



M Ű E G Y E T E M 1 7 8 2

BME Villamosmérnöki és Informatikai Kar  
Mérnök-informatikus szak

## SZAKDOLGOZAT FELADAT

**Sándor József**

részére

# In-silico szimulációs keretrendszer megvalósítása Julia környezetben

Az emberi szervezet metabolikus egyensúlyának fenntartása az intenzív osztályokon fekvő betegek esetén gyakran komoly kihívás az intenzív terápiás szakorvosok számára. A vércukorszint szabályozása az emberi szervezet egyik legösszetettebb élettani folyamatainak eredménye. Az intenzív ápolás során sajnos gyakran szükség van inzulin terápiát alkalmazni a betegek vércukorszintjének normoglikémiás tartományban tartásához. Nemzetközi együttműködésben az Irányítástechnika és Informatika Tanszék Orvosi Informatika Laboratóriumának közreműködésével közelmúltban kidolgozásra került a STAR (Stochastic TARgeted Control) szoros vércukor szabályozási protokoll, mely hatékony megoldást ad az intenzív osztályon alkalmazott ún. szoros vércukor kontroll protokoll megvalósítására.

A protokoll, ill. a protokoll megvalósítását támogató szoftver rendszer fejlesztésének fontos eszköze az in-silico szimulációs környezet, mely segítségével biztonságosan lehet a protokoll végrehajtását valós betegadatok segítségével virtuális betegeken kipróbálni. A hallgató feladata, hogy Julia környezetben valósítsa meg egy ilyen szimulációs környezetet.

Elvégzendő feladatok:

- Irodalomkutatás, korábban a témában született modellek és módszerek megismerése, különös tekintettel a STAR szoros vércukor szabályozási protokollra és az ICING (Intensive Control Insulin-Nutrition-Glucose) modellre.
- Az ICING modell implementációja Julia környezetben, az implementált modell verifikációja.
- In-silico környezet megtervezése és kialakítása, kiértékelő módszerek megvalósítása.
- In-silico környezet verifikációja szimulált és valós betegadatok felhasználásával.
- Az eredmények értékelése, orvosi és mérnöki szempontok alapján.

**Tanszéki konzulens:** Prof. Benyó Balázs, IB325, 463-1416  
dr. Szlávecz Ákos, IB311, 463-4027  
Szabó Bálint, IB311, 463-4027

Budapest, 2021. szeptember 10.

dr. Kiss Bálint  
tanszékvezető



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Irányítástechnika és Informatika Tanszék

Sándor József

**IN-SILICO SZIMULÁCIÓS  
KERETRENDSZER  
MEGVALÓSÍTÁSA JULIA  
KÖRNYEZETBEN**

KONZULENSEK

Prof. Benyó Balázs

dr. Szlávecz Ákos

Szabó Bálint

BUDAPEST, 2021

# Tartalomjegyzék

<b>Összefoglaló .....</b>	<b>6</b>
<b>Abstract.....</b>	<b>7</b>
<b>1 Bevezetés .....</b>	<b>8</b>
1.1 Motiváció .....	8
1.2 Feladat értelmezése .....	9
1.3 Részfeladatok.....	9
1.4 Szakdolgozat felépítése.....	10
<b>2 Irodalmi háttér, módszerek, felhasznált adatok .....</b>	<b>11</b>
2.1 Élettani háttér .....	11
2.2 STAR protokoll.....	12
2.3 In-silico validáció .....	13
2.4 Julia nyelv .....	14
<b>3 Eredmények.....</b>	<b>15</b>
3.1 Elkészült rendszer bemutatása .....	15
3.1.1 Felsőszintű architektúra .....	15
3.1.2 Részletes megvalósítás .....	16
3.1.3 Felhasználói felület .....	26
3.2 Tervezés és implementáció kihívásai.....	27
3.2.1 Adatok beolvasása .....	27
3.2.2 Protokollok beépítése.....	28
3.2.3 JavaCall használata .....	29
3.2.4 Differenciálegyenletek megoldása.....	30
3.2.5 Statisztikai kiértékelés .....	32
3.2.6 Mintavételezés .....	34
3.2.7 Adatok ábrázolása.....	35
3.3 Rendszer validálása.....	36
3.3.1 Rendszer validálása Matlab referencia alapján.....	36
3.3.2 Rendszer validálása statisztikai szempontok alapján.....	40
3.3.3 Rendszer validálása problémás betegeken.....	42
3.3.4 Az ICING2 modell verifikációja .....	44
3.4 Eredmények értékelése .....	48

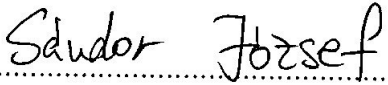
3.4.1 Elkészült rendszer értékelése mérnöki szempontból .....	48
3.4.2 Elkészült rendszer értékelése orvosi szempontból.....	49
<b>4 Összefoglalás.....</b>	<b>51</b>
4.1 További fejlesztési lehetőségek .....	51
<b>Irodalomjegyzék.....</b>	<b>53</b>
<b>Függelék.....</b>	<b>54</b>

## HALLGATÓI NYILATKOZAT

Alulírott **Sándor József**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2021. 12. 09.

  
.....  
Sándor József

# Összefoglaló

Az emberi szervezet metabolikus egyensúlyának fenntartása az intenzív osztályokon fekvő betegek esetén gyakran komoly kihívás az intenzív terápiás szakorvosok számára. Nemzetközi együttműködésben az Irányítástechnika és Informatika Tanszék Orvosi Informatika Laboratóriumának közreműködésével közelmúltban kidolgozásra került a *STAR (Stochastic TARgeted Control)* szoros vércukor szabályozási protokoll, mely hatékony megoldást ad az intenzív osztályon alkalmazott ún. szoros vércukor kontroll protokoll megvalósítására, amellyel az intenzív osztályon ápoltságok mortalitása akár 25-40%-kal csökkenthető.

A protokoll, illetve a protokoll megvalósítását támogató szoftver rendszer fejlesztésének fontos eszköze az *in-silico* szimulációs környezet, mely segítségével biztonságosan lehet a protokoll végrehajtását valós betegadatok segítségével virtuális betegeken tesztelni. A jelen munka célja, hogy *Julia* környezetben valósuljon meg egy ilyen szimulációs környezet.

A munkám során megterveztem, implementáltam, majd leteszteltem a *Julia* alapú *in-silico* szimulációs környezetet. A rendszer konvertálja *Java* bináris, illetve *Matlab* fájlokból a betegek klinikai adatait *Julia* adatstruktúrákba. *Julia* kódból kellett futtatni a *STAR* protokollt megvalósító *Java* kódot, vagyis *Java* függvényeket kellett hívni *Julia* környezetből. A *STAR* protokollon kívül, még másik két protokollt megvalósító kontroller is integrálva lett a szimulációs környezetbe. Kihívást jelentett az úgynevezett *ICING2* modell differenciálegyenleteinek az implementálása és megoldása, amelyek leírják az emberi szervezet metabolikus működését, illetve folyamatait. Végül elkészült egy olyan modul is, amely a szimulálás, illetve valós klinikai kezelés során mért vércukor adatokból különböző statisztikákat készít, valamint különböző szempontok alapján megjeleníti az egyes szimulációk eredményét.

A szakdolgozatban részletesen szó lesz az élettani háttérrel, a *STAR* protokollról, az *ICING2* modellről, a megvalósításhoz használt *Julia* programozási nyelv előnyeiről, a tervezés, illetve implementáció kihívásairól és az elért eredményekről, valamint bemutatásra kerül a szimuláció, illetve a statisztikai kiértékelés folyamata.

Az így elkészült keretrendszer biztonságos, virtuális körülmények között teszi lehetővé különböző kezelési forgatókönyvek kísérletezését, illetve validációját.

## Abstract

Maintaining the metabolic balance of the patients in intensive care units is often a serious challenge for intensive care specialists. In international cooperation with the Medical Informatics Laboratory of the Department of Control Engineering and Information Technology, the STAR (Stochastic TARgeted Control) tight glycemic control protocol was recently developed, which provides an effective solution for the implementation of the close blood glucose control protocol in the intensive care unit, with that the mortality of patients in intensive care units can be reduced by up to 25-40%.

An important tool for the development of the protocol and the supporting software system is the in-silico simulation environment, which allows you to safely test the execution of the protocol on virtual patients using real patient data. The aim of this work is to create such a simulation environment in Julia programming language.

During my work, I designed, implemented and tested the prepared system. During the implementation, firstly, it was necessary to read the clinical data of patients from Java binary and Matlab files into Julia data structures. Then I had to call Java functions from Julia code to implement the STAR protocol. In addition to the STAR protocol, two other protocol controllers have been integrated into the simulation environment. Another challenge was the implementation and solution of differential equations of the ICING2 model, which describe the metabolic functioning and processes of the human body. Finally, a module was created that generates different statistics from blood glucose data measured during simulation or real-world clinical treatment and creates graphs of each simulation based on different criteria.

In the following, we will discuss in detail the physiological background, the STAR protocol, the ICING2 model, the advantages of the Julia programming language used for implementation, the challenges of planning and implementation and the results achieved, as well as the simulation and statistical evaluation process.

The created framework enables experimentation and validation of different treatment scenarios in secure, virtual conditions.

# 1 Bevezetés

Az emberi szervezet metabolikus egyensúlyának fenntartása az intenzív osztályokon fekvő betegek esetén gyakran komoly kihívás az intenzív terápiás szakorvosok számára. A vércukorszint szabályozása az emberi szervezet egyik legösszetettebb élettani folyamatainak eredménye. Az intenzív ápolás során sajnos gyakran szükség van inzulin terápiát alkalmazni a betegek vércukorszintjének normoglikémiás tartományban tartásához. Nemzetközi együttműködésben az Irányítástechnika és Informatika Tanszék Orvosi Informatika Laboratóriumának közreműködésével közelmúltban kidolgozásra került a *STAR* (Stochastic TARgeted Control) szoros vércukor szabályozási protokoll [2], mely hatékony megoldást ad az intenzív osztályon alkalmazott ún. szoros vércukor kontroll protokoll megvalósítására.

A protokoll, illetve a protokoll megvalósítását támogató szoftver rendszer fejlesztésének fontos eszköze az *in-silico* szimulációs környezet, mely segítségével biztonságosan lehet a protokoll végrehajtását valós betegadatok segítségével virtuális betegeken tesztelni, a kezelés eredményét elemezni. A jelen munka célja, hogy *Julia* környezetben valósuljon meg egy ilyen szimulációs környezet [5].

## 1.1 Motiváció

Intenzív osztályon komoly kihívást jelent a páciens vércukorszintjének normoglikémiás tartományban tartása, amellyel az intenzív osztályon ápolott betegek mortalitása akár 25-40%-kal csökkenthető. A betegek tápanyag bevitelét, illetve inzulin adagolását mesterséges úton kell biztosítani. Erre úgynevezett szoros vércukor szabályozást (*Tight Glycemic Control, TGC*) alkalmaznak. A *STAR* (*STochastic TARgeted*) protokoll egy ilyen szoros vércukor szabályozó protokoll [3], melyet a világ több pontján használnak.

Minden biztonságkritikus rendszert az éles üzembehelyezés előtt először alaposan le kell tesztelni. Ez különösen igaz az egészségügyben használatos szoftverekre. Ehhez a verifikációhoz, illetve validációhoz nyújt segítséget a megalkotott keretrendszer.



## 1.2 Feladat értelmezése

A feladat egy úgynevezett *in-silico* validációs környezet megalkotása volt, amelynek lényege, hogy valós betegek adatai alapján, de biztonságos virtuális környezetben szimulált kezelés adatairól, a kezelés eredményességéről szolgáltat információt.

A *STAR* protokollnak létezik egy szimulációs környezete, melyet *Matlab* környezetben implementáltak. A szakdolgozat célja, hogy a szimulációs környezet új funkciókkal kiegészített, *Julia* nyelven implementált változata létrejöjjön, amely hatékonyan támogatja a szimulációs eredmények statisztikai összehasonlítását, illetve ábrázolását, valamint rugalmasan lehet konfigurálni benne különböző kutatási célú elemzési feladatokat. A *Julia* egy intenzíven fejlesztett programozási nyelv, ráadásul rendkívül rugalmas, így számos előnnyel bír a *Matlabhoz* képest.

## 1.3 Részfeladatok

A szakdolgozatom keretében végzett munka során megterveztem az *in-silico* szimulációs környezetet a meglévő változat működése, illetve az új funkciók figyelembevételével. Ezután implementáltam a rendszert *Julia* nyelven. A rendszer verifikációja, illetve a helyes működésének validációja – orvosi döntéstámogató rendszerről lévén szó – fontos lépése volt a munkámnak.

A kiírásban definiált feladatot több részfeladatra bontottam. Először be kellett olvasni *Java* bináris, illetve *Matlab* fájlokból a betegek klinikai adatait *Julia* adatstruktúrákba. Ezután *Julia* kódból kellett futtatni a *STAR* protokollt megvalósító *Java* kódot, vagyis *Java* függvényeket kellett hívni *Julia* környezetből. A *STAR* protokollon kívül, még másik két protokollt megvalósító kontrollert is integráltam a szimulációs környezetbe. További kihívást jelentett az úgynevezett *ICING2* modell differenciálegyenleteinek az implementálása és megoldása, amelyek leírják az emberi szervezet metabolikus működését, illetve folyamatait. Végül elkészült egy olyan modul is, amely a szimuláció vagy valós klinikai kezelés során mért vércukor adatokból különböző statisztikákat készít, valamint különböző szempontok alapján grafikonokon ábrázolja az egyes szimulációkat. Az előbbieken kívül elkészült két különálló grafikus felhasználói felület a szimulátorhoz és a statisztikai kiértékelő modulhoz egyaránt.

## 1.4 Szakdolgozat felépítése

A szakdolgozatnak négy fő fejezete van, a bevezetés, az irodalmi háttér, módszereket, felhasznált adatokat bemutató fejezet, ez után következik, egy terjedelmében hosszabb, az elért eredmények leírását tartalmazó rész, végül az összefoglaló fejezet.

A továbbiakban részletesen szó lesz az élettani háttérrel, a *STAR* protokollról, az *ICING2* modellről és az azt leíró differenciálegyenletrendszerrel, a megvalósításhoz használt Julia programozási nyelv előnyeiről, a tervezés, illetve implementáció kihívásairól és az elért eredményekről. Részletesen bemutatásra kerül a szimuláció, illetve a statisztikai kiértékelés folyamata. Végül, de nem utolsó sorban pedig szó lesz a továbblépési lehetőségekről is.

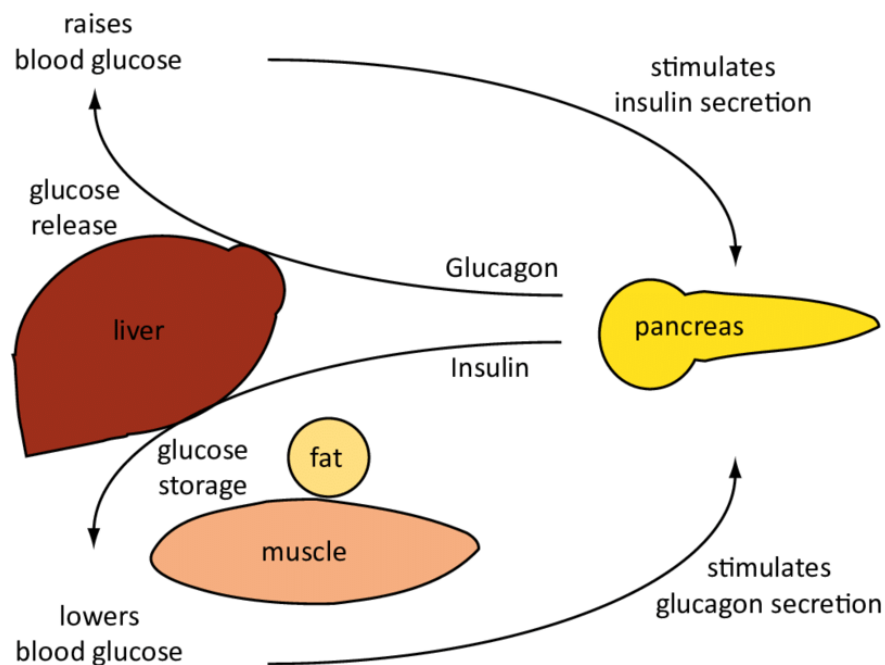
## 2 Irodalmi háttér, módszerek, felhasznált adatok

A keretrendszer tervezésének megkezdése előtt, először az ehhez a területhez tartozó korábbi szakirodalmakkal, kutatásokkal kellett megismerkedni. Jelen fejezetben a felhasznált irodalmi háttér, módszerek és adatok kerülnek bemutatásra.

### 2.1 Élettani háttér

A szoros vércukor szabályozás megvalósítása nehéz feladat, hiszen az az emberi szervezetet, egy nagyon összetett fiziológiai rendszert kell modellezni, ráadásul a betegek egymástól lényegesen különböznek és gyorsan változhat az állapotuk is. Nagyon sok paramétere van a rendszernek, amelyek közül csak néhányat lehet közvetlenül megmérni.

A normoglikémiás vagyis optimális vércukorszint tartomány 4,4 – 6 (8) mmol/l között van. Ha ez a tartomány alá esik a vércukorszint, akkor hipoglikémiáról, ha e fölé, akkor hiperglikémiáról beszélhetünk (1. ábra). Különböző mértékben és időtávon, de mindkettő negatív következményekkel jár [1].



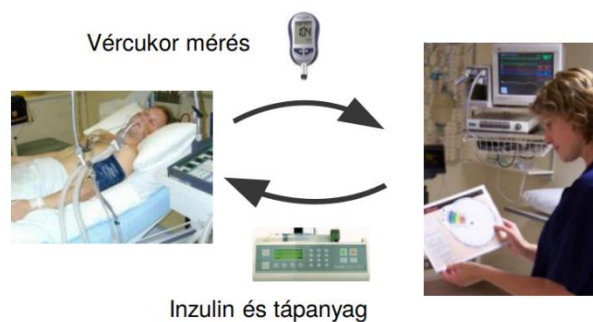
1. ábra: Inzulin és glükóz hatásmechanizmusa

Intenzív osztályon ápolott betegek 35-45 százaléka inzulinterápiára szorul, mert a metabolikus rendszerük egyensúlya felborult. Általában a metabolikus rendszer

zavarának a magas vércukorszint (hiperglikémia) a tünete. Hosszú ideig tartó magas vércukorszint káros következményekkel járhat a beteg kezelésének kimenetelére, ezért alkalmazzák az inzulinterápiát, melynek célja a vércukorszint normoglikémiás intervallumban tartása. Az orvosok számára a kihívást, a megfelelő inzulinadagolás meghatározása jelenti, mivel az ember vércukorszintjét befolyásoló élettani folyamatok igen bonyolultak, emiatt az inzulinbevétel hatása nehezen kiszámítható, hatása késleltetett, pontos kiszámítása bonyolult. Ezen orvosi probléma megoldására szolgálnak az úgynevezett szoros vércukorszabályozó protokollok, mely a szakszemélyzet számára támpontot adnak a megfelelő inzulinadagolás kiválasztására.

## 2.2 STAR protokoll

A *STAR* (Stochastic Targeted Control) protokoll egy szoros vércukor szabályozó protokoll. „*Nurse-in-the-loop*” típusú rendszer (2. ábra), ami azt jelenti, hogy nem avatkozik be közvetlenül, csak terápiás javaslatot ad, amit a nővér vagy orvos felülbírállhat. A nővér megméri bizonyos beteg paramétereit (jelen esetben a páciens vércukorszintjét), a döntéstámogató rendszer ezekből meghatározza a közvetlenül nem mérhető paramétereit, majd előállít terápiás javaslatokat. A nővér a kiválasztott terápiás javaslat alapján végül beállítja a javasolt értékeket az infúziós pumpákon [2][3][4].



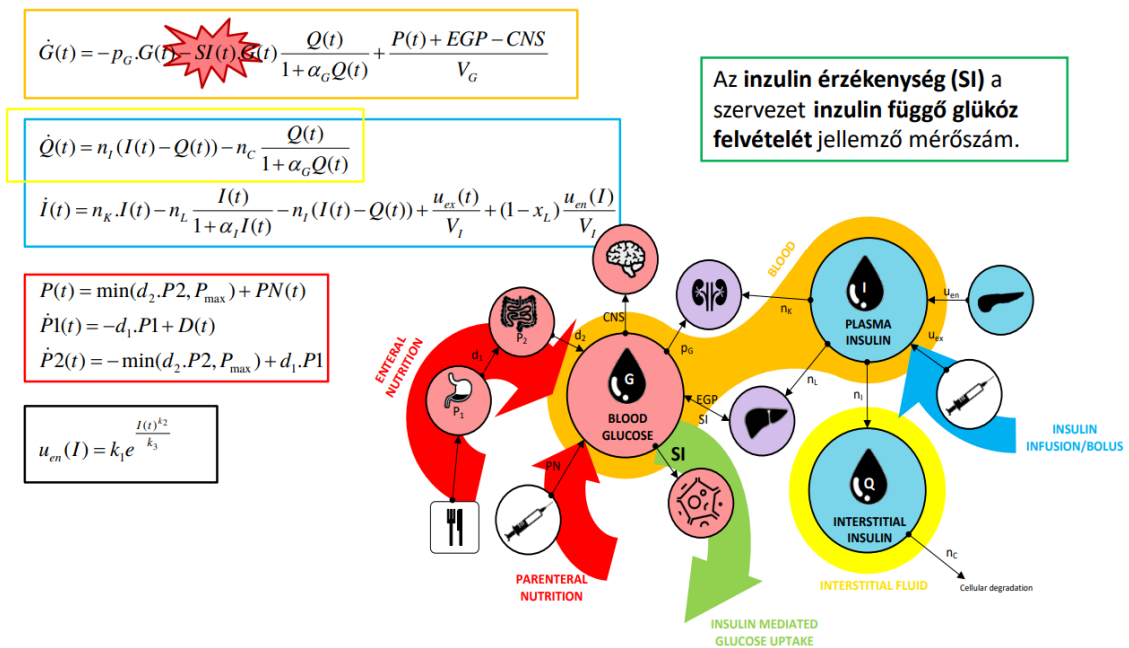
2. ábra: „Nurse-in-the-loop” folyamat

Nagy előnye a *STAR* protokollnak, hogy személyre szabott terápiát lehet segítségével megvalósítani. A protokollt a nővérek és orvosok egy Android platformra, *Java* nyelven implementált, felhasználóbarát kezelői felülettel rendelkező környezetben, egy tablet-en használják mindennapi munkájuk során.

A protokoll az *ICING2* matematikai modellt használja [7], amely egy klinikailag validált metabolikus modell (3. ábra).

A modell számos egyénspecifikus paramétert tartalmaz, amelyek közül a STAR protokoll alkalmazása során az inzulin szenzitivitás paraméter (SI) maradt az egyetlen változó, ezzel reprezentálva a páciens állapotát. Az inzulin érzékenység (inzulin szenzitivitás) (SI) a szervezet inzulin függő glükóz felvételét jellemző mérőszám. Ha ez az érték nő, akkor a beteg érzékenyebb lesz az inzulinra, tehát már kevesebb inzulin hatására is jobban csökken a vércukorszintje.

A protokoll a vércukorszint predikcióját az erre vonatkozó változást leíró differenciálegyenletek megoldásával határozza meg, amelyek az alábbi ábrán láthatóak (3. ábra).



3. ábra: Vércukorszintet befolyásoló fiziológiai modell

## 2.3 In-silico validáció

Az *in-silico* olyan technikára vonatkozik, amikor egy adott kísérletet számítógépes szimuláció segítségével hajtunk végre.

Az *in-silico* szimuláció jelen esetben két fázisból áll:

- Az első fázisban a nyers klinikai adatokat felhasználva létrehozuk a virtuális beteget. Majd, a mért vércukorszintek, inzulinadagolás, illetve táplálás alapján meghatározzuk az időben változó inzulinszenzitivitás görbét.

- A második fázisban futtatjuk a szimulációt a virtuális betegeken, elmentve a mért vércukorszinteket. A végén összehasonlítjuk az eredeti és a szimulált eredményeket és validáljuk a rendszer működését.

Az in-silico szimuláció segítségével lehetséges különböző kezelési protokollok vagy különböző betegcsoportok kezelési adatainak kvantitatív összehasonlítása.

## 2.4 Julia nyelv

Az intenzív osztályokon ápolott betegek szoros vércukor szabályozó protokolljának támogatását végző szoftver működését szimuláló környezet megvalósításához a *Julia* programozási nyelvet [5] használtam. Ennek a nyelvnek számos olyan tulajdonsága van, amely ideálissá tette ennek a feladatnak a megoldására:

- Gyors, úgynevezett *JIT (Just in time)* fordítója van, amely a kód hatékony futásidejű fordítását teszi lehetővé.
- Széleskörű, gazdag könyvtárak érhetőek el benne, számos tudományterületről.
- Alapvetően kicsi, csak a szükséges könyvtárakat tartalmazza. Célzottan lehet importálni az adott feladathoz szükséges könyvtárakat, így magas a hatékonysága.
- Nyílt forráskódú.
- Főként a tudományos számítás (*Scientific Computation*) a fókusza, ami miatt az ilyen irányú tudományos munka előszeretettel alkalmazza és úgy néz ki, hogy a tudományos közösség ebbe az irányba mozdul el. Tipikusan ilyen lehetőség például a *Machine Learning* megoldások, amivel elkezdünk foglalkozni az SI predikció kapcsán.

## 3 Eredmények

A következőkben az elért eredmények kerülnek bemutatásra. Többek között ismertetem a tervezés és implementáció kihívásait, az elkészült rendszer felépítését, értékelem az elkészült környezetet mérnöki, illetve orvosi szempontból, valamint bemutatom annak validációját.

### 3.1 Elkészült rendszer bemutatása

Ebben a fejezetben részletesen bemutatásra kerül a rendszer felépítése és működése, illetve a felhasználói felület.

#### 3.1.1 Felsőszintű architektúra

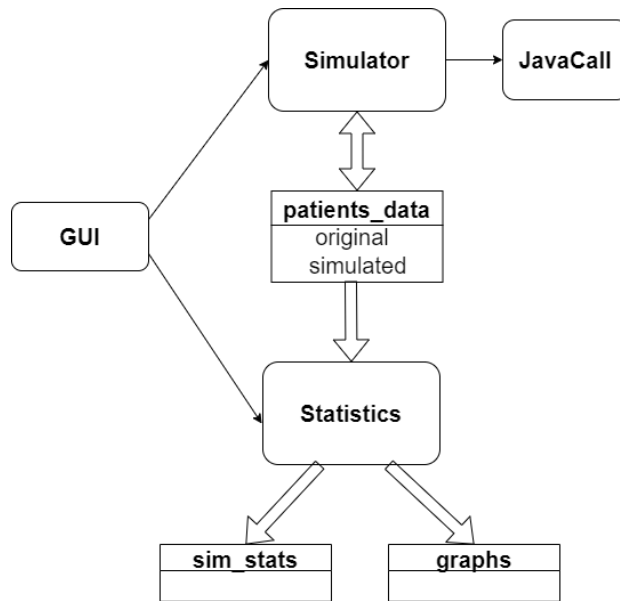
A rendszer megvalósítását megfontolt tervezési fázis előzte meg, így hoztam létre a jelen fejezetben bemutatott architektúrát. Az elkészült rendszer 4 nagyobb komponensből áll, amik 23 *Julia* forrásfájlban vannak megvalósítva.

A *JavaCall* komponens intézi az összes *JVM*-mel való kommunikációt, egy interfészt kínál a többi modul számára.

A *Simulator* komponens felhasználja a *patients\_data/original* mappában lévő betegek adatait és elvégzi a kívánt szimulációt. Ez után az eredményt elmenti a *patients\_data/simulated* mappába.

A *Statistics* komponens felelős minden statisztikai feladat végrehajtásáért. A konkrét statisztikai számítások mellett az adatok szerializációja, előfeldolgozása, illetve megjelenítése is a felelősségi körébe tartozik. Az eredményeit a *graphs*, illetve *sim\_stats* mappákba menti.

A *GUI* komponens feladata a felhasználói interakció konvertálása a *Simulator*, illetve *Statistics* modulok függvényhívásaivá.



4. ábra: Magasszintű architektúra a négy komponenssel és a legfontosabb adatszerkezetekkel

A 4. ábra által mutatott architektúra jelölései a következő jelentéssel bírnak: a lekerekített sarkú téglalapok a komponenseket, a nem lekerekítettek pedig az adattároló egységeket jelölik; az egyszerű nyilak a komponensek függőségeit, a vastag nyilak a fájlrendszerbe mentést, illetve beolvasást jelölik.

### 3.1.2 Részletes megvalósítás

A hatékony áttekinthetőség kedvéért jelen fejezetben felsorolom az elkészült *Julia* keretrendszert alkotó fájlokat. A projekt strukturális felépítését a következő felsoroláson láthatjuk, amelyen jól láthatóak az egyes komponensek és a komponenseket megvalósító forrásfájlok:



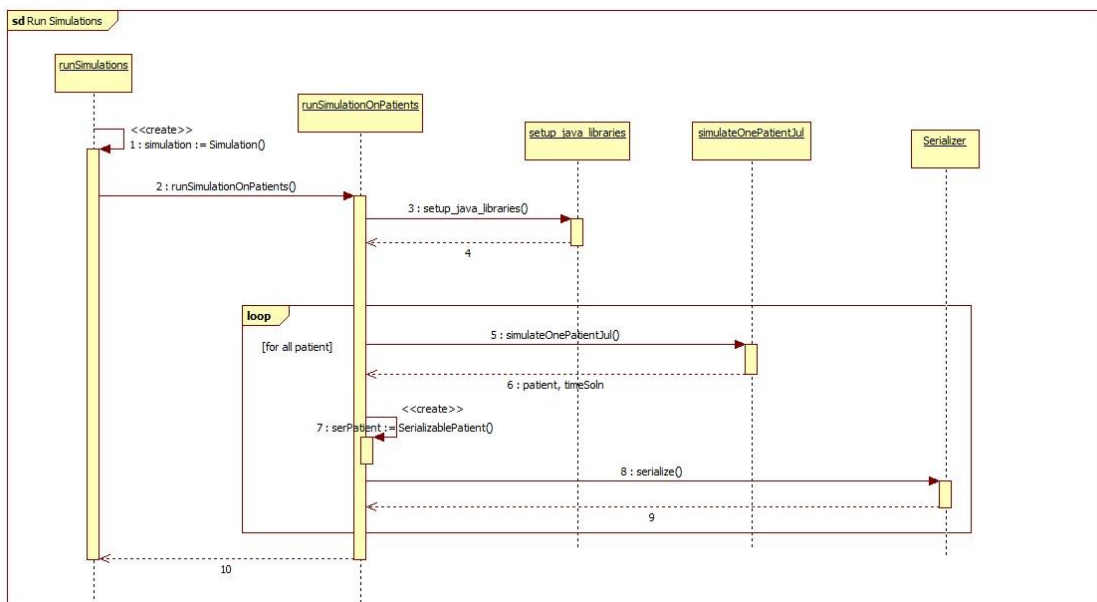
- src
  - JavaCall
    - Star1libs
    - JavaCallHelper.jl
    - JavaClasses.jl
    - setup\_java\_libraries.jl
    - SPRINT\_whole\_cohort.StochasticModel
  - Simulator
    - runSimulations.jl
    - runSimulationOnPatients.jl
    - simulateOnePatientJul.jl
    - HISTORIC\_controller\_simulator.jl
    - SIMPLE\_controller\_simulator.jl
    - STAR\_controller\_simulator.jl
    - ICING2\_model\_sim\_init.jl
    - ICING2\_model\_solver.jl
    - BG\_sensor.jl
    - Simulation\_Structs.jl
  - Statistics
    - runStatistics.jl
    - JuliaStatistics.jl
    - StatisticsCalculator.jl
    - StatisticsExporter.jl
    - Visualiser.jl
    - VisualiserExporter.jl
    - Serializer.jl
    - Resampler.jl
  - GUI
    - SimulatorGUI.jl
    - SimulatorWindow.glade
    - StatisticsGUI.jl
    - StatisticsWindow.glade
- patients\_data
  - original
  - simulated
- graphs
- sim\_stats

### 3.1.2.1 A szimuláció folyamata

A felhasznált betegek historikus adatai a *patients\_data/original/HU\_java* és *patients\_data/original/HU\_mat* mappákban vannak. A *Java* fájlok esetében, betegenként külön mappákba szervezve egy-egy *.GUIData*, *.TimeSoln* meg *.PatientStruct Java* bináris fájlban vannak eltárolva az adatok. Sikerült olyan deszerializáló interfészt létrehozni, amely ugyanúgy képes működni *Java* bináris, *MAT*, illetve *JLD2* (*Julia* szerializációs adattípus) bemeneti fájlok esetén.

A szimuláció futási eredményeképpen létrejön egy *Simresults-actual-date* nevű mappa a *patients\_data/simulated/julia\_results* elérési útvonalon, amelybe páciensenként egy *JLD2* fájlba szerializálódnak ki az aktuális szimuláció eredményei.

A *runSimulations.jl* a program szimulációs részének a belépési pontja, itt lehet beállítani, hogy milyen paraméterekkel fusson a rendszer, ebből kerül meghívásra a *runSimulationOnPatients()* függvény. Ez az utóbbi modul végzi az eredménymappa létrehozását, ha szükséges meghívja a *setup\_java\_libraries()* függvényt, ami inicializálja a *JVM*-et, majd végighaladva a bemeneti mappa tartalmán, minden betegre meghívja a *simulateOnePatientJul()* függvényt (5. ábra).



5. ábra: Szimulációk szekvencia diagramja

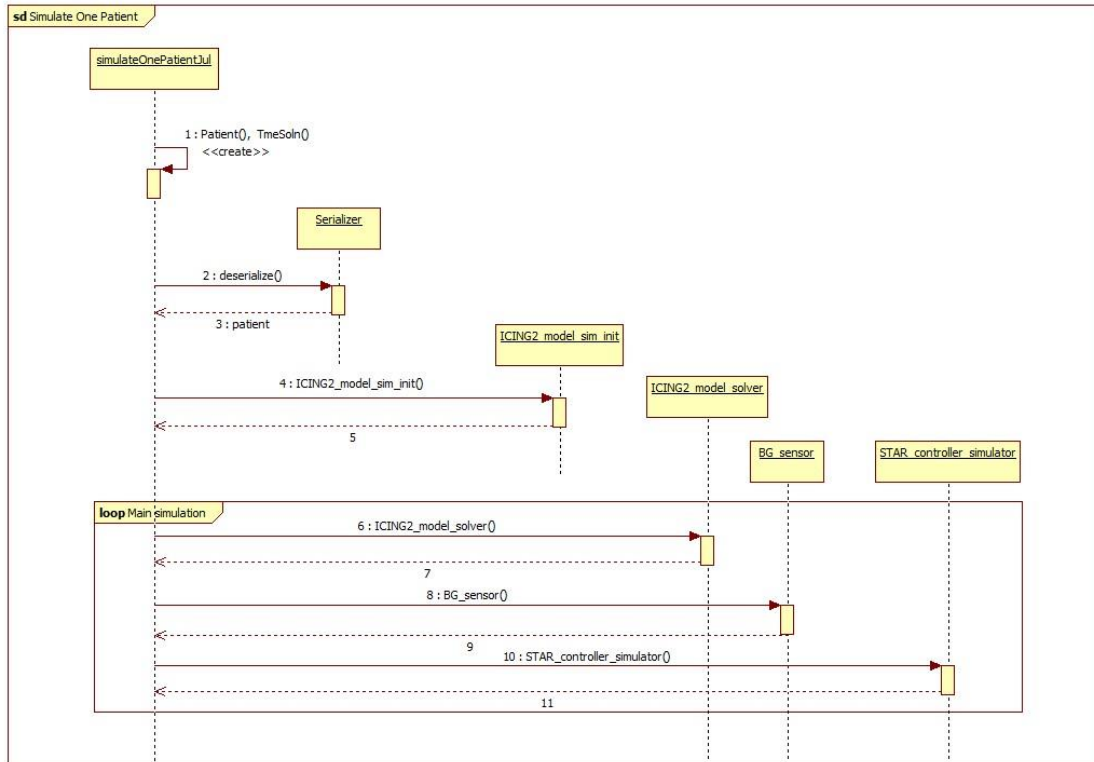
A *SimulateOnePatientJul()* függvény futtatja egy betegre a kiválasztott típusú szimulációt. Először beolvassa a beteg adatait a bemeneti mappából, amely az egységes

interfésznek köszönhetően elrejt, hogy valójában milyen típusú fájlból is történik a beolvasás. Ez a 1. kódrészleten látható módon végezhető el.

```
patient = Simulation_Structs.Patient()
serPatient = Serializer.deserialize(srcPath, name)
patient.Treal_orig = serPatient.Treal
patient.Greal_orig = serPatient.Greal
patient.u_orig = serPatient.u
patient.P_orig = serPatient.P
patient.PN_orig = serPatient.PN
patient.Uo = serPatient.Uo
patient.Po = serPatient.Po
patient.rawSI = serPatient.rawSI
patient.GoalFeed = serPatient.GoalFeed
```

### 1. kódrészlet: Páciens adatainak beolvasása

A fenti módon létrejön a *Patient()* adatstruktúra, melyen kívül még egy *TimeSoln()* változó is szükséges a szimulációhoz. Ezek után van meghívva az *ICING2\_model\_sim\_init()* függvény, ezzel inicializálódnak a használt adatstruktúrák. Majd ciklusban futtatódik az *ICING2\_model\_solver()*, *BG\_sensor()*, illetve a kiválasztott kontrollert megvalósító függvények. Az *ICING2\_model\_solver()* modul felelős a differenciálegyenlet-rendszer vagyis az *ICING2* modell megoldásáért. A *BG\_sensor()* valósítja meg a virtuális „vércukorszint mérést”, vagyis veszi az előzőleg megoldott differenciálegyenlet utolsó értékét és eltárolja azt. A kontrollerek a tápanyag, illetve inzulin bevitelt szabályozzák különböző módokon. A *STAR\_controller\_simulator()* a *JAR* fájllokba hív bele, vagyis a *STAR* protokollt futtatja (6. ábra). A *SIMPLE\_controller\_simulator()* egy egyszerű controller, amely csak az aktuális vércukorszint alapján javasol kezelési módot. A *HISTORIC\_controller\_simulator()* az eredeti kezelési adatokat felhasználva számítja a ki a vércukorszinteket, ez azt jelenti, hogy a virtuális méréseket a valós mérési időpontokban végzi el (a *STAR* ugyanis pontosan 1-2 esetleg 3 óránként), valamint az inzulin és tápanyag adagolás is olyan időben és mennyiségben történik, ahogy az a valós kezelés során is.



6. ábra: Egy páciens szimulációjának szekvencia diagramja

*Simulation\_Structs.jl* fájlban vannak definiálva a szimulációhoz használt *Julia* adatstruktúrák.

A *Java* hívásokat két modul hajtja végre. Az egyik a *JavaClasses.jl*, amelyben a használt *Java* osztályok vannak felsorolva, a másik a *JavaCallHelper.jl*, amelyben *Julia* függvényekbe vannak becsomagolva *Java* függvényhívások. A *SPRINT\_whole\_cohort.StochasticModel* az inzulinérzékenység jövőbeli értékének becslésére szolgáló modellt tartalmazza, az aktuális inzulinérzékenységből bizonyos valószínűséggel meghatározható, hogy egy, két, illetve háromóra múlva milyen tartományba fog esni ez az érték.

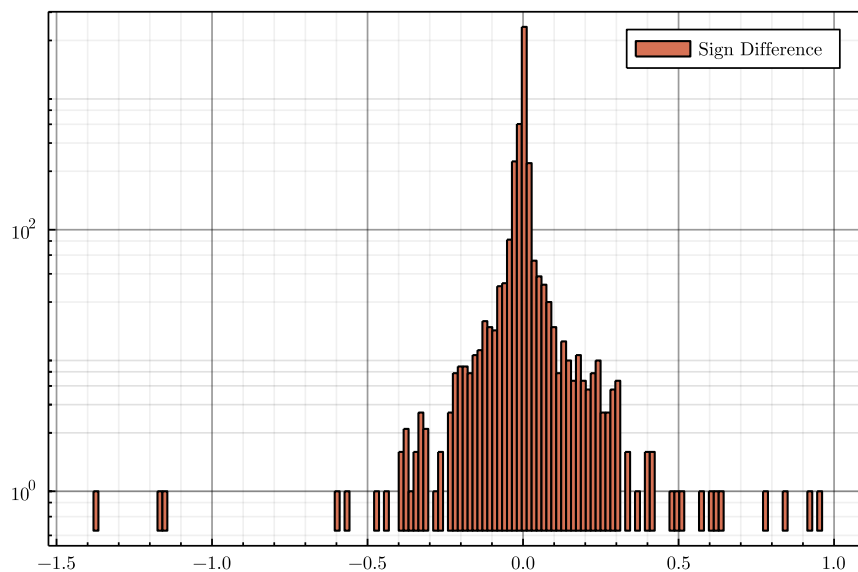
### 3.1.2.2 A statisztikai kiértékelés folyamata

A keretrendszer második fő aspektusa a statisztikai kiértékelés, így ennek a résznek a programnak a belépési pontja a *runStatistics.jl* fájl. Itt lehet felkonfigurálni az elvégzendő feladat paramétereit. A statisztikai kiértékelésért felelős modul alapvetően nagyon rugalmasra lett megtervezve és implementálva, sok kisebb részt lehet kombinálni egymással tetszőlegesen, az orvosi szempontok szerinti következtetést segítő célzattal.

A következőkben felsorolásra kerülő műveletek identikusan működnek *JAVA*, *MAT* és *JLD2* fájlokban tárolt betegadatokra egyaránt, a korábbiakban taglalt deszerializációs interfésznek köszönhetően.

Jelenleg öt fő statisztikai műveletet végző függvény van a programban:

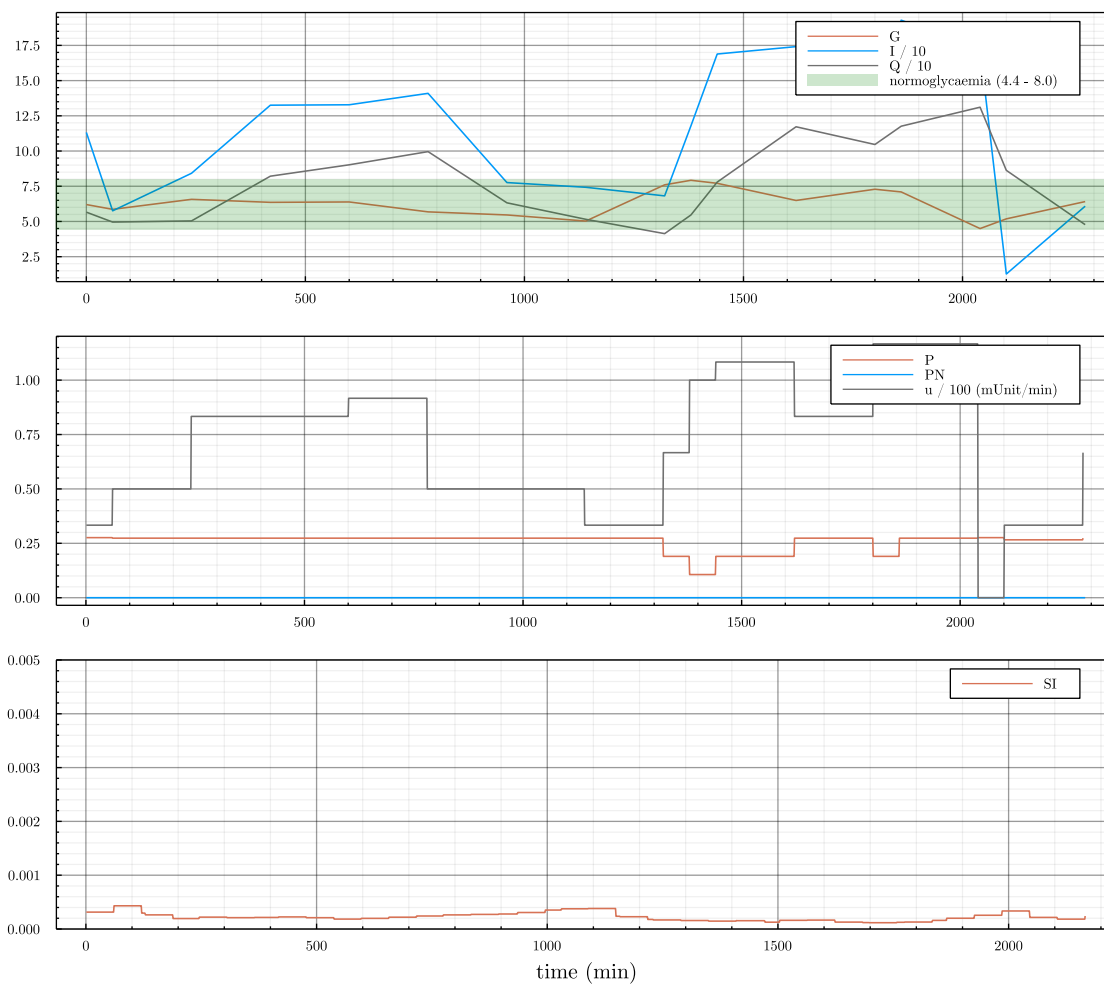
- A *calculate\_treatments\_signDiffBG(path1, path2)*, statisztikai elemzésre lehetőséget adó függvény, előjeles különbséget számol a két paraméterként átadott mappában lévő, azonos betegen végrehajtott kezelés vércukorértékeiből. A számítás eredménye egy hisztogram, amely szemlélteti, hogy mennyire tér el a két szimuláció. Különbség csak azonos időpillanatokban történt vércukormérések esetén van kiszámolva, hiszen csak ez az eset hordoz releváns információt. A 7. ábra egy példa hisztogramot tartalmaz egy *Matlabos* és *Julia* szimuláció vércukorértékeinek az előjeles különbségéről.



7. ábra: Két szimuláció vércukorértékeinek előjeles különbsége hisztogramon ábrázolva

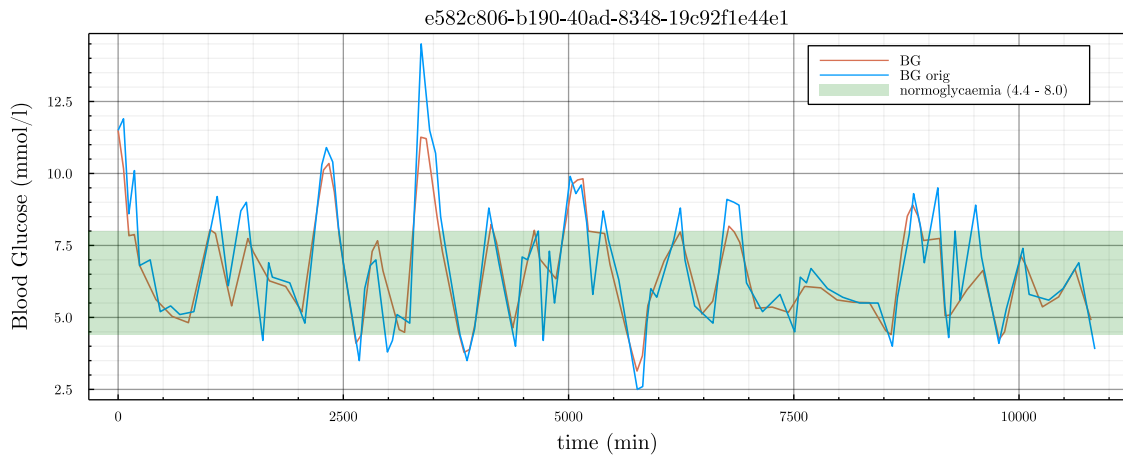
- A *plot\_simulation(path)*, főleg az eredmények vizuális megjelenítéséért felelős eljárás, a paraméterként kapott *path* mappában levő szimulációkról, betegenként három grafikonon állít elő. A *plot\_patient\_metabolics(Patient)* egy átfogó képet ad a páciens kezeléséről, illetve metabolikus állapotáról (8. ábra).

b814a3bf-339f-4411-aff3-0c0d1fbc7d30



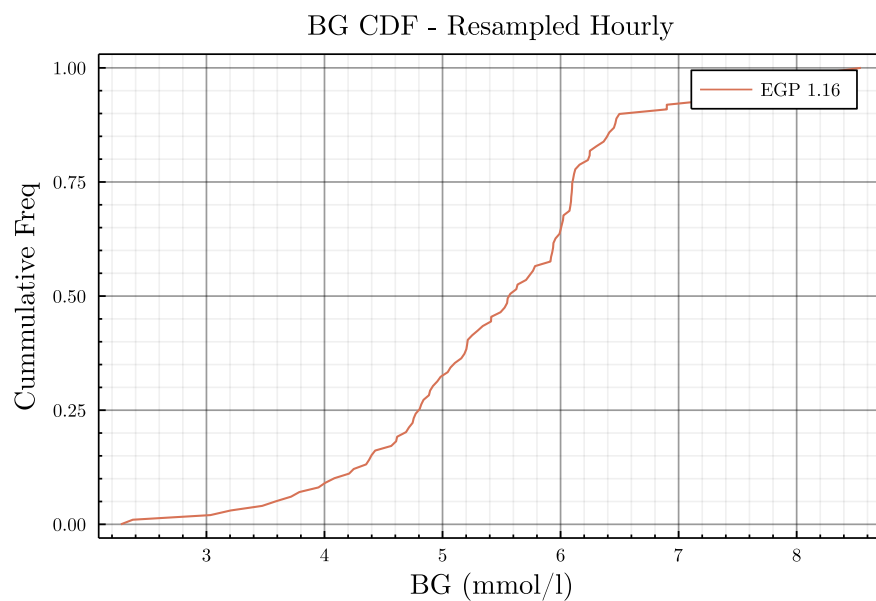
**8. ábra: Példa grafikon egy páciens metabolikus állapotáról**

A *plot\_patient\_BG(Patient)* célja a paraméterként kapott beteg vércukorértékeinek (mmol/l) idő szerinti ábrázolása, ezen kívül feltünteti a beteg historikus, valós korházi körülmények között mért vércukorértékeit is, amely sok alkalommal szolgálhat referenciaként (9. ábra).



**9. ábra: Példa grafikon a szimulált és eredeti vércukorértékekről**

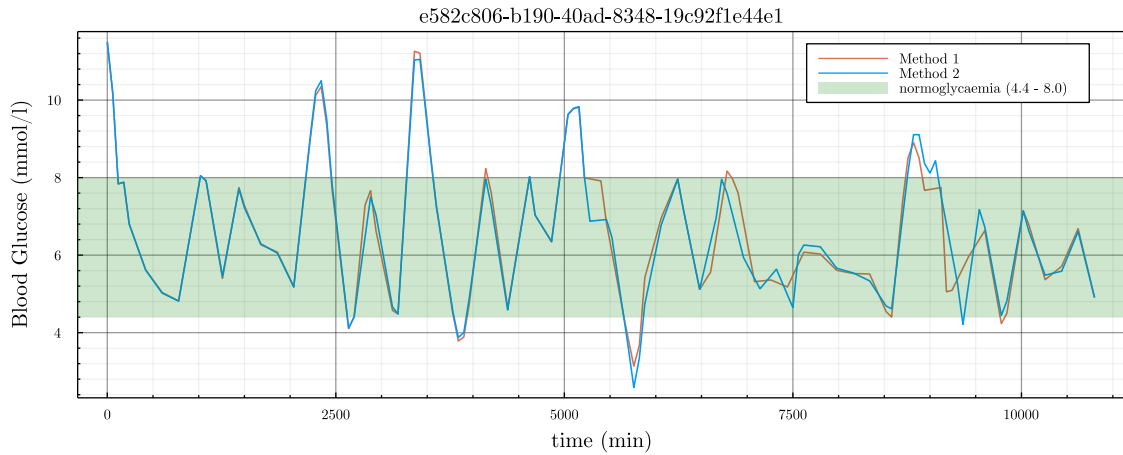
A  $plot\_CDF(Patient.hourlyBG)$  ábrázolja a paraméterként kapott beteg óránként mintavételezett vércukorértékeit  $CDF$  (*Cumulative Distribution Function*) formájában (10. ábra).



**10. ábra: Példa grafikon a betegek szerinti CDF-re**

- A  $create\_statistics(srcpath, dstpath)$  függvény az első paraméterként kapott mappában található páciensek szimulációs adataiból készít összesítő statisztikát, amelyet elment egy *CSV* kiterjesztésű fájlként a második paraméterként kapott elérési útvonalra.
- A  $compare\_treatments(path1, path2)$  függvény vizuálisan hasonlítja össze két szimulációt, tehát egy grafikonon van ábrázolva ugyanannak a betegnek kétfajta kezelés szerinti adatai (*Method 1* és *Method 2*). Az így

elkészült ábrák alapján jól lehet értékelni két kezelés minőségét, illetve a kettő relációját ugyanebből a szempontból. A 11. ábra és 12. ábra ugyanannak a betegnek, de két különböző kezelésből származó szimulációjának adatait tartalmazzák, azzal a különbséggel, hogy a 12. ábra egy részletesebb képet mutat az adott kezelésről.



**11. ábra: Példa grafikon két szimuláció vércukorértékeinek összehasonlítására**

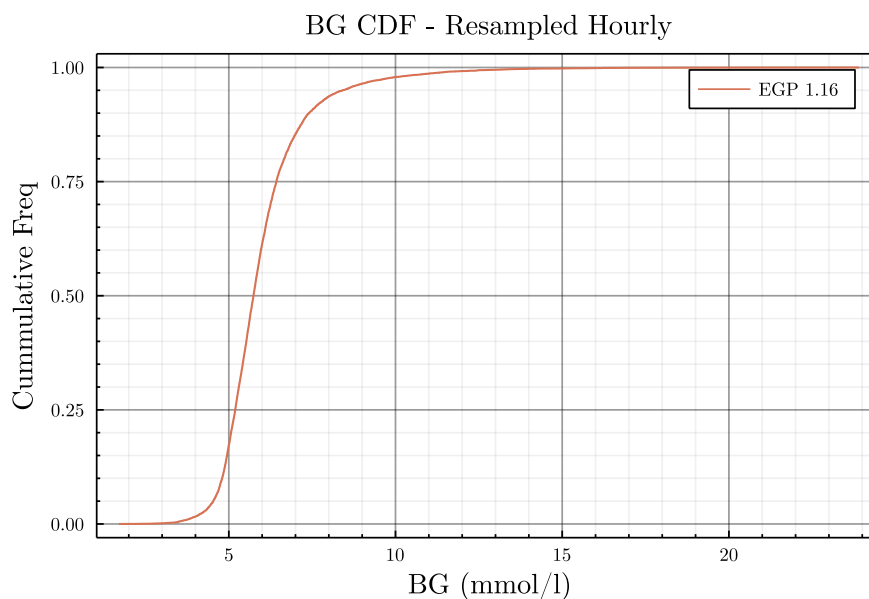


e582c806-b190-40ad-8348-19c92f1e44e1



12. ábra: Példa grafikon két különböző kezelés összehasonlítására

- A  $plot\_cohort\_CDF(path)$  függvény a paraméterként kapott mappában lévő összes szimuláció óránként mintavételezett vércukorértékeiből készít egy összesített  $CDF$ -et ábrázoló grafikont (13. ábra).

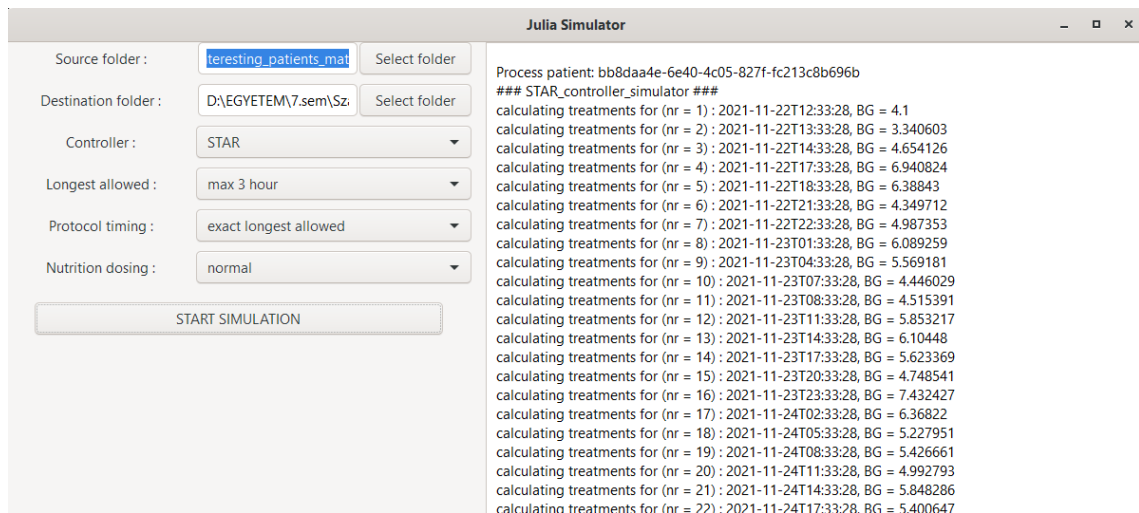


13. ábra: Példa CDF grafikon az összesített óránként mintavételezett vércukorértékekről

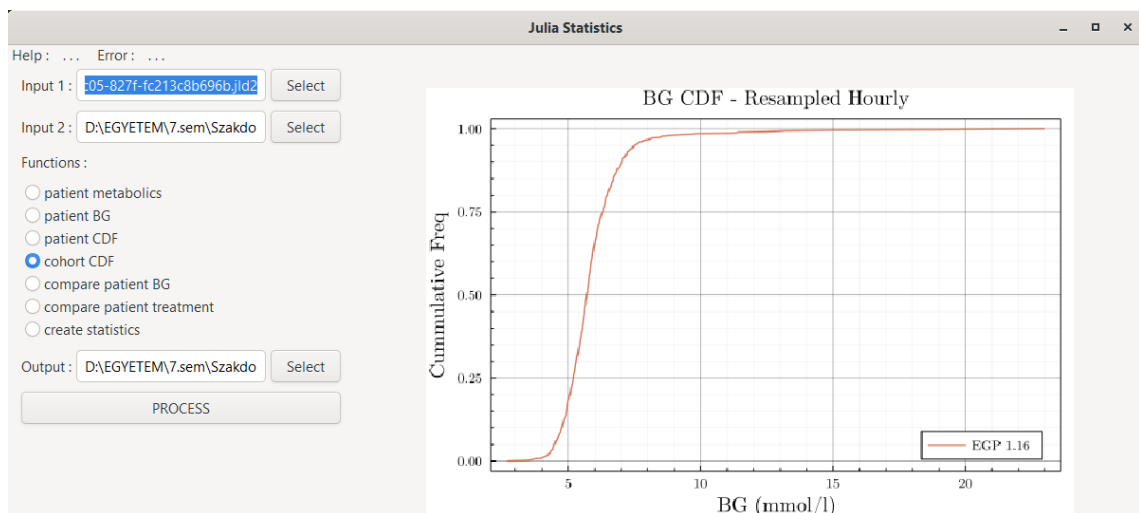
### 3.1.3 Felhasználói felület

A *Julia* nyelv alapvetően a tudományos kutatási területeken terjedt el, a tudományos számításokra lett optimalizálva, tehát már magának a nyelvnek sem elsődleges célja, hogy egy felhasználóbarát alkalmazás legyen elkészíthető a segítségével. A szakdolgozat tervezése során sem volt a követelmények között a grafikus felhasználói felület (*GUI*) kialakítása. Ennek ellenére mégis készült *GUI*, annak érdekében, hogy a *Julia* nyelvben kevésbé jártas kollégák is könnyedén tudják használni az elkészült keretrendszert.

Két különálló felhasználói felületet valósítottam meg, a rendszer két alapfunkciójához egyet-egyét, a szimulációhoz és a statisztikai kiértékeléshez egyaránt (14. ábra, 15. ábra).



14. ábra: A szimulátor felhasználói felülete



15. ábra: A statisztikai kiértékelő felhasználói felülete

Alapvetően a *GUI* segítségével lényegében azonos, az előbbieken említett funkciók hajthatók végre, csak nem kell közvetlenül a *Julia* kódban manuálisan átállítani a paramétereket a kívánt funkció eléréséhez.

## 3.2 Tervezés és implementáció kihívásai

A fejlesztés során számos mérföldkövet definiáltunk, számos módszert és technológiát kellett felhasználni. A továbbiakban ezek kerülnek bemutatásra.

### 3.2.1 Adatok beolvasása

A szimuláció végrehajtásához először be kell olvasni a páciens adatokat. Ezek lehetnek *Java* bináris vagy *Matlab* formátumú fájlokban tárolva. *Matlab* esetén rendelkezésre állt egy beolvasást segítő könyvtár, viszont *Java* fájlok esetében először

létre kellett hozni *Juliában* a *Java* objektumot a szerializált adatból, majd abból áttemelni az adatot a *Julia* által is könnyen használható adatstruktúrákba. A *Julia* keretrendszerben szimulált eredmények a futtatás után *JLD2* formátumú fájlokba kerültek elmentésre.

Az előbbieken taglalt problémára egy deserializációs interfész megvalósítása szolgált megoldásként, amely elfedi, hogy valójában milyen típusú fájlokból történik a páciens, illetve szimuláció adatainak a beolvasása. A deserializációs interfész kódját az 2. kódrészlet tartalmazza.

```
function deserialize(path, patientName)
    type = splitext(readdir(path)[1])[2]
    if type == ".jld2"
        return deserialize_jul(path, patientName)
    elseif type == ".mat"
        return deserialize_mat(path, patientName)
    elseif type == ""
        return deserialize_java(path, patientName)
    else
        throw(ArgumentError("Not existing serializable type."))
    end
end
```

### 2. kódrészlet: Deserializációs interfész

A függvény felismeri, hogy milyen kiterjesztésű a beolvasandó mappa tartalma, a konkrét beolvasást megvalósító függvények további metódusokba vannak kiszervezve. A függvény használatát a 3. kódrészleten láthatjuk.

```
patientName = splitext(filename)[1]
JuliaPatient = Serializer.deserialize(srcpath, patientName)
```

### 3. kódrészlet: Fájlból beolvasás

## 3.2.2 Protokollok beépítése

Mivel a cél egy *in-silico* szimulációs környezet megalkotása volt, ezért a validálni kívánt *STAR* protokoll mellé, még két másik protokoll megvalósítását is beépítettük a rendszerbe. Az egyik egy egyszerű (*SIMPLE*) terápiás eszköz, amely csak az aktuális vércukorszint alapján javasol kezelési módot. A másik, a historikus (*HISTORIC*) kezelési adatokat felhasználva számítja ki a vércukorszinteket. Ez azt jelenti, hogy a virtuális méréseket a valós mérési időpontokban végzi el (a *STAR* ugyanis pontosan 1-2 esetleg 3 óránként), valamint az inzulin és tápanyag adagolás is olyan időben és mennyiségben történik, ahogy az a valós kezelés során is történt a kérdéses beteg esetén. Az utóbbi kontroller rendkívül hasznos az *ICING2* matematikai modell tesztelésére is, hiszen ezzel jól látszik, hogy a virtuális kezelés vércukor értékei mennyire közelítik meg a valós

terápiáét, vagyis, ha változtatunk az *ICING2* modellen, akkor rögtön látszik az okozott eltérés is.

A szimuláció indításakor lehet választani a felsorolt háromfajta protokoll közül, illetve más beállítási módokra is van lehetőség. A *STAR* és *SIMPLE* kontrollerek esetén ki lehet választani, hogy milyen típusú mérési időzítés kerüljön alkalmazásra, a historikus, illetve 1-2, vagy 3 órás. Továbbá a *SIMPLE* kontrollert be lehet állítani úgy, hogy alacsony, közepes vagy magas szén-hidrát bevitelt alkalmazzon.

A felsorolt protokollok segítségével számos elemzési és kutatási feladat elvégezhető, melyek fontos lépései mind a kutatási, mind a kezeléseket elemző kérdések megválaszolásának.

### 3.2.3 JavaCall használata

A legnagyobb nehézséget a munkám során talán a *Juliából* történő *Java* kód hívás megvalósítása jelentette. Ennek a problémának a megoldása elkerülhetetlen volt, hiszen a validálni kívánt protokoll *JAR* fájlok formájában áll rendelkezésre.

A *JavaCall* hívások előtt először be kell tölteni és inicializálni kell egy *JVM*-et (*Java Virtual Machine*), ezt a 4. kódrészleten látható kódsorral tehetjük meg, ahol az átadott paraméterek *JVM* inicializációs argumentumok, jelen esetben ez csak a maximális *Java Heap* mérete.

```
JavaCall.init(["-Xmx128M"])
```

#### 4. kódrészlet: JVM inicializációja a Java Heap paraméter megadásával

*Juliából* a konstruktor híváson kívül csak *Java* függvényeket lehet hívni [6], illetve osztály adattagokat lekérdezni *Java Reflection* szerűen. Azonban például adattagokat beállítani nem lehet közvetlenül, csak függvényhíváson keresztül.

A *JavaCall* hívásokat sikerült hatékonyan leválasztani a program többi részéről, létrehozva egy interfészt a *Java* és *Julia* kód között.

#### 3.2.3.1 Konstruktor használata Java objektumok példányosításához

A konstruktor hívás két lépésben történik. Az első lépésben létre kell hozni egy hivatkozást az osztályra, ez látható a 5. kódrészleten.

```
Java_class_name = @jimport java.net.URL
```

#### 5. kódrészlet: Hivatkozás létrehozása a Java osztályra

Ez után következik az 6. kódrészleten látható példányosítás.

```
Java_object = Java_class_name((param1_type, param2_type, ...), param1_value,  
                               param2_value, ...)
```

#### 6. kódrészlet: Java objektum példányosítása

A példányosítás után már rendelkezésünkre áll a *Java* objektum *Julia* kódban.

### 3.2.3.2 Jcall használata

A létrehozott Java objektum metódusait a 7. kódrészleten látható formában lehet meghívni.

```
Return_value = jcall(Java_object, „method_name”, return_type,  
                    (param1_type, param2_type, ...), param1_value, param2_value, ...)
```

#### 7. kódrészlet: Java metódushívás

### 3.2.3.3 Jfield használata

A *JavaCall* könyvtár csak az adattagok lekérdezését teszi lehetővé, a változtatását kizárólag *setter* függvényhívásokkal lehet megtenni. A *jfield* szintaktikája hasonló a *jcall*-hoz, amit a 8. kódrészlet mutat be.

```
Field_value = jfield(Java_object, „field_name”, field_type);
```

#### 8. kódrészlet: Java adattag lekérdezése

## 3.2.4 Differenciálegyenletek megoldása

A protokoll végrehajtásának szimulációja során, egy adott terápia alkalmazása után ki kell számítani a virtuális beteg következő vércukormérési értékét, amit valós esetben a nővér mér meg kb. 1, 2 vagy 3 órával az előző kezelés után. Mivel rendelkezésünkre áll az *ICING2* validált matematikai modell [7], amely helyesen írja le az emberi szervezet metabolikus folyamatait a vércukor háztartás szempontjából, valamint rendelkezésünkre állnak a páciens historikus SI adatai (állapotának változása), ezért az aktuális kiindulási állapotból az adott táplálási és inzulin bemeneti adatok ismeretében számítani tudjuk, hogy mi lesz a páciens következő mért vércukor értéke (mmol/l).

A szimuláció során szisztematikusan egy kezdeti értékből próbálom megjósolni a vércukorszint további alakulását a változó paraméterek függvényében a következő mérési időpontig. Ennek a változásnak az időbeli alakulása leírható az alább látható ICING2 modell differenciálegyenleteivel (lásd (1) - (7) egyenletek). A differenciálegyenletben a  $G$  a vércukor koncentrációt, az  $I$  a vérben mérhető inzulin koncentrációt, míg a  $Q$  a szövetközi térben mérhető inzulin koncentrációt, a  $P$  és  $PN$  az enterális, illetve parenterális tápanyagbevitelt jelöli. A többi változó jelentését az 1. táblázat tartalmazza.

<b>Model var.</b>	<b>Description</b>	<b>Numerical value [typical range]</b>
$p_G$	Endogenous glucose clearance	$0.006 \text{ min}^{-1}$
$S_I$	Insulin sensitivity	
$\alpha_G$	Saturation of insulin-dependent glucose clearance and receptor-bound insulin clearance from interstitium	165 LmU
$d_1$	Rate of glucose transfer between the stomach and gut	$\frac{-\ln(0.5)}{20} \text{ min}^{-1}$
$d_2$	Rate of glucose transfer from the gut to the bloodstream	$\frac{-\ln(0.5)}{100} \text{ min}^{-1}$
$P_{max}$	Maximal disposal rate from the gut	$6.11 \text{ mmol/min}$
$EGP_b$	Basal endogenous glucose production (unsuppressed by glucose and insulin concentration)	$1.16 \text{ mmol/min}$
$CNS$	Non-insulin mediated glucose uptake by the central nervous system	$0.3 \text{ mmol/L}$
$V_G$	Glucose distribution volume	$13.3 \text{ L}$
$n_I, n_C$	Rate of transport between plasma and interstitial insulin compartments	$0.006 \text{ 1/min}$
$\alpha_I$	Saturation of plasma insulin clearance by the liver	$1.7e - 3 \text{ L/mU}$
$V_I$	Insulin distribution volume	$4 \text{ L}$
$x_L$	First-pass hepatic insulin clearance	$0.67 -$
$n_K$	Clearance of insulin from plasma via the renal route	$0.0542 \text{ 1/min}$
$n_L$	Clearance of insulin from plasma via the hepatic route	$0.1578 \text{ 1/min}$

1. táblázat: Az ICING2 modellben használt élettani paraméterek

$$\dot{G} = -p_G \cdot G(t) - S_I G(t) \frac{Q(t)}{1+\alpha_G Q(t)} + \frac{P(t)+EGP_b-CNS+PN(t)}{V_G} \quad (1)$$

$$\dot{I} = -\frac{n_I I(t)}{1+\alpha_I I(t)} - n_K I(t) - (I(t) - Q(t))n_I + \frac{u_e x(t)}{V_I} + (1 - x_L) \frac{u_{en}}{V_I} \quad (2)$$

$$\dot{Q} = (I(t) - Q(t))n_I - n_C \frac{Q(t)}{1+\alpha_G Q(t)} \quad (3)$$

$$\dot{P}_1 = -d_1 P_1 + P(t) \quad (4)$$

$$\dot{P}_2 = -\min(d_2 P_2, P_{max}) + d_1 P_1 \quad (5)$$

$$u_{en} = k_1 e^{-I(t)^{k_2/k_3}} \quad (6)$$

$$P(t) = \min(d_2 P_2, P_{max}) + PN(t) \quad (7)$$

A vércukorszint meghatározásához tehát implementálni kellett és meg kellett oldani az *ICING2* modell differenciálegyenlet rendszerét *Julia* nyelven.

Ezeknek az egyenleteknek az implementálására tökéletesen alkalmas volt a *Julia OrdinaryDiffEq* könyvtára, amely 4-ed rendű *Runge-Kutta* módszerrel dolgozik (*Tsit5*). A könyvtár a kezdeti értékek megadása után numerikusan, tetszőleges pontossággal oldja meg az egyenletrendszert.

```
prob=ODEProblem(ICING_model_ODE!,ODEinit,(insulinTime[i],insulinTime[i+1]))
Ints = solve(prob, Tsit5(), reltol=1e-8, abstol=1e-8)
```

### 9. kódrészlet: Differenciálegyenletek megoldása

A felhasznált könyvtár segítségével, a 9. kódrészletben szereplő két sorral meg lehet oldani egy definiált differenciálegyenletrendszert két időpont között, adott kezdeti értékek mellett. A megoldást tartalmazó adatsor az *Ints* változóba kerül.

### 3.2.5 Statisztikai kiértékelés

A mért vércukorértékek kiértékelése céljából elkészült egy statisztikát előkészítő, számoló és ábrázoló modul.

A szimulátor által előállított nyers adatokat, páciensenként egy-egy *JLD2* kiterjesztésű fájlba menti el a program. A statisztikát számoló modul ezeket használja fel bemeneti adatként, majd az adott szimuláció statisztikai kiértékelését egy *CSV* kiterjesztésű fájlba menti el.

A statisztika rendkívül széles spektrumú, több megközelítést alkalmaz:



- *Whole Cohort Statistics*

Ezek az adatok egy általános képet adnak a szimulációról olyan adatokkal, mint például a kezelési órák száma, vércukormérések száma (2. táblázat).

- *Raw BG stats*

Ezek a statisztikák a nyers szimulált adatokon számolnak olyan értékeket ki, mint például a mért vércukor értékek mediánja, átlaga, szórása, valamint a mért értékek hány százaléka esik egy bizonyos intervallumba (3. táblázat).

- *Hourly Resampled BG Stats*

Ezek a statisztikák hasonló értékeket számolnak ki, mint az előző pontban említettek, azzal a különbséggel, hogy itt végbemegy egy előfeldolgozási folyamat is, ami annyit jelent, hogy óránként újramintavételezi a nyers adatokat és ezeken végzi el a statisztikai kiértékelést (4. táblázat).

- *Per-episode statistics*

Ezek a statisztikák a külön kezelésekről fogalmazznak meg információkat, nem az összesített adathalmazról (5. táblázat).

- *Intervention Cohort Stats*

Ezek az adatok arról árulnak el információt, hogy összesítve milyen volt a betegek táplálása és inzulin adagolása (6. táblázat).

- *Intervention Per-episode Stats*

Ezek a statisztikák hasonló értékeket számolnak, mint az előző pontban említettek, azzal a különbséggel, hogy itt nem összesítve, hanem kezelésként hajtódik végre a számítás (7. táblázat).

A statisztikai számítások implementációja a *Julia* kódban egy csővezeték segítségével van megvalósítva, amely átlátható és érthető forráskódot eredményez. Így például az enterálisan bevitt glükóz mennyiségnek az óránként mintavételezett kezelések szerinti átlagának a mediánját a 10. kódrészleten látható módon lehet kiszámítani.

```

Median_Enteral_glucose = (@pipe P |>
  map(x -> Resampler.resample_hourly(x, y -> y ), _) |>
  map(x -> mean(x), _) |>
  map(x -> x * 180 * 60 / 1000, _) |>
  quantile(_, [0.25, 0.5, 0.75]) |>
  map(x -> round(x, digits = 2), _))

```

10. kódrészlet: Pipeline használata a statisztikaszámításhoz

### 3.2.6 Mintavételezés

Az előző fejezetben említésre került az egyik típusú statisztikához szükséges előfeldolgozási folyamat, a mintavételezés. Mivel több különböző protokoll létezik a rendszerben, amelyek különböző időzítéseket használnak, ezért ez nem feltétlenül egy triviális feladat. A szimulált jelek mintavételezése alapvetően kétféle lehet, az egyik amikor a vércukorértékeket kell óránként mintavételezni a differenciálegyenletrendszer által előállított sűrű lépésközű adatfolyamból (11. kódrészlet), a másik, amikor az óránként bevitt inzulin és tápanyag mennyiséget kell kiszámolni (12. kódrészlet).

```

function resampleHourlyBG(timeSolnT::Vector{Float64},
timeSolnGIQ::Matrix{Float64})
  hourlyBG = [timeSolnGIQ[1,1]]
  hours = 1
  for i in 2:length(timeSolnT)
    if timeSolnT[i] >= hours * 60
      push!(hourlyBG, timeSolnGIQ[i,1])
      hours = hours + 1
    end
  end
  return hourlyBG
end

```

11. kódrészlet: Vércukorértékek óránkénti mintavételezése

A *resampleHourlyBG()* függvénynek két paramétere van, mindkettő az *ICING2* differenciálegyenlet numerikus megoldásának az eredménye. A *timeSolnT* az idősort, a *timeSolnGIQ* az egyes időpillanathoz tartozó vércukorszintet, a vérben mérhető inzulinkoncentrációt és szövetközi térben mérhető inzulinkoncentrációt tartalmazza. Mindkettő értelemszerűen azonos hosszúságú és sűrű lépésközű adatsor. Az eljárás végighalad az idősoron és minden 60 perc elteltével az aktuális időpillanathoz tartozó vércukorértéket hozzáadja a *hourlyBG* eredmény tömbhöz.

```

function resample_hourly(mtx::Matrix{Float64}, func)
    mtx_hourly = [ func( mtx[1,2]) ]
    cnt = 0.0
    for i in 2:length(mtx[:,1])
        if (mtx[i,1] - mtx[i-1,1] + cnt) < 60
            mtx_hourly[end] = mtx_hourly[end] + func( mtx[i,2] )
            cnt = cnt + (mtx[i,1] - mtx[i-1,1])
        end
        if (mtx[i,1] -mtx[i-1,1] + cnt ) >= 60
            && (mtx[i,1] - mtx[i-1,1] + cnt) < 120
                push!(mtx_hourly, func( mtx[i,2] ) )
                cnt = cnt - 60 + (mtx[i,1] - mtx[i-1,1])
            end
            if (mtx[i,1] - mtx[i-1,1] + cnt) >= 120
                && (mtx[i,1] - mtx[i-1,1] + cnt) < 180
                    push!(mtx_hourly, func( mtx[i,2] ) )
                    push!(mtx_hourly, func( mtx[i,2] ) )
                    cnt = cnt - 120 + (mtx[i,1] - mtx[i-1,1])
                end
                if (mtx[i,1] - mtx[i-1,1] + cnt) >= 180
                    push!(mtx_hourly, func( mtx[i,2] ) )
                    push!(mtx_hourly, func( mtx[i,2] ) )
                    push!(mtx_hourly, func( mtx[i,2] ) )
                    cnt = cnt - 180 + (mtx[i,1] - mtx[i-1,1])
                end
            end
        end
    end
    return mtx_hourly
end

```

### 12. kódrészlet: Infúziós értékek óránkénti számítása

A 12. kódrészletben szereplő *resample\_hourly()* függvény a szervezetbe óránként bevitt infúziós értékeket számítja ki, amelyek az inzulin, valamint a parenterális és enterális táplálék. Az *mtx* mátrix első sora az idősort, a második a hozzátartozó infúziós értékeket tárolja. A *func* nevű paraméterben átadott függvény felelős az adat megfelelő manipulációjáért, megfelelő mértékegységbe való konvertálásáért. Erre a célra a kódban a hívás pillanatában anonim függvény lesz létrehozva és adódik át a *resample\_hourly()* függvénynek.

### 3.2.7 Adatok ábrázolása

A *Julia* nyelv *Plots* könyvtára nagyon rugalmas és változatos diagram készítési lehetőségeket támogat. A szervezet metabolikus állapotát leíró paraméterek, köztük a vércukor értékek, valamint a kezelési paraméterek grafikus ábrázolása nagyon tanulságos tud lenni. Az előzőekben említett statisztikák mellett a vizuális megjelenítés is nagy mértékben hozzájárul a helyes következtetés levonásához az alkalmazott kezelést illetően. Ezért elkészült egy ábrázoló modul, amely többek közt remekül szemlélteti a

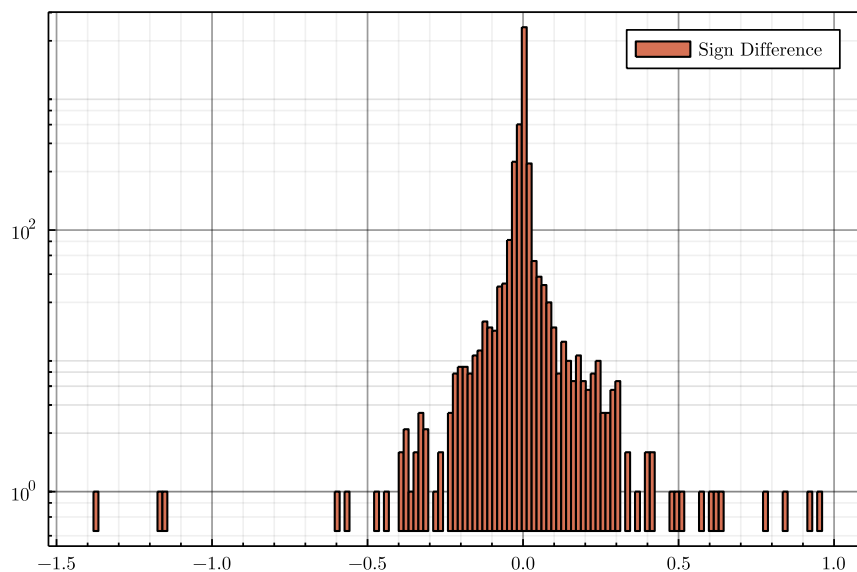
páciens aktuális metabolikus állapotát mérési pillanattól mérési pillanatig. Lehetőség van összehasonlítani segítségével két kezelési módot, egyszerű, időfüggő grafikon, előjeles különbséget ábrázoló hisztogram, valamint *CDF* segítségével.

### 3.3 Rendszer validálása

A rendszer validálásához 112 páciens historikus kezelési adatai állnak rendelkezésre. A rendszert több megközelítéssel is lehet ellenőrizni, háromféle módszert sikerült kidolgozni. Az egyik lehetőség, hogy a már meglévő *Matlab* szimulációs környezet eredményeihez legyenek hasonlítva a *Julia* környezet adatsorai. A másik lehetőség, hogy az egyes protokoll eredményei, statisztikája önmagában, orvosi szempontok alapján értékelődjenek ki. A harmadik lehetőség, hogy kizárólag problémás, vagyis súlyos állapotban lévő pácienseken legyenek vizsgálva a protokollok, ilyenkor közelebbről vizsgáljuk, hogy hogyan viselkedik az egyes protokoll a kritikus pillanatokban. Az előbbi validációs módszereken kívül lehetőség van csak az *ICING2* modellre fókuszálva, annak a verifikációját elvégezni.

#### 3.3.1 Rendszer validálása *Matlab* referencia alapján

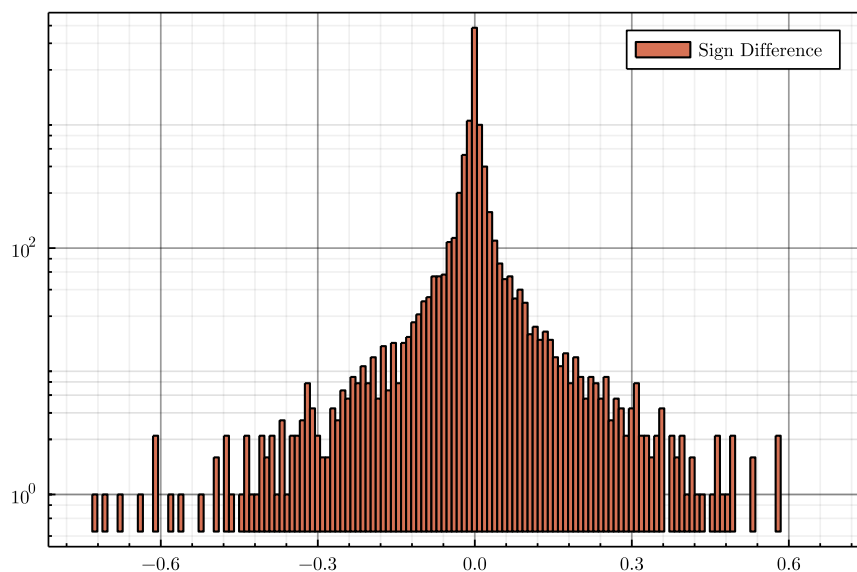
A következőkben egy *Matlab*, illetve *Julia* környezetben futtatott *STAR* kontrollert használó, maximum háromórás kezelési időtartamot alkalmazó szimulációs eredmények kerültek összehasonlításra.



16. ábra: Előjeles különbség *Julia* és *Matlab*, maximum háromórás terápiát alkalmazó szimulációs eredmények között

A 16. ábra által megjelenített hisztogramon láthatóak az előjeles különbségek. Lényeges dolog, hogy csak azonos időpillanatokban mért vércukorértékek lettek feltüntetve. A vizsgált szimulációk során az 5957 mérésből csak 5464 alkalommal egyezett meg az időpont, tehát 493 vércukormérés nem került bele ebbe a statisztikába. Ezek az eltérések a numerikus hibáknak tudhatóak be, valamint a két nyelv különböző differenciálegyenlet megoldó könyvtárat használ, amelyeknek az algoritmusai elviekben azonos (4-ed rendű *Runge-Kutta*), de a konkrét megvalósításban nyilvánvalóan vannak különbségek. A maximális és minimális különbség 0.977 és -1.381, az átlag -0.0006, a szórás 0.0657 volt. Az esetek 99,7 százalékában a hiba +/- 0,5 mmol/l között volt, ami klinikai szempontból fontos, a klinikai gyakorlatban a normoglikémiás tartományban kb. 1 mmol/l-es eltérés felett tekintenek két vércukorszint mérést lényegesen különbözőnek.

Érdeemes két olyan szimulációt vizsgálni, ahol a javasolt kezelési időtartam minden esetben pontosan egy óra, így mindig meg fognak egyezni a vércukormérési időpontok. Ennek az eredménye a 17. ábra hisztogramja.



**17. ábra: Előjeles különbség Julia és Matlab, egyórás terápiát alkalmazó szimulációs eredmények között**

Ebben az esetben a maximális és minimális különbség 0.71 és -0.73, az átlag -0.001, a szórás 0.0657 volt, valamint a 11217 mintavételi időpont nyilvánvalóan mindig megegyezett.

A *STAR* kontroller úgy működik, hogy terápiás javaslatokat állít elő, ha biztonságosnak ítéli meg akkor felajánl három lehetőséget is (egy, kettő és három órára),

jelenleg mindig a lehetséges leghosszabb terápiát választja, tehát ha három óra van beállítva maximálisnak, akkor egy, kettő és háromórás időközönként is történhet a vércukormérés. Ebből kifolyólag nagyobb az esély, hogy a két szimulációban különböző terápiákat javasol ugyanaz a kontroller, ezért a konstans egyórás terápiás szimuláció pontosabb képet ad. Észrevehető ez a két ábrán, meg a hozzá tartozó adatokon is, az egyórás esetben kisebb a szórás, illetve ebben az esetben az értékek eloszlása jobban hasonlít a normál eloszláshoz, mint a háromórás esetben.

Főleg a problémás betegeknél, akiknek az inzulinérzékenysége (SI) erősen változó, van nagyobb eltérés a vércukorértékekben (18. ábra).

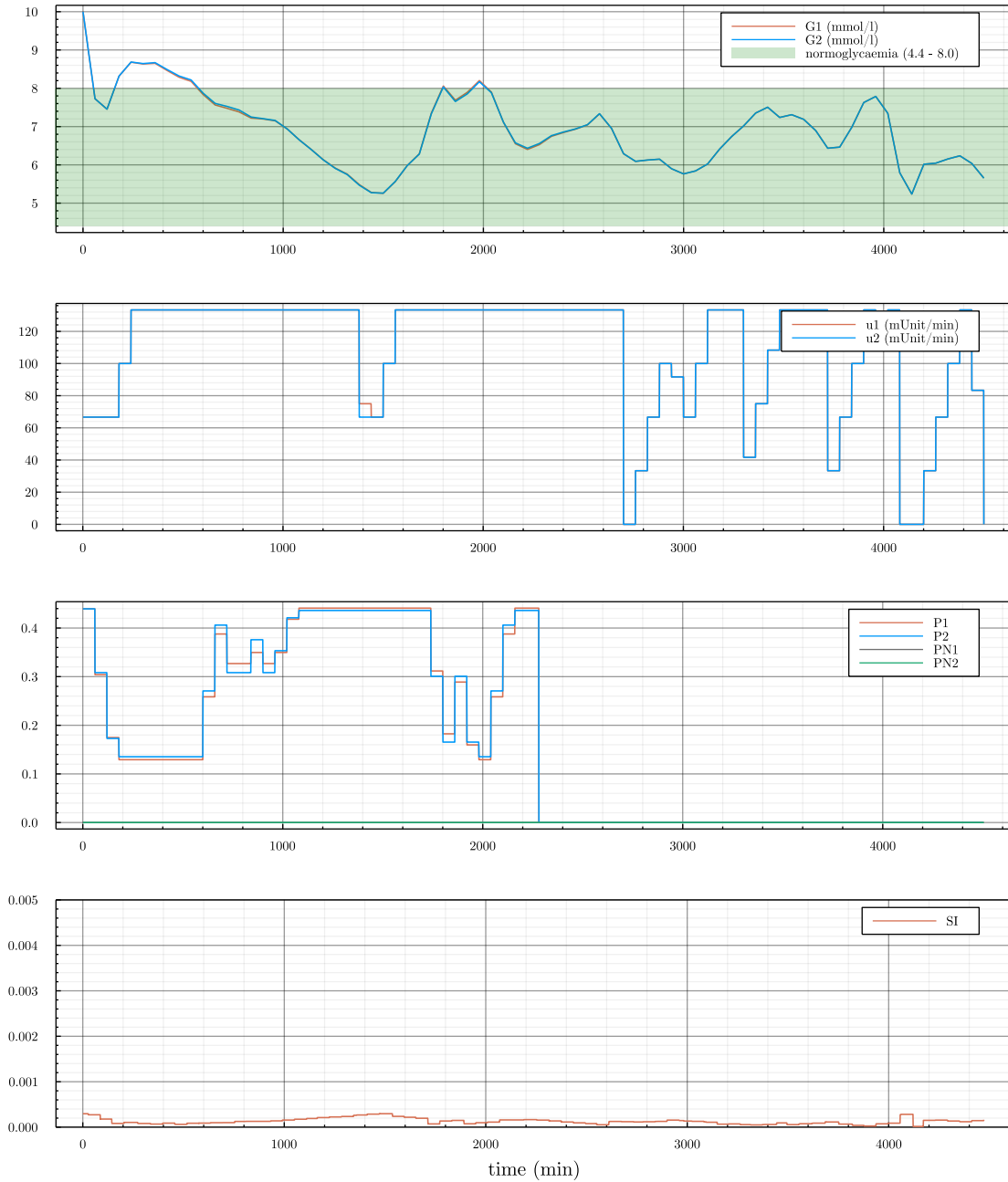
d03fad67-d336-4cf3-a5d4-fcb43fe8f249



**18. ábra: Problémás beteg szimulációja Julia és Matlab környezetben**

Ha az inzulinszenzitivitás stabilabb, akkor az eltérés is kevesebb, hiszen a vércukorpredikció biztosabban végezhető el (19. ábra).

ce26838a-cc3d-40d2-891d-91b741b890dd



19. ábra: Stabil SI értékű beteg szimulációja Julia és Matlab környezetben

### 3.3.2 Rendszer validálása statisztikai szempontok alapján

A következőkben a három controller szimulált, valamint az eredeti bemeneti adatok statisztikai eredményei kerülnek kiértékelésre. Az ebben a fejezetben bemutatott adatok csak töredékét képezik a teljes statisztikának. A függelékben található részletes statisztikai adatok egy orvos számára sokkal többet elárulhatnak egy adott kezelésről, mint amennyi a jelen fejezetben kifejtésre kerül.



### 3.3.2.1 STAR statisztikai kiértékelése

Jelen fejezetben egy maximum háromórás terápiát alkalmazó *STAR* protokoll *Julia* környezetben szimulált eredményei kerültek elemzésre.

A kezelések összesen 11159 órán keresztül történtek és közben 5967 vércukormérés lett rögzítve, ami azt jelenti, hogy a kontroller átlagosan 1.89 órás terápiákat javasolt (2. táblázat).

A 3. táblázatból kiolvasható, hogy a vércukormérések 83 százaléka a normoglikémiás (4.4-8.0 mmol/L) tartományba esik és mindössze két kezelés során fordult elő, hogy a vércukorszint a kritikus 2.2 mmol/L alá essen. Ez igen jó értéknek számít annak tudatában, hogy az adatok olyan betegektől származnak, akik intenzív terápiára szorultak.

### 3.3.2.2 SIMPLE statisztika

Jelen fejezetben háromórás terápiát alkalmazó *SIMPLE* protokoll *Julia* környezetben szimulált eredményei lettek elemezve. Az enterális tápanyagbevitel az összes kezelés során állandó (12. táblázat), parenterális táplálás nincs, valamint a bevitt inzulinegységek száma fix vércukorintervallumokhoz vannak rendelve.

Az elvárások szerint, mivel a *SIMPLE* kevésbé szofisztikált, mint a *STAR*, ezért rosszabbul is teljesít. A különbség legjobban az óránként mintavételezett adatokon figyelhető meg. A *STAR* protokoll óránként mintavételezett vércukorértékeinek 90 százaléka a normoglikémiás tartományba esik, míg *SIMPLE* esetében ez az érték csupán 67 százalék.

### 3.3.2.3 HISTORIC statisztika

Jelen fejezetben egy *HISTORIC* kontrollert alkalmazó, *Julia* környezetben futatott szimuláció eredményei lettek elemezve.

A *HISTORIC* kontroller alkalmas az *ICING2* modell és így a differenciálegyenletmegoldó validálására, hiszen ilyenkor minden infúziós paraméter egyezik az eredeti kezelés paramétereivel.

A *HISTORIC* kontroller az eredeti kezelés vércukormérési időpontjaiban mér, mivel *STAR* protokollt használó kezelés historikus adatai állnak rendelkezésre, ezért ezek az időintervallumok elméletileg egy, két, illetve háromórások lehetnek. Gyakorlatilag 1

és 300 perc között bármilyen érték előfordulhat, ezért ebben az esetben is az óránként mintavételezett statisztika szolgálhat a legrelevánsabb információval.

A 16. táblázat alapján az óránként mintavételezett vércukorértékek 77 százaléka normoglikémiás tartományba esett, amely kevesebb, mint a szimulált *STAR* esetében, de ez a pontatlan mérési időintervallumoknak is betudható.

#### **3.3.2.4 Az eredeti bemeneti adatok statisztikája**

Jelen fejezetben az eredeti kezelési adatokon számolt statisztika kerül bemutatásra, tehát ezek a valós klinikai körülmények között mért értékek kiértékelése. A statisztikát számoló modul bemenete tehát nem a szimulált értékek, épp ezért hiányosak lehetnek néhány helyen, például a valós körülmények között mért értékeket nem lehet óránként mintavételezni.

A 21. táblázat azt mutatja, hogy a valós kezelés során mért vércukorértékek 64 százaléka esett normoglikémiás tartományba, 43 kezelés alkalmával esett a vércukorszint 4,0 mmol/L alá, és 2 alkalommal 2,22 mmol/L alá.

#### **3.3.3 Rendszer validálása problémás betegeken**

A rendszer validálása problémás betegeken módszer megvalósítására a teljes betegállományból három kritikus állapotban lévő páciens lett kiválasztva, mint kísérleti alany. A következőkben elemezve lesz, hogy hogyan működnek a protokollok ezekben az esetekben.

bb8daa4e-6e40-4c05-827f-fc213c8b696b



**20. ábra: Háromórás STAR protokoll működése kritikus helyzetben**

Egy intenzív osztályon ápolott beteg állapota, ha kritikus, az legtöbbször azt jelenti, hogy az inzulinérzékenysége erősen változékony. A fenti grafikonon (20. ábra) látszik, hogy amikor a piros körrel jelölt helyen a vércukorszint 3 mmol/liter alá esik, vagyis hipoglikémia lép fel, akkor a *STAR* protokoll reagál és parenterális glükóz bóluszt adagol a szervezetbe. Az is észrevehető, hogy közvetlen a jelzett időpont előtt a beteg inzulinérzékenysége drasztikusan megnőtt és ettől vált instabillá a rendszer.

c49ddea2-9501-4949-8b13-a8235e2a7f73



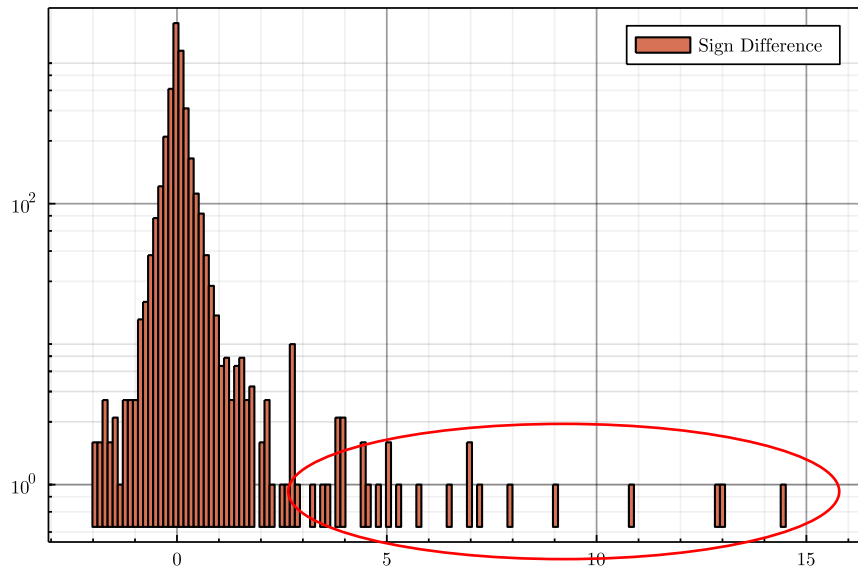
21. ábra: Háromórás STAR protokoll működése kritikus helyzetben

Jelen beteg kezelését láthatóan kritikus állapotban kezdték meg. Ez abból látszik, hogy a vércukorértéke jóval 20 mmol/liter felett van (21. ábra), tehát hiperglikémiás állapotban van a beteg. Alapvetően egészséges embernél akkor lép fel ekkora hiperglikémia, ha nagy stresszhatás (például baleset) éri. Az intenzívterápia megkezdésével határozottan, de nem túl hirtelen kezdett csökkenni a vércukorszint. Kevesebb mint 10 óra alatt sikerült a normoglikémiás tartományba szabályozni a vércukorszintet.

### 3.3.4 Az ICING2 modell verifikációja

Az ICING2 modell implementálásának, vagyis az általa leírt differenciálrendszer megoldásának a helyessége kiemelt fontosságú, hiszen minden egyes szimuláció felhasználja.

A megvalósított kontrollerek közül a *HISTORIC* controller az eredeti táplálási és inzulinadagolási adatokat alkalmazza. Az *ICING2* modell verifikációját úgy lehet a legjobban elvégezni, hogy a *HISTORIC* controllerrel szimulált vércukorértékek legyenek párhuzamba állítva a klinikai körülmények között mért értékekkel. Ez megtehető az egyes páciensek külön grafikonjainak vizsgálatával, valamint a teljes betegállományon vett két adatsornak az előjeles különbségével.

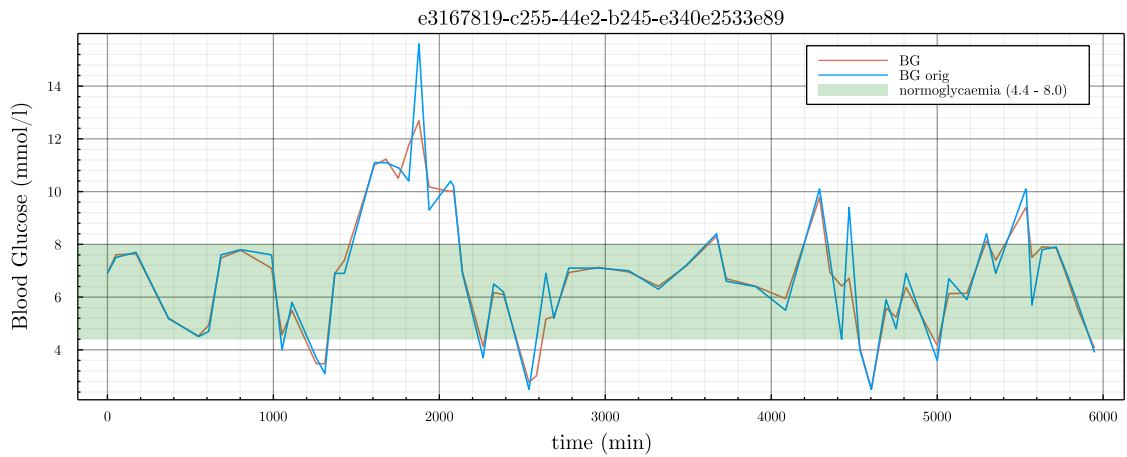


**22. ábra: Eredeti és HISTORIC controller által szimulált vércukorértékeinek előjeles különbségének hisztogramja**

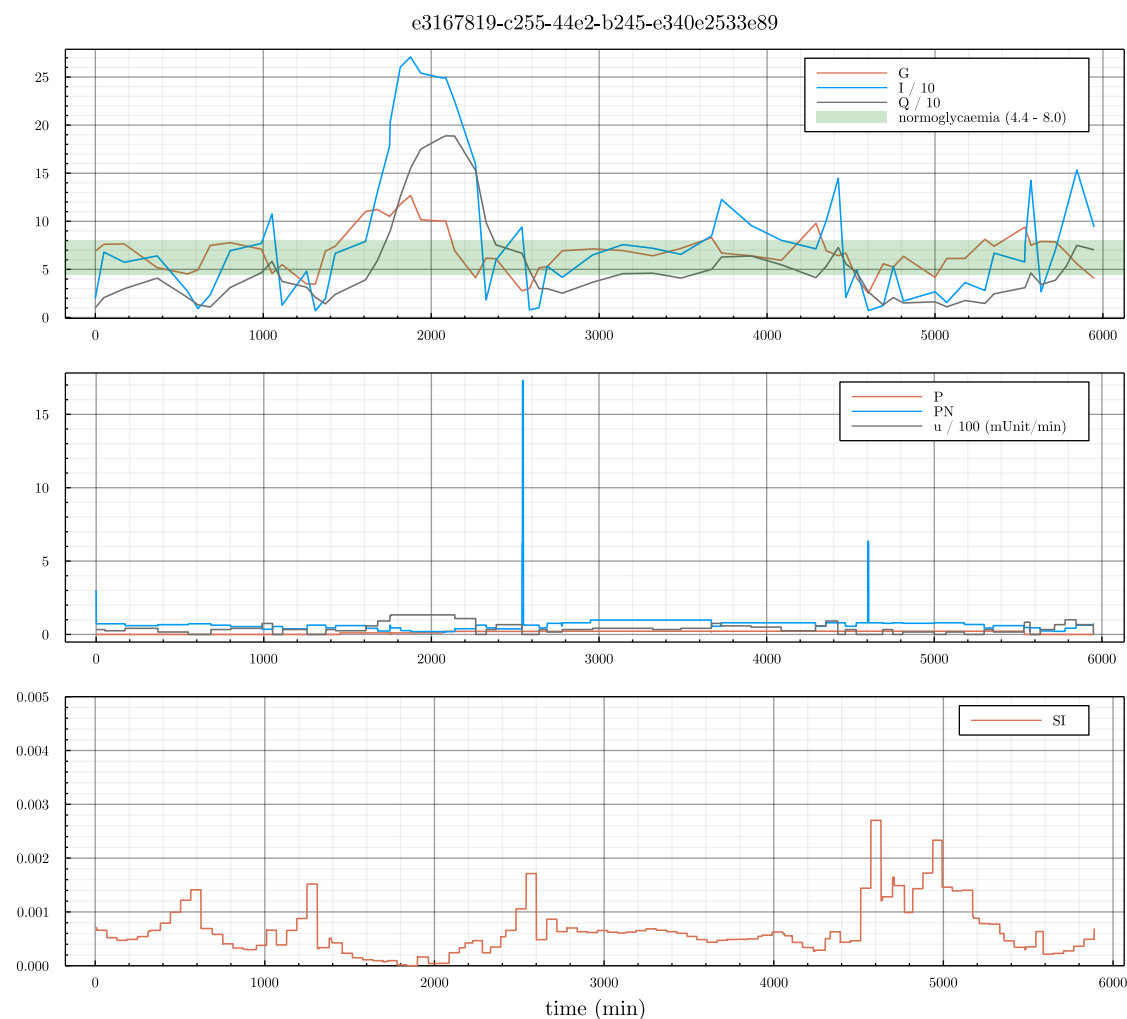
A fent ábrázolt adatsorban a maximális érték 15.95, a minimális -2.01, az átlag 0.0642, a szórás 0.6168. A kiugró esetektől eltekintve az előjeles különbség tekinthető normál eloszlásúnak. Tehát az *ICING2* modell alapvetően jól közelíti a valóságot.

Hangsúlyozandó, hogy az *ICING2* modell megfelelően működik az esetek túlnyomó többségében. A teljesség kedvéért azonban érdemes néhány szót ejteni a 22. ábra piros ellipszissel jelölt kiugró értékek mögött meghúzódó jelenségről.

Több oka is lehet az említett anomáliának, mivel korházban mért adatokról van szó, lehet egyszerű adatrögzítési hiba is a probléma forrása, vagy olyan szcenárió is elképzelhető, hogy a bevitt inzulin kilép a véráramból egy ödémás területre, így késleltetve fejt ki hatását, ez történhetett a 23. ábra és 24. ábra által ábrázolt kezelési esetében is.



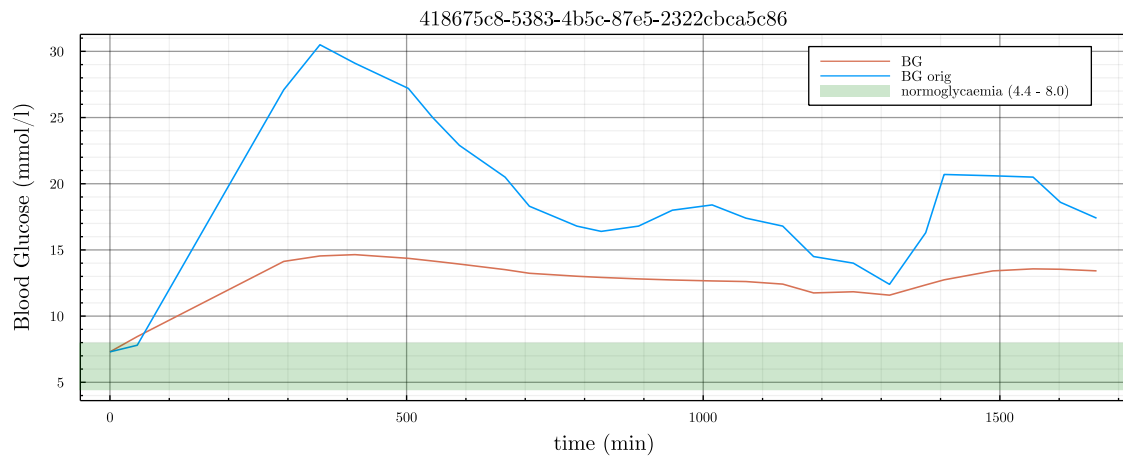
**23. ábra: ICING2 modell tesztelése problémás betegen (vércukorszintek)**



**24. ábra: ICING2 modell tesztelése problémás betegen (teljes terápiás adatok)**

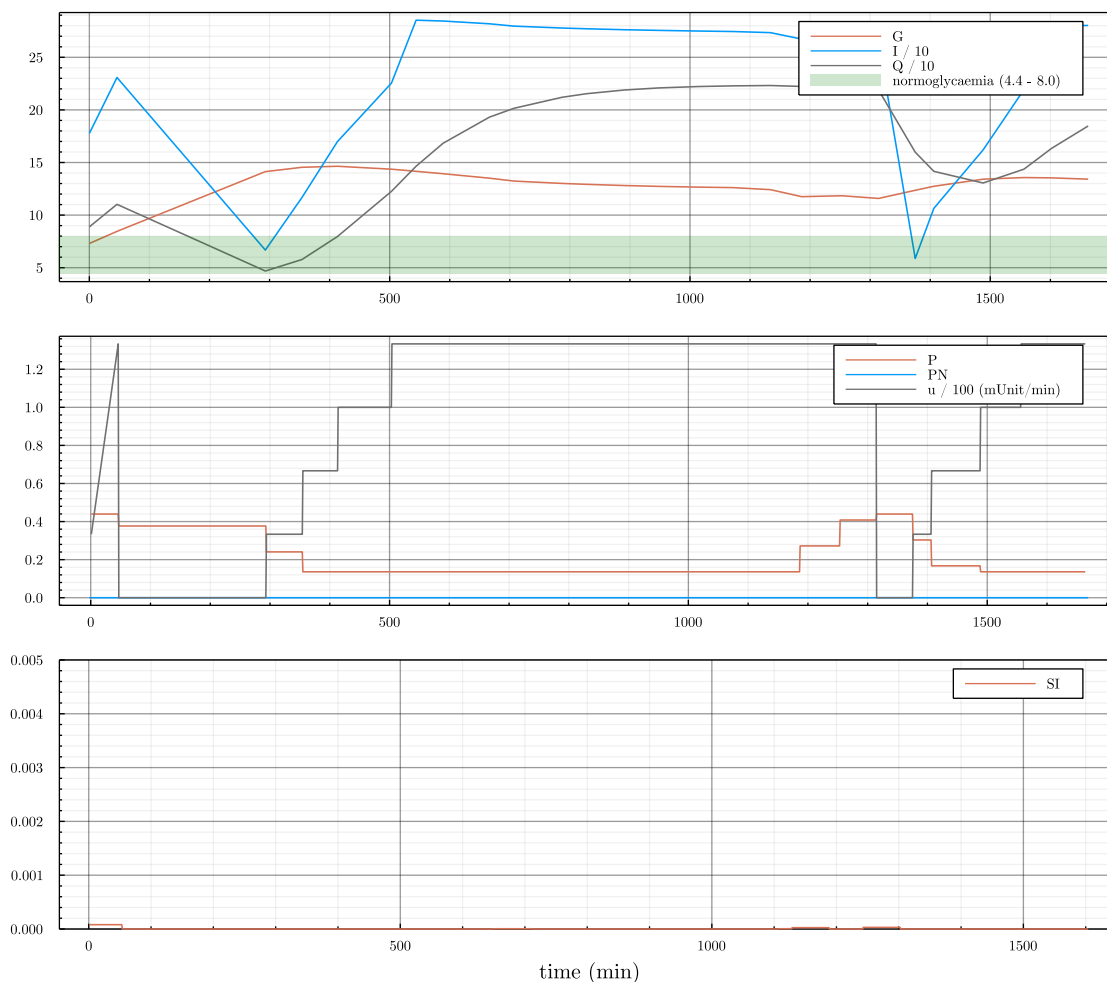
A klinikailag mért és az *ICING2* modell által jósolt értékek közti eltérés fakadhat a modell korlátjaiból is. A modell sok paramétere konstans érték, pedig a valóságban nyilvánvalóan semmi sem állandó. Éppen ezért kevésbé érzékeny, alacsony inzulinszenzitivitású (SI) betegeknél előfordulhat, hogy a kiszámolt SI negatív lenne, így

ebben az esetben egy nagyon kicsi, de pozitív értéket ad vissza a modell. Ez egy ésszerű kompromisszum a modell működésében, hiszen nem lehet minden paramétert pontosan meghatározni, viszont ennek ellenére szemmel láthatóan csak az esetek csekély részében okoz problémát (25. ábra, 26. ábra).



**25. ábra: ICING2 modell korlátjai (vércukorszintek)**

418675c8-5383-4b5c-87e5-2322cbca5c86



26. ábra: ICING2 modell korlátjai (teljes terápiás adatok)

A kisebb-nagyobb eltérések figyelembevételével, biztosan állítható, hogy a modell a tesztesetek egészére nézve kiválóan teljesít, hiszen még, ahol nagyobbak is az eltérések, ott is megegyezik a görbéknek dinamikája.

### 3.4 Eredmények értékelése

Jelen fejezetben a *Julia* nyelven elkészült környezetet értékelem mérnöki és orvosi szempontok szerint.

#### 3.4.1 Elkészült rendszer értékelése mérnöki szempontból

A *Julia* nyelvet tudományos számításokra optimalizálták. A tervezett rendszernek a két fő feladata a differenciálegyenletek megoldása és a számított nagy adathalmazon számolt statisztika, ezért ez nagy előnyt jelent a hatékonyságban és gyorsaságban.



Annak ellenére, hogy a *Julia* nyelv nem támogatja az objektumorientált programozást, objektumorientált elvek szerint létrehoztam egy interfészt a *Julia* és *Java* között, tehát a *Java* hívások jól le vannak választva a program többi részéről, így téve könnyen bővíthetővé, illetve átláthatóbbá a kódot.

A *Julia* alapvetően nem típusos nyelv, ezért nagy a kockázata a futás idejű hibáknak. Ezt azzal küszöböltem ki, hogy ahol ésszerű ott a változóknak, paramétereknek feltüntettem az elvárt típusát is.

A rendszer egyes fázisainak bemenetét szolgáló adatok különböző kiterjesztésű fájlok formájában vannak tárolva (*Java* bináris, *Mat*, *Jld2*). Mérnöki szempontból különösen hasznos volt egy deszerializációs interfész megvalósítása. Ez azt jelenti, hogy azon az absztrakciós szinten, ahol meghívjuk ezt az interfészt, ott nem kell figyelni arra, hogy épp milyen típusú fájlból történik az olvasás.

Sikerült a rendszerbe integrálni három különböző szimulációs kontrollert. A rendszert úgy alakítottam ki, hogy könnyen bővíthető legyen további kontrollerekkel, hiszen mindegyiknek ugyanolyan jól meghatározott bemenete és kimenete van.

A rendszer fejlesztése és tesztelése a kezdetektől fogva párhuzamosan zajlottak, így a munka végére egy műszaki és orvosi szempontok szerint egyaránt letesztelt program készülhetett el. Fejlesztés közben minden függvény alaposan tesztelve lett, ami a *unit* teszteknek felel meg. Integrációs tesztként az szolgált, amikor az egész folyamat egy páciensre lett vizsgálva, kiértékelve. Végül a validációs teszt során statisztikai kiértékelés és grafikus ábrázolás készült az egész betegállományra.

### **3.4.2 Elkészült rendszer értékelése orvosi szempontból**

Intenzív osztályon ápolt betegek 35-45 százaléka inzulinterápiára szorul, mert a metabolikus rendszerük egyensúlya felborult. Általában a metabolikus rendszer zavarának a magas vércukorszint a tünete. Hosszú ideig tartó magas vércukorszint káros következményekkel járhat a beteg kezelésének kimenetelére, ezért alkalmazzák az inzulinterápiát. Az orvosok számára a kihívást, a megfelelő inzulinadagolás meghatározása jelenti, mivel az ember vércukorszintjét befolyásoló élettani folyamatok igen bonyolultak, emiatt az inzulinbevitel hatása nehezen kiszámítható, hatása képlettel, pontos kiszámítása bonyolult.

Ezen orvosi probléma megoldására szolgálnak a szoros vércukorszabályozó protokollok, amelyek a szakszemélyzet számára támpontot adnak a megfelelő inzulinadagolás kiválasztására. Kiemelkedően fontos, hogy ezek a vércukorszabályozó protokollok minden esetben megfelelően és megbízhatóan működjenek. Ehhez azonban még a klinikai tesztek előtt, *in-silico* szimulációs környezetben validálni kell a protokollok működését. Ezt a validációs folyamatot hivatott támogatni és megkönnyíteni a megalkotott környezet.

Az orvosok számára rendkívül hasznos ez a szoftver, hiszen könnyedén lehet vele különböző kezeléseket, különböző betegcsoportokon szimulálni. Továbbá a kezelések eredményét könnyen áttekinthető módon lehet összehasonlítani egymással különböző célzatú grafikonok segítségével. Az előbbieken kívül, a keretrendszer képes különböző betegcsoportok kezeléseiről statisztikai kiértékelést végezni, amely eddig nem látott részleteket fedhet fel az orvos számára az adott terápia minőségéről.

## 4 Összefoglalás

A szakdolgozatom keretében sikerült megterveznem és kifejlesztenem egy olyan rendszert, amely *in-silico* szimulációt valósít meg. A rendszer képes a szimulációs eredmények statisztikai kiértékelésére és grafikonon ábrázolni a szimulált adatsorokat. A jobb felhasználói élmény érdekében a kötelező elemeken túl, még egy grafikus felhasználói felületet is kialakítottam.

A szakdolgozat elkészítése során megterveztem, implementáltam és verifikáltam a rendszert.

A munka során számos technikai kihívást oldottam meg, melyeket a következő eredmények, illetve funkciók foglalnak össze:

- A klinikai adatok beolvasása a virtuális betegek létrehozása céljából;
- A *STAR* protokoll *Java* függvényhívásai *Julia* kódból;
- A szimulált vércukormérést megvalósító differenciálegyenlet rendszer implementálása;
- A három különböző protokollt megvalósító kontroller integrálása a rendszerbe;
- Statisztikai számítások implementációja;
- Szimuláció vizualizációja;
- Az elkészült funkciók tesztelése, validációja;
- Felhasználói felület kialakítása.

Az elvégzett munka előremutató ezen a területen, hiszen az elkészült rendszer könnyen használható kutatási célokra. Új vagy módosított vércukorszabályozó protokollokat, illetve az emberi szervezetet leíró metabolikus modelleket lehet tesztelni, kipróbálni a megvalósított keretrendszerrel.

A bemutatott eredmények alapján megállapítható, hogy orvosi területen és mérnöki vonatkozásban is értékes és jelentős eredményeket sikerült elérnem és a szakdolgozat minden célkitűzését maradéktalanul teljesítettem.

### 4.1 További fejlesztési lehetőségek

Az eddigiekben bemutatott keretrendszer továbbfejlesztésére számos lehetőség van mérnöki és orvosi szempontok szerint egyaránt.

A fejlesztés késői szakaszában felhasználói felület is készült a rendszernek. Mivel ez nem tervezett funkció volt, ezért a programnak alapvetően nem volt szükséges többszálon futnia. Viszont a *GUI* megjelenése miatt már szükségessé válik a többszálúság, tehát további fejlesztési feladat lehet az, hogy a főszálon fut a *GUI* és mindig egy háttérszál dolgozik az aktuális feladaton.

Egy további mérnöki feladat természetesen a meglévő program refaktorálása.

Orvosi szempontból további fejlesztési feladat lehet a meglévő statisztikai kiértékelő módszerek finomítása, illetve bővítése.

Az inzulinszenzitivitás becslésére, ami kritikus fontosságú a vércukorszint predikciójához, a jelenlegi *STAR* protokoll sztochasztikus módszereket használ, a jövőben ezeket kívánjuk helyettesíteni különböző mesterséges intelligencia megoldások integrálásával. Ezen kutatások megvalósításában az elkészített *Julia* környezet nagy segítséget fog jelenteni.

## Irodalomjegyzék

- [1] J. L. Dickson, K. W. Stewart, C. G. Pretty, M. Flechet, T. Desaive, S. Penning és B. C. Lambermont, „Generalisability of a Virtual Trials Method Glycaemic Control in Intensive Care,” *IEEE Transaction on Biomedical Engineering*, pp. 65: 7 pp. 1534-1553, 11 p., 2016.
- [2] F. M. Liam, A. J. Le Compte, G. M. Shaw, S. Penning, T. Desaive és G. J. Chase, „STAR Development and Protocol Comparison,” *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, pp. vol. 59, no. 12, pp. 3357-3364, 2012.
- [3] A. Evans, A. Le Compte, C.-S. Tan, L. Ward, J. Steel, C. G. Pretty, S. Penning, F. Suhaimi, G. M. Shaw, T. Desaive és G. J. Chase, „Stochastic Targeted (STAR) Glycemic Control: Design, Safety, and Performance,” *Journal of Diabetes Science and Technology*, pp. 6(1):102-15, 2012.
- [4] L. M. Fisk, A. J. Le Compte, G. M. Shaw, S. Penning, T. Desaive and J. G. Chase, "STAR Development and Protocol Comparison," in *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 12, pp. 3357-3364, Dec. 2012, doi: 10.1109/TBME.2012.2214384.
- [5] *Julia: A Fast Dynamic Language for Technical Computing*. Jeff Bezanson, Stefan Karpinski, Viral B. Shah, Alan Edelman. (2012).
- [6] Call Java programs from Julia, <https://juliainterop.github.io/JavaCall.jl/> (utolsó látogatás ideje 2021. 05. 16.)
- [7] Jessica Lin, Normy N. Razak, Christopher G. Pretty, Aaron Le Compte, Paul Docherty, Jacquelyn D. Parente, Geoffrey M. Shaw, Christopher E. Hann, J. Geoffrey Chase, „A physiological Intensive Control Insulin-Nutrition-Glucose (ICING) model validated in critically ill patients”, *Computer Methods and Programs in Biomedicine*, Volume 102, Issue 2, 2011.

# Függelék

Num Episodes	112
Total Hours	11159.0
Num BG measurements	5967
Average time of hours analysed (Days)	4.19
Median time of hours analysed [IQR] (Days)	[1.76 2.83 5.09]
Mean Measures/day (Cohort)	12.71
Median [IQR] Measures/day (Per-Patient)	[10.45 12.62 15.31]

**2. táblázat: STAR Whole Cohort Statistics**

BG median [IQR] (mmol/L)	[4.99 5.64 6.67]
BG mean (geometric) (mmol/L)	6.09
BG StDev (geometric) (mmol/L)	1.79
Num episodes < 4.0 mmol/L	47
Num episodes < 2.22 mmol/L	2
% BG < 2.2 mmol/L	0.03
% BG < 4.0 mmol/L	2.82
% BG < 4.4 mmol/L	6.18
% BG within 4.4 - 6.5 mmol/L	66.48
% BG within 4.4 - 7.0 mmol/L	73.3
% BG within 4.4 - 8.0 mmol/L	83.02
% BG within 8.0 - 10 mmol/L	6.85
% BG within > 10 mmol/L	3.87

**3. táblázat: STAR Raw BG stats**

BG median [IQR] (mmol/L)	[5.19 5.74 6.43]
BG mean (geometric) (mmol/L)	6
BG StDev (geometric) (mmol/L)	1.4
% BG < 2.2 mmol/L	0.02
% BG < 4.0 mmol/L	1.62
% BG < 4.4 mmol/L	3.72
% BG within 4.4 - 6.1 mmol/L	61.23
% BG within 4.4 - 7 mmol/L	81.7
% BG within 4.4 - 8 mmol/L	89.92
% BG within 4.4 - 9 mmol/L	92.73
% BG within 6.0 - 9 mmol/L	35.14
% BG within 8.0 - 10 mmol/L	4.23
% BG within > 10 mmol/L	2.09

**4. táblázat: STAR Hourly Resampled BG stats**

Hours of control	[9.25 25.5 40.25]
Num BG measurements	[22.0 36.0 71.0]
Initial BG (mmol/L)	[7.2 9.6 11.3]
BG median (mmol/L)	[5.39 5.93 6.66]
BG mean (mmol/L)	[5.56 6.22 6.96]
BG StDev (mmol/L)	[1.0 1.32 1.88]
%BG > 10.0 mmol/L	[0.0 1.05 8.81]
%BG within 4.0-6.1 mmol/L	[33.47 54.17 74.6]
%BG within 4.0-7.0 mmol/L	[54.44 76.03 87.5]
%BG within 4.0-8.0 mmol/L	[74.01 87.21 94.0]
%BG < 4.4 mmol/L	[0.0 3.45 8.94]
%BG < 4.0 mmol/L	[0.0 0.0 3.28]
%BG < 2.22 mmol/L	[0.0 0.0 0.0]

**5. táblázat: STAR Per-episode statistics (Median [IQR])**

Median insulin rate [IQR] (U/hr)	[0.0 2.0 6.0]
Feed Stats All	
Median glucose rate [IQR] (g/hour)	[0.0 1.76 3.74]
Median glucose rate [IQR] (% goal)	[58.12 97.96 99.96]
Median Enteral glucose [IQR] (g/hour)	[0.0 1.76 3.74]
Median Parental glucose [IQR] (g/hour)	[0.0 0.0 0.0]
Feed Stats Excluding those not fed	
Total hours not fed	4225
Median glucose rate [IQR] (g/hour)	[2.22 3.65 5.42]
Median glucose rate [IQR] (% goal)	[70.21 98.9 100.0]
Median Enteral glucose [IQR] (g/hour)	[2.05 3.65 5.34]
Median Parental glucose [IQR] (g/hour)	[5.0 5.0 5.0]

**6. táblázat: STAR Intervention Cohort Stats (Hourly Average)**

Median insulin rate [IQR] (U/hr)	[2.35 4.16 6.37]
Feed Stats All	
Median glucose rate [IQR] (g/hour)	[0.49 2.89 5.1]
Median glucose rate [IQR] (% goal)	[61.24 78.61 93.11]
Median Enteral glucose [IQR] (g/hour)	[0.0 2.87 5.1]
Median Parental glucose [IQR] (g/hour)	[0.0 0.0 0.0]
Feed Stats Excluding those not fed	
Total hours not fed	4225
Median glucose rate [IQR] (g/hour)	[3.0 4.53 5.65]
Median glucose rate [IQR] (% goal)	[71.39 87.05 94.96]
Median Enteral glucose [IQR] (g/hour)	[2.93 4.36 5.77]
Median Parental glucose [IQR] (g/hour)	[5.0 5.0 5.0]

**7. táblázat: STAR Intervention Per-episode Stats (Hourly Average)**

Num Episodes	112
Total Hours	11220.0
Num BG measurements	3852
Average time of hours analysed (Days)	4.22
Median time of hours analysed [IQR] (Days)	[1.76 2.79 5.1]
Mean Measures/day (Cohort)	8.16
Median [IQR] Measures/day (Per-Patient)	[8.13 8.24 8.38]

**8. táblázat: SIMPLE Whole Cohort Statistics**

BG median [IQR] (mmol/L)	[6.42 7.24 8.23]
BG mean (geometric) (mmol/L)	7.45
BG StDev (geometric) (mmol/L)	1.76
Num episodes < 4.0 mmol/L	33
Num episodes < 2.22 mmol/L	1
% BG < 2.2 mmol/L	0.03
% BG < 4.0 mmol/L	1.43
% BG < 4.4 mmol/L	2.18
% BG within 4.4 - 6.5 mmol/L	24.97
% BG within 4.4 - 7.0 mmol/L	39.8
% BG within 4.4 - 8.0 mmol/L	68.02
% BG within 8.0 - 10 mmol/L	22.72
% BG within > 10 mmol/L	6.98

**9. táblázat: SIMPLE Raw BG stats**

BG median [IQR] (mmol/L)	[6.43 7.21 8.19]
BG mean (geometric) (mmol/L)	7.41
BG StDev (geometric) (mmol/L)	1.69
% BG < 2.2 mmol/L	0.04
% BG < 4.0 mmol/L	1.31
% BG < 4.4 mmol/L	2.06
% BG within 4.4 - 6.1 mmol/L	15.29

% BG within 4.4 - 7 mmol/L	41.02
% BG within 4.4 - 8 mmol/L	68.78
% BG within 4.4 - 9 mmol/L	84.83
% BG within 6.0 - 9 mmol/L	71.28
% BG within 8.0 - 10 mmol/L	22.63
% BG within > 10 mmol/L	6.5

**10. táblázat: SIMPLE Hourly Resampled BG stats**

Hours of control	[15.0 30.0 45.0]
Num BG measurements	[14.75 23.0 41.5]
Initial BG (mmol/L)	[7.2 9.6 11.3]
BG median (mmol/L)	[6.79 7.28 8.02]
BG mean (mmol/L)	[6.9 7.48 8.07]
BG StDev (mmol/L)	[1.11 1.36 1.82]
%BG > 10.0 mmol/L	[0.0 4.0 14.46]
%BG within 4.0-6.1 mmol/L	[2.99 12.02 25.98]
%BG within 4.0-7.0 mmol/L	[21.66 37.5 55.64]
%BG within 4.0-8.0 mmol/L	[49.82 68.69 83.33]
%BG < 4.4 mmol/L	[0.0 0.0 2.78]
%BG < 4.0 mmol/L	[0.0 0.0 1.58]
%BG < 2.22 mmol/L	[0.0 0.0 0.0]

**11. táblázat: SIMPLE Per-episode statistics (Median [IQR])**

Median insulin rate [IQR] (U/hr)	[1.0 1.0 2.0]
Feed Stats All	
Median glucose rate [IQR] (g/hour)	[4.32 4.32 4.32]
Median glucose rate [IQR] (% goal)	[68.12 113.3 133.58]
Median Enteral glucose [IQR] (g/hour)	[4.32 4.32 4.32]
Median Parental glucose [IQR] (g/hour)	[0.0 0.0 0.0]
Feed Stats Excluding those not fed	
Total hours not fed	28
Median glucose rate [IQR] (g/hour)	[4.32 4.32 4.32]
Median glucose rate [IQR] (% goal)	[68.12 113.3 133.58]
Median Enteral glucose [IQR] (g/hour)	[4.32 4.32 4.32]
Median Parental glucose [IQR] (g/hour)	NaN

**12. táblázat: SIMPLE Intervention Cohort Stats (Hourly Average)**

Median insulin rate [IQR] (U/hr)	[1.38 1.76 2.2]
Feed Stats All	
Median glucose rate [IQR] (g/hour)	[4.29 4.32 4.36]
Median glucose rate [IQR] (% goal)	[58.85 91.08 122.69]
Median Enteral glucose [IQR] (g/hour)	[4.29 4.32 4.36]
Median Parental glucose [IQR] (g/hour)	[0.0 0.0 0.0]
Feed Stats Excluding those not fed	
Total hours not fed	28
Median glucose rate [IQR] (g/hour)	[4.32 4.32 4.36]
Median glucose rate [IQR] (% goal)	[58.85 91.08 122.69]
Median Enteral glucose [IQR] (g/hour)	[4.32 4.32 4.36]
Median Parental glucose [IQR] (g/hour)	NaN

**13. táblázat: SIMPLE Intervention Per-episode Stats (Hourly Average)**

Num Episodes	112
Total Hours	11142.98
Num BG measurements	5799
Average time of hours analysed (Days)	4.17
Median time of hours analysed [IQR] (Days)	[1.69 2.79 5.08]
Mean Measures/day (Cohort)	12.42
Median [IQR] Measures/day (Per-Patient)	[11.27 12.88 15.01]



**14. táblázat: HISTORIC Whole Cohort Statistics**

BG median [IQR] (mmol/L)	[6.07 7.03 8.35]
BG mean (geometric) (mmol/L)	7.41
BG StDev (geometric) (mmol/L)	2.09
Num episodes < 4.0 mmol/L	34
Num episodes < 2.22 mmol/L	5
% BG < 2.2 mmol/L	0.09
% BG < 4.0 mmol/L	1.41
% BG < 4.4 mmol/L	2.9
% BG within 4.4 - 6.5 mmol/L	33.16
% BG within 4.4 - 7.0 mmol/L	46.34
% BG within 4.4 - 8.0 mmol/L	66.96
% BG within 8.0 - 10 mmol/L	20.06
% BG within > 10 mmol/L	10.02

**15. táblázat: HISTORIC Raw BG stats**

BG median [IQR] (mmol/L)	[6.04 6.75 7.76]
BG mean (geometric) (mmol/L)	7.09
BG StDev (geometric) (mmol/L)	1.72
% BG < 2.2 mmol/L	0.03
% BG < 4.0 mmol/L	0.84
% BG < 4.4 mmol/L	1.7
% BG within 4.4 - 6.1 mmol/L	25.01
% BG within 4.4 - 7 mmol/L	56.21
% BG within 4.4 - 8 mmol/L	77.07
% BG within 4.4 - 9 mmol/L	87.35
% BG within 6.0 - 9 mmol/L	65.32
% BG within 8.0 - 10 mmol/L	15.26
% BG within > 10 mmol/L	5.94

**16. táblázat: HISTORIC Hourly Resampled BG stats**

Hours of control	[10.13 19.68 37.45]
Num BG measurements	[21.0 39.5 68.25]
Initial BG (mmol/L)	[7.2 9.6 11.3]
BG median (mmol/L)	[6.48 6.87 7.68]
BG mean (mmol/L)	[6.74 7.15 8.06]
BG StDev (mmol/L)	[1.28 1.65 2.12]
%BG > 10.0 mmol/L	[1.93 8.33 17.9]
%BG within 4.0-6.1 mmol/L	[11.71 24.62 36.67]
%BG within 4.0-7.0 mmol/L	[33.33 50.4 62.5]
%BG within 4.0-8.0 mmol/L	[53.27 71.98 82.61]
%BG < 4.4 mmol/L	[0.0 0.0 4.32]
%BG < 4.0 mmol/L	[0.0 0.0 1.63]
%BG < 2.22 mmol/L	[0.0 0.0 0.0]

**17. táblázat: HISTORIC Per-episode statistics (Median [IQR])**

Median insulin rate [IQR] (U/hr)	[2.0 4.0 8.0]
Feed Stats All	
Median glucose rate [IQR] (g/hour)	[1.73 4.06 6.76]
Median glucose rate [IQR] (% goal)	[58.53 100.93 157.12]
Median Enteral glucose [IQR] (g/hour)	[1.13 3.52 6.55]
Median Parental glucose [IQR] (g/hour)	[0.17 0.39 0.61]
Feed Stats Excluding those not fed	
Total hours not fed	9
Median glucose rate [IQR] (g/hour)	[1.74 4.06 6.77]
Median glucose rate [IQR] (% goal)	[59.08 101.19 157.12]
Median Enteral glucose [IQR] (g/hour)	[2.95 4.75 7.39]
Median Parental glucose [IQR] (g/hour)	[0.3 0.45 0.71]

**18. táblázat: HISTORIC Intervention Cohort Stats (Hourly Average)**

Median insulin rate [IQR] (U/hr)	[3.41 4.74 7.07]
Feed Stats All	
Median glucose rate [IQR] (g/hour)	[2.37 4.39 6.0]
Median glucose rate [IQR] (% goal)	[63.17 82.92 119.95]
Median Enteral glucose [IQR] (g/hour)	[1.73 4.06 5.77]
Median Parental glucose [IQR] (g/hour)	[0.18 0.43 0.66]
Feed Stats Excluding those not fed	
Total hours not fed	9
Median glucose rate [IQR] (g/hour)	[2.37 4.39 6.0]
Median glucose rate [IQR] (% goal)	[63.17 82.92 119.95]
Median Enteral glucose [IQR] (g/hour)	[3.36 4.9 6.37]
Median Parental glucose [IQR] (g/hour)	[0.4 0.59 0.78]

**19. táblázat: HISTORIC Intervention Per-episode Stats (Hourly Average)**

Num Episodes	112
Total Hours	11158.93
Num BG measurements	5819
Average time of hours analysed (Days)	2.16
Median time of hours analysed [IQR] (Days)	[0.88 1.65 2.84]
Mean Measures/day (Cohort)	24.0
Median [IQR] Measures/day (Per-Patient)	[24.0 24.0 24.0]

**20. táblázat: Original Whole Cohort Statistics**

BG median [IQR] (mmol/L)	[6.0 7.0 8.4]
BG mean (geometric) (mmol/L)	7.48
BG StDev (geometric) (mmol/L)	2.35
Num episodes < 4.0 mmol/L	43
Num episodes < 2.22 mmol/L	2
% BG < 2.2 mmol/L	0.03
% BG < 4.0 mmol/L	1.98
% BG < 4.4 mmol/L	4.07
% BG within 4.4 - 6.5 mmol/L	30.73
% BG within 4.4 - 7.0 mmol/L	43.96
% BG within 4.4 - 8.0 mmol/L	64.38
% BG within 8.0 - 10 mmol/L	19.44
% BG within > 10 mmol/L	10.47

**21. táblázat: Original Raw BG stats**

BG median [IQR] (mmol/L)	[0.0 0.0 0.0]
BG mean (geometric) (mmol/L)	0.0
BG StDev (geometric) (mmol/L)	0.0
% BG < 2.2 mmol/L	100.0
% BG < 4.0 mmol/L	100.0
% BG < 4.4 mmol/L	100.0
% BG within 4.4 - 6.1 mmol/L	0.0
% BG within 4.4 - 7 mmol/L	0.0
% BG within 4.4 - 8 mmol/L	0.0
% BG within 4.4 - 9 mmol/L	0.0
% BG within 6.0 - 9 mmol/L	0.0
% BG within 8.0 - 10 mmol/L	0.0
% BG within > 10 mmol/L	0.0

**22. táblázat: Original Hourly Resampled BG stats**

Hours of control	[10.13 19.68 37.45]
Num BG measurements	[21.0 39.5 68.25]
Initial BG (mmol/L)	[7.2 9.6 11.3]

BG median (mmol/L)	[6.5 6.9 7.71]
BG mean (mmol/L)	[6.8 7.15 8.06]
BG StDev (mmol/L)	[1.34 1.74 2.35]
%BG > 10.0 mmol/L	[2.08 8.41 18.55]
%BG within 4.0-6.1 mmol/L	[13.49 24.14 33.33]
%BG within 4.0-7.0 mmol/L	[33.33 48.05 58.94]
%BG within 4.0-8.0 mmol/L	[52.35 69.57 79.96]
%BG < 4.4 mmol/L	[0.0 1.2 5.41]
%BG < 4.0 mmol/L	[0.0 0.0 3.23]
%BG < 2.22 mmol/L	[0.0 0.0 0.0]

**23. táblázat: Original Per-episode statistics (Median [IQR])**

Median insulin rate [IQR] (U/hr)	[2.0 4.0 8.0]
Feed Stats All	
Median glucose rate [IQR] (g/hour)	[1.73 4.06 6.76]
Median glucose rate [IQR] (% goal)	[58.53 100.93 157.12]
Median Enteral glucose [IQR] (g/hour)	[1.13 3.52 6.55]
Median Parental glucose [IQR] (g/hour)	[0.17 0.39 0.61]
Feed Stats Excluding those not fed	
Total hours not fed	9
Median glucose rate [IQR] (g/hour)	[1.74 4.06 6.77]
Median glucose rate [IQR] (% goal)	[59.08 101.19 157.12]
Median Enteral glucose [IQR] (g/hour)	[2.95 4.75 7.39]
Median Parental glucose [IQR] (g/hour)	[0.3 0.45 0.71]

**24. táblázat: Original Intervention Cohort Stats (Hourly Average)**

Median insulin rate [IQR] (U/hr)	[3.41 4.74 7.07]
Feed Stats All	
Median glucose rate [IQR] (g/hour)	[2.37 4.39 6.0]
Median glucose rate [IQR] (% goal)	[63.17 82.92 119.95]
Median Enteral glucose [IQR] (g/hour)	[1.73 4.06 5.77]
Median Parental glucose [IQR] (g/hour)	[0.18 0.43 0.66]
Feed Stats Excluding those not fed	
Total hours not fed	9
Median glucose rate [IQR] (g/hour)	[2.37 4.39 6.0]
Median glucose rate [IQR] (% goal)	[63.17 82.92 119.95]
Median Enteral glucose [IQR] (g/hour)	[3.36 4.9 6.37]
Median Parental glucose [IQR] (g/hour)	[0.4 0.59 0.78]

**25. táblázat: Original Intervention Per-episode Stats (Hourly Average)**