MODELLING AND OPTIMISATION OF BLOOD GLUCOSE CONTROL FOR TYPE 1 DIABETES USING MULTI-PARAMETRIC PROGRAMMING AND MODEL-BASED PREDICTIVE CONTROL (mp-MPC)

By:

Associate Professor Dr Ayub Md. Som

Faculty of Chemical Engineering

Universiti Teknologi MARA, Malaysia

with the supervision of :

Professor Efstratios N. Pistikopoulos

Centre for Process Systems Engineering

Department of Chemical Engineering

Imperial College London

United Kingdom

April 2011

TABLE OF CONTENTS

ABSTRACT

This research attempts to develop a new control algorithm to regulate the blood glucose level (BGL) for Type 1 Diabetes. In doing so, Multi-Parametric Programming technique is used to develop the computer algorithm; whereas Model-Based Predictive Control (MPC) is adopted for the design of the controller. Non-Linear Bergman Minimal Model is used to represent the three compartments; plasma glucose, plasma insulin and effective insulin compartments. All simulation and optimisation works are carried out using gPROMS^{TM}. Two types of observations are made so as to study the performance of the proposed control algorithm mainly, the control of BGL values without meal disturbance and the control of BGL values following meals. For the control of BGL without meal disturbance, it is found that the BGL values increase substantially at first and fluctuate around 80 mg/dL to 130 mg/dL. They then tend to level off at 120 mg/dL for sometimes before dropping drastically to 60 mg/dL. However, the BGL values remain at 80 mg/dL prior to reaching its steady state condition at the end of the simulation work. For the control of BGL values following meals, it is found that there are three peaks occurred, which obviously indicate a sudden change in the BGL values in conjunction with the introduction of Fisher meal effect into the glucose-insulin dynamic system. Three simulation works are carried out using three different algorithms so as to refine the performance of the controller. For all three cases, the BGL profiles are almost the same in which they tend to fluctuate initially around 65 mg/dL to 120 mg/dL prior to levelling off at 80 mg/dL throughout the remaining periods. These results match with the works carried out by the previous workers. The only major difference is that the value of exogenous insulin infusion rate, u is on the higher side. This could be due to different diabetic models used and inconsistency in choosing the units for the different parameters. However, it can be concluded that with the proposed control algorithm, both hypoglycemia and hyperglycemia are avoided and it is hoped that the algorithm can be easily installed in the form of microchip for the benefit of the diabetic patient in the near future.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Drug delivery companies have now faced a big challenge to deliver both existing and emerging drug technologies in a manner that improves the benefits to the patients, healthcare workers and the healthcare system. Areas that are being targeted for improvements through device development include: (Brunner, 2004)

- Improved efficacy
- Reduced side effects
- Continuous dosing (sustained released)
- Reduced pain from administration
- Increase ease of use
- Increased use compliance
- Improved mobility
- Decrease involvement of healthcare workers
- Improved safety for healthcare workers
- Reduced environmental impact (elimination of CFC's)

To provide these benefits, a number of approaches are being or in some cases have been developed. The common thread running through the approaches is the concept of self-administered, targeted, sustained release with increased bioavailability. Although from a patient standpoint the elimination of injections is ideal, indications are that injection will remain a necessary means of drug delivery. To minimise the pain, biohazard, cost and inconvenience associated with injections, companies are working to reduce the negative aspects of this delivery. Along these lines, new implants and time release approaches, i.e. MicroCHIPS (programmable MEMS implant) are under development so as to minimise the number of injections required.

One of the areas which are currently being developed to solve the above problems is advanced computer-based systems that can calculate the most effective, safe dose of a drug for an individual patient by using novel mathematical programming methods. The systems can take account of and control for a multitude of different parameters using information on the patient's make up and medical history, together with data on how different drugs perform and interact with each other. The systems would also be used by researchers to create a biomedical devise to help people with diabetes who inject themselves with insulin to manage their condition. This devise would ultimately read the patient's blood sugar level and then calculate and deliver the most effective dose of insulin, depending on this readout. The overall vision of integrated insulin delivery systems is shown in Figure 1.1 (CPSE Annual Report 2008)

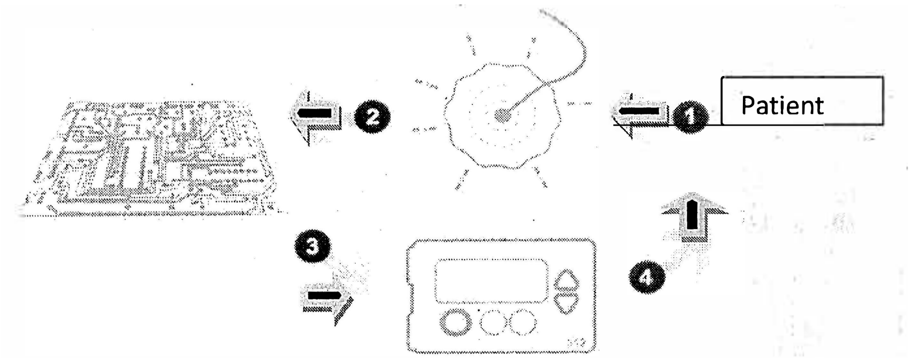


Figure 1.1 Overall vision of integrated insulin delivery systems

As shown in the figure, step 1 measures the glucose concentration from the patient. Step 2, the sensor then inputs the data to the controller which analyses it and implements the algorithm. Step3, after analysing the data the controller then signals the pump to carry out the required action. Step 4, the insulin pump delivers the required dose to the patient, intravenously.

1.1 Objectives of the research

The objectives of this research are as follows:

a) to develop computer algorithm using multi-parametric programming technique and model predictive control (MPC) to control blood glucose level for type 1 diabetes

b) to simulate and optimise the proposed algorithm using gPROMS® prior to being used in the integrated insulin delivery systems.

CHAPTER 2

LITERATURE REVIEW

The role of Chemical Engineers has long been accepted as vital in interfacing Biology and Medicine so as to benefit research efforts worldwide at least for the next decade or so. This has been identified by the United States National Academy of Engineering and National Science Foundation Report entitled 'Beyond the Molecular Frontier'. The system approach instilled by Chemical Engineers, specialising in a process systems engineering, should be able to help train a new type of engineering biologist who can work at the molecular and system wide levels to solve physiological and clinical problems (Bogle *et al.*, 2009).

One of the best examples to demonstrate this role is through numerous studies carried out on blood glucose control for type 1 diabetes. By the year 2025, there will be approximately 300 million people worldwide suffering from diabetes (Dua *et al.*, 2006). Diabetes is a chronic disease characterised by insufficient control of blood glucose concentration in the body. The schematic diagram of the glucose regulation system in the body is shown in Figure 2.1(Bogle *et al.*, 2009). As depicted in Figure 2.1, glucose enters the bloodstream from the gut and is used in all the other organs of the body. It is important that the level of glucose is maintained within certain levels to ensure good response of various human functions, i.e. glucose homeostatis. If glucose levels are high, the pancreas produces insulin, the hormone that instructs liver to convert glucose into glycogen. On the other hand, the glycogen is converted back to glucose as the pancreas produces glucagon if the blood glucose levels are low. As in the case of type 1 diabetes, the destruction of insulin producing beta cells in the pancreas causes the patient to solely rely on blood glucose measurements through regular insulin injections so as to control their desired blood glucose concentration between 60 – 120 mg/dL. A shortage of insulin supply may lead to concentration of blood glucose levels rising above 120 mg/dL and this state is known as hyperglycemia. On the other hand, excessive supply of insulin may lead to the levels of blood glucose concentration is dropping below 60 mg/dL and this state is known as hypoglycemia. The target level of blood glucose concentration should be at 81 mg/dL (Dua *et al.*, 2009).

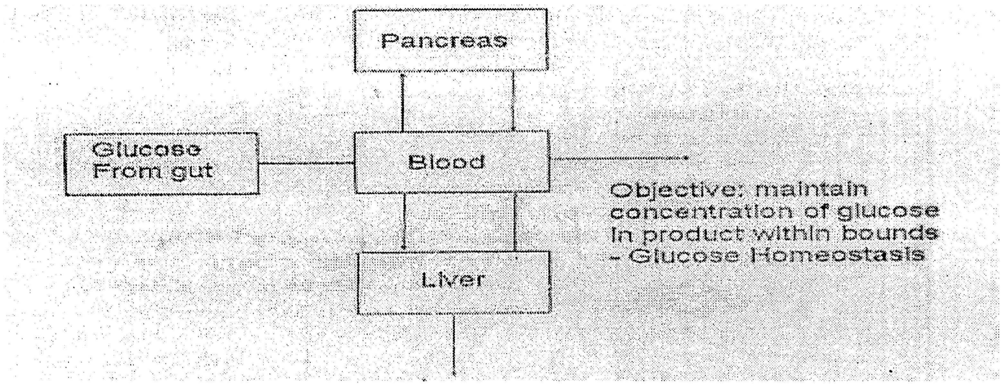

Pancreas
Glucose From gut
Blood
Liver
Objective: maintain concentration of glucose in product within bounds - Glucose Homeostasis

Figure 2.1 Schematic diagram for glucose regulation system in the body

Numerous works pertaining to the control of blood glucose concentration for type 1 diabetes have been carried out and are available in the open literature. These works can be divided into four major areas of applications namely; Simulation Programme, Artificial Intelligence (AI), Mathematical Modelling, and Model-based Predictive Control (MPC).

2.1 Simulation Programme

Lehmann and Deutsch (1995) developed AIDA, a clinical model of glucose-insulin interaction in insulin-dependent (type 1) diabetes for patient and medical staff education. The model attempts to reflect the underlying (patho) physiology of insulin action and carbohydrate absorption in quantitative terms such as insulin sensitivity, volume of glucose, insulin distribution and maximal rate of gastric emptying. The anatomical basis and physiological functions of the AIDA model is shown in Figure 2.2. The model's predictions allow a 24-hour simulation of blood glucose profiles for hypothetical patients to be generated. It is concluded that the model is not refined enough for individual patient simulation and as such the system can only be applied as an educational or demonstration tool.

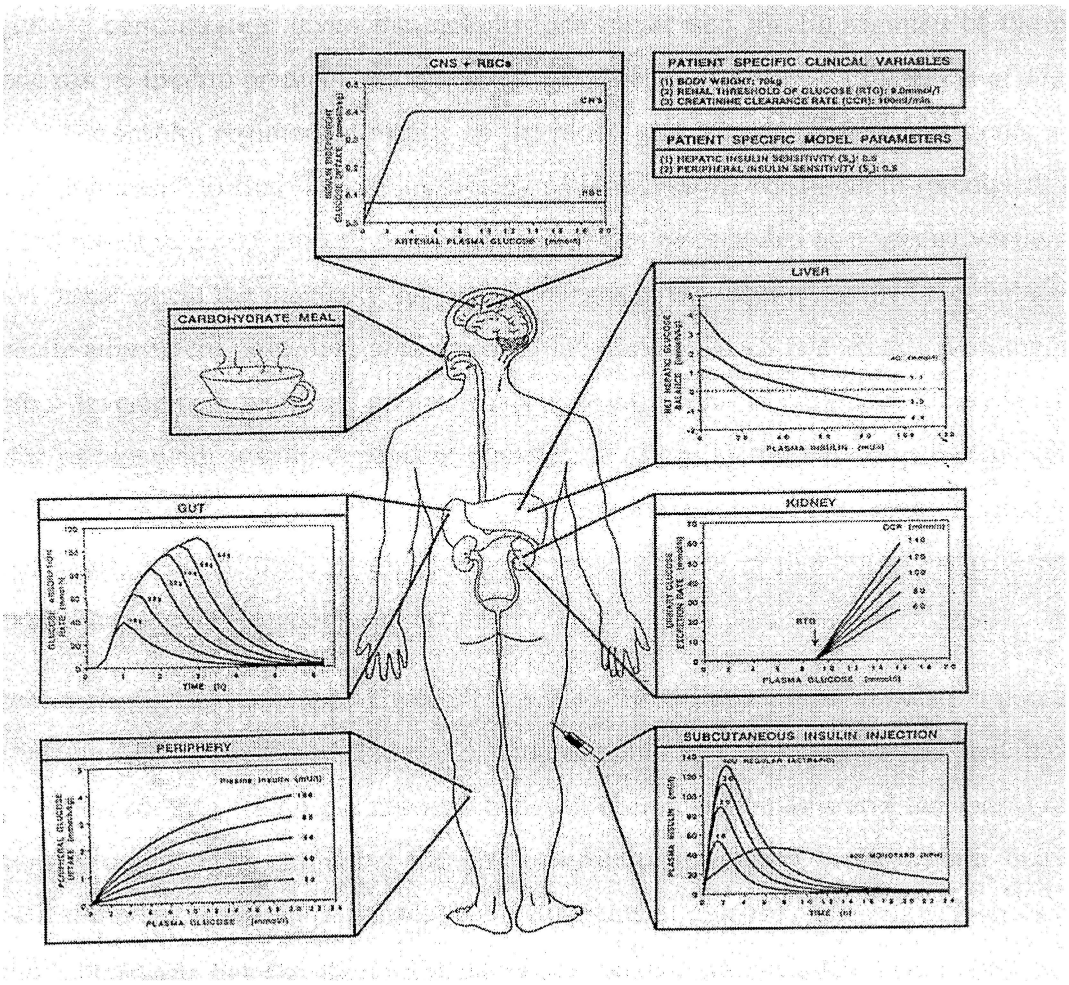

CNS + RBCs
INSULIN INDEPENDENT GLUCOSE UPTAKE
CNS
RBC
ARTERIAL PLASMA GLUCOSE (mmol/l)
PATIENT SPECIFIC CLINICAL VARIABLES
(1) BODY WEIGHT: 70kg
(2) RENAL THRESHOLD OF GLUCOSE (RTG): 9.0mmol/l
(3) CREATININE CLEARANCE RATE (CCR): 100ml/min
PATIENT SPECIFIC MODEL PARAMETERS
(1) HEPATIC INSULIN SENSITIVITY (S_h): 0.5
(2) PERIPHERAL INSULIN SENSITIVITY (S_p): 0.5
CARBOHYORATE MEAL
LIVER
NET HEPATIC GLUCOSE BALANCE
PLASMA INSULIN (mU/l)
GUT
GLUCOSE ABSORPTION RATE
TIME (h)
KIDNEY
URINARY GLUCOSE EXCRETION RATE
CCR
140
120
100
80
60
RTG
PLASMA GLUCOSE (mmol/l)
PERIPHERY
Plasma insulin (mU/l)
PERIPHERAL GLUCOSE UPTAKE
PLASMA GLUCOSE (mmol/l)
SUBCUTANEOUS INSULIN INJECTION
REGULAR (ACTRAPID)
MONOTARD (NPH)
PLASMA INSULIN (mU/l)
TIME (h)

Figure 2.2 Anatomical basis and physiological functions of the AIDA model

Hejlesen *et al.* (1997) developed DIAS, the diabetes advisory system, which provided some forms of intervention for diabetic patients who suffered difficulties in controlling their blood glucose level. DIAS incorporates a model of human carbohydrate metabolism implemented in a Bayesian network (causal probabilistic network or CPN), which gives it the ability to handle the uncertainty; for example, in blood glucose measurements or physiological variations in glucose metabolism. Two adjustable parameters included in the model are namely: the insulin sensitivity and time-to-peak of NPH-type insulin absorption. The system operates in three modes namely; learning, prediction and advisory modes. In learning mode, standard data on blood glucose concentration, insulin injections and carbohydrate content in the meals from one or more days are used to estimate the two adjustable parameters. In the prediction mode, the estimated parameter for insulin sensitivity and time-to-peak of NPH-type insulin are used to make prediction of the blood glucose concentration given the carbohydrate intake and insulin regimen of the patient. This mode can be used to predict unrecognised hypoglycemia or to predict the effect of suggested changes, in the insulin regimen or meals, on the blood glucose. In the advisory mode, a utility measure is minimised to find the insulin therapy which gives the least overall (predicted) risk of too low and too high blood glucose concentrations. It can be regarded as a special version of the prediction mode where the manually suggested changes in the insulin regimen are replaced with an automatic adjustment procedure performed by the system. DIAS is currently a prototype, and with further development work and clinical trials planned, it aims to provide reliable management advice for patient with insulin-dependent diabetes. A study in the UK in patients with well-controlled diabetes showed that DIAS predicted recurrent hypoglycemia at night in more than 50 percent of cases, and that these predictions were more accurate in five out of six patients taken their blood glucose test for hypoglycemia.

Erzen *et al.*, (2002) developed GlucoSIM, a Web-based Educational Simulation package for glucose-insulin levels in human body. An understanding of the physiological and metabolic processes, coupled with forming a network between chemical reactions and transport processes are extremely essential in modelling the glucose-insulin interaction in the human body. Two mathematical models based on pharmacokinetic diagrams of glucose and insulin (Figures 2.3 and 2.4), which represent the transport of glucose and insulin through the major vessels to the capillaries, have been used to develop the simulation package. As shown in Figure 2.3, the glucose diagram contains tissues including heart, brain, liver, kidney and muscle where the glucose is used for energy. The diagram for insulin (Figure 2.4) includes subcutaneous tissue as a source for insulin. The advantage of these models is that the models can yield insight into the physiological processes. However, the disadvantage is that the models do not take into account personal variations in the physiological parameters which, in turn, give only average values for the output.

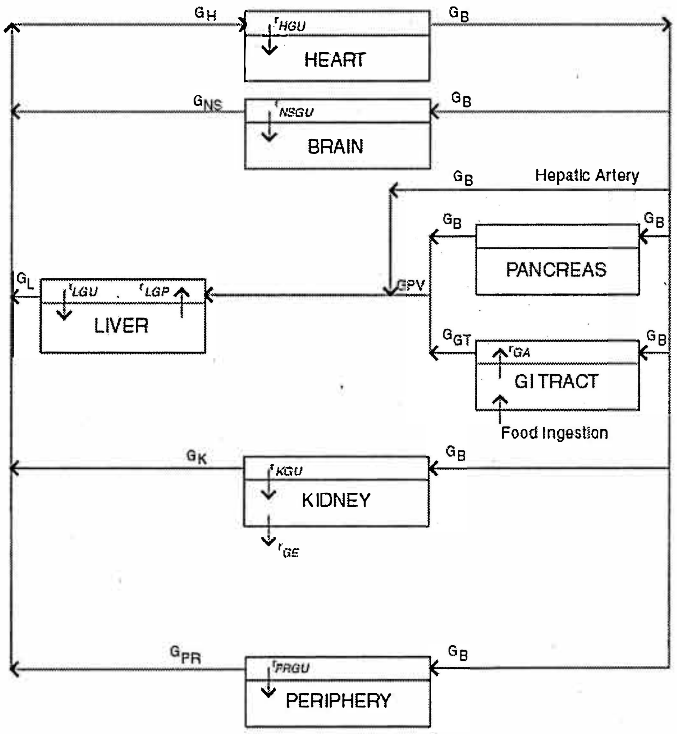


Figure 2.3. Pharmacokinetic Diagram of the Glucose Model

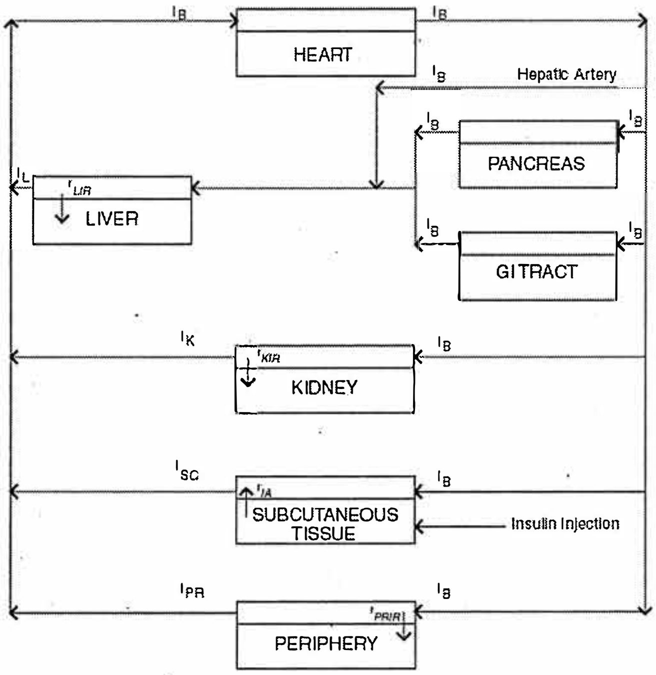


Figure 2.4. Pharmacokinetic Diagram of the Insulin Model

2.2 Artificial Intelligence

Montani *et al.* (2003) presented a multi-modal reasonıng (MMR) methodology, which integrated case-based reasoning (CBR), rule-based reasoning (RBR) and model-based reasoning (MBR) to provide a reliable decision support tool for physicians in the context of type 1 diabetes management. The system architecture of the methodology presented is shown in Figure 2.5. As shown in Figure 2.5, there are four consecutive tasks that need to be executed in the implementation of the therapy revision process for type 1 diabetes care. The completion of the process is scheduled by a RBR system within which each task is mapped into a specific set of rules, fired through a forward chaining mechanism. The four tasks, which are consecutively executed in the reasoning paradigm, are detailed as follows:

a. Data analysis:

The probabilistic description of the modal day of the patient is resorted so as to interpret the effects of a therapy. In particular, the patient's blood glucose level (BGL) modal day is extracted after a discretisation and aggregation of BGL values performed on the basis of qualitative abstractions.

b. Problem identification:

The identification of hypoglycemia problems in the different periods of the day is triggered as a result of the modal day extraction. This task can be independently completed by the RBR system or its behaviour can be specialised resorting to the integration with case-based retrieval. In particular, in the problem identification task, only classification results are exploited, thus tailoring the identification of metabolic alterations to the single patient's needs.

c. Suggestion generation and selection:

In this task, a set of suggestions on how to modify the current insulin therapy are proposed on the basis of insulin competent concept. The most competent insulin which has the stronger effect on the moment of the day in which the problem is in question, is then identified. Competence is evaluated based on the underlying pharmacokinetics of the different insulin types.

d. Therapy revision:

An adjustment to the current insulin therapy is proposed by the RBR system in accordance with the selected suggestions. In the formal evaluation study, it was found that RBR was not sharp enough to promptly face the patient's alteration although its behaviour was judged correctly and satisfactorily. In order to solve this weakness, an integration of the RBR results with MBR or with CBR has to be done. Therefore, MBR and CBR are used in a mutually exclusive way to specialise the rules behaviour. Moreover, the classification procedure provides an added value: the identification of the most probable class(es) for the input case allows to detect a suitable context for interpreting the case itself, the metabolic alterations can be evaluated in the light of the patient's features and the therapeutic suggestion can be adapted to them.

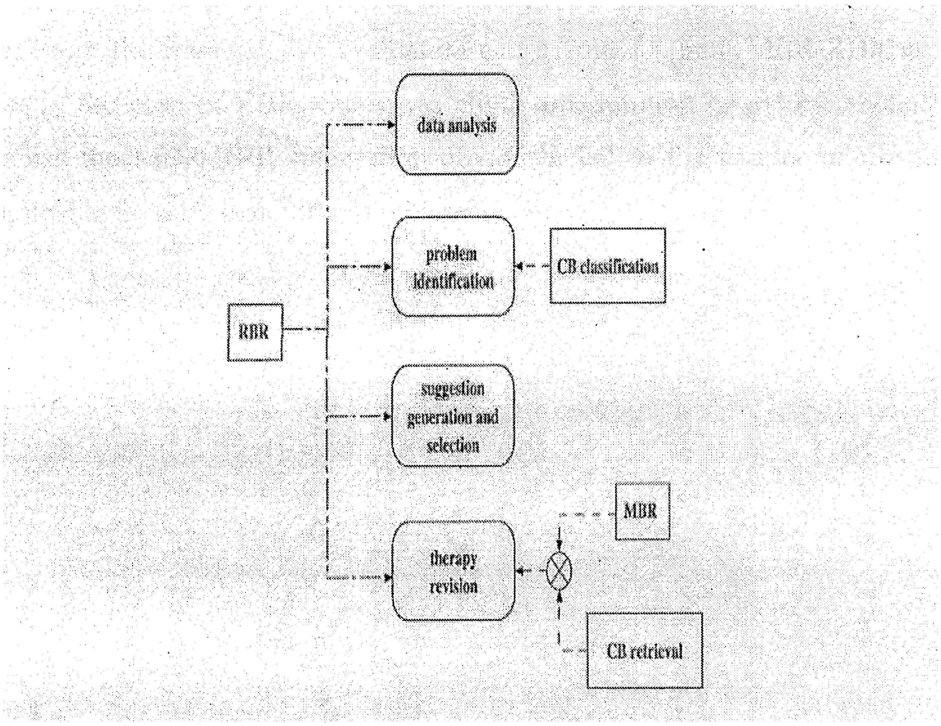


Figure 2.5 Implementation of the integration between RBR, CBR and MBR in the automatic reasoning process for therapy revision (MMR)

The results obtained from the implementation of the MMR methodology are shown in Figures 2.6 and 2.7. As shown in Figure 2.6, a comparison is made between the starting profile (continuous line), the model-based decision support system solution (dash-dotted line) and the optimal solution (dashed line). The circles represent the mean of the measurements available to the model-based system. Figure 2.6(a) represents a simulated patient with normal insulin sensitivity and following meal plan 5. The two approaches show nearly similar profiles, i.e. the patient profile at bed time is only slightly improved in both cases. Figure 2.6(b) represents a simulated patient with high insulin sensitivity and following meal plan 3. The two approaches show again nearly similar profiles, i.e. in both cases the original hyperglycemic profile is properly handled in order to generate a normoglycemic one. Figure 2.6(c) represents a simulated patient with low insulin sensitivity and following meal plan 7. The model-based decision support system shows a sub-optimal capability of normalising the blood glucose profile, although the solution does not show strong differences from the optimal one. Figures 2.7(a) and 2.7(b) compare the results obtained between MBR-RBR and RBR integration, respectively. It is found that both methodologies suggest therapy changes ameliorate the initial metabolic behaviour but the use of the model largely increases efficacy, thus allowing a total recovery from hypoglycemia. From the retrospective evaluation carried out for both MBR-RBR and RBR, it was found that in the case of RBR, there was slight amelioration of the metabolic behaviour obtained, whereas the MBR-RBR integration obviously led to the greater improvement. The model was applicable in more than 80% of the cases.

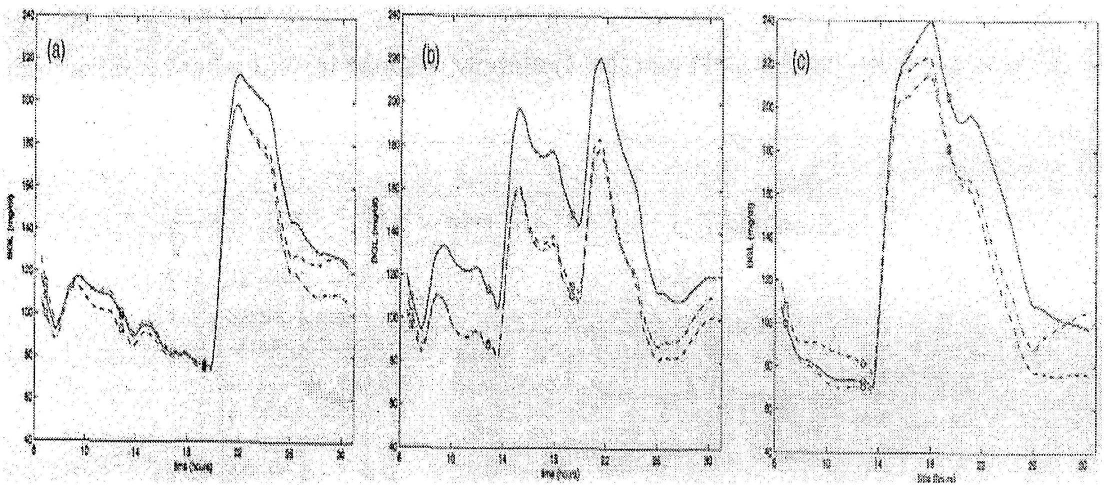


Figure 2.6 Daily simulated BGL profile (24 hr) in response to different therapeutic regimens. A simulated patient with: a) normal insulin sensitivity, b) high insulin sensitivity, and c) low insulin sensitivity.

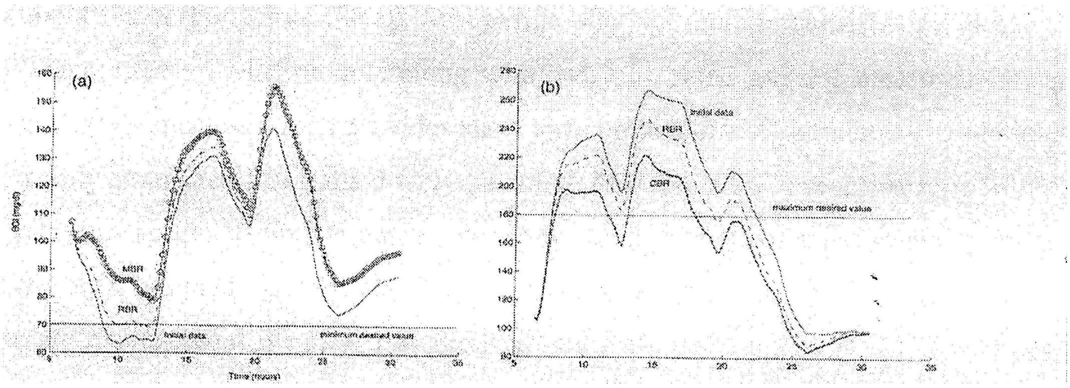


Figure 2.7 Comparison between MBR-RBR and RBR integration in stabilising a simulated patient entering the clinical remission phase.

Mougiakakou *et al.*, (2005) developed a real time simulation model of glucose-insulin metabolism for type 1 diabetic patients using Artificial Neural Networks (ANNs). The model was based on the combination of compartmental models and the ANNs. The outline of the proposed system is shown in Figure 2.8. As shown in Figure 2.8, the system comprises two modules namely; Mathematical Models Module (MM) and Neural Network Module (NN). In the

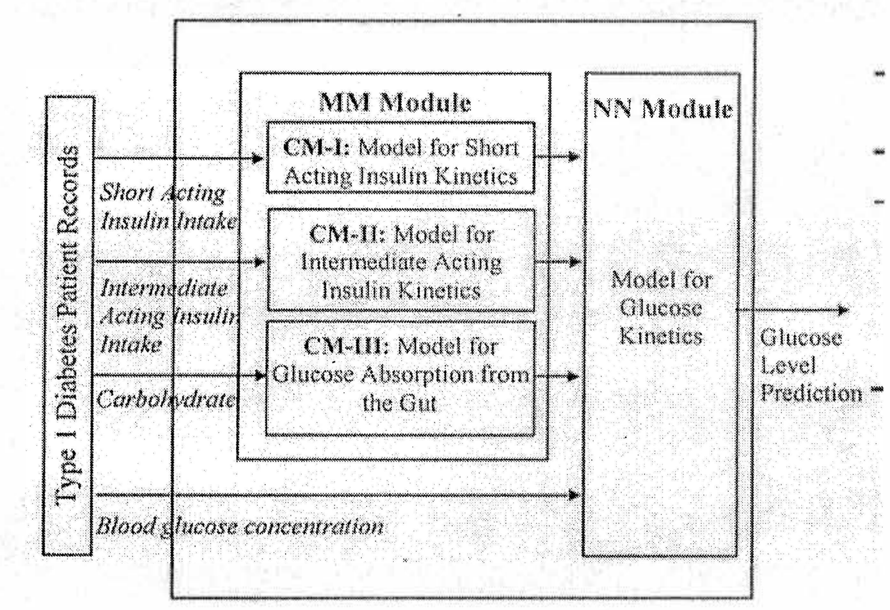

Type 1 Diabetes Patient Records
Short Acting Insulin Intake
Intermediate Acting Insulin Intake
Carbohydrate
Blood glucose concentration
MM Module
CM-I: Model for Short Acting Insulin Kinetics
CM-II: Model for Intermediate Acting Insulin Kinetics
CM-III: Model for Glucose Absorption from the Gut
NN Module
Model for Glucose Kinetics
Glucose Level Prediction

Figure 2.8 Outline of the proposed system

MM module, there are three compartmental models (CM) involved namely: CM-1, CM-II and CM-III. The estimation of plasma insulin concentration based on insulin intake is carried out in CM-I and CM-II, whereas CM-III provides predictions of glucose inputs into the blood based on information about the carbohydrate intake. Meanwhile, the NN module consists of a Recurrent NN (RNN). The outputs of MM module along with previous blood glucose measurements are then passed to the NN module which provides short term predictions of the blood glucose level. The RNN is a fully connected NN trained with the online Real Time Recurrent Learning (RTRL) algorithm which has ability to update online the RNN weights. For comparison purposes, two strategies have been adopted namely, the Free-Run (FR) and the Teacher-Forcing (TF). The available glucose management is ignored by the RNN for the case of online RTRL-FR strategy, whereas in the case of online RTRL-TF strategy the RNN replaces the actual output during training with the corresponding available glucose measurement. The comparisons between those two strategies on the measured and the estimated blood glucose levels verses time are shown in Figure 2.9. It can be concluded that the predictions of the online RTRL-FR trained RNN are relatively better as compared to the online RTRL-TF trained RNN. The obtained results are promising for the accuracy and efficiency of the proposed approach for simulation of glucose-insulin metabolism and prediction of blood glucose levels.

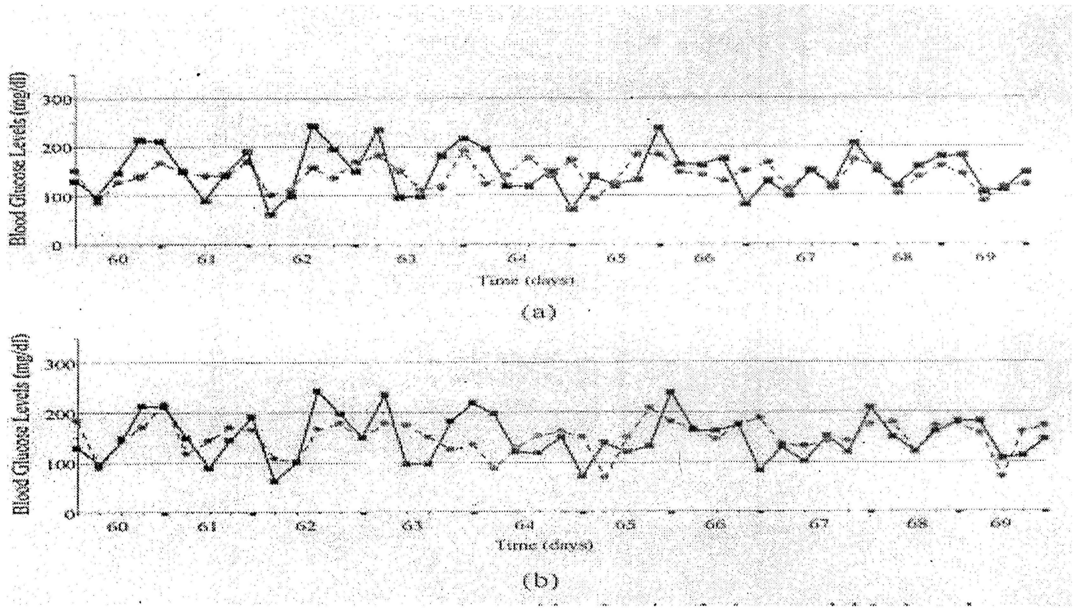


Figure 2.9 Comparison between estimated by the simulation model (♦), and measured by the patient blood glucose level (●), for the testing set (a) using the online RTRL-FR algorithm, and (b) using the online RTRL-TF algorithm

Mougiakakou *et al.*, (2006) presented the combined use of MM and NN Modules for the simulation of glucose-insulin metabolism on children with type 1 diabetes. The schematic diagram of the proposed system is shown in Figure 2.10. In order to predict the glucose level at a time instant, t the most recently measured blood glucose concentration along with parameters reflecting the effects of insulin and food intakes are applied to the NN module. Two different NN architectures have been developed and tested for comparative reasons: a feed-forward NN (FFNN) and a recurrent NN (RNN). As for the RNN, two strategies are applied namely, free-run (FR) and teacher-forcing (TF). In order to access the performance of the developed models, the root mean square error (RMSE) and the correlation coefficient (CC) have been calculated for both data sets and the results are shown in Table 2.1. Based on Table 2.1, it can be deduced that the results obtained from FFNN and the RNN trained with online RTRL-TF are superior to those obtained by online RTRL-FR trained RNN for all diabetic patients. The obtained results have concluded that the model using RTRL-TF trained RNN can simulate more accurate the metabolism of children with a type 1 diabetes.

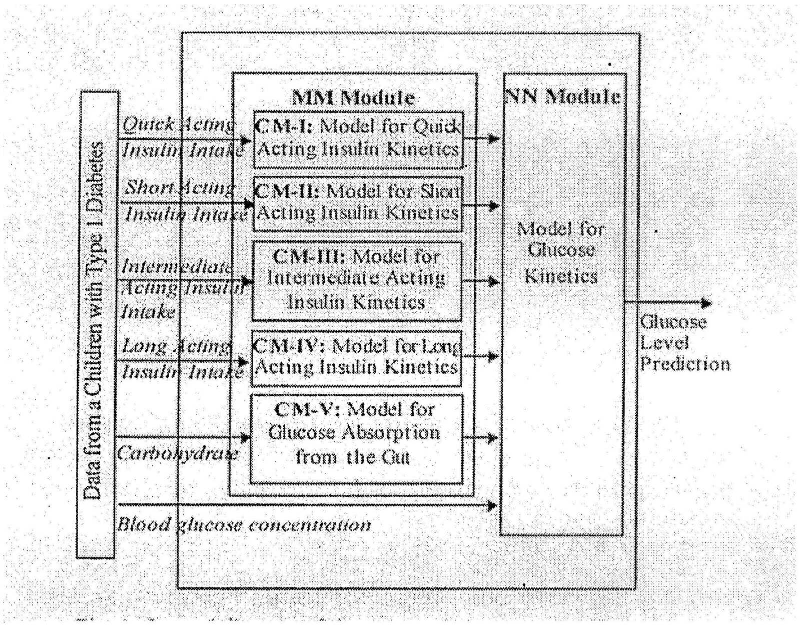
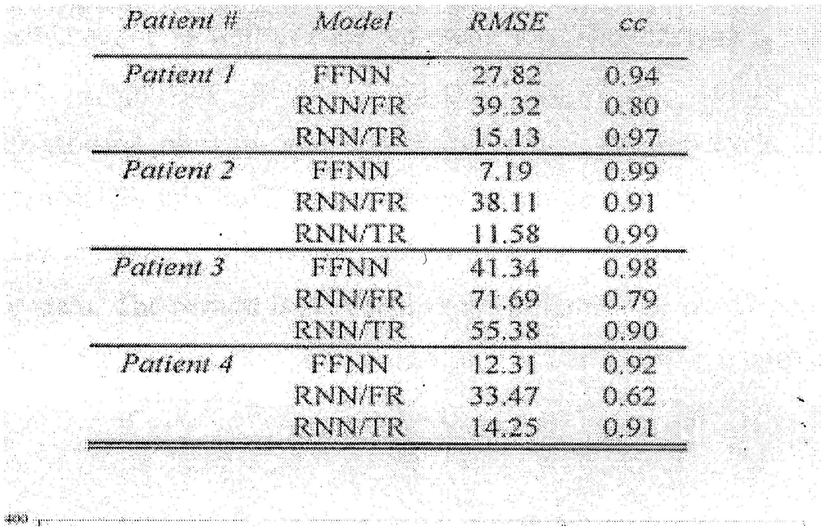

Data from a Children with Type 1 Diabetes
Quick Acting Insulin Intake
Short Acting Insulin Intake
Intermediate Acting Insulin Intake
Long Acting Insulin Intake
Carbohydrate
Blood glucose concentration
MM Module
CM-I: Model for Quick Acting Insulin Kinetics
CM-II: Model for Short Acting Insulin Kinetics
CM-III: Model for Intermediate Acting Insulin Kinetics
CM-IV: Model for Long Acting Insulin Kinetics
CM-V: Model for Glucose Absorption from the Gut
NN Module
Model for Glucose Kinetics
Glucose Level Prediction

Figure 2.10 Schematic diagram of the proposed glucose-insulin metabolism models

Table 2.1 RMSE along with cc between measured and estimated blood glucose levels for testing sets



Patient #	*Model*	*RMSE*	*cc*
Patient 1	FFNN	27.82	0.94
	RNN/FR	39.32	0.80
	RNN/TR	15.13	0.97
Patient 2	FFNN	7.19	0.99
	RNN/FR	38.11	0.91
	RNN/TR	11.58	0.99
Patient 3	FFNN	41.34	0.98
	RNN/FR	71.69	0.79
	RNN/TR	55.38	0.90
Patient 4	FFNN	12.31	0.92
	RNN/FR	33.47	0.62
	RNN/TR	14.25	0.91

Yasini *et al.*, (2009) employed reinforcement learning theory and Q-learning algorithm to regulate the blood glucose level for type 1 diabetic patient. The block diagram of a closed-loop control system for diabetic patients is shown in Figure 2.11. Based on the figure, the optimal insulin delivery rate is calculated by the control algorithm which is designed to keep the patient under metabolic control and allows the desired amount of insulin to be pumped mechanically. Reinforcement learning theory has been developed as a result of man's effort to analyse the behaviour of animal and artificial systems. It focuses on the effect of rewards and punishments on subjects choices in their attempt to achieve a goal. There are two basic elements involved namely: the learner or decision maker called the agent, and everything it interacts with called the environment. This can be described as shown in Figure 2.12. The agent receives the state, x_t of its environment at each time step, t and selects an action, a_t based on this perception and its past experience. Consequently, the agent receives a numerical reward, r_t at one time step later and

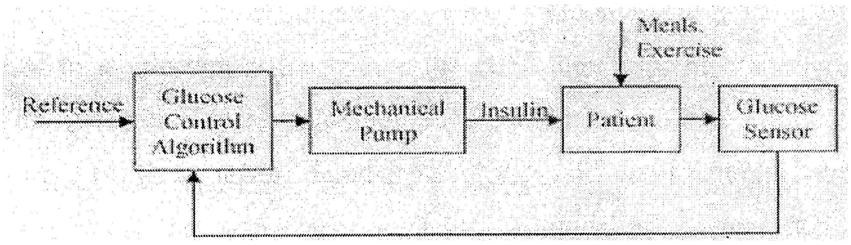


Figure 2.11 Closed-loop control of diabetic patient

finds itself a new state based on its action. A mapping is implemented at each time step the agent reacts to each possible action as well as state representation. The mapping is called the agent's policy. The purpose of applying Q-learning algorithm for diabetes control is to determine appropriate insulin infusion rates in order to stabilise blood glucose levels of the patient at reasonable time frame. In this case, the glucose-insulin regulatory system represents the environment for the simulation. Meanwhile, insulin infusion rates demonstrate actions in the modelling of the system. The reward is set equal to the difference of the glucose concentration from its target value of 80 mg/dL, i.e. a reference set point in normoglycemic range of blood glucose.

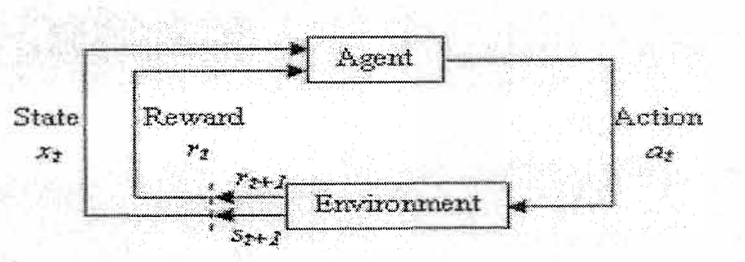


Figure 2.12 The learning process of the agent through its interaction with the environment

The validity of the proposed approach is checked through a MATLAB simulation of the closed-loop system. The results are shown in Figures 2.13 and 2.14. As shown in Figure 2.13, after identifying the environment by the agent, model variables become firm in basal amounts and the Q-value converges to a constant value. Since the controller implementation requires simple function evaluation, it is much easier to compute than solving the online optimisation problem. As shown in Figure 2.14, three sets of parameters for three different patients have been used to check for the robustness of the controller to variations in model parameters. It shows that in all three cases, the plasma glucose and insulin level stabilise in a reasonable time interval. The controller performs quite well and keeps the blood glucose level of the patients around normal value. This work has concluded that the reinforcement learning approach is able to derive the explicit insulin delivery rate for type 1 diabetes and the controller designed using Q-learning scheme has the potential to synthesise knowledge to treat the disease.

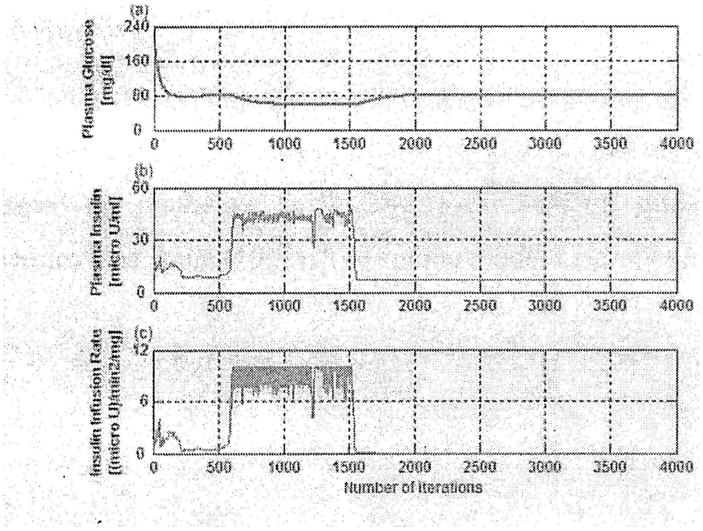


Figure 2.13 Glucose-insulin regulatory system with Q-learning algorithm in off-line state. (a)Plasma glucose concentration (b) Plasma insulin concentration (c) Insulin infusion rate

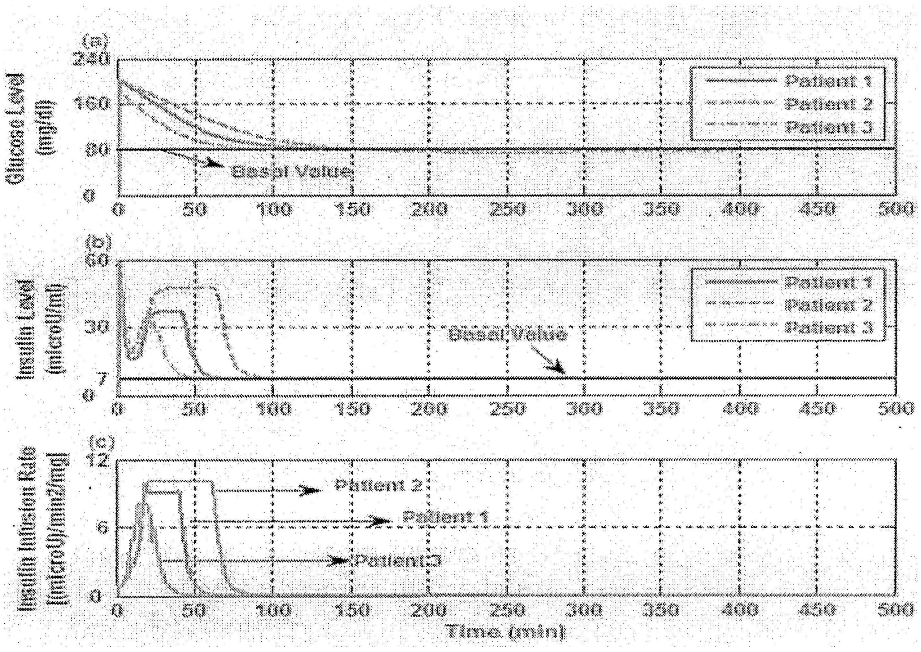


Figure 2.14 Closed-loop glucose regulatory system. (a) Plasma glucose concentration. (b) Plasma insulin concentration. (c) Exogenous insulin rate.

2.3 Mathematical Modelling

Boutayeb and Chetouani (2006) presented a global overvıew of mathematical models dealing with many aspects of diabetes especially on glucose-insulin dynamics. Among the pioneer who developed the mathematical models to estimate the glucose disappearance and glucose-insulin dynamics was Bolie (1961). The simple model he proposed is as follows:

$$\frac{dG}{dt} = -a_1 G - a_2 I + P \qquad (2.1)$$

$$\frac{dI}{dt} = -a_3\, G - a_4\, I \qquad (2.2)$$

where G, $G(t)$ = glucose concentration
I = insulin
P, a_1, a_2, a_3. a_4 = parameters

The real start of modelling of the glucose-insulin dynamics is thought to begin with the so-called Minimal Model proposed by Bergman and Cobelli in the early eighties and the model is as follows:

$$\frac{dG(t)}{dt} = -[P_1 + X(t)]G(t) + P_1 G_b, \quad G(0) = P_o \qquad (2.3)$$

$$\frac{dX(t)}{dt} = -P_2 X(t) + P_3(I(t) - I_b), \qquad X(0) = 0 \qquad (2.4)$$

$$\frac{dI(t)}{dt} = P_4(G(t) - P_5)^+\, t - P_6(I(t) - I_b), \qquad I(0) = P_7 + I_b \quad (2.5)$$

where, $(G(t) - P_5)^+ = G(t)-P_5$ if $G(t) > P_5$ and 0 otherwise
$X(t)$ = insulin-excitable tissue glucose uptake
G_b = subject's baseline glycaemia
I_b = subject's baseline insulinimia
P_0-P_7 = parameters

Professor Bergman was awarded the 2006 Banting medal by the American Diabetes Association so as to indicate the importance of his minimal model as well as his subsequent researches for diabetes understanding. The modified version of the minimal model was proposed by Derowich and Boutayeb (2002) who introduced parameters related to physical exercises. The model is as follows:

$$\frac{dG(t)}{dt} = -(1 + q_2)\, X(t)G(t) + (P_1 + q_1)(G_b - G(t)) \qquad (2.6)$$

$$\frac{dX(t)}{dt} = -P_2X(t) + (P_3 + q_3)(I(t) - I_b) \qquad (2.7)$$

where: q_1, q_2, q_3 = parameters related to physical activity

De Gaetano and Arino (2000) proposed an aggregated delay differential model called a dynamic model which was formulated as follows:

$$\frac{dG(t)}{dt} = -b_1G(t) - b_4I(t)G(t) + b_7, \quad G(0) = G_b + b_o \qquad (2.8)$$

$$\frac{dI(t)}{dt} = -b_2I(t) + \frac{b_6}{b_5}\int_{t-b_5}^{t} G(s)ds, \qquad I\,(0) = I_b + b_3b_o \qquad (2.9)$$

with $G(t) = G_b$ for $-b_5 \leq t < 0$

Mukhopadhyay *et al.*, (2004) proposed an extension to the dynamic model by introducing a generic weight function, ω in the delay integral kernel for the pancreatic response to glucose. The new model is as follows:

$$\frac{dG(t)}{dt} = -b_1G(t) - b_4I(t)G(t) + b_7, \quad G(0) = G_b + b_o \qquad (2.10)$$

$$\frac{dI(t)}{dt} = -b_2I(t) + b_6\int_{0}^{\alpha} \omega(s)G(t-s)ds, \qquad I(0) = I_b + b_3b_o \qquad (2.11)$$

with $G(t) = G_b$ for $t < 0$

Even though the dynamic model solved the problems of the minimal model, Li *et al.*, (2000) noted that some of the interaction terms were too specialised and thus too restricted. They proposed a more general model which was formulated as follows;

$$\frac{dX(t)}{dt} = -f(G(t)) - g(G(t)I(t)) + b_7, \qquad G(0) = G_b + b_0 \qquad (2.12)$$

$$\frac{dI(t)}{dt} = -P(I(t)) + q\big(L(G_t)\big), \qquad I(0) = I_b + b_3b_0 \qquad (2.13)$$

with $G(t) = G_b$ for $- b_5 \leq t < 0$ and $G_t\,(\theta) = G(t + \theta),\ t > 0, - b_5 \leq \theta < 0$

Palerm (2003) presented an interesting survey of mathematical models using control for glucose-insulin and management of diabetes focusing on the Direct Model Reference Adaptive Control (DMRAC). The general DMRAC algorithm is formulated based on the system as follows;

$$\frac{dx}{dt} = Ax(t) + Bu(t) \tag{2.14}$$

$$y(t) = Cx(t) + Du(t)$$

where $x(t)$ = n x 1 state vector

$u(t)$ = m x 1 control vector

$y(t)$ = q x 1 output vector

A, B, C and D = matrices

2.4 Model-based Predictive Control (MPC)

Parker *et al.* (1999) developed a model-based predictive control (MPC) algorithm and constructed a fundamental model to help maintain normoglycemia in type 1 diabetic patient using compartmental modelling techniques. The compartmental diagram of the proposed model is shown schematically in Figure 2.15. The model of human glucose-insulin system used in this study was taken from the initial work of Guyton *et al.*, (1978) which was later updated by Sorensen (1985). As shown in Figure 2.15, mass balances around tissues, which are deemed important to glucose-insulin dynamics, are performed so as to obtain individual compartmental models. In this model, the combined effects of muscle and adipose tissue are represented by the periphery; whereas the stomach and intestine effects are lumped into the gut compartment. An arterial glucose concentration acts as the controlled output which is regulated by the insulin infusion rate, the manipulated variable.

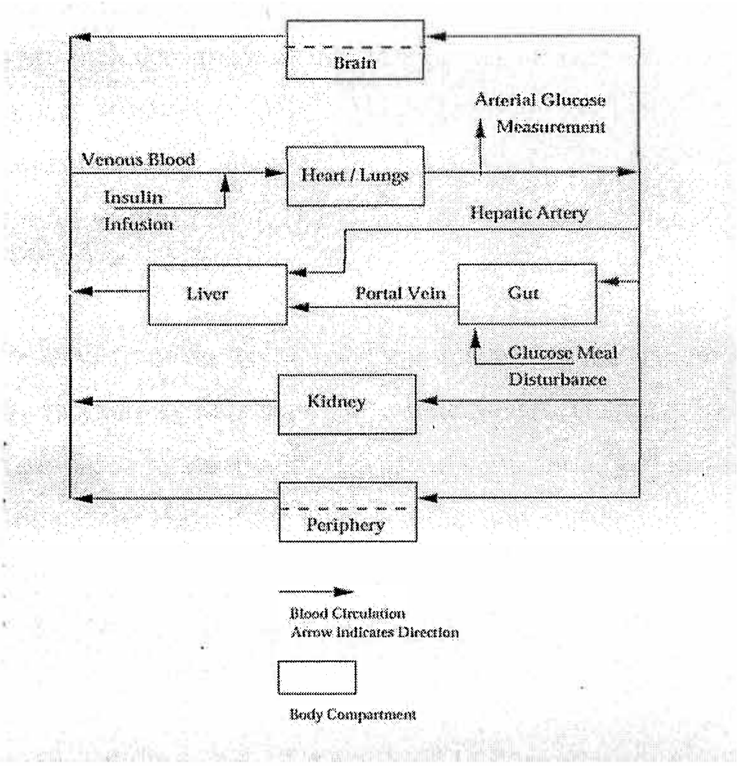


Figure 2.15 Compartmental diagram of the glucose or insulin system in a diabetic patient

Two approaches were used to control blood glucose concentrations namely, linear model predictive control (MPC) and MPC with state estimation. Garcia *et al*,. (1989) presented a MPC which was based on the so-called receding horizon philosophy. An optimal control problem is solved at each sampling time, starting from the current state over a finite horizon. The computation is repeated from the new state and over a shifted horizon leading to a moving horizon policy. The solution of linear MPC relies on a linear dynamic model, incorporates all the input and output constraints as well as the optimisation problems. The optimisation problem solved by MPC is given by:

$$\min_{\Delta U(k)}\left\{||\Gamma_y[y(k+1\,|k) - R(k+1|k)]||^2 + ||\Gamma_u \Delta U(k)||^2\right\} \qquad (2.15)$$

where: $\Delta U(k)$ = future input moves

$R(k+1|k)$ = vector of future reference values

$y(k+1|k)$ = vector of predicted future glucose concentrations

Γ_y = matrices for set point tracking penalty

Γ_u = matrices for insulin move penalty

In the MPC with state estimation (MPC/SE), more additional tuning parameters (a Kalman Filter and reference filter) are included to adjust closed-loop performance. A linear state-space, which forms the internal model structure in MPC/SE, changes the old input-output equations of the system. The model equations are as follows:

$$\begin{aligned} \bar{x}(k+1) &= \Phi\bar{x}(k) + \Gamma_u(k) \\ \bar{y}(k) &= C\bar{x}(k) \end{aligned} \qquad (2.16)$$

The model is constructed from the continuous non-linear diabetic patient model and it is first linearised analytically to yield a linear continuous-time model. The model is then converted to a discrete-time-minimum-phase representation for use in simulation, yielding Φ, Γ, and C. Finally, the controller can estimate the state of the plant and the output using the above equations as well as the equations below:

$$\begin{aligned} \hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma_u(k) + K_F[y(k) - C\hat{x}(k)] \\ \tilde{y}(k) &= C\hat{x}(k) \end{aligned} \qquad (2.17)$$

Simulation study results using the MATLAB/SIMULINK environment are promising for both linear MPC and MPC/SE algorithms. The results are shown in Table 2.2.

Table 2.2 Disturbance rejection result

Controller	**Undershoot (mg/dL)**	**Settling time (min)**
Linear MPC	9.7	343
MPC/SE	4.4	204

As shown in Table 2.2, the increased information available to the MPC/SE algorithm coupled with the Kalman filter yields greater than 40% performance improvement in both undershoot and settling time. This study concludes that MPC algorithms for insulin infusion pump control are capable of providing an excellent framework for glucose control problem. Linear MPC is sufficient in controlling blood glucose but results in glucose concentration near the output lower bound. On the other hand, a MPC/SE, together with Kalman filter and a more internal model, yields improved control performances. In addition, the digital nature of control algorithm allows potential applications onto chip technology in the near future.

Dua and Pistikopoulos (2005) and Dua *et al.*, (2006) developed a control algorithm which took into consideration the model of the patient, constraints on insulin infusion rate and blood glucose concentration. The mathematical model of the diabetic patient can be represented as follows:

$$
\begin{aligned}
x_{t+1} &= Ax_t + Bu_t \qquad\qquad (2.18)\\
s.t. \quad x_{min} &\leq x_t \leq x_{max}\\
u_{min} &\leq u_t \leq u_{max}\\
x_t &\in R^n\\
U_t &\in R^m
\end{aligned}
$$

where: x_t = glucose and insulin concentrations

u_t = insulin delivery rate

R_n = state vectors

R_m = input vectors

min = lower bound

max = upper bound

The optimisation problem using MPC can be represented as follows:

$$
\min_{u} \; J(U, x(t)) = x_{t+N_y|t}^T P x_{t+N_y|t} + \sum_{k=0}^{N_{y-1}} \left[x_{t+k|t}^T Q x_{t+k|t} + u_{t+k}^T R u_{t+k} \right] \quad (2.19)
$$

$$
\begin{aligned}
s.t. \quad x_{min} &\leq x_{t+k|t} \leq x_{max}, k = 1, \ldots, N_c\\
u_{min} &\leq u_{t+k} \leq u_{max,}\ \mathrm{k} = 1, \ldots, N_c\\
x_{t+k+1|t} &= Ax_{t+k|t} + Bu_{t+k,} k \geq 0\\
u_{t+k} &= Kx_{t+k|t}, N_u \leq k \leq N_y
\end{aligned}
$$

where: $u = [\mathrm{u}_{\mathrm{t}}^{\mathrm{T}}, \ldots\ldots\ldots, \mathrm{u}_{\mathrm{t+Nu-1}}^{\mathrm{T}}]^{\mathrm{T}}$

Q and R = constant matrices

P = Riccati equation

N_y = prediction horizon

N_u = control horizon

N_{c} = constraint horizon

K = feedback gain

However, a demanding online computational effort has become the major drawback of MPC for years. As such, an alternative approach to the solution of MPC is long overdue so as to reduce the amount of time, effort and resources needed in terms of its software and hardware implementations. The technique used is parametric programming which is an optimisation framework in which optimisation variable, u as a function of parameter, x_{t} can be obtained using a generic mathematical technique. In other words, when an objective function, a set of constraints and a vector of parameters are given, an optimal value of the optimisation variables is obtained as a set of functions of the parameters and the corresponding regions in the space of parameters where these functions are valid (Pistikopoulos *et al.*, 2002).

Equation (2.19) can be replaced with the elimination of the equalities in its formulation by the following equation:

$$x_{t+k|t} = A^k x_t + \sum_{j=0}^{k-1} A^j B u_{t+k-1-j} \qquad (2.20)$$

so as to obtain the quadratic program (QP) as follows:

$$\min_{u} \tfrac{1}{2} U^T H U + x_t^T F U + \tfrac{1}{2} x_t^T Y x_t \qquad (2.21)$$

$$\text{s.t.} \quad GU \leq W + Ex_t$$

where: : $u = [u_t^T, \ldots\ldots\ldots, u_{t+Nu-1}^T]^{\mathrm{T}} \in R^s$

R^s = vector of optimisation variables

$s = mN_u$

H, F, G, W, Y and E = obtained from Q and R in (2.19)

The QP problem in (2.21) can then be reformulated as a multi-parametric QP (mp-QP) (Dua *et al.*, 2002) as follows;

$$Vz(x) = min\tfrac{1}{2} z^T Hz \qquad (2.22)$$

$$\text{s.t.} \quad Gz \leq W + Sx_t$$

where: $z = U + H^1 F^T x_t$, $z \in R^s$

$s = E + GH^1 F^T$

By treating z as the vector of optimisation variables and x_t as the vector of parameters so as to obtain z as a set of explicit functions of x_t, this mp-QP problem can be solved. Using the equation, $U = z - H^1 F^T x_t$, u is then obtained as a set of explicit functions of x_t. Each of these functions is valid in a polyhedral region in the space of the state variables, x_t.

The Bergman model (Bergman *et al.*, 1981) is used to design the controller to control the blood glucose concentration of the diabetic patient. It is shown schematically in Figure 2.16. As shown in Figure 2.16, there are three compartments which represent the model and it can be written mathematically as follows:

Plasma glucose compartment:

$$\frac{dG}{dt} = -P_1 G - X(G + G_b) + D(t) \qquad (2.23)$$

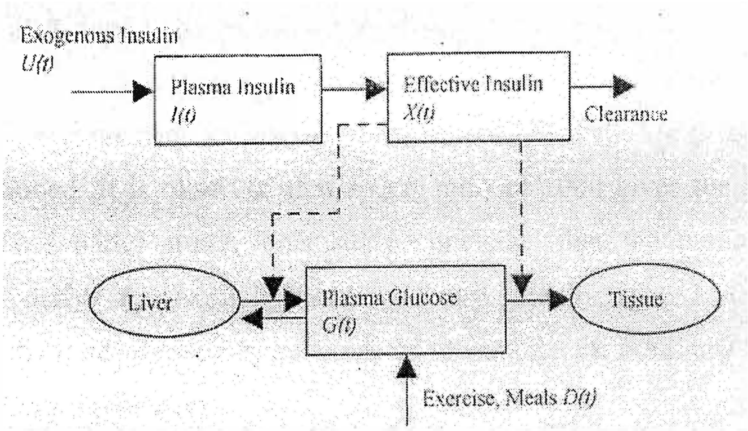


Figure 2.16 Schematic representation of Bergman minimal model

Plasma insulin Compartment:

$$\frac{dI}{dt} = -n(I + I_b) + \frac{u(t)}{V_1} \tag{2.24}$$

Effective insulin compartment:

$$\frac{dX}{dt} = -P_2X + P_3I \tag{2.25}$$

where: G = plasma glucose concentration above basal value, mg/dL

I = plasma insulin concentration above basal value, mU/L

X = proportional to plasma insulin concentration in remote compartment, min^{-1}

D = meal glucose disturbance, mg/dLmin^{-1}

$D(t) = F_G/V_G$

u = exogenous insulin infusion rate, mU/min

G_b = basal value of glucose concentration, mg/dL

I_b = basal value of insulin concentration, mU/L

V_I= insulin distribution volume, L

n = fractional disappearance rate of insulin, min^{-1}

F_G = rate of exogenously infused glucose, mg/min

V_G= glucose distribution space, dL

In this model, parameter values selected among others are: $P_1 = 0$ min^{-1}, $P_2 = 0.025$ min^{-1}, P_3 =0.000013 L/mU min^2 , V_1 = 12 L and n = 5/54 min^{-1}. The model is then linearised about the steady state value of G_b = 81mg/dL, I_b = 15 mU/L and u_b = 16.6667 mU/min so as to obtain the form given in (2.18);

$$X_{t+1} = Ax_t + Bu_t + B_d d_t$$

where A, B and B_d are discrete state space matrices which can be calculated using gPROMS®.

Figures 2.17 and 2.18 depict the performance of the control law under the presence of Fisher meal disturbances of 20, 50 and 40g of carbohydrate intake, on the non-linear Bergman model (2.23)-(2.25). The performances are shown for different Q/R ratios and the number of critical regions (CRs) obtained. It is observed that a Q/R ratio of 1000 gives the best of all the other control performances under study. This study concludes that the parametric programming approach used to derive the explicit insulin delivery rate for type 1 diabetes is simple to implement. Thus, these developments are expected to enhance the automation of insulin delivery and reduce patient inconvenience.

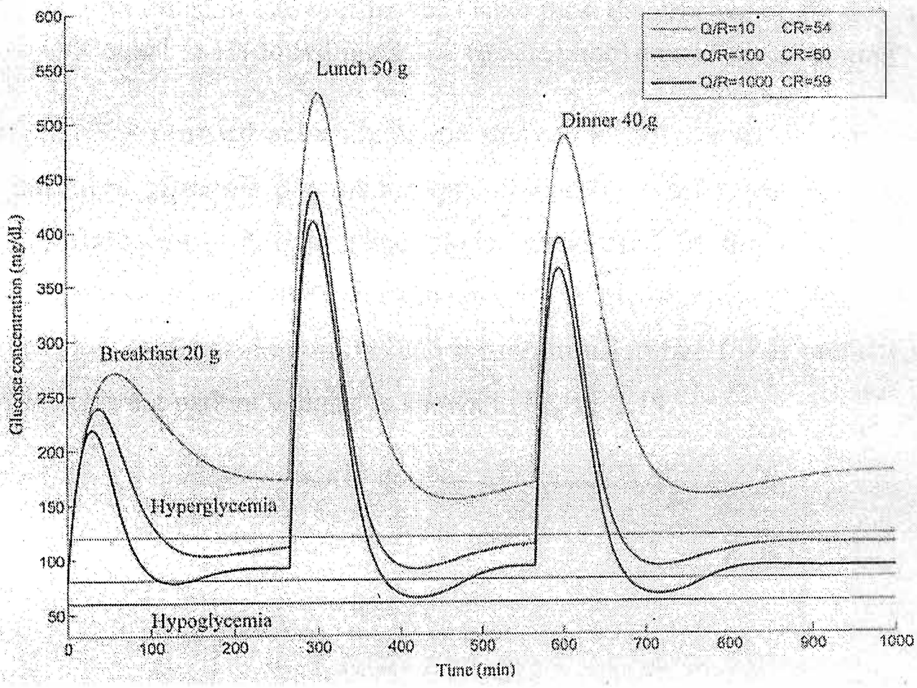


Figure 2.17 Glucose concentration profiles for Fisher meal disturbance of 20, 50 and 40g and for Q/R equal to 10(dotted line), 100 (dashed line) and 1000 (solid line)

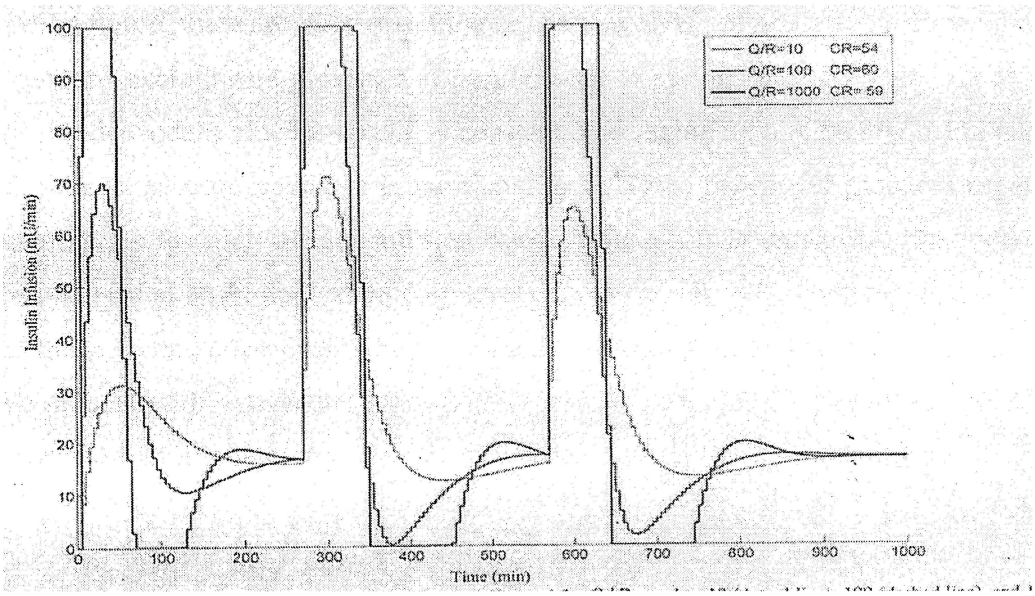


Figure 2.18 Insulin infusion rate profiles for Fisher meal disturbance of 20, 50 and 40g and for Q/R equal to 10(dotted line), 100 (dashed line) and 1000 (solid line)

Finan *et al.*, (2006) carried out a simulation study to identify linear dynamic models to help achieve an improved glycemic control for type 1 diabetes when used in a model predictive control (MPC) framework. Physiological model considered in the research is the model developed by Hovorka *et al.*, (2004) and extended by Wilinska *et al.*, (2005). The steady-state map of plasma glucose concentration (G) and insulin infusion rate (u) as predicted by Hovorka model for three different patient weights is shown in Figure 2.19.

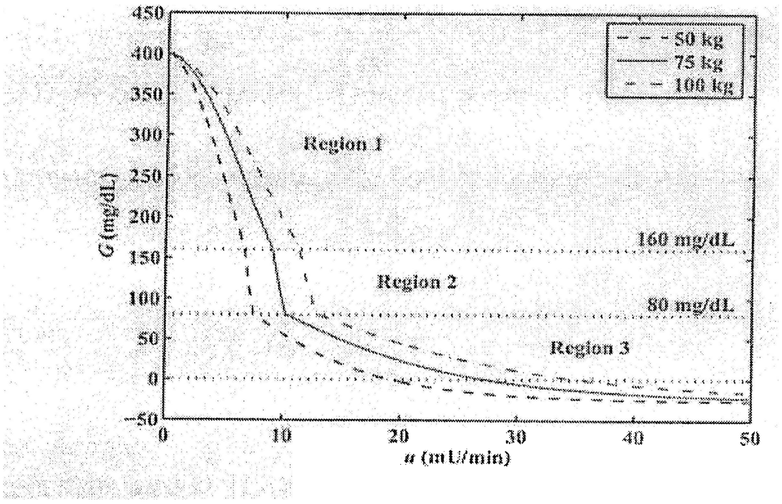


Figure 2.19 Steady-state G-u map for three patient weights. Regions 1-3 represent different operating regions for the model.

As depicted in the figure, there are three operating regions involved: region 1($G \geq 160$ mg/dL) in which renal glucose excretion is present and proportional to G, region 2 ($G \geq 80$ mg/dL) in which non-insulin dependent glucose uptake is constant, and region 3 ($G < 80$ mg/dL) in which non-insulin-dependent glucose uptake is proportional to G. Thus, the model produces negative glucose concentrations for high insulin infusion rates. Although it is unrealistic, the model is intended to be operated at physiological glucose levels, i.e. 40 mg/dL < G < 400 mg/dL.

Two types of linear dynamic models, which are used empirically in this research, are namely autoregressive models with exogenous input (ARX) and Box-Jenkins (BJ) models. The ARX model is represented by this equation:

$$A(q^{-1})G(t) = B_1(q^{-1})\,u(t) + B_2(q^{-1})\,U_G(t) + \varepsilon(t) \qquad (2.26)$$

where: $q^{-1}G(t) = G(t\text{-}1)$, q^{-1} is backward shift operator

A = scalar polynomial in ascending power q^{-1}, starts at q^{o}=1

B_1 and B_2 = scalar polynomials in ascending power q^{-1}, start at q^{-1}

ε = Gaussian process noise

The ARX is a difference equation in which the current output depends on previous outputs and inputs.

Meanwhile, the BJ model is a transfer function model which models both deterministic inputs (i.e. u and U_G) and stochastic inputs (i.e. the noise, ε) which is written as follows:

$$G(t) = \frac{B_1(q^{-1})}{F_1(q^{-1})}u(t) + \frac{B_2(q^{-1})}{F_2(q^{-1})}U_G(t) + \frac{C(q^{-1})}{D(q^{-1})}\varepsilon(t) \quad (2.27)$$

where: B_1, B_2, C, D, F_1 and F_2 = scalar polynomial powers q^{-1}, start at q^{-1}

In the simulation results, model accuracy for both calibration and validation data was quantified by standard coefficient of determination as follows:

$$R^2 = \left(1 - \frac{\sum_{i=1}^{N}(G_i - \hat{G}_i)^2}{\sum_{i=1}^{N}(G_i - \bar{G})^2}\right) x\ 100\% \qquad (2.28)$$

where: N = no. of samples

G = output simulated by Hovorka model

$\hat{G}$ = output predicted by identified model

$\bar{G}$ = average output simulated by Hovorka model

Low order and high order ARX and BJ models were identified from each of the three datasets representative of normal operation (i.e. N_A, N_L, N_H). Table 3 shows the R^2 values of the high-order ARX and BJ models for the three datasets.

Table 2.3 R^2 values of predictions of high-order models identified from N_A for all normal datasets.

Day	Model	N_A	N_L	N_H
N_A	ARX	**66**	48	57
	BJ	**77**	57	63
N_L	ARX	32	**74**	0
	BJ	52	**78**.	42
N_H	ARX	62	27	**74**
	BJ	71	68	**78**

As shown in Table 2.3, both types of model predict their calibration data accuracy (i.e. $R^2_{cal} \geq 66\%$, $\bar{R}^2_{cal} = 74.5\%$). In general, BJ models consistently explain more variability in the data than ARX models. Table 4 shows the R^2 values for the low-order ARX and BJ models for the three normal datasets.

Table 2.4 R^2 values of predictions of low-order models identified from N_A for all normal datasets

Day	Model	N_A	N_L	N_H
N_A	ARX	**66**	43	58
	BJ	**71**	70	79
N_L	ARX	38	**70**	7
	BJ	62	**78**	58
N_H	ARX	61	27	**72**
	BJ	71	66	**80**

As shown in Table 2.4, the low-order models fits are comparable to those of the high-order models (i.e. $R^2_{high} = 56.9\%$, $\bar{R}^2_{low} = 59.7\%$)

The study concludes that accurate linear dynamic models have been identified from a simulated physiological diabetes model: The simulation carried out represents realistic conditions by incorporating measurement noise, reasonable meal times and magnitudes, insulin to carbohydrate ratio and faults.

Markakis *et al.*, (2008) developed a non-parametric / principal dynamic model (PDM) as a MPC strategy to regulate the blood glucose concentrations in type 1 diabetes. The new minimal

Markakis *et al*., (2008) developed a non-parametric / principal dynamic model (PDM) as a MPC strategy to regulate the blood glucose concentrations in type 1 diabetes. The new minimal model structure, termed as an Augmented Minimal Model (AMM) is introduced and it is represented in the form of equations as follows:

$$\frac{dI}{dt} = -\gamma_I \cdot I(t) + \beta \cdot max(G(t) - \theta_I, 0) + D_I(t) \qquad (2.29)$$

$$\frac{dN}{dt} = -\gamma_N \cdot N(t) + \alpha \cdot max(\theta_N - G(t), 0) \qquad (2.30)$$

$$\frac{dX}{dt} = -P_2 \cdot x(t) + P_3 \cdot I(t) \qquad (2.31)$$

$$\frac{dG_I}{dt} = -P_1 \cdot G_I(t) - x(t) \cdot G(t) \qquad (2.32)$$

$$\frac{dG_N}{dt} = -P_4 \cdot G_N(t) + P_5 \cdot N(t) \qquad (2.33)$$

$$G(t) = G_b + G(t) + G_N(t) + D_G(t) \qquad (2.34)$$

where: I = plasma insulin

N = plasma glucagon

X = insulin action

G_I = blood glucose deviated from basal value (G_b=90 mg/dL) due to insulin action

G_N = blood glucose deviated from basal value due to glucagon action

G = blood glucose concentration

D_I = intravenous insulin input

D_G = glucose disturbance (due to meals)

The following set of parameters represents the dynamics of type 1 diabetes based on simulations using Sorensen's model (Sorensen, 1985): P_1 = 0.013, P_2 = 0.063, P_3 = 9x10^{-6}, P_4=0.04, P_5=0.016, β=0, γ_N =3x10^{-3}, α=8x10^{-4} and θ_N=83. By simulating the AMM as mentioned above, discrete time, broadband input-output data is produced and the dynamics capture in the data is decomposed in a linear filter. Thus, the Principal Dynamic Mode (from which quantities like peak value, peak time, time constant and system memory are directly observable) and a non-linearity in series can be obtained. The result is presented in the form of Figure 2.20 in which the left panel presents the linear filter whereas the right panel shows the corresponding non-linearity and its linear approximation.

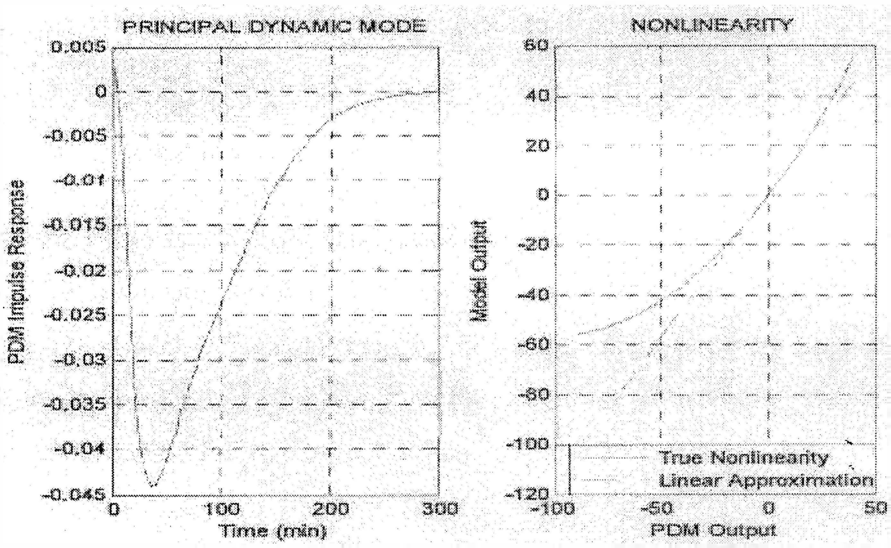


Figure 2.20 The non-linear PDM model of the insulin-glucose dynamics and its linearised counterpart

In order to predict accurately the future values of glucose disturbance, D_G, we need to hypothesise that D_G can be considered as the output of an Auto Regressive (AR) model as follows:

$$D_G\,(n) = D - \alpha + w(n) \qquad (2.35)$$

$$D = [D_G(n\text{-}1)\ D_G\,(n\text{-}2)\ldots\ldots.D_G(n\text{-}K)]$$

$$\alpha = [\alpha_1\ \alpha_2\ldots\ldots\alpha_K]^T$$

where: α = vector of coefficient

w = unknown ‘innovation process’

K = order of AR model

Estimation of coefficient vector is performed using a linear least squares method as follows:

$$\alpha = [A^TA]^{-1}A^Tb \qquad (2.36)$$

where: A = matrix constructed with appropriate values of D vector over discrete times (m>K)

b = corresponding vector of disturbance values

The goal of MPC is to determine the control input value, U(n) at every time instant, n when all the knowledge of the PDM model inclusive of the dynamics between insulin and glucose, past insulin inputs, estimated future values of glucose disturbance are in place. Hence, the cost function can be minimised as follows:

$$J(n) = [G(n+p|n) - R]^T \cdot \Gamma_y \cdot [G(n+p|n) - R] + \Gamma_u \cdot U(n)^2 \qquad (2.37)$$

where: $G(n+p|n)$ = vector of predicted output values over future horizon, p steps
R = target output value
Γ_y= diagonal matrix of weighting coefficient
Γ_u= scalar to determine how expensive insulin input

Simulations of the PDM model have been carried out so as to realise the objective of the MPC (i.e. to maintain the normoglycemic region in which the blood glucose concentrations are in the range of 70 to 110 mg/dL). To achieve this, an insulin micro-pump is used to simulate the imposition of an upper bound of 80 mU/min on the magnitude of the exogenous insulin rate. The block diagram of the closed-loop system is shown in Figure 2.21.

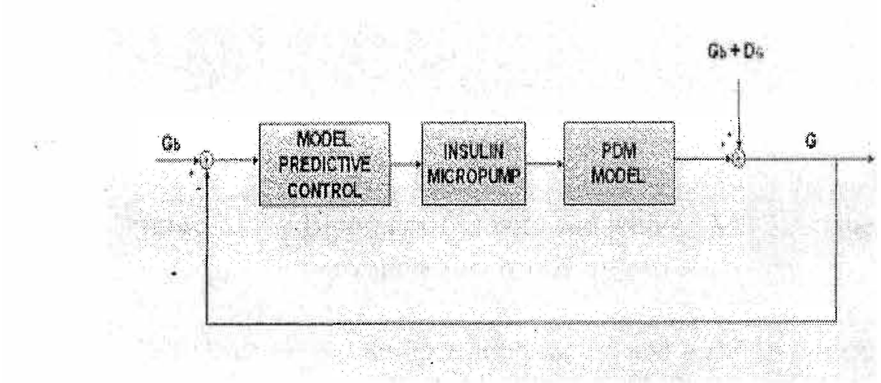


Figure 2.21 Closed-loop system for blood glucose regulation

Figure 2.22 shows the blood glucose concentrations with and without MPC, in its upper panel whereas; the lower panel presents the intravenous insulin infusion rate (superimposed to the basal rate) as determined by MPC. It can be concluded that the proposed algorithm can regulate well blood glucose and is able to deal with both positive and negative deviations of glucose from its basal value. The results and conclusions of this work depend critically on the assumption that Sorensen's model (from which AMM and PDM are derived) is accurate in regulating the blood glucose system as represented in the actual metabolic system.

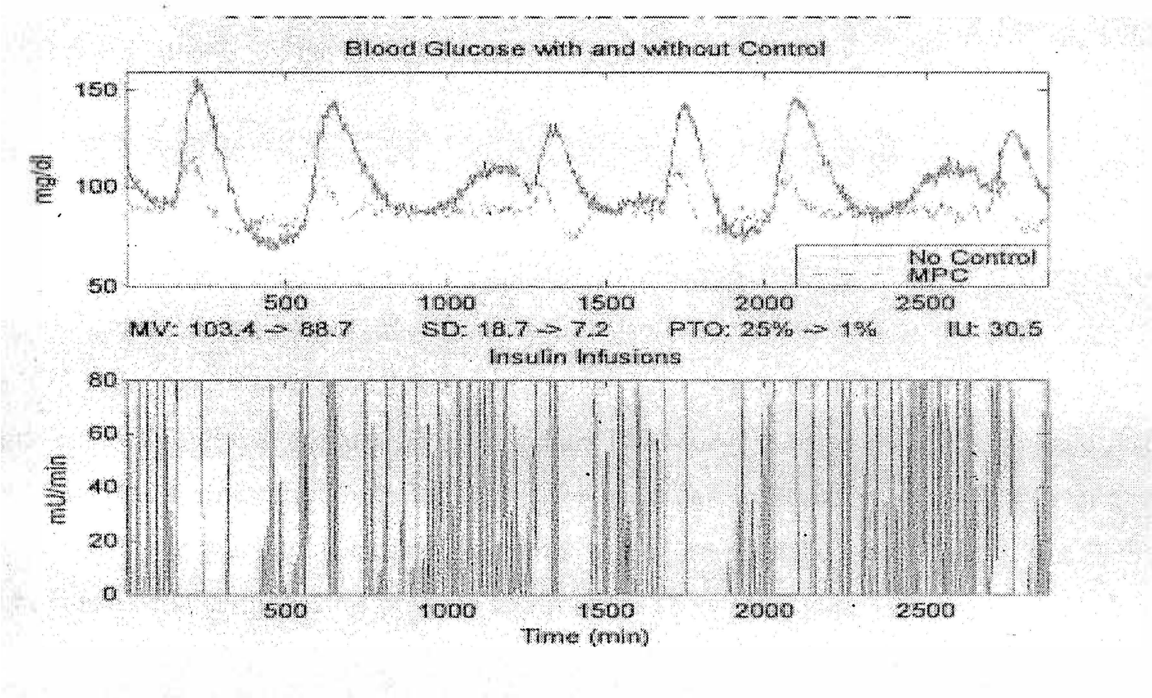


Figure 2.22 Blood glucose with and without MPC and the corresponding insulin infusions

Kovacs *et al.*, (2008) developed an open loop model and a robust controller using a Linear Parameter Varying (LPV) methodology and induced $\mathcal{L}_2$ –norm minimisation, respectively for insulin delivery in type 1 diabetic patients. In doing so, the high complexity non-linear diabetic patient Sorensen-model (Parker *et al.*, 2000) was considered. LPV system is a class of nonlinear system in which the parameter can be an arbitrary time varying; piece-wise continuous and vector valued function denoted by p(t), defined on a compact set $\mathcal{P}$. The dynamics of LPV system can be represented as follows:

$$\dot{x}(t) = A(\rho)x(t) + B(\rho)u(t) \qquad (2.38)$$
$$y(t) = C(\rho)x(t) + D(\rho)u(t)$$

where: $\rho(t) \in F_P$

$P \subset R^S$

$A: R^S \rightarrow R^{n\,x\,n}$

$B: R^S \rightarrow R^{n\,x\,n_u}$

$C: R^S \rightarrow R^{n_y x\,n}$

$D: R^S \rightarrow R^{n_y x\,n_u}$

Another way of describing LPV system is through a polytopic representation. In this case, the validity of the model is captured inside a polytopic region and the model is developed from a linear combination of the linearised models derived in each polytopic point$\left(\Sigma_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}\right)$;

(2.39)

$$\Sigma(t) \subset \{\Sigma_1 \ldots\ldots, \Sigma_2\} = \left\{ \sum_{i=1}^{j} \alpha_i \Sigma_{\mathrm{i}} : \alpha_{\mathrm{i}} \geq 0, \sum_{\mathrm{i}=1}^{\mathrm{j}} \alpha_{\mathrm{i}} = 1 \right\} \quad (2.40)$$

The 19th order Sorensen's model has two inputs: Γ_{meal} (meal disturbance) and Γ_{IVI} (injected insulin amount), and two outputs, G_H^C(the capillary heart-lungs glucose concentration) and I_P^C(the peripheric insulin concentration in the capillaries). The developed polytopic LPV system of the Sorensen's model was simulated and compared with those published in Parker *et al.* (2000). It can be concluded that the results prove to be similar as shown in Figure 2.23.

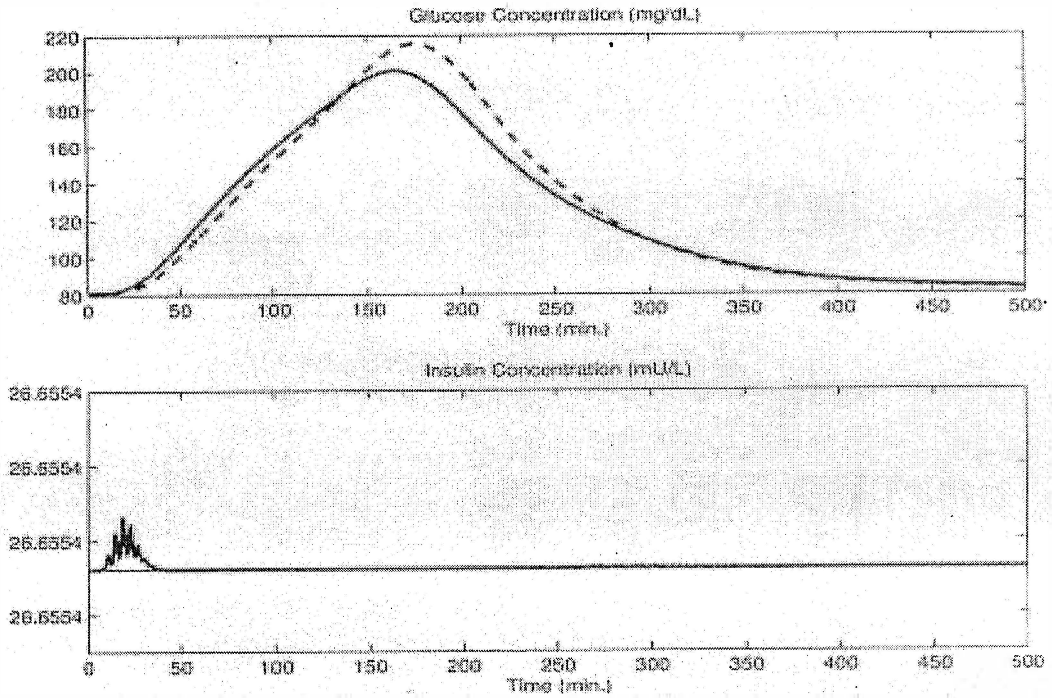


Figure 2.23 The simulation of the nonlinear Sorensen's model (solid) and the considered polytopic region (dashed)

In the case of robust control design using the Sorensen-model (Parker *et al.*, 2000), it appears that a near hypoglycemic situation occurs for the diabetic patient which is considered to be very dangerous. Hence, the aim of the LPV based robust control is to minimise the meal disturbance level over the performance output for all possible variations of the parameter within the polytope F_P, represented as follows:

$$\min_{K} \| G \| = \min_{K} \; \sup_{\rho \epsilon F_P} \; \sup_{\|d\| \neq 0} \frac{\|z y_1\|}{\|d\|} \qquad (2.41)$$

where: d = the meal disturbance input

z = glucose variation

In this study, 5% error on sensor noise and 2% error on glucose measurements are considered. For meal disturbances, a 60g of carbohydrate intake is considered. The control loop is extended with a weighting function for the control signal and an output uncertainty block, so as to avoid the hypoglycemic situation. It is shown in Figure 2.24.

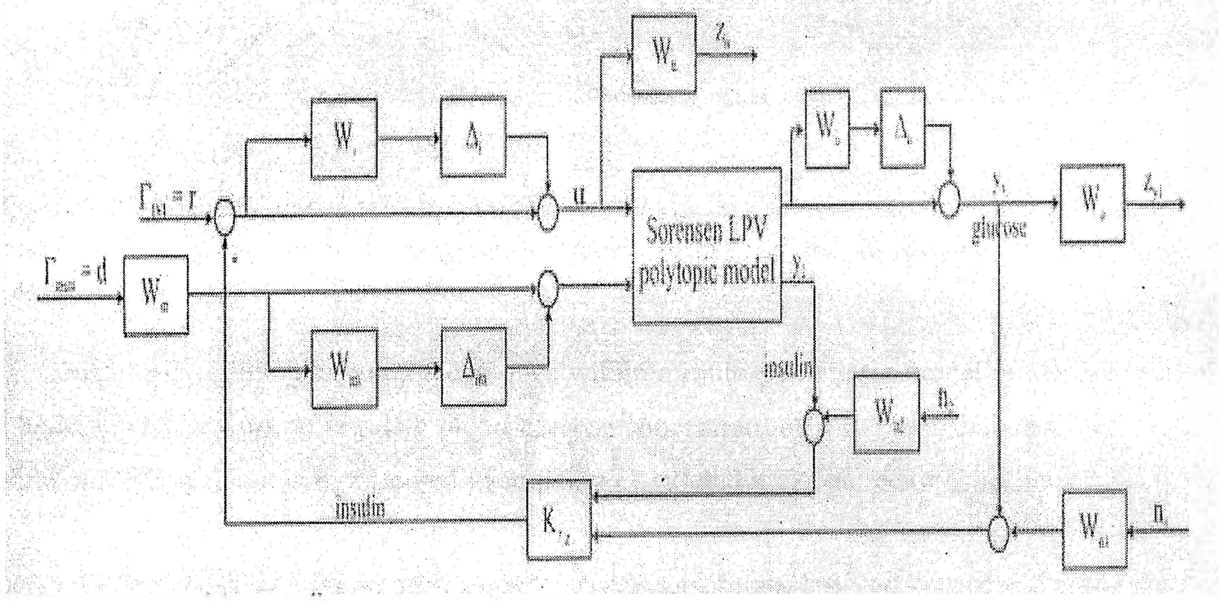


Figure 2.24 The augmented system and the controller

Figure 2.25 shows the performance of the LPV based robust controller with induced $\mathcal{L}_2$ –norm minimisation over the original nonlinear Sorensen-model. It can be concluded that the hypoglycemic situation is avoided and the glucose level is kept inside the normal range at all times.

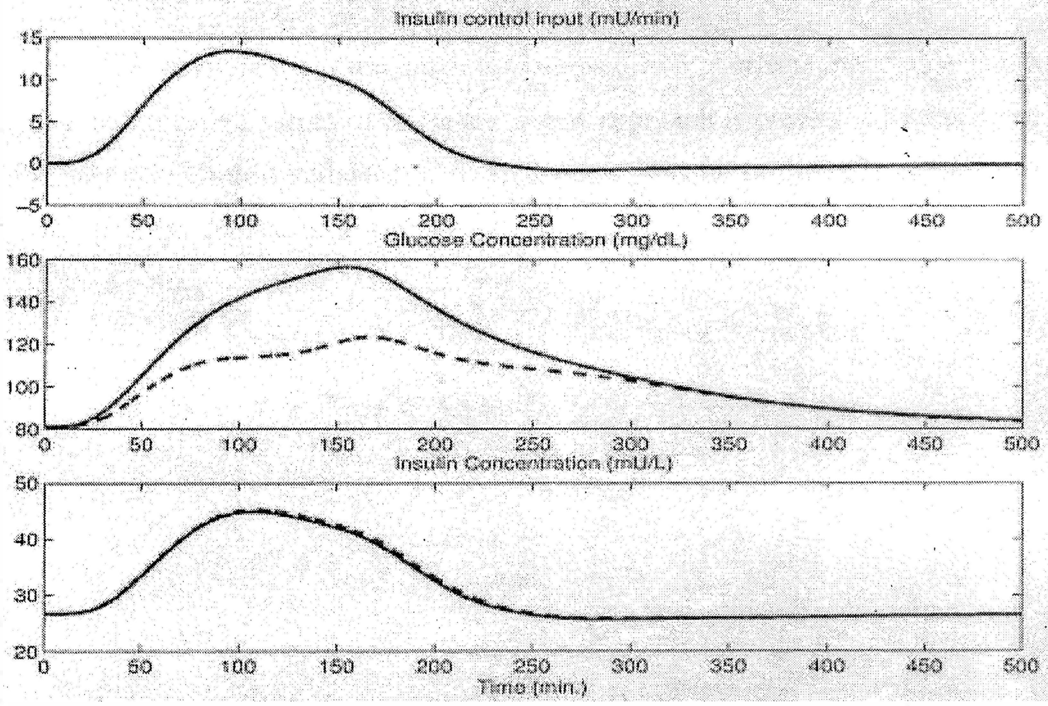


Figure 2.25 The LPV based robust controller with induced $\mathcal{L}_2$ –norm minimisation guaranteed in case of the original nonlinear Sorensen's model (solid) and the considered polytopic region (dashed)

Dua *et al.*, (2009) presented techniques within a multi-parametric model predictive control (mp-MPC) framework to regulate blood glucose concentrations for type 1 diabetes. The typical form of MPC problem is represented in equation (2.19) and it can be recast as follows:

$$\min_{u} \; J(U, x(t)) = \varepsilon_{t+N_y|t}^{T} P \varepsilon_{t+N_y|t} + \sum_{k=0}^{N_{y-1}} \left[\varepsilon_{t+k|t}^{T} Q \varepsilon_{t+k|t} + u_{t+k}^{T} R u_{t+k}\right] \qquad (2.42)$$

s.t. $\varepsilon_{t+k|t} \geq -x_{t+k|t}$, $k \geq 0$ (2.43)

$\varepsilon_{t+k|t} \geq x_{t+k|t}$, $k \geq 0$ (2.44)

$x_{min} \leq x_{t+k|t} \leq x_{max}$, $k = 1, \ldots, N_c$ (2.45)

$u_{min} \leq u_{t+k} \leq u_{max}$, k = 1,..., N_c (2.46)

$x_{t+k+1|t} = A x_{t+k|t} + B u_{t+k}$, $k \geq 0$ (2.47)

Where $\varepsilon(t)$ is a vector which provides $||x(t)||$. Equations (2.43) and (2.44) allow positive and negative values of x to be weighed equally in the objective function (2.42). Constraints (2.43) and (2.44) are modified so as to enforce asymmetric weights on positive and negative deviations as follows:

$$\tau - \varepsilon_{t+k|t} \geq -x_{t+k|t} \ , \ k \geq 0 \tag{2.48}$$

$$\tau + \varepsilon_{t+k|t} \geq x_{t+k|t} \ , \ k \geq 0 \tag{2.49}$$

The objective of this asymmetric formulation is to allow different effects in the objective function on its positive and negative values of x. Hence, a new approach is presented in this paper which provides the vector of control variables, U as an explicit function of the state variables, $x(t)$. For this case, two MPC problems can be formulated:

$$\begin{aligned} \min_{u} \ J(U,x(t)) = x^T_{t+N_y|t}Px_{t+N_y|t} + \sum_{k=0}^{N_{y-1}} \left[x^T_{t+k|t}Q^{-}x_{t+k|t} + u^T_{t+k}Ru_{t+k}\right] \quad (2.50) \\ s.t. \ \ x_{min} \leq x_{t+k|t} \leq 0, \ \ k = 1, \dots, N_c \\ u_{min} \leq u_{t+k} \leq u_{max,} \ \mathrm{k} = 1,\dots, N_c \\ x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k,} \quad k \geq 0 \end{aligned}$$

and

$$\begin{aligned} \min_{u} \ J(U,x(t)) = x^T_{t+N_y|t}Px_{t+N_y|t} + \sum_{k=0}^{N_{y-1}} \left[x^T_{t+k|t}Q^{+}x_{t+k|t} + u^T_{t+k}Ru_{t+k}\right] \quad (2.51) \\ s.t. \quad 0 \leq x_{t+k|t} \leq x_{max}, \ \ k = 1, \dots, N_c \\ u_{min} \leq u_{t+k} \leq u_{max,} \ \ \mathrm{k} = 1,\dots, N_c \\ x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k,} \ k \geq 0 \end{aligned}$$

In the constraint prioritisation work, as an alternative to the asymmetric objective function approach to reduce hypoglycemic excursions, the problem can then be formulated as a set of objectives which represent constraints on blood glucose concentration. In order to achieve this, the control problem similar to (2.19) can then be formulated as follows:

$$\begin{aligned} \min_{U,P,O,v,\varepsilon} \alpha^T P + \beta^T O + \gamma \sum_{k=0}^{N_y} \varepsilon_{t+k|t} + \delta \sum_{k=0}^{N_{y-1}} v_{t+k|t} \quad (2.52) \\ s.t. \quad \varepsilon_{t+k|t} \geq x_{t+k|t} \ , k \geq 0 \\ \varepsilon_{t+k|t} \geq -x_{t+k|t} \ , \ k \geq 0 \end{aligned}$$

$$
\begin{gathered}
v_{t+k} \geq u_{t+k}, k \geq 0 \\
v_{t+k} \geq -u_{t+k}, k \geq 0 \\
u_{min} \leq u_{t+k} \leq u_{max,} \quad \mathrm{k} = 1, \ldots, N_c \\
x_{t+k|t} = Ax_{t+k|t} + Bu_{t+k,} \; k \geq 0 \\
P_1 \geq P_2 \\
P_2 \geq P_3 \\
\vdots \\
P_{N_P-1} \geq P_{N_P} \\
O_1 \geq P_1 \\
O_2 \geq P_2 \\
\vdots \\
O_{N_P} \geq P_{N_P} \\
\psi(O_{i,}x_{t+k|t}) \leq 0, \;\; k \geq 0
\end{gathered}
$$

The results of the study are represented in Figure 2.26. As shown in Figure 2.26, when there are symmetric weightings on hypo and hyperglycemia, there is violation of the constraint on hypoglycemia and the blood glucose concentration falls below 60 mg/dL (Figure 2.26(a)). Hence, two multi-parametric programmes need to be solved in order to avoid this situation; one for glucose concentration $\geq$ 81mg/dL and the other for glucose concentration $\leq$ 81mg/dL. Figure 2.26(b) shows the variation of blood glucose concentrations with time for sinusoidal disturbance, a much smaller negative deviation from 81 mg/dL is obtained and hypoglycemia is avoided. Figure 2.26 (c) shows the performance of the controller, which although is not as good as observed for asymmetric case (Figure 2.26(b)), it is much better than for the symmetric case (Figure 2.26 (a)) and hypoglycemia is avoided. The study concludes that the proposed techniques adequately address the issue of reducing hypoglycemic excursions, which are considered to be more harmful than the hyperglycemic excursions, and hence it is expected to provide a safe and more convenient method for insulin delivery.

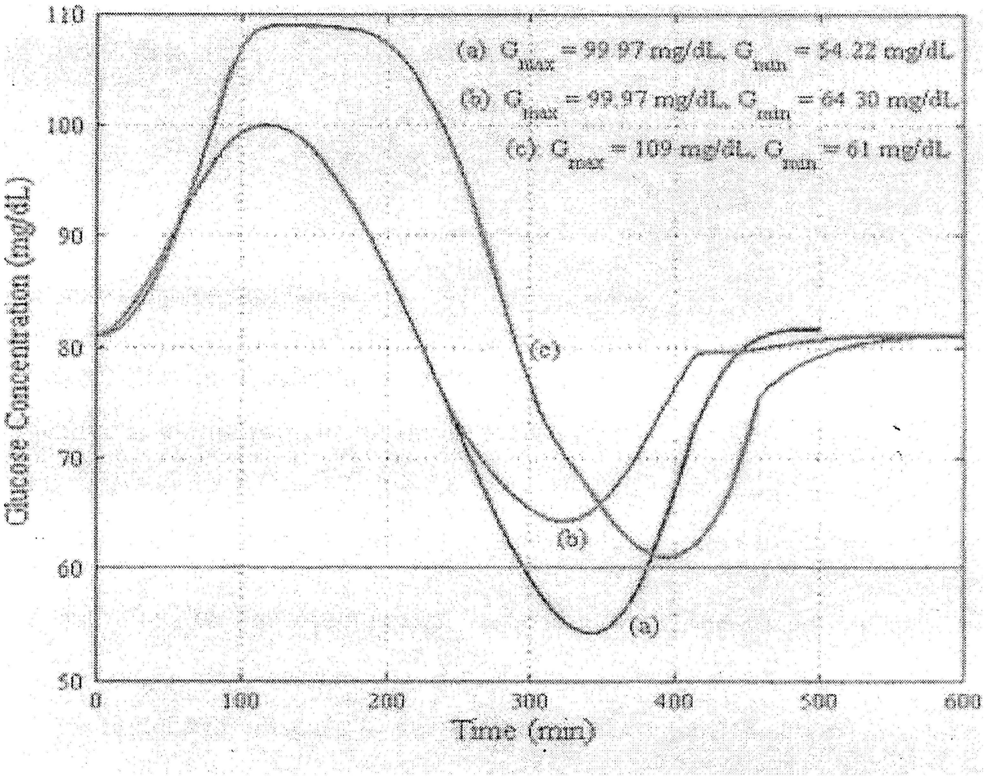


Figure 2. 26 Glucose concentration profile for (a) Symmetric objective function, (b) Asymmetric objective function, and (c) Prioritised constraints

CHAPTER 3

METHODOLOGY

3.1 Multi-parametric programming

Multi-parametric programming is a technique for solving any optimisation problem, where the objective is to minimise or maximise a performance criterion subject to a given set of constraints and where some of the parameters vary between specified lower and upper bounds. The main characteristic of multi-parametric programming is its ability to obtain:

i) the objective and optimisation variable as functions of the varying parameters, and
ii) the regions in the space of the parameters where these functions are valid.

Multi-parametric programming has been applied to a number of applications as follows:

i) hybrid parametric / stochastic programming,
ii) process planning under uncertainty,
iii) material design under uncertainty,
iv) multi-objective optimisation,
v) flexibility analysis
vi) computation of singular multivariate normal probabilities, and
vii) model predictive control.

The advantage of using multi-parametric programming to address these problems is that for problems pertaining to plant operations, such as for process planning, scheduling and control, one can obtain a complete map of all the optimal solutions. Hence, as the operating conditions vary, one does not have to re-optimise for the new set of conditions, since the optimal solution is already available as a function of the operating conditions.

A general parametric nonlinear programming problem can be represented as follows:

$$\begin{aligned} &\min_{K} \int (x,\theta), \\ \text{s.t. } & g_i\,(x,\theta) \le 0, \;\; \forall\; i = 1,\dots\dots.., p, \\ & h_j\,(x,\theta) = 0, \;\; \forall\; i = 1,\dots\dots.., p, \\ & x \in X \subseteq \mathbb{R}^n, \\ & \theta \in \Theta \subseteq \mathbb{R}^m, \end{aligned} \qquad (3.1)$$

where f, g, and h are twice continuously differentiable in x and θ. The first-order Karush-Kuhn-Tucker (KKT) optimality conditions for (Eq. 3.1) are given as follows:

$$\begin{aligned} &\nabla \mathcal{L} = 0, \\ &\lambda_i\, g_{i,}(x,\theta) = 0, \quad \lambda_i \ge 0, \;\forall\; i = 1,\dots\dots.., p, \\ &h_j\,(x,\theta) = 0, \;\; \forall\; j = 1,\dots\dots.., q, \\ &\mathcal{L} = \int (x,\theta) + \sum_{i=1}^{p} \lambda_i\, gi,(x,\theta) + \sum_{j=1}^{q} \mu_j h_j,(x,\theta) \end{aligned} \qquad (3.2)$$

The main sensitivity result for (Eq 3.1) derives directly from system (Eq. 3.2) as shown in Theorem 1.

a) **Theorem 1.** Basic sensitivity theorem.

Let x_0 be a vector of parameter values and (u_0, λ_0, μ_0) a KKT triple corresponding to Eq. (3.2), where λ_0 is nonnegative and u_0 is feasible in Eq. (3.1). Also assume that (i) strict complementary slackness (SCS) holds, (ii) the binding constraint gradients are linearly independent (LICQ: linear independence constraint qualification), and (iii) the second order sufficiently conditions (SOSC) hold. Then, in the neighbourhood of x_0, there exists a unique, once continuously differentiable function, z(x)=[u(x), λ(x), μ(x)], satisfying Eq. (3.2) with: z (x_0)=[u(x_0), λ(x_0), μ(x_0)], where u(x) is a unique isolated minimiser for Eq. (3.1), and

$$\begin{pmatrix} \frac{du(x_0)}{dx} \\ \frac{d\lambda(x_0)}{dx} \\ \frac{d\mu(x_0)}{dx} \end{pmatrix} = -(M_0)^{-1} N_0 \qquad (3.3)$$

Where M_0 and N_0 are the Jacobian of system Eq. (3.2) with respect to z and x:

$$M_0 = \begin{pmatrix} \nabla^2\mathcal{L} & \nabla g_1 & \cdots & \nabla g_p & \nabla h_1 & \cdots & \nabla h_q \\ -\lambda_1 \nabla^T g_1 & -g_1 & & & & & \\ \vdots & & \ddots & & & & \\ -\lambda_p \nabla^T g_p & & \cdots & -g_p & & & \\ \nabla^T h_1 & & & & & & \\ \vdots & & & & & & \\ \nabla^T h_q & & & & & & \end{pmatrix}$$

Collary 1. First order estimation of $x(\theta)$, $\lambda(\theta)$, $\mu(\theta)$, near $\theta = \theta_0$: Under the assumptions of Theorem 1, a first-order approximation of $[x(\theta), \lambda(\theta), \mu(\theta),]$ in a neighbourhood of θ_0 is:

$$\begin{bmatrix} x(\theta) \\ \lambda(\theta) \\ \mu(\theta) \end{bmatrix} = \begin{bmatrix} x_0 \\ \lambda_0 \\ \mu_0 \end{bmatrix} + (M_0)^{-1} \, . \, N_0 \, . \, \theta + o(\|\theta\|), \qquad (3.4)$$

Where $(x_0, \lambda_0, \mu_0) = [x(\theta_0), \lambda(\theta_0), \mu(\theta_0)]$, $M_0 = M(\theta_0)$, $N_0 = N(\theta_0)$, and $\phi(\theta) = o(\|\theta\|)$ means that $\frac{\phi(\theta)}{\|\theta\|} \to 0$ as $\theta \to \theta_0$.

Despite being a simple and linear expression, Eq. (3.4) may lead to complex computational problems, since in the general nonlinear case the Jacobians of system (Eq. 3.2) are in most of the cases complex. Fortunately, it simplifies when Eq. (3.1) has a quadratic objective function, linear constraints, and the parameters appear on the right-hand side of the constraints:

$$z(\theta) = \min_{x} c^T x + \frac{1}{2} x^T Q x$$

$$\text{s.t. } Ax \leq b + F\theta, \qquad (3.5)$$

$$x \in X \subseteq \mathbb{R}^n,$$

$$\theta \in \Theta \subseteq \mathbb{R}^m,$$

where c is a constant vector of dimension n, Q is an (n x n) symmetric positive definite constant matrix, A is a (p x n) constant matrix, F is a (p x m) constant matrix, b is a constant vector of dimension p, and X and Θ are compact polyhedral convex sets of dimensions n and m, respectively. Note that a term of the form θ^T Px in the objective function can also be addressed in the above formulation, as it can be transformed into the form given in Eq. (3.5) by substituting x = s – $Q^{-1}P^T\theta$, where s is a vector of arbitrary variables of dimension n and P is a constant matrix of dimension (m x n).

An application of Theorem 1 to Eq. (3.5) at [$x(\theta_Q)$, θ_Q] gives the following result:

$$\begin{pmatrix} \frac{dx(\theta_Q)}{d\theta} \\ \frac{d\lambda(\theta_Q)}{d\theta} \end{pmatrix} = - (M_Q)^{-1} N^Q \qquad (3.6)$$

Where

$$M_Q = \begin{bmatrix} Q & A_1^T & \dots & A_p^T \\ -\lambda_1 A_1 & -V_1 & & \\ \vdots & & \ddots & \\ -\lambda_p A_p & & & -V_p \end{bmatrix}, \qquad (3.7)$$

$$N_Q = [Y, \lambda_1 F_1, \dots\dots, \lambda_p F_p]^T,$$

$$V_i = A_i x(\theta_Q) - b_i - F_i\,\theta_Q \text{ ,e}$$

And Y is a null matrix of dimension (n x m). Thus, in the linear-quadratic optimisation problem, the Jacobians reduce to a mere algebraic manipulation of the matrices declared in Eq. (3.5).
In the neighbourhood of the KKT point, [$x(\theta_Q)$, θ_Q], Collary 1 writes as follows:

$$\begin{bmatrix} x_Q(\theta) \\ \lambda_Q(\theta) \end{bmatrix} = (-M_Q)^{-1} N_Q (\theta - \theta_Q) + \begin{bmatrix} x_{(\theta_Q)} \\ \lambda_{(\theta_Q)} \end{bmatrix}, \qquad (3.8)$$

Note that when assumptions in Theorem 1 are respected M_Q is always invertible. This is where parametric programming detaches from the sensitivity analysis theory. Whilst sensitivity analysis stops here, where we know what happens if the process conditions deviate from the nominal values to some value in its neighbourhood, parametric programming is concerned with the whole range of parametric variability. The former associates with the uncertainty, whereas the later relates to the variability of the process.

The space of θ where this solution (Eq.3.8) remains optimal is defined as the critical region, CR^Q , and can be obtained by using feasibility and optimality conditions. Note that for convenience and simplicity in presentation, the notation CR is used to denote the set of points in the space of θ that lie in CR as well as to denote the set of inequalities which define CR.

Feasibility is ensured by substituting $x_Q(\theta)$ into the inactive inequalities given in Eq. (3.5), whereas the optimality condition is given by $\bar{\lambda}_Q(\theta) \geq 0$, where $\bar{\lambda}_Q(\theta)$ corresponds to the vector of active inequalities, resulting in a set of parametric constraints. This can be represented by:

$$CR^R = \{\breve{A}\, x_Q(\theta) \leq \breve{b} + \breve{F}\theta, \bar{\lambda}_Q(\theta) \geq 0, CR^{IG}\}, \tag{3.9}$$

where $\breve{A}$, $\breve{b}$ and $\breve{F}$ correspond to the inactive inequalities and CR^{IG} represents a set of linear inequalities defining an initial given region. From the parametric inequalities thus obtained, the redundant inequalities are removed and a compact representation of CR^Q is obtained as follows:

$$CR^Q = \Delta\{CR^R\}, \tag{3.10}$$

where Δ is an operator which removes redundant constraints. Note that a CR^Q is a polyhedral region. Once CR^Q has identified for a solution, $[x(\theta_Q), \theta_Q]$, the next step is to define the rest of the region, CR^{rest} as follows:

$$CR^{rest} = CR^{IG} - CR^Q \tag{3.11}$$

Another set of parametric solutions in each of these regions is then obtained and corresponding CRs are obtained. The algorithm terminates when there are no regions to be explored. In other words, the algorithm terminates when the solution of the differential equation (3.6) has been fully approximated by first-order expansions.

The main steps of the algorithm are outlined in Table 3.1. Note that while defining the rest of the regions, some of the regions are split and hence the same optimal solution may be obtained in more than one region. Therefore, the regions with the same optimal solution are united and a compact representation of the final solution is obtained.

When θ is present on the right-hand side of the constraints, the solution space of Eq.(3.1) is convex and continuous. Since Eq. (3.5) is a special case of Eq. (3.1), its solution has these properties as well. Due to its importance, these properties are proved specifically for Eq. (3.5) in the next theorem.

Table 3.1 Multi-parametric Quadratic Programming (mp-QP) Algorithm

Step 1	In a given region, solve Eq.(3.5) by treating θ as a free variable to obtain a feasible point $[\theta_Q]$.
Step 2	Fix $\theta = \theta_Q$ and solve Eq. (3.5) to obtain $[x(\theta_Q), \lambda(\theta_Q)]$
Step 3	Compute $[-(M_Q)^{-1} N_Q]$ from Eq. (3.6)
Step 4	Obtain $[x_Q(\theta), \lambda_Q(\theta)]$ from Eq. (3.8)

Step 5 Form a set of inequalities, CR^{R} as described in Eq. (3.9)

Step 6 Remove redundant inequalities from this set of inequalities and define the corresponding CR^{Q} as given in Eq. (3.10)

Step 7 Define the rest of the region, CR^{rest} as given in Eq. (3.11)

Step 8 If no more regions to explore, go the next step, otherwise go to Step 1

Step 9 Collect all the solutions and unify the regions having the same solution to obtain a compact representation

b) Theorem 2.

Let us consider the mp-QP (Eq. 3.5) and let Q be positive definitive, Θ convex. Then the set of feasible parameters $\Theta_f \subseteq \Theta$ is convex, the optimizer $x(\theta)$: $\Theta_f \mapsto \mathbb{R}^n$ is continuous and piecewise affine, and the optimal solution $z(\theta) : \Theta_f \mapsto \mathbb{R}$ is continuous, convex, and piecewise quadratic.

Proof. We first prove convexity of Θ_f and $z(\theta)$. Take generic $\theta_1, \theta_2 \in \Theta_f$ and let $z(\theta_1)$, $z(\theta_2)$ and x_1, x_2 be the corresponding optimal values and minimisers. Let $\alpha \in [0,1]$ define $x_\alpha \triangleq \alpha x_1+(1-\alpha)x_2$, $\theta_\alpha \triangleq \alpha\theta_1 + (1-\alpha)\theta_2$. By feasibility, x_1, x_2 satisfy the constraints $Ax_1 \leq b + F\theta_1$, $Ax_2 \leq b+F\theta_2$. These inequalities can be linearly combined to obtain $Ax_\alpha \leq b + F\theta_\alpha$ and therefore x_α is feasible for optimisation problem (Eq. 3.5). Since a feasible solution, $x(\theta_\alpha)$, exists at θ_α , an optimal solution exists at θ_α and hence Θ_f is convex.

The optimal solution at θ_α will be less than or equal to the feasible solution:

$$z(\theta_\alpha) \leq c^T x_\alpha + \frac{1}{2} x_\alpha^T Q x_\alpha$$

and hence,

$$z(\theta_\alpha) - [\alpha\left(c^T x_1 + \frac{1}{2} x_1^T Q x_1\right) + (1-\alpha)(c^T x_2 + \frac{1}{2} x_2^T Q x_2)]$$

$$\leq c^T x_\alpha + \frac{1}{2} x_\alpha^T Q x_\alpha - [\alpha\left(c^T x_1 + \frac{1}{2} x_1^T Q x_1\right) + (1-\alpha)(c^T x_2 + \frac{1}{2} x_2^T Q x_2)]$$

$$= \frac{1}{2} [\alpha^2 x_1^T Q x_1 + (1-\alpha)^2 x_2^T Q x_2 + 2\alpha(1-\alpha) x_2^T Q x_1 - \alpha x_1^T Q x_1 - (1-\alpha) x_2^T Q x_2]$$

$$= -\frac{1}{2}\alpha(1-\alpha)(x_1 - x_2)^T Q(x_1 - x_2) \leq 0,$$

which means that,

$$z(\alpha\theta_1 + (1-\alpha)\theta_2) \leq \alpha z(\theta_1) + (1-\alpha)z(\theta_2), \forall\, \theta_1, \theta_2 \in \Theta, \forall\alpha \in [0.1] \qquad (3.12)$$

proving the convexity of z(θ) on Θ_f.

Within the closed polyhedral regions, CR^Q, in Θ_f the solution x(θ) is affine (Corollary 1). The boundary between two regions belongs to both closed regions. Because the optimum is unique, the solution must be continuous across the boundary. The fact that z(θ) is continuous and piecewise quadratic follows trivially.

Remark: Multi-parametric linear program:

Note that when Q is a null matrix, (Eq. 3.5) reduces to a multi-parametric linear program (mp-LP). This does not affect the solution procedure described above and the algorithm remains the same. This is because the results presented in the theorems are still valid as explained next. The results presented in Theorem 1 continue to hold true and SOSC is valid in spite of the fact that Q is a null matrix. For mp-LPs, x is an affine function of θ and λ remains constant in a CR and therefore Corollary 1 can be used. Whilst the results of Theorem 2 regarding Θ_f and x(θ) are still valid, z(θ) simplifies to a continuous, convex, and piecewise linear function of θ.

Hence, at the end of the algorithm the solution obtained is a conditional piecewise function of the parameters and Theorem 2 implies that the optimal function computed, z(θ), is continuous and convex. (Pistikopoulos *et al.* , 2007)

3.2 Model Predictive Control (MPC)

Consider the general mathematical description of discrete-time, linear time-invariant systems: $x_{t+1} = Ax_t + Bu_t$

$$y_t = Cx_t \qquad (3.13)$$

s.t. $y_{min} \leq y_t \leq y_{max}$,

$u_{min} \leq u_t \leq u_{max}$,

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, and $y_t \in \mathbb{R}^p$, are the state, input and output vectors, respectively, subscripts min and max denote lower and upper bounds, respectively, and the matrix pair (A,B) is stabilisable.

The linear MPC problem for regulating (3.13) to the origin is posed as the following quadratic programming problem (from Section 3.1):

$$\min_{u} \; J(U, x_t)) = x'_{t+N_y|t} P x_{t+N_y|t} + \sum_{k=0}^{N_y-1} \left[x'_{t+k|t} Q x_{t+k|t} + u'_{t+k} R u_{t+k}\right] \quad (3.14)$$

$$\begin{aligned} \text{s.t.} \quad & y_{min} \leq y_{t+k|t} \leq y_{max}, k = 1, \dots, N_c, \\ & u_{min} \leq u_{t+k} \leq u_{max,} \; k = 0,1,\dots, N_c, \\ & x_{t|t} = x_{t,}, \\ & y_{t+k|t} = Cx_{t+k|t,} \quad k \geq 0, \\ & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k,} \quad k \geq 0, \\ & u_{t+k} = Kx_{t+k|t}, \quad N_u \leq k \leq N_y, \end{aligned}$$

where $U \triangleq \{u_t, \dots\dots, u_t + N_u - 1\}$, $Q = Q' \geq 0$, $R = R' > 0$, $P \geq 0$, $(Q^{1/2}, A)$ is detectable, N_u, N_y, N_c are the input, output, and constraint horizon, respectively, such that $N_y \geq N_u$ and $N_c \leq N_y-1$, and K is a stabilising state feedback gain. Eq. (3.14) is solved repetitively at each time t for the current measurement x_t and the vector of predicted state variables, $x_{t+1|t}, \dots, x_{t+k|t}$ at time t+1,..., t+k, respectively, and corresponding optimal control actions, $U^* = \{u^*_{t,\dots\dots}, u^*_{t+k-1}\}$is obtained. The input that is applied to the system is the first control action:

$$u_t = u^*_t$$

and the procedure is repeated at time t+1, based on the new state x_{t+1}.

The state feedback gain K and the terminal cost function matrix P are used to guarantee stability for the MPC (Eq. 3.14). The stability problem of the MPC has been treated extensively; nevertheless we will briefly present two methods to obtain K and P. One possible choice is to set K=0 and P to be the solution of the discrete Lyapunov equation:

$$P = A'PA + Q$$

However, this solution is restricted only to open-loop stable systems, since the control action is stopped after N_u steps. Alternatively, one can choose K and P as the solutions of the unconstrained, infinite-horizon linear quadratic regulation (LQR) problem, i.e., when $N_c=N_u=N_y=\infty$,

$$K = - (R + B'PB)^{-1} B'PA,$$

$$P = (A + BK)'P (A + BK) + K'RK + Q \qquad (3.15)$$

This is possibly the most popular method for obtaining the K and P matrices. Introducing the following relation, derived from (Eq. 3.13),

$$x_{t+k|t} = A^{k} x_t + \sum_{j=0}^{k-1} A^{j} B u_{t+k-1-j} \qquad (3.16)$$

In Eq.(3.14) results in the following quadratic programming or QP problem.

$$J^*(x_t) = \min_U \left\{\frac{1}{2} U'HU + x_t'FU + \frac{1}{2} x_t'Yx(t)\right\} \qquad (3.17)$$

$$\text{s.t. } GU \leq W + Ex_t,$$

where $U \triangleq [u_t', \ldots\ldots, u_{t+Nu-1}']' \in \mathbb{R}^s$, $s \triangleq mN_u$, is the vector of optimisation variables,

$H = H' > 0$, and H, F, Y, G, W, E are obtained from Q, R, and Equations (3.14) – (3.16). Thus, the MPC is applied by repetitively solving the QP problem (Eq. 3.17) at each time $t \geq 0$ for the current value of the state x_t. Due to this formulation, the solution U* of the QP is a function $U^*(x_t)$ of the state x_t, implicitly defined by Eq.(3.17) and the control action u_t is given by:

$$u_t = [I \quad 0 \quad \ldots \quad 0]\, U^*(x_t) \qquad (3.18)$$

The problem in (3.14) obviously describes the constrained linear quadratic regulation problem, while (3.17) is the formulation of the MPC as a QP optimisation problem. Despite the fact that efficient QP solvers are available to solve Eq.(3.17), computing the input u_t online may require significant computational effort. The solution of (3.14) via multi-parametric programming means, which avoids the repetitive optimisation, was previously discussed in Section 3.1. (Pistikopoulos *et al.*, 2007)

3.3 gPROMS^{TM} (general PROcess Modelling Systems)

3.3.1 What is gPROMS^{TM}?

gPROMS^{TM} is a leading advanced process modelling environment for the process industries. It can be applied across many applications areas such as reaction, crystallisation, separation, fuel cell processes, biotechnology, batch processing, control and automation in all process sectors including chemicals, energy, pharmaceuticals, polymers, food, beverage, industrial gases, pulp, paper, minerals, mining, bio-treatment processes, and metals production. gPROMS's process modelling, process simulation and optimisation capabilities are used to generate high-accuracy predictive information for decision support in product and process innovation, design and operation. It is applied by major process and technology organisations throughout the world, as well as for research and teaching at 200 academic institutions worldwide. gPROMS^{TM} has many major advantages over other modelling systems on the market, resulting from its modelling power and the sophistication of the models it is possible to create.

gPROMS^{TM} is an equation-oriented modelling system used for building, validating and executing first-principles models within a flow sheeting framework. Models are constructed in the gPROMS^{TM} Model builder by writing down the fundamental chemistry, physics, chemical engineering, operating procedures and other relationships that govern the process or product behaviour. The resulting model is then validated against observed data typically, laboratory, pilot plant or operating data, to adjust model parameters such as heat transfer coefficients to match reality as closely as possible. Once a model exits, it can be solved in many different ways to perform many different activities; for example, steady state simulation, dynamic simulation, parameter estimation, model-based experiment design, integer optimisation or generation of linearised models for use in control and online optimisation, across the process cycle.

gPROMS^{TM} is the product supplied by Process Systems Enterprise Limited (PSE) which is one of the world's foremost providers of process modelling technology and model-based engineering and innovation services to the process industries and their technology suppliers. They can be contacted at this address: Process Systems Enterprise Limited, 6th. Floor East, 26-28 Hammersmith Grove, London W6 7HA, United Kingdom.

3.3.2 gPROMS^{TM} Fundamentals

gPROMS^{TM} process models are built from a number of fundamental building blocks or entities. A gPROMS^{TM} process model (for a simulation activity) consists of the following entities:

- Variable Types
- Models
- Processes

In addition:

- To execute Optimisation activities, Optimisation entities are required.
- To execute Model Validation activities (parameter estimation and experiment design), Parameter Estimation, Experiment Design and Experiment entities are required.

a) **Variable Types**

Variable types appear under the first entry in the Project Tree. In order to create your own Variable Types, you can either select New entity....from the Entity menu – choosing Variable Type as the Entity type. Once the Variable Type has been introduced, the following information should be provided:

- A default value for Variables of this type. This value will be used as an initial guess for any iterative calculation involving Variables of this type, unless it is overridden for individual variables or a better guess is available from a previous calculation.
- Upper and lower bounds on the values of Variables of this type. Any calculation involving Variables of this type must give results that lie within these bounds. These bounds can be used to ensure that the results of a calculation are physically meaningful. Again, these bounds may be overridden for individual Variables of this type.
- An optional unit of measurement. Users are encouraged to provide this in order to aid Model readability.

b) Models

In gPROMS^{TM}, Model Entities are the central part of any process model. A working TM project will contain at least one model. A Model is defined as a set of quantities and mathematical equations that, coupled with a set of specifications, describe the behaviour of a given system.

The gPROMS^{TM} language declaration for a basic model will typically consist of three parts:

- Parameters section
- Variable section
- Equation section

i) Parameters section

The parameter section is used to declare the parameter of a Model. Parameters are time-invariant quantities that will not, under any circumstances, be the result of a calculation. Quantities such as physical constants (pi, R, etc), Arrhenius coefficient and stoichiometric coefficients usually fall into this category.

Each parameter has a unique name (identifier) by which it can be referenced. Identifiers in the gPROMS^{TM} language start with a letter (a – z and A - Z) and may comprise letters, numbers (1-9) and underscore (_). The gPROMS^{TM} language is not case sensitive. Each parameter is also declared to be of a certain type (e.g. integer, logical or real).

Parameter declarations may optionally include the assignment of default values. For instance:

```
NoComp       AS INTEGER
NoReactions  AS INTEGER DEFAULT 1
```

Finally, note that the categorisation of certain quantities as parameters is sometimes tenuous. Designating a quantity as a parameter has the advantage of reducing the total number of variables in a model. However, this quantity can no longer be treated as an unknown in any future use of the model. Consider, for instance, the quantities that characterise the size and geometry of a vessel. From the point of view of dynamic simulation, these can be viewed as parameters. However, from the point of view of steady-state design calculations performed with the same model, these quantities may be considered unknowns under certain circumstances. It may, therefore, be better to classify them as variables.

ii) Variable section

The variable section is used to declare the Variables of a model. All quantities that are calculated in Model Equations must be declared as Model Variables. For instance:

HoldUp	AS Mass
FlowIn, FlowOut	AS MassFlowrate
Height	AS Length

Like Parameters, Variables are always Real continuous numbers. All Variables must be given a type, however, Variables Types are user-defined.

iii) Equation section

The equation section is used to declare the equations that determine the time trajectories of the variables already declared in the Variable section. The gPROMS^{TM} language is purely declarative. That is, the order in which the equations are declared is of no importance. Simple equations are equalities between two real expressions. These expressions may comprise:

- Integer or real constants
- Parameters that have been declared in the Parameter section
- Variables that have been declared in the Variable section. The special symbol $ preceding a variable name denotes the derivative with respect to time of that variable (e,g. $HoldUp etc).

Similarly to most programming languages, expressions are formed by combining the above operands with the arithmetic operators + (addition), - (subtraction), * (multiplication), / (division) and ^ (exponentiation), as well as built-in intrinsic functions (e.g. square root: SQRT ()). Intrinsic functions have the highest precedence priority, followed by the ^ operator and then the division and multiplication operators. The addition and subtraction operators have the lowest precedence. Naturally, parentheses may be used to alter these precedence rules as required. Finally, comments can be added to clarify the contents of the Model where needed. In gPROMS^{TM}, two types of comments are accepted. One type begins with # and extends to the end of the current line. The other type starts with { and ends with } and may span multiple lines. Moreover, comments of this type may be nested within one another.

c) Process

In the gPROMS^{TM} language, a Model is used to define the physical behaviour of a system and it usually contains Parameter, Variable and Equation declarations. A model can usually be used to study the behaviour of the system under many different circumstances. Each such specific situation is called a simulation activity. The coupling of Models with the particulars of a dynamic simulation activity is done in a Process.

A gPROMS^{TM} Project may contain multiple Processes, each corresponding to a different simulation activity (e.g. simulation of plant start up, simulation of plant shutdown, etc. Each such process must be given a different name and these will be automatically placed in alphabetical order in the gPROMS^{TM} Project Tree. A Process is partitioned into the following key sections:

- The Unit section
- The Set section
- The Assign section
- The initial section
- The SolutionParameters section
- The Schedule section

i) The Unit section

The first item of information required to set up a dynamic simulation activity is the process equipment under investigation. This is declared in the Unit section of a Process. Equipment items are declared as instances of Models. For example:

```
        UNIT
T101 AS BufferTank
```

creates an instance of Model BufferTank, named T101. T101 is described by the variables declared within the BufferTank Model and its time-dependent behaviour is partially determined by the corresponding equations.

ii) The Set section

Before an instance of a Model can actually be used in a simulation, all its parameters must be specified (unless they have been given default values). This is done in the Set section of a Process. For example,

```
SET
        T101.Rho                    :=    1000; # kg/m^3
        T101.CrossSectionalArea     :=    1; # m^2
        T101.Alpha                  :=    10;
```

sets the parameters of T101 to appropriate values. Note that:

- In order to refer to parameter Rho of instance T101 of Model BufferTank, the pathname notation T101.Rho is used.
- It is recommended that pathname completion is used to help construct full and valid pathnames correctly; this is available within all entries in gPROMS^{TM}. Semantic errors, such as referencing a quantity in a lower-level Model that doesn't exist, are only detected when a Model based activity is executed.
- It is also common, particularly for composite Models, to use the Within construct to complete pathnames
- Parameter values are set using the assignment operator (:=). In other words, the arithmetic expression appearing on the right hand side is first evaluated; its value is then given to the parameter appearing on the left hand side. This is another general rule of the gPROMS^{TM} language. gPROMS^{TM} always uses the symbol = to declare the equality of the two general expressions appearing on either side of this symbol.

iii) The Assign section

The set of equations resulting from the instantiation of Models declared in the Unit section is typically underdetermined. This simply means that there are more variables than equations. The number of degrees of freedom in the simulation activity is given by:

Number of degrees of freedom (N_{DOF}) = Number of variables – Number of equations

For the simulation activity to be fully defined, N_{DOF} variables must be specified as either constant values or given functions of time. Variables specified in this way are the input variables (or "inputs") of this simulation activity. The remainder of the variables are the unknown variables, the time variation of which will be determined by the solution of the system equations. Clearly, the number of unknowns is equal to the number of available equations; therefore we have a "square" system of equations.

The specification of input variables is provided in the Assign section of the Process. For instance:

```
ASSIGN
    T1O1.Fin      := 20;
```

Designates the inlet flowrate as an input and assign it a constant value of 20. Again, in order to emphasise the assignment form of these specifications, input specifications use the assignment operator (:=).

iv) The Initial section

Before dynamic simulation can commence, consistent values for the system variables at t=0 must be determined. To this end, a number of additional specifications are needed. These augment the system of equations that describe the behaviour of the system and result in a square system of equations at t=0. The solution of the latter provides the condition of the system at t=0. Traditionally, the term "initial condition" refers to a set of values for the differential variables at t=0. However, gPROMS^{TM} follows a more general approach in which the initial conditions are regarded as additional equations that hold at t=0 and can take any form. This, of course, allows or the traditional specification of "initial values" for the differential variables or, indeed, for any appropriate subset of system variables; however, it also makes possible the specification of much more general initial conditions as equations of arbitrary complexity.

The initial section is used to declare the initial condition information pertaining to a dynamic simulation activity. For instance:

```
INITIAL
    T101.Height = 2.1;
```

specifies an initial condition for the buffer tank system by stating the height of liquid in the tank at t=0 is 2.1. Note that, in contrast to the Set and Assign sections, the equality operators (=) is used here to emphasise the fact that initial conditions are general equations.

An initial condition that is frequently employed for the dynamic simulation of process systems is the assumption of steady-state, constraining the time derivatives of the differential variables to zero. In gPROMS^{TM}, this can be achieved by manually specifying all derivatives to be zero: For example;

```
INITIAL
    T101.$Holdup = 0;
```

However, this would be tedious for models with large numbers of differential variables, so the keyword STEADY_STATE may be utilised to specify this initial condition, as shown below:

```
INITIAL STEADY  STATE
```

v) The Solution Parameter Section

The user also has the option to control various aspects of model-based activities carried out in gPROMS^{TM} such as solver settings and output specifications. The Solution Parameter section is used for this purpose. For example;

```
SOLUTIONPARAMETERS
```

REPORTINGINTERVAL := 60;

The REPORTINGINTERVAL is the interval at which result values will be collected during the dynamic simulation (note that it does not affect the accuracy of the subsequent integration in any way). For this example, an interval of 60 is a reasonable choice. The REPORTINGINTERVAL may be overridden from the simulation execution dialog. The user does not need to give any settings in this section. In such a case, the user will be prompted to enter a REPORTINGINTERVAL in dialog box.

3.4 Dynamic Optimisation in gPROMS^{TM}

3.4.1 What is dynamic optimisation?

In order to introduce the various elements of the definition of the problem of dynamic optimisation, we consider the semi-batch reactor shown in Figure 3.1. Two exothermic reactions are taking place:

$A + B \rightarrow C$

$B + C \rightarrow D$

where A and B are the raw materials, C the desirable product, and D the unwanted by-product.

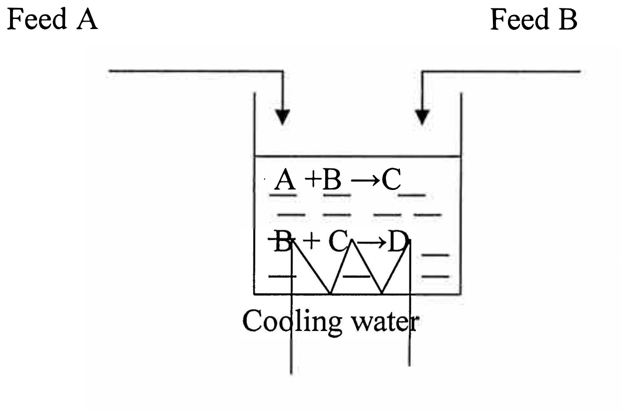


Figure 3.1 Batch Reactor

The reactor receives two independent inputs of pure A and B, and is cooled with cooling water circulating through a coil. Starting with an empty reactor, we are free to vary the in-flows of A and B, as well as the cooling water flowrate. For a given reactor design, our operational objective may be to determine the duration of the operation, and the time variation of the various material and energy flowrates over this duration, so as to maximise the final concentration of C. Of course, equipment design and resource availability usually impose certain limits within which our control manipulations must be maintained – for instance, there is an upper limit on the available flowrate of cooling water.

In general, the design of process operating in the transient domain also leads to problems that are similar to operational optimisation problems, but may have additional degrees of freedom. For instance, we may wish to determine the optimal geometry of the reactor in addition to the optimal way of operating it over time. Because of the transient nature of the underlying process, both the operational and design problems considered above are applications of dynamic optimisation and serve to introduce some of the importance features of this problem in its most basic form.

3.4.2 What is the mathematical problem?

In this section, we provide a mathematical statement of the class of dynamic optimisation solved by gPROMS$^{\mathrm{TM.}}$.

a) The process Model

We consider processes described by mixed differential and algebraic equations of the form: $f(x(t), \dot{x}(t), y(t), u(t), \upsilon) = 0$

Here $x(t)$ and $y(t)$ are the differential and and algebraic variables in the model while $\dot{x}(t)$ are the time derivatives of the $x(t)$ (i.e. $\dot{x}(t) \equiv \frac{dx}{dt}$). $u(t)$ are the control variables and υ then time invariant parameters to be determined by the optimisation. In the context of the batch reactor example considered earlier, the differential variables will typically correspond to fundamental conserved quantities (such as molar component holdups and internal energy), while y will include various quantities related to them (e.g. component molar concentrations and temperature). The input flowrates of A, B and cooling water are the control variables u while, in the design case, the volume of the reactor acts as a time invariant parameter υ.

b) The initial conditions

In general, gPROMS assumes that the initial t=0 condition of the system is described in terms of a set of general non-linear relations of the form:

$I\,(x(0),\, \dot{x}(0),\, y(0),\, u(0),\, \upsilon) = 0$

It is important to note that once we fix the time variation of the controls, *u(t)* and the values of any time invariant parameters, υ, the modelling equations together with the initial conditions completely determine the transient response of the system. In practice, we could determine this response by performing a gPROMS^{TM} dynamic simulation.

c) The objective function

Dynamic optimisation in gPROMS^{TM} seeks to determine:

- the time horizon, t_f,
- the values of the time invariant parameters, υ, and
- the time variation of the control variables, *u(t)*, over the entire time horizon $t \in [0, t_f]$,

so as to minimise (or maximise) the final value of a single variable z. This can be written mathematically as ;

$$\min_{t_f,\, \upsilon, u(t), t\, \in[0, t_f]} z(t_f)$$

Here the objective function variable, z is one of either the differential variables x or the algebraic variables y. In the context of the batch reactor example, t_f would be the duration of the batch reaction while z would be the concentration of component C (either a differential or an algebraic variable, depending on the model used). The above form of the objective function is not restrictive as might appear at first. In particular, it is worth nothing that:

- Maximisation can be carried out as well as minimisation.
- If we wish to optimise a function $\emptyset\ (x(t_f),\ \dot{x}(t_f),\ y(t_f),\ u(t_f),\ \upsilon)$ of several variables instead of a single variable, we can simply add an extra algebraic equation to the model:

$z = \emptyset\ (x, \dot{x}, y, u, v)$

The additional computational cost incurred because of this model extension is usually negligible.

- If we wish minimise or maximise the integral of a function $\emptyset\ (x, \dot{x}, y, u, v)$ over the entire time horizon, we can simply add the differential equation:

  $\dot{z} = \psi\ (x, \dot{x}, y, u, v)$

  together with the initial condition:

  $z\ (0) = 0$

  We can easily verify that this is equivalent to:

  $z\ (t_f) = \int_0^{t_f} \psi\ (\ x(t), \dot{x}(\mathrm{t}), y(t), u(t), v)\ dt$

  Again, very little additional computation cost is incurred in doing this.
- Minimising the time horizon itself can be achieved by adding the equation:

  $\dot{z} = 1.$

  together with the initial condition above.

Bounds on the optimisation decision variables

In practice, the time horizon t_f will often be subject to certain lower and upper bounds:

$$t_f^{min} \leq t_f \leq t_f^{max}$$

In some cases, t_f will, in fact, be fixed at a given value, t_f^*. This can be achieved by setting $t_f^{min} = t_f^{max} = t_f^*$

As we have already seen in the batch reactor example, it is likely that the control variables and time invariant parameters will also be subject to lower and upper bounds.

$u^{min} \leq u(t) \leq u^{max}, \forall\ \mathrm{t} \in [0, \mathrm{t_f}]$

$v^{\mathrm{min}} \leq v \leq v^{\mathrm{max}}$

Other constraint types

a) **End-point constraint**

In some applications, it is necessary to impose certain conditions that the system must satisfy at the end of the operation. These are called end-point constraints. For instance, in the batch reactor example, we may require:

- The final amount of material in the reactor to be at certain prescribed value
  We have an equality end-point constraint, e.g. $w(t_f) = w^*$ where w is one of the system variables (x or y). .
- The final temperature to lie within given limits.
  We have an inequality end-point constraint, e.g. $w^{min} \leq w(t_f) \leq w^{max}$

b) Interior-point constraints

We can also have constraints that hold at one or more distinct times, t_f during the time horizon). These are called interior-point constraints. These may be represented mathematically as:

$$w_I^{min} \leq w(t_I) \leq w_I^{max}$$

Where w is a system variable, and t_I is a given time.

We note that both interior and end-point constraints are special cases of point constraints. However, for convenience, gPROMS^{TM} treats them separately. It also treats any constraints that have to be satisfied at the initial time t=0 as interior-point constraints.

c) Path constraints

We may also have certain constraints that must be satisfied at all times during the system operation. If these path constraints are equalities, then often they can simply be added to the system model effectively converting one of the control variables u(t) into an algebraic variable, y. More often, they are inequalities of the form:

$$w^{min} \leq w(t) \leq w^{max}, \ \forall t \in [0, t_f]$$

For instance, in our batch reactor example, we may require that the temperature never exceed a certain value so as to avoid some unwanted side-reactions that are not explicitly considered by the model.

Classes of control variable profile

For the above dynamic optimisation problem to be well defined, we need to be rather more specific regarding type of the variation of the control variables over time that we are willing to consider. For instance, we could have:

a) Piece-wise constant controls – these remain constant at a certain value over a certain part of the time horizon before jumping discretely value over the next interval.

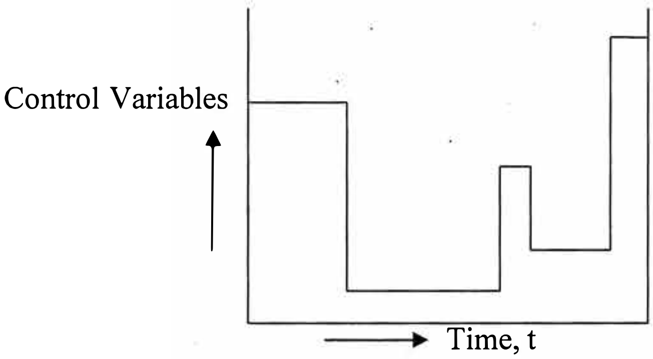


Figure 3.2 Piece-wise constant controls

b) Piece-wise linear controls – these take a certain linear time variation over certain part of , the time horizon before jumping discretely to a different linear variation over the next interval.

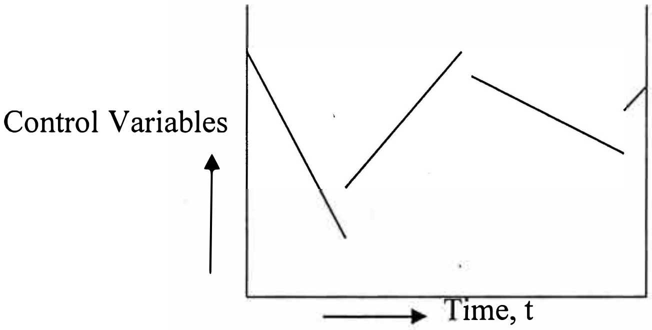


Figure 3.3 Piece-wise linear controls

c) Piece-wise linear continuous controls – these are similar to the piece-wise linear controls described above, with the additional requirement that their values be continuous at the interval boundaries.

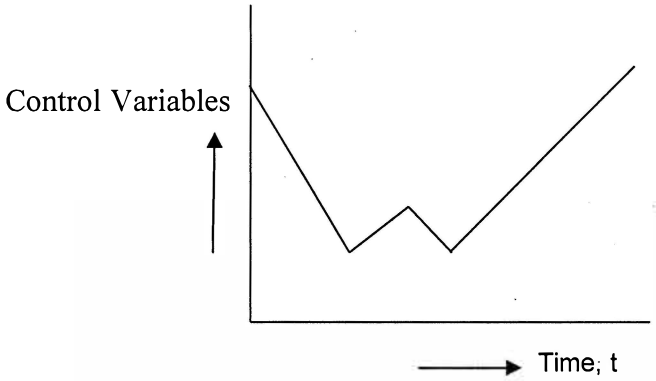


Figure 3.4 Piece-wise linear continuous controls

d) Controls that vary smoothly over time – perhaps as polynomials of a given degree.

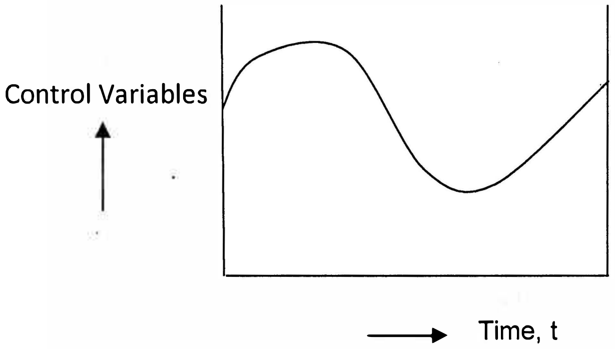


Figure 3.5 Polynomial controls

It is important to appreciate that, in most cases, the choice of the form of the control variables is an engineering rather than a mathematical issue: it very much depends on the capabilities of the actual control systems (automatic or manual) that, we will eventually use to implement these controls on the real plant. For instance, piece-wise constant controls may often be preferable to other types as they are much easier to implement.

Control Variable Profiles in gPROMS^{TM}

The dynamic optimisation facilities in gPROMS^{TM} support piecewise-constant and piecewise-linear controls of the types shown in Figures 3.2- 3.5 respectively. These are by far the most commonly encountered in practical applications. However, if necessary, it is relatively straightforward to introduce several other types of control. For instance, a piecewise-linear continuous control of the type shown in Figure 3.4 can be defined by adding the equations:

$\dot{z} = U$
$U = z + \alpha$
Where:
z is a new differential variable with initial condition z(0) =0;

U is a new piecewise-constant control variable to be determined by the optimisation.

α is a new time invariant parameter representing the initial value of u, to be determined by the optimisation.

We note that this is equivalent to:

$(t) = \alpha + \int_0^t U(\tau)d\tau$

which expresses that the the fact time gradient of the piecewise-linear continuous control is a piecewise-constant function of time.

Also a cubic polynomial control variation of the form:

$$(t) = \alpha + \beta t + \gamma t^2 + \delta t^3$$

can be introduced by adding the following to the model equations:

$\dot{z} = 1$

Together with the initial condition z(0) = 0, this equation effectively defines z as time.

$$u = \alpha + \beta z + \gamma z^2 + \delta z^3$$

By virtue of this equation, the variable u becomes one of the algebraic variables y to be determined by solving the model equations.

The actual control variation is determined by the values of α, β, γ,and δ which should now be treated as time invariant parameters v. *(http://www.psenterprise.com/gproms/index.html)*

CHAPTER 4

RESULTS AND DISCUSSION

Bergman minimal model (Bergman *et al.*, 1981) as shown schematically in Figure 2.16 (Dua *et al.*, 2006) was used in this study. Equations 2.23, 2.24, and 2.25 were employed to represent the three compartments; plasma glucose, plasma insulin, and effective insulin, respectively. The parameter values selected for P_1, P_2, P_3, V_I and n are the same as suggested by Fisher (1991), Dua and Pistikopoulos (2005) and Dua *et al.*, (2006). All ˎsimulation and optimisation works were carried out using gPROMS^{TM}.

4.1 **Control of Blood Glucose Level without meal disturbance**

In this section, we examine means of controlling the BGL without an infusion of exogenous glucose as previously studied by Finan *et al.* (2006).

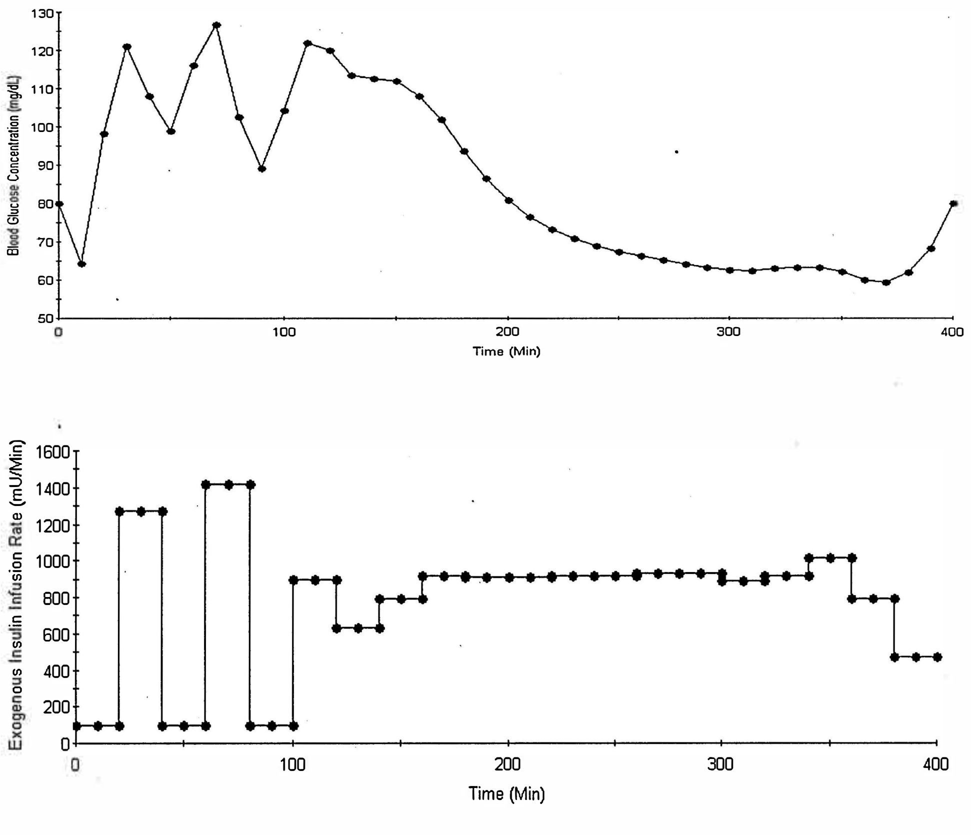


Figure 4.1 Transient glucose, G response to open loop step (basal) and impulse (bolus) inputs in u for 400 minutes

Transient responses to open loop changes in the insulin infusion rate, u were simulated in order to characterise the insulin to glucose dynamics of the Bergman minimal model. Responses of blood glucose level, BGL to open loop (basal) and impulse (bolus) inputs in exogenous insulin infusion rate, u is shown in Figure 4.1. The figure depicts the glucose level increases at first but then decreases prior to reaching its steady state. These results match with the study conducted by Finan *et al.* (2006) which use Hovorka model to simulate the glucose insulin dynamics.

Figures 4.2 and 4.3 show the extension of the above works which incorporate one day graph for the glucose-insulin dynamics representation. As shown in Figure 4.2, the BGL values

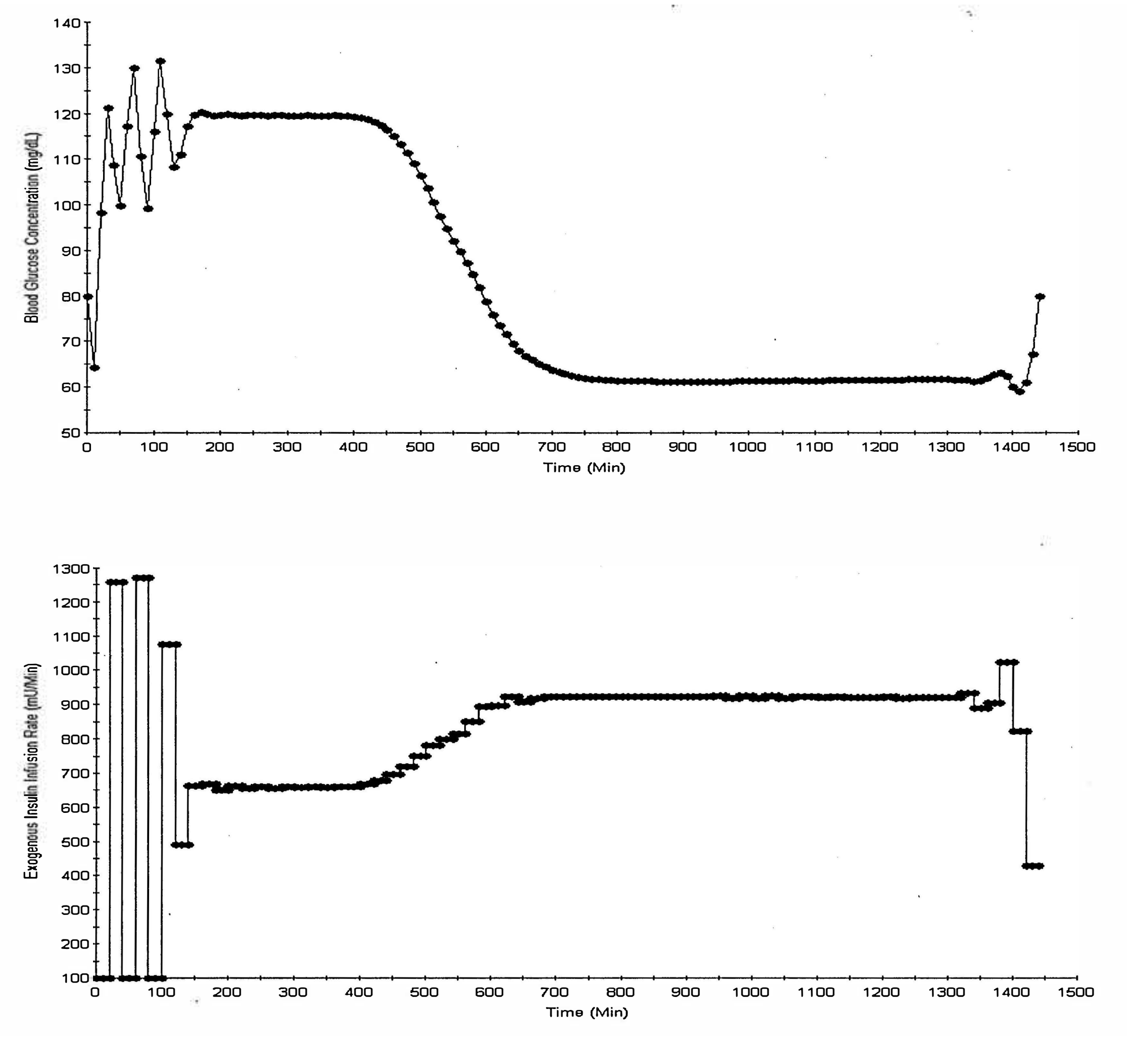


Figure 4.2 Transient glucose response to open loop step (basal) and impulse (bolus) inputs in exogenous insulin infusion rate for 1440 minutes (1st. Trial).

initially fluctuate around 80 to 130 mg/dL for the first 150 minutes but they tend to level off at 120 mg/dL for sometimes before dropping drastically to 60 mg/dL at which they reach their steady-state conditions. The BGL values, however, increase eventually to 80 mg/dL at the end of the simulation work. The graph also depicts the subsequent fluctuation on exogenous insulin infusion rate, u as BGL values change with time prior to levelling off, firstly at u of 700 mU/Min followed by secondly at 900 mU/Min before they finally drop to 400 mU/Min at the end of the simulation work. It can be concluded that, with the proposed algorithm, it can enable us to control BGL values within normoglycemic conditions (except at two highest peaks) so as to avoid hypoglycemia or hyperglycemia to happen. Another simulation work is carried out with a slightly refined algorithm for the purpose of fine tuning the previous results as shown in Figure 4.3.

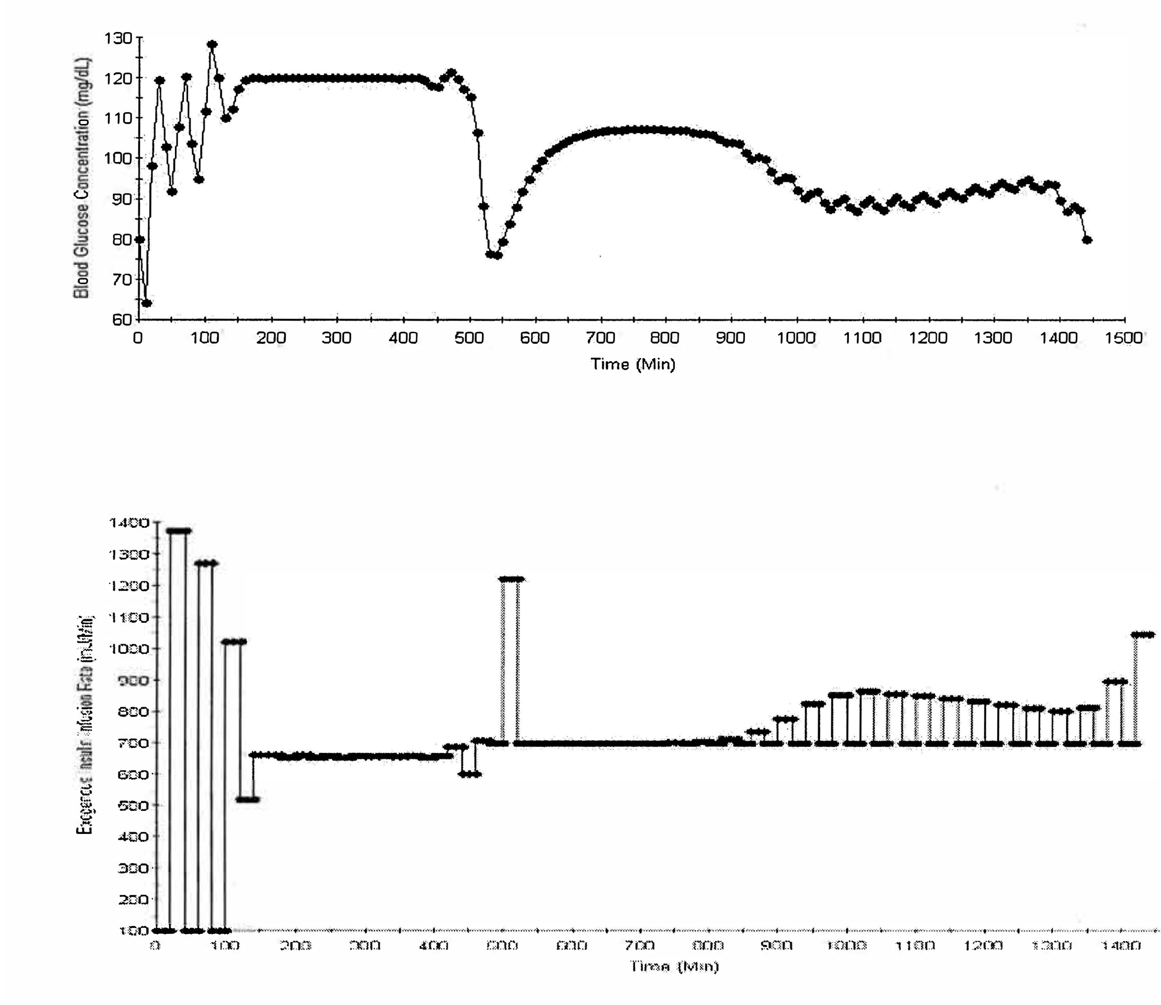


Figure 4.3 Transient glucose response to open loop step (basal) and impulse (bolus) inputs in u for 1440 minutes (2^{nd}. Trial)

As depicted in the figure, the BGL values behave in the same way as the first trial but they tend to increase sharply after the first drop (at 550 minutes). This behaviour has to do with the ability of u, maintaining the BGL values at 120 mg/dL or less (except at one point) as well as avoiding hypoglycemia to happen. This graph is very much similar to the graph obtained in the previous work using Hovorka model (Finan *et al*, 2006). The only major concern would be on the performance of the mechanical pump in which too much fluctuations tend to allow for frequent disruptions on its daily operation.

4.2 Control of Blood Glucose Level (BGL) following meals

In this section, we examine means of controlling the BGL following an infusion of exogenous glucose. In the model, we shall assume that oral glucose infusion commences at t=0 prior to which BGL and plasma insulin are at their fasting levels.

Transient responses to a meal disturbance were simulated using gPROMS^{TM} in order to characterise the postprandial (i.e. post-meal) glucose concentration dynamics. The first run simulated at 1000 minutes, incorporating breakfast at 8 am, lunch at 12 pm and dinner at 5 pm. The sampling period was taken every 5 minutes for a realistic interval of current continuous glucose sensors. The performance of the control law under the presence of Fisher meal disturbances of 20, 50 and 40 g of carbohydrate intake (for breakfast, lunch and dinner, respectively) on the non-linear Bergman model is shown in Figure 4.4. As shown in the figure; the BGL values remain above 60 mg/dL and below 120 mg/dL, thus avoiding both hypoglycemia and hyperglycemia conditions. These results are very much consistent with the previous studies by Finan *et al.* (2006), Dua *et al.* (2006), Markakis *et al.* (2008), Kovacs *et al.* (2008), Eren-Oruklu *et al.* (2009), Dua *et al.* (2009) and Abu-Rmileh and Gabin (2010).

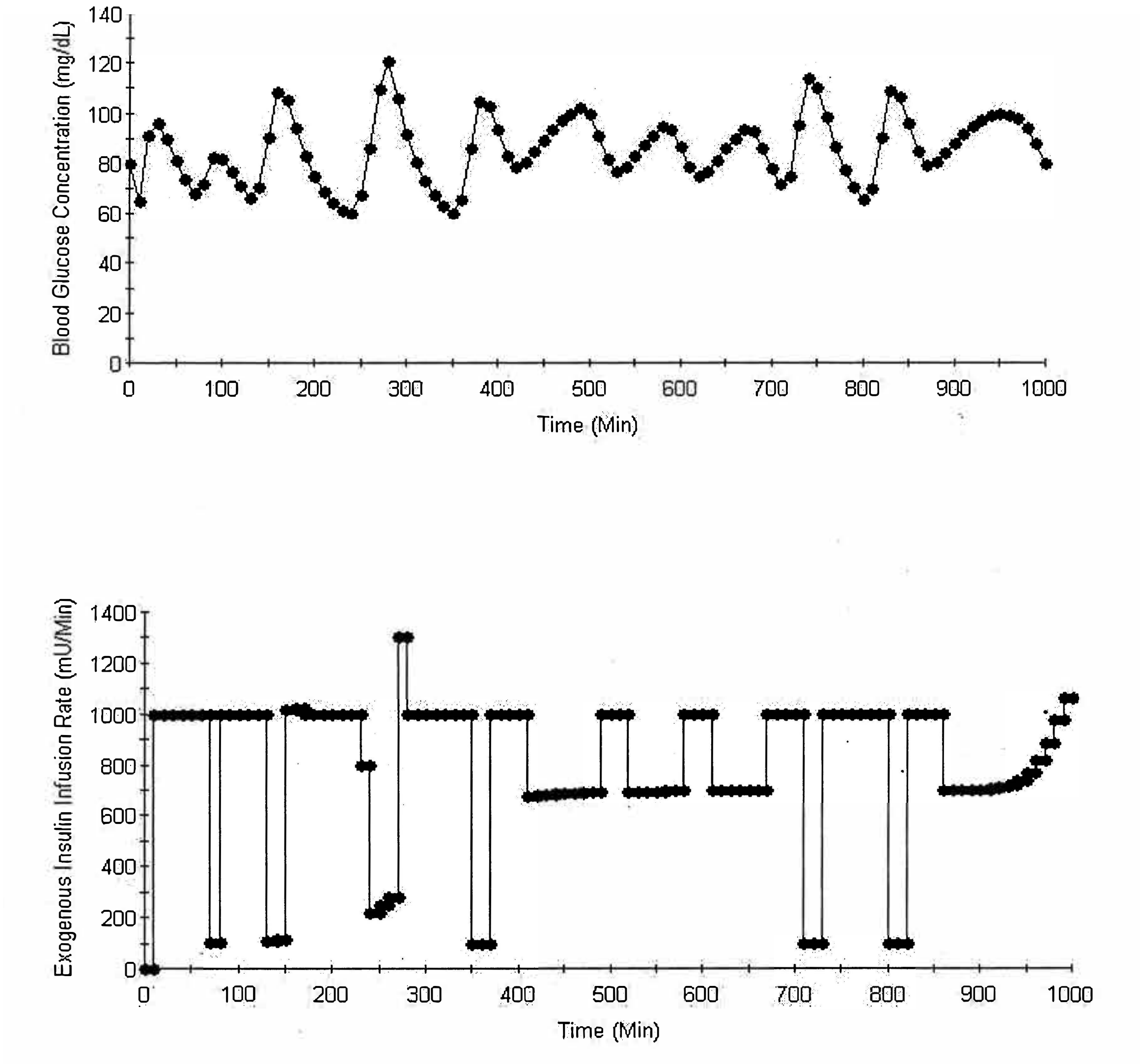


Figure 4.4 Performance of the control law under the presence of Fisher meal disturbances of 20, 50 and 40g of carbohydrate intake (1st. Trial).

Another dynamic simulation works were also carried out using gPROMS^{TM} with a slightly refined algorithm so as to optimise the performance of the control law. The subsequent run was simulated for one day period, starting at 7 am. Breakfast, lunch and dinner were administered at the same amount of carbohydrate intake and time as previously mentioned. These are shown in Figures 4.5 and 4.6. As shown in Figure 4.5, there are three peaks which obviously indicate a sudden change in the BGL values in conjunction with the introduction of meal effect into the

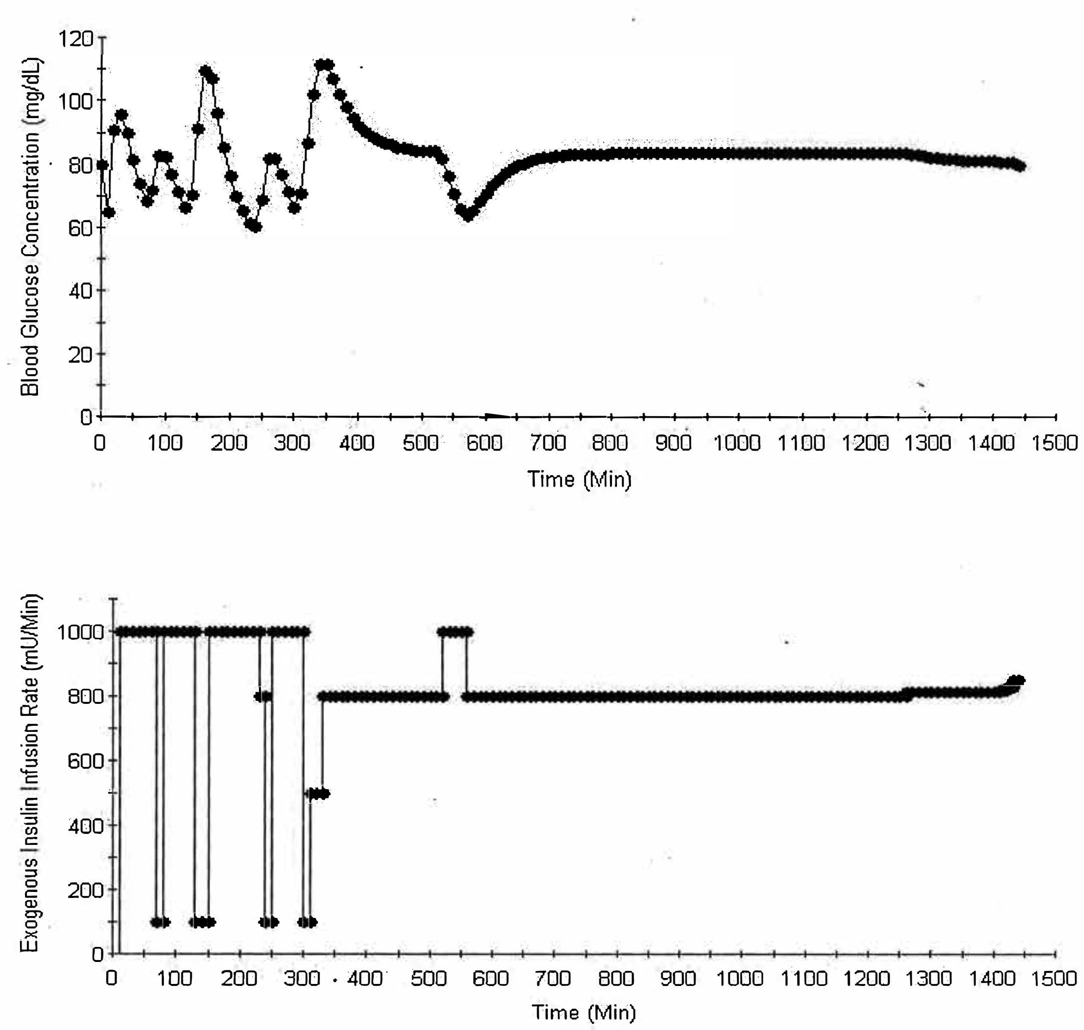


Figure 4.5 Performance of the control law under the presence of Fisher meal disturbances of 20, 50 and 40g of carbohydrate intake (2^{nd}. Trial).

glucose - insulin dynamics system. As can be seen from the graph, the BGL profile is almost the same as in Figure 4.4 before the dinner time but it tends to level off consistently at 80 mg/dL throughout the remaining period. These results coincide with the study carried out by Dua *et al.* (2006), Finan *et al.* (2006), Markakis *et al.* (2008), Eren – Oruklu *et al.* (2009) and Abu-Rmileh and Gabin (2010). The only major difference, which is very obvious, we could find in this study in comparison with the other previous works is that the value of exogenous insulin infusion rate, u is very much on the higher side. This could be due to different diabetic models were employed as well as inconsistent units were used in the parameters which, in turn, result in one decimal point deviation of the u values. It can also be concluded that the non-linear Bergman model needs to be further refined by using consistent units so as to avoid complications and difficulties if we were to incorporate these proposed algorithms into microchips for future work.

Figure 4.6 shows another dynamic simulation work carried out with different algorithm but maintaining the same procedure as previously mentioned. As can be seen from the figure, the BGL profile is almost similar from the previous algorithm but more than three peaks are occurred during the simulation. This could be due to the fact that the u values are kept initially at lower sides which subsequently result in sudden jump on the BGL values. This result, however, keeps the BGL values in normoglycemic conditions throughout the day. It consequently proves

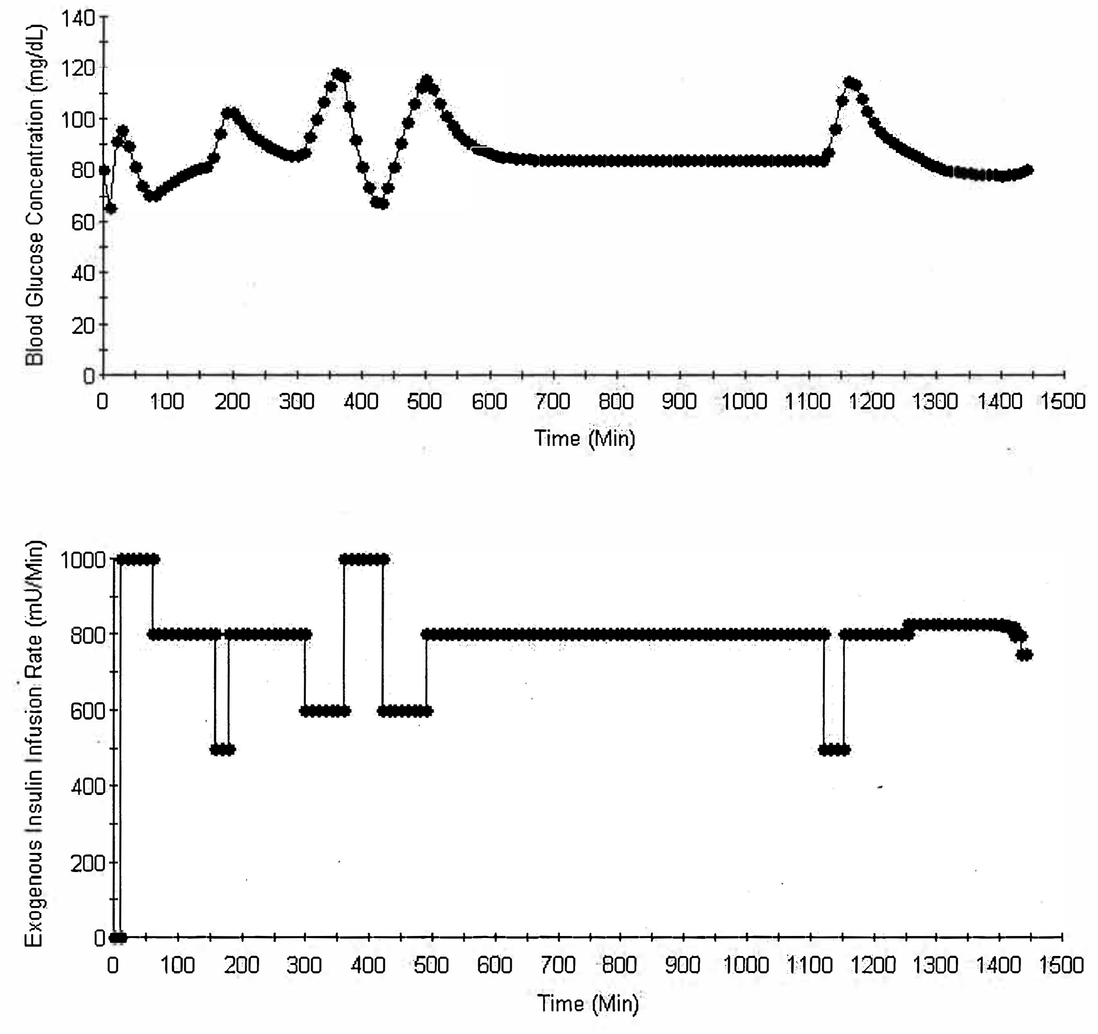


Figure 4.6 Performance of the control law under the presence of Fisher meal disturbances of 20, 50 and 40g of carbohydrate intake (3rd. Trial).

that the proposed algorithm can be easily and cheaply installed in the microchips so as to control the BGL for the use of diabetic patient in the near future.

CHAPTER 5

CONCLUSION

A new computer algorithm based on multi-parametric programmıng technique and model-based predictive control (mp-MPC) has been successfully developed to control the blood glucose level for type 1 diabetes. The non-linear Bergman minimal model is employed to represent the glucose-insulin dynamics as it takes into consideration the physiological aspect of the diabetic patient. In the development of the algorithm, both controls of blood glucose concentration with and without meal disturbances have been carried out so as to compare the results obtained with the previous studies.

The proposed algorithm has also been successfully simulated and optimised using gPROMS® to control the blood glucose level for the period of one day for the sake of simplicity. It is found that the results obtained are very consistent with the previous works. However, the only major difference we could find in this study in comparison with the other previous works is that the value of exogenous insulin infusion rate, u is on the higher side. This could be due to different diabetic models were employed as well as inconsistent units were used in the parameters which, in turn, result in one decimal point deviation of the u values. It can be concluded that the non-linear Bergman model needs to be further refined by using consistent units so as to avoid complications and difficulties in its implementation. It is also hoped that this model-based algorithm can be easily installed in the form of microchips as part of a controller in the integrated insulin delivery systems.

It is high time for a diabetic patient to be able to manage their insulin daily intake in safe and convenient manner. Thus integrated insulin delivery systems, which consist of biosensor, microchips controller and micro-pump can be bio-medically devised to replace the current injection or finger prick methods. This would require intensive researches to be carried out especially on clinical level so as to benefit the diabetic patient in near future.

Bibliography

Abu-Rmileh A, Garcia-Gabin W. (2010). Feedforward-feedback multiple predictive controllers for glucose regulation in type 1 diabetes. *Comput. Methods Programs Biomedicine.* **99**, 113 – 123.

Bergman RN, Phiilips LS, Cinobelli C. (1981). ce evaluation of factors controlling glucose tolerance in man: Measurement of insulin sensitivity and β-cell glucose sensitivity from the response to intravenous glucose. *J. Clin Invest*, **68**: 1456-1467. doi:10.1172/JCII10398.

Bogle IDL, Allen R, Sumner . (2009). The role of computer aided process engineering in physiology and clinical medicine. *Comput. Chem Eng.*doi:10.1016/j.compchemeng.2009.10.021.

Bolie VW. (1961). Coefficients of normal blood glucose regulation. *J Appl Physiol*, **16:** 783-788.

Boutayeb A, Chetouani A. (2006). A critical review of mathematical models and data useed in diabetology. *Biomedical Engineering Online*, **5**: 43. doi:10.1186/1475-925X-5-43

Brunner CS. (2004). Challenges and opportunity in emerging drug delivery technologies. Product Genesis Inc. Cambridge, USA.

Centre for Process Systems Engineering (CPSE) Annual Report 2008, UCL Media Services, London, UK.

De Gaetano A, Arino O. (2000). Mathematical modelling of the intravenous glucose tolerance test. *J Math Biol.* **40**: 136-168.

Derouich M, Boutayeb A. (2002). The effect of physical exercise on the dynamics of glucose and insulin. *Journal of Biomechanics*. **35**: 911-917.

Dua V, Bozinis NA, Pistikopoulos EN. (2002) A multi-parametric programming approach for mixed integer and quadratic engineering problems. *Comput. Chem. Eng*, **26**: 715-733. doi: 10.1016/S0098-1354(01)00797-9.

Dua P, Pistikopoulos, EN. (2005). Modelling and control of drug delivery systems. *Comput. Chem. Eng,*, **29**: 2290-2296.

Dua P, Doyle FJ III, Pistikopoulos EN. (2006). Model-based blood glucose control for type 1 diabetes via parametric programming. *IEEE Trans. Biomed. Eng.*, **53**(8): 1478-1491.

Dua P, Doyle FJ III, Pistikopoulos EN. (2009). Multi-objective blood glucose control for type 1 diabetes. *Med Biol Eng Comput*. **47**: 343-352. doi: 10.1007/s11517-009-0453-0.

Eren-Oruklu M, Cinar A, Quinn L, Smith D. (2009). Adaptive control strategy for regulation of blood glucose levels in patients with type 1 diabetes. *J. Process. Control* **19**, 1333 – 1346.

Erzen FC, Birol G, Tatara E, Cinar A. (2002). Glucosim: A web-based educational simulation package for glucose-insulin levels in the human body.
[*http://216.47.139.198/glucosim/index.html.*]

Finan DA, Zisser H, Jovanovic L, Bevier WC, Seborg, DE. (2006). Identification of linear dynamics models for type 1 diabetes: A simulation study. In *Proceedings of the International Symposium on Advanced Control of Chemical Processes (ADCHEM 2006*), pp. 503-508.

Fisher, ME. (1991). A semiclosed – loop algorithm for the control of blood glucose levels in diabetics. *IEEE Trans. Biomed Eng.*, **38** (1), 57 – 61.

Garcia CE, Prett DM, Morari M. (1989). Model predictive control: theory and practice – a survey. *Automatica.* **25**: 335-348. doi: 10.1016/0005-1098(89)90002-2.

gPROMS Introductory User's Guide Release 3.2 (2009). Process Systems Enterprise Ltd., London, UK. [http://www.psenterprise.com/gproms/index.html].

Guyton JR, Foster RO, Soeldner JS, Tan MH, Kahn CB, Koncz L, Gleason, RE. (1978). A model of glucose-insulin homeostasis in man that incorporates the heterogeneous fast pool theory of pancreatic insulin release. *Diabetes*, **27**: 1027-1042.

Hejlesen OK, Andreassen S, Hovorka R, Cavan DA. (1997). DIAS – the diabetes advisory system: An outline of the system and the evaluation results obtained so far. *Comput. Methods Programs Biomedicine*, **54:** 49-58.

Hovorka R, Canonico V, Chassin LJ, Haueter U, Massi-Benedetti M, Federici MO, Pieber TR Schaller HC, Schaupp L, Vering T, Wilinska ME. (2004). Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiol Meas*, **25**(4): 905-920

Kovacs L, Kulcsar B, Bokor J, Benyo Z. (2008). Model-based nonlinear optimal blood glucose control on type 1 diabetes patients. In *Proc. 30th IEEE EMBS Ann. Int. Conference, Vancouver, British Columbia, Canada,* pp 1607-1610.

Lehmann ED, Deutsch T. (1995). Application of computers in diabetes care – a review. II. Computers for decision support and education, *Med. Inform.* **20**: 303-329.

Li J, Kuang Y, Li B. (2000). Analysis of IVGTT glucose-insulin interaction models with time delays. *Discrete and Continuous Dynamical Systems Series B*. **1**(1): 103-124.

Markakis MG, Mitsis GD, Papavassilopoulos GP, Marmarelis VZ. (2008). Model predictive control of blood glucose in type 1 diabetes: the principal dynamic modes approach. In *Proc. 30th. IEEE EMBS Ann. Int. Conference, Vancouver, British Columbia, Canada,* , pp 5466-5469.

Montani S, Magni P, Bellazzi R, Larizza C, Roudsari AV, Carson ER. (2003). Integrating model-based decision support in a multi-modal reasoning system for managing type 1 diabetic patients. *Artificial Intelligence in Medicine*, **29**: 131-151.

Mougiakakou SG, Prountzou K, Nikita KS. (2005). A real time simulation model of glucose-insulin metabolism for type 1 diabetes patients. In *Proc. 27th IEEE EMBS Ann. Int. Conference, Shanghai, China*, pp. 298-301.

Mougiakakou SG, Prountzou A, Illiopoulou D, Nikita KS, Vazeou A, Bartsocas CS. (2006). Neural network based glucose-insulin metabolism models for children with type 1 diabetes. In *Proc. 28th IEEE EMBS Ann. Int. Conference, New York* , pp 3545-3548.

Mukhopadhyay A, De Gaetano A, Arino O. (2004). Modelling the intravenous glucose tolerance test: A global study for single-distributed-delay model. *Discrete and Continuous Dynamical Systems Series B*, **4**(2):407-417.

Palerm CCR. (2003). Drug infusion control: An extended direct model reference adaptive control strategy. In *Ph.D Thesis* chap 3-4. Rensselear Polytechnic Institute, Troy, New York: 45-60.

Parker RS, Doyle FJ III, Peppas NA. (1999). A model-based algorithm for blood glucose control in type 1 diabetic patients. *IEEE Trans. Biomed. Eng.*, **46**(2): 1-10.

Parker RS, Doyle FJ III, Ward JH, Peppas NA (2000). Robust H_∞ glucose control in diabetes using a physiological model. *AIChEJ,* **46:** 2537-2549.

Pistikopoulos EN, Dua V, Boziniz NA, Bemporad A, Morari M. (2002). On-line optimisation via off-line parametric optimisation tools. *Comput Chem Eng*. **26**: 175-185.

Pistikopoulos EN, Georgiadis, MC, Dua V. (2007). Multi–Parametric Programming. WILEY-VCH Verlag Gmbh & Co. KGaA, Weinheim. ISBN: 978-3-527-31691-5.

Pistikopoulos EN, Georgiadis, MC, Dua V. (2007). Multi–Parametric Model-Based Control. WILEY-VCH Verlag Gmbh & Co. KGaA, Weinheim. ISBN: 978-3-527-31692-2.

Sorensen JT. (1985). A physiologic model of glucose metabolism in man and its use to design and assess improved insulin therapies for diabetes. *PhD thesis*. Massachusetts Institute of Technology (MIT), Cambridge, USA.

Wilinska ME, Chassin LJ, Schaller HC, Schaupp L, Pieber TR, Hovorka R. (2005). Insulin kinetics in type 1 diabetes: Continuous and bolus delivery of rapid acting insulin. *IEEE Trans Biomed Eng*, **52**(1): 3-12.

Yasini SH, Naghibi-Sistani MB, Karimpour A. (2009). Agent-based simulation for blood glucose control in diabetic patients. *International Journal of Applied Science, Engineering and Technology*, **5**(1): 40-47.