UVa/Padova

T1DMS

Type 1 Diabetes Metabolic Simulator

User Guide v3.2

July 31, 2020

THE EPSILON GROUP

Driven by Modeling

This page left blank intentionally.

Contents

List of Figures

This page left blank intentionally.

This page left blank intentionally.

1. Getting Started with the T1DMS Type 1 Diabetes Metabolic Simulator

1.1. Prerequisites

- Supported operating system: T1DMS supports Windows 10
- Matlab™ version up to 2020a (V9.8.0) (32-bit or 64-bit) with Simulink™ and Curve Fitting Toolbox.
- Optional: Install the Parallel Computing Toolbox and use matlabpool or parpool for parallel processing (64-bit).

1.2. Installation process

1. Follow the instructions in the 'T1DMS Installation Guide' for saving the model folder and activating the software.
2. After activation, a copy of the simulator model folder can be copied onto your hard drive. You may maintain more than one copy of the simulator model folder. Be sure to maintain access to the License Manager and Activation Code for future use.

1.3. Running the simulator

1. Start Matlab. Make sure that the simulator folder is selected as Matlab's "current folder."
2. Enter ***Simulator*** in the Matlab command line (note that commands are case sensitive). The screen below will appear.

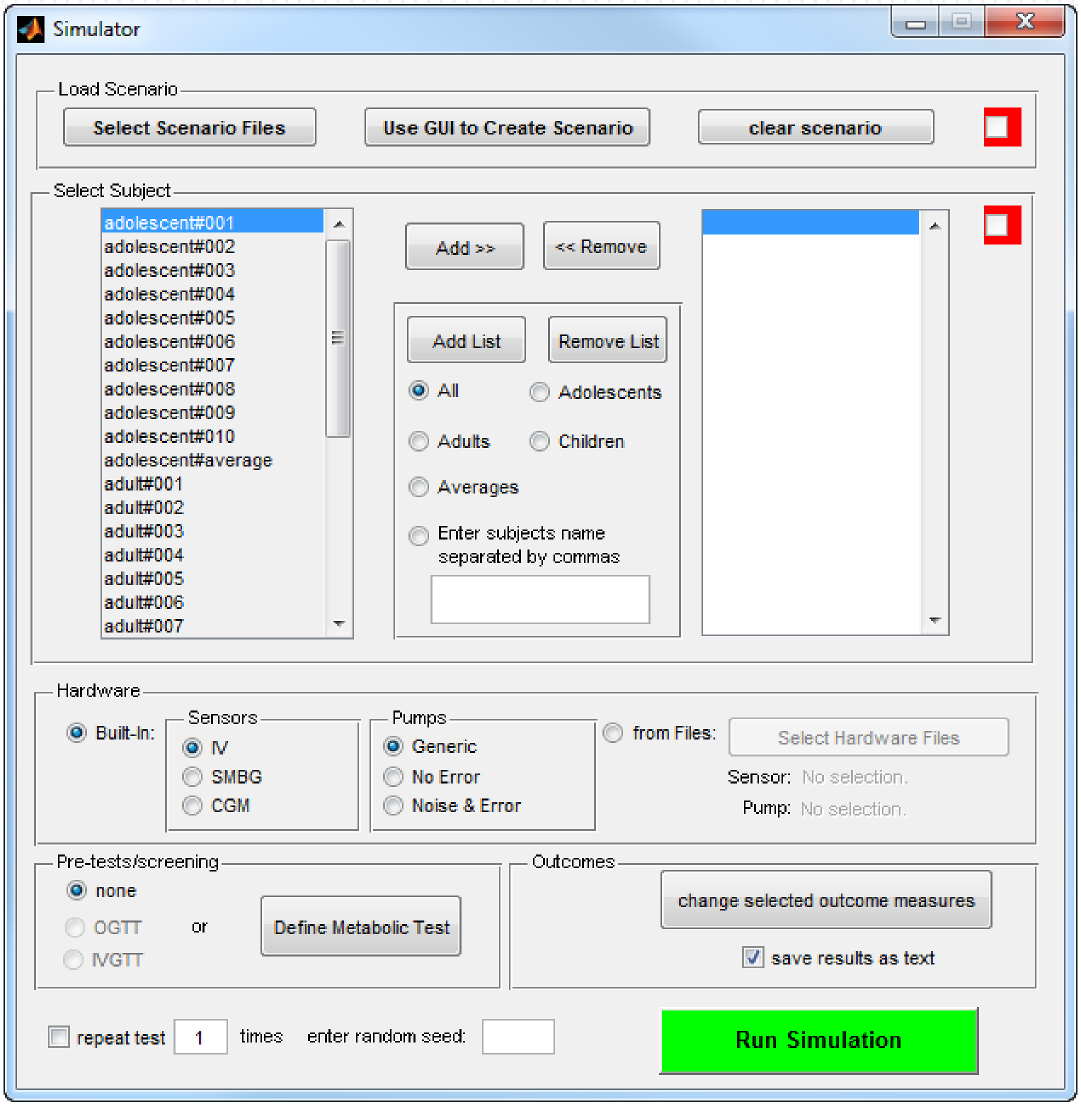

Figure 1: main GUI window

2. Creating a scenario

2.1. Using an ASCII scenario file

Simulation time, initial conditions, meals and open loop treatment parameters are stored in a scenario file in ASCII text format. For the T1DMS application to recognize and load the scenario file, it must have a .scn extension and be placed in the *scenario* folder within the simulator folder.

The simulator uses the specific ASCII file markers (listed below) to load the scenario data and transfer it to Matlab for processing. Click the Select Scenario Files button to choose from a list of available scenarios.

Use a percent sign (%) as the first character in a line to be recognized as input information.

Table 1: List of recognized ASCII file markers

ASCII File Marker	Definition
Tsimul	length of simulation
QTsimul	units of Tsimul (min, hour or day)
Tclosed	start of closed loop
QTclosed	units of Tclosed (see QTsimul)
Treg	start of regulation (in minutes after start of simulation)
simToD	the time of day at start of simulation (in minutes from midnight).
BGinit	allows the user to set the initial blood glucose (mg/dl) at any positive value. If not specified the simulation starts in the fasting state of each subject. It is advised to use realistic initial glucose values that are within the range of human glucose fluctuations.
Note	the initial glucose rate of change is always 0.
Tmeals	a vector of meal times (time begins at the start of the simulation; note minimum 60 minutes between meals; first meal at least 60 minutes after start of simulation).
Ameals	a vector of amount of meals in grams total or grams per Kg body weight
Dmeals	a vector of duration of meals, in minutes (if Dmeals is not specified, default value=15 min). Enter either a single value (all meals equal) or a vector equal in size to Ameals, specifying the duration of each meal. (note: maximum 30 minute meal duration)
QTmeals	units of Tmeals (min, hour, or day)
Qmeals	meal amount units; specifies Ameals as the total (total) amount (grams) or as grams/kg body weight (perkg)
basal	the basal insulin injection during open loop (fixed for all subjects, Units/hour. See Qbasal for subject-specific basal injections)
Qbasal	this tab allows the user to set the basal rate of insulin to a subject specific optimal value. If ‘Qbasal’ is set to quest, the open loop basal rate will be subject specific and corresponds to the basal rate required to maintain the fasting level of the subject (as defined in the ‘questionnaire’, see section 6.3.1). Any other value is ignored. ‘Qbasal’ supersedes any information contained in ‘basal’.
OB	optimal bolus, if set to ***on***, OB value is used to compute the meal bolus as a function of the meals amount (e.g. if subject’s carbohydrate ratio, CR=15 => OB=0.0667 =>1U per 15g of carbohydrates). OB is subject-specific and is part of the ‘questionnaire’ see section 6.3.1). Any other value is equivalent to ***off***. Note: If both OB and Abolus are set then a window will open asking the user to choose one.
Tbolus	times of insulin bolus injections during open loop
QTbolus	units of Tbolus (min, hour, or day)
Abolus	amount of bolus in Units of Insulin during open loop (fixed for all subjects used in the simulation, see OB for subject specific boluses)
TIVINS	a vector of begin and end times of continuous IV insulin injections in minutes past start of simulation

ASCII File Marker	Definition
QIVINS	the dosage of IV insulin injection per kg of weight (perkg) or total (total)
AIVINS	a vector of IV insulin injections matching TIVNS (U of insulin per minute)
TIVD	a vector of begin and end times of continuous IV dextrose injections
QIVD	the dosage of IV dextrose injection per kg of body weight (perkg) or total (total)
AIVD	a vector of IV dextrose injections matching TIVD (grams of CHO)

2.2. Example of a scenario file

The sample file shown below is for a scenario with no BGinit specified, 2 days open loop, and 2 days closed loop.

```
simulation info
%Tsimul=4
%QTsimul=day
%simToD=0
closed loop info
%Tclosed=48
%QTclosed=hour
%Treg=2880
meals info
%Tmeals=[7 12 18 22 31 36 42 46 55 60 66 70 79 84 90 94]
%QTmeals=hour
%Ameals=[45 70 80 20 45 70 80 20 45 70 80 20 45 70 80 20]
%Dmeals=[15]
%Qmeals=total
Open loop insulin info
%Qbasal=quest
%OB=on
```

Figure 2: Example Scenario File

2.2.1. Simulation time vs. start of closed loop vs. start of regulation

There are three important times that are defined as part of a scenario:

1. Simulation Time – identifies how long the simulation is to be run.
2. Start of Closed Loop – time at which the insulin injections are managed by the controller and not the open loop scenario. Any basal or boluses defined to occur after the start of closed loop will be ignored. If you wish to avoid closed loop control, just select a time greater than or equal to the simulation time.
3. Treg – the start of regulation (or end of commutation). The period of time between start of closed loop and start of regulation, the commutation period, represents the warm-up time of the controller. This can be excluded from the performance assessment if need be by setting the regulation time later than the start of closed loop. The later time controls the time period in which the outcome measures are computed.

2.3. Selecting a scenario file

The scenario files stored in the scenario folder are accessible from the simulator window by clicking the ***Select Scenario Files*** button. Locate the desired scenario in the available scenario pane, click to select it, then click the arrow button to move it to the selected scenario pane. Click the ***return to main window*** button to close the scenario selection window.

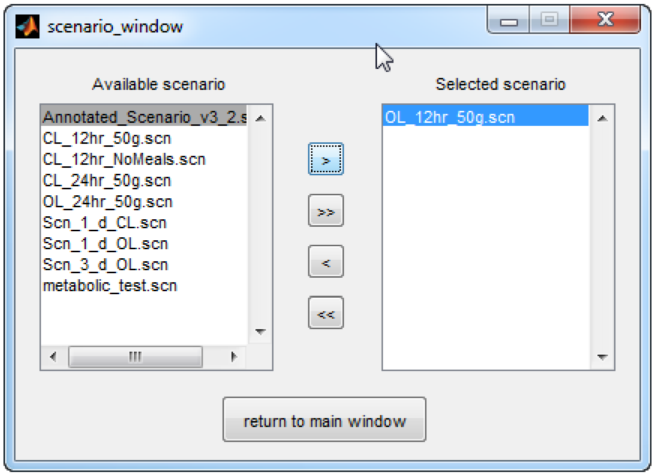

Figure 3: Scenario selection window

2.4. Using the GUI to create a scenario

You can simulate samples by creating a scenario using customizable preloaded scenarios available via the ***Use GUI to Create Scenario*** button:

- 1 meal
- 1 day with 3 meals and 2 snacks
- 1 week with 7 identical days

Note that once a customized scenario is activated (by clicking the ***Generate Scenario*** button) any previously defined scenario (loaded or preloaded) is deactivated. Once deactivated, all previous information, such as simulation length or mealtime, is erased and replaced by the new scenario.

To remove all scenario information loaded in the simulator, click the ***Clear Scenario*** button.

Once you have customized a scenario, you can save it as a .scn file by clicking the ***save as .scn file*** button. Saved scenarios are stored in the scenario folder and can be retrieved in the same manner as other available scenario files. (Save your scenario prior to pushing the Generate Scenario button)

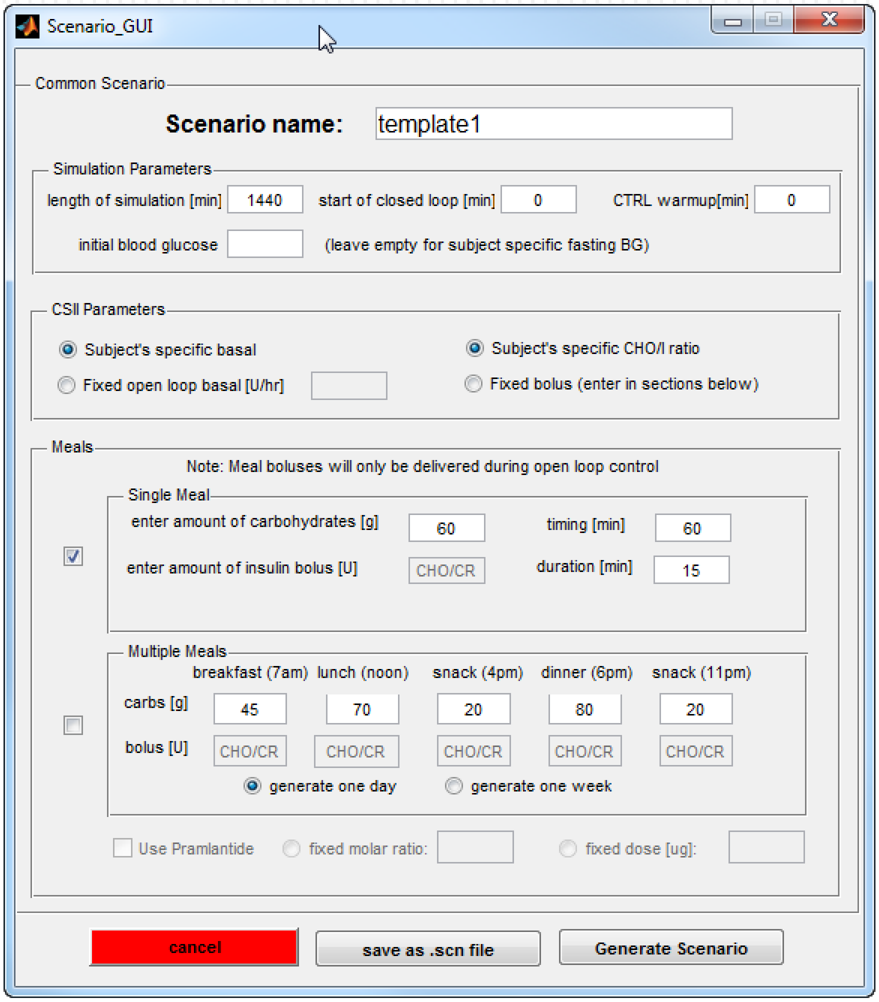

Figure 4: Creating a scenario using the GUI

2.4.1. Simulation Parameters

For all customizable preloaded scenarios, you need to enter the simulation parameters, which include: a simulation time, a start of closed loop control time, a control warm up time, and an initial blood glucose. These parameters are entered at the top of the window in the in the 'Simulation parameters' section.

2.4.2. CSII Parameters

The CSII parameters section allows you to configure continuous subcutaneous insulin infusion (CSII) dosing from an insulin pump. There are two alternatives for configuring the CSII parameters:

1. Subject specific parameters – when you select the radio buttons for ***Subject's specific basal*** and ***Subject's specific CHO/I ratio***, the basal rate will be set to the subject specific optimal rate and the meal bolus will be calculated using the optimal carbohydrate to insulin ratio
2. Fixed parameters – when you select the radio buttons for ***Fixed open loop basal [U/hr]*** and the ***Fixed bolus***, the values you specify in the for the basal rate and the bolus will be applies to all selected subjects during open loop simulation

Important Note: If you wish to simulate multiple meals, you must check the box next to the multiple meals section ***before*** setting the CSII parameters.

2.4.3. Single Meal

The 1 meal scenario has 4 parameters: meal carbohydrate amount in grams, meal timing in minutes, meal duration in minutes, and amount of associated bolus in insulin units. To activate the single meal scenario, check the box next to the single meal section.

Note: any bolus and basal parameters pertain to the open loop section of the scenario. If a bolus is programmed during the closed loop phase (timing of the bolus is greater than the starting time of the closed loop phase) then it is ignored.

2.4.4. Multiple Meals

The multiple meals section allows you to generate either a one day scenario or a one week scenario by selecting the desired radio button. Both options follow the same structure; the one week scenario repeats the one day scenario 7 times.

The multiple meals scenario has 10 parameters: carbohydrate grams for 5 potential meals (breakfast, lunch, dinner, 2 snacks); and the associated boluses in insulin Units. The timing of the meals is fixed and the length of simulation is automatically set to 24 hours. The 24 hour simulation length supersedes the length entered in the simulation parameters section (see above). As with the single meal scenario, boluses and basal rate are ignored during the closed loop phase.

3. Selecting subjects

The Select Subject section (shown in Figure 1: main GUI window) allows you to select subjects for the simulation individually or as a group. Click on a subject from the available list on the left panel and add it to the selected list on the right panel by clicking the ***add*** button. To remove a subject from the selected list, click the subject in the right panel and then click the ***remove*** button.

Use the ***Add List*** and ***Remove List*** buttons to add/remove multiple subjects within a group (children, adolescent, adults or all). Select the radio button that corresponds to the group you wish to add/remove and click either the ***Add List*** or ***Remove List*** button. You can also remove a whole category of subjects by selecting the category and clicking the 'Remove List' button.

To enter a list of specific subject names, type the names in the window separated by commas. Then select the ***Enter subjects name separated by commas*** radio button and click the ***Add List*** button.

4. Outcomes: selecting outcome measures and graphics

The T1DMS simulator provides multiple options for selecting the outcome measures (metrics calculated based on simulation results) and displaying the results in graphical format. It is important to note that graphs will generally include data for the entire simulation, but outcome measures will only be calculated for data during the regulation period. On blood glucose traces, the regulation period is marked by the change in line color from black to green.

Outcome measures and graphs can be selected by clicking on the ***change selected outcome measures*** button. The Outcome Measures window will appear with the available options.

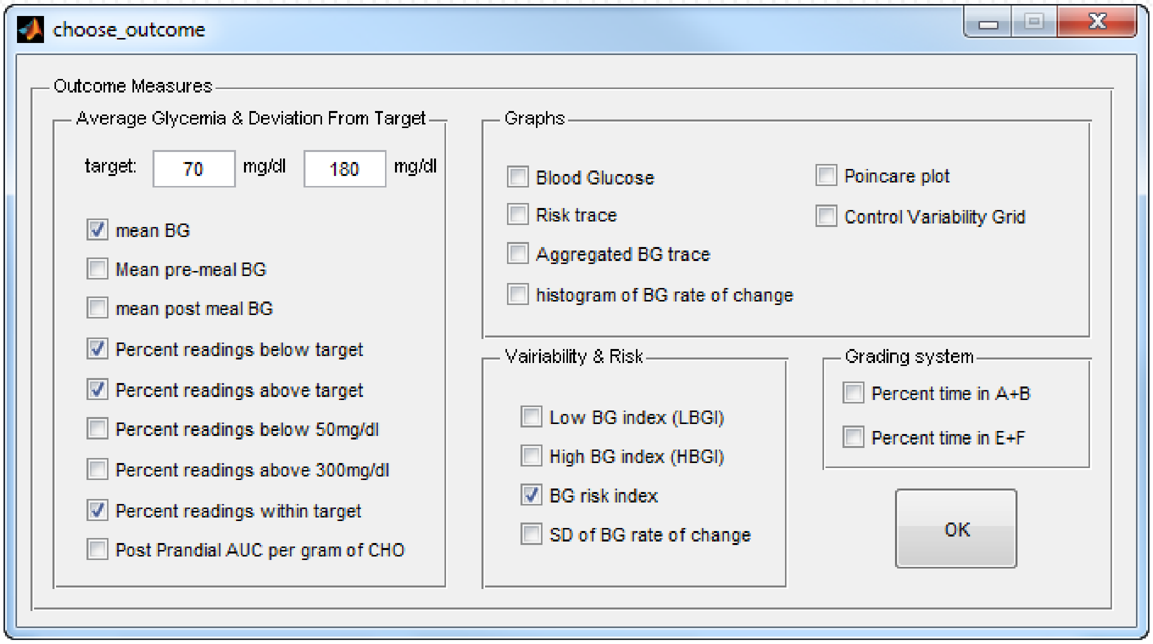

Figure 5: Choosing outcome measures and graphics

Select the desired outcome measures and graphs by checking the appropriate boxes. In order to activate your selections in the simulator, click the ***OK*** button. If you run the simulation prior to clicking ***OK***, the simulator will run based on the previously saved selections.

Details on the available outcome measures and graphs are show in Appendix B: Outcome Measures & Outcome Graphs.

4.1. Outcome measures

The default selections represent a standard set of the most commonly used outcome measures.

- Mean BG
- Percent readings below target
- Percent readings above target
- Percent readings within target

The default target blood glucose range is 70 to 180 mg/dl. You can adjust the lower and upper limit for the target range by clicking in the desired box and changing the values.

You can also select from the following outcome measures:

- Mean pre-meal BG
- Mean post-meal BG
- Percent readings below 50 mg/dl
- Percent readings above 300 mg/dl
- Post Prandial AUC per gram of CHO – the area under the curve (BG concentration) per gram of carbohydrate

4.2. Graphs

The simulator provides the following optional plots to display simulation results:

- Blood Glucose – for the BG graph, you must also select a temporal trace and/or density plot
- Risk Trace
- Aggregated BG trace
- Histogram of BG rate of change
- Poincare plot
- Control Variability Grid – for the CVGA graph, you can choose between per day (one point per day per subject per run) or per subject (one point per subject per run).

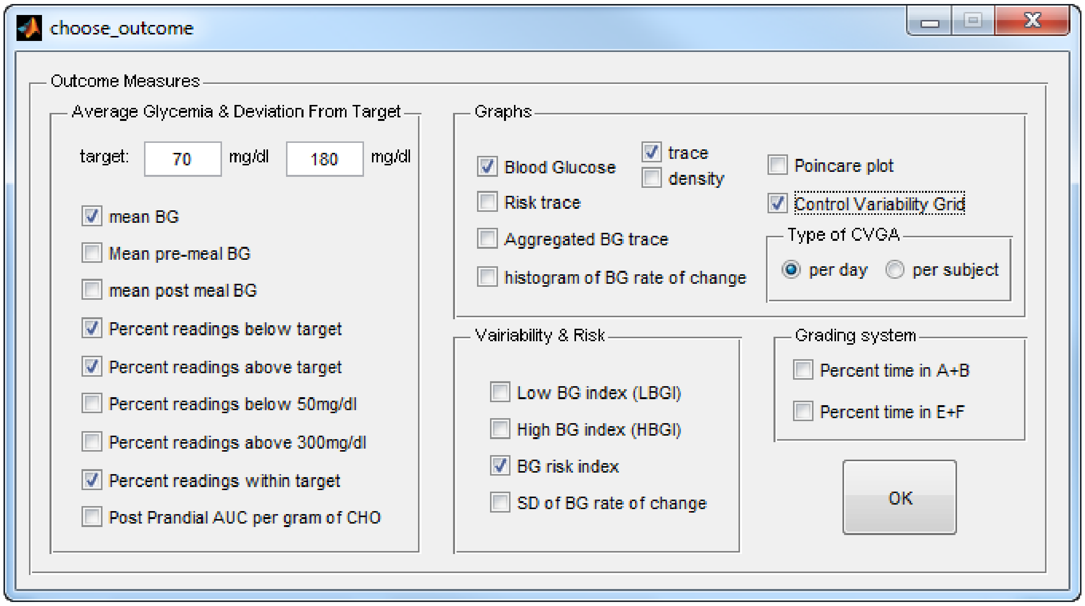

Figure 6: Blood Glucose and Control Variability Grid selections.

4.3. Variability & Risk and Grading system

A detailed explanation of Variability & Risk and Grading system is available in Reference 6 (see section 10).

4.4. Save outcome measures results as a text file

When the ***save results as text*** checkbox is checked, the outcome measures will be stored in a .txt file in the results folder. The default file name is “results_default.txt.” Each time a simulation is run with ***save results as text*** checked, the simulator will over-write any existing file of the same name without warning. To avoid losing the previous outcome measures of a simulation, enter a unique filename in the ‘enter_filename’ window (shown in Figure 10) before saving.

5. Hardware

The hardware section allows you to select from several built in device profiles or to select from any of the hardware files stored in the hardware folder. To use built in devices, click the ***Built-in*** radio button and then click on the desired glucose sensor and pump.

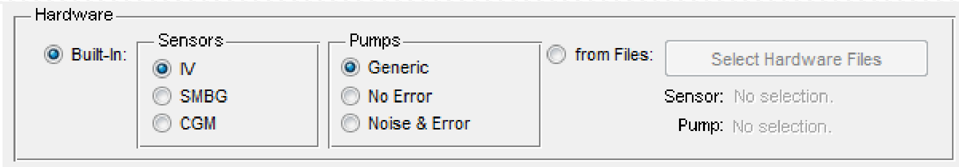

Figure 7: Hardware options on the main GUI

To use hardware described in a hardware file, click on the ***from Files:*** radio button then click the Select Hardware Files button to display the list of available sensors and pumps.

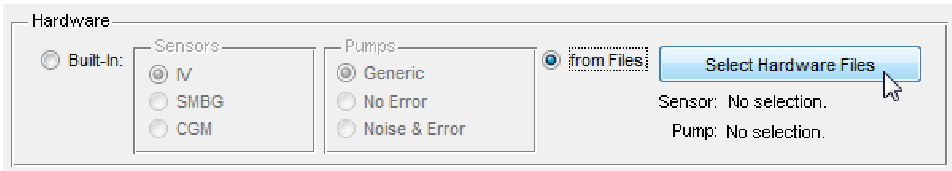

Figure 8: Selecting hardware from a file

Then, click on the desired sensor and the desired pump to select them and click the ***return to main window*** button.

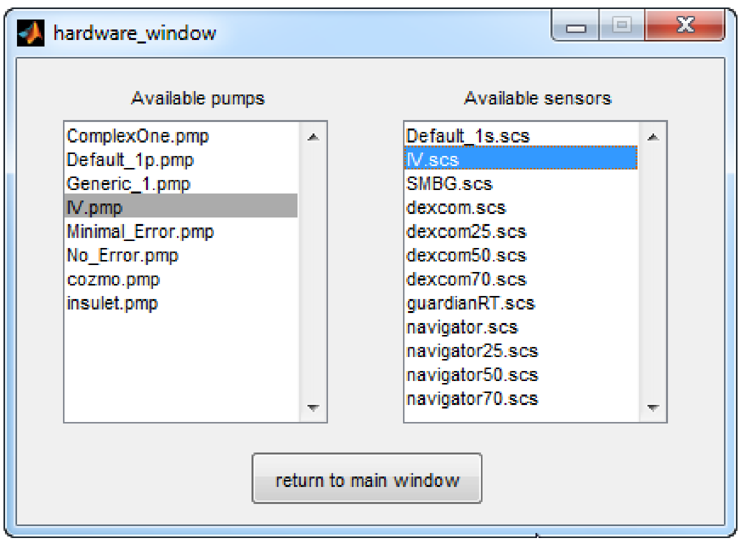

Figure 9: Selecting the pump and sensor files

6. Pre-tests/screening

See Section 9.3.2 Metabolic testing for a description of the use of Pre-tests/screening.

7. Repeating experiments

The repeat option on the left of the ***Run Simulation*** button will repeat the simulation on each subject a specified number of times. Click the ***repeat test*** checkbox and enter the desired number of repetitions.

You can also repeat an exact set of experiments by setting the random seed to a specific number. This will fix the sequence of numbers used to simulate sensor noise allowing you to repeat a simulator run exactly.

Note that the number of subjects, order of the subjects, the number of repeats, and the simulation length are all factors in calculating the simulated sensor noise. As a result, these parameters need to be identical between runs in order to replicate the numeric sequence that produces the sensor noise. For example, if you ran a simulation on a group of subjects and then ran the same simulation on one specific subject from that group, the results will not be the same for that subject. In order to achieve the same results by repeating the test and fixing the random seed, you would need to run the same group of subjects in the same order and for the same length of time.

8. Running a simulation

Clicking on the ***Run Simulation*** button initiates calculations for all selected subjects based on the selected scenario. Once the simulation completes, the simulator generates the outcome measures and graphs.

If the ***save results as text*** check box is checked, a prompt will appear in which you may name the outcome measures text file. This file contains all selected outcome measures (columns) for each subject and run (rows).

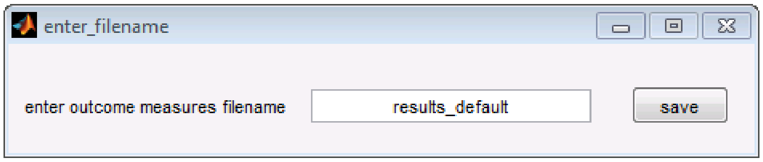

Figure 10: Outcome measures file name prompt

The sim_results.mat file is a structured data file containing the raw data from the simulation. Each time a simulation is run, the simulator will over-write any existing file of the same name.

9. Including a controller in the simulator

In the simulator, a controller is a combination of two main components: a control setup function file (ctrlsetup.m), which fine tunes the chosen control law to each particular subject (section 9.3), and a run-time algorithm, implemented as a Simulink block, which applies the tuned control law (section 9.3).

9.1. Implementing your controller in a Simulink™ block

The control law must be implemented to run from a Simulink™ block, which provides a large choice of possible implementations to each user. The input and output of the control law block are fixed and described below.

9.1.1. Inputs

1. Glucose readings: the selected **sensor** readings in mg/dl

   Note: the readings can be subcutaneous, IV or SMBG, depending upon the sensor selected during the hardware selection. The sensor parameter***, SensorType***, designates the type of sensor.

2. Insulin injection: the subcutaneous insulin injection in pmol.min^{-1} (6000 pmol = 1 U)

3. Meal announcement: a vector containing the amount of an incoming meal and the number of minutes remaining before and after the start of a meal. Both values are 0 if the incoming meal is more than 60 minutes ahead or more than 60 minutes past. As a result, there is no meal announcement more than 60 minutes prior to or past mealtime.
4. ToD: time of day, in minutes past midnight
5. Glucagon injection

9.1.2. Outputs

1. Basal pump rate: the basal rate in pmol.min^{-1} requested of the pump.
2. Bolus pump rate: the bolus rate in pmol.min^{-1} requested of the pump. A bolus can be implemented in more than one minute (e.g. up to 15 for the Cozmo pump).
3. IV insulin injection: the rate of IV insulin injection in pmol.min^{-1}.
4. IV dextrose injection: the rate of IV dextrose injection in mg/kg.min.
5. Oral carbs: grams.min^{-1} of CHO given by the controller (such as glucose tablets).
6. SQ glucagon

Note: the pump will handle the bolus and basal rates differently both in terms of min/max and in terms of increments.

Figure 11: The control law Simulink block

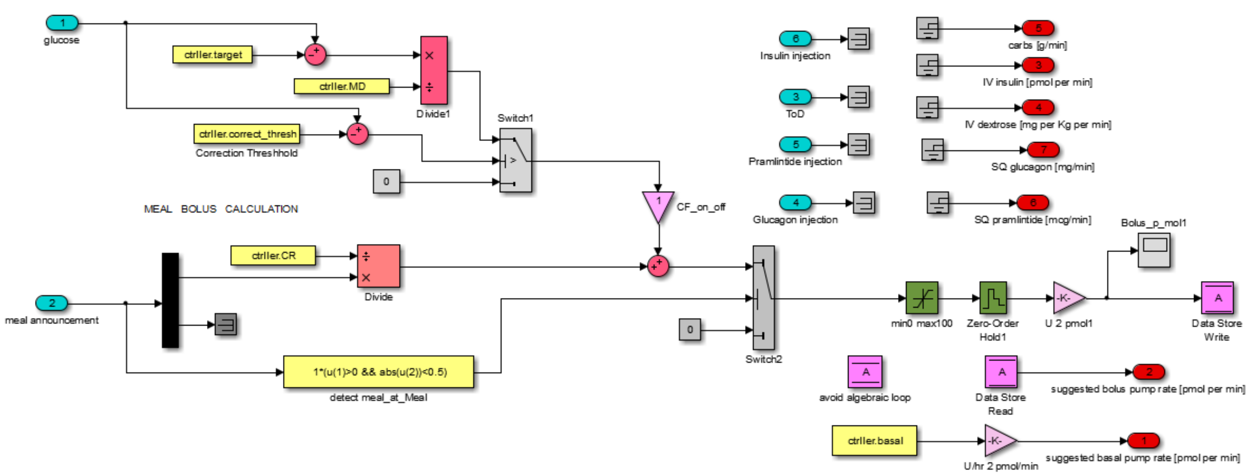

Figure 12: An example of control law implementation

For FDA validation, any Simulink block, m-file, or S-function can be used in the control block as long as it **<u>does not refer</u>** to the structure variables **struttura** or **scenario**, or any variable derived from these two structures.

9.2. Setting up your controller for a simulation (choice of parameters)

It is likely that certain controllers will need tuning to each specific subject, either based on their standard insulin regimen, age, weight, or some characteristics extracted from a metabolic test (e.g. insulin sensitivity from an IVGTT or HOMA). For that purpose, the simulator allows for tuning of the controller before each run.

The tuning procedure must be encapsulated in the ctrlsetup.m file, located in the ‘controller setup’ folder. The ctrlsetup.m file must be a Matlab function file, which allows for the definition of subject specific variables. These variables will then be available to the control law Simulink block during simulation (see green blocks in the ‘standard treatment’ control law provided with the simulator, Figure 12).

<u>Note:</u> For FDA validation the control setup function <u>must not</u> use any information included in the **struttura** and **scenario** structure variables.

9.3. Including your controller block in the simulator

The folder where the simulator is installed contains a file named **testing_platform.mdl**. Open this file in Matlab and the following Simulink model window will open:

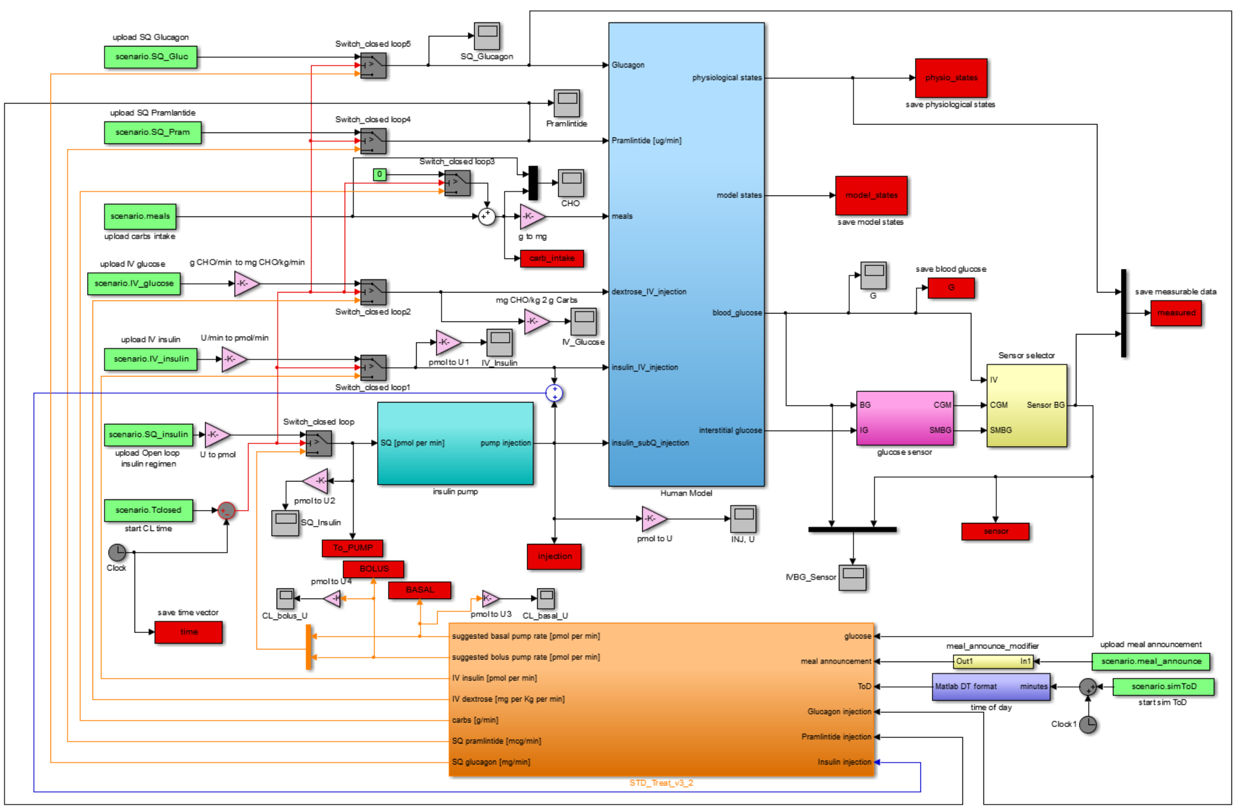

Figure 13: Testing platform

The orange block at the bottom contains the controller.

Replace the standard treatment block with your own controller as constructed in Section 9.1, and save the Simulink model as testing_platform.mdl.

The final step is to copy your ctrlsetup.m function file and any other needed script, function, m-files, mat-files and/or sub-functions into the 'controller setup folder'.

It is important to note that any scripts or sub-functions that are added in a folder other than the 'controller setup' folder (for example if you created a folder in 'controller setup' and then put the files in that folder) **will not be visible to the simulator**, and therefore will not be used.

Note: it is often necessary to restart Matlab after changing a controller.

```
function ctrller=ctrlsetup(Quest,hd,sc,ctrller,struttura);
ctrller.ctrlname='Testing_Std_Control'; % user-specified
path_root=[cd '\controller setup'];
addpath(path_root)
%% Controller Dosing Algorithm Robustness Analysis Modifiers
% % user can use these parameters explore & evaluate response to stress
% NOT included in FDA master file
% SImult will apply for the entire Tsimul
ctrller.SImult=1; % modify SI, insulin sensitivity
% Mealmult will apply for the entire Tsimul
ctrller.Mealmult=1; % modify meal speed (<1=slower)
% test with mis-announcement of the meals amount in the controller
ctrller.meal_announce_modifier=[0 1]; % multiplicative time dependent modifier
% modifier on meal_announce amount, set = 0 for no meal announcement
% first column contains time from which to apply the values in the second column
% ADD additive and/or multiplicative Bias to the CGM
ctrller.CGM_bias=[0 0]; %additive time dependent modifier on CGM signal,
% first column contains time from which to apply the values in the second column
ctrller.CGM_bias_rel=[0 1]; %multiplicative time dependent modifier on CGM signal,
% first column contains time from which to apply the values in the second column
%% write your ctrlsetup code below
ctrller.target=100;
ctrller.thresh=150;
ctrller.meal_time_diff=0; %plus/minus 60 max - see meal announce
ctrller.basal=Quest.basal;
ctrller.MD=Quest.MD;
ctrller.CR=Quest.OB;
```

Figure 14: Example ctrlsetup.m file

9.3.1. Available information on each subject: the questionnaire

When running the ctrlsetup.m function, the user has access to some characteristics of the patient. These characteristics are made readily available to mimic the use of a screening visit during a clinical protocol. They are therefore limited and included in the structure variable Quest. The Quest variable has the following fields:

1. name (from which you can extract the age category: e.g. adolescent#005)
2. weight, in kg
3. fastingBG, the subject's fasting blood glucose under basal insulin injection, in mg/dl.

4. basal, the subject insulin basal rate that keeps the patient at its fasting glucose level, in U/hr.

   DISCLAIMER: The basal rates requested are not necessarily implementable by the chosen pump (e.g. 0.82U/hr will be implemented as 0.8U/hr if using an insulet Omnipod™ which has a 0.05U/hr increment). This can have significant consequences on plasma glucose, especially for very sensitive patients such as children.
5. OB, the optimal bolus in Units of insulin per gram of carbs. This is calculated by optimizing a 24-hour typical day with 3 each of 60g of CHO meals and an 11pm 10g snack. Each meal is associated with a bolus computed as *g of CHO/OB*. The optimization criterion used is the min/max range of glucose in risk space.
6. MD, the maximum drop in mg/dl per Unit of insulin. This is computed as the maximum decrease in glucose concentration (over 6 hours) after a 1 U insulin bolus.
7. TDI, the average total daily insulin regimen, in Units of insulin, per day, using the subject specific basal rate, with boluses calculated using optimal boluses, maximum drop and a target of 120 (maximum drop is used as a correction factor surrogate).

9.3.2. Metabolic testing

One of the most advanced tuning procedures allowed by the simulator is the possibility to simulate a metabolic test before the actual run. Doing so is fairly straightforward but requires great care in variable handling and definition.

First it should be noted that the metabolic test can be any scenario involving ingestion of food, IV dextrose injection, IV insulin injection, or subcutaneous injection of insulin. Length of simulation, timing and number of glucose or insulin challenges is left to the user’s imagination.

At this time the simulator only supports “open loop” metabolic testing, i.e. the insulin/dextrose/meal treatments are decided in advance of the test, not adapted during the test. Therefore clamps types of tests are not available.

To run a metabolic test, you first need to define a set of measurements available in the test (e.g. plasma glucose, or plasma insulin) by clicking the ***define metabolic test*** button at the bottom of the main window. This will activate the following window:

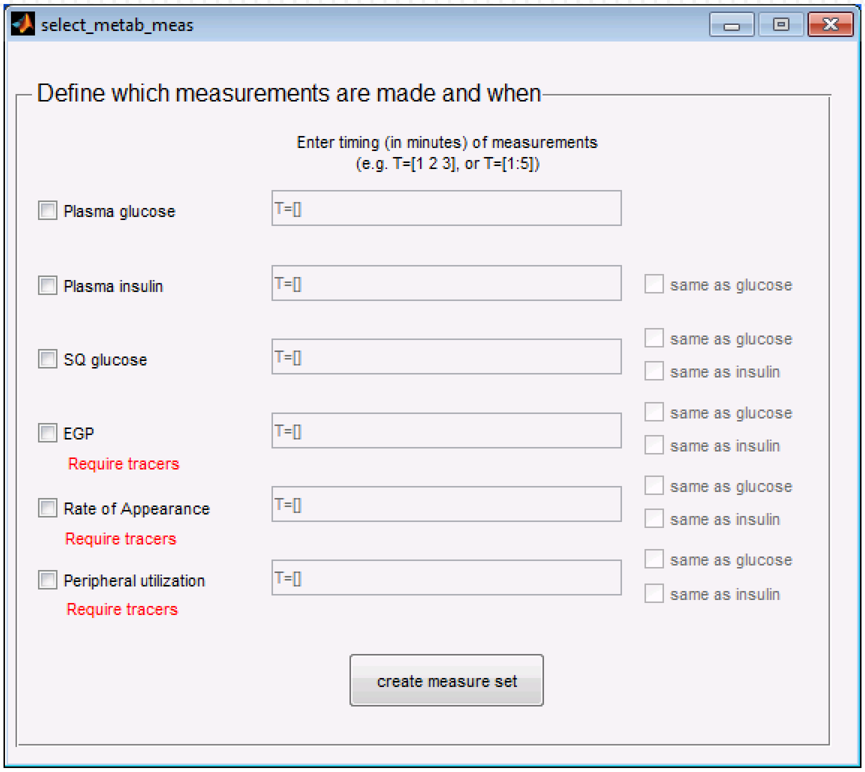

Figure 15: Select metabolic measures

Once a measurement is selected, the timing field becomes editable (unless "same as glucose", or "same as insulin" is selected). The user can enter the timing of measurement in minutes as if defining the vector T in Matlab, e.g. T=[1 2 3 4 5] same as T=[1:5], or T=[0 1 2 3 4 5 10 15 20 25 30 60 90 120 150 180] same as T=[0:5 10:5:30 60:30:180].

By clicking on create measure set you exit the window and save your selection.

To run the actual metabolic test, add the following line in the ctrlsetup.m function:

[dataMetab]=run_metabolic_test(filename,struttura,scenario,hd,Quest)

The results from the test are contained in the structure array dataMetab with one array element per measurement type, and each element with fields time and values; The measurements are always in the following order in dataMetab: Plasma glucose before plasma insulin, before subcutaneous glucose, before endogenous glucose production (EGP), before glucose rate of appearance (RA), before glucose peripheral utilization. (Example files are provided in the controller setup folder and the scenario folder)

Note: for any FDA validation studies, the only metabolic tests allowed are tests that are part of the clinical protocol submitted for an IDE, blood draws included, and that EGP, RA and utilization are ONLY available with the use of tracers.

10. References and Acknowledgements

The T1DMS Type 1 Diabetes Mellitus Metabolic Simulator referenced herein is the 'Distributed Version' of the ***UVa/Padova Type 1 Diabetes Mellitus Metabolic Simulator***. The following references cover some of the scientific literature describing the development of the T1DMS.

10.1 References

1. Chiara Dalla Man, Robert A. Rizza, and Claudio Cobelli Meal Simulation Model of the Glucose-Insulin System IEEE Transactions of Biomedical Engineering, 2007 54(10): 1740-1749
2. Dalla Man C, Camilleri M, Cobelli C. A system model of oral glucose absorption: validation on gold standard data. *IEEE Trans Biomed Eng*. 2006 53(12): 2472-2478.
3. Marc Breton, Ph.D. and Boris Kovatchev, Ph.D. Analysis, Modeling, and Simulation of the Accuracy of Continuous Glucose Sensors J Diabetes Sci Technol 2008; 2(5): 853-862
4. Boris P. Kovatchev, Ph.D., Marc D. Breton, Ph.D., Chiara Dalla Man, Ph.D., and Claudio Cobelli, Ph.D. In Silico Preclinical Trials: A Proof of Concept in Closed-Loop Control of Type 1 Diabetes J. Diabetes Sci Technol 2009 3(1): 44-45
5. Stephen D. Patek, PhD., Wayne Bequette, PhD., Marc Breton, PhD., Bruce A. Buckingham, M.D., Eyal Dassau, PhD., Francis J. Doyle III, PhD., John Lum, Lalo Magni, PhD., and Howard Zisser, M.D. In Silico Preclinical Trials: Methodology and Engineering Guide to Closed-Loop Control in Type I Diabetes Mellitus J. Diabetes Sci Technol 2009 3(2): 269-282.
6. William Clarke, M.D. and Boris P. Kovatchev, Ph.D. Statistical Tools to Analyze Continuous Glucose Monitor Data Diabetes Technol Ther. 2009 11(S1): S45-S54
7. Lalo Magni, Ph.D., Davide M. Raimondo, M.S., Chiara Dalla Man, Ph.D., Marc Breton, Ph.D., Steven Patek, Ph.D., Guissepe De Nicolao, Ph.D., Claudio Cobelli, Ph.D., and Boris P. Kovatchev, Ph.D. Evaluating the Efficacy of Closed-Loop Glucose Regulation via Control-Variability Grid Analysis J. Diabetes Sci Technol 2008 2(4): 630-635.
8. Chiara Dalla Man, Ph.D., Francesco Micheletto, Ph.D., Dayu Lv, Ph.D., Marc Breton, Ph.D., Boris Kovatchev, Ph.D., and Claudio Cobelli, Ph.D. The UVA/PADOVA Type 1 Diabetes Simulator: New Features J. Diabetes Sci Technol 2014 8(1): 26-34.
9. Dayu Lv, Ph.D., Marc D. Breton, Ph.D., and Leon S. Farhy, Ph.D. Pharmacokinetics Modeling of Exogenous Glucagon in Type 1 Diabetes Mellitus Patients Diabetes Technol Ther. 2013 15(11): 935-941.

10.2 Acknowledgements

This User Guide was written by Marc D. Breton and revised by The Epsilon Group.

Appendix A: Sample Scenario.scn and ctrlsetup.m Files

A.1 Annotated Scenario.scn

```
An ANNOTATED Scenario File
Note: "*" Indicates default values
Note: "%" Required for variables to be read, NOT => comment
File Name will be available in sim_results.mat data.scenario file

TIME_of_Simulation_Information
Tsimul     Model Run Time - in QTsimul units
%Tsimul=1440
QTsimul    Tsimul Units - min*, hour, day
%QTsimul=min
simToD     Time of Day at start of sim - 0* in minutes after midnight
%simToD=0

Initial Blood Glucose Value in mg/dl
BGinit not specified* =>subject specific fasting or Specify in mg/dl
Such as this example: %BGinit=[100]

MEALS INFO _ Timing, Amount Carbs in total or perkg, Duration of meals
Tmeals     Time of Meals VECTOR in time after start of simulation
%Tmeals=[[8 12 19 23]
QTmeals    Tmeals unit - min*, hour, day
%QTmeals=hour
Meal_Amounts VECTOR matching_Tmeals_CHO in Total grams or g/KG BW
%Ameals=[40 60 80 10]
DURATION of meals_Fixed* minutes(15*) or Dmeals VECTOR to match Tmeals
%Dmeals=[15]
Qmeals is Ameals Unit_total g cho or g/Kg body weight =total* or perkg
%Qmeals=total

START CLOSED LOOP TREATMENT
Time after start of sim in QTclosed units - min*, hour, day
%Tclosed=0
%QTclosed=hour

START REGULATION _Time past start of the simulation in minutes
REGULATION => data is included in analysis calcs of results data
Built-in Data Analysis
%Treg=0

OPEN_LOOP Basal injections
basal Fixed _ set "basal=[XX]" - Units of Insulin per minute
Qbasal uses subject-specific basal rate (quest overrides basal=)
%Qbasal=quest


OPEN_LOOP SQ insulin BOLUS Treatment
OB on uses subject-specific Unit Insulin/g CHO to compute meal bolus
```

```
%OB=on
User specified Boluses - OB on set overrides any Abolus info
Abolus     Vector of bolus amounts in Units Insulin (total or perkg)
Tbolus     Time_Vector of bolus injections in QTbolus units
QTbolus    Tbolus Units – min*, hour, day
Qbolus     Units of Bolus:'total' Units Insulin or 'perkg' U/Kg B.W.
```

OPEN LOOP INSULIN INJECTION INFO – IV

```
TIVINS     Time_Vector - IV insulin injection begin & end times in
           minutes after start of simulation
QIVins     Dosage Units - IV insulin: 'perkg' b.w. or total 'total'
AIVINS     a vector of IV insulin injection rates (U insulin/minute)
```

OPEN LOOP DEXTROSE INJECTION INFO – IV

```
TIVD       Time_Vector of IV dextrose injections begin & end times in
           minutes after start of simulation
QIVD       units of IVdextrose injections: per kg of of body weight
           ('perkg') or ('total')
AIVD       a vector of IV dextrose injection rates (grams of CHO/min)
```

____

This Annotated Scenario and several sample scenarios are included in the T1DMS Type 1 Diabetes Simulator model folder

____

A.2 Annotated ctrlsetup.m

```
function ctrller=ctrlsetup(Quest,hd,sc,ctrller,struttura)

% OPTIONAL add a name for auditing; can be checked in sim_results
% load sim_results.mat and see data.ctrlsetup
% NOTE actual required filename is always ctrlsetup.m
ctrller.ctrlname='ctrlsetup_Annotated_v3_2';  % user-specified

%% Controller Dosing Algorithm Robustness Analysis Modifiers - Required
% % user can use these parameters explore & evaluate response to stress
% NOT included in FDA master file
% SImult will apply for the entire Tsimul
ctrller.SImult=1; % modify SI, insulin sensitivity

% Mealmult will apply for the entire Tsimul
ctrller.Mealmult=1; % modify meal speed (<1=slower) (>1=faster)

% test with mis-announcement of the meals amount in the controller
ctrller.meal_announce_modifier=[0 1]; % multiplicative time dependent modifier
% modifier on meal_announce amount, set = 0 for no meal announcement
% first column contains time (minutes) from which to apply the values in the second
column
% modified meal amount will be used in meal bolus calculation as applicable

% ADD additive and/or multiplicative Bias to the CGM sensor
ctrller.CGM_bias=[0 0]; % additive time dependent modifier on CGM signal,
% first column contains time from which to apply the values in the second column
ctrller.CGM_bias_rel=[0 1]; % multiplicative time dependent modifier on CGM signal,
% first column contains time from which to apply the values in the second column

%% write your code below
% Subject-Specific: Quest information available to the controller
%Optimal Basal, Optimal Carb Ratio, Max.BG.Drop/Unit Insulin
ctrller.basal=Quest.basal; % U/hr
ctrller.CR=Quest.OB; % Carb Ratio, gCHO/U insulin
ctrller.CF=Quest.MD;   % Correction Factor max BG drop/U insulin

ctrller.name=Quest.names; % subject name- determine if adult, teen or child
ctrller.BW=Quest.weight; % subject weight, kg
ctrller.fastingBG=Quest.fastingBG; % subject fasting BG mg/dL w/basal

%% Some potential controller specific parameters
% Subject independent values can be spec'd directly in Controller

% Target & Threshhold values generally used w/correction bolus calculation
ctrller.corr.tgt=100; % set correction bolus target mg/dL
ctrller.corr_thresh=150; % set correction bolus threshhold mg/dL
ctrller.corr_CFmeal=0; % Correction Bolus w/Meal; =0 => off, =1 => 100%
```

An Annotated ctrlsetup.m file is included in the
T1DMS Type 1 Diabetes Simulator model folder (controller setup folder)

Troubleshooting: Use scopes in your controller. Open the testing_platform.mdl and your controller. Open several scopes prior to running your simulation. Watch your simulation in progress. Check and verify the controller action. (note: scopes do not function during parallel processing)

____

Appendix B: Outcome Measures & Outcome Graphs

B.1 Outcome Measures

The T1DM simulator can create the "per subject" Glucose Control-relevant analyses as selected by the user and calculated at the end of the simulation (see create_output.m for calculation methods). All of the results are evaluated between the start of regulation (Treg) and the end of the simulation.

Mean blood glucose reading:
Per-subject average of glucose readings (mg/dl).

Mean pre-meal blood glucose values:
Average per-subject glucose values (mg/dl) during the hour prior to meal times.

Mean post-meal blood glucose values:
Average per-subject glucose values (mg/dl) over one hour beginning 60 minutes after a given meal.

Per cent time in Severe Hypoglycemia (BG ≤ 50 mg/dL):
Percentage of time (per-subject) spent with a glucose level less than or equal to 50 mg/dL.

Per cent time in Hypoglycemia (BG ≤ 70 mg/dL) (user-selected low target):
Percentage of time (per-subject) spent with a glucose level less than or equal to 70 mg/dL.

Per cent time in Euglycemia (Low Target mg/dL < BG ≤ High Target mg/dL)
Percentage of time (per-subject) spent with a glucose level within the user specified target range.

Per cent time in Hyperglycemia (BG > High Target mg/dL) (user-selected high target):
Percentage of time (per-subject) spent with a glucose level greater than 180 mg/dL.

Per cent time in Severe Hyperglycemia (BG > 300 mg/dL):
Percentage of time (per-subject) spent with a glucose level greater than 300 mg/dL.

Post Prandial AUC per gram of CHO:
Post prandial Area Under the Curve (> pre-meal BG) divided by the total grams of carbohydrates consumed averaged over all of the meals (during regulation).
Mean[(Sum(BG- pre-meal mean BG))/gCHO].

Low BG Index (LBGI):
LBGI is a measure of the frequency and extent of low BG readings. The LBGI provides a per-subject analysis of long-term risk for hypoglycemia. These risk values can be used to identify Minimal (LBGI<1.1), Low (1.1 ≤ LBGI < 2.5), Moderate (2.5 ≤ LBGI < 5), and High-risk groups (LBGI>5.0).

High BG Index (HBGI):
HBGI is a measure of the frequency and extent of high BG readings. The HBGI provides a per-subject analysis of long-term risk for hyperglycemia. These risk values can be used to identify Minimal (HBGI<5.0), Low (5.0 ≤ HBGI < 10.0), Moderate (10.0 ≤ HBGI < 15), and High-risk groups (HBGI>15.0).

BG Risk Index (BGRI):

The BG Risk Index is simply the per-subject sum of the HBGI and LBGI values. The BGRI provides a metric for identifying individual subjects who are prone to extreme Glucose Control-related events.

SD of BG Rate of Change:
The BG Rate of Change is computed as the ratio of glucose readings over intervals of 15 minutes during the simulation's execution. The standard deviation of these values is calculated to provide a measure of the stability of closed-loop control within the model.

B.1 Outcome Graphs

The Distributed T1DMS creates the following Glucose Control-relevant graphs:

Blood Glucose (Trace) -- Population/Subject:
The blood glucose trace graph provides a per-subject or per-population BG trace for the simulation time. BG values are represented as a black line prior to regulation and as a green line during regulation.

Blood Glucose (Density) -- Population/Subject:
The blood glucose density graph provides a probability distribution of glucose values. The graph is divided into three sections based on the simulation's target range. The calculated probabilities of the glucose readings are displayed, numerically, within each target area.

Risk Trace -- Population/Subject:
The glucose risk trace graphs display the fluctuations of LBGI (plotted as < zero) and HBGI (plotted as > 0) for each hour of the simulation (during the regulation period). This graph emphasizes large excursions and suppresses fluctuations within the target range in order to highlight the critical variances in blood glucose throughout the simulation.

Aggregated BG Trace -- Subject:
The Aggregated Glucose Trace provides a per-hour graphical display of the clinical zone of glycemic control.

Histogram of BG Rate of Change -- Population/Subject:
The Histogram of Glucose Rate of Change plots the frequency and distribution of various rates of glycemic change (in mg/dL per-minute) during regulation.

Poincare Plot -- Subject:
The Poincare plot provides a graphical summary of a given subject's glucose stability. A smaller and more concentrated collection of values indicates high glucose stability, whereas a widely dispersed region indicates BG-related irregularity (i.e. frequent and extreme glucose excursions).

Control Variability Grid Analysis (per-day/per-subject) -- Population/Subject:
The Control Variability Grid Analysis (CVGA) points represent per-subject glucose extremes. The individual points are computed as follows: for each subject a point is plotted with an x-coordinate and y-coordinate equal to the minimum and maximum BG values over a given observation period, respectively. NOTE: To reject sensor errors the confidence bounds of the x and y values have been set to a 95% spread of the overall BG distribution.

The Epsilon Group
2000 Holiday Drive
Charlottesville, VA 22901 USA

Telephone: (434) 923-6091

info@tegvirginia.com
www.tegvirginia.com

Follow TEG on:

http://www.linkedin.com/company/2653952

http://www.twitter.com/tegvirginia