Towards a Verified Artificial Pancreas: Challenges and Solutions for Runtime Verification.

Fraser Cameron[1], Georgios Fainekos[2], David M. Maahs[3] and Sriram Sankaranarayanan[4]

[1] Department of Mechanical Engineering, University of Texas, El Paso.
[2] School of Computing, Informatics and Decision Systems Engg., Arizona State Univ., Tempe.
[3] Barbara Davis Center for Childhood Diabetes, University of Colorado, Denver.
[4] Department of Computer Science, University of Colorado, Boulder.

Abstract. In this paper, we briefly examine the recent developments in artificial pancreas controllers, that automate the delivery of insulin to patients with type-1 diabetes. We argue the need for offline and online runtime verification for these devices, and discuss challenges that make verification hard. Next, we examine a promising simulation-based falsification approach based on robustness semantics of temporal logics. These ideas are implemented in the tool S-Taliro that automatically searches for violations of metric temporal logic (MTL) requirements for Simulink(tm)/Stateflow(tm) models. We illustrate the use of S-Taliro for finding interesting property violations in a PID-based hybrid closed loop control system.

1 Introduction: Artificial Pancreas

Type-1 Diabetes (T1D) is a chronic condition caused by the inability of the pancreas to secrete insulin, a hormone that is critical to maintaining blood glucose levels inside a tight *euglycemic* range [42,59]. The standard treatment for T1D consists of delivering insulin externally through injections, or more recently, through insulin pumps that deliver short acting artificial insulin analog, sub-cutaneously. Insulin pumps provide many features, including the accurate delivery of insulin at varying rates over time. However, insulin pumps are controlled manually by the patient, who is ultimately responsible for increasing insulin delivery at meal times (meal bolus), or decreasing/disabling insulin delivery during physical activity [11]. The manual control of insulin delivery poses a heavy burden on the patients themselves, is error-prone and can sometimes lead to dangerous outcomes [57]. Too much insulin causes a dangerous drop in blood glucose levels (hypoglycemia), whereas too little insulin causes the blood glucose levels to remain high (hyperglycemia), resulting in long term damage to organs such as the kidneys, eye and peripheral nerves.

The artificial pancreas (AP) project envisions a series of increasingly sophisticated control systems to automate the delivery of insulin to patients with T1D. At it's core, the AP system combines a continuous glucose monitor (CGM) which senses blood glucose levels periodically, and an insulin pump that delivers insulin in a closed loop managed by a software-based controller. Table 1 shows the original stage wise development for the overall AP concept. A recently revised pathway acknowledges that all stages are

Table 1: Original pathway to the artificial pancreas project with representative papers showing technological feasibility. Source: Juvenile Diabetes Research Foundation (JDRF). See [39] for a recently proposed revised pathway.

ID	Description	Refs.
1	Very Low Glucose Pump Shutoff	[48]
	Pump shutoff during hypoglycemia	
2	Hypoglycemia Minimizer	[10]
	Pump shutoff in advance of predicted future hypoglycemia	
3	Hypo/Hyperglycemia Minimizer	[4,50,32]
	Same as # 2 plus addition of insulin when glucose is above threshold	
4	Hybrid Closed Loop	[35,34,33]
	Closed loop insulin delivery with manual bolus	
5	Fully Autoamted Closed Loop	[7,8,9,15,44,38,19]
	#4 with all manual meal boluses eliminated	
6	Multihormone Closed Loop	[26,25]
	Use glucagon and insulin to achieve bidirectional control	

currently technologically feasible and classifies insulin delivery beyond stage 3 simply as "insulin-only" control and "multihormonal" control [39]. The first (and simplest) stage simply shuts off the pump when the blood glucose level is sensed below a widely accepted threshold for hypoglycemia. Further improvements add the ability to forecast future trends of the blood glucose and perform predictive pump shutoff, introduce extra insulin when blood glucose levels are high, predict the onset of meals and finally a fully closed loop that is expected to completely eliminate the need for manual control of insulin infusions.

The AP project promises a drastically improved approach to treating T1D by improving glucose control and reducing the burden of care to the patient. However, it's use potentially presents numerous risks to the patient. Too much insulin delivered to the patient can drive their blood glucose levels too low, causing seizures, coma or even death [6]. At the other end, a failure to deliver adequate insulin to cover meals can result in too high blood glucose levels that can lead to near-term complications such as ketacidosis. In order to be successful, the AP controller must tolerate significant sensor noise, and unpredictable events such as meals, physical activity and pump/sensor failures [16,36]. Furthermore, software errors in the controller software can have frightening and unexpected consequences [29].

Since it's inception in 2001, the *Runtime Verification* (RV) community has pioneered numerous techniques in efficient monitoring of temporal requirements of systems both during deployment (online monitoring) and development (offline monitoring). Progress in AP controllers bring about two important classes of challenges to the larger verification community, and specifically to the runtime verification community:

1. AP controllers have large state-spaces, a rich set of behaviors and are subject to large disturbances such as meals, exercise, sensor and infusion set failures, that makes these systems hard to reason with for existing symbolic methods. We examine the use of *simulation-based verification* techniques, particularly for the artificial pancreas controllers.

2. Beyond offline monitoring, it is also necessary to perform online monitoring of deployed artificial pancreas control systems to detect failures, caused due to rare events that may be hard to observe in clinical trials. In fact, the idea of robustness of a trace with respect to a logical specification can also be used to perform online monitoring for detecting potential failures early [21,23].

In this paper, we focus mainly on offline monitoring and illustrate how simulation-based falsification techniques can be employed as a first step towards verified artificial pancreas controllers.

2 Simulation-Based Falsification

In this section, we briefly survey simulation-based falsification approaches. We focus primarily on *robustness-guided falsification*, a promising approach that combines the notion of robustness of temporal logic formulas with stochastic optimization techniques for automatically search for falsifying traces.

2.1 Simulation-Based Falsification

Model-based falsification techniques for cyber-physical systems (CPS) seek behaviors of a system that violate a given property φ of interest. Falsification techniques can be *symbolic*, exploring the system behavior using a constraint solver [5], or *numeric*, using numerical simulations of the model to find property violations. In practice, significant strides in symbolic falsification have been made towards faster constraint solvers that support richer logics [20]. Nevertheless, the state-of-the-art for symbolic model checking techniques are currently restricted to linear models that involve controllers with linear assignments/conditionals and plants with linear dynamics [30]. Symbolic model checkers for nonlinear models and nonlinear controllers are currently a topic of ongoing research [13,14,31]. However, significant algorithmic challenges currently limit the scalability of these approaches. Furthermore, the use of these techniques on nonlinear, software-based control system with nonlinear plant models expressed in popular frameworks such as Simulink(tm)/Stateflow(tm) in Matlab(tm) requires a significant tool building effort.

Therefore, in this exposition, we focus on simulation-based approaches. Broadly, a simulation-based falsification technique performs repeated simulations of the system under various inputs and initial conditions, using results of past simulations to guide the future inputs to the system. Simulation-based falsification techniques offer two main features: (a) They are able to handle the system itself as a *black box*. This is an enormous advantage when the model is specified in a widely-used formalism such as Simulink(tm)/Stateflow(tm). Simulink/Stateflow models have complex semantics that change substantially over successive versions of the Matlab(tm) framework. On the other hand, the absence of detailed system knowledge is a drawback: repeated simulations are well-known to be inadequate for exploring systems with large state-spaces. As a result, simulation-based falsification techniques typically have very weak mathematical guarantees. (b) Simulation is cheap, parallelizable and can be performed quite

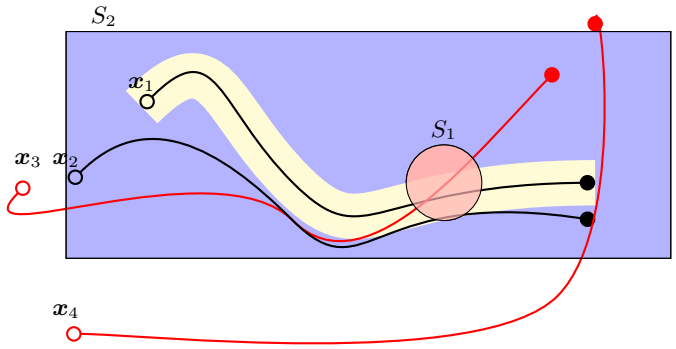


Fig. 1: Trace robustness for temporal property $\varphi : \Diamond S_1 \wedge \Box S_2$. Traces $\boldsymbol{x}_1, \boldsymbol{x}_2$ both satisfy the property: $\rho(\boldsymbol{x}_1, \varphi) > \rho(\boldsymbol{x}_2, \varphi) > 0$. Likewise, $\boldsymbol{x}_3, \boldsymbol{x}_4$ both violate the property: $\rho(\boldsymbol{x}_4, \varphi) < \rho(\boldsymbol{x}_3, \varphi) < 0$. The robustness cylinder for $\boldsymbol{x}_1$ is highlighted.

accurately even for large nonlinear systems that are beyond the reach of many symbolic tools. However, numerical simulation tools are approximate: the simulation trajectories may deviate from the actual system trajectories due to integration and floating-point errors. Many simulation-based approaches have been proposed, especially for falsifying properties of CPS. We restrict our discussion below to two main approaches: (a) Rapid Exploration of Random Trees (RRTs) and (b) Robustness-Guided Falsification.

RRTs explore the behaviors of the system by building a tree whose nodes are system states and edges are trajectories connecting these states [41]. At each step, the tree is grown towards a current target state through a local search technique. Many variants of the basic RRT approach have been explored, some specifically designed for the falsification of temporal logic properties of CPS [54,53,18,55,24]. Recently, the RRT approach has been increasingly successful on larger benchmarks [24]. However, the performance can be quite variable, depending on the specific RRT scheme used. Furthermore, the techniques are also quite sensitive to the choice of distance metrics. Finally, the practical application of RRTs, specifically to Simulink(tm) /Stateflow(tm) models is currently challenging due to the large costs of setting up a simulation run. This is disadvantageous since the standard RRT approach relies on numerous simulations over a short time period for the local search.

In contrast, robustness-guided approaches are based on two main ingredients: (a) First, the notion of temporal property satisfaction is extended to allow us to have a distance metric to property violation [27,56,23]. Such a metric is referred to as the "trace robustness". Intuitively, a trace with a smaller robustness is therefore "closer" to a violation when compared to a trace that has a larger robustness. (b) In turn, the robustness metric can be used as an objective function to guide the system towards property violations in a systematic manner by seeking trajectories of ever decreasing robustness [49,1,3]. This is usually achieved inside a global optimization technique such as Nelder-Mead, simulated annealing, ant-colony optimization or the cross-entropy method that uses the robustness as an objective function to minimize.

We will now briefly outline the robustness-guided approach for falsifying Metric Temporal Logic (MTL) properties of systems [40], following the work of Fainekos and Pappas [27]. The TaLiRo tool implements the MTL monitoring algorithm inside Matlab(tm). The ideas presented are conceptually similar to those of Donzé and Maler, using the alternative formalism of Signal Temporal Logic (STL) [23]. This is implemented in the Breach tool [22]. As mentioned earlier, the notion of robustness extends the standard Boolean notion of property satisfaction of a trace (i.e, a trace either satisfies a property or it does not) to a real-valued notion. Let $\boldsymbol{x} : \mathbb{R}_{\geq 0} \rightarrow X$ be a trajectory, mapping time $t \geq 0$ to state $\boldsymbol{x}(t) \in X$ and φ be a MTL property.

Definition 1 (Robustness Metric). *The* robustness *of $\boldsymbol{x}(\cdot)$ w.r.t φ is a real number $\rho(\boldsymbol{x}, \varphi)$ that has the following properties: (a) $\rho(\boldsymbol{x}, \varphi) > 0$ if $x \models \varphi$, and (b) $\rho(\boldsymbol{x}, \varphi) < 0$ if $x \not\models \varphi$. Furthermore, the magnitude $v : |\rho(\boldsymbol{x}, \varphi)|$ denotes the maximum radius of a* cylinder *around the trace $\boldsymbol{x}$ so that any other trace in the cylinder also has the same outcome for φ as $\boldsymbol{x}$.*

Example 1. Figure 1 illustrates robustness using the property $\varphi : \Diamond S_1 \wedge \Box S_2$ that requires the trace to stay entirely inside the blue rectangle S_2 while intersecting the red circle S_1. We see that traces $\boldsymbol{x}_1, \boldsymbol{x}_2$ satisfy the property. The robustness cylinder around trace $\boldsymbol{x}_1$ is illustrated in the figure. The cylinder represents all perturbations of $\boldsymbol{x}_1$ that also satisfy the property φ and the robustness $\rho(\boldsymbol{x}_1, \varphi)$ is taken to be the radius of the cylinder. It is evident upon a visual inspection that $\rho(\boldsymbol{x}_1, \varphi) > \rho(\boldsymbol{x}_2, \varphi) > 0$.

Similarly, we see that $\boldsymbol{x}_3, \boldsymbol{x}_4$ violate the property. The robustness cylinder around $\boldsymbol{x}_3$ represents all perturbations of $\boldsymbol{x}_3$ that will also violate φ. The robustness $\rho(\boldsymbol{x}_3, \varphi) < 0$ to denote the violation and $|\rho(\boldsymbol{x}_3, \varphi)|$ is set to the radius of the robustness cylinder. It is easy to see that $\rho(\boldsymbol{x}_4, \varphi) < \rho(\boldsymbol{x}_3, \varphi) < 0$.

In fact, robustness for a given trace and property can be computed efficiently using polynomial time in the size of the formula and the number of sample points in the trace $\boldsymbol{x}$ [28]. For convex sets as atomic predicates this requires solving convex optimization problems. However, in practice, the atomic predicates are often described by boxes or half-spaces, and the robustness computation can be optimized significantly.

From Robustness to Falsifications: The problem of finding a violation translates naturally to the problem of finding a negative robustness trace. In turn, we will consider the following optimization problem that seeks to minimize the robustness metric over all traces of a system: $\rho^* : \mathsf{minimize}_{\boldsymbol{x} \in \mathsf{Traces}}\ \rho(\boldsymbol{x}, \varphi)$. If the minimum robustness is $\rho^* < 0$, then we conclude that the system violates the property and the trace $\boldsymbol{x}^*$ that corresponds to the violation is obtained. On the other hand, the robustness function can be quite complicated, even for simple systems. As a result, the optimization problem is hard to solve precisely. To this end, numerous heuristic global optimization algorithms such as simulated annealing [49,1], ant-colony optimization [2], genetic algorithms or the cross-entropy method [58] can be applied to this problem. If these techniques discover a negative robustness trace, then a property violation is concluded. Otherwise, the least robust trace often provides valuable information to the designer, as to how close we get towards violating the property.

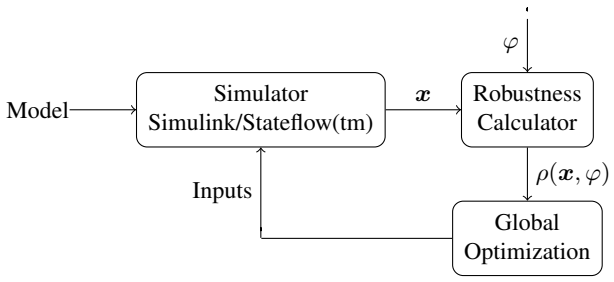


Fig. 2: Illustration of the overall robustness-guided falsification setup.

2.2 S-Taliro Tool

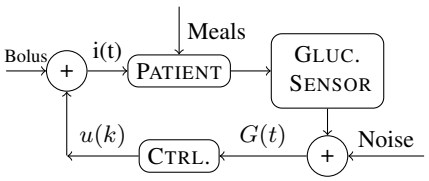
Figure 2 shows a schematic diagram for S-Taliro [5], a robustness guided falsification tool that supports MTL properties [3]. S-Taliro has been implemented inside the Matlab (tm) environment, and can support models described inside Simulink/Stateflow (tm). The tool uses the inbuilt simulator and computes the robustness for a trace. The resulting robustness is used as an objective function by a global optimization engine that seeks to minimize this value. The global optimizer, in turn, decides on future test inputs to the simulator based on the past inputs and the robustness values of the resulting traces. Currently, the tool supports many optimization engines including uniform random exploration, simulated annealing search, ant-colony optimization, cross-entropy method and genetic algorithms. Since no single optimization engine can guarantee finding a global minimum, the typical practice of using the tool consists of using multiple optimization engines, repeatedly and in parallel. If the tool fails to discover a violation, one of the key advantages of robustness metrics is that the least robust trace can provide a relaxed property that can be violated by S-Taliro. S-Taliro is available as an open source tool [6], and is built to be extensible through the addition of new solvers and alternative robustness computation techniques. The latest version uses multiple cores to perform numerous simulations in parallel. It also supports features such as property-directed parameter tuning for models and requirements. These features will be enhanced in future releases of the tool.

3 AP Controller Falsification

We now illustrate the use of S-Taliro on an example PID-based controller design that provides a hybrid closed loop for overnight insulin infusion control. Figure 3 shows the overall diagram of the closed loop system. We note that the controller design here simply serves to illustrate the ideas behind the use of robustness-guided falsification to find potentially harmful scenarios. In particular, the results presented can be improved through systematic and personalized tuning the key controller parameters. We

[5] S-Taliro stands for System TemporAl LogIc RObustness

[6] Cf. `https://sites.google.com/a/asu.edu/s-taliro/s-taliro`



$$
\begin{aligned}
I_e(k) &= I_e(k-1) + (G(k\Delta) - G_0) \\
D(k) &= \tfrac{G(k\Delta) - G((k-1)\Delta)}{\Delta} \\
IOB(k) &= \textstyle\sum_{j=0}^{N} \delta(j) i(k-j) \\
r(k) &= \begin{pmatrix} K_p(G(k\Delta) - G_0) + K_i I_e(k) + \\ K_d D(k) - \gamma IOB(k) \end{pmatrix} \\
u(k) &= \begin{cases} 0 & r(k) \le 0 \\ u_{max} & r(k) \ge u_{max} \\ r(k) & 0 \le r(k) \le u_{max} \end{cases}
\end{aligned}
$$

Fig. 3: Closed loop diagram for the hybrid PID control system, and equations defining the controller. The controller gains and other parameters are shown in blue.

will investigate the overnight use of this control system assuming manual boluses for meals. Such overnight control has the benefit of preventing dangerous seizures due to prolonged hypoglycemia through pump shutoff. Additionally, it helps bring the early morning blood glucose levels inside a tight euglycemic range of $[70, 150]$ mg/dl, leading to desirable longer term outcomes [43].

Controller Design: The controller used in this example is directly inspired by the PID control scheme proposed by Steil et al. [61,60,62]. A detailed analysis of this control scheme was presented by Palerm [51]. Let $G(t)$ represent the value of the blood glucose at time t. The controller operates through periodic sampling of the glucose sensor readings with a period Δ. The insulin level $u(k)$ at the k^{th} time period $t = k\Delta$ is calculated as shown in Figure 3. The insulin infusion rate is held constant for the subsequent time period $[k\Delta, (k+1)\Delta)$. The overall insulin rate $i(t)$ is derived as the sum of the controller input and the patient's manual meal bolus. The terms involved are $I_e(k)$, the integrated error at the k^{th} period, $D(k)$ the derivative term and $IOB(k)$, an *insulin-on-board* compensation term. These are calculated as shown in Figure 3. The parameters for the controller include the target glucose value G_0, taken to be $100mg/dl$, the gains K_p, K_i, K_d, γ that are chosen by trial and error starting from initial values based on the total daily insulin requirements of the patient as explained by Weinzimer et al [62]. Likewise, the cutoff parameters u_{max} that are adjusted by trial and error, starting from the patient's open loop basal rate. The parameters $\delta(j)$ for $j = 0, \ldots, N$ specify the amount of active insulin in the blood at time $t = j\Delta$ corresponding to a unit bolus at time $t = 0$, and is based on available physiological data [61]. Finally, the time period Δ is taken to be 5 minutes for our simulation. The controller code along with the parameter values will be made available to researchers upon request.

Patient Model: Patient modeling is an important part of the overall *in silico* verification process. To this end, many detailed models of insulin-glucose regulation have been proposed. The monograph by Chee and Fernando provides a detailed, historical account of numerous mathematical models [12]. For this simulation study, we use the Dalla-Man et al. model [17,45,47]. This model is a nonlinear ordinary differential equation (ODE) with 10 state variables. The model and corresponding parameters are available as part

Table 2: Inputs that are set by S-Taliro to falsify properties for the AP control system.

T_1	$[0, 60]$ mins.	Dinner time.
X_1	$[50, 150]$ gms	Amount of CHO in dinner.
T_2	$[180, 300]$ mins.	Snack time.
X_2	$[0, 40]$ gms	Amount of CHO in snack.
IC_1, IC_2	$[0, 0.01]$ U/gm	Insulin-to-CHO ratio for meal boluses.
δ_1, δ_2	$[-15, 15]$ min	timing of insulin relative to meal j
$d(100), d(105), \ldots, d(720)$	$[-20, 20]$ mg/dl	sensor error at each sample time.

of the FDA approved T1DM simulator that can now be used as an alternative to animal testing [46]. The model has been increasingly popular inside a simulation environment for "in-silico" or "virtual" clinical trials [52,44].

Nevertheless, to the best of our knowledge, the typical use of this model is through a finite set of fixed "in-silico clinical protocol", that is simulated for multiple sets of patient parameters [52]. The performance statistics such as total time in the euglycemic range or number of hypoglycemic events are reported for each "virtual" patient defined by values of the model parameters. In this exposition, we illustrate a different approach that uses S-Taliro to search over a *set of possible scenarios* to potentially discover the worst case, as defined by the robustness metric, for a given property.

Verification Protocol: For the purposes of falsifying properties of the proposed controller, we use a *set* of possible scenarios, as specified below. Let $t = 0$ represent 7 pm in the evening. Each usage scenario is as follows:

(a) The patient consumes dinner, and manually infuses an insulin bolus at some time $T_1 \in [0, 60]$ minutes. The amount of carbohydrates consumed at dinner X_1 can vary between $[50, 150]$ grams. The bolus is delivered using an insulin-to-CHO ratio IC_1 that can also vary between $[0, 0.01]$ U/gram. Finally, the timing of the bolus relative to the meal time can vary in the range $\delta_1 \in [-15, 15]$ minutes.
(b) The controller is turned on at some time $T_c \in [40, 60]$ minutes, each night.
(c) The patient may possibly consume a snack after the controller is turned on. The snack is consumed sometime between $T_2 \in [180, 300]$, and can vary between $X_2 \in [0, 40]$ grams of carbohydrates. The insulin-to-CHO ratio IC_2 and relative timing δ_2 fall in the same ranges as for dinner.
(d) The controller is turned off at "wake up time" $T_w = 720$.

Finally, we assume that a sensor error of $d(t) \in [-20, 20]$ mg/dl is possible at each sampling instant. This is the error between the sensor output of the Dalla-Man model and the value input to the controller. To decrease the number of parameters, we simply use the values at $d(100), d(105), \ldots, d(720)$ as parameters input to the simulator. These parameters lie in the range $[-20, 20]$ mg/dl, and are also controlled by the S-Taliro tool while exploring the worst-case.

Table 2 summarizes the inputs that S-Taliro can modify to obtain various behaviors of the model. Including the sensor noise inputs, the search space for S-Taliro has nearly 130 parameters. We employed three solvers: uniform random exploration, simulated annealing and the cross-entropy method.

Hypoglycemia: The first property concerns worst case hypoglycemia (low blood glucose levels) possible for this controller. We wish to check whether the system satisfies the MTL property: $\psi_1 : \Box_{[100,700]}(G(t) \geq 70)$, which states that during the time period $t \in [100, 700]$, the blood glucose level should remain above $70mg/dl$ in all scenarios. The time interval $[100, 700]$ is used to allow a run-in period with the controller switched on. As a result, property violations before the controller has warmed up will not be considered. We used three parallel process to search using the simulated-annealing, uniform random and cross-entropy method. While, S-Taliro could not violate the property, the least robust scenario (found by the uniform random search) involves $G(t) \sim 75mg/dl$. In other words, the trace approaches *quite close* to violating the property. The search takes nearly 3200 seconds using three parallel Matlab (tm) R2014 instances on a 4 core, 800 MHz 64 bit AMD Phenom(tm) II processor with 8 GB RAM running Linux. Figure 4 shows the resulting output trace obtained from S-Taliro.

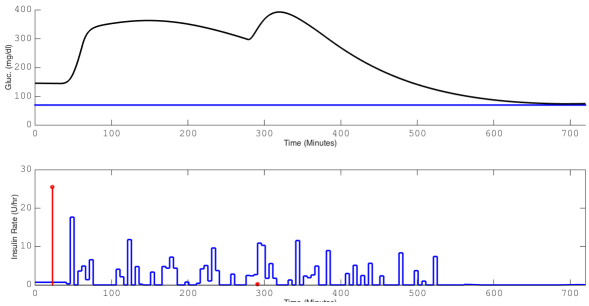


Fig. 4: Least robust trace found by S-Taliro for property $\psi_1 : \Box_{[100,700]}(G(t) \geq 70)$. The top plot for shows the blood glucose levels (mg/dl) over time (minutes) while the bottom plots show the insulin infusion (U/hr) over time (mins). The red impulses represent Bolus amounts in U/hr assuming the bolus amount is delivered over 5 minutes.

Hyperglycemia: The next property concerns whether hyperglycemia (high blood glucose levels) are possible. We wish to check the MTL property $\psi_2 : \Box_{[100,700]}(G(t) \leq 350)$, which states that during the time period $t \in [100, 700]$, can the blood glucose level go above $350mg/dl$. S-Taliro finds a violation of this property with the maximum glucose level of 472 mg/dl. This is found by the cross entropy solver requiring under 5 seconds of total running time. Figure 5 shows the violation trace produced by S-Taliro.

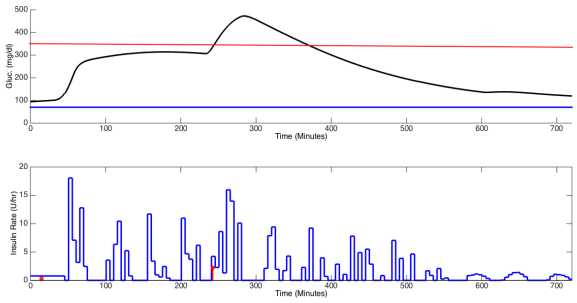


Fig. 5: Least robust trace found by S-Taliro for property $\psi_2 : \Box_{[100,700]}(G(t) \leq 350)$.

Insulin Infusion below Target: The next property concerns whether the controller can infuse insulin while the blood glucose level is below a target level of $90mg/dl$: ψ_3 : $\square_{[100,700]}(G(t) \leq 90 \Rightarrow u(t) = 0)$. The property states that whenever $G(t) \leq 90$ mg/dl, the controller should not command additional insulin, or in other words $u(t) = 0$ should hold. Infusing insulin when the blood glucose is low, can be quite dangerous, worsening the hypoglycemia. The property is violated by S-Taliro in nearly 90 seconds. While all three engines discover a violation, the least robust violation is discovered by the cross entropy (CE) solver.

Hyperglycemia at Wakeup: One of the important objectives of nighttime insulin infusion control is to provide a blood glucose level as close to the normal range as possible at wake up time. Recent clinical evidence indicates that starting off with a normal blood glucose level at wake up time can have beneficial longer term outcomes [43]. To this end, we check whether the morning wakeup blood glucose level can exceed $200mg/dl$.

$$\psi_4 : \square_{[600,700]}(G(t) \leq 200)\,.$$

The property states that the blood glucose levels must remain below 200 mg/dl during the time period $t \in [600, 700]$. S-Taliro cannot violate the property. The minimal robustness trace shows a blood glucose level of $180mg/dl$ at wakeup time, and is discovered by the uniform random search after 3500 seconds.

Prolonged Hyperglycemia: We now focus on the possibility of prolonged hyperglycemia that can potentially give rise to ketacidosis: ψ_5 in Figure 6. The property states that during the time $t \in [200, 600]$ the blood glucose cannot be continuously above 240 mg/dl for more than 180 minutes. S-Taliro easily falsifies this property: the cross-entropy search discovers the least robust trace within 3 seconds.

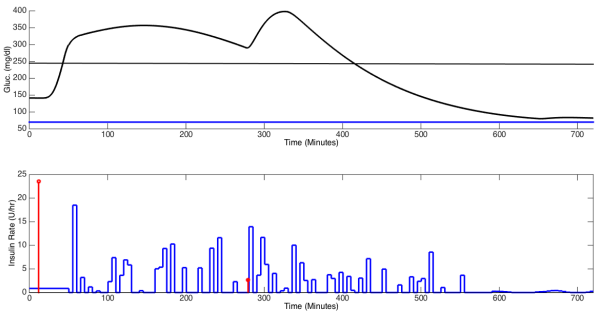


Fig. 6: Least robust trace found by S-Taliro for the prolonged hyperglycemia property ψ_5 : $\neg\Diamond_{[200,600]}\,\square_{[0,180]}\,(G(t) \geq 240)$.

Prolonged Hypoglycemia: Finally, we conclude by searching for the possibility of a prolonged hypoglycemia that can potentially lead to seizures [6]: ψ_6 : $\neg\Diamond_{[200,600]}\,\square_{[0,150]}\,(G(t) \leq 70)$. The property states that there cannot be a contiguous interval of 150 minutes during which $G(t) \leq 70$ mg/dl. The property cannot be violated by S-Taliro. The least robust trace is discovered by uniform random search in 3550 seconds shows a scenario where $G(t) \leq 85$ over a 150 minute interval.

4 Conclusions

In conclusion, we have outlined the need for verifying artificial pancreas controllers and the challenges faced by current verification technique. We have illustrated the use of simulation-based falsification as a first step towards full formal verification. Ongoing work is addressing important gaps in our verification framework including careful modeling of disturbances such as meals, exercise and various sources of sensor noise. We are also working towards making the tool S-Taliro more user friendly to allow control designers to directly use the tool. To this end, we envision simpler and more visual formalisms for specifying temporal properties [37].

Acknowledgments: This material is based upon work supported by the US National Science Foundation (NSF) under grant numbers CPS-1446900, CNS-1319457, CPS-1446751, and CNS-1319560. All opinions expressed are those of the authors, and not necessarily of the NSF.

References

1. Houssam Abbas, Georgios Fainekos, Sriram Sankaranarayanan, Franjo Ivancic, and Aarti Gupta. Probabilistic temporal logic falsification of cyber-physical systems. *Trans. on Embedded Computing Systems (TECS)*, 12:95–, 2013.
2. Yashwanth Singh Rahul Annapureddy and Georgios E. Fainekos. Ant colonies for temporal logic falsification of hybrid systems. In *Proceedings of the 36th Annual Conference of IEEE Industrial Electronics*, pages 91 – 96, 2010.
3. Yashwanth Singh Rahul Annapureddy, Che Liu, Georgios E. Fainekos, and Sriram Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In *Tools and algorithms for the construction and analysis of systems*, volume 6605 of *LNCS*, pages 254–257. Springer, 2011.
4. Eran Atlas, Revital Nimri, Shahar Miller, Eli A. Grunberg, and Moshe Phillip. MD-Logic artificial pancreas system: A pilot study in adults with type 1 diabetes. *Diabetes Care*, 33(5):1072–1076, May 2010.
5. Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.
6. B. Buckingham, D.M. Wilson, T. Lecher, R. Hanas, K. Kaiserman, and F. Cameron. Duration of nocturnal hypoglycemia before seizures. *Diabetes Care*, 31(11):2110–2112, 2008.
7. Fraser Cameron. *Explicitly Minimizing Clinical Risk through Closed-loop Control of Blood Glucose in Patients with Type 1 Diabetes Mellitus*. PhD thesis, Stanford University, 2010.
8. Fraser Cameron, B. Wayne Bequette, D.M. Wilson, Bruce Buckingham, Huyjin Lee, and Günter Niemeyer. Closed-loop artificial pancreas based on risk management. *J. Diabetes Sci Technol.*, 5(2):36879, 2011.
9. Fraser Cameron, Günter Niemeyer, and B. Wayne Bequette. Extended multiple model prediction with application to blood glucose regulation. *Journal of Process Control*, 22(8):1422–1432, Sep 2012.
10. Fraser Cameron, Darrell M. Wilson, Bruce A. Buckingham, Hasmik Arzumanyan, Paula Clinton, H. Peter Chase, John Lum, David M. Maahs, Peter M. Calhoun, and B. Wayne Bequette. Inpatient studies of a kalman-filter-based predictive pump shutoff algorithm. *J. Diabetes Science and Technology*, 6(5):1142–1147, 2012.
11. H. Peter Chase and David Maahs. *Understanding Diabetes (Pink Panther Book)*. Children's Diabetes Foundation, 12 edition, 2011. Available online through CU Denver Barbara Davis Center for Diabetes.

12. Frederick Chee and Tyrone Fernando. *Closed-Loop Control of Blood Glucose*. Springer, 2007.
13. Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *Proc. RTSS'12*, pages 183–192. IEEE, 2012.
14. Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Proc. of CAV'13*, volume 8044 of *LNCS*, pages 258–263. Springer, 2013.
15. Claudio Cobelli et al. and AP@Home Consortium. First use of model predictive control in outpatient wearable artificial pancreas. *Diabetes Care*, 37(5):1212–1215, May 2014.
16. Claudio Cobelli, Chiara Dalla Man, Giovanni Sparacino, Lalo Magni, Giuseppe De Nicolao, and Boris P. Kovatchev. Diabetes: Models, signals and control (methodological review). *IEEE reviews in biomedical engineering*, 2:54–95, 2009.
17. Chiara Dalla Man, Robert A Rizza, and Claudio Cobelli. Meal simulation model of the glucose-insulin system. *IEEE Transactions on Biomedical Engineering*, 1(10):1740–1749, 2006.
18. Thao Dang and Tarik Nahhal. Coverage-guided test generation for continuous and hybrid systems. *Formal Methods in Systems Design*, 34(2):183–213, 2009.
19. E Dassau, H Zisser, Harvey R.A., Percival M.W., B. Grosman, W Bevier, E Atlas, S Miller, R Nimri, L Jovanovic, and Doyle F.J. Clinical evaluation of a personalized artificial pancreas. *Diabetes Care*, 36(4):8019, 2013.
20. Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *TACAS*, volume 4963 of *LNCS*, pages 337–340. Springer, 2008.
21. Adel Dokhanchi, Bardh Hoxha, and Georgios Fainekos. On-line monitoring for temporal logic robustness. In *Runtime Verification*, volume 8734 of *LNCS*, pages 231–246. Springer, 2014.
22. Alexandre Donzé. Breach: A toolbox for verification and parameter synthesis of hybrid systems. In *CAV*, volume 6174 of *Lecture Notes in Computer Science*. Springer, 2010.
23. Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *FORMATS*, volume 6246 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2010.
24. Tommaso Dreossi, Thao Dang, Alexandre Donzé, James Kapinski, Xiaoqing Jin, and Jyotirmoy V. Deshmukh. Efficient guiding strategies for testing of temporal properties of hybrid systems. In *NASA Formal Methods*, volume 9058 of *Lecture Notes in Computer Science*, pages 127–142. Springer, 2015.
25. Firas El-Khatib, J. Jiang J, and Edward R. Damiano. Adaptive closed-loop control provides blood-glucose regulation using dual subcutaneous insulin and glucagon infusion in diabetic swine. *J Diabetes Sci Technol.*, 1(2):18192, 2007.
26. Firas H. El-Khatib, Steven J. Russell, David M. Nathan, Robert G. Sutherlin, and Edward R. Damiano. A bihormonal closed-loop artificial panceras for type 1 diabetes. *Sci. Transl. Med.*, 2, April 2010.
27. Georgios Fainekos and George J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410:4262–4291, 2009.
28. Georgios Fainekos, Sriram Sankaranarayanan, Koichi Ueda, and Hakan Yazarel. Verification of automotive control applications using s-taliro. In *Proceedings of the American Control Conference*, 2012.
29. Gregory P. Forlenza, Sriram Sankaranarayanan, and David M. Maahs. Refining the closed loop in the data age: Research-to-practice transitions in diabetes technology. *Diabetes Technology & Therapeutics*, 17(5), 2015.
30. G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Proc. CAV'11*, volume 6806 of *LNCS*, pages 379–395, 2011.

31. Sicun Gao, Sonhoo Kong, and Edmund M. Clarke. dReal: an SMT solver for nonlinear theories over the reals. In *Proc. CADE'13*, volume 7898 of *Lecture Notes in Computer Science*, pages 208–214. Springer, 2013.
32. B Grosman, E Dassau, H.C Zisser, L Jovanovic, and Doyle F.J. Zone model predictive control: A strategy to minimize hyper- and hypoglycemic events. *J Diabetes Sci Technol.*, 4(4):96175, 2010.
33. R. Hovorka, J. M Allen, D. Elleri, L. J. Chassin, J. Harris, D. Xing, C. Kollman, T. Hovorka, A. M. Larsen, M. Nodale, A. De Palma, M. Wilinska, C. Acerini, and D. Dunger. Manual closed-loop delivery in children and adoloscents with type 1 diabetes: a phase 2 randomised crossover trial. *Lancet*, 375:743–751, February 2010.
34. R. Hovorka, V. Canonico, L.J. Chassin, U. Haueter, M. Massi-Benedetti, M.O. Frederici, T.R. Pieber, H.C. Shaller, L. Schaupp, T. Vering, and M.E. Wilinska. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiological Measurement*, 25:905–920, 2004.
35. R. Hovorka, F. Shojaee-Moradie, P.V. Carroll, L.J. Chassin, I.J. Gowrie, N.C. Jackson, R.S. Tudor, A.M. Umpleby, and R.H. Hones. Partitioning glucose distribution/transport, disposal and endogenous production during IVGTT. *Am. J. Physiol. Endocrinol. Metab.*, 282:992–1007, 2002.
36. Roman Hovorka. Continuous glucose monitoring and closed-loop systems. *Diabetic Medicine*, 23(1):1–12, 2005.
37. B. Hoxha, H. Bach, H. Abbas, A. Dokhanchi, Y. Kobayashi, and G. Fainekos. Towards formal specification visualization for testing and monitoring of cyber-physical systems. In *International Workshop on Design and Implementation of Formal Tools and Systems*, 2014.
38. B Kovatchev, C Cobelli, E Renard, S Anderson, M Breton, S Patek, W Clarke, D Bruttomesso, A Maran, S Costa, A Avogaro, C Dalla Man, A Facchinetti, L Magni, G De Nicolao, J Place, and A Farret. Multinational study of subcutaneous model-predictive closed-loop control in type 1 diabetes mellitus: summary of the results. *J Diabetes Sci Technol.*, 4(6):137481, 2010.
39. Aaron Kowalski. Pathway to artificial pancreas revisited: Moving downstream. *Diabetes Care*, 38:1036–1043, June 2015.
40. Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
41. Steven M. LaValle. Planning algorithms. Cambridge University Press, 2006.
42. David Maahs, Elizabeth Mayer-Davis, Franziska Bishop, L. Wang, M. Mangan, and Robert G. McMurray. Outpatient assessment of determinants of glucose excursions in adolescents with type-1 diabetes. *Diabetes Technology and Therapeutics*, 14(8):658–664, Aug 2012.
43. David M. Maahs, H. Peter Chase, Emily Westfall, Robert Slover, Suiying Huang, John J. Shin, Francine R. Kaufman, Laura Pyle, and Janet K. Snell-Bergeon. The effects of lowering nighttime and breakfast glucose levels with sensor-augmented pump therapy on hemoglobin a1c levels in type 1 diabetes. *Diabetes Technology & Therapeutics*, 16(5):284–291, May 2014.
44. L. Magni, D.M. Raimondo, L. Bossi, C. Dalla Man, G. De Nicolao, B. Kovatchev, and C. Cobelli. Model predictive control of type 1 diabetes: an *in silico* trial. *J. Diabetes Science and Technology*, 1(6):804–12, 2007.
45. Chiara Dalla Man, M. Camilleri, and Claudio Cobelli. A system model of oral glucose absorption: Validation on gold standard data. *Biomedical Engineering, IEEE Transactions on*, 53(12):2472 –2478, dec. 2006.
46. Chiara Dalla Man, F. Micheletto, D. Lv, M. Breton, Boris Kovatchev, and Claudio Cobelli. The UVA/PADOVA type 1 diabetes simulator: New features. *J. Diabetes Science and Technology*, 8(1), January 2014.

47. Chiara Dalla Man, Davide M. Raimondo, Robert A. Rizza, and Claudio Cobelli. GIM, simulation software of meal glucose-insulin model. *J. Diabetes Sci. and Tech.*, 1(3), May 2007.
48. Medtronic Inc. "paradigm" insulin pump with low glucose suspend system, 2012. Cf. `http://www.medtronicdiabetes.ca/en/paradigm_veo_glucose.html`.
49. Truong Nghiem, Sriram Sankaranarayanan, Georgios E. Fainekos, Franjo Ivančić, Aarti Gupta, and George J. Pappas. Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems. In *Hybrid Systems: Computation and Control*, pages 211–220. ACM Press, 2010.
50. R Nimri, I Muller, E Atlas, S Miller, O Kordonouri, N Bratina, C Tsioli, M.A. Stefanija, T Danne, T Battelino, and Phillip M. Night glucose control with md-logic artificial pancreas in home setting: a single blind, randomized crossover trial-interim analysis. *Pediatric Diabetes*, 15(2):91–100, March 2014.
51. Cesar C. Palerm. Physiologic insulin delivery with insulin feedback: A control systems perspective. *Computer Methods and Programs in Biomedicine*, 102(2):130 – 137, 2011.
52. S.D. Patek, B.W. Bequette, M. Breton, B.A. Buckingham, E. Dassau, F.J. Doyle III, J. Lum, L. Magni, and H. Zisser. In silico preclinical trials: methodology and engineering guide to closed-loop control in type 1 diabetes mellitus. *J Diabetes Sci Technol.*, 3(2):269–82, 2009.
53. E. Plaku, Lydia E. Kavraki, and Moshe Y. Vardi. Falsification of LTL safety properties in hybrid systems. In *TACAS*, volume 5505 of *LNCS*, pages 368 – 382, 2009.
54. Erion Plaku, Lydia E. Kavraki, and Moshe Y. Vardi. Hybrid systems: From verification to falsification. In *CAV*, volume 4590 of *LNCS*, pages 463–476. Springer, 2007.
55. Erion Plaku, LydiaE. Kavraki, and MosheY. Vardi. Falsification of ltl safety properties in hybrid systems. *International Journal on Software Tools for Technology Transfer*, 15(4):305–320, 2013.
56. A. Rizk, G. Batt, F. Fages, and S. Soliman. On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In *6th International Conference on Computational Methods in Systems Biology*, number 5307 in LNCS, pages 251–268. Springer, 2008.
57. Sriram Sankaranarayanan and Georgios Fainekos. Simulating insulin infusion pump risks by *In-Silico* modeling of the insulin-glucose regulatory system. In *Computational Methods in Systems Biology (CMSB)*, volume 7605 of *Lecture Notes in Computer Science*, pages 322–339, 2012.
58. Sriram Sankaranarayanan and Georgios E. Fainekos. Falsification of temporal properties of hybrid systems using the cross-entropy method. In *HSCC*, pages 125–134. ACM, 2012.
59. Jay S. Skyler (editor). *Atlas of Diabetes: Fourth Edition*. Springer Science+Business Media, 2012.
60. Garry M. Steil. Algorithms for a closed-loop artificial pancreas: The case for proportional-integral-derivative control. *J. Diabetes Sci. Technol.*, 7:1621–1631, November 2013.
61. G.M. Steil, A.E. Panteleon, and K. Rebrin. Closed-loop insulin delivery - the path to physiological glucose control. *Advanced Drug Delivery Reviews*, 56(2):125–144, 2004.
62. S Weinzimer, G Steil, K Swan, J Dziura, N Kurtz, and W. Tamborlane. Fully automated closed-loop insulin delivery versus semiautomated hybrid control in pediatric patients with type 1 diabetes using an artificial pancreas. *Diabetes Care*, 31:934–939, 2008.