Modeling Blood Glucose and Insulin Regulatory System in Event-B

Albert RIZALDI
Supervised by Dominique Méry

Université de Lorraine
MOSEL-VeriDis

DESEM Summer School, Maynooth 2013

Outline

Outline

Background and Motivation

> *...the estimated number of continuous subcutaneous insulin infusion (CSII, "insulin pump therapy") users in the United States has grown from 70,000 to over 300,000 since 1998.*

Scheiner, G., Sobel, R. J., Smith, D. E., Pick, A. J., Kruger, D., King, J., and Green, K. Insulin pump therapy: guidelines for successful outcomes. *Diabetes Educ 35 Suppl 2* (2009), 29S–41S; quiz 28S, 42S–43S

> *... there were over 5,000 adverse events reported for insulin pumps in the year 2008.*

Zhang, Y., Jones, P. L., and Jetley, R. A hazard analysis for a generic insulin infusion pump. *J Diabetes Sci Technol 4*, 2 (Mar 2010), 263–283

Background and Motivation

> *Attendees agree that ... insulin pumps are subject to poor engineering designs, implementation flaws, manufacturing defects ...*

Zhang, Y., Jones, P. L., and Klonoff, D. C. Second insulin pump safety meeting: summary report. *J Diabetes Sci Technol 4*, 2 (Mar 2010), 488–493

References

Several references of formal methods in medical devices:

- Méry, D., and Singh, N. K. Formal specification of medical systems by proof-based refinement. *ACM Trans. Embedded Comput. Syst. 12*, 1 (2013), 15
- Méry, D., and Singh, N. K. Critical systems development methodology using formal techniques. In *3rd International Symposium on Information and Communication Technology - SoICT 2012* (Ha Long, Viet Nam, Aug. 2012), ACM, pp. 3–12
- Méry, D., and Singh, N. K. Closed-loop modeling of cardiac pacemaker and heart". In *FHIES* (2013), vol. 7789 of *LNCS*

We are interested with closed-loop modeling

> *where formal models of the system and an environment form a closed-loop model.*

Méry, D., and Singh, N. K. Closed-loop modeling of cardiac pacemaker and heart". In *FHIES* (2013), vol. 7789 of *LNCS*

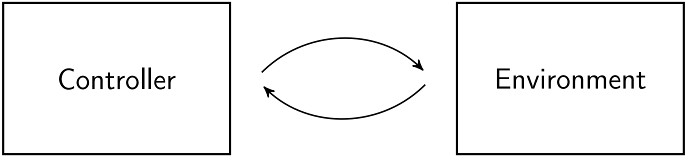


Figure: Closed-loop modeling between controller and environment

Problem Statement

Previous Work

There *exists* a formal model of the *controller* in Event-B but *not* for the *environment*.

Xu, H. Model based system consistency checking using event-b. Master thesis, McMaster University, 2011

Focus

Formalization of *environment* for insulin pumps not the controller

Research Question

How can we model Blood Glucose-Insulin Regulatory System (BGIRS) in Event-B?

- How to translate differential equation into state (event) based formalism?
- How to deal with real numbers in current Event-B?

State of the Art

- BGIRS is hybrid system
  - hybrid system = discrete controller + continuous environment
- State of the art of modeling hybrid system in Event-B:
  - Abrial et al.

    Abrial, J.-R., Su, W., and Zhu, H. Formalizing hybrid systems with event-b. In *Abstract State Machines, Alloy, B, VDM, and Z*, vol. 7316 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 178–193

  - Hybrid Event-B

    Banach, R., and Butler, M. Cruise control in hybrid event-b. *10th International Colloquium on Theoretical Aspects of Computing* (2013). (to appear)

Approach

Rationale

Since the purpose is to obtain a machine checkable formal model of the system, we choose Abrial's instead of Hybrid Event-B.

Problem

But, it is difficult to model differential equation in Event-B!

Quote from *How to solve it*

> *If you cannot solve the proposed problem try to solve first some related problem. . . . Could you solve a part of the (simpler) problem?*

Pólya, G. *How to Solve it: A New Aspects of Mathematical Method*. Anchor Books: Science. Doubleday,Incorporated, 1957

Simplify Problem

Finding

It is "difficult" to model differential equation in Event-B, but it is "easy" to model its solution.

Idea

1. Instead of modeling differential equation directly, we model solution of the differential equation.
2. Instead of considering all possible inputs of differential equation, we focus on a specific class of input ($\mathcal{I}$).

Outline

1. Introduction
2. Mathematical Model
   - Linear Model
   - Input Model
   - Solving Differential Equations
3. Towards Event-B Modeling
4. Logical Model
5. Summary

Ackerman-Bolie's Model[1]

Differential equations for BGIRS:

$$x' = p(t) - \alpha x + \beta y$$

- x is *blood* insulin concentration,
- y is *blood* glucose concentration,
- p is insulin input rate,
- β is sensitivity of insulin to glucose concentration.

$$y' = q(t) - \gamma x - \delta y$$

- q is the glucose intake rate,
- γ is the sensitivity of glucose removal to insulin concentration.

[1] Ghista, D. *Applied Biomedical Engineering Mechanics*. Taylor & Francis, 2008

Input Model (1/3)

Question

After we model the system, how can we specify class of input to BGIRS (model the input) ?

Two inputs:

- p, insulin input (from output of insulin pump)
- q, glucose intake (from what we eat)

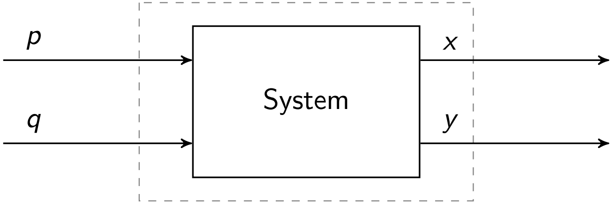


Figure: Inputs to the System

Input Model (2/3)

Insulin pumps have two types of output:

Type	Duration	Characteristic
basal	long	constant rate
bolus	very short	"spike" shape

Table: Basal and Bolus Insulin.

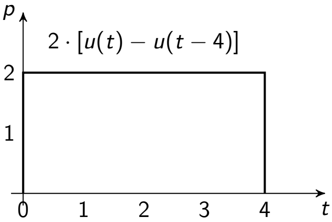
Glucose input can be divided into two types:

Type	Duration	Characteristic
complex	long	constant rate
simple	very short	"spike" shape

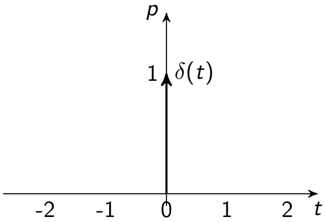
Table: Simple Sugar and Complex Carbohydrate.

Input Model (3/3)

Long and constant rate:



Short and "spike" shape:



Superposition

Linear systems imply *superposition*.

Superposition

1. If $(x_1 \mapsto y_1)$ and $(x_2 \mapsto y_2)$ are two input-output pairs of a linear system
2. and k_1 and k_2 are any two real or imaginary numbers

then $(k_1 x_1 + k_2 x_2 \mapsto k_1 y_1 + k_2 y_2)$ is also input-output pair of the linear system.

Solution Tree

By using superposition property, we can “divide-and-conquer” the solution of differential equation.

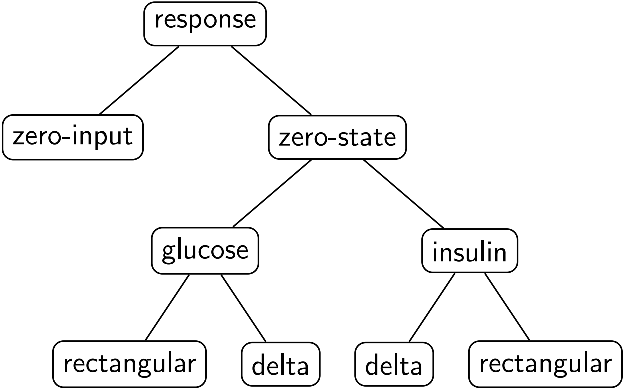


Figure: Decomposing the Solution of ODE.

Complete Solution

The solution of BGIRS differential equations with

- initial condition $y(0) = K$ and $y'(0) = 0$
- glucose input $q_{\text{step}} \cdot u(t) + q_{\text{delta}} \cdot \delta(t)$
- insulin input $p_{\text{step}} \cdot u(t) + p_{\text{delta}} \cdot \delta(t)$

is defined as

$$
\begin{aligned}
y(t) = & -\frac{K \cdot \alpha}{\delta - \alpha} e^{-\delta \cdot t} + \frac{K \cdot \delta}{\delta - \alpha} e^{-\alpha \cdot t} \\
& + q_{\text{step}} \cdot \left(\frac{1 - e^{-\delta \cdot t}}{\delta}\right) + q_{\text{delta}} \cdot e^{-\delta \cdot t} \\
& - \frac{\gamma \cdot p_{\text{step}}}{\delta - \alpha} \cdot \left(\frac{1 - e^{-\alpha \cdot t}}{\alpha} - \frac{1 - e^{-\delta \cdot t}}{\delta}\right) - \frac{\gamma \cdot p_{\text{delta}}}{\delta - \alpha} \cdot \left(e^{-\alpha \cdot t} - e^{-\delta \cdot t}\right)
\end{aligned}
$$

Outline

Real Numbers

Problem

Event-B does not support real numbers!

Possible solutions

- Use another prover
- Construct real type

Plugin	Description	Problem
Isabelle	Translate PO from Event-B to Isabelle	Plugin assume no Real type in Event-B
Theory	Define new type in Event-B	Difficult (e.g. Dedekind Cuts, etc)

Table: Plugin Description and Problem.

Parser

Idea

Build a simple parser (ideally a plugin in RODIN) that read declared *Rodin's real expression* to *Isabelle's expression*

Implicitly,

- Need to make Rodin recognize real type
  - define `Real` type in `Context`
- Need to define operator for real number
  - define *operator* as *function* in `Context`'s constant
- Need to define predicate for real number
  - define *predicate* as *set of pairs* in `Context`

Grammar

$\langle \mathit{expr} \rangle$ ::= <literal> | <variable> | <unExp> | <binExp>

$\langle \mathit{literal} \rangle$ ::= "ZERO" | "ONE"

$\langle \mathit{variable} \rangle$ ::= "Var"(<String>)

$\langle \mathit{unExp} \rangle$ ::= "NEG"(<expr>) | "INV"(<expr>)

$\langle \mathit{binExp} \rangle$ ::= "SUM"(<expr> $\mapsto$ <expr>)
| "MULT"(<expr> $\mapsto$ <expr>)
| "DIVIDE"(<expr> $\mapsto$ <expr>)
| "MINUS"(<expr> $\mapsto$ <expr>)

$\langle \mathit{predicate} \rangle$ ::= (<expr> $\mapsto$ <expr>) $\in$ "LTE"
| (<expr> $\mapsto$ <expr>) $\in$ "GT"
| <expr> $\neq$ <expr>
| <expr> $=$ <expr>

$\langle \mathit{formula} \rangle$ ::= <predicate> $\longleftrightarrow$ <predicate>

Implementation

Requirement

Make a parser with grammar defined previously to expression recognizable by Isabelle

- already implement a simple program in Scala
- Extends the `RegexParser` in Scala

Why Scala?

- can be compiled to Java Virtual Machine
- comply with Eclipse platform
- can be made into Rodin's plugin

Outline

Modeling Strategy

Question

How can we model the solution of diff. equation in event (state) based formalism?

Problems

- Should not introduce time too early
- Time domain of *exponential function* is infinite ($\mathbb{N}$)

Idea

Approximate exponential function.

Approximate Exponential Function

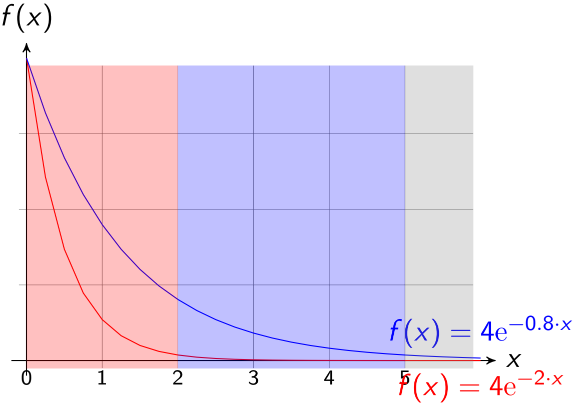


Figure: Approximate Exponential Functions.

Finite State Machine

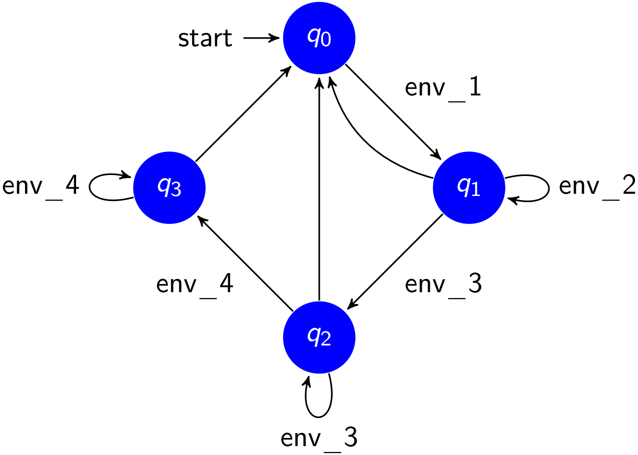


Figure: Finite State Machine for Diff. Equation Solution.

Refinement Strategy

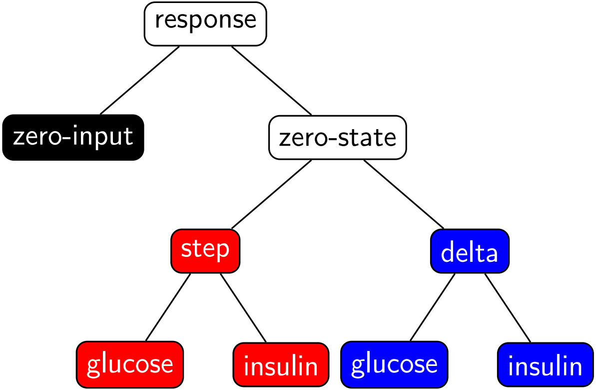


Figure: Decomposing the Solution of ODE.

Refinement

- First
- Second
- Third

Proving

- What have we proved?
  - Type invariant (e.g. $y \in \mathbb{R}^+$)
  - Refinement proof (e.g. guard strenghtening, simulation)
- How to prove it?
  - Pencil-and-paper proof (backward reasoning)
  - Formal proof in Isabelle (forward reasoning)

Findings

During the process of proving, we find important invariants

- At first refinement

$$\gamma \cdot p_{\mathsf{step}} \leq \alpha \cdot q_{\mathsf{step}}$$

- At second refinement

$$\frac{\gamma \cdot p_{\mathsf{step}}}{\alpha\delta} + \frac{\gamma \cdot p_{\mathsf{delta}}}{\delta - \alpha} \leq \frac{q_{\mathsf{step}}}{\delta}$$

Outline

Conclusion (1 / 2)

Question

How to translate differential equation into state (event) based formalism?

1. Model the *solution* instead of the differential equation itself
2. Focus on specific class of input (ie. linear combination of delta and step signal)
3. Solve with superposition principle (ie. zero-state, zero-input, etc)
4. Approximate exponential function (ie. consider it to be zero after $3 \cdot \tau$)
5. Copy the way we decompose the solution as refinement strategy

Conclusion (2 / 2)

Question

How to deal with real numbers in current Event-B?

1. Declare real numbers' type, operator, and predicate as carrier sets, function, and set of pairs respectively
2. Export all real numbers-related proof obligations to Isabelle

Future Works[2]

1. Use Isabelle's support of exponent function to simplify the model
2. Add more properties to prove
   - upper and lower bound, timing properties, etc
3. Formalization of insulin pump through closed-loop modeling
   - Connect the *controller* and *environment* model
4. Validation
   - Validate the assumption (with expert)
   - Validate the expressivity of input model

[2] Our work will be presented in FHIES 2013 Macao, China

Bibliography I

Abrial, J.-R., Su, W., and Zhu, H.
Formalizing hybrid systems with event-b.
In *Abstract State Machines, Alloy, B, VDM, and Z*, vol. 7316 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 178–193.

Banach, R., and Butler, M.
Cruise control in hybrid event-b.
10th International Colloquium on Theoretical Aspects of Computing (2013).
(to appear).

Ghista, D.
Applied Biomedical Engineering Mechanics.
Taylor & Francis, 2008.

Bibliography II

Hartlove, J., Shaffer, D., and Ragan, S.
Glucose-Insulin Model: A Core Learning Goals Activity for Science and Mathematics.
Maryland Virtual High School of Science and Mathematics, January 2001.

Méry, D., and Singh, N. K.
Critical systems development methodology using formal techniques.
In *3rd International Symposium on Information and Communication Technology - SoICT 2012* (Ha Long, Viet Nam, Aug. 2012), ACM, pp. 3–12.

Méry, D., and Singh, N. K.
Closed-loop modeling of cardiac pacemaker and heart".
In *FHIES* (2013), vol. 7789 of *LNCS*.

Bibliography III

Méry, D., and Singh, N. K.
Formal specification of medical systems by proof-based refinement.
ACM Trans. Embedded Comput. Syst. 12, 1 (2013), 15.

Pólya, G.
How to Solve it: A New Aspects of Mathematical Method.
Anchor Books: Science. Doubleday,Incorporated, 1957.

Rodgers, J.
Using Insulin Pumps in Diabetes: A Guide for Nurses and Other Health Professionals.
Wiley, 2008.

Sarshogh, M. R., and Butler, M.
Specification and refinement of discrete timing properties in event-b.
ECEASST 46 (2011).

Bibliography IV

Scheiner, G., Sobel, R. J., Smith, D. E., Pick, A. J., Kruger, D., King, J., and Green, K.
Insulin pump therapy: guidelines for successful outcomes.
Diabetes Educ 35 Suppl 2 (2009), 29S–41S; quiz 28S, 42S–43S.

Sommerville, I.
Software Engineering: (Update) (8th Edition) (International Computer Science).
Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

Xu, H.
Model based system consistency checking using event-b.
Master thesis, McMaster University, 2011.

Bibliography V

Zhang, Y., Jones, P. L., and Jetley, R.
A hazard analysis for a generic insulin infusion pump.
J Diabetes Sci Technol 4, 2 (Mar 2010), 263–283.

Zhang, Y., Jones, P. L., and Klonoff, D. C.
Second insulin pump safety meeting: summary report.
J Diabetes Sci Technol 4, 2 (Mar 2010), 488–493.

Merci!
Thank you!
Go raibh milé maith agaibh!
Tapadh leibh!