Using Modeling and Simulation in the Design of Closed-Loop Insulin Delivery System

David Hoadley
Arvind Ananthan
MathWorks, Inc.

1 Background

The complexity of medical devices is rapidly increasing, enabled by innovations in technology. This same technology has enabled health care to expand from institutional environments to home care and mobile environments. Software plays a critical role in controlling these devices. It is essential that the dependability of these devices is assured because many of them are safety critical; i.e. can cause serious injury or death.

Modeling and simulation of device designs is emerging among medical device manufacturers as a technique to help address such challenges. Modeling and simulation has a long history of improving product quality by helping designers to detect defects that may be overlooked in a traditional software development workflow. For example, design requirements can be modeled, and then the model checked for properties of requirement consistency and completeness. Subsequently, the model can be run in-silico permitting rapid-prototyping and comprehensive real-time checking of design behavior which can ultimately be used to help verify actual device implementation.

Modeling can be thought of as an incremental process improvement step that happens early in a design phase well before any implementation (writing code or building physical hardware), and can be integrated into any software/hardware development lifecycle. Medical Devices manufacturers and companies in other safety-critical industries such as aerospace and automotive have been using modeling and simulation approaches to verify and validate designs early in their development process for many years now [1][2][4].

Today, many diabetic patients use insulin pumps to manage their own medication delivery through a combination of basal and bolus doses of insulin to maintain healthy blood glucose (BG) levels. With the addition of Continuous Glucose Monitoring (CGM) devices, the opportunity exists to create a closed-loop, automatic control or “artificial pancreas” means of delivering the insulin.

In this technical brief, we present a highly abstracted insulin infusion pump system as an example modeling exercise. A simulated environment of the glycemic system of the body, along with the insulin computation algorithm are modeled in Simulink® and tested against various real-world parameters from clinical testing.

2 Methods

The holy grail of the next generation of insulin pumps is to make the device fully automatic and closed-loop in its operation by adapting insulin delivery to the optimum amount of the body needs at any point during the day, and doing so with minimal user intervention. Among the challenges of designing such a closed-loop system is the population variances, such as insulin need, eating habits, physical activity, and stress level. In addition, there are common challenges, such as time delays of readings from CGM devices and the loop-sided dynamics of BG changes: rapid increase on ingestion vs. slower consumption rate [2].

Two common approaches to develop and tune a robust control system include statistical modeling and analytical plant modeling. The former is a data-based approach where clinical or field use data containing input/output observations are used to identify the model of the entire system - treating it as a “black-box”. The latter technique is mathematically modeling complex system dynamics as close to reality as allowed by current understanding of the system; human physiology in our case. The latter, referred to as a plant model in controls engineering, allows more exploratory simulation scenarios, but requires substantial validation from observed data. The two leading models currently proposed to approximate the behavior of glycemic absorption mechanism of the body come from Roman Havorka, and Claudio Cobelli along with their collaborators, respectively[1][2][4].

Typically, a simulation model is a mix of algorithm development and system engineering. When starting to develop the model of a system like the one built in Simulink for this exercise, one could take a bottom-up approach where the algorithms are developed first while letting the system architecture evolve over time. The Top-down approach focuses on defining the system architecture early on instead. In both cases, the test-bench or the environment around this algorithm can be built up progressively to add realism depending on the level of fidelity required for both the algorithm and the system environment.

Our model consists of the test harness, algorithm subsystem for determining the amount of insulin infusion needed, and the glycemic system approximation using the Havorka model to replicate the behavior of the human body to insulin and glucose (from food) infusions. The test harness includes simulated inputs from the user (manual entry) and different meal profiles that include meal sizes and times during the day the meals are taken. Simulation of the closed-loop algorithm performance against these various realistic factors provides us with a reasonable approximation of a clinical trial setting while providing all the advantages of early discovery and algorithm refinement from an in-silico trial setup.

It can be a substantial challenge to model complex human physiological systems at a level of fidelity that is useful for control design and practical for computational capabilities. Sufficient research may not be available for analytical modeling, and thus the plant model could be a statistical representation of real-world clinical research data that attempts to model the systems a black-box. An example of such a statistical model in case of the glycemic system is the in silico Virtual Clinical Trial simulator run by University of Virginia in conjunction with the FDA [1].

3 Results

Our model demonstrates a combination design that relies on closed-loop response for basal control plus meal and BG excursion correction boluses that are initiated by the user and estimated by a separate algorithm. The bolus estimator can
be parameterized by a healthcare professional for each patient.

For instance, for the algorithm we chose a very simple PID (Proportional, Integral, Derivative) controller for our entire control strategy initially. It was then elaborated over time to include power up and alarm states and basal and bolus modes, including user-driven logic to switch between them. Research continues on other controller strategies, such as predictive modeling [2][3].

In order to evaluate an insulin control algorithm, engineers need a way to test and validate its efficacy in real world scenarios. Traditionally, this is done through expensive clinical studies. It is a good idea to first test it out with a representative model of the actual environment within which it will operate.

In our example, environmental stimuli (meals) and user inputs are provided to the algorithm model, and the results of its calculations are provided to a plant model. We modeled our physiological plant using the mathematical representation of the dynamics of blood glucose metabolism documented by Havorka, et.al. [2]. The different components of the virtual patient’s response are parameterized with experimental data and are used to test how different variations of the insulin delivery algorithm performs under differing environmental and user-controlled conditions.

Such an executable model provides a robust and flexible platform for testing different scenarios that the device might encounter in the real world as well as multiple variants of the controller algorithms suited for different types of patients and their lifestyles.

In our simulations, for instance, we noted a reduction in overnight low-BG excursions when we added meal boluses to our basic PID algorithm. However, we also realized that in order to achieve a fully automatic insulin response (without any need for user boluses based on meals and exercise schedules), it’s imperative to have some type of glucose level predictive algorithms (possibly based on learning algorithms and past history of infusions).

4 Interpretation

Creating an executable model enables design exploration through simulation. In our experiments, such executable models were instrumental in exploring various scenarios that an actual device user might run into, and evaluating the algorithms we come up with under those circumstances. However, we were limited in our depth of these simulations, as well as in our ability to validate the results with real clinical trial results that an actual device manufacturer or clinical researcher needs to complete a design.

Another benefits of using such executable models is that it also acts as executable specification allowing the engineers to verify and validate the design requirements and document the results as required by regulatory standards such as FDA’s QSR 820.30 or IEC 62304. Various software verification and validation tasks can be performed at the model abstraction before these designs are translated to software code. While the resulting code still must be tested, verified and validated, doing so at the model level increases the design confidence early on in the design cycle and reduces translation defects (from words to software) primarily caused by incorrect/insufficient requirements that linger into the implementation stages [5].

While we focused on some of the V&V tasks in this use case, we have deferred the details for a separate technical article. Some of the specific steps we focused on were:

Requirements traceability: Throughout the modeling and simulation phase, requirements can be continuously traced/validated by integrating the linking of requirements to various design elements, or conversely, updating the requirements to make them clear and unambiguous as the design evolves.

Design robustness: In addition to predicted stimulus/response testing, parameter sweeps; random and simulated user input; fault conditions; and so on can be applied to the model to determine the expected reliability of the system under a wide variety of conditions. Anomalous observations can be reconstructed and analyzed. Modeling is not a replacement for clinical data, but can significantly increase the confidence of the design before it is ready to enter the implementation and validation phases of the device’s life cycle.

Design Validation: Simulations can also be a platform for maximizing the value of clinical test data, by reproducing events deterministically and allowing exploration around the parameters sets of interest in a real-world clinical setting. Simulation data thus validated through clinical tests can be useful evidence to demonstrate the safety and efficacy of a device in order to obtain regulator approval to market it to the public.

References


6. When code can kill or cure. The Economist, Jun 2nd 2012.
