Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2016

Project Title:	**An Adaptive Multi-Model Recommender Engine for Diabetes Mellitus**
Student:	**Hao Ding**
CID:	**00734091**
Course:	**EE4T**
Project Supervisor:	**Professor Demiris, Y.K.**
Second Marker:	**Dr Mikolajczyk K.M.**

Abstract

This project proposed a recommender engine to assist the decision making and the self-management in diabetes mellitus healthcare. The recommender system is able to combine the knowledge of the system and the healthcare expert in action recommendation. The adaptive mechanism allows the recommender to learn from the patient behaviour and to show improved performance in glycemic control over time. The recommender system utilises a distributed and hierarchical multi-model architecture proposed at Imperial College London in action generation and selection. With a virtual patient simulator, the recommender system is tested by recommending insulin injection and glucose ingestion to introduce the glycemic control on the virtual patient. Experiment results have shown satisfying results in two scenarios: using the system alone in recommendation, or allowing the system to include the knowledge of the expert in action recommendation. This thesis discusses the design of the recommender system and evaluates the performance of the overall architecture, the design choices and the parameter settings in the recommender.

Acknowledgements

I would like to express my deepest appreciation to Professor Yiannis Demiris, as my interviewer to this university and my final year project supervisor. I would like to thank him for invaluable advice and unstinting support throughout the project. Also, a big thanks to Dr. Antoine Cully and Dr. Miguel Sarabia for their technical guidance and insightful advice through this project.

I also wish to thank all my friends for their support and accompany during my study at Imperial College. I wish them every success and every happiness in their future endeavours. Finally, I would like thank my family for their unconditional love over the years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Objectives

Diabetes Mellitus is a metabolic chronic disease which causes long-term complications. However, the complications can be less often and less severe with effective glycemic management [8, 9]. In the current mobile healthcare industry, data tracking is the main functionality in most applications, which is less effective compared to the multifaceted intervention in diabetes management [10]. Therefore, this project aims to provide a recommender engine which follows the 'proactive' intervention mechanism to assist patients in diabetes management. The recommender can potentially be used as the back-end for diabetes management apps.

It is important for the recommender engine to be adaptive, which is the key in the long-term diabetic healthcare. Chronic disease usually involves long-term medical treatment where fundamental changes in user physical and psychological conditions are likely to occur. A previous study has shown that the adaptive mechanism is likely to improve treatment outcomes in chronic disease healthcare [11]. This project aims to introduce the feature of adaptivity in the recommender to benefit the long-term diabetic healthcare.

1.2 Project Achievements

The final deliverable of the project is an adaptive recommender system which utilises the built-in treatment strategies and the knowledge of healthcare professionals to make recommendations on diabetes management. The recommender can potentially be the back-end server for mobile apps. The major achievements of the project are summarised as the following:

- The recommender system can be used alone to suggest actions to diabetic patients. It uses the built-in knowledge, i.e. combinations of common treatment methods, and patient prediction models, to recommend actions.
- The recommender system is able to include the knowledge of experts in recommendation. Assuming an expert is available to demonstrate a treatment method, the recommender system adapts itself throughout the demonstration. After the demonstration, the recommender alone uses the combined knowledge of the system and the expert in later recommendations.
- The recommender always learns from patient glycemic response and shows improved performance in glycemic control over time.

Implementations and experiment results are available in later chapters.

1.3 Project Scope

This project puts a specific emphasis on the glycemic control which is the basic barometer in diabetes management. The project focuses on managing glucose level through insulin and meal suggestions, at typical meal and bed times in daily life. The recommender system is evaluated on a virtual patient model based on the glucose-insulin dynamics proposed by Dalla Man et al [12, 13], which have been widely used in recent diabetes research and has been included in the Matlab SimBiology Toolbox [14]. A similar Java version has been implemented to test the recommender through real-time patient simulation and action intervention. The proposed recommender has a great potential to extend from glycemic control to other aspects in diabetes management such as physical exercise and psychological management. Research on future developments have been included in the last chapter.

1.4 Structure of the Thesis

This thesis consists of requirements capture, background context, design and implementation, evaluation, results and future work.

The 'Requirements Capture' shows the overall and component deliverables. It aims to help readers capture the key points of the thesis.

The 'Background and Context' chapter introduces chronic disease and diabetes mellitus, and how these diseases are tackled in the current mobile health industry. It also introduces the similar work in recent years and the 'HAMMER' architecture, which is the foundation of this project.

The 'Design and Implementation' chapter provides the detailed explanation on the recommender development, including the overall architecture implementation, model training, etc. The evaluation section shows the methodology of the experiments performed on the virtual patient and discusses the effect of parameter settings.

Finally, a summary of the project achievements and future development are included in the last chapter. It extends the potential application of the recommender system to other fields in diabetes management, such as physical activity in glycemic control, psychological management, self-management ability and user engagement.

Chapter 2

Background and Context

2.1 Introduction

This section introduces the background knowledge relevant to this thesis. It introduces the situation of chronic disease treatment with a focus on diabetes mellitus. Then a broad overview of the current mobile healthcare industry is provided and comparisons on the existing diabetes management solution have been made. Mobile healthcare has been a popular topic in recent years. The self-management methods for patients with chronic diseases have also been actively researched and diabetes mellitus is of particular interest since it can be reasonably managed through daily routine activities. Furthermore, this section includes some of the common recommendation techniques used for healthcare applications and research purposes. In particular, this section introduces the basic concepts of the 'HAMMER' Architecture, a multi-model hierarchical structure, used in this project.

2.2 Diabetes Mellitus: Disease and Management

2.2.1 Chronic Disease and Diabetes Mellitus

Chronic diseases is defined as: "long in durationoften with a long latency period and protracted clinical course; of multi-factorial aetiology; with no definite cure; gradual changes over time, asynchronous evolution and heterogeneity in population susceptibility [15]." Examples of chronic diseases include diabetes mellitus, cancer, viral diseases, arthritis and asthma. Often accompanying with chronic diseases is chronic illness, which can be described by the long-term experience of physical or psychological disturbance [16]. It is estimated that approximately 35 million out of the 58 million death worldwide were caused by chronic diseases and in the next decade, the death figure resulting from chronic diseases will rise up to 388 million [17].

Diabetes mellitus is recognised as a type of chronic disease. The World Health Organisation has defined diabetes by "the level of hyperglycaemia giving rise to risk of microvascular damage (retinopathy, nephropathy and neuropathy)" [18]. Diabetes is the leading cause of heart disease, stroke, kidney failure, blindness and it is projected to be the seventh leading cause of death in 2030 [19, 20]. Diabetes is costly and the total cost of diabetes in the United States in 2007 is estimated to be around $174 billion [19].

Diabetes is commonly categorised into three types according to its causes:

- Type 1 Diabetes Mellitus (T1DM) is resulted from the human immune system destroying beta cells in the pancreas, which produces insulin and introduces glycemic control [19]. It is necessary for patients with type 1 diabetes to inject insulin and therefore, the type 1 diabetes mellitus is also referred as insulin-dependent diabetes mellitus (IDDM) [19].
- Type 2 diabetes, also named non-insulin-dependent diabetes mellitus (NIDDM) is caused by the pancreas losing ability to produce insulin effectively [19]. NIDDM accounts for more than 90% diabetes cases in the world and it is usually associated with obesity, physical inactivity and family history [19].
- Gestational diabetes, which is defined as glucose resistance during pregnancy. It is also associated with obesity and family history of diabetes[19].

Recently, the self-management of patients with chronic disease has become a popular topic [21]. Clinical trials have shown that the self-management education improves the chronic disease outcome and reduces costs in some circumstances [21, 22]. Regarding diabetes self-management education, some common measures include self-monitoring of blood glucose level, insulin injection and carbohydrate ingestion rules [23]. Studies have shown that the diabetes self-management education is able to reduce the hospitalisation of patient with diabetes and the self-management is most effective with the professional suggestions on medication and the reinforced learning process with the assistance from healthcare professionals [23]. This project has suggested a method to provide on-going assistance in diabetes self-management process and includes the potential to incorporate professional adviser into the long-term self-management process.

2.2.2 Insulin Medication

Insulin is a type of hormone, produced by beta cells in the pancreas to help the physical body to use glucose for energy [24]. Sir Edward Schafer, a professor formerly graduated from University College London, first used the word 'insulin' and describe the its effect in his book in 1916 [25]. Sir Frederick Grant Banting won the Nobel Prize in Physiology in 1923 for using insulin on humans [26]. The early use of insulin are purified animal-sourced insulin until the first synthesised human insulin was produced in 1978 [27].

The international unit of insulin is in IU, equivalent to 34.7 g pure crystalline insulin, corresponding to the old United States Pharmacopeia unit for insulin (U), which is measured as the amount to reduce the blood glucose to 45 mg/dl. Insulin is categorised into various types based on its characteristics:

- Fast-acting insulin, such as insulin lispro and glulisine, has an onset time of 5 to 15 minutes and a duration of around 3 hours.
- Short-acting insulin, such as regular insulin, has an onset time of around 30 minutes and a duration of approximately 5 hours
- Intermediate-acting insulin, such as NPH insulin, has an onset time of 1 to 3 hours and is active for 16 to 24 hours
- Long-acting insulin, like insulin glargine, has an onset time of one and a half hours, has no peak time and lasts for about 24 hours.
- Ultra-long acting insulin, currently only includes analogue degludec, lasts for more than 24 hours

- Mixed insulin, combinations of fast, short and long acting insulin. It has various effects depending on the mixture.

Effects of insulin can also be measured in glucose infusion rate, which describes how fast the human body consumes carbohydrates, shown in the figure 2.1. The improved glucose infusion rate shows the effect of insulin on reducing blood glucose level.

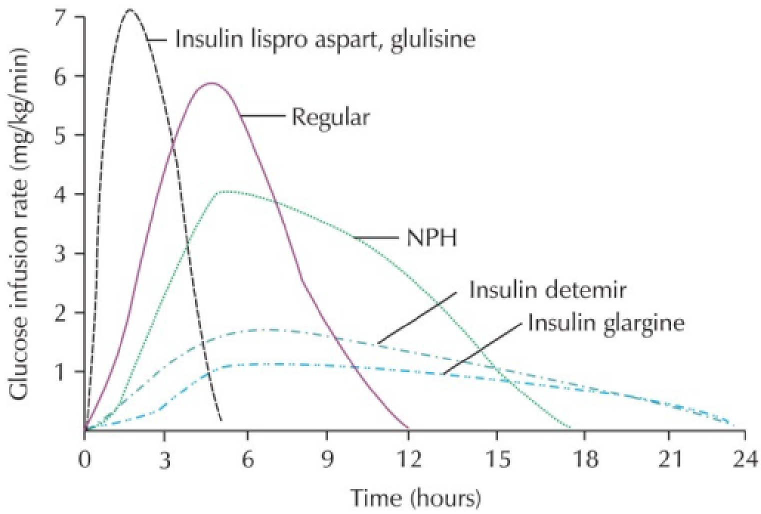


Figure 2.1: Glucose infusion rate of different types of insulin [3].

2.2.3 Diabetes Meal Plans

Diabetic diet is commonly used in glycemic control to reduce hyperglycemia and clinical trials have shown that diet has beneficial metabolic effect in glycemic control [28, 29]. However, how the diet plan should be designed is controversial. The American Diabetes Association has recommended a daily carbohydrate intake of 130 grams/day which provides enough energy for the central nervous system without relying on consuming protein or fat [30]. While a study has claimed that no more than 40% of daily caloric intake should come from carbohydrate for patients in diabetic diet [31]. Another study has shown a high-fiber, 75% carbohydrate diet is likely to benefit diabetic patients [32].

2.3 M-Health and Diabetes Mellitus

M-health is the abbreviation of mobile health. It can be defined as utilising mobile devices in healthcare [33]. It has been increasingly popular to use m-health technology in managing chronic diseases such as diabetes. Understanding the current industry has contributed to the motivation of this project. The m-health market is seen as an opportunity for the idea of the thesis to commercialise. The recent situation of m-health industry and particularly the diabetes mobile solutions are discussed in the following sections.

2.3.1 Current m-Health Industry

M-health is not a new idea. In 2007, the number of mobile phone subscribers is estimated to be 3.1 billion, accounting for 47% of the world population [34]. The broader coverage of mobile devices might contribute to the popularity of m-health. The m-health industry generated a revenue of approximately $718 million worldwide in 2011 [35] and around one fifth of the smartphone phone owners have at least one health app. A recent study on the sustainable m-health strategy has concluded that m-health will inevitably play an increasingly crucial role in future [36]. There is significant potential for mobile technology to solve problems in healthcare area.

M-health can be utilised to support healthcare in a variety of ways. The United Nation Foundation and the Vodaphone Foundation have provided a summary of m-health applications [37]:

- Education and Awareness
- Remote Data Collection
- Remote Monitoring
- Communication and Training for Healthcare Workers
- Disease and Epidemic Outbreak Tracking
- Diagnostics and Treatment Support

2.3.2 Comparison between m-Health Products

This section compares some commercialised m-health products following two different mechanisms, either 'passive' or 'proactive'. In the m-health market exist a wide range of products and a large number of them are focused on data collection. The Health app, a pre-installed app for iPhones, acts as a personal data hub. It can be connected to wearable devices and other third-party accessories to collect and analyse data. Another app, the AliveCor Mobile ECG, measures and records user heart rate. This type of product emphasises on data collection and analysis, which provides information for user to track and understand their health conditions. The product itself does not provide any suggestion, but requires the user to take further steps after presenting them with data.

Some other products follow different mechanisms. Those apps not only collects and analyses data, but also provides feedback to users and tells them the actions to take. For example, the 'HidrateSpark', tracks the amount of water consumed and reminds users to drink water in order to meet the customised daily water goal. The 'Pip', estimates the user's stress level by measuring the Electrodermal Activiey (EDA). It uses gamification to encourage users to reduce their stress levels. Such m-health applications provide proactive suggestions or solutions after obtaining user data.

This project follows the 'proactive' design mechanism, actively providing suggestions to the user-end. After collecting and analysing user data, the recommender engine generates the optimal action to take in order the reach the target state. The details of the recommending mechanism are explained in the following section. In the past project in Jamaica, guidance and instructions are considered important in preventing chronic lifestyle diseases. Therefore, the proactive design of m-health system is likely to lead to higher healthcare efficiency and more engagement of patients in improving the lifestyle of patients with chronic diseases.

2.3.3 M-health Solutions for Diabetes Mellitus

Self-management of Diabetes Mellitus with the assistance of health advisers has shown improvement on the living outcomes [23]. There are a wide range of m-health frameworks, platforms and applications targeting diabetes management on the current market.

M-Health applications are commonly used for data storage and analysis purpose in diabetes management. There are a range of diabetes management applications in the Apple store providing services such as diabetes diary, glucose record and carbohydrate tracking. Examples of software applications emphasising data tracking include ‘mySugr’, ‘Diabetes Pilot Pro’, ‘Glooko’ and ‘Diabetes in Check’. The mentioned applications usually integrate with hardware monitoring devices such as Fitbit and glucose monitors to enable the automatic data logging. Data streams are used to help patients tracking the disease condition and to assist clinicians in decision-making.

M-Health is capable of playing a more significant role in the clinical decision-making process by embedding clinical knowledge in software and formulating a decision based on the real-time patient data. The decision support system can be used collaboratively with the clinician’s decision and the combined effort is likely to lead to better outcome than the decision without collaboration [38]. Such clinical decision support system was initially developed to assist clinicians during hospital visits and now these applications are capable of automatically making decisions based on patient status, like glucose and physiological measures [38]. Personalised decision making based on user preference and individual differences is also available under such mechanism [38].

In 2013, a group of researchers have analysed the current status of the diabetes self-management applications [39]. They found that the data monitoring function is most common in diabetes management applications. 100% of diabetes apps in the Apple store contain blood glucose tracking. Other data monitoring tasks include weight, diet, medication, physical activity, etc. Compared with data monitoring functions, the decision support applications are less commonly seen and only 17% of relevant apps in the Apple store have included such function [39]. This study has also concluded that the efficiency of intervention can be improved by considering the expectations of patients and needs of care providers [39].

2.4 Decision Support Systems and Recommendation Engines

2.4.1 Introduction

This section introduces recent research on the Clinical Decision Support Systems (CDSS) in general healthcare and for diabetes management. As mentioned in the previous section, decision support system remains a minority group in healthcare market but recent study have shown increasing interest in such system and found exciting outcomes using such strategy in clinical healthcare.

As a subsidiary of the decision support system, the recommender system is not only limited to incorporate knowledge base, but also make decisions suggestions. This section also discusses the state-of-the-art of recommender systems in diabetes management.

2.4.2 The CDSS Industry

CDSS have been increasingly popular in clinical healthcare [40]. Research has shown that such system is able to reduce medication error [41], improve the doctors' confidence in making decision [42] and improve patient medication adherence [43]. Compared to other methods, the decision support system is likely to have higher efficiency and long-lasting effect in clinical decision support [44].

However, decision support system cannot guarantee an improvement in clinical practice. A systematic review has pointed out that around one third of such systems did not show significant improvement in practice [40]. Also, it has been difficult to explain the success and failure of the decision support system [45, 46]. Further research is needed to develop our understanding about how such systems precisely change the behaviour care providers but the potential benefit brought by decision support system is still significant.

2.4.3 Decision Support Systems in Healthcare

Though CDSS is likely to be effective in chronic disease management, it is difficult to implement the system and to apply the idea in practice. Some trial programs have been set up to verify the feasibility of such mechanism.

CDSS for diabetes management has been developed and tested in Vermount, the U.S. The main purpose of the system is to collect patient information and generate accurate flow sheet, mailed or faxed reminder and population status for patients and healthcare professionals [47]. The support system provides appropriate action suggestions to care providers based on patient reports and actively engages the patients in self-management through developing their understanding on diseases [47]. Researchers in that experiment have founded improvement in testing frequency of patients and the long-term cost of implementing such decision support system is around $1 per patient per month [47].

Another experiment has integrated diabetes clinical guidelines into the decision support system and the recommendations generated by the system have doubled the clinical compliance [48]. It is also concluded in the report that incorporating clinical guidelines can improve the quality of medical care and help clinicians to meet care standards [48].

2.4.4 Recommender Systems in Diabetes Management

The project METABO is a collaborative project between nine European countries targeting 'Controlling Chronic Diseases related to Metabolic Disorders' [49]. Researchers of the METABO project has proposed a lifestyle recommender system for the self-management of diabetes patients. They aim to improve the lifestyle of patients with diabetes mellitus by providing personalised health plans and daily activity recommendations. The report proposed to use collaborative filter or Kalman filter to determine the thresholds for hypo- and hyper-glycemia. The project group argued that such clinical recommender requires particular attention to details compared to commercial recommender systems [49]. Another study has integrated the collaborative filter to process high-dimensional patient data to reduce the complexity and processing time in recommender systems [50].

A study has discussed the design of 'mobile peer support' system for patients with diabetes, which enhances the self-management efficacy through sharing information with patients with similar profiles [51]. The healthcare education system then delivers personalised education content to various groups of patients [51].

A personalised food recommender to assist diet control of patients with diabetes has been designed [52]. The recommender system constructed the food ontology and utilises the knowledge-based rules to suggest food for patients in diet.

2.5 Recommendation Action Selector: The 'HAMMER' Architecture

2.5.1 Introduction

This section introduces the 'HAMMER' architecture, which the recommender is operating on. The HAMMER is the abbreviation for the 'Hierarchical Attentive Multi Models for Execution and Recognition' [5], proposed by Professor Yiannis Demiris from Imperial College London in 2006. This is a distributed and hierarchical multi-model structure which heavily uses the concepts of motor control. Before going deep into the implementation of recommender, it is necessary to explain the components, the structure and the working mechanism of the 'HAMMER'. Note that this section introduces the general 'HAMMER' architecture, how the 'HAMMER' is implemented and utilised in diabetes recommender is discussed in the design and implementation section.

2.5.2 Overview

The basic functionality of the 'HAMMER' is recommending actions and learning from experts. Such function is realised by pairs of inverse and forward models, which are the elementary units in the 'HAMMER'. Multiple model pairs can be simulated with the same target, which allows model pairs to compete in recommending actions. Action selector and confidence allocation mechanisms step in to make decisions on recommendations. Furthermore, a number of such model pairs are organised into the hierarchy and execute at different abstract levels. Details of hierarchical structure will be discussed in the following sections.

2.5.3 Inverse Model and Forward Model

In control theory, inverse models simulate the behaviour of a natural process and invert the causal flow [53]. Inverse models generate motor commands which are likely to lead the object from the current state to the target state (Specifying target state is optional). Forward models, on the contrary, take the current state and the action commands to predict the resulting future state. A simple diagram that summarised the functions of inverse and forward models as shown in figure 2.2.

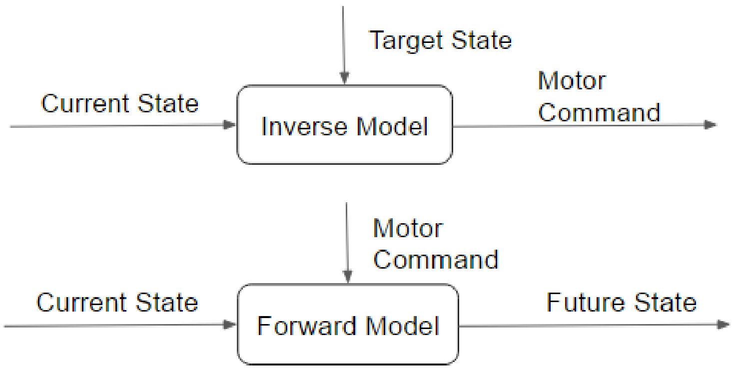


Figure 2.2: The structures of an inverse model(top) and a forward model(bottom).

The inverse model generates motor commands while the forward model imitates the response of the system. Linking an inverse model and a forward model gives the picture of the motor control system, shown in figure 2.3. Wiring the outputs of inverse models to the inputs of forward models allows forward models to predict future outcomes following the inverse model commands. Assuming the motor state is available, the models can be updated and improved over time. By comparing the predicted state of the forward model and the actual state of the system, detection error is produced and can be used to modify the confidence levels of inverse models. Forward models can be updated with the latest system response to enable adaptive prediction of system states.

In the diabetes recommender system, inverse models simulate various therapies that might benefit diabetic patients. Forward models learn patient responses after taking therapies and predict the likely outcomes of the patient if suggested actions from inverse models are executed.

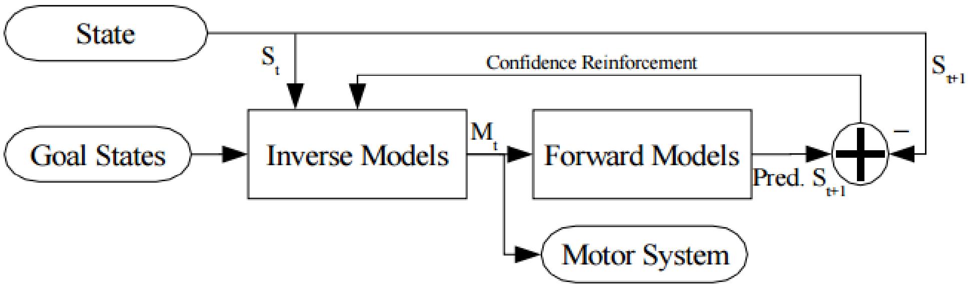


Figure 2.3: The mechanism of a inverse-forward model pair [4].

2.5.4 The Multi Model Hierarchy

Introduction

The multi model hierarchy is a key feature of the 'HAMMER'. Organising the inverse-forward model pairs hierarchically allows the 'HAMMER' to perform more complex tasks. Hierarchies of the model pairs can be horizontal, vertical or combined. The intuition behind each structure is different.

The Parallel Structure

The horizontal structure of the 'HAMMER' means organising model pairs parallel and allowing them to work on the same aspect, for instance, providing various diet plans. Model pairs are operating at the same level. Inverse models generate various action commands while forward models predict corresponding outcomes. A sample diagram is shown in figure 2.4.

In figure 2.4, it shows the parallel arrangement of model pairs. Inverse models B1 to Bn take the system state at time t and produce motor commands M1 to Mn. Each inverse model is paired with a forward model F1 to Fn, which predicts the next state at t+1, shown as P1 to Pn in the diagram. The predicted states are compared with the actual state and the comparison errors E1 to En are produced and used to adjust the confident levels of inverse models.

The parallel hierarchy is the most fundamental structure in the 'HAMMER' architecture. Multiple inverse models are generating the actions that might transfer the current state to the desired one. The intuition behind the parallel hierarchy is the competition between model pairs. Various model pairs generate motor commands competitively and the 'best controller' is labelled

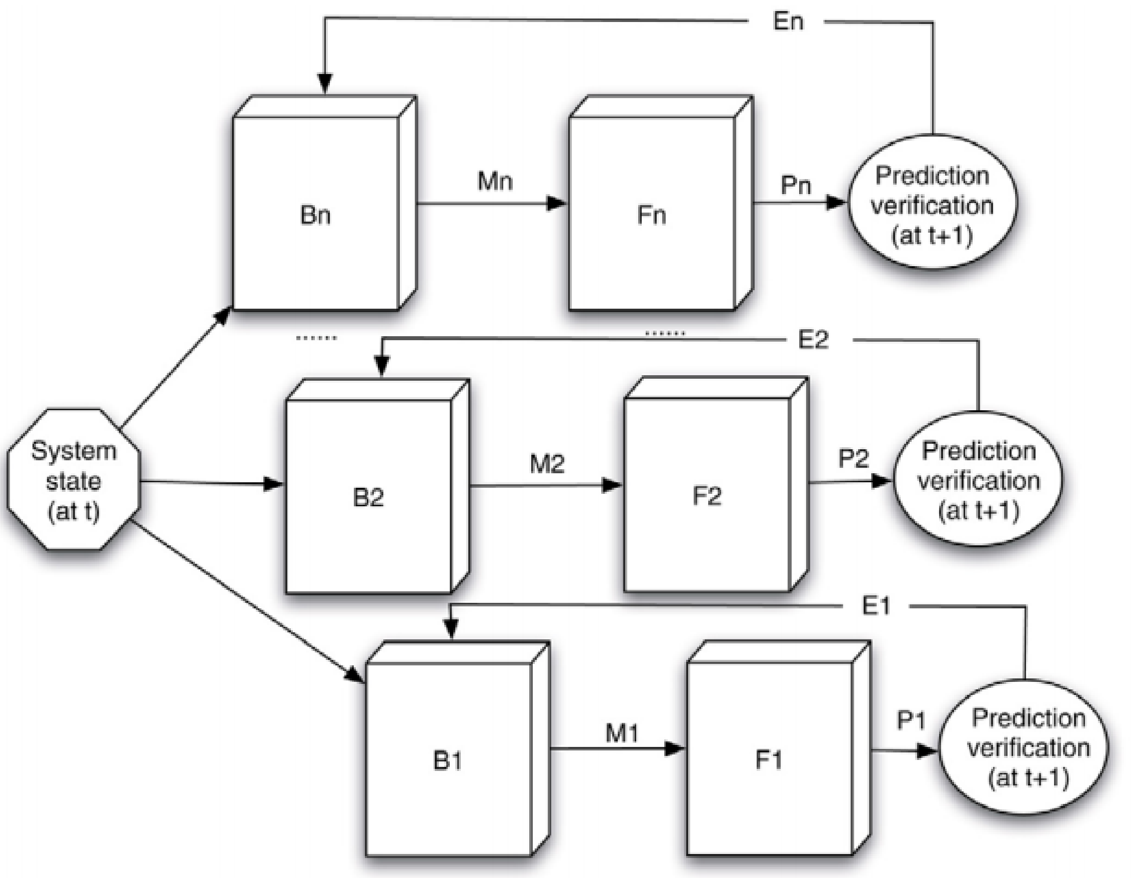


Figure 2.4: The parallel hierarchy of model pairs [5].

with the highest confidence level, assuming a good forward model. Using the parallel hierarchy in action recommendation requires an action selector which chooses the most suitable action to deliver. The selected model does not necessarily have the highest confidence level. The different possible action selection mechanisms are discussed in the next chapter.

In other implementation of the 'HAMMER', The parallel hierarchical architecture has been implemented and tested with robot simulation [54] and robot platforms [5]. The architecture has obtained satisfying result in understanding actions in the imitation learning problem [55]. In the diabetes recommender system, the parallel structure enables the competition between various treatment strategies and allows the selector to choose the most suitable one to deliver to the patient.

The Vertical Structure

The vertical hierarchy of HAMMER is more complicated than the parallel structure. The vertical relationship between model pairs can be defined and understood in a variety of ways. In general, the higher-level models are defined as models in higher abstraction levels, which gives more flexible and more complex descriptions than low-level models.

Experiment has shown that the abstract mechanism has outperformed the mechanism without abstraction [56]. In the experiment, the 'HAMMER' structure is used to recognise the actions of moving an object from one place to the other. The higher level model is described as a sequence of low-level models and aims to find the exact matching sequence of actions. However, it is likely that the action of the demonstrator cannot be recognised owing to different morphology or missing out actions. The abstraction mechanism solves the problem by allowing actions not recognised and checking future potential actions [56]. Thus, the sequence of detailed actions is described by a fewer steps of recognised actions, resulting in higher recognition accuracy [56]. The abstraction mechanism above is achieved by sacrificing specificity and ignoring missing actions in sequence recognition [56].

In the diabetes recommender system, higher abstraction level means considering more aspects and accessing a wider range of information in treatment process. High-level models generate complex strategies based on low-level models. Detailed are discussed in the 'Design and Implementation' chapter.

The Combined Hierarchy

The horizontal and vertical organisations of model pairs form a more complicated hierarchy with a combined feature of both arrangements. The horizontal organisation allows model pairs to operate competitively in recognition and recommendation. The vertical arrangement utilised the information from the low-level model pairs to operate in a high abstract level. The combination of both structures give rise to more flexibility and more power in action recommendation.

2.5.5 The Attention Mechanism

Since the attention mechanism is only implemented to show the idea but not used in the final deliverable, the attention mechanism is only briefly explained in section. The 'HAMMER' is inspired by the simulation theory of action perception of the human brain [5]. Though there is a variety of definitions on human attention, from an engineering point of view, attention can be considered as prioritising the resource allocation assuming resource extraction and transmission consumes computational and time resources. The attention mechanism is designed to optimise the action perception process by allowing models to subscribe a subset of user information and then prioritise the model requests. The attention mechanism in the diabetes recommender is implemented as a ordered list of information that the recommender subscribes.

2.5.6 User Modelling and Learning Algorithms

In order to maintain the long-term engagement of patients in chronic disease healthcare, the recommender system is designed to be adaptive and be able to automatically adjust to the changes happened to users. For example, the system can learn the changes in users preference, physical and mental condition, personality and interests. To provide long-term personalised service, the recommender has to change its behaviour based on the changes of user behaviour. Research has shown that the adaptive mechanism is preferred in human-robot interaction[57]. Machine learning is becoming increasingly popular in the user modelling field[58].

User preference is one of the most commonly used information in learning about new users. There are a range of machine learning techniques that are beneficial to user modelling. The collaborative filtering recommender system is good at selecting items that the new user is likely to have an opinion about [59, 60], especially in the case that the information of the new user is limited. This is likely to be the situation of chronic disease treatment for young children. Bayesian network are commonly used in distinguishing user abilities and learning styles[61, 62, 63, 64]. Feedback based on learning style has shown to contribute to better learning outcomes[64]. Support vector machine has been used to recommend learning resource to user[65]. Regression models have been used to learn the user preference on performance feedback[66].

In the application of machine learning for user modelling, large and labelled data sets might be required and the computational complexity might limit the performance of machine learning algorithms[58]. Therefore, the careful selection of machine learning techniques based on the challenges in different situations is vital to the system performance. Furthermore, machine learning techniques are also likely to be combined to provide personalised learning experience[67].

Chapter 3

Requirements Capture

3.1 The Overall Project Deliverable

The project has proposed a recommender system to assist diabetes self-management with a focus on glycemic control through insulin and meal interventions. The three major features are listed in the following:

- The recommender is able to provide recommendations proactively using the built-in knowledge in the recommender system. The recommender simulates combinations of treatment strategies and uses predictive patient model to justify the their quality and to select the optimal strategy as recommendations.
- The recommender is able to combine the knowledge of an expert and its system built-in knowledge to produce recommendations. The recommender adapts itself during the demonstration of the expert. In the recommendation phase, the recommender is more likely to deliver the actions similar to the demonstrated one.
- The adaptivity of the recommender system improves its performance over time and adapts the recommender if patient response pattern changes. This is achieved by an adaptive patient model, which is updated with patient response and allows the dynamic prediction of treatment outcomes. The patient predictive model has higher accuracy over time and can adapts to any change in patient behaviour pattern.

In the ‘Test and Evaluation’ chapter, the operations of the recommender with and without the expert have been demonstrated. Results have shown the improved glycemic control over time. Effects of parameter settings and design options have also been discussed.

3.2 The Component Deliverables

The final project deliverable can be decomposed into the following components:

- The overall architecture design of the recommender system, based on the ‘HAMMER’ architecture. The architecture itself is generalised and not explicitly designed for the diabetes management.
- Inverse model design, which generates treatment methods, for instance, the treatment strategies like diet plans and insulin injection plans.
- Forward model design, which imitates the behaviour of patients, for example, the glucose level response after injecting insulin.
- The virtual patient implementation which simulates the glucose-insulin dynamics for the evaluation purpose of the recommender engine. The design is proposed by Dr Dalla Man et al in 2007 [68].

The ‘Design and Implementation’ chapter explains the implementation details of the component deliverables. The effects of parameter settings and design options are investigated in the ‘Test and Evaluation’ chapter.

Chapter 4

Design and Implementation

4.1 Introduction

This chapter shows the design and implementation details of the project. Beginning with the design of the overall architecture, this chapter goes through the explicit component design and implementations. After that, the virtual patient implementation is introduced. The implemented system discussed in this chapter is then tested in the following chapter.

4.2 Overview

This section provides an intuitive understanding on the operation of the recommender system. Recall that the recommender system is based on the 'HAMMER' architecture, which allows models to suggest treatment methods competitively and select the most suitable one based on various criteria. Models that suggest actions based on patient state are named inverse models. Each inverse model is paired with a forward model, which is a patient predictive model to predict the likely outcome following the suggested action. Thus the selector may select the action that has the best result.

To improve the recommender performance, hierarchy is introduced by combining simple models to form complex models. Complex models operate competitively at a higher level just as the simple models. Also, forward models are designed to be adaptive and they learn from patient behaviour to improve prediction accuracy. In each iteration, the hierarchical models generate actions, forward models make predictions, and then a selector choose an action to deliver to the patient.

So far, the basic recommending procedure has been explained. The recommender is operating alone in the above scenario. The recommender can also be used after learning from an expert.

When there is an expert available, the recommender is able to learn from the expert by observing the treatment process. During the demonstration of the expert, confidence levels of models in the recommender change, based on the error between the predicted patient state caused by the inverse model's actions and the actual patient state caused by the expert's actions. In other words, models that causes similar results as the expert gains confidence. After demonstration is over, the recommender starts the recommendation procedure again. In the action selection process, the selector considers both confidence level and the predicted outcome to make a decision. In this case, both the built-in knowledge of the system and the knowledge of the expert are considered in the recommendation.

4.3 Inverse Models

In motor control theory, inverse models generate motor commands to approach the target state [69]. In this particular example, inverse models produce treatment methods to assist glycemic control, shown in the figure 4.1.

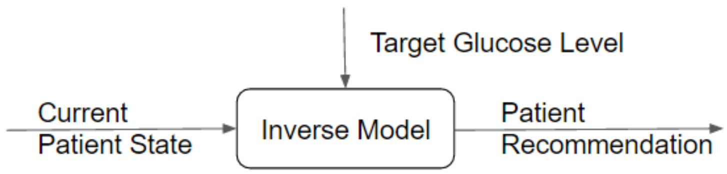


Figure 4.1: The explicit inverse model in diabetes management.

The glycemic control is specifically emphasised in this project. The glucose level is selected as the only target for inverse models, in this case, 112.5 mg/dl . Other inputs of the inverse models are the current patient states, such as the current glucose level, patient weight, sexuality, etc. The main intervention methods in glycemic control include meal plans and different insulin doses. For patients with T1DM or IDDM, insulin treatment is necessary owing to pancreas's failure in insulin production [19]. While for patients with T2DM or NIDDM, which is mainly caused by obesity and physical inactivity, insulin injection might not be necessary [19].

4.3.1 Insulin Models

In the project experiments, the effects of various types of insulin are categorised into fast-acting and long-acting insulin only, to reduce the complexity in simulation. The fuzzy classification of insulin types allows the actions selected from the recommender to have differentiable effects on the virtual patient for the purpose of demonstration. Details about calculating the glucose infusion rate and the implementation of the virtual patient insulin mechanism are explained in later sections.

A number of common insulin strategies are implemented as inverse models. In diabetes treatment, long-acting insulin is usually used to simulation the basal secretion of insulin in pancreas, while the fast-acting insulin simulates the endogenous insulin production during eating [70]. The sliding scale strategy is a still controversial, which was first described in 1924 and used for more than 90 years without sufficient evidence of support [71, 72]. It uses fixed units of long-acting insulin combined with a varied amount of fast-acting insulin routinely before each meal.

The long-acting and fast-acting insulin methods are implemented as separate inverse models to allow combinations of different strategies in higher level models, which will be shown in the following sections. Many studies have concluded that the sliding-scale strategy performs poorly in glycemic control, it is still implemented as inverse models owing to its simplicity and convenience [1, 73, 74]. It is also implemented for comparison with other insulin strategies.

The Dose Adjustment For Normal Eating (DAFNE) is another common approach which takes account into the carbohydrate patient is likely to take to adjust the insulin dose amount. The method was first used by the Diabetes Control and Complications Trial (DCCT) as an educational tool in 1990s and it becomes increasing popular owing to the promotion of short-acting insulins [75]. A clinical trial on flexible insulin management to dietary freedom has shown

	Before Breakfast	**Before Lunch**	**Before Dinner**	**Bedtime**
NPH Insulin	12 Units		6 Units	
Glucose Measure	**Regular Insulin Dose Amount**			
70-100 (mg/dl)	4 Units		4 Units	
101-150 (mg/dl)	5 Units		5 Units	
151-200 (mg/dl)	6 Units		6 Units	
201-250 (mg/dl)	7 Units		7 Units	
251-300 (mg/dl)	8 Units	1 Units	8 Units	1 Units
>301 (mg/dl)	9 Units	2 Units	9 Units	2 Units

Table 4.1: Sample sliding-scale insulin strategy using NPH insulin and regular insulin dose [1].

	Before Breakfast	**Before Lunch**	**Before Dinner**	**Bedtime**
Insulin Glarine				20 Units
Glucose Measure	**Regular Insulin Dose Amount**			
70-100 (mg/dl)	5 Units	5 Units	5 Units	
101-150 (mg/dl)	6 Units	6 Units	6 Units	
151-200 (mg/dl)	7 Units	7 Units	7 Units	
201-250 (mg/dl)	8 Units	8 Units	8 Units	1 Unit
251-300 (mg/dl)	9 Units	9 Units	9 Units	2 Units
>301 (mg/dl)	10 Units	10 Units	10 Units	3 Units

Table 4.2: Sample sliding-scale insulin strategy using insulin lispro and insulin glarine [1].

that the life quality and the glycemic control of patients are improved without worsening hypoglycemia and cardiovascular risk.

The most basic DAFNE regime is described as for every 10 grams of carbohydrates to eat, inject 1 UI of insulin [1]. In practice, the relation between insulin dose and carbohydrate is determined by the insulin-to-carb ratio, which might vary for different patient in different times of the day. Therefore, the DAFNE requires the patient to develop good understanding on his own body from experience.

In the inverse model implementation, the insulin-to-carb ratio for the virtual patient is fixed to be ten as approximation. The inverse model takes the carbohydrate to eat as an input to estimate the fast-acting insulin required to cover it. The adjustment of this parameter can be considered as a future development area.

4.3.2 Diet Models

The daily carbohydrate intake for male and female without diabetes are 230 grams and 300grams according to Diabetes UK. In the recommender system, diet plans are categorised into four different levels, shown in the table 4.3. Since the virtual patient does not include sexuality as an input, the implemented diet plans are same for both genders.

Diet Plans	**Carbohydrate Intake per Meal (grams)**
High-carb Meal Plan	$60 \sim 100$
Normal Meal	$45 \sim 60$
Moderate Diet	$10 \sim 53$
Very-low Diet	$0 \sim 10$

Table 4.3: Diet plans with carbohydrate intakes.

Diet Models suggest carbohydrate intake based on an arbitrary parameter ‘Hunger’ in the virtual patient, which is a random variable that determines how hungry the patient is. The probability of the carb recommendation has a uniform distribution over the range specified by meal plans in the table 4.3.

4.4 Forward Model

In control theory, forward models allow the system to predict future states following motor commands [69]. The quick error detection obtained by comparing actual and predicted states allows adjustment to the processing strategies in cognitive control [69]. In the diabetes recommender, an explicit explanation about forward model is shown in the figure 4.2.

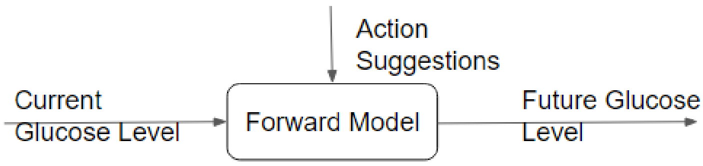


Figure 4.2: The explicit forward model in the application of diabetes treatment

Assuming the glycemic control is the only target in the experiment, forward model takes various action commands as inputs and make predictions on the future glucose level based on the current value. Intuitively, the forward model simulates the response of diabetic patients, predicting the likely glucose level variation of patients following different actions.

4.4.1 Adaptive Forward Model

Adaptivity is the key feature of the recommender system as the patient response mechanism is likely to change throughout the long treatment process of chronic diseases. Machine learning techniques are used, which updates the training data frequently to learn the patient behaviour.

Gaussian Process Regression (GPR) is chosen to implement the forward model. GPR has advantages over the parametric regression models their complex counter-parties like neural networks [76] in solving practical problems. Conventional regression algorithms like linear regression, the function $f(x)$ to learn is assumed to have the form $f(x) = mx + c$ or in polynomial forms. While GPR does not make such assumption but rather asks the training data to speak for themselves [77]. The parametric regression models generally do not perform well in complex dataset, while complex models such as neural networks are not easy to work with in practical applications [76]. The following section explains how to use Gaussian process in regression and how it is implemented in the final deliverable.

Gaussian Process is used as the *prior* in GPR. It is defined as a collection of random variables where any subsets of them form a joint multivariate Gaussian distribution. Gaussian process defines the distribution of over functions and it can be characterised by the formula:

$$f \sim GP(m, k), \tag{4.1}$$

where m is the mean function and k is the covariance function [76]. It is common to the mean of the Gaussian Process is zero everywhere and the relation between each sample only depends on the covariance function $k(x, x')$, where x and x' are input data. There are a range of options to compute the covariance function and a popular one is the 'squared exponential function':

$$k(x, x') = \sigma_f^2 exp[\frac{-(x - x')^2}{2l^2}] \tag{4.2}$$

In the final deliverable of the recommender, the kernel function is implemented as:

$$k(x, x') = \sigma_f^2 exp[\frac{-|x - x'|_f}{2l^2}] \tag{4.3}$$

where the$|x|_f$ stands for the Frobenius norm of the matrix and σ_f is the function deviation. The kernel parameter l is chosen to separate training data effectively. The covariance $k(x, x')$ is expected to be high if $f(x)$ is more likely to be positively correlated with $f(x')$ and vice versa. In practice, the observations can be noisy, which lead to another noise term in the expression, assuming the noise is white Gaussian noise with variance σ_n^2:

$$y = f(x) + N(0, \sigma_n^2) \tag{4.4}$$

Hence, the covariance matrix is modified into the following, owing to the independence of the noise term:

$$k(x, x') = \sigma_f^2 exp[\frac{-(x - x')^2}{2l^2}] + \sigma_n^2 \delta(x, x') \tag{4.5}$$

Using the key assumption in GPR that the observed data form multivariate Gaussian distribution. We have the following equation:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim N(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix})$$

where $*$ denotes the new generated data and T denotes matrix transpose. y_* is what we need to predict and the rest covariance matrixes are:

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_2) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix} \tag{4.6}$$

$$K_* = \begin{bmatrix} k(x_*, x_1) & k(x_*, x_2) & \dots & k(x_*, x_n) \end{bmatrix}, K_{**} = k(x_*, x_*) \tag{4.7}$$

Recall that our aim is to compute y_* given current input x_*, training observations y and training inputs x assuming that they form multi-variate Gaussian distributions.

The probability distribution of $y_*|y$ can therefore be computed as the following:

$$y_*|y \sim N(K_*K^{-1}y, K_{**} - K_*K^{-1}K_*^T) \tag{4.8}$$

A detailed derivation using Bayesian inference is available in the article [78]. Therefore, the best prediction of y_* is the mean of the distribution:

$$\bar{y}_* = K_*K^{-1}y \tag{4.9}$$

Three are many parameter settings that require careful tuning before the forward model can be used effectively in later experiments. σ_f and σ_n is the function variance and noise variance, which simulate the disturbance in the observation and data acquisition process. They should be adjusted carefully in real-world application where sensor data usually come with noise, but in the virtual patient simulation, σ_f is set to 1 and σ_n is set to 0.001 since negligible noise is introduced in the simulation. The kernel parameter l is chosen to separate training data effectively, and the suitable value of l is closely related to the training data, as different dimensions of x lead to largely different values after computing the covariance function. In the implementation, l is estimated to be the Frobenius norm averaged over all training data entries, which is tested and shown the reasonably performance in separating the training data.

In the implementation of GPR in the recommender system, the training inputs x include the current glucose level before taking action, and the quantified action inputs such as insulin units and carbohydrate intake. The training observations y is the resulting glucose level a certain time interval after the action is taken. In most experiments, the prediction interval is chosen to be one hour. Note that the one hour prediction interval is chosen arbitrarily to allow insulin and carbohydrate to take effect on the patient. According to the Diabetes Action Research and Education Foundation, glucose level usually peaks around 1 hour after meal. As a result, the intuitive understanding of the prediction interval is to minimise the worst-case scenario in glycemic control.

4.4.2 Explicit Static Forward Model

In the early-stage experiments, explicit static forward models are used for testing purposes, but static models are abandoned in the final deliverable owing to its disadvantage in coping with changes in patients. This section gives a brief explanation on how explicit and static models can be implemented for testing purposes.

The main idea of the explicit model is to estimate the time effect of actions explicitly and then build linear models on glucose level. Note the time effect of insulin and meal is generally nonlinear and it can be approximated with reasonable accuracy using an FIR filter [6], shown in figure 4.3.

This is a truncated Gaussian distribution which simulates the effect of insulin and meal over a certain period. Various parameters can be chosen to modify the magnitude and time scope to obtain good accuracy in estimation [6]. Linear models can be trained using filtered training data

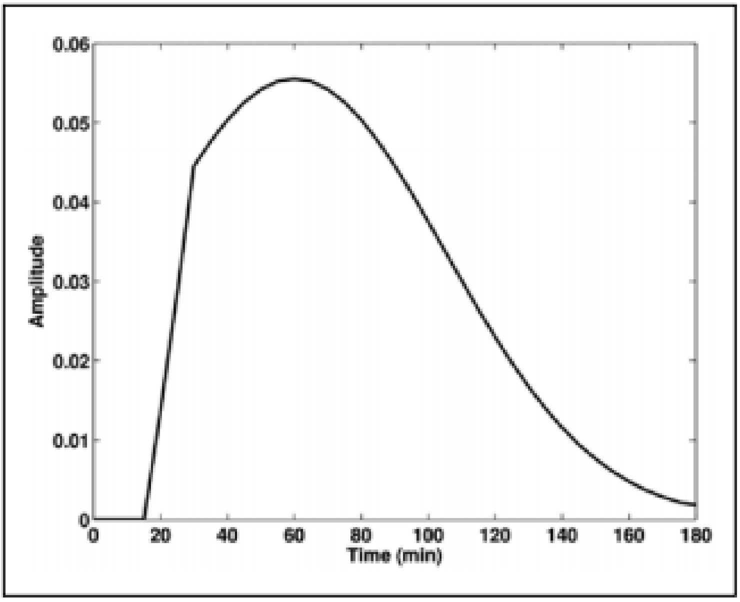


Figure 4.3: An example FIR filter to approximate the effect of subcutaneous insulin injection and meal ingestion within 3 hours [6].

in Matlab and then hardcoded into the recommender system. Such linear model has reasonable performance in prediction but lost its adaptivity. Also, the model is explicitly designed for each inverse model, it leads to higher complexity when the number of inverse models increases.

4.5 Inverse-Forward Model Pair and Confidence Function

This section explains the operation of the inverse-forward model pair in the recommender system for diabetes treatment.

As discussed in previous sections, inverse models generate diabetes therapies and forward models learn the behaviour of the diabetic patient. A more explicit explanation about the operation of the inverse-forward model pair is shown in the figure 4.4.

The action command for glycemic control is wired directly to the forward model to predict the future glucose level. The detection error obtained from comparing the actual and predicted states is fed back to the forward model to improve the prediction accuracy and also used to modify confidence levels of the inverse models.

The confidence function should increase the confidence level when the predicted glucose level is

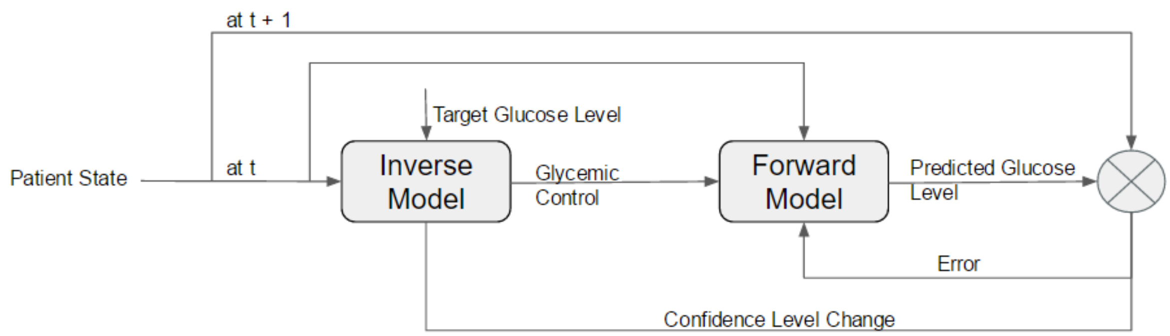


Figure 4.4: The mechanism of the model pair as a basic unit in the recommender system

close to the actual level. The equation 4.10 has shown a simple implementation of the confidence function.

$$C(t) = C(t-1) + Max(0, k - |G_{actual} - G_{predicted}|) \tag{4.10}$$

The coefficient k is an arbitrary number chosen that only if the prediction error is smaller than k, the confidence level of the inverse model increases. The coefficient k should be large enough to cover most range of the prediction error and at the same time, prevent the confidence level from negative numbers, which results in error in some action selection processes. The confidence functions can also be in other forms and not necessarily positive, depending on the choice of action selection algorithms. Effects of confidence level function parameter settings are discussed in the 'Test and Evaluation' section.

4.6 Hierarchies of Models

This section discusses the implementation of hierarchy in the recommender engine, including both the parallel and vertical structure.

4.6.1 Parallel Structure

Recall that the parallel arrangement of model pairs allows inverse models to generate action commands to achieve the same target competitively. The action selector then selects the one to deliver based on some criteria. The figure 4.5 gives an example of parallel arrangement of diet inverse models, which generate diet plans (DP1,DP2,DP3) to realise the target of glucose level. Forward model predicts the results of the diet plans and modifies the confidence levels of inverse models if an expert doctor is available. The different diet plans are sent to selector for selection.

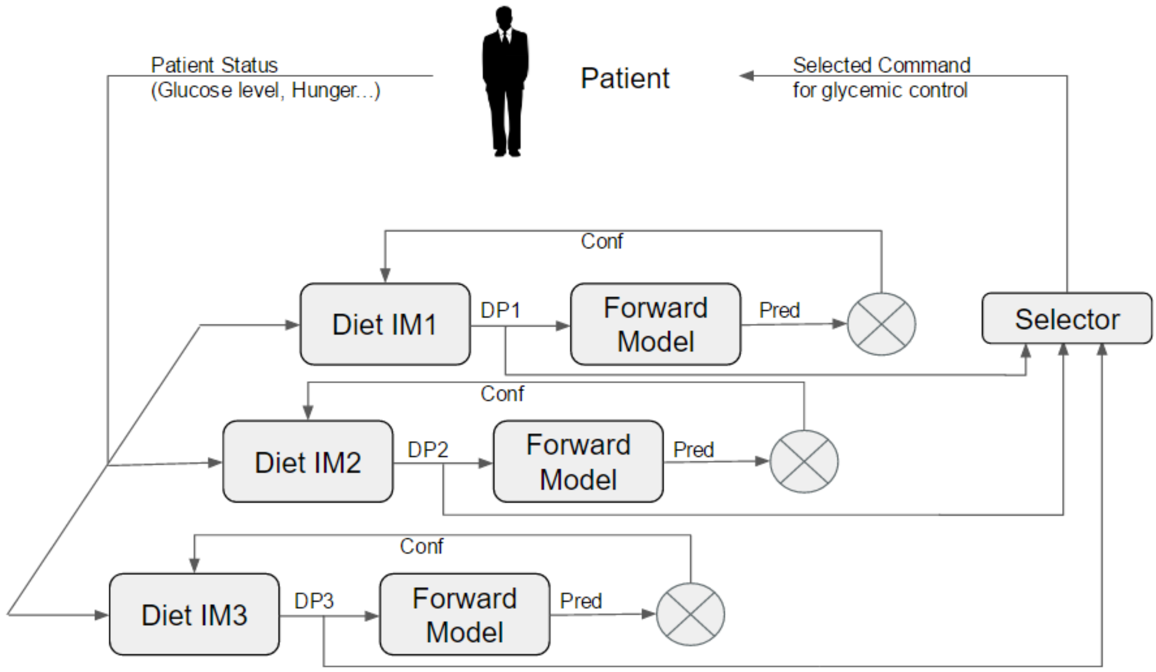


Figure 4.5: The parallel arrangement of model pairs

A intuitive understanding on the parallel structure is that there are various doctors to give different suggestions on the same aspects while the patient only takes the most suitable one. The action selection can be implemented in different ways, for example, the lowest risk, the best quality, etc. Details of action selection are in the following sections.

4.6.2 Vertical Structure

The idea of vertical structure is closely related to the abstraction level. In this project, higher level models are defined as the models that can access a wider range of aspects to make recommendations. One example is to combine inverse models of various aspects to form a higher-level inverse model. In another word, combining the treatment strategies of single aspects to form multiple-therapy strategies.

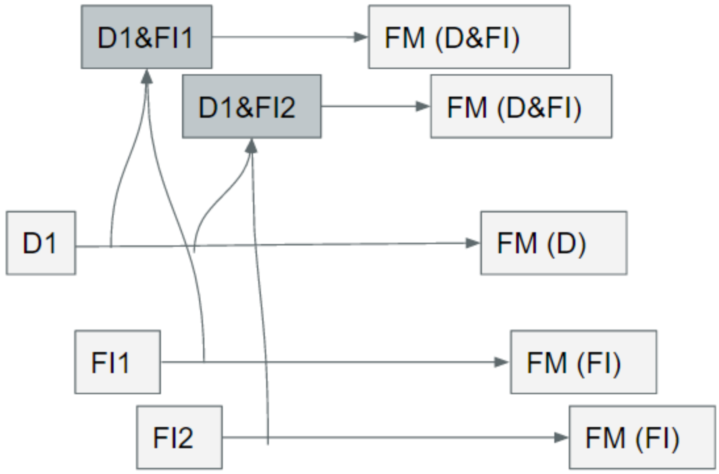


Figure 4.6: The vertical structure of model pairs from the perspective of implementation.

Commands from low-level models targeting different aspects are directly wired to a high-level model. Higher-level model can either output action commands of the low-level models directly or perform some processing based on the actions suggested from low-level models. The figure 4.6 shows an example of such structure. In the diagram, 'D' denotes diet model, 'FI' denotes fast-acting insulin model. 'D1&F1' denotes the combination of diet model 'D1' and fast-acting insulin model 'FI1'. 'FM' stands for forward model and the bracket after FM contains the aspects for the forward model to cover. The diet plan 'D1' are paired with insulin plan 'FI1' and 'FI2' to form two higher-level models. The combinations of inverse models that target different aspects are automatically performed. Also note that the advantage of using generalised machine learning algorithm to implement forward models is also shown in this example. Forward model looks at the signals coming from the corresponding inverse model and then determines the input and output variables. It then automatically requests corresponding training data from the patient for training and predicting purposes. So only one generalised forward model is implemented which can be paired with various inverse models.

4.6.3 An Overview of Hierarchy

The actual hierarchical structure combines the parallel and vertical arrangements of model pairs from the perspective of implementation. The figure 4.7 shows part of the implemented hierarchy. Primitive inverse models cover three aspects, namely diet (D), fast-acting insulin (FI), and long-acting insulin (LI). Two inverse models are implemented for each aspect and two high-level models are formed by combining different primitive inverse models.

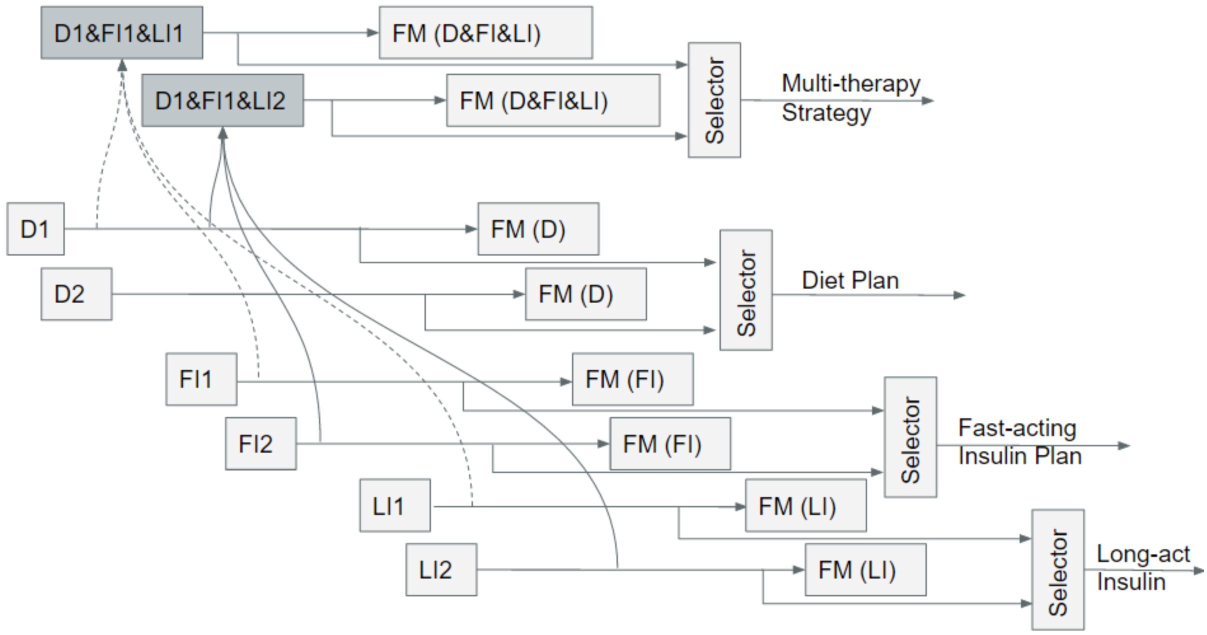


Figure 4.7: An overview of the hierarchical structure from the perspective of implementation.

During execution, model pairs in the same level operate parallel. Lower level models execute before higher level models since the execution of higher-level models requires the signals from lower-level models. The action selection process happens at all levels, either focusing on single aspect or a range of aspects.

Intuitively, high-level models are capable of accessing a wider range of information and such bottom-up architecture allows high-level models to have a broader overview in the action generation and recommendation. In the example of glycemic control, higher-level models produce the multi-therapy strategies, which are essentially combinations of basic strategies. It is interesting to know that a study has shown that multi-therapy treatment strategy is beneficial for diabetic patients in the long term [79]. More potential applications of the hierarchical structure are discussed in later chapters.

4.7 Action Selection Algorithms

Several action selection algorithms have been implemented to achieve different targets. There are many trade-offs to consider in action selection, for example, confidence-quality trade-off and the exploitation-exploration trade-off. This section introduces the different action selection algorithms implemented.

4.7.1 The Confidence Selector

Confidence selector uses the confidence level as the only selection criteria. For inverse models operating at the same level and has the same aspect, the confidence selector inspects their confidence levels and selects the action suggested by the most confident inverse model.

Recall that the confidence level shows how 'confident' the inverse model is in action generation. The confidence selector essentially selects the inverse model which is most similar to the expert. Assuming a perfect expert is available, the confidence selector is likely to be a good choice. However, in practice, the non-ideal treatment strategy is almost inevitable due to the complexity of clinical treatment problems. Therefore, the confidence selector is not necessarily the best choice at all situations.

4.7.2 The Quality Selector

As its name suggests, the quality selector always selects the action with the highest quality. Quality is defined as how close the predicted state is to the target state. In the example of glycemic control, the predicted glucose level after a time interval is compared with the target level, and the action which is most likely to achieve the target glucose level will be selected.

The quality selector is likely to be a good choice if the forward model performs well and the target is set properly. Assuming a perfect forward model and a ideal target, it makes sense to select the action which is believed to help the patient achieve the target state. However, the perfect forward model hardly exists and the target state is difficult to determine as the normal glucose level varies for different patient characteristics or even in different times of a day.

4.7.3 The Roulette Wheel Selector

The Roulette Wheel Selector is also referred as the fitness proportional selector. It is a commonly used selection algorithm which assigns weights on different options. The probability to selection each option is proportional to the weights assigned.

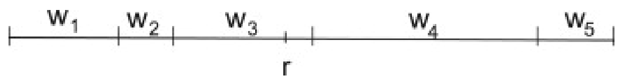


Figure 4.8: A visualisation of the roulette wheel selection algorithm [7].

The figure 4.8 visualise the selection mechanism of the roulette wheel selector. 'w' denotes weight. A random number r ($0 < r < \sum_{i=1}^{N} w_i$, N is the number of options) is generated. The location of r determines selects the option. In this example, option 3 is selected. In the recommender system, weights are assigned with values of confidence levels. Therefore, the model with higher confidence level is more likely to be selected while probability to select other models still exists. The idea behind the roulette wheel selector is to make decisions based on confidence levels but at the same time, keep the randomness. The roulette wheel selector introduces the exploitation and exploration trade-off. Exploitation means making the best decision given all available information while exploration means developing potential opportunities to gather more information. It is also possible to realise the exploitation exploration trade-off by assigning weights according to the quality of the action.

The roulette wheel selector has great advantage in its computational complexity. It has complexity of $O(1)$ [7], which gives rise to quick selection when the number of options is large. However, the roulette wheel selector might introduce extra risk owing to its randomness. In the case of medical treatment, the design of the recommender system needs extra attention to details [49]. The tiny probability to select a poor option is likely to lead to severe result in clinical trials. This problem might be solved using modified roulette wheel selector to eliminate the probability of dangerous options.

4.7.4 Multi-Objective Selector

In the situation where more than one objective is required to consider, multi-objective algorithms are used to make decision based on multiple criteria. This section uses Pareto Dominant Selector as an example.

The Pareto Optimality is named after an Italian economist and engineer Vilfredo Pareto. It describes the efficiency allocation of resource to optimise the overall utility using multiple criteria. This idea is used here to find the trade-off point between confidence and quality.

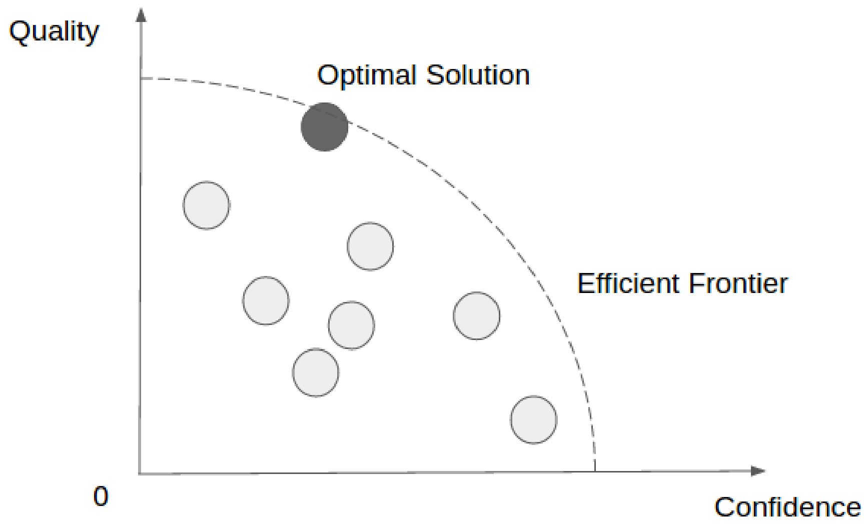


Figure 4.9: An example of Pareto frontier based on quality and confidence level.

The figure 4.9 has shown an example of Pareto frontier. Each inverse model is evaluated on its quality and confidence value separately and assigned with a score on each criteria. The model with the highest overall score (Dominant option) is selected. Note that the confidence level values and quality values are scaled and weighted before converting to a meaningful score. The overall score is calculated by summing up weighted scores on each criteria, where weights represents the user preference in each criteria. Modifying weights directly reflects the user biases on each criteria, which makes the Pareto Dominant Selector convenient and flexible.

4.8 The 'HAMMER' Architecture

This section provides an overview of the 'HAMMER' architecture implementation from the perspective of software engineering.

Project development is mostly under Eclipse JAVA EE with some data processing and visualisation tasks on Matlab. JAVA is explicitly chosen for this project to allow future development on the project, for example, integrating the recommender system with the front-end mobile application and providing cloud service to mobile users. The software project management is done by the Apache Maven, owing to its advantage in dependency management and version management.

The figure 4.10 has shown a brief overview of the 'HAMMER' architecture implementation. Each component represents a JAVA object and a detailed UML diagram including attributes, functions, package names, etc, is available in the Appendix A.1.

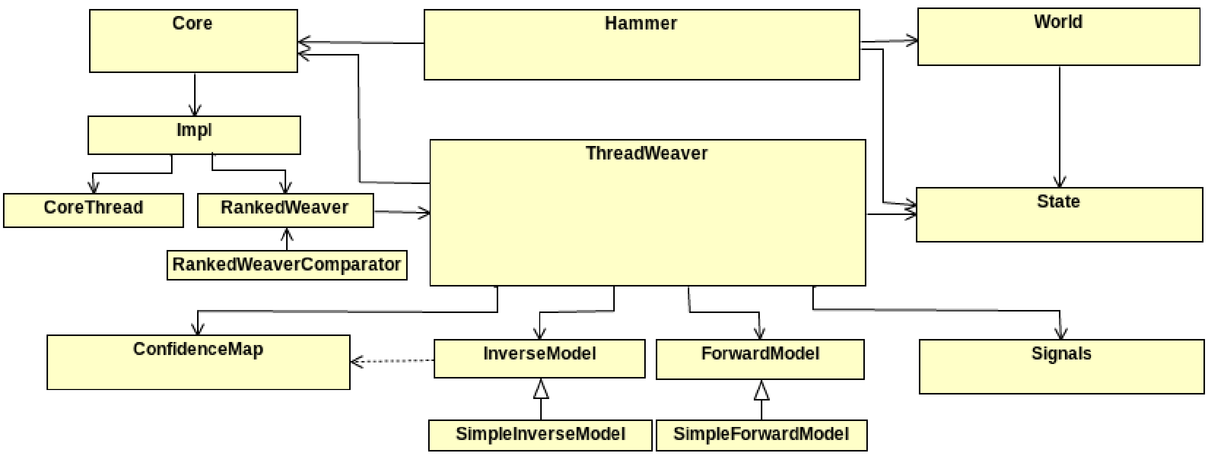


Figure 4.10: The UML diagram of the 'HAMMER' architecture

The 'HAMMER' is the main interface for the recommender system. It connects the 'Core' of the 'HAMMER' architecture to the 'World' simulation and provides interface for 'HAMMER' settings, such as action selector options, confidence function choices and automatic model pair registration.

The 'Core' is the control hub of the 'HAMMER' architecture, providing access to various threads, models and functions. The 'Core' usually contains a number of 'ThreadWeavers', each operates in a single thread and consists of a pair of inverse model and forward model. Such design allows the parallel and sequential operation of model pairs. The execution sequence of the threads is realised by assigning execution ranks. The determination of the execution ranks is based on the model dependencies, which will be elaborated further in later sections. The inverse and forward model classes are implemented with a sets of common functions like clone and simulation, as well as an unspecified main 'Functor' shell, which allows the developer to hardcode the model functionalities externally and then attach the functor to models. Similar to inverse model and forward model, the action selection method and the confidence function are designed as functors which are implemented externally and then linked to the Java class.

The 'World' class contains all the information used available for HAMMER, for example, the virtual patient glucose level or the position of a robot. The state of 'world' is iterated through the execution of HAMMER. The object simulated accepts the actions recommended from HAMMER, which lead to the next state. The 'Signal' class defines the communication method between inverse and forward models. It is essentially a data structure which is capable of storing various types of information and it also contains supporting functions like cloning and data extraction. In order to compromise various data types that are possibly used in practice, general class types are widely used in the implementation, i.e. the inputs and outputs of functions are usually not specified, which allows different types of objects to be passed around.

4.9 Virtual Patient

4.9.1 Introduction

This section briefly introduces the virtual patient dynamic system, proposed by Dalla Man. It focuses on the insulin injection and glucose absorption dynamics, and how the virtual patient interfaces with the recommender system through insulin and glucose interventions.

4.9.2 An Overview of Glucose-Insulin Subsystems

The implementation of the virtual patient follows the glucose-insulin system proposed in [2, 68] by Dalla Man in 2007. This model has now been included in the SimBiology Toolbox in Matlab 2016a. It has been commonly used in recent studies in diabetes patient simulation before clinical trials. The glucose-insulin system decomposes the whole system into several subsystems and uses forcing function strategy to develop parametric models to approximate major glucose and insulin influx, for example, endogenous glucose production, utilisation of glucose, etc. Different types of impairments, including T1DM, T2DM as well as normal patients, are represented as different parameter settings and initial conditions in parametric models. The list of implemented parameter settings is included in the Appendix B.1.

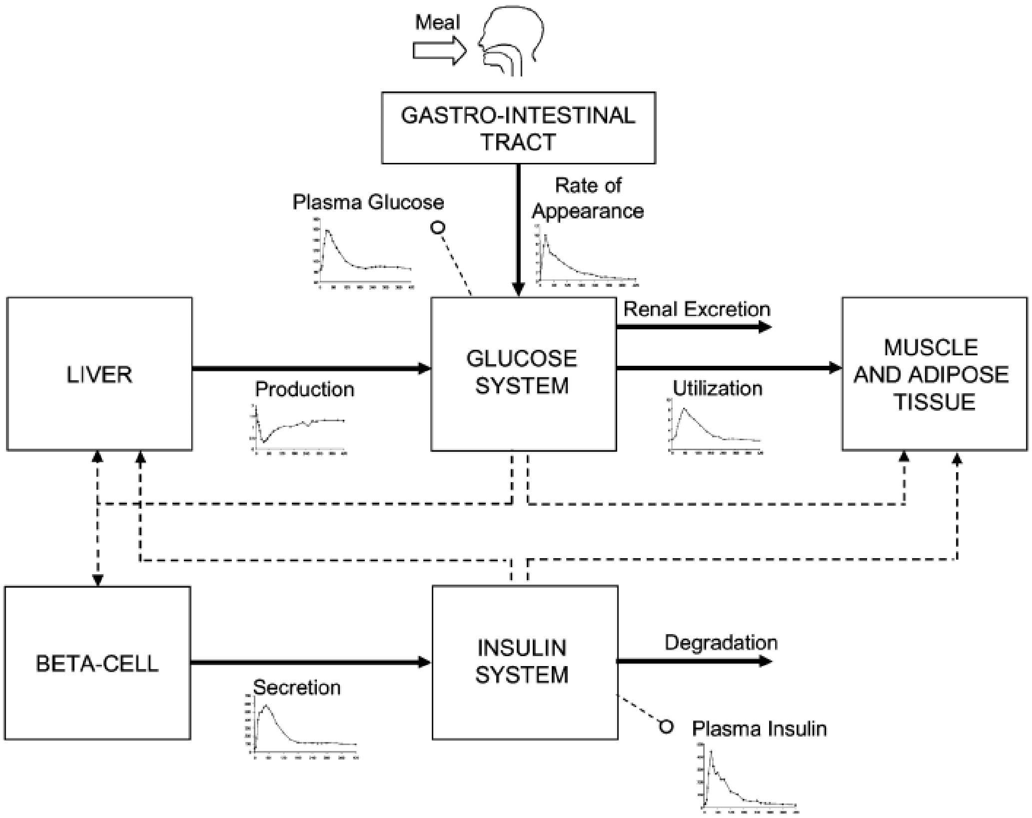


Figure 4.11: An overview of the glucose-insulin subsystems [2].

The figure 4.11 shows an overview of the glucose-insulin system proposed by Dalla Man [2]. The glucose subsystem takes account into the meal rate of appearance, renal extraction, the glucose production from liver, and utilisation from muscle and adipose tissue. The insulin subsystem considers the insulin secretion from pancreas beta-cells and insulin degradation to liver and periphery organs.

The subcutaneous insulin subsystem and intestinal absorption model is of particular importance in this project since the insulin injection amount and the meal ingestion will be quantified and input to the glucose-insulin dynamics to simulate the response of diabetic patient after taking actions suggested by the recommender engine. A thorough explanation on the glucose-insulin model is available in [2, 68].

4.9.3 Subcutaneous Insulin Dynamics

The subcutaneous insulin affects the insulin system by changing the monomeric and non-monomeric insulin concentration in subcutaneous space, which partly enters the circulation and affects the plasma insulin level.

$$\begin{aligned} \dot{S_1}(t) &= -(k_{a1} + k_d)S_1(t) + u(t) \\ \dot{S_2}(t) &= k_d S_1(t) - k_{a2} S(t) \end{aligned} \tag{4.11}$$

The ordinary differential equation set 4.11 represents the subcutaneous insulin mechanism, described in [80]. $S_1(t)$ and $S_2(t)$ stand for the amount of monomeric and non-monomeric insulin amount in subcutaneous space in real time. Their first-order derivative $\dot{S_1}(t)$ and $\dot{S_2}(t)$ represent the rate of change in insulin amount. The parameters k_{a1} and k_{a2} are the absorption rates of non-monomeric and monomeric insulin. The coefficient k_d the the rate of conversion for non-monomeric insulin to enter the circulation. The variable $u(t)$ represents the insulin administration, which is used to simulate the patient response after taking actions from the recommender system.

In the virtual patient simulation, insulin administration categorised into fast and long action insulin, each uses an approximation to convert insulin unit into $u(t)$. One unit insulin is biologically equivalent to 34.7 g pure crystalline insulin while 1 ug of insulin is approximately 172.1 pmol, according to CTDS Laboratories [1, 81]. The insulin administration is in the unit of pmol/kg/min. The duration of the fast-acting insulin is assumed to be one minute. While the duration of long-acting insulin is estimated to be 23 hours and $u(t)$ is equally distributed over this period. The suggested insulin unit is converted into $u(t)$ taking account of the conversion rates above, the patient body mass and duration assumptions.

4.9.4 Glucose Absorption Dynamics

The glucose absorption model used here is described in [82]. Intuitively, meal ingestion affects the glucose system through affecting the amount of amount of glucose to digest and rate of digestion. The equation sets 4.12 has shown the effect of ingested glucose $d(t)$ (mg/min).

$$\begin{aligned}
&Q_{sto}(t) = Q_{sto1}(t) + Q_{sto_2}(t)\\
&\dot{Q_{sto1}}(t) = -k_{gri}Q_{sto_1}(t) + d(t)\\
&\dot{Q_{sto2}}(t) = -k_{gut}(t, Q_{sto})Q_{sto2}(t) + k_{gri}Q_{sto_1}(t)\\
&\dot{Q_{gut}}(t) = -k_{abs}Q_{gut}(t) + k_{gut}(t, Q_{sto})Q_{sto_2}(t)\\
&Ra(t) = \frac{fk_{abs}Q_{gut}(t)}{BW}
\end{aligned} \tag{4.12}$$

The variable $Q_{sto}(t)$ stands for the glucose amount in stomach in mg, which is composed of solid and liquid glucose, Q_{sto_1} and Q_{sto_2}. The first-order derivatives represent the rate of change of variables. $Q_{gut}(t)$ is the intestinal glucose mass. The parameters k_{gri} and k_{abs} are rate of grinding and rate of intestinal absorption respectively. $Ra(t)$ is the glucose rate of appearance in plasma, which relates to the body weight (BW in kg), the proportion of intestinal absorption f, the rate of grinding k_{gri} and intestinal glucose mass Q_{gut}.

$$\begin{aligned}
&k_{gut}(t, Q_{sto}) = k_{min} + \frac{k_{max} - k_{min}}{2}\{tanh[\alpha(Q_{sto} - b\bar{D}(t))] - tanh[\beta(Q_{sto} - a\bar{D}(t))] + 2\}\\
&\alpha = \frac{5}{2\bar{D}(t)(1-b)}, \beta = \frac{5}{2\bar{D}(t)a}\\
&\bar{D}(t) = Q_{sto}(\bar{t}) + \int_{\bar{t}_F}^{\bar{t}} d(\tau)d\tau
\end{aligned} \tag{4.13}$$

The rate of stomach emptying k_{gut} is also affected by glucose ingestion and the process is approximated in the equation 4.13, where a, b, k_{max}, k_{min} are model parameters. $\bar{D}(t)$ accumulates the glucose intake when eating, where $\bar{t}_F$ and $\bar{t}$ are the initial and final times of last food intake.

In practice, the rate of ingested glucose is difficult to measure. As a result, the meal ingestion process is approximated to happen in one minute in our simulation. Since the $\bar{D}(t)$ tracks the accumulation of glucose intake, the assumption of instant food intake might not be accurate in the very short run, but can be a reasonable approximation sometime after the meal intake.

The unit for ingestion rate is mg/min and the meal suggestion from the recommender can easily be converted to mg. Note that in both meal model and insulin model, the delayed effect of meal and insulin have been taken into consideration by the glucose-insulin system [2].

Chapter 5

Test and Evaluation

5.1 Introduction

This section shows the experiment results of the deliverbles. It starts with virtual patient simulation results on different types of patients. Then the effects of design choices and parameter settings of the recommender system are discussed. Finally, the overall performance of the recommender is evaluated, in the scenarios with and without an expert doctor.

5.2 Virtual Patient Simulation

5.2.1 Daily Activity Simulation

This section aims to provide an intuitive understanding about simulation results on different types of virtual patient. This model proposed by Dalla Man has been widely used in many recent research and a complete evaluation of the design is available in [2]. The implementation of the virtual patient has shown the same result as the model used in Imperial College Personal Robotics Laboratory. Therefore, this section only focuses on giving an overview of the virtual patient behaviour rather than any clinical or engineering evaluation on the virtual patient.

The virtual patient simulates the glucose-insulin system of human orgasm continuously from a specified initial state. Interventions like meal intake and insulin injection can be simulated as discussed in previous sections. Simulated patient types include T1DDM, T2DDM and normal people, each of them corresponds to different parameters and initial conditions. A detailed parameter setting is available in Appendix B.1.

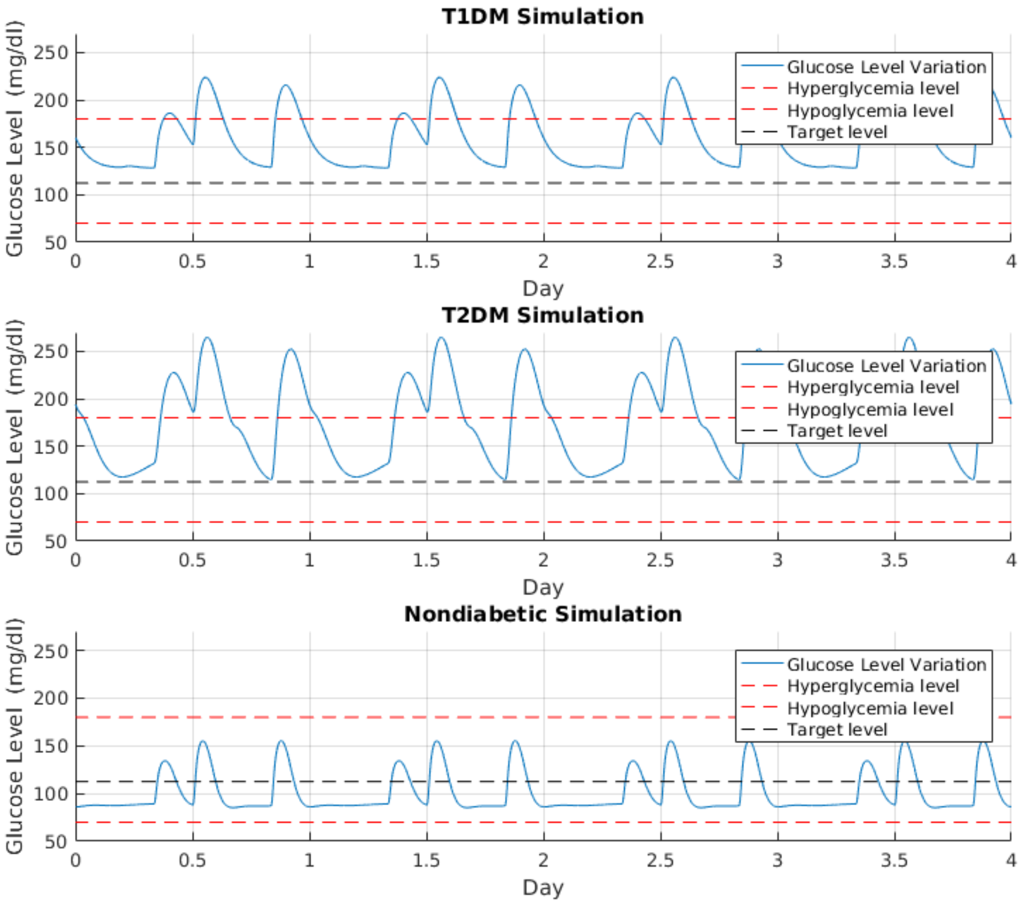


Figure 5.1: Glucose level simulations with typical meal ingestion and no insulin medication, for T1DM (top), T2DM (Middle) and nondiabetic people (bottom).

The figure 5.1 simulates the glucose level variations of a typical daily life, i.e. regular meals and no insulin medication, for patient with T1DM, T2DM and nondiabetic people. Meal times are assumed to happen at 8 a.m., 12 a.m. and 6 p.m, corresponding to the peaks in daily simulations. Different types of patients are given same amount of carbohydrate for the purpose of comparison. It is observed that the glucose levels for diabetic patients exceed the hyperglycemia line frequently during after meal ingestion, without the support of insulin medication. For nondiabetic people, their glucose-insulin dynamics is capable of glycemic control without external insulin injection.

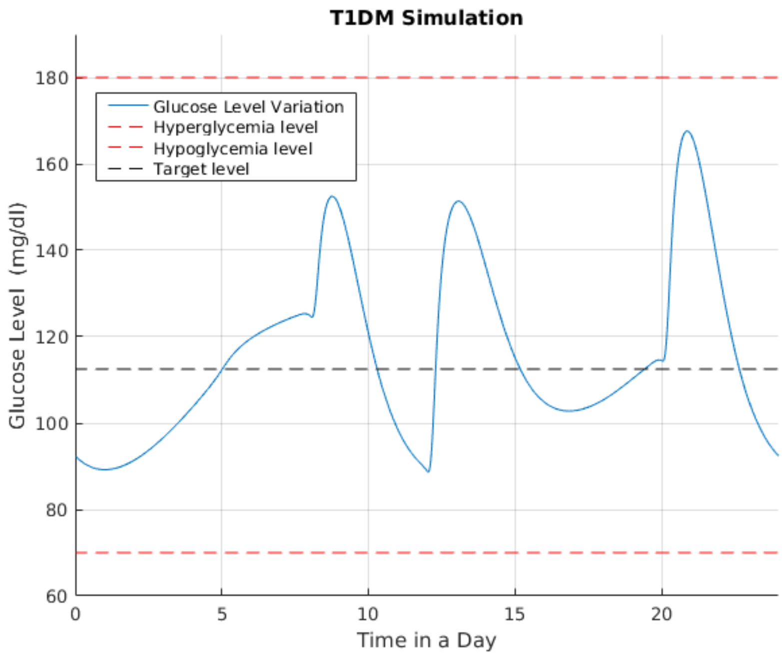


Figure 5.2: A close look at the T1DM glucose simulation in a typical day with suitable insulin medication.

The figure 5.2 shows the example of the glucose variation throughout a typical day with insulin management. The simulated activities includes breakfast, lunch, dinner and insulin intake before each meal. The three peaks shown in the diagram indicates the high glucose concentration after each meal, while the peak is quickly eliminated by the combined effect of insulin injection and glucose utilisation of muscle and other tissue. It can be noticed that under suitable insulin medication, the glycemic level is under control, shown as small variance compared to the top figure in the figure 5.1. Also, no hyper or hypo-glycemia condition is observed throughout the glucose level simulation.

The figure 5.3 shows that the convergence of the glucose levels of the virtual patient with and without diabetes mellitus are close to our expectations. The simulation is performed with no action intervention, that is, the patient does not eat or use insulin. It is observed that patients with diabetes mellitus end up in a state with higher glucose concentration while the glucose level of the normal people remains in a lower level. Note that in practice, patient without any meal intake is less likely to maintain a high glucose level. But the simulation incorporates the idea that the glucose variation for diabetic patients trends to be higher without intervention. It is also observed in the figure that patient with T2DM has a higher steady state glucose level

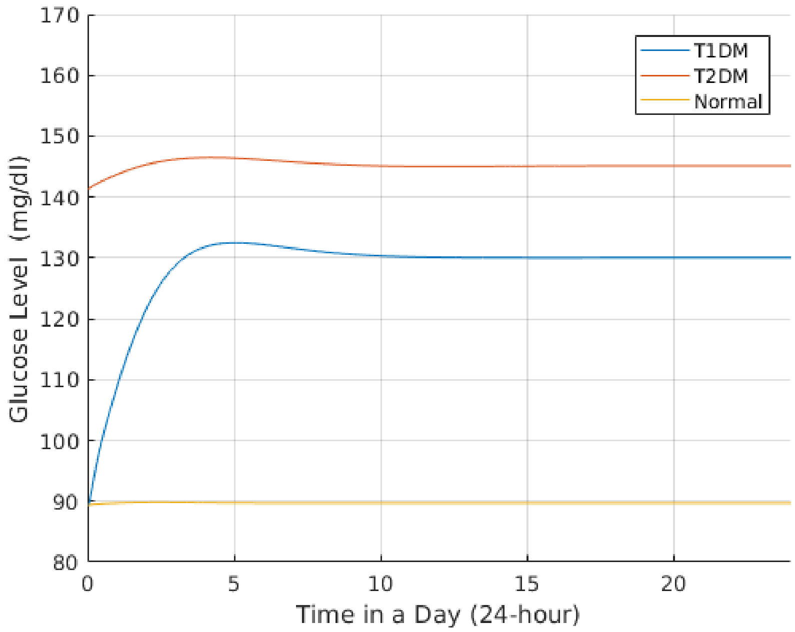


Figure 5.3: The simulation of glucose level variation from initial state to steady state without any activity for patient with T1DM, T2DM, and normal people.

than the patient with T1DM. This is largely caused by estimated parameters and initial states and can be personalised to simulate a real patient.

5.2.2 Assumptions on Daily Life Pattern

The experiment aims to simulate a typical daily life of a diabetic patient. Assumptions are made to simplify the simulation for testing purpose of the recommender system. First of all, it is assumed that the virtual patient follows a typical living pattern. The virtual patient has food breakfast at 8 a.m., lunch at 12 a.m. and dinner at 6 p.m. The patient takes fast-acting insulin 30 minutes before each meal to compensate the effect of glucose intake, which is a common strategy to allow fast-acting insulin to take effect during eating. The patient injects long-acting insulin at 10 p.m. at night before he goes to bed. Therefore, the recommender system is only activated at fixed times of a day. The implementation of the inverse model takes the time in a day into account and prevents the model to make any recommendation at wrong time.

5.3 Design Options

This section introduces the design choices of the recommender system, including hierarchy, confidence level function, action selector and the pre-training phase of forward models.

5.3.1 Role of Hierarchy

In this section, hierarchy has lead to improved performance in action recommendation. This section evaluates the different performance of the recommender system using low-level and high-level models to recommend actions.

Recall that high-level inverse model are formed by combining primitive inverse models. Forward models always predict future states based on the aspects covered by inverse models. As a result, forward models paired with high-level inverse models automatically predict future states based on more variables. Therefore, it is expected that high-level model outperforms low-level model in glycemic control owing to the better forward model prediction accuracy. Experiments agree with the expectation.

Abstract levels	**Variance**	**Hyperglycemia Duration (Hour)**	**Hypoglycemia Duration (Hour)**
Low-level	918.92	5.40	5.61
High-level	794.31	3.86	3.06

Table 5.1: Comparison on recommender performance using models with different abstract levels.

In experiments, models are implemented in two abstraction levels only for simplicity. The primitive models are diet plans, fast-acting insulin plans and long-acting insulin strategies, while the higher level models are combinations of the three treatment strategies. Repetitive experiments are carried out using models in different levels to suggest actions. Results are shown in the table 5.1.

Each simulation lasts 100 days and the performance of recommender in the last 10 days are shown in the table. Evaluating the last ten days' performance is because the benefit of using high-level model is more significant in the long-run. While in the short run, there is not sufficient data to train forward models, resulting in significant noise, so that the benefit of covering more variables is less obvious.

Variance is computed by averaging the glucose level variance around the 112.5 (mg/dl) target level. The hyper and hypo-glycemia duration is computed by summing up the length of hypo- or hyper-glycemia throughout the last ten days. It is observed that using high-level model to recommend action leads to less variation around the target glucose level and less hypo- or hyper-glycemia condition in the long-run.

5.3.2 Confidence Function

The confidence function modifies the confident levels of models based on the prediction error on glucose levels. Confidence function choices have different results during the expert demonstration, which then affect the action selection process. This section discusses the different confidence functions and parameter settings.

$$C(t) = C(t-1) + Max(0, k - |G_{actual} - G_{predicted}|) \tag{5.1}$$

Recall that a simple confidence function can be implemented as 5.1, where k is the error threshold that any error less than that help inverse models gain confidence.

The figure 5.4 shows the effect of confidence function coefficient k. The confidence levels of the demonstrated action (blue line) and other actions that not demosntrated are included for comparison. It is observed that the demonstrated strategy is assigned with the highest confidence level for all k values. Small k value reduces the chance of confidence increment for poor-quality models. When k is below 10, the confidence levels of poor-quality models are reduced to a negligible level.

$$C(t) = C(t-1) + Max(0, [k - |G_{actual} - G_{predicted}|]^n) \tag{5.2}$$

The equation 5.2 is an extension on the simple linear confidence function, where k is the error threshold and n is the polynomial order. The non-linear confidence function is used to magnify

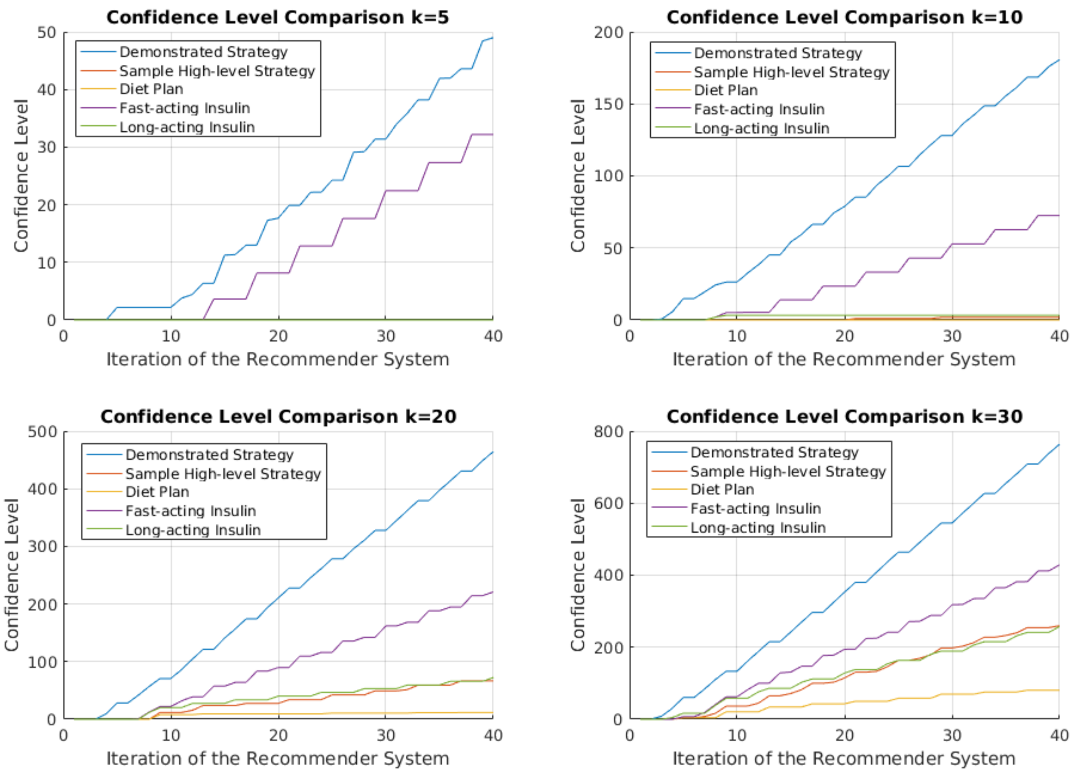


Figure 5.4: Confidence level comparison on different k values in the simple linear confidence function.

the impact of error signal. The figure 5.5 shows result of the second-order confidence function with various error thresholds. The confidence level of the demonstrated strategy is compared with other strategies not in performed by the expert.

It is observed that although the numeral difference between confidence level lines is larger with the non-linear confidence function, the differences in ratios decrease which leads to worse performance in ratio-dependent action selector, like the roulette-wheel selector. Also, when error threshold k is small, the demonstrated strategy is no longer assigned with the highest confidence level. The non-linear confidence selector introduces more uncertainty into the confidence assignment. For example, if a poor-quality inverse model accidentally makes a good guess, a significant confidence level will be assigned. Compared to a good-quality inverse model keep making relatively well guess, the overall confidence of the poor-quality model might still be higher owing to the non-linear feature. Therefore, the simple linear confidence function is considered a better option than non-linear confidence function, and it is used in the final deliverable.

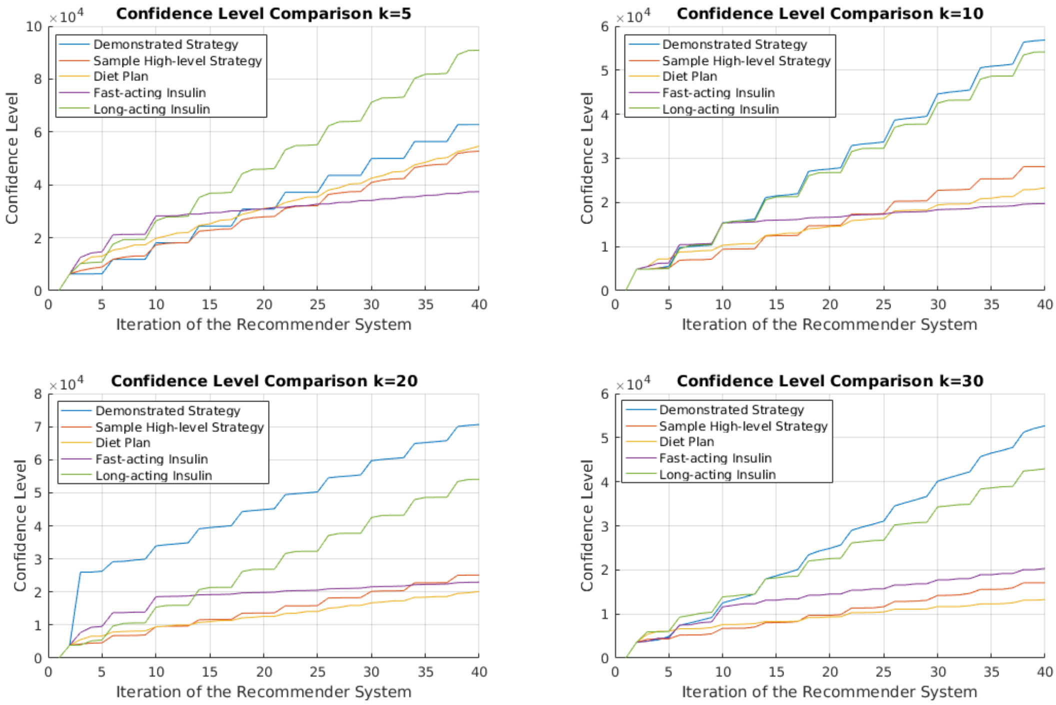


Figure 5.5: Comparison on the second-order confidence function with different k settings.

5.3.3 Pre-training of Forward Model

Forward model is responsible for the dynamic prediction of patient treatment outcomes following the actions suggested by the paired inverse model. However, the forward model might not perform well in early-stage recommendation owing to the limited training data. This section discusses the importance and feasibility of including a pre-training phase before delivering recommendation.

The figure 5.6 shows the glucose variation with randomly suggested action command. The training data is generated by collecting initial glucose levels, actions suggested and resulted glucose levels. Note that the time interval between action suggestions is set to 24 hours arbitrarily to eliminate the effect brought by previous commands. From figure 2.1, the glucose infusion caused by long-acting insulin nearly vanishes after 24 hours. From the virtual patient simulation [2], the glucose effect of meal is negligible after 24 hours.

First of all, including the pre-training of forward model significantly improves the prediction accuracy of the forward model from the beginning of the simulation. It then leads to the

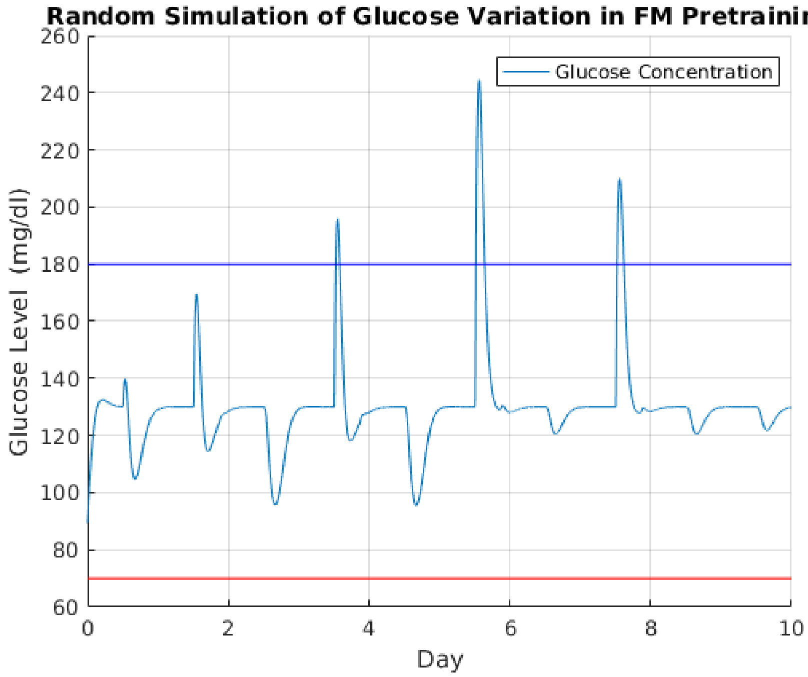


Figure 5.6: Glucose Simulation with Randomly Recommended Actions.

improved quality of action recommendationa using the built-in knowledge of the recommender.

The following experiment proves the improved quality using built-in knowledge of the recommender and the pre-training of forward models, shown in the table 5.2.

Forward Model Pretraining	Variance	Hyperglycemia Duration (Hour)	Hypoglycemia Duration (Hour)
Yes	657.37	14.23	26.87
No	921.08	15.40	15.35

Table 5.2: Comparison on recommender performance with and without the pretraining on forward models.

Results are obtained by performing 20-day simulations on virtual patients with and without forward model pretraining. It can be observed that the pretraining process significantly improves the performance of the recommender, shown as improved variance around the 112.5 (mg/dl) target glucose level and reduced durations of hyper and hypo-glycemia. The extra training data generated in pretraining improves the prediction accuracy of the forward model.

Secondly, including the pre-training phase improves the performance in combining the knowledge of expert, shown in figure 5.7

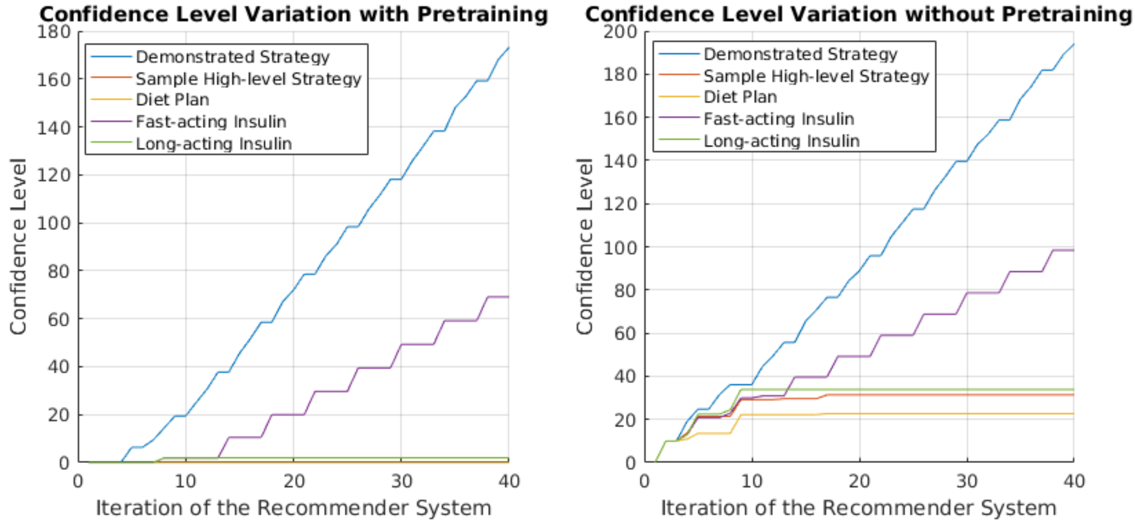


Figure 5.7: Confidence level comparison with (Left) and without (Right) the pretraining of forward model.

It is observed that with the pre-training process of the forward model, the confidence level of irrelevant models are kept to minimum from the beginning of the simulation, shown in the left figure 5.7. Without the pre-training, confidence levels of models are not differentiated until ten iterations of the recommender system where sufficient data has been collected to train the forward model. Therefore, pretraining forward model reduces the noise when combining the knowledge of expert to the recommender system.

It can be concluded that forward model pretraining benefits the recommender performance in both scenarios. However, forward model pretraining requires collecting extra training data by randomly generated actions, which might cause moral issues and extra risk in clinical trial. The feasibility of including the pretraining in clinical trial requires further research.

5.3.4 Action Selection Algorithms

Action selector chooses the most suitable action among the competitive action recommendation of models. When the recommender is operating without the expert, action selector only evaluates the quality of actions by predicting treatment outcomes. When the expert is present, there is a trade-off between the knowledge of expert and the knowledge of the recommender system. This section discusses the latter situation where action selector choice affects the performance of the recommender system.

This section includes the comparison between confidence selector, best quality selector, roulette wheel selector and Pareto Dominant selector. Also, two different situations are considered: Firstly, the doctor's knowledge is 'better' than the built-in knowledge of the recommender. Secondly, the doctor's knowledge is worse than the built-in knowledge of the recommender. The second situation is important to consider since in the real world, there is no perfect doctor. People make mistakes and this is the case where built-in knowledge of the recommender is beneficial.

Simulation with a Perfect Expert

In figure 5.8, a good expert is hardcoded with a suitable treatment method after a trial-and-error process. It is observed that the confidence selector performs better than the best quality selector in this situation. The confidence select choose the model which is most similar to the good demonstrator. The glucose level is well managed with small variation and close to the target glucose level. The proportionate select also has a larger chance to suggest a good action, and therefore, also shows reasonable performance in this situation.

The Pareto Dominant selector gives confidence and quality scores ranging from 0 to 1 to each inverse model. The quality and confidence are weighted at 0.7 and 0.3 respectively, indicating that quality level is slightly preferred. It is observed that the Pareto Dominant selector is able to achieve satisfying glycemic control after adapting the recommender to the expert with perfect knowledge.

Simulation with an Imperfect Doctor

Considering the situation that the doctor do not perform perfectly and the demonstrated action includes high carbohydrate intake which causes immediate hyperglycemia after food intake. The knowledge of the recommender system sets in and prevents such suggestion to be delivered to the patient.

In the figure 5.9, the confidence selector always selects the model that is closest to the doctor.

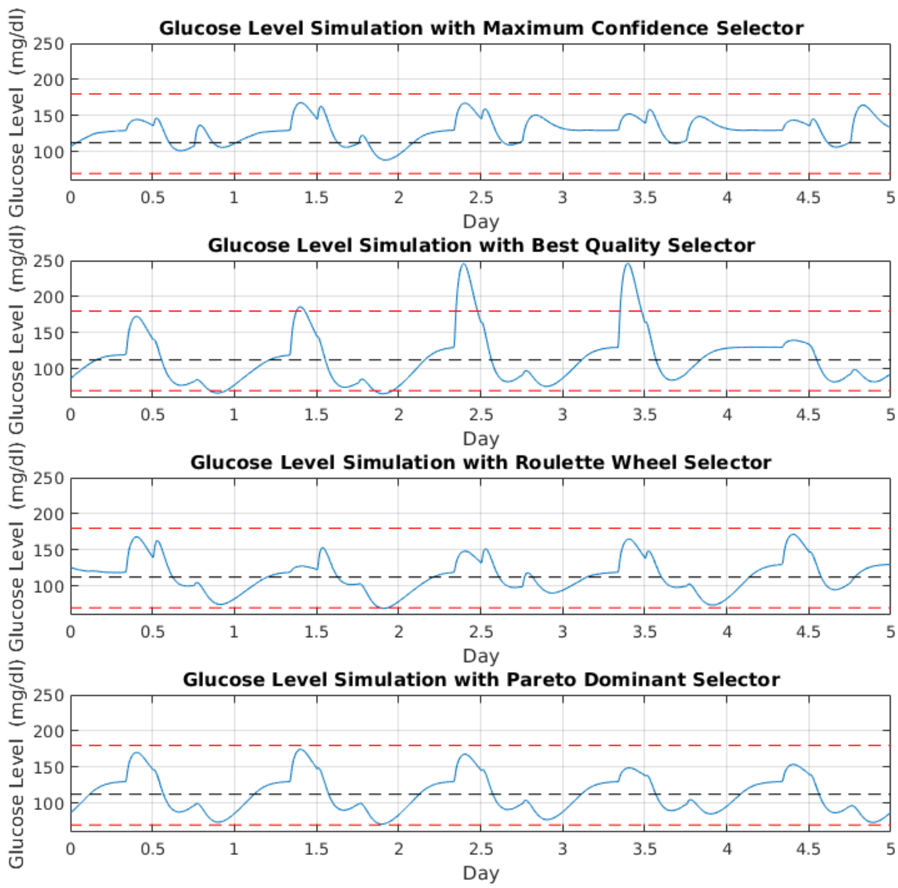


Figure 5.8: Glucose level simulation with three different selectors assuming a good demonstrator. Hypo- or hyper-glycemia lines (Red dashed lines) and target glucose levels (Black lines) are labelled.

The imperfect knowledge of the doctor leads to the poor performance of confidence selector. In contrast, the quality function only utilise the knowledge of the recommender and shows good performance. Regarding the roulette wheel selector, it selects actions with probabilities that are proportionate to model confidence levels, which are negatively affected by the imperfect doctor.

The Pareto Dominant Selector setting is same as the last section. Quality score and confidence score are weighted at 0.7 and 0.3 respectively. Therefore, the Pareto Dominant Selector takes account into the the knowledge of the recommender system and prevents the action that has risk of hyperglycemia to be recommended.

It can be concluded that the Pareto Dominant selector is better than other single-objective selector in terms of both performance and flexibility. It combines the knowledge of the recommender

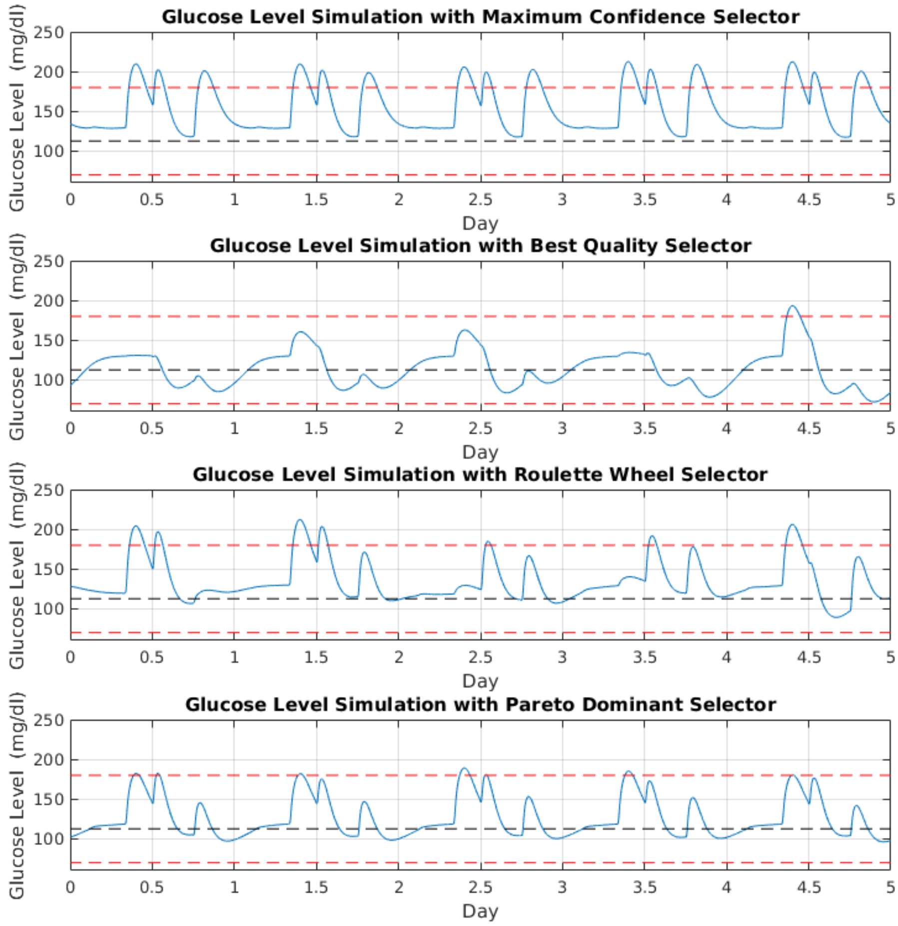


Figure 5.9: Glucose level simulation with three different selectors assuming an imperfect doctor. Hypo- or hyper-glycemia lines (Red dashed lines) and target glucose levels (Black lines) are labelled.

and the doctor to make recommendations. The weights parameter in the Pareto Dominant Selector can be easily modified to show the preference towards either quality (recommender) or confidence (doctor).

5.4 Overall Performance Evaluation

5.4.1 Introduction

This section includes the overall performance evaluation on two operation mode. Firstly, the recommender uses built-in knowledge in the system only to provide suggestion. Secondly, the recommender combines the knowledge of the system and the expert in action recommendation. The performance is evluated in terms of variance around the target glucose level 112.5 (mg/dl) and the length of hyper- or hypo-glycemia condition after taking suggested actions.

Regarding parameter settings, the following experiments use linear confidence function and Pareto Dominant Selector, without the pretraining process. Experiments simulate the glucose variation of T1DM patients as an example and it is assumed that the virtual patient always takes the actions suggested by the recommender.

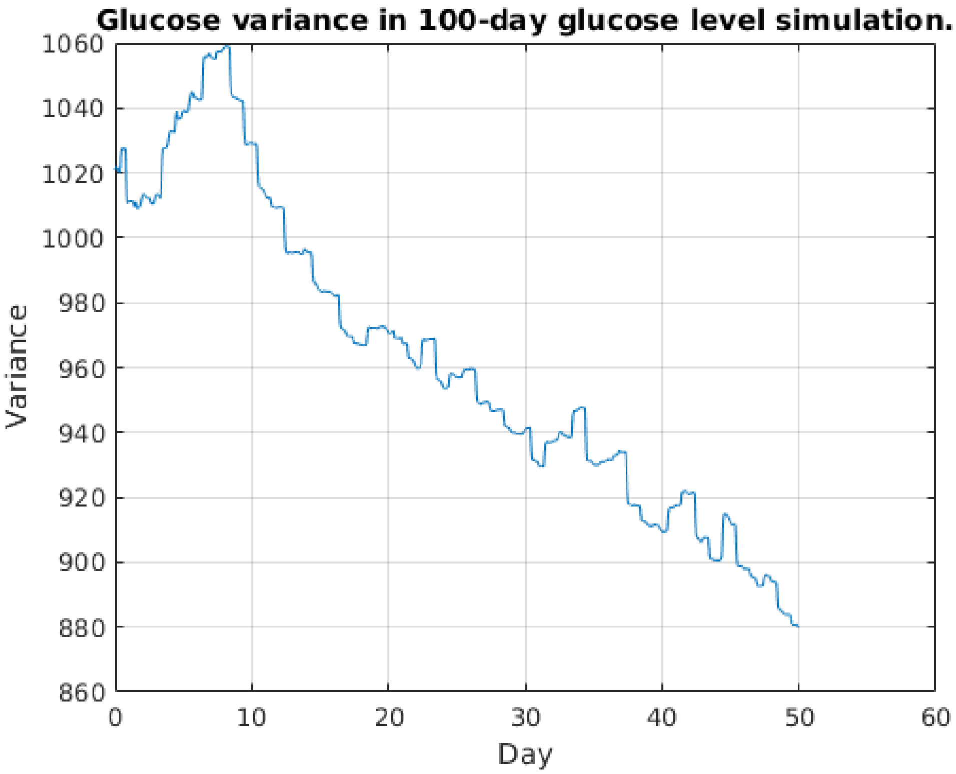
5.4.2 Recommending Using Built-in Knowledge Only

When the recommender is operating alone, it utilises the built-in knowledge to justify action quality and deliver actions. The experiment that simulates the 100-day glucose variations of a T1DM patient. The computed variances are averaged over ten simulations is shown in the figure 5.10.

The top figure in the figure 5.10 shows the glucose variation around the target level through the 100-day simulations. The variance is calculated as the glucose variation around the target level over 50-day period. A long period (50-day) is chosen to find the variance and the plotted variance is averaged over ten simulations, in order to reduce the noise and find the trend.

The bottom figure in the figure 5.10 shows the duration change of hypo- and hyper-glycemia. Each point represents the length of the condition in hours in the 10-day interval. Note that the small value at the beginning of the simulation is caused by the limitation of the virtual patient, since the glucose variation is simulated from a steady state.

It can be observed that the glycemic variance is reduced significantly throughout the simulation and the duration of hypo- and hyper-glycemia is also reduced throughout the simulation. It indicates that glycemic control is successfully introduced by the recommender system and the recommender is improving over time owing to its adaptive feature.



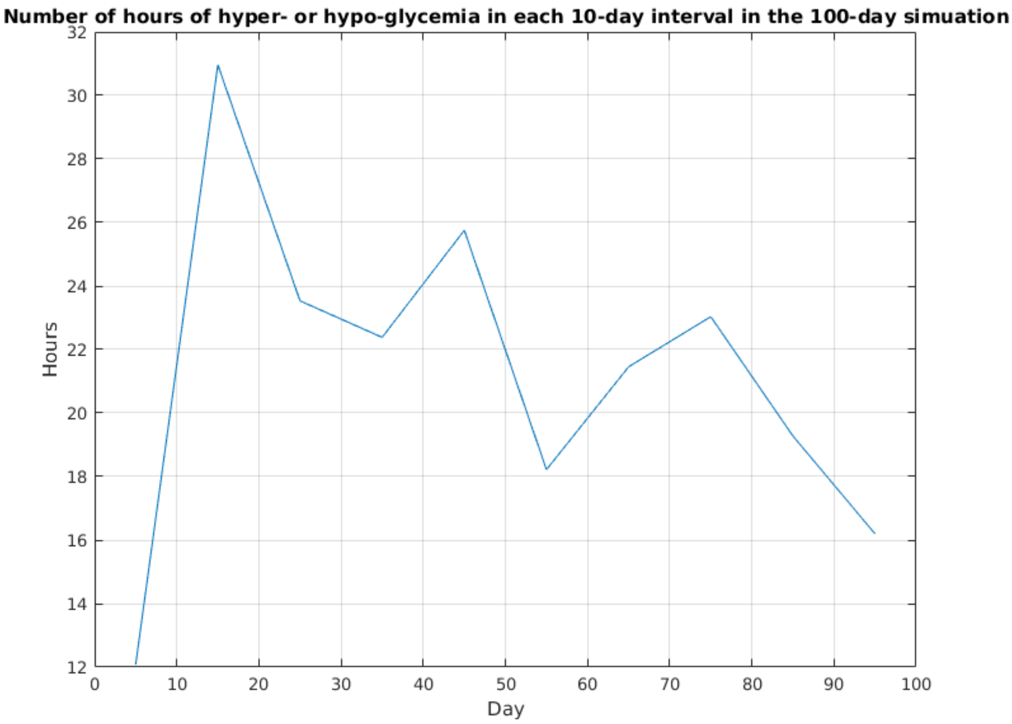


Figure 5.10: Glucose variance throughout the 100-day simulation (Top). Duration of hyper- and hypo-glycemia condition for each 10-day interval in the 100-day simulation (Bottom).

5.4.3 Recommending Using Combined Knowledge

Assuming there is an expert available to demonstrate the ideal treatment strategy for the patient, the recommender system is able to adapt to the actions of the expert. During the demonstration of the expert, the recommender system increase the confidence levels of the inverse models that are leads to similar result as the expert. The modified confidence levels are then kept in the recommender as the knowledge of the expert. Confidence levels contribute to higher scores in the action selection process.

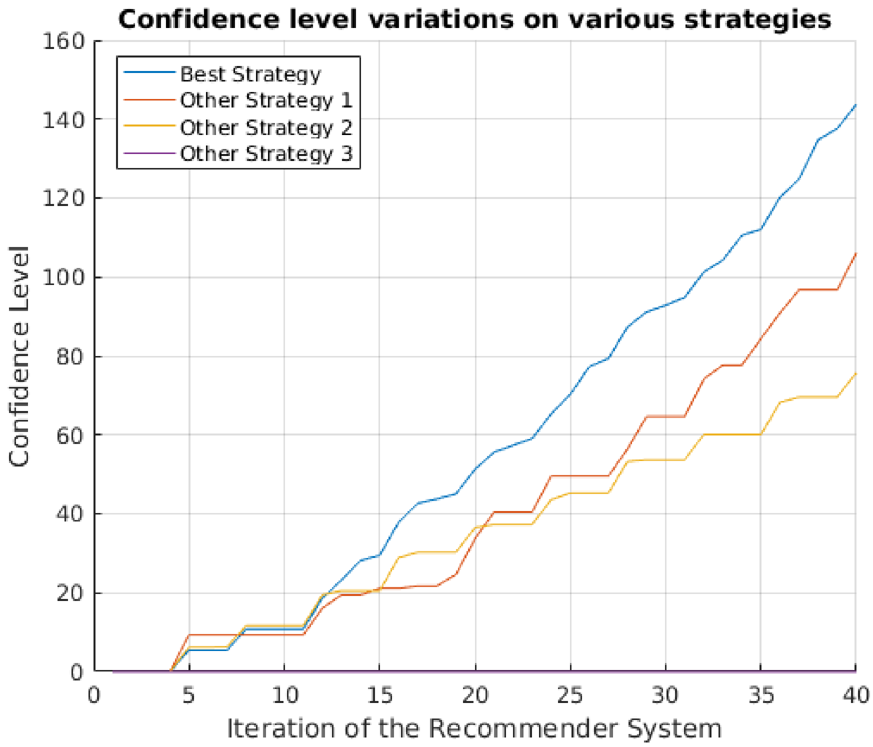


Figure 5.11: Model confidence level variations during expert's demonstration.

The figure 5.11 has shown that the modification of model confidence levels throughout the demonstration of the expert. The model assigned with the highest confidence level (blue line) leads to the most similar outcome as the expert and therefore, has higher opportunity to be selected by the Pareto Dominant Selector.

After the expert demonstrating its actions, the recommender replaces the expert to provide suggestions to the virtual patient.

The figure 5.12 shows the variance of the glucose level using system knowledge only and using

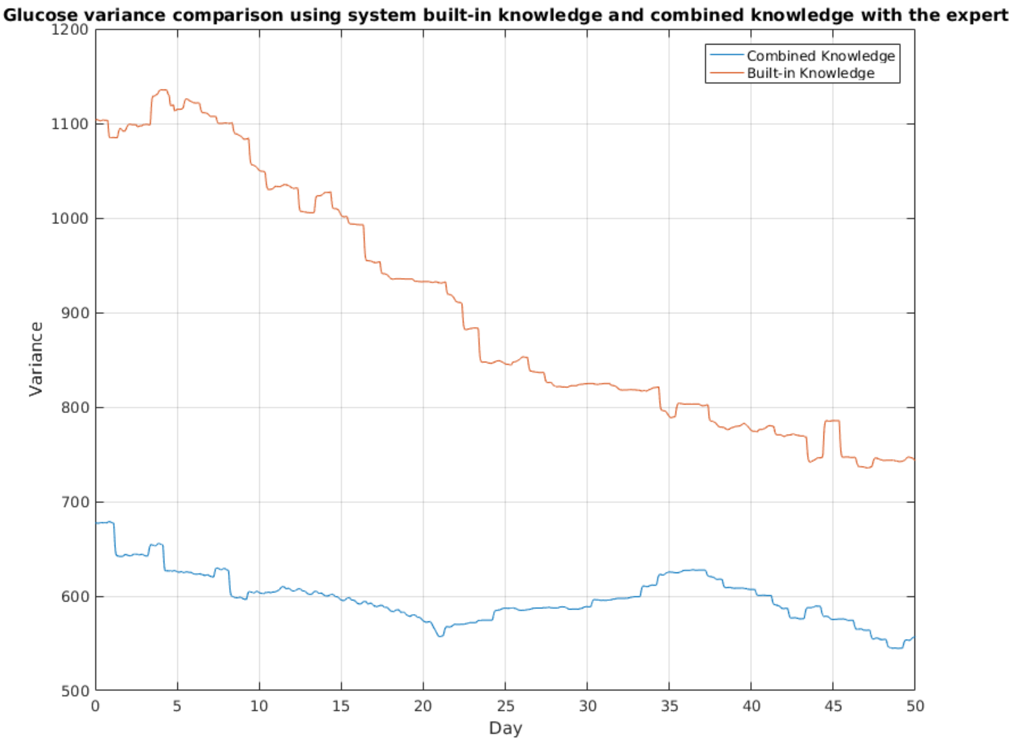


Figure 5.12: Glucose variance comparison with and without the knowledge of the expert.

the combined knowledge of the system and the expert. The variance value is computed by as the glucose variation around the target value over the 50-day period.

It can be observed that the glucose variation after including the expert's knowledge is significantly less the the variation using system built-in knowledge throughout the simulation process. The overall variance in the simulation using the combined knowledge is approximately 40% less than the simulation using system knowledge only. The performance improvement is more obvious when only built-in knowledge is used (Orange line). The downward slope trend in the scenario of combined knowledge is also observable. In the simulation shown in the figure 5.12, the decrease in the 50-day glucose variance is approximately 30% and 12% for using built-in knowledge and combined knowledge respectively.

To sum up, the recommender system is able to provide recommendations using its built-in knowledge. The overall performance improves over time owing to the feature of adaptivity. If an expert with perfect knowledge is available, the recommender system is able to learn from its demonstration and improve its performance in the recommendation process.

Chapter 6

Results and Discussion

6.1 The Final Deliverable

This thesis has proposed an adaptive multi-model recommender system, which is capable of using combined knowledge from the system and the expert (if available) to provide suggestions on diabetes management. It benefits from its adaptivity and has shown improved performance in glycemic control over time. The proposed recommender system is a distributed, hierarchical multi-model architecture. The architecture of the recommender system is not explicitly designed for the diabetic treatment and it has a great potential to extend current implementation to cover a wider range of aspects in diabetic treatment in the real world.

The component implementation is specifically designed for the application of diabetic treatment. The implemented components include inverse models, forward models, confidence functions and action selectors. Inverse model implementations cover the most common insulin strategies and diet plans. The forward model is implemented using Gaussian process regression, a non-parametric supervised machine learning algorithm, to simulate patient response to medical interventions. Various options of confidence functions and action selection algorithms have been implemented and discussed.

The virtual patient implementation follows the design described in [2, 68], which uses forcing function strategies to simulate the glucose-insulin dynamics of human body. Three types of patients, including T1DM, T2DM and normal patient, are simulated by modifying initial conditions and parameter settings. The meal and glucose intervention are also included as inputs in the simulation.

6.2 Experiment Results and Discussion

The project experiments include both component test and overall performance evaluation of the recommender. The overall recommender has shown to introduce effective glycemic control using the built-in knowledge. Using the knowledge of the expert improves the result of simulated glycemic control, shown as the smaller glucose variance compared to using built-in knowledge only. In both scenarios, the recommender shows the improved performance over time.

In the 'Test and Evaluation' section, the virtual patient simulations for T1DM, T2DM and non-diabetic people are included. After that, different design options and parameter settings are discussed. It has been concluded that the linear confidence function and Pareto Dominant Selector outperform other options listed in the thesis. The hierarchical model structure improves the recommender performance, compared to using low-level model in action recommendation. The forward model pretraining improves the recommender performance, but its feasibility requires further research.

Chapter 7

Conclusion and Future Work

7.1 Thesis Achievements

This thesis has proposed an adaptive multi-model recommender system for diabetes management. It is capable of providing recommendation using the combined the knowledge of the recommender system and the expert. The final deliverable contains a general architecture based on the 'HAMMER' and explicit component implementations for diabetes treatment. The project also implemented a virtual diabetic patient for testing purpose. This thesis also discusses various design choices and shows the experiment result on different options and parameter settings.

7.2 Future Work

The proposed recommender system is a prototype that has shown reasonable performance in glycemic control in the virtual patient simulation. It has the great potential to be further developed and utilised in the real world. The potential development areas include improving model implementation and extending the current application to cover other topics in diabetes management.

7.2.1 Improvement on Current Implementation

This section explains the potential improvement on inverse models, forward models.

Inverse models that are hardcoded in the final deliverable including sliding-scale and carb-counting fast insulin method, long-acting insulin method and various diet plans. These strategies are common strategies used in recent diabetes healthcare, but currently the hard-coded treatment strategies are static. Adaptivity can be introduced into inverse models to enable personalised treatment strategy. For example, the detection error between the predicted and actual states can be used to adjust inverse model parameters. Furthermore, the recommender system has limited ability in learning from the expert if the demonstrated action is not similar to any hardcoded inverse model. As a result, machine learning techniques can be utilised to imitate the action of the demonstrator. Inverse models with learning ability are most likely to outperform hardcoded models when there is a good demonstrator give.

Regarding forward model implementation, the Gaussian process regression, a non-parametric machine learning algorithm is selected as an example in the final deliverable. In Gaussian process regression, there are a range of covariance functions that are available [76]. The performance of GPR depends on the hyperparameters such as l, σ, which are estimated in the final deliverable using training data dimensions. However, hyperparameters can be estimated through Maximum a posteriori estimation which incorporates prior distribution in the maximum likelihood estimation. The careful selection of hyperparameters might improve the forward model performance in prediction future patient state.

7.2.2 Extension to Other Aspects

This project choose glycemic control as the starting point of the implementation because glycemic control can be easily quantified and evaluated with the virtual patient. In diabetes management, there are a wide range of topic to investigate and most of these aspects are difficult to quantify and implement.

Research has done to explore areas to cover in diabetes management. This section discusses the aspects that should be taken into consideration but are difficult to implement from an engineering perspective.

This project proposed four main areas that worth consideration to construct a complete recommender system in diabetes management, namely, physical status, psychological status, self-management ability and user engagement with the recommender system.

Physical Status

The final deliverable of the recommender system has a specific focus on the glycemic control. However, there are more than one way to determine the goal of glycemic control and the glucose level is not the only gauge.

Generally speaking, the physical health in diabetes management can be measured by glucose level, blood pressure, lipid and long-term risk associated with diabetes. Regarding the treatment targets in glycemic control, it can be measured by Glycated hemoglobin, Preprandial blood glucose and 2-hour postprandial blood glucose [83, 84]. The long-term health target covers a range of complications associated with diabetes mellitus, such as life expectancy, cardiovascular disease, etc [84].

Psychological Status

Traditionally, glycemic control has been the major target in diabetes management. But the very demanding treatment regime such as diet, insulin and exercise often results in poor adherence and negative mood. The psychological status is complicated to evaluate and quantify. The quantitative research in psychological study describes the characteristics of social phenomena in terms of acts, meanings, relationships, settings, participation and activities. The 'Handbook of psychology and diabetes' has proposed detailed quantitative measure on the psychological status of diabetic patients, mainly in the form of questionnaire [85]. Early research has developed the diabetes quality-of-life (DQOL) measure, which has been employed in a variety of studies [86]. Another study has concluded that psychological intervention is likely to improve depression and life quality of diabetic patients [87].The psychological interventions used in the study includes social support, relaxation, behaviour therapy, cognitive therapy, biofeedback and a set of educational program [87].

Self-management Ability

The self-management ability is another important aspect to consider in diabetes management. Compared to physical and psychological targets, which are direct target of diabetes management, improving the self-management ability of diabetic patient is particularly important in the long-term treatment process of diabetes mellitus.

A Health Education Impact Questionnaire (heiQ) has been developed to measure the patient education and self-management ability in chronic disease management [88]. It has categorised the self-management target into eight sections: life engagement, health directed behaviour, skill acquisition, constructive attitude, self-monitor, health service navigation, social integration and emotional well-being [88]. Rigorous evaluation has been done on questionnaire items and scales [88].

User Engagement

Similar to the self-management ability, the user engagement of the recommender system has important influence on the long run diabetes management process. It might have effects on the user adherence to the recommended treatment regime and psychological health of diabetic patients. The attributes of engagement with technology suggested by previous research include: Aesthetic appeal, attention, challenge, endurability, feedback, interactivity, perceived user control, pleasure, sensory appeal and novelty [89]. The study does not include the quantitative measure of user engagement, but have mentioned the three main experience threads in user engagement: the sensory thread, the emotional thread and the spatiotemporal thread. Apart from questionnaire design, which is commonly used to quantify abstract measures, other potential quantitative measure of user engagement include the frequency and duration of user engagement, the acoustic emotion of patient and the attention probe using eye-tracking techniques.

7.2.3 Integration with Mobile Apps

The recommender system is designed as the back-end of mobile applications and it is possible to integrate the recommender with front-end user interface.

The front-end user interface is not implemented in this project, owing to time limit. But similar work has been done in the 'Mobile Healthcare and Machine Learning' Project, in which an IOS app has been developed to track diabetes management results. In that project, a simple version of the recommender is wrapped into the Tomcat Server using RESTful API so that the recommender is capable of providing cloud service to the user end. The recommender system can be easily integrated with the app to form a complete system and to provide user service.

Appendix A

UML diagram of current HAMMER implementation

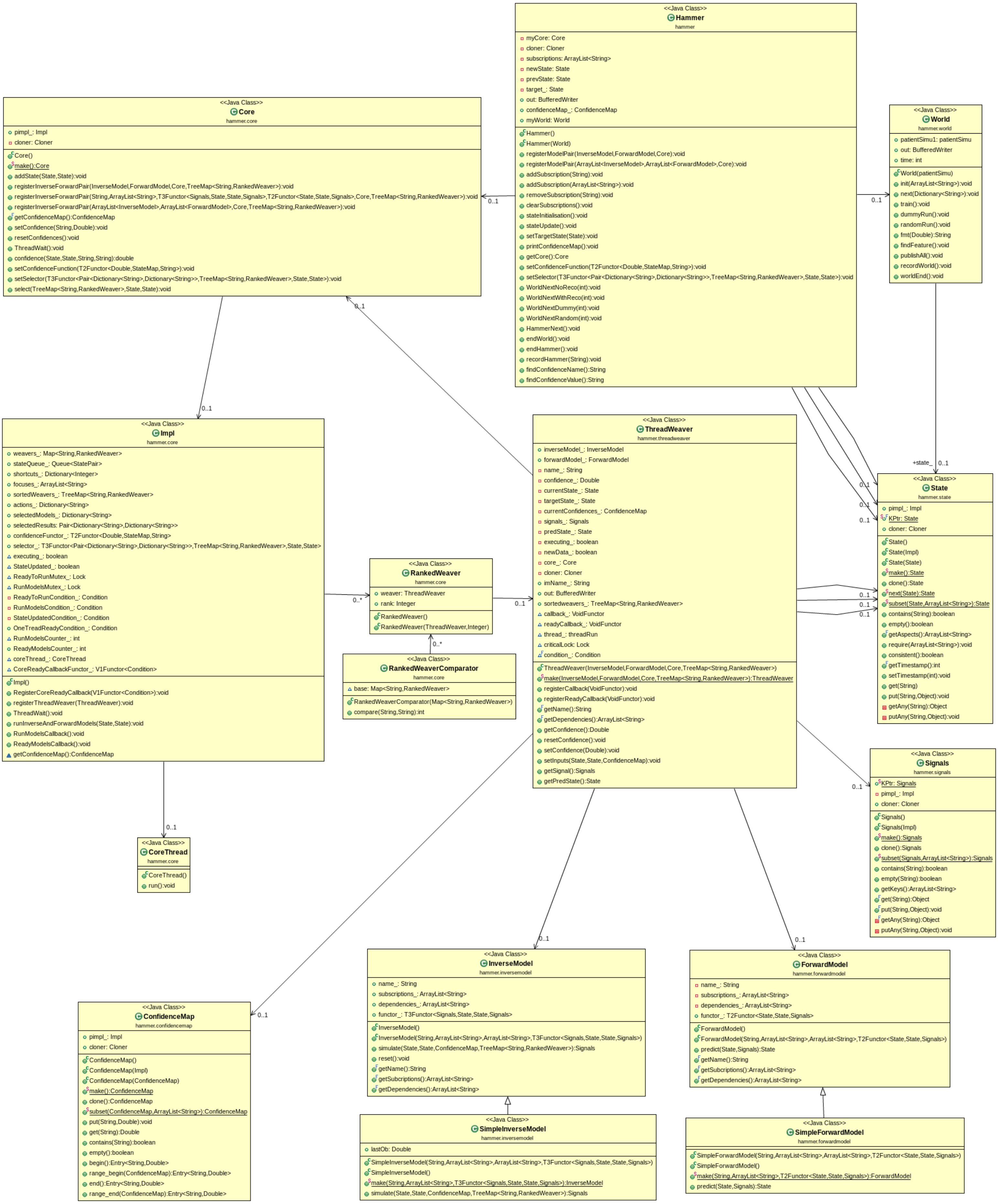


Figure A.1: The structures of an inverse model(top) and forward model(bottom).

Appendix B

Virtual Patient Parameter setting for T1DM, T2DM and normal patients

Process	Parameter	Normal Value	T1DDM	T2DDM	Unit
Glucose Kinetics	V_G	1.88	1.88	1.49	dl/kg
	k_1	0.065	0.065	0.042	min^{-1}
	k_2	0.079	0.079	0.071	min^{-1}
Insulin Kinetics	V_I	0.05	0.05	0.04	l/kg
	m_1	0.19	0.190	0.379	min^{-1}
	m_2	0.484	0.484	0.673	min^{-1}
	m_4	0.194	0.128	0.269	min^{-1}
	m_5	0.0304	0.0274	0.0526	$min * kg/pmol$
	m_6	0.6471	0.5824	0.8118	$dimensionless$
	HE_b	0.6	0.6	0.6	$dimensionless$
Rate of Appearance	k_{max}	0.0558	0.0558	0.0465	min^{-1}
	k_{min}	0.008	0.008	0.0076	min^{-1}
	k_{abs}	0.057	0.057	0.023	min^{-1}
	k_{gri}	0.0558	0.0558	0.0465	min^{-1}
	f	0.9	0.9	0.9	$dimensionless$
	a	0.00013	0.01	0.00006	min^{-1}
	b	0.82	0.82	0.68	$dimensionless$
	c	0.00236	0.00236	0.00023	min^{-1}
	d	0.01	0.01	0.09	$dimensionless$
Endogenous Production	k_{p1}	2.7	3.645	3.09	$mg/lg/min$
	k_{p2}	0.0021	0.0021	0.0007	min^{-1}
	k_{p3}	0.009	0.0594	0.005	$mg/kg/minperpmol/l$
	k_{p4}	0.0618	0.4079	0.0786	$mg/kg/minperpmol/ke$
	k_i	0.0079	0.0079	0.0066	min^{-1}
Secretion	F_{cns}	1	1	1	$mg/kg/min$
	V_{m0}	2.5	2.5	4.65	$mg/kg/min$
	V_{mx}	0.047	0.047	0.034	$mg/kg/minperpmol/l$
	K_{m0}	225.59	225.59	446.21	mg/kg
	p_{2U}	0.0331	0.02185	0.0840	min^{-1}
Renal Excretion	K	2.3	0.759	0.99	$pmol/kgper(mg/dl)$
	α	0.05	0.15	0.013	min^{-1}
	β	0.11	0.33	0.05	$pmol/kg/minper(mg/dl)$
	γ	0.5	1.5	0.5	min^{-1}
Glucose Kinetics	k_{e1}	0.0005	0.0005	0.0007	min^{-1}
	k_{e2}	339	339	269	mg/kg

Table B.1: Virtual patient parameter settings for T1DM, T2DM and normal patient. [2]

Bibliography

[1] Wikipedia, "Insulin (medication)." https://en.wikipedia.org/wiki/Insulin_(medication), 2016.

[2] C. D. Man, R. A. Rizza, and C. Cobelli, "Meal simulation model of the glucose-insulin system," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 10, pp. 1740–1749, 2007.

[3] S.-M. Lee, D. Koh, W. K. Chui, and C.-F. Sum, "Diabetes management and hypoglycemia in safety sensitive jobs," *Safety and health at work*, vol. 2, no. 1, pp. 9–16, 2011.

[4] M. Johnson and Y. Demiris, "Abstraction in recognition to solve the correspondence problem for robot imitation," *Proceedings of TAROS*, pp. 63–70, 2004.

[5] Y. Demiris and B. Khadhouri, "Hierarchical attentive multiple models for execution and recognition of actions," *Robotics and autonomous systems*, vol. 54, no. 5, pp. 361–369, 2006.

[6] Q. Wang, P. Molenaar, S. Harsh, K. Freeman, J. Xie, C. Gold, M. Rovine, and J. Ulbrecht, "Personalized state-space modeling of glucose dynamics for type 1 diabetes using continuously monitored glucose, insulin dose, and meal intake an extended kalman filter approach," *Journal of diabetes science and technology*, vol. 8, no. 2, pp. 331–345, 2014.

[7] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 6, pp. 2193–2196, 2012.

[8] D. Control, C. Trial, *et al.*, "Intensive diabetes treatment and cardiovascular disease in patients with type 1 diabetes," *The New England journal of medicine*, vol. 353, no. 25, p. 2643, 2005.

[9] Y. Ohkubo, H. Kishikawa, E. Araki, T. Miyata, S. Isami, S. Motoyoshi, Y. Kojima, N. Furuyoshi, and M. Shichiri, "Intensive insulin therapy prevents the progression of diabetic

microvascular complications in japanese patients with non-insulin-dependent diabetes mellitus: a randomized prospective 6-year study," *Diabetes research and clinical practice*, vol. 28, no. 2, pp. 103–117, 1995.

[10] C. M. Renders, G. D. Valk, S. J. Griffin, E. H. Wagner, W. J. Assendelft, *et al.*, "Interventions to improve the management of diabetes in primary care, outpatient, and community settings a systematic review," *Diabetes care*, vol. 24, no. 10, pp. 1821–1833, 2001.

[11] A. Sekmen and P. Challa, "Assessment of adaptive human–robot interactions," *Knowledge-Based Systems*, vol. 42, pp. 49–59, 2013.

[12] C. D. Man, R. A. Rizza, and C. Cobelli, "Meal simulation model of the glucose-insulin system," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 10, pp. 1740–1749, 2007.

[13] L. Magni, D. M. Raimondo, L. Bossi, C. Dalla Man, G. De Nicolao, B. Kovatchev, and C. Cobelli, "Model predictive control of type 1 diabetes: an in silico trial," *Journal of diabetes science and technology*, vol. 1, no. 6, pp. 804–812, 2007.

[14] M. R. Documentation, "Simulating the glucose-insulin response." `http://uk.mathworks.com/help/simbio/examples/simulating-the-glucose-insulin-response.html`, 2016.

[15] N. Bentzen, *WONCA dictionary of general/family practice.* Wonca International Classification Committee, 2003.

[16] E. H. Wagner, B. T. Austin, C. Davis, M. Hindmarsh, J. Schaefer, and A. Bonomi, "Improving chronic illness care: translating evidence into action," *Health affairs*, vol. 20, no. 6, pp. 64–78, 2001.

[17] D. Abegunde and A. Stanciole, "An estimation of the economic impact of chronic non-communicable diseases in selected countries," *World Health Organization, Department of Chronic Diseases and Health Promotion*, 2006.

[18] K. G. M. M. Alberti and P. f. Zimmet, "Definition, diagnosis and classification of diabetes mellitus and its complications. part 1: diagnosis and classification of diabetes mellitus. provisional report of a who consultation," *Diabetic medicine*, vol. 15, no. 7, pp. 539–553, 1998.

[19] C. for Disease Control, P. (CDC), C. for Disease Control, P. (CDC), *et al.*, "National diabetes fact sheet: national estimates and general information on diabetes and prediabetes in the united states, 2011," *Atlanta, GA: US Department of Health and Human Services, Centers for Disease Control and Prevention*, vol. 201, 2011.

[20] A. Alwan *et al.*, *Global status report on noncommunicable diseases 2010.* World Health Organization, 2011.

[21] K. R. Lorig, D. S. Sobel, P. L. Ritter, D. Laurent, and M. Hobbs, "Effect of a self-management program on patients with chronic disease.," *Effective clinical practice: ECP*, vol. 4, no. 6, pp. 256–262, 2000.

[22] T. Bodenheimer, K. Lorig, H. Holman, and K. Grumbach, "Patient self-management of chronic disease in primary care," *Jama*, vol. 288, no. 19, pp. 2469–2475, 2002.

[23] S. Clement, "Diabetes self-management education," *Diabetes care*, vol. 18, no. 8, pp. 1204–1214, 1995.

[24] "Insulin." `https://www.diabetes.org.uk/Guide-to-diabetes/What-is-diabetes/Diabetes-treatments/Insulin/`, 2016.

[25] P. Sonksen and J. Sonksen, "Insulin: understanding its action in health and disease," *British journal of anaesthesia*, vol. 85, no. 1, pp. 69–79, 2000.

[26] M. J. Williams, "Jjr macleod: the co-discoverer of insulin.," *Proceedings of the Royal College of Physicians of Edinburgh*, vol. 23, no. 3, p. 1, 1993.

[27] I. Tof, "Recombinant dna technology in the synthesis of human insulin," *Little Tree Publishing, retrieved*, vol. 19, p. 2010, 1994.

[28] E. Farnsworth, N. D. Luscombe, M. Noakes, G. Wittert, E. Argyiou, and P. M. Clifton, "Effect of a high-protein, energy-restricted diet on body composition, glycemic control, and lipid concentrations in overweight and obese hyperinsulinemic men and women," *The American journal of clinical nutrition*, vol. 78, no. 1, pp. 31–39, 2003.

[29] A. E. Järvi, B. E. Karlström, Y. E. Granfeldt, I. E. Björck, N. Asp, and B. Vessby, "Improved glycemic control and lipid profile and normalized fibrinolytic activity on a low-glycemic index diet in type 2 diabetic patients.," *Diabetes Care*, vol. 22, no. 1, pp. 10–18, 1999.

[30] J. P. Bantle, J. Wylie-Rosett, A. L. Albright, C. M. Apovian, N. G. Clark, M. J. Franz, B. J. Hoogwerf, A. H. Lichtenstein, E. Mayer-Davis, A. D. Mooradian, *et al.*, "Nutrition recommendations and interventions for diabetes: a position statement of the american diabetes association.," *Diabetes care*, vol. 31, pp. S61–S78, 2008.

[31] A. Garg, J. P. Bantle, R. R. Henry, A. M. Coulston, K. A. Griver, S. K. Raatz, L. Brinkley, Y. I. Chen, S. M. Grundy, B. A. Huet, *et al.*, "Effects of varying carbohydrate content of diet in patients with noninsulin-dependent diabetes mellitus," *Jama*, vol. 271, no. 18, pp. 1421–1428, 1994.

[32] T. G. Kiehm, J. W. Anderson, and K. Ward, "Beneficial effects of a high carbohydrate, high fiber diet on hyperglycemic diabetic men.," *The American journal of clinical nutrition*, vol. 29, no. 8, pp. 895–899, 1976.

[33] S. Adibi, *Mobile Health: A Technology Road Map*, vol. 5. Springer, 2015.

[34] Cellunar News, "Global Mobile Phone Subscribers to Reach 4.5 Billion by 2012." `http://www.cellular-news.com/story/29824.php`, 2008.

[35] D. El Boghdady, "Health-care apps for smartphones pit fda against tech industry," *The Washington Post*, 2012.

[36] A. C. Norris, R. Stockdale, and S. Sharma, "A strategic approach to m-health," *Health informatics journal*, vol. 15, no. 3, pp. 244–253, 2009.

[37] V. W. Consulting, "mhealth for development: the opportunity of mobile technology for healthcare in the developing world," *Washington Dc and Berkshire, UK*, 2009.

[38] D. C. Klonoff, "The current status of mhealth for diabetes: will it be the next big thing?," *Journal of diabetes science and technology*, vol. 7, no. 3, pp. 749–758, 2013.

[39] O. El-Gayar, P. Timsina, N. Nawar, and W. Eid, "Mobile applications for diabetes self-management: status and potential," *Journal of diabetes science and technology*, vol. 7, no. 1, pp. 247–262, 2013.

[40] D. L. Hunt, R. B. Haynes, S. E. Hanna, and K. Smith, "Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review," *Jama*, vol. 280, no. 15, pp. 1339–1346, 1998.

[41] R. Kaushal, K. G. Shojania, and D. W. Bates, "Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review," *Archives of internal medicine*, vol. 163, no. 12, pp. 1409–1416, 2003.

[42] J. W. Bennett and P. P. Glasziou, "Computerised reminders and feedback in medication management: a systematic review of randomised controlled trials," *Medical Journal of Australia*, vol. 178, no. 5, pp. 217–222, 2003.

[43] R. Walton, E. Harvey, S. Dovey, and N. Freemantle, "Computerised advice on drug dosage to improve prescribing practice," *The Cochrane Library*, 2001.

[44] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach, "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success," *Bmj*, vol. 330, no. 7494, p. 765, 2005.

[45] D. E. Kanouse, J. D. Kallich, and J. P. Kahan, "Dissemination of effectiveness and outcomes research," *Health policy*, vol. 34, no. 3, pp. 167–192, 1995.

[46] B. Kaplan, "Evaluating informatics applicationssome alternative approaches: theory, social interactionism, and call for methodological pluralism," *International journal of medical informatics*, vol. 64, no. 1, pp. 39–56, 2001.

[47] C. D. MacLean, B. Littenberg, and M. Gagnon, "Diabetes decision support: initial experience with the vermont diabetes information system," *American Journal of Public Health*, vol. 96, no. 4, pp. 593–595, 2006.

[48] D. F. Lobach and W. E. Hammond, "Computerized decision support based on a clinical practice guideline improves compliance with care standards," *The American journal of medicine*, vol. 102, no. 1, pp. 89–98, 1997.

[49] S. Hammer, J. Kim, and E. André, "Med-styler: Metabo diabetes-lifestyle recommender," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 285–288, ACM, 2010.

[50] A. S. Hussein, W. M. Omar, X. Li, and M. A. Hatem, "Smart collaboration framework for managing chronic disease using recommender system," *Health Systems*, vol. 3, no. 1, pp. 12–17, 2014.

[51] T. Chomutare, E. Arsand, and G. Hartvigsen, "Mobile peer support in diabetes.," *Studies in health technology and informatics*, vol. 169, pp. 48–52, 2010.

[52] N. Suksom, M. Buranarach, Y. M. Thein, T. Supnithi, and P. Netisopakul, "A knowledge-based framework for development of personalized food recommender system," in *Proc. of the 5th Int. Conf. on Knowledge, Information and Creativity Support Systems*, 2010.

[53] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan, "An internal model for sensorimotor integration," *Science*, vol. 269, no. 5232, p. 1880, 1995.

[54] K. Dautenhahn and C. Nehaniv, "Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model,"

[55] Y. Demiris and M. Johnson, "Simulation theory of understanding others: A robotics perspective," 2006.

[56] M. Johnson and Y. Demiris, "Abstraction in recognition to solve the correspondence problem for robot imitation," *Proceedings of TAROS*, pp. 63–70, 2004.

[57] A. Sekmen and P. Challa, "Assessment of adaptive human–robot interactions," *Knowledge-Based Systems*, vol. 42, pp. 49–59, 2013.

[58] G. I. Webb, M. J. Pazzani, and D. Billsus, "Machine learning for user modeling," *User modeling and user-adapted interaction*, vol. 11, no. 1-2, pp. 19–29, 2001.

[59] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: learning new user preferences in recommender systems," in *Proceedings of the 7th international conference on Intelligent user interfaces*, pp. 127–134, ACM, 2002.

[60] K. A. Bruffee, *Collaborative learning: Higher education, interdependence, and the authority of knowledge.* ERIC, 1999.

[61] N. Li, W. W. Cohen, K. R. Koedinger, and N. Matsuda, "A machine learning approach for automatic student model discovery.," in *EDM*, pp. 31–40, 2011.

[62] A. Bunt and C. Conati, "Probabilistic student modelling to improve exploratory behaviour," *User Modeling and User-Adapted Interaction*, vol. 13, no. 3, pp. 269–309, 2003.

[63] S. Schiaffino, P. Garcia, and A. Amandi, "eteacher: Providing personalized assistance to e-learning students," *Computers & Education*, vol. 51, no. 4, pp. 1744–1754, 2008.

[64] S. M. Parvez and G. D. Blank, "Individualizing tutoring with learning style based feedback," in *Intelligent tutoring systems*, pp. 291–301, Springer, 2008.

[65] A. Al-Hmouz, J. Shen, J. Yan, and R. Al-Hmouz, "Enhanced learner model for adaptive mobile learning," in *Proceedings of the 12th international conference on information integration and web-based applications & services*, pp. 783–786, ACM, 2010.

[66] P. S. Inventado, R. Legaspi, T. Bui, and M. Suarez, "Predicting students appraisal of feedback in an its using previous affective states and continuous affect labels from eeg data," in *Proceedings of the 18th International Conference on Computers in Education, Putrajaya, Malaysia*, 2010.

[67] A. Al-Hmouz, J. Shen, J. Yan, and R. Al-Hmouz, "Enhanced learner model for adaptive mobile learning," in *Proceedings of the 12th international conference on information integration and web-based applications & services*, pp. 783–786, ACM, 2010.

[68] L. Magni, D. M. Raimondo, L. Bossi, C. Dalla Man, G. De Nicolao, B. Kovatchev, and C. Cobelli, "Model predictive control of type 1 diabetes: an in silico trial," *Journal of diabetes science and technology*, vol. 1, no. 6, pp. 804–812, 2007.

[69] R. P. Cooper, "Forward and inverse models in motor control and cognitive control," in *Symposium on AI-Inspired Biology, De Montfort University, Leicester, UK. Retrieved from: http://www. cs. bham. ac. uk/research/projects/cogaff/aiib/papers/invited. d/cooper-rpaper-aiib10. pdf*, 2010.

[70] G. Scheiner and B. A. Boyer, "Characteristics of basal insulin requirements by age and gender in type-1 diabetes patients using insulin pump therapy," *Diabetes research and clinical practice*, vol. 69, no. 1, pp. 14–21, 2005.

[71] E. P. Joslin, *A diabetic manual for the mutual use of doctor and patient.* Lea & Febiger, 1919.

[72] G. E. Umpierrez, A. Palacio, and D. Smiley, "Sliding scale insulin use: myth or insanity?," *The American journal of medicine*, vol. 120, no. 7, pp. 563–567, 2007.

[73] I. B. Hirsch, "Sliding scale insulintime to stop sliding," *Jama*, vol. 301, no. 2, pp. 213–214, 2009.

[74] J. Gearhart, J. Duncan 3rd, W. Replogle, R. Forbes, and E. Walley, "Efficacy of sliding-scale insulin therapy: a comparison with prospective regimens.," *Family practice research journal*, vol. 14, no. 4, pp. 313–322, 1994.

[75] D. L. Souto and E. L. Rosado, "Use of carb counting in the dietary treatment of diabetes mellitus," *Nutrición Hospitalaria*, vol. 25, no. 1, pp. 18–25, 2010.

[76] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[77] C. K. Williams and C. E. Rasmussen, "Gaussian processes for regression," 1996.

[78] M. Ebden, "Gaussian processes for regression: A quick introduction," *The Website of Robotics Research Group in Department on Engineering Science, University of Oxford*, 2008.

[79] R. C. Turner, C. A. Cull, V. Frighi, R. R. Holman, U. P. D. S. U. Group, *et al.*, "Glycemic control with diet, sulfonylurea, metformin, or insulin in patients with type 2 diabetes mellitus: progressive requirement for multiple therapies (ukpds 49)," *Jama*, vol. 281, no. 21, pp. 2005–2012, 1999.

[80] C. A. Verdonk, R. A. Rizza, and J. E. Gerich, "Effects of plasma glucose concentration on glucose utilization and glucose clearance in normal man," *Diabetes*, vol. 30, no. 6, pp. 535–537, 1981.

[81] C. Ltd, "Conversion table (si units/traditional units)." `http://www.ctdslab.co.uk/conversion.html`, 2005.

[82] C. Man, M. Camilleri, and C. Cobelli, "A system model of oral glucose absorption: validation on gold standard data," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 12, pp. 2472–2478, 2006.

[83] A. D. Association *et al.*, "Standards of medical care in diabetes-2006," *Diabetes care*, vol. 29, no. 1, p. S4, 2006.

[84] A. D. Association *et al.*, "Standards of medical care in diabetes–2014.," *Diabetes care*, vol. 37, p. S14, 2014.

[85] C. Bradley, *Handbook of psychology and diabetes: a guide to psychological measurement in diabetes research and practice.* Routledge, 2013.

[86] D. R. Group *et al.*, “Reliability and validity of a diabetes quality-of-life measure for the diabetes control and complications trial (dcct),” *Diabetes care*, vol. 11, no. 9, pp. 725–732, 1988.

[87] L. Steed, D. Cooke, and S. Newman, “A systematic review of psychosocial outcomes following education, self-management and psychological interventions in diabetes mellitus,” *Patient education and counseling*, vol. 51, no. 1, pp. 5–15, 2003.

[88] R. H. Osborne, G. R. Elsworth, and K. Whitfield, “The health education impact questionnaire (heiq): an outcomes and evaluation measure for patient education and self-management interventions for people with chronic conditions,” *Patient education and counseling*, vol. 66, no. 2, pp. 192–201, 2007.

[89] H. L. O’Brien and E. G. Toms, “What is user engagement? a conceptual framework for defining user engagement with technology,” *Journal of the American Society for Information Science and Technology*, vol. 59, no. 6, pp. 938–955, 2008.