JCSE Journal of Computing Science and Engineering

Invited Paper

Journal of Computing Science and Engineering,
Vol. 10, No. 3, September 2016, pp. 75-84

Data-driven Adaptive Safety Monitoring Using Virtual Subjects in Medical Cyber-Physical Systems: A Glucose Control Case Study

Sanjian Chen, Oleg Sokolsky, James Weimer, and Insup Lee*
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA
{sanjian, sokolsky}@cis.upenn.edu, weimerj@seas.upenn.edu, lee@cis.upenn.edu

Abstract
Medical cyber-physical systems (MCPS) integrate sensors, actuators, and software to improve patient safety and quality of healthcare. These systems introduce major challenges to safety analysis because the patient's physiology is complex, nonlinear, unobservable, and uncertain. To cope with the challenge that unidentified physiological parameters may exhibit short-term variances in certain clinical scenarios, we propose a novel run-time predictive safety monitoring technique that leverages a maximal model coupled with online training of a computational virtual subject (CVS) set. The proposed monitor predicts safety-critical events at run-time using only clinically available measurements. We apply the technique to a surgical glucose control case study. Evaluation on retrospective real clinical data shows that the algorithm achieves 96% sensitivity with a low average false alarm rate of 0.5 false alarm per surgery.

Category: Smart and intelligent computing

Keywords: Medical cyber-physical systems; Data-driven approach; Computational Virtual Subjects; Safety monitoring; Glucose control

I. INTRODUCTION

Medical cyber-physical systems (MCPS) introduce new design and verification challenges due to their increasing complexity and life-critical nature [1]. Physiological closed-loop systems are an important class of MCPS and can improve both patient safety and effectiveness of healthcare. From a control perspective, modern healthcare involves many concurrent feedback loops in which patients are the plants. Current medical workflow relies on human caregivers to make all control decisions where patient safety can be compromised by human errors and miscalculations. An automatic controller can be used to partially close the loops by performing normal control actions when patients' conditions are stable. Caregivers can therefore concentrate on making important decisions and will be alerted by the controller in the presence of adverse events. Such closed-loop systems can ultimately improve patient safety and also the quality of care since computers are good at carrying out complicated calculations that enable individualized clinical decision support. However, physiological closed-loop systems share a few key characteristics that present major safety challenges:

- **Models contain unobservable states.** Physiological modeling involves a trade-off between accuracy and controllability. Coarse-grain models with a few state

Open Access http://dx.doi.org/10.5626/JCSE.2016.10.3.75 http://jcse.kiise.org

Received 12 September 2016; **Accepted** 13 September 2016
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

variables abstract away a lot of complicated plant dynamics, which could result in inaccuracy. On the other hand, high dimensional models capture the plant behavior at a fine-grained level but usually include many physiological state variables that are not observable by current technologies. For example, diabetes control has been extensively studied for decades. This research has resulted in a variety of physiological models. A comprehensive review on this subject can be found in [2]. The glucose physiologic models are broadly classified into two categories: 1) the coarse-grain, minimal models that describe the key dynamics using a compact set of state variables and parameters [3, 4]; 2) the fine-grain, maximal models that characterize the glucose metabolism using high-dimensional nonlinear equations and many patient-dependent parameters [2, 5-8]. The maximal models include variables such as total insulin mass in liver and insulin signals governing glucose consumption in interstitial fluids, none of which can be directly measured.

- **Unidentifiable model parameters.** Every patient responds to medications differently. The difference manifests itself as notable model parameter variances across patients. Although the parameters are subject to physiological limits and it is possible to estimate population-wide distributions, it is in general very hard or prohibitively costly to identify the parameters for each individual [2]. For example, maximal glucose models contain multiple rate parameters describing the transportation of glucose and insulin between body compartments (organs and tissues), and they can only be measured for individuals by advanced isotope marker tracking technologies in a tightly-control experimental environment, which is too resource and time consuming to perform on all patients.
- **Unidentifiable short-term physiological variance.** In certain clinical scenarios, e.g., surgical glucose control, patients can suffer from stress-induced hyperglycemia during surgeries, which can lead to elevated infection risk [9-11]. From a modeling perspective, those transient changes in the physiology can manifest as temporal fluctuations in certain physiological parameters, e.g., insulin sensitivity. Coping with such short-term changes of physiological parameters is an especially challenging problem considering that most parameters cannot be directly measured, and there currently lacks quantitative clinical understanding on how the parameters may change over short periods.

In this paper, we present a run-time safety monitoring technique that leverages physiological models to track transient changes in patient's physiology. We apply the technique to a surgical glucose control application. The monitor predicts safety-critical events by online training of a computational virtual subject (CVS) set using real-time glucose measurements. Specifically, the monitor predicts when glucose levels deviate from a specified region in a near future time window. This data driven approach and several related research thrusts are discussed in [12].

To the best of our knowledge, the idea of using CVS for data-driven adaptive safety monitoring has not been explored before. An implementation of the proposed methodology is evaluated on retrospective real patient data, and the results illustrate that our prediction algorithm achieves 96% sensitivity with an average false alarm rate of 0.5 false alarm per surgery. The proposed system is designed to provide clinical decision and support—the 96% detection rate and 0.5 false alarms per surgery are significantly better than most safety critical threshold monitors used in surgery today, e.g., pulse oximeters [13].

II. PROBLEM DESCRIPTION

In this paper, we consider the glucose control for surgical patients, who need insulin infusion for blood glucose level (BGL) regulation, either because they are diabetics or they are non-diabetics but may be at risk of surgery-induced hyperglycemia. Here are a few assumptions for surgical glucose control:

- The BGL is measured and the control input (insulin) is updated at check points separated by a fixed time interval *T*. For example, according to the Hospital of the University of Pennsylvania Cardiac ICU Intraoperative Insulin Protocol (HUP IIP), BGL should be measured and the insulin dosage should be updated every 30 minutes.
- The past BGL readings and insulin rates are documented and accessible by the safety monitor.
- Patients do not eat during surgeries, i.e., no meal disturbances need to be considered.
- Insulin is only given by intravenous (i.v.) infusion.

Fig. 1 shows the overall system architecture. BGL sensor readings are passed to both the controller (ICU protocol or other control algorithm implemented by caregivers) and the safety monitor. The controller calculates an insulin dosage based on certain algorithms. The safety monitor (details are explained in subsequent sections) predicts the range of the next BGL reading, assuming the suggested insulin dosage is given. If the predicted range is safe, then the suggested dosage is passed to the actuator (pumps). Otherwise, the safety monitor raises an alarm of possible unsafe insulin dosage and feed the information back to caregivers for further assessment.

We use the maximal glucose metabolism model that has been introduced in [6, 14], which is the model that the FDA-accepted T1DM Simulator is based on [15]. Given

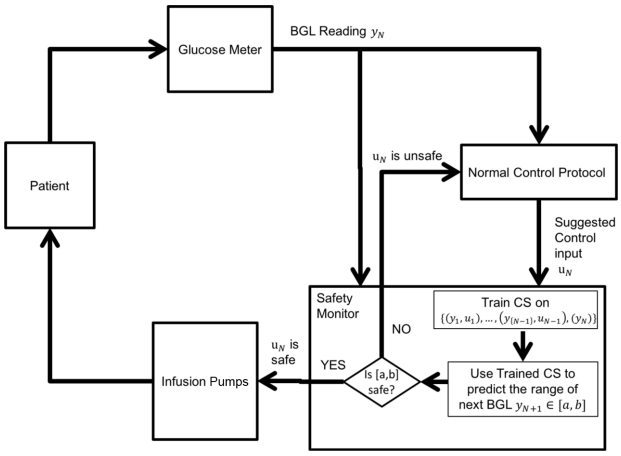


Fig. 1. Architecture of safety monitor for surgical glucose control.

Table 1. State variables

State variable	Meaning
I_p	Plasma insulin mass
X'	Remote insulin signal
I_1	Delayed insulin signal auxiliary variable
I_d	Delayed insulin signal
I_l	Liver insulin mass
G_p	Plasma glucose mass
G_t	Tissue glucose mass

Table 2. Parameters

Parameters	Meaning
m_1	Rate parameter
m_2	Rate parameter
m_3	Rate parameter
m_4	Rate parameter
k_i	Rate parameter
k_{e1}	Glomerular filtration rate
k_{e2}	Renal threshold of glucose
P_{2u}	Rate parameter
V_i	Insulin volume
V_{m0}	Model parameter
V_{mx}	Model parameter
I_b	Basal insulin level
BW	Body weight
K_{m0}	Model parameter
k_1	Rate parameter
k_2	Rate parameter
k_{p1}	Extrapolated EGP
k_{p2}	Liver glucose effectiveness
k_{p3}	Insulin action on liver
V_g	Glucose volume

the assumptions listed above, we can trim certain components from the original model (e.g., the meal absorption process and subcutaneous insulin dynamics). We rewrite the model consisting of insulin and glucose sub-systems as follows, using states and parameters listed in Tables 1 and 2. More details of the model can be found in [6, 14].

$$\dot{I}_p(t) = -(m_2 + m_4)I_p(t) + m_1 I_l(t) + u(t) * 10^2/BW \quad (1a)$$

$$\dot{X}'(t) = P_{2U}/V_i I_p(t) - P_{2U}X(t) - P_{2U} * I_b \quad (1b)$$

$$\dot{I}_1(t) = k_i/V_i I_p(t) - k_i I_1(t) \quad (1c)$$

$$\dot{I}_d(t) = k_i I_1(t) - k_i I_d(t) \quad (1d)$$

$$\dot{I}_l(t) = m_2 * I_p(t) - (m_1 + m_3)I_l(t)\,. \quad (1e)$$

The glucose system:

$$\dot{G}_p(t) = -k_1 * G_p(t) + k_2 * G_t(t) + \max(0,\ k_{p1} - k_{p2} * G_p(t) - k_{p3} * I_d(t)) - 1 - \max(0,\ k_{e1} * (G_p(t) - k_{e2})) \quad (2a)$$

$$\dot{G}_t(t) = -\frac{(V_{m0} + V_{mx} * X'(t)) * G_t(t)}{K_{m0} + G_t(t)} + k_1 * G_p(t) - k_2 * G_t(t) \quad (2b)$$

The only observable input and output to the system are u (insulin infusion rate in pmol/min) and $y = G_p/V_g$ (plasma BGL in mg/dL). The connection between the insulin and glucose subsystems is that the states X' and I_d of the insulin sub-system affect the glucose dynamics.

The glucose subsystem contains nonlinear terms: max $(0, k_{p1} - k_{p2} * Gp - k_{p3} * I_d)$ for endogenous glucose production, max $(0, k_{e1} * (Gp - k_{e2}))$ for renal glucose clearance, and $-\frac{(V_{m0} + V_{mx} * (X' - I_b)) * G_t}{K_{m0} + G_t}$ for insulin's impact on glucose consumption.

In the rest of the paper, we use X to denote the 7-dimensional state vector $X = [I_p, X', I_1, I_d, I_l, G_p, G_t]^T$, y and u to denote the model output and control input, respectively, and P to denote the 20-dimensional parameter vector (see Table 2). The complete model can be written in an abstract form $\dot{X} = f(X, P, u)$, $y = X(6)/V_g$.

A "virtual subject" is an instantiation of P. The parameters are subject to physiological limits, and their distributions, nominal values, and bounds have been studied

clinically [6, 14]. In this paper, we assume upper and lower bounds of the parameters are given, i.e., a parameter P_i is drawn from the interval $[MIN_{P_i}, MAX_{P_i}]$.

The safety property is derived from the clinical requirement, i.e., the BGL should not drop below a critical threshold. This is the most important safety requirement for BGL regulation [16].

The problem can be formulated as follows: for an individual patient, at any check point N, given the past sequences of BGL measurements and insulin rates (i.e., y_1, ..., y_{N-1} and u_1, ..., u_{N-1}, respectively), current BGL y_N, and suggested control input u_N, is u_N safe for the patient, that is, for a given u_N, is it possible that $y_{N+1} < L$?

III. SAFETY MONITOR DESIGN

In this section we present our safety monitor algorithm. The algorithm consists of two phases.

1. In Phase A, we generate a set of CVS (called the covering set [CS]) by randomly sampling parameter vectors from the bounding hypercube and we show that the CS gives predictions of BGL that cover a large possible range with certain degree of uniformity.
2. In Phase B, we use the CS for predictive safety monitoring. At each check point, we use the CS to simulate a patient's past BGL sequence and compute the prediction error of each CVS. The CVS in the CS are then ordered by how well their predictions match the true past BGL. Then we take the suggested control input and use the sorted CVS to predict the range of the patient's next BGL. If the predicted range is unsafe, then an alarm is raised to notify caregivers.

In the rest of this section, we explain the technical details of the two phases of our method.

Covering set generation and validation. As mentioned before, a fundamental challenge of using maximal models in glucose control is that most elements of the parameter vector cannot be identified given the currently available clinical data. Suppose the patient is at a certain state (X_0, y_0) and has a true parameter vector P^T, and we know nothing yet about P^T except that P^T is bounded in a 20-dimension hypercube H_P. But we know that each vector $P^k \in H_P$ will "drive" the physiological process $\dot{X} = f(X, P, u)$ from the initial state (X_0, y_0) to another state (X_1^k, y_1^k) after one sample interval T, assuming u_0 is given during the interval. The first question we consider is: if we only know that P^T is somewhere in H_P, what is the distribution of $\{y_1^k\}$ and which vectors P^k will drive y_1^k to a specific range?

This question has an intuitive interpretation in glucose control: assuming the patient's current BGL is at y_0 and he/she is given a certain amount of insulin u_0, what is the BGL after a time interval T?

The second question we consider is as follows: given a certain initial state (X_0, y_0), can we pick a set of parameter vectors P^{CS} so that the corresponding predictions $\{y_1^k\}$ "cover" the range of possible BGLs? P^{CS} is called the covering set (CS) and the vectors in it are computational virtual subjects (CVS). The first question is a CS generation problem and the second one is a CS validation problem.

We cannot use the existing physiological virtual subjects (PVS) in the T1DM Simulator as the CS for two reasons. First, the size of the PVS set is quite limited (300), and we need a much larger set of virtual subjects to generate a good covering range of BGL, which is essential for the safety monitor. Second, the PVS were derived from experimental clinical data such that they can mimic a group of real human patients, but there is no guarantee that those PVS cover all patients.

The details of the Phase A are explained as follows.

A.1 CS generation

- **A.1.1** Choosing the initial condition (X_0, y_0). The quality of CS will be related to how (X_0, y_0) is chosen. The difficulty is that the model is unobservable, i.e., for a given y_0, there are infinitely many corresponding X_0's. Inspired by the idea that an "insulin sensitivity test" tries to stabilize the patient's physiological states before the test starts, we tackle the difficulty in the following way: the academic version of T1DM Simulator includes 10 PVS that are drawn in the same way as the FDA approved PVS population, we run the 10 PVS on the simulator and obtained the initial states generated by the simulator. To fully excite the dynamics and explore a large range of $\{y_1^k\}$, we choose a high initial BGL y_0 = 250 mg/dL and let u be the "250 mg/dL BGL" action item defined in the previously mentioned HUP IIP, which is 10 U/hr infusion for T = 30 minutes and 10 U insulin bolus given immediately. We experiment with the 10 PVS' X_0's given by the T1DM Simulator and pick the one that gives the largest distribution range of $\{y_1^k\}$.
- **A.1.2** Given the (X_0, y_0, u_0) and T, we randomly sample P^{CS} from the bounding hypercube H_P (Due to the lack of accessible clinical knowledge of how the physiological parameters are correlated statistically, the sampling process does not assume any prior correlation between different parameters. One of the current limitations of this work is the lack of statistical guarantee that the sampled virtual population sufficiently represent all real patients), simulate the model on each vector $P^k \in CS$, and get the corresponding y_1^k after T. We collect a large set (1.5 million in our implementation) of sample vectors P^k so that the distribution of $\{y_1^i\}$ covers a sufficiently large range. The sufficiency can be justified by clini-

cal knowledge of how fast human BGL can drop: the BGL decline/rise rate is subject to physiological limits, e.g., it is written in HUP Glycemic Control Protocol for cardiac surgical patients that "the average rate of decline should be no more than 50 mg/dL per hour".

- **A.1.3** From the large set of randomly sampled $\{P^k\}$ and the corresponding $\{y_1^k\}$, we select a subset of $\{P^k\}$ as the CS, such that the corresponding $\{y_1^{CS}\}$ uniformly cover the entire range of $\{y_1^k\}$. The CS is not constrained by clinical data and can be much larger than the FDA-accepted 300 PVS (in the implementation we have 10,000 CVS in the CS).

A.2 CS validation

In Step A.1.3, the CS is selected to uniformly cover $\{y_1^k\}$ given the (X_0, y_0, u_0) picked in Step A.1.1 We cannot simply conjecture that CS will generate a uniformly distributed predictions of y_{N+1} for any previous state and input tuple (X_N, y_N, u_N). Therefore, we need to validate CS for different tuples (X_N, y_N, u_N) and test the coverage of predicted $\{y_{N+1}\}$. The validation algorithm works as follows:

- **A.2.1** First, we generate a set of test cases to validate the CS. Each test case is a four-value tuple (X_N, y_N, u_N, y_{N+1}). Ideally, such test cases should be from real clinical data, but as mentioned before, it is impossible to fully measure X_N directly from patients. So instead, we run a real hospital protocol (the HUP IIP) on the T1DM Simulator (with its 10 PVS) starting at different initial conditions, and we obtain a large set of simulated patient BGL trajectories.
- **A.2.2** We then test the entire CS on every single test case obtained in the previous step. Specifically, for each test case (X_N, y_N, u_N, y_{N+1}), we simulate the model on every $P^k \in P^{CS}$ for one sample interval, starting from (X_N, y_N) and using u_N as the control input. At the end of the interval, we get a y_{N+1}^k which is the prediction of the true value y_{N+1}. We then measure the quality of coverage of all the predictions $\{y_{N+1}^k\}$ by two criteria: 1) $\{y_{N+1}^k\}$ should contain y_{N+1}, i.e., $y_{N+1} \in [\min\{y_{N+1}^k\}, \max\{y_{N+1}^k\}]$; 2) $\{y_{N+1}^k\}$ is distributed around y_{N+1} with a certain degree of uniformity, which will be defined in the next step.
- **A.2.3** In Step A.1.3, the CS is selected such that the $\{y_1^k\}$ has a uniform distribution. However, due to the nonlinear nature of the model, it should not be expected that the same CS will generate $\{y_{N+1}^k\}$ that have the same near-uniform distribution starting from any initial state (X_N, y_N, u_N, y_{N+1}). In addition, for the safety prediction purpose, we do not need a perfectly uniform distribution. Instead, what we need is that there are enough candidate predictions in $\{y_{N+1}^k\}$ that fill a neighboring region of y_{N+1}. The uniformity metric we use to test the coverage of $\{y_{N+1}^k\}$ is therefore defined as follows:
  - We first determine the neighboring region of y_{N+1}. Again, the size of the region depends on the size of the maximum prediction range in the control step, which is set to 60 mg/dL, i.e., the [max $(y_{N+1} - 30, 30)$, $y_{N+1} + 30$] neighboring region. The lower bound saturates at 30 because BGL < 30 mg/dL is clinically considered to be life-threatening (extreme hypoglycemia). Also, 30 mg/dL is significantly lower than the commonly-used hypoglycemia alarming threshold, which is typically in the range of 70–110 mg/dL.
  - Next, we test the minimum density of predicted BGL values in $\{y_{N+1}^k\}$ that fall into the neighboring region. The density is defined as the average number of values in $\{y_{N+1}^k\}$ that fall into a unit length (1 mg/dL) BGL interval. The density is computed by a binning algorithm: put the values in $\{y_{N+1}^k\}$ that are in the neighboring region into small-sized (5 mg/dL) bins and find a bin with the minimum counts to calculate the minimum density. If the "density" is no less than 1 counts per mg/dL, then the CS passes the coverage test on the case (X_N, y_N, u_N, y_{N+1}). For extra redundancy, our algorithm actually achieves minimum density of 8 counts per mg/dL.

Data-driven adaptive safety monitoring. The Phase B is repeated at each check point. We use the identified and validated CS to predict a patient's BGL one sample interval ahead, given only the past BGL readings and insulin inputs. This is a challenging problem. Existing model-predictive control (MPC) approaches for glucose control either use a simple linear model to approximate the nonlinear dynamics, or require parameter pre-tuning. It is known that there is a fundamental analytical limitation of parameter identification on high-dimension unobservable nonlinear glucose models: it is not possible to uniquely identify all unknown parameters given only the limited single input and single output data [2]. Instead of directly identifying P^T, we propose a data driven technique to adaptively train CS on past sequence (y, u) (the past sequence gets updated each step as new measurements and control actions are taken), and then use the trained CS to predict the range of next BGL reading. The predicted range is then used for the safety monitoring.

B.1 CS Training

The training set is a sequence of past BGL and control actions, $\{(y_i, u_i), ..., (y_{N-1}, u_{N-1}), (y_N)\}$. N is the current check point and y_N is the current BGL. The training set contains the latest $N - i + 1$ BGL and $N - i$ control actions (a control action is effective for an interval between two y's so there is one less control action than the number of y's). Initially $i = 1$. These records are accessible in ICUs, e.g., at HUP, the BGL readings, control information, and patient-related information are recorded electronically. The CS is trained as follows:

• **B.1.1** We simulate the model on the CS, starting from y_i. Each $P^k \in P_{CS}$ generates a corresponding simulated trace $\{y_i^k, ..., y_N^k\}$.

• **B.1.2** The simulated traces are compared to the true trace $\{y_i, \ldots, y_N\}$, and L-2 norm errors are calculated for each trace.

• **B.1.3** We sort the CS by the non-decreasing order of the prediction errors.

B.2 Range prediction

To predict the range of the next BGL y_{N+1}, we initialize a set R_{N+1} as empty.

• **B.2.1** We start from the top of the trained CS, retrieve a vector P^k, extend the simulation $y_i^k, ..., y_N^k$ by one step (assuming the suggested control input u_N is given at check point N), put the predicted y_{N+1}^k into R_{N+1}, move onto the next vector in the trained CS, and repeat the process.

• **B.2.2** R_{N+1} holds predictions of y_{N+1} of a top subset of CVS in the trained CS. The minimum and maximal values of R_{N+1} are the predicted range for y_{N+1}. We put predictions y_{N+1}^k into R_{N+1} until at least one of the following two stopping conditions becomes true:

- The predicted range exceeds a pre-defined window size. This window size directly affects the R_{N+1} and the performance of the prediction algorithm. We develop a double-zone strategy to determine it. If $\min(R_{N+1}) > W_b$, i.e., $\min(R_{N+1})$ is above some threshold W_b, then the stop condition is

$$(\max(R_{N+1}) - \min(R_{N+1})) > W_H,$$

where W_H is the maximum window size of the "high" zone. If

$$\min(R_{N+1}) \le W_b,$$

then the stop condition is

$$(\max(R_{N+1}) - \min(R_{N+1})) > W_L,$$

where W_L is the maximum window size of the "low" zone. The idea is that when the predicted BGL in R_{N+1} are relatively high (above W_b), we allow a larger prediction window. And when some predicted BGL in R_{N+1} are in the low zone, we narrow down the prediction window because now the predicted BGL are closer to the unsafe region and a narrower window can reduce the false positive rate.
- The bottom of the trained CS is reached; i.e., all predictions by the CS have been put into R_{N+1}.

The predicted range of y_{N+1} is given by $[\min(R_{N+1}), \max(R_{N+1})]$.

B.3 Robust safety monitoring

Using the predicted range

$$[\min(R_{N+1}), \max(R_{N+1})],$$

if $\min(R_{N+1})$ is less than the pre-defined safety limit L, then it implies the suggested control input u_N may drive y_{N+1} into the unsafe region, and an alarm is to be raised and fed back to caregivers. This is currently a binary classification alarm, but one can also use $[\min(R_{N+1}), \max(R_{N+1})]$ to generate a more informative, fuzzy-logic alarms: for example, generating alarms with different levels of urgency depending on how much $[\min(R_{N+1}), \max(R_{N+1})]$ intersects with the unsafe region.

B.4 Adaptive training window adjustment

The training sequence grows as more BGL readings are collected. Sometimes the real BGL trajectory could exhibit unpredicted fluctuations (In the patient data, we have seen some BGL changes that cannot be well explained by models: for example, when the BGL tends to stabilize around a certain level and insulin infusion rate does not change for a while, there are sometimes sudden BGL increases, i.e., the patient appears to be more resistant to insulin for a short period. Anesthesiologists at HUP are interested in such scenario when we replayed the retrospect BGL data and specifically pointed out a few such "turns". Their medical opinions are that there are some other factors, e.g., body temperature and surgery-induced stress level, that they believe also change patients' insulin sensitivity, but those effects are not modeled by even the state-of-the-art maximal models. The doctors think it would be useful if we can compare the model-predicted BGL with the true BGL in real time and alert them when such "turns" happen (which can be done with our framework) as they believe these events could suggest some physiological state changes that they concern.). An interesting phenomenon is that the unmodeled dynamics not only causes prediction errors at the "turns", but also affects subsequent predictions after the "turns" even when the patient's physiological states stabilize again. This is because in the learning step, the CS is trained by prediction errors calculated over the entire training window. If there are unpredicted "turns" in the window and we keep them for future predictions, then the CS will always try to learn the "turns" because they dominate the prediction error.

To cope with such cascading error issue, we dynamically adjust the training window in real time. Whenever a true reading y_{N+1} lands outside the predicted range, we remove all past readings before y_N and the new training sequence starts from y_N for future predictions (we need to keep y_N because we need at least one interval, i.e., two past readings, for training when predicting y_{N+2}). The rationality is that a BGL "turn" indicates a possible change in the patient's physiological parameters, so the safety monitor should also be reset to match such change, instead of keeping old past readings and still trying to learn the "turns" starting from old initial states.

IV. EVALUATION

We implement the complete safety monitoring algorithm and evaluate it on de-identified, retrospective patient glucose data collected from the Hospital of the University of Pennsylvania (with the Institutional Review Board approval). In this section, we present the results of the implementation and the evaluation using real patient data. We also discuss several design trade-offs regarding how to configure our algorithm according to different clinical needs.

CS generation and validation. Starting from y_0 = 250 mg/dL, u_0 = 10 U/hr infusion plus 10 U insulin bolus and setting the sampling interval T to be 30 minutes, we explore the distribution of predicted $\{y_1^k\}$ given randomly sampled CVS. To fully explore the possible distribution of $\{y_1^k\}$, we sample 1,500,000 CVS in the bounding hypercube HP and the distribution of $\{y_1^k\}$ is shown in Fig. 2.

The distribution covers a large range from 20 mg/dL to as high as 400 mg/dL. Hospital protocols consider 50 mg/dL per hour a dangerously high BGL decline rate. The lowest predicted BGL after 30 minutes is 20 mg/dL, which translates into a 460 mg/dL per hour drop rate, more than 9 times larger than what the protocols consider dangerous. On the highest end, the highest predicted BGL is around 400 mg/dL. Clinicians consider 10 U/hr to be a very high insulin dose, and it is very unlikely that a patient's BGL can even increase from 250 mg/dL to 400 mg/dL in 30 minutes given such a high insulin rate. The simulated $\{y_1\}$ clearly covers a sufficiently large BGL range. From the 1.5 million candidate CVS, we select 10,000 CVS into the CS such that the $\{y_1\}$ coverage of the CS is uniformly distributed, as shown in Fig. 3.

To validate the coverage of $\{y_{N+1}^k\}$ produced by the CS given any starting state, we extract test cases from the simulated BGL trajectories that are obtained by running the T1DM Simulator together with the HUP IIP protocol controller at different initial conditions. We simulate the 10 PVS included in the T1DM simulator starting from 18 different initial BGLs ranging from 70 mg/dL to 240 mg/dL (sampled every 10 mg/dL) and obtain 180 simulated BGL trajectories. A test case is extracted at each 30 minutes check point of a trajectory, and for each trajectory we extract test cases from the first 24 check points (time 0 to 12 hour), because after 12 hours the simulated BGL trajectories are oscillating around an equilibrium (the initial transient response fades away). Therefore we get 4,320 (10*18*24) test cases. For each test case, we calculate the minimum density of CVS in the neighboring window of y_{N+1}.

Fig. 4 shows the distribution of the minimum density values of CVS in all test cases. The overall minimum density value in all 4,320 test cases is 8.6 counts per mg/dL, which is greater than the required 1 per mg/dL, i.e., the generated CS passes all 4,320 coverage tests.

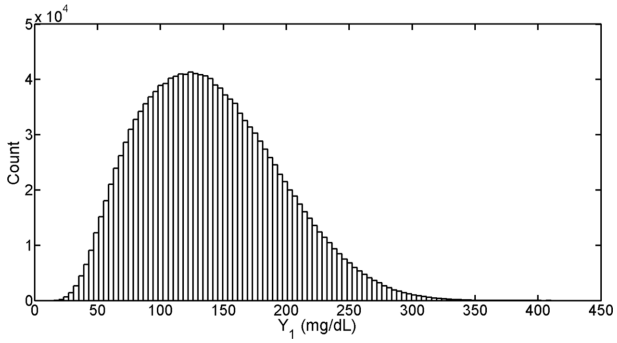


Fig. 2. Distribution of 1,500,000 simulated y_1's.

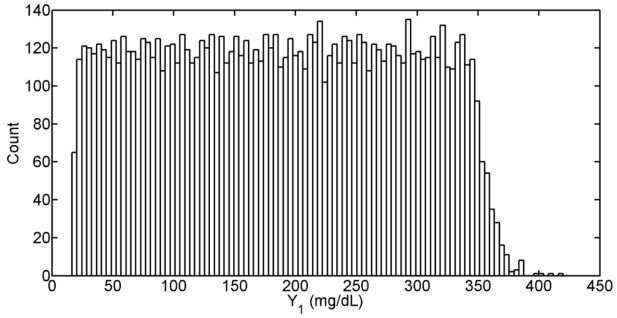


Fig. 3. Distribution of the simulated y_1's of the 10,000 CVS chosen by CS.

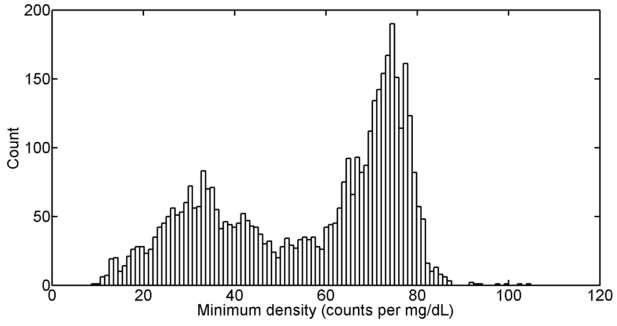


Fig. 4. Distribution of minimum density of CVS in 4,320 test cases.

Safety monitor evaluation. We use the generated CS to test our safety monitor. For safety and regulatory reasons, we cannot directly test this newly designed monitor technique on human patients without extensive offline experiments. We demonstrate the validity of our approach by replaying retrospective patient BGL data on the safety monitor, which, from a computational perspective, is the same as if the safety monitor is tested in the real clinical environment. The data is collected from 51 de-identified type 1 diabetic patients who received cardiac surgery and were on the HUP IIP (so insulin inputs are known). As defined by the protocol, the BGL readings were taken

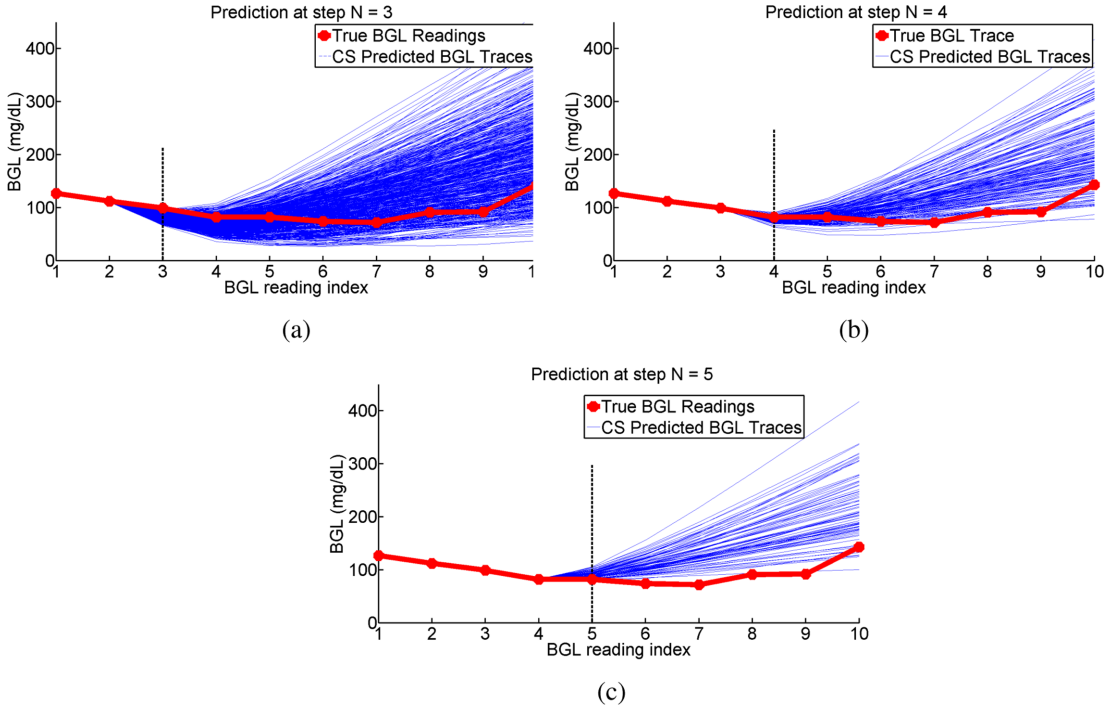


Fig. 5. Prediction snapshots at (a) $N = 3$, (b) $N = 4$, and (c) $N = 5$ on a sample patient.

every 30 minutes.

The evaluation algorithm works as follows. We retroactively run the HUP IIP and our safety monitor on the real BGL data. At each real BGL reading, the safety monitor computes the range of the next BGL reading given the control input determined by the HUP IIP and predicts whether or not the next BGL reading will be safe. Then we move on to the next BGL reading, check it against the predicted range, and repeat the process.

CS training and range prediction. For each patient, we start prediction on y_3 (y_1 and y_2 are used as the initial training set) and proceed until the end of each data trace. Fig. 5 illustrates how the adaptive learning algorithm works on a patient's data trace. At each check point, the algorithm predicts y_{N+1} one interval ahead by picking those CVS that achieve lowest prediction errors on the past sequence (y_i, …, y_N). As shown in Fig. 5, our algorithm adaptively tracks the true BGL trend. At $N = 3$ there are only 2 history readings and 1 interval in the training set. The training set is so small at that point that it cannot fully separate the large CS. That is why the predicted values of y_3 cluster around the true y_3 but the BGL trajectories are further apart in the future. The algorithm only needs to predict one step ahead at a time, so future divergences are irrelevant (as Fig. 5 shows, the predicted range converges as the monitor moves to $N = 4$ and 5).

We test the monitor on 51 patients' data. Overall there are 246 BGL readings, among which 144 BGL readings can be used to test the prediction algorithm (the first two readings of each patient are needed for initial training and cannot be predicted by the monitor). The performance of the safety monitoring algorithm can be tuned by setting the threshold L and window sizes differently. In practice, those parameters should be set according to clinical needs. According to the IIP, clinicians consider BGLs less than 60 mg/dL as critical condition and start to take precautions when the BGL drops below 100 mg/dL. Therefore, we set the alarming threshold L to be 100 mg/dL. Table 3 reports the performance matrices of the monitor (the parameters are set as follows: $W_b = 110$, $W_H = 60$, and $W_L = 30$). The result shows that the monitor achieves 96% sensitivity (24 out of 25 unsafe events are correctly identified) with less than 0.5 false alarms per surgery on average (24 false alarms on 51 patients and each patient's data is collected during one surgery).

Table 3. Evaluation results on 51 patients' data (144 prediction points) when L is set to 100 mg/dL

L = 100		
	Safe (true)	**Unsafe (true)**
Safe (predicted)	95	1
Unsafe (predicted)	24	24

V. CONCLUSION

In this paper, we have developed a data-driven safety

monitoring technique to adaptively track real-time physiological changes and predictively alert caregivers to future critical events. We applied the technique to a surgical glucose control application and developed a novel CVS-based approach for robust safety monitoring. Evaluation on a clinical dataset shows that the proposed safety monitor achieves high sensitivity with a low false alarm rate.

ACKNOWLEDGMENTS

This research was supported in part by NSF CNS-1035715. We would like to thank Benjamin Kohl, M.D. (Anesthesiologist) and Margaret Mullen-Fortino, R.N. (Operations Director) at HUP for advising us with expert knowledge, providing data, and helping with clinical interpretation of our results. We are also grateful to Andrew King and Alex Roederer whose insightful suggestions have helped to improve the paper significantly.

REFERENCES

1. I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, et al., "Challenges and research directions in medical cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 75-90, 2012.
2. C. Cobelli, C. Dalla Man, G. Sparacino, L. Magni, G. De Nicolao, and B. P. Kovatchev, "Diabetes: models, signals, and control," *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 54-96, 2009.
3. P. A. Insel, J. E. Liljenquist, J. D. Tobin, R. S. Sherwin, P. Watkins, R. Andres, and M. Berman, "*Insulin control of glucose metabolism in man: a new kinetic analysis*," *Journal of Clinical Investigation*, vol. 55, no. 5, pp. 1057-1066, 1975.
4. C. Cobelli, G. Toffolo, and E. Ferrannini, "A model of glucose kinetics and their control by insulin, compartmental and noncompartmental approaches," *Mathematical Biosciences*, vol. 72, no. 2, pp. 291-315, 1984.
5. R. Hovorka, L. J. Chassin, M. Ellmerer, J. Plank, and M. E. Wilinska, "A simulation model of glucose regulation in the critically ill," *Physiological Measurement*, vol. 29, no. 8, pp. 959-978, 2008.
6. C. Dalla Man, R. A. Rizza, and C. Cobelli, "Meal simulation model of the glucose-insulin system," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 10, pp. 1740-1749, 2007.
7. B. P. Kovatchev, M. Breton, C. Dalla Man, and C. Cobelli, "In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes," *Journal of Diabetes Science and Technology*, vol. 3, no. 1, pp. 44-55, 2009.
8. D. Bruttomesso, A. Farret, S. Costa, M. C. Marescotti, M. Vettore, A. Avogaro, et al., "Closed-loop artificial pancreas using subcutaneous glucose sensing and insulin delivery and a model predictive control algorithm: preliminary studies in Padova and Montpellier," *Journal of Diabetes Science and Technology*, vol. 3, no. 5, pp. 1014-1021, 2009.
9. K. C. McCowen, A. Malhotra, and B. R. Bistrian, "Stress-induced hyperglycemia," *Critical Care Clinics*, vol. 17, no. 1, pp. 107-124, 2001.
10. J. E. Richards, R. M. Kauffmann, W. T. Obremskey, and A. K. May, "Stress-induced hyperglycemia as a risk factor for surgical-site infection in non-diabetic orthopaedic trauma patients admitted to the intensive care unit," *Journal of Orthopaedic Trauma*, vol. 27, no. 1, pp. 16-21, 2013.
11. M. A. Karunakar and K. S. Staples, "Does stress-induced hyperglycemia increase the risk of perioperative infectious complications in orthopaedic trauma patients?" *Journal of Orthopaedic Trauma*, vol. 24, no. 12, pp. 752-756, 2010.
12. S. Chen, "Model-based analysis of user behaviors in medical cyber-physical systems," Ph.D. dissertation, University of Pennsylvania, PA, 2016.
13. S. Nizami, K. Greenwood, N. Barrowman, and J. Harrold, "Performance evaluation of new-generation pulse oximeters in the NICU: observational study," *Cardiovascular Engineering and Technology*, vol. 6, no. 3, pp. 383-391, 2015.
14. 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," *Journal of Diabetes Science and Technology*, vol. 1, no. 6, pp. 804-812, 2007.
15. T1DMS: type 1 diabetes metabolic simulator, http://tegvirginia.com/solutions/t1dms/.
16. P. E. Cryer, "Hypoglycemia is the limiting factor in the management of diabetes," *Diabetes/Metabolism Research and Reviews*, vol. 15, no. 1, pp. 42-46, 1999.

Sanjian Chen

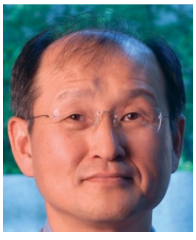
Sanjian Chen received his Ph.D. in Computer and Information Science from the University of Pennsylvania. His research interests include data-driven modeling, machine learning, and decision-support systems. He received the Best Paper Award at the 2012 IEEE Real-Time Systems Symposium (RTSS).

Oleg Sokolsky

Oleg Sokolsky is a Research Professor of Computer and Information Science at the University of Pennsylvania. His research interests include the application of formal methods to the development of cyber-physical systems, architecture modeling and analysis, specification-based monitoring, as well as software safety certification. He received his Ph.D. in Computer Science from Stony Brook University.

James Weimer

James Weimer is a Research Assistant Professor of Computer and Information Science at the University of Pennsylvania. His research interests include the design and analysis of cyber-physical systems with application to medical devices/monitors, networked systems, building energy management, and security. He received his Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University.



Insup Lee

Insup Lee is Cecilia Fitler Moore Professor of Computer and Information Science and Director of PRECISE Center at the University of Pennsylvania. He also holds a secondary appointment in the Department of Electrical and Systems Engineering. He received the B.S. in Mathematics with Honors from the University of North Carolina, Chapel Hill and the Ph.D. in Computer Science from the University of Wisconsin, Madison. His research interests include cyber-physical systems (CPS), real-time systems, embedded systems, high-confidence medical device systems, formal methods and tools, run-time verification, software certification, and trust management. The theme of his research activities has been to assure and improve the correctness, safety, and timeliness of life-critical embedded systems. His papers received the five best paper awards in conferences. Recently, he has been working in medical cyber-physical systems and security of cyber physical systems. He has served on many program committees, chaired many international conferences and workshops and served on various steering and advisory committees of technical societies. He has also served on the editorial boards on the several scientific journals, including Journal of ACM, ACM Transactions on Cyber-Physical Systems, IEEE Transactions on Computers, Formal Methods in System Design, and Real-Time Systems Journal. He is Chair of ACM SIGBED (2015-2017) and was Chair of IEEE TC-RTS (2003-2004). He was a member of Technical Advisory Group (TAG) of President's Council of Advisors on Science and Technology (PCAST) Networking and Information Technology (2006-2007). He is a member of the National Research Council's committee on 21st Century Cyber-Physical Systems Education (2014-2015). He received an appreciation award from Ministry of Science, IT and Future Planning, South Korea in 2013. He is IEEE fellow and received IEEE TC-RTS Outstanding Technical Achievement and Leadership Award in 2008.