Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 210 (2022) 116–123

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2022)
October 26-28, 2022, Leuven, Belgium

A Novel Approach to Multi-Compartmental Model Implementation to Achieve Metabolic Model Identifiability on Patient's CGM Data

Martin Ubl[a,*], Tomas Koutny[b]

[a] *Department of Computer Science and Engineering, University of West Bohemia, Technicka 2967/8, Pilsen, 301 00, Czechia*
[b] *New Technologies for Information Society, University of West Bohemia, Technicka 2967/8, Pilsen, 301 00, Czechia*

Abstract

Diabetes is a widespread civilization disease. When developing a new treatment method, *in-silico* benefits the development process by reducing the need for *in-vivo* subjects. *In-silico* evaluation requires a reliable metabolic model, often created as a multi-compartmental model. A common approach to implementing a multi-compartmental model is to use a system of ordinary differential equations. This approach utilises exponential transfer functions to transfer substances among the compartments. Using other than an exponential function is complex. Therefore, we propose a novel approach based on a direct, numeric integration of separated compartments, which can be further divided into individual depots. This enables to model substance transfer as a separate process with non-exponential characteristics, e.g.; when modelling carbohydrate absorption from the gut. As another benefit, the approach obeys the law of mass conservation on both the computational and architectural levels. This is a key feature when identifying a model on data, that are not measured within a controlled, isolated environment. Moreover, we actually transform the set of equations, i.e.; computer-code functions, into a component model to reduce the total maintenance costs – readability, testing, verification and deployment. We demonstrate the proposed approach by converting the Samadi model to it and enhancing it with a non-exponential transfer function.

Keywords: diabetes; multi-compartmental model; metabolic model; simulation; smartcgms

[*] Corresponding author. Tel.: +420 377 632 453 ; fax: +420 377 634 402.
E-mail address: ublm@kiv.zcu.cz

10.1016/j.procs.2022.10.127

1. Introduction

Diabetes is a heterogeneous group of diseases, which all manifest with elevated blood glucose (BG) levels[1]. Elevated BG leads to organ damage and potentially death. The main cause of diabetes is either an absolute (type 1 diabetes, T1D) or relative (type 2 diabetes, T2D) insulin insufficiency. Insulin is a hormone, which serves as a transfer agent for glucose to pass the cell membrane. It also allows for storing excessive glucose in a form of glycogen. Overall, it lowers the BG. This paper is further concerned with T1D-only.

The patient measures his/her glucose levels using a CGM device[2]. This device measures the glucose levels in the interstitial fluid using a sensor deployed to a patient's subcutaneous tissue. The sensor obtains the interstitial glucose (IG) readings, which are sufficiently similar to BG readings in order to achieve a successful treatment[3].

For a T1D patient, in order to keep the BG in a safe range, effective insulin treatment is needed. The treatment aims to keep BG in between 3.6-6.0$mmol/L$. An insulin pump is a device, that doses insulin into a patient's body[4]. It operates in two modes – bolus and basal insulin. Bolus insulin is a one-time insulin delivery of a greater volume. Basal insulin delivery is a continuous delivery mode, which doses small portions of insulin periodically every few seconds or minutes. T1D insulin treatment is a control problem[5]. The control is either an open-loop, in which the user sets the insulin volumes manually or a closed-loop, in which the pump deploys an algorithm, which estimates the dosage based on BG readings. A hybrid closed-loop control assumes that the system runs as a closed-loop, but no control action is performed without the patient's intervention[5].

Treatment methods are often tested *in-silico* in order to reduce the need for *in-vivo* patients[6]. The *in-silico* evaluation phase requires a physiological model, that allows this type of testing inside a simulation. Such a model comprises a number of compartments, each virtually maintaining at least one substance depot. Compartments are connected using virtual links, which represent transfer functions. These transfer functions are often exponential. This is due to the fact, that substance absorption was observed to be of an exponential character[7].

A common approach to multi-compartmental model implementation is to use a system of ordinary differential equations (ODE), as such a model represents a dynamic system[8]. Exponential transfer functions are modelled linearly in ODE. This implicitly keeps the computational time low. Using ODE as an implementation in conjunction with exponential transfer functions allows for multiple transfers within a single compartment to be commenced at once – i.e.; there is no need for multiple depots with isolated transfers. This is a useful property when modelling e.g.; meal-glucose or insulin dosage absorption.

To solve the ODE system in a computer program, a solver is used – usually a Runge-Kutta method-based solver[9]. Given the initial conditions, the solver integrates the ODE system and calculates quantities in all compartments incrementally.

To achieve optimal treatment based on model-predictive control (MPC), we need to have a fully personalized physiological model. Obtaining a personalized physiological model is a particularly hard problem. With the lack of all relevant quantity measurements, the fitting of any model may exhibit large errors. Moreover, measured data from CGM sensors are always noisy and suffer from immune system-induced sensor numbness. Sensor calibration by an explicit BG measurement partially compensates for this phenomenon[10].

Furthermore, deployment of such a model into a closed-loop system driven by an embedded device requires all components to be properly verified on a component level. Verification of a multi-compartmental model implemented as an ODE system is hard. To be able to easily verify the implementation, we require a high level of decomposition and a well-defined component interface[11].

1.1. SmartCGMS

SmartCGMS is a signal processing framework and architecture. It builds on top of the principles of High-Level Architecture (HLA), which is a well-established simulation paradigm.

A SmartCGMS configuration defines a chain of linearly connected top-level entities called *filters*. Filters communicate by a message passing. Every filter can generate a message, which is then propagated through the chain from the source to the last filter in chain. A filter can send messages to preceding filters only via the feedback link[13]. The message structure is defined by an abstract data type.[12]

Every filter maintains a single, isolated functionality. For example, database reader filter reads previously measured values and sends them to the chain. Signal generator filter uses a configured model to calculate one or more signals, optionally based on previously measured values. A log filter reads every message and transcribes it to a CSV log file. A drawing filter reads all messages containing a signal level and visualizes them in form of a plot.

2. Related work

Multi-compartmental models are an integral part of, among others, diabetes mellitus treatment research. Their primary goal in this area is to model the human glucose metabolism and related systems as accurately, as possible. Bergman [14] devised a model of a glucose disappearance to estimate insulin sensitivity. Hovorka [15] devised a model containing multiple metabolic subsystems to allow for model-predictive BG control. Dalla Man [16] introduced a model specifically for insulin treatment evaluation. This model, deployed as a part of T1DMS simulator, was accepted by the U.S. Food and Drug Administration (FDA) for pre-clinical trial *in-silico* testing on a single-meal scenarios. Later, an improved model was proposed by Visentin, that was accepted by FDA for a single-day scenarios[17]. Samadi [18] proposed an alternative model incorporating a physical activity into the Hovorka model.

All of the above models are implemented as an ODE system. As the models gain complexity, their equations grow and creates a room for a human error when proposing enhancements. Furthermore, all of the above models need various transfer functions other than exponentials. This forces authors to create non-standard constructions, such as multiple compartments to simulate absorption curves, when, in fact, there are only two physical compartments that exchange the substance in a non-exponential way.

3. Proposed approach

We propose a novel approach to multi-compartmental model implementation, considering the best practices of software engineering. The multi-compartmental model implementation comprises a set of entities with a defined interface. The top-level entity is a compartment itself. It contains a dynamic collection of depots. Each depot has a quantity and a volume. Depots are connected with links, which encapsulates a transfer function, and optionally a list of moderator functions. A single depot can be connected to multiple depots by links with various transfer functions.

A transfer function entity maintains the transfer between depots. When using a standard ODE approach, the implementor needs to split the transfer into two separate terms for respective differential equations. For example, the exponential transfer with a transfer coefficient c between compartment A and B is written using an ODE system as follows:

$$\begin{aligned}\frac{dA(t)}{dt} &= -c \cdot A(t)\\ \frac{dB(t)}{dt} &= c \cdot A(t)\end{aligned} \tag{1}$$

Most likely, a computer code representing these equations will comprise a function for each equation. We may use a fraction of a C++ code to demonstrate the form:

```
double dA(TState state, double T) {
  return -c*state.A;
}
double dB(TState state, double T) {
  return c*state.B;
}
```

In this code, `TState` represent a container structure for compartment quantities. The equivalent link is expressed with proposed approach as a computer code as follows:

```
A.Link_To<CConstant_Unbounded_Transfer_Function>(B, c);
```

The above code maintains both the parts of the transfer (the elimination from A and appearance in B), as it is a single entity, which considers both the depots at once during transfer. Although the transfer is of an exponential character, the term is linear, hence the name of the entity.

In order to define virtually any type of transfer function, we use a direct numerical integration of transfer functions to determine the fraction of quantity to be exchanged between depots. Using this approach, it is also possible to create

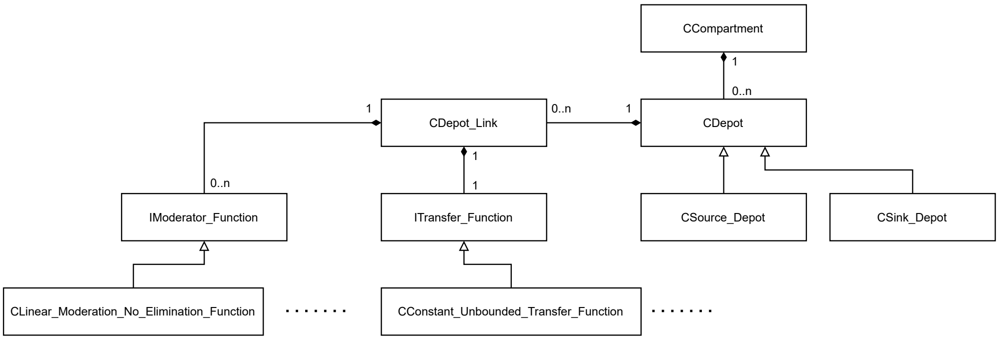

Fig. 1. Class diagram of proposed method

temporary depots with bounded transfer functions. For example to model insulin absorption, we can take the insulin activity curve and implement it directly as a time-bounded function. Then, using multiple types of insulin does not need multiple compartments – we change the upper bound of the insulin transfer function.

The figure 1 shows the class diagram of the proposed method. The `CDepot` class represents the generic depot with usual behavior. This class has two specializations – `CSource_Depot` as a depot with unlimited quantity (always gives the requested amount) and `CSink_Depot` as a depot with unlimited capacity (always takes and consumes the whole amount). These two depot types model a situation, when a substance continuously enters or leaves the modelled system.

Moderator function is a function, that moderates the transfer amount by an amount of other depot. Therefore, it is the source of non-linearity in the original ODE approach. The transfer function in our proposed approach itself cannot take another depot into account and needs a moderator to moderate the transfered amount, optionally with moderator elimination. The moderator function is implemented as a component, which calculates moderation input. The moderator function component is instantiated during transfer function definition. Then, during the simulation, the transfer amount is multiplied by a moderation input. A typical example of a moderated transfer is insulin-dependent glucose elimination – the amount of glucose eliminated from the system either by utilization or glycogenesis is directly dependent on the amount of insulin in respective compartment.

To prove the equivalence between the standard ODE approach and our proposed approach, let us take the `CConstant_Unbounded_Transfer_Function` component and an exponential transfer function expressed in ODE. A single transfer within our proposed method steps through a simple equation with a step of Δt:

$$\begin{aligned} n(Q(t), t) &= Q(t) * \int_{t_i}^{t_{i+1}} f(t)\, dt \\ A(t + \Delta t) &= A(t) - n(A(t), t) \\ B(t + \Delta t) &= B(t) + n(A(t), t), \end{aligned} \tag{2}$$

where $f(t)$ is a transfer function. The chosen function yields just a constant coefficient for every input, thus $f(t) = c$. Let us use the equation for compartment A and substitute the $f(t)$ with a constant term c. Then, by solving the integral on a given time domain, factoring out the constant term c, substituting the time difference $t_{i+1} - t_i$ by a Δt and rearranging the equation we obtain the following difference form:

$$\frac{A(t + \Delta t) - A(t)}{\Delta t} = -c * A(t) \tag{3}$$

Using a limit, the resulting equation is successfully converted into a form of an ODE:

$$\begin{aligned} \lim_{\Delta t \to 0} \frac{A(t + \Delta t) - A(t)}{\Delta t} &= -c * A(t) \\ \frac{dA(t)}{dt} &= -c * A(t) \end{aligned} \tag{4}$$

4. Experimental setup

To verify the equivalence between traditional ODE approach and our proposed novel approach, we choose the Samadi model as the representative of ODE-based multi-compartmental models. The full equation description is presented in the respective paper[18].

We implemented the original Samadi model as a system of ODE, and tuned its parameters to fit a specific time segment extracted from Ohio T1D dataset[19], more specifically a single day segment of a subject 544. We used the MetaDE algorithm to find parameters of the model[20]. The original Samadi model is stepped using Dormand-Prince parametrization of a Runge-Kutta ODE solver[21]. For our proposed approach, we use the Simpson's 1/3 rule to integrate over the transfer function[22].

A large portion of transfer functions is of an exponential character, and therefore can be transformed to a simple link with constant transfer rate, as demonstrated in section 3. Nonetheless, some of the transfer characteristics exhibit a different behavior:

- over-the-threshold linear clearance – e.g.; renal clearance over the threshold of $9.0 mmol/L$ (the $F_R(t)$ term)
- threshold-driven ratio clearance – e.g.; insulin-independent glucose uptake term ($F^C_{01}(t)$ term)
- moderated appearance or clearance – there are many instances, e.g.; physical activity-moderated insulin transfer of glucose from Q_1 to Q_2 (written as $(1 + \alpha E_2(t)^2) \cdot x_1(t) \cdot Q_1(t)$)
- a number of transfers in the exercise subsystem – e.g.; every transfer moderated by the $f(E_1(t))$ function
- unit change during transfer – e.g.; transfer between S_2 and I changes the units with the factor of $\frac{1}{V_I}$; this means, that some quantity leaves the S_2 compartment, and enters the I compartment multiplied by this factor

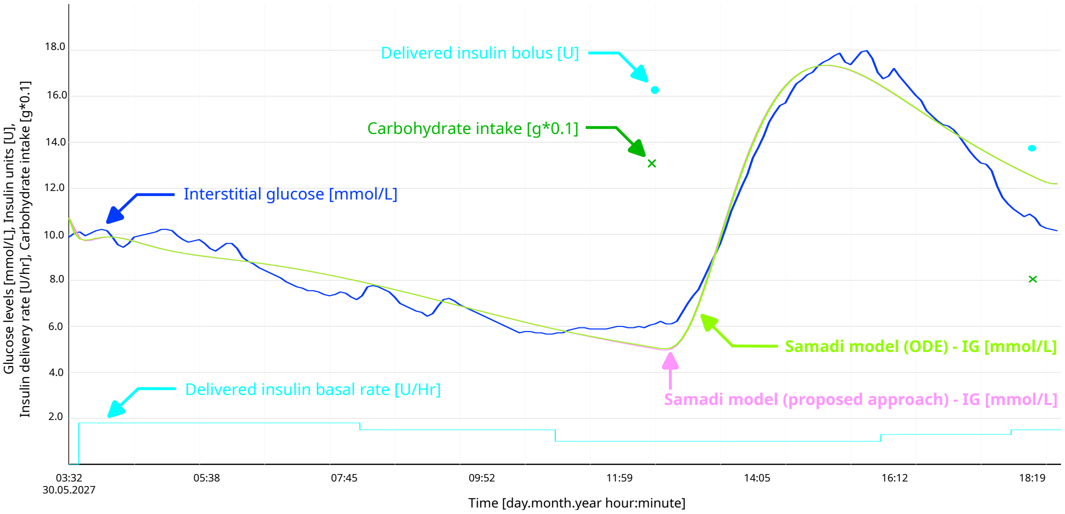

Fig. 2. Graph of the measured input data, and both implementations of Samadi model.

We transformed the Samadi model from its original form of differential equations to our proposed approach. For every type of transfer function, we implemented its own class. Then we use the same set of parameters for both the original and proposed approach to show, that both approaches can represent the original model. Table 1 shows the absolute and relative errors of the original ODE implementation, and our proposed approach, respectively, against the measured data. Figure 3 depicts the empirical cumulative distribution function (ECDF). Both the tables and ECDF shows, that the errors of both approaches are very close. Figure 2 then shows the time-segment with curves from both model implementations.

To demonstrate the capabilities of our proposed approach, let us slightly modify the Samadi model. The glucose flux from Q_1 to G_sc is modelled as a simple exponential transfer from Q_1 to G_sc with unit conversion, and as a clearance from G_sc. This is not physiologically correct, as the substance transfer between these two compartments is

Error type	Method	Avg. ± Std.Dev.	Min.	1. Q	Median	3. Q	95% Pct.	99% Pct.	Max.
Absolute	ODE	0.571 ± 0.429	5.78e-10	0.271	0.516	**0.714**	**1.204**	**1.930**	**1.999**
	Proposed	0.582 ± 0.431	1.673e-03	0.273	0.521	0.752	1.235	**1.930**	**1.999**
	Modified	**0.5693 ± 0.459**	**3.161e-10**	**0.23**	**0.5068**	0.7194	1.408	1.993	2.097
Relative	ODE	6.325% ± 4.935%	4.05e-09%	2.66%	5.076%	8.767%	16.5%	18.93%	19.47%
	Proposed	6.445% ± 5.009%	1.173e-02%	2.761%	5.188%	8.761%	16.71%	19.12%	19.86%
	Modified	**6.116% ± 4.931%**	**2.216e-09%**	**2.55%**	**4.959%**	7.917%	17.03%	18.85%	20.53%

Table 1. Absolute and relative errors of original ODE system implementation of Samadi model, proposed method and proposed method modified with non-exponential transfer

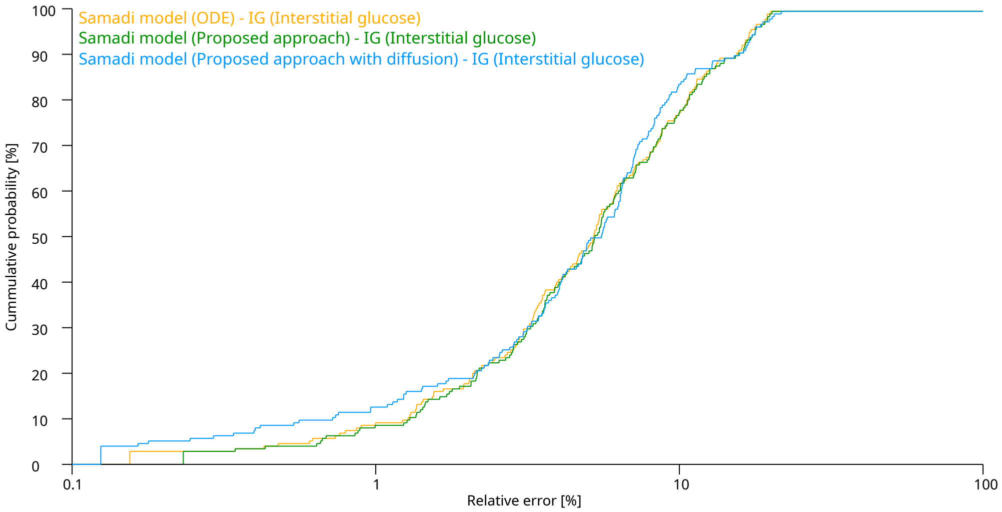

Fig. 3. Empirical cumulative distributon function of both implementations of Samadi model

a facilitated diffusion. Let us modify the model as follows. We rename G_sc to Q_sc and remove unit conversion, so the Q_1 and Q_sc have the same units. Then, we remove all the links regarding this depot and introduce a new one, that uses the `CTwo_Way_Diffusion_Unbounded_Transfer_Function` function. We reuse the τ_g parameter as a diffusion rate parameter. The Q_sc is converted to IG measurements (formerly G_sc) by a control code outside the model definition.

After modifying the model, the parameter set is no longer valid, as the model changed and the meaning of the τ_g parameter changed its meaning. Let us use the MetaDE to obtain a new parameter fit. Table 1 shows absolute and relative errors after modifying the model and obtaining a new parameter set.

As the table 1 shows, the introduction of diffusion transfer between Q_1 and Q_sc slightly improved the model fit. Nonetheless, we introduced this change just to demonstrate the modularity and robustness of our proposed approach. This paper does not suggest any modifications to this model.

For a better comparison of both approaches on the implementation level, let us consider Q_1 and Q_2 compartments. The code of the ODE implementation follows:

```
constexpr double Gthresh_F01 = 4.5;
constexpr double Gthresh_FR = 9.0;

double CSamadi_ODE::dQ1(const double _T, const double _X) const {
  const double Gt = _X / mParams.Vg;
  const double F01C = Gt >= Gthresh_F01 ?
                      mParams.F01 : (mParams.F01 * Gt / Gthresh_F01);
  const double FR = Gt >= Gthresh_FR ?
                    0.003 * (Gt - 9.0) * mParams.Vg : 0;
  const double UG = mState.D2 / mParams.tmaxG + mState.DH2 / (mParams.tmaxG / 2.0);

  return -(1 + mParams.alpha * mState.E2 * mState.E2) * mState.x1 * _X
         + mParams.k12 * mState.Q2
         - F01C - FR + UG + mParams.EGP_0 * (1 - mState.x3);
}

double CSamadi_ODE::dQ2(const double _T, const double _X) const {
  return (1 + mParams.alpha * mState.E2 * mState.E2) * mState.x1 * mState.Q1
         - mParams.k12 * _X
         - mState.x2 * _X * (1 + mParams.alpha * mState.E2 * mState.E2
                             - mParams.beta * mState.E1 / mParams.HRbase);
}
```

The code above is considered a "spaghetti code"[23] – a code with repetitive parts and little to no decomposition. Although all given transfers are well-defined as a mathematical function, the implementation does not allow to verify the program correctness with a standardized testing procedures, e.g.; to test the transfers separately with a set of unit tests.

The most repeated part is the `1 + mParams.alpha * mState.E2 * mState.E2` calculation. In terms of our proposed approach, this is a moderator function. Therefore, we implement the moderator function as a component inheriting the `IModerator_Function` interface:

```
class CQuadratic_Moderation_No_Elimination_Function : public IModeration_Function {
  protected:
    double mModeration_Factor = 1.0;
  public:
    CQuadratic_Moderation_No_Elimination_Function(double modFactor)
      : mModeration_Factor(modFactor) {
    }

    virtual double Get_Moderation_Input(double moderatorAmount) const override {
      return 1.0 + moderatorAmount * moderatorAmount * mModeration_Factor;
    }

    virtual double Get_Elimination_Input(double moderatorAmount) const override {
      return 0.0;
    }
};
```

The link between Q_1 and Q_2 utilizing this moderator function is then established as follows:

```
q1.Moderated_Link_To<CConstant_Unbounded_Transfer_Function>(q2,
  [&E2, &x1, this](CDepot_Link& link) {
    link.Add_Moderator<CQuadratic_Moderation_No_Elimination_Function>(E2, mParams.alpha);
    link.Add_Moderator<CLinear_Moderation_No_Elimination_Function>(x1, 1.0);
  },
  1.0);
```

This component is reusable and therefore, we can use it to define the other two repetitions of the same term.

5. Conclusion

We proposed a novel approach to multi-compartmental model implementation. It respects the software engineering best-practices. The model, its compartments and transfer functions are decomposed on the implementation level into separate, reusable building blocks. The code is cleaner, self-documenting, better constrained and reduces the probability of human error during model modifications. Furthermore, every building block can be verified separately using standard testing procedures. This complies with the basic idea of the SmartCGMS framework and allows for a possible future rapid deployment process.

The proposed approach was verified on the Samadi model. The Samadi model was implemented in two versions – original implementation using ODE system, and using the proposed approach. Results show, that both approaches exhibits very similar errors, while the proposed method is much more scalable and respects the best practices of software engineering. The error difference most likely comes from the differences between the used ODE solver and the integration rule – both exhibit a slightly different behavior in different situations.

The current implementation of the proposed approach is based purely on dynamic components. For the future work, converting a substantial part of the component code-base into a statically instantiated components may greatly improve the performance.

Acknowledgements

This article was created with the financial support of the project Improving the Quality of Internal Grant Schemes at the UWB, project registration number: CZ.02.2.69/0.0/0.0/19_073/0016931, and was also supported by a university-specific research grant SGS-2022-015 of University of West Bohemia.

References

[1] Kasper DL, Fauci AS, Hauser SL, Longo DL, Jameson JL, Loscalzo J. (2015) "Harrison's principles of internal medicine (19th edition.)" in New York: McGraw Hill Education. ISBN: 007174889X

[2] Facchinetti A. (2016) "Continuous Glucose Monitoring Sensors: Past, Present and Future Algorithmic Challenges." in Sensors (Basel). **16** (12): 2093. DOI: 10.3390/s16122093

[3] Hirsch IB, Battelino T, Peters AL, Chamberlain JJ, Aleppo G, Bergenstal RM. (2018) "Role of continuous glucose monitoring in diabetes treatment." Arlington, Va., American Diabetes Association.

[4] Nimri R, Nir J, Phillip M. (2020) "Insulin Pump Therapy" in American Journal of Therapeutics. **27** (1): e30–e41. DOI: 10.1097/MJT.0000000000001097

[5] Fuchs J, Hovorka R. (2020) "Closed-loop control in insulin pumps for type-1 diabetes mellitus: safety and efficacy." in Expert review of medical devices. **17** (7): 707–720.

[6] Colquitt RB, Colquhoun DA, Thiele RH. (2011) "In silico modelling of physiologic systems." in Best practice & research – Clinical anaesthesiology. **25** (4): 499–510.

[7] Debnath S, Kumar TH. (2020) "An Overview on Pharmacokinetics and Pharmacokinetic Modeling." in Asian Journal of Research in Pharmaceutical Science. **10** (2): 124–130.

[8] Close CM, Frederick DK, Newell JC. (2001) "Modeling and analysis of dynamic systems". John Wiley & Sons. ISBN: 978-0-471-39442-6

[9] Zhang J, Mokhtari A, Sra S, Jadbabaie A. (2018) "Direct Runge-Kutta discretization achieves acceleration." in Advances in neural information processing systems. **31**.

[10] Acciaroli G, Vettoretti M, Facchinetti A, Sparacino G. (2018) "Calibration of minimally invasive continuous glucose monitoring sensors: state-of-the-art and current perspectives." in Biosensors. **8** (1): 24. DOI: 10.3390/bios8010024

[11] Hsiung PA, Santambrogio MD, Huang CH. (2009) "Reconfigurable System Design and Verification (1st ed.)." in CRC Press. DOI: 10.1201/9781315219035

[12] Koutny T, Ubl M. (2018) "Parallel software architecture for the next generation of glucose monitoring." in Procedia Computer Science. **141**: 279–286. DOI: 10.1016/j.procs.2018.10.197

[13] Ubl M, Koutny T. (2019) "SmartCGMS as an Environment for an Insulin-Pump Development with FDA-Accepted In-Silico Pre-Clinical Trials." in Procedia Computer Science. **160**: 322–329. DOI: 10.1016/j.procs.2019.11.084

[14] Bergman RN, Ider YZ, Bowden CR, Cobelli C. (1979) "Quantitative estimation of insulin sensitivity." in The American Journal of Physiology. **236** (6): 667–677. DOI: 10.1152/ajpendo.1979.236.6.E667

[15] Hovorka R, Canonico V, Chassin LJ, Haueter U, Massi-Benedetti M, Orsini Federici M, Pieber TR, et al. (2004) "Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes." in Physiological measurement. **25** (4): 905–920. DOI: 10.1088/0967-3334/25/4/010

[16] Dalla Man C, Micheletto F, Lv D, Breton M, Kovatchev BP, Cobelli C. (2014) "The UVA/PADOVA Type 1 Diabetes Simulator: new features." in Journal of Diabetes Science and Technology. **8** (1): 26–34. DOI: 10.1177/1932296813514502

[17] Visentin R, Campos-Náñez E, Schiavon M, et al. (2018) "The UVA/Padova Type 1 Diabetes Simulator Goes From Single Meal to Single Day.""in Journal of Diabetes Science and Technology. **12** (2): 273–281. DOI: 10.1177/1932296818757747

[18] Rashid M, Samadi S, Sevil M, Hajizadeh I, Kolodziej P, Hobbs N, Maloney Z, et al. (2019) "Simulation software for assessment of nonlinear and adaptive multivariable control algorithms: Glucose–insulin dynamics in Type 1 diabetes." in Computers & Chemical Engineering. **130**. DOI: 10.1016/j.compchemeng.2019.106565

[19] Marling C, Bunescu R. (2020) "The OhioT1DM dataset for blood glucose level prediction: Update 2020." in CEUR workshop proceedings. **2675**: 71.

[20] Koutny T. (2016) "Using meta-differential evolution to enhance a calculation of a continuous blood glucose level." in Computer methods and programs in biomedicine. **133**: 45–54.

[21] Kimura T. (2009) "On dormand-prince method." Japan Malaysia Technical Institute. **40**: 1–9.

[22] Uddin MJ, Moheuddin MM, Kowsher M. (2019) "A New Study of Trapezoidal, Simpson's 1/3 and Simpson's 3/8 Rules of Numerical Integral Problems." in Applied Mathematics and Sciences: An International Journal. **6** (4): 1–14. DOI: 10.5121/mathsj.2019.6401

[23] Politowski C, Khomh F, Romano S, Scanniello G, Petrillo F, Guéhéneuc YG, Maiga A. (2020) "A large scale empirical study of the impact of Spaghetti Code and Blob anti-patterns on program comprehension." in Information and Software Technology. **122**. DOI: 10.1016/j.infsof.2020.106278