MODEL-BASED CLOSED-LOOP GLUCOSE CONTROL IN CRITICAL ILLNESS

by

Timothy D. Knab

B.S., University of Rochester, 2011

Submitted to the Graduate Faculty of
the Swanson School of Engineering in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Timothy D. Knab

It was defended on

February 13, 2017

and approved by

Robert S. Parker, Ph.D., Professor

Department of Chemical and Petroleum Engineering

Gilles Clermont, MD, MSc., Professor

Department of Critical Care Medicine, Department of Chemical and Petroleum Engineering

Joseph J. McCarthy,Ph.D., Professor

Department of Chemical and Petroleum Engineering

Daniel G. Cole, Ph.D., Professor

Department of Mechanical Engineering and Materials Science

Dissertation Director: Robert S. Parker, Ph.D., Professor

Department of Chemical and Petroleum Engineering

MODEL-BASED CLOSED-LOOP GLUCOSE CONTROL IN CRITICAL ILLNESS

Timothy D. Knab, PhD

University of Pittsburgh, 2017

Stress hyperglycemia is a common complication in critically ill patients and is associated with increased mortality and morbidity. Tight glucose control (TGC) has shown promise in reducing mean glucose levels in critically ill patients and may mitigate the harmful repercussions of stress hyperglycemia. Despite the promise of TGC, care must be taken to avoid hypoglycemia, which has been implicated in the failure of some previous clinical attempts at TGC using intensive insulin therapies. In fact, a single hypoglycemic event has been shown to result in worsened patient outcomes.

The nature of tight glucose regulation lends itself to automatic monitoring and control, thereby reducing the burden on clinical staff. A blood glucose target range of 110-130 mg/dL has been identified in the High-Density Intensive Care (HIDENIC) database at the University of Pittsburgh Medical Center (UPMC). A control framework comprised of a zone model predictive controller (zMPC) with moving horizon estimation (MHE) is proposed to maintain euglycemia in critically ill patients. Using continuous glucose monitoring (CGM) the proposed control scheme calculates optimized insulin and glucose infusion to maintain blood glucose concentrations within the target zone.

Results from an observational study employing continuous glucose monitors at UPMC are used to reconstruct blood glucose from noisy CGM data, identify a model of CGM error in critically ill patients, and develop an *in silico* virtual patient cohort. The virtual patient cohort recapitulates expected physiologic trends with respect to insulin sensitivity and glycemic variability. Furthermore, a mechanism is introduced utilizing proportional-integral-derivative (PID) to modulate basal pancreatic insulin secretion rates in virtual patients. The result is virtual patients who behave realistically in simulated oral glucose tolerance tests and insulin tolerance tests and match clinically observed responses.

Finally, *in silico* trials are used to simulate clinical conditions and test the developed control system under realistic conditions. Under normal conditions the control system is able to tightly control glucose concentrations within the target zone while avoiding hypoglycemia. To safely counteract the effect of faulty CGMs a system to detect sensor error and request CGM recalibration is introduced. Simulated *in silico* tests of this system results in accurate detection of excessive error leading to higher quality control and hypoglycemia reduction.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

As my graduate school journey comes to an end and I reflect upon where I have come from I realize what a rewarding and formative experience this has been. A huge part of this journey has been the people who have been involved along the way and this endeavor comes to an end I owe them thanks a debt of gratitude. There are certainly a multitude a multitude of people who deserve thanks here for their support and encouragement along this journey. Without them I wouldn't be where I am and you would no doubt not be reading a preface to this thesis.

First and foremost I owe everything to my parents. They are responsible for encouraging and fostering a lifelong desire to both continually learn and constantly improve. Without them none of this would have been possible. They have supported me every step of this journey from my first interest in science to completion of this PhD and kept me going when I had my doubts. Natasa, I know this adventure hasn't always been easy, dealing with a cranky grad student intent on working science and engineering into every discussion is probably not what you had in mind but you have always there for me and provided much needed support, encouragement and motivation. Thank you for being by my side, always.

Graduate school has been an incredibly gratifying and transformative process. Although it has at times been one of the most challenging periods of my life it has also been one of thost rewarding and I owe much of that to my advisers Bob, and Gilles. Bob, your boundless optimism fortified me on more occasions than I can count and, although I may never admit it again, has rubbed off. Your unrelenting passion for research and learning will continue to influence me far into the future, and although I may not choose a path of academia, I hope I can impart some of that optimism, wisdom and dedication on those I come across in my future endeavours. I also need to thank my co-advisor, Gilles, who did his best to keep me grounded and keep my research relevant to clinical matters. Most importantly, thank you, Gilles, for always reminding me that "perfection is the enemy of great". Without this guiding principle I doubt this would have ever been written.

Finally, I would like to express my gratitude to my labmates and colleages. Those of you who had to put up with my pessimism, sarcasm and cynicism on a daily basis deserve a special thanks. Ari, you were always there for discussion both about shared research interests and the greater world and scientific community. Thang, your guidance as I entered the lab as a new graduate student was invaluable and you provided, on many occasions, much needed advice in both research and life. Li Ang, Matt and Christy, thank you for being great colleagues, and more importantly wonderful friends. Your friendship and camaraderie has been beyond reproach and invaluable throughout this process.

1.0 INTRODUCTION

Stress hyperglycemia is the elevation of blood glucose in response to illness [1, 2] and is prevalent in up to 32.2 % of critically ill patients [3] where it appears to be a marker of disease severity [4]. Every year more than 5.7 million patients are admitted to intensive care units (ICUs) in the United states for treatment. Between 2000 and 2005 alone critical care expenses increased from \$56.6 billion to \$81.7 billion representing 13.4% of hospital costs, 4.1\$ of national health care expenditures and 0.66% of the U.S. GDP [5]. The Society of Critical Care Medicine estimates that savings of up to \$1 billion per quality life year gained can be attained with better management of critically-ill patients [5]. Careful management of stress hyperglycemia may present a route to achieving a significant portion of these savings.

Although hyperglycemia in the ICU has not been consistently shown to be detrimental in patients with preexisting diabetes it is correlated with worse outcomes in patients not known to have diabetes [6]. There is a large body of literature including observational and prospective randomized clinical trials which shows a strong association between hyperglycemia and poor clinical outcomes such as mortality, infections and hospital complications [7–15]. Furthermore, this association appears to be correlated with both the severity of hyperglycemia at the time of admission to the ICU as well as the duration of hyperglycemia during the hospital stay [7, 9, 10]. In addition to the deleterious effects of hyperglycemia there is mounting evidence that glycemic variability may also be negatively associated with patient outcomes [16–19].

The work here is grounded in the hypothesis that with an automated closed-loop system hyperglycemia may be mitigated, hypoglycemia avoided and overall glycemic variability reduced resulting in improved patient outcomes. To this end, a control algorithm utilizing moving horizon estimation and zone model-predictive control is developed to affect zone glucose control (ZGC) through the automated infusion of subcutaneously delivered insulin and intravenous glucose and demonstrated to be effective in *in silico* trials on a “virtual patient” cohort.

1.1 TARGETED GLUCOSE CONTROL AND INTENSIVE INSULIN THERAPY IN CRITICAL CARE

A seminal study conducted in Leuven, Belgium [20] on over 1500 patients implemented a protocol with the goal of tightly controlling blood glucose concentrations in critical care patients between 80-110 $\frac{mg}{dL}$ established the efficacy of reducing hyperglycemia on improved patient outcomes. This study resulted in a significant ($p<0.001$)reduction in mean morning blood glucose levels between the conventional and intensive insulin therapy groups from 173 to 103 $\frac{mg}{dL}$. There was an associated reduction in 12 month mortality in intensive care patients under intensive insulin therapy to 4.6 % compared to 8.0 % for the conventional treatment group ($P<0.04$) (Kaplan-Meier survival curve shown in Figure 1). This study established a new paradigm for insulin administration protocols in critical care units [21]. In 2004 a similar study [22] on 800 critically ill patients found that targeted glucose control (TGC) through intensive insulin therapy reduced mortality by 29.3% and reduced overall length of stay by 10.8% with no significant changes in hypoglycemia.

Despite these initial successes in improving patient outcome with targeted glucose control the results of these studies[20, 22] study have not always been borne out in subsequent studies. The follow-up to the Leuven study found only a reduction in morbidity, but not mortality when attempting targeted glucose control [23] Both the Glucontrol [24] and the CREATE-ECLA [25] found no significant changes in patient outcomes with targeted glucose control. Perhaps the most damning evidence for the limited or non-existent benefits of reducing stress hyperglycemia in critically ill patients resulted from the NICE-SUGAR study [26]. This multi-center study of over 6000 patients reported an increase in mortality within the group receiving intensive insulin therapy as a means to control or prevent stress hyperglycemia. Retrospective analysis of the NICE-SUGAR study [27] has since shown that

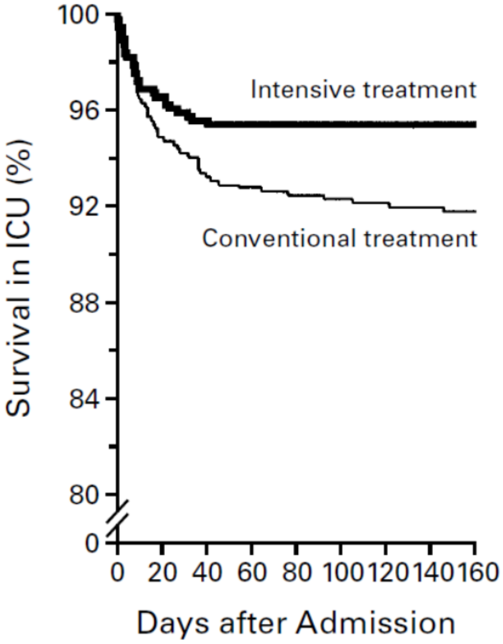


Figure 1: Kaplan-Meier curve showing cumulative survival of patients who received intensive insulin treatment or conventional treatment in the Intensive Care Unit [20])

insulin therapy to combat stress hyperglycemia increases the incidence of moderate and severe hypoglycemia and the risks associated with hypoglycemia in critically ill patients [22, 28] outweigh the potential benefits of targeted glucose control. The VISEP study [29], conducted a year prior, with a much smaller patient cohort corroborated these results and had to be stopped early due to a high incidence of hypoglycemia and associated adverse effects. Due to the inconsistency in outcomes across studies, current clinical guidelines allow for hyperglycemia to avoid the potential for harm associated with hypoglycemia despite the fact that hypoglycemic blood glucose concentrations do not promote optimal patient outcomes.

A review of the literature highlights discrepancies and inconsistencies in protocols for tight glucose control [30] as well as a variety of TGC outcomes dependent on the critically-ill population considered [31–33]. Due to varying reports about the efficacy and safety of TGC there is significant disagreement regarding insulin treatment protocols [34, 35]. An overview of these studies and several additional trials attempting TGC on adult, non-diabetic patients

Table 1: Targeted Glucose Control clinical trials (adapted from [3]). ∗: Study underpowered due to premature discontinuation. Mixed: non-diabetic and diabetic subjects. CABG: Coronary Artery Bypass Graft.

Study	Setting	Population	Clinical Outcome
Malmberg, 1994 [36]	CCU	Mixed	28% decrease in mortality after 1 year
Van den Berghe, 2001 [20]	Surgical ICU	Mixed, with CABG	34% decrease in mortality∗
Krinsley, 2003 [10]	Med-surgical ICU	Mixed	27% decrease in mortality
Van den Berghe, 2006 [23]	Medical ICU	Mixed	18% decrease in morbidity
Gandhi, 2007 [37]	Operating Room	Mixed, cardiac surgery	No difference in mortality; increase in stroke rate in TGC arm
VISEP, 2008 [38]	Medical ICU	Mixed w/ sepsis	No difference in 28- or 90-day mortality, end-organ failure, length of stay ∗
De La Rosa, 2008 [39]	Med-surgical ICU	Mixed	No difference in 28-day mortality
Glucontrol, 2009 [24]	Med-surgical ICU	Mixed	No difference in 28-day mortality∗
NICE-SUGAR 2009/2012 [26, 27]	Med-surgical ICU	Mixed	Increase in 90-day mortality
CGAO-REA, 2014 [40, 41]	Medical ICU	Mixed	No difference in 90-day mortality, increased hypoglycemia in the TGC group
Okabayashi, 2014 [42]	Surgical ICU	Mixed	Decrease in surgical site infections in the TGC group

is given in table 1. In aggregate these studies seem to support the hypothesis that mitigating stress hyperglycemia through insulin therapy has the potential to improve patient outcomes provided hypoglycemia is avoided.

1.2 PATHOPHYSIOLOGY OF STRESS HYPERGLYCEMIA AND OBSTACLES TO GLUCOSE CONTROL

The stress response results in a metabolic shift from an anabolic to a highly catabolic state to provide additional fuel for tissue repair and immune function. Governed by a complex neural-immune-neuroedocrine signal intermediary metabolic pathways are redirected and reenginered to support the shift to catabolism which ultimately provides the necessary amino acids for healing and recovery [43]. A result of stress-induced metabolic changes is increased counterregulatory hormone activity which works to increase hepatic glucose production and decrease peripheral glucose uptake as result of insulin resistance, factors which may be antecedent to stress hyperglycemia.

The insulin signaling cascade is initiated by binding with an insulin receptor (IR) [44] followed by activation of the IRS/PI3K/Akt pathway and the Ras/mitogen-activated protein kinase/extracellular signal-regulated kinase (MEK/ERK) pathway [45]. Decreased activation and tyrosine phosphorylation of IR, and the insulin receptor substrates IRS-1 and IRS-2 has been found to result from extreme stress [46] leading to a loss of fidelity in insulin signal transduction. The loss of insulin-activated phosphorylation of IR, IRS-1, IRS-2 and Akt has been detected in as little as 15 minutes in the liver and within 60 minutes in skeletal muscle following trauma [45] resulting in a decreased sensitivity to insulin.

Systemic glucose balance is maintained by dynamic, minute-to-minute regulation of endogenous glucose production and of glucose utilization by peripheral tissues [47–49]. Glucose production is accomplished by gluconeogenesis or glycogenolysis primarily in the liver and in a lesser degree by the kidneys [3, 49]. The constantly evolving nature of the metabolic landscape in critical illness and the delicate homeostatic balance between glucose production and glucose uptake which may be upset by fluctuations in insulin sensitivity may explain the varying degrees of success in clinical trials of TGC. In light of these data, one thing appears clear, if stress hyperglycemia is to be treated with insulin administration careful monitoring of blood glucose concentration with frequent adjustment to insulin administration is required. A task ideally suited to a closed-loop control system.

1.2.1 Targeted Glucose Control: Ways and Means

The first steps toward glucose control in critically ill patients used a variety of paper-based protocols to guide nurses in adjusting insulin concentrations and infusion rates based on intermittent glucose measurements [50]. Although this approach is simple, cost-effective and can be deployed in hospitals worldwide it is not patient specific and may introduce personal bias if not entirely prescriptive [50]. The next step in ICU glucose control is the use of computerized algorithms or decision support systems.

A number of computer-based decision support systems have been developed including the Stochastic TARgeted (STAR) system [51], the LOGIC-Insulin algorithm [52], an enhanced model-predictive control (eMPC) algorithm [53, 54], the Glucose Regulation for Intensive care Patients (GRIP) system [55], Glucosafe [56], and the Contrôle Glycémique Assisté par Ordinateur (CGAO) system [57]. These systems benefit from mathematical models describing basic biology and therefore a higher degree of physiologic relevance including the ability to adapt to inter- and intrapatient variability and respond to changes in insulin sensitivity. Furthermore, because of the use of models describing glucose dynamics they can respond to trends and act pro-actively rather than take reactive action based on a single glucose measurement as with the paper protocols.

In general, the use of computer-based decision support systems has resulted in reduced error rates in treatment and overall better glucose control [58]. However, despite the outlined successes, these systems suffer from a number of key issues. First and foremost, they still require a clinical effort to obtain glucose measurements and administer treatment. As decision-support systems these computerized algorithms are not automated and require blood glucose measurements at constant intervals, in some cases [51, 56] or at variable algorithm requested times in others [52, 53, 55, 57]. In general these algorithms require a new blood glucose sample approximately every 1-3 hours. One study reported that hourly blood glucose monitoring required nearly 2 hours of direct nursing time per patient per day to achieve TGC [59]. Hooijdonk et al. [60] reported a 1.8% increase in total hospital costs using frequent point-of-care glucose testing. Furthermore, given the reliance of TGC on frequent and accurate glucose measurements to avoid hypoglycemia [61] any non-compliance with the recommendations of these decision support systems by clinical staff may have serious and costly consequences.

Using continuous glucose monitors (GCMs), which automatically measure glucose concentrations at a high frequency, a closed-loop glucose control system becomes feasible and may provide next step in targeted glucose control in critically ill patients. The use of continuous glucose monitors diminishes the burden imposed on clinical staff by frequent bedside glucose testing reducing it to much less frequent sensor calibrations. Furthermore, the automated administration of treatment by a closed-loop algorithm eliminates issues of noncompliance and further assuages the clinical burden.

1.3 CONTINUOUS GLUCOSE MONITORING

A key component of any closed-loop control system is a feedback mechanism which monitors the state of the controlled variable so that adjustments can be made to drive the deviation between the controlled variable and set point or target to zero. Here continuous glucose monitors are used to for high frequency measurements of glucose concentrations. It has been shown by Hanazaki et al. that closed-loop glucose control using continuous monitoring of venous glucose levels can reduce the incidence of hyperglycemia in the critically ill without inducing hypoglycemia [62, 63]. As a point of comparison, a randomized clinical trial using a computerized decision support system (CDSS) with sparse glucose measurements using PID control was unable to significantly decrease the incidence of hyperglycemia compared to the control arm and doubled the incidence of hypoglycemia [57]. This trial had a median time between glucose measurements of 139 minutes. The stark difference in outcomes between the Hanazaki and the CDSS trial can be attributed in large part to the frequency of glucose measurements. Continuous glucose monitoring is fundamental to enabling reliable and effective closed-loop glucose control in critically ill patients.

In a number of additional studies, control algorithms were successful in limiting hypoglycemia but generally were ineffective in controlling glucose to a target[64–66]. Blood glucose values in these samples were typically sampled at an hourly or slower rate necessitating very conservative control leading to inefficacy in reducing hyperglycemia. These results establish high-frequency glucose monitoring as one of the cornerstones enabling reliable and effective closed-loop glucose control in critically ill patients. In the Hanazaki study, glucose was continuously measured from venous blood sampled at a rate of $\frac{2mL}{hr}$. However, continuous venous sampling requires catheterization which carries significantly increased risks of infection and thrombotic complications in critically ill patients [67] and is impractical for routine deployment in a critical care setting. Furthermore, high frequency sampling by clinical staff a poor use of a medical professionals' time and is clearly untenable given the responsibilities of a clinician in a critical care setting.

Commercially available continuous glucose monitoring systems (CGMs) provide a safe, minimally invasive alternative to arterial sampling that requires only nominal clinician effort and oversight. CGMs are generally comprised of a small sensor implanted subcutaneously which generates a signal in the presence (and proportional to the concentration) of interstitial glucose. Typically a signal is generated through electron transfer in the enzyme catalyzed glucose oxidase reaction, an increase in fluorescence or decrease in Förster resonance energy transfer (FRET) of fluorescein labeled dextrin due to preferential glucose binding affinity for concavelin A [68]. The signal from the subcutaneous sensors is processed by a receiving unit that serves to both calibrate the sensors and filter the raw signal.

1.3.0.1 CGM in Critical Care Although the intended patient population for continuous glucose monitors is ambulatory diabetics, CGMs have found their way into critical care units as means to prevent hypoglycemia [61] and identify hyperglycemia [69] as well as investigate the effects of intensive insulin therapy [70].

The accuracy of CGMs in critical or intensive care units has been investigated in a variety of studies. Block et al. [70] reported that in 50 adults admitted to a medical (non-surgical) intensive care (MICU) glucose concentrations recorded by the Glucoday® S CGM correlated with arterial glucose measurements with r=0.85 and P<0.0001 for n=555 with 97% of the data falling within regions A and B on a Clarke Error Grid (See Figure 2 for a description of a Clarke Error Grid).

Another study of CGMs in a critical care setting enrolled 22 patients at the Yale New Haven Hospital[72]. Patients were implanted with the Continuous Glucose Monitoring System® by Medtronic Minimed. In total, 41 sensors (1-5 devices per patient) yielded 546 sensor-reference measurement pairs resulting in a mean absolute deviation (MAD) of 19.7 ± 18.3 $\frac{mg}{dL}$. GCM values correlated to fingerstick measurements with r=0.88 and 98.7% of the CGM values fell within the clinically acceptable A and B regions of the Clarke Error Grid.

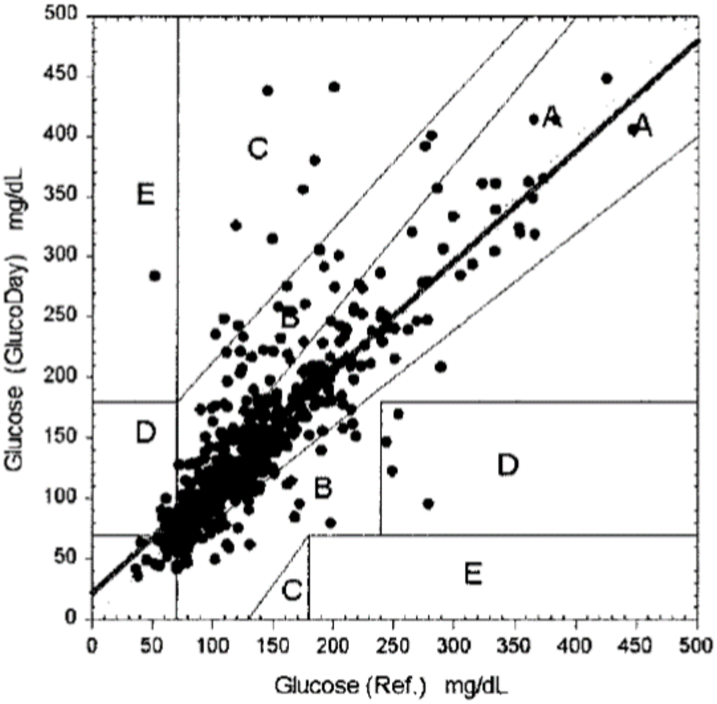


Figure 2: Example Clarke Error Grid for GlucoDay CGM in MICU patients (reproduced from [71]). Region A represents values that are within 20% of the reference measurement. Region B contains points that are outside of the 20% region but would not lead to inappropriate treatment. Region C contains points that would lead to unnecessary treatment. Region D represents a potentially dangerous failure to detect hyper- or hypoglycemia. Region E corresponds to a region where hyperglycemia could be detected as hypoglycemia and vice-versa. [71]

A separate study of continuous glucose monitoring in the Surgical Intensive Care Unit (SICU) at the Yale New Haven Hospital enrolled 25 patients to be implanted with a Guardian REAL-Time Continuous Glucose Monitoring System from Medtronic Minimed [73]. This study aimed to determine if CGM resulted in an acceptable accuracy for clinical use in an intensive care unit and found CGMS values to correlate with blood glucose fingerstick values with r=0.61 (p<0.001) with a MAD of 16.0 $\frac{mg}{dL}$. A Clarke Error Grid analysis showed that 98.92% of the points were within zone A or B. The authors of this study concluded that CGMS in the SICU were reasonably accurate despite the widespread use of pressors and large-volume resuscitation and state that further work investigating the use of continuous glucose monitoring to enable targeted glucose control with intensive insulin therapy is warranted.

A different 50 patient study of the Clinical Glucose Monitoring System® (CGMS) Gold™ resulted in 736 CGM-reference blood glucose pairs resulting in a correlation coefficient of 0.93 between CGM sensed blood glucose and reference arterial blood glucose [74]. According to regulations set by the International Organization for Standardization (ISO), 94.2% of the data points from this study can be deemed accurate. This study further established that CGMs in a clinical setting are unaffected by shock and the subsequent required administration of norepinephrine.

Through a number of studies in both medical and surgical intensive care units, it has been shown that continuous glucose monitoring achieves a reasonably high accuracy with approximately 94-99% of the CGM measurements falling within zones A and B on a Clarke Error Grid. This suggests that the error inherent to these sensors will not lead to insulin administration based on traditional scales that will lead to patient harm in a critical or intensive care unit. Despite the apparent accuracy of CGMs in critical care settings, the results and statistical outcomes of the aforementioned studies are best viewed somewhat more critically. The Clarke Error analyses presented in these studies is predicated on a traditional schedule or scale of insulin administration, however due to the ability of an automated control algorithm to deliver insulin at a much higher frequency based on CGM measurements, chronically biased CGM readings and large errors could result in glucose or insulin administration that would drive patient blood glucose levels to unsafe levels. Ultimately, the control of glycemia is intimately related to the performance of the continuous glucose monitors, the frequency of measurement and the control algorithm or treatment protocol [75].

A common measure used to assess the accuracy of continuous glucose monitoring is mean absolute relative deviation (MARD) [76, 77] (equation (1.1)), where $G(t_i)$ is true blood glucose at time t_i and $\hat{G_{t_i}}$ is blood glucose as measured by CGM at time t_i.

$$\text{MARD} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{G}(t_i) - G(t_i)|}{G(t_i)} \tag{1.1}$$

A recent study by Wilinska and Hovorka [78] suggested that MARD $\leq$ 11% is optimal for control and resulted in the lowest frequency of hypoglycemia. This result was corroborated by a similar study [79] which found that MARD $\leq$ 10% was optimal. The conclusion to be drawn from these studies is that the high frequency acquisition of glucose data via a CGM system carries a definitive advantage for glycemic control compared to low frequency glucose monitoring techniques, such as capillary fingerstick, even though the measurement quality of CGM systems is inferior to intermittent monitoring [80].

An analysis of unadulterated CGM data from an observational study at UPMC results in a MARD of 15.15%, greater than the maximum acceptable values established by Boyd and Bruns, Wilinska and Hovorka and Van Herpe[78–80]. The large MARD observed in our clinical dataset is likely due to improper calibration, contraindicated medications and pressure-induced loss of sensitivity as a result of rolling onto or sleeping on a sensor [81]. In light of these studies and a preliminary analysis of our clinical dataset, it is necessary to ensure that, through a combination of multiple CGMs and state estimation, estimates of blood glucose serving as the basis for control action in the proposed control algorithm meets a minimum criterion of MARD $\leq$ 10%.

1.4 AUTOMATED GLUCOSE CONTROL

1.4.1 In Type 1 Diabetes

Although the work here is focused on treating stress hyperglycemia in critically ill patients, the use and study of automated glucose control in Type 1 Diabetes (T1D) predates the study of TGC in critical care and continues to work toward the development of an artificial pancreas

(AP), enabled largely by the development and advancement of CGM technology. Initial work on APs began in the '70s by Albisser et al.[82], Pfeiffer et al.[83], and Mirouze et al.[84] culminating in the development of the first commercial closed-loop beside glucose control device, the Biostator[85]. The heart of any closed-loop glucose control system is the control scheme and APs over the years have employed a variety of algorithms. Control algorithms that have shown promise over the years include proportional-integrative-derivative (PID) control [86, 87], fuzzy-logic[88–90], bio-inspired controllers [91] which mimic β-cell insulin secretion in response to glucose and notably, model-predictive control (MPC) [92–97].

MPC for AP usage has been studied in both linear[92] nonlinear [93] formulations and have been shown [98] to be effective in maintaining euglycemia and decreasing hypoglycemia (particularly overnight where conventional self-monitoring is difficult). MPC's have also exhibited success in a bihormonal systems where the MPC is responsible for insulin infusion and a PID controller controls glucagon infusion [99]. The option for glucagon infusions to counteract the effects of insulin allows for better control by enabling control of blood glucose in both directions.

1.4.2 In Critical Care:

Drawing from work in the development of an artificial pancreas several studies to date have explored the efficacy and safety of closed-loop glucose control systems utilizing continuous glucose monitoring in critically-ill patients. In 2003 a small five subject trial was conducted to determine if continuous glucose monitor could be used in real-time with a closed-loop system to control blood glucose concentrations [100]. The system used subcutaneous CGMs with a proportional-integral (PI) control algorithm based on a sliding scale approach which automatically infused intravenous insulin to patients. Error grid analysis of the CGM data indicated that just 64.6% of sensor readings were accurate compared to measurements by a highly accurate glucometer with blood drawn from an indwelling arterial cannula. That is CGM measurements exhibited $\leq$ 20% deviation from the arterial line measurements [100]. Of the five patients who underwent closed-loop control only one patient did not require manual intervention due to sensor deviations $\geq$ 20% from glucometer measurements. Despite issues of CGM error Chee et al. found that the closed-loop performance was comparable to manual TGC, although it is difficult to ascertain the extent to which manual intervention affected closed-loop outcomes.

A 2011 study utilized the STG-22 (Nikkiso, Tokyo, Japan), an artificial pancreas system, to attempt TGC with a glucose target of $90 - 110\frac{mg}{dL}$[101]. In this study the STG-22 infused insulin continuously if blood glucose concentrations were $\geq 110\frac{mg}{dL}$ and a 10% glucose solution $BG \leq 90\frac{mg}{dL}$[101]. Infusion rates were calculated by the STG-22 using a PID control algorithm [102]. In 208 ICU patients the STG-22 was able to maintain glucose concentrations within the target zone for 50% of the study period with no hypoglycemia (defined by Yatabe et al. as $BG \leq 70\frac{mg}{dL}$[101]).

In a further trial using the STG-22 closed-loop glucose control system 450 patients in a Japanese intensive care unit were treated following surgery for hepato-biliary-pancreatic disease [42]. The patients were randomly assigned to either an intensive treatment group with a glucose target range of $79.29 - 109.90\frac{mg}{dL}$ or an intermediate treatment group with a target zone between $138.72\frac{mg}{dL}$ and $194.57\frac{mg}{dL}$. Under closed-loop control neither group experienced any hypoglycemia and the intensive therapy group saw a decrease in surgical site infections which may be a component of the improved outcomes observed in successful TGC.

In 2013 a study [103] was conducted in which twenty-four critical ill adults were randomized into two groups. One group received fully automated closed-loop intensive insulin therapy with a nonlinear model-predictive controller (nMPC) directing the intravenous delivery of insulin and a 20% dextrose solution using a subcutaneous CGM to monitor glucose. The other group was treated with a local protocol utilizing intravenous sliding-scale insulin. The target blood glucose concentration for the nMPC arm was $108 - 144\frac{mg}{dL}$. With each new CGM blood glucose measurement the nMPC updates parameters characterizing a rapidly changing glucose flux to correct for errors in model based predictions, and a slowly changing estimate of a basal endogenous insulin production rate required to maintain euglycemia [103]. In the experimental closed-loop treatment arm nMPC resulted in an increased time within the target band (54.3% with a range of 44.1% to 72.8%) compared to the conventional group treated with intravenous sliding-scale insulin (18.5% with a range of 0.1% to 39.9%).

Furthermore, the nMPC arm was associated with decreased mean glucose concentrations ($140\frac{mg}{dL}$) compared to the conventional arm ($164\frac{mg}{dL}$). No hypoglycemic events were reported in either treatment arm.

Despite the general success of these closed-loop glucose controllers in achieving TGC, there are a number of drawbacks to these methods. The PI or PID based control algorithms have no basis in physiology or ability to adapt to different patient phenotypes and may not be suitable for the wide range both inter- and intrapatient variability expected in a critical care setting. The trials utilizing the STG-22 system rely on a dual-lumen catheter to draw and return blood at a rate of $2\frac{mL}{hr}$ for blood glucose measurements. The use of vascular catheters carry an increased risk of foreign body reactions which may lead to thrombus formation and catheter occlusion necessitating catheter replacement [104]. Vascular catheterization also carries risk of infection [105], sepsis [106] and hemorrhaging [107]. Furthermore, the closed-loop glucose control systems used in all the trials discussed above utilize intravenous insulin infusion to affect TGC, which carries the same risks as vascular catheterization for glucose monitoring. The decision to utilize intravenous delivery likely stems from reasons of simplicity. Compared to diabetic patients who utilize subcutaneous insulin, critically ill patients may present a higher variability in insulin requirements and the use of intravenous insulin eliminates adsorption delays associated with subcutaneous insulin delivery [50].

In addition to the potential issues associated with vascular catheterization and PID control, the 2003 study by Chee et al. [100] demonstrated the potential pitfalls of CGMs in an intensive care setting where they may fail to meet the accuracy requirements for safe TGC. The 2013 study utilizing model-predictive control by Leelarathna et al. [103] relied on a single subcutaneous glucose monitor inserted either in the anterior abdominal wall or the upper arm. This represents a single point of failure and poor calibration techniques or a faulty CGM could derail control and result in dangerous hypo- or hyperglycemia. As a preventative or safety measure the MPC algorithm in this study requests reference arterial blood glucose measurements made using a blood gas analyzer every 1 to 6 hours. In practice arterial samples were drawn every hour which represents a significant clinical burden or workload.

The closed-loop glucose control scheme developed here (see fig. 3 for an overview) utilizes dual CGMs inserted in the abdominal wall to provide glucose concentration measurements. Moving horizon estimation is used to update patient specific parameters based on glucose measurements and a model-predictive controller optimizes glucose and insulin infusion rates to maintain blood glucose concentrations within a target zone. The key advantages of this arrangement over PID controllers is the ability to account for intra- and interpatient variability and the capacity to identify complex glucose trends and take proactive corrective measures to prevent hypo- and hyperglycemia. Compared to clinically tested model-based methods [103] the use of dual or redundant CGMs mitigates the risk of CGM error, and through sensor fusion in the controller results in a MARD which is sufficient for safe TGC with less frequent reference measurements and calibrations required. The enhanced accuracy of fused measurements from dual CGMs also enables more accurate estimates of patient-specific parameters which improves theoretically achievable control quality [108]. Furthermore the use of multiple CGMs provides a "built-in" reference that enables the identification of CGM faults.

The basis for the internal model used in the closed-loop glucose control algorithm developed here was developed specifically to describe the glucose-insulin trajectories of critically ill patients [110], including saturating nonlinear insulin dynamics. This is in contrast to the model used in the MPC developed by Leelarathna et al. [103] as well as in a semi-closed-loop MPC trials which required manual glucose draws for measurement and manual insulin administration [111]. The controller model employed in [103] and [111] is developed to describe glucose-insulin dynamics following an intravenous glucose tolerance test (IVGTT) and was identified using data from IVGTTs over only a few hours. Owing to the original model purpose and the relatively short time scale over which the parameters are identified the model does not contain a mechanism for the saturating effect of insulin on glucose uptake [112, 113] which may result in significant model-patient mismatch at longer timescales with elevated rates of insulin infusion and affect the ability of the algorithm to maintain glucose control.

This model is supplemented with a validated linear model describing the adsorption kinetics of subcutaneously injected insulin [114]. As a result the glucose controller developed here can accurately predict the rate of insulin appearance in plasma following subcutaneous administration and adsorption to optimize changes in subcutaneous infusion rates to maintain blood glucose concentrations within the target zone. This confers the advantages of continuous subcutaneous insulin infusion which are largely the elimination of complications associated with an indwelling intravenous delivery mechanism. Moreover, in contrast to regular insulin infusion, with subcutaneous insulin an increase in delivered dose as a bolus or spike in infusion rates does not alter the time to peak and duration of action [115]. Additionally, CSII can reduce postmeal glucose spikes, decrease the incidence of severe hypoglycemia [116, 117] and is consistent with the preference method for insulin delivery for hospitalized patients in the U.S. [118]. Finally, the use of subcutaneous insulin and CSII has been associated with decreased glycemic variability [117, 119, 120] which has beneficial implications for patient outcomes [16–19].

Like existing closed-loop systems, the glucose control system developed here allows for the infusion of intravenous glucose. This is a key component of this system as it provides a rapid "rescue" mechanism to avoid hypoglycemia. The safety margin and protection against hypoglycemia that the availability of glucose infusion provides enables more aggressive control to mitigate hyperglycemia.

1.5 GLUCOSE TARGETS IN CRITICALLY ILL PATIENTS

The benefits of hyperglycemia reduction in critically ill patients, balanced with a necessary avoidance of hypoglycemia implies the existence of an ideal target blood glucose range for critically ill patients associated with reduced mortality and improved patient outcomes. Data in the High-Density Intensive Care (HIDENIC) database from the University of Pittsburgh Medical Center (UPMC) was compiled in Figure 4 and suggests that maintaining blood glucose levels between 90 and 110 $\frac{mg}{dL}$ minimizes patient mortality. The proposed target range is 110-130 $\frac{mg}{dL}$ where the lower bound is chosen as it provides a comfortable buffer against hypoglycemia and the steeply increasing mortality rates associated with even mild hypoglycemia.

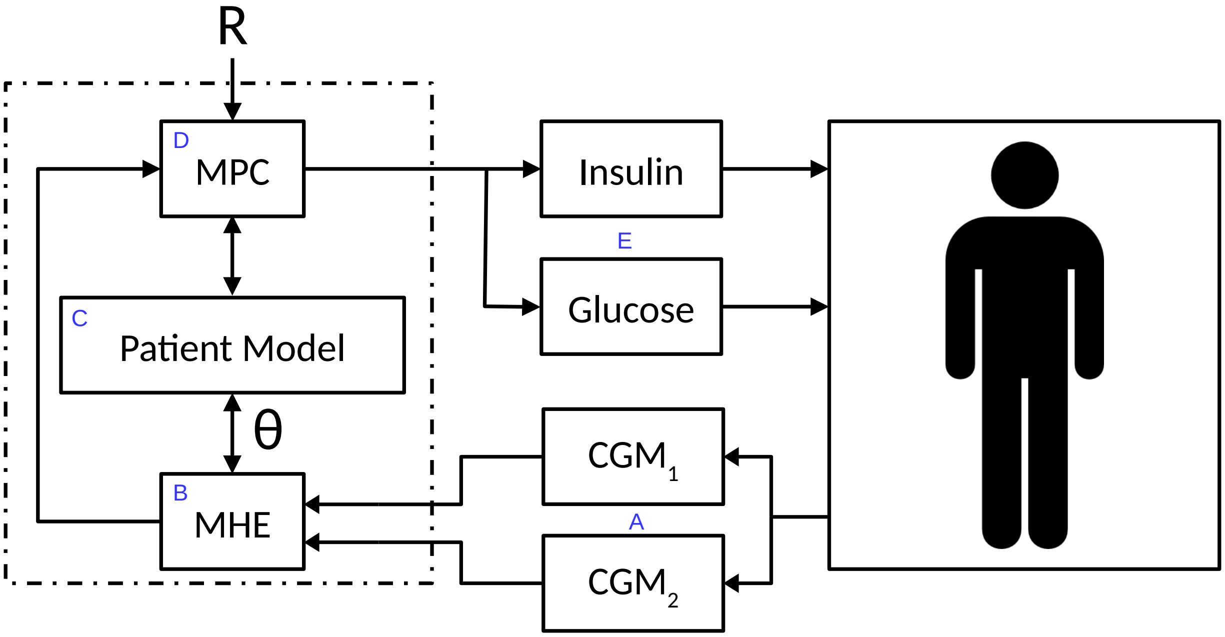


Figure 3: Diagram of the closed-loop glucose control scheme developed here. Dual CGMs (A) measure patient glucose concentrations which are provided to the moving horizon estimator (B). The MHE fuses the CGM measurements and estimates a more accurate blood glucose concentration, insulin concentrations, insulin sensitivity, insulin production and glucose production. These patient-specific estimates and parameters are passed to the MPC (D) which optimizes a series of subcutaneous insulin and intravenous glucose infusion changes (E) to maintain predicted glucose concentrations within a target zone over a future horizon. The first of these infusion rate changes is implemented by infusion pumps and the cycle repeats. The dashed box taken as a whole is the control algorithm developed here. The MHE and MPC used in this work are both linear as the linear estimation and control problems can be formulated as a constrained quadratic program which are well studied and a large number of software tools are available for the efficient solution of constrained quadratic programs. Moreover, the MPC is a convex quadratic problem which implies that any local minimum is also a global minimum thereby guaranteeing the optimality of control action [109]. Furthermore, the highly structured nature of linear MPC means that a generic control algorithm can be developed which has use in problems outside of the work presented here. Finally, a linear control scheme may be more amenable to regulatory bodies, such as the FDA owing to the well-studied nature of the field and the multitude of mathematical techniques developed for the analysis of linear-time-invariant systems.

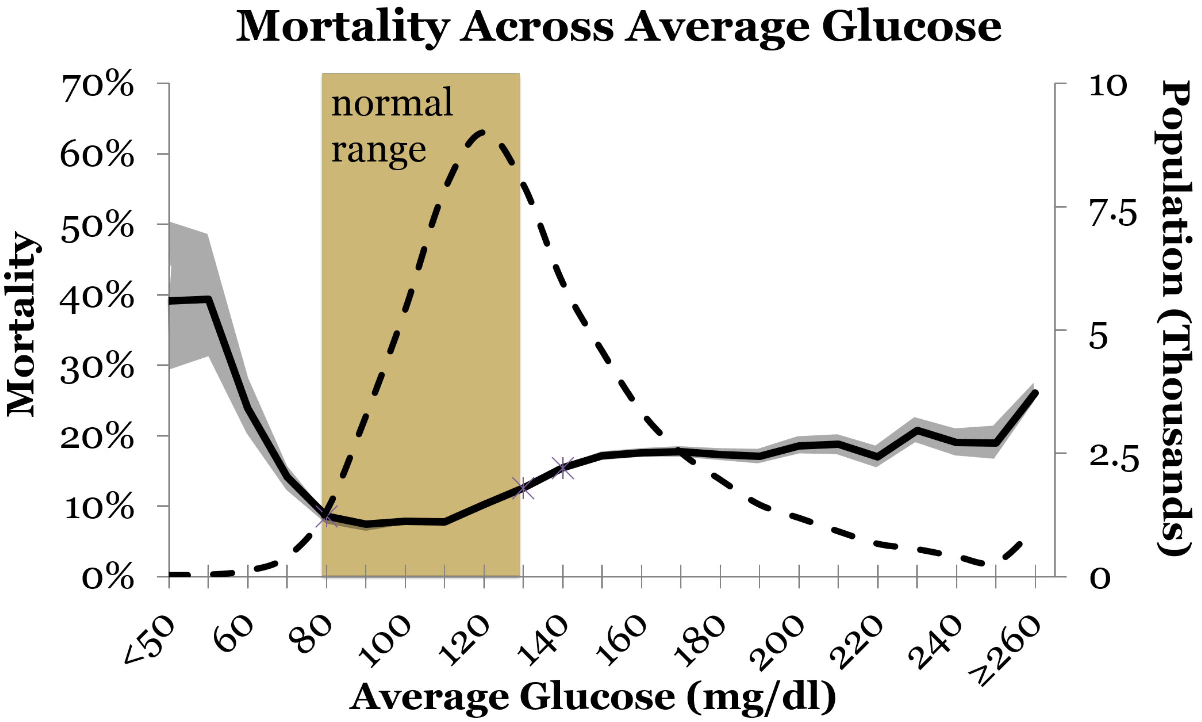


Figure 4: Increased mortality is associated with both hyper- and hypoglycemia in critical care patients (-). Mean blood glucose over the duration of stay shows that the critical care population (- -) tends to present with hyperglycemia [121]. The "bathtub" shaped region between 80 and 130 $\frac{mg}{dL}$ shows a clear minimum in mortality which is centered around lower glucose concentrations than the critical care population mean.

1.6 VIRTUAL PATIENTS AND *IN SILICO* TRIALS

in silico trials utilizing virtual patients offer the ability to design and test protocols and algorithms to optimize performance and safety without risk to real patients [122]. The development of virtual patients and simulated trials have a long history in glucose control, both for diabetic patients [123–125] and critically ill patients [52, 78, 122, 126–129] and here a novel virtual patient cohort is developed which is used for *in silico* trials of the ZGC algorithm. Through simulated trials using this virtual patient cohort the controller is iteratively tuned mitigate hyperglycemia while avoiding hypoglycemia even in the face of CGM inaccuracies. Furthermore, the virtual patient cohort is used to inform controller design decisions by exploring advantages and disadvantages of announced meals. Finally, the virtual patient cohort can be used to test error and fault remediation strategies and provide assurances of safety in the face of CGM or insulin infusion set failure.

The novel virtual patient here is based on the model developed by Lin et al. [110] which has been previously validated in a critically ill patient population. This model is extended to account for the effects of insulin suppression on endogenous glucose production [130] and calibrated to high-frequency clinical data from 17 critically ill patients via a unique approach utilizing a number of heuristic regularizations and physiologic constraints. A realistic pancreatic response is then introduced via a proportional-integral-derivative control mechanism which modulates insulin secretion rates in response to plasma glucose concentrations. The result is a set of physiologically realistic and clinically relevant virtual patients which capture a variety of glucose profiles, exhibit a range insulin sensitivity phenotypes and respond realistically to clinical treatments as tested via simulated oral glucose tolerance tests and insulin tolerance tests.

1.6.1 Simulation Outcomes and Glucose Control Metrics

Through *in silico* studies it is trivial to calculate any number of statistics, indices and metrics related to glucose and insulin dynamics, however it is important to identify and utilize metrics which maintain relevance to measurable clinical outcomes. An ideal index of the quality of glucose control should, according to Preiser et al. [50]:

- be measurable and assessable over time and retrospectively
- be linked to clinical outcomes
- allow robust statistical comparison between studies and paradigms
- be calculated and calculable from regular interpolated measurements
- be equally applicable to individuals and patient cohorts
- relate to known physiological effects
- be easy to understand and interpret

To this end several metrics are used throughout to quantify the quality of glucose in *in silico* trials. The first and perhaps most obvious is mean blood glucose concentrations over the course of the simulated trial. With a goal of reducing hyperglycemia the expectation is that successful control will present with a reduction in average glucose concentrations. Similarly, as a result of the negative association of blood glucose variability with patient outcomes a reasonable expectation of quality glucose control would be a decrease in the variance or standard deviation of blood glucose concentrations over the course of a simulated trial. To quickly and efficiently quantify the severity of any incident hyper- or hypoglycemia minimum and maximum glucose concentrations during *in silico* trials are reported.

In addition to these simple statistical indices time(s) in relevant bands or zones has been widely used as a potentially useful metric of glucose control quality [50, 131]. Here, percentages of total time within certain bands are used to assess the quality of glucose control: The bands used throughout this work are:

- the target zone ($110 \leq BG \leq 130\frac{mg}{dL}$)
- below the target zone ($BG \leq 110\frac{mg}{dL}$)
- above the target zone ($BG \geq 130\frac{mg}{dL}$)
- hypoglycemia ($BG \leq 74\frac{mg}{dL}$)
- hyperglycemia ($BG \geq 180\frac{mg}{dL}$)

Time in band metrics have an increasing accuracy with higher measurement frequency and are thus ideal for *in silico* simulations where near continuous glucose values are available. The clinical data used here is also available at a high frequency with a sampling rate of 5 minutes which further supports the use of time in band metrics for comparison between *in silico* trials and clinical practice.

1.7 OVERVIEW OF THIS THESIS

Chapter 2 provides a method to reconcile differences between measurements from multiple CGMs by removing bias and drift to reconstruct an accurate blood glucose trajectory. Using the newly generated glucose data from the reconciled sensors a model of CGM error is constructed for both nominally functioning CGMs with a low level of noise and CGMs deemed to be malfunctioning with a much greater degree of error. The error models are able to accurately regenerate, to second order statistics, the observed error in stochastic simulations and suggest that the primary driver of CGM error is improper calibration

In chapter 3 the reconstructed glucose data is used to generate a virtual patient cohort through optimization of patient specific parameters including a time-varying insulin sensitivity profile and rate of endogenous glucose production. The insulin sensitivity profiles show two distinct patient phenotypes which have been described as clinically stable and clinically unstable. Regressed endogenous insulin production trajectories exhibit ultradian rhythms consistent with physiology and overall insulin sensitivity and blood glucose trends are congruent with clinical expectations.

Chapter 4 sees the introduction of a feedback mechanism for pancreatic insulin secretion dependent on plasma glucose concentrations. To maximize the realism and physiologic relevance of the virtual patient cohort a PID-based mechanism for the control of basal insulin secretion is fit to patient-specific endogenous insulin secretion profiles. The introduction of this feedback mechanism is shown to result in patient responses to oral glucose tolerance tests and insulin tolerance tests which accurately recapitulate the dynamics observed in human experiments.

Chapter 5 details the formulation for a linear moving horizon estimator and zone model predictive controller. The MHE and zMPC developed within are generic, and can be used for the control of any process described by a LTI state-space model. A description of the linearized internal models including linearization points is given and controller tuning parameters provided. In this chapter it is shown in *in silico* trials that dual CGMs, even with realistic noise provide measurements which following sensor fusion using the MHE result in a MARD which is acceptable for safe glucose control in a clinical setting. Examples of controller action in *in silico* trials on individual patients and the virtual patient cohort are provided and it is shown that the MHE/zMPC control scheme is able to successfully control glucose to a target zone even when simulated with realistic CGM sensor noise. The use of announced or unannounced patient meals in the MPC is explored and finally the results of the ZGC system developed here is compared to commercially available solutions.

In Chapter 6 the detection of CGM faults and excessive error is discussed and a system to detect such failures is developed. The system is shown via *in silico* trials to mitigate the effects of improper CGM calibration and ensure patient safety. An additional safety mechanism to detect insulin infusion set failures is developed using a feature developed with the MHE that allows for disturbance estimation. Finally, a clinical interface or information system is detailed which provides a simple color-based display of the controller's ability to maintain glucose control over an approximately 3 hour horizon.

In addition to demonstrating that a linear MHE/zMPC is effective in treating stress hyperglycemia by reducing blood glucose levels and variability while avoiding hypoglycemia in *in silico* trials this work has resulted in a comprehensive python package for linear estimation and model-based control of LTI systems with an extensive feature set which, to our knowledge, has not been available to this point. A novel virtual patient cohort with a high degree of physiologic realism and clinical relevance is also developed and can be used in the future to test further control schemes and paradigms.

1.8 NOMENCLATURE

Abbreviations

TGC	Tight Glucose Control
IIT	Intensive Insulin Therapy
ZGC	Zone Glucose Control
(z)MPC	(Zone) Model Predictive Control
MHE	Moving Horizon Estimator/Estimation
CGM	Continuous Glucose Monitor/Monitoring
SQ	Subcutaneous
MARD	Mean Absolute Relative Deviation
VP	Virtual Patient
PID	Proportiona-Integral-Derivative Control
ICING	Intensive Care Insulin Nutrition Glucose Model
eICING	Extended ICING Model
(e)ICING + SQ	(Extended) ICING Model with Subcutaneous Insulin Absorption Kinetics

Modeling

BG	Blood Glucose Concentration	$\frac{mg}{dL}$
Q_I	Effective Interstitial Insulin Concentration	$\frac{mU}{L}$
I	Plasma Insulin Concentration	$\frac{mU}{L}$
P_1	Mass of glucose in the stomach	mg
P_2	Mass of glucose in the gut	mg
$Q_{SC,r/f}$	Subcutaneous insulin (regular/fast-acting) mass in the interstitium	mU
$Q_{1,r/f}$	1st intermediate SQ insulin compartment (regular/fast-acting)	mU
$Q_{2,r/f}$	2nd intermediate SQ insulin compartment(regular/fast-acting)	mU
Q_L	Insulin concentration in the liver	$\frac{mU}{L}$
G_{ISF}	Interstitial Glucose Concentration	$\frac{mg}{dL}$

EGP	Endogenous Glucose Production	$\frac{mg}{min}$
S_I	Insulin Sensitivity	$\frac{L}{mU \cdot min}$
U_{en}	Endogenous Insulin Production/Pancreatic Insulin Secretion	$\frac{mU}{min}$
$U_{SC,f/r/s}$	Rate of SQ insulin infusion (fast/regular/slow)	$\frac{mU}{min}$
U_I	Rate of IV insulin infusion	$\frac{mU}{min}$
U_G	Rate of IV glucose infusion	$\frac{mg}{min}$
U_E	Rate of Enteral Feeding	$\frac{mg}{min}$
I_{SS}	Steady State Plasma Insulin Concentration	$\frac{mU}{L}$
$Q_{I_{SS}}$	Steady State Interstitial Insulin Concentration	$\frac{mU}{L}$
$U_{en,b}$	Basal rate of insulin production	$\frac{mU}{min}$
$\underline{x}(t)$	Vector of model states at time t	
A, B, C, D etc.	Matrices	

2.0 CONTINUOUS GLUCOSE MONITORING

2.1 INTRODUCTION

Continuous glucose monitors (CGM) were originally developed for use in diabetic patients to provide doctors with a continuous record of glucose measures. Coupled with records of patient meals, exercise and medicine this enabled a detailed understanding of patients glucose dynamics, as well as insulin sensitivity and insulin requirements. Needle-based subcutaneous continuous glucose monitoring systems were introduced in the early 2000s and currently form the basis for the majority of commercially available CGM devices [132, 133]. In numerous studies, the use of continuous glucose monitoring has been shown to result in significant improvements in HbA1c or glycated hemoglobin, which reflects long-term average blood glucose levels [134–139]

Despite initial approval for use in type 1 and type 2 diabetics, and primarily as an adjunct to traditional self-monitoring of blood glucose levels (although recent FDA approval of a Medtronic device opens the door to closed-loop control [140]), continuous glucose monitoring has found its way into intensive care units as a means to alert to, and potentially forewarn of, dangerous hyper- and hypoglycemia. The impact on patient outcomes from continuous glucose monitoring in a clinical setting has been evaluated in multiple studies and found to be generally favorable [72, 76, 77, 141, 142]. However, there remains some debate over the efficacy of CGM in a critical care setting, as well as over the accuracy requirements for the benefits of CGM to be realized for critically ill-patients.

In ambulatory diabetics, continuous glucose monitors are known to suffer from drift, be subject to the effects of improper or erroneous calibration, and exhibit significant fluctuations in sensor gain in response to pressure [143, 144] and certain medications such as acetaminophen [145]. We would expect these effects to be more pronounced in non-ambulatory patients in critical care units, especially considering the highly variable nature of blood glucose levels in this patient population. As a result, there is a need to evaluate the error and noise characteristics of a continuous glucose monitor in a critically-ill patient populations and develop a model of CGM error because, any closed-loop glucose control system Chapter 5 will have to demonstrate efficacy and safety in the face of this error.

2.2 RETROSPECTIVE CGM CORRECTION

There is a reasonable expectation that two continuous glucose monitors placed subcutaneously in each patient would record glucose measurements with little difference between the sensors. In practice, however, there exist varying degrees of inconsistency between the sensors as a result of improper calibration, poorly chosen calibration times, pressure induced losses of sensitivity, non-specific CGM drift, and random noise (see Figure 5). YSI® analyzer blood glucose measurements, and to a lesser degree, capillary fingersticks, are considered to be significantly more accurate than the CGM measurements, but are available at a greatly reduced frequency compared to the CGM. The reference measurements collected from the study patients, when available, are taken as a basis for retrospective correction of the individual CGM signals so that they are consistent with glucose measurements of known accuracy.

2.2.1 Materials and Methods

2.2.1.1 Clinical Data and Medical Records Twenty four Patients enrolled in an IRB-approved study at the University of Pittsburgh Medical Center (UPMC) each received two Dexcom® Platinum™ G4 continuous glucose monitors, one on each side of the abdomen

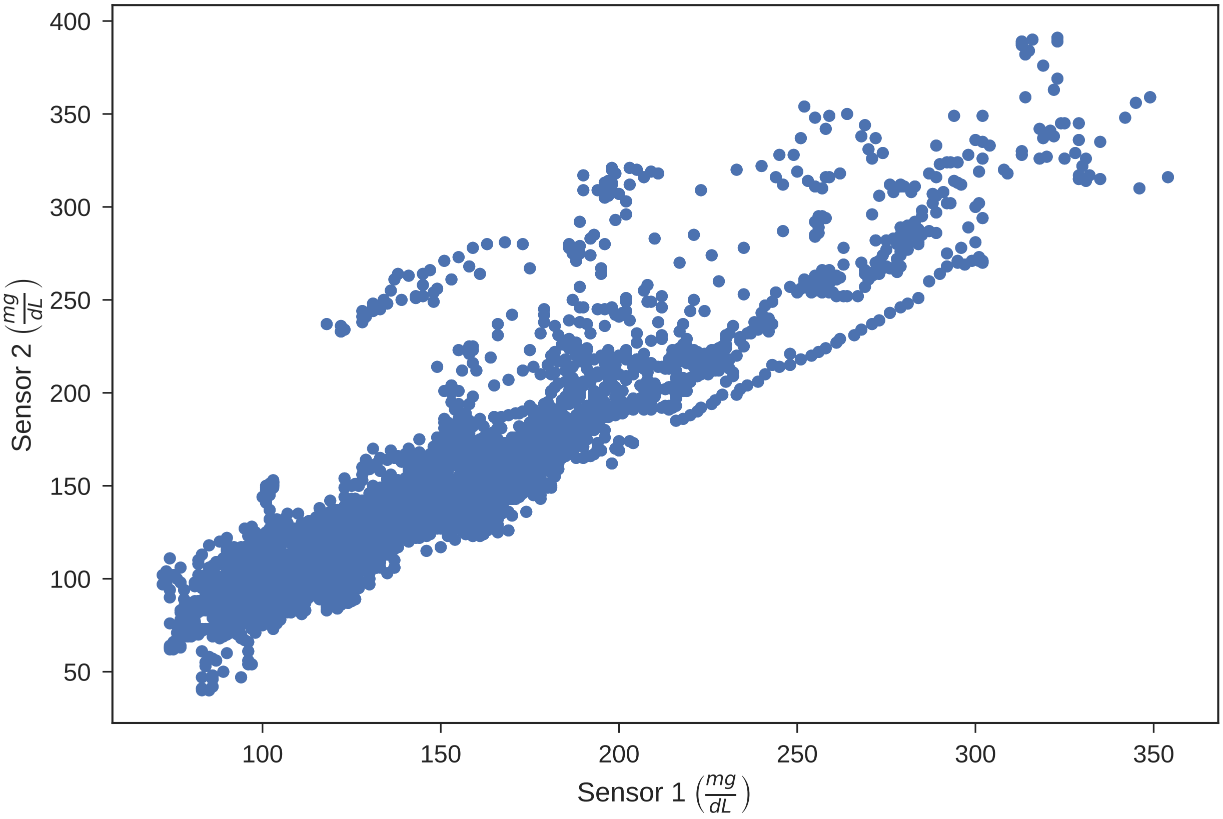


Figure 5: Sensor 1 vs. Sensor 2 across N=17 patients with a total of 641.5 hours of measurement. The average per patient time of measurement is 37.7 hours, with a minimum of 24.2 hours and a maximum of 49.8 hours. There appears to be a linear correlation described by the line $y = x$, as would be expected for two sensors ostensibly measuring the same blood glucose level. However, significant variability around $y = x$ is observed, indicating that the CGMs are not always in close agreement and therefore some amount of error is inherent in at least one, but possibly both, CGMs.

(henceforth referred to as CGM$_1$ and CGM$_2$). These sensors recorded subcutaneous glucose levels every five minutes over a period not less than 12 hours. 17 patients were available for analysis; 7 patients were omitted due to missing data and sensor-dropout. Clinical staff recalibrated the CGMs using reference capillary fingerstick measurements approximately every 6 hours. These recalibration measurements were available with the CGM data. Additional reference blood glucose measurements were collected via capillary fingerstick and YSI® analyzer and were recorded in the UPMC electronic medical records.

2.2.1.2 Dynamically Weighted Linear Combination of CGM Measurements A simple and intuitive approach to retrospective sensor correction is based on the assumption that overall a "true" or accurate series of glucose measurements by the CGMs should be representable as a linear combination of the measurements from the individual sensors as shown in Equation (2.1).

$$BG_{true}(t_i) = \lambda_1(t_i)CGM_1(t_1) + \lambda_2(t_i)CGM_2(t_2) \qquad \forall i = 1...N_R \tag{2.1}$$

In this approach, the weights, $\lambda_1(t_i)$ and $\lambda_2(t_i)$, for CGM$_1$ and CGM$_2$, respectively, are dynamic and may take different values for a set of glucose measurements, from both fingerstick reference and the continuous glucose monitors, at time t. Here, $(G(t))$, defined for a particular reference measurement (from 1 to N_R reference measurements) as

$$G_i \equiv \{BG_{ref}(t_i), CGM_a(t_a), CGM_b(t_b)\}$$

where BG_{ref} is the value of the i-th reference measurement which was taken at time t_i. $CGM_1(t_1)$ and $CGM_2(t_2)$ are the CGM values from each CGM recorded at times t_1 and t_2. t_1 and t_2 are the measurement times from each CGM data set closest to the reference time.

A simple quadratic program (QP) for determining the weights $\lambda_1(t_1)$ and $\lambda_2(t_2)$, via minimization of the sum squared error between the reconstruction and reference measurements is shown in Equation (2.2a).

$$\underset{\lambda_{1,i},\lambda_{2,i}}{\text{minimize}} \quad J = \sum_{i=1}^{N_R} \left(BG_{ref}(t_i) - [\lambda_1(t_i)CGM_1(t_1) + \lambda_2(t_i)CGM_2(t_2)]\right)^2 \tag{2.2a}$$

subject to:

$$\lambda_{1,2}(t_i) \geq 0 \quad \forall i = 1...N_R \tag{2.2b}$$

In Equation (2.3b), every reference value is associated with two variables, $\lambda_1(t_1)$ and $\lambda_2(t_2)$ resulting in an ill-posed problem with an infinite number of solutions for $\{\lambda_1(t_1), \lambda_2(t_2)\} \in \mathbb{R}$.

To combat the ill-posedness of this problem, a regularization term is used to generate solutions that trend toward some physical assumption. In this case, if both sensors and the reference measurements were perfectly accurate, the CGM profiles would match exactly and require no modification, thereby implying that in the ideal limiting case a true mean of the CGM measurements would be appropriate. As such, solutions that trend toward a true mean are preferred. Starting from this assumption, a Tikhonov regularization term is employed which applies a quadratic penalty on weights that deviate from 0.5. The quadratic nature of the regularization term means that solutions that deviate further from a simple mean of the two continuous glucose monitors are penalized more heavily. The resultant objective function is shown in Equation (2.3a).

$$\begin{aligned}\underset{\lambda_{a,i},\lambda_{b,i}}{\text{minimize}} \quad J = \sum_{i=1}^{N_R} & \left[\left(BG_{ref}(t_i) - [\lambda_1(t_i)CGM_1(t_1) + \lambda_2(t_i)CGM_2(t_2)]\right)^2 \right. \\ & \left. + \Gamma_\mu \left\{(\lambda_1(t_i) - 0.5)^2 + (\lambda_2(t_i) - 0.5)^2\right\}\right]\end{aligned} \tag{2.3a}$$

subject to:

$$\lambda_{1,2}(t_i) > 0 \quad \forall i = 1...N_R \tag{2.3b}$$

Looking again to the limit of perfect data, a simple mean of the CGM measurements would result in an exact match to the data at every point with constant weights (with a value of 0.5). As such, solutions which limit variability in the weights are preferred. In addition to preferentially generating solutions approximating a true mean, the regularization term in

Equation (2.3a) penalizes deviations from a constant value, so it beneficially tends to result in solutions which favor weights close to 0.5 and thereby minimizes changes in the weights from one reference measurement to the next. The magnitude of this regularization is set by Γ_μ in Equation (2.3a)

Solutions to the quadratic program of Equation (2.3a) result in a set of two weights for each (sparsely sampled) reference measurement but do not generate a full reconstruction of the glucose profile from the CGM data at the native high-frequency (5 minute) sampling rate of the continuous glucose monitors. To fully reconstruct the glucose trace from the two CGMs, a certain dynamic for the weights must be assumed in the times between reference measurements. The previous assertion that the variability of the weights should be minimized in an ideal solution motivates the choice of simple linear dynamics between reference CGM measurements. A linearly changing weight, from $BG_{ref}(t_i)$ to $BG_{ref}(t_{i+1})$, is the minimum variance path. Given the relative sparsity of the reference data and the reduction to a simple mean in the ideal limit, the values of the weights at intervening CGM measurement times between reference measurements are calculated using a simple linear interpolation. Following interpolation, the CGM measurement values and the associated interpolated weights are paired and a reconstructed glucose trajectory is then calculated as the weighted linear combination of the CGMs and their associated weights.

A Note on Pairing: Reference measurements are paired with their corresponding CGM measurement by simply finding the CGM measurement within the calibration window containing the reference measurement with a timestamp closest to that of the reference measurement. If the difference between the reference and CGM measurement times exceeds 5 minutes, it implies a missing CGM measurement and that reference measurement is discarded.

2.2.1.3 Dynamically Weighted Average of CGM Measurements (I) An evolution of the method discussed in Section 2.2.1.2 utilizes a dynamically weighted average of CGM measurements rather than a dynamically weighted linear combination to retrospectively reconstruct a glucose profile which is in closer agreement with reference measurements than either set of CGM data independently. For a given measurement time, the weighted average is shown in Equation (2.4).

$$BG_{true}(t_i) = \frac{1}{2}\left[\lambda_1(t_i)CGM_1(t_1) + \lambda_2(t_i)CGM_2(t_2)\right] \tag{2.4}$$

Although outwardly this appears to be an equivalent formulation to the linear combination presented in Section 2.2.1.2, this evolution allows weights to be regressed from the reference measurement for each CGM independently, then subsequently averaged, whereas the previous formulation required simultaneous regression of both weights for each reference measurement.

The primary benefit in enabling independent regression of the weights is that it allows reconstruction at times or over intervals where data may be missing from one CGM due to any number of technical or patient comfort and safety-related reasons. The previously discussed technique (Section 2.2.1.2) relies on a weighted linear combination of the CGM measurements, and as such, it requires values from all sensors at every time to reconstruct a more accurate glucose value. In this approach, a glucose trajectory can be safely reconstructed using both sensors, or in the case of missing data, a single CGM. The formulation for this modification is given in Equation (2.5).

In practice, the reference data and the CGM data are treated as piecewise continuous between recalibration points for each CGM, inclusive of the calibration points. This means that each interior calibration measurement is included both at the end of one interval and the beginning of the next. If the quadratic program shown in Equations (2.5a) and (2.5b) is solved for each interval using all reference measurements contained within that interval inclusive of the calibration measurements, the resultant profile will tend toward continuity around recalibrations.

$$\underset{\lambda_{(1,2),i}}{\text{minimize}} \quad J = \sum_{i=1}^{N_R}\left[\left(BG_{ref}(t_i) - \frac{1}{2}\boldsymbol{\lambda}(t_i)\mathbf{CGM}^T(t_i)\right)^2 + \Gamma_\mu\left(\boldsymbol{\lambda}(t_i) - 1.0\right)^2\right] \tag{2.5a}$$

$$\text{subject to:} \quad \boldsymbol{\lambda}(t_i) > 0 \quad \forall i = 1...N_R \tag{2.5b}$$

$$\text{where :} \quad \mathbf{CGM}(t_i) = [\text{CGM}_1(t_i), \quad \text{CGM}_2(t_i)]$$

$$\boldsymbol{\lambda}(t_i) = [\lambda_1(t_i), \quad \lambda_2(t_i)]$$

The quadratic program defined by Equation (2.5) is similar to Equation (2.3), but differs in that it minimizes the sum squared error between a reference glucose measurement and a *single* weighted CGM measurement at a corresponding time compared to the linear combination previously discussed. Following the logic detailed in Section 2.2.1.2 we would expect the weighted average to approximate an unweighted average. Because this approach reconstructs a blood glucose profile as a weighted average of the CGM data, rather than a linear combination, and because weights are regressed to the data from a single CGM independent of the other sensor, the regularization term in Equation (2.5) preferentially results in solutions with weights that take on the value of 1 in the ideal case (rather than 0.5).

A Note on Intervals: Throughout this discussion, intervals have been assumed to be the subset of all CGM, capillary fingerstick, and YSI® blood chemistry measurements between (re)-calibrations of that sensor. The final interval therefore remains unbounded due to the lack of a calibration point before the CGM was removed or stopped collecting data. As such, the final interval is treated as all points between the last sensor recalibration and the last available CGM measurement. In the event that the final interval does not contain any reference measurements in addition to the calibration measurement, Equation (2.9) is ill-posed with an infinite number of degenerate solutions. Under these circumstances, Equation (2.9) is solved for a bias alone under the assumption that $\lambda = 1$.

Inherent in this piecewise approach is the possibility of discontinuous weights as calibration points shift from the end of the preceding interval to the beginning of the current interval (see Equation (2.6)).

$$\lambda_{1,2}(t_i) = \begin{cases} \lambda_1 & \lim\limits_{t \to t_i^-} \\ \lambda_2 & \lim\limits_{t \to t_i^+} \end{cases} \tag{2.6}$$

As such, the linear interpolation of the weights for a given CGM at CGM measurement times is also calculated piecewise within intervals. Although these discontinuities result in significantly increased variability in the weights after recombination of the intervals, the increase occurs around recalibrations which are assumed to reset the internal state of the CGM, and as such there should be no expectations of continuity in the weights, whereas there is a definite physiologic expectation of continuity in glucose levels at all times.

Following the solution of Equation (2.5) for all intervals and interpolation of the weights, the piecewise-interval products of the CGM measurements and corresponding weights are concatenated to form the weighted glucose profile for a CGM.

2.2.1.4 Model-Based Approach The approaches to glucose reconstruction detailed in Sections 2.2.1.2 and 2.2.1.3 suffer from a number of significant drawbacks:

1. solutions are heavily dependent on the magnitude of regularization
2. solutions do not consider the differing accuracy of capillary fingerstick and blood chemistry analyzer measurements
3. physiologic constraints are not explicitly accounted for

In the previously described approaches, capillary fingerstick measurements and measurements from a YSI® blood chemistry analyzer are assumed to be equally accurate or valid. However, in practice these two methods have very different margins of error. Fingerstick measurements are assumed to be accurate to within ±10% [146], whereas blood chemistry analyzer glucose readings are significantly more accurate (±2%) [147]. In the previous approaches (Sections 2.2.1.2 and 2.2.1.3), the accuracy could be accounted for through a relative weighting of the reference measurement types in the objective function. However, this introduces more hyper-parameters in addition to the already present regularization, which have a significant impact on the resulting solutions. Furthermore, it becomes increasingly difficult to determine and justify the values of an increasing number of hyper-parameters. Finally, despite the efforts to address issues of continuity in Section 2.2.1.3, aphysiologic rates of change in glucose values may still remain.

To combat these shortcomings, a model-based approach was developed that retains the use of variable weights to correct CGM readings based on reference measurements in accordance with the understood accuracy of those measurements along with known physiologic constraints, while eliminating the need for regularization on the weights. Additionally, this method does not require the pairing of CGM values, thereby eliminating issues such as jumps or discontinuities associated with CGM dropout and missing CGM data.

This method is a two-step approach where the steps are:

i. remove bias and determine linearly varying weights for each CGM, independently, to generate corrected CGM trajectory

ii. fit physiologic model of glucose and insulin dynamics to unpaired corrected CGM trajectories simultaneously

In step i., the CGMs are corrected independently of each other, such that each CGM matches the reference data to within the known error of all measurements (fingerstick or blood chemistry analyzer). Under the assumption that portions of the collected CGM data are affected by poor or improper calibration techniques leading to a bias or constant offset in the measurements following calibration, the data from a single CGM are corrected in a piecewise manner between calibrations. For an interval between calibration points there is assumed to be a constant bias, B, which is additive with all points within the interval. Additionally, every CGM measurement in a CGM-reference pair is assigned a gain factor $\lambda_{(1,2),t}$ that corrects the measurement to within the error of the associated reference value. As before, there is an assumption that the gain factors should all tend toward 1.0. Previously, regularization was necessary for solutions with this characteristic, however the objective in this approach simply minimizes the sum-squared difference between each gain factor and unity subject to constraints that the corrected CGM trajectory is within the known error of each measurement type. This optimization problem is formulated in Equation (2.7).

$$\begin{aligned}
\underset{B,\lambda_{(a,b)},t_i}{\text{minimize}} \quad & J = \sum_{i=1}^{N_I} \left(\lambda(t_i) - 1.0\right)^2 && \text{(2.7a)}\\
\text{subject to:} \quad & \lambda(t_i) > 0 \quad \forall i = 1 ... N_I && \text{(2.7b)}\\
& CGM_z(t_w) + B \geq 0 \qquad \forall z \in \{a,b\}\,,\ \textit{and}\ \forall w \in \{j,k,l\} && \text{(2.7c)}\\
& \frac{BG_{ref}(t_w) - \lambda_z(t_w)\left[CGM_z(t_w) + B\right]}{BG_{ref}(t_w)} \times 100 \leq \alpha_w \\
& \qquad\qquad \forall z \in \{a,b\}\,,\ \textit{and}\ \forall w \in \{j,k,l\} && \text{(2.7d)}
\end{aligned}$$

Equation (2.7b) ensures that the λ's cannot become negative and Equation (2.7c) constrains the biases such that the CGM measurement after bias removal is always positive. Together, Equations (2.7b) and (2.7c) restrict the parameter space such that either a bias or gain correction alone still result in physically feasible solutions (*i.e.*, non-negative glucose concentrations). The constraint given in Equation (2.7d) ensures that each reconstructed glucose measurement in an interval is within α% of it's associated reference measurement where α is the known percent error associated with that measurement type – capillary fingerstick (j), blood chemistry analyzer (k) or calibration measurement (l). Although, in general, calibration measurements are obtained via a capillary fingerstick, they are handled within this constraint separately from other capillary fingerstick measurements so that a tighter constraint may optionally be specified to help ensure continuity around calibrations. For the work presented here, $\alpha_j = 10\%$, $\alpha_k = 2\%$, and $\alpha_l = 5\%$.

Following solution of the optimization problem given in Equation (2.7) for a given interval, gain factors for all CGM measurements within the interval are calculated via linear interpolation as before, and the corrected CGM data are calculated according to Equation (2.8). In contrast to the previous methods, the corrected measurements from the CGMs are not paired and combined but rather left independent for the second step in this approach.

$$CGM_{corr.}(t_i) = \lambda(t_i)\left(CGM(t_i) + B\right) \qquad \forall i \in \text{Interval} \tag{2.8}$$

After determination of the bias and gain correction factors in all intervals of available data from a patient, a physiologically consistent model of insulin and glucose dynamics (see Chapter 3) is fit to the corrected trajectories from both CGMs simultaneously resulting in a reconstructed glucose trajectory ($CGM_R(t_i)$). This approach is preferable to an arithmetic fusion of the corrected signals as seen in Sections 2.2.1.2 and 2.2.1.3 because the model is subject to physiological constraints, such as maximal rates of glucose change, and ensures smoothness in the resultant reconstructed blood glucose profile. This ameliorates issues of discontinuity around recalibrations and prohibits aphysiologic behavior in CGM_R – characteristics not guaranteed by the previously discussed methods. Furthermore, assumptions about the accuracy of a CGM over a particular window, or the instantaneous accuracy of a sensor, can be incorporated into the model regression to the corrected data to de-emphasize contributions of poorly calibrated or potentially faulty sensors.

Under the assumption that CGMs that were improperly calibrated suffer from bias, and therefore are less accurate, every corrected CGM measurement can be weighted by $\frac{1}{1+|B_i|}$, where B_i is the bias associated with the interval of origin for the CGM measurement as determined via the optimization problem in Equation (2.7). Similarly, to mitigate the effects of faulty or erroneous measurement, a corrected CGM measurement ($CGM_{corr}(t_i)$) may also be weighted by $\frac{1}{1+|1-\lambda(t_i)|}$. By weighting CGM values in the model fit to the corrected CGM data, the most accurate data – data that required little correction – is preferentially fit.

This method seeks to ensure continuity across recalibrations by correcting the CGM data in intervals both before and after each sensor calibration. By including the calibration measurement as a reference in Equation (2.7a) and setting a conservative error threshold in Equation (2.7d), the gap between corrected CGM data, both preceding and following calibration can be tightly controlled. The internal recalibration process for Dexcom® Platinum™ G4 continuous glucose monitors is proprietary, but the data appear to show that there may be rapid dynamics in the approximately 20 minutes following recalibration. As noted earlier, reference measurements are paired with the closest CGM measurement within 5 minutes of the reference. Due to the potential for significant glucose changes at calibration points, the point immediately following recalibration may not be re-equilibrated with the calibrated value and may be a poor point of comparison to the calibration point leading to excessive variability in the λ's. In an effort to mitigate the confounding effects of calibration dynamics on the reconstruction, an extension to the model-based reconstruction method using the values further along the CGM trajectory following recalibration and/or the mean of multiple values subsequent to recalibration is described here.

Following a calibration, N points, starting with some time offset by β from the recalibration, are averaged and replace the first CGM measurement following recalibration as the CGM value paired with the recalibration reference. This modification results in the changes to the formulation given in Equation (2.7) shown in Equation (2.9), which is solved for every interval.

$$\underset{B,\lambda_{(a,b)},t_i}{\text{minimize}} \quad J = \sum_{i=1}^{N_I} \left(\lambda(t_i) - 1.0\right)^2 \tag{2.9a}$$

$$\text{subject to:} \quad \lambda(t_i) > 0 \quad \forall i = 1 \ldots N_I \tag{2.9b}$$

$$\rho_z(t_w) + B \geq 0 \qquad \forall z \in \{a,b\}\,,\ and\ \forall w \in \{j,k,l\} \tag{2.9c}$$

$$\frac{BG_{ref}(t_w) - \lambda_z(t_w)\left[\rho_z(t_w) + B\right]}{BG_{ref}(t_w)} \times 100 \leq \alpha_w$$

$$\forall z \in \{a,b\}\,,\ and\ \forall w \in \{j,k,l\} \qquad \text{where:}$$

$$\rho_z(t_i) = \begin{cases} \frac{1}{N_\beta} \sum_{j=\beta}^{\beta+N_\beta} CGM_z(t_j) & i = 1 \\ CGM_z(t_i) & i = 2 \ldots N_I - 1 \end{cases} \tag{2.9d}$$

In the case of $\beta = 0$ and N $= 1$, Equation (2.9) reduces to Equation (2.7).

2.2.1.5 Numerical Methods The optimization problem given in Equations (2.3), (2.5), (2.7) and (2.9) was implemented in Python using the Pyomo modeling and optimization package ([148, 149]). All nonlinear programs were solved using the interior point algorithm IPOPT as the nonlinear solver [150] and ma27 from [151] as the linear solver.

2.2.2 Results

An example showing the raw CGM data for a particular patient and the corrected CGM data following bias removal and weighting via the two step method from Section 2.2.1.4 is shown in Figure 6. From Figure 6 it is clear that the combination of a shift to remove CGM bias and a linearly varying weight to account for CGM drift or fluctuations in the sensor gain is able to correct the raw CGM data such that it is in close agreement with reference capillary fingerstick measurements.

Although a detailed explanation of the modeling efforts and optimization techniques used to generate a model-based reconstruction (CGM_R) of a blood glucose trajectory is reserved for Chapter 3, Figure 7 shows an example of the model-based reconstruction compared to the corrected CGM data along with the shifts, B, and weights, $\lambda_{1,2}$, associated with each CGM. The data collected by CGM 1 from the patient shown in Figures 6 and 7 exhibit what is believed to be a pressure induced loss of sensitivity in the region around 500 minutes. In response to the pressure-induced loss of sensitivity, the magnitudes of the bias and gain factors associated with that CGM, over the interval including the event, are significantly elevated compared to the nominally functioning sensor, as shown in the bottom panel of Figure 7. When contributions to the objective function governing the fit of the model (see Chapter 3) for each corrected CGM value are weighted by $\frac{1}{1+|B_i|}$ and $\frac{1}{1+|1-\lambda(t_i)|}$, the likely faulty data from CGM 1 is effectively ignored as evidence by the close match of the reconstructed glucose trajectory to the data from CGM 2 over the interval containing the pressure-induced loss of sensitivity.

The model-based approach employed here easily allows for the specification of constraints on the maximum rate of change of glucose. Limiting the glucose rate of change to a physiologic regime provides a simple path to dealing with any remaining discontinuities in the corrected CGM data. Applying the approach laid out in Section 2.2.1.4 results in reconstructed blood glucose trajectories which are in good agreement with reference measurements of greater accuracy and ultimately results in a cohort of virtual patients believed to be both physiologically consistent and clinically realistic. A detailed description of the virtual patient cohort and its relation with both physiology and clinical observation is given in Chapter 3.

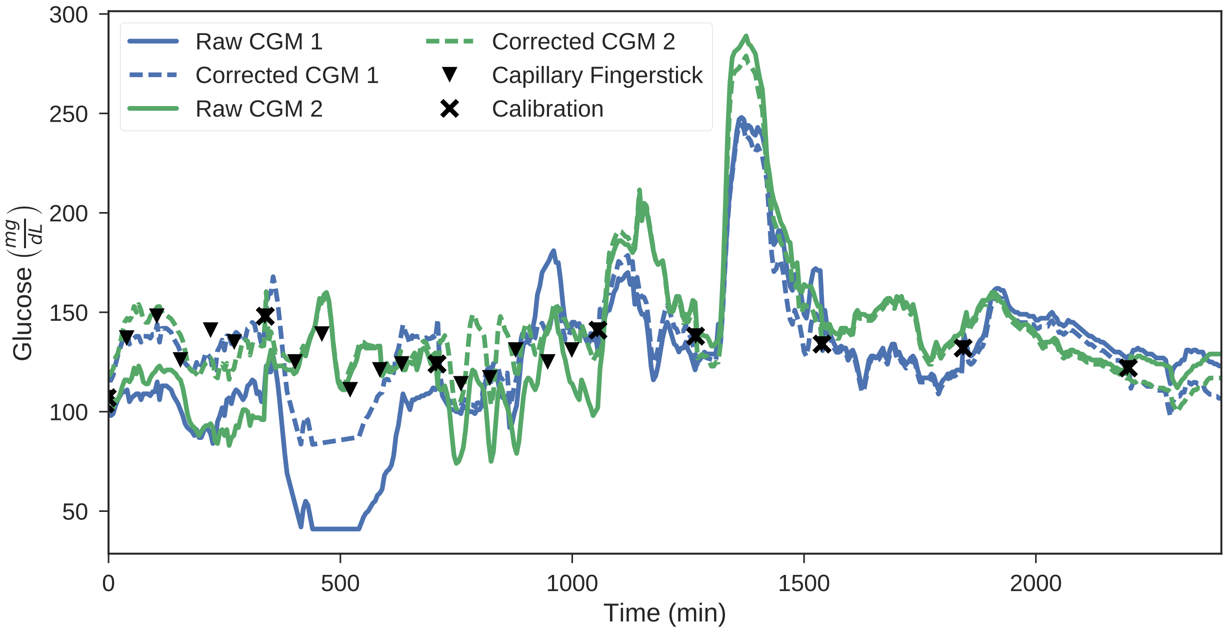


Figure 6: Raw and corrected CGM values for two Dexcom® Platinum™ G4 continuous glucose monitors placed subcutaneously in Patient 16. The solid lines (—) show the raw data and the dashed lines (- - -) denote the corrected signals. Capillary fingerstick measurements and recalibration points are given by the ▼'s and **x**'s, respectively.

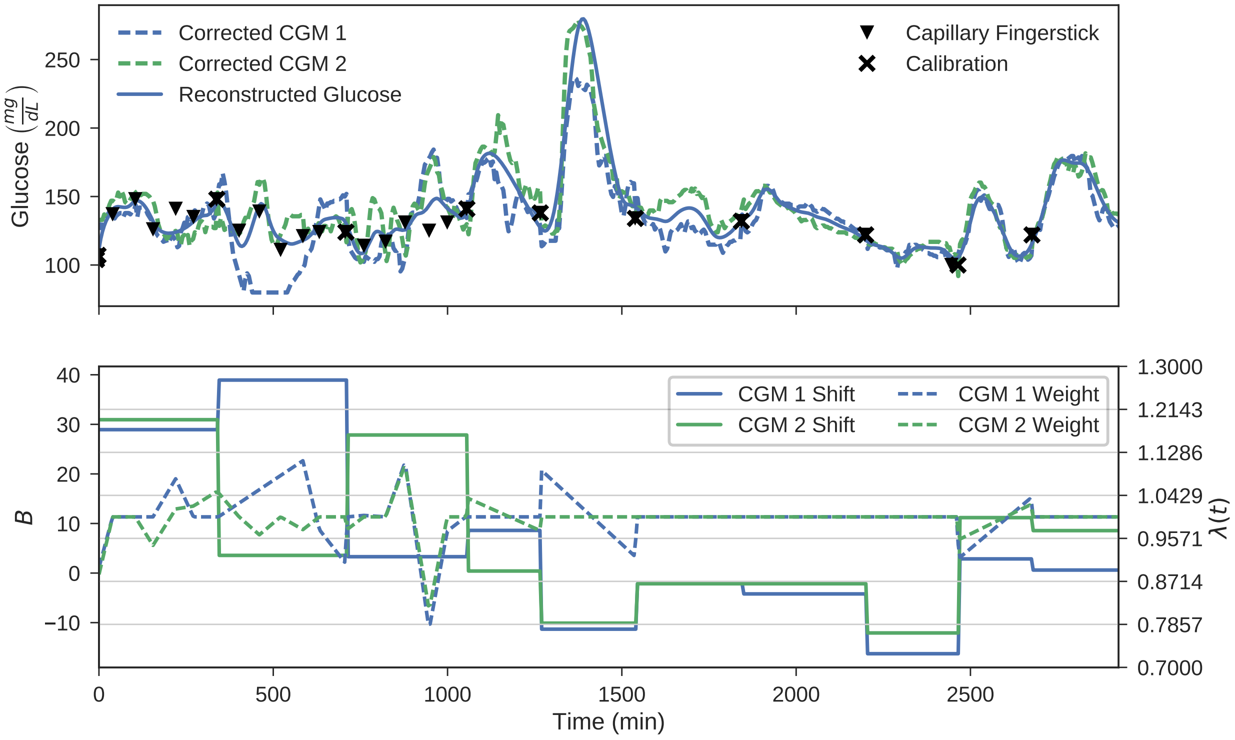


Figure 7: Top: Model-based blood glucose trajectory reconstructed from corrected CGM data using a model of glucose-insulin dynamics and medical record data (see Chapter 3). Bottom: Shift and weights for the corrected CGM data.

Table 2: Statistics for the biases (B) and linearly varying gain correction factors (λ) for all patients and intervals. IQR represents the interquartile range (25% –75%) of the data.

	μ	σ	Median	Min	Max	IQR
$B\left(\frac{mg}{dL}\right)$	6.60	18.625	4.51	-38.37	86.88	-3.06 –13.39
λ	0.98	0.072	0.997	0.54	1.47	0.96 –1.014

2.2.3 Discussion

By identifying an additive bias for each interval of CGM data and a multiplicative gain correction factor for all CGM measurements, raw CGM data can be corrected such that it is in close agreement with reference glucose measurements. The optimization problem governing this correction aims to minimize deviations from unity in the multiplicative gain correction factors but allows for a free range of bias values subject to a non-negativity constraint (Equation (2.7c)) such that percent error of any CGM-reference measurement pair is $\leq$ the known accuracy of the reference measurement type.

Because the bias term is additive across an interval, it preserves, exactly, the shape of the raw CGM data. Conversely, due the fact that the gain correction is a linearly varying multiplicative modifier, if $\lambda(t_i) - \lambda(t_{i-1}) \neq 0$ and $\lambda(t_i) \geq 0$, there is necessarily a stretching or compression of the raw CGM trajectory leading to potentially significant differences in the dynamics of the corrected data compared to the raw data. Larger $|\lambda(t_i) - \lambda(t_{i-1})|$ induces greater deformation. The optimization problem given in Equation (2.9) minimizes $(\lambda(t_i) - 1)^2$, which has the effect of also minimizing $|\lambda(t_i) - \lambda(t_{i-1})|$. $|\lambda(t_i) - \lambda(t_{i-1})|$ (or similarly $(\lambda(t_i) - \lambda(t_{i-1}))^2$ is not minimized directly in the objective function because the limit $\lambda(t_i) - \lambda(t_{i-1}) = 0$ does not imply $\lambda(t_i) = 1.0$, the original assumptions or intuition for the weights yielding ideal CGM function.

The distribution of the biases (B) and gain correction factors (λ) are shown in Figures 8 and 9 respectively. The statistics of the parameter distributions are given in Table 2. Figure 8 and Table 2 demonstrate that the λ's are, in general, tightly grouped around the

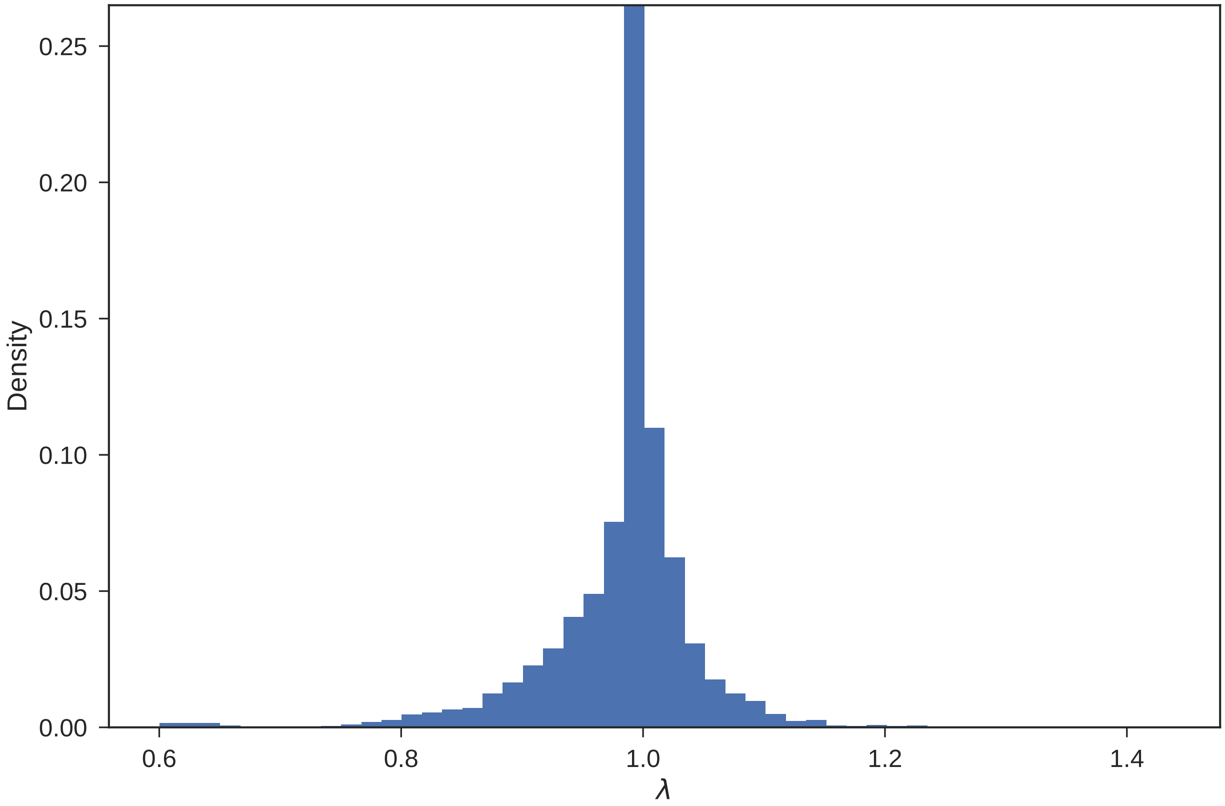


Figure 8: Normalized density of correction factors, $\lambda(t_i)$, for all patients (N=17) after solving the optimization problem in Equation (2.9) for intervals with available reference data.

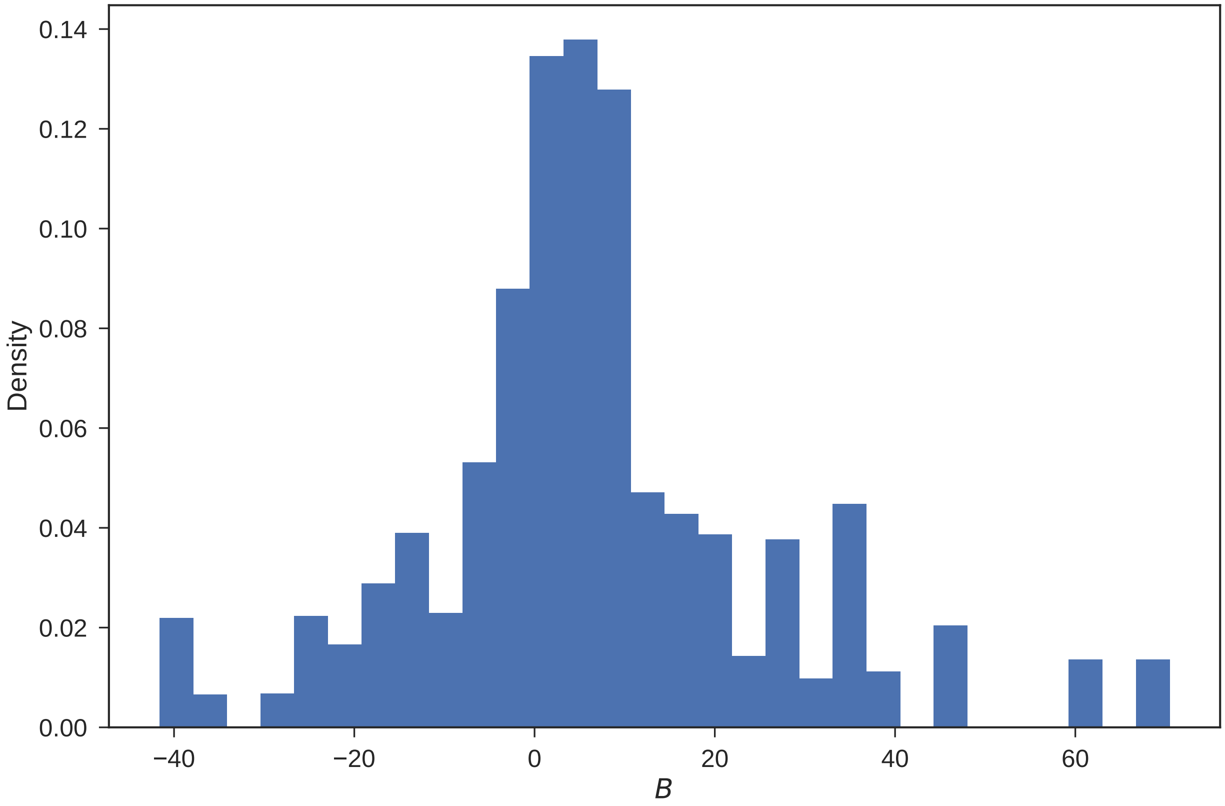


Figure 9: Normalized density of biases, B, for all patients (N=17) after solving the optimization problem in Equation (2.9) for intervals with available reference data.

nominal value of 1 (±0.17). Meanwhile the biases have a much higher degree of variability, as seen in Figure 9, and positive mean indicating that in general the CGMs seem to report glucose values that are too low. The relatively low variability in the λ's is consistent with the objective of minimizing $(\lambda(t_i) - 1)^2 \ \forall i)$ and implies a low level of modification to the raw CGM dynamics. Although, in general, the λs tend strongly toward unity, as evidenced by the standard deviation and interquartile range in Table 2, the minimum and maximum values indicate regions of exceptionally poor CGM performance. The marked difference from one of the λ's in these extreme cases suggests that the corresponding CGM data is fundamentally flawed with respect to glucose dynamics as significant variability over time is required to recapitulate the reference measurements. The patient data resulting in the minimum λ over all patients and intervals is shown in Figure 10 as an example and for reference.

Up to approximately 1000 minutes, both CGMs in Figure 10 exhibit extremely poor performance compared to reference blood glucose measurements. Because the poor performance appears to be endemic to both sensors and recovers immediately following a recalibration it is likely that the sensors were calibrated to an erroneous measurement or at a point during rapidly changing blood glucose levels, in contradiction to the operating procedure for the Dexcom® Platinum™ G4 continuous glucose monitor [152]. Acetaminophen is also known to degrade the performance of continuous glucose monitors [145] but causes falsely elevated glucose readings rather than the under-reported values seen here. With the medical record showing no indication that this patient may have received acetaminophen or any other medication, miscalibration is the likely culprit for the poor performance over the first interval.

Over the region of under-reported glucose measurements, the procedure detailed in Section 2.2.1.4 is able to correct the raw CGM data so that it is in agreement with the reference measurements. The correction requires the removal of a large negative bias and highly variable λ's that deviate significantly from unity (see Figure 11). The steep slopes and large deviations from 1 exhibited by the λs in Figure 11 for the poorly performing CGMs result in significant differences in trajectory and dynamics between the raw and corrected data. This is readily apparent in Figure 10, where close inspection reveals sections of data where the raw and corrected data are trending in different directions. Despite the large differences in magnitude and dynamics from the raw data, the corrected data recapitulates reference data

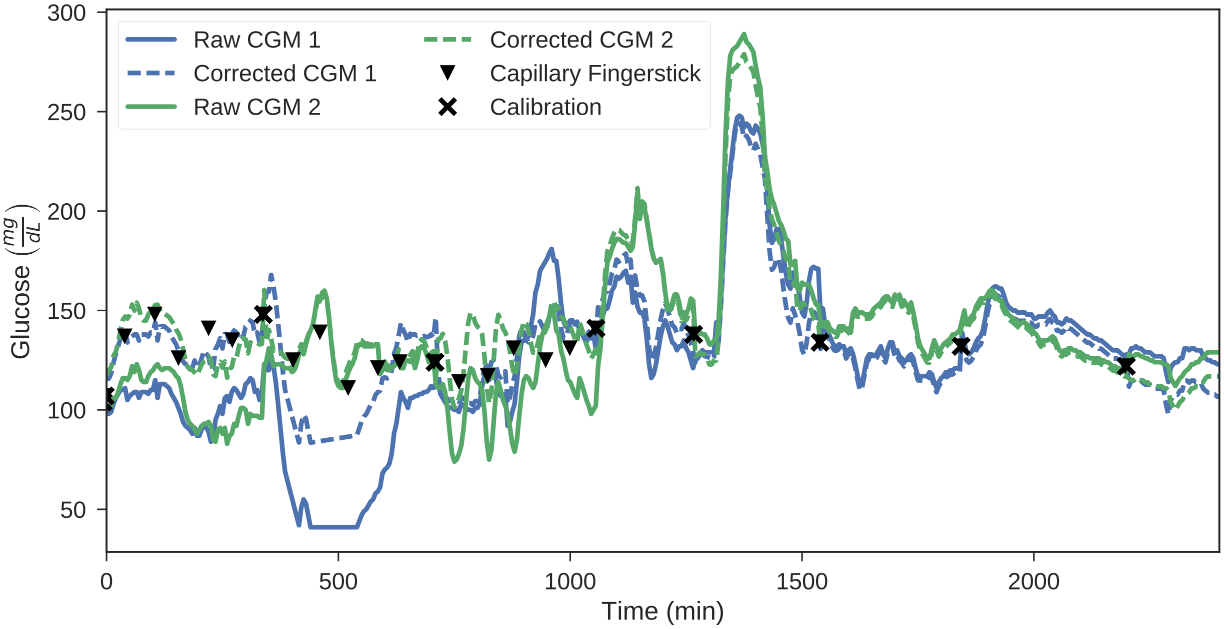


Figure 10: Raw and corrected CGM values for two Dexcom® Platinum™ G4 continuous glucose monitors placed subcutaneously in Patient 5. The solid lines (-) show the raw data and the dashed lines (–) denote the corrected signals. Capillary fingerstick measurements and recalibration points are given by the ▼'s and **x**'s, respectively. The region between 0 and 1000 minutes exhibits extremely poor CGM performance, with both CGMs significantly under-reporting measured glucose concentrations.

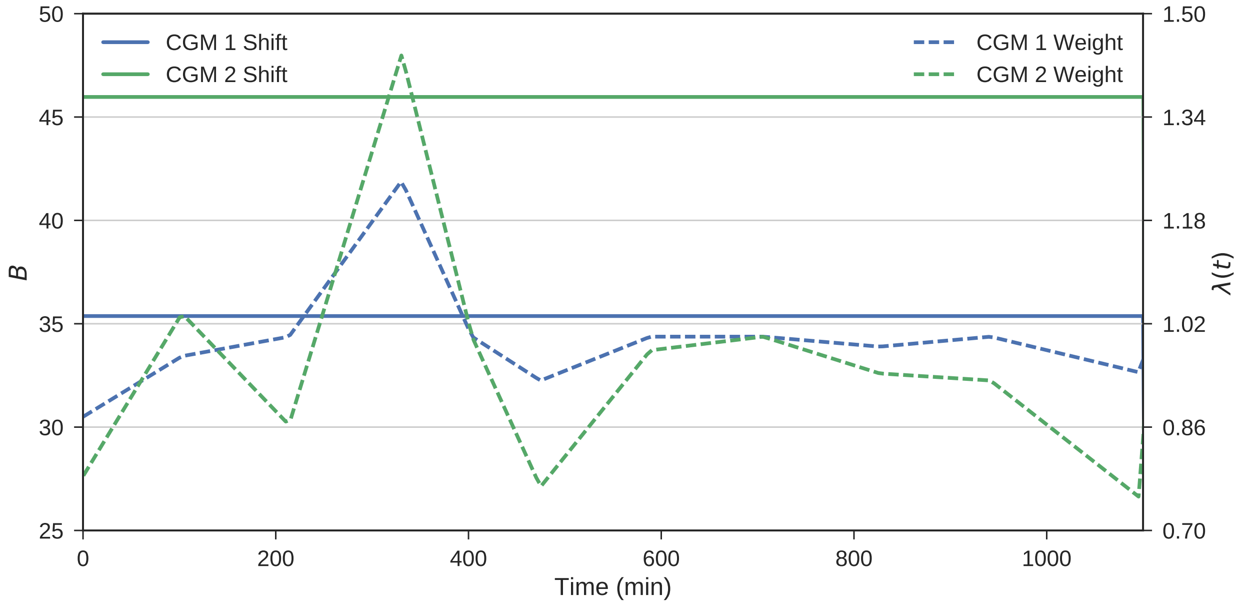


Figure 11: Interval biases (B) and gain corrections (λ's) for a region of poor performance by both CGMs.

collected at a comparatively high frequency and exhibits dynamics that are readily defended compared to raw data. For example, CGM 2 reports rapid drops into a state of severe hypoglycemia followed by a rapid rebound with no glucose or insulin infusions recorded in the medical record to support this trend. Furthermore, the raw CGM data are not particularly coherent or well correlated, whereas the corrected CGM trajectories are significantly more coherent and correlate very well.

Facchinetti *et al.* [153] employed a similar approach, which allowed for a polynomially varying bias and gain over an interval. That study found that both a bias and gain which varied linearly resulted in the best reconstruction. The work by Facchinetti *et al.* differs from the approach discussed here in that the method described in Section 2.2.1.4 utilizes a constant bias and time-varying multiplicative gains within an interval. For intervals containing only calibration points, the method employed here reduces to that of [153] with a constant, instead of linearly varying, bias. For intervals containing one or more non-calibration reference measurements, an argument could be made that this methodology may approach that of [153] with polynomial order N, where N is the number of reference measurements leading to a polynomial gain that redirects the CGM data through all reference measurements. However, Facchinetti *et al.* had the benefit of high frequency reference measurements (approximately every 15 minutes). The data available for this study is much more sparse, with only a small handful of reference measurements available in an interval. As such, fitting a single linearly varying gain correction factor over an interval becomes extremely sensitive to outliers and often results in large discontinuities across calibration windows. Fitting higher order polynomial functions to describe the variations in the gain becomes untenable due to the lack of intuition regarding physical relevance of the polynomial coefficients. The approach discussed here hinges on the assumption that $\boldsymbol{\lambda}$ should tend toward unity. No such assumption exists for quadratic or higher order polynomial functions to drive the coefficients toward some value. As such, with the sparse data available, the problem rapidly becomes ill-posed for higher-order functions. For these reasons, and through the evolution of several approaches, the retrospective glucose reconstruction discussed here was developed and found to be effective, efficient and generate solutions consistent with physiology and certain physical assumptions.

2.3 MODELING CGM ERROR

The previous section (2.2.1.4) detailed a method for correcting high-frequency CGM data by removing biases and linear drifts in the sensor gain. The corrected CGM trajectories are then used to reconstruct a single blood glucose profile and in turn a virtual patient cohort as detailed in Chapter 3. In this section, that reconstructed glucose data is utilized in the synthesis of a stochastic model for the CGM error process.

The true instantaneous error ($E_I(t_i)$) for a CGM at any point can be calculated as the difference between the reference data ($R_G(t_i)$) and the raw CGM data (Equation (2.10)).

$$E_I(t_i) = R_G(t_i) - CGM(t_i) \tag{2.10}$$

However, because $E_I(t_i)$ can be calculated only when reference measurements are available, it is not representative of the error at the CGM sampling rate, but rather it is assumed to be the result of an accumulation (integration) of the error in every CGM measurement from calibration up to the current time. As such, the reconstructed glucose profile ($CGM_R(t_i)$) developed in Section 2.2.1.4 is used as a surrogate reference to calculate a reconstructed error ($E_R(t_i)$, Equation (2.11)) at the native CGM sampling rate.

$$E_R(t_i) = CGM_R(t_i) - CGM(t_i) \tag{2.11}$$

To develop a model of the CGM error process, autoregressive-plus-moving-average ($ARMA$) models are fit to the high-frequency reconstructed error. From the reconstructed error, we can identify properly functioning CGMs and sensors that appear to be malfunctioning. A set of $ARMA$ models are fit to the set either properly functioning or malfunctioning CGMs independently resulting in a stochastic model for "normal" CGM noise or error, and a stochastic model for the rapid growth in error observed in faulty or malfunctioning CGMs.

2.3.1 Materials and Methods

2.3.1.1 Identifying Faulty CGM Data All continuous glucose monitors will suffer from some degree of noise in their measurements, such that small errors are not indicative of a faulty sensor. In Section 2.2.1.4, it was shown that even a single CGM may perform well over certain intervals and at other times may function very poorly, as evidenced by large biases and a highly variable gain which deviates significantly from unity. A common measure used to assess the accuracy of continuous glucose monitoring is mean absolute relative difference (MARD, Equation (2.12)) [76, 77].

$$\text{MARD} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\hat{G}(t_i) - G(t_1)\right|}{G(t_1)} \tag{2.12}$$

where $G(t_i)$ is true blood glucose at time t_i and $\hat{G}(t_i)$ is blood glucose as measured by CGM at time t_i.

A recent study by Wilinska and Hovorka [78] suggested that MARD $\leq 10\%$ is optimal for TGC and resulted in the lowest frequency of hypoglycemia [154]. This result was corroborated by a similar study, which found that MARD $\leq 11\%$ was optimal [79]. The MARD for each CGM over every interval in our data was calculated with $G(t_i) = CGM_R(t_i)$ and $\hat{G}(t_i) = CGM(t_i)$, and the recommendations of these studies are used as a threshold between functioning and malfunctioning sensors. This results in the identification of a set of intervals with $MARD > 10\%$, which are presumed to contain data from malfunctioning CGMs, and a set of nominally functioning sensors with $MARD \leq 10\%$. These two sets are used to develop stochastic models describing both the noise present even in fault-free CGM operation and the error process governing faulty CGMs leading to significant drift and inaccurate glucose measurements.

2.3.1.2 Autoregressive-plus-Moving-Average (ARMA) Models $ARMA$ models are commonly used to describe stochastic time series processes and will be employed here to model CGM error. The notation $ARMA(p, q)$ represents a model consisting of an autoregressive polynomial with p terms and a moving-average polynomial with q terms. The general form for an $ARMA$ model is given in Equation (2.13).

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{2.13}$$

Here c is a constant trend, ε is white noise process with variance σ_ε^2, φ_i are the coefficients of the autoregressive polynomial and θ_j are the coefficients of the moving average polynomial.

$ARMA$ models assume weakly stationary data, meaning that the data and autocovariance do not vary with respect to time. Here, the augmented Dickey-Fuller test is used to check for stationarity. Because it was found, In general, that the reconstructed error was not stationary, the reconstructed error data was differenced once over each interval resulting in a stationary process. Here, stationarity implies that the joint probability distribution, and consequently the interval mean and variances of the error, do not change with time. If X_t

is assumed to be the integrated error of a single differenced error signal ($dX_{t_i} = X_{t_i}X_{t_{i-1}}$), then the integrated form of Equation (2.13) for X_t is given by Equation (2.14).

$$\left(1 - \sum_{i=1}^{p} \varphi_i L^i\right)(1-L)X_t = \delta + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right)\varepsilon_t \tag{2.14}$$

L in Equation (2.14) is the lag operator defined as $LX_t = X_{t-1}$.

2.3.1.3 Identifying p and q in $ARMA(p, q)$

The autocorrelation and partial autocorrelation functions are commonly used to identify the appropriate order for p and q. In general, it was observed that the autocorrelation function for the differenced error data dropped off immediately to levels indicating no significant correlation after a single lag. This is indicative of an $ARMA(0, 1)$ process or simply an integrated moving average model with one term ($MA(1)$, see Equation (2.15)) and a constant (c).

$$X_t = \delta + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right)\varepsilon_t \tag{2.15}$$

2.3.1.4 Akaike Information Criterion

The Akaike Information Criterion(AIC) [155, 156] is commonly used as a measure of the tradeoff between the quality of model fit and the complexity of the model. In general, parsimony as characterized by fewer unknown parameters is favored to avoid over-fitting. The AIC is computed as:

$$\mathrm{AIC} = 2k + n\ln\left(\frac{\mathrm{RSS}}{n}\right) \tag{2.16}$$

where k is the number of estimated parameters, n is the number of data points and RSS is the residual sum of squares.

From Equation (2.16), it is clear that as k, the number of estimated parameters, increases, the AIC value will grow if there is not a corresponding decrease in the residual from a least-squares regression. Conversely, a reduction in the number of parameters, which does not significantly increase the fitting error as measured as the residual sum of squares, will result in a reduced AIC. Here, the AIC is used to validate the choice of an $MA(1)$ model structure and evaluate the need for a non-zero constant trend (c in Equations (2.13) and (2.15)).

2.3.1.5 CGM Bias Because the reconstructed error data were found to be non-stationary and because $ARMA$ models assume stationarity, the data were made stationary via differencing. As a result of the differencing, any constant offset or bias is removed from the data. The reconstruction process in Section 2.2.1.4 enabled the identification of a constant bias for all intervals (see Figure 9 for the distribution of biases) which may be added to the ARMA process. As before, the biases are separated into bias subsets corresponding to the nominally functioning and malfunctioning CGM intervals. The distributions of these subsets can then be sampled when simulating or regenerating a CGM error trajectory to re-introduce a realistic offset post-hoc.

2.3.2 Results

The moving average coefficient (θ), along with σ_{ε}^2, were fit using the *statsmodels* [157] package in Python. As a result of differencing the error data to induce stationarity, any trends or offset are removed. As such, there is assumed to be no constant in the model and c is taken to be equal to 0. Assuming an $MA(1)$ model, the distributions of the MA coefficient θ as well as the error variance σ_{ε}^2 are shown for both the functioning and malfunctioning CGMs in Figure 12 and the statistics for θ are shown in Table 3.

Table 3: Statistics for the moving average parameter (θ) and the variance of the white noise (ε) driving the MA process for all intervals with $MARD \leq 10\%$ (functioning) and all intervals with $MARD > 10\%$ (malfunctioning).

		Mean	Median	Standard Deviation	Min	Max
Functioning	θ	0.401	0.402	0.185	-0.016	0.858
	σ_{ε}^2	5.96	2.93	8.11	0.584	55.86
Malfunctioning	θ	0.401	0.394	0.227	-0.208	0.916
	σ_{ε}^2	11.25	8.72	9.202	0.702	38.89

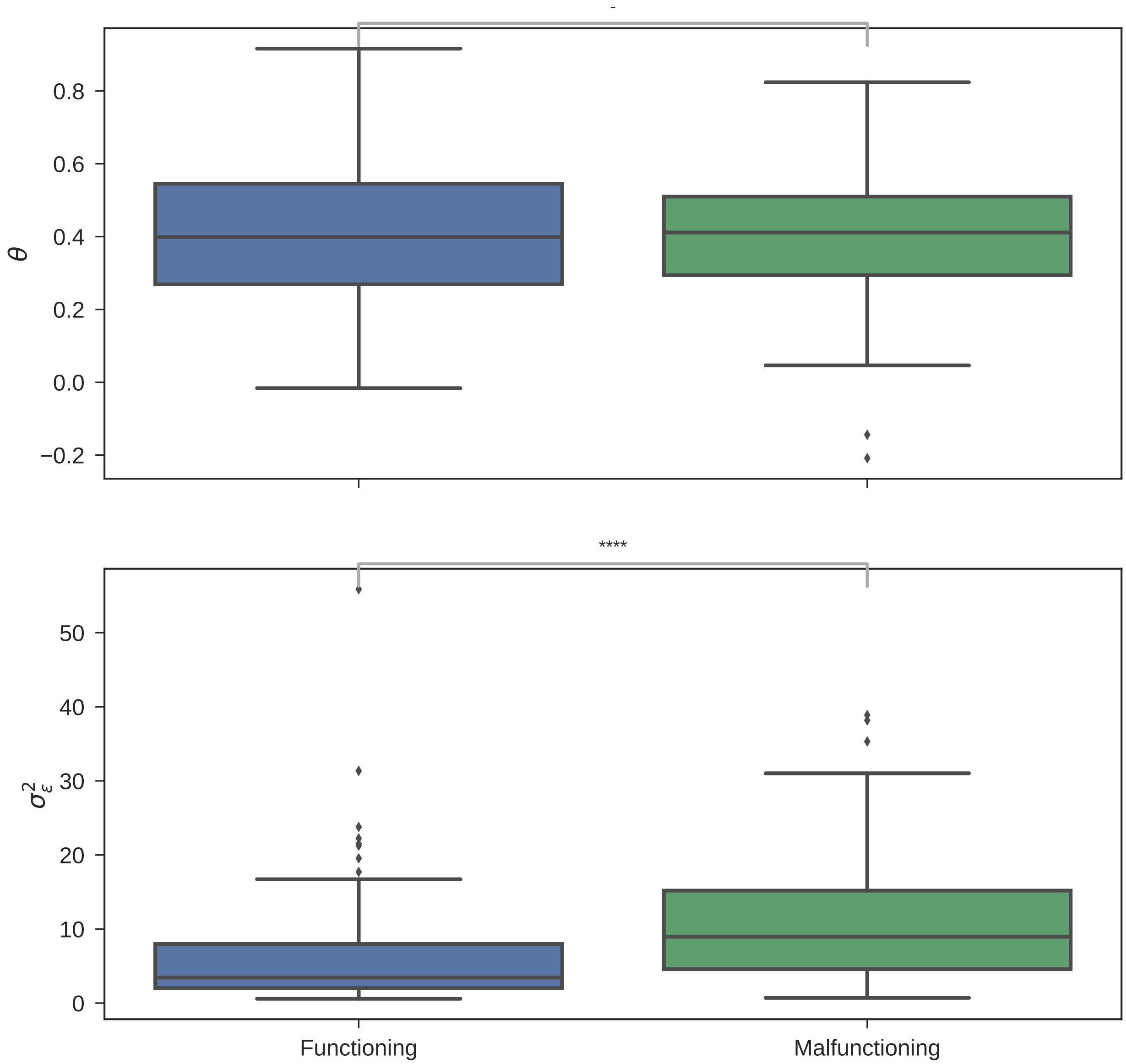


Figure 12: (Top) Distribution of the moving average parameter (θ) and (Bottom) the white noise variance (σ_ε^2). The outer edges of the box denote the interquartile range, the middle bar indicates the median, the boxes extend to $1.5 \times IQR$, and the ⧫s indicate outliers. Significance (via Mann-Whitney U Test) is indicated with $*$ at the following levels: $-=$ no significance, $*$=95%, $**=$ 99%, $***=$ 99.9%, $****=$ 99.99%

Figure 13 shows a histogram of the biases for intervals with proper CGM functioning and intervals where a CGM is presumed to be malfunctioning. Normal distributions were found to provide a good description of the biases, and the fit of the normal distributions are shown in Figure 13. The parameters describing the normal distributions (Equation (2.17)) for the functioning and malfunctioning CGMs are given in Table 4.

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.17}$$

where μ is the mean of the distribution and σ^2 is the variance.

From the $MA(1)$ model and the distribution of CGM biases, a simulation or likely realization of a CGM error profile (E_S) can be constructed from the integration or cumulative sum of the $MA(1)$ process with a constant variance and bias as:

$$E_S(t_j) = \sum_{i=1}^{j} \left[\theta_i E_S(t_{i-1}) + \varepsilon(t_i)\right] + N(\mu_B, \sigma_B^2) \tag{2.18}$$

where $N(\mu_B, \sigma_\varepsilon^2)$ is a random sample from the appropriate normally distributed bias function held constant across an interval. Equation (2.18) gives the full CGM error model for both nominally functioning and faulty CGMs where θ, σ_ε^2 and $N(\mu_B, \sigma_B^2)$ fully determine whether the model is simulating functioning or malfunctioning sensors.

1000 realizations of the integrated moving average model for both nominally functioning and malfunctioning models were generated for each interval length within the functioning or malfunctioning subgroup. For all the moving average parameters (θ, σ_ε^2) median values were used to diminish the effect of outliers on the trajectories. Constant biases were sampled directly from the distribution $N(\mu_B, \sigma_B^2)$ for the corresponding functioning or malfunctioning subgroup. Because the integration of the moving average model results in a process with a variance that increases with time, realizations were sampled with a length equal to the median value of all interval lengths within the CGM subgroup. The cumulative histograms comparing simulated error for both CGM subgroups compared to the true observed error are shown in Figures 14 and 15. The statistics of observed and simulated error trajectories are shown for both groups in Table 5.

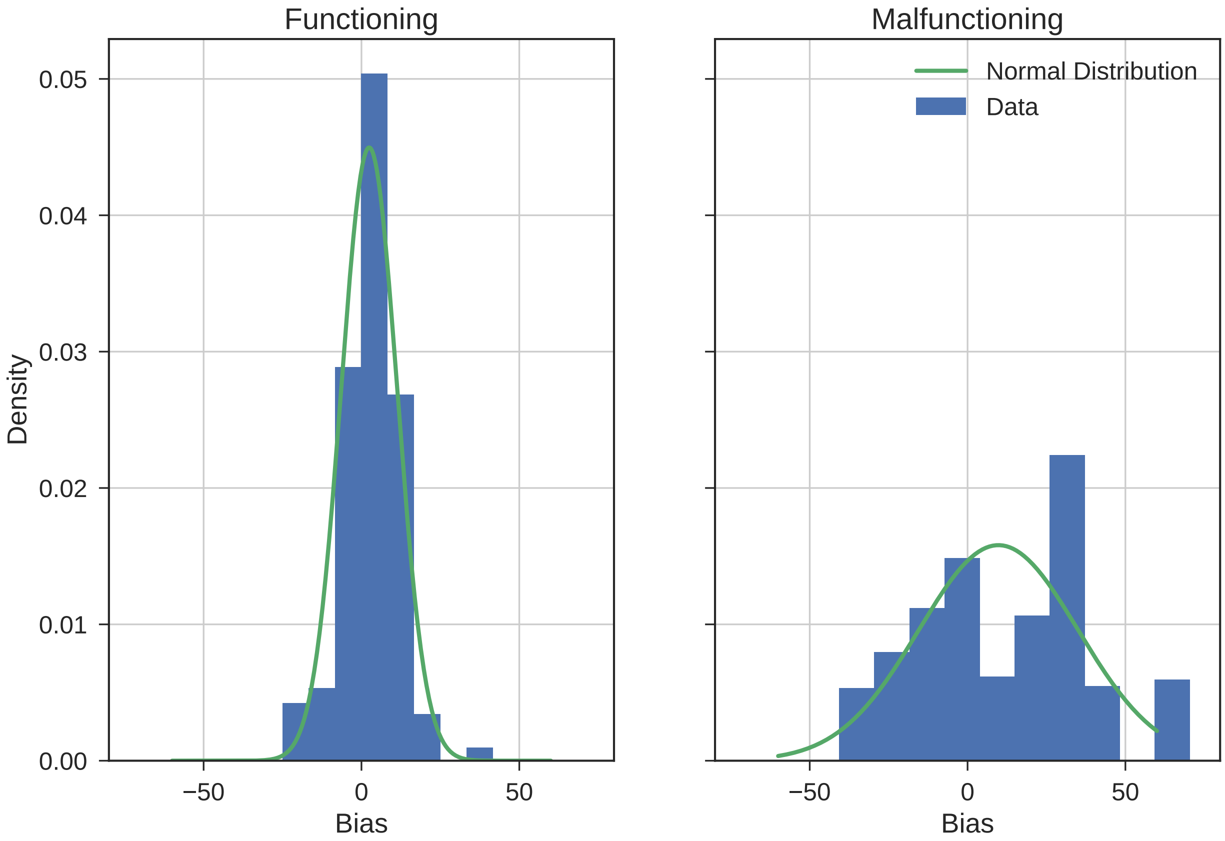


Figure 13: Distribution of bias parameter (B) for nominally functioning continuous glucose monitors (Left) and malfunctioning sensors (Right).

Table 4: Parameters for the normal pdf (Equation (2.17)) describing the distribution of the biases for both functioning and malfunctioning CGMs.

	μ $\left(\frac{mg}{dL}\right)$	σ $\left(\frac{mg}{dL}\right)$
Functioning	2.01	7.15
Malfunctioning	7.90	20.48

An additional 1000 realizations were simulated for the equivalent of 700 minutes of measurements and compared to the true error trajectories for both the nominally functioning and malfunctioning CGM subsets. These results are shown in Figures 16 and 17.

As previously mentioned (Section 2.3.1.3), the autocorrelation function of the differenced error data suggested that the data were described by a moving average model with one history term. Section 2.3.1.4 explained the use of the Akaike Information Criterion (AIC) to assist in model selection in a manner that favors parsimony. Table 6 presents the AIC values for a variety of alternative and/or higher order models.

2.3.3 Discussion

For both the nominally functioning and malfunctioning CGM data, the Aikaike Information Criterion (AIC) was not found to be the lowest for the chosen model structure ($ARMA(0,1)$). However, in general, there is little variance in the AIC across all model structures and an $ARMA(0,1)$ model was found to best represent both the reconstructed and observed CGM error qualitatively (Figures 14 to 17). Although the mean AIC in Table 5 results from a second order moving average ($ARMA(0,2)$) model, it is somewhat misleading in so far as fitting routines fail to converge on a solution for all intervals using higher order models. As such, there are fewer total intervals included in the calculation of mean AIC

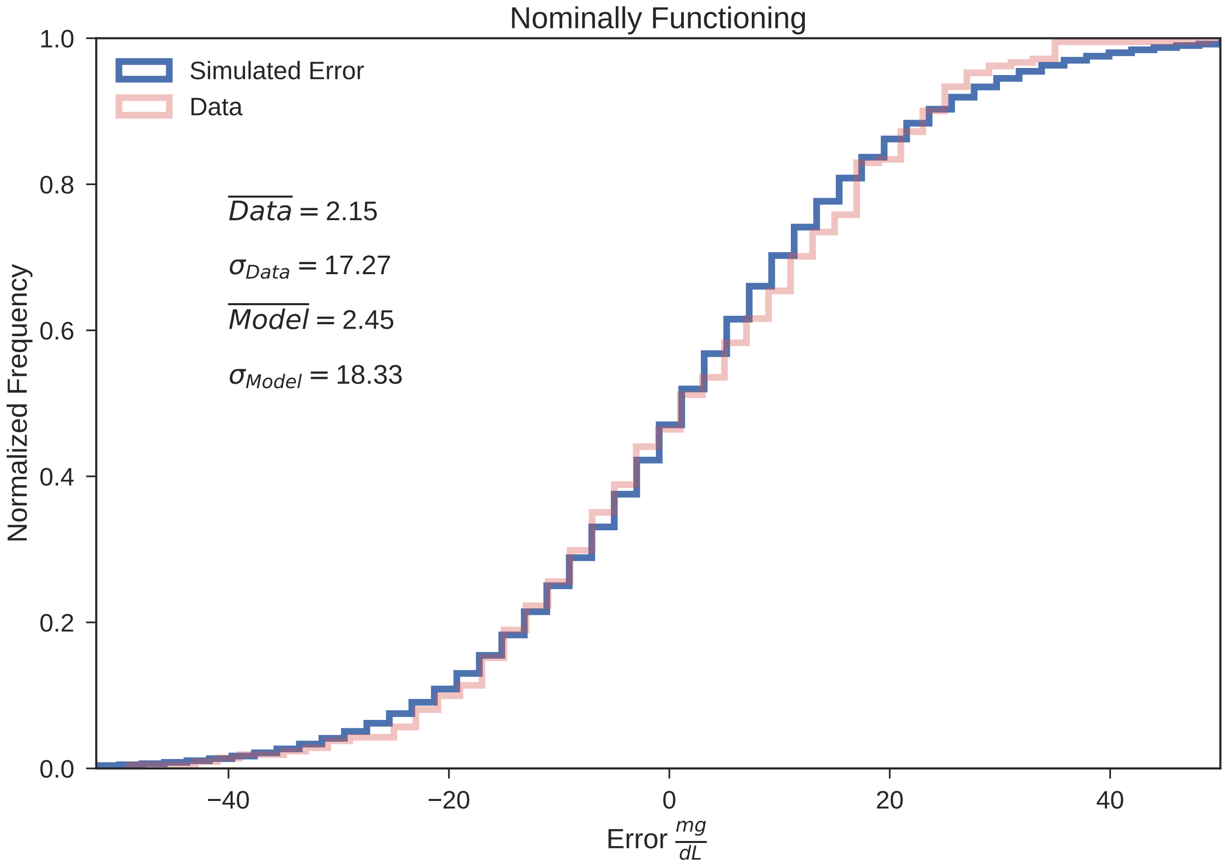


Figure 14: Cumulative histogram showing the qualitative agreement between the observed error calculated as the difference between sparse reference measurements and the corresponding raw CGM data, and simulations (N=1000) of the integrated moving average model (Equation (2.18)) with parameters determined for the nominally functioning subset of CGM data.

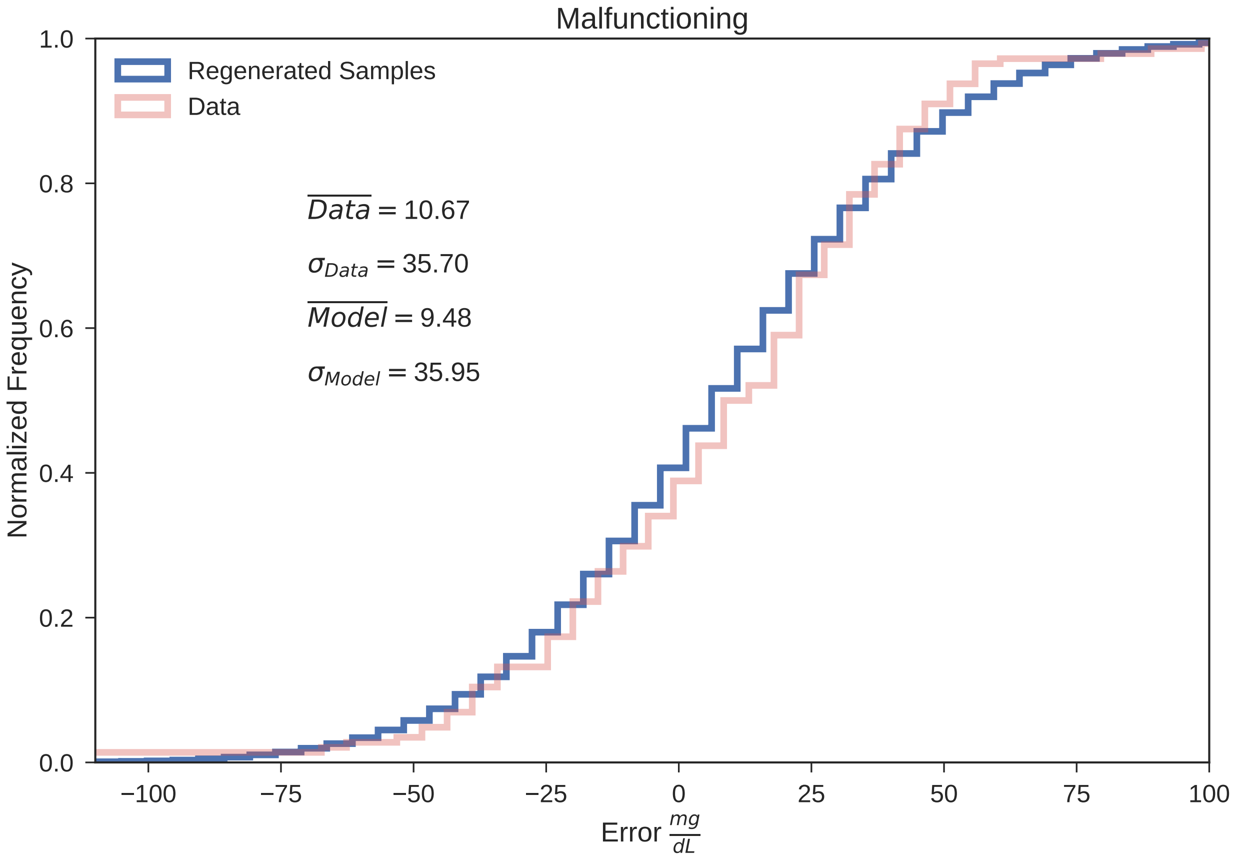


Figure 15: Cumulative histogram showing the qualitative agreement between the observed error calculated as the difference between sparse reference measurements and the corresponding raw CGM data, and simulations of the integrated moving average model (Equation (2.18)) with parameters determined for the malfunctioning or faulty subset of CGM data.

Table 5: Mean and standard deviations of the observed data compared to 1000 simulated trajectories via an integrated moving moving average model with offest for both the nominally functioning and malfunctioning CGM groups

		$\mu\ \left(\frac{mg}{dL}\right)$	$\sigma\ \left(\frac{mg}{dL}\right)$
Functioning	Observed	1.825	17.92
	Simulated	2.343	18.18
Malfunctioning	Observed	10.63	35.82
	Simulated	9.285	35.75

for higher order models and only the "best intervals, that is intervals most well-described by an $ARMA(0, 2+)$ model are included. The deviation observed between the model and reference data in Figure 15 may suggest that a more narrow non-normal distribution is more appropriate to describe the variance of the random effects in the moving average model.

The distributions of parameters for both the nominally functioning and malfunctioning CGM groups are shown in Figure 12, with relevant statistics given in Table 3. Disregarding bias from these parameters and distributions for the moment, as it is purely additive post-hoc, it is apparent that both the nominally functioning and malfunctioning CGM groups can be described by an $MA(1)$ process with, for all practical purposes, an identical moving average coefficient (θ). The differences between them must result from the significant difference in the variance of the white noise (σ_ε^2). This is not necessarily intuitive, but upon further consideration of the calibration and operation of CGM it could be expected, as explained below.

The calibration procedure for the Dexcom® Platinum™ G4 continuous glucose monitor is proprietary, but patent literature suggests that the CGM employs a linear calibration curve calculated via a weighted least-squares regression over (at most) the previous six calibration points where recent calibrations may be weighted more heavily [158]. As such the calibration curve is sensitive to miscalibrations or erroneous calibration measurements.

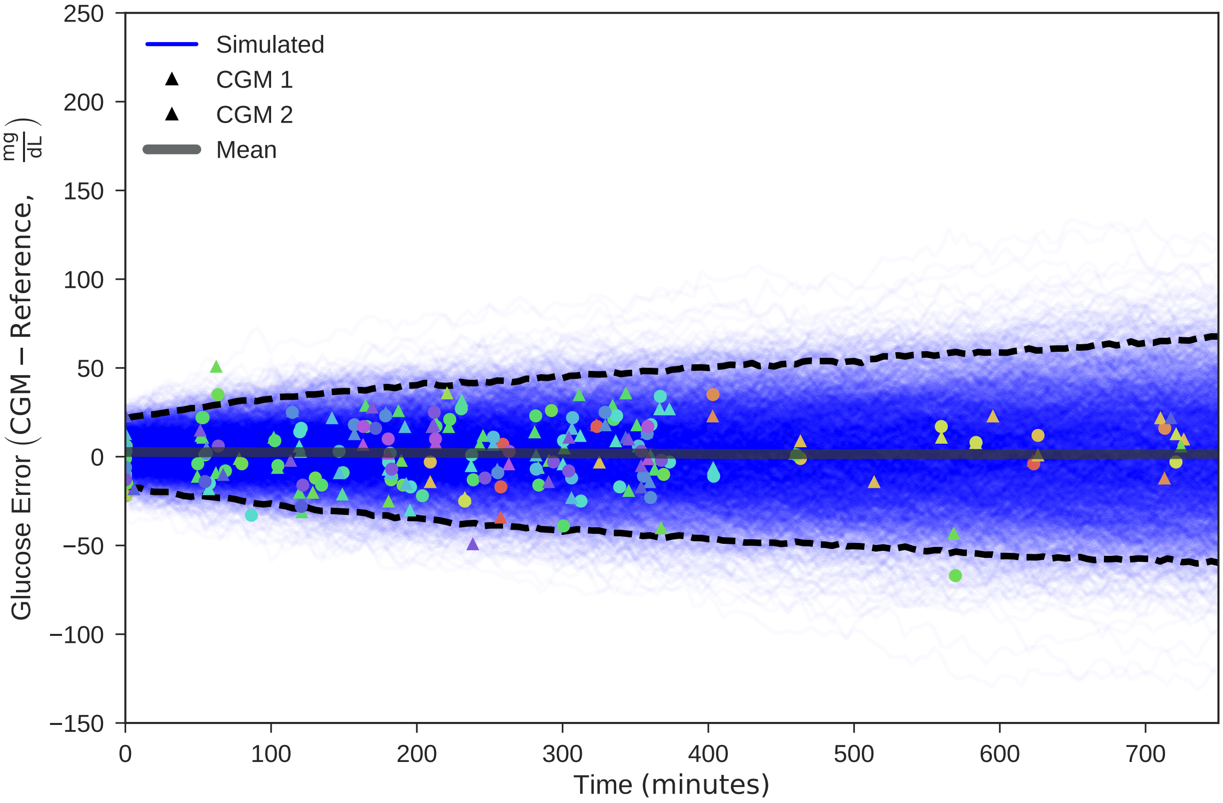


Figure 16: The error in nominally functioning CGMs is simulated over 700 minutes and compared to the observed error trajectories. The dashed black lines denote the 95th percentiles of the simulated data. The color of the reference measurement corresponds to individual patients (all reference measurements from a certain patient will have the same color).

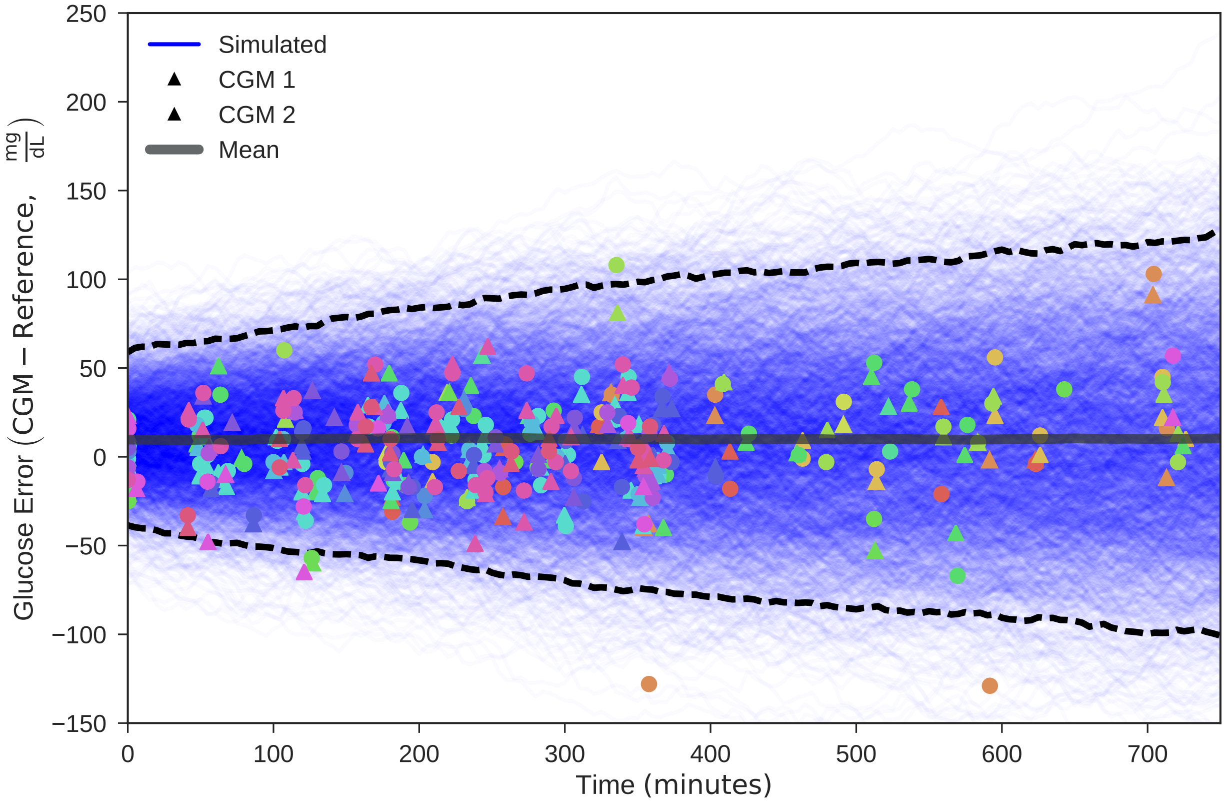


Figure 17: The error in malfunctioning functioning CGMs is simulated over 700 minutes and compared to the observed error trajectories. The dashed black lines denote the 95th percentiles of the simulated data. The color of the reference measurement corresponds to individual patients (all reference measurements from a certain patient will have the same color).

Table 6: The mean AIC over all intervals fit with an $ARMA(p, q)$ model is shown for models with $p \in \{0, 1, 2\}$ autoregressive terms and $q \in \{0, 1, 2\}$ moving average terms for both the nominally functioning and malfunctioning CGM data.

	q p	0	1	2
Functioning	0	–	312.20	319.14
	1	306.72	310.42	305.01
	2	306.22	311.38	312.06
Malfunctioning	0	–	416.83	414.88
	1	409.88	410.81	437.85
	2	409.63	410.29	419.15

A new calibration measurement with significant error compared to the true blood glucose concentration will be given a greater weight than older, possibly more accurate, calibrations, potentially resulting in a significant shift in the slope of the calibration curve. As a result, all succeeding glucose concentrations will be measured with a faulty gain. This leads to significantly more error as glucose concentrations move away from the calibration point. This implies that the observed trend in any of the error trajectories is simply the magnification of error as glucose concentrations naturally vary in response to pancreatic insulin secretion, meals, and a host of other physiologic phenomena. The stochasticity of the trajectories is the result of the magnification of random noise from an erroneous calibration curve. Ultimately, this effect would manifest in the variance of the process with faulty calibrations leading to a significant magnification in the variance of the error as observed here. Because the point-to-point changes in the error are simply the result of random noise and natural blood glucose fluctuations, the dynamics of the process would not be expected to change; hence, it is observed that an $MA(1)$ process with a single θ, but different σ_ε^2s, is able to capture and recapitulate error between the two groups equally well.

Poor calibration does not affect only the slope; it is also used to set the intercept or offset of the calibration curve. Poor calibration, especially at early time points, may therefore also result in continuous bias or offset in the CGM measurements. This is apparent in the distributions for the biases (Figure 13 and table 4), which have a positive mean and significantly greater spread over intervals of malfunctioning CGM data. Due to the extreme difference in the distribution of the biases between the nominally functioning and malfunctioning CGM groups, it is easy to attribute most of the error to faulty or erroneous calibration measurements. However, calibrating the CGM during periods of rapid blood glucose change is also known to result in significant error [159]. This is because interstitial glucose concentrations, as measured by the CGM, lag blood glucose measurements, as measured by a capillary fingerstick, due to the transport time from the blood to interstitium. During rapid changes in blood glucose, this effect becomes most pronounced and interstitial concentrations may lag significantly behind the measured blood glucose concentration, thereby introducing a source of error in the calibration curve. As such, calibration should only be performed during periods of stable blood glucose concentrations, during which plasma and interstitial glucose concentrations will have equilibrated. As a result, it would be expected that the data from malfunctioning CGMs should exhibit greater slopes in blood glucose and greater blood glucose variability preceding calibration compared to data from the nominally functioning CGMs. This phenomenon is in fact seen, as shown in Figure 18, although the magnitude of the effect may be debatable. From this we may ultimately conclude that a combination of erroneous calibration measurements and calibration at inappropriate times combine to produce the observed CGM error, with erroneous calibration appearing as the primary contributor.

Not discussed previously is the potential for drift in the CGM gain. We would expect any drift to be the result of physical processes such as degradation of the sensing element or biofouling [68]. Because the physical processes leading to CGM drift are expected to be consistent, the drift can be considered a component of the error dynamics as governed by the moving average parameter (θ) and therefore conserved between the CGM groups.

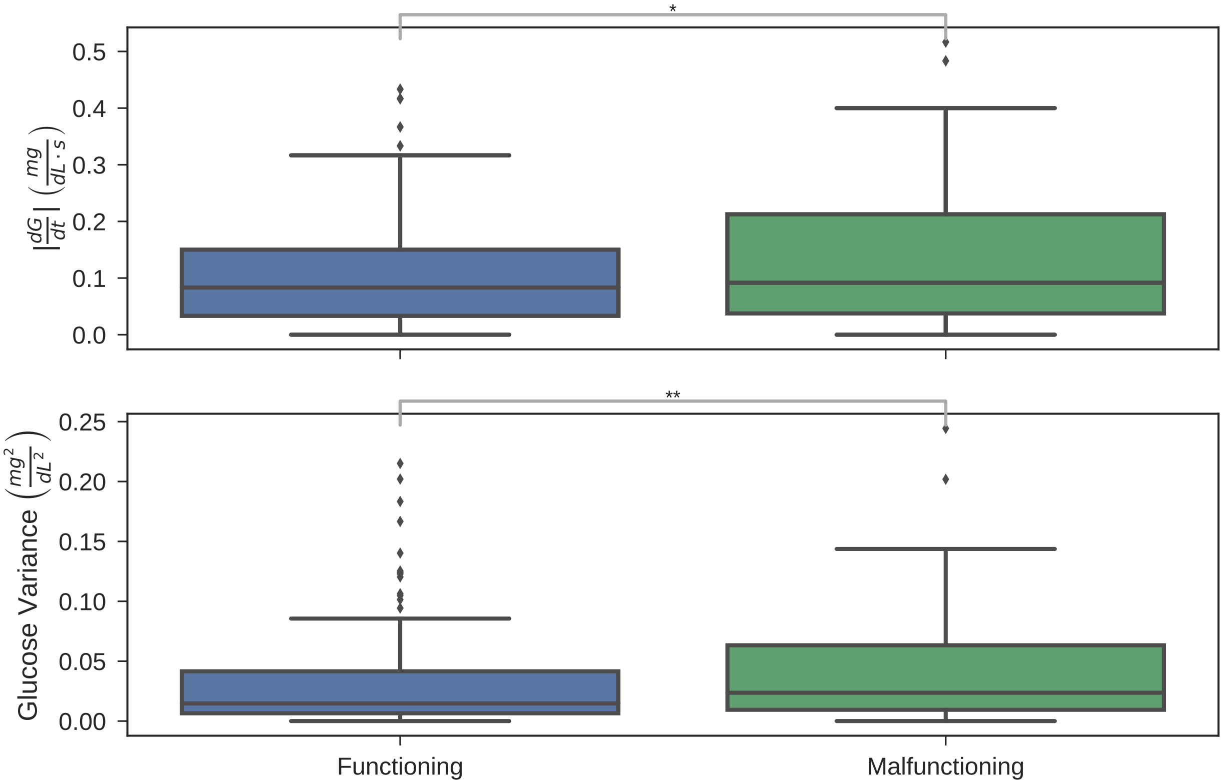


Figure 18: (Top) Glucose rate of change in the 30 minutes preceding a CGM (re)calibration. (Bottom) Blood glucose variability in the 30 minutes preceding (re)calibration. Significance (via Mann-Whitney U Test) is indicated with $*$ at the following levels: $- =$ no significance, $* = 95\%$, $** = 99\%$, $*** = 99.9\%$, $**** = 99.99\%$

2.4 SUMMARY

A model-based technique has been developed that corrects raw CGM data according to reference measurements of varying confidence in a piecewise manner by removing a static bias and allowing for the potential of a linearly varying drift in the gain of the sensors. The approach is able to retrospectively correct the CGM data to recapitulate reference measurements from multiple sources primarily by removing a constant offset. In general, corrections to CGM gain are small and the few cases where the gain correction factors ($\lambda(t_i)$) grow large represent extreme CGM faults such as improper calibration. Fitting a model of glucose-insulin dynamics to the corrected data results in dynamics that will be shown, in Chapter 3, to be physiologically realistic and consistent with clinical observations and expectations.

Using the glucose concentrations fit via the aforementioned model as surrogate reference measurements, the CGM data is segregated based on MARD into two groups of individuals: one with normal sensor function, and a second with malfunctioning CGMs. A time series model of CGM error was then developed for each group independently, and it was found that a first-order integrated moving average model best described each group. Furthermore, the moving-average processes were characterized by a nearly identical distribution of the moving average parameter but differed significantly in the variance of the white noise term. From these observations and the distribution of the biases of each CGM group (where the bias is calculated as part of the retrospective correction) it was suggested that calibration error may be largely responsible for the differences in error where the malfunctioning CGM group was subject to erroneous calibration measurements or calibration at inappropriate times (i.e. rapid rate of glucose change or high glucose variability).

These results highlight the importance of proper calibration technique. Chapter 5 will use the error model developed here to demonstrate that as long as calibration protocol is followed carefully, with dual CGMs and moving horizon estimation the CGMs generally perform well. This system is capable of measuring blood glucose levels with MARD$\leq$ 10%, which is appropriate for closed-loop control. In addition, the distributions of error between the nominally functioning and malfunctioning CGM groups, as well as the error models developed, will be incorporated into the closed-loop glucose control algorithm for *in silico* trials which simulate realistic clinical conditions (see Chapter 5. They will also be used in Chapter 6 to test algorithms used to identify malfunctioning CGMs thus ensuring patient safety.

3.0 VIRTUAL PATIENTS

3.1 INTRODUCTION

In developing a model-based glycemic control scheme, the dangers associated with even a single incidence of hypoglycemia [22] motivate the necessity of simulated trials and testing of a proposed control algorithm. To facilitate this need the following chapter details the development of a mathematical model, and a corresponding data-driven virtual patient cohort, for *in silico* testing of targeted glucose control schemes. The requirements of a virtual patient cohort are that it is able to capture blood glucose dynamics observed in a wide range of critically ill patients with patient-specific parameterizations and trends which are congruous with physiology. We would expect characteristic differences in the dynamics of insulin sensitivity between patients corresponding to stable (healthier/improving patients) and unstable (acutely ill) patients. Namely, healthy patients, or those with stable blood glucose levels near euglycemia, should have a higher degree of sensitivity to insulin as well as a lower variance in their insulin sensitivity trajectories [160]. Conversely, patients exhibiting stress hyperglycemia and poor blood glucose control would be expected to present with suppressed and highly variable insulin sensitivities concomitant with a proinflammatory ICU state [160]. These characteristically different dynamics should emerge in any clinically-relevant virtual patient cohort. Additionally, a virtual patient should support clinically relevant inputs such as enteral and parenteral nutrition as well insulin administration (both intravenous (IV) and subcutaneous (SQ)). Furthermore, parameter ranges should be consistent with physiology and fixed when supported by literature values.

To this end the virtual patient model developed here is composed by linking the Intensive Control Insulin-Nutrition-Glucose (ICING) model [110], a model of subcutaneous insulin delivery [160], and the insulin-suppressed endogenous glucose production component of the “extended” minimal model [130]. The virtual patient cohort synthesized from this model captures the clinical data of patient response at high measurement density while meeting the criteria outlined above. The resultant virtual patients exhibit a range of both intra- and inter-patient variability that could be expected in a population of critically ill patients. These virtual patients will be used in Chapter 5 to study and test the merits of a proposed ZGC paradigm.

3.2 METHODS AND MODELING

The basis of the virtual patient cohort comes from an existing model of clinically validated glucose-insulin dynamics for critically ill patients. The base model makes certain simplifying assumptions such as a constant basal rate of endogenous glucose production (EGP_b), and a simplistic feedback mechanism modeling pancreatic insulin secretion as suppression from a maximal rate in response to Rising plasma insulin levels. In lieu of this feedback mechanism, Lin *et al.* [110] recommend fitting an endogenous insulin production rate using C-Peptide data as a marker of plasma insulin concentration but unfortunately this data is not available in real-time at present. Fitting this model with patient specific variable insulin sensitivities and rates of non-insulin mediated glucose uptake (p_G) to the corrected high frequency CGM data (Section 2.2.1.4) resulted in variability in insulin sensitivity which defied physiologic expectations. The model only resulted in a good match to the corrected CGM data via numerous and implausible extended periods of complete resistance to insulin. In even the most severely type 2 diabetic patients complete insensitivity to insulin is not observed [161]. The excessive variability and poor quality of fit, compared to the results of the model creators [110] may be due to differences in data density. The ICING model was developed and validated with hourly blood glucose measurements whereas the CGM data used here is available with a sampling frequency of 5 minutes. To develop a virtual patient cohort with increased physiologic relevance and clinical consistency, in addition to more physiologically-realistic patient-specific profiles for the aforementioned regressed parameters and pancreatic insulin secretion profiles, the assumption of a constant rate of endogenous glucose production will be relaxed. Rapid rates of change in blood glucose concentrations which may only be observed with the high frequency measurements available from the continuous glucose monitors are believed to drive the excessive variability in insulin sensitivity, which not have been originally observed with the lower rate of measurement in the development of the ICING model. The inclusion of an insulin-dependent rate of endogenous glucose production and more realistic time-dependent insulin secretion trajectories distributes the variability realistically into several key contributors and result in a more believable virtual patients. There is not, to the best of our knowledge, a model of insulin secretion that results in a good fit with our high density clinical data and exhibits dynamics consistent with the observed data. In addition, a number of constraints grounded in physiology are used to guide the regression of insulin sensitivity and secretion profiles.

3.2.1 ICING Model

The Intensive-Control-Insulin-Nutrition-Glucose (ICING) model has been previously validated in a critical-care patient population[110] and will be used as the basis for the glucose-insulin dynamics of this virtual patient cohort. Nominally, the ICING model shares a structure with the Bergman minimal model [162] but includes a Michaelis-Menten saturating insulin effect in comparison to the linear response of glucose to insulin seen in the minimal model. This change is significant for a critically-ill population as it suppresses the effect of insulin[163–166] at high concentrations, thereby limiting the extent of insulin-mediated glucose uptake in the presence of insulin resistance. The basic structure and connectivity of the ICING model is depicted and described in Figure 19 and the nonlinear differential equations representing this five-state model are given in Equations (3.1a) to (3.1h)

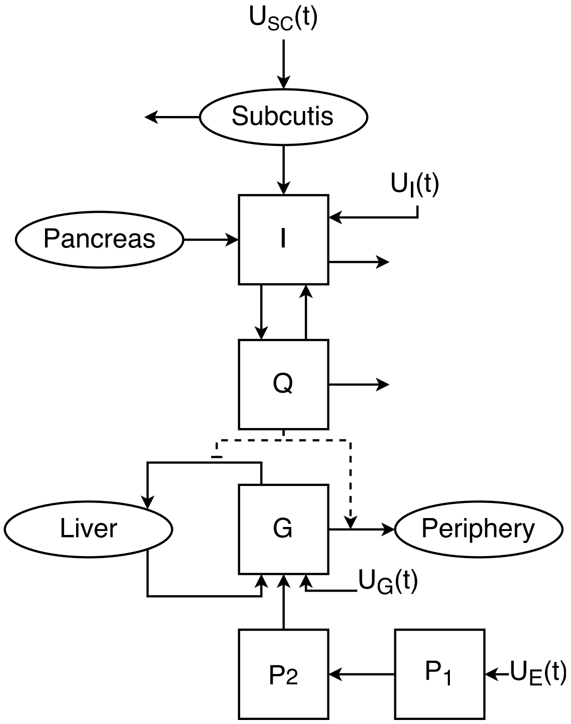


Figure 19: Block diagram depicting the structure of the ICING model for glucose-insulin dynamics. Here, I is plasma insulin, Q_I is interstitial insulin, and G is plasma glucose. Plasma insulin concentrations can change in response to pancreatic insulin production, exogenously administered intravenous insulin ($U_I(t)$) and absorption from the subcutaneous space following subcutaneous insulin administration (see Section 3.2.3). Insulin diffuses between the blood and interstitium and is eliminated via cellular clearance in the interstitium. Glucose is cleared from the blood via an insulin mediated saturating effect on peripheral and hepatic glucose uptake and appears in the blood as a result of endogenous glucose production (EGP) where $\sim$ 95% of (EGP) is found to come from the liver[167]. Additional sources of glucose are exogenous intravenous glucose administration ($U_G(t)$) enteral feeding ($U_E(t)$) where P_1 and P_2 represent gastric emptying and absorption in the gut, respectively.

$$\frac{dBG(t)}{dt} = -p_G BG(t) - S_I(t)\frac{Q_I(t)}{1+\alpha_G Q_I(t)} + \frac{P(t) + EGP_b(t) + U_G(t) - CNS}{V_G} \tag{3.1a}$$

$$\frac{dQ_I(t)}{dt} = n_I(I(t) - Q_I(t)) - n_c\frac{Q_I(t)}{1+\alpha_G Q_I(t)} \tag{3.1b}$$

$$\frac{dI(t)}{dt} = -n_k I(t) - \frac{n_L I(t)}{1+\alpha_I I(t)} - n_I\left(I(t) - Q_I(t)\right) + \frac{u_{ex} + (1-x_L)U_{en}(t)}{V_I} \tag{3.1c}$$

$$\frac{dP_1(t)}{dt} = -d_1P1 + D(t) \tag{3.1d}$$

$$\frac{dP_2(t)}{dt} = -\min(d_2P2, P_{max}) + d_1P1 \tag{3.1e}$$

$$P(t) = min(d_2P2, P_{max}) + PN(t) \tag{3.1f}$$

$$U_{en}(t) = k_1 e^{-I(t)^{k_2}/k_3} \tag{3.1g}$$

$$u_{ex} = U_I(t) + k_{TR}Q_{2,r} + k_{TR}Q_{2,f} + U_{SC,s}(t) \tag{3.1h}$$

Equations (3.1a) to (3.1c) describes the effect of previously infused insulin over time in the interstitium (Q_I) on blood glucose (BG) and the transport of plasma insulin (I) into the interstitium. Equations (3.1d) to (3.1f) govern the appearance of glucose in the plasma from the gut ($P(t)$). P_1 represents glucose in the stomach and P_2, the intestine resulting from oral or parenteral glucose intake ($PN(t)$) [110]. The insulin secretion rate as originally described in [110] is given by Equation (3.1g). The saturating effect of insulin on glucose uptake is given by the Michaelis-Menten function:

$$\frac{Q_I(t)}{1+\alpha_G Q_I(t)}$$

in Equation (3.1a) and is used to represent the binding of insulin to GLUT4 receptors on the cell surface driven by interstitial insulin concentrations. This term appears again in Equation (3.1b) to represent the degradation of receptor bound insulin[168]. A final saturating effect is given by:

$$\frac{I(t)}{1+\alpha_I I(t)}$$

which describes the lumped flux of insulin from the plasma to account for the many various routes of insulin clearance which are not explicitly counted for here. $k_{TR}Q_{2,r}$ and $k_{TR}Q_{2,f}$ in Equation (3.1h) descrbe the rate of appearance of insulin in the plasma following the absorption of subcutaneously administered regular- and fast-acting insulin analogs per the kinetics given by Equations (3.2) to (3.4). $U_{SC,s}(t)$ gives a constant rate of plasma appearance of insulin following a subcutaneous bolus of a long-acting insulin analog.

3.2.2 Glucose and Medical Record Data

The virtual patient cohort detailed herein was developed using high frequency (5 minute) measurements of glucose collected using two Dexcom® Platinum™ G4 continuous glucose monitors per patient. The data from each CGM were independently corrected via the procedure detailed in Section 2.2.1.4, before use here, to enhance the accuracy with respect to reference capillary fingerstick and blood chemistry analyzer measurements.

In addition to the corrected glucose data, amounts and rates of glucose infusion and insulin infusion were obtained from a manual search of the UPMC electronic medical records for each patient. Information pertaining to enteral nutrition (meals and continuous tube feeding) was also obtained from the medical records. For meals, the type (breakfast, lunch, dinner, juice), time, and percentage consumed were specified and for tube feeding the feeding rate in mL/hr was specified along with the solution type enabling conversion to an equivalent rate of glucose appearance in the gut. Because the nutritional content of meals was not specified, the equivalences specified in Table 7[92, 169] were used. These medical record data were included in the virtual patient fitting in order to isolate, to the greatest degree possible, insulin sensitivity and insulin secretion dynamics from exogenous inputs. This is important for simulations of TGC methodologies so that virtual patient responses can be decoupled from the original clinical protocol under which they were treated.

3.2.3 Subcutaneous Insulin Administration Model

Subcutaneous insulin administration is less invasive than intravenous delivery and preferentially employed in patients deemed stable enough to be taken off intravenous insulin drips. The medical record data used here included both continuous infusions and bolus administrations of slow-release, regular- and fast-acting insulin analogs. Regular insulin is standard human insulin whereas the fast-acting analog is modified to be absorbed more readily from

Table 7: Glucose equivalents for three standard meals and oral fluids along with the corresponding rate of appearance in the gut. The rate of appearance is based on the assumption that meals are consumed over the course of an hour and juice over twenty minutes.

Meal	Glucose Equivalent (g)	Rate $\left(\frac{mg}{min}\right)$
Breakfast	40	666.67
Lunch	40	666.67
Dinner	70	1166.67
Oral Fluids (Juice)	14.2	710.0

the subcutaneous injection site and, therefore, to be available for insulin-mediated glucose disposal more rapidly. The long-acting analog is designed to be slowly released over a period of up to 24 hours to mimic the response of basal pancreatic release.

These various insulin analogs have very different pharmacokinetic profiles after subcutaneous administration which must be considered in order to maximally isolate patient-specific, time-varying parameters from exogenous effects in the virtual patient cohort. To this end a linear three-compartment model of subcutaneous insulin absorption is employed[160] to describe the absorption kinetics of regular- and fast-acting insulin. The model as used here results from the reduction of an existing model of subcutaneous insulin absorption[78] and captures the absorption kinetics of regular- and fast-acting insulin through a difference in parameterization. The model structure and differential equations describing regular- and fast-acting insulin analogs are shown in Figure 20 and table 8.

The parameterization of Equations (3.2) to (3.4) for regular and fast-acting analogs is given in Table 8. This model structure does not have a parameterization for long-acting insulin, so instead long-acting analogs are assumed to result in a constant rate of appearance in the blood over 20 hours.

$$\frac{dQ_{SC}(t)}{dt} = U_{SC}(t) - k_{SC}Q_{SC}(t) \tag{3.2}$$

$$\frac{dQ_1(t)}{dt} = k_{SC}Q_{SC}(t) - k_E Q_1(t) - k_{TR}Q_1(t) \tag{3.3}$$

$$\frac{dQ_2(t)}{dt} = k_{TR}Q_1(t) - k_{TR}Q_2(t) \tag{3.4}$$

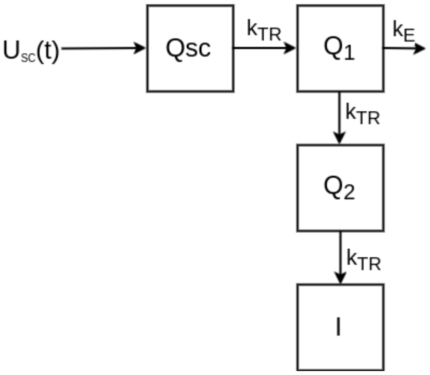


Figure 20: Block diagram depicting the absorption of subcutaneously administered insulin ($U_{SC}(t)$) from the interstitium (Q_{SC}) through two intermediary compartments (Q_1, Q_2) to the plasma (I).

Table 8: Parameters describing the absorption and appearance in the blood plasma of regular- and fast-acting insulin following a subcutaneous injection. The parameterization of the model is largely identical for the two insulin analogs with the differences in action being described by the rate of elimination from the first intermediary compartment. (see [114] and [170] for a further discussion)

Parameter	Units	Regular	Fast
k_E	min^{-1}	0.0268	0.00682
k_{SC}	min^{-1}	1.08	1.08
k_{TR}	min^{-1}	0.0127	0.127

3.2.4 Insulin Mediated Endogenous Glucose Production

The suppression of endogenous glucose production is taken from [171] and is modeled by Equation (3.5a) and Equation (3.5b).

$$\frac{dQ_L(t)}{dt} = PG_4(I(t) - Q_L(t)) \tag{3.5a}$$

$$EGP(t) = EGP_0\left(1 - k_{EGP}\left[\frac{Q_L^3(t)}{Q_L^3(t) + s_{EGP}^3}\right]\right) \tag{3.5b}$$

Equation (3.5a) describes the transport of plasma insulin to a remote liver compartment, Q_L. This effector modulates the rate of endogenous glucose production via the algebraic saturating Hill function in Equation (3.5b) [171]. In this manner rising plasma insulin levels suppress the production and release of glucose from the liver. This addition along with the inclusion of a description of subcutaneous insulin absorption are depicted in Figure 21 and will be referred to from here on as the extended ICING (eICING) model. s_{EGP} in Equation (3.5b) is the remote liver insulin concentration at which endogenous glucose production is suppressed to a rate half of the lowest possible suppressed value. The maximum rate of endogenous glucose production, EGP_0 in Equation (3.5b), is taken to be EGP_b from [110] at a value of $20.9\frac{mg}{min}$ and PG_4 assumes the value from [171]. The full form of the eICING model can be seen in Appendix A

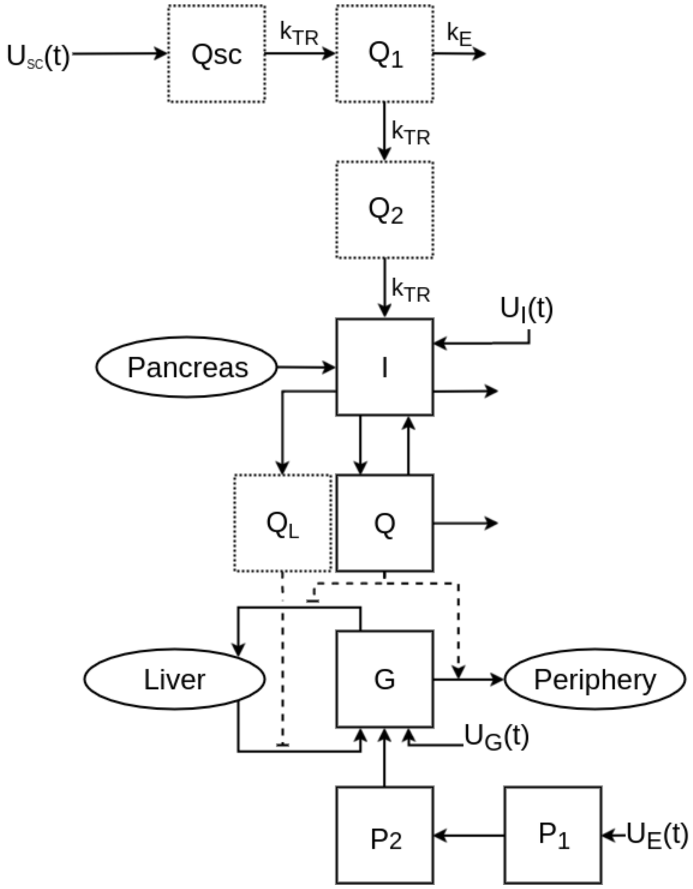


Figure 21: Block diagram depicting the structure of the eICING model. The additions of subcutaneous insulin absorption kinetics and a saturating insulin-mediated endogenous functionality are shown by the dotted lines ($\cdots$).

3.2.5 Glucose Transport Delays

The continuous glucose monitors used in the observation study which generated the glucose data for this work are implanted subcutaneously and therefore measure interstitial glucose concentrations. Due transport dynamics there is a lag between plasma and interstitial glucose concentrations [172–177]. This lag is estimated to be less than 10 minutes [172], and is commonly assumed to be around 5 minutes [175]. This could generally be assumed to result in little difference between plasma and interstitial glucose concentrations during periods of slow glucose changes, however the discrepancy may become significant following rapid plasma glucose fluctuations. As such, this transport delay is added to the ICING model so that it may be fit to interstitial glucose concentrations as measured by subcutaneous continuous glucose monitors as opposed to plasma glucose measurements.

The lag between interstitial and plasma glucose concentrations can be simply modeled by the first order filter given by Equation (3.6) as has been commonly done, previously [178–181].

$$\frac{dG_{ISF}(t)}{dt} = \frac{1}{\tau}(BG(t) - G_{ISF}(t)) \tag{3.6}$$

where $BG(t)$ is the plasma glucose concentration and $G_{ISF}(t)$ is the glucose concentration in interstitial fluid. Assuming a lag time of 5 minutes, τ, the filter time constant becomes 3.15 minutes. Given this addition $G_{ISF}(t)$ is then fit to the CGM data as opposed to $BG(t)$.

3.2.5.1 Regressed Patient-Specific Parameters In addition to $S_I(t)$ and $U_{en}(t)$, the constant parameters n_I and n_C will be regressed individually to each patient. Lin *et al.* [110] identified n_I as a critical parameter governing the quality of model fit. Based on studies indicating that the interstitial to plasma insulin ratio is between 0.4 and 0.6 at steady state the assumption that $n_I = n_C$ was employed in parameterizing the ICING model. In that work, ultimately, the "optimal" values of n_I and n_C were determined via grid search. The formulation of virtual patients as a constrained nonlinear optimization problem allows for relaxation of the assumption that $n_I = n_C$. Here n_I and n_C will be determined through the solution of the nonlinear optimization in Equation (3.9) where the relative values of n_I and n_C bounded by a steady-state constraint.

The value of pG, the insulin-independent rate of glucose uptake in peripheral tissue was established pair-wise with a basal rate of endogenous glucose production (EGP_b) in Lin *et al.* [110]. As EGP in the eICING model introduced here is related algebraically to a remote insulin concentration in the liver, pG is refit as a variable in the nonlinear program given in Equation (3.9).

Because peripheral tissue insulin sensitivity (S_I) is known to exhibit inter- and intrapatient variability, it stands to reason that different patients experience varying degrees of sensitivity in their suppression of endogenous glucose production in response to insulin. As such, s_{EGP} is regressed as a patient-specific constant parameter in our virtual patient cohort.

The constant rate of glucose utilization by the central nervous system (CNS) is allowed to vary in the virtual patient regression between it's bounds of 5.22 $\frac{mg}{min}$ and 6.84 $\frac{mg}{min}$.

Finally, steady state values of plasma and interstitial insulin concentrations (I_{SS} and $Q_{I_{SS}}$, respectively) along with a basal post-absorptive rate of insulin secretion ($U_{en,b}$) are regressed subject to the constraints given in Equations (3.12a), (3.12b) and (3.13). It is worth noting here that $U_{en,b}$ in addition to being a variable is used to regularize insulin secretion rates to a patient-specific basal value (see Section 3.2.6.1) thereby favoring mean-reverting behavior and a return to basal rates in pancreatic insulin production during steady-state conditions.

3.2.6 Virtual Patient Dynamic Optimization

3.2.6.1 Regularization An inverse problem is the process of deducing the causal factors that result in a set of observations. Inverse problems are an archetypal class of ill-posed problems which require additional assumptions such as smoothness of solutions which is introduced here via regularization [182]. Specifically, regularization on changes in the S_I

profile (Equation (3.7a)) results in smoothness in the insulin sensitivity trajectories. A second regularization (Equation (3.7b)) is used to enforce a degree of smoothness in the insulin secretion rate profiles.

$$\Gamma_{S_I} \sum_{i=1}^{N} \left(S_I(t_i) - S_I(t_{i-1})\right)^2, \quad i \in I \tag{3.7a}$$

$$\Gamma_{U_{en}} \sum_{i=1}^{N} \left(U_{en}(t_i) - U_{en,b})\right)^2, \quad i \in I \tag{3.7b}$$

Without the regularization term in Equation (3.7a), the regressed $S_I(t)$ profile would exhibit extreme variability in order to produce an exact match to the clinical data at the expense of expected physiologic behavior resulting in a preponderance of sharp corners. The omission of this regularization term additionally results in extremely tight correlation between the $S_I(t)$ and $U_{I,en}(t)$ profiles as a result of the ill-posedness of the problem. The regularization term of Equation (3.7b) ensures that insulin secretion rates have a mean reverting behavior such that insulin secretion rates tend toward $U_{en,b}$ in a post-absorptive state. Appropriate selection of the magnitudes of the quadratic penalties outlined in Equation (3.7) along with the insulin-mediated suppression of endogenous glucose production (Equation (3.5b)) breaks the strong correlation between insulin sensitivity and insulin secretion rates in this otherwise ill-posed problem. This results in a separation of dynamics between these trajectories and a frequency of oscillation in insulin secretion rates consistent with observed ultradian rhythms [183]. In the nonlinear program (NLP) formulated in Section 3.2.6.2, $U_{en,b}$ is a constant regressed parameter whose value is bounded through the use of a steady-state constraint.

3.2.6.2 Virtual Patient Optimal Formulation

A dynamic optimization approach was employed to regress the virtual patients to clinical data. The differential equations were discretized via a backward finite difference with a five minute time step (corresponding to the native CGM sampling rate). Insulin sensitivity ($S_I(t)$) and the pancreatic insulin secretion rate ($U_{en}(t)$) were discretized on the same timescale. From this discretization, constraints of the form shown in Equation (3.8) can be formulated for all differential equations in the eICING model.

$$\frac{\underset{\sim}{x}(t_i) - \underset{\sim}{x}(t_{i-1})}{\Delta t} :== \frac{d\underset{\sim}{X}(t_i)}{dt} \tag{3.8}$$

The finite differences for all eICING model equations along with the regularization discussed above results in the following optimization problem (Equation (3.9)):

$$\begin{aligned}
\underset{\nu}{\text{minimize}} \qquad & J = \sum_{i=0}^{N} (G_{ISF}(t_i)_{model} - BG(t_i)_{CGM_{a,b}})^2 + \Gamma_{S_I} \sum_{i=1}^{N} (S_I(t_i) - S_I(t_{i-1}))^2 \\
& + \Gamma_{U_{en}} \sum_{i=0}^{N} (U_{en}(t_i) - U_{en,b}))^2 \quad \forall i \in \{0, \ldots, t_f\} && \text{(3.9a)}
\end{aligned}$$

$$\begin{aligned}
\text{subject to:} \qquad & \frac{dx_i}{dt} = \frac{x(t_i) - x(t_{i-1})}{\Delta t} && \text{(3.9b)} \\
& x(0) \geq 0 && \text{(3.9c)} \\
& 0.0 \ \left(\tfrac{L}{mU \cdot min}\right) \leq S_I(t_i) && \text{(3.9d)} \\
& 0.0 \ \left(\tfrac{mU}{min}\right) \leq U_{en}(t_i) \leq 210.0 \ \left(\tfrac{mU}{min}\right) && \text{(3.9e)} \\
& \forall i = 0 ... t_f \\
& 0.5.58 \left(\tfrac{mg}{min}\right) \leq CNS \leq 6.84 \left(\tfrac{mg}{min}\right) && \text{(3.9f)} \\
\text{where:} \qquad & x = \left\{BG, Q_I, I, P1, P2, Q_{sc,(r,f)}, Q_{1,(r,f)}, Q_{2,(r,f)}\right\} \\
& \nu = \{S_I(t_i), U_{en}(t_i), x(0), pG, s_{EGP}, CNS, n_I, n_C, I_{SS}, Q_{I_{SS}}, U_{en,b}, G_{ISF}\}
\end{aligned}$$

The objective function of the nonlinear program laid out in Equation (3.9) seeks to minimize the sum squared error between the CGM data and the eICING model while ensuring smooth and physiologically motivated insulin sensitivity and pancreatic insulin secretion profiles. This objective function is minimized subject to the algebraic finite differences of the eICING differential equations, a non-negative constraint on the initial conditions for all model states and physiologic limits on insulin sensitivity [110]) and rates of pancreatic insulin secretion [184]). The bounds on the rate of glucose uptake in the brain (CNS, Equation (3.9f)) and and the upper bound on the rate of insulin secretion (U_{en}, Equation (3.9e)) are taken from [110] and [185], respectively.

3.2.6.3 Additional Physiologic Constraints

The nonlinear program given above in Equation (3.9) is readily solved, but for certain patients results in solutions counter to physiology and intuition – returning null rates of insulin-independent rates of glucose disposal (IIGD, pG) and extended periods of time characterized by insulin secretion rates at or near the physiologic maximum.

The first physiology-based constraint is used to maintain a physiologically consistent ratio between non-insulin mediated glucose disposal (NIMGD) and insulin mediated glucose uptake (IMGU). Shown in Equation (3.10), the upper bound in this constraint follows from the observation that non-insulin mediated glucose disposal accounts for up to 75% of glucose metabolism in the post-absorptive (3-5 hours following complete digestion and absorption of a meal) state [186]. The lower bound is based on data suggesting that skeletal, adipose and liver tissues are responsible for approximately 83% of glucose disposal in the postprandial (immediately following a meal) state [187]. Because skeletal, adipose and liver tissues are the primary insulin sensitive tissues in the body [188], it stands to reason that even in the postprandial state NIMGD is responsible for approximately 17% of glucose disposal. By limiting the fraction of glucose disposal occurring through non-insulin mediated pathways between the post-absorptive and postprandial extremes this constrain helps to properly set the balance between insulin sensitivity (S_I), the non-insulin mediated rate of glucose uptake (pG) and rates of insulin production (U_{en}).

$$0.17 \leq \frac{1}{N}\sum_{i=0}^{N} \frac{p_G BG(t_i)}{p_G BG(t_i) + S_I(t_i) BG(t_i) \frac{Q_I(t_i)}{1+\alpha_G Q_I(t_i)}} \leq 0.75 \tag{3.10}$$

Despite the regularization introduced in Equation (3.7b), which penalizes deviations in the insulin secretion rate from a basal value it was observed that optimal solutions often resulted in extended periods of insulin production at the maximum allowable rate. This is not believed to be particularly realistic, and it is well known that the pancreas tends to exhibit a biphasic release phenomenon where a maximum rate of secretion is reached only briefly [189–193]. A study by Polonsky *et al.* [194] measured insulin secretion rates over a 180 minute period in normal and obese individuals receiving a hyperglycemic intravenous glucose infusion. Over this 3 hour period normal subjects (body surface area $\approx 1.7m^2$) secreted insulin at an average rate of $44.29 \pm 2.17 \frac{mU}{min}$ while obese subjects (body surface area $\approx 2.2m^2$) secreted insulin at a significantly higher rate of $79.1 \pm 8.4 \frac{mU}{min}$. From the results of this experiment the upper bound on the average insulin secretion rate over any 3 hour period was set at $87.5 \frac{mU}{min}$. This constraint is formalized in Equation (3.11).

$$\frac{1}{180} \sum_{j=t_i}^{t_i+180} U_{en}(t_j) \leq 87.5 \frac{mU}{min} \quad \forall t_i \in \{0, \cdots, t_f\} \tag{3.11}$$

At steady state it is estimated that the ratio of interstitial (Q_I) to plasma (I) insulin is 0.4 to 0.6 [195–197]. This ratio ultimately determines the the value of n_I and n_C as well as the steady state or basal rate of insulin secretion ($U_{en,b}$). To calculate the the steady state plasma and interstitial insulin concentrations the constraints given in Equations (3.12a) and (3.12b) are used where I_{SS} and $Q_{I_{SS}}$ are the steady state plasma and interstitial insulin concentrations, respectively.

$$n_I \left(I_{SS} - Q_{I_{SS}}\right) - n_C \frac{Q_{I_{SS}}}{1 + \alpha_G Q_{I_{SS}}} = 0 \left(\frac{mU}{L \cdot min}\right) \tag{3.12a}$$

$$-n_K I_{SS} - \frac{n_L I_{SS}}{1 + \alpha_I I_{SS}} - n_I \left(I_{SS} - Q_{I_{SS}}\right) + (1 - x_L) \frac{U_{en,b}}{V_I} = 0 \left(\frac{mU}{L \cdot min}\right) \tag{3.12b}$$

Then the ratio of $Q_{I_{SS}} : I_{SS}$ is constrained by Equation (3.13).

$$0.4 \leq \frac{Q_{I_{SS}}}{I_{SS}} \leq 0.6 \tag{3.13}$$

A final constraint on n_I and n_C (Equation (3.14) is taken from [110] and is based on reported half-lives of "effective" insulin between 25-130 min.

$$0.0053 \text{ min}^{-1} \leq n_I + n_C \leq 0.0277 \text{ min}^{-1} \tag{3.14}$$

3.2.7 Numerical Methods

The discretization and constraints formulated in Section 3.2.6.2 were implemented in Python using the Pyomo optimization package [148, 149]. The nonlinear program from Equation (3.9) was solved using the interior point algorithm IPOPT as the nonlinear solver [150] and ma57 from [151] as the linear solver.

To determine the magnitude of the regularization parameters (Γ_{S_I} and $\Gamma_{U_{en}}$) the nonlinear program from Section 3.2.6.2 was first solved on a 2-dimensional grid of possible values for the regularization parameters. At each point the sum-squared error was returned resulting in the construction of an objective function surface spanning many orders of magnitude for regularization parameter values. The resultant surface is shown in Figure 22.

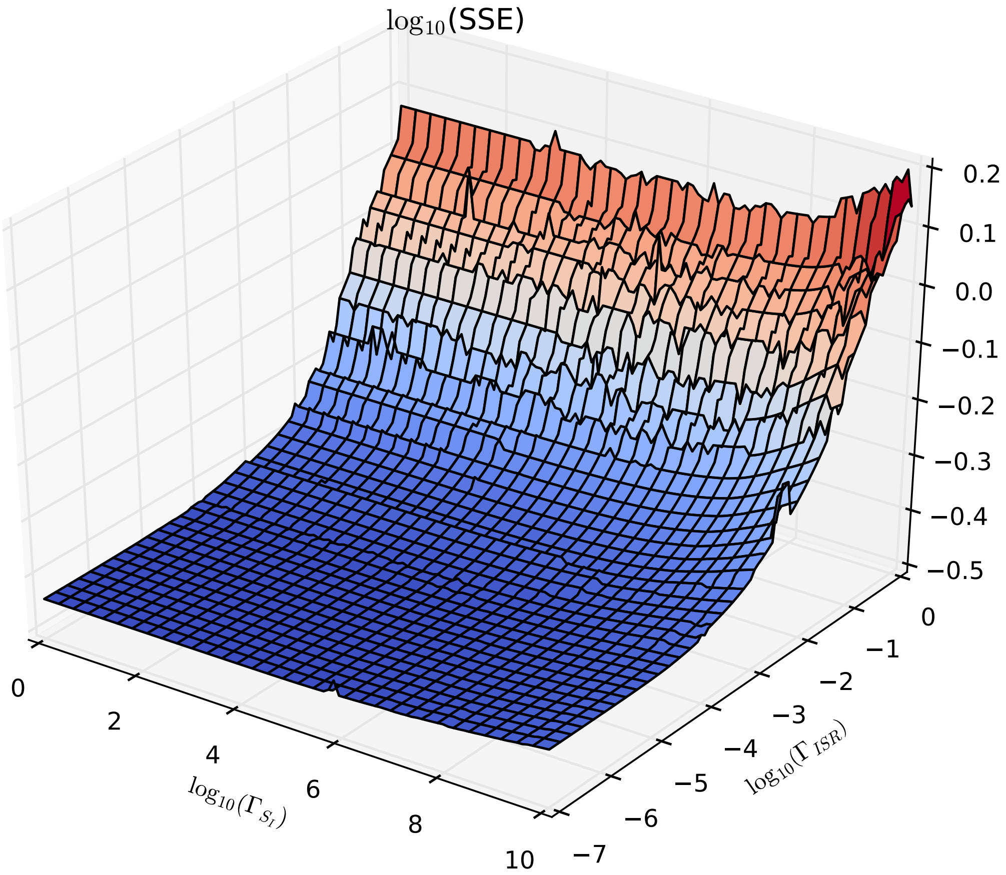


Figure 22: $[\log_{10}(\Gamma_{S_I}), \ \log_{10}(\Gamma_{U_{en}})]$ vs. $\log_{10}(\mathrm{SSE})$

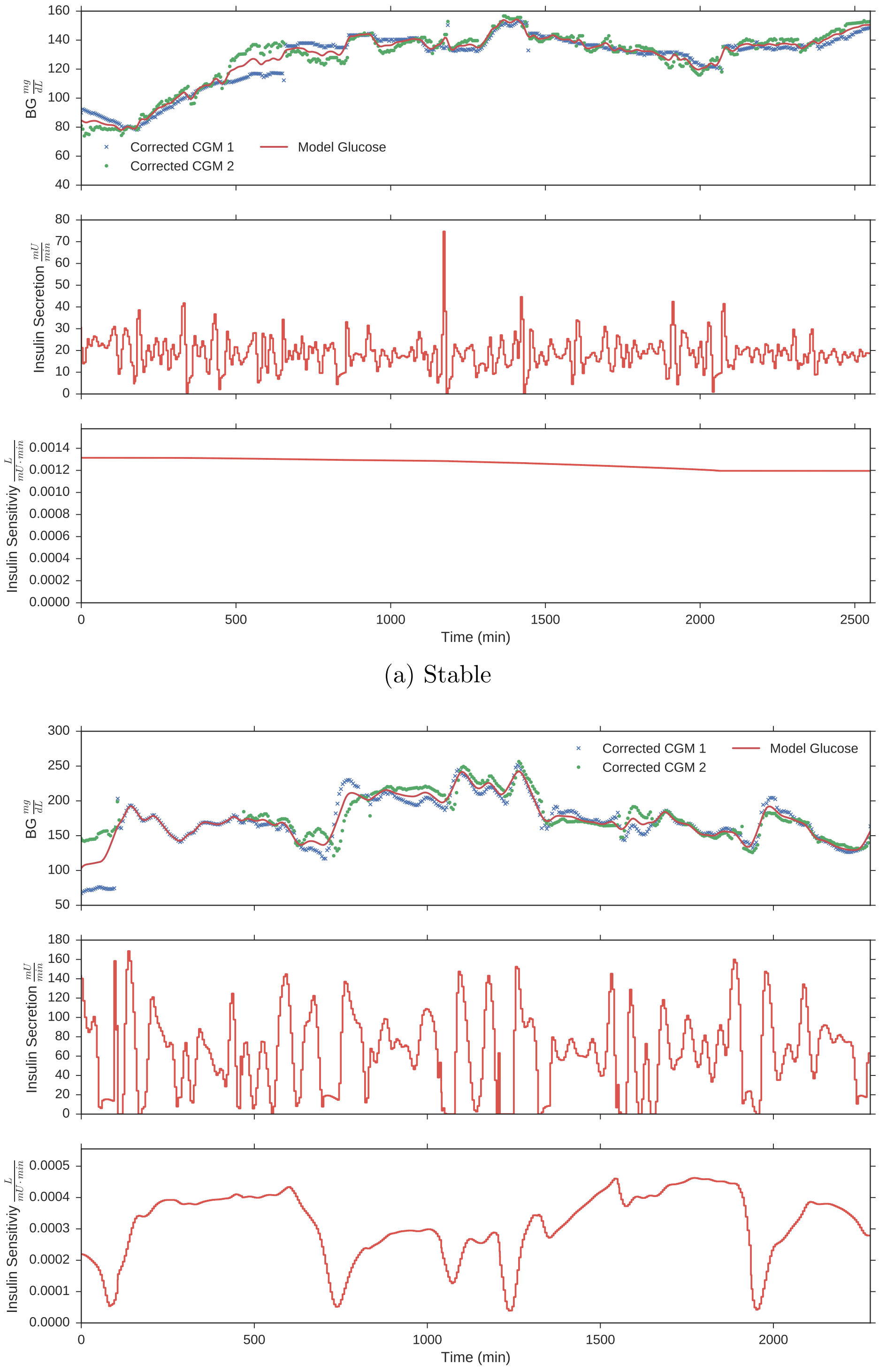
The objective surface shown in Figure 22 indicates that the sum squared error begins to sharply increase at $[\Gamma_{S_I}, \quad \Gamma_{U_{en}}] \approx [1 \times 10^8, \quad 1 \times 10^{-4}]$. Because these values represent the point at which variability and oscillations in the insulin sensitivity and insulin secretion rate profiles can be reduced without significantly increasing the SSE and thereby unduly biasing the solutions, the final virtual patient cohort is taken as the solution to the NLP in Equation (3.9) with $[\Gamma_{S_I}, \quad \Gamma_{U_{en}}] \approx [1 \times 10^8, \quad 1 \times 10^{-4}]$.

3.3 RESULTS

Of 24 patients enrolled in the observational study which provided the data for this work 17 patients had glucose data sets over a time window of at least 24 hours which was sufficient to develop virtual patients. Over these 17 patients, the virtual patients exhibited a mean absolute error per point of 1.79 $\frac{mg}{dL}$. Two representative virtual patients are shown in Figures 23a and 23b. The first virtual patient (Figure 23a) has a relatively stable insulin sensitivity profile and is representative of a healthier patient with blood glucose approaching euglycemic levels. The latter virtual patient (Figure 23b) is characterized by a much higher degree of variability in their insulin sensitivity profile as well as significant hyperglycemia and poor overall control of blood glucose levels. Model fits, insulin sensitivity and secretion profiles are shown for all patients comprising the virtual patient cohort in Appendix B.

The insulin secretion rates associated with both patients in Figures 23a and 23b are oscillatory and are representative of the virtual patient cohort. The amplitude spectral densities (ASD) for the representative virtual patients shown in Figures 23a and 23b are shown in Figures 24a and 24b.

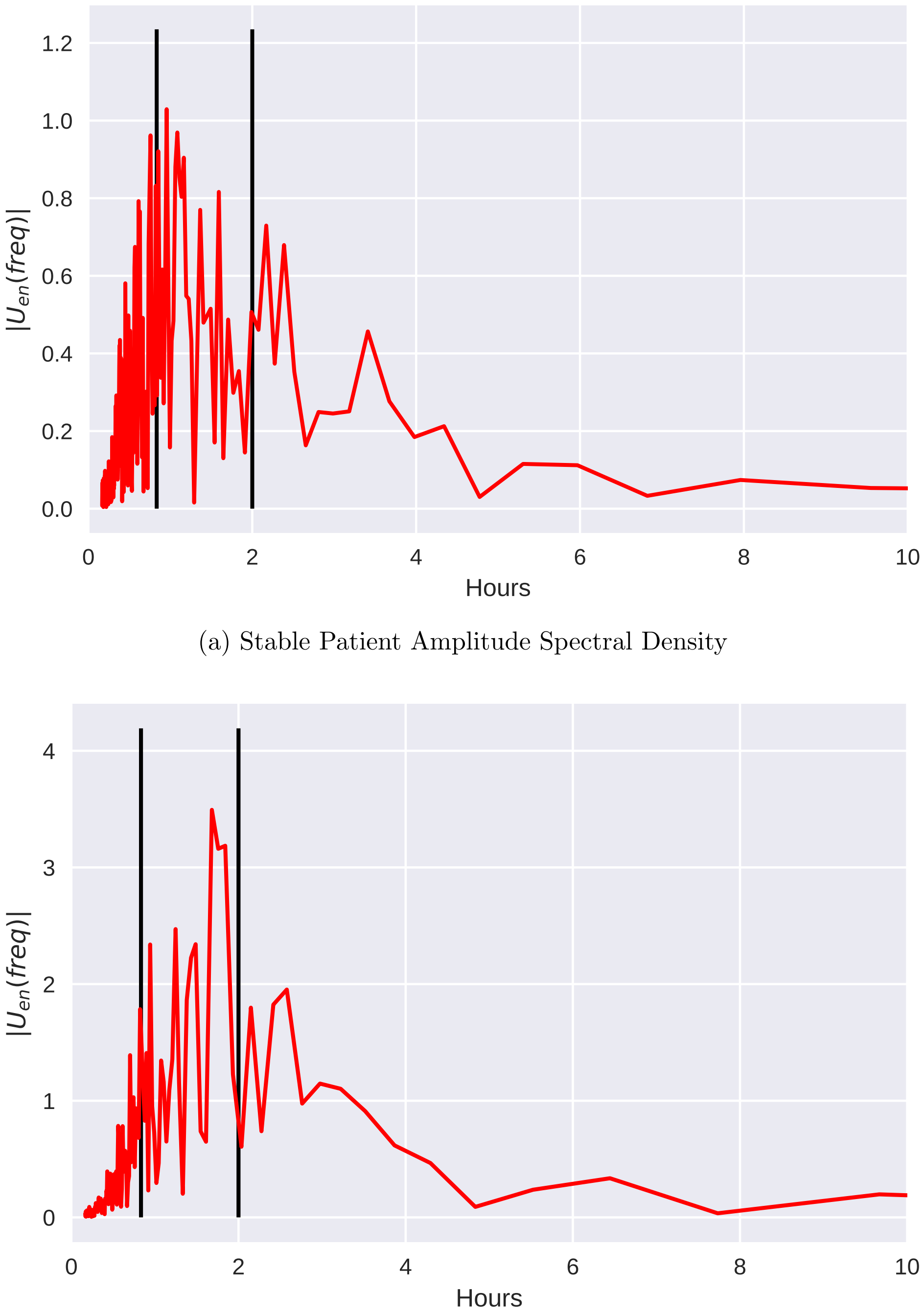
For the ASD of all members of the virtual patient cohort see Appendix C. In general, insulin secretion profiles exhibit a peak in the ASD in the 50-120 minute range consistent with published studies of ultradian insulin oscillations in humans [183, 198] although several virtual patients exhibit oscillation frequencies outside the expected range of the ultradian insulin secretion rhythm which may be indicative of β-cell dysfunction in these patients and loss of glucose control [198].



(a) Stable

(b) Unstable

Figure 23: Representative clinically "stable" (A) and clinically "unstable" (B) virtual patients. **Top Panels:** eICING model glucose concentration (-) fit to clinical CGM data (● and ●). **Middle Panels:** Estimated pancreatic insulin secretion rate ($U_{en}(t)$). **Bottom Panels:** Estimated Insulin sensitivity ($S_I(t)$)



(a) Stable Patient Amplitude Spectral Density

(b) Unstable Patient Amplitude Spectral Density

Figure 24: Amplitude spectral density of Representative clinically “stable” (A) and clinically “unstable” (B) virtual patients. Amplitude spectral density is calculated determined via Fast-Fourier-Transform of the insulin secretion profile after filtering via a high-bandpass Butterworth filter with a cutoff frequency of 5 hours. Vertical black bars indicate physiologic frequency for ultradian insulin secretion oscillations (50-120 minutes) [183, 198].

3.3.0.1 Parameter Distributions The distributions of the regressed static parameters p_G, n_I, n_C, s_{EGP}, $U_{en,b}$, I_{SS} and $Q_{I_{SS}}$ are shown in (half-maximum suppression liver insulin concentration $\left(\frac{mU}{L}\right)$) are shown in Section 3.3.0.1. A table summarizing the virtual patient estimated parameters is shown in Table 9.

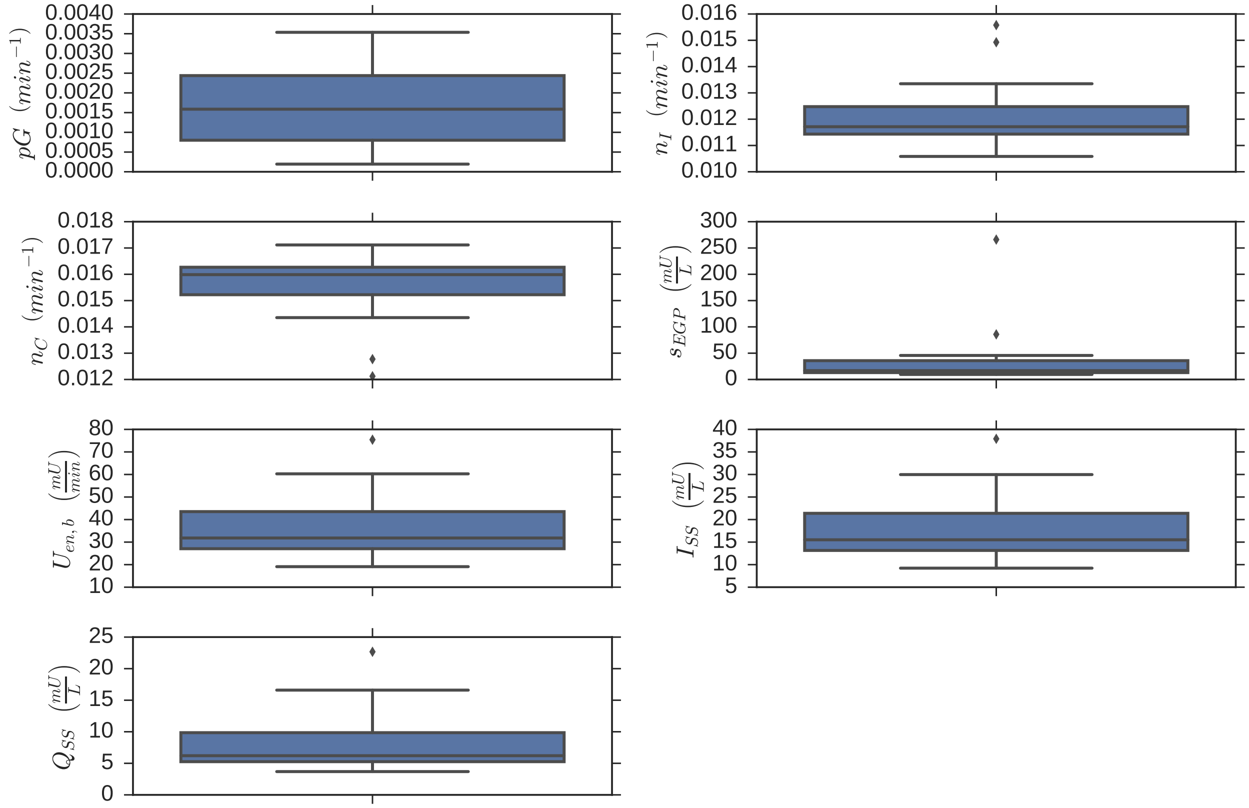


Figure 25: Boxplots showing the parameter distribution across the virtual patient cohort. Box denotes IQR and whiskers denote 1.5 IQR. ♦s denote outliers.

Table 9: Mean (μ), standard deviation (σ), median and range for all regressed parameter across the virtual patient cohort. Constant parameters statistics are calculated between patients. For the time varying parameters intrapatient means are calculated then used for determine interpatient statistics.s

	PG	s_{EGP}	n_I	n_C	$\overline{S_I}$	$\overline{U_{en}}$	$U_{en_{SS}}$	I_{SS}	$Q_{I_{SS}}$
μ	0.00173	39.06444	0.01220	0.01550	0.00044	36.48977	37.67478	18.52343	8.33580
σ	0.00112	61.50377	0.00132	0.00132	0.00029	16.15142	15.89632	8.07137	5.02089
min	0.00019	9.57798	0.01058	0.01212	0.00005	19.09709	19.09894	9.22002	3.68801
50%	0.00159	16.81689	0.01171	0.01599	0.00037	29.88487	31.82060	15.51374	6.20550
max	0.00354	266.00000	0.01558	0.01712	0.00117	75.45382	75.43278	37.91158	22.67296
Constant?	y	y	y	y	n	n	y	y	y

3.4 DISCUSSION

A virtual patient cohort was modeled using high-frequency (5-min) clinical CGM data and a novel synthesis of existing, previously validated models of insulin-glucose dynamics in critically ill patients, insulin-mediated endogenous glucose production and subcutaneous insulin absorption. The resultant virtual patient cohort exhibits variable pancreatic insulin secretion rates and insulin sensitivity profiles in line with physiologic expectation.

One of the main goals in adding an insulin-mediated endogenous glucose production (EGP) mechanism (Equation (3.5b)) was to partially decouple the action of endogenous insulin production and insulin sensitivity. Due to the product of $S_I(t)$ and $Q(t)$ in Equation (3.1a), fitting both dynamic pancreatic insulin secretion rate and insulin sensitivity results in trajectories for U_{en} and S_I which are highly inversely correlated. Without including insulin-mediation in the EGP mechanism, the ill-posedness of the problem is more severe and no differentiation in S_I trajectory shapes across the patient cohort is observed. The addition of a mechanism for insulin-mediated endogenous glucose production provides a structure where pancreatic insulin secretion is able to affect an increase or decrease in blood glucose disposal rates via endogenous glucose production independent of insulin-mediated glucose update. Through the addition of this mechanism, trends in insulin sensitivity trajectories become apparent across the virtual patient cohort and are consistent physiology and previous studies

3.4.0.1 Model Parameterization In general, the model parameterization is well within physical regimes across the virtual patient cohort although some differences from the ICING model [110] arise in this virtual patient cohort. Mainly pG, the non-insulin mediated rate of glucose uptake is, on average, about half as fast as the lower reported in Lin *et al.* [110] which determined a lower bound from an aggregated set of results of the Bergman minimal model fit to a variety of patients [162, 199, 200]. The discrepancy in this rate compared to the virtual patient model described likely results from the fact that the Bergman minimal model supposes a constant rate of endogenous glucose production which is a multiple of the rate of non-insulin mediated rate of glucose uptake (see Equation (3.15)). In contrast, the eICING model presented herein does not treat glucose production as a multiple of p_G and further more the virtual patient cohort incorporates a mechanism for suppression of endogenous glucose production by insulin. These changes to the model structure ultimately lead to an unsurprising and unconcerning decrease in p_G compared to the minimal model.

$$\frac{dBG(t)}{dt} = -(S_G + X(t))BG(t) + S_G G_b \tag{3.15}$$

where

$BG(t)$: Plasma blood glucose concentration $\left(\frac{mg}{dL}\right)$

$X(t)$: Remote insulin concentration $\left(\frac{mU}{L}\right)$

$S_G(t)$: Non-insulin mediated rate of glucose uptake (also referred to as glucose effectiveness in Bergman, min^{-1})

Furthermore, it has been shown [199] that the basic model structure used in the minimal model, the ICING model and the eICING model of this work is insensitive to changes in p_G (or S_G) with respect to regressed insulin sensitivities. The net result is a mitigation of any concerns about the discrepancy in p_G between the virtual patient cohort developed here and previous work.

Despite the discrepancy in p_G values discussed above, the mean insulin sensitivity across this virtual patient cohort ($.44 \times 10^{-3} \quad \frac{L}{mU \cdot min}$) is in good agreement with the value found by Lin *et al.* ($.31 \times 10^{-3} \quad \frac{L}{mU \cdot min}$). Furthermore, the interquartile ranges for this virtual patient cohort ($.22 \times 10^{-3} - .60 \times 10^{-3} \quad \frac{L}{mU \cdot min}$) is consistent with that reported by Lin *et al.* ($.20 \times 10^{-3} - .48 \times 10^{-3} \quad \frac{L}{mU \cdot min}$). The values of n_I and n_C were tightly distributed and correspond to an "effective" insulin half-life of approximately 25 minutes which is consistent with the range of values reported in the literature [164, 201, 202].

The mean basal insulin secretion rate, $U_{en,b}$, fit across the virtual patient cohort was $37.67\frac{mU}{min}$ which is approximately 2× the commonly accepted basal secretion rate in healthy individuals [194, 203–206]. This elevated basal rate of insulin secretion may either be pathogenic or symptomatic of stress hyperglycemia and insulin resistant, but it is consistent with the pancreatic insulin functionality proposed by Lin *et al.* in [110].

Lin *et al.* in [110] proposed the algebraic relationship between plasma insulin concentrations and insulin secretion given in Equation (3.16).

$$U_{en} = k_1 e^{\frac{-I(t)^{k_2}}{k_3}} \tag{3.16}$$

where

k_1 : 45.7 $\frac{mU}{min}$

k_2 : 1.5

k_3 : 1000.0

Across the virtual patient cohort developed using the eICING model detailed here the mean plasma insulin concentration ($I(t)$) was 36.49 $\frac{mU}{L}$. Using this value in Equation (3.16) gives an average rate of insulin secretion of 36.66 $\frac{mU}{min}$ which is very close to the average secretion rate across the virtual patient cohort resulting from the solution to the NLP of Equation (3.9) with a completely non-parametric treatment of S_I and U_{en}. This is the result that would be expected in the limit of extreme regularization resulting in zero variability in insulin secretion rates over time. Despite the agreement in the limit of a constant rate of insulin secretion, the pancreatic insulin functionality shown in Equation (3.16) was not employed here as it was found not to have the required dynamic range to capture the variability in blood glucose concentrations in a large portion of the virtual patient cohort. This is due to the fact that U_{en} in Equation (3.16) is suppressed only by rising insulin concentrations which are affected only by U_{en} itself and exogenous action. As such U_{en} given by Equation (3.16) quickly reaches steady state and remains unaffected by blood glucose concentrations. There is a body of evidence suggestive of autocrine negative-feedback of insulin on insulin secretion rates as results in the ICING model with U_{en} described by Equation (3.16) [207, 208], however there is still a debate as to reality of this mechanism [209]. Furthermore, it is well understood that the primary feedback loop in the pancreas is driven by glucose concentrations [210, 211] – a mechanism which is precluded by Equation (3.16).

In general, the half-maximum suppression concentration of insulin in the liver (s_{EGP}) is tightly distributed around a median value of 16.82 $\frac{mU}{L}$ with a small number of virtual patients exhibiting a significantly higher value. This may be suggestive of a breakdown in the inhibitory mechanism of insulin on endogenous glucose production in certain patients which may result in or exacerbate stress hyperglycemia.

At steady-state the average insulin concentration across the virtual patient cohort would be 18.5 $\frac{mU}{L}$. This is consistent with reported fasting insulin concentrations in the literature with a range of $5.0 - 45.0$ $\frac{mU}{L}$ [212–215]. The steady-state concentrations of interstitial insulin are then tightly governed by Equation (3.13).

3.4.1 Physiologically Consistent Insulin Sensitivity Profiles

Insulin sensitivity in critically-ill patients is often suppressed leading to elevated blood glucose levels. In a physiologically consistent virtual patient cohort we would expect this trend to be borne out. Figure 26 demonstrates that in general there is a weak trend which emerges in our virtual patient population.

Another hallmark of stress hyperglycemia in critically-ill patients is a highly variable sensitivity to insulin reflecting the magnitude of the accompanying inflammatory response and the administration of certain drugs, such as catecholamines. In a clinically-relevant virtual patient cohort, especially one used to explore methods for TGC, we would expect this trend to emerge. Figure 27 shows that increased variability in insulin sensitivity is indeed associated with higher average blood glucose levels consistent with a pro-inflammatory state and stress hyperglycemia.

The data shown in Figures 26 and 27 however fails to account for exogenous infusions of insulin. From the medical record data it is known that many of the patients used to develop this virtual patient cohort received significant exogenous infusions of insulin while in critical care. Despite the potential for insulin resistance and associated hyperglycemia we would expect large amounts of exogenously administered insulin to negatively affect plasma glucose concentrations. As such, it is intuitive to reason that if two patients (a and b) have similar mean plasma glucose concentrations yet patient a received significantly more

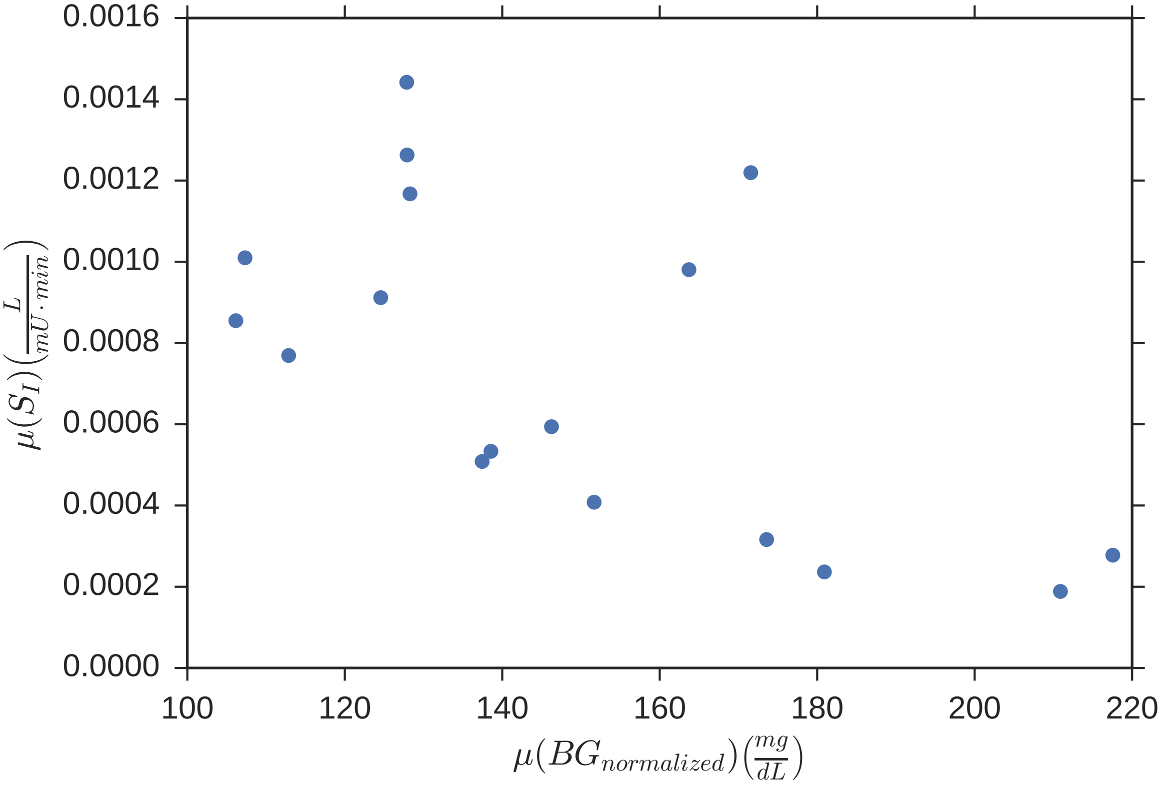


Figure 26: Mean blood glucose vs. mean insulin sensitivity: an inverse relationship is observed between average insulin sensitivity and average blood glucose across our virtual patient cohort. This is consistent with clinical observations, which suggest that insulin resistance is a primary driver of stress hyperglycemia in critically-ill patients.

exogenous insulin, then patient a would be expected to have a lower sensitivity to insulin than patient b. To this end, a weighting based on exogenous insulin infusions should be used to compare average plasma glucose values with insulin sensitivity values and variability. The weighting chosen here normalizes mean blood glucose concentrations by the average hourly rate of exogenous insulin administration. The hourly average is used to correct for the fact that patients have varying lengths of stay in the ICU. (see Equation (3.17).

$$\mu(BG)_{normalized} = \mu(BG)\frac{1}{1+\int_{t=0}^{T} U_{ex}dt} \tag{3.17}$$

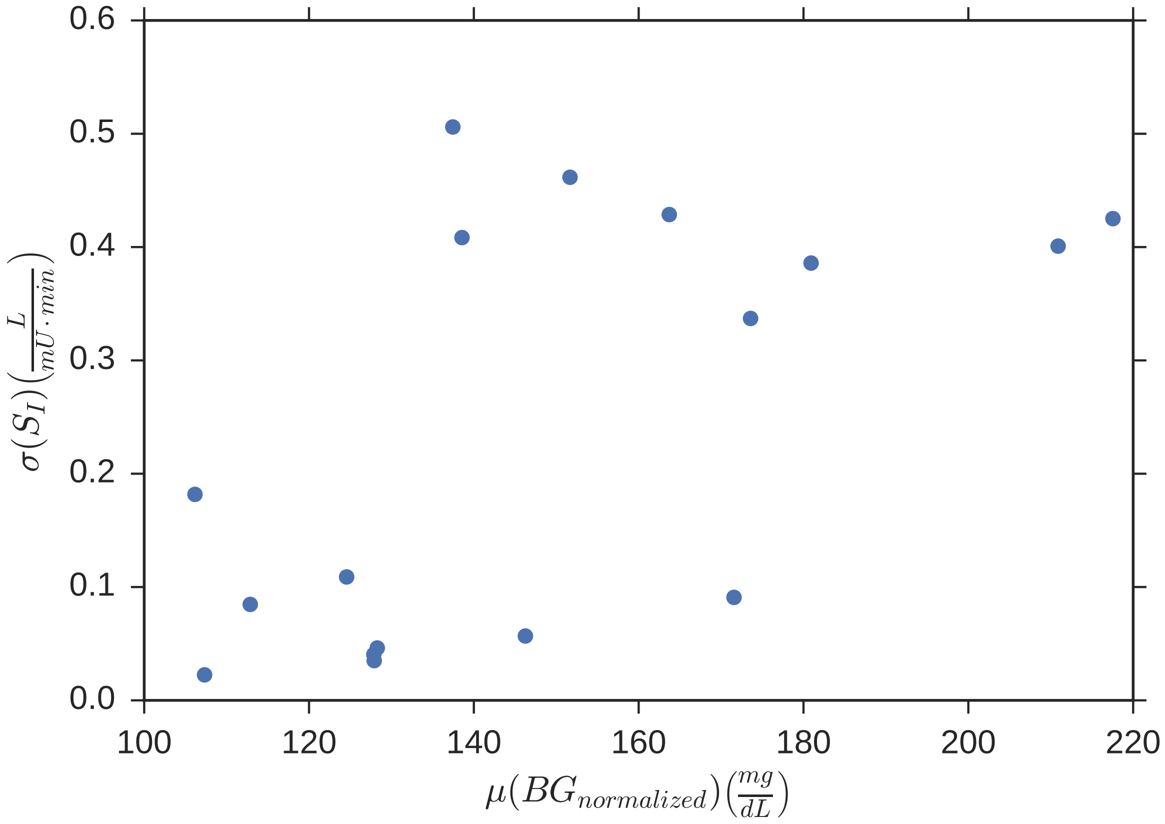


Figure 27: Mean blood glucose vs. mean normalized insulin sensitivity variance: the positive correlation between average blood glucose and the variability in insulin sensitivity across our virtual patient cohort corroborates the clinical expectation that patients exhibiting symptoms of stress hyperglycemia may have significant variability in insulin sensitivity over time.

where T is the total time of stay in hours. The form of the normalization in Equation (3.17) is used to prevent all patients receiving no exogenous insulin from having identically zero normalized mean blood glucose concentrations.

Blood glucose normalized in this manner is shown in comparison with mean insulin sensitivity and mean normalized insulin sensitivity variance in Figures 28 and 29.

In Figure 28 an inverse relationship is observed between average insulin sensitivity and normalized average blood glucose across our virtual patient cohort. As a result of the normalization (Equation (3.17)) patients receiving significant exogenous insulin infusions have a

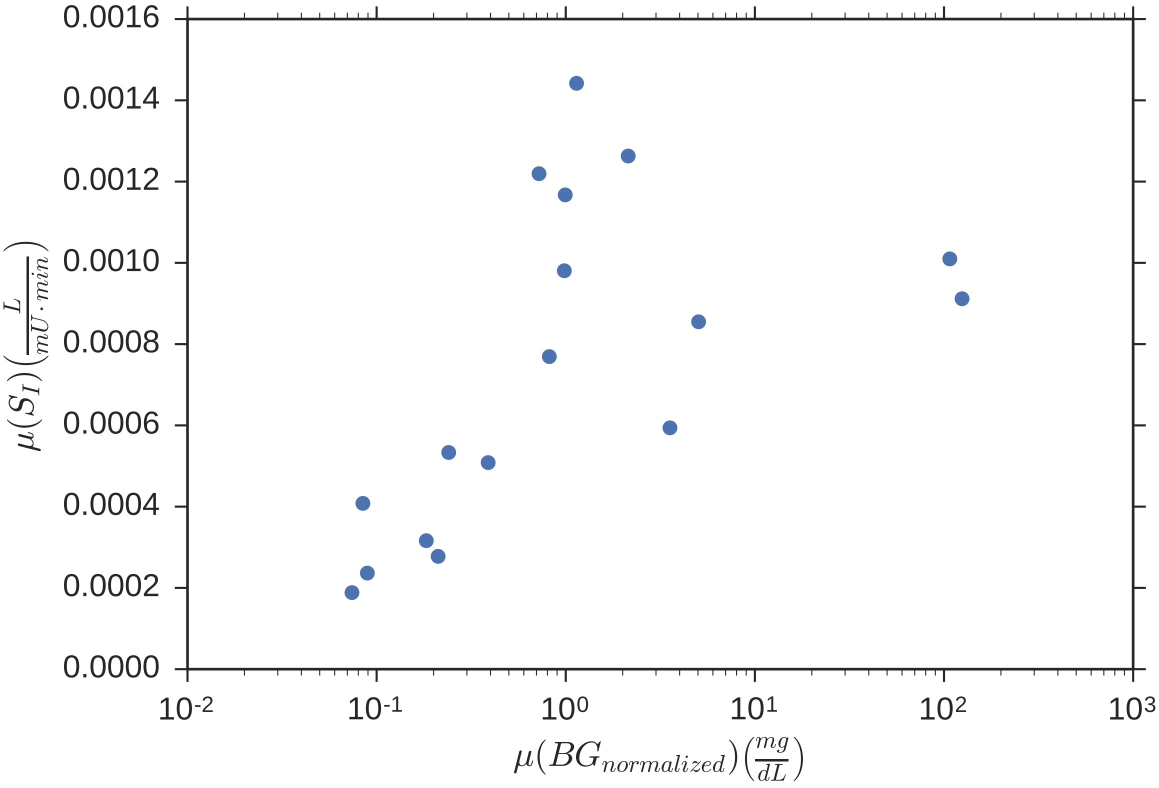


Figure 28: Mean blood glucose normalized by Equation (3.17) vs. mean insulin sensitivity.

much lower normalized average blood glucose concentration compared to patients not receiving exogenous insulin. It could be reasoned that these patients therefore have a depressed sensitivity to insulin necessitating large amounts of exogenous insulin. As such lower values of $BG_{normalized}$ would be expected to be associated with lower mean S_I values, a trend that is borne out in Figure 28.

Similar logic applies to the data in Figure 29 except the expectation is now that "unhealthy" patients receiving significant exogenous insulin intervention exhibit higher insulin sensitivity variability. This expectation seems to be corroborated by Figure 29.

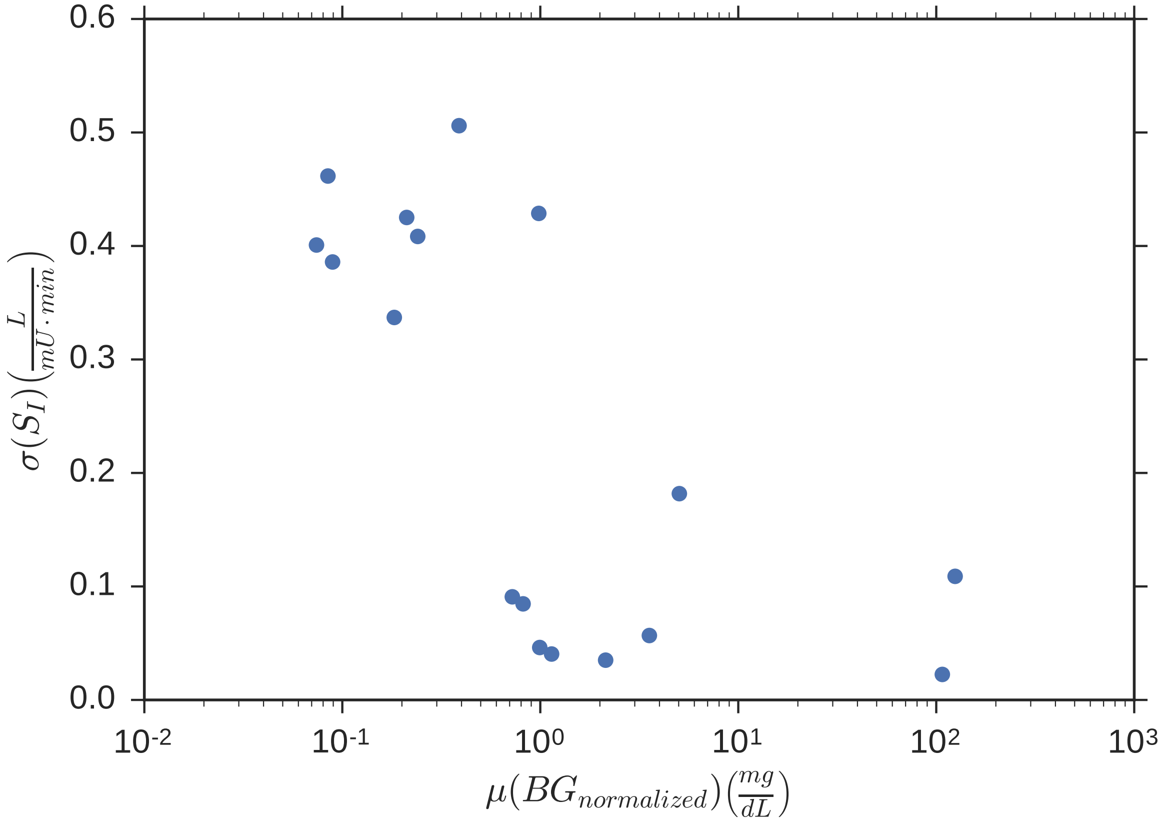


Figure 29: Mean blood glucose normalized by Equation (3.17) vs. mean normalized insulin sensitivity variance: the positive correlation between average blood glucose and the variability in insulin sensitivity across our virtual patient cohort corroborates the clinical expectation that patients exhibiting symptoms of stress hyperglycemia may have significant variability in insulin sensitivity over time.

3.4.2 Conserved Trends

Under the assumption that, patient health generally improves leading up to discharge from the ICU, we would expect to see this trend conserved in within our virtual patient cohort. If insulin sensitivity is used as a marker of health, then the variance in insulin sensitivity should drop. Figures 30a and 30b shows the mean normalized variance of insulin sensitivity, at time “N” over which is calculated using the previous “N-12” hours of data.

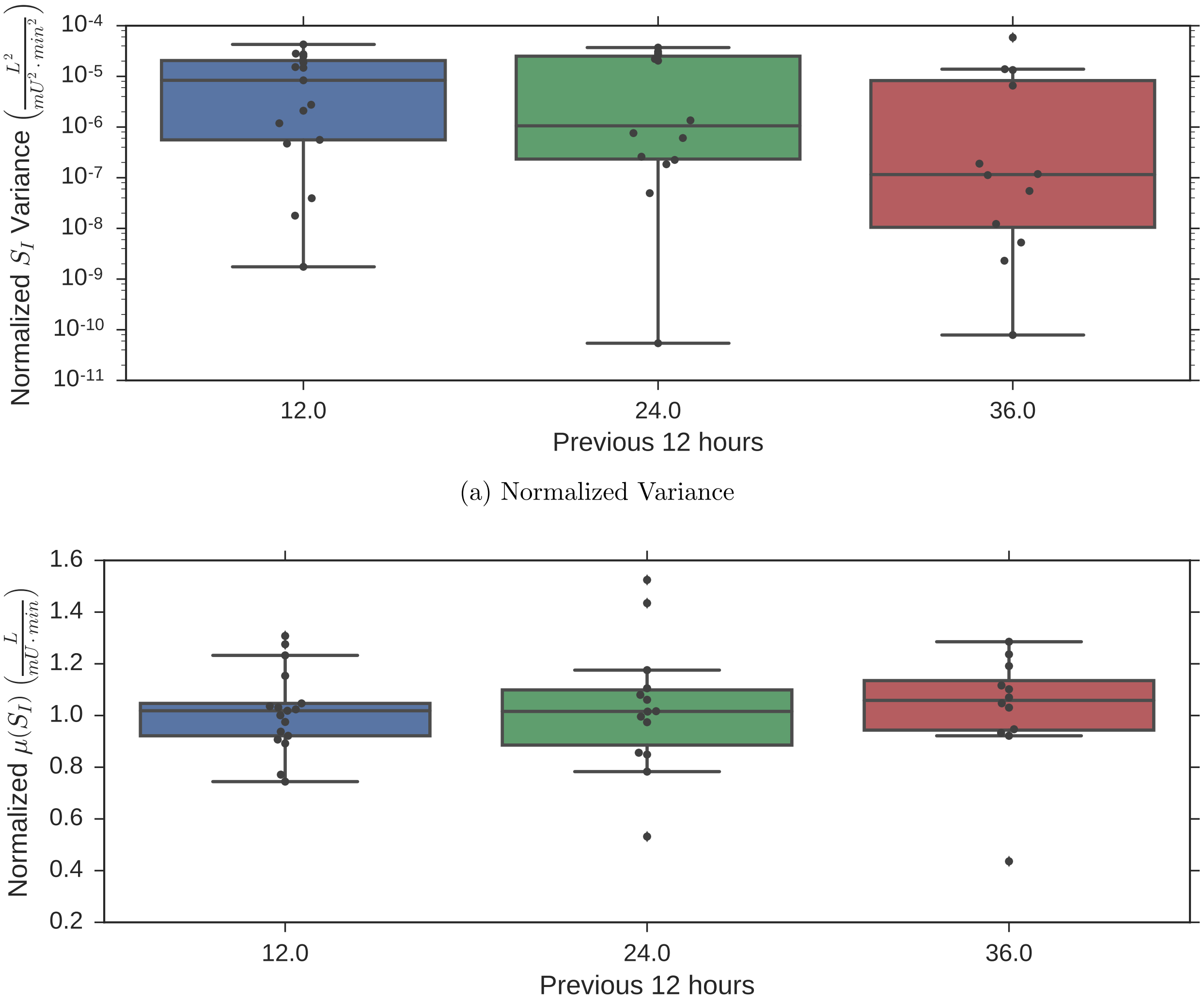
Figures 30a and 30b show that, in general virtual patients exhibit a decrease in S_I variability as well as an increase in insulin sensitivity over the course of treatment in the ICU. This trend recapitulates clinical understanding [216–218]. The conservation of this trend across the virtual patient cohort indicates that the model in Equations (3.1a) to (3.1h) incorporates the necessary physiologically-motivated characteristics to parametrically capture the observed variability in ICU patient glucose response across a cohort of patients.

3.5 SUMMARY

We have developed a virtual patient cohort derived from clinical data collected using continuous glucose monitors with a five minute sampling rate at the University of Pittsburgh Medical Center. The virtual patients within the cohort were synthesized by fitting clinical data using our eICING model, composed of: (i) the ICING [110] model; (ii) a mechanism for insulin-mediated endogenous glucose production from the "extended" minimal model (a model of subcutaneous insulin absorption [160]. Regressed quantities were a constant rate of non-insulin mediated glucose disposal (p_G), a dynamic pancreatic insulin secretion rate ($U_{en}(t)$) and a dynamic insulin sensitivity ($S_I(t)$) profile. Instead of a constant EGP_b an insulin-mediated functionality is added from an alternate model [171] which results in a saturating suppression of endogenous glucose production in response to plasma insulin levels elevated above baseline or basal values, consistent with repeated observations [167, 219–221].

The model closely matched the clinical data, after including medical record data regarding relevant interventions ((par)enteral nutrition, insulin administration, etc.) with a mean error of 1.02%. Through regularization, an oscillatory pancreatic insulin secretion rate of with an average value of approximately 62 mU/min was maintained across the virtual patient cohort. The period and magnitude of oscillation were consistent with studies of the ultradian nature of insulin secretion rates and this resulted in $S_I(t)$ trajectories that were congruous with clinical expectations. In hyperglycemic patients $S_I(t)$ profiles resulted in increased sensitivity and decreased variability as a patients stay in the ICU progressed. These trends are consistent with physiology, conserved across the virtual patient cohort, and, to our knowledge, have not been demonstrated in models of insulin-glucose dynamics in critically ill patients to date.

The result of this work is a virtual patient cohort and model that captures individual patient changes in insulin sensitivity and glucose dynamics, can respond to clinically relevant treatment interventions, and provides a possible platform for *in silico* testing of glucose control schemes and methodologies with minimal risk to patients.



(a) Normalized Variance

(b) Normalized Mean

Figure 30: (a) Normalized S_I variance across patients calculated with a sliding 12 hour window over the course of their stay in the ICU. As might be expected if insulin sensitivity variability is a marker of health there is a clear inverse relationship between insulin sensitivity variability and time spent in the ICU. (b) Normalized $\mu(S_I)$ across all patients also calculated with a sliding 12 hour window shows that in general insulin sensitivity improves over the course of an ICU stay. The 48 hour data point is excluded due to a small number of patients admitted to the ICU for 48 hours or more. For both (a) and (b) patient data points were normalized by S_I averaged across the duration of that patients' ICU stay.

4.0 RESPONSIVE VIRTUAL PATIENT PANCREATIC FUNCTION

4.1 INTRODUCTION

In healthy individuals the pancreas responds to plasma glucose concentrations and secretes insulin to maintain normoglycemia in a closed loop feedback system. Rising plasma glucose concentrations generally lead to rising insulin secretion rates and vice versa. The virtual patient cohort developed in Chapter 3 was developed by fitting a dynamic insulin sensitivity (S_I) and pancreatic insulin secretion rate (U_{en}) subject to a variety of physiologic constraints and heuristic regularizations. The resulting U_{en} trajectories are believed to be consistent with physiologic norms and realistically approximate clinical expectations, however they are specific to the measured blood glucose profile.

The goal of a closed-loop glucose control system for patients in critical care is to treat stress hyperglycemia while avoiding hypoglycemia As such, virtual patient simulations of various treatment methodologies are expected to yield plasma glucose profiles that may differ from the fitted glucose data, such that the fitted U_{en} trajectory is no longer strictly valid or accurate. In light of this discrepancy, a functionality that responds to plasma glucose concentrations through a change in pancreatic insulin secretion rates is developed for the virtual patient cohort.

Neglecting the pancreatic response to rising or falling blood glucose levels in *in silico* trials can endanger patient health. For example, if the pancreatic response to glucose concentrations is ignored during *in silico* trials, the controller may be tuned to aggressively deliver insulin to combat rising glucose concentrations, which results in maintenance of euglycemia in the virtual patient cohort. In a real (non-diabetic) patient, however, the pancreas would be expected to respond, to a degree, by increasing insulin secretion to combat rising glucose concentrations; when coupled with the aggressive response of the controller, which was tuned in the absence of this effect, dangerous hyperglycemia may result.

To develop a responsive pancreas for the virtual patients, insulin secretion is assumed to be comprised of both a glucose dependent and a glucose independent component [222]. The glucose dependent component of insulin secretion provides a baseline rate of secretion that shifts up or down in response to plasma glucose concentrations. The glucose-independent component provides the remainder of the insulin secretion and is found to contain the oscillatory component of insulin secretion resulting in the ultradian rhythms observed in the virtual patient cohort (see Chapter 3). The functionality for a glucose dependent component of insulin secretion is provided using a proportional-integral-derivative (PID) controller as has been done previously [223] and was motivated by the use of PID control in the development of many artificial pancreas systems [224–231].

This approach is inspired by fact that the pancreas acts in a feedback loop and, in a simplified view, acts as a controller for health and homeostasis. The PID parameters can be determined through a simple ordinary least squares regression. The pancreatic response under PID control across the virtual patient cohort is validated using simulated oral glucose tolerance tests and insulin tolerance tests with comparison to typical literature reported responses.

4.2 METHODS

4.2.1 PID Control

A PID controller is comprised of three terms one for each of proportional, integral and derivative action and is shown in it's time domain representation in Equation (4.1).

$$u(t) = K_p e(t) + K_i \int_0^t e(t) + K_d \frac{de(t)}{dt} \tag{4.1}$$

where K_p, K_i, and K_d are all tuning parameters and represent the proportional, integral, and derivative gains, respectively. $e(t)$ represents the error signal or deviation from the setpoint.

The proportional term $(K_p e(t))$ provides a contribution to control action that varies linearly with error. The integral term $\left(K_i \int_0^t e(t)\right)$ provides a correction to control action proportional to the accumulated error and helps to prevent controller offset. The final term in Equation (4.1) is the derivative term $\left(K_d \frac{de(t)}{dt}\right)$ and is used to correct control action based on the rate of change of the error.

4.2.2 Determining PID Parameters

For each virtual patient in the virtual patient cohort, the fitted pancreatic insulin secretion rate $(U_{en}(t))$ and model-predicted plasma glucose $(BG(t))$ are available at each time point. As design criteria for the PID control governing the glucose-dependent rate of insulin secretion, the PID U_{en} should match the virtual patient regressed U_{en} as closely as possible at each time point given $e_{BG}(t)$, where $e_{BG}(t) = BG(t) - BG_{SP}$ and BG_{SP} is a plasma glucose concentration setpoint. Here, BG_{SP} is taken as $110\frac{mg}{dL}$, which is the upper limit of normoglycemia [78, 232]. The integral component is calculated as $\sum_{i=0}^{T} e_{BG}(t_i)(t_i - t_{i-1}) \quad \forall T \in t$. Because the setpoint here is unchanging, $\frac{de_{BG}(t)}{dt}$ can be taken exactly as the value of Equation (3.1a) in Chapter 3.

Given vectors of values for $e_{BG}(t)$, $\sum_{i=0}^{T} e_{BG}(t_i)(t_i - t_{i-1})$, and $\frac{de_{BG}(t)}{dt}$ for all time points, the PID-governed rate of insulin secretion at time $t = t_{i+1}$ is given by a linear combination of those vectors at $t = t_i$ (see Equation (4.2)).

$$\begin{bmatrix} U_{en,PID}(t_1) \\ \vdots \\ U_{en,PID}(t_N) \end{bmatrix} = \begin{bmatrix} e_{BG}(t_0) & \sum_{i=0}^{t_0} e_{BG}(t_i)(t_i - t_{i-1}) & \frac{de_{BG}(t_0)}{dt} \\ \vdots & \vdots & \vdots \\ e_{BG}(t_{N-1}) & \sum_{i=0}^{t_{N-1}} e_{BG}(t_i)(t_i - t_{i-1}) & \frac{de_{BG}(t_{N-1})}{dt} \end{bmatrix} \begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} \tag{4.2}$$

The values of the PID gains (K_p, K_i, and K_d) can then be determined via ordinary least squares (OLS) subject to Equations (4.3) to (4.5), where the virtual patient fitted rate of pancreatic insulin secretion $U_{en}(t)$ is the response variable and $e_{BG}(t)$, $\sum_{i=0}^{T} e_{BG}(t_i)(t_i - t_{i-1})$, and $\frac{de_{BG}(t)}{dt}$ are the regressors.

$$K_p \geq 0 \tag{4.3}$$

$$K_i \geq 0 \tag{4.4}$$

$$K_d \geq 0 \tag{4.5}$$

4.2.2.1 Glucose Independent Insulin Secretion If the PID controller given by Equation (4.1) governs the glucose dependent rate of insulin secretion, then the glucose independent rate of secretion is simply taken as the residual from the OLS results used to determine the PID parameters.

During simulation of a virtual patient, the overall rate of pancreatic insulin secretion (U_{en}) is given as the sum of the PID-controlled rate and the residual from the OLS regression at any time. Under simulated treatment, this leads to the possibility of both $U_{en} < 0\frac{mU}{min}$ and $U_{en} > 210\frac{mU}{min}$. These scenarios are avoided by taking the soft minimum (Equation (4.6)) and soft maximum (Equation (4.7)) of the overall insulin secretion rate. Soft minimum and maximum functions are used to avoid sharp corners and approximate saturating rates of secretion, which are believed to be more physiologically realistic.

$$\min(x, y) = \frac{-\log(e^{-xk} + e^{-yk})}{k} \tag{4.6}$$

$$\max(x, y) = \frac{\log(e^{xk} + e^{yk})}{k} \tag{4.7}$$

4.2.3 Simulated Tolerance Tests

4.2.3.1 Insulin Tolerance Test The insulin tolerance test (ITT) is the gold standard for assessment of the hypothalamic-pituitary-adrenal axis (HPA), which is a measure of pituitary function, as well as for the assessment of the growth hormone (GH) axis [233].

Hypoglycemia stimulates the release of glucagon, epinephrine, growth hormone and cortisol, which promote hepatic glucogenesis, lipolysis and ketogensis [234–236] thereby acting as antagonists to insulin action. In this manner, cortisol and growth hormone, specifically, comprise key components of the counter-regulatory response to hypoglycemia [237–241]. Developed in the 1960s, the ITT uses an insulin bolus to induce hypoglycemia to elicit the counter-regulatory cortisol and growth hormone response. Because cortisol and GH release require proper hypothalamic, pituitary and adrenal function [242–244] the ITT has widely been used as a rapid test of the HPA and GH axes.

Although protocols vary, in general, hypoglycemia at a level at or below 50.4 $\frac{mg}{dL}$ is necessary to properly assess the HPA and GH-axes [245–250]. In a standard insulin tolerance test, between 0.025 and 0.3 $\frac{mU}{kg}$ of insulin are injected intravenously to induce the required hypoglycemic response with 0.15 $\frac{mU}{kg}$ as the typical administration. Glucose data following an ITT are available from the work of Greenwood *et al.* for insulin administration levels of $0.025, 0.05, 0.1$ and 0.15 $\frac{mU}{kg}$.

The goal of the ITT is to induce significant hypoglycemia in order to elicit a cortisol and growth hormone responses and because cortisol and growth hormone are insulin antagonists, However, these effects are not modeled or considered in the virtual patient cohort. As a result, the lowest insulin administration ($0.025\frac{mU}{min}$) is simulated to avoid inducing a level of hypoglycemia which would otherwise mount a counterregulatory response not a component of this work. For all virtual patients, the ITT is simulated during a period of stable plasma glucose ($\sigma(BG(t)) \leq 10\frac{mg}{dL}$ over the previous hour) and during a period in which the virtual patient medical record data was free of meals, exogenous glucose and insulin. Following simulation of the ITT, the recovery index (see Section 4.2.3.2) is calculated.

4.2.3.2 Recovery Index

Greenwood *et al.* [251] define a recovery index as the sum of plasma glucose levels obtained at 60, 90, and 120 minutes following insulin administration expressed as a percentage of the pre-ITT plasma glucose concentration (Equation (4.8)).

$$RI = \sum_{i=[60,90,120]} \frac{BG(t_0+i)}{BG(t_0)} \quad \text{where } t_0 \text{ is the time of ITT insulin injection} \tag{4.8}$$

The recovery index is used as a measure of the rate of recovery of plasma to pre-ITT values.

4.2.3.3 Oral Glucose Tolerance Test The oral glucose tolerance test (OGTT) is the most commonly used method to test whole body glucose tolerance *in vivo* [252] and in some instances to derive information about insulin release and insulin sensitivity [253–255]. Briefly, an OGTT involves oral consumption of some amount of glucose followed by blood glucose draws at several intervals following glucose intake.

The World Health Organization (WHO) recommends a 75 gram oral dose of glucose in all adults [256] to be consumed within 5 minutes. This is the main dosage used for OGTTs in the United States [257] and the version of the test that will be simulated here. Before the test is administered, plasma glucose levels should be below $110\frac{mg}{dL}$, and at 1 hour following glucose intake, plasma glucose levels below $180\frac{mg}{dL}$ are considered normal [257, 258]. After 2 hours, plasma glucose concentrations below $140\frac{mg}{dL}$ are considered normal [257], $140 \leq BG \leq 200\frac{mg}{dL}$ is indicative of impaired glucose tolerance and $BG > 200\frac{mg}{dL}$ confirms a diabetes diagnosis [258].

Very few critical care patients meet the criteria for administering the OGTT (mainly plasma BG $< 110\frac{mg}{dL}$) as they have not necessarily been fasting and are assumed to suffer from stress hyperglycemia. As a result, the plasma glucose limits for normal, impaired and diabetic glucose tolerances in a 2 hour OGTT are not particularly useful in identifying proper dynamics in the virtual patient cohort. However, because the OGTT is a common test, there is plentiful data in the literature detailing plasma glucose dynamics following an OGTT. Here, we are primarily interested in time to peak plasma glucose concentrations and secondarily, time to recover to pre-test levels. The data from [252–254, 259–268] indicate that in healthy patients, a peak plasma glucose concentration following an OGTT is reached within 60 minutes, and in type 2 diabetic patients the same peak is not reached until approximately 90 to 120 minutes (see Figure 31 for example data).

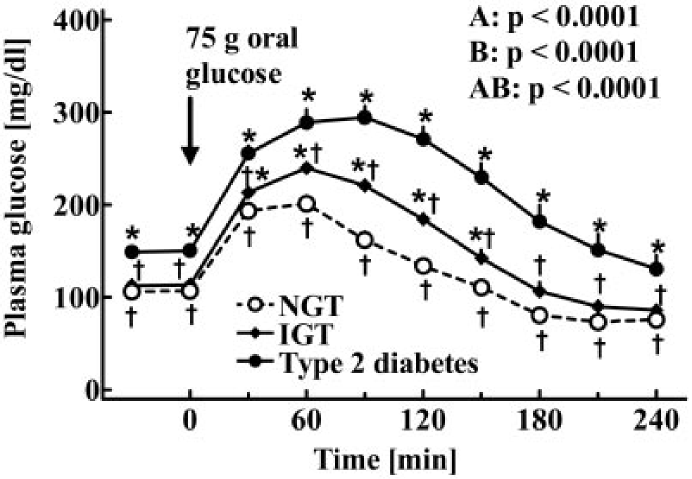


Figure 31: Oral glucose tolerance test plasma glucose dynamics following a standard OGTT for patients with normal glucose tolerance (○), impaired glucose tolerance (●) or Type 2 diabetes (⧫) (Reproduced from Vollmer *et al.* [262]). Note that all patients reach a peak plasma glucose concentration within approximately 90-120 minutes and return to pre-test plasma glucose concentrations within 240 minutes.

For simulations of an OGTT in the virtual patient cohort the standard dose of 75 g of glucose administered orally is used. For all virtual patients the OGTT is simulated during a period of stable plasma glucose ($\sigma(BG(t)) \leq 10\frac{mg}{dL}$ over the previous hour) during a period in which the virtual patient medical record data was free of meals, exogenous glucose and insulin. Following simulation of the OGTT the time until a glucose peak in the following 6 hours is reached is calculated as well as the time until glucose levels fall to within 15% of glucose concentrations simulated without the OGTT.

4.3 RESULTS

4.3.1 PID Parameters

The PID gains resulting from the OLS regression are shown in Figure 32.

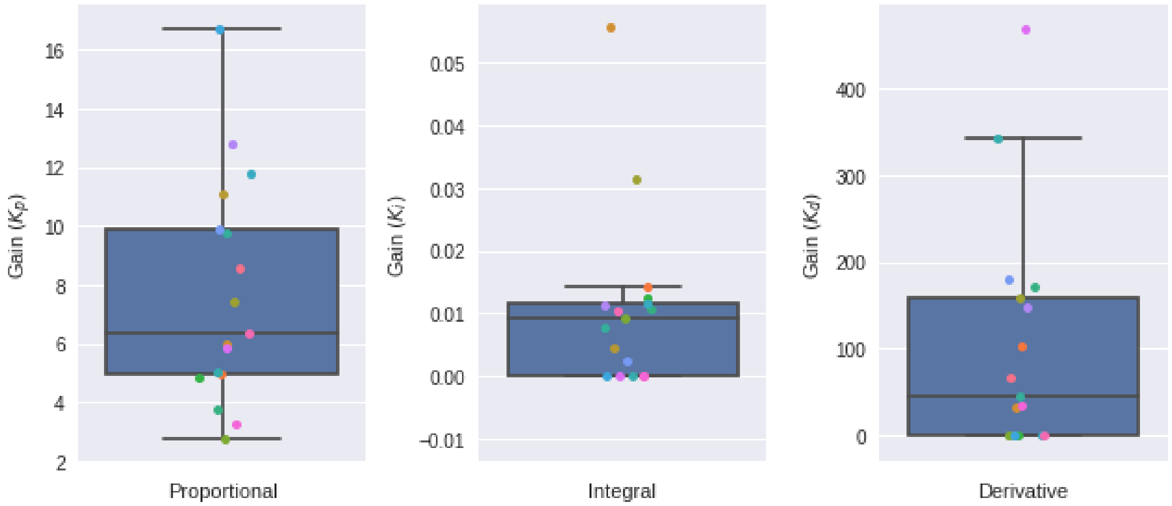


Figure 32: Distribution of regressed values for the proportional, integral and derivative gains, respectively, are shown by box plots. The exact values are shown by colored circles. The colors of the circles correspond to single patient values across all three panes of the figure.

Figure 32 shows that the proportional gain (K_p) is strictly positive for all patients, as would be expected given the definition of the error here as $e_{BG}(t) = BG(t) - BG_{SP}(t)$. As plasma glucose concentrations rise, the error signal grows more positive, which given our understanding of pancreatic function should result in increasing rates of pancreatic insulin secretion. This seems to be borne out in the OLS regression to the virtual patient data which returns $K_p > 0$ across the virtual patient cohort. A similar argument can be made for K_i, the gain on the integral component of the PID controller, which is nonnegative across the virtual patient cohort. Several patients resulted in a K_i exactly equal to zero, as per Equations (4.3) to (4.5), perhaps indicating a chronic pancreatic dysfunction or failure in these patients. This leaves the derivative gain (K_d), which as seen in Figure 32, is nonnegative across the virtual patient cohort. A certain subset of patients have the derivative component of their pancreatic response bounded at zero by Equations (4.3) to (4.5), which may be due to acute pancreatic dysfunction resulting from stress and an inability to respond to rapidly changing blood glucose concentrations. Perhaps in these patients this effect is responsible for the elevated blood glucose variability commonly observed in stress hyperglycemia. For all virtual patients, the PID-controlled rate of insulin secretion ($U_{en,PID}$) and the glucose independent oscillatory rate of insulin secretion ($U_{en,GID}$) are given in Appendix D. Here $U_{en,GID}$ is simply the vector of residuals from the OLS regression of the PID parameters.

4.3.2 Insulin Tolerance Test

The Recovery Index (RI) for all patients meeting the criteria for inclusion in the simulated ITT (see Section 4.2.3.1) is shown in Figure 33 for simulations with a responsive pancreas under PID control and for a nonresponsive pancreas. Four patients were excluded from the ITT, as they did not meet the blood glucose stability criteria established above or did not have a period free of exogenous input in the medical record data. The dynamics following a simulated ITT for an example patient simulated with a responsive, or nonresponsive, pancreas are shown in Figure 34.

4.3.3 Oral Glucose Tolerance Test

The time to peak glucose following a simulated OGTT for all patients meeting the criteria for inclusion in the simulated OGTT (see Section 4.2.3.3) is shown in Figure 35 for simulations with a responsive pancreas under PID control and a nonresponsive pancreas. Four patients were excluded from the OGTT as they did not meet the blood glucose stability criteria established above or did not have a period free of exogenous input in the medical record data. The time to recovery as defined by blood glucose values returning to within 10% of glucose concentrations had the OGTT not been performed is shown for these patients in Figure 36. The dynamics following a simulated OGTTT for an example patient with a responsive and nonresponsive pancreas are shown in Figure 37.

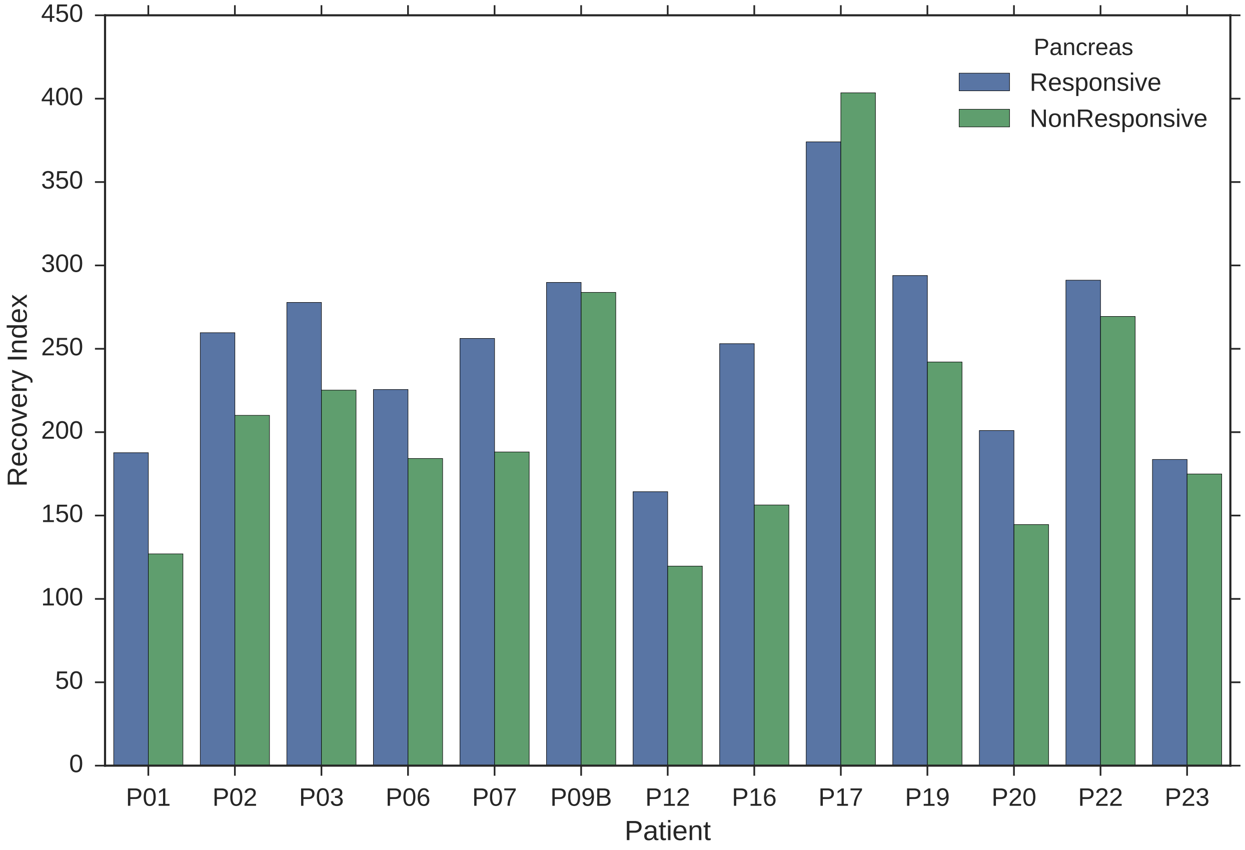


Figure 33: Recovery index for the simulated insulin tolerance test for virtual patients with a responsive pancreas under PID control and a nonresponsive pancreas. 4 Patients were excluded for failure to meet the criteria to conduct an ITT.

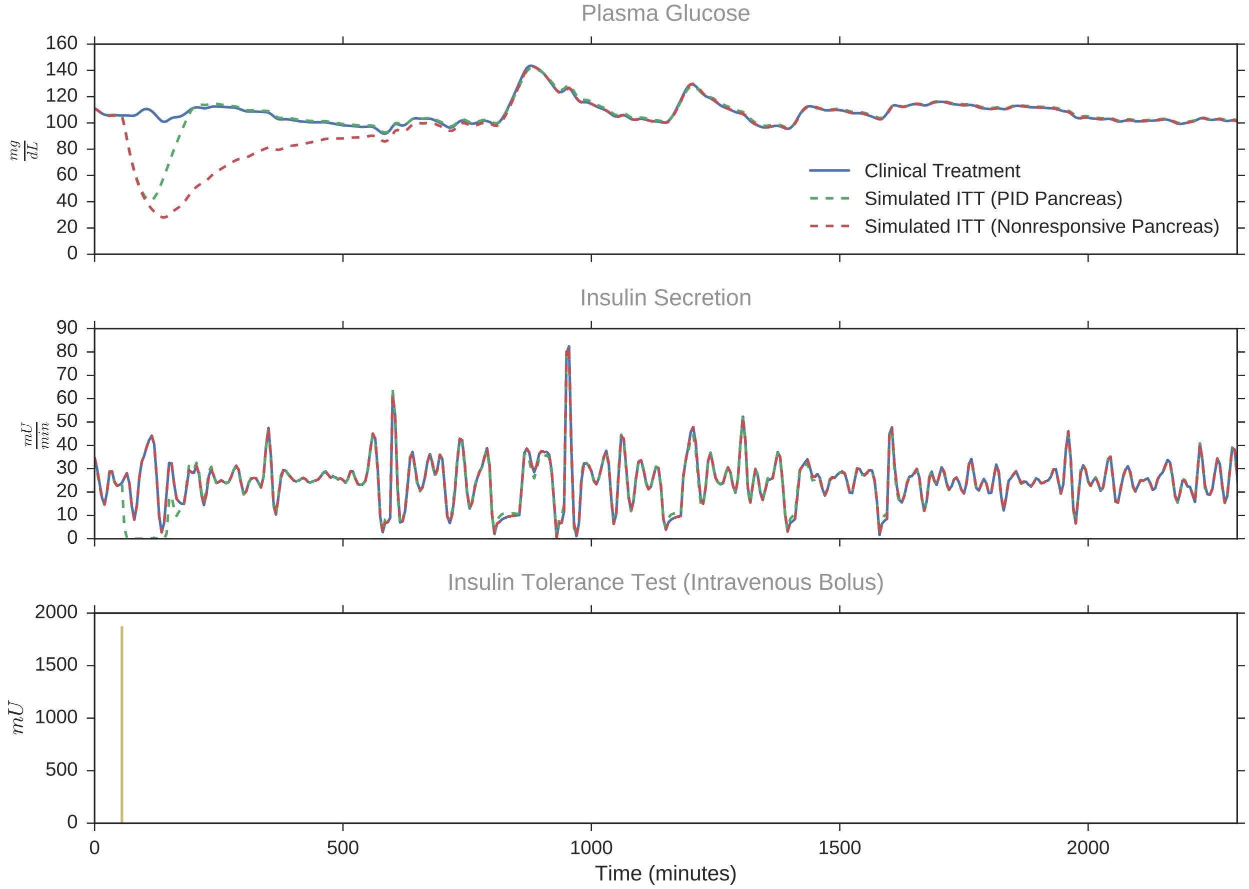


Figure 34: Example virtual patient undergoing an insulin tolerance test with a simulated responsive pancreas and a nonresponsive pancreas. The top panel shows plasma glucose concentrations for the baseline virtual patient (clinical treatment, -), the virtual patient simulated with a responsive pancreas (--) and a nonresponsive pancreas (--). The bottom panel indicates the time and magnitude of the ITT insulin bolus (-).

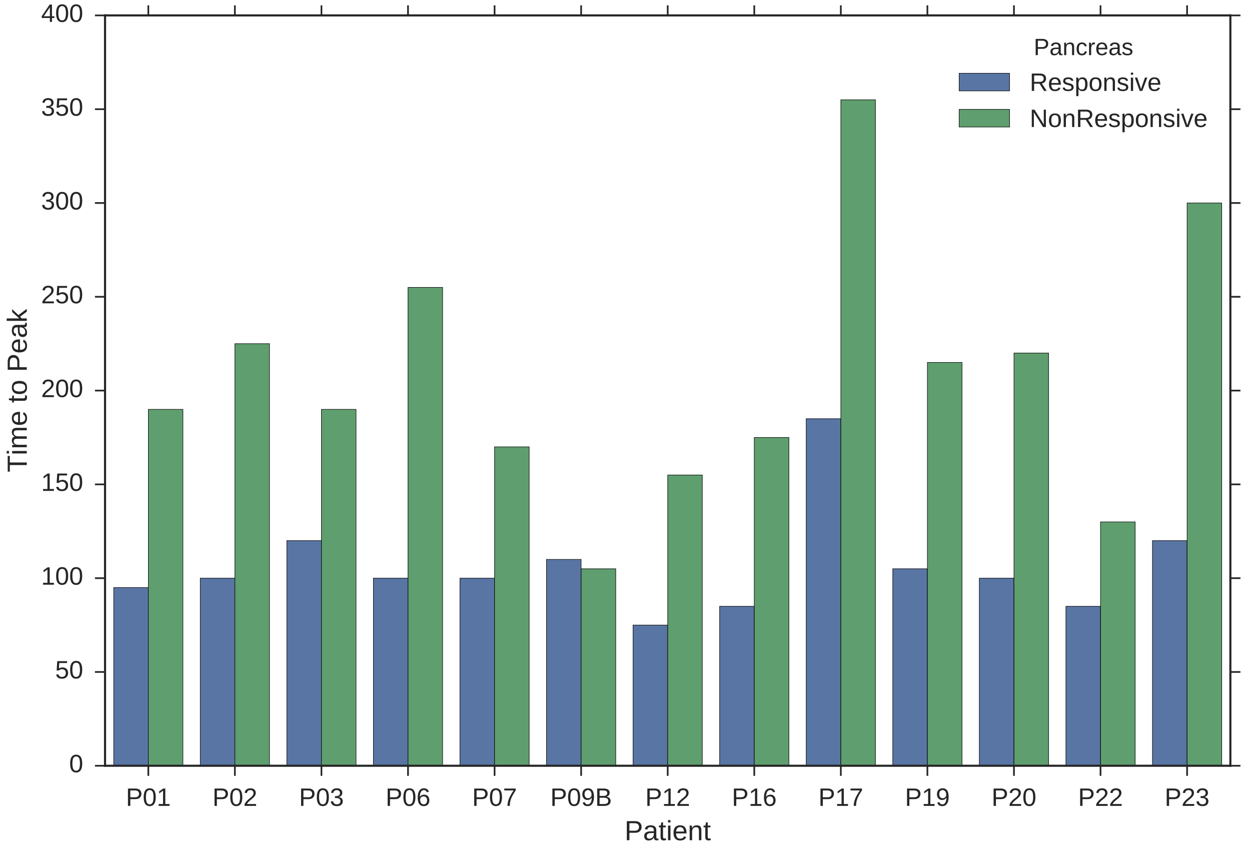


Figure 35: Time to peak plasma glucose concentrations following a simulated oral glucose tolerance test for virtual patients with a responsive pancreas under PID control and a non-responsive pancreas. 4 Patients were excluded for failure to meet the criteria to conduct an OGTT.

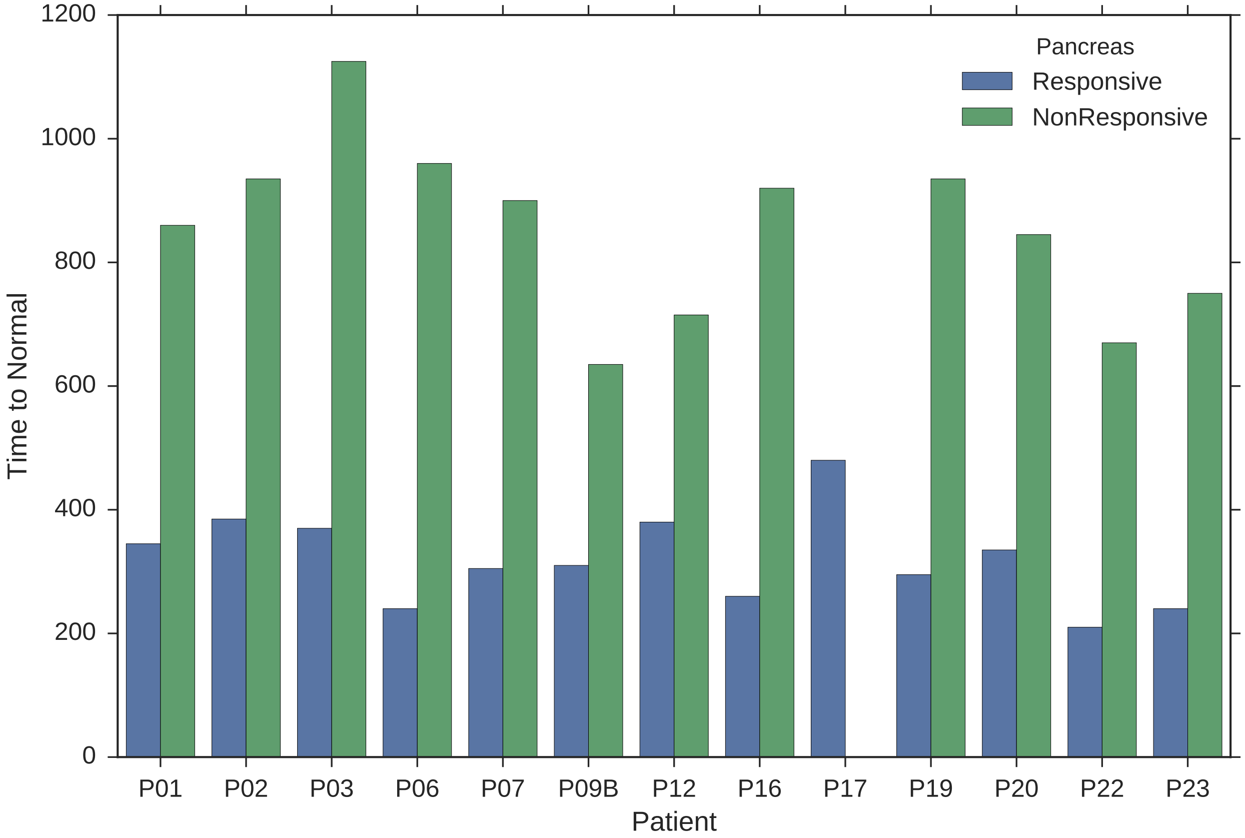


Figure 36: Time to recovery of plasma glucose concentrations following a simulated oral glucose tolerance test for virtual patients with a responsive pancreas under PID control and a nonresponsive pancreas. Recovery is defined as plasma glucose concentrations returning to $\leq$ 10% of plasma glucose concentrations simulated in the absence of the OGTT. 4 Patients were excluded for failure to meet the criteria to conduct an OGTT.

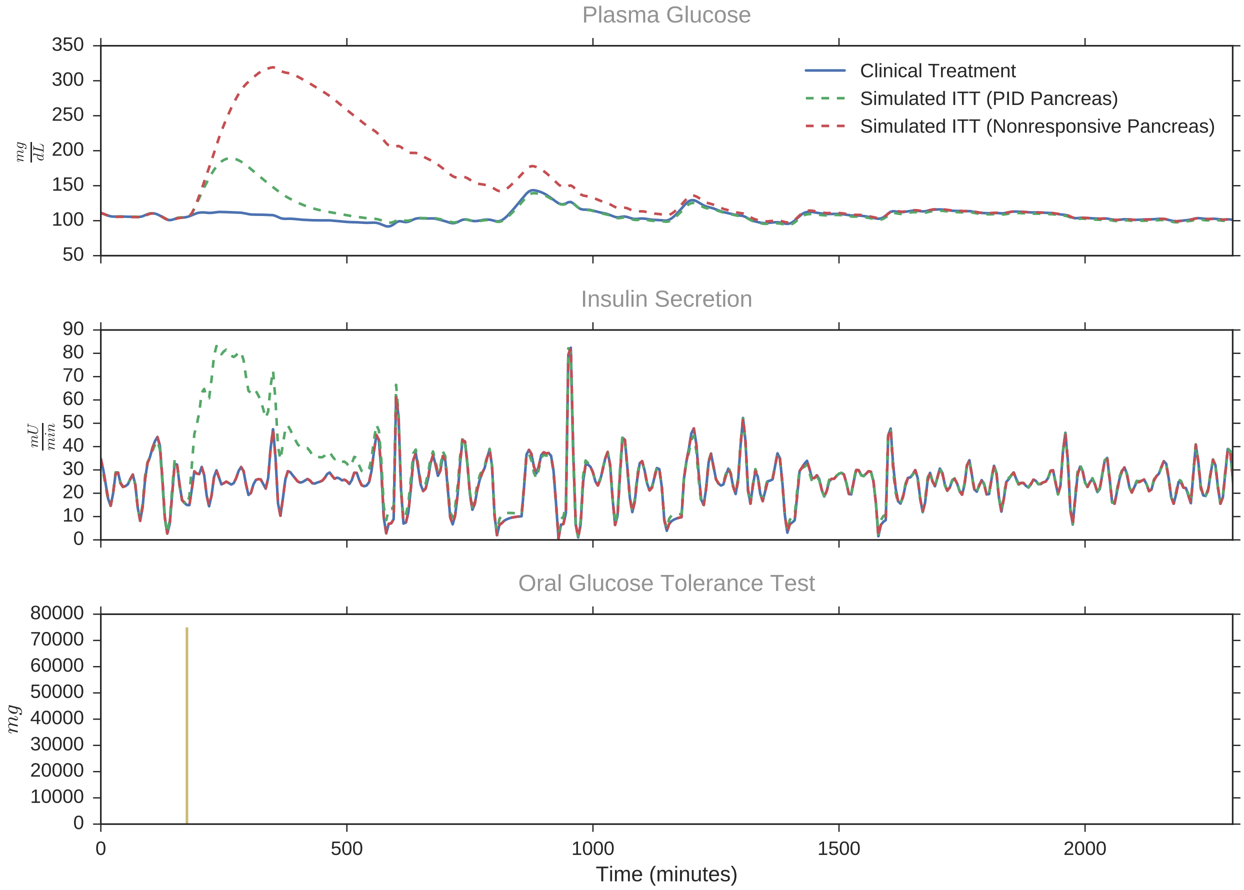


Figure 37: Example virtual patient undergoing an oral glucose tolerance test with a simulated responsive pancreas and a nonresponsive pancreas. The top panel shows plasma glucose concentrations for the baseline virtual patient (clinical treatment, -), the virtual patient simulated with a responsive pancreas (--) and a nonresponsive pancreas (--). The bottom panel indicates the time and magnitude of the OGTT oral intake (-).

4.3.4 Discussion

4.3.4.1 Effects of Pancreatic PID Control As a motivating example, consider simulations of a virtual patient who was treated in the clinic with significant exogenous insulin in the form of subcutaneous boluses of lispro, a fast acting insulin analog [269, 270]. If the clinically administered exogenous insulin is withheld, as would be done in *in silico* trials of a closed-loop glucose control system, the simulated virtual patient response is shown compared to the clinical regimen in Figure 38.

When this insulin regimen is withheld in simulations, the PID component of U_{en} results in higher secretion rates to combat rising plasma glucose concentrations in the virtual patient with a responsive pancreas. From Figure 38, it is clear that not accounting for a pancreatic response results in a much more significant degree of hyperglycemia.

This example demonstrates the effect of withholding clinical treatment, however it does not demonstrate the pancreatic response to a different treatment regimen, one that might include significant, extended exogenous insulin and glucose infusions, such as when a patient is placed in the care of a closed-loop glucose control scheme. This is the motivation for simulating insulin and oral glucose tolerance tests. Furthermore, these are standardized tests with quantifiable outcomes which can be compared to a wealth of literature data.

4.3.4.2 Insulin Tolerance Test The results shown in Figure 33 demonstrate that patients simulated with a pancreas having a PID controlled response to plasma glucose concentrations exhibit a significantly higher recovery index compared to virtual patients with no responsive pancreatic functionality. The results of Greenwood *et al.* [251] suggest that following an insulin tolerance test with an insulin administration of 0.025 $\frac{mU}{kg}$, a recovery index of 291 ± 16 would would be expected. The average recovery index across our virtual patient cohort is 182.1 for the group with a responsive pancreas and 145.2 for the group with a nonresponsive pancreas. As expected, the group with the responsive pancreas has a higher recovery index than the virtual patients with a nonresponsive pancreas due to a decrease in insulin secretion rates by the PID controlled component of pancreatic function (see Figure 34 for the pancreatic insulin suppression).

However, even the value for the virtual patient group with responsive pancreatic function is less than half of that reported by Greenwood *et al.* In light of the assumption that these patients are suffering from stress hyperglycemia believed to be driven, in part, by depressed insulin sensitivity, the expectation would be that these patients should have a

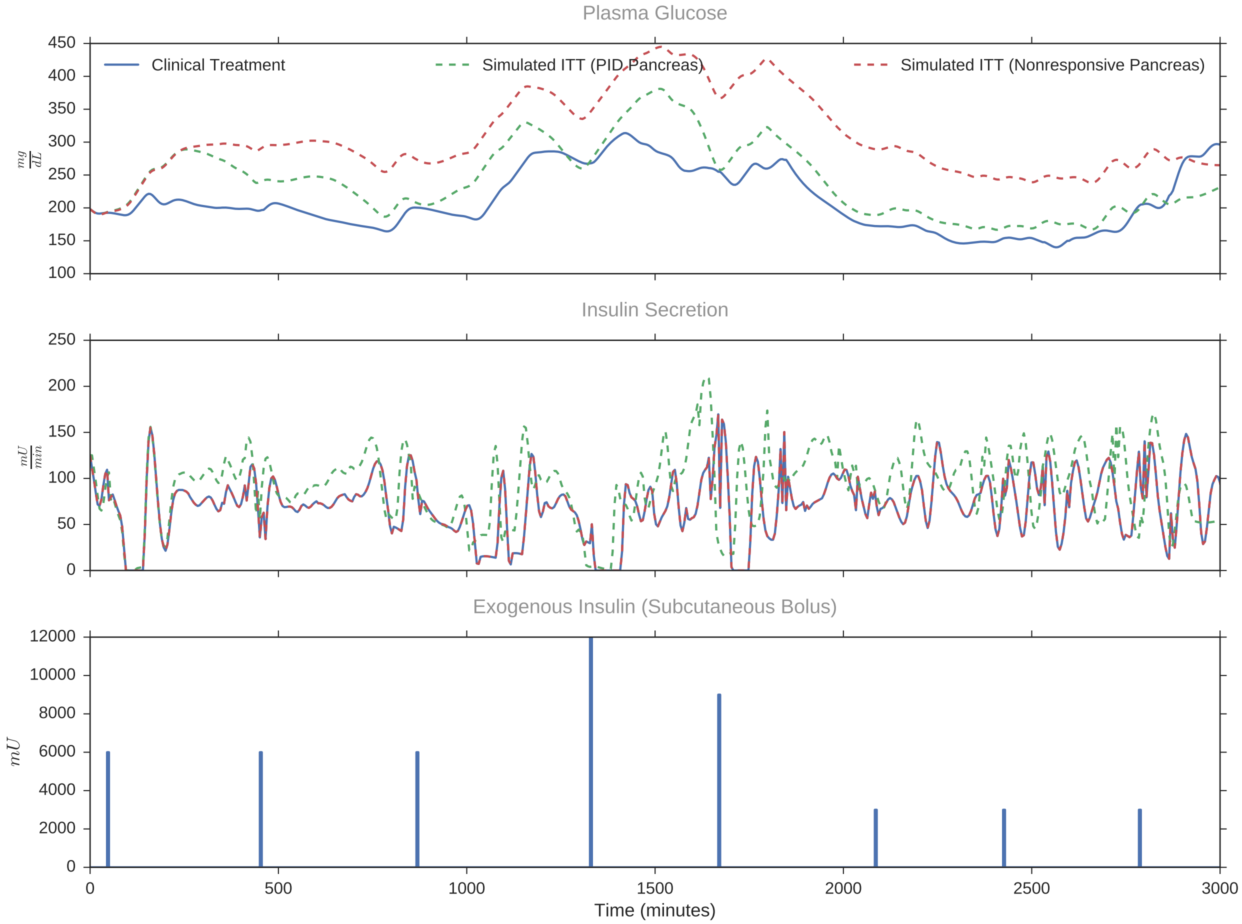


Figure 38: Simulated plasma glucose concentrations (top), and insulin secretion rates (middle) simulated under the virtual patient fitted clinical treatment (-) and with the subcutaneous insulin administration withheld for the virtual patient with a responsive pancreas with a component of insulin secretion under PID control (--) and within a non-responsive pancreas (--). The exogenous insulin schedule regimen used in the clinic and for fitting the virtual patient is shown in the bottom panel.

higher recovery index than the healthy subjects in [251], as the administered insulin should have a diminished effect on plasma glucose concentrations due to insulin resistance. Although this discrepancy seems large, consider that Greenwood *et al.* reported a significant increase in plasma cortisol concentrations compared to a saline control (11.5 ± 2.3 vs. $10.6 \pm 1.9\ \frac{\mu g}{dL}$) as well as elevated plasma growth hormone concentrations compared to the saline control (5.2 ± 3.8 vs. $1.8 \pm 2.4\ \frac{ng}{mL}$). Cortisol and growth hormone are powerful insulin antagonists and are not considered in the extended ICING virtual patient model, which may explain the depressed recovery index in the virtual patient cohort in response to an insulin tolerance test.

Patients P17 and P23 in Figure 33 show an increased recovery index with the nonresponsive pancreas, counter to intuition and the trend across the remainder of the simulated insulin tolerance tests. These results can be explained by a severe lack of insulin sensitivity or extreme insulin resistance in these patients. In both patients, the insulin bolus fails to induce a meaningful decrease in plasma glucose concentrations; the differences between the pancreatic functionalities can be attributed to the pancreatic response to withheld clinical treatment in the patient with PID controlled insulin secretion rates.

Taking a more qualitative approach to the question of realism in the responsive pancreatic functionality introduced via PID control in the virtual patient cohort, consider the simulated plasma glucose dynamics following an ITT in Figure 34. The virtual patient with a responsive pancreas has a higher plasma glucose nadir compared to the virtual patient with a nonresponsive pancreas ($\approx 40\frac{mg}{dL}$ vs. $\approx 20\frac{mg}{dL}$), a value consistent with the approximately 60% reduction in plasma glucose concentrations seen in [251] at this level of insulin administration.

Furthermore, the virtual patient with responsive pancreatic function recovers to pre-ITT conditions in approximately 120 minutes, whereas the virtual patient with a nonresponsive pancreas takes approximately 445 minutes to recover. The data collected by Greenwood *et al.* [251] and others [237, 271–273] seem to suggest that the 120 minute recovery time seen in the virtual patient group with responsive pancreatic function is much closer to reality and physiology than the long tail of recovery seen in the group with a nonresponsive pancreas.

4.3.4.3 Oral Glucose Tolerance Test

The results shown in Figure 35 indicate that virtual patients simulated with a responsive pancreas achieve a peak plasma glucose concentration during an oral glucose tolerance test on average 125 ± 79.6 minutes following enteral glucose intake. In comparison, the virtual patients with no pancreatic responsivity take on average 209 ± 90.04 minutes to reach peak plasma glucose concentrations. The difference in dynamics is due to an upregulation in insulin secretion rates in response to rising plasma glucose concentrations in virtual patients simulated with a responsive pancreas. The increased rate of insulin secretion in these patients results in more rapid metabolism of the ingested glucose leading to a more rapid fall in plasma glucose concentrations and a correspondingly earlier peak. The time to peak of 125 minutes for the responsive pancreas group is much more consistent with literature data and physiologic expectations (see Figure 31) [262].

For the same reason, Figure 36 shows that the virtual patients with a pancreas capable of responding to plasma glucose concentrations show a much more rapid overall response compared to the nonresponsive virtual patients (average time of return to “normal” plasma glucose concentrations of 319 ± 71.3 minutes vs. 854 ± 134,minutes). A response time of 319 minutes is consistent with literature data [252–254, 259–268] and much more believable on a physiological and experiential basis than 854 minutes (over 14 hours), even for insulin resistant patients. See Figure 37 for an example of an OGTT in the same virtual patient used for the example ITT (Figure 34). Note the long aphysiologic hyperglycemic tail that results from the OGTT in a virtual patient without a pancreatic response to glucose and the significant increase in insulin secretion rates in response to rising plasma glucose concentrations following the OGTT in the virtual patient with a pancreatic response.

4.4 SUMMARY

The pancreatic response demonstrated here is shown to be consistent with experimentally observed responses for both insulin and glucose tolerance tests and is believed to add an additional element of realism to the virtual patient cohort. The enhanced realism and accuracy of the virtual patient cohort enabled through the use of a responsive pancreas that adapts insulin secretion rates dependent on plasma glucose concentrations via a PID control law will lead to higher quality controller tuning and enhanced closed-loop glucose control safety and efficacy.

5.0 MODEL BASED GLUCOSE CONTROL

5.1 INTRODUCTION

Maintaining blood glucose concentrations within a tight target zone or range requires close monitoring of blood glucose levels and more frequent intervention than is practical by critical care staff. Furthermore, current care protocols do not address inter- and intrapatient variability in glucose-insulin dynamics stemming from differing insulin sensitivities. High-frequency sampling of blood glucose levels, through continuous glucose monitors, coupled with model-based control, has the capacity to enable zone-targeted glucose control to maintain blood glucose levels within the targeted zone and thereby decrease hyperglycemia while avoiding hypoglycemia, ultimately leading to improved patient outcomes.

The work described herein details the development and virtual testing of an automated linear model predictive controller with state estimation for the delivery of both subcutaneous insulin and intravenous glucose to maintain patient blood glucose levels within a target zone (110-130 mg/dL). Subcutaneous insulin is used as it is the clinical standard in the United States, and complications associated with IV administration, such as infection, are avoided. Output regulation is used to ensure that intravenous glucose infusion rates return to 0 when deemed unnecessary by the controller. The controller model is informed from glucose measurement values, as measured by a continuous glucose monitoring (CGM) system with a 5-minute sampling interval.

Moving horizon estimation is employed to ensure that the controller model closely matches patient dynamics and enables the controller to account for both intra- and inter-patient variability. Insulin sensitivity (S_I) is used as an estimated model parameter due to its stress-driven transient nature in critically ill-patients. The ICING model parameter describing the rate of pancreatic insulin secretion (U_{en}) is estimated, as endogenous insulin production is known to be variable and would be expected to respond via a feedback mechanism to plasma glucose levels. Additionally, the rate of endogenous glucose production, EGP, assumed to be a static parameter in the ICING model, is estimated to account for responsivity to changing plasma insulin concentrations.

In addition to the aforementioned parameters, Q_I, the state describing effective interstitial insulin concentrations, is estimated, with deviations from model dynamics heavily penalized. The logic behind this decision stems from the idea that remote, or effective, insulin concentrations should generally follow model dynamics. However, due to the linearization, the saturating effect of high effective insulin concentrations on blood glucose disposal is lost. As a result, at high values of Q_I, the rate of blood glucose disposal may be such that it becomes impossible for modeled BG concentrations to reach the measured value.

Allowing for error on Q_I enables a decrease in Q_I contrary to linearized ICING dynamics when model-predicted BG would otherwise disagree significantly from measured values. By decreasing Q_I, the rate of insulin-mediated glucose disposal drops, thereby mimicking the saturation of the insulin-mediated glucose disposal. A large penalty on Q_I is used in both Q and P_0 so that Q_I varies primarily according to model dynamics and differs only when significant error from the measurements would otherwise arise.

The glucose control system designed here is a zone model predictive controller (zMPC) with moving horizon estimation (MHE) formulation that includes glucose output regulation, which will be referred to from here forward as zMPC/MHE. Diagrammatic examples of MHE, MPC, and zone control for the glucose control problem are shown in Figure 39.

The zMPC/MHE/GOR control scheme developed here is tested and tuned via *in silico* trials on the virtual patient cohort developed in Chapter 3, with the responsive pancreas explained in Chapter 4. The controller is evaluated under the assumption of perfect sensors, or in the face of realistic CGM noise, via simulations of the error process detailed in Chapter 2. Finally, the glucose control algorithm of this work is compared to commercially available solutions such as the computerized Yale protocol [275] and the GlucoStabilizer® algorithm [276].

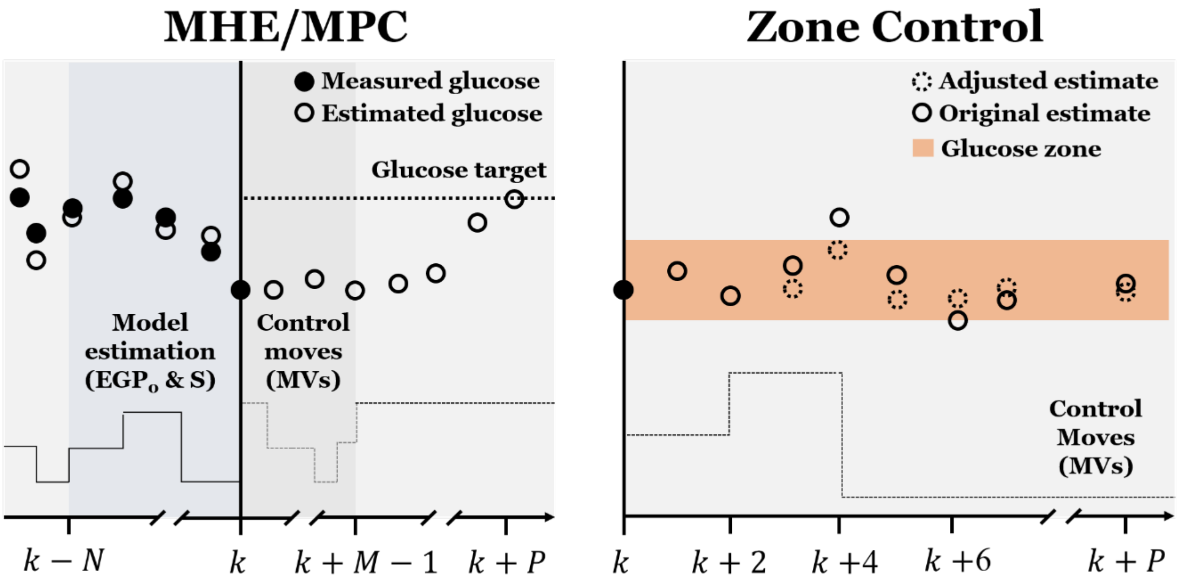


Figure 39: MPC/MHE schematic (left) showing prediction and estimation horizons along with optimal control actuation (adapted from [274]). MHE minimizes the error between past glucose measurements and model predictions by adjusting insulin sensitivity and insulin secretion rates based on past data over a moving horizon of length N. Zone control schematic (right) shows changes in infusion rates to maintain predicted blood glucose within a specified zone.

5.2 METHODS

5.2.1 Zone Model-Predictive Control (zMPC)

Given that an optimal range of blood glucose concentrations resulting in improved patient outcomes has been established ([121]), an MPC formulation for control to zone (zMPC), rather than a single setpoint, is employed. Using an internal model, a zone model-predictive controller minimizes the objective function of an optimization problem where model-predicted excursions of the controlled variable(s) from the target zone are penalized over a prediction horizon. Changes in the manipulated variables are also penalized. The resulting quadratic objective function is subject to constraints governing the model dynamics, maximum and minimum values for the manipulated variables, and maximum and minimum point-to-point changes in the manipulated variables.

For the discrete state-space model given by:

$$\begin{aligned} x_{k+1} &= Ax_k + B_u u_{k-1} + B_u \Delta u_k + B_d d_k \\ y_k &= Cx_k + D_u u_k + D_d d_k \end{aligned} \tag{5.1}$$

where x_{k+1} is the predicted state vector at the next time step, x_k is the state vector at the current time, u_{k-1} is a vector of previous manipulated inputs, Δu_k is the change in manipulated inputs at the current time, d_k is a vector of disturbance inputs at the current time, and y_k is the observed output at the current time, a linear, constrained model predictive controller is formulated as shown in the following quadratic program([277]):

$$\underset{z}{\text{minimize}} \quad \sum_{i=1}^{P} \|y_{k+i} - \delta\|_\Gamma + \sum_{i=1}^{M} \|\Delta u_{k+i-1}\|_S \tag{5.2a}$$

$$\text{subject to:} \quad x_{k+1} = Ax_k + B_u u_{k-1} + B_u \Delta u_k + B_d d_k \tag{5.2b}$$

$$y_{k+1} = Cx_{k+1} + D_u u_k + D_d d_k \tag{5.2c}$$

$$\Delta u_{min} \le \mathbb{I}\Delta u \le \Delta u_{max} \tag{5.2d}$$

$$u_{min} \le \mathbb{I}u \le u_{max} \tag{5.2e}$$

$$\text{Zone}_{lower} \le \delta \le \text{Zone}_{upper} \tag{5.2f}$$

$$\text{where :} \quad z = \Delta u_k, \ldots, \Delta u_{k+N-1}$$

Here Γ is the matrix penalizing predicted model deviations from the target zone and S is the matrix penalizing control moves. Γ and S, along with the prediction horizon, P, and the control horizon, M, are controller tuning parameters.

This formulation results in targeted zone control by allowing δ to move between its lower and upper bounds (Zone$_{lower}$ and Zone$_{upper}$, respectively) as given by constraint (5.2f) to minimize the difference between the measured output and predicted output. Δu_{min} and Δu_{max} in constraint (5.2d) are the lower and upper bounds, respectively, on the allowable change in the manipulated input at any given time step. The vector of non-manipulated variables is generally assumed to remain constant at d_k over the prediction horizon. Solution of this quadratic program results in a vector of optimal input changes at every time point along the control horizon, however only the first suggested input change is implemented by the controller.

For the glucose control problem, $u_k = [U_{I,k}, U_{G,k}]'$ and $y_k = [BG_k, U_{G,k}]'$ where $U_{I,k}$ and $U_{G,k}$ are the subcutaneous insulin and glucose infusion rates, respectively. BG_k is the model-predicted blood glucose concentration. The fact that $U_{G,k}$ appears in both u_k and y_k is what gives rise to glucose output regulation in this controller. The addition of this output regulation serves to diminish spiking behavior and ensures that glucose and insulin are not infused unnecessarily. This formulation also:

(i) prevents a situation where the controller infuses glucose constantly while simultaneously infusing insulin at a constant (elevated) rate

(ii) reduces control effort by minimizing exogenous insulin usage, because continuous insulin infusion suppresses endogenous insulin production, which would thereby increase the control effort required to maintain blood glucose concentrations within the desired zone.

For this work there is no limit on the maximum rate of change of insulin and glucose infusion rates, but the rates of subcutaneous insulin and glucose infusion are constrained to:

$$0\frac{mU}{min} \leq U_I \leq 250\frac{mU}{min} 0\frac{mg}{min} \leq U_G \leq 600\frac{mg}{min} \tag{5.3}$$

The formulation in Equation (5.2) [278–280] is generalized for any linear model and includes functionality for systems with direct feedthrough (via matrices D_u and D_d)) as well as announced disturbances. The controller is implemented as a standalone python module, termed PyMPC, and given the generic formulation of the QP Equation (5.2) and thoughtful python class design, the controller as currently coded may be employed for the control of any process or system given a linear plant model *in silico*. The quadratic program at the heart of the controller is formulated in Pyomo [148, 149] and solved using an interior point method [150]. For implementation details, including code as well as documentation for use of the python module in a generic control problem, please consult Appendix E.

5.2.2 Moving Horizon Estimation (MHE)

For linear systems, a Kalman filter is an optimal state estimator; however, it does not allow for inequality constraints. As such, a moving horizon estimation (MHE) scheme was chosen as it allows model states to be upper and lower bounded by physiological constraints ([281]), and in the unconstrained case it reduces to a Kalman filter ([282]).

The MHE for the glucose control system developed here is formulated similar to [283]. Assuming that model-patient mismatch is due to process noise on both the states (ω_k) and inputs (σ_k), as well as measurement noise on the output (ν_k), the state-space model can be written as:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + B\sigma_k + \omega_k \\ y_k &= Cx_k + Du_k + \nu_k \end{aligned} \tag{5.4}$$

For the state-space model in equation (5.4), the MHE is defined by the following quadratic program:

$$\underset{z(k)}{\text{minimize}} \qquad \sum_{i=k-N+1}^{k} \nu_i^T R \nu_i + \sum_{i=k-N+1}^{k-1} \omega_i^T Q \omega_i + (x^e_{k-N+1})^T P^{-1}_{k-N+1|k-N} (x^e_{k-N+1}) \tag{5.5a}$$

$$\text{subject to:} \quad \nu_i = y_i - (Cx_i + Du_i) \tag{5.5b}$$

$$x_{k+1} = Ax_k + Bu_k + B\sigma_k + \omega_k \tag{5.5c}$$

$$\alpha\omega_k = \mathbf{0} \tag{5.5d}$$

$$\beta\sigma_k = \mathbf{0} \tag{5.5e}$$

$$\omega_{i,j} = \omega_{i-1,j}, \quad i = k-N+2\dots k-1, \quad \forall j \in \text{Constant Parameters} \tag{5.5f}$$

$$\text{where:} \quad x^e_{k-N+1} \triangleq x_{k-N+1} - \hat{x}_{k-N+1|k-N} \tag{5.5g}$$

$$z = \left[x^e_{k-N+1}, \omega_{k-N+1}, \dots\ \omega_{k-1}, \sigma_{k-N+1}, \dots, \sigma_{k-1}\right] \tag{5.5h}$$

$$P^{-1}_{k-N|k-N} = P^{-1}_{k-N|k-N-1} + C^T R^{-1} C \tag{5.5i}$$

$$P^{-1}_{k-N+1|k-N} = Q^{-1} - Q^{-1}A\left(P^{-1}_{k-N|k-N} + A^T Q^{-1} A\right)^{-1} A^T Q^{-1} \tag{5.5j}$$

$$\textbf{s.t } P^{-1}_{1|0} = P^{-1}_0$$

Through this formulation unmeasured inputs, or noise on measured inputs (σ) or put another way, unmeasured disturbances, can be estimated. Constraints (5.5d) and (5.5e) in the MHE optimization problem (5.5) force the process noise on manipulated inputs and states that are not estimated to be zero (through suitable selection of vectors α and β). R and Q are matrices that penalize deviations of the model from measurements and added state noise, respectively, and are used to tune the estimator in conjunction with the estimation horizon, N.

The final term in Equation (5.5b) is the arrival cost, which summarizes previous information that is not included in the estimation horizon window. This term can be loosely viewed as the conditional density function, $p(x_{k-N}|y_0, \cdots, y_{k-N-1})$ and penalizes prediction of initial conditions [281]. By penalizing deviations of x_{k-N+1} away from predictions of x_{k-N+1} given previous data and estimates ($\hat{x}_{k-N+1|k-N}$) if there is high confidence in projections of the optimal estimate $\hat{x}_{k-N+1|k-N}$ then the cost of estimating x_{k-N} far away from $\hat{x}_{k-N+1|k-N}$ is large [282].

For unconstrained linear systems, the arrival cost can be expressed explicitly because the MHE optimization simplifies to that of a Kalman filter, and the Kalman filter covariance update formula (Equation (5.5i) and Equation (5.5i)) with initial condition P_0 can be used [284]. In the case of a constrained linear MHE, as implemented here, general analytical expressions for the arrival cost are not available [285]. A commonly used strategy is to approximate the arrival cost by the Kalman update; such is the method used in this implementation of a constrained linear MHE. This has the benefit of reducing to an exact algebraic expression of the arrival cost when all inequality constraints are inactive [285]

In this formulation time-dependent parameters are estimated by reformulating the state space model such that parameters are states with no dynamics:

$$\frac{d\theta(t)}{dt} = 0$$

or in a discrete time formulation

$$\theta_{k+1} = \theta_k$$

This reformulation is used to estimate variable insulin sensitivities, insulin secretion rates, and endogenous glucose production rates, where $\theta_0 + \omega_{k,\theta}$ exactly determines the parameter value at each point, k over the estimation horizon, N and θ_0 is the nominal parameter value. For linear realizations of nonlinear models θ_0 is the operating point of the linearization.

Although not used here, this formulation allows for the estimation of constant parameters (across the estimation horizon) via Equation (5.5f). By enforcing Equation (5.5f) over a subset of estimated parameters (which in this formulation are equivalent to estimated states) the state noise $\omega_{i,j}$ for state/parameter j at point i in the estimation horizon is constrained to be equal to the state noise at the previous point for all points. In this way, the parameter is held constant across the estimation horizon.

The inclusion of this functionality, in addition to allowing for the possibility of feedthrough, meaning model inputs which directly and immediately affect an estimator output, and estimation of unmeasured or noisy inputs, results in a moving horizon estimator that is generalizable across multiple input-multiple output (MIMO) linear problems described with state transition, input, feedthrough and output matrices. As with the model predictive controller (Section 5.2.1), the MHE is implemented as a standalone Python module (given the name CoPyMHE for **Co**nstrained **P**ython **M**oving **H**orizon **E**stimator). The MHE quadratic program is formulated in Pyomo [148, 149] and solved via the open source interior point solver IPOPT [150]. For implementation details, including code and documentation for using the python module in a generic state estimation problem, please consult Appendix F.

5.2.3 Internal Model

Both the zMPC and MHE make use of an internal linear model for predictions and estimation. Here, the basic ICING model, with the addition of the model describing subcutaneous insulin absorption dynamics, is linearized and used as the internal model for both the zMPC and MHE. For a system defined by

$$\frac{dx}{dt} = F(x, t)$$

where

$$x = [x_1, x_2 \ \ldots \ x_n]$$

the linearization is the first order term of the Taylor expansion of $F(x, t)$ around a point of interest, x_0, as given by Equation (5.6)

$$\frac{dx}{dt} \approx F(\underline{x}_0, t) + \nabla F(x_0, t)(\underline{x}(t) - \underline{x}_0) \tag{5.6}$$

where $\nabla F(\underline{x}_0, t)$ is the Jacobian of $F(\underline{x}, t)$ evaluated at the linearization point, $\underline{x}_0$.

Linearization and discretization of the ICING + subcutaneous insulin model is accomplished by finding the Jacobian of the model with respect to all state variables, estimated parameters and inputs ($[BG, Q_I, I, P_1, P_2, Q_{scr}, Q_{1r}, Q_{2r}, SI, U_{en}, U_G, U_I, O_G]$), where U_G, U_I, and O_G are rates of intravenous glucose infusion, subcutaneous insulin infusion and equivalent oral glucose intake, respectively. The resulting Jacobian, broken into the state space representation of the ICING model with dynamics for absorption of subcutaneously delivered regular acting insulin, is given in Table 11 below.

The ICING model equation describing the mass of glucose in the gut (P_2) contains a term that requires finding the maximum of truncated linearly varying transport rate between the stomach and the gut (see Equation (3.1e)). For linearization purposes, the rate is assumed to varying linearly over all ranges, that is to say it is not truncated.

The state space matrices (A, B, and C) in Table 11 are evaluated at ICING model values for all constant parameters and at the virtual patient cohort population averages for S_I and U_{en}. The states ($[BG, Q_I, I, P_1, P_2, Q_{scr}, Q_{1r}, Q_{2r}]$) are evaluated at steady state values from the ICING+SQ model [160] with $S_I = 0.0005\frac{L}{mU \cdot min}$ and $U_{en} = 20\frac{mU}{min}$. Intravenous glucose and insulin infusion rates are assumed to be 0 at steady state, as are subcutaneous insulin infusion rates and enteral glucose intake. The steady state values are given in Table 10.

Table 10: Steady state values for the ICING+SQ model with $S_I = 0.0005 \left(\frac{L}{mU \cdot min}\right)$ and $U_{en} = 20 \left(\frac{mU}{min}\right)$

Variable:	**Steady State Value**
$BG \left(\frac{mg}{dL}\right)$	117.23
$Q_I \left(\frac{mU}{L}\right)$	8.961
$I \left(\frac{mU}{L}\right)$	16.83
$P_1 \, (mg)$	0.0
$P_2 \, (mg)$	0.0
$Q_{sc} \, (mU)$	0.0
$Q_1 \, (mU)$	0.0
$Q_2 \, (mU)$	0.0
$S_I \left(\frac{L}{mU \cdot min}\right)$	0.0005
$U_{en} \left(\frac{mU}{min}\right)$	20

After evaluation of A, B and C at the linearization point, the system is discretized with a 5 minute sampling time to correspond to the CGM sampling rate [286]. Discretization is accomplished through the scipy cont2discrete module [287, 288], which uses the following transformations:

$$A_d = e^{AT} \tag{5.7a}$$

$$B_d = \left(\int_{\tau=0}^{T} e^{A\tau} d\tau \right) B = A^{-1} \tag{5.7b}$$

$$C_d = C \tag{5.7c}$$

Here $\tau = 5$ minutes is the sampling time.

5.2.4 *in silico* Trials

in silico trials are conducted on simulations of all patients within the virtual patient cohort for the entirety of each patients' stay in the critical care unit. For *in silico* trials, clinically administered infusions or injections through any route (subcutaneous or intravenous) are withheld from the virtual patients, however simulated virtual patients still consume meals as indicated in the medical record data. Virtual patients with the responsive pancreatic functionality described previously (Chapter 4) are simulated in python using the scipy [287, 289] *odeint* library, which employs lsoda [290] from the FORTRAN odepack library. Simulations are conducted in 5 minute steps to match the sampling rate of the CGM system [286], where a new CGM "measurement" is taken as the interstitial glucose concentration at the end of 5 minutes simulated time (G_{ISF}, see Equation (3.6)) each time the virtual patient eICING model differential equations are solved.

For each virtual patient, insulin sensitivity (S_I) is calculated by the MHE at each glucose measurement, collected at 5-minute intervals, and is assumed to be constant until the next glucose measurement. The pancreatic insulin secretion rate, U_{en}, is also assumed to remain constant over the 5 minute interval. In Chapter 4 a physiologically-motivated PID control scheme is developed to determine pancreatic insulin secretion rates in response to blood glucose concentrations. Briefly, insulin secretion is assumed to be comprised of a basal component, which is determined by the aforementioned PID controller and a pseudo-random

Table 11: Linearization of the ICING + Subcutaneous Insulin Absorption Model

$$A = \begin{bmatrix} - \frac{Q_{I} S_{I}}{Q_{I} \alpha_{G} + 1.0} - p_{G} & \frac{B G Q_{I} S_{I} \alpha_{G}}{\left(Q_{I} \alpha_{G} + 1.0\right)^{2}} - \frac{B G S_{I}}{Q_{I} \alpha_{G} + 1.0} & 0 & 0 & 0 & 0 & 0 & 0 & - \frac{B G Q_{I}}{Q_{I} \alpha_{G} + 1.0} & 0 \\ 0 & \frac{Q_{I} \alpha_{G} n_{C}}{\left(Q_{I} \alpha_{G} + 1.0\right)^{2}} - \frac{n_{C}}{Q_{I} \alpha_{G} + 1.0} - n_{I} & n_{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & n_{I} & \frac{I \alpha_{I} n_{L}}{\left(I \alpha_{I} + 1.0\right)^{2}} - n_{I} - n_{K} - \frac{n_{L}}{I \alpha_{I} + 1.0} & 0 & 0 & 0 & 0 & \frac{k_{a1r}}{V_{I}} & 0 & \frac{1}{V_{I}} \left(- x_{L} + 1\right) \\ 0 & 0 & 0 & - d_{1} & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & d_{1} & - \frac{d_{2} e^{- P_{2} d_{2}}}{e^{- P_{2} d_{2}} + e^{- P_{max}}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & - k_{scr} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{scr} & - k_{a1r} - k_{vr} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{a1r} & - k_{a1r} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

ICING + Subcutaneous Insulin model state (A) matrix

$$B = \begin{bmatrix} \frac{1}{V_{G}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{V_{I}} \left(- x_{L} + 1\right) \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

ICING + Subcutaneous Insulin model input (B) matrix

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

ICING + Subcutaneous Insulin model output (C) matrix residual component. Here, the insulin secretion rate is taken as the sum of the PID calculated basal component and the residual at the start of the 5 minute interval (as described in Chapter 4). As noted earlier, secretion rates are constrained between $0\frac{mU}{min}$ and $210\frac{mU}{min}$.

At the start of simulated closed-loop glucose control treatment, the MHE and zMPC are both initialized with the A, B, and C matrices from Table 11 evaluated at the linearization point. At each 5 minute time step, the MHE module returns a vector of all states and the two estimated parameters (S_I and U_{en}). The MHE results are inputs to the MPC module and serve as the initial conditions for the MPC internal model. The estimated parameters S_I and U_{en} are assumed to remain constant throughout the prediction horizon. The MPC module returns predictions of the controlled variable (BG in this case) at all steps in the prediction horizon and optimized values for the control variables at all steps in the control horizon. The first values of the MPC optimized control variables (glucose and subcutaneous insulin infusion rates) become the inputs to the virtual patient simulation and remain constant across the 5 minute simulation for that time step.

5.2.5 Simulated CGM Error

To more closely imitate clinical conditions, and the unfortunate but inescapable reality of CGM sensor noise, the zMPC/MHE glucose control system is tested *in silico* with simulated noisy CGMs. In practice, each patient has glucose concentrations measured by two continuous glucose monitors; simulated virtual patient glucose concentrations are also “measured” using two simulated CGMs. The availability of glucose measurements from two sensors necessitates a modification of the C matrix in the MHE such that:

$$C_{MHE} = \begin{bmatrix} C \\ C \end{bmatrix} \tag{5.8}$$

and therefore

$$\begin{bmatrix} BG_{CGM,1} \\ BG_{CGM,2} \end{bmatrix} = [C_{MHE}]\,[G_{ISF}] \tag{5.9}$$

This modification enables the MHE to estimate S_I and U_{en} based on both CGM measurements. If CGM noise is not simulated $BG_{CGM,1}$ and $BG_{CGM,2}$ are identical and estimations would be no different than if a single “measurement” with C were used. When

$BG_{CGM,1} \neq BG_{CGM,2}$ and with symmetric R this modification results in estimations based on the simple average of $BG_{CGM,1}$ and $BG_{CGM,2}$. For asymmetric R, the estimation is based on a weighted linear combination of $BG_{CGM,1}$ and $BG_{CGM,2}$.

CGM noise is simulated here using a first-order moving average model (MA(1), as described in Chapter 2) with parameters taken as the medians from Table 3 for properly functioning sensors. An independent MA(1) process is instantiated for each CGM and realizations of the process are added to G_{ISF} from the virtual patient simulation at eac time step to generate two unique CGM measurements with realistic noise characteristics. If the addition of noise would result in $BG_{CGM,i} < 0$ new values are generated from the MA(1) process until $BG_{CGM,i} + \text{noise} > 0$

Recalibration of the simulated CGMs is performed every 6 hours by re-initializing the integrated moving average error model and letting error accumulate from a random starting error over the next 6 hour window. The CGM recalibration often produces a jump in the measured glucose values as the error decreases significantly due to recalibration. This introduces large derivatives in a number of states in the MHE and results in estimates that introduce corresponding large derivatives int he MPC leading to rapid control moves leading to a temporary loss of control of blood glucose levels. To combat this, the error, for each sensor, at the recalibration point, where error is defined as the difference between the previous CGM measurement and the accurate measurement used for recalibration is subtracted from the previous N CGM measurements (where N is the estimation horizon) used by the MHE. Failure to correct the measurements internal to the MHE at the time of recalibration and subsequent MHE re-initialization was found to result in significant measurement and estimation errors which ultimately lead to poor control and dangerous hyper- or hypoglycemia.

5.2.6 zMPC and MHE Tuning

With a sample time of 5 minutes ($T_s = 5$) the prediction horizon, P, is set at 36 intervals (corresponding to predictions 3 hours into the future) per the common recommendation [291] that $T = PT_s$ where T is the desired closed-loop response time – in this case 3 hours. In general it is recommended that the control horizon (M) is much less than P ($M << P$) for the following reasons [291]:

- Smaller M reduces the computational size of the problem
- $M < P$ is essential for plants with delay. If $M = P$ it is possible that later control moves might not affect plant outputs leading to an ill-posed optimization problem
- Small M promotes stability, although there is no guarantee of stability

For this work it was found that $M = 8$ resulted in good controller performance. Through trial and error the following weighting matrices were found to result in a controller which was resistant to high frequency oscillations in infusion rates or "spiking", especially in the face of CGM noise and which was able to mitigate hyperglycemia while avoiding hypoglycemia.

$$\Gamma = \begin{bmatrix} 30 & 0 \\ 0 & 10 \end{bmatrix}$$

$$S = \begin{bmatrix} 500 & 0 \\ 0 & 0.005 \end{bmatrix}$$

The MHE horizon, N, was set at 36 to utilize measurements over the previous 3 hours to estimate insulin sensitivity and insulin secretion rates. R, the matrix penalizing deviations in model predicted glucose concentrations from measured concentrations is simply taken as 2×2 identity matrix. The matrix is 2×2 and symmetric to account for the availability of measurements from 2 CGMs and the belief that neither CGM is inherently more accurate. Using $R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ as a baseline the weighting matrices Q and P_0 (given below) are assumed to be identical.The values for the first two diagonals take (corresponding to ω_{S_I} and $\omega_{U_{en}}$)their values from the regularization on S_I and U_{en}, respectively, used in developing the virtual patient cohort by virtue of their analogous functionality in the MHE. The latter two diagonal values, corresponding to ω_Q and ω_{EGP} were found via trial and error.

$$Q = P = \begin{bmatrix} 1\times10^{-5} & 0 & 0 & 0 \\ 0 & 1\times10^{8} & 0 & 0 \\ 0 & 0 & 1\times10^{-4} & 0 \\ 0 & 0 & 0 & 1\times10^{-4} \end{bmatrix}'$$

The corresponding vector of state noises is given by:

$$\omega = \begin{bmatrix} U_{en}, & S_I, & Q_I, & EGP \end{bmatrix}$$

A Note on Weighting: A value of 1×10^{-4} is used as the weight on noise for both Q_I and U_{en}, however the ultimate penalty is the product of the weight and squared state noise value. Because U_{en} is generally much larger in magnitude $(0 - 210\frac{mU}{min})$ compared to EGP $(0 - 1.16\frac{mmol}{min})$ in the ICING model the weight, on Q_I represents a significantly larger penalty.

5.2.7 Moving Horizon Estimation and Accuracy Requirements

It has been suggested that in order to safely affect TGC in a critically-ill patient population the accuracy of high frequency CGM measurements, as described by Mean Absolute Relative Deviation (MARD), needs to be less than 11% according to one study [78] or less than 10% according to another [79]. The Dexcom® Platinum™ G4 continuous glucose monitors used here have a MARD of 15.15% which seems to suggest that they may be insufficient for control. However, it is hypothesized here that the use of moving horizon estimation with data from two CGMs and a sufficient recalibration frequency will result in a MARD value for the estimated blood glucose concentration which is sufficient for control. Using simulated CGM error an *in silico* trial was conducted across all in patients in the virtual patient cohort in replicate with N=100. MARD was calculated using the MHE estimated blood glucose concentration as a surrogate measurement and the virtual patient simulated blood glucose concentration as the reference. The results utilizing both a single and dual CGMs and an array of recalibration frequencies are summarized in Figure 40

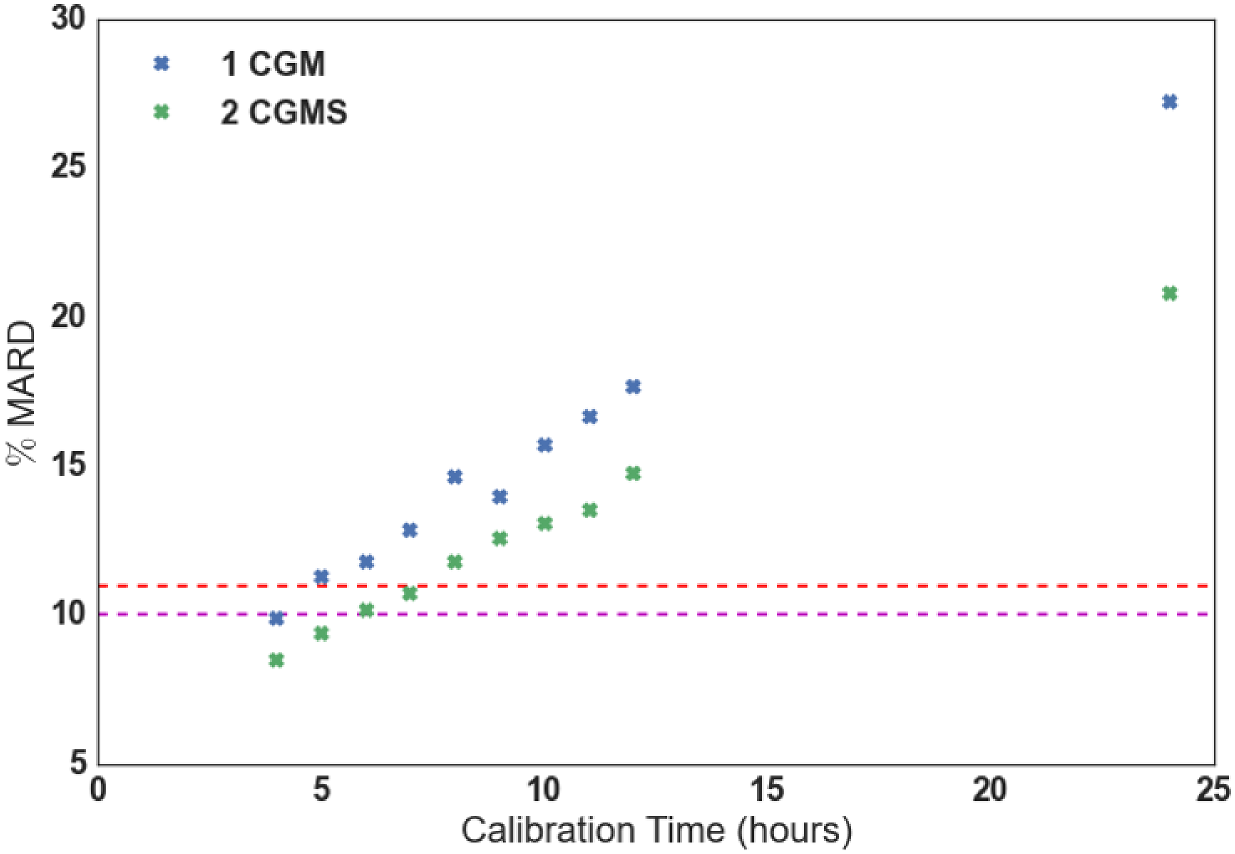


Figure 40: CGM recalibration frequency vs. MARD of estimated blood glucose concentrations across the virtual patient cohort over N=100 trials using 1 and 2 CGMs. With 2 CGMs the MHE fuses the sensor measurements into an overall significantly more accurate estimate of blood glucose concentrations which results in an increase in the requisite recalibration frequency to achieve an acceptable MARD for closed-loop glucose control.

5.3 CONTROL RESULTS

Control results for a simulated virtual patient suffering from significant hyperglycemia are shown in Figure 41, assuming perfectly functioning CGMs with no sensor noise. If simulated CGM noise is added to more closely mimic clinical conditions the closed-loop glucose control action and resulting plasma glucose concentrations are shown in Figure 42.

As seen in Figure 41, in simulations with realistic CGM noise, Figure 42, the zMPC/MHE glucose control algorithm developed here is able to tightly control plasma glucose concentrations in a simulated virtual patient. To demonstrate efficacy and safety, repeated *in silico* control trials with simulated CGM noise are conducted across the virtual patient cohort. In this *in silico* trial, each virtual patient undergoes 100 simulations to account for the stochastic nature of the CGM noise, and statistics are aggregated from all simulated control runs. The results of this *in silico* trial are shown in Figure 43 and Table 12.

5.4 DISCUSSION

Through the use of redundant or dual CGMs and the MHE for sensor fusion, a MARD which is acceptable for TGC is achievable with a calibration period of approximately 6-7 hours despite the deleterious effects of CGM noise. Although this recalibration frequency is significantly greater than would be utilized for ambulatory Type 1 Diabetic patients [286], it represents a significant time-savings over control schemes which require blood sample draws and manual glucose measurement up to every hour [103, 292], which represents a significant clinical workload [59]. If only a single CGM is used, simulation results indicate that the recalibration frequency necessary for acceptable TGC and the associated clinical effort increases by approximately 40%.

Utilizing a zone MPC/MHE control algorithm to automate the delivery of glucose and insulin to the virtual patients resulted in a much higher fraction of time spent within the target glucose zone, a significant decrease in the incidence of hyperglycemia, and dramatic decrease in hypoglycemic events as shown in Figure 43 and Table 12. The zMPC/MHE with

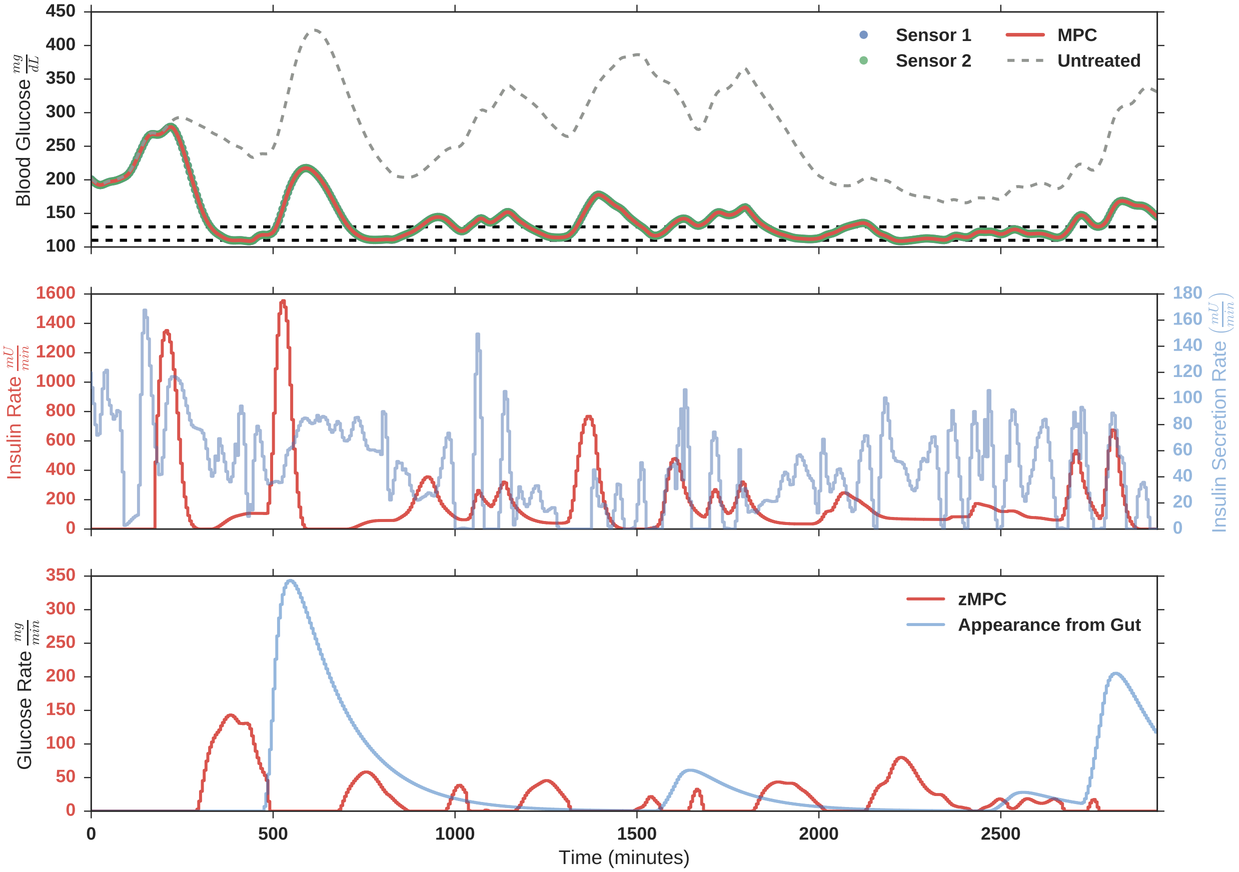


Figure 41: Plasma glucose concentrations (Top) for the simulated virtual patient under zMPC/MHE control (-) compared to plasma glucose concentrations if the patient consumed meals as indicated in the medical record and went untreated (--). Measurements taken by the two CGMs are shown by ● and ●. (Middle) zMPC optimized insulin infusion rate (-) and pancreatic insulin secretion rate (-). (Bottom) zMPC optimized glucose infusion rate (-). Note: Perfect measurements result in perfect agreement between the CGMs and an exact match to simulated virtual patient plasma glucose concentrations.

output regulation algorithm significantly outperforms the clinical protocol in controlling blood glucose levels while completely avoiding any hypoglycemic events. The controller

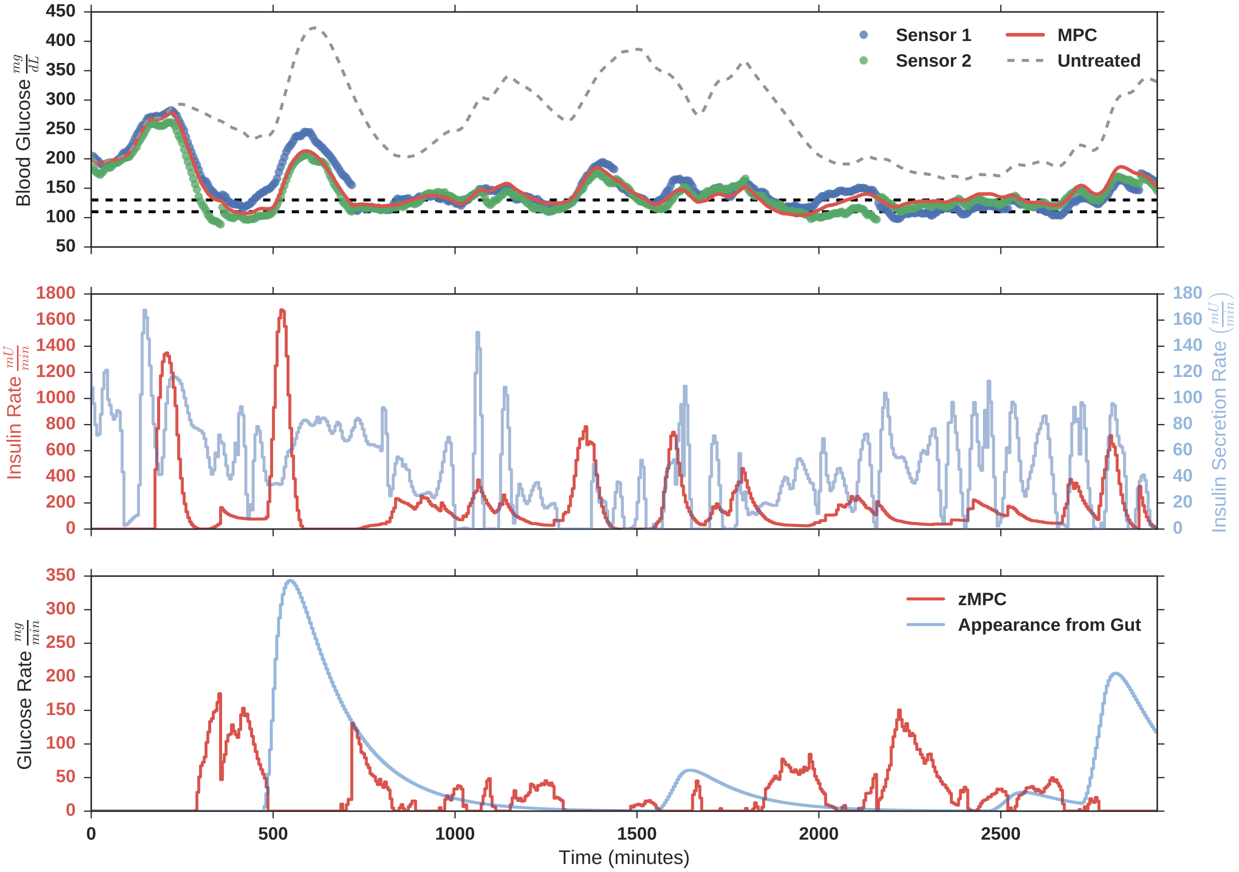
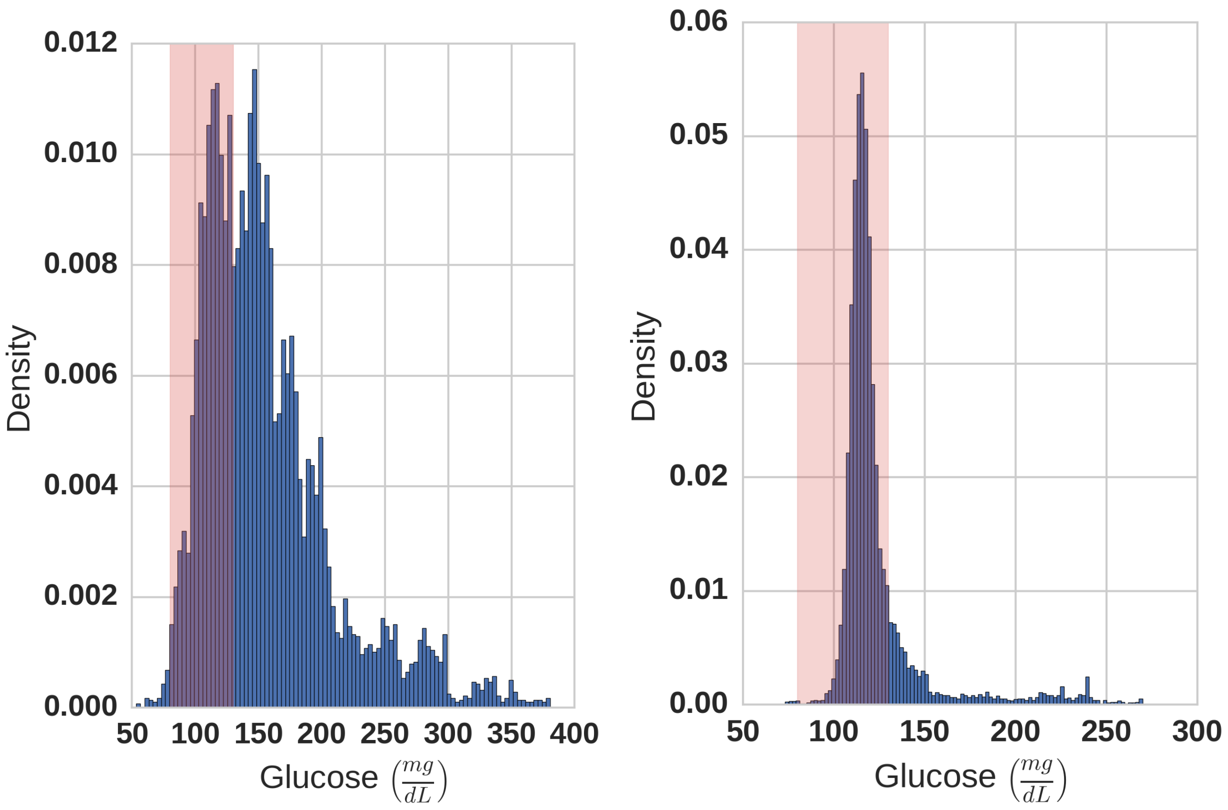


Figure 42: Plasma glucose concentrations (Top) for the simulated virtual patient under zMPC/MHE control (-) compared to plasma glucose concentrations if the patient consumed meals as indicated in the medical record and went untreated (--). Measurements taken by the two CGMs are shown by ● and ●. (Middle) zMPC optimized insulin infusion rate (-) and pancreatic insulin secretion rate (-). (Bottom) zMPC optimized glucose infusion rate (-). Note: Simulated noise on CGM measurements results in mismatch agreement between the CGMs (top panel) as well as simulated virtual patient plasma glucose concentrations.

appears to be well-tuned, delivering insulin in decaying “bursts”, and the output regulation on glucose infusion rates appears to be functioning as intended. Glucose is infused as a rescue to avoid hypoglycemia followed by a return to no glucose infusion once the danger of hypoglycemia has passed.



(a) Clinical Treatment

(b) zMPC/MHE Control

Figure 43: Distribution of blood glucose concentrations resulting from clinical treatment and zMPC/MHE administered treatment. **(a)** Clinical Treatment. **(b)** zMPC/MHE administered subcutaneous insulin and intravenous glucose infusions during an *in silico* trial with simulated CGM noise. Shaded region indicates healthy blood glucose zone ($80\frac{mg}{dL} \leq BG \leq 130\frac{mg}{dL}$)

Table 12: Mean, average standard deviation, minimum and maximum values as well as mean percentage of time spent within relevant zones for the clinical treatment regimen and the *in silico* trial with simulated CGM noise using a zMPC/MHE algorithm to control subcutaneous insulin and intravenous glucose infusions.

	Mean $(\frac{mg}{dL})$	σ $(\frac{mg}{dL})$	Min $(\frac{mg}{dL})$	Max $(\frac{mg}{dL})$	%Time$_{Zone}$	%Time$_{80\leq BG\leq 130}$
Clinic	156.11	26.04	53.44	387.20	23.73	37.39
zMPC/MHE	119.38	13.79	74.14	360.02	75.54	86.59

Examining the example results in Figures 41 and 42 shows that there are significant windows where both subcutaneous insulin and intravenous glucose infusion rates are non-zero. This may initially appear to be the result of poor or improper controller tunings or an improperly estimated model. The intuitive expectation would be a zeroing of insulin infusion rates when glucose infusion rates are non-zero so the controller is not "fighting" itself, and vice versa. However, the fact that subcutaneously, and not intravenously, administered insulin is being used must be considered. Subcutaneously administered regular acting insulin must be absorbed into the blood stream and distributed throughout the body and peripheral interstitial tissue to be effective.

The subcutaneous absorption process is slow compared to distribution in the plasma and results in a delay between administration and action, such that ceasing subcutaneous insulin infusion immediately in response to projected hypoglycemia would result in predicted hyperglycemia at some point during the prediction horizon. Instead, the optimal solution to the MPC formulation results in a continued subcutaneous insulin infusion (albeit at a decreased rate) and impending hypoglycemia is countered with a non-zero intravenous glucose infusion rate.

As a further explanation of the observed overlap between insulin and glucose infusions; consider the structure of the subcutaneous insulin absorption model Equations (3.2) to (3.4) as well as Equation (3.1h). Note that there is only insulin degradation in Q_1, otherwise there is strictly transport from $Q_{sc} \rightarrow Q_1$ and from $Q_2 \rightarrow I$. The transport rate constant, k_{TR} for $Q_2 \rightarrow I$, the final step in regular acting insulin absorption to plasma is approximately 5× slower than n_K (0.0127 min^{-1} vs. 0.0542 min^{-1}), not even considering the saturating rate of plasma insulin disposal or the transport of insulin to the interstitium. All of this is to say that there is a significant tail to insulin action even after subcutaneous infusion has been stopped as insulin continues to appear in plasma due to the relatively slow absorption process from the subcutis. This is in comparison to intravenous insulin infusion in which case the effect of infused insulin will become negligible rapidly following cessation of infusion due to the rapid elimination of insulin from plasma.

Consider also that intravenously infused glucose distributes rapidly in plasma as blood completes a circulation of the body in approximately 2 minutes [293] (Note: in the virtual patients developed here equilibration is immediate). As a result, in response to dropping blood glucose concentrations that may portend hypoglycemia, when subcutaneous insulin is used the optimal control action may not be a total restriction on insulin infusion due to the delay introduced by absorption, but rather a continued infusion (at a lower level to prevent future hyperglycemia) with a rapid rise in glucose infusion rates to combat hypoglycemia. There are controller tunings which reduce the overlap between insulin and glucose infusions, however it requires much more aggressive controller action. The net effect is that overall both more insulin and glucose are infused with these tunings compared to to presented controller tuning results despite the significant overlap.

5.4.1 Announced vs. Unannounced Meals

The zMPC algorithm as designed and implemented here has the functionality to tailor control action around predicted disturbances – in this case patient meals. By announcing future meals the zMPC can take pre-emptive action, by infusing additional subcutaneous insulin before a meal, to ensure that blood glucose concentrations do not spike in response to a meal. The results presented in Figures 41 to 43 do not utilize this feature but rather the control algorithm assumes that there are no non-controller administered sources of exogenous glucose and insulin, including oral intake over the prediction horizon.

A Note on Unannounced vs. Announced Meals: Here unannounced meals imply that the zMPC is totally "blind" to meals and the rate of oral glucose intake over the prediction horizon is exactly zero. For an announced meal the current rate of oral glucose intake is held constant (Zero-Order Hold) across the prediction horizon. In the case of predicted meals, the anticipated rate of oral glucose intake over every step in the prediction horizon for any forecaster meals within the prediction horizon are provided to the zMPC.

The logic behind this decision is perhaps most easily explained with the following demonstration. Here, a virtual patient (different from the one presented in Figures 41 and 42) is placed under the control of the zMPC/MHE algorithm in the following scenarios:

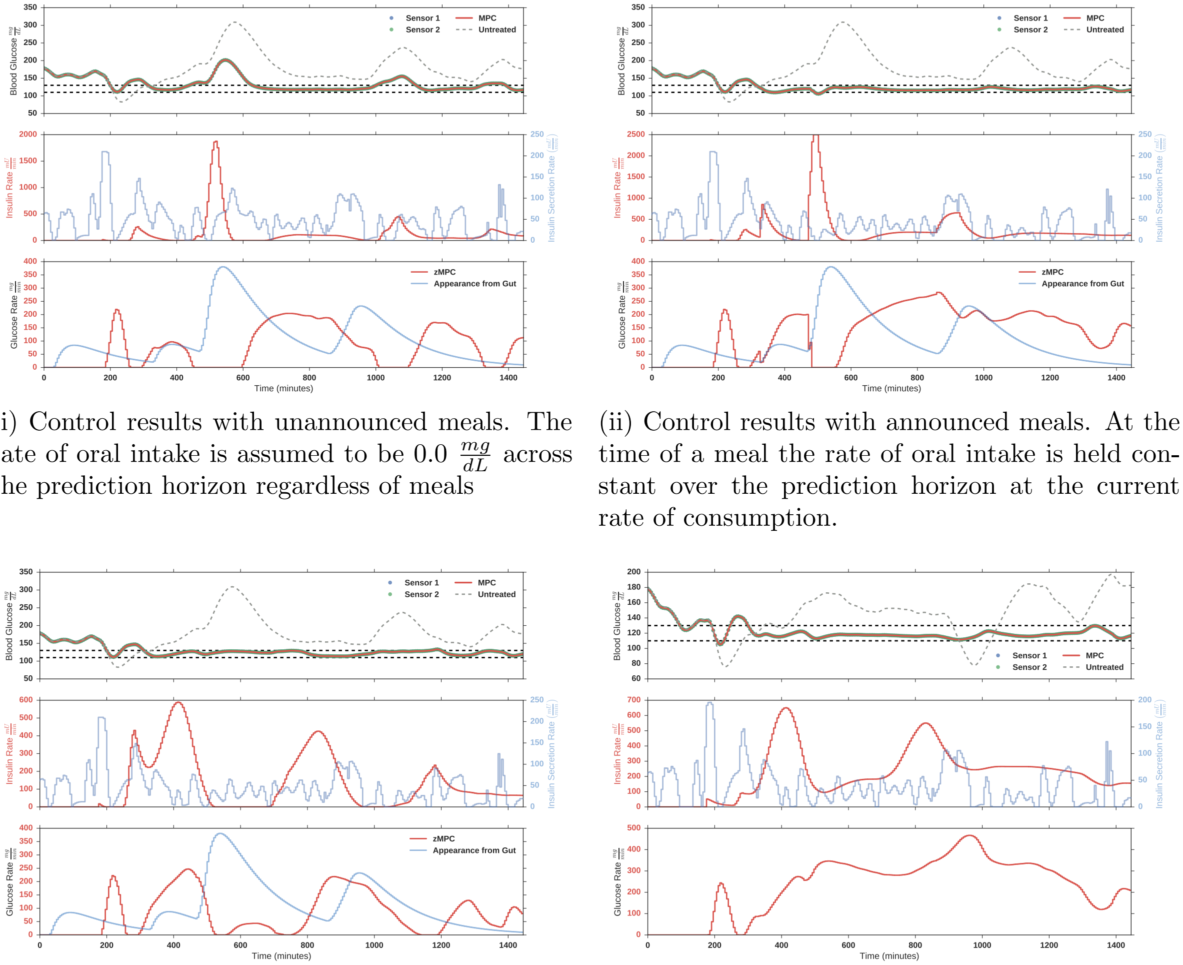
(i) Unannounced meals

(ii) Announced meals

(iii) Predicted meals – consumed as predicted

(iv) Predicted meals – not consumed

Figure 44 shows the control results for a certain virtual patient in all of the above scenarios. For scenarios in which meals are consumed the rate of glucose appearance in the blood from absorption in the guy is shown. As seen in Figure 44, the use of predicted and announced meals results in a significantly better controller to response in meals as evidenced by the lack of spike in blood glucose concentrations around 400 minutes. This result matches expectations, however from a clinical standpoint the use of announced or predicted meals does not necessarily represent the optimal design decision.



(i) Control results with unannounced meals. The rate of oral intake is assumed to be 0.0 $\frac{mg}{dL}$ across the prediction horizon regardless of meals

(ii) Control results with announced meals. At the time of a meal the rate of oral intake is held constant over the prediction horizon at the current rate of consumption.

(iii) Control results with forecasted meal information. The assumed rate of oral intake for the duration of any meal which falls within the prediction horizon is provided to the zMPC.

(iv) Control results with forecasted meal information. The assumed rate of oral intake for the duration of any meal which falls within the prediction horizon is provided to the zMPC. In this scenario meals the forecasted meals are never actually consumed

Figure 44: Control results for a certain virtual patient under four different scenarios demonstrating the affects of unannounced vs. announced meals. CGM error is not simulated here for clarity and ease of comparison across figures. The untreated trajectory (--) in the top panel of all figures shows the simulated virtual patient blood glucose profile in the absence of any zMPC calculated infusions or any clinical treatment.

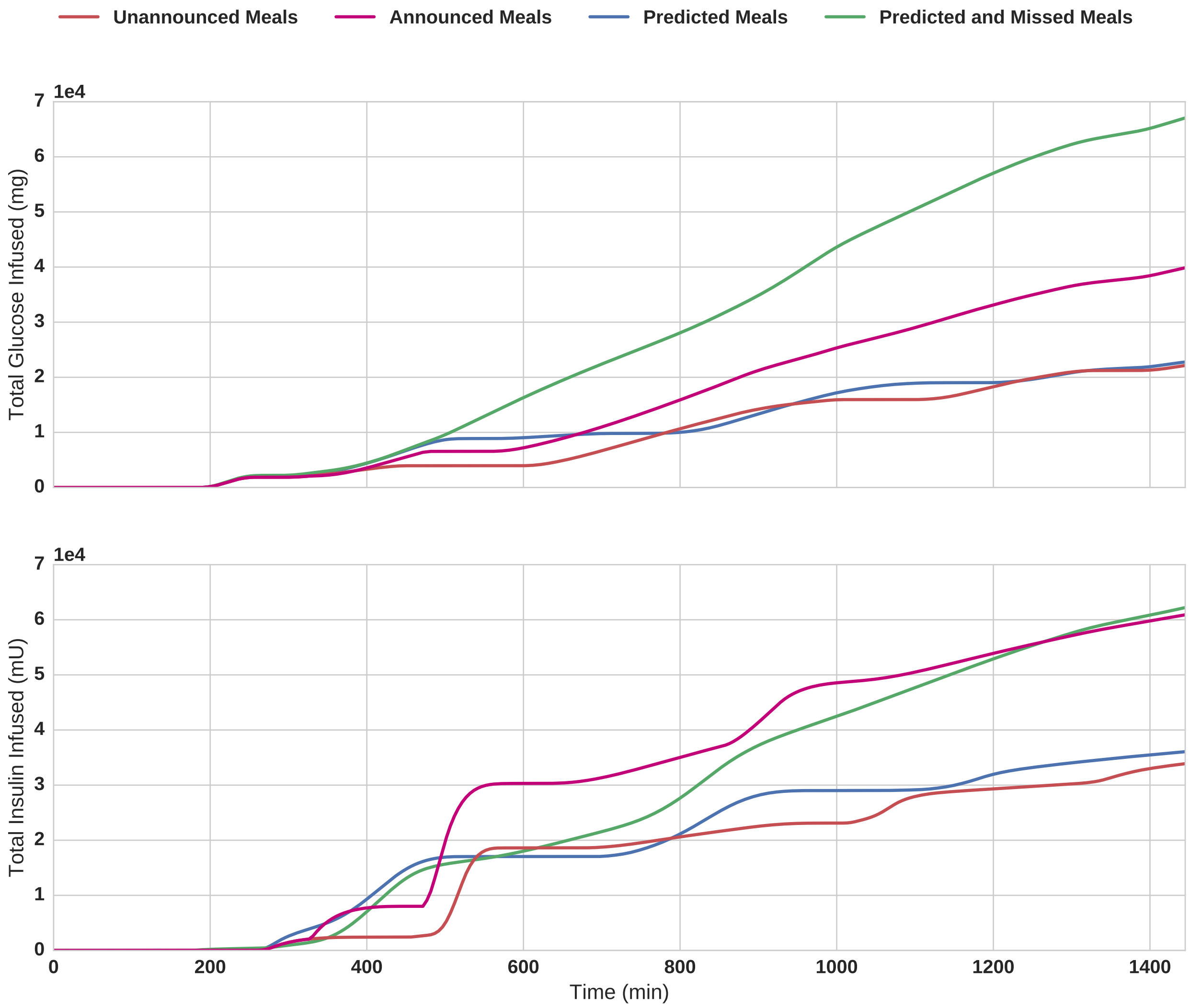


Figure 45: Cumulative plot of intravenous glucose (top) and subcutaneous insulin (bottom) showing the total mass of glucose and insulin delivered over time for the 4 scenarios shown in Figure 44. For all scenarios the MHE is provided accurate meal information.

Figure 45 shows the total amount of intravenous glucose and subcutaneous insulin delivered over time for the 4 scenarios presented above. From this figure and the results shown in Figure 44 it is clear that predicted and announced meals result in good maintenance of glucose levels within or near the zone with a minimum of control effort. In the case of unannounced meals (Figure 44i) insulin infusion rates rise rapidly to very high levels to combat the influx of glucose resulting from consumed meals, however, there is still a small a degree of hyperglycemia resulting from the meal consumed between 400 and 600 minutes. Using an announced meal (Figure 44ii) the controller is able to combat any hyperglycemia with large, rapidly rising rates of subcutaneous insulin infusion but significant rates of glucose infusion are required to avoid hypoglycemia. This is due to the fact that the rate of oral intake is held at a constant elevated value over the prediction horizon (360 minutes) which is far longer than the meal is consumed leading to very (overly) aggressive insulin delivery. If the controller is provided predicted meal information, including the duration of the meal significantly lower rates of insulin infusion are required to completely avoid hyperglycemia as insulin is infused pre-emptively to combat the predicted rise in blood glucose concentrations following a meal. Because the magnitude and duration of oral intake is accurately provided to the controller prior to the meal there is a significantly diminished need for glucose to avoid hypoglycemia following oral intake in this mode of operation.

It would seem then, that predicting meals is the optimum strategy for controlling plasma glucose concentrations to a target zone. However, consider the final scenario (Figure 44iv) in which predicted meals are provided to the controller but ultimately the meals are missed and not consumed. As in the previous scenario (Figure 44iii) the controller ramps up insulin infusion rates pre-emptively to combat the affects of oral glucose intake, however the expected oral intake never arrives requiring significant sustained infusions of glucose to avoid hypoglycemia. In the results presented in Figure 44 hypoglycemia is avoided but at the cost of significantly increased total insulin and glucose infusions (see Figure 45).

Figure 45 shows that unannounced and predicted meals both lead to the infusion of similar masses of insulin and glucose, however, as seen in Figures 44i and 44iii predicted meals result in significantly tighter glucose control within the target zone. Figure 45 also demonstrates the significant increase in both glucose and insulin requirements if announced meals are used and the further increased glucose requirement if predicted meals are used but ultimately the meals are not consumed. For this patient missing predicted meals precipitated a 300+% increase in the mass of glucose infused to prevent hypoglycemia over the scenario where oral intake was as predicted.

Although hypoglycemia is prevented here, this may not always be the case, especially as the upper limit of allowable rates of glucose infusion is reached. Furthermore, the use of announced or predicted meals is predicated upon the assumption of absolute reliability int he glucose infusion system. Any failure in glucose infusion carries with it a significant risk of hypoglycemia and cannot be recommended as the safest mode of operation. For these reason unannounced meals are assumed throughout and the results described in Figures 41 to 43 and table 12 were generated using simulations with unannounced meals.

5.4.2 Comparison to Commercial Solutions

As a point of comparison to commercially available and approved solutions *in silico* trials on the virtual patient cohort with the Computerized Yale Protocol and GlucoStabilizer® Algorithm were conducted.

5.4.2.1 GlucoStabilizer GlucoStabilizer® is marketed as a tool for “trusted, Safe, Effective Glycemic Management” [294] in intensive care units and claims an 87% reduction in hypoglycemia and a 35% reduction in nursing workload [294]. The details of the algorithm are detailed in [276, 295] and for brevity are not reproduced in their entirety here. Briefly, the program recommends an insulin dose based on an insulin sensitivity factor (ISF) which increases or decreases, typically in increments of 0.1, when blood glucose is below or above, respectively, the target range. The recommended insulin dose ($\frac{U}{hr}$) is then calculated as $(BG - 60) \times ISF$ where BG is the current blood glucose concentration. Additionally, GlucoStabilizer® can recommend a bolus of Dextrose 50% based on the formula $(100 - BG) \times 0.4$ when blood glucose concentrations fall below $70\frac{mg}{dL}$. For direct comparison with the model-based control algorithm developed here the target zone in GlucoStabilizer® is set at $110-130\frac{mg}{dL}$. GlucoStabilizer® requires new glucose measurements and adjustments to the insulin infusion rate every hour.

The results of the GlucoStabilizer® on the same virtual patient from Figures 41 and 42 are shown in Figure 46. The aggregate results of blood glucose management using GlucoStabilizer® are shown in Figure 47 and table 13. As seen in Figure 46 the GlucoStabilizer® algorithm significantly reduces blood glucose concentrations but induces an oscillatory glucose trajectory and results in significant hypoglycemia, defined here as blood glucose concentrations $\leq 70\frac{mg}{dL}$. Due to the oscillatory nature of blood glucose concentrations when treated with GlucoStabilizer® the mean blood glucose concentration for this patient is $133.65\frac{mg}{dL}$ which is very close to the target zone. Furthermore, the target blood glucose $(110-130\frac{mg}{dL})$ is reached in 240 minutes (4 hours), which is consistent with claims that the average time to target is 3.84 hours [294]. GlucoStabilizer® further claims a hypoglycemia rate of 0.13% [294] where in GlucoStabilizer defines hypoglycemia as $BG \leq 40\frac{mg}{dL}$. This is born out in the in silico trials here with a hypoglycemia incidence rate of 0.23%. However, it is important to consider the shaded region region of Figure 47 which denotes the minimum mortality zone $(80-130\frac{mg}{dL})$ and note that a significant portion of blood glucose concentration density lies to the left of this zone into a region of rapidly rising mortality rates [121] when using the GlucoStabilizer® algorithm.

Although the GlucoStabilizer® algorithm generally maintains blood glucose concentrations within a target zone and avoids severe hypoglycemia it tends to result in oscillatory blood glucose profiles with relatively large amplitudes, especially compared with the zMPC/MHE controller developed here. Blood glucose variability has been negatively correlated with patient outcomes in a number of studies [296–300, 300, 301]. In fact, according to one study [38] patients with lower mean glucose concentrations but significantly increased glucose variability had an almost five-fold increase in odds of hospital mortality compared to patients with higher glucose concentrations but lower glucose variability. Notice from Table 13 the GlucoStabilizer® decreases mean blood glucose concentrations to the middle of the target zone compared to the clinical protocol but does so at the expense of increased blood glucose variability $(156.11 \pm 26.04\frac{mg}{dL}$ vs. $118.12 \pm 31.49\frac{mg}{dL})$.

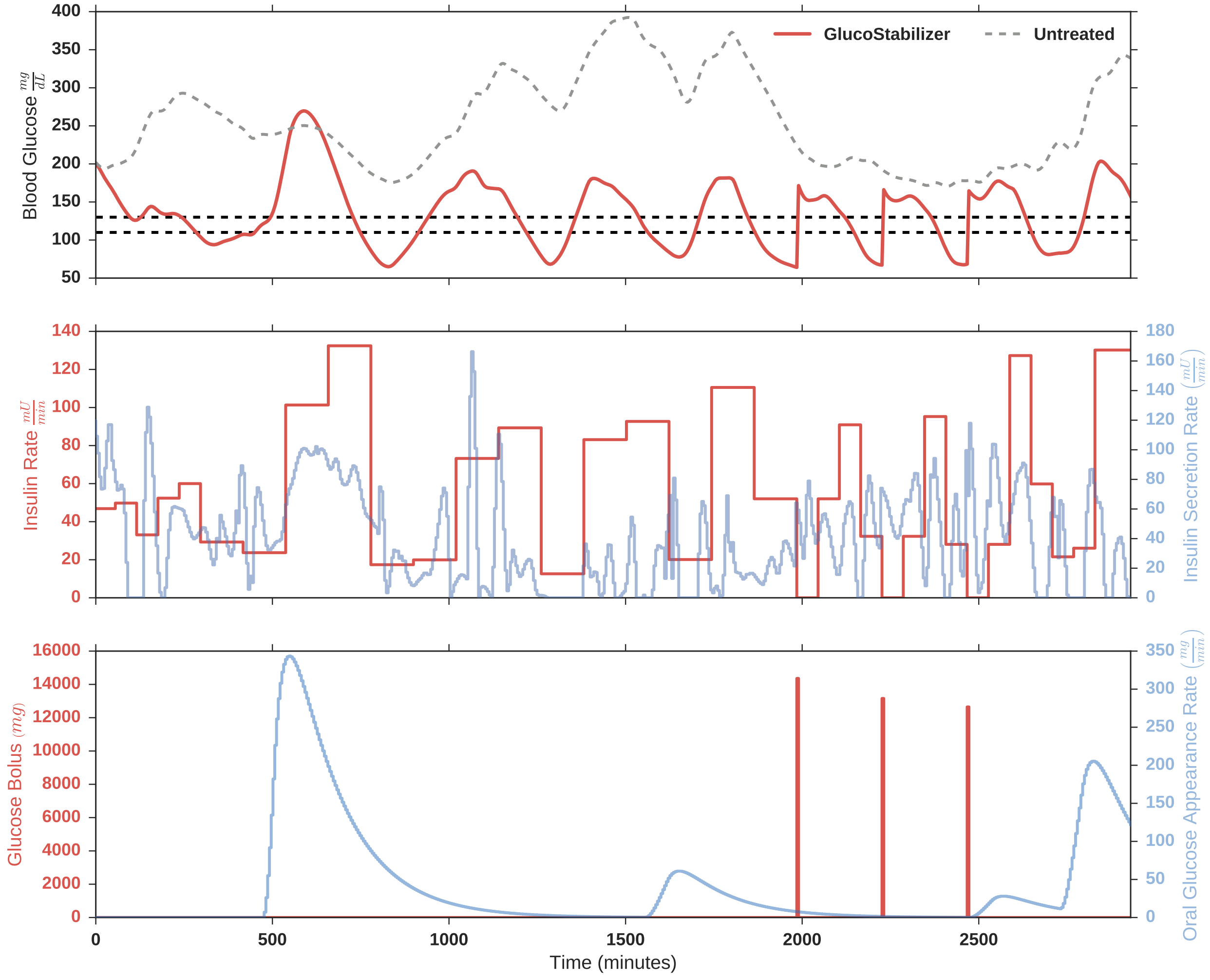


Figure 46: Glucose control using the GlucoStabilizer® algorithm on a virtual patient. The top panel illustrates the GlucoStabilizer blood glucose concentrations with meals compared to the untreated virtual patient consuming the same meals. The middle panel shows the rate of insulin infusion by the GlucoStabilizer® algorithm and the pancreatic insulin secretion response to blood glucose concentrations. The bottom panel shows any glucose boluses recommended and delivered by GlucoStabilizer® as well as the rate of glucose appearance in the blood from oral intake.

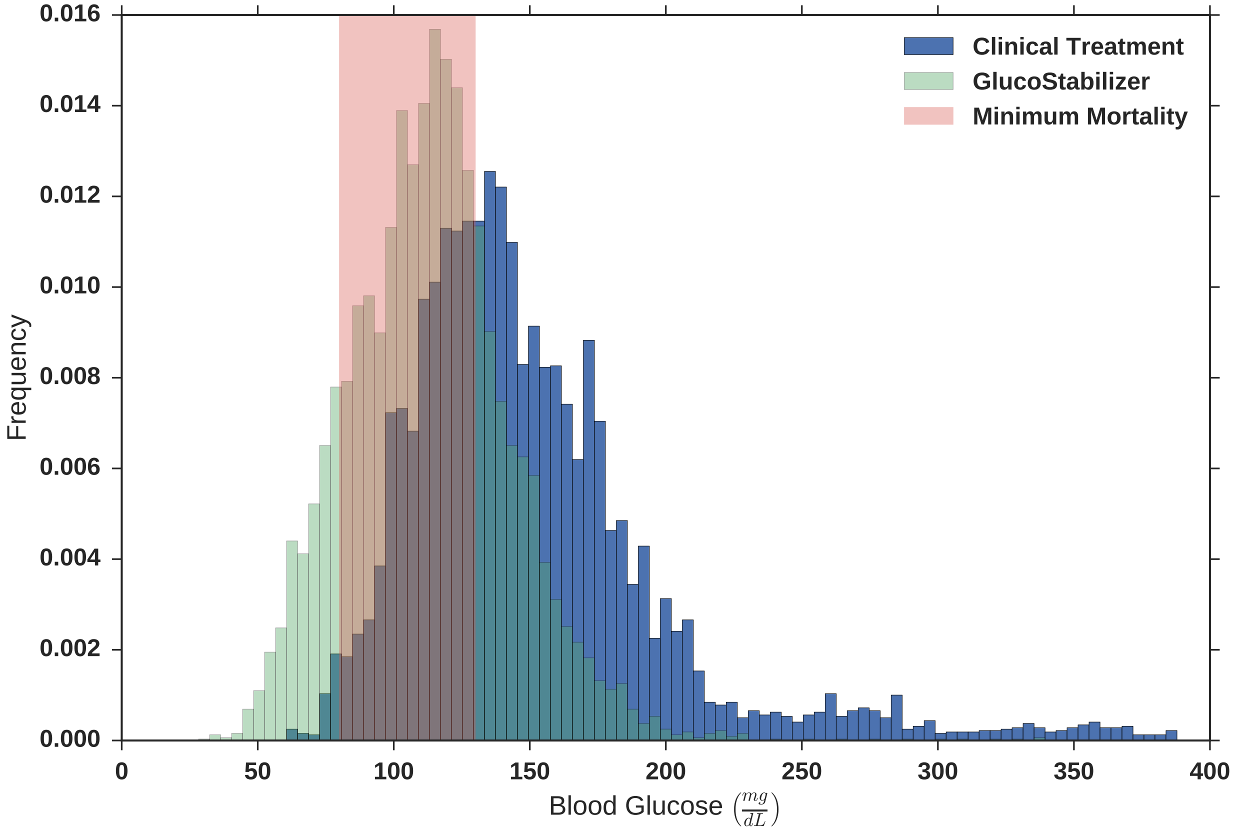


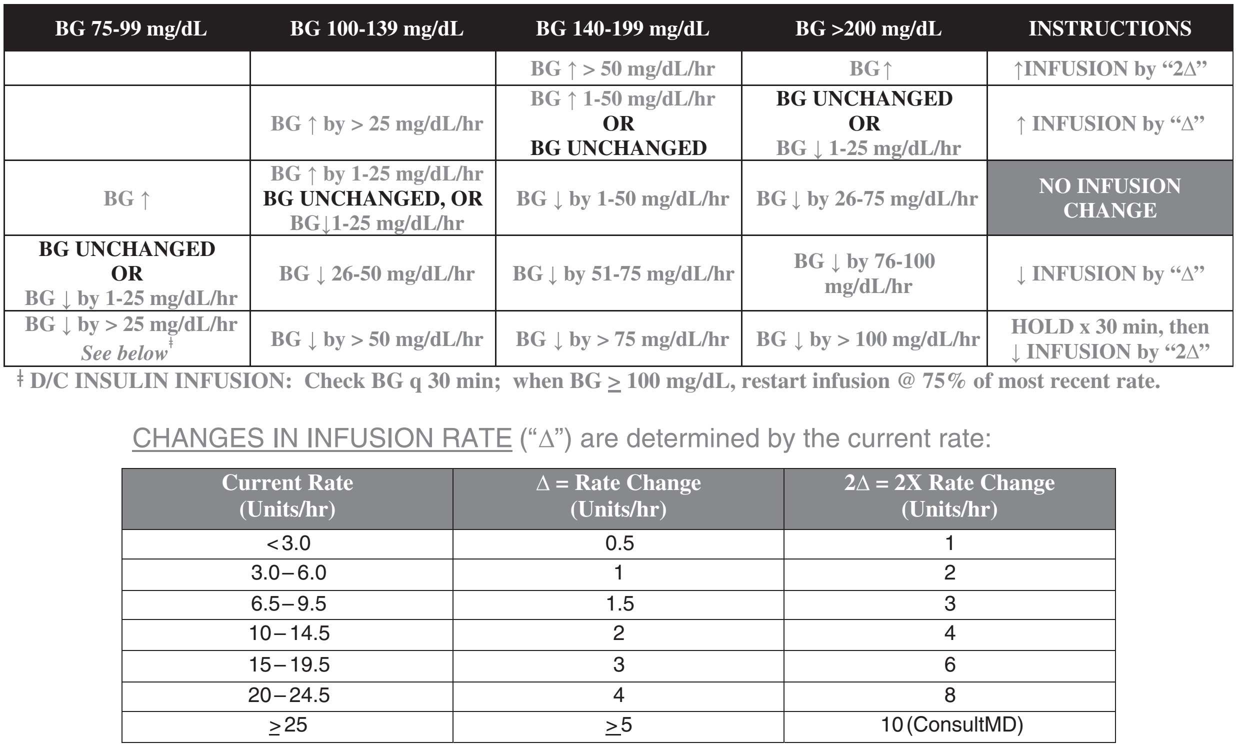
Figure 47: Histogram of blood glucose concentrations in *in silico* trials of the GlucoStabilizer® algorithm on the virtual patient cohort. The histogram for the clinical treatment is provided for comparison.

Table 13: Mean, average standard deviation, minimum and maximum values as well as percentage of time spent within relevant zones for the GlucoStabilizer® in *in silico* trials with the virtual patient cohort.

	Mean $\left(\frac{mg}{dL}\right)$	σ $\left(\frac{mg}{dL}\right)$	Min $\left(\frac{mg}{dL}\right)$	Max $\left(\frac{mg}{dL}\right)$	$\%\text{Time}_{Zone}$	$\%\text{Time}_{80\leq BG\leq 130}$
Clinic	156.11	26.04	53.44	387.20	23.73	37.39
GlucoStabilizer	118.12	31.49	19.24	338.86	30.12	59.75

5.4.2.2 Computerized Yale Protocol: The Computerized Yale Protocol (trademarked as the GlucoCare™IGC System by Pronia Medical Systems, LLC, Louisville, KY) is a FDA cleared insulin-dosing calculator based on the 2004 Yale insulin infusion protocol [302]. The GlucoCare™IGC system targets a blood glucose concentration range of $110 - 140\frac{mg}{dL}$ based on the Yale Protocol by recommending bolus administration of insulin and glucose as well as continuous intravenous infusion of insulin based on a patient's response to prior rates [275]. In essence, the Yale protocol adapts to patients' varying degrees of insulin sensitivity. The details of the GlucoCare™IGC System are detailed in [275] and the basics are given in Table 14.

Table 14: Main instructions from the 2004 Yale Insulin Infusion Protocol (100140 mg/dL). See column 1, row 4 of the first table which demonstrates the recommendation that insulin continue with a BG decrease of 125 mg/dL even while in the range of 7599 mg/dL (reproduced from [275]).



BG 75-99 mg/dL	BG 100-139 mg/dL	BG 140-199 mg/dL	BG >200 mg/dL	INSTRUCTIONS
		BG ↑ > 50 mg/dL/hr	BG↑	↑INFUSION by "2Δ"
	BG ↑ by > 25 mg/dL/hr	BG ↑ 1-50 mg/dL/hr **OR** **BG UNCHANGED**	**BG UNCHANGED** **OR** BG ↓ 1-25 mg/dL/hr	↑ INFUSION by "Δ"
BG ↑	BG ↑ by 1-25 mg/dL/hr **BG UNCHANGED, OR** BG↓1-25 mg/dL/hr	BG ↓ by 1-50 mg/dL/hr	BG ↓ by 26-75 mg/dL/hr	NO INFUSION CHANGE
BG UNCHANGED **OR** BG ↓ by 1-25 mg/dL/hr	BG ↓ 26-50 mg/dL/hr	BG ↓ by 51-75 mg/dL/hr	BG ↓ by 76-100 mg/dL/hr	↓ INFUSION by "Δ"
BG ↓ by > 25 mg/dL/hr *See below*‡	BG ↓ by > 50 mg/dL/hr	BG ↓ by > 75 mg/dL/hr	BG ↓ by > 100 mg/dL/hr	HOLD x 30 min, then ↓ INFUSION by "2Δ"

‡ D/C INSULIN INFUSION: Check BG q 30 min; when BG ≥ 100 mg/dL, restart infusion @ 75% of most recent rate.

CHANGES IN INFUSION RATE ("Δ") are determined by the current rate:

Current Rate (Units/hr)	Δ = Rate Change (Units/hr)	2Δ = 2X Rate Change (Units/hr)
<3.0	0.5	1
3.0−6.0	1	2
6.5−9.5	1.5	3
10−14.5	2	4
15−19.5	3	6
20−24.5	4	8
≥25	≥5	10 (ConsultMD)

The GlucoCare™IGC System is not intended for patients who are consuming intermittent meals and must be handled separately from the protocol's directions. As such, for the *in silico* trial of the GlucoCare™IGC System conducted here with the virtual patient cohort meals are not included. The results of the GlucoCare™IGC System in this *in silico* trial for an example patient (the same patient used in testing the GlucoStabilizer® algorithm and demonstrating the zMPC/MHE control system in Figures 41 and 42) is shown in Figure 48 as an example. As shown in Figure 48 the GlucoCare system significantly reduces blood glucose concentrations through the continuous infusion of intravenous insulin. At two points during treatment dextrose boluses are delivered intravenously to prevent hypoglycemia and for this particular patient severe hypoglycemia ($BG \leq 40\frac{mg}{dL}$) is avoided, however similar to the results observed using the GlucoStabilizer algorithm (see Figure 46) the GlucoCare system induces significant oscillations in blood glucose concentrations and appears to increase blood glucose variability. The aggregated results of the GlucoCare system in this *in silico* trial using the virtual patient cohort are presented in Figure 49 and table 15.

As with the GlucoStabilizer algorithm and the zMPC/MHE control scheme developed here the computerized Yale Protocol of the GlucoCare system successfully shifts the distribution of blood glucose concentrations to lower values and places a larger mass of the distribution within the healthy zone ($80 - 130\frac{mg}{dL}$) as demonstrated in Figure 49. Compared to the GlucoStabilizer algorithm significantly less of the blood glucose density falls to the left of the healthy zone in the region of rapidly rising mortality.

Using the GlucoCare™IGC System mean blood glucose concentrations are reduced to $130.26\frac{mg}{dL}$ with a minimum value of $29.59\frac{mg}{dL}$. The GlucoCare system induces severe hypoglycemia ($BG \leq 40\frac{mg}{dL}$) 0.10% of the time – similar to the rates observed using the GlucoStabilizer system and overall a very low incidence rate. However, as postulated above and as observed with the GlucoStabilizer algorithm, the GlucoCare™IGC System results in a significant increase in mean blood glucose variability over the clinical treatment ($\pm 26.04\frac{mg}{dL}$ to $\pm 41.70\frac{mg}{dL}$) which raises the same concerns and suffers from the same drawbacks as the GlucoStabilizer algorithm discussed previously in Section 5.4.2.1.

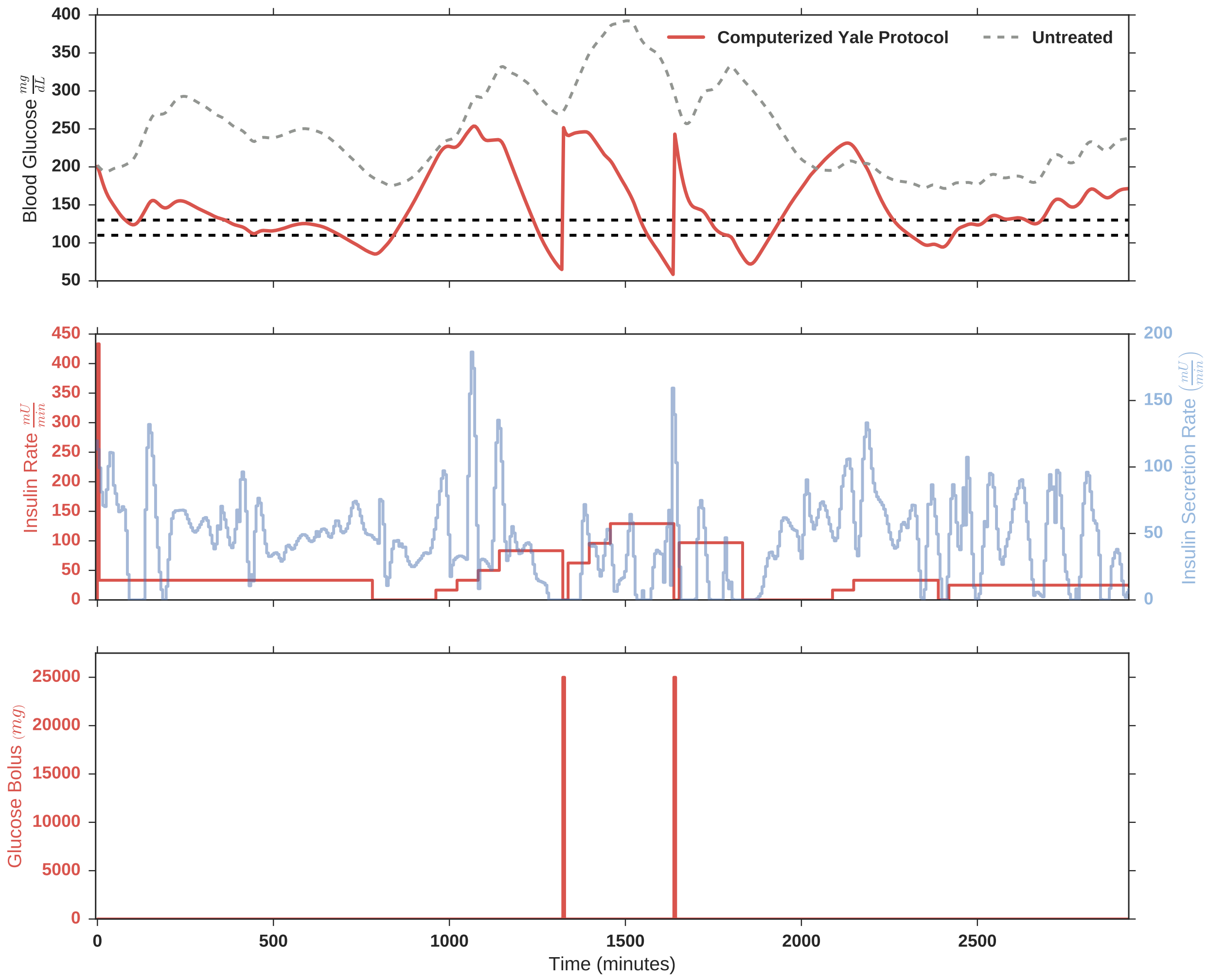


Figure 48: Glucose control using the GlucoCare™IGC System to manage blood glucose concentrations in a virtual patient. The top panel illustrates the GlucoCare system blood glucose concentrations in the absence of meals compared to the untreated virtual patient. The middle panel shows the rate of insulin infusion by the GlucoStabilizer® algorithm and the pancreatic insulin secretion response to blood glucose concentrations. The bottom panel shows any glucose boluses recommended and delivered by the GlucoCare™IGC System.

Both the GlucoStabilizer and GlucoCare systems effectively reduce mean blood glucose concentrations and largely avoid severe hypoglycemic incidents, albeit, with significant blood glucose variability. However, two of the stated driving forces behind the development of both

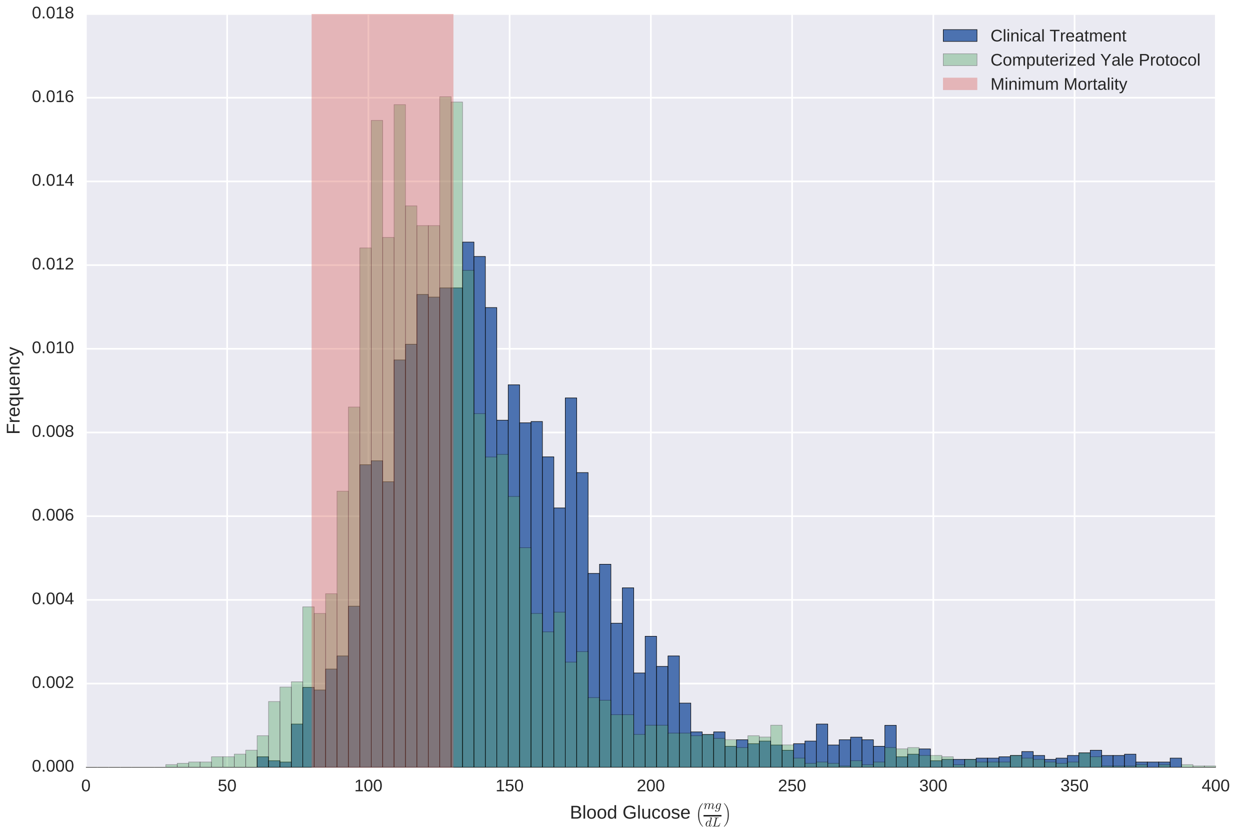


Figure 49: Histogram of blood glucose concentrations in *in silico* trials of the Computerized Yale Protocol on the virtual patient cohort. The histogram for the clinical treatment is provided for comparison.

systems is a reduction in clinical workload and an increase in adherence to protocol [275, 276]. Both systems require, at a minimum a manually obtained blood glucose measurement every hour and adjustment of insulin infusion rates with the possibility of a glucose bolus. In situations where there is a high risk of hypoglycemia the required rate of intervention may increase to once every 15 minutes. Compared to the proposed model-based control system developed here which ostensibly requires manual blood glucose measurements only once every 6 hours for CGM recalibration the FDA approved and commercially available solutions still represent a significantly increased clinical burden over what may be possible.

Table 15: Mean, average standard deviation, minimum and maximum values as well as percentage of time spent within relevant zones for the Computerized Yale Protocol tested in *in silico* trials with the virtual patient cohort.

	Mean ($\frac{mg}{dL}$)	σ ($\frac{mg}{dL}$)	Min ($\frac{mg}{dL}$)	Max ($\frac{mg}{dL}$)	%Time$_{Zone}$	%Time$_{80\leq BG\leq 130}$
Clinic	156.11	26.04	53.44	387.20	23.73	37.39
Computerized Yale Protocol	130.26	41.70	29.59	415.94	28.6	56.3

5.4.3 Summary

In this chapter the efficacy and safety of a model-based controller utilizing continuous subcutaneous insulin infusions and intravenous glucose delivery has been demonstrated. Using the zone model-predictive controller (zMPC) and moving horizon estimation (MHE) system developed and described here blood glucose concentrations in a virtual patient cohort are successfully and tightly maintained within a target zone.

In silico trials on the virtual patient cohort were successfully employed to tune zMPC/MHE controller and further used to justify the decision to operate the controller in a mode in which patient meals are unannounced. In the *in silico* trials the zMPC/MHE was able to significantly reduce average blood glucose levels compared to the clinical standard of care (156.11 to 119.38 $\frac{mg}{dL}$), reduce blood glucose variability by 50% and maintain blood glucose concentrations within a zone of minimum mortality ($80 - 130\frac{mg}{dL}$) 85.69% of the time while completely avoiding hypoglycemia, even with normal, or expected CGM error.

The significantly improved glucose control demonstrated with the zMPC/MHE system on the virtual patient cohort is achieved using subcutaneous insulin thereby recognizing all the benefits of subcutaneous delivery and still outperforms other commercially available solutions, especially on measures of glucose variability, such as GlucoStabilizer® and the GlucoCare™IGC System which utilize intravenous insulin delivery. If intravenous insulin was utilized, the model-based system developed here would likely see a further increase in performance with a higher percentage of glucose measurements falling within the target zone and zone of minimum mortality. When the control performance of the model-based zMPC/MHE glucose control system developed here is compared with commercially available and FDA approved systems in *in silico* trials with a realistic virtual patient cohort it becomes clear that this model-based system may be a safe, effective and viable path to closed-loop glucose control to a target, all while reducing the burden or workload imposed on clinical staff when tight glucose control is considered.

6.0 ALARMS AND FAULT DETECTION FOR CLOSED-LOOP MODEL-BASED GLUCOSE CONTROL

6.1 INTRODUCTION

The automation of advanced systems allow for more efficient and safe operation, however an important consideration in any process under automatic control is the response or actions required when one or more components fail [303]. Like all closed-loop control systems, the glucose control scheme developed here is comprised of measurement devices, the continuous glucose monitors, an actuator, the glucose infusion pump and insulin infusion set, and the zone model-predictive controller with moving horizon estimator. Failure of either the measurment device or the actuator significantly diminish the achievable controller performance and negatively impact patient safety. In this section techniques to identify faults or failures in the measurement and actuation mechanisms will be discussed and evaluated via *in silico* trials. Furthermore, a system to assess the “health” of the estimator and controller with a simple clinically relevant presentation is developed and demonstrated.

Subcutaneous continuous glucose monitors may fail due to a significantly inhibited responses to glucose concentrations due to pressure induced losses of sensitivity (PILS) [81, 143, 144] and long term attenuation of sensor sensitivity [304–306] or due to improper calibration leading to bias and the potential for significant drift. Due to the variability in insulin sensitivity, the possibility for rapid shifts in insulin sensitivity, and the not yet well characterized or modeled etiology of insulin resistance, it is extremely difficult to identify the cause of unexpected variations in blood glucose as a change in insulin sensitivity or a potential CGM fault. Fortunately, the use of paired or redundant CGMs in this work helps limit the severity of this issue by enabling the controller to track instantaneous and accumulated error between the sensors. Furthermore, techniques for identifying PILS events in CGMs [307, 308] have been developed based on maximal physiologic rates of change, which can be employed here.

The potential for failure in control action stems largely from the possibility of insulin infusion set failure. During normal infusion set operation, an infusion pump creates a bubble of insulin under the skin which then osmoses into the blood stream where it affects glucose uptake [307]. The body may react to the infusion set and exogenous insulin through swelling at the site. Swelling and skin contortions may allow insulin to leak out of the body which creates a mismatch between controller commanded insulin and the mass of insulin actually delivered, resulting in poor control due to inadequate insulin [307].

The degradation in control resulting from an insulin infusion set failure is confounded in the control system developed here due to the use of a moving horizon estimation scheme to keep the internal controller models current and accurate. An unaccounted loss of insulin at the infusion site leading to a low circulating concentration of effective insulin will result in some inaccurate combination of reduced estimates of insulin sensitivity, reduced estimates of endogenous insulin production and increased estimates of the rate of endogenous glucose production. The inaccuracies in parameter estimates due to an insulin infusion set failure will further degrade controller performance and may lead to significant hypoglycemia when the infusion set fault is rectified as the likely controller response to what appears to be an ineffective rate of insulin infusion is an increase in the infusion rate.

The primary strategy for the mitigation of insulin infusion set failures is replacement of the infusion set every three days [129]. Significant variability in the lifespan of insulin infusion sets [309] means that sets may fail before they are changed, necessitating a means of detecting an infusion set failure. Here, taking the work of Tyler *et al.* [310] and Bemporad *et al.* [311] as inspiration, the ICING model state space is augmented with an additional disturbance which is used to identify infusion set failures.

Problems with alarms in health care systems have been noted for decades [312–315], and alarm fatigue [316], associated with excessive alarms, has been linked to over 200 deaths between January 2005 and May 2013 [317]. So, although CGM and controller faults need to be handled in a fail-safe manner to ensure patient safety, automated responses to such situations should be designed and tuned to minimize false alarms.

6.2 METHODS

6.2.1 CGM Performance Degradation

6.2.1.1 Faulty Calibration and CGM Drift The use of paired or redundant continuous glucose monitors provides a convenient means of determining potential CGM by offering a reference point for each CGM in the form of a measurement from the paired sensor. The following two simple metrics are used here to identify excessive CGM error and are tested in *in silico* trials:

(i) Percent deviation between CGMs (PDC)

(ii) Cumulative area between CGMs (ABC)

If both sensors were functioning perfectly, exact agreement between glucose measurements from the two CGMs would be expected. As one, or both, CGMs begin to drift, the deviation between the measurements from the sensors might be expected to grow. If an excessive deviation, expressed as a percent error between the CGMs (Equation (6.1)) is detected, it is grounds for an immediate alarm per metric (i).

$$PDC = \frac{|CGM_1 - CGM_2|}{\frac{1}{2}(CGM_1 + CGM_2)} \times 100 \tag{6.1}$$

The use of metric (i) for detection of CGM faults and alarm purposes utilizes a high percent error threshold (30%) to avoid false positives but enables rapid detection of excessive CGM error.

Metric (i) enables rapid detection of excessive deviation but does not account for a chronic less severe deviation between the CGMs, which may eventually manifest as hyper- or hypoglycemia as a result of controller offset. Metric (ii) integrates the difference in glucose measurements between the two CGMs at 5 minute intervals over a sliding 3 hour window (Equation (6.2)).

$$ABC = \sum_{t=0}^{180} 5\left(CGM_1(t) - CGM_2(t)\right) \tag{6.2}$$

This metric is slower to detect faults than the aforementioned metric (PDC) but enables a high threshold for PDC as errors that fall below the PDC threshold will accumulate and signal a problem via metric (ii). The error threshold for this metric is set based on a heuristic which assumes the sensors are diverging at a physiologic maximum rate of glucose change ($5\frac{mg}{dL \cdot min}$ [318]) continually for 20 minutes. This extreme rate of change for an extended period of time is assumed to occur rarely in the clinic, and therefore likely denotes a potential fault in CGM measurements.

To simulate faulty continuous glucose monitors *in silico*, a stochastic additive error is generated using a first order moving-average model with the parameters identified from faulty sensor data in Section 2.3.1. Compared to random noise in nominally functioning sensors, the error signals for faulty sensors are characterized by comparatively large biases and variability (see Chapter 2 for a discussion of the differences between nominally functioning and faulty CGMs).

6.2.2 Infusion Set Failure

To rapidly detect an infusion set failure, an additional input, $U_{SC,D}$, is added to Equation (3.2) such that:

$$\frac{dQ_{SC}(t)}{dt} = U_{SC}(t) - k_{SC}Q_{SC}(t) - U_{SC,D} \tag{6.3}$$

Here $U_{SC,D}$ is a disturbance in subcutaneous insulin infusion rates. As discussed in Section 5.2.2, the linear moving horizon estimation algorithm developed in this work is able to estimate unmeasured disturbances in addition to states and parameters. This feature is used here to allow the estimator to account for "missing" insulin in the event of an infusion set failure. A positive value for $U_{SC,D} \geq 0$ represents a decrease in the rate of subcutaneous insulin infusion analogous to a failed and leaking insulin infusion set. In the linearization of the ICING+SQ model augmented with $U_{SC,D}$, the nominal operating point for $U_{SC,D}$ is taken as $0\frac{mU}{min}$. An additional row and column are added to both Q and P, the MHE weighting matrices, to account for $U_{SC,D}$. Through *in silico* trials, the matrix elements corresponding to $U_{SC,D}$ are tuned such that $U_{SC,D}$ does not deviate significantly from zero except in the case of simulated infusion set failures. The onset of an insulin infusion set failure is simulated by withholding zMPC-recommended insulin infusions from the virtual patients at a randomly selected time. An infusion set failure is considered to have been detected when estimations of the disturbance variable $U_{SC,D}$ rise above a specified threshold. Through these stochastic *in silico* trials, the average time to detection, as well as false and true positive rates, can be determined.

6.2.3 Clinical Interface

To provide constant feedback about controller operation in a simple, easy to parse manner, a clinically facing "traffic light" system is introduced that codifies the projected ability of the controller to maintain euglycemia by color. The three defined regimes correspond to the colors green, yellow, and red, and are rooted in controller projections of its ability to maintain glycemic control as detailed here:

- **Green**: The mean projected blood glucose concentration over the prediction horizon falls within the target zone, and no hypoglycemia ($BG \leq 74\frac{mg}{dL}$) or hyperglycemia ($BG \geq 180\frac{mg}{dL}$) is projected.

- Yellow: The mean projected blood glucose concentration over the prediction horizon falls outside of the target zone, but no hypo- or hyperglycemia is projected.

- **Red**: A hypo- or hyperglycemic event is projected despite controller action, **OR** a CGM fault or infusion set failure is detected as described above

6.3 RESULTS

6.3.1 CGM Fault Detection

The results of the CGM fault detection algorithm described above in Section 6.2.1.1 in closed-loop with error simulated using the model developed in Section 2.3.1 for malfunctioning or improperly calibrated CGMs is demonstrated on a virtual patient in Figure 50.

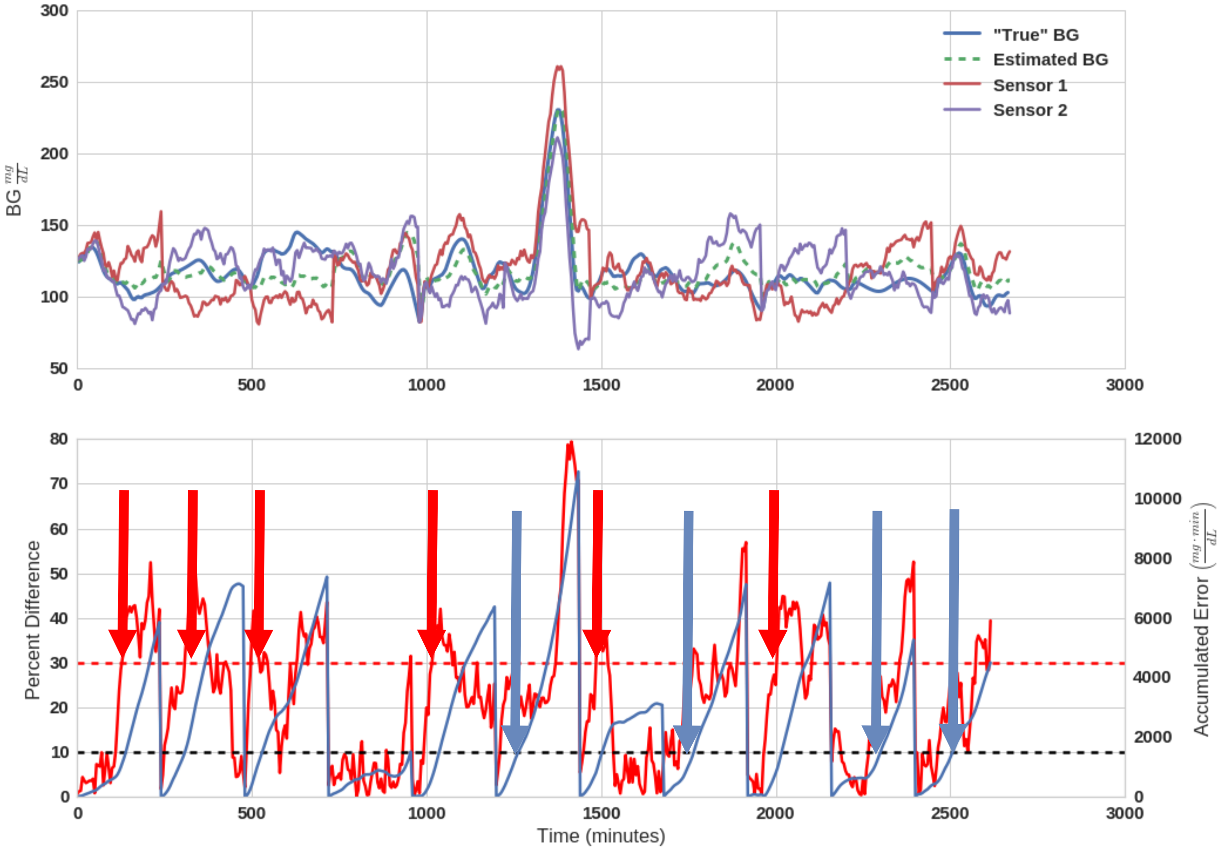


Figure 50: Plasma glucose concentrations (Top) as measured by two "faulty" CGMs (--, -) compared to estimated concentrations (--) and true blood glucose concentration (-). (Bottom) CGM Fault metrics: Metric (i.-) percent deviation between CGMs, Metric (ii.-) cumulative area between CGMs. The colored arrows indicate when the corresponding CGM fault metric reaches the alarm threshold.

Table 16: Percentage of CGM measurements within the zone, above, and below the zone, as well as the percentage of points classified as either hypo- or hyperglycemic for the clinical treatment, the zMPC/MHE algorithm on virtual patients with undetected simulated CGM faults, and with CGM faults detected by the metrics detailed above. For *in silico* trials with simulated faulty CGMs, each virtual patient was simulated in replicate with N=100 to calculate aggregate statistics.

	%Zone	%Above	%Hyper $\left(BG \geq 180\frac{mg}{dL}\right)$	%Below	%Hypo $\left(BG \leq 74\frac{mg}{dL}\right)$
Clinic	23.73	69.89	17.79	1.20	0.0
zMPC/MHE (undetected)	54.11	16.84	2.93	27.3	5.50
zMPC/MHE (detected)	61.49	16.84	2.93	21.65	1.93

Assuming a worst-case scenario of always improperly calibrated CGMs an *in silico* trial was conducted using simulated faulty CGMs and the fault detection metrics to ascertain the efficacy of the alarm metrics. The results of 100 such simulations or each patient in the virtual patient cohort are displayed in Table 16 for both undetected and detected CGM faults. In the latter scenario, when a fault is detected recalibration is simulated by resetting the integrating error process, although the error model continues to be parameterized with the faulty parameter set to capture the worst case scenario where improper (re-)calibration is assumed.

6.3.2 Infusion Set Failure

It was found that including the estimated infusion set failure disturbance variable, $U_{SC,D}$ with:

$$Q = P = \begin{bmatrix} 1 \times 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 1 \times 10^{8} & 0 & 0 & 0 \\ 0 & 0 & 1 \times 10^{-4} & 0 & 0 \\ 0 & 0 & 0 & 1 \times 10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 1 \times 10^{-3} \end{bmatrix}$$

where the corresponding vector of estimated noise is given by:

$$\omega_d = \begin{bmatrix} \omega, & d \end{bmatrix}' = \begin{bmatrix} U_{en}, & S_I, & Q, & EGP, & U_{SC,D} \end{bmatrix}'$$

results in low estimated values for $U_{SC,D}$ (on the order of $\approx\leq 5\frac{mU}{min}$, which is generally insignificant compared to controller prescribed insulin infusion rates). During an infusion set failure, the magnitude of $U_{SC,D}$ increases by orders of magnitude.

The threshold used for detecting an infusion set failure is not a constant but rather taken as a percentage of the current insulin infusion rate. Because the inclusion of $U_{SC,D}$ in Equation (6.3) directly subtracts from the rate of insulin infusion, the magnitude of $U_{SC,D}$, or the rate of insulin leakage from a railed infusion set, would be expected to be proportional to the insulin infusion rate. The following metric (ISF_D, see Equation (6.4)) is used to detect infusion set failure:

$$ISF_D(t) = \begin{cases} 0, & \text{for } 0 \leq U_{SC} \leq 10 \\ \frac{U_{SC,D}(t)}{U_{SC}(t)} \times 100, & \text{for } 10 > U_{SC} \end{cases} \tag{6.4}$$

This formulation is used to avoid divide by zero scenarios at low levels of subcutaneous insulin infusion. The threshold used here to detect an infusion set failure is $ISF_D(t) \geq 25$ for any t – corresponding to an estimated loss of at least 10% of the infused insulin.

As an example, the simulation in Figure 51 shows an infusion failure which is initiated at $t = 600$ minutes and which is rectified at $t = 1500$ minutes.

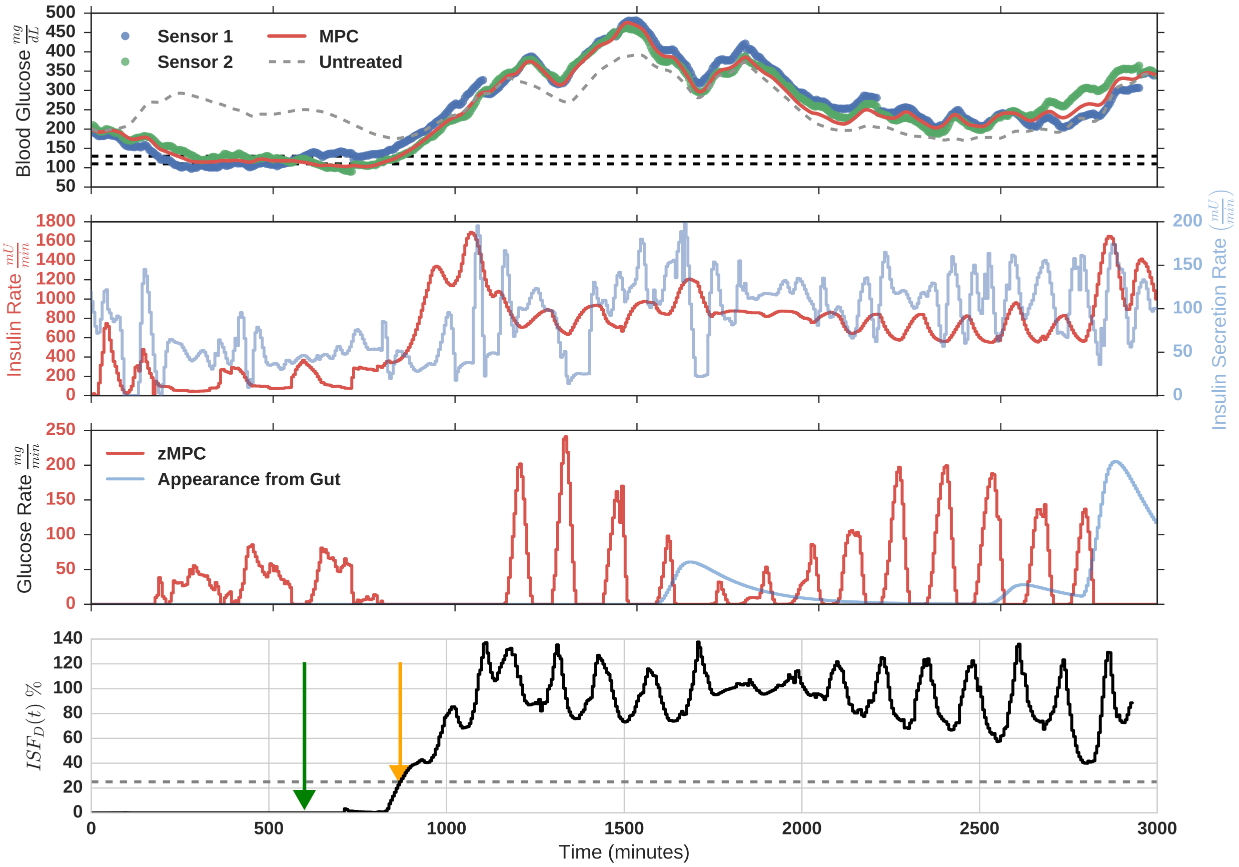


Figure 51: Plasma glucose concentrations (Top) for the simulated virtual patient under zMPC/MHE control (-) compared to plasma glucose concentrations if the patient consumed meals as indicated in the medical record and went untreated (--). Measurements taken by the two CGMs are shown by ● and ●. (2nd Panel) zMPC optimized insulin infusion rate (-) and pancreatic insulin secretion rate (-). (3rd Panel) zMPC optimized glucose infusion rate (-). (Bottom) $ISF_D(t)$ shown in black with the infusion set failure detection threshold indicated by dashed line (--). The green arrow indicates the onset of infusion set failure and the orange arrow indicates when the failure was detected.

Using an estimated insulin infusion disturbance to identify infusion set failures in stochastic *in silico* trials 82.6% of infusion set faults were identified with an average time to detection of 5.8 hours. Furthermore, even with a relatively low detection threshold of 10% the false positive rate over 1700 simulations was 2%.

6.3.2.1 Controller "Health" Figure 52 demonstrates the output of a possible clinician facing interface which provide a means to rapidly parse the projected ability of the controller to control glucose concentrations over the next 3 hours. Every five minutes during the *in silico* experiment a new pair of CGM measurements is received, the MHE estimates a new set of states and parameters, andthe zMPC uses these estimates to optimize M control moves to maintain blood glucose concentrations within the target zone over a prediction horizon of length P. Based on the criteria outlined above in Section 6.2.3, each new measurement, estimation and prediction results in a color assignment to the controller operation, which is displayed in Figure 52.

6.4 DISCUSSION

As seen in Figure 50, fault metric (i.) is able to rapidly detect significant disagreement in CGMs and fault metric (ii.) enables detection of slowly accumulating error from CGM drift. The statistics aggregated in Table 16 show that the use of the CGM fault detection metrics results in a nearly three-fold decrease in hypoglycemia with no change in hyperglycemia – a significant improvement. Furthermore, this reduction in hypoglycemia is realized with an average of only 3.75 alarms per 8 hour ICU shift. This represents a significant savings in workload and easing of the monitoring burden compared to the previously discussed commercially available solutions.

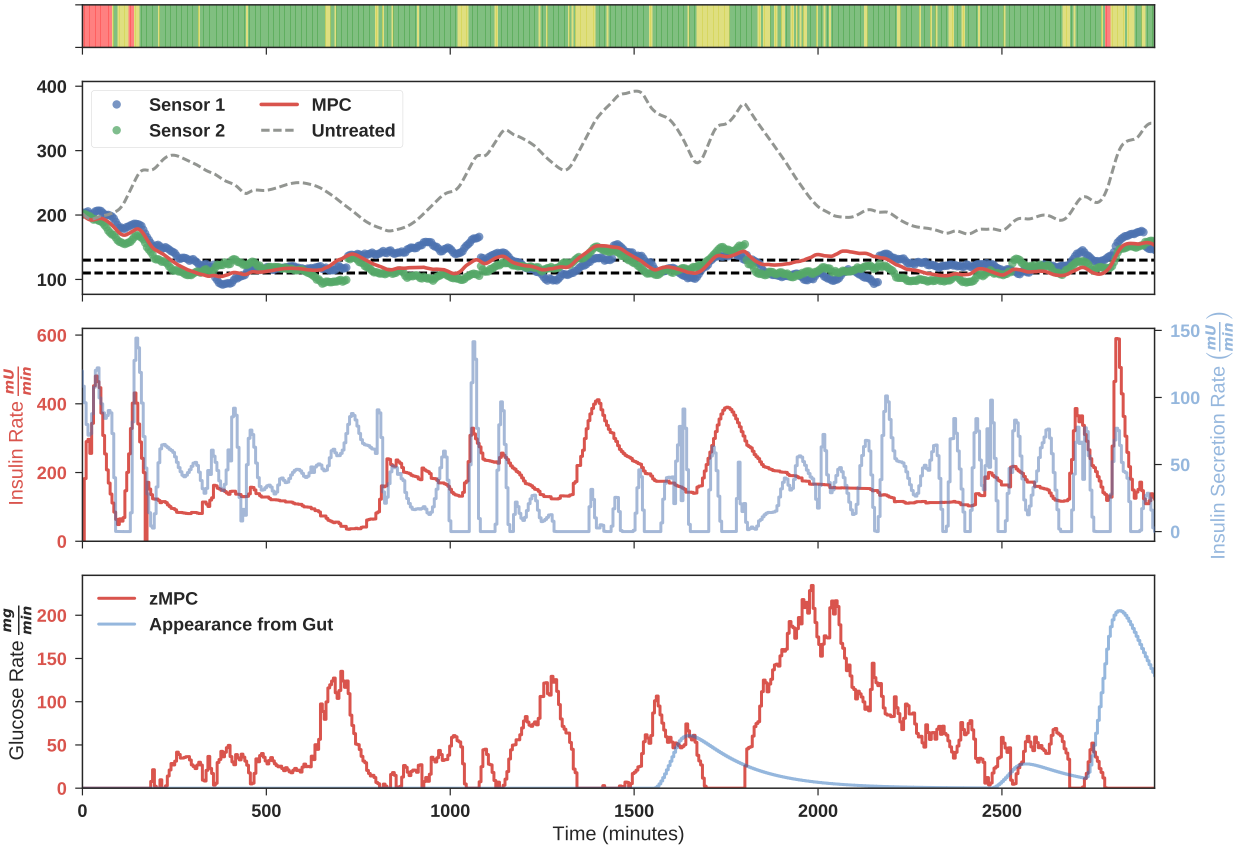


Figure 52: The control outcomes with simulated normal CGM noise for the virtual patient used in previous control examples is shown. The bottom 3 panels remain unchanged from before, however the top panel is shaded in accordance with the color scheme defined in Section 6.2.3.

Although the alarm metrics used here to detect excessive CGM error do not completely eliminate incidences of moderate hypoglycemia ($BG \leq 74\frac{mg}{dL}$), it is important to remember that these simulated trials are truly a worst case scenario. In all cases of (re-)calibration, the CGM error model is reinitialized with the parameter set characterizing faulty CGMs. In clinical practice we would expect such a scenario to be exceedingly unlikely.

The infusion set failure detection mechanism accurately detects a large majority of infusion set failures with a low false positive rate. Although approximately 18% of infusion set failures appear to remain undetected, it is important to note that an infusion set failure may only be detected if insulin is actually being infused. In many of the "missed" cases, insulin is not constantly being infused and, due to subcutaneous insulin on board, glucose control does not immediately or significantly suffer.

The average time to detection of an infusion set failure is 5.8 hours. This may seem excessive, but the fact that subcutaneous insulin is being used here must be considered. As discussed previously (Section 5.4), there is a significant difference in dynamics between subcutaneous and intravenously delivered insulin, with subcutaneous insulin having a prolonged tail on effective action. As such, it takes a good deal longer to detect a failure in subcutaneous insulin delivery compared to intravenous infusion, leading to seemingly lengthy detection times compared to other methods [307].

With regards to the "traffic light" information system, the example provided in Figure 52 is representative of simulations with normal CGM noise. As seen in Figure 52, yellow regions tend to precede excursions from the target zone and red regions are associated with significant excursions from the target zone. As such, this simple system may provide a means for clinicians to quickly identify when further non-automated intervention may be required to ward off significant hypo- or hyperglycemia. Furthermore, such an interface may work to enhance clinical confidence in the closed-loop model-based control loop system developed here by providing output or feedback and making the system appear like less of a "black box".

6.4.1 Pressure Induced Loss of Sensitivity

Not yet considered here is a CGM fault via a pressure-induced loss of sensitivity. CGMs are known to return a significantly decreased measurement of interstitial blood glucose in response to a pressure challenge [81, 143, 144]. This pressure-induced loss of sensitivity (PILS) is hypothesized to be due to a temporary reduction in local blood flow resulting in a slowing of the glucose-diffusion process in the region of the sensing element [143, 144, 319]. In humans, anomalous sensor readings are more commonly reported at night and are believed to be due to compression of the sensor particularly when patients lie directly on the sensor [320]. This is especially concerning in the largely bedridden critical care patient population where the possibility for a PILS due to a patient lying on a sensor is much greater.

The data available for one virtual patient appears to exhibit a plausible, but unconfirmed pressure induced loss of sensitivity. Otherwise, the remainder of the data does not appear to have any segments which believably resemble a pressure induced loss of sensitivity in a single CGM. As such, and because the algorithm discussed above has been tested and validated in the work of Baysal *et al.* it will not be tested further for the detection of pressure induced losses of sensitivity . An overview of this algorithm has been included in Appendix G to provide a complete picture of the alarm systems envisioned for this model-based glucose control system.

6.5 SUMMARY

A method to identify failing continuous glucose monitors and recommend recalibration based on two metrics was introduced. Using this system in a simulated worst-case clinical scenario, hypoglycemia was significantly reduced and patient safety enhanced. Using the moving horizon estimator integrated into the closed-loop model-based control algorithm augmented with an insulin infusion disturbance variable allowed infusion set failures to be reliably detected within an average of 6 hours after the onset of failure. Furthermore, the infusion set failure detection mechanism has a low false positive rate, thereby reducing the potential for alarm fatigue and ignored faults. Finally, a system to output simple, relevant information about the projected ability of the control system to manage hypo- and hyperglycemia was introduced. This system provides an interface for clinicians to remain involved in glucose management and intervene when deemed necessary.

7.0 SUMMARY AND FUTURE RESEARCH

7.1 CONTRIBUTIONS

Mitigating hypeglycemia and reducing glucose variability in critically ill patients presenting with stress hyperglycemia may be able to improve patient outcomes such as mortality, length of hospital stay, infections and other complications. Previous attempts to treat stress hyperglycemia through tight glucose control and intensive insulin therapy have been met with varying degrees of success, largely due to the increased incidence and risk of hypoglycemia inherent in IIT. The work presented in this thesis details the development of a model-based control system to automate the delivery of glucose insulin for the control of blood glucose concentrations within a target zone.

To this end, the feedback component of the closed-loop control scheme in the form of continuous glucose monitors are characterized to develop models of CGM error processes and to reconstruct accurate patient blood glucose trajectories from dual CGM measurements. The reconstructed glucose trajectories are used to develop a novel physiologically realistic and clinically relevant virtual patient cohort which recapitulates anticipated physiologic trends, exhibits expected inter- and intrapatient variability with individual virtual patients who respond in a biologically consistent manner to insulin and glucose challenges. To affect zone glucose control a moving horizon estimator and model-predictive control are implemented in python and tuned using the virtual patient cohort so as to safely and effectively control blood glucose concentrations even in the presence of CGM noise. Finally, a system to detect CGM faults and insulin infusion set failures is developed and tested via *in silico* trials on the virtual patient cohort and a clinical information system utilizing a simple color-coded scheme to display the expected capacity of the controller to maintain glucose control is provided.

7.1.1 Continuous Glucose Monitors: Blood Glucose Reconstruction and Error Models

In an observational study conducted at the University of Pittsburgh Medical Center 24 critically ill patients each had two Dexcom® Platinum™ G4 continuous glucose monitors inserted on opposing sides of the abdominal wall. These CGMs measured glucose concentrations every five minutes for no less than 24 hours. Additional reference blood glucose measurements were collected using capillary fingerstick point-of-care glucometers and a blood gas analyzer. Analysis of the data indicated significant error between the CGMs and the reference measurements. To correct the error a methodology was developed to remove presumed bias or offset and CGM drift (Section 2.2.1.3). The result is a single blood glucose profile which combines high-density measurements from independent CGMs and high-accuracy reference measurements to produce a more accurate high-frequency vector of glucose concentrations.

Although there have been previous attempts to reconcile CGM measurements with high accuracy reference measurements and model glucose sensor error [153], these techniques had the advantage of high frequency reference measurement with a capillary fingerstick available every 15 minute. This enabled a convolutional approach and explicit modeling of glucose transport dynamics. Here, reference measurements are much more sparse necessitating the approach taken in this work where a heuristic regularization is used to generate solutions which trend toward an ideal situation with perfect agreement between CGMs and reference measurements.

Utilizing the high-frequency reconstructed glucose trajectories as surrogate references a noise model for properly functioning CGMs and an error model for malfunctioning CGMS are developed. Nominally functioning and malfunctioning intervals of CGM data between sensor recalibration are classified based on a MARD threshold of 5%. It was found that a first order integrated moving average model fit both the data from nominally and malfunctioning CGMs equally well with a nearly identical moving average parameter ($\theta = 0.402$, and $\theta 0.394$, respectively) but vastly different variances in the white noise driving these stochastic process. This suggests that the primary component of CGM error may be erroneous calibration measurements or calibration during times of rapid glucose fluctuation. The error models developed in Chapter 2 are used in Chapter 5 and Chapter 6 to test the efficacy and safety of the control algorithm under realistic clinical conditions.

7.1.2 Virtual Patients

The use of *in silico* trials offers the ability to design and test treatment paradigms, protocols and algorithms to optimize performance and safety without risk to real patients [122]. Useful and credible *in silico* trials require accurate computer models of not only treatment and the clinical realization of a treatment but also simulated or "virtual" patients [321–324]. In this work, a virtual patient cohort is developed using reconstructed blood glucose data from CGM measurements and medical record data of exogenous glucose and insulin administration, enteral nutrition and pertinent medications.

The virtual patient cohort constructed here is rooted in the Intensive Care Insulin Nutrition Glucose (ICING) model developed by Lin *et al.* [110]. This model was developed specifically to describe the glucose-insulin dynamics of critically-ill patients and has been validated for both it's explanatory and predictive power in an intensive care population. The model, as originally formulated accounts for only intravenously insulin administration, yet the data used here to construct the virtual patient cohort indicates that a vast majority of the patients received subcutaneous insulin injections and infusions. Furthermore, the controller developed here is designed to administer subcutaneous insulin to avoid complications associated with intravenous delivery and conform to the U.S. clinical standard of care. As such, the ICING model was extended with a model developed by the Parker group which characterizes the absorption kinetics of subcutaneously delivered fast-acting and regular-acting insulin [114, 160]. This model was found to be insufficient to capture the glucose variability observed in the clinical data and as such was extended with a functionality to describe the insulin-induced suppression of endogenous glucose production in the liver [130] resulting in the extended ICING (eICING) model.

To calibrate the eICING model to patient data, time-varying insulin sensitivity and rate of endogenous glucose production (pancreatic insulin release) was regressed. Additional patient specific parameters included the the Michaelis-Menten half-maximum insulin concentration for the suppression of hepatic EGP and the rates of insulin transport from the plasma to interstitium and subsequent insulin degradation. The latter rate parameters are constrained by a physiologically-based constraint on a steady state ratio of insulin concentrations in the plasma and interstitium [195–197].

Through a unique use of heuristic Tikhonov regularization terms on insulin sensitivity and production changes and physiologically motivated constraints a novel virtual patient cohort was developed. The virtual patients exhibited oscillations in pancreatic insulin secretion consistent with observed ultradian rhythms of insulin secretion [183, 185, 198] and insulin sensitivity trends which are conserved across the virtual patient cohort and congruous with physiologic expectation and clinical assumptions. Namely, a negative correlation between mean regressed insulin sensitivity values and blood glucose concentrations and a positive correlation between mean regressed insulin sensitivity variability and blood glucose concentrations – especially once normalized for exogenous insulin.

Although the virtual patient cohort as constructed from the eICING+SQ model exhibits physiologic realism and the proper clinical tendencies, the lack of pancreatic response to changing glucose concentrations cannot be ignored. In simulated trials of various treatment methodologies it is extremely unlikely that a blood glucose trajectory identical to the one observed in the clinical data would arise. As such, the pancreas would be expected to respond to rising and falling blood glucose concentrations by increasing or decreasing insulin secretion rates and the static regressed endogenous insulin production profile would no longer be valid. The solution employed here is to modulate a basal rate of insulin secretion in response to plasma glucose concentrations using a PID controller. The proportional, integral and derivative gains are fit to a glucose error signal with a set point of $110\frac{mg}{dL}$ (the upper limit of normoglycemia [78, 232]) via a linear least squares regression. The pancreatic response using PID control is tested via simulated oral glucose tolerance tests and insulin tolerance tests. The virtual patients equipped with the responsive PID pancreas show a much more realistic response to the tolerance tests with times to peak glucose, recovery indices, and total recovery times which are consistent with literature studies.

7.1.3 Control

To maintain blood glucose concentrations within the target zone $\left(110 - 130\frac{mg}{dL}\right)$ a controller comprised of a zone model predictive controller and a moving horizon estimator is developed and tested. The moving horizon estimator is implemented as a python module (CoPyMHE) and allows for state, parameter and disturbance estimation of linear time invariant (LTI) systems in a standard state space format. The update of the MHE arrival cost is the same as the Kalman Filter covariance update cost and is exact if the estimation is unconstrained. For constrained estimates the Kalman Filter arrival cost update is an approximation. The model-predictive controller is also implemented as a python module (PyMPC) and enables multiple input multiple output zone control of linear systems. Like the CoPyMHE module, PyMPC is generic and can be used for any LTI system in a standard state space format. PyMPC outputs can be constrained and PyMPC includes built in functionality to specify any manipulated variable for output regulation, that is treatment of a manipulated variable as a controlled variable with a target zone or setpoint.

For *in silico* trials using the CoPyMHE and PyMPC modules the ICING model was supplemented with the subcutaneous insulin model and linearized around a steady state operating point. The linearized model in state space form was discretized with a 5 minute sampling period to correspond to the sampling rate of the Dexcom® Platinum™ G4 and provided to the MHE and zMPC. For the MHE the output matrix C was modified to generate two output measurements of blood glucose such that the state space mapped to the clinical reality where two CGMs are used to measure glucose levels. In this way, the

MHE fuses CGM measurements via a maximum likelihood estimate of insulin sensitivity, insulin production, glucose production and interstitial insulin over a receding horizon of CGM measurements. Using estimates of model state the MPC optimizes glucose and insulin infusion rates which are administered to the virtual patient.

Using the MHE to fuse CGM measurements with simulated noise the MARD of the blood glucose concentration as estimated by the MHE is calculated against the simulated virtual patient glucose concentration. Although analysis of the data from individual CGMs shows a MARD greater than 15%, larger than the suggested maximum for glucose control in critical care [78, 80] the MHE estimates exhibit a significantly smaller MARD. With a simulated recalibration frequency of 6 hours using 2 CGMs MHE estimates result in MARD $\leq 11\%$, acceptable for glucose control. If only a single CGM is used with the MHE the necessary recalibration frequency increases to once every 4 hours requiring a significant increase in clinical effort.

In *in silico* trials using both "perfect" (i.e no error) CGMs and CGMs with simulated normal levels of noise the controller was able to significantly reduce mean glucose in the virtual patient cohort, compared to the clinical data as well as glycemic variability. The effects of announcing meals to the MPC were studied via simulation and the decision was made that the danger of missing announced meals outweighed the benefits of announced meals (mainly avoidance of post-meal glucose speaks). Under these simulated conditions, hypoglycemia was avoided entirely and a significant portion (86.59%) of time was spent in the target zone for the virtual patients under closed-loop control (compared to 37.39%) for the clinical treatment. Additionally, the controller developed here is compared in *in silico* trials to two commercially available, FDA approved systems for glucose management in the ICU; the GlucoStabilizer [294] and the Computerized Yale Protocol [275]. For these trials the GlucoStabilizer and Computerized Yale Protocol algorithms were re-implemented and used to treat the virtual patient cohort. Although these commercially available solutions were successful in mitigating hyperglycemia, a low level of moderate hypoglycemia resulted (although severe hyperglycemia was avoided). Furthermore, these systems were not effective in reducing glucose variability and both systems actually increased variability compared to an untreated patient. Finally, these systems require frequent measures of blood glucose concentrations, in some cases at 15 minute intervals which is a significant clinical burden. As such, these systems are not as safe or effective as the closed-loop controller developed here.

7.1.4 Safety

Through the use of dual CGMs with moving horizon estimation MARD is reduced to acceptable levels. However, there is still a possibility of improper recalibration and/or CGM failure. To detect CGM faults or failures a method which tracks instantaneous percent error between the CGMs as well as integrated error is introduced. With an instantaneous threshold of 30% and an accumulated error threshold of $5000 \frac{mg \cdot min}{dL}$, in *in silico* trials with simulated faulty CGM error (including after every recalibration) the use of these error metrics reduces the incidence of hypoglycemia from 5.5% to 1.93%. Although hypoglycemic events are still present in this scenario it is important to note that this is a worst case scenario simulating improper CGM recalibration at every turn. In practice, this seems incredibly unlikely.

In addition to CGM faults or failure, the other likely point of failure in the control system developed here is in the insulin infusion set. Insulin infusion sets are used to infuse subcutaneous insulin below the skin and may fail due to a foreign body reaction and swelling at the site or skin contortions which allow infused insulin to leak out of the body [307]. The resultant degradation in glucose control quality is made more severe by the use of an MHE which will likely estimate a significant decrease in insulin sensitivity and/or pancreatic secretion to account for the loss of insulin through infusion set failure thereby rendering the internal models inaccurate.

To detect infusion set failures the disturbance estimation feature of the MHE is used. An input, $U_{SC,d}$ is added which is constrained in the MHE to be negative such that the effective insulin infusion rate is the sum of U_{SC} and $U_{SC,d}$. By setting the term in the MHE weighting matrix Q which penalizes deviations $U_{SC,d}$ from zero sufficiently high, $U_{SC,d}$ will be estimated to be significantly non-zero only in the case of infusion set failures. An infusion set failure is ultimately detected by the ratio of $U_{SC,d}$ to U_{SC}. When the negative disturbance reaches a substantial fraction of the actual positive subcutaneous insulin infusion rate a failure is detected. In *in silico* trials this technique is sensitive, detecting $\sim$ 82% of infusion set failures with a 2% false positive rate.

As a final component of the safety system a clinical information system is designed and presented here. In essence, this system uses a color-coded system to indicate to clinicians the predicted ability of the MPC to adequately manage blood glucose concentrations within the target zone. In doing so it provides a mean to keep clinicians involved in the automated treatment process and warn of potential ineffective control so that clinicians may intervene if they feel it is necessary.

7.2 FUTURE DIRECTIONS

The control system developed here for zone glucose control in critically ill non-diabetic patients has been tested extensively through *in silico* trials in a virtual patient cohort and shown to be effective and safe. Despite the benefits of *in silico* trials using mathematical models to simulate patient behavior, it must be conceded that all models are wrong yet some are useful [325]. As such, at some point clinical trials are necessary. The likely next step in testing the control algorithm would be in open-loop operation where the MHE collects data from CGMs as normal and the MPC calculates optimal control action, however infusion rates are changed only subject to clinical approval. This keeps a doctor in the loop to ensure patient safety and enables immediate intervention or controller override in the event of an emergency or unexpected controller failure. Furthermore, it may be beneficial during such a trial to explore the effects of blood glucose dynamics of CGM recalibration. A significant component of CGM error observed here is postulated to be due to improper recalibration or recalibration during periods of rapid blood glucose change. Deliberate recalibration during periods of both steady blood glucose concentrations and rapid change may help provide evidence of this hypothesis and through analysis of glucose variability and CGM MARD may help set proper recalibration guidelines for CGM in critical care.

The virtual patient cohort developed here is believed to be physiologically realistic and responds appropriately to intervention with insulin and glucose. However, the PID pancreatic response used here lacks a meaningful relation to physiology and only accounts for the basal rate of insulin secretion, a fraction of the total pancreatic insulin secretion. The remaining fraction of insulin secretion is patient-specific yet independent of plasma glucose concentrations. The addition of a functionality which fully describes pancreatic insulin secretion would be another step towards more realistic virtual patients and would further enhance confidence in *in silico* trials and simulated control results. Furthermore, a fully functional description of pancreatic insulin secretion could be incorporated into the controller model which would result in more accurate estimations and predictions ultimately leading to better controller performance

A multitude of mathematical models to describe pancreatic insulin secretion ranging from cellular and subcellular models [326–330] to physiologically-based models [222, 331–335]. However these models are developed for healthy patients or to explain the beta cell component of metabolic dysfunction in diabetic patients. In a critically-ill patient population due to the complex interplay between inflammation, stress and cytokine dynamics the parameterization of these models may differ significantly from both a healthy and diabetic patient population. Unfortunately, existing models contain a large number of parameters, and therefore require a significant data density in order to proper calibrate.

Here, no insulin or C-peptide measurements were available leading to the use of a Tikhonov regularization term and a number of physiologically motivated constraints in developing the virtual patient cohort. If open-loop trials were conducted or additional observation studies carried out it would be beneficial to collect C-peptide data to help further identify the contribution of pancreatic insulin secretion to blood glucose dynamics. Even if only sparse C-peptide measurements were available, perhaps model reduction techniques such as balanced truncation and parametric clustering based on sensitivity analysis, as previously used in the Parker group [336] could be employed to ensure the identifiability of pancreatic insulin secretion parameters.

7.2.1 Counterregulatory Response

The endogenous response to hypoglycemia consists of the release of numerous counterregulatory hormones including epinephrine, norepinephrine, glucagon, cortisol, and growth hormone [170, 337, 338]. The collective action of the counterregulatory response reduces the rate of glucose absorption into peripheral tissues by lowering insulin sensitivity and triggers the release of endogenous glucose supplies from the liver into the bloodstream to counteract the causative hypoglycemia [170]. Although the use of a closed-loop model-based controller results in zone glucose control while avoiding hypoglycemia in *in silico* trials, counterregulatory hormone concentrations may be increased from exogenous sources.

Epinephrine may be used to treat asthma and abnormal heart rhythm [339] and norepinephrine is commonly used in ICU settings for the treatment of critically low blood pressure [340]. Hydrocortisone is commonly used to treat a litany of conditions including rheumatoid arthritis, asthma and COPD [341]. Due to the prevalence of exogenous administration of these hormones and their significant action in the regulation of blood glucose it may be important to include a mechanistic description of their effects both in the virtual patient cohort and within the controller. Otherwise, a number of conditions treated with these hormones may preclude the use of the glucose control system developed here and a number of ICU medications may be contraindicated for use with automated treatment.

A colleague in the Parker lab has developed a number of modular mathematical models describing various components of the counterregulatory response [170] which may be used for this purpose. However, there is a relative lack of individualized data concerning the dynamics of the counterregulatory response and patient specific responses, particularly where insulin sensitivity is concerned, may be difficult to quantify. Ultimately, inclusion of explanatory mechanisms for the counterregulatory response would broaden the applicability of the glucose control system developed here and potentially improve patient outcomes in a larger population with a significant savings in terms of clinical effort and health care costs.

APPENDIX A

EXTENDED ICING MODEL

The full extended ICING model with subcutaneous insulin adsorption is given in eqs. (A.1) to (A.13). Here eqs. (A.1) to (A.7) are taken from the ICING model [110] and describe the saturating effect of effective interstitial insulin concentrations (Q_I) on blood glucose (BG) disposal with the magnitude of effect modulated by insulin sensitivity (S_I). Additionally, glucose is disposed of through a non-insulin dependent pathway at a proportional to glucose concentration with a rate constant of p_G and used at a constant rate by the central nervous system (CNS). Glucose is replenished via enteral nutrition ($P(t)$), exogenous infusion (U_G) and endogenous glucose production (EGP). Insulin appears in the interstitium from the plasma at a rate of n_I and is degraded at a saturating rate of n_C. In the plasma insulin degrades with a saturating rate of n_L and at a non-saturating rate of n_k. Insulin is replenished in the plasma via pancreatic secretion (U_{en}) and exogenous input (u_{ex}). Equations (A.4) to (A.6) describe the transport of oral nutrition from the stomach to the gut and subsequent adsorption into the bloodstream. Equations (A.8) to (A.10) describe the linear adsorption kinetics of both regular- and fast-acting insulin analogs [114]. In this three compartment model subcutaneously administered insulin is degraded in the second compartment at a rate specific to regular- or fast-acting analogs. Equations (A.11) to (A.12) descibe the transport of insulin to liver tissue and the subsequent saturating insulin mediated endogenous glucose production [130]. Finally, eq. (A.13) governs the transport of glucose from plasma to the interstitial fluid (ISF) which is the glucose concentration measured by CGMs in *in silico* simulations.

A.1 MODEL EQUATIONS

$$\frac{dBG(t)}{dt} = -p_G BG(t) - S_I(t)\frac{Q(t)}{1+\alpha_G Q(t)} + \frac{P(t)+EGP(t)+U_G(t)-CNS}{V_G} \tag{A.1}$$

$$\frac{dQ(t)}{dt} = n_I(I(t)-Q(t)) - n_c\frac{Q(t)}{1+\alpha_G Q(t)} \tag{A.2}$$

$$\frac{dI(t)}{dt} = -n_k I(t) - \frac{n_L I(t)}{1+\alpha_I I(t)} - n_I\left(I(t)-Q(t)\right) + \frac{u_{ex}+(1-x_L)U_{en}(t)}{V_I} \tag{A.3}$$

$$\frac{dP1}{dt} = -d_1 P1 + D(t) \tag{A.4}$$

$$\frac{dP2}{dt} = -min(d_2 P2, P_{max}) + d_1 P1 \tag{A.5}$$

$$P(t) = min(d_2 P2, P_{max}) + PN(t) \tag{A.6}$$

$$u_{ex} = U_{IV} + k_{a_1} Q_{2,r} + k_{a_1} Q_{2,f} \tag{A.7}$$

$$\frac{dQ_{sc,(r,f)}}{dt} = -k_{1,sc} Q_{sc,(r,f)} + U_{sc,(r,f)} \tag{A.8}$$

$$\frac{dQ_{1,(r,f)}}{dt} = k_{1,sc} Q_{sc,(r,f)} - k_{v,(r,f)} Q_{1,(r,f)} - k_{a_1} Q_{1,(r,f)} \tag{A.9}$$

$$\frac{dQ_{2,(r,f)}}{dt} = k_{a_1} Q_{1,(r,f)} - k_{a_1} Q_{2,(r,f)} \tag{A.10}$$

$$\frac{dQ_L(t)}{dt} = PG_4(I(t)-Q_L(t)) \tag{A.11}$$

$$EGP(t) = EGP_0\left(1 - k_{EGP}\left[\frac{Q_L^3(t)}{Q_L^3(t)+s_{EGP}^3}\right]\right) \tag{A.12}$$

$$\frac{dG_{ISF}(t)}{dt} = \frac{1}{\tau}(BG(t) - G_{ISF}(t)) \tag{A.13}$$

A.2 MODEL PARAMETERS

Table 17: Parameter values and units for the extended ICING model with subcutaneous insulin adsorption kinetics. $*$: regressed patient-specific parameter, $**$: regressed, dynamic (time-varying) patient-specific parameter vector

Paramter	Value	Unit
p_G	$*$	$\frac{mg}{dL\cdot min}$
S_I	$*, **$	$\frac{L}{mU\cdot min}$
α_G	0.0154	$\frac{L}{mU}$
V_G	133	dL
CNS	$*$	$\frac{mg}{min}$
n_I	$*$	min^{-1}
n_C	$*$	min^{-1}
α_I	00017	$\frac{L}{mU}$
n_k	0.0542	min^{-1}
n_L	0.1578	min^{-1}
x_L	0.67	$-$
v_I	3.5	L
U_{en}	$*, **$	$\frac{mU}{min}$
d_1	0.047	min^{-1}
d_2	0.0069	min^{-1}
P_{max}	1100.75	$\frac{mg}{min}$
$k_{1,SC}$	1.08	min^{-1}
$k_{v,r}$	0.0268	min^{-1}
$k_{v,f}$	0.00683	min^{-1}
k_{a_1}	0.0127	min^{-1}
PG_4	0.07353	min^{-1}
EGP_0	208.98	$\frac{mg}{min}$
k_{EGP}	0.9909	$-$
s_{EGP}	$*$	$\frac{mU}{L}$

APPENDIX B

VIRTUAL PATIENTS

For all virtual patients:

- **Top Panels:** eICING model glucose concentration (-) fit to clinical CGM data (● and ●)
- **Middle Panels:** Estimated pancreatic insulin secretion rate ($U_{en}(t)$)
- **Bottom Panels:** Estimated Insulin sensitivity ($S_I(t)$)

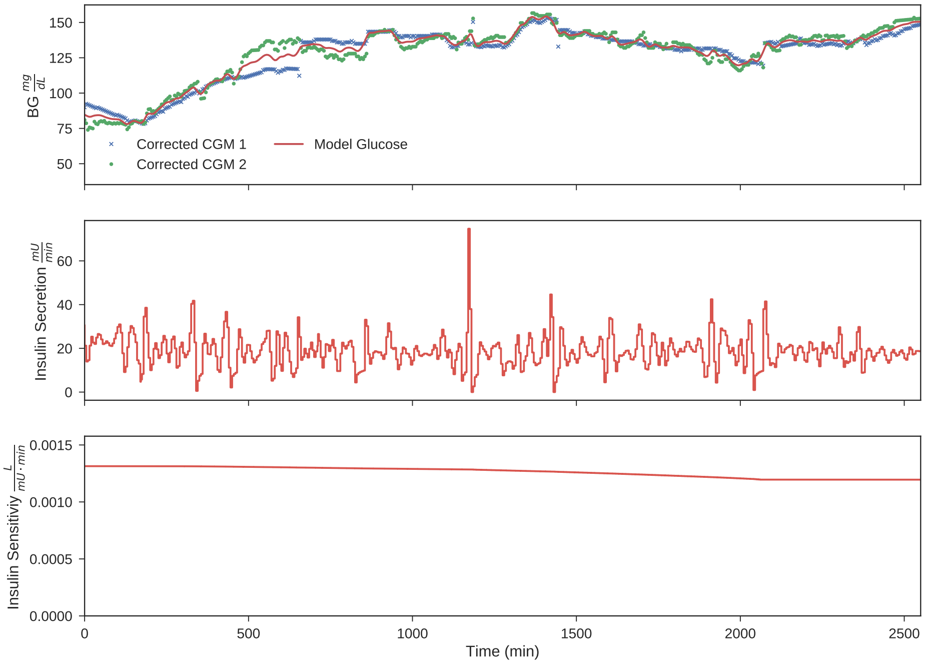


Figure 53: Virtual Patient 1

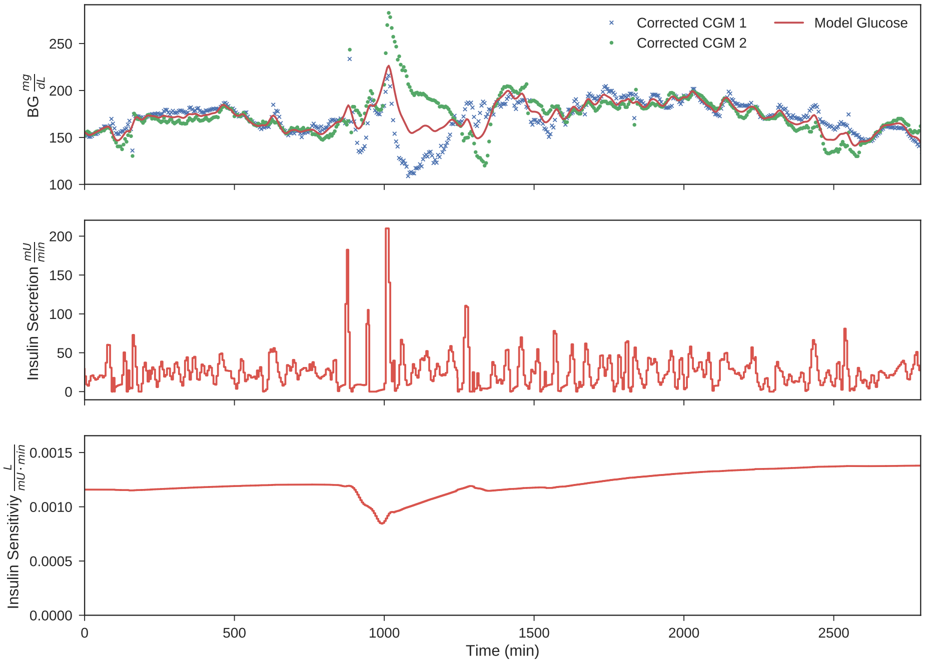


Figure 54: Virtual Patient 2

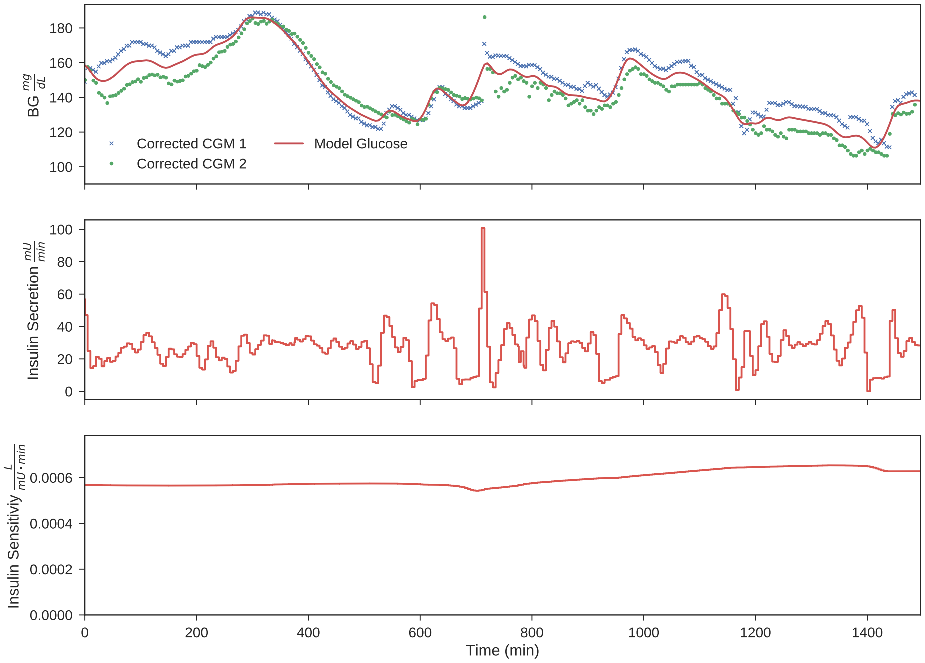


Figure 55: Virtual Patient 3

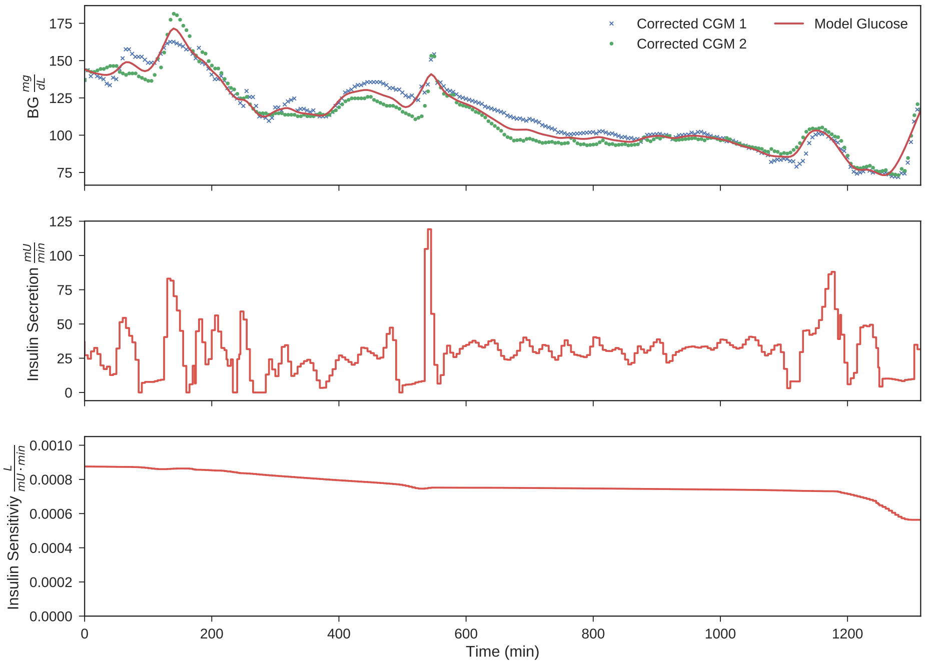


Figure 56: Virtual Patient 4

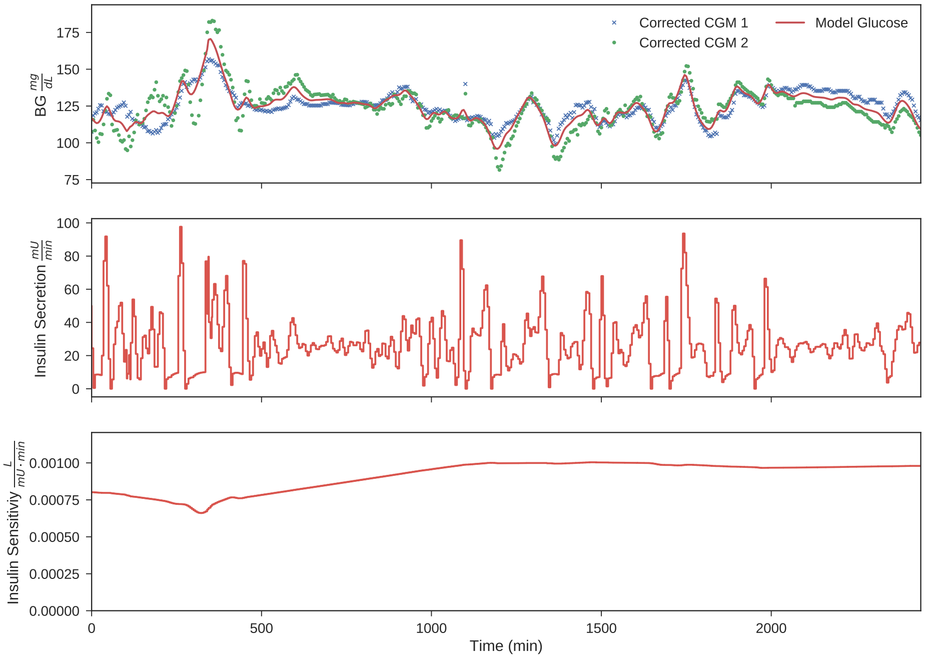


Figure 57: Virtual Patient 5

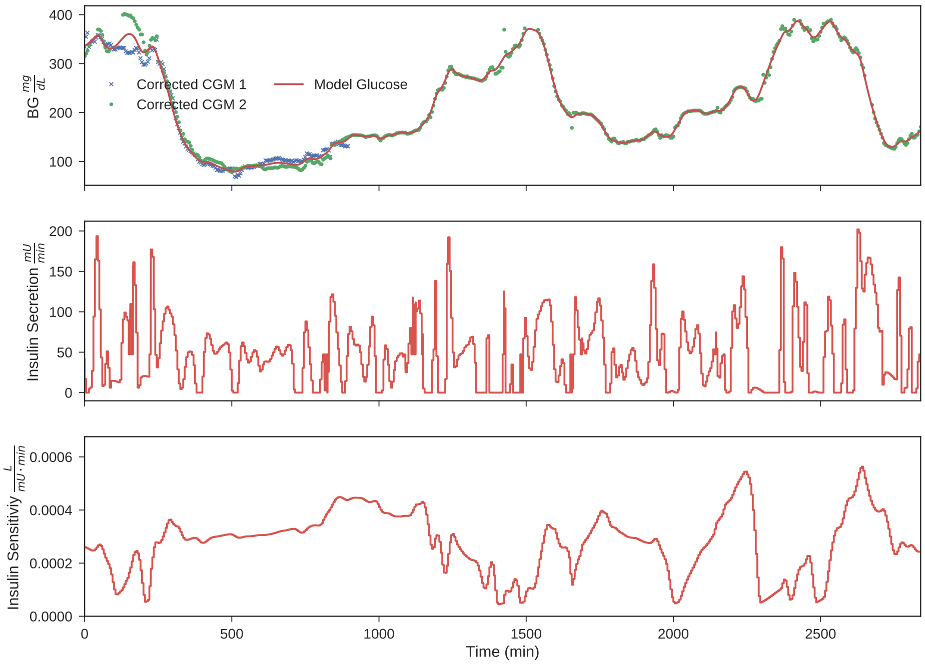


Figure 58: Virtual Patient 6

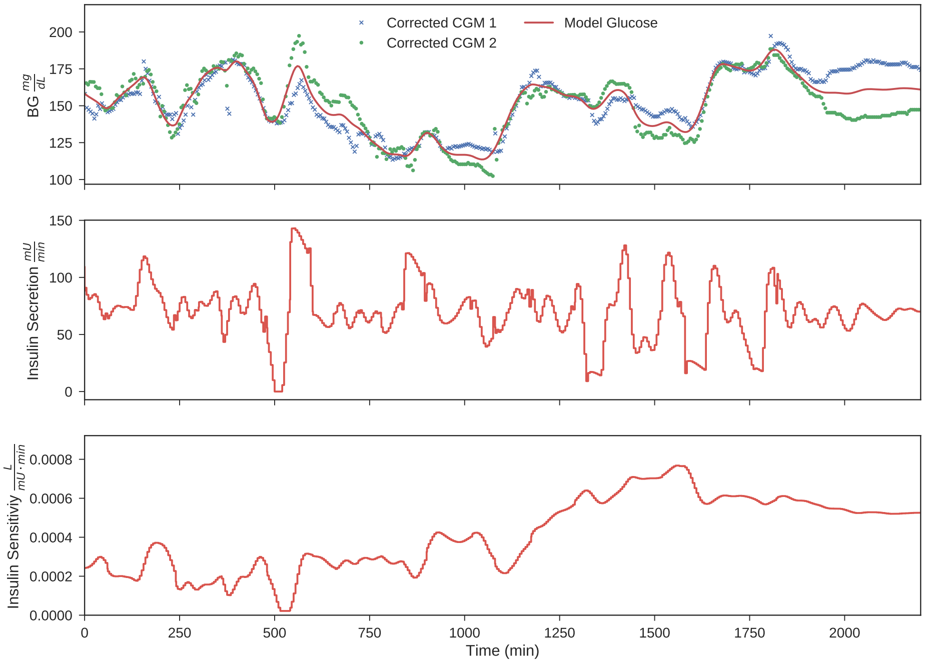


Figure 59: Virtual Patient 7

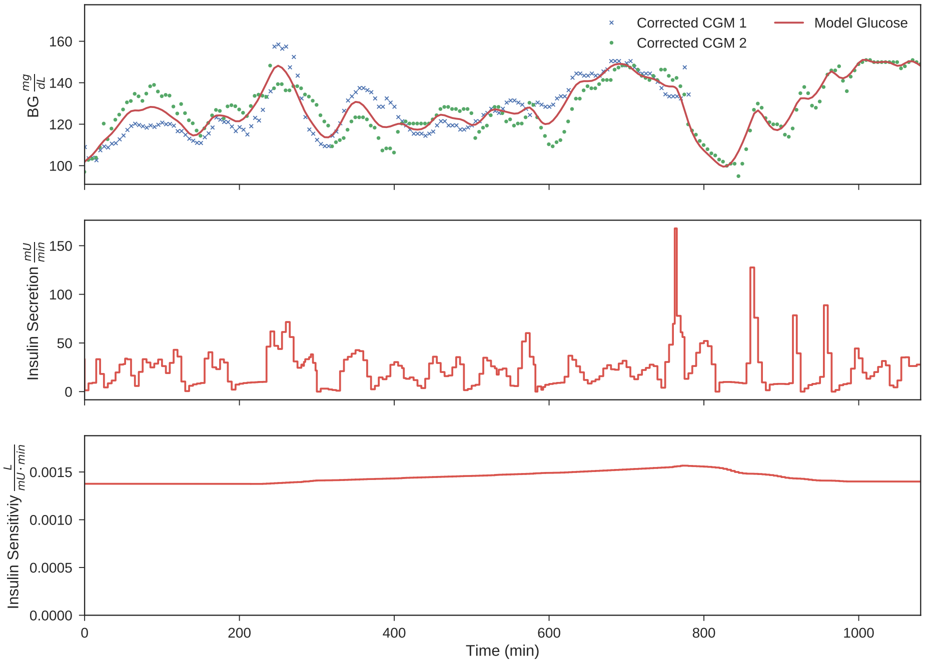


Figure 60: Virtual Patient 8

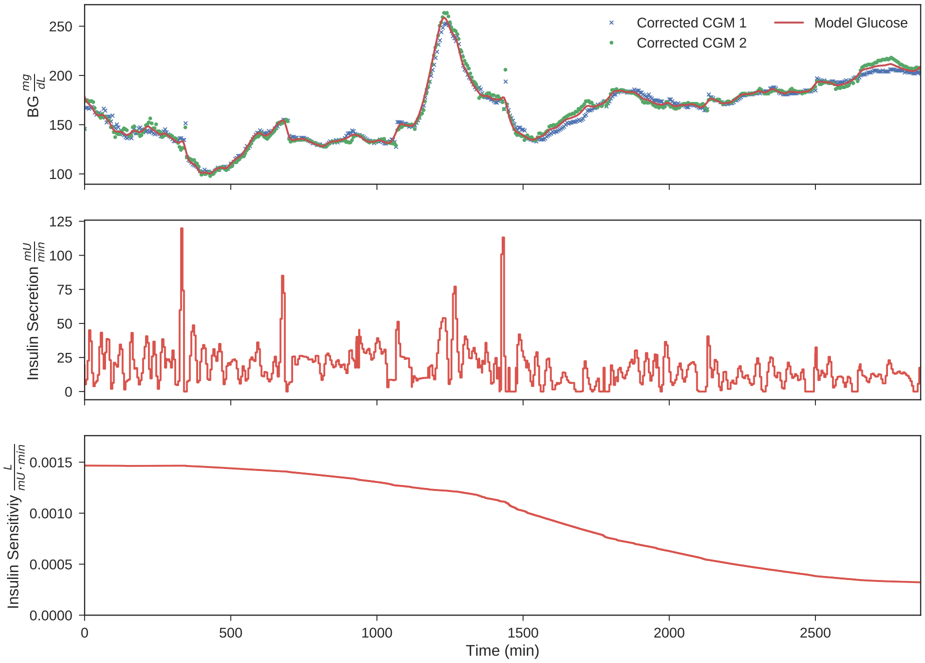


Figure 61: Virtual Patient 9

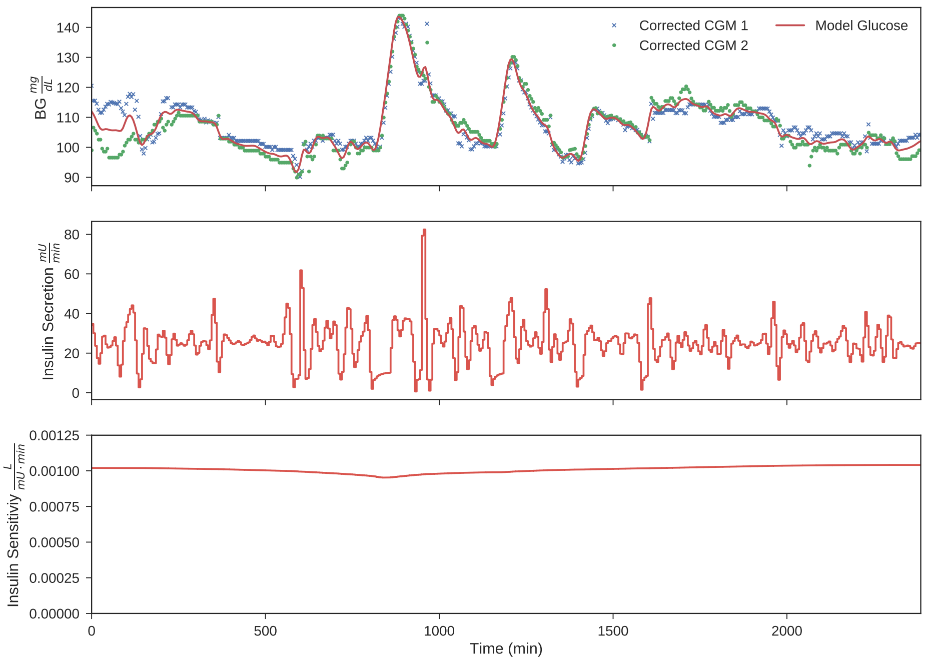


Figure 62: Virtual Patient 10

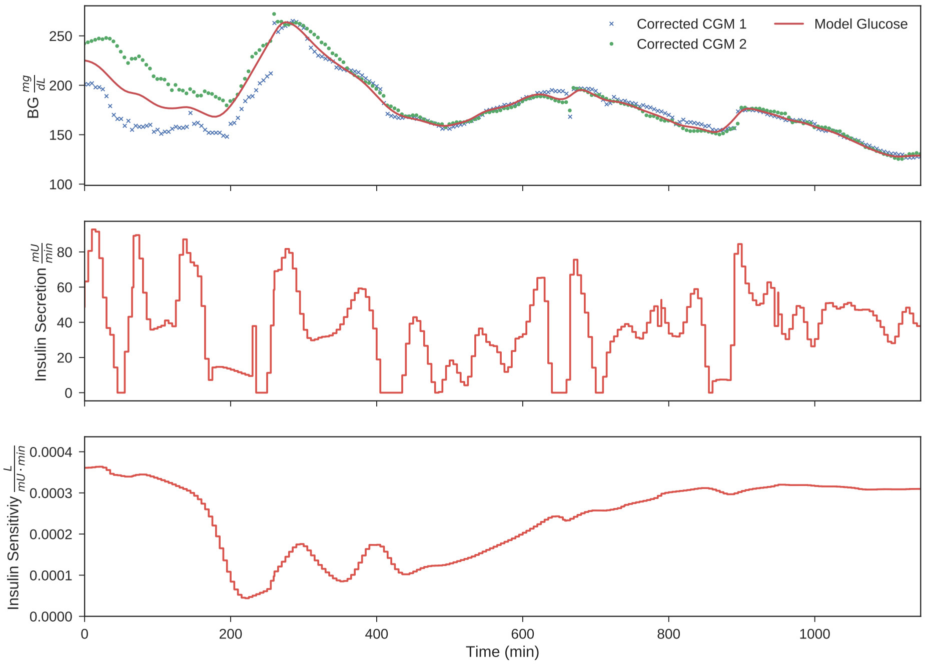


Figure 63: Virtual Patient 11

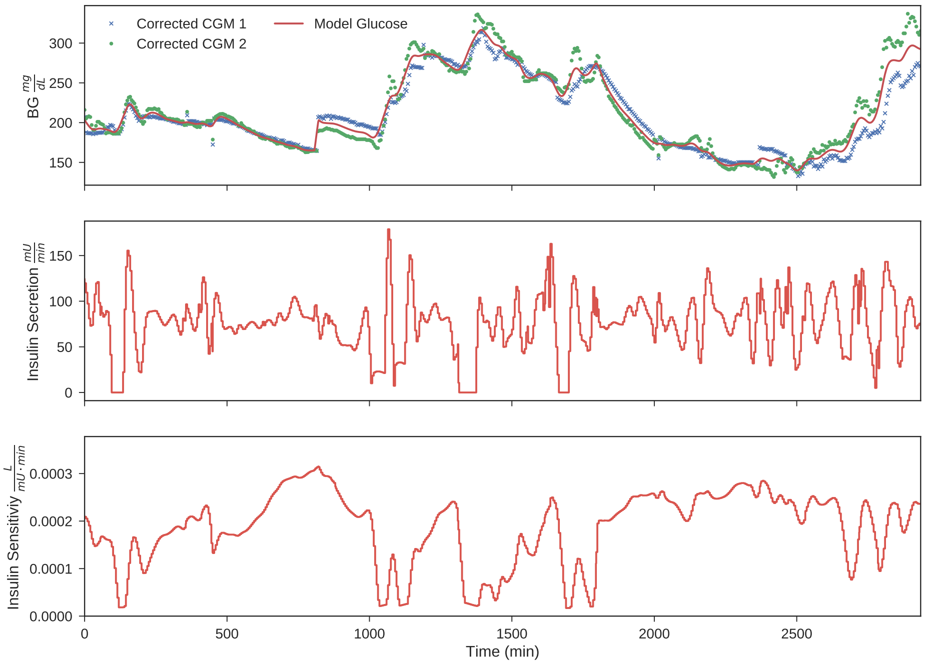


Figure 64: Virtual Patient 12

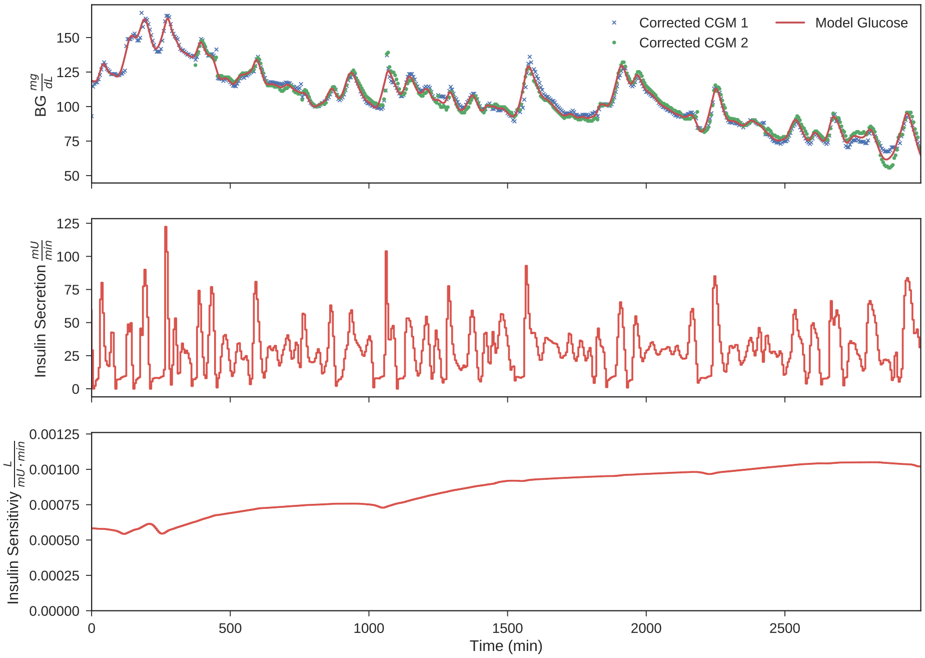


Figure 65: Virtual Patient 13

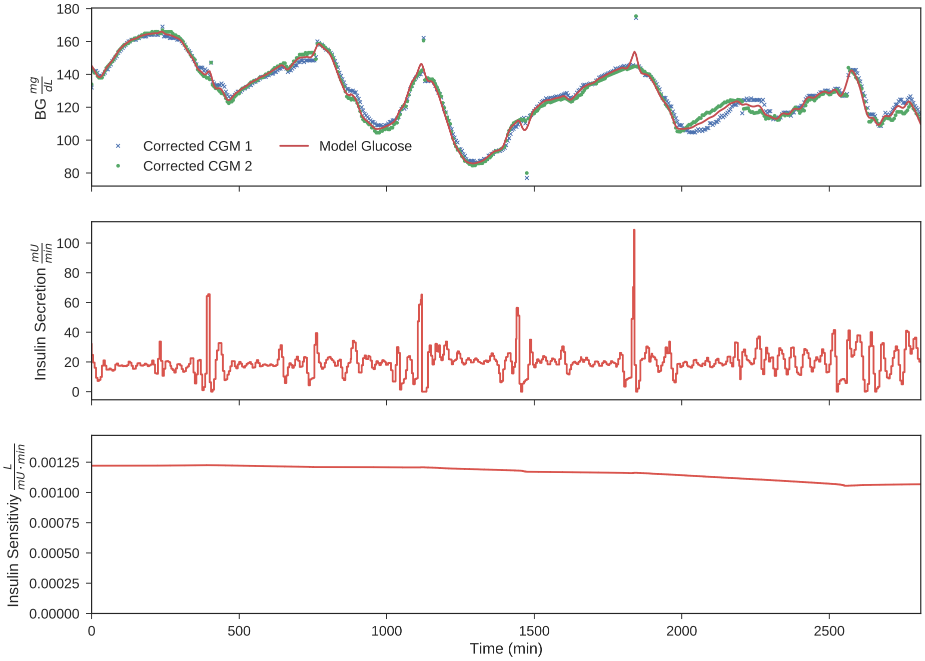


Figure 66: Virtual Patient 14

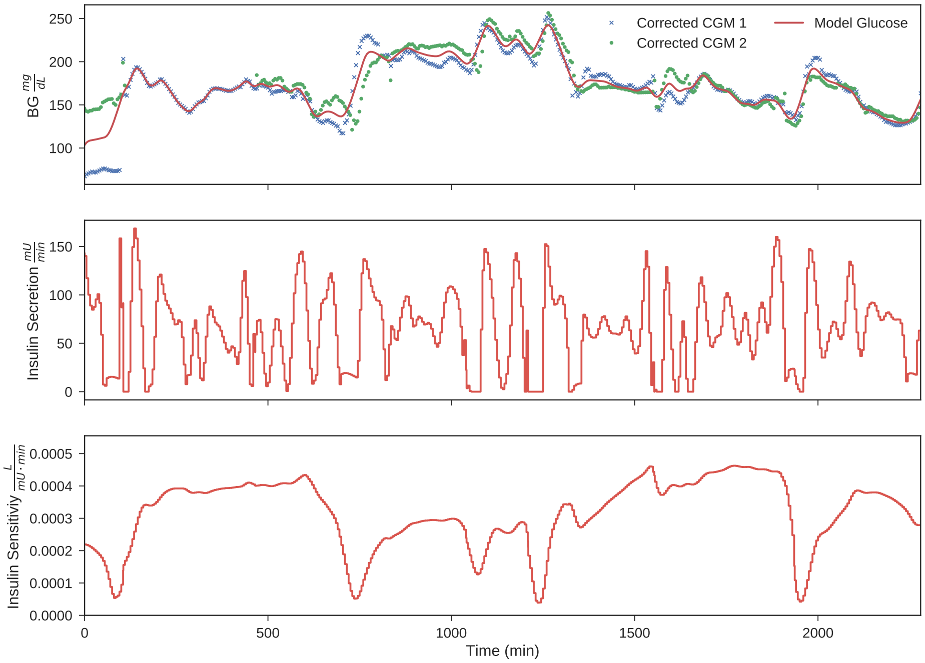


Figure 67: Virtual Patient 15

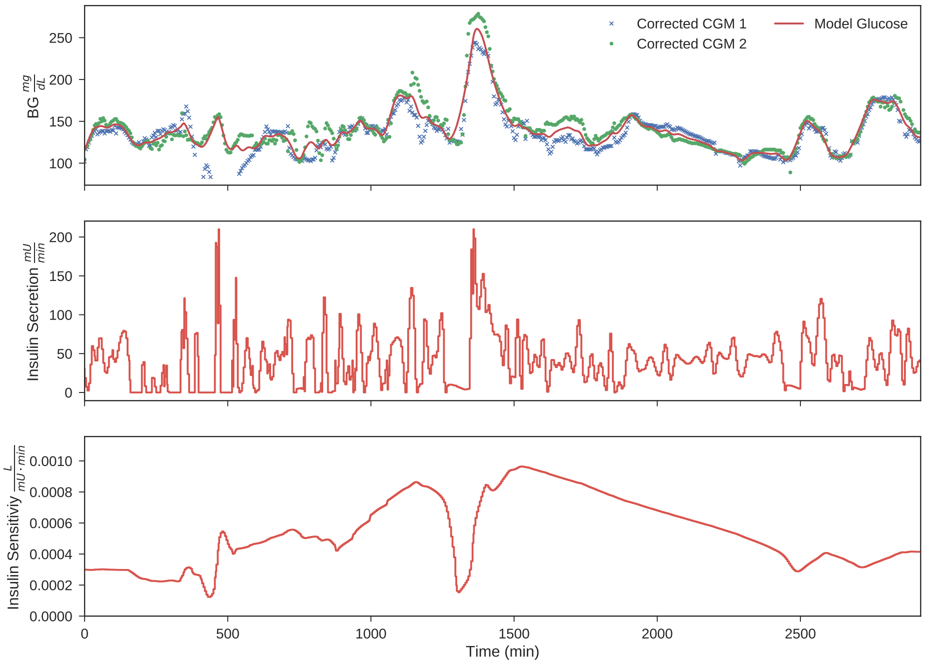


Figure 68: Virtual Patient 16

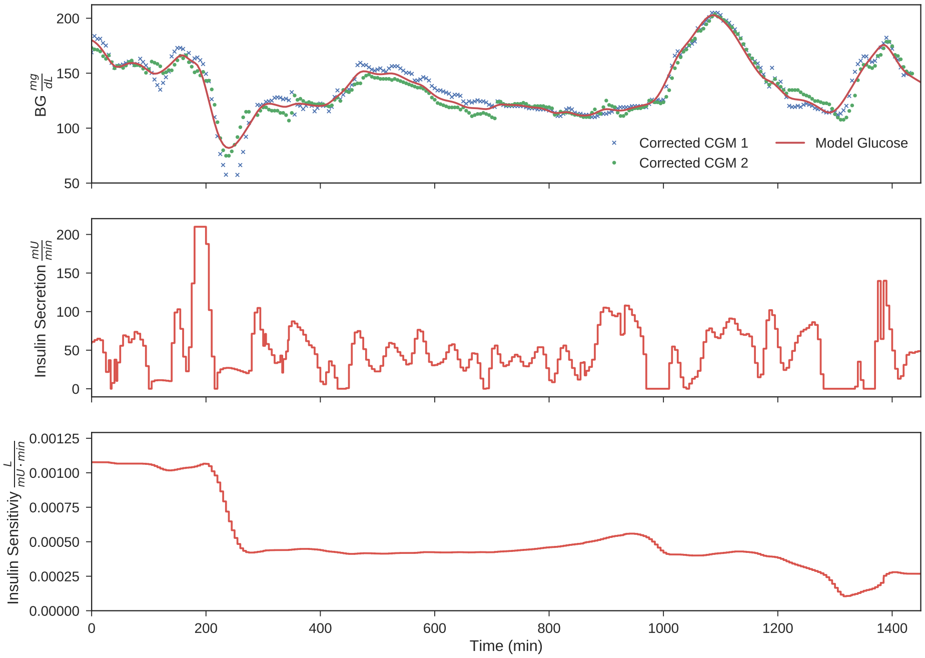


Figure 69: Virtual Patient 17

APPENDIX C

ULTRADIAN INSULIN RHYTHMS

For all virtual patients:

-: Amplitude Spectral Density

-: Observed frequency limits of ultradian insulin secretion rhythms in the literature

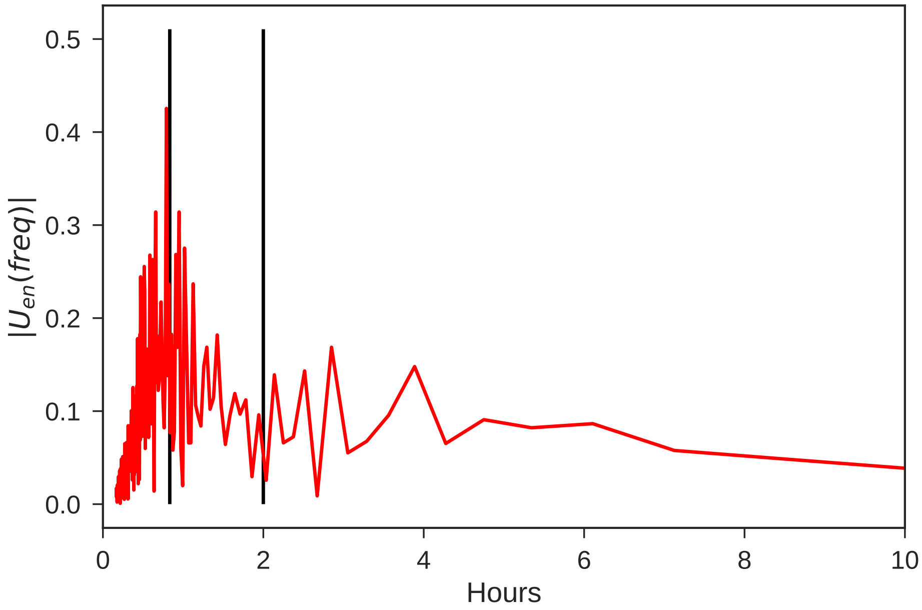


Figure 70: Virtual Patient 1

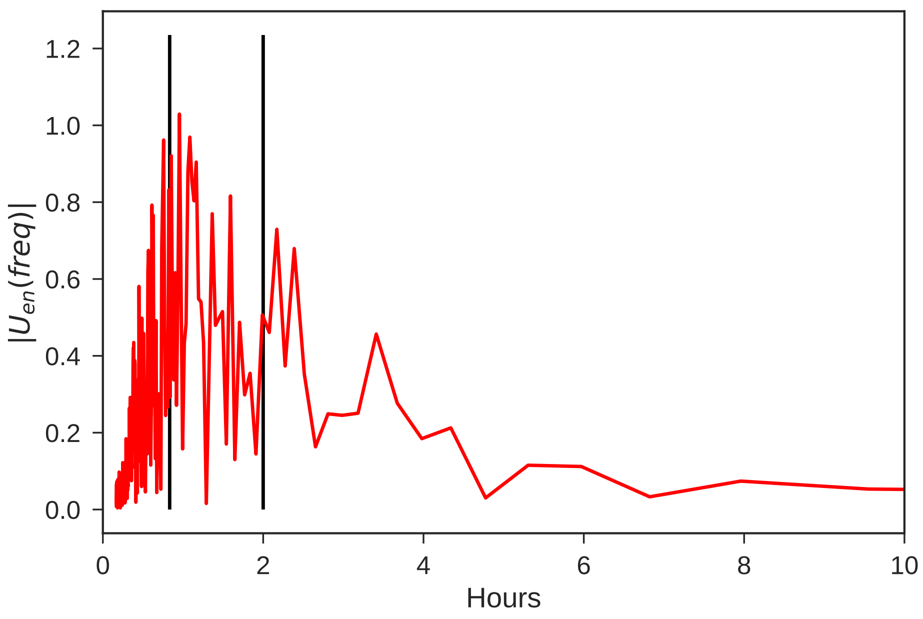


Figure 71: Virtual Patient 2

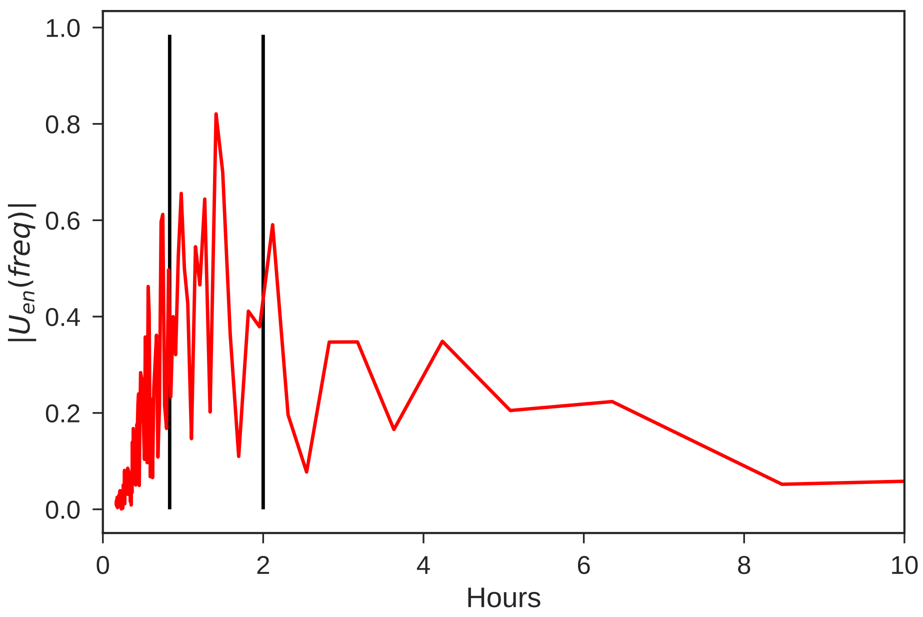


Figure 72: Virtual Patient 3

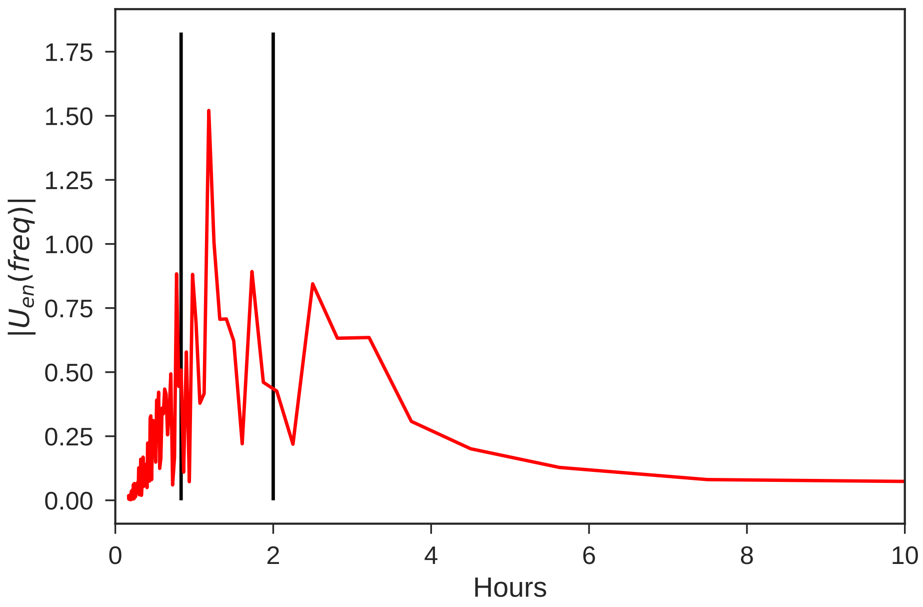


Figure 73: Virtual Patient 4

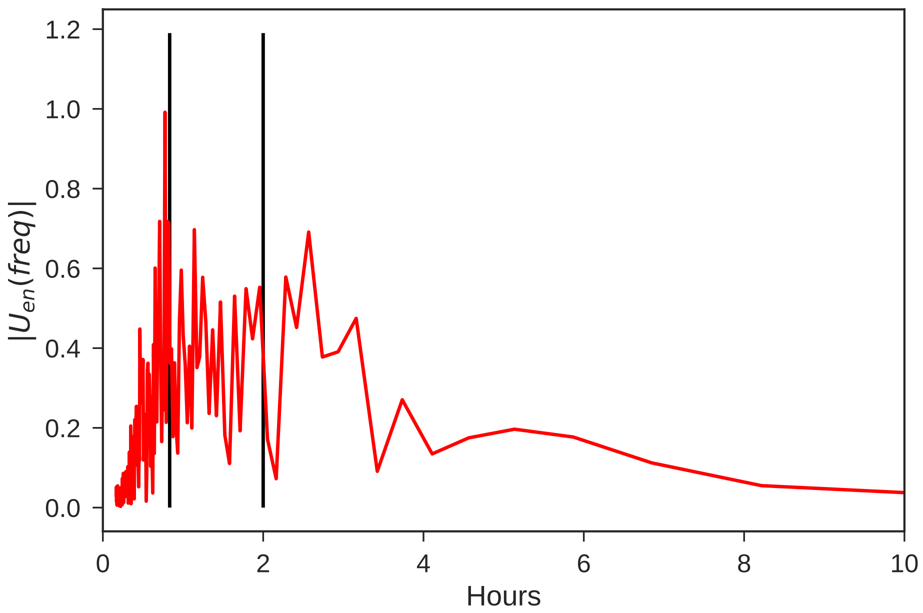


Figure 74: Virtual Patient 5

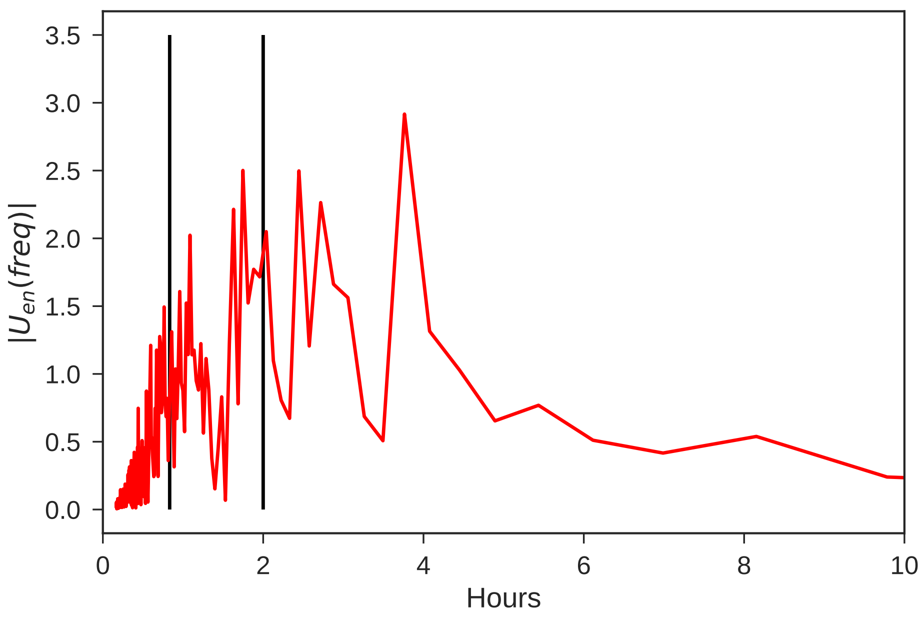


Figure 75: Virtual Patient 6

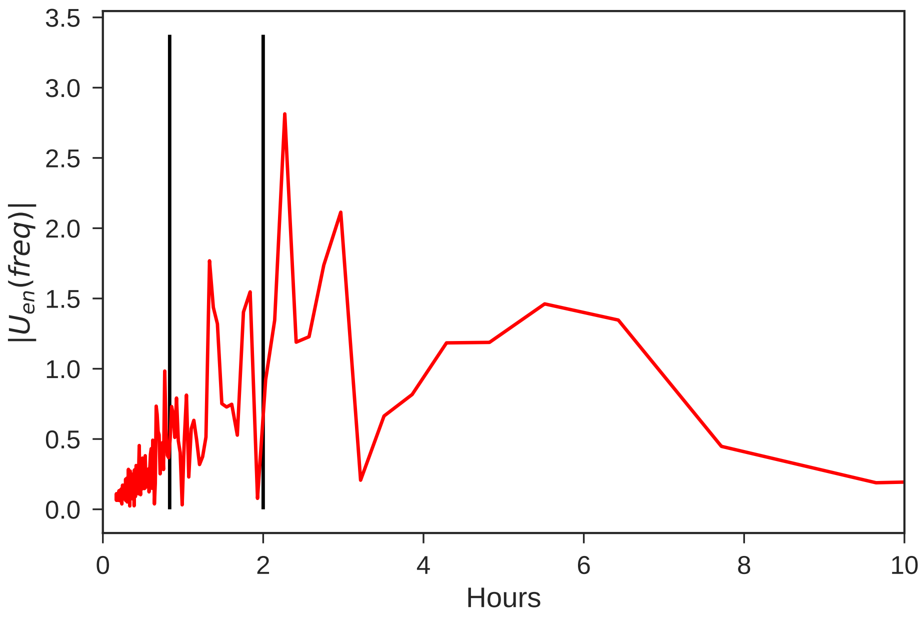


Figure 76: Virtual Patient 7

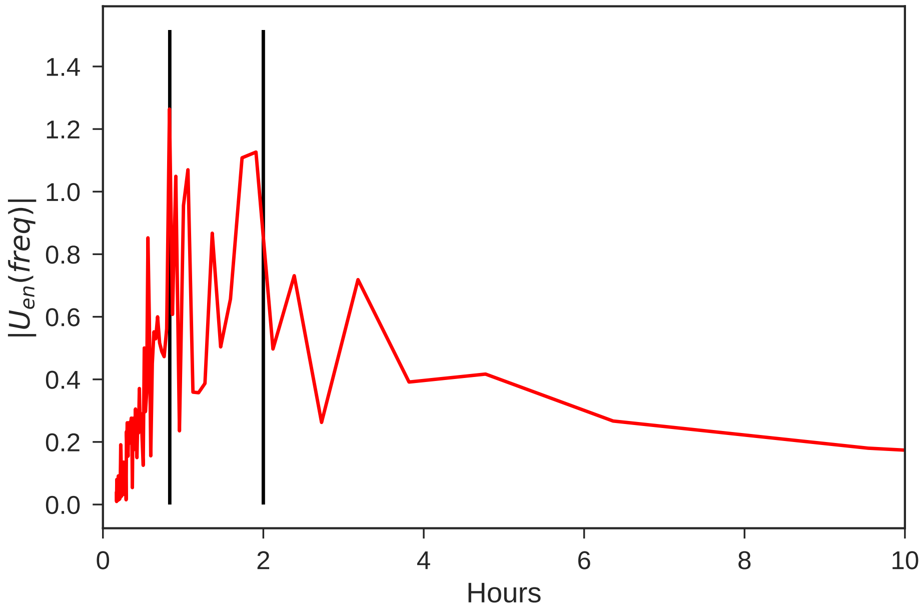


Figure 77: Virtual Patient 8

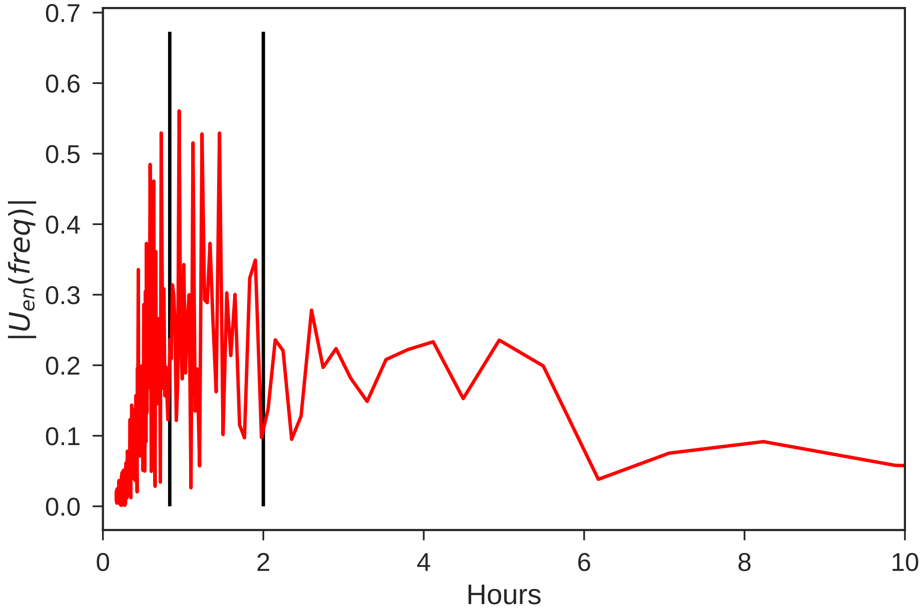


Figure 78: Virtual Patient 9

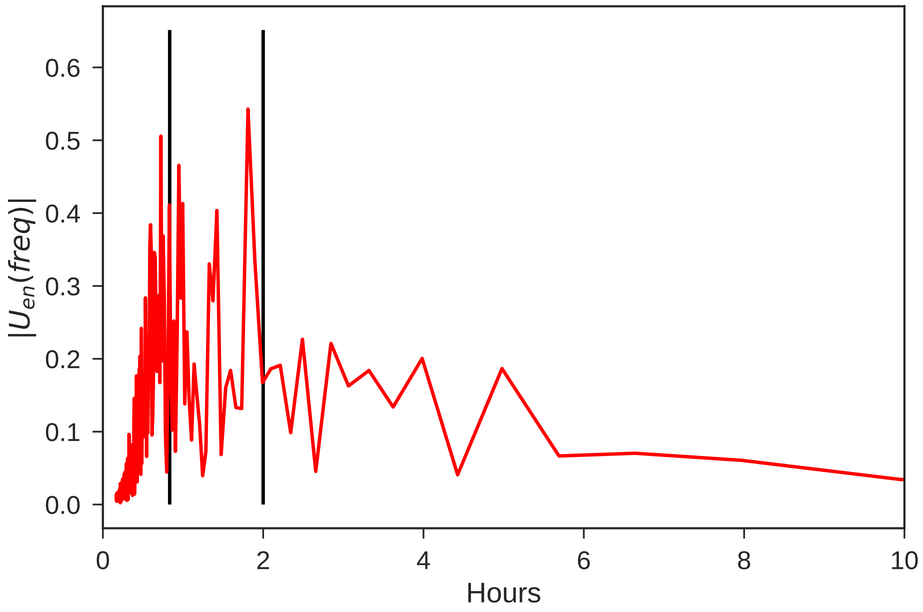


Figure 79: Virtual Patient 10

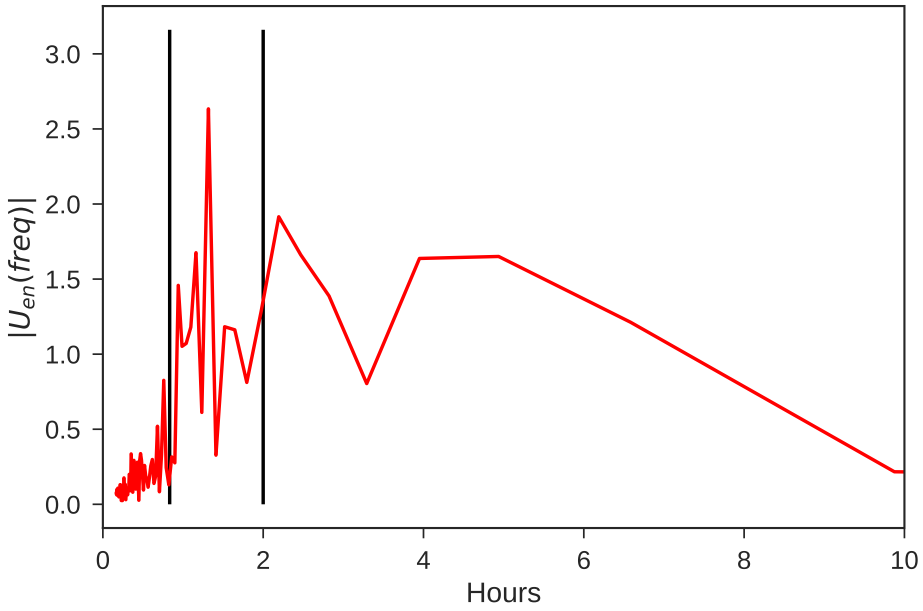


Figure 80: Virtual Patient 11

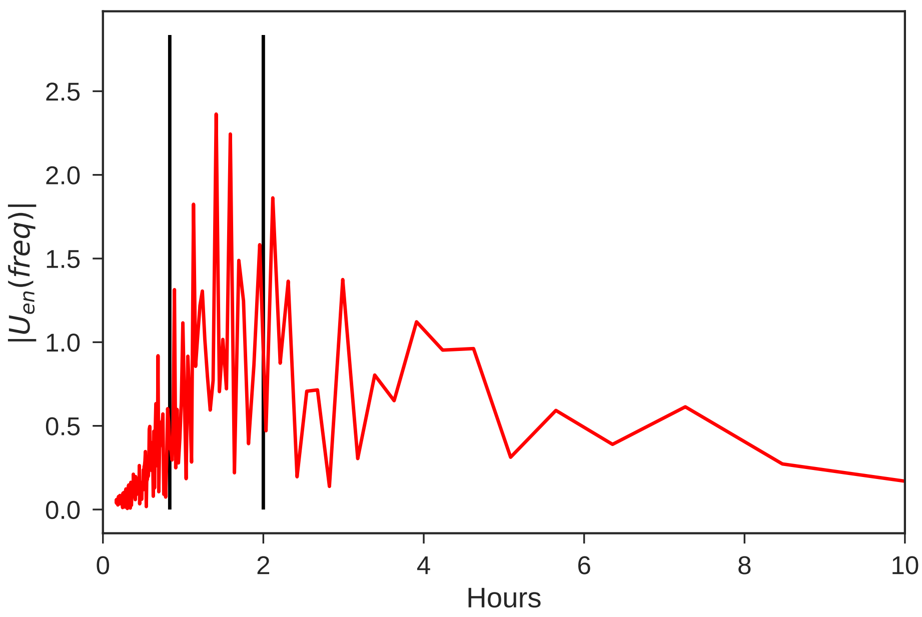


Figure 81: Virtual Patient 12

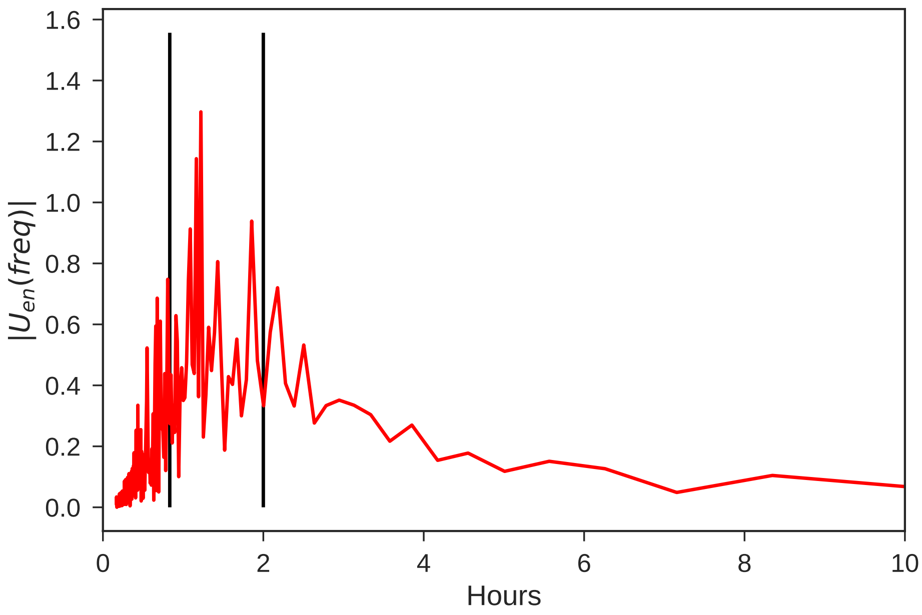


Figure 82: Virtual Patient 13

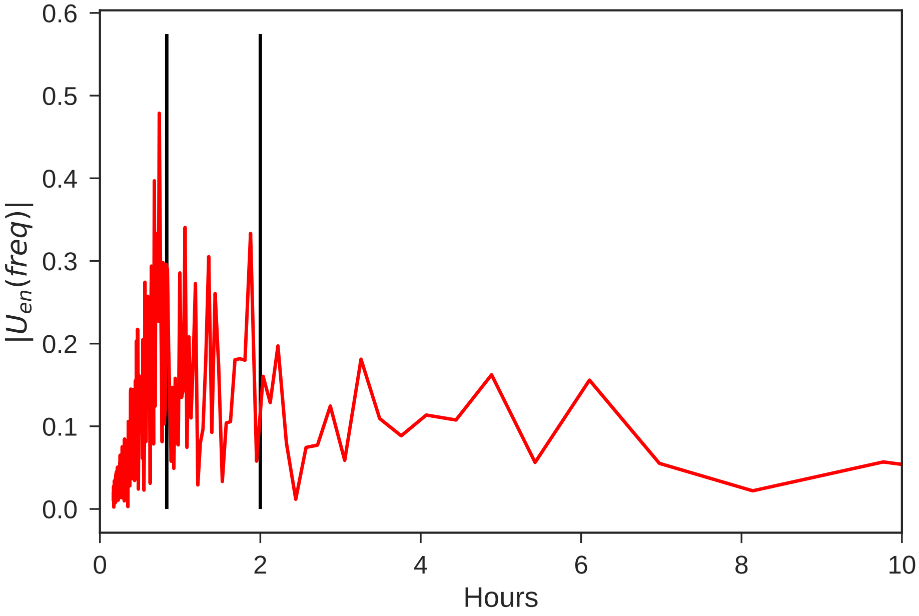


Figure 83: Virtual Patient 14

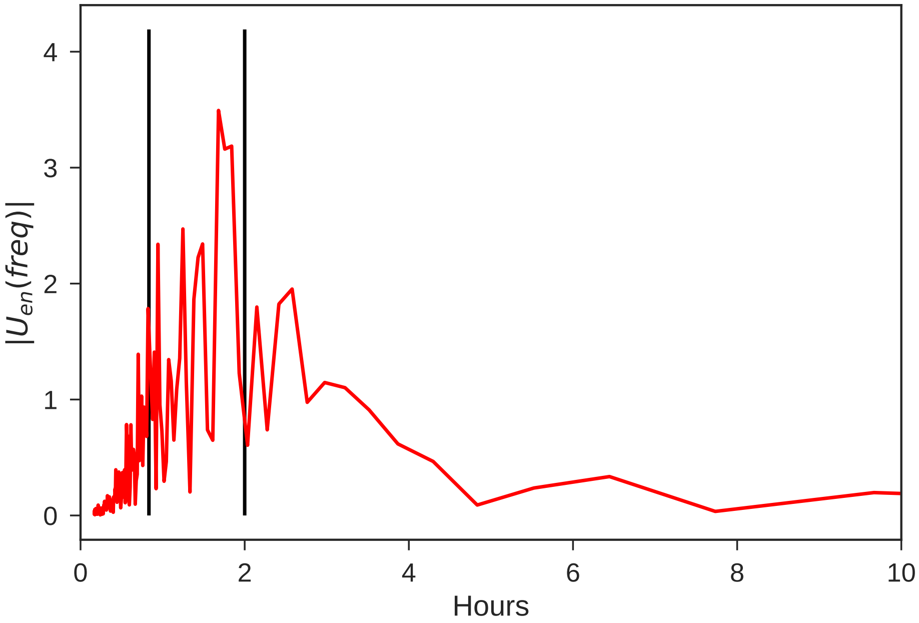


Figure 84: Virtual Patient 15

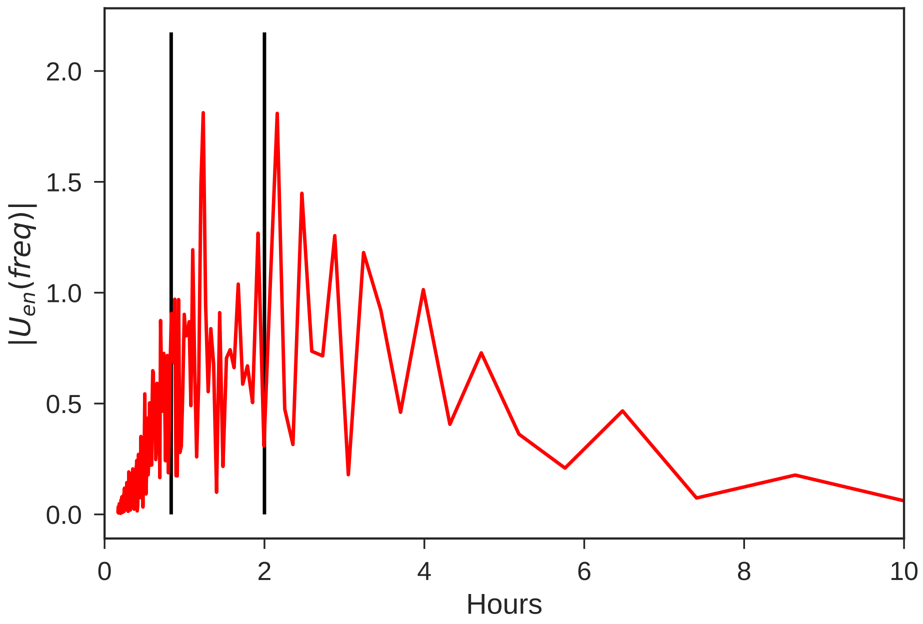


Figure 85: Virtual Patient 16

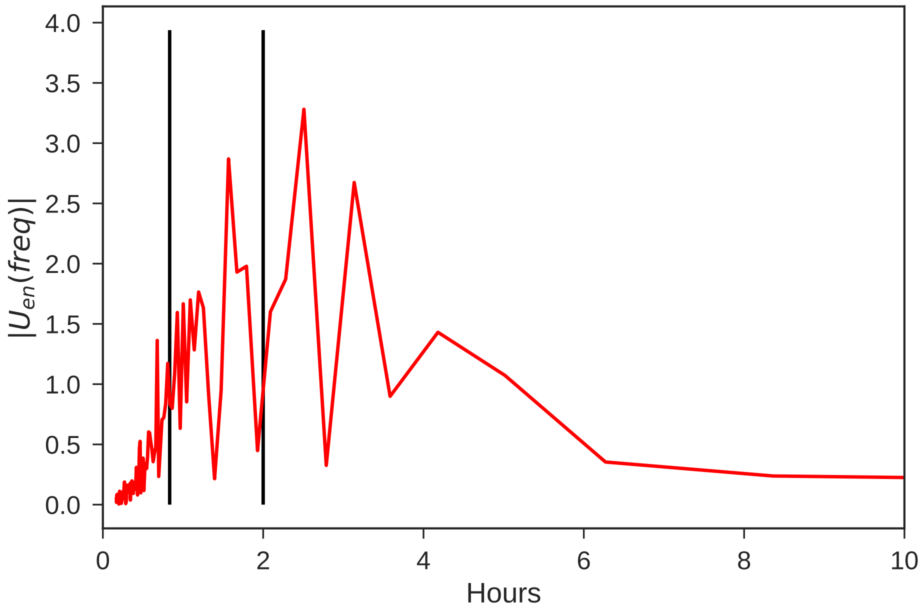


Figure 86: Virtual Patient 17

APPENDIX D

RESPONSIVE PANCREAS BASAL AND RESIDUAL INSULIN SECRETION RATES

Figure 87 below shows the always positive basal rate of insulin secretion as well as the residual component which gives rise to ultradian oscillations in insulin secretion.

As seen in fig. 87, the glucose dependent (-) contribution to the overall insulin secretion rate is slowly varying compared to the glucose independent rate (-))but exhibits long-term trends not present in the high frequency glucose independent component. Rather, the glucose independent component of insulin secretion is generally centered around zero and will give rise to expected ultradian rhythms in any simulation of a virtual patient.

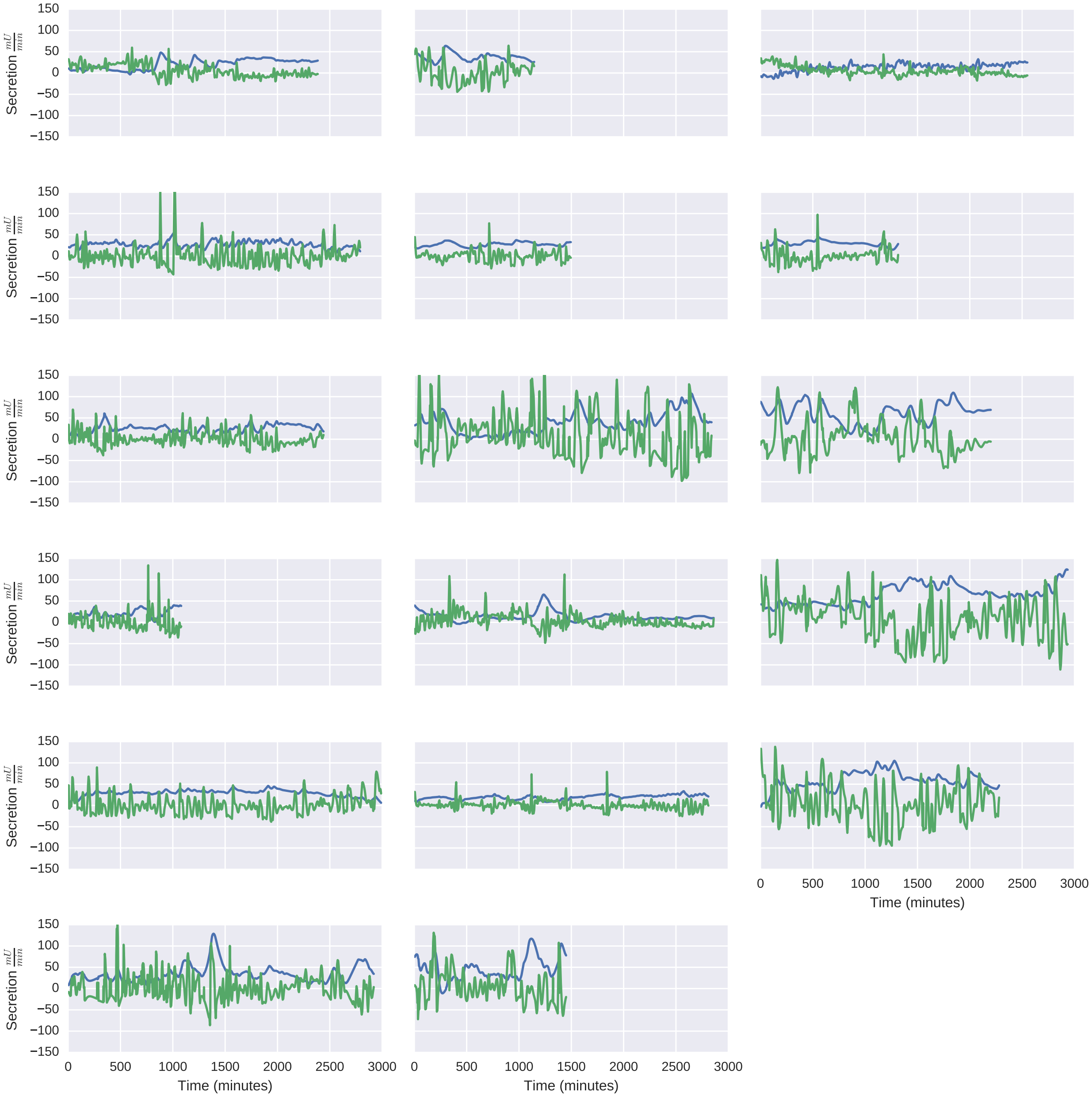


Figure 87: (-) denotes the glucose dependent (PID) rates of insulin secretion and (-) shows the glucose independent residual.

APPENDIX E

MPC CODE

Listing 1: PyMPC Module Source Code: Unlike most research grade code, the code here takes pains to check for improperly formatted/configured inputs and provide useful and informative error messages. Thorough documentation and API information is available upon request but is not included here for brevity.

```
from pyomo.environ import (ConcreteModel, Constraint, Var, Param,
↪  Objective, RangeSet)
from pyomo.opt import SolverFactory
import numpy as np
from numbers import Number


class _IncompatibleMatrix(Exception):
    pass


class _StateSpace_Error(Exception):
    pass


class _MPC_Error(Exception):
    pass


class _Bounds_Error(Exception):
    pass


class _Update_Error(Exception):
    pass


class MPC(object):

    def __init__(self, A=np.array([]), B=np.atleast_2d(np.array([])),
    ↪  C=np.array([]), D=np.atleast_2d(np.array([])),
                 P=20, M=3, Zone={}, Gamma=np.atleast_2d(np.array([])),
                 ↪  S=np.atleast_2d(np.array([])), **kwargs):
        self.A = np.atleast_2d(np.array(A)).copy()
```

```python
        self.B = np.atleast_2d(np.array(B)).copy()
        self.C = np.atleast_2d(np.array(C)).copy()
        self.D = np.atleast_2d(np.array(D)).copy()
        self.P = P
        self.M = M
        self.Zone = Zone
        self.Gamma = np.atleast_2d(np.array(Gamma)).copy()
        self.S = np.atleast_2d(np.array(S)).copy()
        self._eval_kwargs(kwargs)
        self._checkSS()
        self._checkZone()
        self.__setupMPC()

    def _checkSS(self):
        if self.A.shape[1] == 0:
            raise _StateSpace_Error('A is missing!')
        if self.B.shape[1] == 0:
            raise _StateSpace_Error('B is missing, control not possible!')
        if self.B.shape[1] != 0 and self.A.shape[0] != self.B.shape[0]:
            raise _IncompatibleMatrix('Dimensions of A and B are incompatible')
        if self.A.shape[1] != self.C.shape[1]:
            raise _IncompatibleMatrix('Dimensions of A and C are incompatible')
        if self.D.shape[1] != 0 and self.C.shape[0] != self.D.shape[1]:
            raise _IncompatibleMatrix('Dimensions of C and D are incompatible')
        if self.D.shape[1] != 0 and self.B.shape[1] != 0 and self.D.shape[1] != self.B.shape[1]:
            raise _IncompatibleMatrix('Dimensions of B and D are incompatibe')
        if self.A.shape[0] == 0:
            raise _StateSpace_Error('A is missing')
        if self.C.shape[1] == 0:
            raise _StateSpace_Error('C is missing')
        if self.A.shape[0] != len(self.States):
            raise _StateSpace_Error('Dimensions of A do not match given States')
        if self.B.shape[1] != 0 and self.B.shape[1] != len(self.Inputs):
            raise _StateSpace_Error('Dimensions of B do not match given Inputs')
        if self.D.shape[1] != 0 and self.D.shape[1] != len(self.Inputs):
            raise _StateSpace_Error('Dimensions of D do not match given Inputs')
        if self.D.shape[1] != 0 and self.D.shape[0] != len(self.Outputs):
            raise _StateSpace_Error('Dimensions of D do not match given Outputs')

    def __addstr(self, attribute, string):
        tmp = []
        if type(string) == list:
            setattr(self, attribute, string)
        else:
            tmp.append(string)
            setattr(self, attribute, tmp)

    def _eval_kwargs(self, kwargs):
        if 'States' in kwargs:
            self.__addstr('States', kwargs['States'].copy())
            if len(set(self.States)) != len(self.States):
                raise _StateSpace_Error('List of state names cannot have any duplicates')
        else:
            self.States = ['s' + str(i) for i in range(0, self.A.shape[0])]
        if 'Inputs' in kwargs and self.B.shape[0] == 0:
            raise _StateSpace_Error('Inputs provided but B is missing')
        elif 'Inputs' in kwargs:
            self.__addstr('Inputs', kwargs['Inputs'].copy())
            if len(set(self.Inputs)) != len(self.Inputs):
                raise _StateSpace_Error('List of input names cannot have any duplicates')
        else:
            self.Inputs = ['i' + str(i) for i in range(0, self.B.shape[1])]
        if 'Outputs' in kwargs:
            self.__addstr('Outputs', kwargs['Outputs'].copy())
            if len(set(self.Outputs)) != len(self.Outputs):
                raise _StateSpace_Error('List of output names cannot have any duplicates')
        else:
            self.Outputs = ['o' + str(i) for i in range(0, self.C.shape[0])]
        if 'Manipulated' in kwargs:
            self.__addstr('Manipulated', kwargs['Manipulated'].copy())
            if len(set(self.Manipulated)) != len(self.Manipulated):
                raise _MPC_Error('List of manipulated variables cannot have any duplicates')
        else:
            self.Manipulated = self.Inputs
        if 'OutputRegulated' in kwargs:
            self.__addstr('OutputRegulated', kwargs['OutputRegulated'].copy())
        if 'Controlled' in kwargs:
            self.__addstr('Controlled', kwargs['Controlled'].copy())
            if len(set(self.Controlled)) != len(self.Controlled):
                raise _MPC_Error('List of controlled variables cannot have any duplicates')
        else:
            self.Controlled = self.Outputs
        if 'Bounds' in kwargs:
            self.Bounds = kwargs.get('Bounds')
            self._checkBounds()
        else:
            self._checkBounds()
        if 'dBounds' in kwargs:
            self.dBounds = kwargs.get('dBounds')
            self._checkdBounds()
        else:
            self._checkdBounds()
        if 'Solver' in kwargs:
            self.solver = kwargs['Solver']
        else:
            self.solver = 'ipopt'
        if 'Hold_Type' in kwargs:
            self.Hold_Type = kwargs['Hold_Type']
            if self.Hold_Type not in ['Zero_Order', 'First_Order']:
                raise _MPC_Error('Hold type %s is not recognized. Currently supported options are Zero Order and First Order holds' % self.Hold_Type)
        else:
            self.Hold_Type = 'First_Order'
        list_all = self.States + self.Inputs + self.Outputs
        if len(set(list_all)) != len(list_all):
            raise _MPC_Error('Listed states, inputs, and outputs CANNOT have any duplicates or overlap')
        self._checkMPC()

    def _checkMPC(self):
        if hasattr(self, 'OutputRegulated'):
            OutputRegulated = self.OutputRegulated
        else:
            OutputRegulated = []
        if self.B.shape[1] == 0:
            raise _MPC_Error('No input matrix provided, cannot calculate control action!')
        for m in self.Manipulated:
            if m not in self.Inputs:
                raise _MPC_Error('Manipulated variable %s not found in list of model inputs' % m)
        for c in self.Controlled:
            if c not in self.Outputs:
                raise _MPC_Error('Controlled variable %s not found in list of model outputs' % c)
        if self.Gamma.shape[0] != len(self.Controlled) + len(OutputRegulated):
            raise _MPC_Error('Dimensions of Gamma do not agree with number of controlled variables')
        if self.Gamma.shape[0] != self.Gamma.shape[1]:
            raise _MPC_Error('Gamma must be square!')
        if self.S.shape[0] != len(self.Manipulated):
            raise _MPC_Error('Dimensions of S do not agree with number of manipulated variables')
        if self.S.shape[0] != self.S.shape[1]:
            raise _MPC_Error('S must be square!')

    def _checkZone(self):
        if hasattr(self, 'OutputRegulated'):
            OutputRegulated = self.OutputRegulated
        else:
            OutputRegulated = []
        for CV in self.Controlled + OutputRegulated:
            if CV not in self.Zone:
                raise _MPC_Error('Zone must be provided for controlled variable %s' % CV)
            tmp = self.Zone[CV]
            if 'Upper' not in tmp and 'Lower' not in tmp:
                raise _MPC_Error('Upper or Lower zone boundaries must be provided for controlled variable %s' % CV)
            if type(tmp) is not dict:
                raise _MPC_Error('Zone for %s must be a dictionary' % CV)
            if len(tmp) == 0:
                raise _MPC_Error('Zone for %s must contain a bound!' % CV)
            for b in tmp.keys():
                if b not in ['Upper', 'Lower']:
                    raise _MPC_Error('%s in zone for %s is not recongnized' % (b, CV))
                tmp2 = tmp[b]
                if not hasattr(tmp2, '__iter__'):
                    tmp2 = [tmp2]
                    self.Zone[CV][b] = tmp2
                elif len(tmp2) != self.P:
                    raise _MPC_Error('%s bound for %s zone must be scalar or defined for all points in prediction horizon' % (b, CV))
                else:
                    for k in tmp2:
                        if not isinstance(k, Number):
                            raise _MPC_Error('all % bounds for %s zone must be numbers!' % (b, CV))

    def _checkBounds(self):
        if not hasattr(self, 'Bounds'):
            self.Bounds = {}
        for m in self.Bounds.keys():
            if m not in self.Manipulated:
                raise _MPC_Error('Error in bounds: %s not found in manipulated variables' % m)
            if type(self.Bounds[m]) is not dict:
                raise _MPC_Error('Bounds for %s must be a dictionary' % m)
            for b in self.Bounds[m].keys():
                if b not in ['Upper', 'Lower']:
                    raise _Bounds_Error('%s in Bounds for %s is not recognized' % (b, m))
                if not hasattr(self.Bounds[m][b], '__iter__'):
                    self.Bounds[m][b] = [self.Bounds[m][b]]
                elif not isinstance(self.Bounds[m][b], Number) or len(self.Bounds[m][b]) != self.P or self.Bounds[m][b].ndim != 1:
                    raise _Bounds_Error('%s bound for manipulated variable %s must be a scalar or an iterable of length equal to P')
                elif isinstance(self.Bounds[m][b], Number):
                    self.Bounds[m][b] = [self.Bounds[m][b]]
                else:
                    self.Bounds[m][b] = list(self.Bounds[m][b])
                for n in self.Bounds[m][b]:
                    if not isinstance(n, Number):
                        raise _Bounds_Error('All values for %s %s bound must be numbers' % (m, b))

    def _checkdBounds(self):
        if not hasattr(self, 'dBounds'):
            self.dBounds = {}
        for m in self.dBounds.keys():
            if m not in self.Manipulated:
                raise _MPC_Error('Error in dbounds: %s not found in manipulated variables' % m)
            if type(self.dBounds[m]) is not dict:
                raise _MPC_Error('dBounds for %s must be a dictionary' % m)
            for b in self.dBounds[m].keys():
                if b not in ['Upper', 'Lower']:
                    raise _Bounds_Error('%s in dBounds for %s is not recognized' % (b, m))
                if not hasattr(self.dBounds[m][b], '__iter__'):
                    self.dBounds[m][b] = [self.dBounds[m][b]]
                elif not isinstance(self.dBounds[m][b], Number) or len(self.dBounds[m][b]) != self.P or self.dBounds[m][b].ndim != 1:
                    raise _Bounds_Error('%s dbound for manipulated variable %s must be a scalar or an iterable of length equal to P')
                elif isinstance(self.dBounds[m][b], Number):
                    self.dBounds[m][b] = [self.dBounds[m][b]]
                else:
                    self.dBounds[m][b] = list(self.dBounds[m][b])
                for n in self.dBounds[m][b]:
                    if not isinstance(n, Number):
                        raise _Bounds_Error('All values for %s %s dbound must be numbers' % (m, b))

    def __setupMPC(self):
        if hasattr(self, 'OutputRegulated'):
            OutputRegulated = self.OutputRegulated
        else:
            OutputRegulated = []
        model = ConcreteModel()
        model.statespace = {'A': self.A, 'B': self.B, 'C': self.C, 'D': self.D}
        model.weights = {'Gamma': self.Gamma, 'S': self.S}
        model.states_i = {s: i for i, s in enumerate(self.States)}
        model.inputs_i = {ii: i for i, ii in enumerate(self.Inputs)}
        model.manipulated_i = {ii: i for i, ii in enumerate(self.Manipulated)}
        model.outputs_i = {oo: i for i, oo in enumerate(self.Outputs)}
        model.OutputRegulated = OutputRegulated
        model.controlled = self.Controlled
        model.output_regulated_i = {oo: i for i, oo in enumerate(model.OutputRegulated)}
        model.controlled_i = {oo: i for i, oo in enumerate(self.Controlled)}
        model.A = Param(self.States, self.States, initialize=self._defineA)
        if self.B.shape[1] != 0:
            model.B = Param(self.States, self.Inputs, initialize=self._defineB)
        model.C = Param(self.Outputs, self.States, initialize=self._defineC)
        if self.D.shape[1] != 0:
            model.D = Param(self.Outputs, self.Inputs, initialize=self._defineD)
        model.bounds = self.Bounds
        model.dbounds = self.dBounds
        model.Phorizon = RangeSet(0, self.P - 1)
        model.Mhorizon = RangeSet(0, self.M - 1)
        model = self._setZone(model)
        model = self._setBounds(model)
        model = self._setdBounds(model)
        model.CVSP = Param(self.Controlled + model.OutputRegulated, ['Upper', 'Lower'], model.Phorizon, initialize=self._defineSP, mutable=True)
        model.MVbounds = Param(self.Manipulated, ['Upper', 'Lower'], model.Mhorizon, initialize=self._defineBounds, mutable=True)
        model.dMVbounds = Param(self.Manipulated, ['Upper', 'Lower'], model.Mhorizon, initialize=self._definedBounds, mutable=True)
        model.X = Var(self.States, model.Phorizon, initialize=0.0)
        model.MV = Var(self.Manipulated, model.Mhorizon, bounds=self._assignBounds, initialize=0.0)
        model.dMV = Var(self.Manipulated, model.Mhorizon, bounds=self._assigndBounds, initialize=0.0)
        model.Y = Var(self.Outputs, model.Phorizon, initialize=0.0)
        model.deltaSP = Var(self.Controlled + model.OutputRegulated, model.Phorizon, bounds=self._assignSP, initialize=0.0)
        model.U = Param(self.Inputs, model.Phorizon, initialize=0.0, mutable=True)
        model.MVprev = Param(self.Manipulated, initialize=0.0, mutable=True)
        model.x0 = Param(self.States, initialize=0.0, mutable=True)
        model.Hold_Type = self.Hold_Type
        model.Gamma = Param(self.Controlled + self.OutputRegulated, self.Controlled + self.OutputRegulated, initialize=self._defineGamma, mutable=True)
        model.S = Param(self.Manipulated, self.Manipulated, initialize=self._defineS, mutable=True)

        def _stepSS(model, s, i):
            if i == 0:
                return model.X[s, i] == model.x0[s]
            else:
                asum = sum([model.A[s, si] * model.X[si, i - 1] for si in self.States])
                bsum = 0.0
                for ii in self.Inputs:
                    if ii in self.Manipulated and i < self.M:
                        bsum += model.B[s, ii] * (model.U[ii, i - 1] + model.MV[ii, i - 1])
                    elif ii in self.Manipulated and self.Hold_Type == "First_Order":
                        bsum += model.B[s, ii] * (model.U[ii, i - 1] + model.MV[ii, model.Mhorizon.last()])
                    else:
                        bsum += model.B[s, ii] * model.U[ii, i - 1]
                return model.X[s, i] == asum + bsum
        model.SS_Constraint = Constraint(self.States, model.Phorizon, rule=_stepSS)

        def _getY(model, o, i):
            csum = sum([model.C[o, s] * model.X[s, i] for s in self.States])
            dsum = 0.0
            if hasattr(model, 'D'):
                for ii in self.Inputs:
                    if ii in self.Manipulated and i < self.M:
                        dsum += model.D[o, ii] * (model.U[ii, i] + model.MV[ii, i])
                    else:
                        dsum += model.D[o, ii] * model.U[ii, i]
            return model.Y[o, i] == csum + dsum
        model.Y_Constraint = Constraint(self.Outputs, model.Phorizon, rule=_getY)

        def _getdMV(model, m, i):
            if i == 0:
                # print("fooey!")
                return model.dMV[m, i] == model.MV[m, i] - model.MVprev[m]
            else:
                return model.dMV[m, i] == model.MV[m, i] - model.MV[m, i - 1]
        model.dMV_Constraint = Constraint(self.Manipulated, model.Mhorizon, rule=_getdMV)

        def _Obj(model):
            ysum = 0.0
            for o1 in self.Controlled:
                for o2 in self.Controlled:
                    for i in model.Phorizon:
                        if o1 in self.Controlled:
                            elem1 = model.Y[o1, i]
                        if o2 in self.Controlled:
                            elem2 = model.Y[o2, i]
                        ysum += (elem1 - model.deltaSP[o1, i]) * model.Gamma[o1, o2] * (elem2 - model.deltaSP[o2, i])
            for o1 in model.OutputRegulated:
                for o2 in model.OutputRegulated:
                    for i in model.Mhorizon:
                        if o1 in self.OutputRegulated:
                            if i <= model.Mhorizon.last():
                                elem1 = model.MV[o1, i]
                            elif i > model.Mhorizon.last() and self.Hold_Type == 'First_Order':
                                elem1 = model.MV[o1, model.Mhorizon.last()]
                                # elem1 = 0.0
                            elif i > model.Mhorizon.last() and self.Hold_Type == 'Zero_Order':
                                elem1 = 0.0
                            else:
                                elem1 = 0.0
                        if o2 in self.OutputRegulated:
                            if i <= model.Mhorizon.last():
                                elem2 = model.MV[o2, i]
                            elif i > model.Mhorizon.last() and self.Hold_Type == 'First_Order':
                                elem2 = model.MV[o2, model.Mhorizon.last()]
                                # elem2 = 0.0
                            elif i > model.Mhorizon.last() and self.Hold_Type == 'Zero_Order':
                                elem2 = 0.0
                            else:
                                elem2 = 0.0
                        ysum += (elem1 - model.deltaSP[o1, i]) * model.Gamma[o1, o2] * (elem2 - model.deltaSP[o2, i])
            dssum = 0.0
            for mv1 in self.Manipulated:
                for mv2 in self.Manipulated:
                    for i in model.Mhorizon:
                        dssum += model.dMV[mv1, i] * model.S[mv1, mv2] * model.dMV[mv2, i]


            return ysum + dssum
        model.Objective = Objective(rule=_Obj)
        self._pyomoMPC = model
        self.__opt = SolverFactory(self.solver)

    def _setZone(self, model):
        model._zone = self.Zone.copy()
        for CV in self.Controlled + model.OutputRegulated:
            tmp = model._zone[CV]
            for b in ['Upper', 'Lower']:
                if b not in tmp and b == 'Upper':
                    model._zone[CV][b] = list(np.inf * np.ones(self.P))
                elif b not in tmp and b == 'Lower':
                    model._zone[CV][b] == list(-np.inf * np.ones(self.P))
                elif len(tmp[b]) == 1:
                    model._zone[CV][b] = list(model._zone[CV][b][0] * np.ones(self.P))
        return model

    def _setBounds(self, model):
        model._bounds = self.Bounds.copy()
        for MV in self.Manipulated:
            if MV not in model._bounds:
                model._bounds[MV] = {'Lower': -np.inf * np.ones(self.M), 'Upper': np.inf * np.ones(self.M)}
            tmp = model._bounds[MV]
            for b in ['Upper', 'Lower']:
                if b not in tmp and b == 'Upper':
                    model._bounds[MV][b] = list(np.inf * np.ones(self.M))
                elif b not in tmp and b == 'Lower':
                    model._bounds[MV][b] = list(-np.inf * np.ones(self.M))
                elif len(tmp[b]) == 1:
                    model._bounds[MV][b] = list(model._bounds[MV][b][0] *
                     ↪  np.ones(self.M))
        return model

    def _setdBounds(self, model):
        model._dbounds = self.dBounds.copy()
        for MV in self.Manipulated:
            if MV not in model._dbounds:
                model._dbounds[MV] = {'Lower': -np.inf * np.ones(self.M),
                 ↪  'Upper': np.inf * np.ones(self.M)}
            tmp = model._dbounds[MV]
            for b in ['Upper', 'Lower']:
                if b not in tmp and b == 'Upper':
                    model._dbounds[MV][b] = list(np.inf * np.ones(self.M))
                elif b not in tmp and b == 'Lower':
                    model._dbounds[MV][b] = list(-np.inf * np.ones(self.M))
                elif len(tmp[b]) == 1:
                    model._dbounds[MV][b] = list(model._dbounds[MV][b][0] *
                     ↪  np.ones(self.M))
        return model

    def MPCUpdate(self, x0=np.array([]), **kwargs):
        x0 = np.array(x0)
        if len(x0) == 0:
            x0 = np.zeros(self.A.shape[0])
        elif len(x0) != self.A.shape[0]:
            raise _Update_Error('Dimensions of given states does not jive
             ↪  with state space!')
        for s in self.States:
            i = self._pyomoMPC.states_i[s]
            self._pyomoMPC.x0[s] = x0[i]
        if 'U' in kwargs:
            Utmp = np.atleast_2d(np.array(kwargs['U'])).T
            if Utmp.shape[0] != self.B.shape[1]:
                raise _Update_Error('Input vector must have the same number
                 ↪  of rows as inputs (Columns of B)')
            if Utmp.shape[1] != 1 and Utmp.shape[1] != self.P:
                raise _Update_Error('Input vector must be constant over
                 ↪  prediction horizon or a value provided at every point
                 ↪  over the prediction horizon')
            if Utmp.shape[1] == 1:
                for ii in self.Inputs:
                    for i in self._pyomoMPC.Phorizon:
                        j = self._pyomoMPC.inputs_i[ii]
                        self._pyomoMPC.U[ii, i] = Utmp[j, 0]
            elif Utmp.shape[1] == self.P:
                for ii in self.Inputs:
                    for i in self._pyomoMPC.Phorizon:
                        j = self._pyomoMPC.inputs_i[ii]
                        self._pyomoMPC.U[ii, i] = Utmp[j, i]
        if 'Ucontrol' in kwargs:
            Uctmp = np.array(kwargs['Ucontrol'])
            if Uctmp.shape[0] != self.B.shape[1]:
                raise _Update_Error('Previous control action vector must
                 ↪  have the same number of rows as inputs (Rows of B)')
            if Uctmp.shape[1] != 1:
                raise _Update_Error('Please provide the previous control action only!')
            for i in self.Inputs:
                j = self._pyomoMPC.inputs_i[i]
                self._pyomoMPC.MVprev[i] = Uctmp[j]
        self._pyomoMPC.preprocess()
        soln = self.__opt.solve(self._pyomoMPC, keepfiles=False, tee=False)
        self._pyomoMPC.solutions.load_from(soln)
        control_action = np.zeros((len(self._pyomoMPC.manipulated_i), self.M))
        predictedx = np.zeros((self.A.shape[0], self.P - 1))
        predictedy = np.zeros((self.C.shape[0], self.P))
        for iidx, i in enumerate(self._pyomoMPC.Mhorizon):
            for mv in self.Manipulated:
                j = self._pyomoMPC.manipulated_i[mv]
                control_action[j, iidx] = self._pyomoMPC.MV[mv, i]()
        for s in self.States:
            ctr = 0
            for iidx, i in enumerate(self._pyomoMPC.Phorizon):
                if i != 0:
                    j = self._pyomoMPC.states_i[s]
                    predictedx[j, ctr] = self._pyomoMPC.X[s, i]()
                    ctr += 1
                for o in self.Outputs:
                    jy = self._pyomoMPC.outputs_i[o]
                    predictedy[jy, iidx] = self._pyomoMPC.Y[o, i]()
        for mv in self.Manipulated:
            self._pyomoMPC.MVprev[mv] = self._pyomoMPC.MV[mv, 0]
        return control_action, predictedx, predictedy

    @staticmethod
    def _defineA(model, s1, s2):
        i = model.states_i[s1]
        j = model.states_i[s2]
        return model.statespace['A'][i, j]

    @staticmethod
    def _defineB(model, s, inp):
        i = model.states_i[s]
        j = model.inputs_i[inp]
        return model.statespace['B'][i, j]

    @staticmethod
    def _defineC(model, o, s):
        i = model.outputs_i[o]
        j = model.states_i[s]
        return model.statespace['C'][i, j]

    @staticmethod
    def _defineD(model, o, inp):
        i = model.outputs_i[o]
        j = model.inputs_i[inp]
        return model.statespace['D'][i, j]

    @staticmethod
    def _defineGamma(model, cv1, cv2):
        if cv1 in model.controlled:
            i = model.controlled_i[cv1]
        elif cv1 in model.OutputRegulated:
            i = model.output_regulated_i[cv1] + len(model.controlled)
        if cv2 in model.controlled:
            j = model.controlled_i[cv2]
        elif cv2 in model.OutputRegulated:
            j = model.output_regulated_i[cv2] + len(model.controlled)
        return model.weights['Gamma'][i, j]

    @staticmethod
    def _defineS(model, mv1, mv2):
        i = model.manipulated_i[mv1]
        j = model.manipulated_i[mv2]
        return model.weights['S'][i, j]

    @staticmethod
    def _defineBounds(model, e, ul, i):
        return model._bounds[e][ul][i]

    @staticmethod
    def _assignBounds(model, e, i):
        return (model.MVbounds[e, 'Lower', i], model.MVbounds[e, 'Upper',
        ↪  i])

    @staticmethod
    def _definedBounds(model, e, ul, i):
        return model._dbounds[e][ul][i]

    @staticmethod
    def _assigndBounds(model, e, i):
        return (model.dMVbounds[e, 'Lower', i], model.dMVbounds[e, 'Upper',
        ↪  i])

    @staticmethod
    def _defineSP(model, cv, ul, i):
        return model._zone[cv][ul][i]

    @staticmethod
    def _assignSP(model, cv, i):
        return (model.CVSP[cv, 'Lower', i], model.CVSP[cv, 'Upper', i])
```

APPENDIX F

MHE CODE

Listing 2: CoPyMHE Module Source Code: Unlike most research grade code, the code here takes pains to check for improperly formatted/configured inputs and provide useful and informative error messages. Thorough documentation and API information is available upon request but is not included here for brevity.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun  2 14:23:05 2015
@author: Tim
REQUIRED ARGUMENTS
---------------------------------------------------------------------------
A: State transition matrix
B: Input matrix
C: Output/Observation matrix
m: estimation horizon
P0: Initial State Error Covariance Matrix
R: Measurement Noise Covariance Matrix
Q: State Noise Covariance Matrix
----------------------------------------------------
OPTIONAL/KEYWORK ARGUMENTS
---------------------------------------------------------------------------
x0: Vector of states at first measurement (Y[k-m+1])
    - defaults to 0 if not provided.
States: List of state names
    - defaults to 1...Ns if not provided.
Inputs: List of input names
    - defaults to 1...Ni if not provided.
Outputs: List of output names
    - defaults to 1...No if not provided.
EstimatedStates: List of states to be estimated (must match names in States)
    - defaults to all if not provided.
Bd: Disturbance matrix
    - Provides input matrix for unmeasured disturbances
```

```python
Disturbances:
    - List of names for disturbances in Bd
Qd: Disturbance noise covariance matrix
    - Must be provided if Bd is given!
"""
from pyomo.environ import (SolverFactory, ConcreteModel, RangeSet,
                           Param, Objective, Constraint, ConstraintList, Var)
import numpy as np
from numbers import Number
from numpy.linalg import inv as inv
import pdb

# Define Exception Classes


class _IncompatibleMatrix(Exception):
    pass


class _StateSpace_Error(Exception):
    pass


class _MHE_Error(Exception):
    pass


class _Bounds_Error(Exception):
    pass


class _Update_Error(Exception):
    pass


class MHE:

    def __init__(self, A=np.array([]), B=np.atleast_2d(np.array([])), C=np.array([]), D=np.atleast_2d(np.array([])), H=10,
                 R=np.array([]), P0=np.array([]), Q=np.array([]), Regularize_Disturbances="All", Estimate_IC=[], IC={}, Ts_ratio=1, **kwargs):
        self.A = np.atleast_2d(np.array(A)).copy()
        self.B = np.atleast_2d(np.array(B)).copy()
        self.C = np.atleast_2d(np.array(C)).copy()
        self.D = np.atleast_2d(np.array(D)).copy()
        self.H = H
        self.R = np.atleast_2d(np.array(R)).copy()
        self.Q = np.atleast_2d(np.array(Q)).copy()
        self.P = np.atleast_2d(np.array(P0)).copy()
        self._Regularize_Disturbances = Regularize_Disturbances
        self.Estimate_IC = Estimate_IC
        self.Ts_ratio = Ts_ratio
        if type(Ts_ratio) is not int:
            raise _MHE_Error("Ts_ratio must be an integer > 1 multiple of the state space sampling rate")
        self._eval_kwargs(kwargs)
        self._checkSS()

    def _checkSS(self):
        if self.A.shape[0] != self.A.shape[1]:
            raise _StateSpace_Error('A must be square!')
        if self.B.shape[1] != 0 and self.A.shape[0] != self.B.shape[0]:
            raise _IncompatibleMatrix('Dimensions of A and B are incompatible')
        if self.A.shape[1] != self.C.shape[1]:
            raise _IncompatibleMatrix('Dimensions of A and C are
            ↪ incompatible')
        if self.D.shape[1] != 0 and self.C.shape[0] != self.D.shape[1]:
            raise _IncompatibleMatrix('Dimensions of C and D are
            ↪ incompatible')
        if self.D.shape[1] != 0 and self.B.shape[1] != 0 and
        ↪ self.D.shape[1] != self.B.shape[1]:
            raise _IncompatibleMatrix('Dimensions of B and D are
            ↪ incompatibe')
        if self.A.shape[0] == 0:
            raise _StateSpace_Error('A is missing')
        if self.C.shape[0] == 0:
            raise _StateSpace_Error('C is missing')
        if self.A.shape[0] != len(self.States):
            raise _StateSpace_Error(
                'Dimensions of A do not match given States')
        if self.B.shape[1] != 0 and self.B.shape[1] !=
        ↪ len(list(set(self.Inputs + self.Disturbances))):
            raise _StateSpace_Error(
                'Dimensions of B do not match given Inputs')
        if self.D.shape[1] != 0 and self.D.shape[1] != len(self.Inputs):
            raise _StateSpace_Error(
                'Dimensions of D do not match given Inputs')
        if self.D.shape[1] != 0 and self.D.shape[0] != len(self.Outputs):
            raise _StateSpace_Error(
                'Dimensions of D do not match given Outputs')

    def __addstr(self, attribute, string):
        tmp = []
        if type(string) == list:
            setattr(self, attribute, string)
        else:
            tmp.append(string)
            setattr(self, attribute, tmp)

    def _eval_kwargs(self, kwargs):
        if 'States' in kwargs:
            self.__addstr('States', kwargs['States'].copy())
            if len(set(self.States)) != len(self.States):
                raise _MHE_Error(
                    'List of state names cannot have any duplicates')
        else:
            self.States = ['s' + str(i) for i in range(0, self.A.shape[0])]
        if 'Inputs' in kwargs and self.B.shape[0] == 0:
            raise _MHE_Error('Inputs provided but B is missing')
        elif 'Inputs' in kwargs:
            self.__addstr('Inputs', kwargs['Inputs'].copy())
            if len(set(self.Inputs)) != len(self.Inputs):
                raise _MHE_Error(
                    'List of input names cannot have any duplicates')
        else:
            self.Inputs = ['i' + str(i) for i in range(0, self.B.shape[1])]
        if 'Outputs' in kwargs:
            self.__addstr('Outputs', kwargs['Outputs'].copy())
            if len(set(self.Outputs)) != len(self.Outputs):
                raise _MHE_Error(
                    'List of output names cannot have any duplicates')
        else:
            self.Outputs = ['o' + str(i) for i in range(0, self.C.shape[0])]
        if 'Estimated' in kwargs:
```

```python
            self.__addstr('Estimated', kwargs['Estimated'].copy())
            if len(set(self.Estimated)) != len(self.Estimated):
                raise _MHE_Error(
                    'List of estimated state names cannot have any duplicates')
        else:
            self.Estimated = self.States
        if 'Measured' in kwargs:
            self.__addstr('Measured', kwargs['Measured'].copy())
            if len(set(self.Measured)) != len(self.Measured):
                raise _MHE_Error(
                    'List of measured outputs cannot have any duplicates')
        else:
            self.Measured = self.Outputs
        if 'Disturbances' in kwargs and self.B.shape[1] == 0:
            raise _MHE_Error('Disturbances provided but B is missing')
        elif 'Disturbances' in kwargs:
            self.__addstr('Disturbances', kwargs['Disturbances'].copy())
            if len(set(self.Disturbances)) != len(self.Disturbances):
                raise _MHE_Error(
                    'List of estimated disturbances cannot have any duplicates')
        else:
            self.Disturbances = []
        if 'L' in kwargs and hasattr(self, 'Disturbances'):
            self.L = np.atleast_2d(np.array(kwargs['L'])).copy()
        elif 'L' in kwargs:
            raise _MHE_Error(
                'Disturbance regularization matrix, L, provided with no disturbances')
        else:
            self.L = np.atleast_2d(np.array([]))
        if self._Regularize_Disturbances not in ['All', 'First', None]:
            raise _MHE_Error(
                'Disturbance regularization method not understood')
        if self._Regularize_Disturbances in ["All", "First"] and not hasattr(self, 'L'):
            raise _MHE_Error(
                'Disturbance regularization requested with no weighting matrix. Please provide L')
        self._checkMHE()
        if 'Bounds' in kwargs:
            self.Bounds = kwargs.get('Bounds')
            self._checkBounds()
        else:
            self._checkBounds()
        if 'dBounds' in kwargs:
            self.dBounds = kwargs.get('dBounds')
            self._checkdBounds()
        else:
            self._checkdBounds()
        if 'Solver' in kwargs:
            self.solver = kwargs['Solver']
        else:
            self.solver = 'ipopt'
        list_all = self.States + self.Inputs + self.Outputs
        if len(set(list_all)) != len(list_all):
            raise _MHE_Error(
                'Listed states, inputs, and outputs CANNOT have any duplicates or overlap')

    def _checkMHE(self):
        est_all = self.Estimated.copy()
        if hasattr(self, 'Disturbances'):
            est_all += self.Disturbances
        self._estimated_all = est_all
        if len(est_all) == 0:
            raise _MHE_Error('Nothing given to estimate!')
        if self.R.shape[0] != len(self.Measured):
            raise _MHE_Error(
                'The size of R is incompatible with given measured outputs')
        if self.R.shape[0] != self.R.shape[1]:
            raise _MHE_Error('R must be square!')
        # pdb.set_trace()
        if self.Q.shape[0] != (len(self.Estimated) +
        ↪  len(self.Disturbances)):
            raise _MHE_Error(
                'The size of Q is incompatible with given estimated states
                ↪  and disturbances')
        if self.Q.shape[0] != self.Q.shape[1]:
            raise _MHE_Error('Q must be square!')
        elif self.P.shape[1] != 0 and self.P.shape[0] !=
        ↪  (len(self.Estimated) + len(self.Disturbances)):
            raise _MHE_Error(
                'The size of P0 is incompatible with given estimated states
                ↪  and parameters')
        if self.P.shape[0] != self.P.shape[1] and len(self.Estimated) > 0:
            raise _MHE_Error('P must be square!')
        if hasattr(self, 'L') and self.L.shape[1] != 0 and self.L.shape[0]
        ↪  != len(self.Disturbances):
            raise _MHE_Error(
                'The size   of L is incompatible with given estimated
                ↪  disturbances')
        for s in self.Estimated:
            if s not in self.States:
                raise _MHE_Error('State %s not found in statespace' % s)

    def _checkBounds(self):
        if not hasattr(self, 'Bounds'):
            self.Bounds = {}
        for s in self.Bounds.keys():
            if s not in self.States + self.Disturbances:
                raise _MHE_Error(
                    'Error in bounds: %s not found in states or
                    ↪  disturbances' % s)
            if type(self.Bounds[s]) is not dict:
                raise _MHE_Error('Bounds for %s must be a dictionary' % s)
            for b in self.Bounds[s].keys():
                if b not in ['Upper', 'Lower']:
                    raise _Bounds_Error(
                        '%s in Bounds for %s is not recognized' % (b, s))
                if not isinstance(self.Bounds[s][b], Number):
                    raise _Bounds_Error(
                        'Value for %s  %s bound must be a number' % (s, b))
            if 'Lower' not in self.Bounds[s]:
                self.Bounds[s]['Lower'] = -np.inf
            if 'Upper' not in self.Bounds[s]:
                self.Bounds[s]['Upper'] = np.inf
        for s in self.States + self.Disturbances:
            if s not in self.Bounds:
                self.Bounds[s] = {'Lower': -np.inf, 'Upper': np.inf}

    def _checkdBounds(self):
        if not hasattr(self, 'dBounds'):
            self.dBounds = {}
        for s in self.dBounds.keys():
            if s not in self.States + self.Disturbances:
                raise _MHE_Error(
                    'Error in dbounds: %s not found in states or
                    ↪  disturbances' % s)
            if type(self.dBounds[s]) is not dict:
                raise _MHE_Error('dBounds for %s must be a dictionary' % s)
            for b in self.dBounds[s].keys():
                if b not in ['Upper', 'Lower']:
                    raise _Bounds_Error(
                        '%s in dBounds for %s is not recognized' % (b, s))
                if not isinstance(self.dBounds[s][b], Number):
                    raise _Bounds_Error(
                        'Value for %s  %s dbound must be a number' % (s, b))
            if 'Lower' not in self.dBounds[s]:
                self.dBounds[s]['Lower'] = -np.inf
            if 'Upper' not in self.dBounds[s]:
                self.dBounds[s]['Upper'] = np.inf
        for s in self.States + self.Disturbances:
            if s not in self.dBounds:
                self.dBounds[s] = {'Lower': -np.inf, 'Upper': np.inf}

    def __setupMHE(self, Y):
        model = ConcreteModel()
        model.estimated = self.Estimated
        model.disturbances = self.Disturbances
        model.m_mhe = Param(initialize=(self._ctr - 1) * self.Ts_ratio,
        ↪  mutable=True)
        model.statespace = {'A': self.A, 'B': self.B, 'C': self.C, 'D':
        ↪  self.D}
        model.weights = {'P': self.P, 'Q': self.Q, 'R': self.R, 'L': self.L}
        model.states_i = {s: i for i, s in enumerate(self.States)}
        model.states_est_i = {s: i for i, s in enumerate(self.Estimated)}
        model.estimated_i = {s: i for i, s in enumerate(self.Estimated +
        ↪  self.Disturbances)}
        model.inputs_i = {ii: i for i, ii in enumerate(self.Inputs)}
        model.disturbances_i = {ii: i for i, ii in
        ↪  enumerate(self.Disturbances)}
        model.outputs_i = {oo: i for i, oo in enumerate(self.Outputs)}
        model.disturbances_i = {ii: i for i,
                                ii in enumerate(self.Disturbances)}
        model.A = Param(self.States, self.States, initialize=self._defineA)
        if self.B.shape[1] != 0:
            model.B = Param(self.States, list(set(self.Inputs +
            ↪  self.Disturbances)), initialize=self._defineB)
        model.C = Param(self.Outputs, self.States, initialize=self._defineC)
        if self.D.shape[1] != 0:
            model.D = Param(self.Outputs, self.Inputs,
                            initialize=self._defineD)
        model.bounds = self.Bounds
        model.dbounds = self.dBounds
        model.horizon = RangeSet(0, model.m_mhe)
        # print(model.m_mhe.display())
        model.horizon_1 = RangeSet(0, model.m_mhe - 1)
        model.ihorizon = RangeSet(0, model.m_mhe)
        Ts_ddict = {}
        Ts_wdict = {}
        Ts_ydict = {}

        tmp = model.horizon_1.first()
        Winit = []
        Dinit = []
        Yinit = []
        for h in model.horizon_1:
            if np.mod((h), self.Ts_ratio) == 0:
                tmp = h
                Winit.append(tmp)
            Ts_wdict[h] = tmp
        tmp = model.horizon.first()
        for h in model.ihorizon:
            if np.mod((h), self.Ts_ratio) == 0:
                tmp = h
                Dinit.append(tmp)
            Ts_ddict[h] = tmp
        tmp = model.horizon.first()
        for h in model.horizon:
            if np.mod((h), self.Ts_ratio) == 0:
                tmp = h
                Yinit.append(tmp)
            Ts_ydict[h] = tmp

        self.Winit = Winit
        self.Dinit = Dinit
        self.Yinit = Yinit
        model.Xbounds = Param(
            self.States, ['Upper', 'Lower'], initialize=self._defineBounds, mutable=True)
        model.X = Var(self.States, model.horizon,
                      bounds=self._assignBounds, initialize=0.0)
        model.dXbounds = Param(
            self.States, ['Upper', 'Lower'], initialize=self._definedBounds, mutable=True)
        model.dX = Var(self.States, model.horizon_1,
                       bounds=self._assigndBounds, initialize=0.0)
        model.w = Var(self.Estimated, Winit, initialize=0.0)
        model.Ibounds = Param(self.Disturbances, [
                              'Upper', 'Lower'], initialize=self._defineBounds, mutable=True)
        model.ICons = Var(self.Disturbances, model.ihorizon,
                          bounds=self._assignBounds, initialize=0.0)
        model.dIbounds = Param(self.Disturbances, [
                               'Upper', 'Lower'], initialize=self._definedBounds, mutable=True)
        model.dICons = Var(self.Disturbances, model.ihorizon,
                           bounds=self._assigndBounds, initialize=0.0)
        model.d = Var(self.Disturbances, Dinit, initialize=0.0)
        model.Ymeas = Param(self.Outputs, Yinit,
                            initialize=0.0, mutable=True)
        model.Reset = Param(self.States, initialize=0, mutable=True)
        model.vL = Var(self.Outputs, Yinit, initialize=0.0)
        # Set initial conditions for estimated states and parameters
        model.x0 = Param(self.States, initialize=0.0, mutable=True)
        model.d0 = Param(self.Disturbances, initialize=0)
        model.X_1 = Param(self.States, initialize=0.0, mutable=True)
        model.X_1Constraints = ConstraintList()
        model._ICLOCK = ConstraintList()
        model.XE = Var(self.States, initialize=0.0)
        model.U = Param(self.Inputs, Dinit,
                        initialize=0.0, mutable=True)
        model.U_1 = Param(self.Inputs, initialize=0.0, mutable=True)
        model.D_1 = Param(self.Disturbances, initialize=0.0, mutable=True)
        model.P = Param(self.Estimated + self.Disturbances, self.Estimated + self.Disturbances,
                        initialize=self._defineP, mutable=True)
        model.Q = Param(self._estimated_all, self._estimated_all,
                        initialize=self._defineQ)
        model.R = Param(self.Outputs, self.Outputs,
                        initialize=self._defineR, mutable=True)
        if self.L.shape[1] != 0:
            model.L = Param(self.Disturbances, self.Disturbances,
                            initialize=self._defineL, mutable=True)

        def _Ic(model, d, i):
            if d in self.Inputs:
                U = model.U[d, Ts_ddict[i]]
            else:
                U = 0.0
            return model.ICons[d, i] == U + model.d[d, Ts_ddict[i]]
            # return model.ICons[d, i] == model.d[d, i]
        model._IC = Constraint(self.Disturbances, model.ihorizon, rule=_Ic)

        def _dIc(model, d, i):
            if i == 0:
                if d in self.Inputs:
                    U = model.U[d, Ts_ddict[i]]
                    U_1 = model.U_1[d]
                else:
                    U = 0.0
                    U_1 = 0.0
                return model.dICons[d, i] == (U + model.d[d, Ts_ddict[i]]) - (U_1 + model.D_1[d])
            else:
                if d in self.Inputs:
                    U = model.U[d, Ts_ddict[i]]
                    U_1 = model.U[d, Ts_ddict[i - 1]]
                else:
                    U = 0.0
                    U_1 = 0.0
                return model.dICons[d, i] == (U + model.d[d, Ts_ddict[i]]) - (U_1 + model.d[d, Ts_ddict[i - 1]])
        model._dIC = Constraint(self.Disturbances, model.ihorizon, rule=_dIc)

        # Calculate changes for ROC constraints
        def _dXC_f(model, s, i):
            return model.dX[s, i] == model.X[s, i + 1] - model.X[s, i]
        model.dXC = Constraint(self.States, model.horizon_1, rule=_dXC_f)

        def _stepSS(model, i, s):
            asum = sum([model.A[s, si] * model.X[si, i] for si in self.States])
            bsum = 0.0
            for ii in self.Inputs:
                bsum += model.B[s, ii] * model.U[ii, Ts_ddict[i]]
            for ii in self.Disturbances:
                bsum += model.B[s, ii] * model.d[ii, Ts_ddict[i]]
            if s in self.Estimated:
                wsum = model.w[s, Ts_wdict[i]]
            else:
                wsum = 0.0
            return model.X[s, i + 1] == asum + bsum + wsum
        model.SS_Constraint = Constraint(model.horizon_1, self.States, rule=_stepSS)

        def _outputConstraints(model, s):
            summer = 0
            for s2 in self.Outputs:
                summer += model.C[s2, s]
            if summer == 0 and s in self.Estimated:
                return model.w[s, model.horizon_1.last()] == 0
            else:
                return Constraint.Skip
        model.OutputConstraintsC = Constraint(self.States, rule=_outputConstraints)

        def _vL(model, i, o):
            cout = sum([model.C[o, si] * model.X[si, i] for si in self.States])
            if hasattr(model, 'D'):
                for ii in self.Inputs:
                    cout += model.D[o, ii] * model.U[ii, i]
                for ii in self.Disturbances:
                    cout += model.D[o, ii] * model.D[ii, i]
            return model.vL[o, i] == model.Ymeas[o, i] - cout
        model.vLC = Constraint(Yinit, self.Outputs, rule=_vL)

        def _NoFeedThroughConstraint(model, d):
            if not hasattr(model, 'D') and self._ctr > 1:
                return model.d[d, Ts_ddict[model.ihorizon.last()]] == model.d[d, Ts_ddict[model.ihorizon.last() - 1]]
            else:
                return Constraint.Skip
        model._NFTC = Constraint(self.Disturbances, rule=_NoFeedThroughConstraint)

        def _Obj(model):
            ArrivalCost = 0.
            for s1 in self.Estimated:
                for s2 in self.Estimated:
                    ArrivalCost += model.XE[s1] * \
                        model.P[s1, s2] * model.XE[s2]
            SSE = 0.
            for k in Yinit:
                for o1 in self.Outputs:
                    for o2 in self.Outputs:
                        SSE += model.vL[o1, k] * \
                            model.R[o1, o2] * model.vL[o2, k]
            NoiseCost = 0.
            for k in model.w_index_1:
                for s1 in self._estimated_all:
                    for s2 in self._estimated_all:
                        if s1 in self.States:
                            elem1 = model.w[s1, Ts_wdict[k]]
                        elif s1 in self.Disturbances:
                            elem1 = model.d[s1, Ts_ddict[k]]
                        if s2 in self.States:
                            elem2 = model.w[s2, Ts_wdict[k]]
                        elif s2 in self.Disturbances:
                            elem2 = model.d[s2, Ts_ddict[k]]
                        NoiseCost += elem1 * model.Q[s1, s2] * elem2
            if hasattr(model, 'D'):
                for s1 in self._estimated_all:
                    for s2 in self._estimated_all:
                        if s1 in self.Disturbances and s2 in self.Disturbances:
                            NoiseCost += model.d[s1, Ts_ddict[model.ihorizon.last()]] * model.Q[s1, s2] * model.d[
                                s2, Ts_ddict[model.ihorizon.last()]]

            deltaDCost = 0.
            if hasattr(model, 'L'):
                if self._Regularize_Disturbances == "All":
                    for k in Dinit:
                        for d1 in self.Disturbances:
                            for d2 in self.Disturbances:
                                if k > 0:
                                    deltaDCost += (model.d[d1, Ts_ddict[k]] - model.d[d1, Ts_ddict[k - 1]]) * model.L[
                                        d1, d2] * (model.d[d2, Ts_ddict[k]] - model.d[d2, Ts_ddict[k - 1]])
                                else:
                                    print(k)
                                    deltaDCost += (model.d[d1, Ts_ddict[k]] - model.D_1[d1]) * model.P[
                                        d1, d2] * (model.d[d2, Ts_ddict[k]] - model.D_1[d2])
                elif self._Regularize_Disturbances == "First":
                    deltaDCost += (model.d[d1, Ts_ddict[k]] - model.D_1[d1]) * model.P[
                        d1, d2] * (model.d[d2, Ts_ddict[k]] - model.D_1[d2])
                elif self._Regularize_Disturbances is None:
                    deltaDCost += 0
            return ArrivalCost + SSE + NoiseCost + deltaDCost

        model._obj = Objective(rule=_Obj)
        self._pyomoMHE = model
        self.__opt = SolverFactory(self.solver)
        self.Ts_ddict = Ts_ddict
        self.Ts_wdict = Ts_wdict
        self.Ts_ydict = Ts_ydict

    def MHEUpdate(self, Y, U=None, x0=None, d0=None, returnfull=False, reset=[], **kwargs):
        self.Reset = reset.copy()
        if hasattr(self, '_x0'):
            del self._x0
        if self.B.shape[1] == 0 and d0 is not None:
            raise _Update_Error(
                'Initial disturbance supplied in the absence of a B matrix')
        elif not hasattr(self, 'Disturbances') and d0 is not None:
            raise _Update_Error(
                'Initial disturbance supplied in the absence of estimated disturbances')
        elif d0 is None:
            d0 = np.zeros(len(self.Disturbances))
        else:
            d0 = np.array(d0)
        if U is None:
            U = np.zeros(self.B.shape[1])
        if (np.isnan(Y).any()):
            raise _Update_Error('NaN detected in meaurement!')
        if (np.isnan(U).any()):
            raise _Update_Error('NaN detected in input!')
        Y = np.matrix(Y)
        U = np.matrix(U)
        if Y.shape[1] != U.shape[1]:
            raise _Update_Error(
                ('Y and U do not contain the same number of values'))
        if Y.shape[0] != self.C.shape[0]:
            raise _Update_Error(
                ('Dimensions of measurement do not match State Space'))
        if not hasattr(self, 'y'):
            self.y = Y
        else:
            self.y = np.hstack((self.y, Y))
        if U is not None and self.B.shape[1] == 0:
            raise _Update_Error(
                'An input is provided in update, but no B matrix is
                ↪ specified')
        else:
            if U.shape[0] != len(self.Inputs):
                raise _Update_Error(
                    'Dimensions of input do not match State Space')
        if not hasattr(self, 'u'):
            self.u = U
        else:
            # pdb.set_trace()
            self.u = np.hstack((self.u, U))
        self._ctr = self.y.shape[1]

        if self._ctr * self.Ts_ratio <= float(self.H * self.Ts_ratio) or
        ↪ not hasattr(self, '_pyomoMHE'):
            self._x0 = {}
            if x0 is None:
                x0 = {}
                for s in self.States:
                    if s not in self.Estimate_IC:
                        x0[s] = 0.0
            else:
                self._x0 = x0
            for s in self.States:
                if s in self.Estimate_IC:
                    self.Reset.append(s)
                else:
                    self._x0[s] = x0[s]
            _m = self._ctr
            if hasattr(self, '_pyomoMHE'):
                delattr(self, '_pyomoMHE')
            self.__setupMHE(Y)
            self._pyomoMHE.m_mhe = np.array([[self._ctr * self.Ts_ratio]])
            for d in self.Disturbances:
                didx = self._pyomoMHE.disturbances_i[d]
                self._pyomoMHE.D_1[d] = float(d0[didx])
            # self._x0 = {}
            for s in self.States:
                self._pyomoMHE.X_1[s] = 0.0

        else:
            self._pyomoMHE.m_mhe = self.H * self.Ts_ratio
            _m = int(self.H)
            # FORCE INITIAL CONDITONS:
            if x0 is None:
                self._x0 = {}
            else:
                for s in x0.keys():
                    self._x0[s] = x0[s]
            # print(reset)
            self.Reset = reset.copy()
        del self._pyomoMHE._ICLOCK
        del self._pyomoMHE._ICLOCK_index
        self._pyomoMHE._ICLOCK = ConstraintList()
        asum = {}
        for sk in self.States:
            asum[sk] = self._pyomoMHE.X_1[sk]()
        for k in range(0, self.Ts_ratio):
            asum2 = {}
            for sj in self.States:
                bsum = 0.0
                for ii in self.Inputs:
                    bsum += self._pyomoMHE.B[sj, ii] *
                    ↪  self._pyomoMHE.U_1[ii]()
                    # print(bsum)
                for ii in self.Disturbances:
                    bsum += self._pyomoMHE.B[sj, ii] *
                    ↪  self._pyomoMHE.D_1[ii]()
                asum2[sj] = sum([self._pyomoMHE.A[sj, sk] * asum[sk] for sk
                ↪  in self.States]) + bsum
            for sk in self.States:
                asum[sk] = asum2[sk]
        for s in self.States:
            if s not in self.Reset and s not in self._x0 and s not in
            ↪  self.Estimated:
                self._pyomoMHE._ICLOCK.add(expr=(self._pyomoMHE.X[s, 0] ==
                ↪  asum[s]))
            elif s in self._x0:
                self._pyomoMHE._ICLOCK.add(expr=(self._pyomoMHE.X[s, 0] ==
                ↪  self._x0[s]))
            elif s in self.Estimated and s not in self.Reset:
                self._pyomoMHE._ICLOCK.add(expr=(self._pyomoMHE.XE[s] ==
                ↪  self._pyomoMHE.X[s, 0] - asum[s]))
        for i in range(1, _m + 1):
            for o in self.Outputs:
                yidx = self._pyomoMHE.outputs_i[o]
                if i == 1:
                    self._pyomoMHE.Ymeas[o, 0] = float(self.y[yidx, - _m -
                    ↪  1 + i])
                else:
                    self._pyomoMHE.Ymeas[o, (i - 1) * self.Ts_ratio] =
                    ↪  float(self.y[yidx, -_m - 1 + i])

        for i in range(1, _m + 1):
            for inp in self.Inputs:
                uidx = self._pyomoMHE.inputs_i[inp]
                if i == 1:
                    self._pyomoMHE.U[inp, 0] = float(self.u[uidx, - _m - 1
                    ↪  + i])
                else:
                    self._pyomoMHE.U[inp, (i - 1) * self.Ts_ratio] =
                    ↪  float(self.u[uidx, - _m - 1 + i])
        if self.P.shape[1] != 0:
            self.P = self.P
        for s1 in self.Estimated:
            for s2 in self.Estimated:
                i = self._pyomoMHE.states_est_i[s1]
                j = self._pyomoMHE.states_est_i[s2]
                self._pyomoMHE.P[s1, s2] = self.P[i, j]
        self._pyomoMHE.preprocess()
        soln = self.__opt.solve(self._pyomoMHE, keepfiles=False, tee=False)
        self._pyomoMHE.solutions.load_from(soln)
        for s in self.States:
            self._pyomoMHE.X_1[s] = float(self._pyomoMHE.X[s, 0]())
        for d in self.Disturbances:
            self._pyomoMHE.D_1[d] = self._pyomoMHE.d[d, 0]()
        for u in self.Inputs:
            self._pyomoMHE.U_1[u] = self._pyomoMHE.U[u, 0]()
        xout = np.matrix(np.zeros((self.A.shape[0], 1)))
        xout2 = np.matrix(np.zeros((self.A.shape[0], 1)))
        wout = np.matrix(np.zeros((len(self.Estimated), 1)))
        for s in self.States:
            i = self._pyomoMHE.states_i[s]
            xout[i, 0] = self._pyomoMHE.X[s, self._pyomoMHE.horizon.last()]()
            xout2[i, 0] = self._pyomoMHE.X[s, self._pyomoMHE.horizon.first()]()
        if len(self.Estimated) > 0:
            if len(self._pyomoMHE.w) > 0:
                for s in self.Estimated:
                    i = self._pyomoMHE.states_est_i[s]
                    wout[i, 0] = self._pyomoMHE.w[
                        s, self._pyomoMHE.horizon_1.first()]()
            if not hasattr(self, 'w'):
                self.w = wout
                self.w = np.array(self.w)
            else:
                self.w = np.hstack((self.w, wout))
                self.w = np.array(self.w)
        if not hasattr(self, 'x'):
            self.x = xout
            self.x = np.array(self.x)
        else:
            self.x = np.hstack((self.x, xout))
            self.x = np.array(self.x)
        if len(self.Disturbances) > 0:
            dout = np.array(np.zeros(len(self.Disturbances)))
            if self._pyomoMHE.horizon.last() > 0:
                for d in self.Disturbances:
                    didx = self._pyomoMHE.disturbances_i[d]
                    dout[didx] = (self._pyomoMHE.d[
                                  d, self.Ts_ddict[self._pyomoMHE.horizon.last()]]())
                if not hasattr(self, 'd'):
                    self.d = dout
                    self.d = np.array(self.d)
                else:
                    self.d = np.hstack((self.d, dout))
                    self.d = np.array(self.d)
            else:
                tmplist = []
                for d in self.Disturbances:
                    tmplist.append(self._pyomoMHE.d0[d])
                dout = np.array(tmplist)
            return xout, dout
        else:
            return xout, xout2

    def updateBounds(self, e, Lower=None, Upper=None):
        if e not in self._estimated:
            KeyError(e + ' not found in estimated objects')
        if Lower is not None:
            if (not isinstance(Lower, Number) or Lower != '-inf'):
                KeyError('Must enter an integer or',
                         ' float for lower bound!')
            self.Bounds[e]['Lower'] = float(Lower)
            if hasattr(self, '_MHE__MHEInstance'):
                if e in self.States:
                    self._mHEInstance._Xbounds[e, 'Lower'] = float(Lower)
                else:
                    self._mHEInstance._Dbounds[e, 'Lower'] = float(Lower)
        if Upper is not None:
            if (not isinstance(Upper, Number) or Upper != 'inf'):
                KeyError('Must enter an integer or',
                         ' float for upper bound!')
            self.Bounds[e]['Upper'] = Upper
            if hasattr(self, '_MHE__MHEInstance'):
                if e in self.States:
                    self._mHEInstance._Xbounds[e, 'Upper'] = float(Upper)
                else:
                    self._mHEInstance._Dbounds[e, 'Upper'] = float(Upper)

    def updateROCBounds(self, e, Lower=None, Upper=None):
        if e not in self.States:
            KeyError(e + ' not found in defined states')
        if Lower is not None:
            if (not isinstance(Lower, Number) or Lower != '-inf'):
                KeyError('Must enter an integer or',
                         ' float for lower bound!')
            self.dBounds[e]['Lower'] = Lower
            if hasattr(self, '_MHE__MHEInstance'):
                if e in self.States:
                    self._mHEInstance._dXbounds[e, 'Lower'] = float(Lower)
                else:
                    self._mHEInstance._dDbounds[e, 'Lower'] = float(Lower)
        if Upper is not None:
            if (not isinstance(Upper, Number) or Upper != 'inf'):
                KeyError('Must enter an integer or',
                         ' float for upper bound!')
            self.dBounds[e]['Upper'] = Upper
            if hasattr(self, '_MHE__MHEInstance'):
                if e in self.States:
                    self._mHEInstance._dXbounds[e, 'Upper'] = float(Upper)
                else:
                    self._mHEInstance._dDbounds[e, 'Upper'] = float(Upper)

    @staticmethod
    def _defineA(model, s1, s2):
        i = model.states_i[s1]
        j = model.states_i[s2]
        return model.statespace['A'][i, j]

    @staticmethod
    def _defineB(model, s, inp):
        i = model.states_i[s]
        if inp in model.inputs_i:
            j = model.inputs_i[inp]
        elif inp in model.disturbances_i and inp not in model.inputs_i:
            j = model.disturbances_i[inp] + len(model.inputs_i)
        return model.statespace['B'][i, j]

    @staticmethod
    def _defineC(model, o, s):
        i = model.outputs_i[o]
        j = model.states_i[s]
        return model.statespace['C'][i, j]

    @staticmethod
    def _defineD(model, o, i):
        i = model.outputs_i[o]
        j = model.inputs_i[i]
        return model.statespace['D'][i, j]

    @staticmethod
    def _defineL(model, d1, d2):
```

```python
        i = model.disturbances_i[d1]
        j = model.disturbances_i[d2]
        return model.weights['L'][i, j]

    @staticmethod
    def _defineP(model, e1, e2):
        i = model.estimated_i[e1]
        j = model.estimated_i[e2]
        return model.weights['P'][i, j]

    @staticmethod
    def _defineR(model, o1, o2):
        i = model.outputs_i[o1]
        j = model.outputs_i[o2]
        return model.weights['R'][i, j]

    @staticmethod
    def _defineQ(model, s1, s2):
        if s1 in model.states_est_i.keys():
            i = model.states_est_i[s1]
        elif s1 in model.disturbances_i.keys():
            i = model.disturbances_i[s1] + len(model.states_est_i.keys())
        if s2 in model.states_est_i.keys():
            j = model.states_est_i[s2]
        elif s2 in model.disturbances_i.keys():
            j = model.disturbances_i[s2] + len(model.states_est_i.keys())
        return model.weights['Q'][i, j]

    @staticmethod
    def _defineXprev(model, s):
        # print(model.x0[s])
        return float(model.x0[s])

    @staticmethod
    def _defineU(model, inp, k):
        i = model.inputs_i[inp]
        return model.u[i, -model.m_mhe - 2 + k]

    @staticmethod
    def _defineUprev(model, d):
        return float(model.u0[d])

    @staticmethod
    def _defineDprev(model, d):
        return float(model.d0[d])

    @staticmethod
    def _defineBounds(model, e, ul):
        return model.bounds[e][ul]

    @staticmethod
    def _assignBounds(model, e, i):
        if e in model.states_i.keys():
            return (model.Xbounds[e, 'Lower'], model.Xbounds[e, 'Upper'])
        else:
            return (model.Ibounds[e, 'Lower'], model.Ibounds[e, 'Upper'])

    @staticmethod
    def _definedBounds(model, e, ul):
        return model.dbounds[e][ul]

    @staticmethod
    def _assigndBounds(model, e, i):
        if e in model.states_i.keys():
            return (model.dXbounds[e, 'Lower'], model.dXbounds[e, 'Upper'])
        else:
            return (model.dIbounds[e, 'Lower'], model.dIbounds[e, 'Upper'])

    @staticmethod
    def _kalmanUpdateP(mhe):
        # A, C, P, Q, R
        A = mhe.A
        C = mhe.C
        P = mhe.P
        Q = mhe.Q
        R = mhe.R
        ii = []
        for e in mhe.Estimated:
            i = mhe._pyomoMHE.states_est_i[e]
            ii.append(i)
        Cin = np.empty((C.shape[0], len(mhe.Estimated)))
        Ain = np.empty((len(mhe.Estimated), len(mhe.Estimated)))
        Qin = np.empty((len(mhe.Estimated), len(mhe.Estimated)))
        for iidx, i in enumerate(ii):
            Cin[0, i] = C[0, i]
            for j in enumerate(ii):
                Ain[i, j] = A[i, j]
                Qin[i, j] = Q[i, j]
        C = Cin
        A = Ain
        Q = Qin
        Pkmkminv = P + C.T * inv(R) * C
        P = inv(Q) - inv(Q) * A * inv(Pkmkminv +
                                      A.T * inv(Q) * A) * A.T * inv(Q)
        return P
```

APPENDIX G

CGM DETECTION OF A PRESSURE INDUCED LOSS OF SENSITIVITY

G.1 DETECTION

A PILS is characterized by a rate of decrease in glucose levels that violate physiologic limits [308]. Following an approach similar to that of Baysal et al.[308] at each new CGM measurement the PILS detection algorithm calculates the rate of change in the CGM signal $g' = \frac{CGM_k - CGM_{k-1}}{t_k - t_{k-1}}$. The algorithm then checks the following:

$$\begin{array}{rl} \text{i} & k \geq 3 \\ \text{ii} & g' < g'_c \\ \text{iii} & \frac{g'_k}{g'_{k-1}} > g'_{ratio,c} \end{array}$$

These three checks ensure that (i) at least three CGM measurements are available, (ii) the rate of glucose decrease as measured by CGM is greater than some critical physiologic value and (iii), the relative change in magnitude of the rate of decrease between measurements is increasing above some critical threshold. When all three criterion are met a pressure-induced loss of sensitivity alarm is signaled. The criterion above are used to identify the start of a PILS event, however it is also necessary to determine when or if the PILS event has ended and the CGM has returned to normal functionality. Following the start of a PILS event, at each new reading from the faulty CGM the algorithm checks for:

$$\begin{array}{rl} \text{i} & g' > g'_c \\ \text{ii} & CGM_{k,est} \leq CGM_k \\ \text{iii} & g'_{c,ratio} \leq \frac{g'_{k-i}}{g'_{k-1-i}} \quad [i = 0 ... N_{PILS}] \end{array}$$

G.2 REMEDIATION

In the event of a detected pressure induced loss of sensitivity the moving horizon estimator reverts to a single observer mode where the CGM not influenced by the PILS event is used in the MHE observation vector in place of the CGM affected by pressure. Direct replacement of the pressure-affected CGM in the MHE observation vector may lead to a jump in glucose estimations and a corresponding "jump" in some combination of insulin sensitivity, endogenous insulin production rate and endogenous glucose production rate. To prevent such a discontinuity, a constant offest is added to the replacement CGM value such that the deviation between the CGM signals before the pressure induced loss of sensitivity is maintained. To signal the end of a PILS event the CGM measured glucose rate of change needs to return to a physiologic regime (i). Additionally, the most recent CGM measured glucose value must be greater than or equal to the glucose level estimated by the MHE using the remaining functioning sensor (ii) ***or*** the relative magnitude of the rate of glucose change between subsequent measurements must greater than some threshold value ($g'_{c,ratio}$) for a certain number of measurements (N_{PILS}) (iii). If the PILS is not resolved within the allotted time following the loss of a single sensor infusions are stopped and an alarm is raised.

BIBLIOGRAPHY

[1] K. M. Dungan, S. S. Braithwaite, and J.-C. Preiser, “Stress hyperglycaemia,” vol. 373, no. 9677, pp. 1798–1807.

[2] K. C. McCowen, A. Malhotra, and B. R. Bistrian, “Stress-Induced Hyperglycemia,” vol. 17, no. 1, pp. 107–124.

[3] L. Corsino, K. Dhatariya, and G. Umpierrez, “Management of Diabetes and Hyperglycemia in Hospitalized Patients,” in *Endotext* (L. J. De Groot, G. Chrousos, K. Dungan, K. R. Feingold, A. Grossman, J. M. Hershman, C. Koch, M. Korbonits, R. McLachlan, M. New, J. Purnell, R. Rebar, F. Singer, and A. Vinik, eds.), MD-Text.com, Inc.

[4] P. E. Marik and R. Bellomo, “Stress hyperglycemia: An essential survival response!,” vol. 17, no. 2, p. 305.

[5] “SCCM — Critical Care Statistics.”

[6] M. Egi, R. Bellomo, E. Stachowski, C. J. French, G. K. Hart, C. Hegarty, and M. Bailey, “Blood glucose concentration and outcome of critical illness: The impact of diabetes*:,” vol. 36, no. 8, pp. 2249–2255.

[7] G. E. Umpierrez, S. D. Isaacs, N. Bazargan, X. You, L. M. Thaler, and A. E. Kitabchi, “Hyperglycemia: An Independent Marker of In-Hospital Mortality in Patients with Undiagnosed Diabetes,” vol. 87, no. 3, pp. 978–982.

[8] M. Falciglia, R. W. Freyberg, P. L. Almenoff, D. A. D’Alessio, and M. L. Render, “Hyperglycemia-related mortality in critically ill patients varies with admission diagnosis,” vol. 37, no. 12, pp. 3001–3009.

[9] A. P. Furnary, G. Gao, G. L. Grunkemeier, Y. Wu, K. J. Zerr, S. O. Bookin, H. S. Floten, and A. Starr, “Continuous insulin infusion reduces mortality in patients with diabetes undergoing coronary artery bypass grafting,” vol. 125, no. 5, pp. 1007–1021.

[10] J. S. Krinsley, “Association between hyperglycemia and increased hospital mortality in a heterogeneous population of critically ill patients,” vol. 78, no. 12, pp. 1471–1478.

[11] G. V. M. Bochicchio, M. Joshi, K. M. R. Bochicchio, A. R. Pyle, S. B. M. Johnson, W. Meyer, K. Lumpkins, and T. M. M. Scalea, "Early hyperglycemic control is important in critically injured trauma patients," vol. 63, no. 6, pp. 1353–1359.

[12] B. P. Kavanagh and K. C. McCowen, "Glycemic Control in the ICU," vol. 363, no. 26, pp. 2540–2546.

[13] M. I. Worthley, F. M. Shrive, T. J. Anderson, and M. Traboulsi, "Prognostic Implication of Hyperglycemia in Myocardial Infarction and Primary Angioplasty," vol. 120, no. 7, pp. 643.e1–643.e7.

[14] R. Fogelholm, K. Murros, A. Rissanen, and S. Avikainen, "Admission blood glucose and short term survival in primary intracerebral haemorrhage: A population based study," vol. 76, no. 3, pp. 349–353.

[15] P. L. Bosarge and J. D. Kerby, "Stress-induced Hyperglycemia," vol. 47, no. 1, pp. 287–297.

[16] J. Hermanides, T. M. Vriesendorp, R. J. Bosman, D. F. Zandstra, J. B. Hoekstra, and J. H. DeVries, "Glucose variability is associated with intensive care unit mortality*:," vol. 38, no. 3, pp. 838–842.

[17] M. Egi and R. Bellomo, "Reducing Glycemic Variability in Intensive Care Unit Patients: A New Therapeutic Target?," vol. 3, no. 6, pp. 1302–1308.

[18] A. Ouattara, A. Grimaldi, and B. Riou, "Blood Glucose VariabilityA New Paradigm in Critical Care?," vol. 105, no. 2, pp. 233–234.

[19] N. A. Ali, J. S. Krinsley, and J.-C. Preiser, "Glucose Variability in Critically III Patients," in *Yearbook of Intensive Care and Emergency Medicine* (P. J.-L. Vincent, ed.), no. 2009 in Yearbook of Intensive Care and Emergency Medicine, pp. 728–737, Springer Berlin Heidelberg.

[20] G. Van den Berghe, P. Wouters, F. Weekers, C. Verwaest, F. Bruyninckx, M. Schetz, D. Vlasselaers, P. Ferdinande, P. Lauwers, and R. Bouillon, "Intensive Insulin Therapy in Critically Ill Patients," vol. 345, no. 19, pp. 1359–1367.

[21] J.-C. Preiser, P. Devos, and G. Van den Berghe, "Tight control of glycaemia in critically ill patients," vol. 5, no. 5, pp. 533–537.

[22] J. S. Krinsley and A. Grover, "Severe hypoglycemia in critically ill patients: Risk factors and outcomes," vol. 35, no. 10, pp. 2262–2267.

[23] G. Van den Berghe, A. Wilmer, G. Hermans, W. Meersseman, P. J. Wouters, I. Milants, E. Van Wijngaerden, H. Bobbaers, and R. Bouillon, "Intensive Insulin Therapy in the Medical ICU," vol. 354, no. 5, pp. 449–461.

[24] J.-C. Preiser, P. Devos, S. Ruiz-Santana, C. Mlot, D. Annane, J. Groeneveld, G. Iapichino, X. Leverve, G. Nitenberg, P. Singer, J. Wernerman, M. Joannidis, A. Stecher, and R. Chiolro, “A prospective randomised multi-centre controlled trial on tight glucose control by intensive insulin therapy in adult intensive care units: The Glucontrol study,” vol. 35, no. 10, pp. 1738–1748.

[25] S. R. Mehta, S. Yusuf, R. Daz, J. Zhu, P. Pais, D. Xavier, E. Paolasso, R. Ahmed, C. Xie, K. Kazmi, J. Tai, A. Orlandini, J. Pogue, L. Liu, and CREATE-ECLA Trial Group Investigators, “Effect of glucose-insulin-potassium infusion on mortality in patients with acute ST-segment elevation myocardial infarction: The CREATE-ECLA randomized controlled trial,” vol. 293, no. 4, pp. 437–446.

[26] T. N.-S. S. Investigators, “Intensive versus Conventional Glucose Control in Critically Ill Patients,” vol. 360, no. 13, pp. 1283–1297.

[27] T. N.-S. S. Investigators, “Hypoglycemia and Risk of Death in Critically Ill Patients,” vol. 367, no. 12, pp. 1108–1118.

[28] J. Hermanides, R. J. Bosman, T. M. Vriesendorp, R. Dotsch, F. R. Rosendaal, D. F. Zandstra, J. B. L. Hoekstra, and J. H. DeVries, “Hypoglycemia is associated with intensive care unit mortality,” vol. 38, no. 6, pp. 1430–1434.

[29] F. M. Brunkhorst, C. Engel, F. Bloos, A. Meier-Hellmann, M. Ragaller, N. Weiler, O. Moerer, M. Gruendling, M. Oppert, S. Grond, D. Olthoff, U. Jaschinski, S. John, R. Rossaint, T. Welte, M. Schaefer, P. Kern, E. Kuhnt, M. Kiehntopf, C. Hartog, C. Natanson, M. Loeffler, and K. Reinhart, “Intensive Insulin Therapy and Pentastarch Resuscitation in Severe Sepsis,” vol. 358, no. 2, pp. 125–139.

[30] M. Wilson, J. Weinreb, and G. W. S. Hoo, “Intensive Insulin Therapy in Critical Care,” vol. 30, no. 4, pp. 1005–1011.

[31] R. Tiruvoipati, B. Chiezey, D. Lewis, K. Ong, E. Villanueva, K. Haji, and J. Botha, “Stress hyperglycemia may not be harmful in critically ill patients with sepsis,” vol. 27, no. 2, pp. 153–158.

[32] H. Hirasawa, S. Oda, and M. Nakamura, “Blood glucose control in patients with severe sepsis and septic shock,” vol. 15, no. 33, pp. 4132–4136.

[33] B. W. Whitcomb, E. K. Pradhan, A. G. Pittas, M.-C. Roghmann, and E. N. Perencevich, “Impact of admission hyperglycemia on hospital mortality in various intensive care unit populations*:,” vol. 33, no. 12, pp. 2772–2777.

[34] P. E. Marik and J.-C. Preiser, “Toward understanding tight glycemic control in the ICU: A systematic review and metaanalysis,” vol. 137, no. 3, pp. 544–551.

[35] P. Parsons and P. Watkinson, “Blood glucose control in critical care patients a review of the literature,” vol. 12, no. 4, pp. 202–210.

[36] K. A. Malmberg, S. Efendic, L. E. Rydn, and F. T. M. S. Group, “Feasibility of Insulin-Glucose Infusion in Diabetic Patients With Acute Myocardial Infarction: A report from the multicenter trial: DIGAMI,” vol. 17, no. 9, pp. 1007–1014.

[37] G. Y. Gandhi, G. A. Nuttall, M. D. Abel, C. J. Mullany, H. V. Schaff, P. C. O’Brien, M. G. Johnson, A. R. Williams, S. M. Cutshall, L. M. Mundy, R. A. Rizza, and M. M. McMahon, “Intensive intraoperative insulin therapy versus conventional glucose management during cardiac surgery: A randomized trial,” vol. 146, no. 4, pp. 233–243.

[38] N. A. Ali, J. M. OBrien, K. Dungan, G. Phillips, C. B. Marsh, S. Lemeshow, A. F. Connors, and J.-C. Preiser, “Glucose variability and mortality in patients with sepsis,” vol. 36, no. 8, pp. 2316–2321.

[39] G. D. C. De La Rosa, J. H. Donado, A. H. Restrepo, A. M. Quintero, L. G. Gonzlez, N. E. Saldarriaga, M. Bedoya, J. M. Toro, J. B. Velsquez, J. C. Valencia, C. M. Arango, P. H. Aleman, E. M. Vasquez, J. C. Chavarriaga, A. Yepes, W. Pulido, and C. A. Cadavid, “Strict glycaemic control in patients hospitalised in a mixed medical and surgical intensive care unit: A randomised clinical trial,” vol. 12, no. 5, p. R120.

[40] P. C. Davidson, R. D. Steed, and B. W. Bode, “Glucommander,” vol. 28, no. 10, pp. 2418–2423.

[41] R. Cinotti, C. Ichai, J.-C. Orban, P. Kalfon, F. Feuillet, A. Roquilly, B. Riou, Y. Blanloeil, K. Asehnoune, and B. Rozec, “Effects of tight computerized glucose control on neurological outcome in severely brain injured patients: A multicenter sub-group analysis of the randomized-controlled open-label CGAO-REA study,” vol. 18, p. 498.

[42] T. Okabayashi, Y. Shima, T. Sumiyoshi, A. Kozuki, T. Tokumaru, T. Iiyama, T. Sugimoto, M. Kobayashi, M. Yokoyama, and K. Hanazaki, “Intensive Versus Intermediate Glucose Control in Surgical Intensive Care Unit Patients,” vol. 37, no. 6, pp. 1516–1524.

[43] J. I. Mechanick, “Metabolic Mechanisms of Stress Hyperglycemia,” vol. 30, no. 2, pp. 157–163.

[44] A. Herschkovitz, Y.-F. Liu, E. Ilan, D. Ronen, S. Boura-Halfon, and Y. Zick, “Common Inhibitory Serine Sites Phosphorylated by IRS-1 Kinases, Triggered by Insulin and Inducers of Insulin Resistance,” vol. 282, no. 25, pp. 18018–18027.

[45] L. Li and J. L. Messina, “Acute Insulin Resistance Following Injury,” vol. 20, no. 9, pp. 429–435.

[46] K. C. McCowen, P. R. Ling, A. Ciccarone, Y. Mao, J. C. Chow, B. R. Bistrian, and R. J. Smith, “Sustained endotoxemia leads to marked down-regulation of early steps in the insulin-signaling cascade,” vol. 29, no. 4, pp. 839–846.

[47] P. E. Cryer, “Hypoglycemia, functional brain failure, and brain death,” vol. 117, no. 4, pp. 868–870.

[48] G. Boden, “Gluconeogenesis and glycogenolysis in health and diabetes,” vol. 52, no. 6, pp. 375–378.

[49] M. E. McDonnell and G. E. Umpierrez, “Insulin Therapy for the Management of Hyperglycemia in Hospitalized Patients,” vol. 41, no. 1, pp. 175–201.

[50] J.-C. Preiser, J. G. Chase, R. Hovorka, J. I. Joseph, J. S. Krinsley, C. D. Block, T. Desaive, L. Foubert, P. Kalfon, U. Pielmeier, T. V. Herpe, and J. Wernerman, “Glucose Control in the ICU A Continuing Story,” p. 1932296816648713.

[51] A. Evans, A. Le Compte, C.-S. Tan, L. Ward, J. Steel, C. G. Pretty, S. Penning, F. Suhaimi, G. M. Shaw, T. Desaive, and J. G. Chase, “Stochastic Targeted (STAR) Glycemic Control: Design, Safety, and Performance,” vol. 6, no. 1, pp. 102–115.

[52] T. Van Herpe, D. Mesotten, P. J. Wouters, J. Herbots, E. Voets, J. Buyens, B. De Moor, and G. Van den Berghe, “LOGIC-Insulin AlgorithmGuided Versus Nurse-Directed Blood Glucose Control During Critical Illness,” vol. 36, no. 2, pp. 188–194.

[53] C. Pachler, J. Plank, H. Weinhandl, L. J. Chassin, M. E. Wilinska, R. Kulnik, P. Kaufmann, K.-H. Smolle, E. Pilger, T. R. Pieber, M. Ellmerer, and R. Hovorka, “Tight glycaemic control by an automated algorithm with time-variant sampling in medical ICU patients,” vol. 34, no. 7, pp. 1224–1230.

[54] J. B. Lee, E. Dassau, R. Gondhalekar, D. E. Seborg, J. E. Pinsker, and F. J. Doyle, “Enhanced Model Predictive Control (eMPC) Strategy for Automated Glucose Control,” vol. 55, no. 46, pp. 11857–11868.

[55] M. Vogelzang, F. Zijlstra, and M. W. Nijsten, “Design and implementation of GRIP: A computerized glucose control system at a surgical intensive care unit,” vol. 5, p. 38.

[56] U. Pielmeier, S. Andreassen, B. Juliussen, J. G. Chase, B. S. Nielsen, and P. Haure, “The Glucosafe system for tight glycemic control in critical care: A pilot evaluation study,” vol. 25, no. 1, pp. 97–104.

[57] P. Kalfon, B. Giraudeau, C. Ichai, A. Guerrini, N. Brechot, R. Cinotti, P.-F. Dequin, B. Riu-Poulenc, P. Montravers, D. Annane, H. Dupont, M. Sorine, B. Riou, and O. b. o. t. C.-R. S. Group, “Tight computerized versus conventional glucose control in the ICU: A randomized controlled trial,” vol. 40, no. 2, pp. 171–181.

[58] E. Rood, R. J. Bosman, J. I. van der Spoel, P. Taylor, and D. F. Zandstra, “Use of a Computerized Guideline for Glucose Regulation in the Intensive Care Unit Improved Both Guideline Adherence and Glucose Regulation,” vol. 12, no. 2, pp. 172–180.

[59] D. Aragon, “Evaluation of Nursing Work Effort and Perceptions About Blood Glucose Testing in Tight Glycemic Control,” vol. 15, no. 4, pp. 370–377.

[60] R. T. M. van Hooijdonk, L. M. G. Steuten, M. M. A. Kip, H. Monteban, M. R. Mulder, F. v. B. Houckgeest, J. P. van der Sluijs, A. Abu-Hanna, P. E. Spronk, and M. J.

Schultz, “Health Economic Evaluation of a Strict Glucose Control Guideline Implemented Using Point-of-Care Testing in Three Intensive Care Units in The Netherlands,” vol. 13, no. 4, pp. 399–407.

[61] U. Holzinger, J. Warszawska, R. Kitzberger, M. Wewalka, W. Miehsler, H. Herkner, and C. Madl, “Real-Time Continuous Glucose Monitoring in Critically Ill Patients,” vol. 33, no. 3, pp. 467–472.

[62] K. Hanazaki, H. Maeda, and T. Okabayashi, “Tight perioperative glycemic control using an artificial endocrine pancreas,” vol. 40, no. 1, pp. 1–7.

[63] K. Hanazaki, H. Kitagawa, T. Yatabe, M. Munekage, K. Dabanaka, Y. Takezaki, Y. Tsukamoto, T. Asano, Y. Kinoshita, and T. Namikawa, “Perioperative intensive insulin therapy using an artificial endocrine pancreas with closed-loop glycemic control system: The effects of no hypoglycemia,” vol. 207, no. 6, pp. 935–941.

[64] R. Hovorka, J. Kremen, J. Blaha, M. Matias, K. Anderlova, L. Bosanska, T. Roubicek, M. E. Wilinska, L. J. Chassin, S. Svacina, and M. Haluzik, “Blood Glucose Control by a Model Predictive Control Algorithm with Variable Sampling Rate Versus a Routine Glucose Management Protocol in Cardiac Surgery Patients: A Randomized Controlled Trial,” vol. 92, no. 8, pp. 2960–2964.

[65] J. J. Cordingley, D. Vlasselaers, N. C. Dormand, P. J. Wouters, S. D. Squire, L. J. Chassin, M. E. Wilinska, C. J. Morgan, R. Hovorka, and G. V. den Berghe, “Intensive insulin therapy: Enhanced Model Predictive Control algorithm versus standard care,” vol. 35, no. 1, pp. 123–128.

[66] J. G. Chase, G. Shaw, A. L. Compte, T. Lonergan, M. Willacy, X.-W. Wong, J. Lin, T. Lotz, D. Lee, and C. Hann, “Implementation and evaluation of the SPRINT protocol for tight glycaemic control in critically ill patients: A clinical practice change,” vol. 12, no. 2, p. R49.

[67] Merrer J, De Jonghe B, Golliot F, and et al, “Complications of femoral and subclavian venous catheterization in critically ill patients: A randomized controlled trial,” vol. 286, no. 6, pp. 700–707.

[68] S. Vaddiraju, D. J. Burgess, I. Tomazos, F. C. Jain, and F. Papadimitrakopoulos, “Technologies for Continuous Glucose Monitoring: Current Problems and Future Promises,” vol. 4, no. 6, pp. 1540–1562.

[69] R. P. Radermecker, A. Sultan, C. Piot, A. S. Remy, A. Avignon, and E. Renard, “Continuous glucose monitoring as a tool to identify hyperglycaemia in non-diabetic patients with acute coronary syndromes,” vol. 26, no. 2, pp. 167–170.

[70] C. D. Block, B. Manuel-y Keenoy, L. V. Gaal, and P. Rogiers, “Intensive Insulin Therapy in the Intensive Care Unit Assessment by continuous glucose monitoring,” vol. 29, no. 8, pp. 1750–1756.

[71] W. L. Clarke, D. Cox, L. A. Gonder-Frederick, W. Carter, and S. L. Pohl, “Evaluating Clinical Accuracy of Systems for Self-Monitoring of Blood Glucose,” vol. 10, no. 5, pp. 622–628.

[72] P. A. Goldberg, M. D. Siegel, R. R. Russell, R. S. Sherwin, J. I. Halickman, D. A. Cooper, J. D. Dziura, and S. E. Inzucchi, “Experience with the Continuous Glucose Monitoring System in a Medical Intensive Care Unit,” vol. 6, no. 3, pp. 339–347.

[73] K. M. Schuster, K. Barre, S. E. Inzucchi, R. Udelsman, and K. A. Davis, “Continuous glucose monitoring in the surgical intensive care unit: Concordance with capillary glucose,” vol. 76, no. 3, pp. 798–803.

[74] U. Holzinger, J. Warszawska, R. Kitzberger, H. Herkner, P. G. H. Metnitz, and C. Madl, “Impact of shock requiring norepinephrine on the accuracy and reliability of subcutaneous continuous glucose monitoring,” vol. 35, no. 8, pp. 1383–1389.

[75] J. C. Boyd and D. E. Bruns, “Effects of Measurement Frequency on Analytical Quality Required for Glucose Measurements in Intensive Care Units: Assessments by Simulation Models,” vol. 60, no. 4, pp. 644–650.

[76] B. Kovatchev, S. Anderson, L. Heinemann, and W. Clarke, “Comparison of the Numerical and Clinical Accuracy of Four Continuous Glucose Monitors,” vol. 31, no. 6, pp. 1160–1164.

[77] S. E. Noujaim, D. Horwitz, M. Sharma, and J. Marhoul, “Accuracy Requirements for a Hypoglycemia Detector: An Analytical Model to Evaluate the Effects of Bias, Precision, and Rate of Glucose Change,” vol. 1, no. 5, pp. 652–668.

[78] M. E. Wilinska and R. Hovorka, “Glucose Control in the Intensive Care Unit by Use of Continuous Glucose Monitoring: What Level of Measurement Error Is Acceptable?,” vol. 60, no. 12, pp. 1500–1509.

[79] T. V. Herpe, B. D. Moor, G. V. den Berghe, and D. Mesotten, “Modeling of Effect of Glucose Sensor Errors on Insulin Dosage and Glucose Bolus Computed by LOGIC-Insulin,” vol. 60, no. 12, pp. 1510–1518.

[80] J. C. Boyd and D. E. Bruns, “Performance Requirements for Glucose Assays in Intensive Care Units,” vol. 60, no. 12, pp. 1463–1465.

[81] B. D. Mensh, N. A. Wisniewski, B. M. Neil, and D. R. Burnett, “Susceptibility of Interstitial Continuous Glucose Monitor Performance to Sleeping Position,” vol. 7, no. 4, pp. 863–870.

[82] A. M. Albisser, B. S. Leibel, T. G. Ewart, Z. Davidovac, C. K. Botz, and W. Zingg, “An artificial endocrine pancreas,” vol. 23, no. 5, pp. 389–396.

[83] E. Pfeiffer, C. Thum, and A. Clemens, "The Artificial Beta Cell - A Continuous Control of Blood Sugar by External Regulation of Insulin Infusion (Glucose Controlled Insulin Infusion System)," vol. 6, pp. 339–342.

[84] J. Mirouze, J. L. Selam, T. C. Pham, and D. Cavadore, "Evaluation of exogenous insulin homoeostasis by the artificial pancreas in insulin-dependent diabetes," vol. 13, no. 3, pp. 273–278.

[85] A. H. Clemens, D. L. Hough, and P. A. D'Orazio, "Development of the Biostator Glucose clamping algorithm.," vol. 28, no. 9, pp. 1899–1904.

[86] G. M. Steil, C. C. Palerm, N. Kurtz, G. Voskanyan, A. Roy, S. Paz, and F. R. Kandeel, "The Effect of Insulin Feedback on Closed Loop Glucose Control," vol. 96, no. 5, pp. 1402–1408.

[87] F. J. Doyle, L. M. Huyett, J. B. Lee, H. C. Zisser, and E. Dassau, "Closed-Loop Artificial Pancreas Systems: Engineering the Algorithms," vol. 37, no. 5, pp. 1191–1197.

[88] E. Atlas, R. Nimri, S. Miller, E. A. Grunberg, and M. Phillip, "MD-Logic Artificial Pancreas System A pilot study in adults with type 1 diabetes," vol. 33, no. 5, pp. 1072–1076.

[89] M. S. Ibbini and M. A. Masadeh, "A fuzzy logic based closed-loop control system for blood glucose level regulation in diabetics," vol. 29, no. 2, pp. 64–69, 2005/03//Mar/Apr2005.

[90] R. Mauseth, I. B. Hirsch, J. Bollyky, R. Kircher, D. Matheson, S. Sanda, and C. Greenbaum, "Use of a Fuzzy Logic Controller in a Closed-Loop Artificial Pancreas," vol. 15, no. 8, pp. 628–633.

[91] P. Herrero, R. Calm, J. Veh, J. Armengol, P. Georgiou, N. Oliver, and C. Tomazou, "Robust Fault Detection System for Insulin Pump Therapy Using Continuous Glucose Monitoring," vol. 6, no. 5, pp. 1131–1141.

[92] L. Magni, D. M. Raimondo, L. Bossi, C. D. Man, G. De Nicolao, B. Kovatchev, and C. Cobelli, "Model Predictive Control of Type 1 Diabetes: An in Silico Trial," vol. 1, no. 6, pp. 804–812.

[93] R. Hovorka, V. Canonico, L. J. Chassin, U. Haueter, M. Massi-Benedetti, M. O. Federici, T. R. Pieber, H. C. Schaller, L. Schaupp, T. Vering, and M. E. Wilinska, "Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes," vol. 25, no. 4, p. 905.

[94] B. Grosman, E. Dassau, H. C. Zisser, L. Jovanovi, and F. J. Doyle, "Zone Model Predictive Control: A Strategy to Minimize Hyper- and Hypoglycemic Events," vol. 4, no. 4, pp. 961–975.

[95] H. Lee, B. A. Buckingham, D. M. Wilson, and B. W. Bequette, “A Closed-Loop Artificial Pancreas Using Model Predictive Control and a Sliding Meal Size Estimator,” vol. 3, no. 5, pp. 1082–1090.

[96] F. Cameron, B. W. Bequette, D. M. Wilson, B. A. Buckingham, H. Lee, and G. Niemeyer, “A Closed-Loop Artificial Pancreas Based on Risk Management,” vol. 5, no. 2, pp. 368–379.

[97] S. Schmidt, D. Boiroux, A. K. Duun-Henriksen, L. Frssing, O. Skyggebjerg, J. B. Jrgensen, N. K. Poulsen, H. Madsen, S. Madsbad, and K. Nrgaard, “Model-Based Closed-Loop Glucose Control in Type 1 Diabetes: The DiaCon Experience,” vol. 7, no. 5, pp. 1255–1264.

[98] S. Del Favero, D. Bruttomesso, and C. Cobelli, “Artificial Pancreas: A Review of Fundamentals and Inpatient and Outpatient Studies,” in *Frontiers in Diabetes* (D. Bruttomesso and G. Grassi, eds.), vol. 24, pp. 166–189, S. KARGER AG.

[99] F. H. El-Khatib, S. J. Russell, D. M. Nathan, R. G. Sutherlin, and E. R. Damiano, “A bihormonal closed-loop artificial pancreas for type 1 diabetes,” vol. 2, no. 27, p. 27ra27.

[100] F. Chee, T. Fernando, and P. van Heerden, “Closed-loop glucose control in critically ill patients using continuous glucose monitoring system (CGMS) in real time,” vol. 7, no. 1, pp. 43–53.

[101] T. Yatabe, R. Yamazaki, H. Kitagawa, T. Okabayashi, K. Yamashita, K. Hanazaki, and M. Yokoyama, “The evaluation of the ability of closed-loop glycemic control device to maintain the blood glucose concentration in intensive care unit patients*:,” vol. 39, no. 3, pp. 575–578.

[102] K. Yamashita and T. Yatabe, “Intraoperative glycemic control procedures and the use of an artificial pancreas,” vol. 15, no. 33, pp. 4126–4131.

[103] L. Leelarathna, S. W. English, H. Thabit, K. Caldwell, J. M. Allen, K. Kumareswaran, M. E. Wilinska, M. Nodale, J. Mangat, M. L. Evans, R. Burnstein, and R. Hovorka, “Feasibility of fully automated closed-loop glucose control using continuous subcutaneous glucose measurements in critical illness: A randomized controlled trial,” vol. 17, no. 4, p. R159.

[104] P. J. Strasma, S. Finfer, O. Flower, B. Hipszer, M. Kosiborod, L. Macken, M. Sechterberger, P. H. J. van der Voort, J. H. DeVries, and J. I. Joseph, “Use of an Intravascular Fluorescent Continuous Glucose Sensor in ICU Patients,” vol. 9, no. 4, pp. 762–770.

[105] I. I. Raad and G. P. Bodey, “Infectious Complications of Indwelling Vascular Catheters,” vol. 15, no. 2, pp. 197–208.

[106] “Catheter Complications in Total Parenteral Nutrition A Prospective Study of 200 Consecutive Patients NEJM.”

[107] I. I. Raad, M. Luna, S.-A. M. Khalil, J. W. Costerton, C. Lam, and G. P. Bodey, "The Relationship Between the Thrombotic and Infectious Complications of Central Venous Catheters," vol. 271, no. 13, pp. 1014–1016.

[108] S. Skogestad and M. Morari, "Understanding the dynamic behavior of distillation columns," vol. 27, no. 10, pp. 1848–1862.

[109] S. P. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press.

[110] J. Lin, N. N. Razak, C. G. Pretty, A. Le Compte, P. Docherty, J. D. Parente, G. M. Shaw, C. E. Hann, and J. Geoffrey Chase, "A physiological Intensive Control Insulin-Nutrition-Glucose (ICING) model validated in critically ill patients," vol. 102, no. 2, pp. 192–205.

[111] J. Plank, J. Blaha, J. Cordingley, M. E. Wilinska, L. J. Chassin, C. Morgan, S. Squire, M. Haluzik, J. Kremen, S. Svacina, W. Toller, A. Plasnik, M. Ellmerer, R. Hovorka, and T. R. Pieber, "Multicentric, Randomized, Controlled Trial to Evaluate Blood Glucose Control by the Model Predictive Control Algorithm Versus Routine Glucose Management Protocols in Intensive Care Unit Patients," vol. 29, no. 2, pp. 271–276.

[112] A. Roy, "Dynamic Modeling of Free Fatty Acid, Glucose, and Insulin During Rest and Exercise in Insulin Dependent Diabetes Mellitus Patients."

[113] P. D. Docherty, J. G. Chase, C. E. Hann, T. F. Lotz, J. Lin, K. A. McAuley, and G. M. Shaw, "The Identification of Insulin Saturation Effects During the Dynamic Insulin Sensitivity Test," vol. 4, pp. 141–148.

[114] A. Pritchard-Bell, G. Clermont, T. D. Knab, J. Maalouf, and R. S. Parker, "Subcutaneous Insulin Dynamics in a Critical Care Population," (Submitted).

[115] D. C. Howey, R. R. Bowsher, R. L. Brunelle, and J. R. Woodworth, "[Lys(B28), Pro(B29)]-Human Insulin: A Rapidly Absorbed Analogue of Human Insulin," vol. 43, no. 3, pp. 396–402.

[116] V. Melki, E. Renard, V. Lassmann-Vague, S. Boivin, B. Guerci, H. Hanaire-Broutin, J. Bringer, P. Belicar, N. Jeandidier, L. Meyer, P. Blin, B. Augendre-Ferrante, and J.-P. Tauber, "Improvement of HbA1c and Blood Glucose Stability in IDDM Patients Treated With Lispro Insulin Analog in External Pumps," vol. 21, no. 6, pp. 977–982.

[117] L. Crenier, C. Abou-Elias, and B. Corvilain, "Glucose Variability Assessed by Low Blood Glucose Index Is Predictive of Hypoglycemic Events in Patients With Type 1 Diabetes Switched to Pump Therapy," vol. 36, no. 8, pp. 2148–2153.

[118] S. Ellahham, "Insulin therapy in critically ill patients," vol. 6, pp. 1089–1101.

[119] A. Prieto-Tenreiro, R. Villar-Taibo, M. Pazos-Couselo, M. Gonzlez-Rodrguez, F. Casanueva, and J. M. Garca-Lpez, "[benefits of subcutaneous continuous insulin

infusion in type 1 diabetic patients with high glycemic variability],” vol. 59, no. 4, pp. 246–253.

[120] B. Guerci and J. P. Sauvanet, “Subcutaneous insulin: Pharmacokinetic variability and glycemic variability,” vol. 31, pp. 4S7–4S24.

[121] B. Yegneswaran, R. Parket, S. Gawel, A. Pritchard-Bell, T. Ho, and G. Clermont, “Association between average glucose levels and hospital mortality among critically ill patients,” vol. 17, p. P458.

[122] L. M. Fisk, A. J. Le Compte, G. M. Shaw, and J. G. Chase, “Improving safety of glucose control in intensive care using virtual patients and simulated clinical trials,” vol. 3, no. 3, pp. 415–430.

[123] K. Choi, T. J. Oh, J. C. Lee, M. Kim, H. C. Kim, Y. M. Cho, and S. Kim, “In-Silico Trials for Glucose Control in Hospitalized Patients with Type 2 Diabetes,” vol. 31, no. 2, pp. 231–239.

[124] S. S. Kanderian, S. A. Weinzimer, and G. M. Steil, “The Identifiable Virtual Patient Model: Comparison of Simulation and Clinical Closed-Loop Study Results,” vol. 6, no. 2, pp. 371–379.

[125] P. Palumbo, G. Pizzichelli, S. Panunzi, P. Pepe, and A. D. Gaetano, “Tests on a virtual patient for an observer-based, closed-loop control of plasma glycemia,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 6936–6941.

[126] A. Guerrini, M. Sorine, and P. Kalfon, “Assessing Performances of Glucose Control Algorithms on a Set of Virtual ICU Patients,”

[127] J. G. Chase, F. Suhaimi, S. Penning, J.-C. Preiser, A. J. Le Compte, J. Lin, C. G. Pretty, G. M. Shaw, K. T. Moorhead, and T. Desaive, “Validation of a model-based virtual trials method for tight glycemic control in intensive care,” vol. 9, p. 84.

[128] J. C. Lee, M. Kim, K. R. Choi, T. J. Oh, M. Y. Kim, Y. M. Cho, K. Kim, H. C. Kim, and S. Kim, “In Silico Evaluation of Glucose Control Protocols for Critically Ill Patients,” vol. 59, no. 1, pp. 54–57.

[129] B. Bequette, F. Cameron, N. Baysal, D. Howsmon, B. Buckingham, D. Maahs, and C. Levy, “Algorithms for a Single Hormone Closed-Loop Artificial Pancreas: Challenges Pertinent to Chemical Process Operations and Control,” vol. 4, no. 4, p. 39.

[130] A. Roy and R. S. Parker, “Dynamic modeling of free fatty acid, glucose, and insulin: An extended ”minimal model”,” vol. 8, no. 6, pp. 617–626.

[131] S. Penning, C. Pretty, J.-C. Preiser, G. M. Shaw, T. Desaive, and J. G. Chase, “Glucose control positively influences patient outcome: A retrospective study,” vol. 30, no. 3, pp. 455–459.

[132] D. C. Klonoff, "Continuous Glucose Monitoring Roadmap for 21st century diabetes therapy," vol. 28, no. 5, pp. 1231–1239.

[133] D. C. Klonoff, C. Lias, S. Beck, J. L. Parkes, B. Kovatchev, R. A. Vigersky, G. Arreaza-Rubin, R. D. Burk, A. Kowalski, R. Little, J. Nichols, M. Petersen, K. Rawlings, D. B. Sacks, E. Sampson, S. Scott, J. J. Seley, R. Slingerland, and H. W. Vesper, "Development of the Diabetes Technology Society Blood Glucose Monitor System Surveillance Protocol," p. 1932296815614587.

[134] B. W. Bode, T. M. Gross, K. R. Thornton, and J. J. Mastrototaro, "Continuous glucose monitoring used to adjust diabetes therapy improves glycosylated hemoglobin: A pilot study," vol. 46, no. 3, pp. 183–190.

[135] H. P. Chase, L. M. Kim, S. L. Owen, T. A. MacKenzie, G. J. Klingensmith, R. Murtfeldt, and S. K. Garg, "Continuous subcutaneous glucose monitoring in children with type 1 diabetes," vol. 107, no. 2, pp. 222–226.

[136] C. L. Rohlfing, H.-M. Wiedmeyer, R. R. Little, J. D. England, A. Tennill, and D. E. Goldstein, "Defining the Relationship Between Plasma Glucose and HbA1c Analysis of glucose profiles and HbA1c in the Diabetes Control and Complications Trial," vol. 25, no. 2, pp. 275–278.

[137] J. C. Pickup, S. C. Freeman, and A. J. Sutton, "Glycaemic control in type 1 diabetes during real time continuous glucose monitoring compared with self monitoring of blood glucose: Meta-analysis of randomised controlled trials using individual patient data," vol. 343, p. d3805.

[138] S. Garg, H. Zisser, S. Schwartz, T. Bailey, R. Kaplan, S. Ellis, and L. Jovanovic, "Improvement in Glycemic Excursions With a Transcutaneous, Real-Time Continuous Glucose Sensor A randomized controlled trial," vol. 29, no. 1, pp. 44–50.

[139] S. Garg and L. Jovanovic, "Relationship of Fasting and Hourly Blood Glucose Levels to HbA1c Values Safety, accuracy, and improvements in glucose profiles obtained using a 7-day continuous glucose sensor," vol. 29, no. 12, pp. 2644–2649.

[140] "Press Announcements - FDA approves first automated insulin delivery device for type 1 diabetes."

[141] A. M. Corstjens, J. J. Ligtenberg, I. C. van der Horst, R. Spanjersberg, J. S. Lind, J. E. Tulleken, J. H. Meertens, and J. G. Zijlstra, "Accuracy and feasibility of point-of-care and continuous blood glucose analysis in critically ill ICU patients," vol. 10, no. 5, p. R135.

[142] R. T. van Hooijdonk, T. Winters, J. C. Fischer, E. C. van Dongen-Lases, J. S. Krinsley, J.-C. Preiser, and M. J. Schultz, "Accuracy and limitations of continuous glucose monitoring using spectroscopy in critically ill patients," vol. 4, p. 8.

[143] K. L. Helton, B. D. Ratner, and N. A. Wisniewski, "Biomechanics of the SensorTissue InterfaceEffects of Motion, Pressure, and Design on Sensor Performance and the Foreign Body ResponsePart I: Theoretical Framework," vol. 5, no. 3, pp. 632–646.

[144] K. L. Helton, B. D. Ratner, and N. A. Wisniewski, "Biomechanics of the SensorTissue InterfaceEffects of Motion, Pressure, and Design on Sensor Performance and Foreign Body ResponsePart II: Examples and Application," vol. 5, no. 3, pp. 647–656.

[145] D. M. Maahs, D. DeSalvo, L. Pyle, T. Ly, L. Messer, P. Clinton, E. Westfall, R. P. Wadwa, and B. Buckingham, "Effect of Acetaminophen on CGM Glucose in an Outpatient Setting," vol. 38, no. 10, pp. e158–e159.

[146] K. Tonyushkina and J. H. Nichols, "Glucose Meters: A Review of Technical Challenges to Obtaining Accurate Results," vol. 3, no. 4, pp. 971–980.

[147] YSI, "YSI 2300 Stat Plus Glucose & Lactate Ananlyzer — ysi.com."

[148] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: modeling and solving mathematical programs in python," vol. 3, no. 3, pp. 219–260.

[149] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff, *Pyomo Optimization Modeling in Python*, vol. 67 of *Springer Optimization and Its Applications.* Springer US.

[150] A. Wchter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," vol. 106, no. 1, pp. 25–57.

[151] HSL, "A collection of Fortran codes for large scale scientific computation.."

[152] cofa1, "Dexcom product guides - user guides, quick start, tutorials."

[153] A. Facchinetti, S. Del Favero, G. Sparacino, J. Castle, W. Ward, and C. Cobelli, "Modeling the Glucose Sensor Error," vol. 61, no. 3, pp. 620–629.

[154] B. P. Kovatchev, "Hypoglycemia Reduction and Accuracy of Continuous Glucose Monitoring," vol. 17, no. 8, pp. 530–533.

[155] H. Akaike, "A Bayesian extension of the minimum AIC procedure of autoregressive model fitting," vol. 66, no. 2, pp. 237–242.

[156] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach.* Springer, 2nd ed ed. OCLC: ocm48557578.

[157] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, pp. 57–61.

[158] "System and methods for processing analyte sensor data for sensor calibration."

[159] H. Kirchsteiger, J. B. Jrgensen, E. Renard, and L. del Re, *Prediction Methods for Blood Glucose Concentration: Design, Use and Evaluation.* Springer.

[160] A. Pritchard-Bell, G. Clermont, T. Knab, D., J. Maalouf, M. Vilkhovoy, and R. S. Parker, "Modeling Glucose and Subcutaneous Insulin Dynamics in Critical Care," In Press.

[161] C. R. Kahn, "Insulin resistance, insulin insensitivity, and insulin unresponsiveness: A necessary distinction," vol. 27, no. 12, pp. 1893–1902.

[162] R. N. Bergman, L. S. Phillips, and C. Cobelli, "Physiologic evaluation of factors controlling glucose tolerance in man: Measurement of insulin sensitivity and beta-cell glucose sensitivity from the response to intravenous glucose.," vol. 68, no. 6, pp. 1456–1467.

[163] R. L. Prigeon, M. E. Rder, D. Porte, and S. E. Kahn, "The effect of insulin dose on the measurement of insulin sensitivity by the minimal model technique. Evidence for saturable insulin transport in humans.," vol. 97, no. 2, pp. 501–507.

[164] A. Natali, A. Gastaldelli, S. Camastra, A. M. Sironi, E. Toschi, A. Masoni, E. Ferrannini, and A. Mari, "Dose-response characteristics of insulin action on glucose metabolism: A non-steady-state approach," vol. 278, no. 5, pp. E794–E801.

[165] J. G. Chase, G. M. Shaw, J. Lin, C. V. Doran, M. Bloomfield, G. C. Wake, B. Broughton, C. Hann, and T. Lotz, "Impact of Insulin-Stimulated Glucose Removal Saturation on Dynamic Modelling and Control of Hyperglycaemia," vol. 1, pp. 79–94.

[166] R. A. Rizza, L. J. Mandarino, and J. E. Gerich, "Dose-response characteristics for effects of insulin on production and utilization of glucose in man," vol. 240, no. 6, pp. E630–E639.

[167] P. N. Bvenholm, J. Pigon, C.-G. stenson, and S. Efendic, "Insulin Sensitivity of Suppression of Endogenous Glucose Production Is the Single Most Important Determinant of Glucose Tolerance," vol. 50, no. 6, pp. 1449–1454.

[168] W. C. Duckworth, R. G. Bennett, and F. G. Hamel, "Insulin Degradation: Progress and Potential," vol. 19, no. 5, pp. 608–624.

[169] C. D. Man, R. A. Rizza, and C. Cobelli, "Meal Simulation Model of the Glucose-Insulin System," vol. 54, no. 10, pp. 1740–1749.

[170] A. Pritchard-Bell, "Mathematical Modeling in Systems Medicine: New Paradigms for Glucose Control in Critical Care."

[171] A. Roy and R. S. Parker, "A phenomenological model of plasma FFA, glucose, and insulin concentrations during rest and exercise," in *American Control Conference (ACC), 2010*, pp. 5161–5166.

[172] E. Cengiz and W. V. Tamborlane, "A Tale of Two Compartments: Interstitial Versus Blood Glucose Monitoring," vol. 11, pp. S–11–S–16.

[173] P. J. Stout, N. Peled, B. J. Erickson, M. E. Hilgers, J. R. Racchini, and T. B. Hoegh, “Comparison of Glucose Levels in Dermal Interstitial Fluid and Finger Capillary Blood,” vol. 3, no. 1, pp. 81–90.

[174] R. J. Davey, C. Low, T. W. Jones, and P. A. Fournier, “Contribution of an intrinsic lag of continuous glucose monitoring systems to differences in measured and actual glucose concentrations changing at variable rates in vitro,” vol. 4, no. 6, pp. 1393–1399.

[175] E. Kulcu, J. A. Tamada, G. Reach, R. O. Potts, and M. J. Lesho, “Physiological Differences Between Interstitial Glucose and Blood Glucose Measured in Human Subjects,” vol. 26, no. 8, pp. 2405–2409.

[176] A. Basu, S. Dube, M. Slama, I. Errazuriz, J. C. Amezcua, Y. C. Kudva, T. Peyser, R. E. Carter, C. Cobelli, and R. Basu, “Time Lag of Glucose From Intravascular to Interstitial Compartment in Humans,” vol. 62, no. 12, pp. 4083–4087.

[177] A. Basu, S. Dube, S. Veettil, M. Slama, Y. C. Kudva, T. Peyser, R. E. Carter, C. Cobelli, and R. Basu, “Time lag of glucose from intravascular to interstitial compartment in type 1 diabetes,” vol. 9, no. 1, pp. 63–68.

[178] A. Facchinetti, G. Sparacino, and C. Cobelli, “Enhanced accuracy of continuous glucose monitoring by online extended kalman filtering,” vol. 12, no. 5, pp. 353–363.

[179] M. Breton and B. Kovatchev, “Analysis, Modeling, and Simulation of the Accuracy of Continuous Glucose Sensors,” vol. 2, no. 5, pp. 853–862.

[180] D. J. Lunn, C. Wei, and R. Hovorka, “Fitting dynamic models with forcing functions: Application to continuous glucose monitoring in insulin therapy,” vol. 30, no. 18, pp. 2234–2250.

[181] S. Guerra, A. Facchinetti, G. Sparacino, G. D. Nicolao, and C. Cobelli, “Enhancing the accuracy of subcutaneous glucose sensors: A real-time deconvolution-based approach,” vol. 59, no. 6, pp. 1658–1669.

[182] O. M. Alifanov, E. A. Artiukhin, and S. V. Rumiantsev, *Extreme Methods for Solving Ill-Posed Problems with Applications to Inverse Heat Transfer Problems.* Begell House.

[183] C. Simon and G. Brandenberger, “Ultradian oscillations of insulin secretion in humans,” vol. 51, pp. S258–S261.

[184] C. Pretty, P. Docherty, J. Lin, L. Pfeifer, U. Jamaludin, G. Shaw, A. Le Compte, and J. Chase, “Endogenous insulin secretion and suppression during and after sepsis in critically ill patients: Implications for tight glycemic control protocols,” vol. 15, p. P389.

[185] J. Li, Y. Kuang, and C. C. Mason, “Modeling the glucoseinsulin regulatory system and ultradian insulin secretory oscillations with two explicit time delays,” vol. 242, no. 3, pp. 722–735.

[186] A. D. Baron, O. G. Kolterman, J. Bell, L. J. Mandarino, and J. M. Olefsky, “Rates of noninsulin-mediated glucose uptake are elevated in type II diabetic subjects.,” vol. 76, no. 5, pp. 1782–1788.

[187] M. P. Macedo, I. S. Lima, J. M. Gaspar, R. A. Afonso, R. S. Patarro, Y.-B. Kim, and R. T. Ribeiro, “Risk of postprandial insulin resistance: The liver/vagus rapport,” vol. 15, no. 1, pp. 67–77.

[188] R. A. DeFronzo and D. Tripathy, “Skeletal Muscle Insulin Resistance Is the Primary Defect in Type 2 Diabetes,” vol. 32, pp. S157–S163.

[189] R. Nesher and E. Cerasi, “Modeling phasic insulin release immediate and time-dependent effects of glucose,” vol. 51, pp. S53–S59.

[190] J.-C. Henquin, N. Ishiyama, M. Nenquin, M. A. Ravier, and J.-C. Jonas, “Signals and Pools Underlying Biphasic Insulin Secretion,” vol. 51, pp. S60–S67.

[191] P. Rorsman, L. Eliasson, E. Renstrm, J. Gromada, S. Barg, and S. Gpel, “The Cell Physiology of Biphasic Insulin Secretion,” vol. 15, no. 2, pp. 72–77.

[192] Z. Wang and D. C. Thurmond, “Mechanisms of biphasic insulin-granule exocytosis - roles of the cytoskeleton, small GTPases and SNARE proteins,” vol. 122, no. 7, pp. 893–903.

[193] S. G. Straub and G. W. G. Sharp, “Glucose-stimulated signaling pathways in biphasic insulin secretion,” vol. 18, no. 6, pp. 451–463.

[194] K. S. Polonsky, B. D. Given, L. Hirsch, E. T. Shapiro, H. Tillil, C. Beebe, J. A. Galloway, B. H. Frank, T. Karrison, and E. Van Cauter, “Quantitative study of insulin secretion and clearance in normal and obese subjects.,” vol. 81, no. 2, pp. 435–441.

[195] S. Gudbjrnsdttir, M. Sjstrand, L. Strindberg, J. Wahren, and P. Lnnroth, “Direct Measurements of the Permeability Surface Area for Insulin and Glucose in Human Skeletal Muscle,” vol. 88, no. 10, pp. 4559–4564.

[196] M. Sjstrand, A. Holmng, and P. Lnnroth, “Measurement of interstitial insulin in human muscle,” vol. 276, no. 1, pp. E151–E154.

[197] M. Sjstrand, A. Holmng, L. Strindberg, and P. Lnnroth, “Estimations of muscle interstitial insulin, glucose, and lactate in type 2 diabetic subjects,” vol. 279, no. 5, pp. E1097–E1103.

[198] N. M. O’Meara, J. Sturis, E. Van Cauter, and K. S. Polonsky, “Lack of control by glucose of ultradian insulin secretory oscillations in impaired glucose tolerance and in non-insulin-dependent diabetes mellitus.,” vol. 92, no. 1, p. 262.

[199] C. McDonald, A. Dunaif, and D. T. Finegood, "Minimal-Model Estimates of Insulin Sensitivity Are Insensitive to Errors in Glucose Effectiveness 1," vol. 85, no. 7, pp. 2504–2508.

[200] G. Pillonetto, G. Sparacino, P. Magni, R. Bellazzi, and C. Cobelli, "Minimal model SI=0 problem in NIDDM subjects: Nonzero Bayesian estimates with credible confidence intervals," vol. 282, no. 3, pp. E564–E573.

[201] A. Mari and A. Valerio, "A circulatory model for the estimation of insulin sensitivity," vol. 5, no. 12, pp. 1747–1752.

[202] K. Turnheim and W. K. Waldhusl, "Essentials of insulin pharmacokinetics," vol. 100, no. 3, pp. 65–72.

[203] Y. T. Kruszynska, P. D. Home, I. Hanning, and K. G. Alberti, "Basal and 24-h C-peptide and insulin secretion rate in normal man," vol. 30, no. 1, pp. 16–21.

[204] T. E. Shapiro, H. Tillil, A. Rubenstein, and K. Polonsky, "Peripheral Insulin Parallels Changes in Insulin Secretion More Closely Than C-Peptide After Bolus Intravenous Glucose Administration*," vol. 67, no. 5, pp. 1094–1099.

[205] M. H. Shanik, Y. Xu, J. krha, R. Dankner, Y. Zick, and J. Roth, "Insulin Resistance and Hyperinsulinemia Is hyperinsulinemia the cart or the horse?," vol. 31, pp. S262–S268.

[206] B. A. Cooperberg and P. E. Cryer, "Glucagon supports postabsorptive plasma glucose concentrations in humans with biologically optimal insulin levels," vol. 59, no. 11, pp. 2941–2944.

[207] J. Iversen and D. W. Miles, "Evidence for a Feedback Inhibition of Insulin on Insulin Secretion in the Isolated, Perfused Canine Pancreas," vol. 20, no. 1, pp. 1–9.

[208] I. B. Leibiger, B. Leibiger, and P.-O. Berggren, "Insulin feedback action on pancreatic -cell function," vol. 532, pp. 1–6.

[209] C. J. Rhodes, M. F. White, J. L. Leahy, and S. E. Kahn, "Direct Autocrine Action of Insulin on -Cells: Does It Make Physiological Sense?," vol. 62, no. 7, pp. 2157–2163.

[210] A. R. Saltiel and C. R. Kahn, "Insulin signalling and the regulation of glucose and lipid metabolism," vol. 414, no. 6865, pp. 799–806.

[211] C. J. Hedeskov, "Mechanism of glucose-induced insulin secretion.," vol. 60, no. 2, pp. 442–509.

[212] J. Olefsky, J. W. Farquhar, and G. Reaven, "Relationship between fasting plasma insulin level and resistance to insulin-mediated glucose uptake in normal and diabetic subjects," vol. 22, no. 7, pp. 507–513.

[213] A. Laws, A. C. King, W. L. Haskell, and G. M. Reaven, "Relation of fasting plasma insulin concentration to high density lipoprotein cholesterol and triglyceride concentrations in men.," vol. 11, no. 6, pp. 1636–1642.

[214] T. Rnnemaa, M. Laakso, K. Pyrl, V. Kallio, and P. Puukka, "High fasting plasma insulin is an indicator of coronary heart disease in non-insulin-dependent diabetic patients and nondiabetic subjects.," vol. 11, no. 1, pp. 80–90.

[215] C. Weyer, R. L. Hanson, P. A. Tataranni, C. Bogardus, and R. E. Pratley, "A high fasting plasma insulin concentration predicts type 2 diabetes independent of insulin resistance: Evidence for a pathogenic role of relative hyperinsulinemia.," vol. 49, no. 12, pp. 2094–2101.

[216] C. G. Pretty, A. J. Le Compte, J. G. Chase, G. M. Shaw, J.-C. Preiser, S. Penning, and T. Desaive, "Variability of insulin sensitivity during the first 4 days of critical illness: Implications for tight glycemic control," vol. 2, no. 1, p. 17.

[217] A. Blakemore, S.-H. Wang, A. Le Compte, G. M. Shaw, X.-W. Wong, J. Lin, T. Lotz, C. E. Hann, and J. G. Chase, "Model-Based Insulin Sensitivity as a Sepsis Diagnostic in Critical Care," vol. 2, no. 3, pp. 468–477.

[218] T. Ferenci, B. Beny, L. Kovcs, L. Fisk, G. M. Shaw, and J. G. Chase, "Daily Evolution of Insulin Sensitivity Variability with Respect to Diagnosis in the Critically Ill," vol. 8, no. 2, p. e57119.

[219] S. D. Mittelman, Y. Y. Fu, K. Rebrin, G. Steil, and R. N. Bergman, "Indirect effect of insulin to suppress endogenous glucose production is dominant, even with hyperglucagonemia.," vol. 100, no. 12, pp. 3121–3130.

[220] A. Seppl-Lindroos, S. Vehkavaara, A.-M. Hkkinen, T. Goto, J. Westerbacka, A. Sovijrvi, J. Halavaara, and H. Yki-Jrvinen, "Fat Accumulation in the Liver Is Associated with Defects in Insulin Suppression of Glucose Production and Serum Free Fatty Acids Independent of Obesity in Normal Men," vol. 87, no. 7, pp. 3023–3028.

[221] A. Mitrakou, D. Kelley, M. Mokan, T. Veneman, T. Pangburn, J. Reilly, and J. Gerich, "Role of reduced suppression of glucose production and diminished early insulin release in impaired glucose tolerance," vol. 326, no. 1, pp. 22–29.

[222] I. M. Toli, E. Mosekilde, and J. Sturis, "Modeling the InsulinGlucose Feedback System: The Significance of Pulsatile Insulin Secretion," vol. 207, no. 3, pp. 361–375.

[223] E. Watson, M. Chappell, F. Ducrozet, S. Poucher, and J. Yates, "A new general glucose homeostatic model using a proportional-integral-derivative controller," vol. 102, no. 2, pp. 119–129.

[224] S. A. Weinzimer, G. M. Steil, K. L. Swan, J. Dziura, N. Kurtz, and W. V. Tamborlane, "Fully Automated Closed-Loop Insulin Delivery Versus Semiautomated Hybrid Control

in Pediatric Patients With Type 1 Diabetes Using an Artificial Pancreas," vol. 31, no. 5, pp. 934–939.

[225] E. Renard, J. Place, M. Cantwell, H. Chevassus, and C. C. Palerm, "Closed-Loop Insulin Delivery Using a Subcutaneous Glucose Sensor and Intraperitoneal Insulin Delivery," vol. 33, no. 1, pp. 121–127.

[226] T. Peyser, E. Dassau, M. Breton, and J. S. Skyler, "The artificial pancreas: Current status and future prospects in the management of diabetes," vol. 1311, no. 1, pp. 102–123.

[227] G. Marchetti, M. Barolo, L. Jovanovic, H. Zisser, and D. E. Seborg, "An Improved PID Switching Control Strategy for Type 1 Diabetes," vol. 55, no. 3, pp. 857–865.

[228] C. Cobelli, E. Renard, and B. Kovatchev, "Artificial Pancreas: Past, Present, Future," vol. 60, no. 11, pp. 2672–2682.

[229] W. L. Clarke, S. Anderson, M. Breton, S. Patek, L. Kashmer, and B. Kovatchev, "Closed-Loop Artificial Pancreas Using Subcutaneous Glucose Sensing and Insulin Delivery and a Model Predictive Control Algorithm: The Virginia Experience," vol. 3, no. 5, pp. 1031–1038.

[230] D. Bruttomesso, A. Farret, S. Costa, M. C. Marescotti, M. Vettore, A. Avogaro, A. Tiengo, C. D. Man, J. Place, A. Facchinetti, S. Guerra, L. Magni, G. D. Nicolao, C. Cobelli, E. Renard, and A. Maran, "Closed-Loop Artificial Pancreas Using Subcutaneous Glucose Sensing and Insulin Delivery and a Model Predictive Control Algorithm: Preliminary Studies in Padova and Montpellier," vol. 3, no. 5, pp. 1014–1021.

[231] B. W. Bequette, "A Critical Assessment of Algorithms and Challenges in the Development of a Closed-Loop Artificial Pancreas," vol. 7, no. 1, pp. 28–47.

[232] R. A. Ritzel, A. E. Butler, R. A. Rizza, J. D. Veldhuis, and P. C. Butler, "Relationship Between -Cell Mass and Fasting Blood Glucose Concentration in Humans," vol. 29, no. 3, pp. 717–718.

[233] O. Ajala, H. Lockett, G. Twine, and D. E. Flanagan, "Depth and duration of hypoglycaemia achieved during the insulin tolerance test," vol. 167, no. 1, pp. 59–65.

[234] A. Caraty, M. Grino, A. Locatelli, V. Guillaume, F. Boudouresque, B. Conte-Devolx, and C. Oliver, "Insulin-induced hypoglycemia stimulates corticotropin-releasing factor and arginine vasopressin secretion into hypophysial portal blood of conscious, unrestrained rams.," vol. 85, no. 6, pp. 1716–1721.

[235] N. Tesfaye and E. R. Seaquist, "Neuroendocrine Responses to Hypoglycemia," vol. 1212, pp. 12–28.

[236] J. E. Sprague and A. M. Arbelez, "Glucose Counterregulatory Responses to Hypoglycemia," vol. 9, no. 1, pp. 463–475.

[237] J. Gerich, P. Cryer, and R. Rizza, "Hormonal mechanisms in acute glucose counterregulation: The relative roles of glucagon, epinephrine, norepinephrine, growth hormone, and cortisol," vol. 29, no. 11, pp. 1164–1175.

[238] P. D. Feo, G. Perriello, E. Torlone, M. M. Ventura, C. Fanelli, F. Santeusanio, P. Brunetti, J. E. Gerich, and G. B. Bolli, "Contribution of cortisol to glucose counterregulation in humans," vol. 257, no. 1, pp. E35–E42.

[239] I. Lager, "The insulin-antagonistic effect of the counterregulatory hormones," vol. 735, pp. 41–47.

[240] S. N. Davis, C. Shavers, F. Costa, and R. Mosqueda-Garcia, "Role of cortisol in the pathogenesis of deficient counterregulation after antecedent hypoglycemia in normal humans.," vol. 98, no. 3, pp. 680–691.

[241] S. N. Davis, C. Shavers, B. Davis, and F. Costa, "Prevention of an increase in plasma cortisol during hypoglycemia preserves subsequent counterregulatory responses.," vol. 100, no. 2, pp. 429–438.

[242] A. V. Turnbull and C. L. Rivier, "Regulation of the hypothalamic-pituitary-adrenal axis by cytokines: Actions and mechanisms of action," vol. 79, no. 1, pp. 1–71.

[243] C. Tsigos and G. P. Chrousos, "Hypothalamicpituitaryadrenal axis, neuroendocrine factors and stress," vol. 53, no. 4, pp. 865–871.

[244] S. M. Smith and W. W. Vale, "The role of the hypothalamic-pituitary-adrenal axis in neuroendocrine responses to stress," vol. 8, no. 4, pp. 383–395.

[245] C. W. Burke, "The Pituitary Megatest: Outdated?," vol. 36, no. 2, pp. 133–134.

[246] S. L. Jones, P. J. Trainer, L. Perry, J. a. H. Wass, G. M. Besser, and A. Grossman, "An audit of the insulin tolerance test in adult subjects in an acute investigation unit over one year," vol. 41, no. 1, pp. 123–128.

[247] P. Vestergaard, H. C. Hoeck, P. E. Jakobsen, and P. Laurberg, "Reproducibility of Growth Hormone and Cortisol Responses to the Insulin Tolerance Test and the Short ACTH Test in Normal Adults," vol. 29, pp. 106–110.

[248] E. Erturk, C. A. Jaffe, and A. L. Barkan, "Evaluation of the Integrity of the Hypothalamic-Pituitary-Adrenal Axis by Insulin Hypoglycemia Test," vol. 83, no. 7, pp. 2350–2354.

[249] M. Lange, O. L. Svendsen, N. E. Skakkebaek, J. Muller, A. Juul, M. Schmiegelow, and U. Feldt-Rasmussen, "An audit of the insulin-tolerance test in 255 patients with pituitary disease," vol. 147, no. 1, pp. 41–47.

[250] G. Dickstein, "The assessment of the hypothalamo-pituitary-adrenal axis in pituitary disease: Are there short cuts?," vol. 26, pp. 25–30.

[251] F. C. Greenwood, J. Landon, and T. C. Stamp, "The plasma sugar, free fatty acid, cortisol, and growth hormone response to insulin. I. In control subjects.," vol. 45, no. 4, pp. 429–436.

[252] M. Matsuda and R. A. DeFronzo, "Insulin sensitivity indices obtained from oral glucose tolerance testing: Comparison with the euglycemic insulin clamp.," vol. 22, no. 9, pp. 1462–1470.

[253] A. Mari, G. Pacini, E. Murphy, B. Ludvik, and J. J. Nolan, "A Model-Based Method for Assessing Insulin Sensitivity From the Oral Glucose Tolerance Test," vol. 24, no. 3, pp. 539–548.

[254] M. Stumvoll, A. Mitrakou, W. Pimenta, T. Jenssen, H. Yki-Jrvinen, T. V. Haeften, W. Renn, and J. Gerich, "Use of the oral glucose tolerance test to assess insulin release and insulin sensitivity.," vol. 23, no. 3, pp. 295–301.

[255] G. Pacini and R. N. Bergman, "MINMOD: A computer program to calculate insulin sensitivity and pancreatic responsivity from the frequently sampled intravenous glucose tolerance test," vol. 23, no. 2, pp. 113–122.

[256] K. Alberti and P. Zimmet, "Definition, diagnosis and classification of diabetes mellitus and its complications. Part 1: Diagnosis and classification of diabetes mellitus. Provisional report of a WHO Consultation," vol. 15, no. 7, pp. 539–553.

[257] "Glucose tolerance test Results - Mayo Clinic."

[258] "Glucose Tests: The Test — Glucose Test: Blood Sugar; Blood Glucose; Fasting Blood Glucose; Oral Glucose Tolerance Test; OGTT; GTT; Urine Glucose — Lab Tests Online."

[259] T. Vilsbll, T. Krarup, J. Sonne, S. Madsbad, A. Vlund, A. G. Juul, and J. J. Holst, "Incretin Secretion in Relation to Meal Size and Body Weight in Healthy Subjects and People with Type 1 and Type 2 Diabetes Mellitus," vol. 88, no. 6, pp. 2706–2713.

[260] E. Muscelli, A. Mari, A. Casolaro, S. Camastra, G. Seghieri, A. Gastaldelli, J. J. Holst, and E. Ferrannini, "Separate Impact of Obesity and Glucose Tolerance on the Incretin Effect in Normal Subjects and Type 2 Diabetic Patients," vol. 57, no. 5, pp. 1340–1348.

[261] S. H. Knudsen, K. Karstoft, B. K. Pedersen, G. van Hall, and T. P. J. Solomon, "The immediate effects of a single bout of aerobic exercise on oral glucose tolerance across the glucose tolerance continuum," vol. 2, no. 8, p. e12114.

[262] K. Vollmer, J. J. Holst, B. Baller, M. Ellrichmann, M. A. Nauck, W. E. Schmidt, and J. J. Meier, "Predictors of Incretin Concentrations in Subjects With Normal, Impaired, and Diabetic Glucose Tolerance," vol. 57, no. 3, pp. 678–687.

[263] M. Axelsen, "Postprandial Hypertriglyceridemia and Insulin Resistance in Normoglycemic First-Degree Relatives of Patients with Type 2 Diabetes," vol. 131, no. 1, p. 27.

[264] E. Breda, M. K. Cavaghan, G. Toffolo, K. S. Polonsky, and C. Cobelli, "Oral Glucose Tolerance Test Minimal Model Indexes of -Cell Function and Insulin Sensitivity," vol. 50, no. 1, pp. 150–158.

[265] K. Frch, G. Pacini, J. J. Nolan, T. Hansen, A. Tura, and D. Vistisen, "Impact of Glucose Tolerance Status, Sex, and Body Size on Glucose Absorption Patterns During OGTTs," vol. 36, no. 11, pp. 3691–3697.

[266] G. Bock, C. D. Man, M. Campioni, E. Chittilapilly, R. Basu, G. Toffolo, C. Cobelli, and R. Rizza, "Pathogenesis of Pre," vol. 55, no. 12, pp. 3536–3549.

[267] E. Breda, G. Toffolo, K. S. Polonsky, and C. Cobelli, "Insulin Release in Impaired Glucose Tolerance Oral Minimal Model Predicts Normal Sensitivity to Glucose but Defective Response Times," vol. 51, pp. S227–S233.

[268] M. C. Moore, S. N. Davis, S. L. Mann, and A. D. Cherrington, "Acute Fructose Administration Improves Oral Glucose Tolerance in Adults With Type 2 Diabetes," vol. 24, no. 11, pp. 1882–1887.

[269] "Drugs@FDA: FDA Approved Drug Products."

[270] "HUMALOG INSULIN LISPRO INJECTION, USP(rDNA ORIGIN)100 UNITS PER ML (U-100)."

[271] R. A. Rizza, P. E. Cryer, and J. E. Gerich, "Role of Glucagon, Catecholamines, and Growth Hormone in Human Glucose Counterregulation," vol. 64, no. 1, pp. 62–71.

[272] J. Gerich, J. Davis, M. Lorenzi, R. Rizza, N. Bohannon, J. Karam, S. Lewis, R. Kaplan, T. Schultz, and P. Cryer, "Hormonal mechanisms of recovery from insulin-induced hypoglycemia in man," vol. 236, no. 4, pp. E380–385.

[273] K. Nonaka, F. Ono, M. Ishibashi, and N. Okita, "No Delay in Glucose Change at Antecubital Skin in Hypoglycemia of Normal Subjects," vol. 26, no. 3, pp. 957–958.

[274] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. D. III, *Process Dynamics and Control.* Wiley, 3 edition ed.

[275] M. R. Marvin, S. E. Inzucchi, and B. J. Besterman, "Computerization of the Yale Insulin Infusion Protocol and Potential Insights into Causes of Hypoglycemia with Intravenous Insulin," vol. 15, no. 3, pp. 246–252.

[276] R. Juneja, C. Roudebush, N. Kumar, A. Macy, A. Golas, D. Wall, C. Wolverton, D. Nelson, J. Carroll, and S. J. Flanders, "Utilization of a Computerized Intravenous

Insulin Infusion Program to Control Blood Glucose in the Intensive Care Unit," vol. 9, no. 3, pp. 232–240.

[277] K. R. Muske and J. B. Rawlings, "Model predictive control with linear models," vol. 39, no. 2, pp. 262–287.

[278] G. Grunberger, J. Abelseth, T. Bailey, B. Bode, Y. Handelsman, R. Hellman, L. Jovanovi, W. Lane, P. Raskin, W. Tamborlane, and C. Rothermel, "Consensus Statement by the American Association of Clinical Endocrinologists/American College of Endocrinology Insulin Pump Management Task Force," vol. 20, no. 5, pp. 463–489.

[279] "10% Dextrose Injection, USPConcentrated Dextrose in Water."

[280] C. F. Cori and G. T. Cori, "The Fate of Sugar in The Animal Bod IV. The Tolerance of Normal and Insulinized Rats for Intravenously Injected Glucose and Fructose," vol. 72, no. 2, pp. 597–614.

[281] C. V. Rao, J. B. Rawlings, and J. H. Lee, "Constrained linear state estimationa moving horizon approach," vol. 37, no. 10, pp. 1619–1628.

[282] C. Rao, J. Rawlings, and D. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," vol. 48, no. 2, pp. 246–258.

[283] D. G. Robertson, J. H. Lee, and J. B. Rawlings, "A moving horizon-based approach for least-squares estimation," vol. 42, no. 8, pp. 2209–2224.

[284] C. V. Rao and J. B. Rawlings, "Constrained process monitoring: Moving-horizon approach," vol. 48, no. 1, pp. 97–109.

[285] C. C. Qu and J. Hahn, "Computation of arrival cost for moving horizon estimation via unscented Kalman filtering," vol. 19, no. 2, pp. 358–363.

[286] Dexcom, "Dexcom Platinum G4 Continuous Glucose Monitoring System: Users Manual."

[287] E. Jones, T. Oliphant, and P. Peterson, *SciPy: Open Source Scientific Tools for Python.* 2001.

[288] G. Zhang, X. Chen, and T. Chen, "Digital redesign via the generalised bilinear transformation," vol. 82, no. 4, pp. 741–754.

[289] E. Hairer, S. P. Nrsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems.* No. 8 in Springer series in computational mathematics, Springer, 2nd rev. ed ed. OCLC: ocn620251790.

[290] A. Hindmarsh and L. Petzold, "LSODA, Ordinary Differential Equation Solver for Stiff or Non-Stiff System,"

[291] "Choosing Sample Time and Horizons - MATLAB & Simulink."

[292] F. Chee and T. Fernando, *Closed-Loop Control of Blood Glucose.* Springer Science & Business Media.

[293] L. Tarr, B. S. Oppenheimer, and R. V. Sager, "The circulation time in various clinical conditions determined by the use of sodium dehydrocholate," vol. 8, no. 6, pp. 766–786.

[294] "Welcome to GlucoStabilizer."

[295] I. G. M. Team, "IV GlucoStabilizer: How Insulin and Dextrose 50% are calculated."

[296] S. Todi and M. Bhattacharya, "Glycemic variability and outcome in critically ill," vol. 18, no. 5, pp. 285–290.

[297] L. Monnier, E. Mas, C. Ginet, F. Michel, L. Villon, J.-P. Cristol, and C. Colette, "Activation of Oxidative Stress by Acute Glucose Fluctuations Compared With Sustained Chronic Hyperglycemia in Patients With Type 2 Diabetes," vol. 295, no. 14, pp. 1681–1687.

[298] L. A. Dossett, H. Cao, N. T. Mowery, M. J. Dortch, J. M. Morris, and A. K. May, "Blood Glucose Variability Is Associated with Mortality in the Surgical Intensive Care Unit," vol. 74, no. 8, pp. 679–685.

[299] G. Meyfroidt, D. M. Keenan, X. Wang, P. J. Wouters, J. D. Veldhuis, and G. Van den Berghe, "Dynamic characteristics of blood glucose time series during the course of critical illness: Effects of intensive insulin therapy and relative association with mortality," vol. 38, no. 4, pp. 1021–1029.

[300] J. S. Krinsley, "Glycemic variability and mortality in critically ill patients: The impact of diabetes," vol. 3, no. 6, pp. 1292–1301.

[301] M. Egi, R. Bellomo, E. Stachowski, C. J. French, and G. Hart, "Variability of Blood Glucose Concentration and Short-term Mortality in Critically Ill Patients," vol. 105, no. 2, pp. 244–252.

[302] P. A. Goldberg, M. D. Siegel, R. S. Sherwin, J. I. Halickman, M. Lee, V. A. Bailey, S. L. Lee, J. D. Dziura, and S. E. Inzucchi, "Implementation of a Safe and Effective Insulin Infusion Protocol in a Medical Intensive Care Unit," vol. 27, no. 2, pp. 461–467.

[303] B. W. Bequette, "Fault Detection and Safety in Closed-Loop Artificial Pancreas Systems," vol. 8, no. 6, pp. 1204–1214.

[304] U. Klueh, Z. Liu, B. Feldman, T. P. Henning, B. Cho, T. Ouyang, and D. Kreutzer, "Metabolic biofouling of glucose sensors in vivo: Role of tissue microhemorrhages," vol. 5, no. 3, pp. 583–595.

[305] A. Koh, S. P. Nichols, and M. H. Schoenfisch, “Glucose Sensor Membranes for Mitigating the Foreign Body Response,” vol. 5, no. 5, pp. 1052–1059.

[306] S. Vashist, “Continuous Glucose Monitoring Systems: A Review,” vol. 3, no. 4, pp. 385–412.

[307] N. Baysal, F. Cameron, B. Buckingham, D. Wilson, and B. Bequette, “Detecting sensor and insulin infusion set anomalies in an artificial pancreas,” in *American Control Conference (ACC), 2013*, pp. 2929–2933.

[308] N. Baysal, F. Cameron, B. A. Buckingham, D. M. Wilson, H. P. Chase, D. M. Maahs, B. W. Bequette, f. t. I. H. C.-L. S. G. (ihcl), B. A. Buckingham, D. M. Wilson, T. Aye, P. Clinton, B. P. Harris, H. P. Chase, D. M. Maahs, R. Slover, P. Wadwa, J. Realsen, L. Messer, I. Hramiak, T. Paul, S. Tereschyn, M. Driscoll, B. W. Bequette, F. Cameron, N. Baysal, R. W. Beck, J. Lum, C. Kollman, P. Calhoun, J. Sibayan, N. M. Njeru, W. Sauer, J. Lott, J. C. Pickup, I. Hirsch, and H. Wolpert, “A Novel Method to Detect Pressure-Induced Sensor Attenuations (PISA) in an Artificial Pancreas,” vol. 8, no. 6, pp. 1091–1096.

[309] A. W. Karlin, T. T. Ly, L. Pyle, G. P. Forlenza, L. Messer, R. P. Wadwa, D. J. DeSalvo, S. L. Payne, S. Hanes, P. Clinton, D. M. Maahs, and B. Buckingham, “Duration of Infusion Set Survival in Lipohypertrophy Versus Nonlipohypertrophied Tissue in Patients with Type 1 Diabetes,” vol. 18, no. 7, pp. 429–435.

[310] M. L. Tyler, K. Asano, and M. Morari, “Application of moving horizon estimation based fault detection to cold tandem steel mill,” vol. 73, no. 5, pp. 427–438.

[311] A. Bemporad, D. Mignone, and M. Morari, “Moving horizon estimation for hybrid systems and fault detection,” in *American Control Conference, 1999. Proceedings of the 1999*, vol. 4, pp. 2471–2475 vol.4.

[312] W. K. Hamilton, “Do we monitor enough? We monitor too much,” vol. 2, no. 4, pp. 264–266.

[313] J. Phillips and J. H. Barnsteiner, “Clinical alarms: Improving efficiency and effectiveness,” vol. 28, no. 4, pp. 317–323.

[314] B. J. Drew, R. M. Califf, M. Funk, E. S. Kaufman, M. W. Krucoff, M. M. Laks, P. W. Macfarlane, C. Sommargren, S. Swiryn, and G. F. Van Hare, “AHA Scientific Statement: Practice Standards for Electrocardiographic Monitoring in Hospital Settings: An American Heart Association Scientific Statement From the Councils on Cardiovascular Nursing, Clinical Cardiology, and Cardiovascular Disease in the Young: Endorsed by the International Society of Computerized Electrocardiology and the American Association of Critical-Care Nurses,” vol. 20, no. 2, pp. 76–106.

[315] C. Atzema, M. J. Schull, B. Borgundvaag, G. R. D. Slaughter, and C. K. Lee, “ALARMED: Adverse events in Low-risk patients with chest pain Receiving contin-

uous electrocardiographic Monitoring in the Emergency Department. A pilot study," vol. 24, no. 1, pp. 62–67.

[316] K. C. Graham and M. Cvach, "Monitor Alarm Fatigue: Standardizing Use of Physiological Monitors and Decreasing Nuisance Alarms," vol. 19, no. 1, pp. 28–34.

[317] J. P. Shivers, L. Mackowiak, H. Anhalt, and H. Zisser, "Turn It Off!: Diabetes Device Alarm Fatigue Considerations for the Present and the Future," vol. 7, no. 3, pp. 789–794.

[318] T. C. Dunn, R. C. Eastman, and J. A. Tamada, "Rates of glucose change measured by blood glucose meter and the GlucoWatch Biographer during day, night, and around mealtimes," vol. 27, no. 9, pp. 2161–2165.

[319] S. D. Favero, M. Monaro, A. Facchinetti, A. Tagliavini, G. Sparacino, and C. Cobelli, "Real-time detection of Glucose Sensor and Insulin Pump Faults in an Artificial Pancreas.," vol. 47, no. 3, pp. 1941–1946.

[320] J. D. R. F. C. G. M. S. Group, "Lack of accuracy of continuous glucose sensors in healthy, nondiabetic children: Results of the Diabetes Research in Children Network (DirecNet) Accuracy Study," vol. 144, no. 6, pp. 770–775.

[321] G. Clermont, J. Bartels, R. Kumar, G. Constantine, Y. Vodovotz, and C. Chow, "In silico design of clinical trials: A method coming of age:," vol. 32, no. 10, pp. 2061–2070.

[322] M. R. Davies, H. B. Mistry, L. Hussein, C. E. Pollard, J.-P. Valentin, J. Swinton, and N. Abi-Gerges, "An in silico canine cardiac midmyocardial action potential duration model as a tool for early drug safety assessment," vol. 302, no. 7, pp. H1466–H1480.

[323] P. Hunter, T. Chapman, P. V. Coveney, B. de Bono, V. Diaz, J. Fenner, A. F. Frangi, P. Harris, R. Hose, P. Kohl, P. Lawford, K. McCormack, M. Mendes, S. Omholt, A. Quarteroni, N. Shublaq, J. Skr, K. Stroetmann, J. Tegner, S. R. Thomas, I. Tollis, I. Tsamardinos, J. H. G. M. van Beek, and M. Viceconti, "A vision and strategy for the virtual physiological human: 2012 update," vol. 3, no. 2, p. 20130004.

[324] A. G. Erdman, D. F. Keefe, and R. Schiestl, "Grand Challenge: Applying Regulatory Science and Big Data to Improve Medical Device Innovation," vol. 60, no. 3, pp. 700–706.

[325] G. E. P. Box, "Robustness in the Strategy of Scientific Model Building.."

[326] M. G. Pedersen, A. Corradin, G. M. Toffolo, and C. Cobelli, "A subcellular model of glucose-stimulated pancreatic insulin secretion," vol. 366, no. 1880, pp. 3525–3543.

[327] L. E. Fridlyand and L. H. Philipson, "Glucose sensing in the pancreatic beta cell: A computational systems analysis," vol. 7, p. 15.

[328] R. V. Overgaard, K. Jelic, M. Karlsson, J. E. Henriksen, and H. Madsen, “Mathematical Beta Cell Model for Insulin Secretion following IVGTT and OGTT,” vol. 34, no. 8, pp. 1343–1354.

[329] I. R. Sweet and F. M. Matschinsky, “Mathematical model of beta-cell glucose metabolism and insulin release. I. Glucokinase as glucosensor hypothesis,” vol. 268, no. 4, pp. E775–E788.

[330] R. Bertram, J. Previte, A. Sherman, T. A. Kinard, and L. S. Satin, “The phantom burster model for pancreatic beta-cells.,” vol. 79, no. 6, pp. 2880–2892.

[331] E. Cerasi, G. Fick, and M. Rudemo, “A Mathematical Model for the Glucose Induced Insulin Release in Man,” vol. 4, no. 4, pp. 267–278.

[332] A. R. Sedaghat, A. Sherman, and M. J. Quon, “A mathematical model of metabolic insulin signaling pathways,” vol. 283, no. 5, pp. E1084–E1101.

[333] M. Koschorreck and E. D. Gilles, “Mathematical modeling and analysis of insulin clearance in vivo,” vol. 2, p. 43.

[334] I. J. Stamper and X. Wang, “Mathematical modeling of insulin secretion and the role of glucose-dependent mobilization, docking, priming and fusion of insulin granules,” vol. 318, pp. 210–225.

[335] C. Cobelli, G. Pacini, and A. Salvan, “On a simple model of insulin secretion,” vol. 18, no. 4, pp. 457–463.

[336] T. HO, “A model-based clinically-relevant chemotherapy scheduling algorithm for anticancer agents.”

[337] P. D. Feo, G. Perriello, M. M. Ventura, P. Brunetti, F. Santeusanio, J. E. Gerich, and G. B. Bolli, “The pancreatic-adrenocortical-pituitary clamp technique for study of counterregulation in humans,” vol. 252, no. 4, pp. E565–E570.

[338] P. E. Cryer, “Glucose counterregulation: Prevention and correction of hypoglycemia in humans,” vol. 264, no. 2, pp. E149–E155.

[339] “Epinephrine Monograph for Professionals - Drugs.com.”

[340] “Norepinephrine medical facts from Drugs.com.”

[341] “Hydrocortisone Monograph for Professionals - Drugs.com.”