



(51) International Patent Classification:  
G16H 50/20 (2018.01)

(21) International Application Number:  
PCT/US2023/085495

(22) International Filing Date:  
21 December 2023 (21.12.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
63/434,833 22 December 2022 (22.12.2022) US

(71) Applicant: **PRESIDENT AND FELLOWS OF HARVARD COLLEGE** [US/US]; 17 Quincy Street, Cambridge, Massachusetts 02138 (US).

(72) Inventors: **OZASLAN, Basak**; c/o Harvard University, Smith Center – Suite 727, 1350 Massachusetts Ave., Cambridge, Massachusetts 02138 (US). **AIELLO, Eleonora Maria**; c/o Harvard University, Smith Center – Suite 727, 1350 Massachusetts Ave., Cambridge, Massachusetts 02138 (US). **DOYLE, III, Francis J.**; c/o Harvard University, Smith Center – Suite 727, 1350 Massachusetts Ave., Cambridge, Massachusetts 02138 (US). **DASSAU, Eyal**; c/

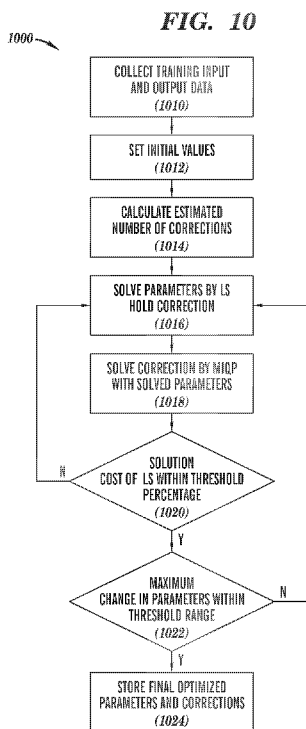
o Harvard University, Smith Center – Suite 727, 1350 Massachusetts Ave., Cambridge, Massachusetts 02138 (US).

(74) Agent: **YOUNG, Alissa R.** et al.; Nixon Peabody LLP, 53 State Street, Exchange Place, Boston, Massachusetts 02109 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE,

(54) Title: MODEL IDENTIFICATION WITH INCOMPLETE INPUT DATA IN TYPE I DIABETES



(57) Abstract: A system and method for identifying an individualized glucose dynamics model for managing glucose of a patient is disclosed. The method includes collecting glucose concentration output data from the patient over a period of time. Carbohydrate size and insulin infusion input data is collected from the patient over the period of time. Parameters of a glucose dynamics model are optimized via solving an iterative optimization problem such as a nonlinear least squares problem based on the collected data while fixing a constant time and amount of a correction value. The time and amount of a modified correction value are determined based on executing a mixed integer quadratic program on the collected data while fixing a constant value for the parameters of the glucose dynamics model. A final correction value and final optimized parameters of the model are provided based on iterating the optimizing parameters and determining the modified correction value.

WO 2024/138024 A2

SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

**Published:**

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

## **MODEL IDENTIFICATION WITH INCOMPLETE INPUT DATA IN TYPE I DIABETES**

### **PRIORITY CLAIM**

**[0001]** The present disclosure claims benefit of and priority to U.S. Provisional Application No. 63/434,833 filed December 22, 2022. The contents of that application are hereby incorporated in their entirety.

### **STATEMENT OF GOVERNMENT SUPPORT**

**[0002]** This invention was made with government support under Grant No. R01DK120358, awarded by the National Institute of Health (NIH). The government has certain rights in the invention.

### **TECHNICAL FIELD**

**[0003]** The present disclosure relates generally to treatments for diabetes and more specifically to a method of identification of individualized model parameters via determining corrections for incomplete input data aiming for a correct prediction of glucose outcomes.

### **BACKGROUND**

**[0004]** Type 1 Diabetes (T1D) is associated with the lack of endogenous insulin production, leading to high blood glucose levels. Short of a cure, strategic planning of insulin injection and carbohydrate intake are essential to keep blood glucose levels within a safe range. Yet, glucose management is a challenging task in daily life as (i) glucose metabolism is, in general, difficult to predict, and (ii) the insulin effect happens on a timescale that makes timely glucose regulation non-trivial. Therefore, mathematical models that capture the relationship between the main inputs (i.e., insulin and meal) and glucose outcomes are highly sought to guide immediate and long-term treatment decisions for achieving satisfactory glucose management.

**[0005]** With the increasingly available data recorded through Continuous Glucose Monitors (CGM) and insulin pumps, the focus has been directed toward data-driven methods

to obtain personalized models. It has been shown that machine learning methods, such as training neural networks on T1D data may help increase model performance but they may also lead to inadvertent outcomes on test data. Thus, models that are based on physiological knowledge and are parameterized to allow interindividual differences, while not as flexible as neural networks, are considered safer. Therefore, the identification problem involves fitting few model parameters to data.

**[0006]** Despite advances in engineering and diabetes technologies, obtaining personalized mathematical models that can fit real-life data remains a notoriously difficult problem. This is not surprising given the fact that human glucose metabolism is a complex system to be described with a small set of equations. There are numerous internal (e.g., stress) and external (e.g., physical exercise) disturbances, the effect of circadian rhythm (e.g., dawn phenomenon) on glucose dynamics with significant intra- and inter-patient variations. While there are average population models for some of these factors, such as the effect of meals, the majority of factors is yet to be modeled.

**[0007]** Further, free-living data may be affected by many confounding factors and may also be incomplete in many cases. A major challenge in fitting models to glucose metabolism in people with Type 1 diabetes is incomplete data as its collection partially relies on self-reporting and does not include all relevant events. In particular, since patients with T1D self-report their carbohydrate intake, the time and amount of these meals can be wrongly reported or not reported at all (e.g., the person may simply choose not to or forget to report meals or snacks). Such inaccuracies may lead to inaccurate models that result in inappropriate treatment decisions.

**[0008]** Mixed integer optimization has previously been employed in model identification context for hybrid models, fault detection, and best subset selection in regression analysis. However, optimized sparse corrections for incomplete input data while jointly identifying the model parameters of a dynamical system is not currently available. To date, no generic technique has been developed to tackle the problem of incomplete data, as even noticing the fact that the input data is incomplete/incorrect may be non-trivial. This problem is often faced in clinical data collected in real life in cases such as unreported meal intake, unknown macro-nutrient composition, unreported physical activity, and unreported events of device malfunction. In practice, obvious problems in the data are handled by either imputation or discarding the part of the data with missing/incorrect information in the preprocessing phase for model identification. However, this practice is not ideal since (i) the discarded data may carry important information, (ii) using the generic imputation methods may lead to a bias in

parameter identification, and (iii) one often may not even know if there are incomplete/incorrect portions of the information available.

**[0009]** Thus, there is a need to identify personalized metabolic parameters for a dynamics model in a way that the identification becomes inherently robust against significant inaccuracies in the collected input data.

### **SUMMARY**

**[0010]** The term embodiment and like terms, e.g., implementation, configuration, aspect, example, and option, are intended to refer broadly to all of the subject matter of this disclosure and the claims below. Statements containing these terms should be understood not to limit the subject matter described herein or to limit the meaning or scope of the claims below. Embodiments of the present disclosure covered herein are defined by the claims below, not this summary. This summary is a high-level overview of various aspects of the disclosure and introduces some of the concepts that are further described in the Detailed Description section below. This summary is not intended to identify key or essential features of the claimed subject matter. This summary is also not intended to be used in isolation to determine the scope of the claimed subject matter. The subject matter should be understood by reference to appropriate portions of the entire specification of this disclosure, any or all drawings, and each claim.

**[0011]** According to certain aspects of the present disclosure, a method for determining a modified dynamic system model for predicting an output from a dynamic system based on an input is disclosed. Output data is collected from the dynamic system over a period of time. Input data is collected from the dynamic system over a period of time. Parameters of a dynamic system model are optimized via solving a nonlinear least squares problem based on the collected data while fixing a correction value. A modified correction value is determined based on a mixed integer quadratic program executed on the collected data while fixing a constant value for the parameters of the dynamic system model. A final correction value and final optimized parameters are provided to the dynamic system model from iterating the optimizing the parameters of the dynamic system model and determining the modified correction value.

**[0012]** A further implementation of the example method is where the dynamic system is a patient and the input data relates to a drug taken by the patient. Another implementation is where the dynamic system is one of a biological system, an industrial system, a population system, a market system, or a resource distribution system. Another implementation is where the example method further includes determining a maximum number of correction values

based the number of times an absolute relative difference value of collected output data and an output of an initial fit from the least squares method exceeds a threshold value.

**[0013]** Another disclosed example is method for determining an individualized model for predicting glucose levels of a patient. Glucose concentration output data is collected from the patient over a period of time. Carbohydrate size and insulin infusion input data is collected from the patient over a period of time. Parameters of a glucose dynamics model are optimized via solving a nonlinear least squares problem based on the collected data while fixing a constant time and a constant amount of a correction value. The time and the amount of a modified correction value is determined based on executing a mixed integer quadratic program on the collected data while incorporating the optimized parameters of the glucose dynamics model. A final correction value and final optimized parameters of the glucose dynamics model are provided based on iterating the optimizing parameters of the glucose dynamics model and determining the time and the amount of the modified correction value.

**[0014]** A further implementation of the example method is where the parameters of the glucose dynamics model include glucose at equilibrium, insulin at equilibrium, insulin sensitivity, meal transport rate, real absorption rate, insulin utilization rate, and insulin clearance rate. Another implementation is where the collecting glucose data includes periodically collecting sets of recorded glucose measurements from the patient, and wherein the collecting the input data includes periodically collecting data on insulin injections and carbohydrate intakes from the patient. Another implementation is where the glucose measurements are collected from a glucose monitor, the carbohydrate intake is entered from a patient input to a computing device, and the insulin injections are determined from an insulin pump. Another implementation is where optimizing parameters of the glucose dynamics model includes minimizing the sum of the squared errors from the nonlinear least squares problem while the number of corrections does not exceed a predetermined value. Another implementation is where the predetermined value is determined by the number of times an absolute relative difference value of a recorded glucose trace and an output of an initial fit from the least squares method exceeds a threshold value. Another implementation is where the predetermined number of corrections is limited by a predetermined time period. Another implementation is where the correction value compensates for incomplete input data including at least one of missing meal reports, miscalculated meals, delay in eating a meal, and insulin pump occlusion. Another implementation is where the example method includes storing the model with the final optimized parameters and final correction value in a storage device.

**[0015]** Another disclosed example is a system for determining an individualized glucose dynamics model for a patient. The system includes a memory storing machine executable code. A control system is coupled to the memory having one or more processors. The control system is configured to execute the machine executable code to cause the control system to collect glucose data from the patient over a period of time. The control system is configured to collect carbohydrate input data from the patient over a period of time. The control system is configured to optimize parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected data while fixing a constant time and a constant amount of a correction value. The control system is configured to determine the time and the amount of a modified correction value based on executing a mixed integer quadratic program on the collected data while incorporating the optimized parameters of the glucose dynamics model. The control system is configured to provide a final correction signal value and final optimized parameters of the glucose dynamics model based on iterating the optimizing parameters of the glucose dynamics model and determining the time and the amount of the modified correction value.

**[0016]** A further implementation of the example system is where the parameters of the glucose dynamics model include glucose at equilibrium, insulin at equilibrium, insulin sensitivity, meal transport rate, real absorption rate, insulin utilization rate, and insulin clearance rate. Another implementation is where the collection of glucose data includes periodically collecting sets of recorded glucose measurements from the patient, and where the collection of the input data includes periodically collecting data on insulin injections and carbohydrate intakes from the patient. Another implementation is where the example system includes a glucose monitor recording the glucose measurements. The example system includes a computing device with a patient input to enter the carbohydrate intakes. The example system includes an insulin pump configured to collect the insulin injections. Another implementation is where the optimization of parameters of the glucose dynamics model includes minimizing the sum of the squared errors from the nonlinear least squares problem while the number of corrections does not exceed a predetermined value. Another implementation is where the predetermined value is determined by the number of times an absolute relative difference value of a recorded glucose trace and an output of an initial fit from the least squares method exceeds a threshold value. Another implementation is where the predetermined number of corrections is limited by a predetermined time period. Another implementation is where the correction value compensates for incomplete input data including at least one of missing meal reports, miscalculated meals, delay in eating a meal, and insulin pump occlusion. Another

implementation is where the example system includes a storage device storing the model with the final optimized parameters and final correction value.

**[0017]** Another disclosed example is a non-transitory machine readable medium having stored thereon instructions for performing a method comprising machine executable code which when executed by at least one machine, causes the machine to collect glucose data from the patient over a period of time. The instructions cause the machine to collect carbohydrate input data from the patient over a period of time. The instructions cause the machine to optimize parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected data while fixing a constant time and a constant amount of a correction value. The instructions cause the machine to determine the time and the amount of a modified correction value based on executing a mixed integer quadratic program on the collected data while incorporating the optimized parameters of the glucose dynamics model. The instructions cause the machine to provide a final correction value and final optimized parameters of the glucose dynamics model based on iterating the optimizing parameters of the glucose dynamics model and determining the time and the amount of the modified correction value.

**[0018]** Another disclosed example is a system for managing glucose of a patient. The system includes an insulin pump configured to deliver insulin into a patient. The system includes a memory storing machine executable code. The system includes a control system coupled to the memory. The control system has one or more processors and is configured to execute the machine executable code to cause the control system to accept a carbohydrate size input and accept an insulin infusion input from the patient. The control system is configured to modify the carbohydrate size input with a correction value of an individualized glucose dynamics model including optimized parameters. The correction value and optimized parameters are determined by collecting glucose concentration output training data from the patient and collecting carbohydrate size and insulin infusion input training data from the patient over a period of time. The determining includes iteratively optimizing parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected training data while fixing a constant time and amount of a correction value; and determining the time and amount of a modified correction value based on executing a mixed integer quadratic program on the collected training data while fixing a constant value for the parameters of the glucose dynamics model. The control system provides the insulin infusion input and the modified carbohydrate size input to the individualized glucose dynamics model to determine a prediction of glucose outcome. The control system is configured to send a command to the insulin pump to deliver a determined insulin dosage based on the predicted glucose outcome.

**[0019]** A further implementation of the example system is where the parameters of the glucose dynamics model include glucose at equilibrium, insulin at equilibrium, insulin sensitivity, meal transport rate, real absorption rate, insulin utilization rate, and insulin clearance rate. Another implementation is where the example system includes a computing device in communication with the control system. The computing device includes a patient input to enter the carbohydrate intake. The insulin pump is configured to collect the insulin infusion input. Another implementation is where the optimization of parameters of the glucose dynamics model includes minimizing the sum of the squared errors from the nonlinear least squares problem while the number of corrections does not exceed a predetermined value. Another implementation is where the predetermined value is determined by determining the number of times an absolute relative difference value of a recorded glucose trace and an output of an initial fit from the least squares method exceeds a threshold value. Another implementation is where the predetermined number of corrections is limited by a predetermined time period. Another implementation is where the correction value compensates for incomplete input data including at least one of missing meal reports, miscalculated meals, delay in eating a meal, and insulin pump occlusion.

**[0020]** Another disclosed example is a method of providing treatment for a diabetic patient. A carbohydrate size input from a patient is accepted. An insulin infusion input from the patient is accepted. The carbohydrate size input is modified with a correction value of an individualized glucose dynamics model including optimized parameters. The correction value and optimized parameters are determined by: collecting glucose concentration output training data from the patient and collecting carbohydrate size and insulin infusion input training data from the patient over a period of time; and iteratively optimizing parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected training data while fixing a constant time and amount of a correction value; and determining the time and amount of a modified correction value based on executing a mixed integer quadratic program on the collected training data while fixing a constant value for the parameters of the glucose dynamics model. The insulin infusion input and the modified carbohydrate size input to the model are provided to determine a prediction of glucose outcome. A command to an insulin pump is sent to deliver a determined insulin dosage based on the predicted glucose outcome.

**[0021]** A further implementation of the example method is where the parameters of the glucose dynamics model include glucose at equilibrium, insulin at equilibrium, insulin sensitivity, meal transport rate, real absorption rate, insulin utilization rate, and insulin

clearance rate. Another implementation is where the carbohydrate intake input is received from a computing device including a patient input to enter the carbohydrate intake, and where the insulin infusion input is collected from an insulin pump. Another implementation is where the optimization of parameters of the glucose dynamics model includes minimizing the sum of the squared errors from the nonlinear least squares problem while the number of corrections does not exceed a predetermined value. Another implementation is where the predetermined value is determined by determining the number of times an absolute relative difference value of a recorded glucose trace and an output of an initial fit from the least squares method exceeds a threshold value. Another implementation is where the predetermined number of corrections is limited by a predetermined time period. Another implementation is where the correction value compensates for incomplete input data including at least one of missing meal reports, miscalculated meals, delay in eating a meal, and insulin pump occlusion.

**[0022]** Another example is a non-transitory machine readable medium having stored thereon instructions for performing a method comprising machine executable code which when executed by at least one machine, causes the machine to accept a carbohydrate size input from a patient. The instructions cause the machine to accept an insulin infusion input from the patient. The instructions cause the machine to modify the carbohydrate size input with a correction value of an individualized glucose dynamics model including optimized parameters. The correction value and optimized parameters are determined by: collecting glucose concentration output training data from the patient and collecting carbohydrate size and insulin infusion input training data from the patient over a period of time; and iteratively optimizing parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected training data while fixing a constant time and amount of a correction value; and determining the time and amount of a modified correction value based on executing a mixed integer quadratic program on the collected training data while fixing a constant value for the parameters of the glucose dynamics model. The instructions cause the machine to provide the insulin infusion input and the modified carbohydrate size input to the model to determine a prediction of glucose outcome. The instructions cause the machine to send a command to an insulin pump to deliver a determined insulin dosage based on the predicted glucose outcome.

**[0023]** There has thus been outlined, rather broadly, the more important features of the invention in order that the detailed description thereof may be better understood, and in order that the present contribution to the art may be better appreciated. There are additional features

of the invention that will be described hereinafter and that will form the subject matter of the claims appended hereto.

**[0024]** In this respect, before explaining several embodiments of the invention in detail, it is to be understood that the invention is not limited in its application to the details of the construction, configuration or arrangement of the components set forth in the following descriptions or illustrated in the drawings. The invention is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of description and should not be regarded as limiting.

**[0025]** To the accomplishment of the above and related objects, this invention may be embodied in the form illustrated in the accompanying drawings, attention being called to the fact, however, that the drawings are illustrative only, and that changes may be made in the specific configuration and processes illustrated and described within the scope of the appended claims.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0026]** Various other objects, features and attendant advantages of the present invention will become fully appreciated as the same becomes better understood when considered in conjunction with the accompanying drawings, wherein:

**[0027]** FIG. 1 is an insulin delivery system with a controller for insulin doses with an individualized model for dosage prediction;

**[0028]** FIG. 2 is a set of graphs showing the need for correction for inaccurate information towards the identification of a dynamic glucose model;

**[0029]** FIG. 3 is an example algorithm to determine an estimated number of corrections to be applied to the example individualized model;

**[0030]** FIG. 4 is an example algorithm to provide optimized parameters and corrections for an example individualized model;

**[0031]** FIG. 5A is a table of corresponding symbols for the plots in FIG. 5B;

**[0032]** FIG. 5B is a series of plots that show different scenarios of accurate and inaccurate input data;

**[0033]** FIG. 6 is a table of root mean square error values across subjects for a comparison of models including a model generated by the example method applied to the different scenarios of accurate and inaccurate input data;

[0034] FIG. 7A is a series of boxplots showing the estimated number of corrections required for each of the different scenarios;

[0035] FIG. 7B is a series of boxplots showing how the number of estimated corrections change with different ARD thresholds;

[0036] FIG. 8A shows graphs of model fits compared with the original output and corresponding input data in a training dataset along with the identified input corrections by the example method;

[0037] FIG. 8B shows graphs of model fits compared with the original output and corresponding input data in a testing dataset;

[0038] FIG. 9 is a set of graphs showing an example application on real-life data to obtain individualized glucose model for pregnant women with diabetes; and

[0039] FIG. 10 is a flow diagram of the example routine to obtain corrections and optimized model parameters for an individualized model.

#### **DETAILED DESCRIPTION**

[0040] Various embodiments are described with reference to the attached figures, where like reference numerals are used throughout the figures to designate similar or equivalent elements. The figures are not necessarily drawn to scale and are provided merely to illustrate aspects and features of the present disclosure. Numerous specific details, relationships, and methods are set forth to provide a full understanding of certain aspects and features of the present disclosure, although one having ordinary skill in the relevant art will recognize that these aspects and features can be practiced without one or more of the specific details, with other relationships, or with other methods. In some instances, well-known structures or operations are not shown in detail for illustrative purposes. The various embodiments disclosed herein are not necessarily limited by the illustrated ordering of acts or events, as some acts may occur in different orders and/or concurrently with other acts or events. Furthermore, not all illustrated acts or events are necessarily required to implement certain aspects and features of the present disclosure.

[0041] For purposes of the present detailed description, unless specifically disclaimed, and where appropriate, the singular includes the plural and vice versa. The word “including” means “including without limitation.” Moreover, words of approximation, such as “about,” “almost,” “substantially,” “approximately,” and the like, can be used herein to mean “at,” “near,” “nearly at,” “within 3-5% of,” “within acceptable manufacturing tolerances of,” or any logical

combination thereof. Similarly, terms “vertical” or “horizontal” are intended to additionally include “within 3-5% of” a vertical or horizontal orientation, respectively. Additionally, words of direction, such as “top,” “bottom,” “left,” “right,” “above,” and “below” are intended to relate to the equivalent direction as depicted in a reference illustration; as understood contextually from the object(s) or element(s) being referenced, such as from a commonly used position for the object(s) or element(s); or as otherwise described herein.

**[0042]** This disclosure relates to a system and method that uses a sequential identification algorithm that alternates between solving a nonlinear least squares problem to optimize dynamic system model parameters and executing Mixed-Integer Quadratic Programming (MIQP) to optimize the time and amount of input corrections. A dynamic system model is mathematical representation of the dynamics between the inputs and outputs of a dynamic system. The input corrections account for missing or incorrect inputs in a dataset for identifying correct dynamic system model parameters. Thus, the unreported or misreported parts in the data are reconciled by adding sparse corrections via the MIQP leading to an improved identification of the model parameters. In the case of an example glucose dynamics model, generally meal intake and insulin injections represent the system inputs, and thus events such as unreported or misreported meals and boluses, can be interpreted as input disturbances. Moreover, as such events are typically characterized by an impulse nature, there is the need for a method of determining insulin injections that automatically recognizes the occurrence of such disturbances and allows for a sparse correction signal in the identification. This effort allows reestablishing a correct input-output relationship in the data leading to an improved identification of the model parameters. The reason behind choosing corrections to be sparse is to mainly target the above disturbances, which are usually few and far between in a day. However, such corrections can also handle other input inaccuracies, which are not strictly sparse by nature, as in the case of insulin pump occlusion (when insulin pump records injected insulin that in fact is not delivered). Sparsity can be enforced in the identification problem by a suitable regularization term in the cost function or with a cardinality constraint in a mixed-integer formulation of the identification problem.

**[0043]** Thus, the present disclosure relates to a model parameter estimation method that is robust against incomplete or incorrect input data. The example method uses available input-output data such as a drug input and body reaction as an output for a patient (dynamic system). An example of a dynamic system model is the Subcutaneous Oral Glucose Minimal Model (SOGMM) having insulin and meal data for the inputs and glucose concentration as the output. The incompleteness/incorrectness of data is a major impediment to parameter identification of

any model. Therefore, the example method has a broad range of potential applications for any dynamic system whose identification may suffer from the incomplete input data, such as human metabolic systems (e.g., the impact of a medication on vital signs), batch distillation, batch reactions, catalyst regeneration, pharmacokinetic and pharmacodynamic models, pricing models for markets, population growth models, resources distribution networks (such as a water distribution network), industrial processes, or aircraft dynamics. For example, in the case of a human metabolic system, the impacts of behavioral and environmental inputs (e.g., physical activity, salt intake, alcohol intake, certain medications, changing time zone) on the metabolic outputs (e.g., blood pressure, glucose, hormones) vary among individuals. An accurate identification of the parameters of a dynamics model would require complete information on the inputs that significantly affect the outputs, which may not be available. This situation can be reconciled by the example method to obtain correct model parameters.

**[0044]** Beyond biological models, the example method can be used in a variety of applications. In the case of industrial processes, incomplete datasets and unknown measurements is a common condition, the estimation of the incomplete information can support decision-making and the detection of fault diagnosis. Additionally, the information on catastrophic events (e.g., famine, wars, pandemics) in a population growth model may be incomplete. For example, incomplete meteorological data, which are inputs to aircraft dynamics, may compromise the correct identification of aircraft dynamics. Similarly, the missing information on the user demands in a water distribution network may alter the input-output relationship. A complete input information can aid the correct management of the water distribution without loss or waste of resources and can help enhance management of the aforementioned systems. These examples demonstrate the importance of compensating for incomplete information on input data to correctly estimate the parameters of a model.

**[0045]** In one disclosed example, the individualized model with the optimized parameters and corrections may be used to improve the control strategy for insulin dosing. Further, controller parameters such as the penalties on the controller input and the system output may be tuned via the individualized model. The predicted output of the individualized model may be output to display for analysis by a clinician or an insulin treatment parameter optimization program for the patient in relation to treatments for the patient.

**[0046]** Due to the nature of its data, the model identification in T1D is amenable to an approach using optimized sparse correction for incomplete input data while jointly identifying the model parameters of a dynamic system. Numerous experiments with incomplete in-silico training data were conducted and show that models obtained from the example method are able

to provide more accurate predictions on test data than models obtained from standard parameter identification methods. Case studies on the performance of the proposed method for ten in-silico subjects from the United States Food and Drug Administration-accepted UVA/Padova T1D Metabolic Simulator and five real-life use-case scenarios demonstrate the effectiveness of the disclosed method in relation to the prediction of glucose levels.

**[0047]** FIG. 1 is a block diagram of an example insulin delivery system 100. The system 100 allows determination of an individualized model and the application of the individualized model for treatment of a person with diabetes such as a patient 110. As explained above, the individualized model includes optimized parameters and input corrections specific to the patient 110. The system 100 includes a glucose concentration measuring device such as a continuous glucose monitor (CGM) 120 that is worn by the patient 110. In this example, the CGM 120 outputs continuous measurements of the blood glucose levels of the patient 110 in this example. An insulin pump 130 is also worn by the person 110. The insulin pump 130 supplies insulin in order to regulate blood glucose levels of the patient 110. Alternatively, other insulin administration devices such as a connected insulin pen may be used.

**[0048]** A controller 140 provides instructions to the pump 130 to provide insulin boluses to the patient 110. The controller 140 may include a control system that has one or more processors, memory and may include one or more control models, such as an individualized model 142, stored on a memory that process glucose data output from the CGM 120 and feedback insulin provided data from the pump 130, and other data to determine a bolus size of insulin that needs to be delivered to the patient 110 and sends the instructions to the pump 130. Alternatively, other sensors may be used, such as an insulin level sensor that measures the insulin level in the patient 110. In the absence of the CGM 120, glucose levels may be determined by periodic testing of glucose by taking blood fluid samples.

**[0049]** The memory is a machine readable medium that stores machine executable code in the form of an application. The controller 140 is a control system coupled to the memory. The controller 140 may include one or more processors. The application is executed by the controller 140 and executes machine executable code to collect individual data from the patient 110 to create the individualized model 142.

**[0050]** The controller 140 may be in communication with the pump 130 by a wired or wireless connection. The controller 140 sends commands to the pump 130 to deliver the determined insulin dosage. The CGM 120 may be any suitable sensor for continuous glucose monitoring and may be an under the skin sensor with a wireless connection to the controller

140. In other examples, it may be a non-invasive sensor and have a wired or wireless connection to the controller 140, for instance the Dexcom G6 CGM system manufactured by Dexcom, Inc.

**[0051]** The pump 130 may be any suitable insulin pump that is capable of receiving instructions from the controller 140 and delivering insulin boluses to the patient 110. For example, an artificial pancreas using a closed-loop system that includes an insulin pump may be used such as the Medtronic MiniMed 670G, the Tandem Control-IQ, the Omnipod 5, the CamAPS, the TidePool iLet Loop, and like products. The pump 130 may also send a signal to the controller 140 indicating the amounts and times that insulin boluses are delivered to the patient 110.

**[0052]** In this example, a mobile computing device 150 such as a smartphone carried by the patient 110 includes the controller 140, that may be a CPU, and a memory 144. As will be explained, the mobile computing device 150 may execute an application that incorporates the individualized model 142 for the individual determination of bolus calculations and the control of the insulin pump 130. The application may also include an input interface to allow the patient 110 to enter carbohydrate intakes for inputs into the individualized model 142. The mobile computing device 150 may include a communication interface such as a network transmitter receiver allowing communications to an external network such as the Internet or the Cloud 160 for upload of data related to the bolus calculation. It is to be understood that the bolus calculation described herein may be performed within applications running in the Cloud 160 in conjunction, or in place of the mobile computing device 150. The availability of collected data as will be explained, may be accessed by other health care actors via the Cloud 160. Other communication interfaces such as NFC and Bluetooth may be employed for communication with external devices such as the CGM 120 and the insulin pump 130.

**[0053]** In this example, the application may be preloaded on the mobile computer device 150 or may be offered as an application that may be downloaded to the mobile computer device 150 from a network device such as a server via a network. As explained above, parts or the entire application may be run on the Cloud 160.

**[0054]** Alternatively, the application may be executed by a controller on an insulin administration device such as the insulin pump 130. The insulin pump 130 in this example would have a communication interface to receive data from the CGM 120. In this case, the output of the application may be displayed to the user on either the pump 130 or on the mobile computer device 150.

**[0055]** The system in FIG. 1 is based on an example individualized model for T1D treatment and insulin-meal-glucose models. The main treatment of T1D is external insulin

injections, which can be administered manually or using automated insulin delivery systems - depending on the availability and personal choice. The goal is to keep blood glucose levels in a desired range. The two main elements of any insulin treatment regimen are (i) monitoring blood glucose levels, and (ii) timing and sizing the amount of insulin injections. In order to guide the decisions on (ii) based on data from (i) and patient inputs (such as meal reports), models are sought to provide predictions on glucose outcomes.

**[0056]** One such model is a Subcutaneous Oral Glucose Minimal Model (SOGMM). SOGMM is a minimal physiological model for glucose dynamics. The SOGMM relates glucose outputs with oral glucose intake and subcutaneous insulin injections. The set of differential equations (1) in SOGMM is as follows:

$$\dot{G}(t) = -\left(S_g + X_I(t)\right) \cdot G(t) + S_g \cdot G_b + \frac{R_a(t)}{BW \cdot V_g}, \quad (1a)$$

$$\dot{X}_I(t) = -p_2 \cdot X_I(t) + p_2 \cdot S_I \left( \frac{I_p(t)}{V_I \cdot BW} - I_b \right), \quad (1b)$$

$$\dot{Q}_1(t) = -k_\tau \cdot Q_1(t) + u_m(t), \quad (1c)$$

$$\dot{Q}_2(t) = -k_{abs} \cdot Q_2(t) + k_\tau \cdot Q_1(t), \quad (1d)$$

$$\dot{I}_{sc1}(t) = -(k_d) \cdot I_{sc1}(t) + u_i(t), \quad (1e)$$

$$\dot{I}_{sc2}(t) = -k_d \cdot I_{sc2}(t) + k_d \cdot I_{sc1}(t), \quad (1f)$$

$$\dot{I}_p(t) = -k_{cl} \cdot I_p(t) + k_d \cdot I_{sc2}(t), \quad (1g)$$

$$R_a(t) = f_c \cdot k_{abs} \cdot Q_2(t), \quad (1h)$$

where  $G$  (mg/dL) is the plasma glucose concentration,  $G_b$  (mg/dL) is the basal plasma glucose concentration,  $X_I$  (l/min) is the remote insulin compartment that corresponds to the portion of insulin that acts on glucose,  $Q_1$  (mg) and  $Q_2$  (mg) are the two compartments representing the oral glucose transport,  $I_{sc1}$  and  $I_{sc2}$  (mU) are the interstitial insulin transport compartments,  $I_p$  (mU) and  $I$  (mU/l) represent the insulin mass in plasma, and  $R_a$  (mg/min·kg) represents the glucose rate of appearance.  $BW$ (kg) is the person's bodyweight  $V_g$ (kg/dL) and  $V_I$  (L/kg) represent the distribution volume of glucose and insulin, respectively.  $p_2$ ,  $k_d$ ,  $k_{cl}$  are transport rates that act on the insulin-related compartments (i.e.,  $I_{sc1}$ ,  $I_{sc2}$ ,  $I_p$ ,  $X_I$ ) while  $k_\tau$  and  $k_{abs}$  are the transport rates that act on meal related compartments (i.e.,  $Q_1$ ,  $Q_2$ ). Finally,  $S_g$  (1/min) is the fractional glucose effectiveness, and  $f_c$  is the fractional meal absorption. This model has two inputs:  $u_m$  (mg/min) and  $u_i$  (mU/min) that are the carbohydrate size in an ingested meal and insulin infusion, respectively. The parameters in the set of Equations (1) are generally assumed to be time-invariant.

**[0057]** The example process uses the SOGMM from the above description with an example model identification with input correction process to create the individualized model 142 in FIG. 1. The majority of the parameters in the SOGMM are structurally identifiable. In this example, the parameters in the set of Equations (1) are categorized into  $\theta_p = (G_b, I_b, S_I, k_\tau, k_{abs}, k_d, k_{cl})$  as the set of parameters to be identified for an individual. In this example, the nominal values of the parameters  $G_b$ ,  $I_b$ , and  $S_I$  can be obtained based on average glucose of a patient over a period of time, such as three months, the prescribed basal insulin profile, and total daily insulin use, respectively. In this example, plasma glucose concentration,  $G_b$  may be derived from data collected via a sensor such as the CGM 120 in FIG. 1. The insulin at equilibrium  $I_b$ , may be read from any suitable insulin recording device or application. In this example, the insulin at equilibrium is read from a sensor on the insulin pump 130 in FIG. 1. In this example the meal transport rate,  $k_\tau$  has a nominal value of 0.0018, the meal absorption rate,  $k_{abs}$  has a nominal value of 0.0182, the insulin utilization rate,  $k_d$  has a nominal value of 0.0100, and the insulin clearance rate,  $k_{cl}$  has a nominal value of 0.1600. The nominal values are obtained from existing literature for an average population.  $\theta_f = (S_g, V_g, V_I, p_2, f_c)$  are fixed parameters that are set to their nominal values. Due to discrete-time measurements, the system in its discrete-time form is used by defining  $k$  as the discrete time-index and the metabolic state vector as  $x(k) = (G(k), X_I(k), Q_1(k), Q_2(k), I_{sc1}(k), I_{sc2}(k), I_p(k))^T$ .

**[0058]** By fixing the parameters,  $\theta_p$ , numerical simulations of the SOGMM model may be run. First, the continuous-time model is discretized via forward Euler integration to obtain the following evolution:

$$x(k+1) = f(x(k), \theta) + B_i u_i(k) + B_m u_m(k), \hat{y}(k) = Cx(k), \quad (2)$$

where  $u_i(k)$  and  $u_m(k)$  are the insulin and meal inputs, respectively, and  $\hat{y}(k) = G(k)$  is the estimated output.  $B_i = (0, 0, 0, 0, 0, \Delta t, 0)^T$ ,  $B_m = (0, 0, \Delta t, 0, 0, 0, 0)^T$ ,  $C = (1, 0, 0, 0, 0, 0, 0)^T$  with  $\Delta t$  being the time-step, and  $\theta = \{\theta_p, \theta_f\}$  are the model parameters.  $\Delta t$  is set to 5 min. The meal inputs may be collected by patient input on an application on a computing device such as the mobile computing device 150.

**[0059]** All equations in the set of Equations (1) except Equation (1a) are linear in state. As will be explained the model is required to be linear in state and inputs. A linear approximation is made to the  $X_I(k)G(k)$  term in Equation (1a) around the equilibrium point ( $X_I = 0, G = G_b$ ) and arrives in:

$$\dot{G}(k+1) \approx S_g(G_b - G(k)) - X_I(k)G_b + \frac{R_a(k)}{V_g}. \quad (3)$$

By replacing Equation (1a) with Equation (3) and some algebraic manipulation, the state evolution may be written as:

$$\begin{aligned} x(k+1) &= Ax(k) + c + B_i u_i(k) + B_m u_m(k), \\ y(k) &= Cx(k), \end{aligned} \quad (4)$$

where  $A \in \mathbb{R}^{7 \times 7}$  is derived from packing the terms in Equation (2) and  $c = (S_g G_b, 0, 0, 0, 0, 0, 0)^T$  is the term derived from Equation (3).

**[0060]** Given a time length,  $N$ , a vector of insulin inputs  $\mathbf{u}_i \in \mathbb{R}^N$ , and a vector of meal inputs  $\mathbf{u}_m \in \mathbb{R}^N$ , Equation (4) may be simulated forward to obtain the simulated outputs  $\hat{\mathbf{y}} \in \mathbb{R}^N$  in the following compact form:

$$\hat{\mathbf{y}} = \mathcal{F}(\mathbf{u}_i, \mathbf{u}_m, \theta_p), \quad (5)$$

where the  $k^{\text{th}}$  values of  $\hat{\mathbf{y}}$ ,  $\mathbf{u}_i$ ,  $\mathbf{u}_m$  are  $\hat{y}(k)$ ,  $u_i(k)$ ,  $u_m(k)$  in Equation (2), respectively. Note that  $\mathcal{F}$  is linear in inputs  $\mathbf{u}_i$  and  $\mathbf{u}_m$  but nonlinear in  $\theta_p$ . The initial state  $x(0)$  plays a negligible role in long-term simulations as its effect is dissipated after few hours into the simulation (warm-up period). In the case studies, the states are always initialized to their nominal values, which correspond to their equilibrium values derived from Equation (2) and the first 5 hours of data are used only for warm-up and then excluded from the performance evaluation.

**[0061]** The data collected from a person with T1D can be described with the following tuple:

$$\mathcal{D} := (\mathbf{y}, \mathbf{u}_i, \mathbf{u}_m). \quad (6)$$

where  $\mathbf{y} \in \mathbb{R}^N$ ,  $\mathbf{u}_i \in \mathbb{R}^N$ , and  $\mathbf{u}_m \in \mathbb{R}^N$  are the vectors of recorded glucose measurements, insulin injections, and carbohydrate intakes, respectively from the patient, and  $N$  is the length of data. As mentioned earlier, the values are recorded every 5 minutes (i.e., one day of data means  $N = 288$ ) for a patient in this example. Of course other different measurement periods may be used.

**[0062]** The standard procedure to find a personalized model is via identifying optimal parameters, denoted by  $\theta_p^*$ , through solving the following Least-Squares (LS) problem:

$$\boxed{LS} \min_{\theta_p} \|\mathbf{y} - \mathcal{F}(\mathbf{u}_i, \mathbf{u}_m, \theta_p)\|_2^2, \quad (7)$$

Solving Equation (7) is the standard approach to find  $\theta_p^*$ , but this solution is known to suffer from misreported/unreported input data. The noise in  $y$  and  $\mathbf{u}_i$  are relatively negligible, but the values in  $\mathbf{u}_m$  are more prone to errors since the estimation can be subjective and recording is at individual discretion (e.g., a person may forget to report a meal during a busy time).

**[0063]** The goal of the present method is to identify the optimized parameters,  $\theta_p^*$  in a way that the optimized parameters capture the true dynamics regardless of inaccuracies/incompleteness in the data used for model identification. The approach is to correct the carbohydrate concentrations,  $\mathbf{u}_m$  by adding a correction signal,  $\mathbf{u}_c$ , such that resulting parameter values represent the dynamics of the individual's metabolism with high fidelity as shown in FIG. 2. Given  $\mathcal{D}$  as in Equation (6), integer  $\gamma \geq 0$ , the optimal model parameters and the optimal corrections to meal input are found by minimizing the sum of the squared errors, while the number of non-zero meal correction instances does not exceed  $\gamma$ :

$$\boxed{LS - C} \min_{\theta_p, \mathbf{u}_c} \|\mathbf{y} - \mathcal{F}(\mathbf{u}_i, \mathbf{u}_m + \mathbf{u}_c, \theta_p)\|_2^2, \quad (8)$$

subject to  $\|\mathbf{u}_c\|_0 \leq \gamma$ .

where  $\|\cdot\|_2$  and  $\|\cdot\|_0$  are used to represent the Euclidean norm and the number of non-zero elements (zero semi-norm), respectively.

**[0064]** FIG. 2 shows a graph 200 of training data that includes a plot 202 representing a true output of glucose. A second graph 210 represents the training input data with input inaccuracies. A line 212 represents a recorded meal and a line 214 represents an unrecorded meal, which results in the training dataset to have inaccurate input. A second set of graphs 220 and 230 relate to the parameter identification process if the model parameters are identified without correcting the training input. The graph 220 represents uncorrected training input data. Thus, the graph 220 only includes the recorded meal in a line 222. The unrecorded meal is not used in the training procedure. The graph 230 represents the glucose output from the true measurement as shown in a plot 232 and the output predicted through a least squares model in Equation 7 based on the uncorrected training data as shown in a plot 234. The graphs 240 and 250 represent the effect of the desired corrected data that is generated from the example method. The graph 240 represents the recorded input data in a line 242. The input correction obtained by the example method is shown as a line 244. The correction obtained in a line 244 is similar to the unrecorded input data in the line 214. The graph 250 represents the glucose output from the true measurement as shown in a plot 252 and that which is predicted through a least squares model with corrections such as that shown in the graph 240 as shown in a plot 254. The outputs of the graph 250 that shows the glucose level predicted by the example optimized model is ideally reflective of the true glucose level due to application of the correction.

**[0065]** Two final graphs 270 and 280 show the learned dynamics of the model on the testing data. The graph 270 is a graph of sample test data with accurate inputs as shown in a

line 272 of an input meal. The graph 280 is a set of outputs of the glucose level. The graph 280 includes a true output data curve 282 and an identified output curve 284 generated with the model parameters obtained from the training phase with the example method. The example method compensates for the missing data in the training phase that results in the output curve 284 being similar to the true output data curve 282. The result of a model using the least square (LS) method with inaccurate data without corrections is shown as a curve 286.

**[0066]** The value of the number of corrections,  $\gamma$  is determined prior to solving Equation (8). An example algorithm for choosing  $\gamma$  is shown in FIG. 3. It is desired that the corrections,  $u_c$  is also sparse such that the corrections only act on significant differences between the model predicted and observed output.

**[0067]** An alternative to  $\|u_c\|_0 \leq \gamma$  is adding  $\|u_c\|_2^2$  to the cost in Equation (8) so that adding more and larger corrections would be penalized. Equation (8) is better suited to deal with incomplete inputs as corrections occur without a cost. The only restriction is sparsity imposed by the cardinality constraint which suits the nature of typical errors in input data in T1D (e.g., incorrect meal report).

**[0068]** The problem defined by Equation (8) falls into the category of nonlinear optimization as  $\mathcal{F}$  is nonlinear in  $\theta_p$  in Equation (1). Furthermore, the cardinality constraint  $\|u_c\|_0 \leq \gamma$  is non-convex and makes the problem combinatorial. Therefore, solving Equation (8) is challenging as (i) gradient-based optimization is not suitable to handle the cardinality constraint and (ii) the problem size (the number of decision variables in  $\theta_p$  and  $u_c$ ) is too large for global search (using methods such as simulated annealing, genetic algorithm, etc.).

**[0069]** By fixing  $u_c$ , Equation (8) becomes a standard, smooth, nonlinear least squares problem, which can be solved to local optimality using standard gradient-based methods. Furthermore,  $\|u_c\|_0 \leq \gamma$  can be cast as a set of mixed-integer constraints. While mixed integer optimization is computationally prohibitive, mature commercial solvers, such as Gurobi Optimization, are able to solve moderately sized mixed-integer linear/quadratic programs to (near) global optimality in reasonable times. However, such solvers cannot handle nonlinear constraints on  $\theta_p$ .

**[0070]** The disclosed method is an alternating minimization approach to solve Equation (8) to obtain corrections for the model. In the first step, the correction signal,  $u_c$ , is fixed and the model parameters,  $\theta_p$  is solved using a standard nonlinear least-squares solver. In the second step, the model parameters,  $\theta_p$  is fixed and the correction signal,  $u_c$ , is solved using Mixed-Integer Quadratic Programming (MIQP). The two steps are alternated until convergence. In

the example, convergence is defined as the latest solution cost of Equation (9) below being within 5% of the previous iteration's cost while the maximum change in the parameters in  $\theta_p$  remains within 10% range of their values from the previous iteration. The pseudo-code of the described two-step process is shown in an Algorithm in FIG. 4. The details are explained in the following.

**[0071]** Given  $\mathbf{u}_c$ , the following non-linear least squares problem in Equation (8) is solved:

$$\min_{\theta_p} \|\mathbf{y} - \mathcal{F}(\mathbf{u}_i, \mathbf{u}_m + \mathbf{u}_c, \theta_p)\|_2^2 + (\theta_p - \bar{\theta}_p)' \Lambda (\theta_p - \bar{\theta}_p), \quad (9)$$

where  $\bar{\theta}_p$  stands for the nominal parameters and  $\Lambda$  is the regularization matrix. While Equation (9) is a nonlinear program, it is smooth and can be solved to local optimality. When solving Equation (9) in the intermediate steps of Algorithm 2 in FIG. 4, Equation (9) is warm started with the previous model parameters  $\theta_p$ . In the very first iteration, all elements of  $\mathbf{u}_c$  are set to zero and the population-level values for the model parameters  $\theta_p$  are taken from known studies to provide a reasonable initial guess of the initial values for the model parameters  $\theta_p$ .

**[0072]** The next phase is to choose the number of corrections ( $\gamma$ ) via an example algorithm such as the algorithm in FIG. 3. A “significant difference” between the model prediction and the real output implies incomplete input within the few hours leading to the difference. The significant difference is defined as an Absolute Relative Difference (ARD) value  $\geq 20/20\%$ . The ARD is calculated as below:

$$\text{ARD}(y, \hat{y}) = \begin{cases} |y - \hat{y}| & y \leq 100 \frac{\text{mg}}{\text{dL}}, \\ 100 \frac{|y - \hat{y}|}{y} & \text{otherwise.} \end{cases} \quad (10)$$

The selected ARD threshold is about twice the mean ARD of the FDA-approved CGM devices between 2016 and 2021, hence regular CGM noise does not increase the estimated number of corrections. After solving Equation (9) for the first time, an estimate of the number of corrections  $\gamma$  is obtained such as through the Algorithm in FIG. 4.

**[0073]** Note that every time the number of corrections,  $\gamma$  increases, increasing the corrections again is avoided for the subsequent four hours regardless of the ARD value within that period (see Algorithm 1). This is due to the observation that meal intake disturbs glucose levels significantly for three hours and the total absorption can take up to six hours. Thus, it is safe to assume that each correction can be used to restore the compromised input-output relationship for the subsequent four hours in the identification process in this example.

**[0074]** The Mixed-Integer Quadratic Programming (MIQP) technique is used to determine optimum input corrections,  $\mathbf{u}_c$ , and model parameters for a given training data set. Using the

big-M method,  $\|\mathbf{u}_c\|_0 \leq \gamma$  is cast as mixed-integer constraints and the following MIQP is obtained:

$$\begin{aligned} \min_{\mathbf{u}_c, \mathbf{v}, \mathbf{b}} \quad & \|\mathbf{y} - \mathcal{F}(\mathbf{u}_i, \mathbf{u}_m + \mathbf{u}_c, \theta_p)\|_2^2, \\ \text{subject to} \quad & uc(k) - v(k) \geq -M(1 - b(k)), \\ & uc(k) - v(k) \leq M(1 - b(k)), \\ & -Mb(k) \leq uc(k) \leq Mb(k), \\ & -M \leq v(k) \leq M, \\ & b(k) \in \{0, 1\}, k = 1, \dots, N, \\ & \sum_{k=1}^N b(k) = \gamma, \end{aligned} \tag{11}$$

where  $v$  and  $b$  are auxiliary continuous and binary variables, respectively, and  $M$  is a large number that is an upper bound for any correction magnitude. It is straightforward to check that if  $b(k) = 0$  in the constraints, then  $uc(k) = 0$  and if  $b(k) = 1$ , then  $uc(k)$  can take any value in the range  $[-M, M]$ .

**[0075]** The above described method for producing an individualized model was evaluated through in-silico studies using 10 adult subjects in the UVA/Padova simulator. The experiments were two days long with three main meals and one snack on each day. Total carbohydrate intake is selected similar to real life situations. Five distinct scenarios for different unreported or misreported information in  $D$  reflect situations that can be experienced in real life. To better resemble the real life, CGM noise was also incorporated.

**[0076]** A first scenario was complete information where all inputs are recorded accurately. A second scenario is unreported snacks. This scenario reflects that persons tend to not report the snacks that they do not bolus for. This scenario implements this behavior for the bed-time snack on both days that are 40g and 30g, respectively. A third scenario is miscalculated meals. Carbohydrate counting is a major challenge in daily T1D care and errors are frequent. For this scenario, a 30% overestimation on the biggest meal, lunch on Day 1, and a 30% underestimation error on the breakfast of Day 2 was implemented. A fourth scenario is delays in eating the bolused meal. While this scenario is less frequent compared to the earlier scenarios, persons may have delays in eating the meal that they already bolused for. This may be due to situations such as a delay in the arrival of meal order. In terms of the challenge, this scenario involves both misreporting (a full meal is reported when no meal is actually eaten) and unreporting (a meal is eaten at a later time with no report) leading to a combined information inaccuracy in the input data. This situation is implemented by introducing a one-hour delay to eating the lunch on Day 1 that was bolused for and reported at 2PM but eaten at

3PM. A fifth scenario is pump occlusion where the misrecorded input is insulin. The input data suggests insulin being delivered while the insulin is not actually delivered, leading to high glucose values. It is difficult to detect pump occlusion in practice. Pump occlusion for the breakfast bolus and the following two-hours on Day 2 were implemented.

**[0077]** The five scenarios are illustrated in the graphs in FIG. 5A-5B. FIG. 5A is a table 500 showing the references symbols in the graphs 510, 520, 530, 540, and 550 in FIG. 5B. The table 500 includes a column 502 that lists whether the data correctly reported, misreported, or unreported. A second column 504 lists bolus insulin input for both injected and not injected bolus. These show either insulin injections from the pump or other source and meal-independent insulin injections, i.e., basal insulin and correction bolus. A third column 506 lists meal inputs.

**[0078]** Recorded data for meal and insulin are reported with a line symbol and a dot symbol and a cross, respectively in the graphs in FIG. 5B. The inaccuracies for meal inputs are shown as a dashed line, while inaccuracies for injected insulin is shown in a second type of dot and uninjected insulin is shown as a second type of cross. A final line pattern shows unreported meal inputs.

**[0079]** The graph 510 in FIG. 5B shows the description of a complete information scenario. Thus, the graph 510 includes a series of dots such as the dot 512 that represent times of injected insulin and a series of lines such as the line 514 that represents a meal input. A line 516 represents meal independent injected insulin. The graph 520 shows the results of an unreported snacks scenario. The graph 520 thus includes two lines 522 that represent unreported snacks at different times. The graph 530 shows the results of a miscalculated meals scenario. The graph 530 includes two dashed lines 532 that represent miscalculated meals at different times. The graph 540 shows the results of delays in eating a bolused meal. The graph 540 shows a line 542 that represents a miscalculated meal and a dashed line 544 that represents a misreported meal. The graph 550 shows the results of a pump occlusion scenario. The graph 550 shows a cross symbol 552 that represents a misreported not injected bolus input.

**[0080]** After obtaining model parameters for each in-silico subject, the prediction capabilities of the models were evaluated on a separate test data where input-output information were reported accurately. If the example least-squares correction (LS-C) method is successful in recovering the effects of inaccurate input data in the training data, the resulting model will have improved prediction performance on the test data. A two-day long test scenario was designed with three meals per day and one snack on each day. Meal times, meal sizes, and unbolused snack times were different from the training scenario and were designed as follows:

Day 1 (breakfast of 40g at 7AM, snack of 20g at 10AM, lunch of 70g at 2PM, dinner of 60g at 8PM), Day 2 (breakfast of 30g at 8AM, snack of 10g at 10AM, lunch of 50g at 1PM, dinner of 80g at 8PM).

**[0081]** To assess the performances of the nominal model, the model identified with least squares (LS) (from solving Equation (7)), and the model identified by using the example LS-C method, root mean-square error (RMSE) predictions for each model were calculated on the test data. FIG. 6 shows a table 600 that shows the RMSE values across 10 subjects for models obtained from each of the five scenarios and the three methods. Thus, results are obtained for the glucose predictions using the nominal model, a model identified by the known least squares (LS) method, and a model identified by the example least square correction (LS-C) method. A paired t-test was used to evaluate whether the difference in performance was significant between the least squares method and the example least squares with correction method. The sample size was equal to the number of in-silico subjects, 10, and two-tailed significance threshold  $\alpha = 0.05$  was used in all evaluations reported in the table 600 in FIG. 6.

**[0082]** As expected, no difference was observed for models trained on data with complete input information, while the example LS-C algorithm learns a more accurate model that leads to significantly lower test scenario RMSE for the four other scenarios ( $p < 0.05$ ) that had input data inaccuracies. The largest performance improvement was obtained in the second scenario of missing recording of snacks. This scenario represents a critical, yet common, condition of two unreported meals that has a significant impact on the glucose levels. A small improvement is reported on the third scenario of miscalculated meals and incomplete data is successfully recovered in the fourth scenario. The fifth scenario of CGM occlusion demonstrates that the example algorithm can also successfully handle a condition where the inaccuracy does not involve a meal. The example LS-C based model suggested the lowest number of corrections for the scenario of complete information and the highest number of corrections for the fifth scenario of CGM pump occlusion, when evaluated via median [IQR]. Thus the median number of corrections estimated across in-silico subjects was 0 for Scenario I with the interquartile range of the number of corrections estimated across in -silico subjects being 0. Similarly, the respective medium number of corrections and interquartile ranges were 2.5 [1] for Scenario II, 3 [1] for Scenario III, 3 [2] for Scenario IV, and 6 [1.75] for Scenario V. These results show how the algorithm successfully estimated the correct numbers of incorrectly reported inputs for each scenario.

**[0083]** FIG. 7A shows a series of boxplots 700 showing the estimated number of corrections ( $\gamma$ ) required for each of the five scenarios. FIG. 7B shows a series of boxplots 750

of the estimated number of corrections ( $\gamma$ ) required for different ARD thresholds. The data for the boxplot 700 was collected across all subjects per scenario for the selected ARD threshold (20/20%). The boxplot 700 illustrates the distribution of the corrections,  $\gamma$  across all subjects per scenario. A posteriori analysis of how estimated  $\gamma$  changes for different thresholds of ARD across all scenarios is shown in the boxplot 750 in FIG. 7B. The plot 750 shows data collected across all subjects and scenarios for different ARD threshold choices.

**[0084]** FIGs. 8A shows a graph 810 of the training input data and a graph 820 of the model training for glucose outputs and for the fourth scenario of delays in eating the bolused meal. The graph 810 of training input data for Subject #6 when a 2PM meal was digested one hour after it was recorded. The graph 810 shows a set of insulin injections symbolized by dots 812, a carbohydrate input symbolized by lines 814, and lines 816 representing optimal corrections generated by the example method.

**[0085]** The graph 820 shows the sample training output for Subject #6 when 2PM meal was digested one hour after it was recorded. The graph 820 includes a plot 822 of the true glucose output, a plot 824 of the model based on the least square method, and a plot 826 of the model based on the example least squares with corrections (LS-C) method. The graph 820 shows model fits for identification with standard approach described in Equation 7 in the plot 824 and the identification of the example LS-C method in the plot 826 (green line) compared with the true data in the plot 822.

**[0086]** FIG. 8B shows a graph 850 shows the testing outcomes obtained by the models identified on the training data shown in the graph 810 for the fourth scenario of delays in eating the bloused meal. The graph 850 includes a plot 852 of the true glucose output, a plot 854 of the output of the model based on the least square (LS) method, and a plot 856 of the output of the model based on the example least squares with corrections (LS-C) method. Sample test output for Subject #6 with the models identified with standard approach described in Equation (7) in the plot 854 and with the example least-squares with correction (LS-C) algorithm in the plot 856 are compared with the test data in the plot 852. The graph 860 shows the meal and insulin input data that are reported with a series of lines 862 and dots 864, respectively.

**[0087]** The graphs in FIG. 8A-8B show how the presence of inaccuracies in the data, e.g., 2PM meal on day 1 digested one hour after it was recorded, can compromise the correct identification of the model parameters when using the standard LS approach from Equation (7). The graph 850 in FIG. 8B shows how the example LS-C method allows to handle the inaccuracy in the identification phase achieving a superior glucose prediction performance on the test data compared to LS.

**[0088]** The example method described above may be used to identify metabolic parameters of 8 participants who participated in a supervised pregnancy study. The run-in portion of the study is used to identify model parameters and the supervised portion of the study is used to test the performance of the identified model.

**[0089]** While the parameters in the example individualized model are assumed to be time-invariant, it is known that the insulin sensitivity has a periodic nature with a 24-hour period and most of the time individuals have their own specific diurnal pattern - albeit non-periodic events such as physical exercise may cause temporary significant changes. In order to implement the method on the free-living data, a truncated Fourier series is used to model the periodic changes of insulin sensitivity:

**[0090]** For this evaluation, the supervised study data with available ground truth was used. FIG. 9 shows a graph 910 that shows a comparison of the true glucose data in a plot 912, a plot 914 of glucose simulated from the least squares method, and a plot 916 of glucose simulated from the example least square with correction method over a time index. A graph 920 shows the carbohydrate input data. A set of lines 922 constitute announced carbohydrate data. A set of lines 924 show the corrections generated from the example least squares with correction method that lead to the more accurate simulation in the plot 916. A graph 930 shows a series of lines 932 of bolus insulin levels relative to the time index. The graph 930 also shows a series of line 934 of basal insulin levels relative to the time index.

**[0091]** A flow diagram 1000 in FIG. 10 is representative of example machine readable instructions for the process of generating an individualized model for an individual. In this example, the machine readable instructions comprise an algorithm for execution by: (a) a processor; (b) a controller; and/or (c) one or more other suitable processing device(s). The algorithm may be embodied in software stored on tangible media such as flash memory, CD-ROM, floppy disk, hard drive, digital video (versatile) disk (DVD), or other memory devices. However, persons of ordinary skill in the art will readily appreciate that the entire algorithm and/or parts thereof can alternatively be executed by a device other than a processor and/or embodied in firmware or dedicated hardware in a well-known manner (e.g., it may be implemented by an application specific integrated circuit [ASIC], a programmable logic device [PLD], a field programmable logic device [FPLD], a field programmable gate array [FPGA], discrete logic, etc.). For example, any or all of the components of the interfaces can be implemented by software, hardware, and/or firmware. Also, some or all of the machine readable instructions represented by the flowcharts may be implemented manually. Further, although the example algorithm is described with reference to the flowchart illustrated in FIG.

10, persons of ordinary skill in the art will readily appreciate that many other methods of implementing the example machine readable instructions may alternatively be used. For example, the order of execution of the blocks may be changed, and/or some of the blocks described may be changed, eliminated, or combined.

**[0092]** A set of training input data such as carbohydrate levels over a period of time and corresponding times and a set of training output data such as glucose concentrations over the period of time is collected from the individual patient (1010). The initial values for model parameters are set (1012). The estimated maximum number of corrections is then determined from the input data and output data (1014). The model parameters are solved with a non-linear least squares solver while holding the correction constant (1016). The correction is solved by executing Mixed-Integer Quadratic Programming while incorporating the solved model parameters in the dynamic model (1018).

**[0093]** The routine determines whether the solution cost of the least squares solver is within a predetermined percentage of the previous iteration (1020). If the solution cost is outside of the predetermined percentage, the routine loops back to 1016 for another iteration of determining parameters and corrections. If the solution cost is within the predetermined percentage, the routine determines whether the maximum change in the parameters is within a predetermined percentage range (1022). If the maximum change is outside of the predetermined percentage range, the routine loops back to 1016 for another iteration of determining parameters and corrections. If the maximum change is inside the predetermined percentage range, the model with the final optimized parameters and the final corrections are stored (1024).

**[0094]** The results show the importance of accurate input information in model identification and that the example least squares with correction method (LS-C), is effective in reconciling the unfavorable effects of imperfect input in model identification for T1D. While misreported meals are the primary motivation to apply corrections through the meal input, other correction may be applied for phenomena such as different macronutrients, acute psychological stress, and physical activities are unreported and unmodeled, that may have profound effects on insulin-glucose dynamics. The effects of these can also be approximately modeled by artificial meals. For instance, physical exercise is known to cause a sharp decrease in glucose values and hence can be roughly viewed as the person has taken negative meal. Conditions that impact the model parameters, such as sickness or pregnancy, appear in the model parameter values instead of meal corrections, as long as the parameters are sufficient to capture these dynamics. Similarly, the example can be extended to correct significant output

inaccuracies that can affect model identification outcomes, such as low glucose readings due to compression on the CGM sensor. The example least square with correction method can also handle inputs in the form of a step or a ramp instead of an impulse allowing the integration of the effect of combined inaccuracies.

**[0095]** Other objects and advantages of the present invention will become obvious to the reader and it is intended that these objects and advantages are within the scope of the present invention.

**[0096]** Embodiments of the present disclosure may comprise or utilize a special purpose or general-purpose computer including computer hardware, such as, for example, one or more processors and system memory, as discussed in greater detail below. Embodiments within the scope of the present disclosure also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. In particular, one or more of the processes described herein may be implemented at least in part as instructions embodied in a non-transitory computer-readable medium and executable by one or more computing devices (e.g., any of the media content access devices described herein). In general, a processor (e.g., a microprocessor) receives instructions, from a non-transitory computer-readable medium, (e.g., a memory, etc.), and executes those instructions, thereby performing one or more processes, including one or more of the processes described herein.

**[0097]** Computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-executable instructions are non-transitory computer-readable storage media (devices). Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, embodiments of the disclosure can comprise at least two distinctly different kinds of computer-readable media: non-transitory computer-readable storage media (devices) and transmission media.

**[0098]** Non-transitory computer-readable storage media (devices) includes RAM, ROM, EEPROM, CD-ROM, solid state drives (“SSDs”) (e.g., based on RAM), Flash memory, phase-change memory (“PCM”), other types of memory, other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

**[0099]** A “network” is defined as one or more data links that enable the transport of electronic data between computer systems and/or modules and/or other electronic devices. When information is transferred or provided over a network or another communications

connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmissions media can include a network and/or data links which can be used to carry desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. Combinations of the above should also be included within the scope of computer-readable media.

**[00100]** Further, upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures can be transferred automatically from transmission media to non-transitory computer-readable storage media (devices) (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a “NIC”), and then eventually transferred to computer system RAM and/or to less volatile computer storage media (devices) at a computer system. Thus, it should be understood that non-transitory computer-readable storage media (devices) can be included in computer system components that also (or even primarily) utilize transmission media.

**[00101]** Computer-executable instructions comprise, for example, instructions and data which, when executed at a processor, cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. In one or more embodiments, computer-executable instructions are executed on a general purpose computer to turn the general purpose computer into a special purpose computer implementing elements of the disclosure. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural marketing features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described marketing features or acts described above. Rather, the described marketing features and acts are disclosed as example forms of implementing the claims.

**[00102]** Those skilled in the art will appreciate that the disclosure may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, tablets, pagers, routers, switches, and the like. The disclosure may also be practiced in distributed system environments where local and remote computer systems, which are linked

(either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

**[00103]** Embodiments of the present disclosure can also be implemented in cloud computing environments. In this description, “cloud computing” is defined as an un-subscription model for enabling on-demand network access to a shared pool of configurable computing resources. For example, cloud computing can be employed in the marketplace to offer ubiquitous and convenient on-demand access to the shared pool of configurable computing resources. The shared pool of configurable computing resources can be rapidly provisioned via virtualization and released with low management effort or service provider interaction, and then scaled accordingly.

**[00104]** A cloud-computing un-subscription model can be composed of various characteristics such as, for example, on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service, and so forth. A cloud-computing un-subscription model can also expose various service un-subscription models, such as, for example, Software as a Service (“SaaS”), a web service, Platform as a Service (“PaaS”), and Infrastructure as a Service (“IaaS”). A cloud-computing un-subscription model can also be deployed using different deployment un-subscription models such as private cloud, community cloud, public cloud, hybrid cloud, and so forth. In this description and in the claims, a “cloud-computing environment” is an environment in which cloud computing is employed.

**[00105]** In one example, a computing device may be configured to perform one or more of the processes described above. the computing device can comprise a processor, a memory, a storage device, an I/O interface, and a communication interface, which may be communicatively coupled by way of a communication infrastructure. In certain embodiments, the computing device can include fewer or more components than those described above.

**[00106]** In one or more embodiments, the processor includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute instructions for digitizing real-world objects, the processor may retrieve (or fetch) the instructions from an internal register, an internal cache, the memory, or the storage device and decode and execute them. The memory may be a volatile or non-volatile memory used for storing data, metadata, and programs for execution by the processor(s). The storage device includes storage, such as a hard disk, flash disk drive, or other digital storage

device, for storing data or instructions related to object digitizing processes (e.g., digital scans, digital models).

**[00107]** The I/O interface allows a user to provide input to, receive output from, and otherwise transfer data to and receive data from computing device. The I/O interface may include a mouse, a keypad or a keyboard, a touch screen, a camera, an optical scanner, network interface, modem, other known I/O devices or a combination of such I/O interfaces. The I/O interface may include one or more devices for presenting output to a user, including, but not limited to, a graphics engine, a display (e.g., a display screen), one or more output drivers (e.g., display drivers), one or more audio speakers, and one or more audio drivers. In certain embodiments, the I/O interface is configured to provide graphical data to a display for presentation to a user. The graphical data may be representative of one or more graphical user interfaces and/or any other graphical content as may serve a particular implementation.

**[00108]** The communication interface can include hardware, software, or both. In any event, the communication interface can provide one or more interfaces for communication (such as, for example, packet-based communication) between the computing device and one or more other computing devices or networks. As an example and not by way of limitation, the communication interface may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for communicating with a wireless network, such as a WI-FI.

**[00109]** Additionally, the communication interface may facilitate communications with various types of wired or wireless networks. The communication interface may also facilitate communications using various communication protocols. The communication infrastructure may also include hardware, software, or both that couples components of the computing device to each other. For example, the communication interface may use one or more networks and/or protocols to enable a plurality of computing devices connected by a particular infrastructure to communicate with each other to perform one or more aspects of the digitizing processes described herein. To illustrate, the image compression process can allow a plurality of devices (e.g., server devices for performing image processing tasks of a large number of images) to exchange information using various communication networks and protocols for exchanging information about a selected workflow and image data for a plurality of images.

**[00110]** It should initially be understood that the disclosure herein may be implemented with any type of hardware and/or software, and may be a pre-programmed general purpose computing device. For example, the system may be implemented using a server, a personal computer, a portable computer, a thin client, or any suitable device or devices. The disclosure

and/or components thereof may be a single device at a single location, or multiple devices at a single, or multiple, locations that are connected together using any appropriate communication protocols over any communication medium such as electric cable, fiber optic cable, or in a wireless manner.

**[00111]** It should also be noted that the disclosure is illustrated and discussed herein as having a plurality of modules which perform particular functions. It should be understood that these modules are merely schematically illustrated based on their function for clarity purposes only, and do not necessarily represent specific hardware or software. In this regard, these modules may be hardware and/or software implemented to substantially perform the particular functions discussed. Moreover, the modules may be combined together within the disclosure, or divided into additional modules based on the particular function desired. Thus, the disclosure should not be construed to limit the present invention, but merely be understood to illustrate one example implementation thereof.

**[00112]** The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In some implementations, a server transmits data (e.g., an HTML page) to a client device (e.g., for purposes of displaying data to and receiving user input from a user interacting with the client device). Data generated at the client device (e.g., a result of the user interaction) can be received from the client device at the server.

**[00113]** Implementations of the subject matter described in this specification can be implemented in a computing system that includes a back end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network (“LAN”) and a wide area network (“WAN”), an inter-network (e.g., the Internet), and peer-to-peer networks (e.g., ad hoc peer to-peer networks).

**[00114]** Implementations of the subject matter and the operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their

structural equivalents, or in combinations of one or more of them. Implementations of the subject matter described in this specification can be implemented as one or more computer programs, i.e., one or more modules of computer program instructions, encoded on computer storage medium for execution by, or to control the operation of, data processing apparatus. Alternatively or in addition, the program instructions can be encoded on an artificially generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. A computer storage medium can be, or be included in, a computer-readable storage device, a computer-readable storage substrate, a random or serial access memory array or device, or a combination of one or more of them. Moreover, while a computer storage medium is not a propagated signal, a computer storage medium can be a source or destination of computer program instructions encoded in an artificially generated propagated signal. The computer storage medium can also be, or be included in, one or more separate physical components or media (e.g., multiple CDs, disks, or other storage devices).

**[00115]** The operations described in this specification can be implemented as operations performed by a “control system” on data stored on one or more computer-readable storage devices or received from other sources.

**[00116]** The term “control system” encompasses all kinds of apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, a system on a chip, or multiple ones, or combinations, of the foregoing. The apparatus can include special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit). The apparatus can also include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, a cross-platform runtime environment, a virtual machine, or a combination of one or more of them. The apparatus and execution environment can realize various different computing model infrastructures, such as web services, distributed computing and grid computing infrastructures.

**[00117]** A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, declarative or procedural languages, and it can be deployed in any form, including as a stand alone program or as a module, component, subroutine, object, or other unit suitable for use in a computing environment. A computer program may, but need not,

correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

**[00118]** The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform actions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

**[00119]** Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a processor for performing actions in accordance with instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a Global Positioning System (GPS) receiver, or a portable storage device (e.g., a universal serial bus (USB) flash drive), to name just a few. Devices suitable for storing computer program instructions and data include all forms of non volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

**[00120]** While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Although the invention has been illustrated and described with respect to one or more implementations, equivalent alterations and modifications will occur or be known to others skilled in the art upon the reading and understanding of this specification and the annexed

drawings. In addition, while a particular feature of the invention may have been disclosed with respect to only one of several implementations, such feature may be combined with one or more other features of the other implementations as may be desired and advantageous for any given or particular application. Thus, the breadth and scope of the present invention should not be limited by any of the above-described embodiments. Rather, the scope of the invention should be defined in accordance with the following claims and their equivalents.

**[00121]** While various embodiments of the present disclosure have been described above, it should be understood that they have been presented by way of example only, and not limitation. Numerous changes to the disclosed embodiments can be made in accordance with the disclosure herein, without departing from the spirit or scope of the disclosure. Thus, the breadth and scope of the present disclosure should not be limited by any of the above described embodiments. Rather, the scope of the disclosure should be defined in accordance with the following claims and their equivalents.

## WHAT IS CLAIMED IS:

1. A method for determining a modified dynamic model for predicting an output from a dynamic system based on an input, the method comprising:
  - collecting output data from the dynamic system over a period of time;
  - collecting input data from the dynamic system over a period of time;
  - optimizing parameters of a dynamic system model via solving a nonlinear least squares problem based on the collected data while fixing a correction value;
  - determining a modified correction value based on a mixed integer quadratic program executed on the collected data while fixing a constant value for the parameters of the dynamic system model; and
  - providing a final correction value and final optimized parameters to the dynamic system model from iterating the optimizing the parameters of the dynamic system model and determining the modified correction value.
2. The method of claim 1, wherein the dynamic system is a patient and the input data relates to a drug taken by the patient.
3. The method of claim 1, wherein the dynamic system is one of a biological system, an industrial system, a population system, a market system, or a resource distribution system.
4. The method of claim 1, further comprising determining a maximum number of correction values based the number of times an absolute relative difference value of collected output data and an output of an initial fit from the least squares method exceeds a threshold value.
5. A method for determining an individualized model for predicting glucose levels of a patient, the method comprising:
  - collecting glucose concentration output data from the patient over a period of time;
  - collecting carbohydrate size and insulin infusion input data from the patient over a period of time;
  - optimizing parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected data while fixing a constant time and a constant amount of a correction value;
  - determining the time and the amount of a modified correction value based on executing a mixed integer quadratic program on the collected data while incorporating the optimized parameters of the glucose dynamics model; and

providing a final correction value and final optimized parameters of the glucose dynamics model based on iterating the optimizing parameters of the glucose dynamics model and determining the time and the amount of the modified correction value.

6. The method of claim 5, wherein the parameters of the glucose dynamics model include glucose at equilibrium, insulin at equilibrium, insulin sensitivity, meal transport rate, meal absorption rate, insulin utilization rate, and insulin clearance rate.
7. The method of claim 6, wherein the collecting glucose data includes periodically collecting sets of recorded glucose measurements from the patient, and wherein the collecting the input data includes periodically collecting data on insulin injections and carbohydrate intakes from the patient.
8. The method of claim 6, wherein the glucose measurements are collected from a glucose monitor, the carbohydrate intake is entered from a patient input to a computing device, and the insulin injections are determined from an insulin pump.
9. The method of claim 7, wherein optimizing parameters of the glucose dynamics model includes minimizing the sum of the squared errors from the nonlinear least squares problem while the number of corrections does not exceed a predetermined value.
10. The method of claim 9, wherein the predetermined value is determined by the number of times an absolute relative difference value of a recorded glucose trace and an output of an initial fit from the least squares method exceeds a threshold value.
11. The method of claim 10, wherein the predetermined number of corrections is limited by a predetermined time period.
12. The method of claim 6, wherein the correction value compensates for incomplete input data including at least one of missing meal reports, miscalculated meals, delay in eating a meal, and insulin pump occlusion.
13. The method of claim 6 further comprising storing the model with the final optimized parameters and final correction value in a storage device.
14. A system for determining an individualized glucose dynamics model for a patient, the system comprising:
  - a memory storing machine executable code;
  - a control system coupled to the memory comprising one or more processors, the control system configured to execute the machine executable code to cause the control system to:
    - collect glucose data from the patient over a period of time;
    - collect carbohydrate input data from the patient over a period of time;

optimize parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected data while fixing a constant time and a constant amount of a correction value;

determine the time and the amount of a modified correction value based on executing a mixed integer quadratic program on the collected data while incorporating the optimized parameters of the glucose dynamics model; and

provide a final correction signal value and final optimized parameters of the glucose dynamics model based on iterating the optimizing parameters of the glucose dynamics model and determining the time and the amount of the modified correction value.

15. The system of claim 14, wherein the parameters of the glucose dynamics model include glucose at equilibrium, insulin at equilibrium, insulin sensitivity, meal transport rate, real absorption rate, insulin utilization rate, and insulin clearance rate.

16. The system of claim 15, wherein the collection of glucose data includes periodically collecting sets of recorded glucose measurements from the patient, and wherein the collection of the input data includes periodically collecting data on insulin injections and carbohydrate intakes from the patient.

17. The system of claim 14, further comprising:

a glucose monitor recording the glucose measurements;

a computing device with a patient input to enter the carbohydrate intakes; and

an insulin pump configured to collect the insulin injections.

18. The system of claim 15, wherein the optimization of parameters of the glucose dynamics model includes minimizing the sum of the squared errors from the nonlinear least squares problem while the number of corrections does not exceed a predetermined value.

19. The system of claim 18, wherein the predetermined value is determined by the number of times an absolute relative difference value of a recorded glucose trace and an output of an initial fit from the least squares method exceeds a threshold value.

20. The system of claim 19, wherein the predetermined number of corrections is limited by a predetermined time period.

21. The system of claim 14, wherein the correction value compensates for incomplete input data including at least one of missing meal reports, miscalculated meals, delay in eating a meal, and insulin pump occlusion.

22. The system of claim 14 further comprising a storage device storing the model with the final optimized parameters and final correction value.

23. A non-transitory machine readable medium having stored thereon instructions for performing a method comprising machine executable code which when executed by at least one machine, causes the machine to:

collect glucose data from the patient over a period of time;

collect carbohydrate input data from the patient over a period of time;

optimize parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected data while fixing a constant time and a constant amount of a correction value;

determine the time and the amount of a modified correction value based on executing a mixed integer quadratic program on the collected data while incorporating the optimized parameters of the glucose dynamics model; and

provide a final correction value and final optimized parameters of the glucose dynamics model based on iterating the optimizing parameters of the glucose dynamics model and determining the time and the amount of the modified correction value.

24. A system for managing glucose of a patient, the system comprising:

an insulin pump configured to deliver insulin into a patient;

a memory storing machine executable code;

a control system coupled to the memory comprising one or more processors, the control system configured to execute the machine executable code to cause the control system to:

accept a carbohydrate size input from the patient;

accept an insulin infusion input from the patient;

modify the carbohydrate size input with a correction value of an individualized glucose dynamics model including optimized parameters, wherein the correction value and optimized parameters are determined by: collecting glucose concentration output training data from the patient and collecting carbohydrate size and insulin infusion input training data from the patient over a period of time; and iteratively optimizing parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected training data while fixing a constant time and amount of a correction value; and determining the time and amount of a modified correction value based on executing a mixed integer quadratic program on the collected training data while fixing a constant value for the parameters of the glucose dynamics model;

provide the insulin infusion input and the modified carbohydrate size input to the individualized glucose dynamics model to determine a prediction of glucose outcome; and

send a command to the insulin pump to deliver a determined insulin dosage based on the predicted glucose outcome.

25. The system of claim 24, wherein the parameters of the glucose dynamics model include glucose at equilibrium, insulin at equilibrium, insulin sensitivity, meal transport rate, real absorption rate, insulin utilization rate, and insulin clearance rate.
26. The system of claim 24, further comprising a computing device in communication with the control system, the computing device including a patient input to enter the carbohydrate intake, and wherein the insulin pump is configured to collect the insulin infusion input.
27. The system of claim 25, wherein the optimization of parameters of the glucose dynamics model includes minimizing the sum of the squared errors from the nonlinear least squares problem while the number of corrections does not exceed a predetermined value.
28. The system of claim 27, wherein the predetermined value is determined by determining the number of times an absolute relative difference value of a recorded glucose trace and an output of an initial fit from the least squares method exceeds a threshold value.
29. The system of claim 28, wherein the predetermined number of corrections is limited by a predetermined time period.
30. The system of claim 24, wherein the correction value compensates for incomplete input data including at least one of missing meal reports, miscalculated meals, delay in eating a meal, and insulin pump occlusion.
31. A method of providing treatment for a diabetic patient, the method comprising:
  - accepting a carbohydrate size input from the patient;
  - accepting an insulin infusion input from the patient;
  - modifying the carbohydrate size input with a correction value of an individualized glucose dynamics model including optimized parameters, wherein the correction value and optimized parameters are determined by: collecting glucose concentration output training data from the patient and collecting carbohydrate size and insulin infusion input training data from the patient over a period of time; and iteratively optimizing parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected training data while fixing a constant time and amount of a correction value; and determining the time and amount of a modified correction value based on executing a mixed integer quadratic

program on the collected training data while fixing a constant value for the parameters of the glucose dynamics model;

providing the insulin infusion input and the modified carbohydrate size input to the model to determine a prediction of glucose outcome; and

sending a command to an insulin pump to deliver a determined insulin dosage based on the predicted glucose outcome.

32. The method of claim 31, wherein the parameters of the glucose dynamics model include glucose at equilibrium, insulin at equilibrium, insulin sensitivity, meal transport rate, real absorption rate, insulin utilization rate, and insulin clearance rate.

33. The method of claim 31, wherein the carbohydrate intake input is received from a computing device including a patient input to enter the carbohydrate intake, and wherein the insulin infusion input is collected from an insulin pump.

34. The method of claim 32, wherein the optimization of parameters of the glucose dynamics model includes minimizing the sum of the squared errors from the nonlinear least squares problem while the number of corrections does not exceed a predetermined value.

35. The method of claim 34, wherein the predetermined value is determined by determining the number of times an absolute relative difference value of a recorded glucose trace and an output of an initial fit from the least squares method exceeds a threshold value.

36. The system of claim 35, wherein the predetermined number of corrections is limited by a predetermined time period.

37. The system of claim 31, wherein the correction value compensates for incomplete input data including at least one of missing meal reports, miscalculated meals, delay in eating a meal, and insulin pump occlusion.

38. A non-transitory machine readable medium having stored thereon instructions for performing a method comprising machine executable code which when executed by at least one machine, causes the machine to:

accept a carbohydrate size input from a patient;

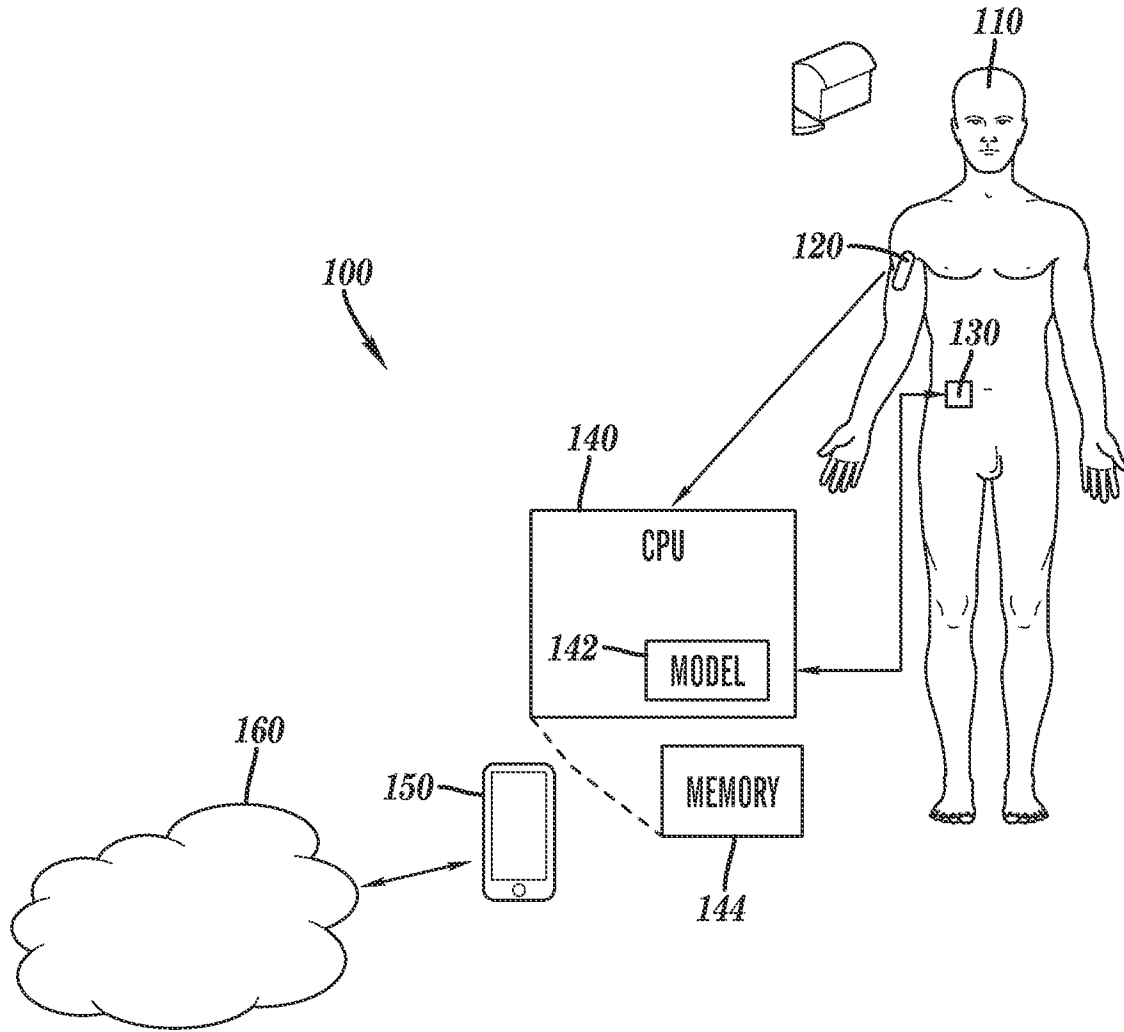
accept an insulin infusion input from the patient; and

modify the carbohydrate size input with a correction value of an individualized glucose dynamics model including optimized parameters, wherein the correction value and optimized parameters are determined by: collecting glucose concentration output training data from the patient and collecting carbohydrate size and insulin infusion input training data from the patient over a period of time; and iteratively optimizing parameters of a glucose dynamics model via solving a nonlinear least squares problem based on the collected training

data while fixing a constant time and amount of a correction value; and determining the time and amount of a modified correction value based on executing a mixed integer quadratic program on the collected training data while fixing a constant value for the parameters of the glucose dynamics model;

provide the insulin infusion input and the modified carbohydrate size input to the model to determine a prediction of glucose outcome; and

send a command to an insulin pump to deliver a determined insulin dosage based on the predicted glucose outcome.



**FIG. 1**

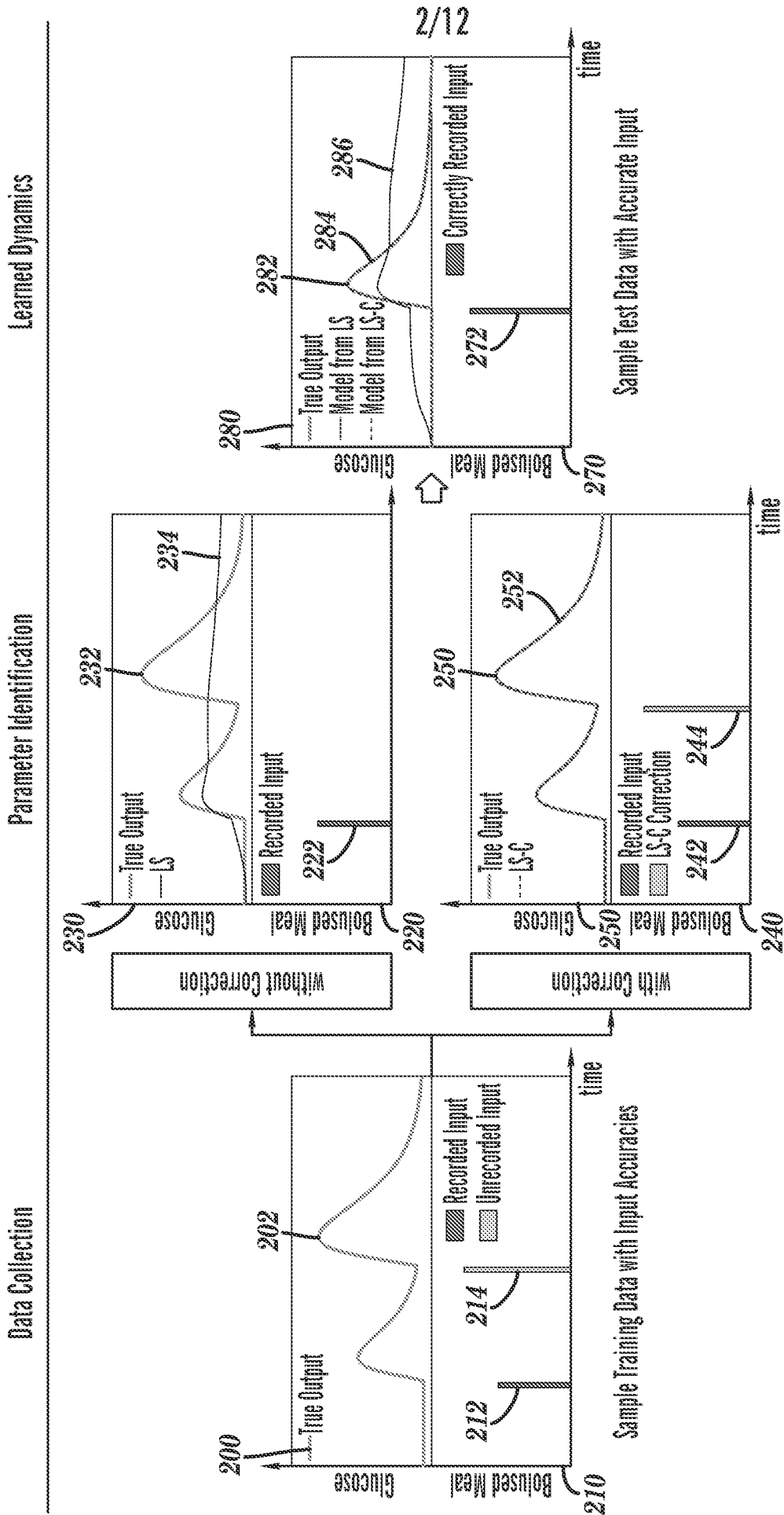


FIG. 2

3/12

---

**Algorithm 1:** Estimation of  $\gamma$

---

**Data:**  $y$ : recorded glucose trace  
 $\hat{y}$ : output of initial fit from the LS method

**Result:** Number of corrections  $\gamma$

**Initialize:**  $\gamma = 0, j = 1$

**while**  $j \leq N$  **do**

**if**  $ARD(y(j), \hat{y}(j)) > 20$  **then**

$\gamma \leftarrow \gamma + 1, j \leftarrow j + 48$    ▷ Advance 4 hours

**else**

$j \leftarrow j + 1$    ▷ Advance 5 minutes

**end**

**end**

**return**  $\gamma$

---

FIG. 3

---

**Algorithm 2:** Sequential Optimization for LS-C
 

---

**Data:** Recorded data  $\mathcal{D} = (\mathbf{y}, \mathbf{u}_i, \mathbf{u}_m)$

**Result:** Optimized SOGMM parameters  $\theta_p^*$

**Initialize:**  $\mathbf{u}_c = \mathbf{0}$ ,  $i = 0$ ,  $\theta_{p,0} = \bar{\theta}_p$

**while Not Convergence Criteria do**

$\mathbf{u}_m \leftarrow \mathbf{u}_m + \mathbf{u}_c$  and solve (9) to get  $\theta_{p,i}^*$

**if**  $i = 0$  **then**

        | Estimate  $\gamma$  from Algorithm 1

**end**

$\theta_p \leftarrow \theta_{p,i}^*$  and solve (11) to get  $\mathbf{u}_c$

$i \leftarrow i + 1$

**end**

**Return**  $\theta_p^* = \theta_p$

---

**FIG. 4**

500  
↓

	504 Insulin Bolus Input		506 Meal Input	
	Injected	Not Injected		
502 Correctly reported	●	✘	▬	
Misreported	●	✘	▬ ▬	
Unreported	N/A		▬	

**FIG. 5A**

FIG. 5B

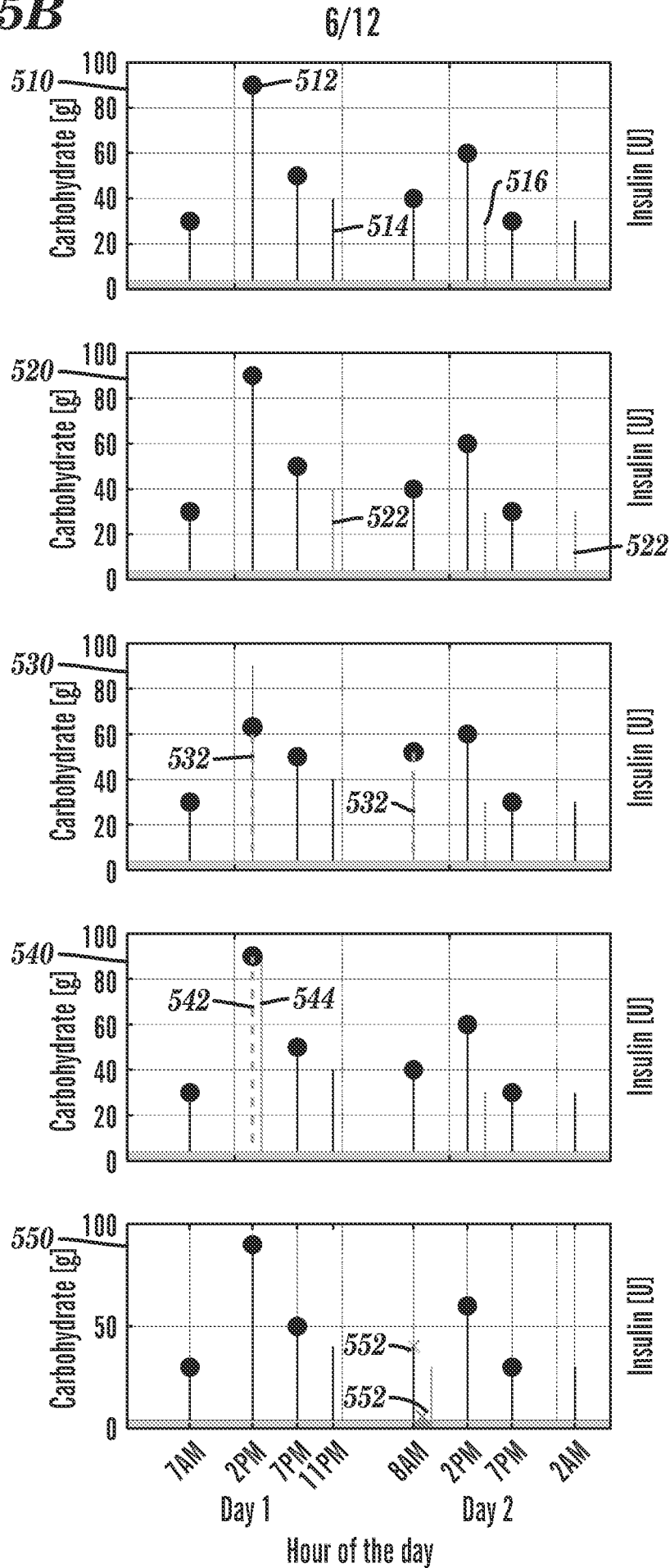


Table 2. RMSE (mg/dL) on test data

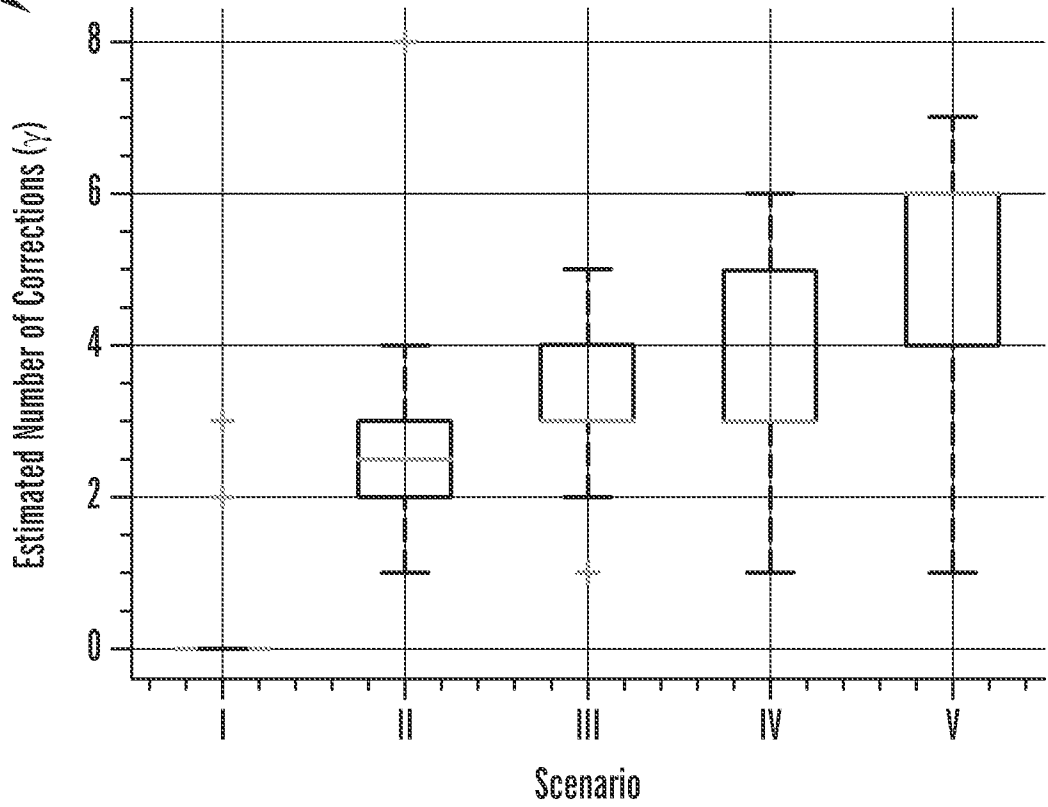
Model	Scenario				
	I	II	III	IV	V
Nominal	43±8	43±8	43±8	43±8	40±6
LS	12±3	20±7	17±3	17±2	29±9
LS-C	12±4	14±7	15±4	12±3	24±9
p-value*	0.746	0.004	0.009	0.0008	0.007

Values are reported as mean±standard deviation across 10 adult subjects.

FIG. 6

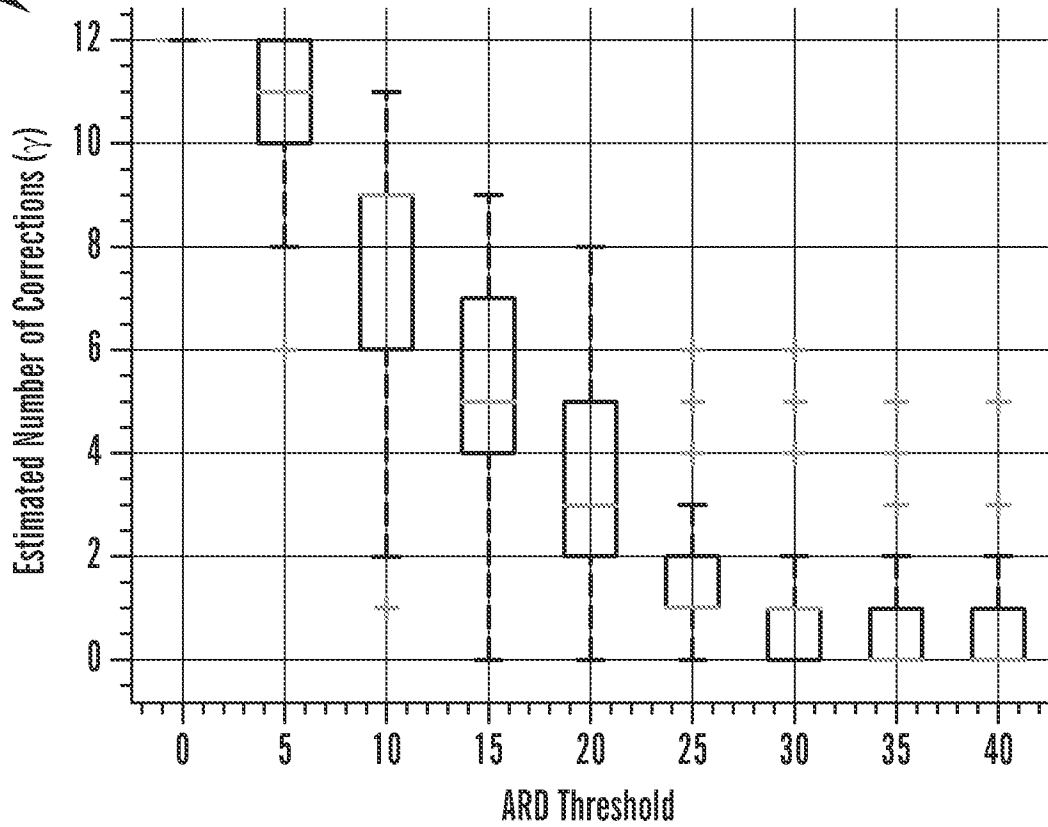
8/12

700



**FIG. 7A**

750



**FIG. 7B**

9/12

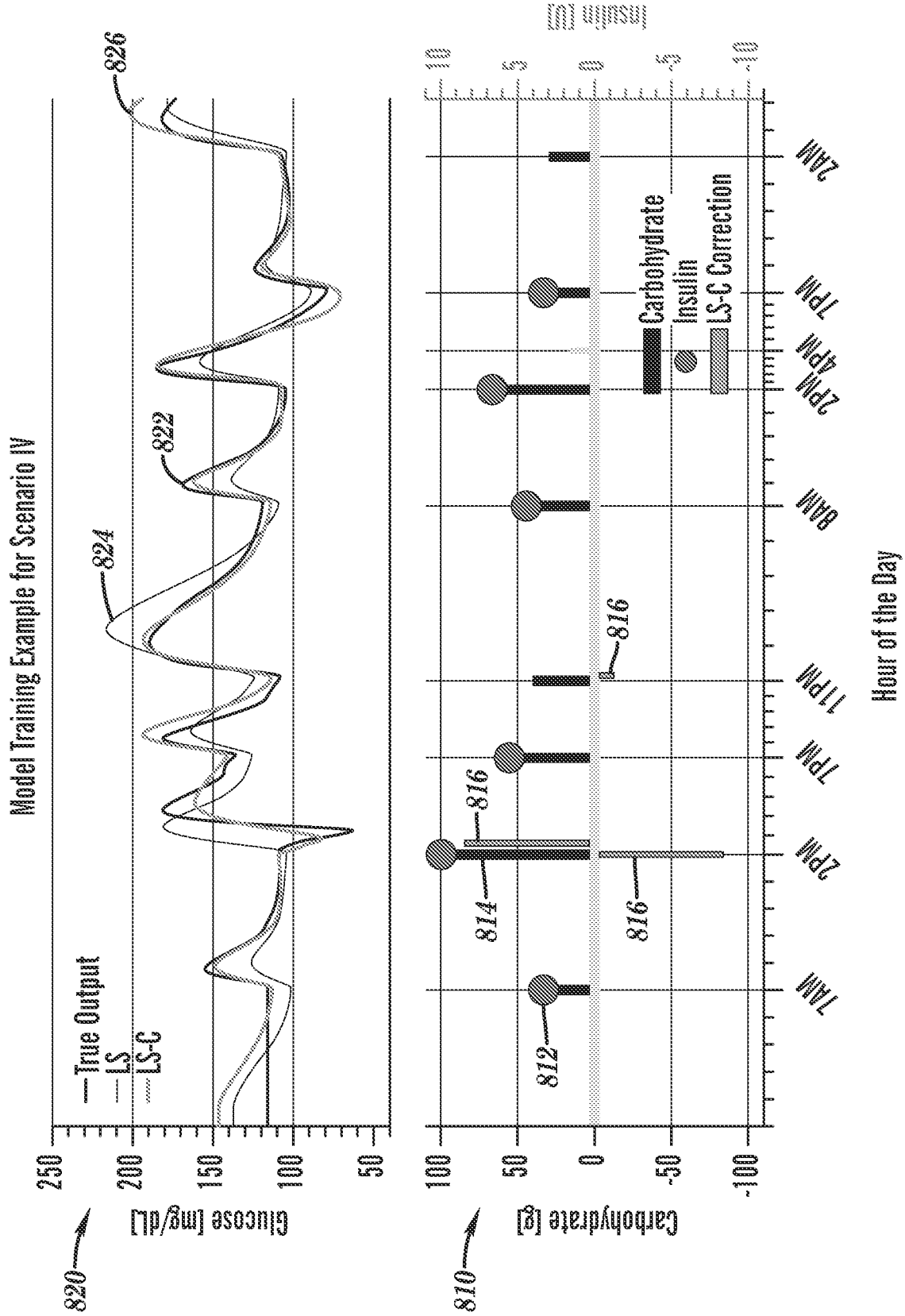
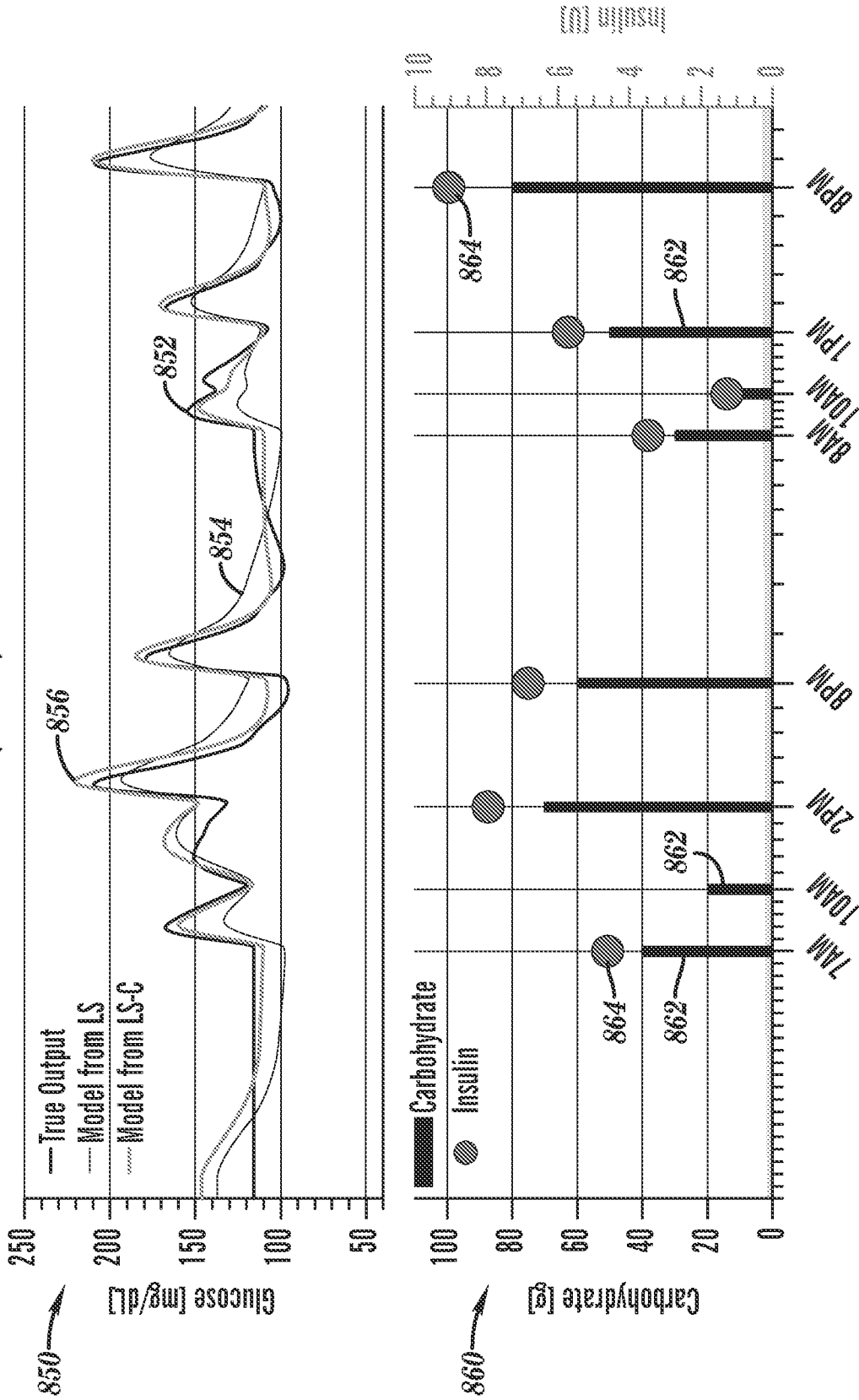


FIG. 8A

10/12

Test Output Example for Scenario IV



Hour of the Day

FIG. 8B

Participant PREG2MSSM02 on August 19th, 2020

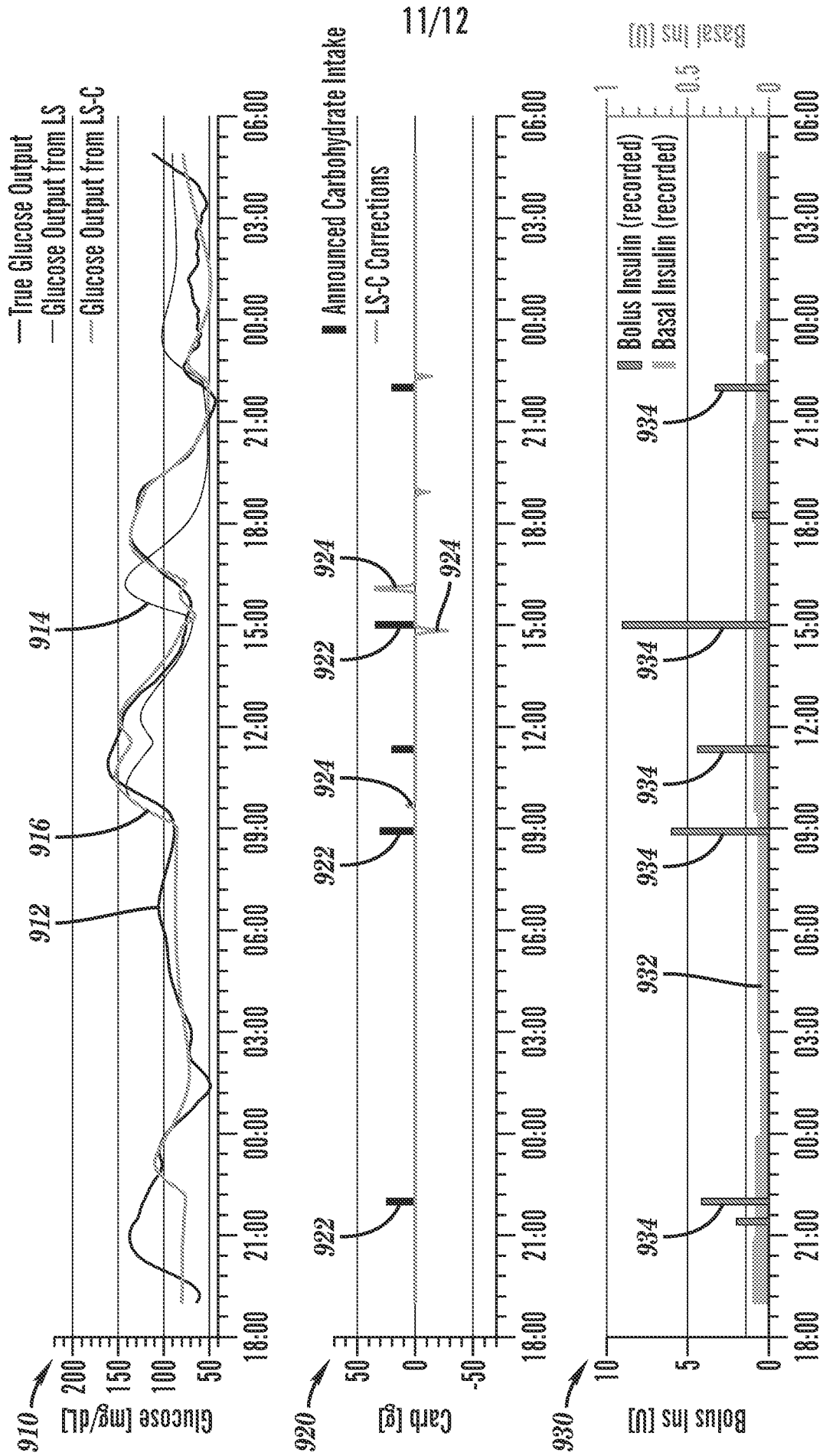
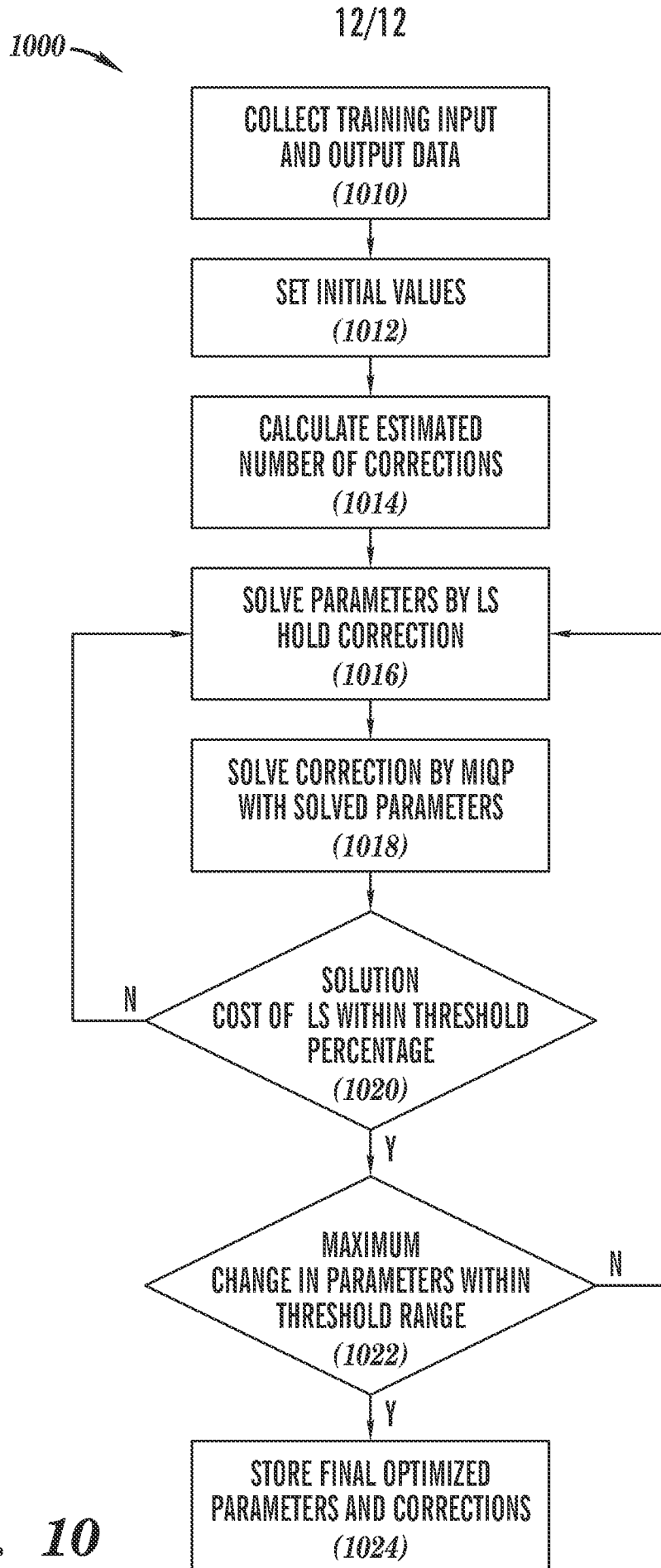


FIG. 9



**FIG. 10**