POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

Data-driven nonlinear control of glucose-insulin dynamics in patients affected by Type I Diabetes

Supervisors

Prof. Diego REGRUTO

Candidate

Alice GIORDANO

July 2024

You can't control the wind,
but you can adjust your sails.

Abstract

Diabetes is one of the top ten causes of death in adults and, in the recent decades, became a global health problem. People with T1DM suffer from a metabolic disorder characterized by pancreas's inability to produce a sufficient amount of insulin, the peptide hormone that plays a vital role in regulating the way cells absorb glucose and use it as an energy source. As a result, these individuals need to constantly control their blood glucose levels through insulin administration. In recent years, significant technological progress in continuous glucose monitoring and insulin delivery systems has enabled researchers to develop automated methods for the management of diabetes, often referred to as the Artificial Pancreas. The development of control algorithms for this purpose is a highly active field of research. While traditional control approaches have been the primary focus up until now, machine learning (in particular neural networks) seems to offer a promising alternative framework that has not yet been thoroughly explored. This thesis focuses on data-driven approaches, aiming to design controllers, both linear and nonlinear, starting from the input-output data, collected through simulation of an accurate diabetic patient simulator available in the literature, bypassing the modeling step. The work is divided into two main sections. The first one revolves around the development of two linear controllers employing the Least squares and the Set-membership methods. The second, instead, concerns the design of a nonlinear controller taking advantage of recurrent neural networks. Through extensive testing and validation, the results demonstrate that, while linear controllers perform adequately in maintaining safe glucose levels, the nonlinear controller significantly outperforms them. The neural network-based approach provides more accurate and responsive insulin delivery, offering a superior and more robust solution for AP systems. These findings highlight the potential for advanced neural network algorithms to become a new milestone for diabetes management through improved automated control mechanisms.

Table of Contents

List of Tables VII

List of Figures VIII

Acronyms XI

1 Introduction 1
- 1.1 Research questions and objectives . . . 1
  - 1.1.1 General description of the project . . . 2
- 1.2 Structure of the thesis . . . 3

2 Background 4
- 2.1 Diabetes . . . 4
  - 2.1.1 Type I Diabetes . . . 5
  - 2.1.2 Type II Diabetes . . . 5
- 2.2 Artificial Pancreas Systems . . . 7
- 2.3 Mathematical models of the glucose-insulin system . . . 11
  - 2.3.1 Sorensen model . . . 11
  - 2.3.2 Hovorka model . . . 13
  - 2.3.3 UVAPadova model . . . 13
- 2.4 Comparison among three models . . . 14

3 Literature review 15
- 3.1 Problem's description . . . 15
- 3.2 An overview of Control Systems . . . 16
  - 3.2.1 Modern Control Theory . . . 18
  - 3.2.2 Data-Driven Control . . . 19
- 3.3 An overview of Machine Learning . . . 22
  - 3.3.1 Neural Networks . . . 24
  - 3.3.2 Recurrent Neural Network . . . 27
- 3.4 State of Art of the Artificial Pancreas . . . 30

4 Methodology 34
4.1 Linear approaches . . . 34
4.1.1 Least Square . . . 39
4.1.2 Set-membership . . . 40
4.2 Nonlinear Neural Networks approach . . . 42

5 Development and experiments 44
5.1 Linear approaches . . . 44
5.1.1 Least Square . . . 46
5.1.2 Set Membership . . . 52
5.2 Nonlinear Neural Networks approach . . . 55

6 Results' analysis 63
6.1 Linear approaches . . . 63
6.1.1 LS: Pros and Cons . . . 63
6.1.2 SM: Pros and Cons . . . 64
6.1.3 Comparison among LS and SM . . . 65
6.1.4 The overfitting problem . . . 66
6.2 Nonlinear Neural Networks approach . . . 67
6.2.1 NNOE: Pros and Cons . . . 67
6.2.2 Comparison among linear and nonlinear methods . . . 68
6.2.3 The overfitting problem . . . 70

7 Conclusions and Future Improvements 71
7.1 Conclusions . . . 71
7.2 Future Improvements . . . 72

A Sorensen model equations 73

Bibliography 75

List of Tables

5.1 Least Square tests' metrics 50
5.2 Set Membership tests' metrics 54

List of Figures

2.1 Artificial Pancreas control system scheme 7
2.2 Compartment diagram of the Sorensen model 12

3.1 Open-loop control system block diagram 17
3.2 Closed-loop control system block diagram 17
3.3 Feedback control system to be designed compared with the reference model 19
3.4 Direct data-driven 20
3.5 Indirect data-driven 21
3.6 Supervised, unsupervised and reinforcement learning 23
3.7 Architectures of Neural Networks 24
3.8 Activation functions 25
3.9 Overfitting and underfitting 26
3.10 Architecture of a single hidden layer RNN 27
3.11 Timeline of the development of the AP 30
3.12 PID control algorithm 31
3.13 MPC control algorithm 32
3.14 FLC control algorithm 32
3.15 Single hormone closed-loop system 33
3.16 Dual hormone closed-loop systems 33

4.1 Simulink scheme of the system where data are taken 35
4.2 Input u(t) 35
4.3 Output y(t) 35
4.4 Reference model step response 37
4.5 Delayed reference model step response 37
4.6 $G_{empt}(t)$ 38
4.7 $\tau_{pasto}(t)$ 38
4.8 System 1 42
4.9 System 2 42

5.1 Simulink scheme of the system 45
5.2 LS n = 2 and Delta_eta = 0 49
5.3 LS new meal model 50
5.4 SM n = 2 and Delta_eta = 0 53
5.5 SM new meal model 54
5.6 Simulink scheme of the neural network system 55
5.7 Simulink scheme of the neural network 55
5.8 Simulink scheme of the system where data were taken from 57
5.9 Input of the controller, *out_u* 57
5.10 Output of the controller, *out_y* 57
5.11 Simulink scheme of the system where data were taken from 59
5.12 Output of the controller, *uscita_controllore* 59
5.13 Output of the system, *uscita_sys* 59
5.14 First experiment plot 60
5.15 Second experiment plot 61
5.16 Validation of the neural network controller 62

6.1 LS with Delta_eta = 0.1 65
6.2 SM with Delta_eta = 0.1 65
6.3 LS with Delta_eta = 2 65
6.4 SM with Delta_eta = 2 65
6.5 Least Square controller 68
6.6 Set-Membership controller 69
6.7 Neural Network controller 69
6.8 Best RNN controller 70
6.9 Validation of the neural network controller 70

Acronyms

AI
Artificial Intelligence

AID
Automated Insulin Delivery

AP
Artificial Pancreas

BG
Blood Glucose

BPTT
Back Propagation Through Time

CGM
Continuous Glucose Monitoring

CT
Continuous Time

DDC
Data-Driven Control

DDDC
Direct Data-Driven Control

FLC
Fuzzy Logic Control

HCL

Hybrid Closed-Loop

IDDC

Indirect Data-Driven Control

LADA

Latent Autoimmune Diabetes of the Adult

LTI

Linear Time Invariant

MBC

Model-Based Control

ML

Machine Learning

MPC

Model Predictive Control

NN

Neural Network

NNOE

Non-Linear Output Error

OE

Output Error

PID

Proportional Integral Derivative

RMSE

Root Mean Square Error

RNN

Recurrent Neural Network

SM

Set-Membership

T1DM

Type 1 Diabetes Mellitus

Chapter 1

Introduction

In the following chapter, an introduction of the work will be presented, focusing on research questions and objectives, supplying a general description of the project. Moreover, a brief overview of the structure of the thesis will be provided.

1.1 Research questions and objectives

This thesis aims to investigate a novel framework for the development of data-driven based control algorithms for the *Artificial Pancreas* system. Specifically, the objective of the work is to develop a nonlinear control algorithm by employing recurrent neural networks. The research begins with an analysis of diabetes, examining its different types and their consequences, and introduces the concept of the Artificial Pancreas as a potential tool for managing the disease. It is represented as a closed-loop system that automatically controls glycemia by infusing an adequate amount of insulin, according to the measured glucose level. The primary purpose of this device is, thus, to preserve safe and healthy BG levels, preventing episodes of hypoglycemia or hyperglycemia. Despite the AP_{sys} showing incredible promises for treatment of T1DM, they currently face limitations: they are capable of dosing insulin to decrease blood glucose levels, but unable to counteract the effects of a too high dose. Additionally, they are not completely "hands off": in fact, the amount of insulin to administrate has to be manually set by the patient.
Thus, the next step is to develop a fully closed-loop system capable of automatically determining the insulin quantity using a control algorithm. This thesis focuses on this context, proposing a new approach for the development of a control system able to govern the operation of the Artificial Pancreas, aiming to eliminate the need for human intervention. This is achieved through the use of recurrent neural networks: adapting to each patient, they would be able to predict the glucose-insulin dynamic, thereby autonomously managing insulin administration.

1.1.1 General description of the project

The development process that led to the satisfaction of the aforementioned purpose, started with the collection of input-output data (i.e., the insulin and blood glucose values) using a highly accurate diabetic patient simulator available in the literature. These data were subsequently employed to design two linear controllers, following, respectively, LS (Least Squares) and SM (Set-Membership) approaches, with the goal of managing the glucose-insulin dynamics of a patient affected by type 1 diabetes. This was done in order to perform a preliminary evaluation of the state of the art and to identify the weaknesses of these methodologies, to use them as foundations for the development of a neural networks nonlinear controller.
Both linear approaches were tested by integrating them into a closed-loop system that replicates the patient's behavior and validating their effectiveness in response to varying inputs (i.e., different values of insulin metabolized from meals).
To identify the limits of these two control strategies, additional controllers were developed by introducing progressively increasing measurement errors to the output data and conducting repeated tests. Performance evaluation of the two methods included the definition of a "safe range" within which the controlled system's output curve had to remain. Finally, after developing the nonlinear controller, the same experiments were replicated to verify the resolution of previously identified limitations and to validate the formulated hypothesis.

1.2 Structure of the thesis

The thesis work is composed of seven chapters, guiding through the process leading to the complete design, test and validation of the linear and nonlinear controllers:

- **Chapter 2**, *Background*, provides an overview of diabetes, its classification and introduces the concept of the Artificial Pancreas. It also presents the three main mathematical models of the glucose-insulin dynamics, which will be compared in the end.
- **Chapter 3**, *Literature Review*, focuses on the description of the problem from an engineering point of view. It introduces concepts of Control Systems and Machine Learning, that will serve as a basis for understanding the employed approaches and methodologies to address the proposed problem.
- **Chapter 4**, *Methodology*, presents the utilized approaches and the applied methodologies. In particular it will delve into the specific theory of the Linear and Nonlinear methods.
- **Chapter 5**, *Development and Experiments*, contains the development of the different algorithms, the experimental setup and experiments' results.
- **Chapter 6**, *Results' analysis*, aims to provide the pros and cons of the used methods, offering a comprehensive assessment of their effectiveness.
- **Chapter 7**, *Conclusions and Future Improvements*, summarizes the relevant findings and the contribution of this thesis work, providing a brief overview of the results and proposing some future improvements.

Chapter 2

Background

This chapter will provide a comprehensive examination of the diabetes (encompassing its classification and its consequences), the Artificial Pancreas system (its composition and its limitations) and, lastly, three different mathematical models of the glucose-insulin dynamic.

2.1 Diabetes

Diabetes is one of the top ten causes of death in adults and, in the last decades, became a global health problem.
As written in [1], the rate of diabetes in the world population reached 9% (463 million people) in 2019 and is projected to increase to 10.2% (578 million) by 2030. This chronic disease occurs when the immune system mistakenly attacks and destroys the beta cells of the pancreatic islets; thus, patients affected by diabetes mellitus suffer from a metabolic disorder characterized by pancreas inability to provide a sufficient amount of insulin, the peptide hormone that plays a vital role in regulating the way the cells absorb glucose and use it as an energy source. When this mechanism is altered, the glucose accumulated in the bloodstream turns into the direct cause of diabetes onset.
From a medical perspective, diabetes is a very complex disease as it represents the container of multiple syndromes and disorders; it is associated with problems affecting the eyes, kidneys, nerves, and blood vessels.

2.1.1 Type I Diabetes

Type I diabetes is of autoimmune origin, so it has no connection with eating habits. It affects about 10% of people with diabetes; it usually occurs in youth, but, in predisposed individuals, it can also manifest in older age, presenting itself in a variant called LADA (Latent Autoimmune Diabetes of the Adult), where the autoimmune attack on beta cells is slow and less pronounced, and the disease develops over the course of the years.
Type I diabetes is portrayed by insufficient or absent production of insulin [2].
The causes of type I diabetes are unknown, but it is characterized by the presence in the blood of antibodies directed against antigens positioned on the cells that produce the hormone. This damage could be related to environmental or genetic factors. Genetic predisposition to react against external agents, such as viruses and bacteria, could be involved. Therefore, type I diabetes is classified among autoimmune diseases, in which the immune system attacks the body itself. Type I diabetes involves acute reactions: the patient may develop diabetic ketoacidosis coma caused by the accumulation of ketones, leading to loss of consciousness, dehydration, and severe blood alterations.
People with type I diabetes need to receive lifelong insulin replacement therapy (through injections or by an insulin pump), in which the hormone is supplied externally to recreate the effect of naturally produced insulin, maintaining blood glucose (BG) levels within the "normal", euglycemic[1], range.
In order to assist patients in managing glucose levels, the *Artificial Pancreas* (AP) has been developed.

2.1.2 Type II Diabetes

Type II diabetes is the most frequent and widespread form of diabetes. It affects about 90% of diabetic patients and it is characterized by insulin resistance in the body, often associated with reduced insulin release. It begins with insulin resistance, leading to insulin deficiency (a toxic condition for pancreatic cells, which are destroyed).
It typically occurs in older age, and genetic factors (as the predisposition to reduced response of beta cells to glucose stimulus) and environmental factors (such as obesity, which leads to insulin resistance) are involved in the genesis of this disease. Prevention plays a crucial role in type II diabetes, as initially, the patient

[1] *Euglycemic range:* refers to BG levels that are within the normal and healty range. Typically, this range is considered to be between 70 to 100 mg/dL for fasting glucose level and less than 140 mg/dL 2 hours after the assumption of glucose

only presents insulin resistance and hyperinsulinemia[1]; the pancreas can maintain adequate glucose levels thanks to increased insulin production. When pancreatic dysfunction occurs, the organ fails to provide appropriate insulin, leading to the onset of type II diabetes. In this type of diabetes, unlike type I, chronic complications primarily manifest, affecting various organs and tissues. The therapeutic plan for managing diabetes involves implementing a proper lifestyle, focusing on correct dietary habits, physical activity, and abstaining from smoking.
For type II diabetics, who generally suffer from obesity, adopting a healthy lifestyle is crucial. This includes reducing calories intake, especially from fats, and increasing physical activity to improve blood glucose, dyslipidemia[2], and blood pressure levels. Approximately 80% of type II diabetes cases could be prevented by adopting a healthy lifestyle.

[1] *Hyperinsulinemia:* condition characterized by higher-than-normal levels of insulin in the blood

[2] *Dyslipidemia:* refers to an abnormal amount of lipids (fats) in the blood

2.2 Artificial Pancreas Systems

In the history of diabetes treatments, several approaches have been tried as alternatives to insulin, but many limitations still remain. The best option to date for T1DM results thus to be a proper blood glucose monitoring and appropriate insulin administration [3].
Therefore, the Artificial Pancreas has become the "golden standard" for the treatment of type I diabetes. The AP is a closed-loop system that automatically controls glycemia by infusing an adequate amount of insulin, according to the measured glucose level.
The primary purpose of the Artificial Pancreas is to preserve safe and healtly BG levels, preventing episodes of hypoglycemia[1] [4] or hyperglycemia[2].
To maintain the BG levels, despite many daily alterations (mainly caused by meals or physical exercise), the device was developed to automate information acquisition, decision making and insulin administration.
Thus, the three main components of all AP_{sys} are: *sensors*, *insulin pumps* and *control algorithms*.

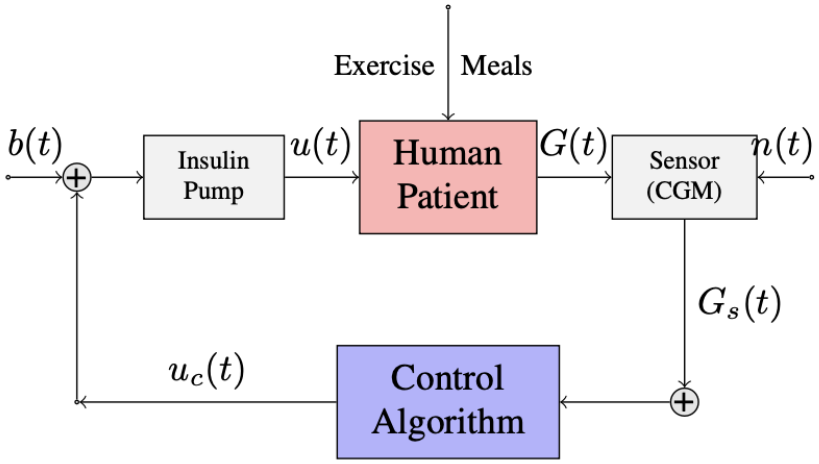

Figure 2.1: Artificial Pancreas control system scheme

[1] *Hypoglycemia*: condition in which the BG levels fall below 70 mg/dL, that can lead to loss of consciousness, coma or even death

[2] *Hyperglycemia*: condition in which the BG levels are above 140-180 mg/dL, that occurs when the body doesn't produce enough insulin

- **Sensor:** a CGM[3] [5] used for for subcutaneous measurements (with, in general, 5 minutes sampling time) of glucose concentration. The sensor wirelessly sends the information to a program stored on the smartphone and calculates the amount of insulin needed, signaling the insulin infusion pump when the hormome needs to be delivered. A CGM has three parts: a *tiny sensor*, inserted under skin (in general on the belly or an arm), which estimates the glucose level in the fluid between cells, very similar to the glucose level in blood; a *transmitter*, which sends information to the third part, a *software program* stored on a separate device, the receiver.

- **Insulin pump:** needed for the delivery of the insulin, which must be delivered subcutaneously, since it cannot be administrated orally; for this reason a small cannula is inserted through the skin into the subcutaneous tissue.
  The insulin infusion pump will deliver small doses of insulin throughout the day, when BG levels are not in the desired range. The insulin pump connects to a small plastic tube and a very small needle. The plastic tube will stay inserted for several days while attached to the insulin pump.
  A second type of pump has, instead, no tubes; this pump is attached directly on skin with a self-adhesive pad and is controlled by a hand-held device. The plastic tube and pump device have to be changed every several days.

- **Control algorithm:** based on CGM measurements, is able to determine the necessary insulin dosage to be injected.

  In the last decades, numerous AP algorithms have been proposed, major categories include: proportional integral derivative (PID) control algorithms, model-predictive control (MPC) algorithms and fuzzy rule-based controllers (FLC).

  - **MPC** (Model Predictive Control): applies a mathematical model of the glucoregolatory system to determine the optimal insulin infusion rate.
  - **PID** (Proportional Integral Derivative): calculates insulin delivery based on excursions from desired glucose (P component), difference between measured and desired glucose (I component) and rate of glucose change (D component).
  - **FLC** (Fuzzy Logic algorithms): approximate the decision-making of diabetes clinicians.

[3] *Continuous Glucose Monitoring*: system that consists of a small sensor, placed under the skin on the abdomen, which continuously measures glucose levels in the interstitial fluid

This topic will be further expounded upon in the next chapter, where the research delves deeper into its nuances and implications.

Benefits

An AP offers various benefits for people with T1DM, including:

- improved glycemic control;
- non-invasive insulin administration (no need for injections);
- remote monitoring and supervision by doctors;
- monitoring of BG levels from smartphone.

Disadvantages

AP devices still need to be perfected, they present disadvantages, such as:

- routine upkeep to ensure the system works properly;
- settings may need adjustments;
- mandatory meal size recordings.

Types of AP_{sys}

There exists several types of AP systems, including:

- **threshold suspend and predictive suspend system:** it stops delivering insulin when BG level drops to a pre-set level. Stopping insulin delivery at the right moment can help avoiding low blood sugar, or hypoglycemia;
- **insulin-only system:** it keeps BG level within a target range by automatically increasing or decreasing the amount of insulin delivered;
- **dual hormone system:** it uses two hormones to control BG, similarly to the way the pancreas works in people who do not have diabetes: insulin to lower glucose levels and glucagon to raise blood glucose levels [6].

Limits of AP_{sys}

Despite AP_{sys} showing incredible promises for treatment of T1DM, they present a one-sided control problem: they are capable of dosing insulin to decrease blood glucose levels, but unable to counteract the effects of a too high dose.
Moreover, they are not completely "hands off": in fact, the insulin amount has to be set by the patient.
Therefore, the next step is a closed-loop system able to automatically determine the insulin dose using a control algorithm.

2.3 Mathematical models of the glucose-insulin system

Mathematical models of glucose-insulin system have a long history.
In the last decades, several models have been proposed with the aim of describing the glucose-insulin dynamics, in both healthy and diabetic patients.
Some of them may be relatively simple approximations (known as "compact" models), others may be relatively complex representations of the glycemic control (known as "maximal" models).
What is needed for the control algorithm's development is therefore a physiologically correct maximal model.
The most well-known and widely used mathematical representations of the physiology of a diabetic individual are the *Sorensen* and the *Hovorka* models, as well as the *UVAPadova Simulator*.
While the Hovorka model and the UVAPadova Simulator are only able to describe the glucose metabolism of a T1DM subject, the Sorensen model has been formulated to simulate the behavior of both normal and diabetic individuals, making it the most complete.

2.3.1 Sorensen model

The *Sorensen* model [7] is the most complex out of the available one, it is composed of three sub-models: one for glucose, one for insulin and one for glucagon. Moreover, it provides the subdivision of the organism in six compartments and each of them corresponds to a specific organ or a set of organs: *brain*, *heart and lungs*, *liver*, *gut*, *periphery (tissue and muscles)* and *kidney*.
Overall, the adopted model is composed of a system of 19 equations, among which non-linear ordinary ordinary differential equations can be found: 8 for glucose, 7 for insulin, and 1 for glucagon (proper balance equations), plus 3 empirically derived equations to define corresponding auxiliary variables (dimensionless), used in some of the balance equations.
In the most general case, for the considered physiological system, the equations of the model can, thus, be written in the following form:

$$\frac{dZ_i}{dt} = \sum_{e=1}^{ne} k_{ie} Z_e - \sum_{u=1}^{ne} k_{iu} Z_u + \sum_{l=1}^{ns} \Gamma_l \tag{2.1}$$

where:

- Z_i indicates the term on which the balance is set on the control volume (or compartmental volume) V_i;
- Z_e indicates a generic incoming flow term into the control volume V_i;

- Z_u denotes a generic outgoing flow term from the control volume V_i;
- Γ_l indicates a source term (or sink, depending on its sign);
- n_e, n_u and n_s indicate the total number of terms entering and leaving the compartment and sources;
- flow constants: $k_{ie} = \frac{Q_{ie}}{V_i}$, $k_{iu} = \frac{Q_{iu}}{V_i}$, with Q_{ie} and Q_{iu} volumetric flows of blood entering and leaving V_i from compartment i.

It takes into account factors such as: insulin sensitivity, glucose effectiveness, insulin secretion dynamics and the effects of meals and physical activity on blood glucose levels.

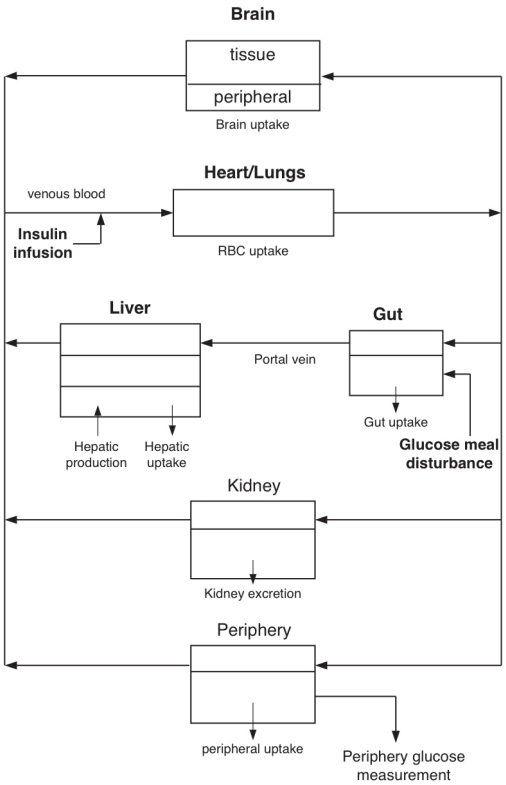

Figure 2.2: Compartment diagram of the Sorensen model

2.3.2 Hovorka model

The *Hovorka* is a sophisticated model that incorporates several physiological processes involved in glucose metabolism and insulin action. As for the Sorensen model, it aims to accurately represent the interaction between glucose production by the liver, glucose uptake by peripheral tissues, insulin secretion by the pancreas and insulin action on tissues. It includes two equations for glucose kinetics, describing its the amount in plasma and tissues. One remarkable feature of this model is its ability to simulate the effects of the subcutaneous insulin infusion on blood glucose level over time.

2.3.3 UVAPadova model

The *UVA Padova Type 1 Diabetes Simulator* is a comprehensive computer simulation model, developed by the University of Virginia and the University of Padova. It is specifically designed to simulate the dynamics of the glucose-insulin in T1DM patients. The representation of the physiological processes involved in glucose metabolism and in insulin action, such as glucose absorption from the gut, insulin secretion by the pancreas, hepatic glucose production and insulin action on peripheral tissues are key features of the simulator. This model includes the glucose and the insulin dynamics, the meal absorption, the hepatic glucose production, the exercises effects and a feedback control. Its ability to simulate various scenarios, accounting for different insulin dosing regimens, meal compositions and exercises levels, make the model a valid candidate for the study of the impact of different interventions on blood glucose control and for the optimizing diabetes management strategies.

2.4 Comparison among three models

- Sorensen model:
  - *Strengths:* ability to reproduce exactly the glycemic pattern; moreover, thanks to the presence of a delay different equation, the model is able to represent the delay between changes in BG levels and corresponding insulin secretion response.
  - *Weaknesses:* the model's complexity may pose challenges for real-time implementation in AP systems; it may also require significant computational resources to simulate.
- Hovorka model:
  - *Strengths:* detailed representation of the glucose-insulin dynamics and accuracy in the simulation of the effects of insulin pump therapy.
  - *Weaknesses:* as for the Sorensen model, the complexity could make it challenging to implement it in real-time applications; in addition, the performances may depend on the specific clinical scenario considered.
- UVAPadova model:
  - *Strengths:* flexibility and versatility to explore different aspects of diabetes management.
  - *Weaknesses:* although the comprehensive representation of the glucose-insulin dynamics, the performances may depend on the accuracy of the input parameters.

As reported in [8], the Sorensen model results, among these three, the most complete and exhaustive in term of physiological description and parameters values, with 22 nonlinear equations and 135 parameters.
Lastly, after considering the above, taking into account factors as the accurate representation of dynamics, the physiological realism, the research validation and the potential for real-time implementations, the choice of the model, for this thesis work, falls on the Sorensen one.

Chapter 3

Literature review

This chapter will provide a short description of this thesis problem from an engineering perspective. For this reason, a brief overview of *Control Systems* and *Machine Learning* will be presented. Regarding control systems, concepts as *modern control theory* and the *data-driven* approaches will be introduced. In terms of machine learning, *neural networks* (particularly recurrent ones) and their role in nonlinear control design will be examined. Finally, the *state of art of the artificial pancreas* will be presented.

3.1 Problem's description

The aim of this thesis work is to make a contribution, albeit a small one, to the development of the Artificial Pancreas. As will be discussed afterwards in this chapter, the research on the development of the AP has been ongoing for decades, and, despite the availability of effective devices on the market, the challenge has not been fully overcome yet. In this thesis, three approaches (two linear and one nonlinear) are proposed for the design of a controller, which aims to manage the glucose level (keeping it within the desired range) as the amount of insulin varies and in the presence of the meal disturbance. They will work by handling the difference between the equilibrium value (of an average adult) of glucose and the actual value (i.e., the one measured by the sensor), aiming to minimize this difference as much as possible. The first two approaches, as will be seen, are valid and functional (in simulation), but they have a major drawback: they are linear methods. Given the complexity of the human body and the equations describing it, it is difficult to believe that a linear controller can manage the nonlinearities of the system. For this reason, a nonlinear approach was considered, involving the use of neural networks. These will play a fundamental role, acting as the controller of the system. The goal will be to "teach" the network (through training) to manage the

patient system, learning to handle its nonlinearities as well.

3.2 An overview of Control Systems

A control system is defined as a system of devices that manages, commands, directs, or regulates the behavior of other devices or systems to achieve a desired result. It plays a crucial role in regulating and optimizing the attitude of different systems, leading to improve efficiency, safety, and performances.
A control system typically consists of four principal components:

1. **Input:** signals or data that the system receives from sensors or other sources;
2. **Controller:** component that processes the input data and determines the appropriate action to take, in order to control the given system;
3. **Plant or Process:** system or device that the controller is regulating or controlling;
4. **Output:** the output of the controlled system.

The main feature of a control system is that there should be a clear mathematical relationship between the input and output of the system; when this relation can be represented by a linear equation, the system is called *linear*.
Contrary, when the relationship between input and output cannot be represented by a linear proportionality, rather the input and output are related by some nonlinear relation, the system is defined *nonlinear*.

Two main types of control system can be distinguished: the open loop and the closed loop:

Open-loop control system

A control system in which the control action is totally independent of the output of the system is called open-loop control system.

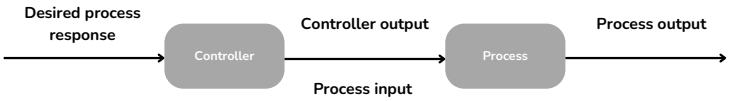

Figure 3.1: Open-loop control system block diagram

Closed-loop control system

A control system in which the output has an effect on the input quantity in such a manner that it will adjust itself based on the output generated, is called closed-loop control system.

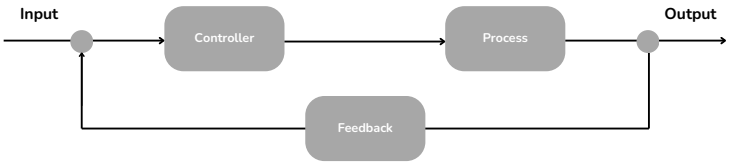

Figure 3.2: Closed-loop control system block diagram

The feedback is what differentiates open loop and closed loop systems.
It feeds a part of the output back to the input; when the feedback is added to the input, it is called *positive feedback*, while, when it is subtracted, it is called *negative feedback*.

3.2.1 Modern Control Theory

Modern control theory [9] is a branch of engineering and applied mathematics that deals with the analysis and the design of control systems. Differently from the classical control theory, which focuses on *LTI*[1] systems, modern control theory manages more complex systems, that may be nonlinear, time-variant or have uncertainties. Modern control methods include optimal control, adaptive control, model-predictive control, sliding mode control, and so on.
A modern approach consists of four main steps:

- modeling the plant;
- design the controller based on the model;
- analyze the control system properties under some assumptions made on the model;
- apply the designed method.

Since the design of the controller, the analysis of the stability and the performances evaluation all depend on the structure, dynamics and assumptions imposed on the system model, it is also called *Model-Based Control*.
MBC strategies, however, may fall short due to the complexity of process dynamics and, moreover, the establishment of models, that are both accurate and easy to use, can be challenging.

In recent years, the acquirement, processing and analysis of "big data[2]" has become a relevant topic in science and engineering.
Therefore, can a controller be designed directly from input-output (I/O) data, bypassing the modeling step?
This question lies at the heart of the concept of *Data-Driven Control*.

[1] *LTI (Linear Time Invariant):* system which is linear and time-invariant. This means that its behavior does not change over time

[2] *Big data*: refers to extremely large and complex datasets (online and offline data) generated by industries

3.2.2 Data-Driven Control

Data-driven control refers to the control theory and method involving the direct use of data for the design of the controller, instead of the explicit information from the mathematical model of the controlled process; thus, in general, DDC is employed for the systems whose models are unavailable.
By harnessing the wealth of data obtainable from sensors, actuators, and other sources in modern systems, and so, by learning the behavior of the plants and the controllers as *black boxes*[1], it is possible to design them directly from input-output data, without relying on explicit system models. For this reason, this approach is also known as *model-free control*.
DDC provides a powerful approach to control system design, particularly in complex and dynamic environments, offering several advantages, such as: adaptability, robustness and flexibility.

As reported in [10], the general framework of Data-Driven Controller Design can be illustrated by the following block diagram structure:

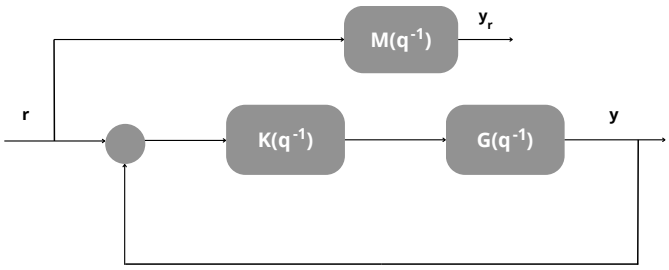

Figure 3.3: Feedback control system to be designed compared with the reference model

where, q^{-1} denotes the standard backward shift operator, and:

- $K(q^{-1})$: the controller to be designed;
- $G(q^{-1})$: the discrete-time plant to be controlled, assumed to be unknown;

[1] *Black box*: refers to a system whose internal structure, dynamics, and mechanisms are not fully understood; it is treated as an entity with known inputs and outputs, and its behavior is characterized solely by the relationship between them

- $M(q^{-1})$: the reference model that describes the desired input-output behavior of the feedback system to be designed.

The aim of DDC is to design a controller $K(q^{-1})$ able to make the *complementary sensitivity function*[1] $T(q^{-1})$ as close as possible to $M(q^{-1})$, $T(q^{-1}) \simeq M(q^{-1})$, relying solely on the input-output dataset garnered from an open loop experiment on the plant:

$$\frac{y}{r} = T(q^{-1}) = \frac{L(q^{-1})}{1 + L(q^{-1})} = \frac{K(q^{-1})G(q^{-1})}{1 + K(q^{-1})G(q^{-1})} \tag{3.1}$$

Direct Data-Driven

Direct data-driven control, also known as model-free control, is a method where, the control system directly uses input-output data to derive the mathematical expression of the controller. Direct data-driven approaches, in fact, do not rely on model identification; the reason behind it is that the available data, experimentally collected, are used to design the controller. Its parameters are computed by formulating the problem in terms of model matching and the control specifications are usually given in terms of a desired closed-loop map. DDDC offers adaptability and robustness, particularly when complex and dynamic systems are considered.

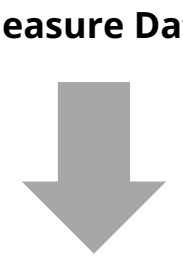

Figure 3.4: Direct data-driven

[1] *Complementary sensitivity function*: is a measure of how well a control system rejects disturbances or uncertainties in the system; it is related to the sensitivity function S through the relationship: $T = 1 - S$

Indirect Data-Driven

Indirect data-driven approaches, instead, involve the development of mathematical models, starting from the identification of the plant model, used to design a control system, via model-based design, able to regulate the system's behavior.
Therefore, *system identification* and *parameter estimation* are fundamental techniques to produce an accurate model from data. Indirect data-driven methods, however, rely on the model's accuracy to make control decisions; for this reason, IDDC is preferable in real-world practical applications where precise models of the plant are obtainable and system dynamics are stable.
Moreover, it is able to guarantee reliability[1] and stability.

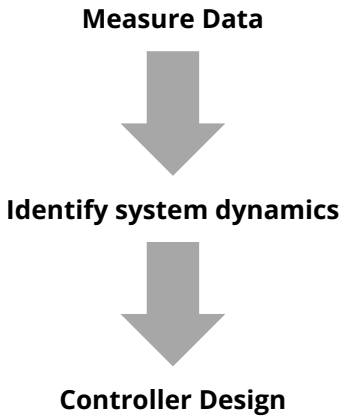

Figure 3.5: Indirect data-driven

In this thesis work, a Direct Data Driven approach will be used.

[1] *Reliability*: ability of a system to perform its intended function consistently and accurately under specific conditions for a specified period of time

3.3 An overview of Machine Learning

Machine Learning is a branch of AI and computer science that focuses on the use of data and algorithms to mimic the way humans learn, gradually improving their accuracy. ML, thus, describes an approach to develop algorithms and statistical models, that enable machines to make decisions and predictions, based on previously collected data samples.
It is possible to distinguish between three main approaches in ML: *supervised learning*, *unsupervised learning* and *reinforcement learning*.

Supervised learning

In supervised learning, algorithms are trained on a labeled dataset, meaning that each input data point is associated with a corresponding output label. The aim of this approach is to learn a mapping from input features to output labels, based on examples provided in the training data. Classification and regression are fundamental tasks of supervised learning. The first one aims to assign input data points into predefined categories or classes based on their features. In classification, the algorithm learns a mapping from input features to discrete output labels, representing the different classes or categories. The second one is used to predict continuous numerical values. In regression analysis, the algorithm learns the relationship between input features and a continuous target variable, allowing it to make predictions about new data points.

Unsupervised learning

In unsupervised learning, algorithms are trained on unlabeled data, meaning that the input data points do not have corresponding output labels. The goal is to discover hidden patterns, structures or relationships within the data. Clustering is considered an unsupervised learning task, as it aims to describe the hidden structure of the object, with no prior labeling. Its goal is to partition the data into subsets such that data points within the same cluster are more similar to each other than to those in other clusters.

Reinforcement learning

In reinforcement learning, an agent learns to perform a task through repeated "t*rial-and-error*[1]" interactions with a dynamic environment, receiving feedback from the environment in the form of rewards or penalties based on its actions. Contrary to supervised and unsupervised, this task does not rely on a static dataset, but operates in a dynamic environment and learns from collected experiences.

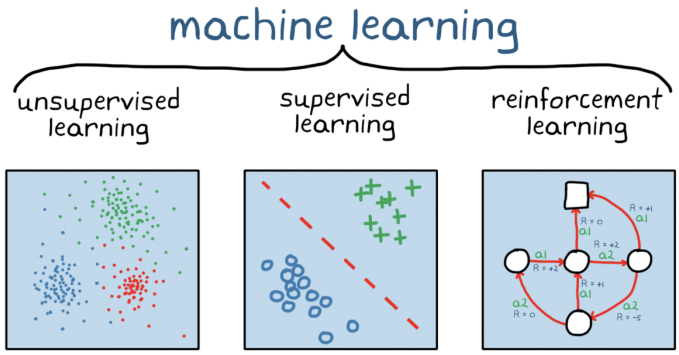

Figure 3.6: Supervised, unsupervised and reinforcement learning

[1] *Trial and error*: problem-solving method in which various solutions are attempted until the desired outcome is achieved; it is characterized by experimentation, learning from mistakes, and refining approaches based on observed outcomes

3.3.1 Neural Networks

Neural networks are at the heart of deep learning[1] models. A neural network is a computational model[2], that makes decisions in a manner similar to the human brain, by using processes that mimic the way biological neurons work together, with the aim to identify phenomena, weigh options and arrive at conclusions.
NNs give a way of defining a complex and non-linear form of hypothesis $h_{W,b}(x)$, with parameters W and b, that can be fitted to data.
Each NN consists of interconnected nodes, *neurons*, organized in layers: an input layer (the leftmost), one or more hidden layers (middle layers), and an output layer (the rightmost).
A Neural Network can be classified as *simple*, if composed of a single hidden layer, or *deep*, if more hidden layers are present.

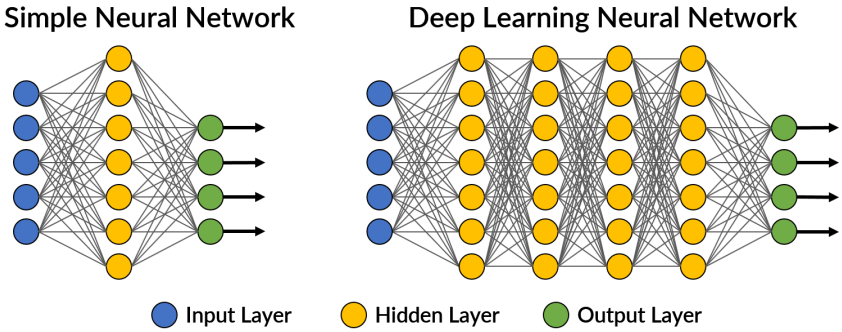

Figure 3.7: Architectures of Neural Networks

To each node are associated weights W and biases b and, being $x = (x_1, x_2, .., x_n)$ a vector of real numbers (inputs), the output y will result to be:

$$y = h_{W,b}(x) = f(b + w^T x) = f(b + \sum_{i=1}^{n} w_i x_i) \tag{3.2}$$

where f is called the *activation function*.
The activation function is mathematical function that determines the output of a neuron. It adds non-linearity to the network, enabling it to learn complex patterns in the data.

[1] *Deep learning*: class of algorithms based on artificial neural network with multiple layers, hence the term "deep"

[2] *Computational model*: mathematical or algorithmic representations of how neurons interact and learn

In the following figure, the typical activation functions are depicted, along with their respective mathematical expressions and graphs.

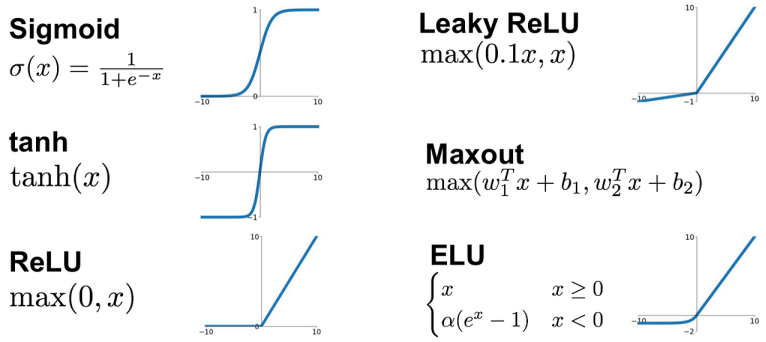

Figure 3.8: Activation functions

Training and validation

Training and validation are key components of ML workflow, particularly when developing predictive models like NN.

- **Training:** during the training phase, the model learns from the labeled data and captures relationships between input features and target outputs.
  The weights and biases are tuned in order to minimize the discrepancy between predicted outputs and true labels.
  The model computes predictions, compares them with the actual labels using a loss function, and updates its parameters using optimization algorithms (e.g., gradient descent) to minimize the loss.
  The result of the training phase is a trained model, with optimized parameters, able to make predictions on new and unseen data.

- **Validation:** The validation phase, instead, is used to evaluate the performances of the trained model, through a separate dataset, called validation set.
  The model makes predictions on the validation set, and performance metrics (e.g., accuracy) are computed.
  Hyperparameters, such as learning rate, regularization strength, or model architecture, may be adjusted based on validation performance to improve the model's generalization ability.
  The result of the validation phase is an optimized model, with tuned hyperparameters, which demonstrates good performance on previously unseen data.

Overfitting and underfitting

Overfitting and underfitting are two common problems encountered in machine learning models, especially in supervised learning tasks.

- **Overfitting:** occurs when a model learns the training data too well, capturing noise or random fluctuations in the data, rather than the underlying pattern. As a result, the model performs well on the training data, but poorly on new and unseen data. In order to reduce the overfitting, increasing the training data provided to the model can help in achieving a better generalization level.
- **Underfitting:** occurs when a model is too simple to capture the underlying structure of the data. In this case, the model fails to learn the patterns in the training data and performs poorly not only on the unseen data, but also on training data. In order to reduce underfitting, instead, using a more complex model or adjusting hyperparameters to increase model complexity can help capturing the underlying patterns in the data.

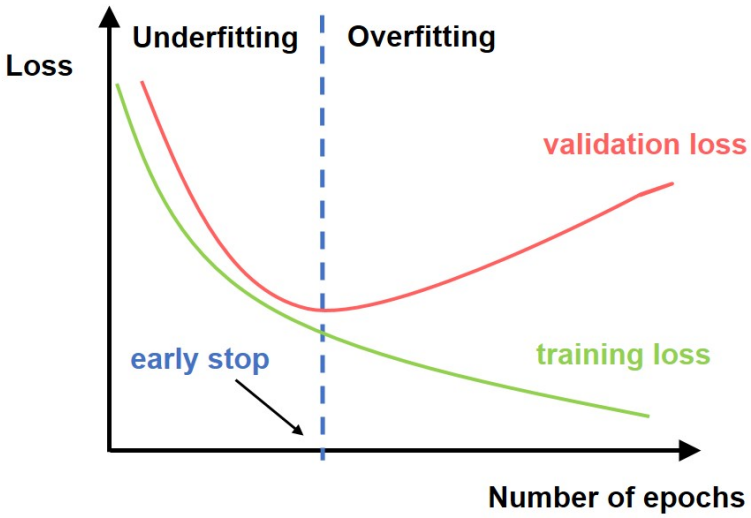

Figure 3.9: Overfitting and underfitting

Anyway, to address these issues, some techniques can be adopted, such as:
l1 or l2 *regularization*, *cross-validation*, *ensemble methods* and *early stopping*.

3.3.2 Recurrent Neural Network

A Recurrent Neural Network (RNN) is a particular type of artificial neural network, that utilizes sequential data or time series data as training data to learn.
While traditional deep neural networks, like *feedforward neural networks* (FFNN) (which process data from input to output in a single direction) and *convolutional neural networks* (CNN), assume that input and output are independent of each other, the output of recurrent neural networks depends on previous elements within the sequence.
Thus, the key feature of RNNs is their ability to maintain a state or memory of previous inputs, as they process sequential data; this memory enables them to capture dependencies and patterns existing in the data.
Another distinctive feature of RNN is that they share parameters across each layer of the network (same weight parameter within each layer), contrary to the feedforward where each node has a different weight.

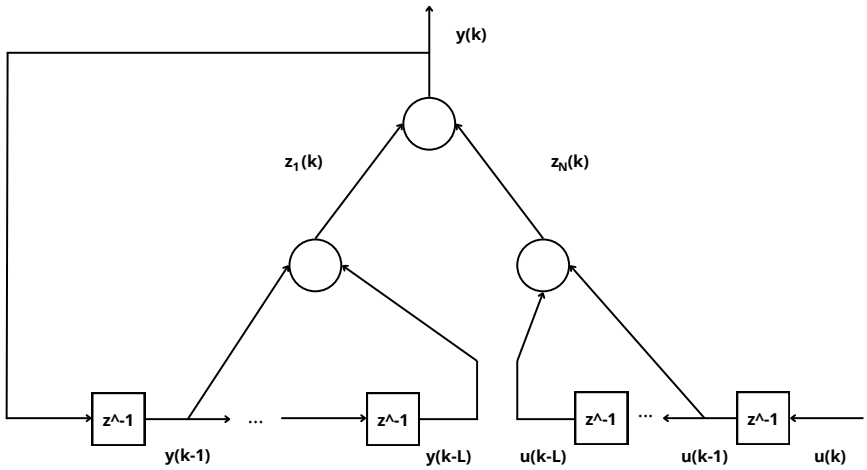

Figure 3.10: Architecture of a single hidden layer RNN

The output of this RNN can be written in this way:

$$y_t = W_2\, tanh\left(W_{1u}\left[u_t; ..; u_{t-n}\right] + W_{1y}\left[y_{t-1}; ..; y_{t-n}\right] + b_1\right) + b_2 \tag{3.3}$$

where, u is the input vector, y is the output vector, W_1 and W_2 are the weights and b_1 and b_2 are the biases; the hyperbolic tangent is the activation function of the layer.

Why Neural Networks for non-linear control design

The use of neural networks for non-linear systems' control design offers a powerful approach to address the complexities inherent in such systems. NNs provide flexibility in modeling relationships between inputs and outputs, enabling them to accurately capture the nonlinear dynamics of the system. Their adaptive nature allows them to continuously learn and, adjusting their parameters based on feedback, ensure robust performance, even in the presence of uncertainties and disturbances. Unlike traditional control techniques, NNs, particularly the recursive ones, excel at handling non-linearities, making them well-suited for a wide range of nonlinear systems.

Vanishing gradient and exploding gradient

Recurrent neural networks, during training phase, leverage the backpropagation through time (BPTT) algorithm to determine gradients, which is slightly different from traditional backpropagation, as it is specific to sequential data. The principles of BPTT are the same as traditional backpropagation, where the model trains by evaluating errors from its output layer to its input layer. BPTT differs from the traditional approach since errors are summed at each time step, while instead feed-forward networks do not need to sum errors since they do not share parameters at each layer. During this process, RNNs tend to encounter two problems, known as *exploding gradients* and *vanishing gradients*. These problems are determined by the size of the gradient, represented by the slope of the loss function along the error curve. When the gradient is too small, it continues to shrink, updating the weight parameters until they become insignificant, i.e., 0. When this happens, the algorithm stops learning. The phenomenon of exploding gradients, instead, occurs when the gradient is too large, thus creating an unstable model. In this case, the magnitude of the model weights will increase too much and will be represented as NaN (Not a Number).
A solution to gradient problems is proposed in [7], and reported below.

Controller Multiplier Optimization

A novel procedure to design a group of optimization algorithms for non-convex, differentiable, and equality constrained optimization problem is proposed. This approach allows to train the NNOE-RNN avoiding limitations showed by standard gradient based algorithm, when applied to RNN network training. It is based on the solution of a fictitious control problem:

$$\begin{cases} min_x = f(x) \\ s.t.\ h(x) = 0 \end{cases}$$

where, f(x) is the objective function and x are the optimization variables (which, in this case, are the neural networks parameters).
The problem can be written in the following way:

$$\begin{cases} \dot{x}(t) = -(\frac{\partial f}{\partial x})^T - \sum_{i=1}^{m} (\frac{\partial h_i}{\partial x})^T \lambda_i \\ y(t) = h(x) \end{cases}$$

The key idea is thus to build the descent direction $\dot{x}$ as a combination of a projected gradient and of a Gauss-Newton, which drives the solution towards the feasible set. In [7], a new CT (continuous-time) framework for convex and non-convex constrained optimization is proposed, particularly focusing on equality constraints.
The presented framework leverages a feedback control perspective: it starts from the solution of first-order necessary conditions for minima and builds a CT dynamic system, whose control input is the vector of the Lagrange multipliers. The output represents the regulated constraints.

3.4 State of Art of the Artificial Pancreas

Since Banting and Best isolated insulin in the 1920s, great progress has been made in the treatment of T1DM, as it can be seen in the following timeline.

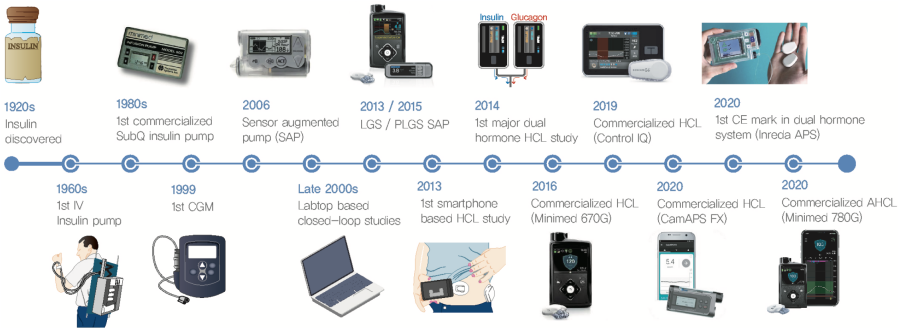

Figure 3.11: Timeline of the development of the AP

The idea of an Artificial Pancreas has existed for a considerable period; however, significant progress depends on advancements in continuous glucose monitoring technology.
Despite the development of numerous non-invasive blood glucose monitoring devices, the current method involves attaching a sensor to the subcutaneous tissue, where it measures glucose levels in the interstitial fluid every 1 to 5 minutes.
Currently, there are several approved AP systems available on the market, most of them are close to being a nearly fully functional pancreas replacement option, but, as anticipated in Section 2.2, they are not completely hands off.
Companies that have developed APs include: *Medtronic*, *Tandem Diabetes Care*, *Insultet Corporation* and *Beta Bionics*.

Challengers and limitations

Despite the AP showing great promises, some challenges and limitations are still present:

- **Accuracy and reliability:** the CGM must be highly accurate for the control algorithm to make the correct insulin dosing decisions;
- **Cost:** current systems can cost thousands of dollars;
- **User training and education:** users must be trained and educated on using these devices.

Control algorithms

As anticipated in Section 2.2, currently, the main control algorithms used for closed-loops are: PID (proportional, integral and derivative control) algorithms, MPC (model predictive control) algorithms and FLC (fuzzy logic control) algorithms.

PID algorithms are composed of three contributions:

1. **proportional:** corresponds to the present, because it detects how far away the glucose level is from the target right now;
2. **integral:** corresponds to the past, because it sees the area deviating from the target in previous trajectory;
3. **derivative:** corresponds to the future, because it predicts the future direction of changes in the glucose level.

The PID control algorithm requires only a sensor glucose variable to operate, which makes it easier to be implemented.

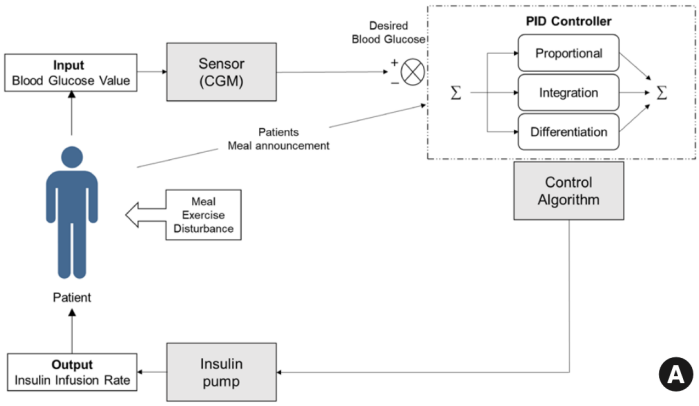

Figure 3.12: PID control algorithm

MPC algorithms predict changes in the dependent variables, after a specific time, by adjusting the independent variables.
In the context of AP_{sys}, covered also in [11], MPC models the glucose level as the dependent variable and adjusts independent variables such as body mass index, insulin-on-board and carbohydrate intake. This modeling is performed every few minutes with updated information. In general, MPC has better performances than PID, but it is more complex and challenging to implement.

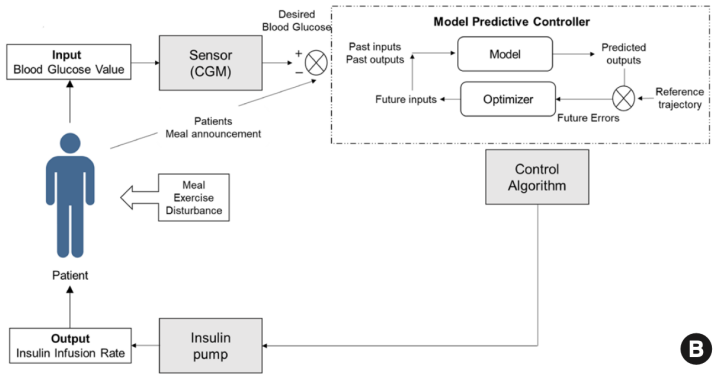

Figure 3.13: MPC control algorithm

Fuzzy Logic control algorithms operate through supervised learning based on expert opinions to establish a specific decision.

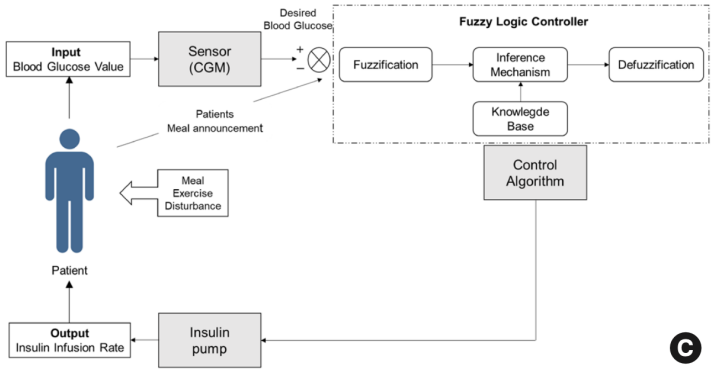

Figure 3.14: FLC control algorithm

A various number of APs have been developed and clinically validated, and, currently, most automated insulin delivery (AID) systems are hybrid closed-loop (HCL) systems, in which the basal insulin is automatically determined and delivered, while mealtime insulin boluses must be managed manually, meaning that people are also required to provide information about the sizes and the assumption times of the meals. Ideally, AID systems should be fully closed-loop system completely automated, requiring minimal user input (no information about meals or exercise). These systems are, in general, single hormone, delivering insulin only, Section 2.2, while, an ideal AID system should also be a dual-hormone system, in order to more closely mimic a biological pancreas and reduce the risk of hypoglycaemia by countering aggressive insulin delivery through the administration of glucagon in addition to insulin. However, the cost and complexity of dual-hormone systems have thus significantly limited their development [12].

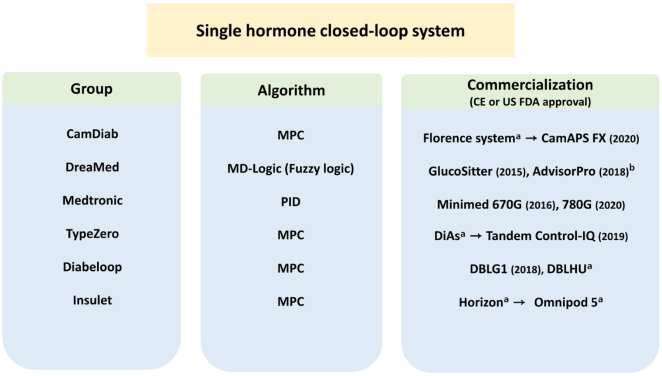

Figure 3.15: Single hormone closed-loop system

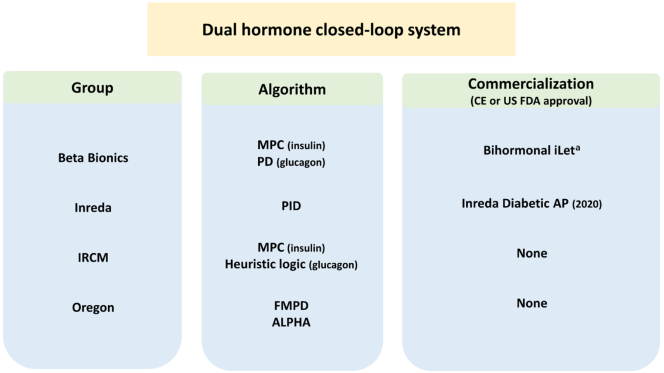

Figure 3.16: Dual hormone closed-loop systems

Chapter 4

Methodology

In this chapter, the methodology used will be presented.
It will be divided into two sections: in the first section, the methodology of the two linear approaches will be explained, starting from data collection and proceeding to the theory behind the two control strategies. In the second section, the methodology used for the non-linear approach will be illustrated.

4.1 Linear approaches

This section aims to provide a comprehensive comparative analysis of two widely utilized linear methods: the *Least Squares* and the *Set Membership*.
The first one yields parameters' estimates that best fit the given data, rendering it a stalwart in regression analysis and system identification.
The second one, instead, takes a divergent approach, emphasizing robustness when facing imperfect or noisy data and, contrary to the LS, delineates a feasible set of parameters consistent with both observed data and predefined bounds and constraints. Before going into detail about the two methods used, however, it's necessary to introduce some fundamental concepts, such as: data collection, reference model and meal model.

Data collection

Data collection was performed using the following scheme:

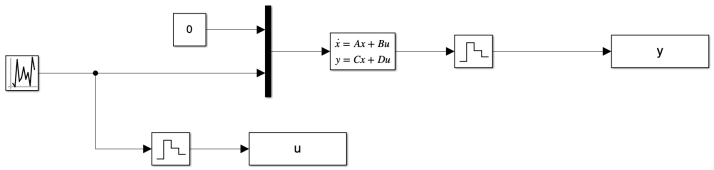

Figure 4.1: Simulink scheme of the system where data are taken

Two inputs were given to the linearized system: a constant at 0, to avoid disturbance (the meal), and a *uniform random number* block, ranging from -1 to 2, with a step of 0.1. Hence, from the system, four thousand input data u, and the corresponding output y, have been collected. Both u and y signals have been garnered with a sample time equal to 5 (imposed by the *zero-order hold*[1]) to simulate the behavior of sensors used in AP_{sys}, which, in general, collect the values of the insulin every five minutes. The same data have been used in both of the following control techniques.

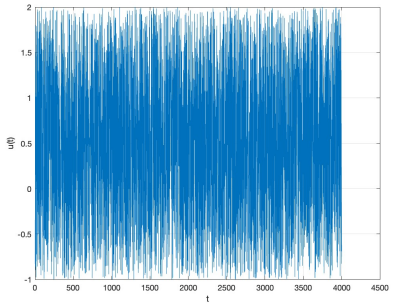

Figure 4.2: Input u(t)

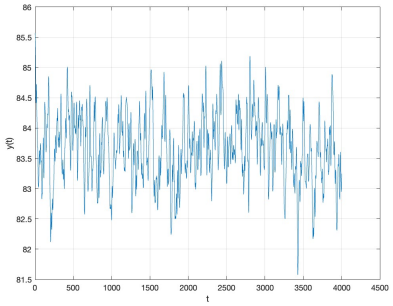

Figure 4.3: Output y(t)

[1] *Zero-order hold*: method used in digital signal processing to convert a continuous-time signal into a discrete one; it samples the continuous signal at regular intervals and holds the sampled value constant until the next sampling instant

Reference Model

As already mentioned in the previous chapter, the reference model serves as standard against which the performance of a system is evaluated; it represents the desired behavior that the system should achieve. In this context, it seemed reasonable to choose a first-order discrete reference model, for T = M, with a single pole and a steady-state gain equal to 1.
In this way, decreasing the value of the pole, from 1 (excluded) to 0, it was possible to select a wider bandwidth, making S smaller, gradually attenuating the effect of the meal.
On the other hand, expanding the bandwidth, the effects of the non-modeled dynamics increase, making the stabilization of the system very difficult.
For this reason, a trade-off with a pole in 0.6 was found, and the model turned out to be:

$$M(z) = \frac{1 - pole}{z - pole} = \frac{0.4}{z - 0.6} \tag{4.1}$$

As for the data, also this model has been used in both LS and SM, but in the second technique a delayed version of it was preferred; this due to the *relative degree*[1] concept; in some cases, the model matching works better if the reference model has a delay equal to the relative degree of the system. Hence, by looking at the transfer function of the linearized system, the relative degree outcomes to be four, so the delayed reference model results:

$$M(z) = \frac{1 - pole}{z - pole} * z^{-4} \tag{4.2}$$

[1] *Relative degree:* measure of how many times the output of a system must be differentiated with respect to time to obtain a specific control input response

The following are the *step responses* of the two reference models:

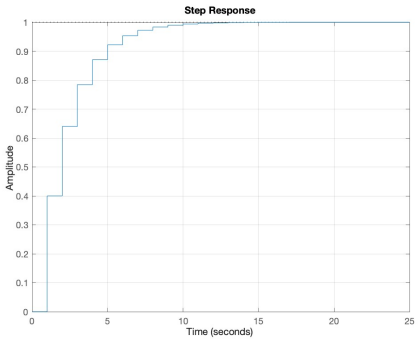

Figure 4.4: Reference model step response

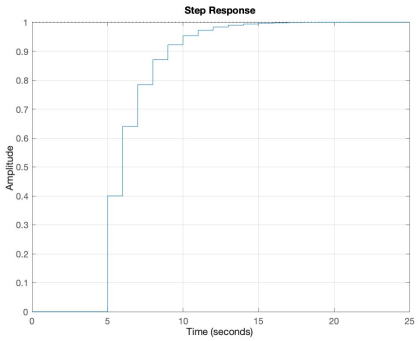

Figure 4.5: Delayed reference model step response

Meal model

The intake of food, which equates to a certain amount of glucose, is considered as a disturbance entering the "patient system".
According to the Lehmann and Deutsch model, the absorption of glucose through the intestinal wall can be described by the following equation:

$$\tau_{pasto}(t) = k_{gabs} G_{gut}(t) \tag{4.3}$$

where $G_{gut}\,[mg]$ represents the quantity of glucose in the digestive system and k_{gabs} is the constant which regulates the glucose absorption, while the flow of glucose $G_{empt}[mg/min]$ from the stomach to the intestine is governed by the following equation:

$$G_{empt} = \frac{dG_{gut}}{dt} - k_{gabs} G_{gut} \tag{4.4}$$

The analytical expression of $G_{empt}(t)$ is:

$$G_{empt}(t) = \begin{cases} mt & \text{if } t_0 \leq t < t_1 \\ Vmax_{ge} & \text{if } t_1 \leq t < t_2 \\ 1 & \text{if } t_2 \leq t \leq t_f \\ 0 & \text{otherwise} \end{cases}$$

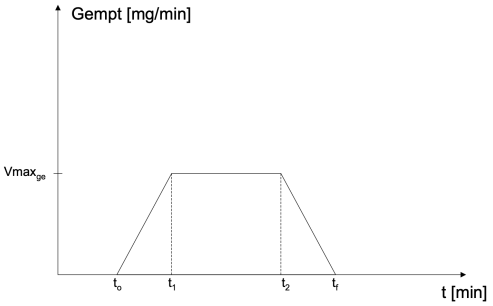

Figure 4.6: $G_{empt}(t)$

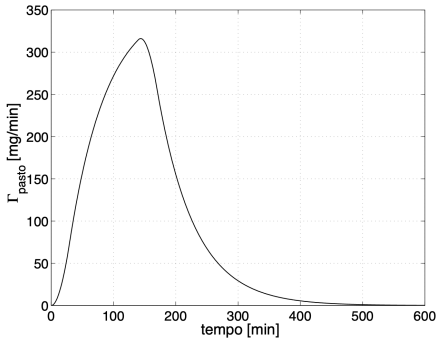

Figure 4.7: $\tau_{pasto}(t)$

In the right figure is reported the trend of the curve, trapezoidal, corresponding to the function $G_{empt}(t)$.
While, in the left figure, the trend of glucose absorption rate in the digestive system is shown for a meal equivalent to 50g of glucose.

4.1.1 Least Square

The Least Square is a parameter estimation method, based on the minimization of the sum of the squares of the residuals.
The formulation of the LS problem, in matrix form, can be written as follows:

$$\theta_{LS} : arg\ min_{\theta \in R}\ ||b - A\theta||_2^2 \tag{4.5}$$

with:

- b: vector of the observed values, i.e. dependent variables;
- A: matrix of the independent variables;
- θ: vector of parameters to be estimated.

The objective is thus to find the θ_{LS} that minimizes the squared Euclidean norm (*l2-norm*) of the residual vector $b - A\theta$, which represents the difference between the observed values and the predicted ones.
In order to solve the linear regression problem, specifically in the context where the matrix A is invertible (square with rows/columns linearly independent), a direct solution, by explicitly solving for θ, can be found:

$$\theta = A^{-1} * b \tag{4.6}$$

In the field of system identification, the LS solution of the system of equations $b = A\theta$ is called *least-square parameter estimate*.
Moreover, it is possible to notice that:

$$lim_{N \to \infty}\ \theta_{LS}\ = \theta_{true} \tag{4.7}$$

where θ_{LS} is the least square estimate, computed with N data, and θ_{true} is the true value of theta.
The 4.7 statement underlines how the estimates, provided by the least squares method, become increasingly accurate and converge to the true one when the number of data points N increases towards infinity.
This implies that with a sufficiently large dataset, the method provides reliable estimates of the true parameters, allowing to understand and model the relationships between data.

4.1.2 Set-membership

Set-membership theory plays an important role in system identification by adding uncertainty and bounded disturbances intrinsic in real-world systems.
SM theory is based on the following three ingredients:

- a priori information about the system to be modeled;
- a priori information on the noise/uncertainty affecting the collected data;
- a set of I/O data experimentally collected.

Unlike traditional modeling approaches, relying on precise mathematical equations, set-membership deals with systems dynamics by defining *feasible parameter sets*, D_θ, which are by definition the sets of all the values of θ, the parameter to identify, which are consistent with the three ingredients. The FPS is thus an implicit description of all the feasible solutions to the system identification problem. By including SM theory into system identification techniques, it is possible to develop *robust control systems*, offering guarantees on performances and stability.
According to set-membership identification theory, as reported in [10], given a set D_θ of all the feasible controller parameters θ, a specific single value is taken in the set to design the controller $K(q^{-1})$ to be implemented in the feedback control system.
The point is selected by looking for the values of the parameter θ that minimize the worst case l_∞ estimation error computed over the entire feasible controller parameter set; the estimate θ^c is the so called l_p-*Chebyshev center* of D_θ, also known as *central estimate* in the set-membership literature.
In particular, for each single component θ_j of the parameter vector θ, the central estimate θ_j^c is given by:

$$\theta_j^c = \frac{\bar{\theta_j} + \underline{\theta_j}}{2} \tag{4.8}$$

where:

$$\bar{\theta_j} = min_{\theta,\, \eta,\, \epsilon\, D_\theta}\, \theta_j, \quad \underline{\theta_j} = max_{\theta,\, \eta,\, \epsilon\, D_\theta}\, \theta_j \tag{4.9}$$

which compose the PUI[1]:

$$PUI = [\bar{\theta_j}, \underline{\theta_j}] \tag{4.10}$$

[1] *PUI*: Possibly Unbounded Interval

Therefore, the vector θ^c of all the θ_j^c is the vector of the unknown parameters estimated that will be used in the creation of the controller K:

$$K(q^{-1}) = \frac{\theta_1\, q^{-1} + \theta_2\, q^{-2} + .. + \theta_{n+1}\, q^{-n}}{1 + \theta_{n+2}\, q^{-1} + .. + \theta_{end}\, q^{-n}} \tag{4.11}$$

being n the order of the controller.

Output Error structure

As anticipated at the beginning of the chapter, the collected data have also been used for SM identification, but, in this case, the output measurements y are assumed to be corrupted by bounded additive noise η, that satisfies the following bounding condition $|\eta(t)| \leq \Delta\eta$, for a known bound $\Delta\eta$, $\forall t = 1, .., N$.
Hence, the input u will be the same and the output y will result:

$$\tilde{y}(k) = y(k) + \eta(k) \tag{4.12}$$

Output Error problem

After generating, through *lsim* command on MATLAB, the desired output s, obtained by simulating the time response of dynamic system, M/(1+M), to arbitrary inputs, r, the OE problem can be formulated as follows:

$$min_{k,s_K}\ ||s - s_K||_\infty \tag{4.13}$$

$$s.t.\ Ky = s_K,\ |\eta| = |y - \tilde{y}| \leq \Delta\eta \tag{4.14}$$

being s_K the output from the controller, $||s - s_K||_\infty$ results to be the simulation error to minimize.

SM main results:

- no matter what the values of N and the noise bounds are, θ_{true} will belong to D_θ;
- no matter what the values of N and the noise bounds are, θ_j will belong to PUI_{θ_j};
- although in practice D_θ is going to become smaller and smaller for increasing values of N, it is not possible to prove, contrary to LS case, that:

$$lim_{N->\infty}\ D_\theta = \theta_{true}. \tag{4.15}$$

4.2 Nonlinear Neural Networks approach

In this section the non-linear approach will be introduced, which will include the use of neural networks. The problem of non-linear model matching, the data collection and the problem development will be presented.

Non-linear model matching problem

Starting from the scheme of the Direct Data Driven Control reported in Figure 3.3, it is possible to divide the system in two branches: the first one, in Figure 4.8, is composed of the reference system, which, in this non-linear case, is L; while the second one, in Figure 4.9, consists of the controller K, to be designed, and the plant of the model P. Both the systems have as input z, and, respectively, as outputs y_r and y.
As for the linear case, the aim is thus to match y and y_r, i.e. to match the nonlinear open-loop functions of the two systems.

Figure 4.8: System 1

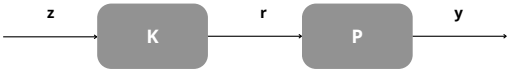

Figure 4.9: System 2

From system 1 it is obtained:

$$y_r(k) = L(q^{-1})\, z(k) \tag{4.16}$$

where $L(q^{-1})$ is defined as:

$$L(q^{-1}) = \frac{M(q^{-1})}{1 - M(q^{-1})} \tag{4.17}$$

Being both the controller K and the plant P non-linear, the formulation of the output y cannot be a linear combination of blocks anymore and so, from system 2, it results:

$$y(k) = P[r(k)] = P[K[z(k)]] \tag{4.18}$$

From equations 4.16 and 4.18, it comes that, in order to achieve $y = y_r$, it is needed to have $P[K(z)] = L_z$.
Since $y_r(k) = L(q^{-1})\, z(k)$, it is possible to invert the relation:

$$z(k) = L^{-1} y(k) \tag{4.19}$$

where y is collected from the experiment on the plant P.

Training optimization problem

The NN has to be trained by using the collected signal r(k) and the signal z(k); in order to match the two systems above, the NN controller to train can be described by the following equation:

$$r'(k) = W_2\, \sigma\left(W_1\left[z(k); ..; z(k-n); r'(k-1); ..; r'(k-n)\right] + b_1\right) + b_2 \tag{4.20}$$

where r'(k) is thus the prediction output of the neural network.
In order to train the NN, which is described by parameters $\theta = [W_1; W_2; b_1; b_2]$, it is needed to solve the following optimization problem:

$$min_{\,y}\, ||y_r(k) - y(k)||_\infty \tag{4.21}$$

The problem results to be a minimization of the infinity-norm of the difference between the reference system's response and the output of the system.
Therefore, the l_∞-norm, object of the minimization problem, represents the loss function employed in the training phase.

The aforementioned optimization algorithm used for the training process of the neural network follows a novel *gradient-free* approach presented in [13].

Chapter 5

Development and experiments

Similarly to the previous one, this chapter will also consist of two sections: one for the linear approaches and one for the neural network's approach.
They will include the explanation of the development of the Simulink models and relative scripts, along with an analysis of some obtained results.
Moreover, some preliminary considerations will be provided.

5.1 Linear approaches

This section will provide a thorough examination of the two linear methods, including their algorithms, experimental setups and results.

Simulink scheme of the complete system

In Figure 5.1 is depicted the Simulink scheme used for all the performed simulations. Two main systems, "patient systems", compose the scheme:

- the upper one, represents the patient with the AP (in other terms, it is the one in which the controller K is implemented);
- the lower one, instead, only describes the "behavior of the patient" and so, it is devoid of the AP.

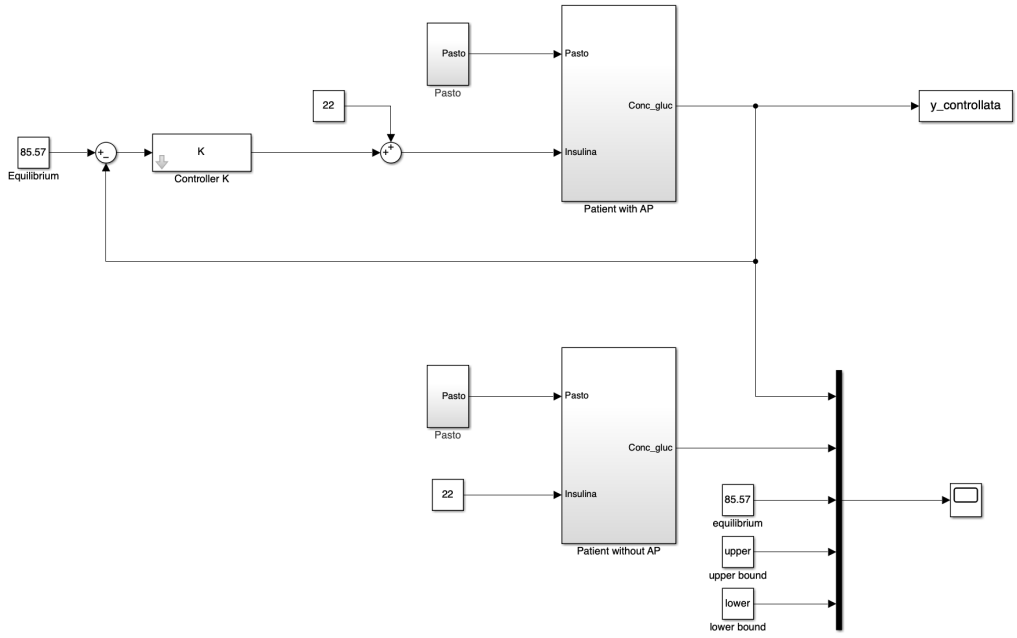

Figure 5.1: Simulink scheme of the system

Each of them has two inputs: the fist one is equal for both systems, the meal, while the second one is the insulin input that, for the lower system, is just a constant block with value 22 (the insulin's equilibrium value), whereas, in the upper system, is composed of a sum between the same constant and the output of the LTI block (in which the controller K is implemented). At the input of the controller there is a sum between the constant 85.57 (the equilibrium value of the glucose) and the output of the system. In the lower section of the scheme are also present three constant blocks:

- *85.57*: the glucose equilibrium value that needs to be achieved;
- *upper bound*: the maximum acceptable value of glucose, fixed to 110;
- *lower bound*: the minimum acceptable value of glucose, fixed to 70;

inserted with the aim of seeing them in simulations' plots employed in the evaluation of the different controller's performance.

5.1.1 Least Square

Least Square development

The Least Square algorithm has been implemented in MATLAB as follows:

Algorithm 1 Least Square algorithm:

```
1: procedure
2:     ▷ load u                                          ▷ loading of input data
3:     ▷ load y                                          ▷ loading of output data
4:     ▷ r = u
5:     ▷ Delta_eta = 0                                   ▷ error bound
6:     ▷ eta = 2*Delta_eta*rand(size(r))-Delta_eta;
7:     ▷ y_mis = y-85.57+eta;
8:     ▷ z = tf('z');
9:     ▷ polo = 0.6;
10:    ▷ M = (1-polo)/(z-polo);                          ▷ Reference Model definition
11:    ▷ [numM,denM] = tfdata(M,'v');
12:    ▷ N = length(y_mis);
13:    ▷ s = lsim(M/(1-M),r);
14:    ▷ n = 2;                                          ▷ order of the controller
15:    ▷ b = s(n+1:N);
16:    ▷ Aout = [ ];
17:    ▷ Ain = [ ];
18:    for k=1:1:n do
19:        Aout = [-s(k:N-(n-k+1)), Aout]
20:        Ain = [y_mis(k:N-(n-k+1)), Ain]
21:    end for
22:    ▷ Ain = [y_mis(n+1:N), Ain];
23:    ▷ A = [Aout Ain];
24:    ▷ A1 = [-s(2:N-1),-s(1:N-2),y_mis(3:N),y_mis(2:N-1),y_mis(1:N-2)];
25:    ▷ theta = A^-1 b;
26: end procedure
```

At first, the input u and the output y data are loaded; then, eta (random and centered in zero) defined through the value of *Delta_eta*, *y_mis*, is calculated by subtracting the glucose equilibrium value from the output y and by adding the value of eta. The reference model is also defined and the s data is calculated. The vector b and the matrix A are built, and so the values of θ_s can be found through the matrix LS formulation (see Equation 4.6).
The script has been written in such a way to be able to easily change the value of Delta_eta and the controller order.

Least Square experiments

Several tests have been done, varying the order of the controller and the value of the Delta_eta. Starting from a controller order n equal to 2, some experiments have been executed changing the value of Delta_eta.
Initially, it was set equal to 0, then increasing bounds were tested, from 0.1 up to 2 (the value 2 has been chosen accordingly to the fact that eta is added to y_{mis}, which has mean magnitude 1.88, meaning that the added error is bigger than the output mean).
In order to evaluate the performances of the found controllers, two metrics were chosen. The first one, the RMSE, is calculated between the desired output (the equilibrium glucose value) and the output of the controlled system.
It is obtained as can be seen in Algorithm 2:

Algorithm 2 RMSE:

procedure
▷ $y_c = y\_controllata.signals.values$; ▷ from the Simulink scheme
▷ $error = y_c - 85.57$;
▷ $RMSE = sqrt(sum(error.^2)/length(error))$
end procedure

The second one consists in the calculation of "*out of boundaries*" percentages of the curve as reported in Algorithm 3:

Algorithm 3 %

procedure

▷ upper = 110;
▷ lower = 70;
▷ $if\ max(y_c) <= upper\ \&\&\ min(y_c) >= lower$
▷ $disp(\text{"}y_c\ always\ in\ healthy\ range\text{"})$
▷ else
▷ $up_{diff} = max(y_c) - upper;$
▷ $low_{diff} = min(y_c) - lower;$
▷ $if\ up_{diff} > 0$
▷ $up_{diff\_perc} = max(y_c) * 100/upper - 100;$
▷ $disp\,(\text{"}Out\ of\ the\ upper\ bound\ by: \text{"} + num2str(up_{diff\_perc}) + \text{"\%"})$
▷ end
▷ $if\ low_{diff} < 0$
▷ $low_{diff\_perc} = abs(min(y_c) * 100/lower - 100);$
▷ $disp\,(\text{"}Out\ of\ the\ lower\ bound\ by: \text{"} + low_{diff\_perc} + \text{"\%"})$
▷ end
▷ end

end procedure

Least Square results

The figure 5.2 shows the plot of the simulation with the best performances, in which the controller K has order 2 and the value of Delta_eta is 0.

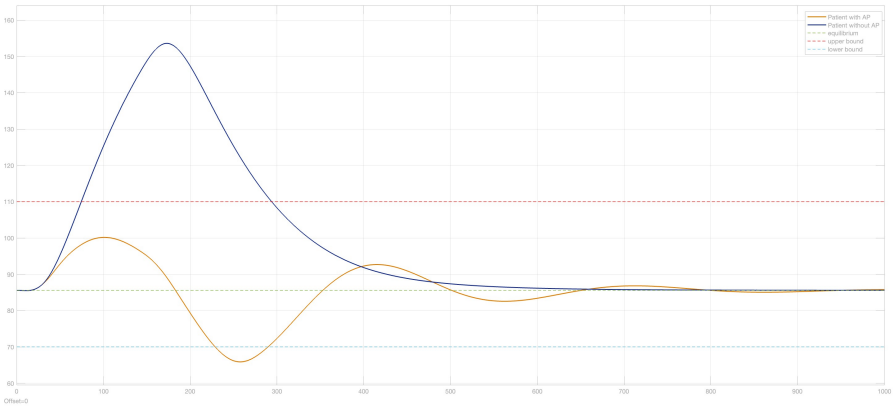

Figure 5.2: LS n = 2 and Delta_eta = 0

The controller K outcome to be:

$$K(z) = \frac{-0.3062z^2 + -0.7026z + 0.9641}{z^2 - 0.7815z - 0.2188} \tag{5.1}$$

It is possible to see, how the orange curve, which is the output of the system with the controller, results to be always below the upper limit (the red dashed line), but by a small amount out of the lower (the light blue dashed line). This simulation, in fact, has the lowest value of RMSE and the lowest, different from zero, percentage of "out of lower bound".

Below, the Table 5.1 reports metrics' values for some of the more significant simulations done:

Controller's order	Delta_eta	RMSE	% out of upper bound	% out of lower bound
2	0	7.5433	0	5.8494%
2	0.1	24.6429	10.973%	37.1357%
2	0.5	38.7320	43.9611%	68.6408%
4	0	25.7445	0	23.9253%
4	0.1	17.7332	2.6508%	27.6272%
4	0.5	20.6413	4.3553%	27.3707%
6	0	25.7711	0	23.9184%
6	0.1	35.0698	37.5454%	62.5339%
6	0.5	97.6203	201.4564%	121.8688%

Table 5.1: Least Square tests' metrics

From the Table above, it can be observed that the LS achieves better results in the absence of an error on the output and, as its magnitude increases, the performances get worse: both the RMSE and percentages tend to higher values.
In fact, while the LS method is a powerful and widely used technique for fitting models to data, its working behavior can be compromised in the presence of errors; it seemed pointless, therefore, to report the values obtained from simulations with larger errors, such as 2.

Other simulations have been performed using a different meal model. It has been developed following the eating pattern of a standard adult man, composed of three meals (breakfast, lunch and dinner) distributed throughout the day with different glycemic peaks: the light blue curve in the following figure.

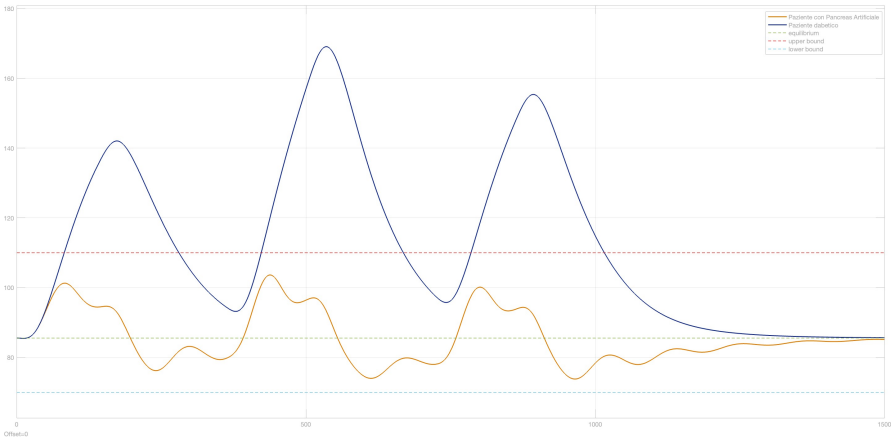

Figure 5.3: LS new meal model

It can be noticed that, even changing the shape of the disturbance added to the system, the LS performs as well as before. The controlled curve, the orange one, results to be always in the desired range. As for the previous case, adding a measurement error on the output, during the design phase, the resulting algorithm is not able to control the system anymore.

5.1.2 Set Membership

Set Membership development

The Set-Membership algorithm has been implemented on *Julia*[1] as follows:

Algorithm 4 Set Membership:

procedure
▷ Ntrain = 300;
▷ s = s[1:Ntrain];
▷ y_t = y_t[1:Ntrain]-85.57*ones(300);
▷ Delta_eta = 0;
▷ y_t=y_t+2*Delta_eta*rand(length(y_t))-Delta_eta*ones(length(y_t))
▷ n = 6;
▷ @polyvar Np[1:n+1];
▷ @polyvar Dp[1:n]; Dp1 = [1; Dp];
▷ @time theta_hat,bound = smi_dt_minOE(y_t, s, Np, Dp1, [Np;Dp], false, Delta_eta, 2, "Mosek", "linf");
▷ @show theta_hat
end procedure

Through the Algorithm 4, the number of data (Ntrain) can be chosen, and, as for the LS, the value of the glucose equilibrium has been subtracted from y_t.
In lines 4 and 5, it can be decided whether to add or not the error (random and centered in zero) on y_t. Moreover, the order of the controller can be defined and through the command @polyvar it is possible to make the numerator and the denominator of the controller K the variables to be identified.
At the end, using the function *smi_dt_minOE*, the values of theta_hat and the minimum bound can be found.

Algorithm 5 smi_dt_minOE function:

procedure
▷ function smi_dt_minOE(uk, yk, num, den, theta, stability_cns, Delta_eps, relaxOrder, solver, mode);
end procedure

The arguments of the function are, in order: the input (y_t), the output (s), the numerator and the denominator previously defined, the variable theta (which

[1] *Julia*: high-level programming language specifically designed for scientific computing, numerical analysis, and data science.

composes the Np and the Dp), the stability constraints (set to false in this case), the value of Delta_eta, the relaxation order of the problem (set to 2), the solver used (Mosek) and the mode, which is the norm to be minimize (infinity-norm in this case).

Set Membership experiments

As for the LS, also for the SM, several test have been executed, varying the order of the controller and the value of the Delta_eta. In order to evaluate the performances of the found controllers, in this case too, the same metrics have been employed.

Set Membership results

The figure 5.4 shows the plot of the simulation with the best performances, in which, as for the LS, the controller has order 2 and the value of Delta_eta is 0.

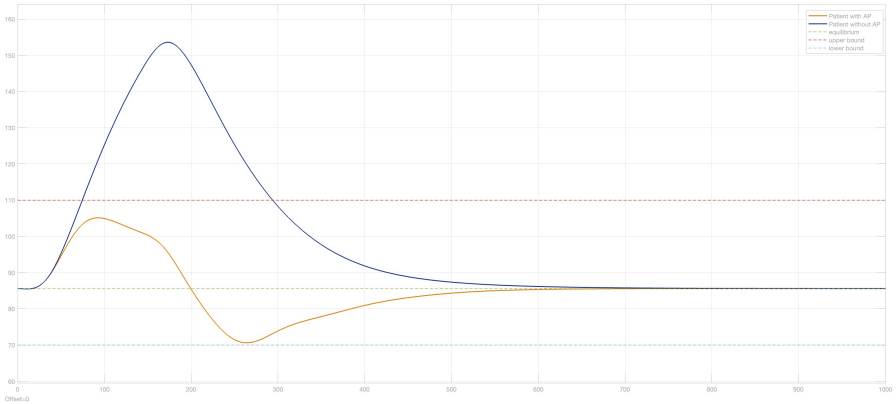

Figure 5.4: SM n = 2 and Delta_eta = 0

The controller K outcomes to be:

$$K(z) = \frac{-1.2088z^2 + 2.0978z - 0.9069}{z^2 - 1.5608z + 0.5608} \tag{5.2}$$

The RMSE results to be the lowest and both the percentages are 0. As it can be seen, in fact, the orange curve is always in the desired range and moreover, it does not present oscillations.
The same behavior can be found in the case with Delta_eta equal to 0.1, with the only difference that the RMSE is slightly higher.

Controller's order	Delta_eta	RMSE	% out of upper range	% out of lower range
2	0	7.4388	0	0
2	0.1	7.4585	0	0
2	2	16.4829	10.077%	39.0426%
4	0	7.7333	0	0.29379%
4	0.1	7.5321	0	0
4	2	11.8898	6.4232%	21.4855%
6	0	10.6795	0	0.46956%
6	0.1	10.6616	0	0.11076%

Table 5.2: Set Membership tests' metrics

Acceptable results are also obtained with controller's order of 4 and 6 and with values of Delta_eta 0 and 0.1, as reported in Table 5.2. All the values of RMSE are restricted and also the percentages are low, except for the cases in which the error is 2, but, contrary to LS, it was possible to insert the results obtained in this setting due to the fact that the system does not diverge.

Also for the SM, some experiments have been performed with the new model of the meal. The following figure represents the behavior of the system with the same controller used in Figure 5.5.

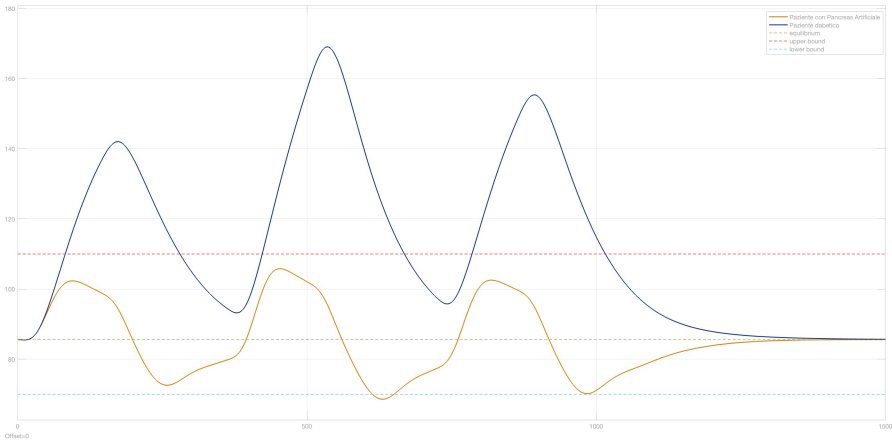

Figure 5.5: SM new meal model

Identical considerations of the LS can be done: the controller K is able to control the system in the presence of different shape of disturbances, and, also in this case, increasing too much the value of the error bound, the system is, again, not correctly controlled anymore.

5.2 Nonlinear Neural Networks approach

This section will provide a thorough examination of the non linear method, including its algorithms, experimental setups and results.

Simulink scheme of the non linear system

Starting from Figure 5.1, which includes the patient systems and the one with the linear controller (either the least square and the set membership one), the following block will be added, obtaining a scheme able to compare the three scenarios. The latter has the same structure of the system with the linear controller, the only difference is represented by the linear controller being replaced by the non linear one, i.e. the neural network controller.

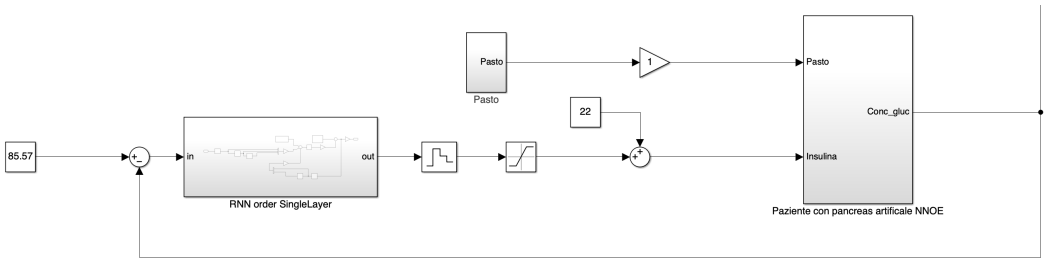

Figure 5.6: Simulink scheme of the neural network system

In the following figure, the simulink scheme of the recurrent neural network is reported, having the same structure shown in Figure 3.10:

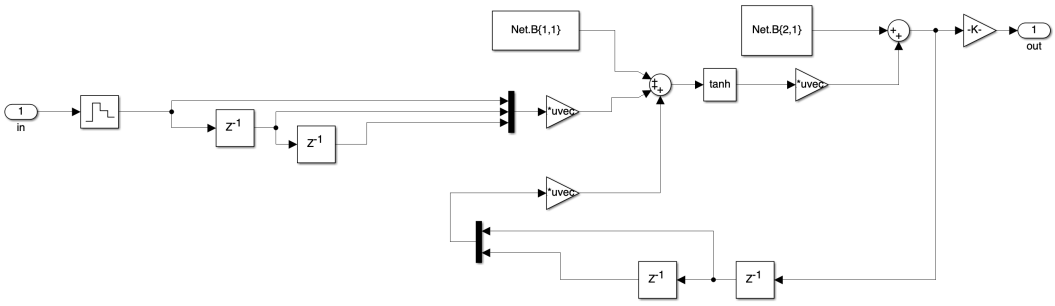

Figure 5.7: Simulink scheme of the neural network

It can be noticed, from the delays present at the input and output, that the dynamic order of the system (i.e. the order of the controller) is 2. Moreover, the number of neurons in the hidden layer has been set to 6.

Neural Network controller development

The Neural Network approach has been implemented on MATLAB through different functions and scripts. In particular, the script devoted to the training of the network is structured in the following main sections:

- **Data preparation:** loading and pre-processing of the data;
- **Network initialization:** definition of the architecture of the neural network, including the input and the output layers, the number of hidden layers and the number of neurons;
- **Training configurations:** initialization of the optimization variables, definition of the regularization coefficient and of the initial conditions;
- **Training loop:** implemented through a for loop for the training of the network.

Moreover, an *Auxiliary function* has been used. It performs the computation of the recurrent neural network parameters by implementing the optimization algorithm with the objective function and constraints presented in Section 4.2. In this computation, the regulation coefficient is responsible of shrinking to zero the error in the minimization problem (the tuning of this parameter has been executed by a trial and error procedure), and gradient computation is performed.
At the end, a section responsible for the plotting of the *evolution of parameters' estimates* and of the *RMSE evolution during the training* is implemented, in order to evaluate the success of the training process.

Neural Network controller experiments

In this section, two experiments will be considered, both concerning the training phase of the network. In particular, data are collected, then implemented and used to train the NN.

First experiment

As first approach, in order to verify that the neural network was learning correctly, it was trained using the data (input and output) taken from the best linear controller found (the SM one). The purpose was, thus, to ensure that the network could learn and replicate the behavior of the SM linear controller.

Data collection

The data collection was performed using the following scheme, namely the patient system with the linear controller implemented. The input and the output of the linear controller were collected with a sample time equal to 5.

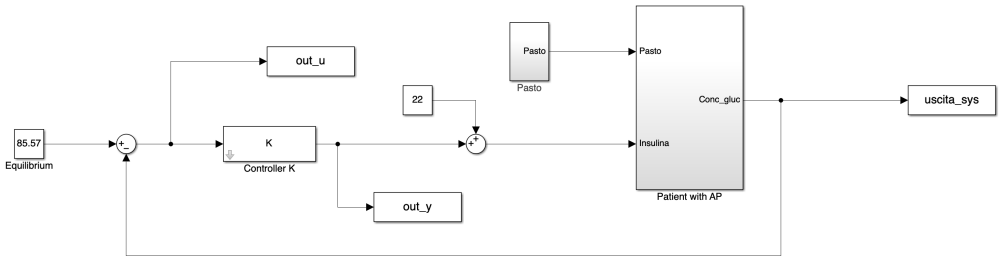

Figure 5.8: Simulink scheme of the system where data were taken from

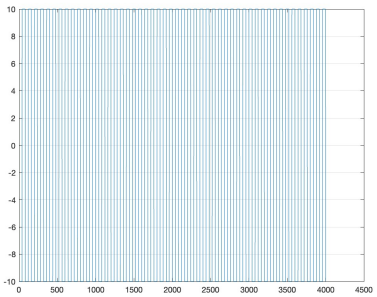

Figure 5.9: Input of the controller, *out_u*

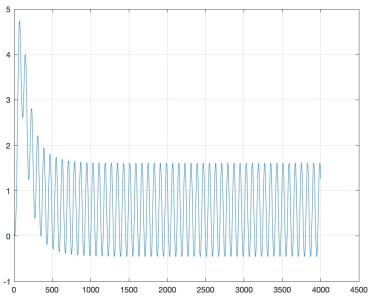

Figure 5.10: Output of the controller, *out_y*

Algorithm 6 Load of the data

```
procedure
    ▷ load out_u                          ▷ input from the controller
    ▷ load out_y                          ▷ output from the controller
    ▷ y_NNOE = out_y - 85.57;
    ▷ u_NNOE = out_u;
    ▷ utrain = u_NNOE;                    ▷ neural network input
    ▷ ytrain = y_NNOE;                    ▷ neural network output
end procedure
```

From the above algorithm, it can be seen how data are loaded; after that, the equilibrium value of the glucose is removed from the output and y_NNOE is used as output of the neural network, while the controller input u_NNOE is used directly as neural network input.

Second experiment

As second approach, the neural network was trained using the data taken from the output of the SM linear controller and from the output of the complete system (i.e. from the plant).
The linear controller's final result has been used as ground truth of the neural network; while, as input, the data z, obtained through the Equation 4.19, was employed.

Data collection

The data collection was performed through the same scheme used in the first experiment. In this case, the output of the linear controller and the output of the complete system were collected, as always, considering a sampling time equal to 5.

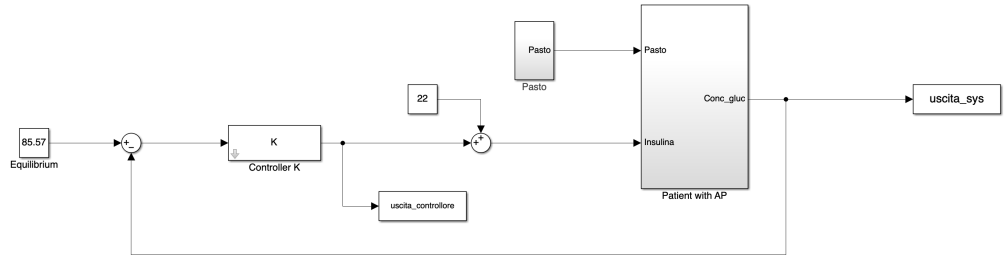

Figure 5.11: Simulink scheme of the system where data were taken from

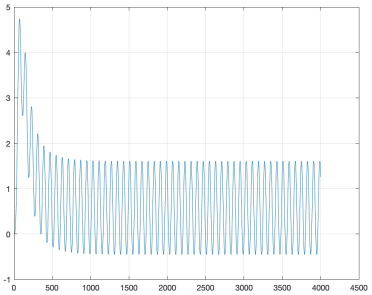

Figure 5.12: Output of the controller, *uscita_controllore*

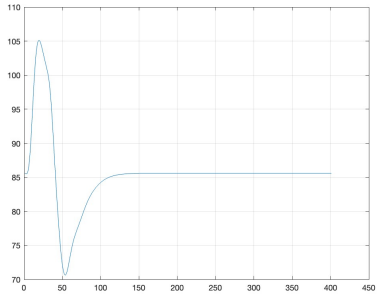

Figure 5.13: Output of the system, *uscita_sys*

Algorithm 7 Load of the data and z

procedure
▷ load uscita_controllore ▷ output from the controller
▷ load uscita_sys ▷ output from the plant
▷ y_NNOE = uscita_sys - 85.57; ▷ system's output minus equilibrium
▷ u_NNOE = uscita_controllore;
▷ z = tf('z');
▷ polo = 0.6;
▷ $M = (1 - polo)(z - polo) * z^{-}4$; ▷ Reference model
▷ for k=1:1:size(y_NNOE)-1
▷ z_NNOE(k) = (1/polo)*(y_NNOE(k+1)-y_NNOE(k));
▷ utrain = z_NNOE'; ▷ neural network input
▷ ytrain = u_NNOE; ▷ neural network ground truth
end procedure

Neural Network controller results

First experiment

In the following figure, the result of the first experiment is shown. As usual, the blue line represents the diabetic patient's curve, the orange line represents the curve obtained by implementing the set-membership controller in the system, and the green line represents the curve obtained by training the neural network with the input/output data obtained from the linear controller. It can be observed that the two curves are almost overlapping, indicating that the network is correctly capable of learning.

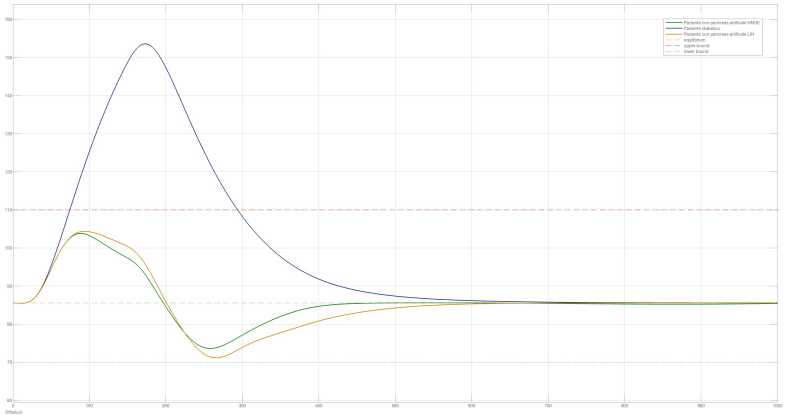

Figure 5.14: First experiment plot

Second experiment

As second experiment, the neural network was trained as previously explained, and after several training sessions, the best network was obtained, which turned out to be the one achieved at epoch 2400, out of 2500 that were set.

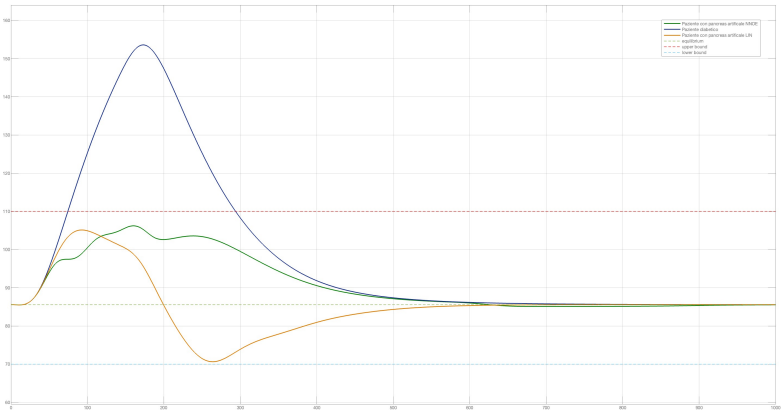

Figure 5.15: Second experiment plot

Starting from Figure 5.4, in which the yellow curve represents the behavior of the system patient with the best SM linear controller, it has been added the green curve that depicts the system patient with the neural network controller.
It can be seen that the green curve never falls outside the desired safe range, which is why the metric used to evaluate the linear controllers in this case was not calculated; on the contrary, the RMSE value was computed, which results to be:
RMSE: 7.3818.

Validation and results

The validation phase took place by experimenting the NNOE controller in the patient system by testing different sets of input, in particular the meal model (that consists of breakfast, lunch and dinner) already employed in the LS and the SM cases. As shown in the following figure, the neural controller is able to control the system well, without oscillations and always remaining within the safe range, never dropping below equilibrium, unlike the SM one.
This is especially relevant from a medical perspective, as it ensures a situation of constant safety for the patient.

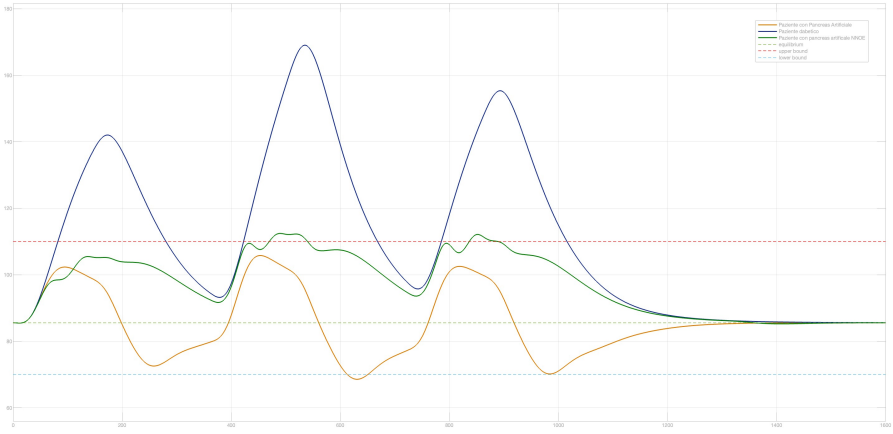

Figure 5.16: Validation of the neural network controller

In this specific case, the RMSE turns out to be 12.3682.

Chapter 6

Results' analysis

In this chapter, an analysis of the obtained results will be provided, the pros and cons of both approaches will be presented, and comparisons will be made.

6.1 Linear approaches

In the upcoming section, the linear approaches will be analyzed, starting from the pros and cons identified for both methods, followed by a comparison between them.

6.1.1 LS: Pros and Cons

Pros:

- **Computational efficiency:** the algorithm proves to be very efficient, and therefore fast, even when using a large amount of data;
- **Optimal under certain conditions:** when the assumptions of the linear model hold, the LS estimates are unbiased and efficient;

Cons:

- **Assumption of linearity:** LS assumes a linear relationship between the predictors and the target variable;
- **Sensitive to outliers:** LS is highly sensitive to outliers in the data.
  A single outlier can significantly impact the resulting regression coefficients and predictions;
- **Not suitable for non-linear relationships:** LS assumes a linear relationship between the predictors and the target variables. If the relationship is non-linear, it may produce biased or inefficient estimates.

- **Overfitting.**

Overall, the least squares approach is a powerful and widely used method for data analysis and model fitting, but it requires careful consideration of its assumptions and potential limitations.

6.1.2 SM: Pros and Cons

Pros:

- **Robustness to uncertainty:** this approach can effectively handle uncertainty and variability in data by considering all possible values within a set;
- **Non-probabilistic:** SM does not require assumptions about the probability distribution of errors
- **Computational efficiency:** the algorithm can be computationally efficient, particularly when dealing with linear constraints and bounded sets.

Cons:

- **Implementation's complexity:** implementing the SM can become mathematically and computationally challenging;
- **Overfitting.**

Overall, the set membership approach is a powerful tool for dealing with uncertainty and variability in data, but it requires careful consideration of its limitations and the specific context in which it is applied.

6.1.3 Comparison among LS and SM

After listing pros and cons of the two methods, it is important to underline the causes behind the choice of switching to a SM approach. The main reason revolves around the behavior of the LS in the presence of errors. In fact, as it possible to see in Figure 6.1, when errors are added, even if very small (Delta_eta equal to 0.1), the latter is not capable to control the system: the orange curve, not only exits several times from the range, but it is also characterized by an oscillatory behavior. Contrarily, the SM approach, Figure 6.2, even when facing errors, it is able to discretely control the system and, even more, considering smaller ones, like the one in the figure (Delta_eta equal to 0.1), its behavior appears to be very similar to the one with noiseless data.

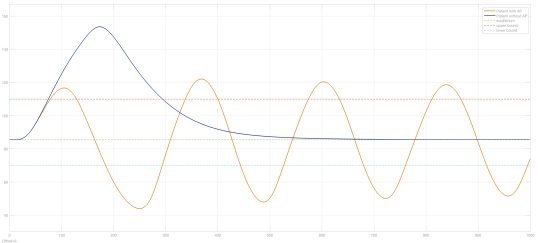

Figure 6.1: LS with Delta_eta = 0.1

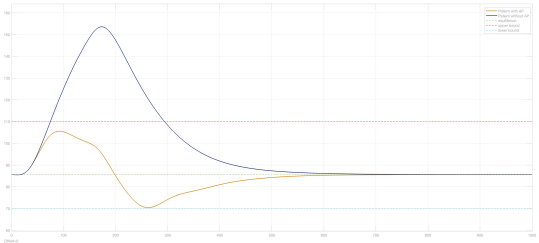

Figure 6.2: SM with Delta_eta = 0.1

Another comparison can be done in the case Delta_eta equal to 2:

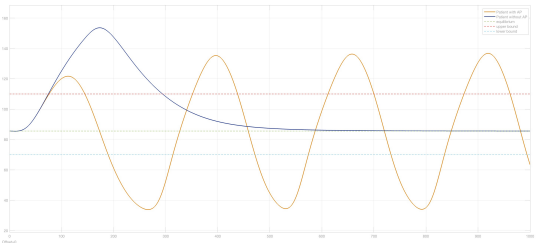

Figure 6.3: LS with Delta_eta = 2

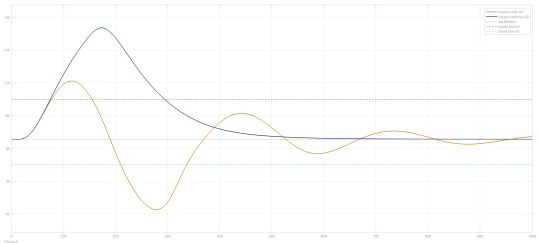

Figure 6.4: SM with Delta_eta = 2

Both the simulations have been performed with n equal to 4 and Delta_eta equal to 2. For the LS case, Figure 6.3, it is possible to notice that the orange curve exits the desired range several times, while, in the SM case, Figure 6.4, the curve results to be in the range after 300 minutes. It must be, however, highlighted the presence of a peak below 60, that, from a medical point of view, is not acceptable. In conclusion, the LS is not able to perform a control action on the system in presence

of errors, whereas, despite the SM being capable to control the system, from a medical point of view, the results are not sufficiently accurate to be accepted.

6.1.4 The overfitting problem

In the previous sections, among the cons of the two linear methods, overfitting was mentioned. It is an undesirable machine learning behavior that occurs when the model gives accurate predictions for training data, but not for new data. The model learns the training data too well, becoming too complex and so fitting the training data too closely, leading to poor generalization to new and unseen samples. In this context, there exists a point at which, if the order of the controller is increased too much, it stops improving and further destabilizes the system.
In fact, if the model is too complex, during the identification phase, it tries to employ all degrees of freedom to explain the data well and resulting in overfitting. This is the reason why in the Tables 5.1 and 5.2, provided in the previous chapter, the order of the controller only goes up to 6, as it was observed that, beyond this value, starting from order 7 or 8, the controller was unable to stabilize the system, which eventually diverged.

6.2 Nonlinear Neural Networks approach

In the next section, the nonlinear approach will be analyzed, starting from the pros and cons identified, followed by a comparison with the linear methods.

6.2.1 NNOE: Pros and Cons

Pros:

- **Nonlinear Control:** neural networks can handle the nonlinear dynamics of glucose metabolism better than traditional linear control systems;
- **Adaptive Learning:** neural networks can adapt to individual variations in physiology and lifestyle, learning from historical data and improving their performance over time;
- **Personalization:** they can be tailored to individual patients' data, leading to highly costomized treatment plans;
- **Predictive Capabilities:** with sufficient training data, neural networks can predict future glucose levels and adjust insulin delivery preemptively, potentially reducing hypoglycemic and hyperglycemic episodes;
- **Integration with other data sources:** NN can easily integrate data from multiple sources, such as CGM or insulin pumps, providing a comprehensive description of the patient's health.

Cons:

- **Data Requirements:** the training phase requires large amounts of high-quality data, which may not always be available;
- **Computational Complexity:** neural networks can be computationally intensive, which could be a challenge for real-time processing on wearable devices with limited power and resources;
- **Lack of Transparency:** NN are often seen as "black boxes" due to their complex inner principles, making it difficult to understand how decisions are made. This can be a concern in a medical context where transparency is crucial;
- **Resource Intensive:** developing, training, and maintaining neural network models requires significant resources, including expertise in machine learning and data science;

- **Overfitting.**

Overall, a neural network controller for an AP offers promising benefits in terms of: adaptability, precision, and predictive capabilities. However, challenges related to: data requirements, computational complexity, and transparency must be addressed to ensure these systems are safe, reliable, and effective in managing diabetes.

6.2.2 Comparison among linear and nonlinear methods

In the following figures, the best controllers found for each method used are shown for comparison. Starting with Figure 6.5, which shows the behavior of the system managed by the controller obtained using the LS technique, it can be observed that, despite being a valid and functional controller, it exhibits several oscillations. Additionally, it is important to remember that the controller was obtained using only error-free data.

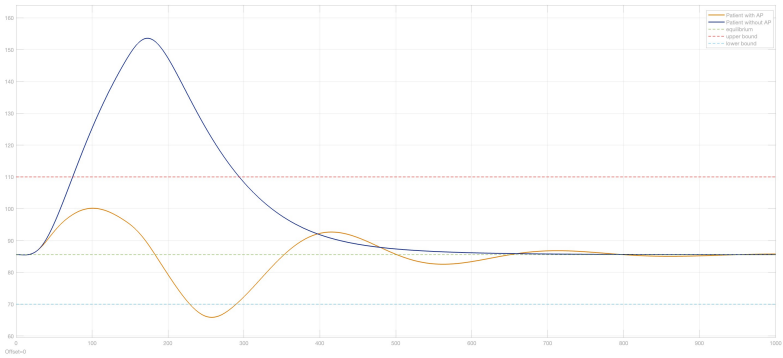

Figure 6.5: Least Square controller

Conversely, the Figure 6.6 displays the behavior of the system with the SM controller. Although it demonstrates a significantly better performance compared to the previous one and operates effectively in the presence of limited errors, it is still a linear controller and thus incapable of managing all the non linearities present within the system.

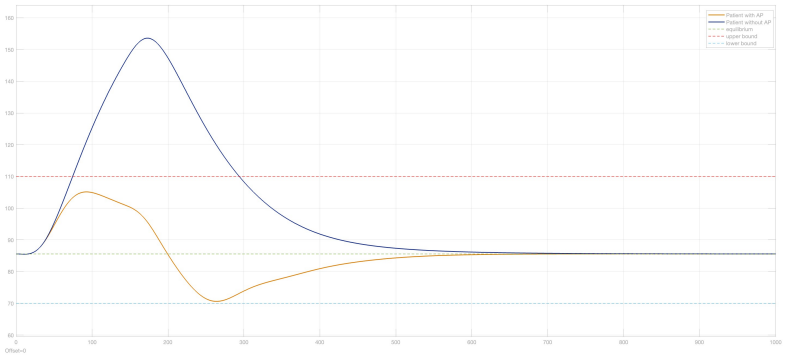

Figure 6.6: Set-Membership controller

At the end, the behavior of the system controlled by the RNN is reported in Figure 6.7. It performs significantly better than any other controller presented so far, exhibiting no oscillations, always staying within the safe range, and never dropping below a certain threshold, thus ensuring the patient's safety. Moreover, it encompasses all the aforementioned disadvantages, 6.1.4, particularly the ability to handle the non linearities of the system.

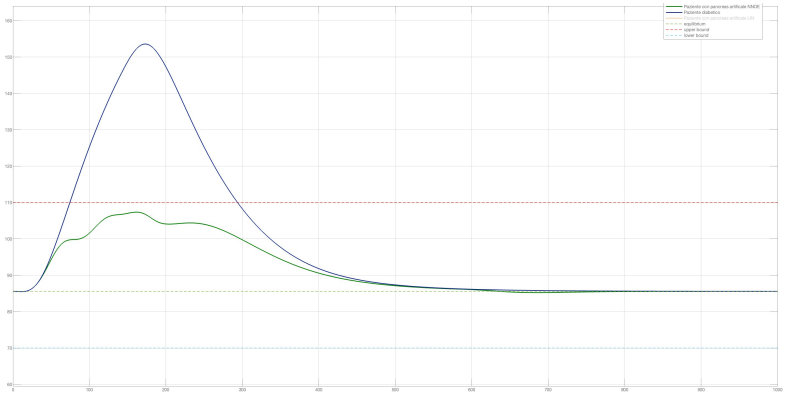

Figure 6.7: Neural Network controller

6.2.3 The overfitting problem

As anticipated in 3.3.1, one of the main issues that can arise when using neural networks is the overfitting. An example of this phenomenon was observed during the network's validation phase. The model that had shown the best performance at the end of training, Figure 6.8, during a test where a different meal was applied (representing new and previously unseen inputs for the network), exhibited unacceptable behavior, contrary to expectations.

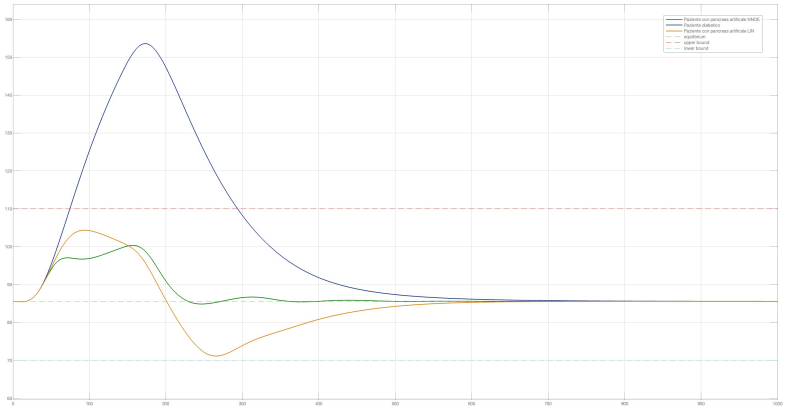

Figure 6.8: Best RNN controller

As shown in the Figure 6.9 below, the network is no longer able to control the system, converging to a point far from equilibrium, which is dangerous for the patient's safety. This behavior is typical of overfitting: the model had learned the training data too well, thus failing to generalize correctly when faced with previously unseen data.

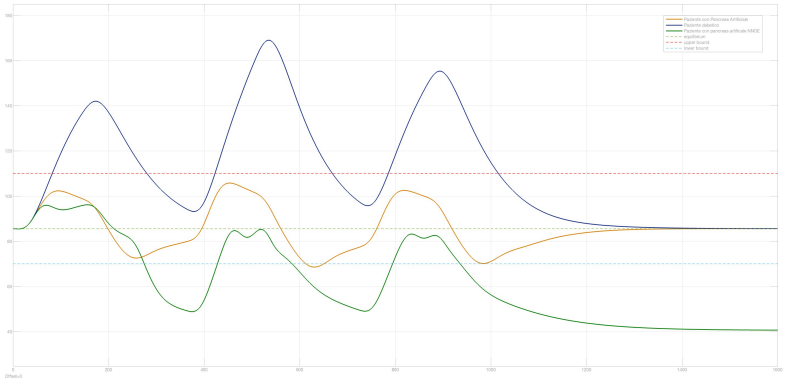

Figure 6.9: Validation of the neural network controller

Chapter 7

Conclusions and Future Improvements

In this chapter, a brief summary of the obtained results will be provided, and future improvements will be presented.

7.1 Conclusions

An *Artificial Pancreas* has the potential to revolutionize the management of T1DM. It offers numerous advantages, such as improved blood sugar control, a reduced risk of hypoglycemia and an enhanced quality of life. Despite some existing challenges, the outlook for this medical device is promising. Continued research and development are likely to bring additional benefits, with advancements in closed-loop systems and the possibility of an AP implant on the horizon. The aim was to contribute to the research and development of valid and innovative control systems for AP_{sys}. The work progressed through two main phases: in the first phase, two linear control algorithms were developed using the Least Squares and Set-Membership techniques. In the second phase, the attention was directed towards designing a novel nonlinear controller, employing recurrent neural networks. Both the linear and nonlinear controllers have been trained using input-output data collected through the use of an accurate diabetic patient simulator available in the literature. Through extensive testing and validation, the results demonstrate that, while linear controllers perform adequately in maintaining safe glucose levels, the nonlinear controller significantly outperforms them. The neural network-based approach provides more accurate and responsive insulin delivery, offering a superior and more robust solution for AP systems. These findings highlight the potential for advanced neural network algorithms to become a new milestone for diabetes management through improved automated control mechanisms.

7.2 Future Improvements

While the current framework demonstrates satisfactory performance and reliability, there are numerous opportunities for enhancements that could elevate its overall effectiveness and efficiency. This section outlines various potential improvements and refinements to the architecture. Below is reported a comprehensive list of proposed variations:

- **Glucose-insulin system's model:** as mentioned in Section 2.3, there exist two models, besides the one currently employed, that describe the glucose-insulin system. Therefore, one potential modification would be to shift the *UVAPadova* model and test the developed controllers to validate them on an implementation different from the one used for their design. This change would allow verification of the control strategy's portability across different models, making it applicable to a wider range of future studies.
- **Dataset:** for what concerns the dataset, an improvement that could be made would be the use of a dataset composed of data collected from real patients. Therefore, it could be possible to create a mixed dataset (composed of both real and synthetic data) with the aim of increasing the network's generalization capabilities, validating it subsequently employing real data. This approach could demonstrate that the model (trained with the aid of limited real data) can be applied to real patients. This achievement would significantly simplify future development steps, partially solving the problem of requiring large amount of real medical data.
- **Neural network:** through additional and detailed research on literature in the field of machine learning, alternative models of RNNs could be identified, that might be more suitable for the task, thus achieving potentially better and more precise results.
- **Online training:** the strategy of online training can mitigate issues stemming from the prolonged use of a neural network (i.e., performance degradation over time and exposure to continuous new stimuli) and its application to each specific patient. Hence, the pre-trained network (using the aforementioned dataset) could be utilized with online training, repeated at specific intervals, as a method for both fine-tuning the parameters to better suit individual needs and compensating for potential degradation.

Appendix A

Sorensen model equations

Glucose equations:

- *Brain:*

$$\frac{dG_{BV}}{dt} = \frac{Q_B^G}{V_{BV}^G}(G_H - G_{BV}) - \frac{V_{BI}}{V_{BV}^G T_B}(G_{BV} - G_{BI}) \tag{A.1}$$

$$\frac{dG_{BI}}{dt} = \frac{1}{T_B}(G_{BV} - G_{BI}) - \frac{\tau_{BGU}}{V_{BI}^G} \tag{A.2}$$

- *Heart and lungs:*

$$\frac{dG_H}{dt} = \frac{1}{V_H^G}(Q_B^G G_{BV} + Q_L^G G_L + Q_K^G G_K + Q_P^G G_{PV} - Q_H^G G_H - \tau_{RBCU}) \tag{A.3}$$

- *Liver:*

$$\frac{dG_L}{dt} = \frac{1}{V_L^G}(Q_A^G G_H + Q_G^G G_G - Q_L^G G_L + \tau_{HGP} - \tau_{HGU}) \tag{A.4}$$

- *Gut:*

$$\frac{dG_G}{dt} = \frac{Q_G^G}{V_G^G}(G_H - G_G) - \frac{1}{V_G^G}(\tau_{meal} - \tau_{GGU}) \tag{A.5}$$

- *Kidney:*

$$\frac{dG_K}{dt} = \frac{Q_K^G}{V_K^G}(G_H - G_K) - \frac{\tau_{KGE}}{V_K^G} \tag{A.6}$$

- *Periphery (tissue and muscles):*

$$\frac{dG_{PV}}{dt} = \frac{Q_P^G}{V_{PV}^G}(G_H - G_{PV}) - \frac{V_{PI}^G}{V_{PV}^G T_P^G}(G_{PV} - G_{PI}) \tag{A.7}$$

$$\frac{dG_{PI}}{dt} = \frac{1}{T_P^G}(G_{PV} - G_{PI}) - \frac{\tau_{PGU}}{V_{PI}^G} \tag{A.8}$$

Insulin equations:

- *Brain:*

$$\frac{dI_B}{dt} = \frac{Q_B^I}{V_{BV}^I}(I_H - I_{BV}) \tag{A.9}$$

- *Heart and lungs:*

$$\frac{dI_H}{dt} = \frac{1}{V_H^I}(Q_B^I I_B + Q_L^I I_L + Q_K^I I_K + Q_P^I I_{PV} - Q_H^I I_H + i(t)) \tag{A.10}$$

- *Liver:*

$$\frac{dI_L}{dt} = \frac{1}{V_L^I}(Q_A^I I_H + Q_G^I I_G - Q_L^I I_L + \tau PIR - \tau LIC) \tag{A.11}$$

- *Gut:*

$$\frac{dI_G}{dt} = \frac{Q_G^I}{V_G^I}(I_H - I_G) \tag{A.12}$$

- *Kidney:*

$$\frac{dI_K}{dt} = \frac{Q_K^I}{V_K^I}(I_H - I_k) - \frac{\tau_{KIC}}{V_K^I} \tag{A.13}$$

- *Periphery (tissue and muscles):*

$$\frac{dI_{PV}}{dt} = \frac{Q_P^I}{V_{PV}^I}(I_H - IPV) - \frac{V_{PI}^I}{V_{PV}^1 T_P^I}(I_{PV} - I_{PI}) \tag{A.14}$$

$$\frac{dI_{PV}}{dt} = \frac{1}{T_P^I}(I_{PV} - I_{PI}) - \frac{\tau_{PIC}}{V_{PI}^I} \tag{A.15}$$

Glucagon equation:

$$\frac{dN}{dt} = 0.0916(\tau_{PNR} - N) \tag{A.16}$$

Additional equations:

$$\frac{dA_1}{dt} = \frac{1}{25}[1.21 - 1.41\, tanh(1.66(\frac{I_L}{21.43})) - A_1] \tag{A.17}$$

$$\frac{dA_2}{dt} = \frac{1}{65}[\frac{2.7\, tanh(0.39N)}{2} - A_2] \tag{A.18}$$

$$\frac{dA_3}{dt} = \frac{1}{25}[2\, tanh(0.55\frac{I_L}{21.43}) - A_3] \tag{A.19}$$

Bibliography

[1] Iqra Shafeeq Mughal, Luca Patanè, and Riccardo Caponetto. «A comprehensive review of models and nonlinear control strategies for blood glucose regulation in artificial pancreas». In: *Annual Reviews in Control* 57 (2024), p. 100937. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2024.100937 (cit. on p. 4).

[2] Linda A DiMeglio, Carmella Evans-Molina, and Richard A Oram. «Type 1 diabetes». In: *The Lancet* 391.10138 (2024/06/04 2018), pp. 2449–2462. DOI: 10.1016/S0140-6736(18)31320-5. URL: https://doi.org/10.1016/S0140-6736(18)31320-5 (cit. on p. 5).

[3] Sun Joon Moon, Inha Jung, and Cheol-Young Park. «Current Advances of Artificial Pancreas Systems: A Comprehensive Review of the Clinical Evidence». en. In: *Diabetes Metab J* 45.6 (Nov. 2021), pp. 813–839 (cit. on p. 7).

[4] Jeremy H. Pettus, Fang Liz Zhou, Leah Shepherd, Ronald Preblick, Phillip R. Hunt, Sachin Paranjape, Kellee M. Miller, and Steven V. Edelman. «Incidences of Severe Hypoglycemia and Diabetic Ketoacidosis and Prevalence of Microvascular Complications Stratified by Age and Glycemic Control in U.S. Adult Patients With Type 1 Diabetes: A Real-World Study». In: *Diabetes Care* 42.12 (Sept. 2019), pp. 2220–2227. DOI: 10.2337/dc19-0830. eprint: https://diabetesjournals.org/care/article-pdf/42/12/2220/528751/dc190830.pdf. URL: https://doi.org/10.2337/dc19-0830 (cit. on p. 7).

[5] Giacomo Cappon, Martina Vettoretti, Giovanni Sparacino, and Andrea Facchinetti. «Continuous Glucose Monitoring Sensors for Diabetes Management: A Review of Technologies and Applications». en. In: *Diabetes Metab J* 43.4 (Aug. 2019), pp. 383–397 (cit. on p. 8).

[6] Ahmad Haidar, Laurent Legault, Virginie Messier, Tina Maria Mitre, Catherine Leroux, and Rémi Rabasa-Lhoret. «Comparison of dual-hormone artificial pancreas, single-hormone artificial pancreas, and conventional insulin pump therapy for glycaemic control in patients with type 1 diabetes: an open-label

randomised controlled crossover trial». In: *The Lancet Diabetes & Endocrinology* 3.1 (2024/06/04 2015), pp. 17–26. DOI: 10.1016/S2213-8587(14)70226-8. URL: https://doi.org/10.1016/S2213-8587(14)70226-8 (cit. on p. 9).

[7] Diego Regruto Vito Cerone. «La patologia del diabete mellito: Modello fisiologico delle dinamiche glucosio insulina.» In: (2005-2006) (cit. on pp. 11, 28, 29).

[8] Marcello Pompa, Simona Panunzi, Alessandro Borri, and Andrea De Gaetano. «A comparison among three maximal mathematical models of the glucose-insulin system». In: *PLOS ONE* 16.9 (Sept. 2021), pp. 1–43. DOI: 10.1371/journal.pone.0257789. URL: https://doi.org/10.1371/journal.pone.0257789 (cit. on p. 14).

[9] Zhongsheng Hou, Ronghu Chi, and Huijun Gao. «An Overview of Dynamic-Linearization-Based Data-Driven Control and Applications». In: *IEEE Transactions on Industrial Electronics* 64.5 (2017), pp. 4076–4090. DOI: 10.1109/TIE.2016.2636126 (cit. on p. 18).

[10] V. Cerone, D. Regruto, and M. Abuabiah. «Direct data-driven control design through set-membership errors-in-variables identification techniques». In: *2017 American Control Conference (ACC)*. 2017, pp. 388–393. DOI: 10.23919/ACC.2017.7962984 (cit. on pp. 19, 40).

[11] Nicola Paoletti, Kin Sum Liu, Hongkai Chen, Scott A. Smolka, and Shan Lin. «Data-Driven Robust Control for a Closed-Loop Artificial Pancreas». In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 17.6 (2020), pp. 1981–1993. DOI: 10.1109/TCBB.2019.2912609 (cit. on p. 31).

[12] Sofia Peacock, Isolda Frizelle, and Sufyan Hussain. «A Systematic Review of Commercial Hybrid Closed-Loop Automated Insulin Delivery Systems». In: *Diabetes Therapy* 14.5 (2023), pp. 839–855. DOI: 10.1007/s13300-023-01394-5. URL: https://doi.org/10.1007/s13300-023-01394-5 (cit. on p. 33).

[13] V. Cerone, S. M. Fosson, S. Pirrera, and D. Regruto. *A new framework for constrained optimization via feedback control of Lagrange multipliers*. 2024. arXiv: 2403.12738 [math.OC]. URL: https://arxiv.org/abs/2403.12738 (cit. on p. 43).