LUND UNIVERSITY

Glycemic Control and Temperature Control in Buildings

Stemmann, Meike

2016

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Stemmann, M. (2016). *Glycemic Control and Temperature Control in Buildings*. [Doctoral Thesis (monograph), Department of Automatic Control]. Department of Automatic Control, Lund Institute of Technology, Lund University.

Total number of authors:
1

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Glycemic Control
and
Temperature Control in Buildings

Meike Rönn

LUND
UNIVERSITY

Department of Automatic Control

PhD Thesis TFRT-1114
ISBN 978-91-7623-864-6 (print)
ISBN 978-91-7623-865-3 (web)
ISSN 0280–5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2016 by Meike Rönn. All rights reserved.
Printed in Sweden by Holmbergs i Malmö AB.
Lund 2016

Till Roger och Mia

Abstract

This thesis is comprised of two parts, applying concepts from automatic control to different application areas.

The first part of this thesis concerns the development of an optimization-based algorithm, determining the size of insulin doses for patients suffering from Diabetes Mellitus and treated with multiple insulin injections.

Diabetes Mellitus is a chronic disease characterized by elevated blood glucose levels, either because of missing insulin production due to failure of the pancreatic beta-cells, or because of the body cell's reduced sensitivity to insulin. The therapy usually consists of insulin injections to substitute for the missing insulin. The amount of insulin to be administered is decided by the patient, usually using empirically developed rules of thumb.

Thus, the first part of this thesis presents an algorithm determining the dose intakes of insulin and counteracting glucose, which bring the blood glucose concentration back to a normal range. The proposed algorithm uses optimization methods and predictions of the blood glucose concentration to determine these doses. The predictions are based on individualized patient models describing the blood glucose dynamics. The cost function used in the optimization algorithm reflects the risk associated with the blood glucose values. For evaluation, the control algorithm was tested in a simulation study using a virtual patient and was compared to a simple bolus calculator from the literature. It was found that the proposed control algorithm could improve the time of the simulated patient's blood glucose in a safe range, as compared to a bolus calculator.

The second part of this thesis aims at applying inverted decoupling to the area of temperature control in buildings. Inverted decoupling is commonly used in process control and chemical engineering. Using this decoupling technique, it is possible to cancel out cross-couplings in a multi-variable system. With this, the multi-variable system can be controlled as several single input single output systems, for example using PID control.

Buildings are multi-variable systems with a variety of different interacting variables, for example the temperature dynamics of several adjacent

rooms, or variables such as air flow rate and the temperature of the air in a ventilation system. In the second part of this thesis, inverted decoupling is applied to these two examples to decrease the couplings in the dynamics. For the first example, the aim was to use the decoupling method to decrease the interactions of the temperature dynamics of adjacent rooms, in order to be able to regulate the temperature of each room without influence from another room. In the second example, the room temperature was to be regulated using the temperature of the air in a ventilation system. However, changing the air flow rate in the ventilation system, for example to regulate the CO_2 concentration, influences the room temperature as well. The aim was to use inverted decoupling to decrease this coupling, so that a change in air flow rate does not influence the room temperature. In simulation studies, the proposed decoupled controller could reduce the effect of the couplings in both examples.

Preface

After all these years, my time as a PhD student now reaches the end. It has been a very challenging, but also an exciting journey, during which I learned a lot.

When I came to Lund as an exchange student in 2006, the plan was to stay for half a year and then go back to my studies in Hamburg. But Lund changed my mind. Both the city, the student life and the courses at department of Automatic Control I read during that time. Unfortunately, after one year it was time to go back to Hamburg. Thanks to the help of Rolf Johansson, I could come back and do my master thesis project at the department. During that time, I got to know the department at all the coffee breaks and developed the wish to stay and work at the department for longer than just the master thesis. So I applied for a PhD position, and I was very happy when I got to know that I was accepted.

When I started as a PhD student in Lund, I joined the DIAdvisor project. This project was aimed at developing a prediction-based tool to help diabetic patients optimize their therapy. It was very fascinating to see that I could apply control theory to a field so different from the one I studied. After my Licentiate Thesis in 2013, the DIAdvisor project had been ending, and I wanted to get back to something a little bit closer to what I studied during my master. So I decided to switch to the application of control theory to the field of energy and buildings. It demanded some extra effort to get to know this new application area, but it has been very interesting. If I could choose an area to continue in, it would be in the area of energy and buildings.

Acknowledgements

First I would like to thank my current main supervisor Anders Rantzer for his supervision and guidance during the later part of my PhD, and for his help and valuable feedback during the first part. I would like to thank my current second supervisor Bo Bernhardsson for his helpful feedback on

my thesis. Furthermore, I would like to thank Rolf Johansson, who was my main supervisor during the first years of my PhD, for his supervision and for introducing me to graduate studies. Moreover, I would like to thank Tore Hägglund for inspiring discussions during the latest part of my PhD.

Thanks to Anders Rantzer, Tore Hägglund, Bo Bernhardsson and Carolina Lidström for taking the time to proofread this thesis. Your input was very valuable.

A big thanks also to all the administrative staff at the department. Eva Westin, Ingrid Nilsson, Mika Nishimura and Monika Rasmusson, without your commitment and your kindness the department would not be the same. Also a big thanks to Leif Andersson for his help with computer issues and for all the Latex help during the writing of this thesis. Without him, this thesis would not look as pretty as it does now. Furthermore, I would like to thank Anders Nilsson for his help with software issues.

A big thank you to the whole Department of Automatic Control here in Lund, for a very welcoming atmosphere for everyone, and a very creative and communicative environment, especially due to all the coffee breaks.

Thanks to Marzia and Fredrik, who were my DIAdvisor colleagues here in Lund, for their collaboration, support and encouragement. I would also like to thank Alessandra Parisio who made it possible for me to spend two weeks at KTH in Stockholm, Sweden, so I could try out my control algorithm on a real-life test-bed.

Thanks also to Alina and Mahdi for fun discussions and collaboration on course work. Although it has been a while ago, I would like to thank Alina for our occasional third coffee break. I would also like to thank all the colleagues I have shared an office with during all these years: Tommi, Martin K., Marzia, Martin H., Andreas and Olof.

I am also sending a big thank you to my family in Germany for always supporting and encouraging me. Ein ganz großes Dankeschön an meine Familie, die mich immer unterstützt und ermutigt hat, und immer an mich geglaubt hat.

Finally, a very big thanks to my husband Roger for his understanding, his patience and for believing in me even more than I do myself. And of course a big thanks to our daughter Mia, because she always makes me smile and is so happy when I come home from work. Now I am looking forward to new adventures, as a family.

Financial Support

Financial support for this work was provided by the Swedish Research Council through the LCCC Linnaeus Center, the ELLIIT Excellence Center and the European FP7 IST-216592 Project DIAdvisor.

Publications and Contributions

Contributions Part 1

- The amount of insulin and glucose a diabetic patient should take around mealtimes is determined using an algorithm based on optimization and using predictions of future blood glucose levels. Furthermore, patient models tailored to the individual patient's blood glucose dynamics are used.
- The blood glucose predictions and the algorithm determining the insulin and glucose dose sized are based on dynamic models describing the blood glucose dynamics of an individual patient.
- The results of this research show that, when using multiple insulin injections, a trade-off is to be made between the time which the blood glucose values lie in a safe range, and the amount of insulin taken per day. It seems that the higher the amount of time, which blood glucose values lie in a range safe for a diabetic patient, the larger the amount of insulin taken per day.
- The proposed algorithm gives the opportunity to tune this previously mentioned trade-off, i.e., to decide upon the importance of time in safe range versus amount of insulin per day. The conventional insulin therapy does not provide this opportunity.

Contributions Part 2

- Inverted decoupling, commonly used within the process industry, is applied to two examples within the control of temperature in buildings.

- Inverted decoupling provides a possibility to improve temperature regulation is multi-zone buildings with several interacting variables, by adding a decoupling network to already existing local PI controllers with supervisory control.
- In simulation studies, inverted decoupling could reduce the temperature coupling between two adjacent room, by using first-order models.
- The interaction between the air flow rate and the air temperature in a ventilation system could be reduced using inverted decoupling in simulation studies.

Publications

Note: the author of this thesis has changed name from *Meike Stemmann* to *Meike Rönn*.

This thesis is based on the following publications:

Stemmann, M. and R. Johansson (2012). "Control of type 1 diabetes via risk-minimization for multi dose injection patients". In: *5th International Conference on Advanced Technologies and Treatments for Diabetes (ATTD 2012), Barcelona, Spain*.

A control algorithm based on solving an optimization algorithm is presented, which aims at helping patients using multiple dose injection therapy to determine the doses of insulin and glucose intakes which stabilize their blood glucose concentration. As a cost function for the optimization, an asymmetrical cost function representing the risk related to a certain blood glucose value is used.
M Stemmann was the main author of this publication. She came up with the details of the control algorithm and was the main responsible for algorithm development, implementation and testing. R Johansson contributed by the conceptual idea of the algorithm and feedback both concerning algorithm development and preparation of the publication.

Stemmann, M. and R. Johansson (2012). "Diabetic blood glucose control via optimization over insulin and glucose doses". In: *8th IFAC Symposium on Biological and Medical Systems (IFAC BMS 2012), Budapest, Hungary*.

The algorithm proposed in this publication aims at helping diabetic patients using multi-dose injection therapy to determine the dose and time of insulin and glucose intakes to stabilize their blood glucose concentration. The

control was done by minimizing the risk associated with the blood glucose concentrations over the doses and times of insulin and glucose intakes.
M Stemmann was the main author of this publication. She came up with the details of the control algorithm and was the main responsible for algorithm development, implementation and testing. R Johansson contributed by the conceptual idea of the algorithm and feedback both concerning algorithm development and preparation of the publication.

Cescon, M., M. Stemmann, and R. Johansson (2012). "Impulsive predictive control of T1DM glycemia: an in-silico study". In: *5th Annual Dynamic Systems and Control Conference (ASME 2012), Ft. Lauderdale, FL, USA*.

In this publication, a basal-bolus regiment for a diabetic patient is implemented for a virtual patient. A low-order transfer function model representing the physiological patient dynamics for an individual patient was estimated for each virtual patient. Using this model, an optimization-based control algorithm was implemented, determining insulin-boluses to be administered to the virtual patient.
The estimation of the patient models was done by M Cescon. M Stemmann was responsible for the control algorithm, using the estimated models. Manuscript preparation was done by both M Cescon and M Stemmann in collaboration. R Johansson contributed by the conceptual idea of the algorithm and feedback both concerning algorithm development and preparation of the publication.

Stemmann, M. and A. Rantzer (2014). "Temperature control of two interacting rooms with decoupled PI control". In: *19th IFAC World Congress*. Cape Town, South Africa.

The room temperature of two adjacent rooms with interacting temperature dynamics were controlled using a PI controller plus a decoupling network. The idea of the decoupling network is to cancel the interactions arising from the coupled temperature dynamics, such that the temperature of each room can be regulated by its respective PI controller without being influenced by the temperature of the adjacent room.
M Stemmann was responsible for algorithm development, implementation and testing. A Rantzer contributed by feedback on both the algorithm development and the manuscript preparation, and by the conceptual idea.

The material in Part 1 of this thesis has been published in the Licentiate Thesis of the author of this thesis:

Stemmann, M. (2013). *Predictive Control of Diabetic Glycemia*. Licentiate Thesis TFRT-3258. Department of Automatic Control, Lund University, Sweden.

The author of this thesis was also involved in the following publications, but they are not part of this thesis:

Stemmann, M., F. Stahl, J. Lallemand, E. Renard, and R. Johansson (2010). "Sensor calibration models for a non-invasive blood glucose measurement sensor". In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2010), Buenos Aires, Argentina*, pp. 4979–4982.

A calibration model was developed for a noninvasive blood glucose sensor, to determine how the blood glucose data measured by this sensor is related to blood glucose data measured with laboratory capillary finger sticks and to corrupting noise. The variability of calibration models for different patients was analyzed as well as the dynamics of the non-invasive blood glucose sensor according to reference blood glucose measurements and corrupting noise.
M Stemmann was the main author of this publication. She came up with the details of the control algorithm and was the main responsible for algorithm development, implementation and testing. R Johansson contributed by the conceptual idea of the algorithm and feedback both concerning algorithm development and preparation of the publication.

Ghazaei, M., M. Stemmann, A. Robertsson, and R. Johansson (2015). "An analytic solution to fixed-time point-to-point trajectory planning". In: *2015 IEEE Conference on Control Applications (CCA)*. Sydney, Australia.

An analytic solution to the problem of fixed-time trajectory generation with a quadratic cost function under velocity and acceleration constraints was developed.
M Ghazaei came up with the idea for the problem and the concept for the solution. The solution was worked out together by M Ghazaei and M Stemmann, within a project for a course about Optimal Control. R Johansson and A Robertsson contributed by feedback concerning algorithm development and preparation of the publication.

Contents

Part I

Glycemic Control

1

Introduction

Diabetes Mellitus is a chronic disease caused either by the inability of the body to produce insulin (Type 1) or because the cells in the body do not respond to the effect of insulin (Type 2). In type 1 diabetes, a failure of the β-cells in the pancreas caused by, e.g., injuries, infections or autoimmune disorders, leads to a lack of insulin secretion. This lack of insulin secretion leads to chronically elevated blood glucose levels, which results in complications affecting, for example, the heart, liver, kidneys or nerves [Guyton and Hall, 2006].

Diabetes is one of the leading causes of blindness, amputation and kidney failure in the world [World Health Organization, 2013]. There are 331 million people with diabetes worldwide and more than 55 million in Europe. The number of people with diabetes is projected to increase from 8.3 % of the world population in 2012 to 9.9 % of the world population in 2030 [International Federation of Diabetes, 2012b]. Although $85 - 95$ % of adult people with diabetes are suffering from type 2 diabetes, the number of people with type 1 diabetes is increasing around the world each year. Furthermore, the majority of young people suffering from diabetes have type 1 diabetes [International Federation of Diabetes, 2011].

For treatment, patients with type 1 diabetes have to substitute the missing insulin by administrating insulin externally. This is done either through multiple insulin injections during the day or by continuous insulin infusion subcutaneously with a pump. Patients treated with multiple daily injections usually take basal insulin to cover the body's basal insulin needs and additional bolus insulin doses at times when the blood glucose concentration is high, e.g., around meal times. The challenge in diabetes treatment is that the patient needs to self-reliantly determine the doses of insulin required to maintain the blood glucose concentration within the normoglycemia range. This means solving an optimization problem several times throughout the day.

To help the patient with this task, blood glucose prediction algorithms as well as many different control algorithms have been proposed [Ståhl

and Johansson, 2009] [Cescon et al., 2009]. These control algorithms reach from proportional–integral–derivative (PID) control, pole placement over adaptive and run-to-run methods to model predictive control (MPC) [Cobelli et al., 2009] [Harvey et al., 2010] [Parker et al., 2001]. Many of the proposed controllers aim at having a continuous insulin signal injected into the diabetic patient via an insulin pump.

However, in the scope of the European project DIAdvisor^{TM} [Poulsen et al., 2010] [The DIAdvisor Consortium, 2012] aiming at developing a blood glucose prediction and treatment advisory system, the patient should be able to use, e.g., an insulin pen instead of a pump. Whereas continuous insulin administration can be useful for patients using an insulin pump, this is not suitable for patients using for example insulin pens to administer insulin. Instead, insulin dose advices should in such a case be in the form of impulses.

This thesis aimed at developing a control algorithm, that calculates impulse-formed insulin and glucose dose advices. These advices should not be given very frequently, but rather only a few times a day. Here the need for insulin and glucose intakes was determined in case of a meal or if the blood glucose concentration left a safe range. To determine the amounts of insulin and glucose to be taken, two different approaches were proposed. The first approach was to formulate an optimization problem, whose solution gave the amounts of insulin and glucose to be taken. The second approach was to evaluate a cost function for given sets of insulin and glucose intakes. The insulin and glucose doses resulting in the lowest cost were then chosen to be applied to the patient. It is assumed that the basal insulin need of the patient is covered.

This first part of the thesis is structured in the following way:

- Chapter 1 gives an introduction, formulates the problem for this thesis and presents the European project DIAdvisor, within which this work was done.

- Chapter 2 gives a background about the *Diabetes Mellitus* disease, how the blood glucose is regulated by the human body and how *Diabetes Mellitus* is treated. Also, an overview of research about automatic control in diabetes is given.

- Chapter 3 presents the nonlinear model used as a virtual patient to simulate a patient with type 1 diabetes.

- Chapter 4 introduces the mathematical models describing the patient dynamics used in the prediction and control algorithms, since the proposed control algorithms are model-based.

- Chapter 5 describes the optimization-based control algorithm developed in this thesis. The cost function used as well as the formulation of the optimization problem are presented. Furthermore, it is described how this optimization problem was solved and how it was embedded within an algorithm to form a controller.
- Chapter 6 describes the simulation environment, the metrics used for controller evaluation and presents the results. Moreover, a bolus calculator serving as a comparative reference for the proposed control algorithm is introduced.
- Chapter 7 discusses the results and Ch. 8 concludes the work.

1.1 The DIAdvisor Project

The work presented in this thesis was done within the European Project DIAdvisor^{TM} [Poulsen et al., 2010; The DIAdvisor Consortium, 2016]. The goal of the project was to develop a mobile short-term blood glucose predictor and treatment advisory system, that helps diabetic patients to manage their therapy, minimize the time spent outside the normal glycemic range and give them an improved quality of life. The negative effects of long-term hyperglycemia, i.e. too high blood glucose values, should be reduced without increasing the occurrence of hypoglycemia, i.e. too low blood glucose values. The objective of the project was to develop a system, which predicts blood glucose levels and gives treatment advice to the patient [Poulsen et al., 2010; The DIAdvisor Consortium, 2016].

The concept of the DIAdvisor^{TM} project is shown in Figure 1.1. The DIAdvisor^{TM} system incorporates both a blood glucose predictor, predicting the future blood glucose development, and an advisor informing the patient about corrective actions needed to best reach a blood glucose target range. It needs inputs provided by the patient, e.g., about meals and insulin intakes and from sensors such as blood glucose sensors. With the help of these inputs, the DIAdvisor^{TM} system provides a prediction of future blood glucose development and based on this calculates advices on corrective actions in terms of insulin and food intakes. The blood glucose prediction and advices are given to the patient as an instruction. The patient then has to administer the doses using, e.g., an insulin pen or pump. The patient data are also sent to the health care provider.

The DIAdvisor^{TM} system was evaluated in clinical trials at three different sites, collecting data from 50 different patients [The DIAdvisor Consortium, 2016].

Up to date patients decide upon insulin doses either using personal experience, or using rules of thumbs to calculate the correct insulin dose based

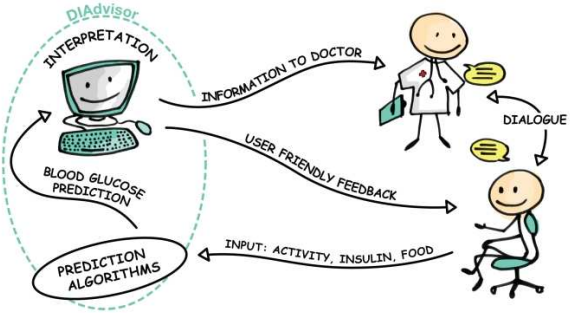

Figure 1.1 The concept of the DIAdvisor mobile short-term blood glucose predictor and treatment advisory system [The DIAdvisor Consortium, 2012].

on consumed carbohydrates and measured blood glucose values. Simple bolus calculators can assist with the calculations. By giving information about future blood glucose development and insulin intake recommendations to the patient, the DIAdvisor^{TM} system intends to empower patients into taking own decisions in their diabetes treatment [The DIAdvisor Consortium, 2016].

1.2 Problem Statement: Diabetic Impulse Control

In the sense of the DIAdvisor^{TM} project described in Sec. 1.1, the aim of this thesis was to develop a control algorithm, which gives advice about insulin injections to the patient while still allowing the patient some flexibility during the day. In this context, to keep flexibility means that the dose of insulin is determined when needed, e.g. at mealtime, and does not need to be planned for the whole coming day. To fit the DIAdvisor^{TM} system, advices should not be given in a continuous manner, but instead be in the shape of impulses, and they should be given rather seldom. Based on blood glucose predictions and measurements, the control algorithm should determine the insulin doses to be administered by the patient and the amount of glucose to be consumed with a food intake. Furthermore, the control algorithm should be based on an individual mathematical patient model describing the patient dynamics. Similar as described in Sec. 1.1, the goal was to minimize time spent outside the normal glycemic range.

2

Background

2.1 Blood Glucose Regulation in the Body

Glucose is one of the most important energy sources of the human body and is used as fuel by almost all cells, e.g., muscles, adipose tissue or the cells in the brain. A stable concentration of glucose in the blood is essential. The healthy human body has an in-built, complex feedback system to regulate the concentration of glucose in the blood and make sure it remains in balance. The main regulator of this so called glucose homeostasis is the hormone insulin, which is produced by the β-cells in the pancreas [Shrayyef and Gerich, 2010]. It stimulates the uptake of glucose by the cells of insulin-dependent tissue and the storage of glucose in the liver. In addition, there are counter regulatory hormones such as glucagon, epinephrine, cortisol or growth hormone, which work to prevent hypoglycemia, i.e., too low blood glucose concentration [Guyton and Hall, 2006] [Warrell et al., 2012].

When a patient consumes a meal, the blood glucose concentration increases, which stimulates the pancreas to secrete insulin. Insulin then promotes the utilization of the glucose by insulin-dependent body tissues, such as muscles and adipose tissue, as well as the storage of glucose in the liver and muscles as glycogen, compare to Fig. 2.1. Moreover, it inhibits the liver from producing more glucose and thus brings the blood glucose back to normal. Between meals, when the blood glucose concentration drops, the secretion of insulin from the pancreas is decreased, stimulated by the counter regulatory hormones mentioned above. If the blood glucose concentration drops too low, glucose is released into the blood by splitting glycogen from the liver back into glucose [Guyton and Hall, 2006] [Cobelli et al., 2009].

In a healthy patient, the insulin production is a process that is very closely controlled by the body, and in which the body closely monitors the glucose levels. If the blood glucose levels drop too low nevertheless, there are three mechanisms in the body that should protect from hypoglycemia, i.e., too low blood glucose level [Fowler, 2011]. When the blood glucose level drops under the normal level of 80 mg/dL, the release of insulin is decreased

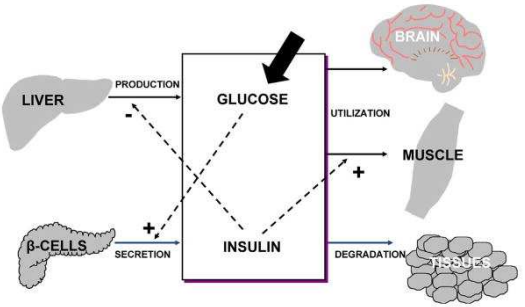

Figure 2.1 Scheme of the insulin-glucose regulating system in the body [Cobelli et al., 2009]

and kept at low levels. Moreover, the low glucose level stimulates the glucose production in liver and kidneys in order to increase the blood glucose levels. If the blood glucose level falls further, i.e., below $65 - 70$ mg/dL, the hormone glucagon is released from the alpha cells in the pancreas and also epinephrine is released. Glucagon and epinephrine both stimulate the production of glucose in the liver, while epinephrine also stimulates glucose production in the kidneys. It also induces symptoms that can be felt by the patient like increased heart rate, nervousness or anxiousness. An increased release of cortisol and growth hormone occurs, if the blood glucose level stays low for a prolonged time. If the blood glucose levels fall below 50 mg/dL, cognitive dysfunction, seizures and loss of consciousness may occur, since the delivery of glucose to the brain is insufficient [Fowler, 2011].

2.2 Diabetes Mellitus

Due to either a deficiency of insulin or the inability of the body to use insulin efficiently, the chronic disease *Diabetes Mellitus*, often only called diabetes, leads to hyperglycemia. There are several kinds of diabetes, each having different pathophysiological mechanisms. The two most common kinds are type 1 and type 2 diabetes [Williams and J.C., 2001].

The most common form is type 2 diabetes (T2DM). It mostly appears in adults, but has lately also occurred in children and adolescents [International Federation of Diabetes, 2011]. It is caused by a combination of insulin resistance and relative insulin deficiency with increased glucose production in the liver. This means, that the body can usually produce its own insulin, but either the amount is not sufficient, or the body is not responding to its effects, due to a decreased sensitivity of the insulin-dependent tissues

to insulin [Guyton and Hall, 2006] [Fowler, 2010a] [Fowler, 2010b]. Some important factors that may promote T2DM are obesity, poor diet, physical inactivity or increasing age. Not all patients require daily insulin injections, instead they are treated with a combination of diet advice, oral medications and physical activity [International Federation of Diabetes, 2011]. In later stages of T2DM, deterioration of the pancreas can occur so that daily insulin treatment is necessary.

Type 1 diabetes (T1DM) is characterized by absolute insulin deficiency. Through an autoimmune reaction, the body's own immune system attacks the insulin-producing β-cells in the pancreas and destroys them. This decreases the ability of the body to produce insulin and eventually leads to hyperglycemia. The first symptoms appear when approx. 80 % of the β-cells are destroyed. Patients suffering from T1DM need to substitute the missing insulin with externally injected insulin every day to control their blood glucose concentration. To achieve good control, they need to closely monitor their blood glucose concentration and adjust insulin intakes and diet to achieve normal blood glucose levels every day [Guyton and Hall, 2006] [Fowler, 2010a] [Fowler, 2010b].

During long-term high levels of blood glucose concentration, diabetes leads to diseases affecting the heart and blood vessels, eyes, kidneys and nerves. It is one of the main causes of cardiovascular disease, blindness and kidney failure in high-income countries [International Federation of Diabetes, 2011].

2.3 Insulin

To control hyperglycemia in diabetes, insulin has been used since the 1920s. Pioneering efforts to use insulin in diabetes treatment were made in 1922 [Banting and Best, 1922]. Initially, insulin was extracted from animal pancreatic tissue, but nowadays a recombinant DNA technique employing microorganisms is used to produce insulin [Fowler, 2008].

There are different kinds of insulin available with different on-set and duration times. For regular insulin, it takes about 30 to 60 minutes until it starts having an effect. Its peak effect is reached after $2-3$ hours and the total duration is 8 to 10 hours [Fowler, 2008]. It is a short acting insulin that is used to cover meal-time glucose excursions. Through modification and combination with additives at the molecular level, the insulin can be modified to be absorbed more quickly. Such modified insulin analogs reach the maximum peak in less time and have a higher maximum insulin concentration [Fowler, 2008].

Regular insulin can also be modified to prolong its pharmacokinetic profile. This modification is called long-acting insulin and is used to replace

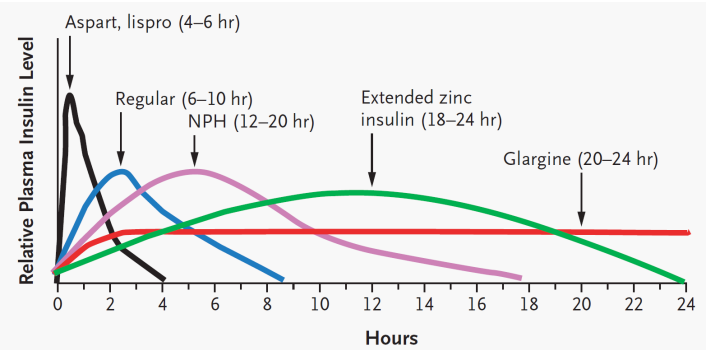

Figure 2.2 Approximate Pharmacokinetic Profiles of Human Insulin and Insulin Analogues [Hirsch, 2005]

the body's long acting insulin requirements. One such long-acting insulin is for example NPH (neutral protamine Hagedorn), with an onset action after 2 to 4 hours, a peak action after 4 to 10 hours and a duration of 10 to 16 hours [Fowler, 2008]. There are also other long-acting insulins available, which have different maximum peak or half-life. Glargine for example has a reduced peaking effect, an on-set of 1-2 hours and a duration of 20-30 hours [Heukamp et al., 2012].

2.4 Diabetes Treatment

For both T1DM and T2DM, lifestyle treatment consisting of a healthy diet and regular physical exercise is important. Because of the absolute insulin deficiency in patients with T1DM, these patients need to be treated with externally administered insulin intakes [Fowler, 2008].

The conventional therapy consists of one or two injections of intermediate and rapid-acting insulin per day, and includes self-monitoring of urine or blood glucose as well as education about diet and exercise. This therapy approach does not include daily adjustments of insulin doses [Group, 1993], which means that a strict daily schedule of meal times with little flexibility is required.

A more intensive approach [Group, 1993] consists of long-acting insulin analogs, that are administered one to two times per day, to cover the body's basal insulin requirements and rapid-acting insulin analogs to cover meal-times. The amount of rapid-acting insulin is determined according to the amount of carbohydrates in a meal, considering the measured blood glucose concentration at mealtime. The total amount of insulin injections needed

each day are larger than with the conventional approach, but the intensive approach replicates more closely the body's insulin secretion. It also allows for a greater flexibility concerning mealtimes [Fowler, 2008].

A major risk for patients treated with any kind of insulin therapy is hypoglycemia, which can be life threatening [Warrell et al., 2012]. While the intensive insulin treatment is associated with a decrease in micro vascular complications and decreased risk of kidney failure, it increases the risk for hypoglycemia [Group, 1993]. The drop of blood glucose can be induced not only by too large insulin administration, but also by increased activity or decreased appetite among others. To prevent hypoglycemia, patients are presented with the task to match the rate of insulin into the bloodstream with the rate of glucose entering the bloodstream, only by subcutaneous insulin injections. Usually patients estimate the amount of glucose in the meal and use a carbohydrate-to-insulin ratio to determine the amount of insulin that is needed to cover the meal. Furthermore, patients calculate the amount of insulin necessary to lower the blood glucose concentration using an insulin sensitivity factor. These ratios and factors are determined with the help of the health care provider. Factors like stress and physical activity can alter the amount of insulin needed. This means that these patients have to solve an optimization problem every day [Fowler, 2011].

2.5 Automatic Control for Diabetes

Many efforts have been made to develop control algorithms that help with the therapy for diabetic patients. The first closed-loop algorithms were established in the '60s and '70s and used intravenous glucose measurements and insulin infusion. In [Albisser et al., 1974] and [Kraegen et al., 1977] for example, a hyperbolic tangent function relating blood glucose concentration and rate of change to insulin infusion rate is used as a control law. The first product commercially available was the Biostator [Connor et al., 1982] [Cobelli et al., 2011].

In [Steil et al., 2006], the feasibility of using subcutaneous glucose measurements and insulin infusions for closed-loop control of diabetes was shown using PID control, which became a widely researched algorithm for control of diabetes [Steil et al., 2004] [Weinzimer et al., 2008] [Panteleon et al., 2006].

More recently, the most commonly used algorithm for closed-loop control in diabetes is the model predictive control (MPC) algorithm [Cobelli et al., 2009] [Harvey et al., 2010] [Parker et al., 2001] [Cobelli et al., 2011]. As the name suggests, it includes a model of the patient metabolic system. Since predictions of the future development of the blood glucose concentration are possible, the effect of insulin on the future blood glucose concentration

can be determined. Furthermore, constraints can be included to take into account limitations on, e.g., insulin delivery and the permitted blood glucose concentration of the patient.

MPC has in earlier years been developed for intravenous insulin infusions and glucose measurements, e.g., in [Parker et al., 1996], [Parker et al., 2000], where an MPC with constraints on the insulin infusion rate was used. In [Parker et al., 2000], glucose infusion was added as an additional controlled variable. Furthermore, an asymmetric cost function was used to take into consideration that hypoglycemia is much more dangerous than hyperglycemia.

Later, MPC was mainly used with subcutaneous glucose measurements and insulin infusions. Using an unconstrained MPC controller gave performance improvements over PID control [Magni et al., 2007] in in-silico trials. To include a prediction of the amount of insulin available in the blood over time after an intake of insulin, insulin on board was incorporated into the MPC controller [Ellingsen et al., 2009]. An approach to combine the advantages of MPC control with conventional therapy to cover disturbances like meals was presented in [Magni et al., 2011]. The conventional therapy was used as a feed-forward compensation, while the MPC controller provided feedback control to cover, e.g., meal uncertainties. A way to adjust the controller parameters on a day to day basis was done in [Magni et al., 2009] by adapting the MPC controller parameters with a run-to-run strategy.

An advisory algorithm based on fuzzy control, which determines the amounts of insulin to be injected by patients using multiple insulin doses per day instead of a pump, was developed in [Campos-Delgado et al., 2006]. Using expert knowledge about diabetes treatment, the amounts of insulin to be injected were determined on a day to day basis. Another algorithm aimed at determining insulin doses for patients using multiple daily insulin injections based on run-to-run control was developed in [Campos-Cornejo et al., 2010], which assumed a prescribed diet regime for the patient.

In [Bondia et al., 2009], basal insulin infusion rate and bolus insulin intakes were calculated determining a set of bolus intakes and basal infusion rates guaranteeing a good postprandial blood glucose response.

An approach to fit the MPC control scheme to optimize therapy for patients using multiple insulin doses per day was presented in [Kirchsteiger et al., 2009]. The control signal determined by the MPC controller was approximated by single control outputs at single time points through summation of the MPC controller's output signal. To take into account different patient dynamics for different blood glucose concentrations, gain scheduling was used. In an in-silico study it was found that approximating the insulin signal determined by the MPC by single insulin injections did not reduce the performance significantly.

3

The Virtual Patient

A simulator for the glucose-insulin system of a Type 1 diabetic patient was developed within the DIAdvisor project [The DIAdvisor Consortium, 2016; Dalla Man et al., 2007a], and used as an in-silico test-bed to test control and prediction algorithms. In this thesis, this virtual patient test-bed was used as a simulation model to test the control algorithm described in Chapter 5.

The Virtual Patient is a nonlinear model simulating the glucose-insulin system in the body. It describes the physiological events happening in the glucose-insulin system after a meal intake by using compartment modeling [Dalla Man et al., 2007a]. It is a complex model aiming at modeling the glucose-insulin dynamics in detail. Because of its complexity, it is not easily identifiable. However, in this thesis the virtual patient model was used as a test-bed to evaluate the proposed control algorithm. Models to be used by the control algorithm are presented in Chapter 4.

The virtual patient model was implemented in Matlab and Simulink using S-functions [MATLAB, 2011]. Three different sets of parameters were provided through the DIAdvisor^{TM} project, so that three different patients could be simulated.

In this chapter the in-silico model used as a test-bed is described, and a example simulation is given.

3.1 The Glucose-Insulin System

A scheme of the glucose-insulin system is shown in Fig. 3.1. This section gives an overview over the different parts the virtual patient model is made up from. For details see [Dalla Man et al., 2007b; Dalla Man et al., 2007a; Dalla Man et al., 2007a].

The ***gastro-intestinal tract*** provides the glucose rate of appearance for the glucose system after a glucose intake though a meal. It describes the transit of glucose through the stomach and the upper small intestine [Dalla Man et al., 2006]. The solid and the liquid phase of the stomach as

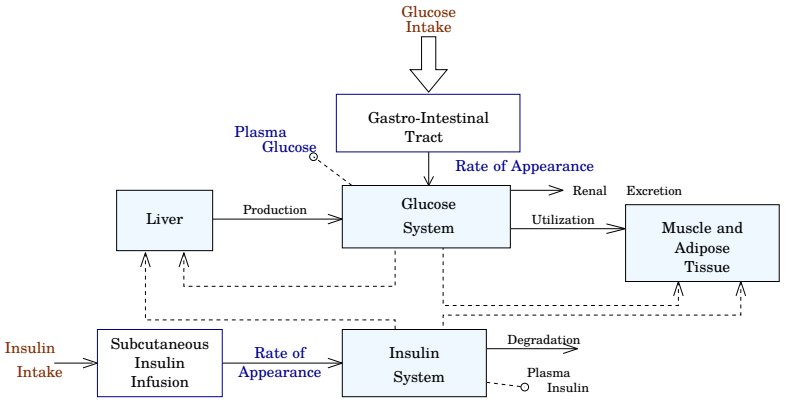

Figure 3.1 Scheme of the glucose-insulin system [Dalla Man et al., 2007b]. The continuous lines denote fluxes of insulin or glucose, the dashed lines denote insulin or glucose signals controlling the sub-parts of the virtual patient. Glucose intake and insulin intake are the inputs to the virtual patient, plasma glucose the measured output.

well as the intestine are modeled here. The model equations for the glucose absorption are shown in Ep. (3.1)

$$\begin{aligned} Q_{\text{sto}} &= Q_{\text{sto1}} + Q_{\text{sto2}} \\ \dot{Q}_{\text{sto1}}(t) &= -k_{\text{gri}} Q_{\text{sto1}}(t) + d(t) \\ \dot{Q}_{\text{sto2}}(t) &= k_{\text{gri}} Q_{\text{sto1}}(t) - k_{\text{gut}}(t, Q_{\text{sto}}) Q_{\text{sto2}} \\ \dot{Q}_{\text{gut}}(t) &= k_{\text{gut}}(t, Q_{\text{sto}}) Q_{\text{sto2}} - k_{\text{abs}} Q_{\text{gut}}(t) \\ R_a(t) &= \frac{f k_{\text{abs}}}{W_{\text{body}}} Q_{\text{gut}}(t) \end{aligned} \tag{3.1}$$

Here Q_{sto} [mg] is the amount of glucose in the stomach, where Q_{sto1} [mg] represents the solid phase and Q_{sto2} [mg] the liquid phase. The glucose mass in the intestine is denoted by Q_{gut} [mg], k_{gri} [min^{-1}] denotes the rate of grinding and k_{abs} [min^{-1}] the rate constant of intestinal absorption. The fraction of intestinal absorption that appears in the plasma is f, and d [mg/min] is the rate of ingested glucose. The body weight is denoted by W_{body} [kg] and the rate of appearance of glucose in the plasma by R_a [mg/kg/min]. The rate constant of gastric emptying $k_{\text{gut}}(t, Q_{\text{sto}})$ [min^{-1}] is a nonlinear function of Q_{sto}. For details see [Dalla Man et al., 2006]. The glucose rate of appearance R_a [mg/kg/min] was used by the control algorithm presented in Chapter 5 as one of the inputs for the prediction algorithm, which the controller uses.

The ***glucose system*** describes the insulin-independent glucose utilization in, e.g., the brain, kidneys and red blood cells, and the insulin-dependent glucose utilization in, e.g., muscle and adipose tissue. Taking into account the meal glucose intake and the glucose produced by the liver, as well as the mentioned insulin-dependent and insulin-independent utilization, the concentration of glucose in the plasma is calculated.

The ***insulin system*** models the degradation of insulin in the liver and in the plasma. Using the rate of appearance of insulin into the plasma after an insulin intake, the plasma insulin concentration and the degradation of insulin are determined.

The ***subcutaneous insulin infusion*** determines the rate of appearance of insulin in the plasma after a subcutaneous intake of insulin by the patient [Dalla Man et al., 2007a]. The model equations for the subcutaneous insulin kinetics are shown in Eq. (3.2).

$$\begin{aligned} \dot{I}_{\mathrm{sc1}}(t) &= R_{\mathrm{Inf}}(t) - (k_d + k_{a1}) I_{\mathrm{sc1}}(t) \\ \dot{I}_{\mathrm{sc2}} &= k_d I_{\mathrm{sc1}}(t) - k_{a2} I_{\mathrm{sc2}}(t) \\ R_i(t) &= k_{a1} I_{\mathrm{sc1}}(t) + k_{a2} I_{\mathrm{sc2}}(t) \end{aligned} \tag{3.2}$$

Here I_{sc1} is the amount of non-monomeric insulin in the subcutaneous space, I_{sc2} is the amount of monomeric insulin in the subcutaneous space, $R_{\mathrm{Inf}}(t)$ [pmol/kg/min] is the infusion rate of exogenous insulin, k_d [min^{-1}] is the rate constant of insulin dissociation and k_{a1} [min^{-1}] and k_{a2} [min^{-1}] are rate constants connected to non-monomeric and monomeric insulin absorption. The rate of appearance of insulin in plasma is denoted by $R_i(t)$. This plasma insulin was used by the control algorithm presented in Chapter 5 as one of the inputs for the prediction algorithm, which the controller uses.

The ***liver*** produces glucose, controlled by insulin and glucose levels in the plasma. There is a delay between the insulin plasma concentration and the effect of the insulin on the glucose production by the liver.

The glucose in the plasma is used by the ***muscle and adipose tissues***, which is as well controlled by insulin and glucose concentrations in plasma.

3.2 Virtual Patient Simulation

The glucose-insulin system shown in Fig. 3.1 has a glucose intake in [g] and a subcutaneous insulin intake in insulin units (IU) as inputs, and gives out the concentration of glucose in the blood plasma.

As an example, 10 [g] of glucose were applied to the glucose input of the virtual patient model shown in Fig. 3.1, while the insulin input was set to zero. In another example, 1 unit of insulin was applied to the insulin input of the virtual patient, while the glucose input was set to zero.

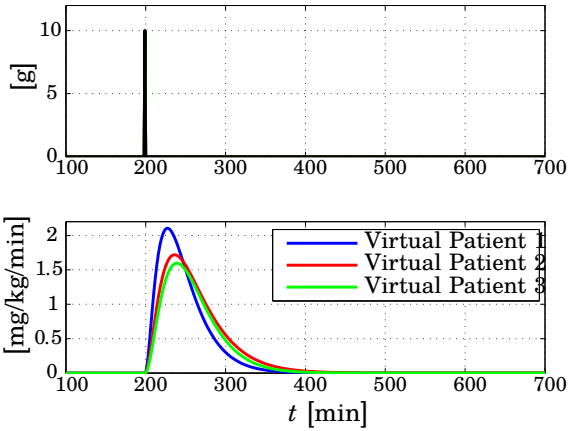

Figure 3.2 Glucose rate of appearance $R_a(t)$ (lower panel) as a response to 10 g glucose intake (upper panel). The rate of appearance is shown for three different virtual patients.

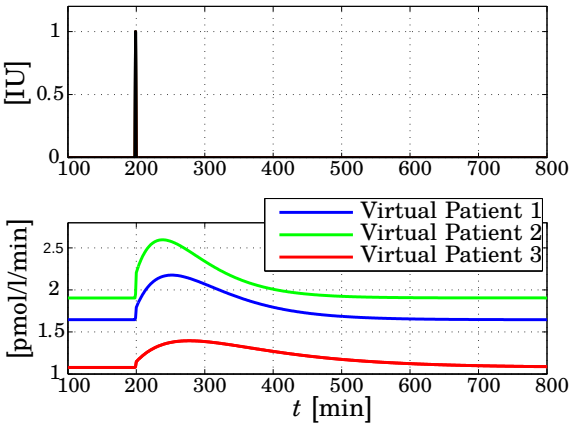

Figure 3.3 Insulin rate of appearance $R_i(t)$ (lower panel) as a response to 1 unit insulin intake (upper panel). The rate of appearance is shown for three different virtual patients.

First, the virtual patient model calculates the rates of appearances of insulin and glucose in the blood plasma. The glucose rate of appearance resulting from an intake of 10 [g] of glucose as calculated by the virtual patient is shown in Fig. 3.2. The insulin rate of appearance resulting from a subcutaneous insulin intake of 1 unit is shown in Fig. 3.3.

Then, these rates of appearance are used by the glucose and insulin system models to calculate the blood glucose concentration. The blood glucose concentrations resulting from the two examples are shown in Fig. 3.4. The upper panel shows the blood glucose concentration when there are 10 [g] of glucose intake, and no insulin intake. The lower panel shows the blood glucose concentration when there is no glucose intake, but 1 unit of insulin intake.

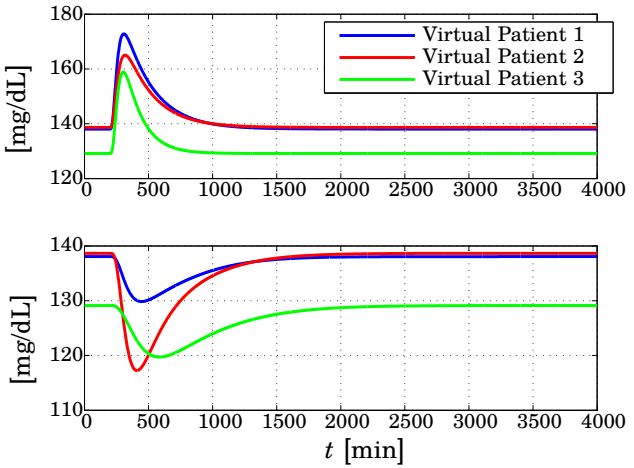

Figure 3.4 The blood glucose concentration of the three virtual patients as a response to 10 g of glucose intake (upper panel) and 1 unit of insulin (lower panel). A glucose intake increases the blood glucose concentration, while an insulin intake decreases it.

Rates of appearances and plasma glucose concentrations are shown for the three virtual patients used in this thesis. It can be seen, that 10 g of glucose increase the blood glucose concentration between ca. 20 and 30 [mg/dL]. It takes ca 16 hours, until a steady-state value is reached again. For an intake of 1 unit insulin, the blood glucose concentration decreases between 10 and 20 [mg/dL], and a steady-state is reached after ca. 25 hours. Hence, the dynamics from an insulin intake to the blood glucose concentration are slower than the dynamics from a glucose intake to the blood glucose concentration.

4

Patient Models and Predictions

The control algorithms proposed in this thesis to calculate the doses of insulin and glucose were based on a mathematical model of the patient dynamics. Moreover, they used predictions of the blood glucose concentration to take decisions about insulin and glucose intakes. Hence, models of the glucose-insulin dynamics of the diabetic patient were needed.

Two different model structures were used to describe the dynamics of the diabetic patient to be controlled. While the model in the previous chapter served as a simulator of virtual patients, the models presented here were used in the control and prediction algorithms. These models are lower in complexity than the nonlinear model used as a virtual patient. A linear state-space model was used in a prediction algorithm to predict future blood glucose concentration. For the optimization-based control algorithm, a model taking the amount of glucose and insulin and the time they are applied and giving out the change of blood glucose con concentration over time, was used. In this chapter, these two models and the prediction algorithm are presented.

4.1 State-Space Models

The linear state-space model used in the prediction algorithm and in one of the control algorithms to describe the insulin-glucose dynamics of a diabetics patient is shown in Eq. (4.1).

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ke_k \\ y_{M,k} &= Cx_k + e_k \end{aligned} \tag{4.1}$$

Here, the input u_k is a vector including the insulin rate of appearance $R_i(t)$ and glucose rate of appearance $R_a(t)$. The output $y_{M,k}$ is the blood glucose concentration in [mg/dL].

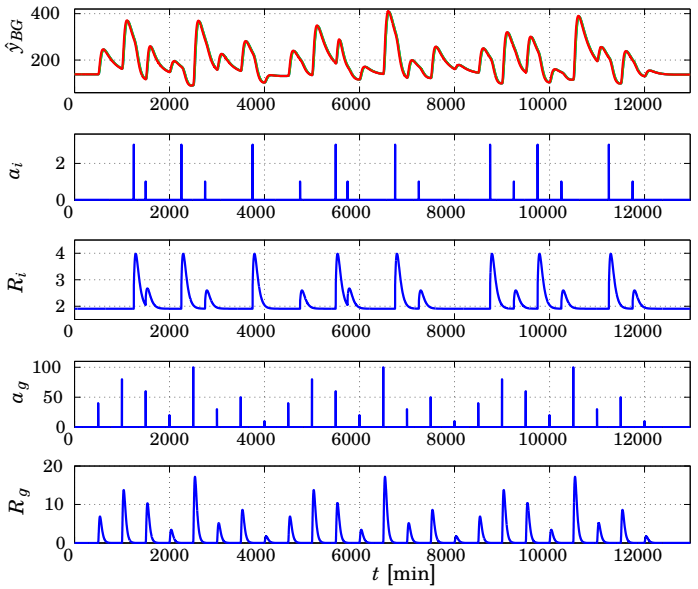

Figure 4.1 The data used for estimating the state-space model for patient 2. The first panel shows the blood glucose concentration in [mg/dL], which is the output of the estimated state-space model. The second panel shows the insulin intakes in [IU], and the third panel the insulin rate of appearance in [pmol/l/min] resulting from the insulin intakes. The fourth panel shows the glucose intakes in [g], and the fifth panel the glucose rate of appearance in [mg/kg/min] resulting from the glucose intakes. The glucose and insulin rates of appearance were the inputs for the estimated state-space model.

Since a discrete time model was to be estimated, the rates of appearance and the blood glucose concentration at the output of the virtual patient were sampled with 1 minute sampling time. For model estimation, the function `n4sid.m` from the Matlab System Identification Toolbox [Ljung and MathWorks, Inc, 2011] was used, following guidelines in [Cescon, 2011].

The estimation data consisted of 24 glucose intakes a_g [g] distributed over 9 days and 24 insulin intakes a_i [IU] for all three patients, which were placed in between the glucose intakes. The data for patient 2 is shown in Fig. 4.1. The sizes and times of the intakes were not chosen in a physiologically correct manner, but in order to get a good fit between the output of the estimated model and the output of the virtual patient. A discussion about identifying patient models using data measured from real patients can be found in, e.g., [Cescon, 2011] and [Ståhl, 2012].

For validation data, a data set of 3 days was taken, with meals three

time per day and reasonable amounts of glucose in the meals. The insulin dose was given with the meal and its size was chosen in order to cover the carbohydrate amount of the meal, according to, e.g., [Gross et al., 2003].

A state-space model was estimated using `n4sid` for model orders between 1 and 12 for the three virtual patients. The model with the model order leading to the best FIT value for a 300 steps ahead prediction for both estimation and validation data was chosen. The FIT value $\text{FIT}\ [\%] = (1 - (||y_{BG} - y_M||)/(||y_{BG} - \bar{y}_{BG}||)) \cdot 100$, where y_{BG} is the blood glucose concentration at the output of the virtual patient, $\bar{y}_{BG}$ its mean value and y_M the output of the estimated model, represents the percentage of the output variation that is explained by the estimated model [Ljung and MathWorks, Inc, 2011] and was calculated for the 300 step ahead prediction using the Matlab function `compare.m`. For the validation data, a FIT of 86%, 88% and 87% could be reached for virtual patients 1, 2 and 3, respectively.

4.2 Predictions

Since the control algorithms used predictions of the blood glucose concentration to determine the amounts of glucose and insulin to be applied to the patient, a prediction algorithm was needed. This prediction algorithm used the state-space model in Eq. (4.1) to estimate the blood glucose concentration at the output of the virtual patient for a future horizon H_p.

A scheme of the inputs and outputs for the prediction algorithm is shown in Fig. 4.2. The insulin and glucose intake doses are recalculated into their rates of appearance in plasma using Eq. (3.2) and Eq. (3.1). These are used as inputs u_k to the prediction algorithm.

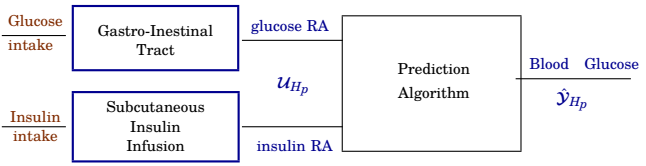

Figure 4.2 Scheme of the prediction algorithm predicting the blood glucose concentration. The prediction algorithm uses the insulin and glucose rates of appearances to calculate blood glucose predictions.

Using the rates of appearance of insulin and glucose as inputs u_k, and the state-space model in Eq. (4.1), the blood glucose concentration p time steps ahead can be calculated as shown in Eq. 4.2. The current time step is denoted by k.

$$\hat{y}_{k+p} = CA^p x_k + \sum_{m=0}^{p-1} CA^{p-1-m} B u_{k+m} \tag{4.2}$$

Collecting the predictions for $p = 1...H_p$ in one vector $\hat{\mathcal{Y}}_{H_p}$ results in Eq. (4.3), representing all blood glucose predictions for the prediction horizon H_p.

$$\hat{\mathcal{Y}}_{H_p} = \mathcal{S}_x x_k + \mathcal{S}_u \mathcal{U}_{H_p} \tag{4.3}$$

with

$$\hat{\mathcal{Y}}_{H_p} = \begin{pmatrix} \hat{y}_{k+1} & \dots & \hat{y}_{k+H_p} \end{pmatrix}^T$$
$$\mathcal{U}_{H_p} = \begin{pmatrix} u_k & \dots & u_{k+H_p} \end{pmatrix}^T$$

$$\mathcal{S}_x = \begin{pmatrix} CA & CA^2 & \dots & CA^{H_p} \end{pmatrix}^T$$
$$\mathcal{S}_u = \begin{pmatrix} CB & D & 0 & 0 & \dots & 0 \\ CAB & CB & D & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ CA^{H_p-1}B & CA^{H_p-2}B & \dots & CB & D & 0 \\ CA^{H_p}B & CA^{H_p-1}B & \dots & CAB & CB & D \end{pmatrix}$$

where here $D = 0$.

The states x_k of the patient model were estimated using a Kalman filter [Åström and Wittenmark, 1997]. The inputs u_i for $i = k...k + p - 1$ were the glucose and insulin rate of appearance calculated for the future horizon H_p. The rates of appearance were determined from the insulin and glucose intakes using Eq. (3.1) and Eq, (3.2), respectively.

4.3 The Controller Model

The optimization-based control algorithm proposed in this thesis needed a model having single insulin and glucose intakes as inputs, not their rates of appearance, and the change in blood glucose concentration as a response to those inputs over time as an output. A model in this form can be found in [Trogmann et al., 2010b] [Trogmann et al., 2010a], which was used here.

For use in a discrete optimization problem, this model was discretized with sampling time h, here $h = 1$ min. The sampling instant is denoted by k.

The model takes the doses and times of glucose or insulin intakes as inputs and has the change of blood glucose concentration over time as an output. Equation (4.4) shows this model for insulin and glucose intakes, where $y_{h,g}(k, k_g, a_g)$ is the change in blood glucose concentration as response to a glucose intake of size a_g [g] at time $t_g = h \cdot k_g$ [min] and $y_{h,i}(k, k_i, a_i)$ the change in blood glucose concentration for an insulin intake with the size a_i [IU] at time $t_i = h \cdot k_i$ [min]. The model is linear in the intake doses a_i and a_g.

$$
\begin{aligned}
y_{h,g}(k; k_g, a_g) &= \begin{cases} n_1 e^{-n_2 h(k-k_g)} h^{n_3} (k-k_g)^{n_3} a_g & \text{if } k \geq k_g \\ 0 & \text{else} \end{cases} \\
y_{h,i}(k; k_i, a_i) &= \begin{cases} n_4 e^{-n_5 h(k-k_i)} h^{n_6} (k-k_i)^{n_6} a_i & \text{if } k \geq k_i \\ 0 & \text{else} \end{cases}
\end{aligned}
\tag{4.4}
$$

The parameters n_1 to n_6 were estimated individually for the patient to be controlled. This was done using nonlinear constrained optimization [The Mathworks, 2011], solving the optimization problem (4.5). The blood glucose concentration at the output of the virtual patient is denoted by. An initial offset $c_{\text{offset},BG}$ in the blood glucose data y_{BG} was subtracted from y_{BG}. The parameters n_1 and n_4 were constrained to ensure a physiologically correct gain of the model considered.

$$
\begin{aligned}
&\min_{n_1 \ldots n_6} \sum_{k=0}^{N} \left| y_{h,g}(k; k_g, a_g) + y_{h,i}(k; k_i, a_i) - (y_{BG}(k) - c_{\text{offset},BG}) \right|^2 \\
&n_1 > 0 \\
&n_4 < 0
\end{aligned}
\tag{4.5}
$$

Figure 4.3 shows $y_{h,g}(k, k_g, a_g)$ and $y_{h,i}(k, k_i, a_i)$ with $a_g = 10$ g and $a_i = 1$ unit at times $k_g = k_i = 100$ as inputs for a sample patient. To form the total change of blood glucose concentration from an initial value when both insulin and glucose are taken at different times, those two functions were added.

Note that both $y_{h,g}(k, k_g, a_g)$ and $y_{h,i}(k, k_i, a_i)$ start at zero, since they describe the deviation of the blood glucose concentration caused by intakes of insulin or glucose.

The model presented here was used in the optimization-based control algorithm to describe the effects of intakes of insulin and glucose on the blood glucose concentration. Based on this and predictions of future blood glucose concentration, the optimization-based control algorithm calculated doses of insulin and glucose.

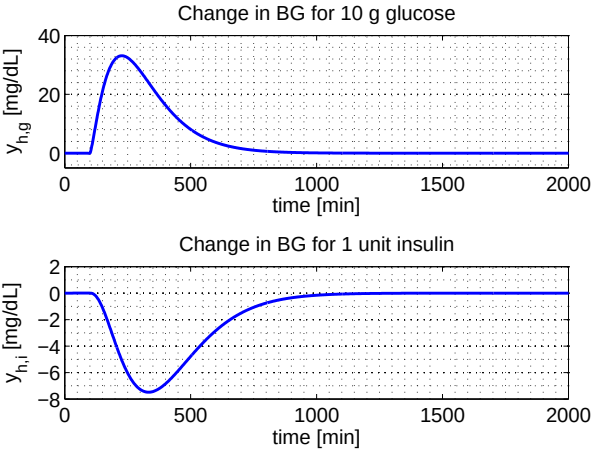

Figure 4.3 Output of the nonlinear model for patient 2. Upper panel: Change of blood glucose as response to 10 g of Glucose intake at time $t_g = 100$. Lower panel: Change of blood glucose as response to 1 unit of insulin intake at time $t_i = 100$.

5

Diabetic Glycemia Control via Optimization

The proposed control algorithm determined the amounts of insulin and glucose a patient with type 1 diabetes should take in order to bring the blood glucose concentration back to normoglycemia through solving an optimization problem. This optimization problem was solved at distinct points in time, e.g., when a meal occurs or when the blood glucose concentration left the normal range of $70 - 180$ [mg/dL]. Hence, the insulin and glucose control signals were pulse shaped.

While insulin treatment is important in most of the every day situations, additional glucose intakes can be important to prevent hyperglycemia under special conditions like exercise or stress [Guyton and Hall, 2006]. Therefore, apart from insulin dose advice, also glucose dose advices were determined by the control algorithm proposed in this chapter.

The structure of the control loop is shown in Fig. 5.1. A *virtual patient* as described in Sec. 4.3 was used to simulate the diabetic patient to be controlled. A *prediction* algorithm (see Sec. 4.2) determined the blood glucose concentration in a future horizon using information about the blood glucose concentration at the output of the virtual patient and the insulin and glucose applied to the virtual patient. The *controller* used the blood glucose predictions and the blood glucose concentration at the output of the virtual patient in an optimization algorithm to determine the amount of insulin and glucose the patient should take. The optimization problem was solved at specific points in time. When a meal occurred or when the blood glucose concentration was predicted to leave the normal range of $70 - 180$ [mg/dL], the optimization was called to determine the dose of insulin and glucose that should be administered to the virtual patient.

This chapter is organized as follows. First, the cost function used by the optimization problem is introduced. Next, the optimization problem

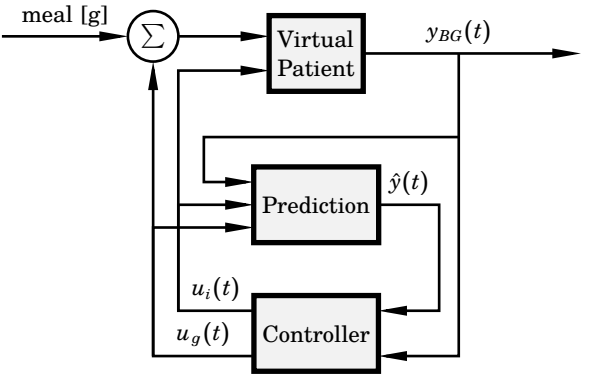

Figure 5.1 Scheme showing the structure of the control algorithm. The prediction algorithm uses the measured blood glucose concentration and past insulin and glucose intakes to predict the future blood glucose concentration. This predicted blood glucose concentration and the measured blood glucose concentration are used by the optimization-based controller to determine insulin and glucose doses to be administered to the virtual patient.

is formulated and last, the control algorithm invoking the optimization problem is described.

5.1 The Asymmetric Cost Function

The cost function used in the optimization problem had an asymmetric shape over the blood glucose concentration. The reason for this was that low blood glucose values were associated with much worse complications than high blood glucose values. Through the asymmetric shape, the cost function in Eq. (5.1) [Cameron et al., 2011] used here took this circumstance into account.

$$\begin{aligned} J &= \sum_{k=k_0}^{H_p} L(k) \\ L(y(k)) &= a \cdot y(k) + b + c \cdot \max\{(d - y(k))^3, 0\} \end{aligned} \tag{5.1}$$

The advantage of an asymmetric over a quadratic cost function for control was already addressed in [Kirchsteiger and del Re, 2009] and [Dua et al., 2009] for example.

The blood glucose concentration is denoted by $y(k)$. The time at which the optimization algorithm is called is denoted by k_0 and the prediction horizon over which the blood glucose concentration $y(k)$ is predicted is $H_p = 120$ [min] (compare to Sec. 4.2). The parameters a, b, c and d have

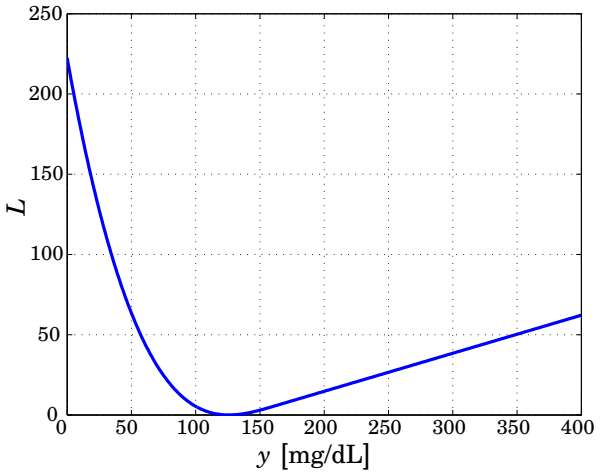

Figure 5.2 The asymmetric cost function in Eq. (5.1). This cost function takes into account, that different risks are related to high and low blood glucose values. Moreover, it is a convex cost function.

been adjusted to have a the minimum of the asymmetric cost $L(y(k))$ at 126 [mg/dL], so that $a = 0.237$, $b = -32.66$, $c = 6 \cdot 10^{-5}$ and $d = 162$. Figure 5.2 shows L over the blood glucose concentration y. One of the advantages of the cost function (5.1) over, e.g., the asymmetric cost presented in [Kovatchev et al., 2000] is that it is convex. The convexity is shown in Appendix A.

For the optimization problem convexity means that the optimal solution to the problem is guaranteed to be found [Boyd and Vandenberghe, 2004].

A technical advantage of the cost function (5.1) is that it handles negative blood glucose values [Cameron et al., 2011]. Negative blood glucose values are indeed not possible in reality. However, if the prediction algorithm calculates negative value for the predicted blood glucose concentration, the cost function should work with these negative values.

5.2 The Optimization Problem

In order to determine the amounts of insulin and glucose to be taken by the diabetic patient, an optimization problem was solved. The solution to this problem was calculated if a meal occured, if the blood glucose concentration was predicted to fall below 80 [mg/dL] or if it rose over 180 [mg/dL], see also Sec. 5.3. The optimization used the predicted blood glucose values $\hat{y}$ from Eq. (4.2) and added the effect of insulin and glucose intakes on the blood glucose concentration according to Eq. (4.4). Together, they form the

expected future blood glucose concentration as shown in Eq. (5.2):

$$y(k; a_g, a_i) = y_{h,g}(k; k_g, a_g) + y_{h,i}(k; k_i, a_i) + \hat{y}(k) \tag{5.2}$$

The effect that intakes of insulin and glucose have on the blood glucose concentration is denoted by $y_{h,i}(k; k_i, a_i)$ and $y_{h,g}(k; k_g, a_g)$, respectively. The size of the insulin intake a_i and the size of the glucose intake a_g are the optimization variables to be determined by the optimization problem. The times of insulin and glucose intake k_i and k_g were set before the optimization problem was solved. Hence, in Eq. (5.2), $y(k; a_g, a_i)$ depends linearly on the dose sizes and not on the intake times.

The optimization problem to be solved is shown in Eq. (5.3).

$$\begin{aligned} \underset{a_i, a_g}{\text{minimize}} \quad & \sum_{k=1}^{H_p} L(y(k; a_g, a_i)) + c_3|a_i| + c_4|a_g| \\ \text{subject to} \quad & 0 < a_i < c_{\text{lim,i}}, \\ & 0 < a_g < c_{\text{lim,g}}, \end{aligned} \tag{5.3}$$

where

$$\begin{aligned} L(y(k; a_g, a_i)) = a\ & (y_{h,g}(k; k_g, a_g) + y_{h,i}(k; k_i, a_i) + \hat{y}(k)) + b \\ & + c\ \max\left(d - (y_{h,g}(k; k_g, a_g) + y_{h,i}(k; k_i, a_i) + \hat{y}(k))^3, 0\right) \end{aligned}$$

It used the cost function (5.1), where the blood glucose concentration $y(k)$ was substituted by Eq. (5.2). This gives a cost function, which depends on the size of the insulin dose a_i [IU] and the size of the glucose dose a_g [g]. Since Eq. (5.2) is affine in a_i and a_g and the cost function (5.1) is convex, the composition is convex as well [Boyd and Vandenberghe, 2004]. Linear constraints were added to the optimization problem, which constrained the amount of insulin and glucose determined by the optimization. The convex cost function and the linear constraints on the optimization variables give a convex optimization problem.

Solving the Optimization Problem

To solve the optimization problem (5.3), the Optimization Toolbox from Matlab [The Mathworks, 2011] was used. This toolbox offers algorithms to solve constrained and unconstrained optimization problems. Among others, functions for linear, quadratic and nonlinear optimization are available.

The optimization problem (5.3) had a smooth and nonlinear cost function with more than one optimization variable. The constraints were bounds on the optimization variables. Therefore, the function `fmincon` was used to solve (5.3). This function solves constrained optimization problems in the form shown in Eq. (5.4).

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & f(z) \\ \text{subject to} \quad & G_i(z) = 0, i = 1, ..., m_e, \\ & G_i(z) \leq 0, i = m_e + 1, ..., m, \end{aligned} \tag{5.4}$$

The function $f(z)$ is the cost function to be minimized over the optimization variables in the vector z, where $f(z)$ can be nonlinear. The constraints $G_i(z)$ represents the constraints on the optimization problem. There are m_e equality constraints and $m - m_e$ inequality constraints, which can be nonlinear as well. Comparing this to the optimization problem (5.3), the cost function $f(z)$ was the nonlinear function J in Eq. (5.1), where $y(k)$ was substituted by $y(k; a_i, a_g)$ in Eq. (5.2). The optimization variables were the doses of insulin and glucose, i.e., $z = [a_g \ a_i]$. There were no equality constraints, but linear inequalities to limit the allowed doses for insulin and glucose.

The active-set algorithm was used here within `fmincon` to solve the optimization problem. The active-set algorithm does not need a gradient to be provided and it can take large steps, to improve simulation speed [The Mathworks, 2011]. The optimization variables, i.e. the amount of insulin and glucose, were constrained to be positive.

5.3 The Control Algorithm

The control algorithm took predictions $\hat{y}$ of the blood glucose concentration during a future horizon, as produced by the predictor (see Sec. 4.2), and the actual measured blood glucose concentration as inputs, and gave out values for the amount of insulin and glucose to be taken. Meals taken by the virtual patient were, if not stated otherwise, assumed to be unknown disturbances.

The amount of glucose and insulin to be taken was determined through the optimization problem (5.3). This was solved in the following cases:

1. A meal occured,
2. The blood glucose concentration increased over 180 [mg/dL] and was rising,
3. The blood glucose concentration was predicted to drop lower than 80 [mg/dL] within the next 15 minutes.

Note that since the blood glucose concentration increases after a meal, no optimization was performed within the two hours following a meal insulin bolus. An exception was a too low blood glucose concentration.

The amounts of insulin and glucose determined by the optimization (5.3) were applied to the virtual patient. Moreover, they were applied to the prediction algorithm. In this way, future predictions take past glucose as well as insulin intakes into account. Hence, the blood glucose predictions $\hat{y}$ included information about the effect of past insulin intakes on the future blood glucose concentration. Through the predictions, the optimization problem within the controller had access to this information and could determine new intakes accordingly.

6

Simulation Studies

In order to test and evaluate the control algorithm proposed in Chapter 5 in a closed-loop manner, the virtual patient described in Sec. 3 was used. The control algorithm, the predictor and the virtual patient were implemented in Matlab Simulink [MATLAB, 2011].

In this chapter, the simulation setup for the control algorithm is presented. Furthermore, a bolus calculator formula from the literature is described. The bolus calculator is used to compare the control algorithm proposed in this thesis.

For simulations with the optimization-based controller described in Ch. 5, the structure shown in Fig 5.1 was implemented. The virtual patient takes glucose $u_g(t)$ and insulin $u_i(t)$ as inputs and gives out the blood glucose concentration signal $y_{BG}(t)$. All these signals are used by a prediction algorithm to predict the measured blood glucose concentration over a future horizon. The predictor uses a linear model of the patient, here a virtual patient, and a Kalman filter to calculate the predictions, see Sec. 4.2. The predictions of the blood glucose concentration and the blood glucose concentration at the output of the virtual patient are used by the optimization-based control algorithm to determine the doses of insulin and glucose to be given to the virtual patient, as described in Ch. 5.

The parameters to be set within the control algorithm are the weights c_3 and c_4, which punish the norm of the insulin and glucose dose sizes in the optimization problem (5.3). Table 6.1 shows these parameters for the virtual patients. They were chosen individually for each of the patients, in order to maximize the time spent in the safe range of blood glucose values.

The algorithm was first tested without constraints on insulin and glucose intakes to test the ability of the controller to keep the blood glucose concentration in the safe range of $70 - 180$ [mg/dL]. Furthermore, the control algorithm was tested with constraints on insulin intakes to reduce the amount of glucose and insulin given to the patient per day.

Table 6.1 Parameters used for the optimization-based controller presented in chapter 5

Virtual Patient	c_3	c_4
1	0	0
2	100	10
3	60	70

6.1 The Bolus Calculator

To compare the performance of the control algorithms presented in this thesis, an insulin bolus calculation formula as described in [Shapira et al., 2010], [Gross et al., 2003] or [Wang et al., 2010] is used to determine the amount of insulin to be taken with meals. As shown in Eq. (6.1), this formula uses the current measured blood glucose concentration y_{BG} of the virtual patient, the amount of carbohydrates in the meal M[g] and a reference blood glucose concentration y_{r}, here $y_{\mathrm{r}} = 126$ [mg/dL], to determine the meal insulin bolus.

$$u_{i,\mathrm{BC}} = \frac{M}{c_{\mathrm{ITC}}} + \frac{y_{\mathrm{BG}} - y_{\mathrm{r}}}{c_{\mathrm{ISF}}} \tag{6.1}$$

The patient's insulin-to-carbohydrate ratio c_{ITC} [g/IU] and insulin sensitivity factor c_{ISF} [mg/dL/IU] can be determined using the total dose of insulin c_{TDD} a patient takes per day through $c_{\mathrm{ITC}} = 500/c_{\mathrm{TDD}}$ and $c_{\mathrm{ISF}} = 1800/c_{\mathrm{TDD}}$ [Diabetes Education Online, 2012], [BD Diabetes, 2012]. If the total dose of insulin per day for a patient is not known through previous treatment, as it is the case for the virtual patients here, it can be approximated using $c_{\mathrm{TDD}} = 0.66 \cdot W_{\mathrm{body}}$, where W_{body} is the patient's body weight in [kg] [Diabetes Education Online, 2012].

6.2 Evaluation Methods

To evaluate the glycemic control for the control algorithm presented and compare it to the bolus calculator, the low blood glucose and high blood glucose indices as presented in [Kovatchev et al., 2000; Kovatchev et al., 2005; Cobelli et al., 2009] and the percentage of time spent in the normal blood glucose range of 70 to 180 [mg/dL] $\mathrm{T}_{\mathrm{safe}}$[%] are used.

The low blood glucose index (LBGI) and high blood glucose index (HBGI) evaluate the glycemic control of a diabetic patient considering the risk for hypoglycemia and hyperglycemia, respectively [Kovatchev et al., 2000]. The LBGI gives a measure of the frequency and magnitude of low blood glucose readings, while the HBGI gives a measure of the frequency and magnitude

of high blood glucose readings [Kovatchev et al., 2005]. They are calculated based on a risk function, which emphasizes the higher risk connected to low blood glucose readings compared to high blood glucose readings. This risk function is given in Eq. (6.2) [Kovatchev et al., 2005], [Cobelli et al., 2009].

$$\begin{aligned} r(y_{\mathrm{BG}}) &= 10 \cdot f(y_{\mathrm{BG}})^2 \\ f(y_{\mathrm{BG}}) &= 1.509 \cdot ((\ln(y_{\mathrm{BG}}))^{1.084} - 5.381) \end{aligned} \tag{6.2}$$

The left branch $r_l(y_{\mathrm{BG}})$ of the risk function (6.2) is connected to the risk for hypoglycemia and its right branch $r_h(y_{\mathrm{BG},i})$ to the risk of hypoglycemia. These branches are calculated as given in Eq. (6.3) [Kovatchev et al., 2005], [Cobelli et al., 2009].

$$\begin{aligned} r_l(y_{\mathrm{BG}}) &= \begin{cases} r(y_{\mathrm{BG}}) & f(y_{\mathrm{BG}}) < 0 \\ 0 & \text{otherwise} \end{cases} \\ r_h(y_{\mathrm{BG}}) &= \begin{cases} r(y_{\mathrm{BG}}) & f(y_{\mathrm{BG}}) > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{6.3}$$

With this, the LBGI and HBGI are calculated as stated in Eq. (6.4) [Kovatchev et al., 2005], [Cobelli et al., 2009].

$$\begin{aligned} \mathrm{LBGI} &= \frac{1}{n} \sum_{i=1}^{n} r_l(y_{\mathrm{BG},i}) \\ \mathrm{HBGI} &= \frac{1}{n} \sum_{i=1}^{n} r_h(y_{\mathrm{BG},i}) \end{aligned} \tag{6.4}$$

In order to avoid both hypoglycemia and hyperglycemia, both LBGI and HBGI should be small. According to [Kovatchev et al., 2005], patients can be classified according to the clinical risk connected to their measured blood glucose concentration into three zones representing low, medium and high risk for hypoglycemia and hyperglycemia for both LBGI and HBGI, as shown in Table 6.2.

Table 6.2 Clinical risk of a a measured blood glucose concentration obtained through LBGI and HBGI, to classify the quality of the glycemic control [Kovatchev et al., 2005]. L (low), M (medium), H (high)

		HBGI		
		< 4.5	$4.5 - 9$	> 9
LBGI	< 2.5	L	L/M	L/H
	$2.5 - 5$	M/L	M	M/H
	> 5	H/L	H/M	H

6.3 Results

In this section the results of the simulation study on the three virtual patient are shown, evaluating the proposed control algorithm, and comparing it to the bolus calculator.

A summary of the results using the evaluation methods from section 6.2 is shown in the table 6.3. Each row of the tables shows the results for one of the control algorithms. The bolus calculator described in Sec. 6.1 is denoted by BC and the optimization-based controller from Ch. 5 by Opt.

Table 6.3 Result Summary for the three Virtual Patients.
BC (Bolus Calculator), Opt (Optimization); LBGI (low blood glucose index), HBGI (high blood glucose index), clin. risk (clinical risk according to Table 6.2), I_{total} (total daily insulin dose), G_{total} (total daily carbohydrate consumption, without meals)

	LBGI	**HBGI**	**clin. risk**	$T_{safe}[\%]$	I_{total} [U]	G_{total} [g]
Virtual patient 1						
BC	0	12.59	L/H	53	17.2	0
Opt	0.22	3.59	L	85	85.7	152.3
Virtual patient 2						
BC	0.19	2.66	L	87	17.6	0
Opt	0.35	1.11	L	100	34.5	136.5
Virtual patient 3						
BC	0.47	4.45	L	76	13.8	0
Opt	0.25	2.61	L	80	42.6	172.3

The goal for the HBGI and LBGI is to have them as low as possible (compare to Sec. 6.2), where the clinical risk connected to the HBGI and LBGI as shown in Table 6.2 helps for the evaluation of these two indices. Furthermore, the time $T_{safe}[\%]$ spent in safe blood glucose range and the total daily insulin and glucose dose advices are supposed to be low as well.

For all three tested virtual patients, the optimization-based control algorithm Opt presented in Ch. 5 achieved a lower LBGI and HBGI than the bolus calculator, which implies a lowered risk for low and high blood glucose values. Furthermore, the percent of time spent in the safe range $70 - 180$ [mg/dL] was higher than the bolus calculator for all tested virtual patients. This decreased risk for blood glucose values outside the safe range had to be paid for by increased doses of insulin and glucose.

For an example patient, the simulation results obtained with the bolus

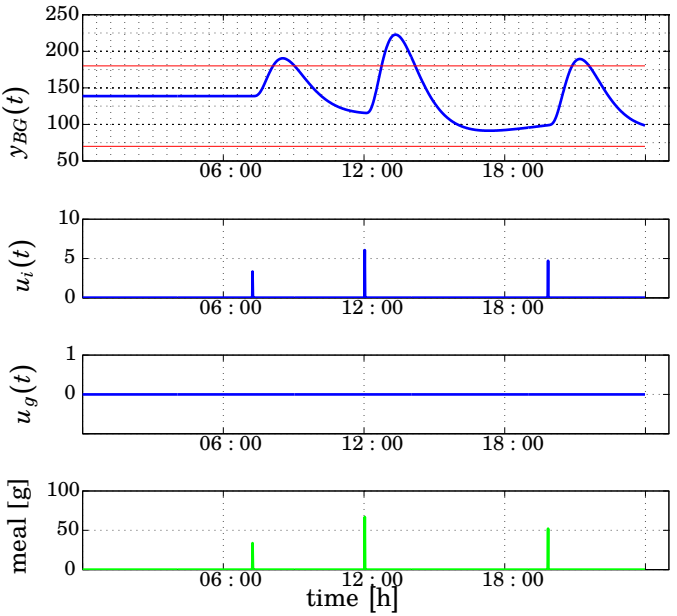

Figure 6.1 Simulation results for Virtual Patient 2 with Bolus Calculator. The bolus calcuator determined an insulin dose for each meal, bringing the blood glucose concentration back into a safe range.

calculator are shown in Fig. 6.1 and those obtained with the optimization-based controller are shown in Fig. 6.2. In this case, the amount of insulin for the optimization-based controller was not constrained. As can be seen in Fig. 6.1, the bolus calculator determined insulin doses bringing the blood glucose concentration back into safe range after a meal. However, it is conservative concerning the size of the insulin doses, leading to a shorter time in safe range than for the optimization-based controller. The results for the optimization-based controller in Fig. 6.2 show that without further restrictions of the size of the insulin and glucose doses, the controller was more aggressive than the bolus calculator. Alongside with the counteracting carbohydrates, this leads to an increase of time spent in safe range compared to the bolus calculator, but also an increase in the amount of insulin and carbohydrates taken by the patient.

Figures showing the simulation results for the other two virtual patients can be found in Appendix B.

By constraining the amounts of insulin and glucose in the optimization problem, less insulin and glucose per day could be achieved. Table 6.4 shows

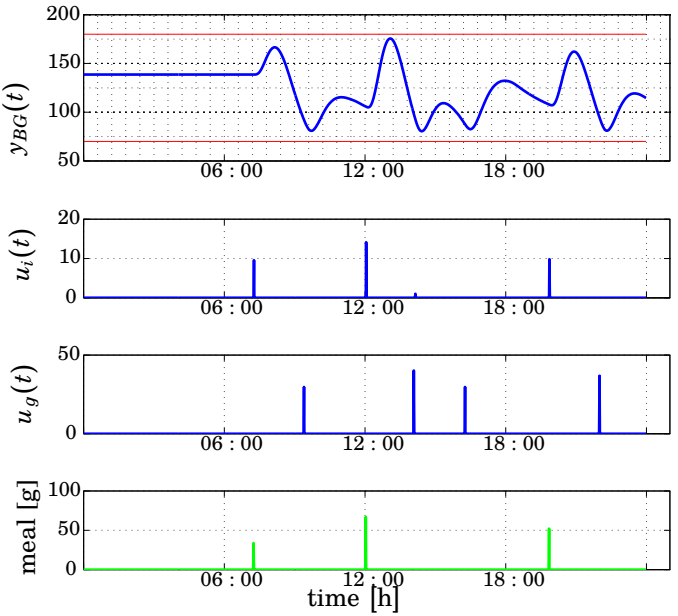

Figure 6.2 Simulation results for Virtual Patient 2 with the optimization-based controller, tuned for good control. The optimizatin-based controller determines insulin doses at meal times, as well as counteracting glucose intakes in between the meals. For this patient, the blood glucose concentration is kept in safe range for 100% of the simulated time.

results when the amount of insulin per injection were restricted to 15 units per intake for patient 1 and to 5 units per intake for patient 2 and patient 3. It can be seen that the amount of insulin and glucose per day was reduced, but the time in safe range $T_{safe}[\%]$ decreased and the HBGI increased, compare also to Fig. 6.2 and Fig. 6.3. Figure 6.2 shows the case when the amount of insulin per injection is not restricted, and Fig. 6.3 shows when it is restricted. When restricting the insulin dose, it seems from Fig. 6.3 that the behavior of the optimization-based control algorithm got closer to that of the bolus calculator. The optimization-based algorithm was still slightly more aggressive than the bolus calculator, and it had a longer residence time in the safe range. Similar results can be seen for other simulated patients, see Appendix B.

Table 6.4 Results of the optimization-based controller, tuned to give less insulin and glucose dose advices. The amount of insulin has been constrained to 15 units for patient 1, 5 units for patient 2 and 5 units for patient 3; LBGI (low blood glucose index), HBGI (high blood glucose index), clin. risk (clinical risk according to Table 6.2), I_{total} (total daily insulin dose), G_{total} (total daily carbohydrate consumption, without meals)

	LBGI	**HBGI**	**clin. risk**	T_{safe} [%]	I_{total} [U]	G_{total} [g]
Pat 1	0.28	5.36	L	78	47.85	24.68
Pat 2	0.20	2.32	L	90	15.01	0
Pat 3	0.17	4.27	L	76	15.01	16.31

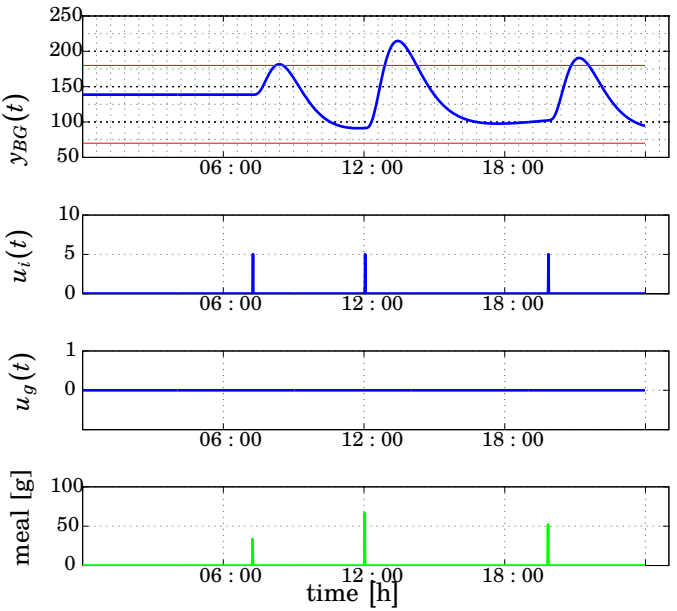

Figure 6.3 Simulation results for Virtual Patient 2 with the optimization-based controller, tuned for less insulin and glucose dose advice. The controller determines insulin intakes at meal times, bringing the blood glucose concentration back into the safe range.

7

Discussion

In order to help diabetic patients using multiple insulin intakes per day, the intention was to develop an algorithm calculating insulin and glucose dose advises using optimization methods. The goal was to spend as much time as possible to the target blood glucose range of $70-180$ [mg/dL], while avoiding hyperglycemia and, more importantly, hypoglycemia (see Sec. 1.2).

Here, a control algorithm giving insulin and glucose dosing advises at mealtimes and in case of too low blood glucose concentration has been proposed. The reason that dosing advice was determined only in the mentioned cases is that it is assumed that the basal insulin needs were covered. There have been many efforts to develop control algorithms, which determine the basal insulin needs (see Section 2.5). Here, insulin and glucose dosing advises for mealtimes and too low blood glucose values are covered instead.

For evaluation, the proposed algorithm was compared to a bolus calculator as found in e.g. [Shapira et al., 2010], [Gross et al., 2003] or [Wang et al., 2010].

As shown in the Table 6.3, the risk for low and high blood glucose values could be reduced using the optimization-based control algorithm presented in Ch. 5. On the other hand, the reduced risk came with an increase of the amount of insulin and glucose to be taken by the patient. This result is not surprising, since the objective function of the optimization problem did not include penalty on the dose sizes.

As shown in Table 6.4, the optimization-based control algorithm could be tuned to give less insulin and glucose to the diabetic patient. However, this led to a decreased control performance, i.e., less time in safe range and a higher HBGI. With the limited insulin and glucose amounts, for two out of the three patients the time in safe range was still higher than for the bolus calculator, and in none of the cases did the performance get worse than the bolus calculator.

In order to have good glycemic control, the absorption of carbohydrates into the blood should be matched by the absorption profile of insulin into the bloodstream to even out the blood glucose concentration [Pickup, 2012].

Because of delays in the insulin absorption, this is not the case when injecting insulin externally. To reduce the increase of blood glucose after a meal, larger amounts of insulin had to be given, increasing the risk for hypoglycemia. According to [Group, 1993], intensified therapy of diabetes leads to an increased risk for hypoglycemia. The results presented in Ch. 6 and also in the Appendix B show that by giving more insulin than the bolus calculator did, the optimization-based algorithm could decrease glucose excursions after meals and the time spent outside safe range. However, in order to avoid hypoglycemia large amounts of extra carbohydrates had to be given to the patient in case of larger insulin intakes. Furthermore, when restricting the amount of insulin that the optimization-based algorithm was allowed to give to the patient per injection, the amount of insulin per day and as a result also the amount of extra glucose per day could be decreased. With these results it seemed there was a trade-off to be made between the blood glucose concentration being in safe range as long as possible and restricting the amount of insulin and glucose to be taken by the patient. In the optimization-based control algorithm, adjusting parameters and constraints on insulin and glucose intakes could be used to tune the trade-off and to adjust the aggressiveness of the controller, which influences the time spent in safe range.

It can be seen in the figures shown in Ch. 6 and in Appendix B, that the optimization-based control algorithm kept away from hypoglycemia for all tested patients. This is due to the fact that the amounts of insulin and glucose doses were determined when the blood glucose concentration was predicted to drop under 80 [mg/dL] within the next 15 minutes. This allows for time to react before the blood glucose drops too low.

The approach in this thesis was to give advises for single insulin injections. Compared to a bolus calculator (see Sec. 6.1), the approaches here aimed at improving glycemic control by including more knowledge about the specific patient than the bolus calculator through an individual patient model in the control algorithm. A mathematical patient model enabled predictions of future blood glucose concentration to be made. These predictions allowed the controller to take the expected future development of the blood glucose concentration and the future expected impact of insulin and glucose intakes into account when making its decisions. This gave the chance for a higher degree of individualization than with the bolus calculator. Although estimating good patient models from real patient data remains challenging [Cescon, 2011], such a model could describe the patient dynamics, and thus how a specific patient's blood glucose concentration reacts to intakes of insulin and glucose, in greater detail than the parameters used in a simple bolus calculation algorithm.

The optimization-based controller used patient models, which had the doses of insulin and glucose as inputs. Here, the model parameters were

estimated from virtual patient data, where the input signals could be chosen with some freedom in order to provide sufficient excitation to the system to be estimated. However, this freedom is restricted when these models are to be estimated from clinical patient data. Hence, estimating a good patient model is more problematic. In [Cescon, 2011], ARMAX models with insulin and glucose rates of appearance were estimated for real patient data.

It can be seen in the figures shown in the Appendix B, that the bolus calculator in general gave smaller insulin doses than the optimization-based controller, in order to safely stay away from hypoglycemia. For patient 2 however, the last insulin dose lead to hypoglycemia. By predicting the effect of an insulin or glucose intake on the blood glucose concentration, the controller had the chance to keep the blood glucose concentration in tighter bounds. Hence, as seen in the results for the optimization-based controller, the control algorithm suggested the administration of larger insulin doses. Furthermore, the prediction of the blood glucose concentration enabled the control algorithm to suggest counteracting carbohydrates before the blood glucose concentration falls under 80 [mg/dL], so that hypoglycemia could be prevented.

Through the mathematical model and the prediction algorithm, the ongoing effect of past insulin intakes on the blood glucose concentration was already included in the control algorithm. If insulin was taken in the past, the blood glucose concentration would be predicted to decrease. This blood glucose prediction was used by the optimization algorithm to determine a new insulin dose. Using a bolus calculator, this would have to be added extra as an insulin-on-board module in order to prevent stacking of insulin intakes. Furthermore, it would be possible to include the effect of future meals on the blood glucose concentration by using a mathematical model and a prediction algorithm.

The optimization problem formulated to determine the insulin and glucose doses did not have output constraints, since it was found that introducing constraints of $70 - 180$ [mg/dL] on the output lead to infeasibility problems when solving the optimization problem. A meal would unavoidably increase the blood glucose concentration to rise over 180 [mg/dL]. A small dose of insulin taken to cover the meal would lead to a blood glucose concentration that still rises over 180 [mg/dL] within the first 1-2 hours after the meal, making the optimization algorithm infeasible. If this increased blood glucose concentration could be avoided by overdosing insulin, the blood glucose concentration would fall below 70 [mg/dL] within the prediction horizon, leading to infeasibility as well. A way around this infeasibility problem could be to introduce output constraints with dynamically adjusted size depending on meal size, within the first 2 hours after a meal. However, it was found in this thesis that good results could be achieved even without the use of output constraints.

Even though it still remained to be tested how the optimization-based control algorithm presented in this thesis performs compared to a bolus calculator in the presence of disturbances like stress or exercise, using optimization algorithms and blood glucose predictions to decide upon insulin and glucose intakes has been found to improve glucose control compared to a bolus calculator in simulation-based tests. Through the use of mathematical models, more detailed knowledge about the patient dynamics could be incorporated into the algorithm than a bolus calculator does.

Looking back at the goal formulated in Sec. 1.2, the time spent in the safe glycemic range could be increased for the optimization-based and the selection-based control algorithms proposed in this thesis, as compared to a bolus calculator algorithm found in literature, for all virtual patients tested here. While the bolus calculator reached maximum 87 % time in safe range and even as low as 53 %, the optimization-based algorithm could reach between 80 and 100 % time in safe range and the selection-based control algorithm between 82 % and 93 %, depending on the individual patient.

8

Conclusion and Future Work

In order to give advice about insulin intakes to diabetic patients, where the insulin dose advice should be a pulse-shaped signal, a mathematical optimization problem was proposed to determine insulin intake at mealtime and extra insulin and glucose intakes when the blood glucose concentration leaves the range of 70−180 [mg/dL]. The control algorithm that incorporated this optimization problem used predictions of future blood glucose concentration, which were determined by a prediction algorithm based on past and present blood glucose measurements and insulin and glucose intakes. The optimization problem as well as the prediction algorithm made use of mathematical models describing the dynamics of a diabetic patient. The control algorithm was tested in simulations using a virtual implementation of a diabetic patient.

Moreover, a bolus calculator algorithm from literature was implemented to compare the results obtained with the proposed control algorithm. It was found that using mathematical optimization, the time in the safe range of 70 − 180 [mg/dL] could be increased compared to the bolus calculator for all tested patients, but at the cost of increased amounts of insulin and glucose intakes. The proposed algorithm could avoid hypoglycemia in all cases.

The optimization-based control algorithm could be tuned such that the amounts of insulin and glucose given to the diabetic patient were reduced. This results in less time spent in safe range and thus characteristics, which approached those of a bolus calculator.

However, careful tuning of the optimization-based control algorithm had the potential to improve glycemic control as compared to the bolus calculator.

The proposed algorithms provided freedom for individualization through individualized mathematical models and controller parameters, so that the controller could be adjusted specifically to the patient to be treated, to give individualized advice.

To improve the algorithm further, insulin intakes should be allowed before meal intake, since the dynamics of appearance of insulin into the blood stream are slower than the dynamics of appearance of glucose from a meal into the blood stream. Furthermore, it should be possible to determine additional glucose intakes at a later time point within the prediction horizon and not necessarily at the same time as the insulin intake.

Another approach could be to constrain the blood glucose concentration of the patient to, e.g., 140 [mg/dL] from two hour after a meal onward. The International Federation of Diabetes recommended that the two-hour postmeal blood glucose concentration should not exceed 140 [mg/dL], as long as hypoglycemia can be avoided [International Federation of Diabetes, 2012a].

Bibliography

Albisser, A., B. Leibel, T. Ewart, Z. Davidovac, C. Botz, and W. Zingg (1974). "An artificial endocrine pancreas". *Diabetes* **23**:5, pp. 389–396.

Åström, K. and B. Wittenmark (1997). *Computer-controlled systems: theory and design*. Prentice Hall, Upper Saddle River, New Jersey.

Banting, F. G. and C. H. Best (1922). "The internal secretion of the pancreas". In: *J Lab Clin Med*. [Reprinted in Vol. 80, 1972, to mark 50th anniversary of the discovery], pp. 251–266.

BD Diabetes (2012). *How to Calculate Your Insulin Sensitivity Factor*. `http://www.bd.com/us/diabetes/page.aspx?cat=7001&id=7605`. Online, accessed November 2012.

Bondia, J., E. Dassau, H. Zisser, R. Calm, J. Vehi, L. Jovanovič, and F. Doyle III (2009). "Coordinated basal–bolus infusion for tighter postprandial glucose control in insulin pump therapy". *Journal of Diabetes Science and Technology (Online)* **3**:1, p. 89.

Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge University Press, Cambridge.

Cameron, F., B. Bequette, D. Wilson, B. Buckingham, H. Lee, and G. Niemeyer (2011). "A closed-loop artificial pancreas based on risk management". *Journal of Diabetes Science and Technology* **5**:2, p. 368.

Campos-Cornejo, F., D. Campos-Delgado, E. Dassau, H. Zisser, L. Jovanovic, and F. Doyle (2010). "Adaptive control algorithm for a rapid and slow acting insulin therapy following run-to-run methodology". In: *American Control Conference (ACC 2010)*. IEEE. Baltimore, Maryland, pp. 2009–2014.

Campos-Delgado, D., M. Hernandez-Ordonez, R. Femat, and A. Gordillo-Moscoso (2006). "Fuzzy-based controller for glucose regulation in type-1 diabetic patients by subcutaneous route". *Biomedical Engineering, IEEE Transactions on* **53**:11, pp. 2201–2210.

Cescon, M. (2011). *Linear Modeling and Prediction in Diabetes Physiology*. Licentiate Thesis 3250. Department of Automatic Control, Lund University, Lund, Sweden.

Cescon, M., F. Stahl, M. Landin-Olsson, and R. Johansson (2009). "Subspace-based model identification of diabetic blood glucose dynamics system identification". In: *15th IFAC Symposium on System Identification (SYSID 2009)*. Vol. 15. 1. Saint-Malo, France.

Cobelli, C., C. Dalla Man, G. Sparacino, G. De Nicolao, and B. Kovatchev (2009). "Diabetes: models, signals and control". *Biomedical Engineering, IEEE Reviews in* **2**, pp. 54–96.

Cobelli, C., E. Renard, and B. Kovatchev (2011). "Artificial pancreas: past, present, future". *Diabetes* **60**:11, pp. 2672–2682.

Connor, H., G. Atkin, and E. Attwood (1982). "Short-term control of brittle diabetes using a biostator". *British Medical Journal (Clinical research ed.)* **285**:6351, p. 1316.

Dalla Man, C., M. Camilleri, and C. Cobelli (2006). "A system model of oral glucose absorption: validation on gold standard data." *Biomedical Engineering, IEEE Transactions on* **53**:12.

Dalla Man, C., D. Raimondo, R. Rizza, and C. Cobelli (2007a). "Gim, simulation software of meal glucose-insulin model". *J Diabetes Sci Technol.* **1**:3, pp. 323–330.

Dalla Man, C., R. Rizza, and C. Cobelli (2007b). "Meal simulation model of glucose-insulin system". *Biomedical Engineering, IEEE Transactions on* **54**:10, pp. 1740–1749.

Diabetes Education Online (2012). *Calculating Insulin Dose*. `http://dtc.ucsf.edu/types-of-diabetes/type2/treatment-of-type-2-diabetes/medications-and-therapies/type-2-insulin-rx/calculating-insulin-dose/`. Online, accessed November 2012.

Dua, P., F. Doyle, and E. Pistikopoulos (2009). "Multi-objective blood glucose control for type 1 diabetes". In: *Medical and Biological Engineering and Computing*, pp. 343–352.

Ellingsen, C., E. Dassau, H. Zisser, B. Grosman, M. Percival, L. Jovanovič, and F. Doyle III (2009). "Safety constraints in an artificial pancreatic beta-cell: an implementation of model predictive control with insulin on board". *Journal of Diabetes Science and Technology (Online)* **3**:3, p. 536.

Fowler, M. J. (2008). "Insulin and Incretins". *Clinical Diabetes* **26** (1), pp. 35–39.

Fowler, M. J. (2010a). "Diabetes: Magnitude and Mechanisms". *Clinical Diabetes* **28** (1), pp. 42–46.

Fowler, M. J. (2010b). "Diagnosis, Classification, and Lifestyle Treatment of Diabetes". *Clinical Diabetes* **28** (2), pp. 79–86.

Fowler, M. J. (2011). "The Diabetes Treatment Trap". *Clinical Diabetes* **29** (1).

Gross, T., D. Kayne, A. King, C. Rother, and S. Juth (2003). "A bolus calculator is an effective means of controlling postprandial glycemia in patients on insulin pump therapy". *Diabetes Technology and Therapeutics* **5**:3, pp. 365–369.

Group, D. R. et al. (1993). "Diabetes control and complications trial (dcct): the effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin dependent diabetes mellitus". *N Engl J Med* **329**:14, pp. 977–86.

Guyton, A. C. and J. E. Hall (2006). "Textbook of medical physiology". In: 11th ed. Elsevier Saunders, Philadelphia, PA, USA. Chap. 78.

Harvey, R., Y. Wang, B. Grosman, M. Percival, W. Bevier, D. Finan, H. Zisser, D. Seborg, L. Jovanovic, F. Doyle, and E. Dassau (2010). "Quest for the artificial pancreas: combining technology with treatment". *Engineering in Medicine and Biology Magazine, IEEE* **29** (2), pp. 53–62.

Heukamp, I., C. Then, A. Lechner, and J. Seissler (2012). "Update on type-1 diabetes." *Internist (Berl)* **54**:2, pp. 201–216. URL: http://www.biomedsearch.com/nih/Update-diabetes/23247771.html.

Hirsch, I. B. (2005). "Insulin analogues". *New England Journal of Medicine* **352**:2, pp. 174–183.

International Federation of Diabetes (2011). *IDF Diabetes Atlas*. http://www.idf.org/diabetesatlas/. Online, accessed 2011.

International Federation of Diabetes (2012a). *Guideline for Management of Postmeal Glucose*. https://www.idf.org/guidelines/postmeal-glucose-2007. Online, accessed 2012.

International Federation of Diabetes (2012b). *IDF Diabetes Atlas, 2012 Update*. http://www.idf.org/diabetesatlas/. Online, accessed 2012.

Kirchsteiger, H. and L. del Re (2009). "Reduced hypoglycemia risk in insulin bolus therapy using asymmetric cost functions". In: *7th Asian Control Conference (ASCC 2009)*. Hong Kong, China, pp. 751–756.

Kirchsteiger, H., L. Del Re, E. Renard, and M. Mayrhofer (2009). "Robustness properties of optimal insulin bolus administrations for type 1 diabetes". In: *American Control Conference (ACC 2009)*. IEEE. St. Louis, Missouri, pp. 2284–2289.

Kovatchev, B., W. Clarke, M. Breton, K. Brayman, and A. McCall (2005). "Quantifying temporal glucose variability in diabetes via continuous glucose monitoring: mathematical methods and clinical application". *Diabetes Technology and Therapeutics* **7**:6, pp. 849–862.

Kovatchev, B., M. Straume, D. Cox, and L. Farhy (2000). "Risk analysis of blood glucose data: a quantitative approach to optimizing the control of insulin dependent diabetes". *Computational and Mathematical Methods in Medicine* **3**:1, pp. 1–10.

Kraegen, E., L. Campbell, Y. Chia, H. Meier, and L. Lazarus (1977). "Control of blood glucose in diabetics using an artificial pancreas". *Australian and New Zealand Journal of Medicine* **7**:3, pp. 280–286.

Ljung, L. and MathWorks, Inc (2011). *Matlab: system identification toolbox*. Online, assessed 2011. `http://www.mathworks.se/help/releases/R2011a/pdf_doc/ident/ident.pdf`.

Magni, L., M. Forgione, C. Toffanin, C. Dalla Man, B. Kovatchev, G. De Nicolao, and C. Cobelli (2009). "Artificial pancreas systems: run-to-run tuning of model predictive control for type 1 diabetes subjects: in silico trial". *Journal of Diabetes Science and Technology (Online)* **3**:5, p. 1091.

Magni, L., D. Raimondo, L. Bossi, C. Dalla Man, G. De Nicolao, B. Kovatchev, and C. Cobelli (2007). "Artificial pancreas: closed-loop control of glucose variability in diabetes: model predictive control of type 1 diabetes: an in silico trial". *Journal of Diabetes Science and Technology (Online)* **1**:6, p. 804.

Magni, L., C. Toffanin, C. Dalla Man, B. Kovatchev, C. Cobelli, and G. De Nicolao (2011). "Model predictive control of type 1 diabetes added to conventional therapy". In: *18th IFAC World Congress (IFAC 2011)*. Milano (Italy).

MATLAB (2011). *version 7.12.0 (R2011a)*. The MathWorks Inc., Natick, Massachusetts.

Panteleon, A., M. Loutseiko, G. Steil, and K. Rebrin (2006). "Evaluation of the effect of gain on the meal response of an automated closed-loop insulin delivery system". *Diabetes* **55**:7, pp. 1995–2000.

Parker, R., J. Doyle III, J. Harting, and N. Peppas (1996). "Model predictive control for infusion pump insulin delivery". In: *Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE*. Vol. 5. IEEE. Amsterdam, pp. 1822–1823.

Parker, R., F. I. Doyle, and N. Peppas (2001). "The intravenous route to blood glucose control". *Engineering in Medicine and Biology Magazine, IEEE* **20** (1), pp. 65–73.

Parker, R., E. Gatzke, and F. Doye III (2000). “Advanced model predictive control (MPC) for type I diabetic patient blood glucose control”. In: *Proceedings of the American Control Conference, 2000*. Vol. 5. IEEE. Chicago, IL, pp. 3483–3487.

Pickup, J. (2012). “Insulin-pump therapy for type 1 diabetes mellitus”. *New England Journal of Medicine* **366**:17, pp. 1616–1624.

Poulsen, J. U., A. Avogaro, F. Chauchard, C. Cobelli, R. Johansson, L. Nita, M. Pogose, L. del Re, E. Renard, S. Sampath, F. Saudek, M. Skillen, and J. Soendergaard (2010). “A diabetes management system empowering patients to reach optimised glucose control: from monitor to advisor”. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2010)*. Buenos Aires , Argentina, pp. 5270–5271.

Shapira, G., O. Yodfat, A. HaCohen, P. Feigin, and R. Rubin (2010). “Bolus guide: a novel insulin bolus dosing decision support tool based on selection of carbohydrate ranges”. *Journal of Diabetes Science and Technology* **4**:4, p. 893.

Shrayyef, M. and J. Gerich (2010). “Normal glucose homeostasis”. In: Poretsky, L. (Ed.). *Principles of Diabetes Mellitus*. Springer, pp. 19–35.

Ståhl, F. and R. Johansson (2009). “Diabetes mellitus modeling and short-term prediction based on blood glucose measurements”. *Mathematical Biosciences* **217**:2, pp. 101–117.

Ståhl, F. (2012). *Diabetes Mellitus Glucose Prediction by Linear and Bayesian Ensemble Modeling*. Licentiate Thesis Licencitate Thesis 3255. Department of Automatic Control, Lund University, Sweden.

Steil, G., A. Panteleon, and K. Rebrin (2004). “Closed-loop insulin delivery—the path to physiological glucose control”. *Advanced drug delivery reviews* **56**:2, pp. 125–144.

Steil, G., K. Rebrin, C. Darwin, F. Hariri, and M. Saad (2006). “Feasibility of automating insulin delivery for the treatment of type 1 diabetes”. *Diabetes* **55**:12, pp. 3344–3350.

The DIAdvisor Consortium (2012). *DIAdvisor*. `http://www.diadvisor.org`. Online, accessed November 2012.

The DIAdvisor Consortium (2016). *The DIAdvisor Consortium (2012): Final publishable summary report.* `http://cordis.europa.eu/project/rcn/85459_en.html`. Online, accessed March 2016.

The Mathworks (2011). *Matlab optimization toolbox user’s guide*.

Trogmann, H., H. Kirchsteiger, G. Castillo Estrada, and L. del Re (2010a). “Fast estimation of meal and insulin bolus effects in T1DM for in silico testing using hybrid approximation of physiological meal and insulin model”. *Diabetes* **59**, A136.

Trogmann, H., H. Kirchsteiger, and H. del Re (2010b). “Hybrid control of type 1 diabetes bolus therapy”. In: *49th IEEE Conference on Decision and Control (CDC 2010)*. Atlanta, Georgia, pp. 4721–4726.

Wang, Y., E. Dassau, H. Zisser, L. Jovanovič, and F. Doyle III (2010). “Automatic bolus and adaptive basal algorithm for the artificial pancreatic beta-cell”. *Diabetes Technology and Therapeutics* **12**:11, pp. 879–887.

Warrell, D. A., T. M. Cox, and J. D. Firth (2012). “Diabetes”. In: *Oxford Textbook of Medicine*. Oxford University Press, New York, NY, oxfordmedicine . com / 10 . 1093 / med / 9780199204854 . 001 . 1 / med - 9780199204854. Chap. 13.11.1.

Weinzimer, S., G. Steil, K. Swan, J. Dziura, N. Kurtz, and W. Tamborlane (2008). “Fully automated closed-loop insulin delivery versus semiautomated hybrid control in pediatric patients with type 1 diabetes using an artificial pancreas”. *Diabetes Care* **31**:5, pp. 934–939.

Williams, G. and P. J.C. (2001). *Handbook of Diabetes*. 2nd ed. Blackwell Science, Maldan, MA.

World Health Organization (2013). *Diabetes Programme*. http://www.who.int/diabetes/en/. Online, accessed January 2013.

Part II

Applications of Decoupled PID Control to Temperature Control in Housing

9

Introduction

In recent years, the use of control algorithms in building management systems has been an active research area. Many different algorithms to regulate the temperature, among others, in buildings have been proposed [Ma et al., 2012; Hazyuk et al., 2012].

Already 40 years ago, the use of control algorithms to regulate the climate in buildings was of interest. According to [Jensen, 1978], the reasons were at that time that buildings were becoming more lightweight, and the energy price was increasing.

Today the energy question is still of concern, many researchers in this area address this question in their research. Advancements in the building sector and EU regulations have led to new houses being built very energy-efficiently with very good insulation. However, many old buildings with poorly insulated walls are still left. Poor insulation of inner walls in a building leads to a coupling of temperature dynamics between different rooms of the building, i.e., the temperature will be influenced not only by the outside conditions, but also by the temperature of adjacent rooms. This leads to couplings between the temperature dynamics of the rooms in a building. Hence, the temperature control of the whole building is a multi-variable control problem.

To regulate the temperature and other variables concerning the climate in a building, HVAC (heating ventilation air conditioning) systems are usually used. These add coupled variables to the system to be controlled, since many of the properties regulated by an HVAC system influence each other. For example, if the CO_2 level in the room needs to be decreased, the HVAC system can increase the flow of fresh air from the outside into the room. However, especially in the winter season in a northern country like Sweden, this can cause the room temperature to decrease and thus deviate from its desired value.

To regulate such coupled building systems, a method very popular in the research community is MPC (model predictive control). This algorithm is often chosen as the most suitable method to regulate occupant comfort as

well as energy usage, since it can consider predictions of disturbances such as weather, occupancy or outside temperature. On the other hand, MPC requires a complex and accurate model of the building to be regulated and accurate disturbance predictions to perform well. Furthermore, it requires the implementation of an optimization algorithm, and thus a solver for this optimization problem. Therefore, a large cost is required for an initial setup in a building, which might be worthwhile for new large-scale commercial buildings, but not necessarily for smaller buildings.

Another control method widely used for temperature control in buildings and for building components is PID control [Salsbury, 2005a; Dounis and Caraiscos, 2009]. This is also still the most used control method in industry. In building management systems, PID controllers are used for low-level controllers for building or HVAC system components. Another example of the use of PID controllers are thermostatic valves on radiators [Peffer et al., 2011]. With this, the temperature of adjacent rooms are often controlled locally, without the knowledge of the temperature in other rooms.

To coordinate all the different variables to be controlled using the sensors and actuators of an HVAC system, radiators or other systems, supervisory control is often implemented. Even though MPC would be beneficial for this task, because of the above mentioned implementation issues for MPC, still rule-based supervisory controllers are often used. These can get very complex with all the couplings present in larger buildings equipped with an HVAC system.

Decreasing the interactions between the different variables of the multi-variable building system might help to reduce the complexity of the supervisory systems. Alternatively, this decoupling could be an addition to the low-level PID controllers operated by an MPC supervisory controller.

The idea in this thesis is thus to build on to already existing methods, by adding a decoupling network [Gagnon et al., 1998] to the PI control of building variables such as the room temperature or the CO2 level. The purpose of this decoupling network is to reduce the interaction between different variables. Two examples of coupled variables are presented in this thesis.

One example is the control of room temperature for adjacent rooms. The assumption is that the walls connecting the rooms are badly insulated, so that a change in temperature in one room influences the temperature in the neighboring rooms. By the use of inverted decoupling [Gagnon et al., 1998], the thermal coupling between the rooms is to be decreased. A model of four adjacent rooms arranged in a square was implemented in Modelica language using the Modelica Building Library [Wetter et al., 2014]. This model was used as a simulation model to test the decoupled PI controller.

The second example is the control of the room temperature, when the flow rate of outside air into the room is changing, e.g., to control the CO2

level in the room. For this, it is assumed that the outside temperature is less than the desired room temperature. Here, the change in air flow rate influences the room temperature to deviate from its desired value. The coupling is thus only partial, since the CO2 level control influences the room temperature, but not vice versa. To reduce the coupling between the two variables, again a decoupling network is connected to the PI controllers regulating the room temperature. The decoupled PI controller was tested in simulations, using a dynamic model of the room's temperature dynamics. Moreover, tests were performed in a laboratory room at KTH in Stockholm.

This part of the thesis is organized as follows. First, background information is presented, including an overview of temperature control in buildings and an introduction to the used decoupling method.

Then, the application of decoupled PI control to temperature regulation in adjacent rooms is presented. For this, the implementation of a simulation model in the Modelica language is explained, as well as estimation of simpler models used for control, the controller implementation and the simulation results.

After this, the use of a decoupled PI controller for a ventilation system is presented. This includes the description of the test-bed at KTH in Stockholm as well as the dynamic model used for simulation tests. Also, the estimation of simple models for control is presented, and the implementation and testing using the simulation model as well as the test-bed. The simulation results and the results from the test-bed experiments are presented in the end of that section.

At last, the results in this part are discussed and a conclusion is drawn.

10

Background

10.1 Temperature Control in Housing

The first digital building automation system was set up in the 60s, which included communication of sensors and devices, and a schedule to control these devices. According to [Wang, 2010], building automation had been done using mechanical and pneumatic equipment before that. When the energy prices increased during the 70s, buildings became more lightweight and with upcoming digital computers, the interest in the use of digital controllers for building automation increased as well [Wong and So, 1997; Jensen, 1976]. The digital controller had become very widespread in the 80s for building automation, and building management systems were programmed using high-level languages [Wong and So, 1997], which made it possible for more advanced control algorithms being used in building automation.

Since then, a variety of different control algorithms have been investigated by the research community, ranging from PID over Fuzzy Control and optimal control to MPC and algorithms based on neural networks and genetic algorithms [Dounis and Caraiscos, 2009; Afram and Janabi-Sharifi, 2014].

MPC has often been seen by the research community as a suitable method for building automation, since it can take into account weather predictions and occupant schedules [Hazyuk et al., 2012]. Hence, this is one of the most popular methods today for building automation, and many papers have been published around this topic. An MPC controlling a building and its HVAC (heating ventilation air conditioning) system on both high-level and low-level control is presented in [Ma et al., 2012]. In [Hazyuk et al., 2012], an MPC is developed, which minimizes the energy usage and places the comfort requirements into the constraints instead of trying to follow a target temperature. A deterministic MPC is compared to a stochastic MPC in [Oldewurtel et al., 2012], finding that a stochastic approach better captures the non-deterministic nature of the disturbances occurring in a

building. In [Parisio et al., 2014b; Parisio et al., 2014a], a stochastic MPC was used to control the HVAC system of a student lab, where the algorithm dynamically learns the weather conditions and the occupant patterns. In these papers, the building is represented as one zone. When the building size increases, it can however be of advantage to model the building dynamics with more than one zone. Then, a centralized MPC can become quite computationally expensive [Ma et al., 2012]. Therefore, the authors of [Moroşan et al., 2010] developed a distributed MPC to control a building consisting of multiple zones, taking into account thermal coupling between the zones while avoiding a centralized solution. In [Lamoudi et al., 2012], a non-linear distributed MPC was developed to control indoor temperature, CO_2 concentration and illumination in a multi-zone building.

However, setting up an MPC to automate a building requires an accurate model describing the building dynamics. Especially in older buildings, additional equipment needs to be installed, leading to significant initial costs to the set-up of MPC [Ma et al., 2012].

Therefore, researchers have investigated other methods for building automation. Another popular algorithm for building automation is Fuzzy Control. A Fuzzy Controller combined with PID designed to fulfill comfort requirements set by the user while giving priority to passive cooling techniques is presented in [Kolokotsa et al., 2005]. In [Paris et al., 2010], combinations of PID and MPC and of PID and Fuzzy Control are used to control a building with multiple energy sources.

In spite of all the research done towards advanced control algorithms for building automation, the most widespread algorithm used in the building automation industry is PID [Salsbury, 2005b]. There are several reasons why the building industry is slow to adopt the new methods popular in the research community [Salsbury, 2005b; Holmberg, 2001]. One is that the low-cost hardware typically used by the building industry cannot run methods that are too computationally demanding. Another reason is that methods, which increase the set-up time of a building management system are seen as being impractical.

Therefore, local PID controllers with a basic logic interconnecting the controllers are still used in the majority of the buildings [Ma et al., 2012]. On the low-level, cascaded PID controllers are designed independently of each other. On the higher level of control, schedules are determining setpoints for the building components [Ma et al., 2012]. In [Doukas et al., 2007] for example, a rule-based decision support system for a building energy management system is developed. There are rules for both energy efficiency and for indoor comfort. [Mathews et al., 2000] developed a rule-based algorithm to improve the CO_2 level as well as the room temperature. [Zhou et al., 2014] presented a demand-based supervisory control of a large-scale room, which was divided into 4 zones, where each zone was equipped

with a VAV (variable air volume) box. Classical control algorithms such as PID control or on/off controllers were then used for building components, such as supply air temperature, supply air pressure, heater control, VAV unit control and various valves on an HVAC system [Afram and Janabi-Sharifi, 2014].

For larger buildings, these rule-based algorithms interconnecting several cascaded PID controllers can quickly become confusing, so that it quickly evolves into a complex task to overview all interactions happening between different control loops.

10.2 Decoupling PID control

A building is a multi-variable system, especially when considering several zones and when it is equipped with an HVAC system. There are interdependencies between many different variables. Particularly for a building with poorly insulated walls considered as a multi-zone building, there are couplings between the temperature dynamics of those different zones. Furthermore, coupling occurs between different variables in an HVAC system, where each of the variables might be regulated by a PID controller.

As depicted in Section 10.1, PID control with rule-based supervisory control is still used for most of the buildings. A way to decrease interactions between variables, while keeping PID controllers is to used a decoupling controller with the PID controllers.

Decoupling methods are quite well-known in the area of distillation, but they are not common in other fields such as process control [Waller et al., 2003], much less in the building automation area. Decoupling controllers are feed-forward controllers having manipulated variables as inputs, which are designed to reduce interaction between different control loops [Seborg et al., 2010]. There are static decoupling methods [Seborg et al., 2010; Astrom and Hagglund, 2006], and dynamical decoupling methods [Liu et al., 2007; Shen et al., 2010; Gagnon et al., 1998; Garrido et al., 2011]. Static decoupling methods decouple only static interactions. They need very little process information, but couplings are still present during transient conditions. In this thesis, the interest is to achieve decoupling during transient conditions, e.g., when the temperature of a neighboring room is changed. Hence, static decoupling methods are not applicable here, and dynamics methods should be used.

Using decoupling methods, the stability of the closed-loop system can be reached by stabilizing the single loops [Seborg et al., 2010]. Moreover, a change in one variable does not affect otherwise coupled variables. On the other hand, the performance of the decoupling control depends strongly on the the accuracy of the underlying models describing the system to be

decoupled. Also, it needs to be ensured that the transfer functions of the decoupling controller are realizable.

A popular decoupling method is simplified decoupling [Gagnon et al., 1998]. For this method, the transfer functions making up the decoupler are quite simple. However, the apparent process becomes a combination of the elements of the process transfer matrix, which makes controller tuning difficult. A less commonly used method is ideal decoupling. Using this decoupling method, the apparent process consists of the diagonal elements of the process transfer function, which facilitates controller design. On the other hand, the elements in the decoupling transfer matrix become more complex than for simplified decoupling.

In this thesis, a third possible method, namely inverted decoupling, is used [Gagnon et al., 1998; Garrido et al., 2011]. This method combines the advantages of simplified decoupling and ideal decoupling. The decoupling transfer matrix is as simple as for simplified decoupling and the apparent process consists of the diagonal elements of the process, as for ideal decoupling. Hence, it is easier to take into account saturation of manipulated variables than ideal and simplified decoupling, and an anti-windup strategy can be implemented in the same way as for SISO processes [Gagnon et al., 1998].

In [Gagnon et al., 1998] it was found that robust performance and robust stability are similar for simplified, ideal and inverted decoupling. Hence this does not play a role when choosing the method. An extension of inverted decoupling to systems with multiple time delays and non-minimum-phase zeros is presented in [Chen and Zhang, 2007]. In [Garrido et al., 2011], an extension to $n \times n$ processes as well as different configurations of inverted decoupling to ensure realizability are presented.

Inverted Decoupling for 2×2 Processes

A block diagram of a 2×2 process with inverted decoupling is shown in Fig. 10.1. The decoupling network in Fig 10.1 is shown in Eq. (10.1).

$$\begin{aligned} u_1(s) &= -\frac{G_{12}(s)}{G_{11}(s)} u_2(s) + c_1(s) \\ u_2(s) &= -\frac{G_{21}(s)}{G_{22}(s)} u_1(s) + c_2(s) \end{aligned} \tag{10.1}$$

As can be seen in Fig. 10.1, the input to the process is composed of the output of the respective controller and the output of one of the decoupling transfer functions. The decoupling transfer functions have a manipulated variable as their input. In this way, the decoupling transfer functions are a feed-forward term with a having a manipulated variable as an input instead of a disturbance. For example, the decoupling transfer function $-\frac{G_{12}}{G_{11}}$ in Fig.

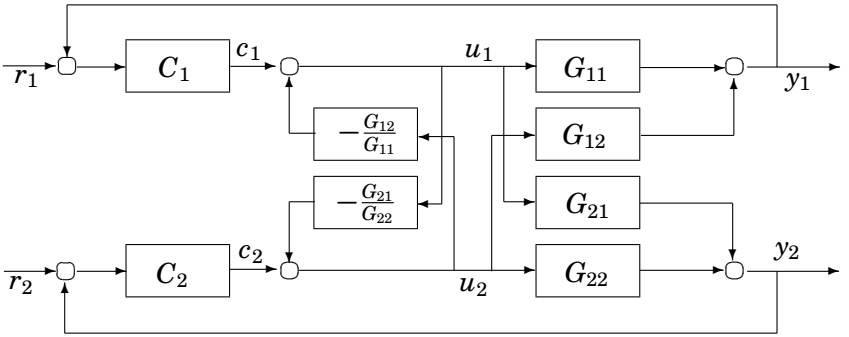

Figure 10.1 The control structure of inverted decoupling for a system with two inputs and two outputs.

10.1 has the manipulated variable u_2 as its input. The output is added to the output of controller C_1. Thus, the effect of u_2 on the process output y_1 is canceled out. This can be seen as follows. The transfer function from u_2 to y_1, assuming $c_1 = 0$ and going through the decoupling transfer function is:

$$\begin{aligned} y_1 &= G_{11} \cdot \left(-\frac{G_{12}}{G_{11}} u_2 \right) \\ &= -G_{12} u_2 \end{aligned}$$

The transfer function from u_2 to y_1, now going through the process and assuming $u_1 = 0$ is:

$$y_1 = G_{12} u_2$$

Taking everything together, the output y_1 now becomes:

$$\begin{aligned} y_1 &= G_{11} u_1 + G_{12} u_2 - G_{12} u_2 \\ &= G_{11} u_1 \end{aligned}$$

Hence, the decoupling transfer function $-\frac{G_{12}}{G_{11}}$ cancels the cross-coupling between y_1 and u_2. Similarly, the transfer function $-\frac{G_{12}}{G_{11}}$ cancels the cross-coupling between y_2 and u_1. The apparent process now consists of the diagonal elements of the process transfer matrix, and controllers can be designed in the same way as for two SISO processes.

Inverted Decoupling for $n \times n$ Processes

Inverted decoupling can be extended to higher order $n \times n$ processes, as it was done in [Garrido2011] for example. Denoting the process by $G(s)$

and the decoupling transfer matrix by $D(s)$, the apparent process as seen by the controllers is $T(s) = D(s) \cdot G(s)$. For inverted decoupling, $T(s) = \text{diag}\,(G_{11}, G_{22}, ..., G_{nn})$ is a diagonal matrix containing the diagonal elements of the process transfer matrix $G(s)$. This diagonal transfer matrix for the apparent process is achieved by choosing the decoupling matrix as shown in Eq. (10.2).

$$D(s) = G(s)^{-1}T(s) \tag{10.2}$$

The input signals to the decoupling system $D(s)$ are the controller outputs $c(s)$, and the outputs are the manipulated variables $u(s)$, so that

$$u(s) = D(s) \cdot c(s)$$

Using Eq. (10.2), the above equation becomes

$$\begin{aligned} & u(s) = G^{-1}(s)T(s) \cdot c(s) \\ \Leftrightarrow\ & c(s) = T^{-1}(s)G(s) \cdot u(s) \\ \Leftrightarrow\ & c(s) = \big(I - I + T^{-1}(s)G(s)\big) \cdot u(s) \\ \Leftrightarrow\ & c(s) = u(s) - \big(I - T^{-1}(s)G(s)\big) \cdot u(s) \\ \Leftrightarrow\ & u(s) = \big(I - T^{-1}(s)G(s)\big) \cdot u(s) + c(s) \end{aligned} \tag{10.3}$$

Since $T(s)$ is a diagonal matrix, its inverse is $T^{-1}(s) = \text{diag}\,\big(G_{11}^{-1}, G_{22}^{-1}, ..., G_{nn}^{-1}\big)$, where $G_{ii}(s),\ i = 1, ..., n$ are the 1×1 diagonal elements of the process transfer matrix $G(s)$.

With this, the decoupling network for $n \times n$ processes has the structure shown in Eq. (10.4).

$$\begin{bmatrix} u_1(s) \\ u_2(s) \\ \vdots \\ u_n(s) \end{bmatrix} = D(s) \begin{bmatrix} u_1(s) \\ u_2(s) \\ \vdots \\ u_n(s) \end{bmatrix} + \begin{bmatrix} c_1(s) \\ c_2(s) \\ \vdots \\ c_n(s) \end{bmatrix} \tag{10.4}$$

$$D(s) = \begin{bmatrix} 0 & -\frac{G_{12}}{G_{11}} & -\frac{G_{13}}{G_{11}} & \cdots & -\frac{G_{1n}}{G_{11}} \\ -\frac{G_{21}}{G_{22}} & 0 & -\frac{G_{23}}{G_{22}} & \cdots & -\frac{G_{2n}}{G_{22}} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ -\frac{G_{n1}}{G_{nn}} & -\frac{G_{n2}}{G_{nn}} & \cdots & -\frac{G_{n,n-1}}{G_{nn}} & 0 \end{bmatrix}.$$

As can be seen, extending the decoupling network to $n \times n$ systems of order higher than $n = 2$ keeps the structure of the 2×2 system in Eq. (10.1).

It can be noted, that here one requirement is that $G_{ii} \neq 0$. Furthermore, the number of poles and zeros in the G_{ij}, $i, j \in [1, n]$, have to be such that the transfer functions $\frac{G_{ij}}{G_{ii}}$, $i, j \in [1, n]$, are realizable. For cases where realizability is not given for inverted decoupling as used here, an extension to inverted decoupling to keep realizability is presented in [Garrido et al., 2011].

An Example of Inverted Decoupling

To show the effect of inverted decoupling, it is applied to a simple example. The equations of the coupled process is shown in Eq. (10.5).

$$G(s) = \begin{bmatrix} \frac{1}{s+1} & \frac{2}{3s+5} \\ \frac{1}{s+1} & \frac{1}{s+1} \end{bmatrix} \tag{10.5}$$

The process is to be controlled by two PI controllers $C_1(s)$ and $C_2(s)$, both of which have the same parameters, see Eq. (10.7).

$$C_1(s) = 2 + \frac{2}{s} \tag{10.6}$$

$$C_2(s) = 2 + \frac{2}{s} \tag{10.7}$$

A decoupling network is designed to decouple the process dynamics. The decoupling network according to Eq. (10.1) is here

$$\begin{aligned} u_1(s) &= 2\frac{(s+1)}{(3s+5)}u_2(s) + c_1(s) \\ u_2(s) &= 1 \cdot u_1(s) + c_2(s) \end{aligned} \tag{10.8}$$

Note that for $u_2(s)$, the decoupling turned out to be static here, while for $u_1(s)$ the decoupling is dynamic.

Figure 10.2 shows step responses of the closed-loop system with the process from Eq. (10.5) and the PI controller from Eq. (10.7) without a decoupling network in between them, and with the decoupling network (10.8) between them. The case without decoupling is shown as a blue line, and the case with decoupling as a red line. A unit step is applied to the first reference input r_1 at time $t = 10$, and to r_2 at time $t = 40$.

As can be seen in Fig 10.2, without the decoupling network a step in one of the inputs influences not only the respective output, but also the other output. Using the decoupling network, the closed-loop system behaves as if two single-loop processes were regulated by their respective controller. Both static and dynamic interactions could be eliminated.

In the next two chapters, the inverted decoupling method is used to cancel interactions present in buildings and their heating and ventilation systems.

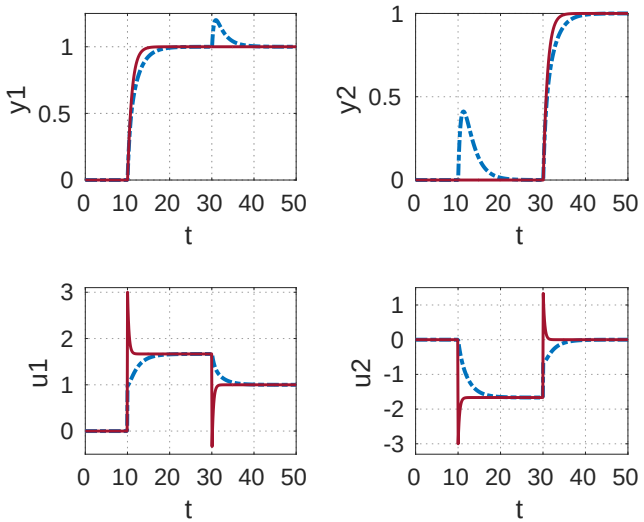

Figure 10.2 Step responses of the closed-loop system with the process in Eq. (10.5) and the PI controller from Eq. (10.7), without decoupling (dashed) and with decoupling (solid). Using decoupling, the control signals react quicker, and the step-responses resemble those of two SISO systems.

11

Temperature Control of Adjacent Rooms

The application of decoupled PI control on the temperature control of adjacent rooms is described in this chapter.

First, the simulation environment used for the simulations is introduced shortly, and the models implemented in it are presented. Then, the estimation of first-order models representing the temperature dynamics is described. After that, the proposed control algorithm is explained and simulation results are presented. At last, the results are discussed.

11.1 Adjacent Room Models in Modelica

For simulation of the temperature dynamics of four adjacent rooms, the Modelica language [The Modelica Association, 2016] was used. As a simulation environment Dymola (Dynamic Modeling Laboratory) [Dassault System, 2016] was used in this thesis.

This section starts with a short description of Modelica and the components used for implementing the simulation model used in this thesis. After that follows a description of the simulation model and the four-room building implemented with the Modelica language.

Modelica

Modelica [The Modelica Association, 2016] is an equation-based, object oriented language aimed at modeling the dynamic behavior of complex physical systems. It supports different physical domains such as electrical, mechanical, thermal, pneumatic, fluid, control and others. A model can include parts from several of these domains, for example combining the modeling of heat transfer dynamics with the fluid dynamics of the water-based heating system in one model.

With Modelica, the simulation model with components from all the different physical domains can be developed conveniently using a graphical editor, or using textual description. To build up a simulation model, either existing components can be used directly from a library, or these models can be altered and new ones can be added.

The Buildings Library

The Buildings Library [Lawrence Berkeley National Laboratory, 2016a; Wetter et al., 2011b] is a library for the Modelica language, which offers component and system models for buildings and its energy and control systems. It was developed at the Lawrence Berkley National Laboratory, starting in 2007. Since the library is developed in the Modelica language, existing components or system models can be modified or extended to create new ones. In this way, heating and ventilation systems can be tested even in unconventional setups. Furthermore, the developed system models can be used to develop and test various control algorithms [Wetter et al., 2011b].

The Buildings Library consists of several packages. Using components from these packages, HVAC systems, water-based heating systems and heat transfer between rooms and rooms and the outside, among others, can be implemented [Wetter et al., 2011b; Wetter et al., 2011a]. Some of these packages are listed below:

Airflow The Airflow package contains models for simulation of air flow between different rooms or zones in a building system simulation, as well as between the room and the outside.

BoudaryCondition A weather data reader and models for solar radiation and sky temperatures are available in the BoudaryCondition package.

Fluid Models and components for air- and water-based heating and ventilation systems are provided in the Fluid package. Depending on the component, models are based on either first principles, steady-state performance curves or, for transient behavior, differential equations.

HeatTransfer The package HeatTransfer contains components modeling heat transfer through, for example, walls consisting of multiple layers and through windows.

Rooms The Rooms package includes models for the heat transfer between rooms as well as through the building envelope. After generating several instances of a room model from this package, the different room models can be connected to a multi-room or multi-zone building. Furthermore, the room models can be linked to models of heating and ventilation systems from the Fluid package.

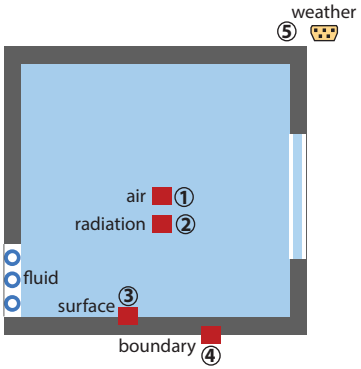

Figure 11.1 Icon for a room from the Room package in the Buildings Library. [Lawrence Berkeley National Laboratory, 2016b].

In this thesis, the Rooms package was used to simulate several adjacent rooms. This simulation model was used to develop and test decoupled PI controllers.

The Room Model in the Buildings Library

A model of a room is available in the package **Rooms** [Wetter et al., 2011a]. The icon for the room in shown in Fig. 11.1. This model implements a room assuming perfectly mixed air.

The room can have any number of walls, ceilings, floors or other constructions and surfaces. These take part in heat exchange through conduction, convection or infrared or solar radiation.

Within a room, different components from the Buildings Library are used to compute the heat conduction through the constructions and surfaces. There are components for constructions with windows and for those without, and components for constructions which partition several zones. In order to connect one room to another, there are constructions with only one side taking part in the heat exchange of the room. The conduction of heat through the components is implemented as one-dimensional heat conduction though multi-layered materials.

The room also contains a component modeling the solar radiation absorbed by the glass of the windows and by shadings. This is used by the heat transfer models to take part in the heat balance of the room. Radiative and convective heat transfer to the sky and the outside environment are modeled through boundary conditions. Construction parameters can be specified for all components the room consists of. These parameters describe the geometry of the constructions and the material properties.

Components from other packages of the Building Library can be con-

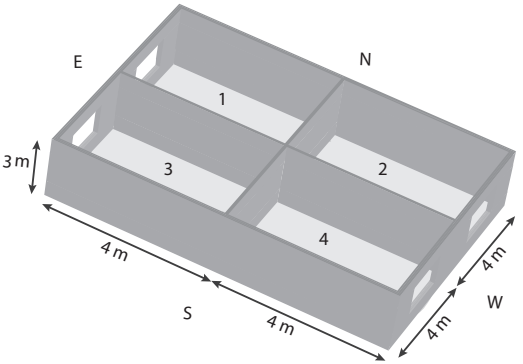

Figure 11.2 Floor plan of a 2-by-2 setup of rooms, which are modeled using the Modelica Buildings Library. Each room has a floor size of 4×4 [m], and has one window.

nected to a room model. For example, an air conditioning system can be connected by using components from the Fluid package. Using the Airflow package, an open door between two rooms can be modeled.

The room model has several connectors in order to integrate it into a larger model. These connectors are depicted as red squares in the symbol in Fig. 11.1. With the help of the weather data connector (5 in Fig. 11.1), weather data can be loaded from a file, or the conditions can be entered manually.

The room temperature can be measured at connector port 1 in Fig. 11.1. Through the same connector port, convective heat can be added to the room model. To add radiative heat to the room model, connector port 2 in Fig. 11.1 can be used.

The connector ports 3 and 4 in Fig. 11.1 give the possibility to connect the room model to a floor or a radiant slap, or to connect several room models. These ports were used in this thesis to build up models of adjacent rooms.

Modeling of Adjacent Rooms in Modelica

For the purpose of the thesis, a model of a two-by-two structure of four rooms are implemented with the Modelica Buildings Library. This model was then used to test the proposed decoupled PI controller.

Figure 11.2 shows the floor plan of the building implemented with the Buildings Library. The model consists of four rooms arranged in a 2-by-2 square. Each room has a size of 4×4 m^2, and the rooms have a hight of 3 m. The east and west facing walls of each room have a window.

In Modelica, this model is implemented using the room model from the

Buildings library described in Sec. 11.1. Each of the four rooms is modeled with such a room model. These room models can then be interconnected for simulation of heat transfer. This coupling of the all the room's heat transfer is done using the port labeled number 3 in Fig. 11.1. To model the wall between the rooms, a multi-layer wall component is connected between two of these ports. The outside walls of the rooms are modeled as 120 mm brick walls, and the walls between the rooms as 5 *mm* concrete walls.

To heat the rooms, heating power in [W] is added directly into the rooms via the port number 1 in Fig. 11.1. Adding a positive heat flow Q corresponds to heating, adding a negative amount of heat corresponds to cooling, and $Q = 0$ corresponds to neither heating nor cooling. Here, only heating is considered. Since the air in the room is assumed to be completely mixed, the whole room is heated evenly when heating power is added. The outside temperature is set to constant $-20°C$. There are no disturbances from occupants or appliances.

11.2 Estimating Temperature Dynamics

In order to implement the decoupled PI controller, simple models of the adjacent rooms in Section 11.1 are needed. For this purpose, first order models are estimated using data from step-response test. This section describes the estimation of these first-order models.

Estimation Data

To identify the model parameters, step responses were recorded from the Modelica model. For this, as many experiments are recorded as there are rooms in the models. For each experiment, the heat flow into one of the rooms was set to 3000 W, while the heat flow into the remaining rooms was zero. The temperatures in all rooms were recorded. This was repeated for all rooms in the model.

The step-response data used to estimate the first-order models for the 2-by-2 room configuration is shown in Fig. 11.3. Four Experiments were performed, each shown in its respective row in the figure. The columns in the figure correspond to the measured temperatures in rooms 1, 2, 3 and 4, respectively.

For each experiment, a step in heat flow was applied to one of the rooms, while the heat flow in the other rooms remained unchanged. In Experiment 1, a step-change from zero to 3000 [W] was applied to the heat flow into the first room at $t = 500$, while the heat flow to the other rooms was zero. For experiments 2, 3 and 4, the step-change in heat flow was done for room 2, 3 and 4, respectively, while the heat flow in other rooms remained zero. In each of the experiments, the temperatures of all four rooms were recorded.

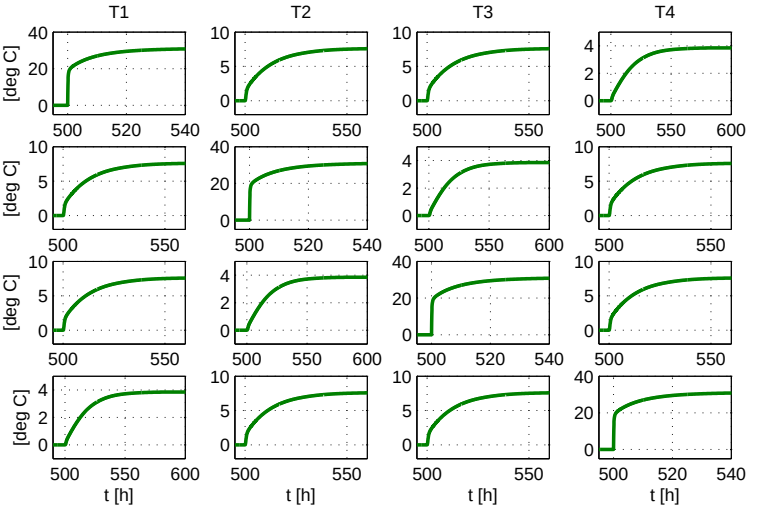

Figure 11.3 Identification data for estimating first-order models for the 2-by-2 room configuration. The outside temperature was −20 °C and the input step hight was 3000 [W]. The effect of the outside temperature was subtracted from the measured data, resulting in the step response data starting at zero °C. The first row shows the four room temperatures, when a step in heat flow was applied to room number 1. Rows 2, 3 and 4 show the room temperatures when a step in heat flow was applied to rooms 2, 3 and 4, respectively.

Since the first-order models to be estimated should describe the effect of a change in heat flow into the rooms on the room temperatures, and not the effect of the outside temperature as a disturbance, the effect of the outside temperature in the room temperatures was subtracted from the recorded data. The initial room temperature in the recorded data was −20 °C, since the outside temperature is −20 °C. After subtraction the step-response data has an initial value of zero degree Celsius, as shown in Fig. 11.3.

As can be seen in the estimation data, the dynamic behavior of the system does not resemble that of first-order systems. This is most pronounced in the subplots on the main diagonal of Fig. 11.3. These are the room temperatures of the rooms, to which the heating power was added. This is not surprising, since the simulation model is non-linear. However, the data in Fig. 11.3 suggest that there are two time constants involved in the heating of the rooms. When heat is added to a room, the air in room as well as the constructions, such as walls and floors, are heated up. It takes a longer time for walls and floors to take up the heat than for the room air. Hence, the

heating up of room air results in a shorter time constant than the heating up of the walls and floors. Hence, a second-order model would probably describe the step-response data better than a first-order model. However, for simplicity of the identification process, first-order models are used here.

Here, the step-response data was generated by applying a step in the heat flow Q from zero to 3000 [W]. Since the heat flow Q in the closed-loop experiments in Section 11.4 is approximately between 500 [W] and 1500 [W], and the simulation model is nonlinear, using estimation data with Q in the same range would lead to a more accurate model for the dynamic behavior around the operating point. However, the idea for identification here was to have heat flow added to only one of the rooms, and have no heating or cooling in the other rooms. As described above, the temperature dynamics of the 2-by-2 rooms are approximated by first-order models, although the step-response data suggests models of 2nd order. Hence accuracy of the first-oder models might not be increased by estimation around a different operating point for the heat flow input Q.

Estimation of First Order Models

With the heat flow into the rooms as inputs and the room temperature as outputs, the four rooms shown in Fig. 11.2 become a MIMO system with four inputs and four output.

The system of transfer functions describing the heat transfer between the four rooms is shown in Eq. (11.1).

$$\begin{aligned} T_1(s) &= G_{11}(s)Q_1(s) + G_{12}(s)Q_2(s) + G_{13}(s)Q_3(s) + G_{14}(s)Q_4(s) \\ T_2(s) &= G_{21}(s)Q_1(s) + G_{22}(s)Q_2(s) + G_{23}(s)Q_3(s) + G_{24}(s)Q_4(s) \\ T_3(s) &= G_{31}(s)Q_1(s) + G_{32}(s)Q_2(s) + G_{33}(s)Q_3(s) + G_{34}(s)Q_4(s) \\ T_4(s) &= G_{41}(s)Q_1(s) + G_{42}(s)Q_2(s) + G_{43}(s)Q_3(s) + G_{44}(s)Q_4(s) \end{aligned} \tag{11.1}$$

The inputs to this process are the heat flows $Q_i(s),\ i = 1, ..., 4$ into the four rooms, and the output are the room temperatures $T_i(s),\ i = 1, ..., 4$ in the four rooms.

Comparing to the estimation data in Fig. 11.3, the transfer function $G_{ij}(s)$ is estimated using the step-response data in row j and column i in Fig. 11.3. Utilizing this step-response data, the gains K_{ij} and the time constants d_{ij} for first-order models of the following form were estimated:

$$G_{ij}(s) = \frac{K_{ij}}{1 + d_{ij}s} \tag{11.2}$$

The **gains $\mathbf{K_{ij}}$** were estimated using the steady-state form of equation (11.1). The steady-state equation for the temperature in room number i is

obtained by setting $s = 0$ in Eq. (11.1).

$$T_i(0) = G_{i1}(0)Q_1(0) + G_{i2}(0)Q_2(0) + G_{i3}(0)Q_3(0) + G_{i4}(0)Q_4(0) \quad (11.3)$$

Since there are four rooms, it is $i = 1, ..., 4$.

During collection of estimation data, the heat flow into room j was $Q_j \neq 0$, and the heat flow into the remaining rooms was $Q_i = 0, i \in [1, 4] \backslash k$. This was repeated for all 4 rooms. Thus, Eq. (11.3) becomes

$$T_i(0) = G_{ij}(0)Q_j(0), \ i \in [1, 4]$$

Hence, the gain K_{ij} can be computed using the steady-state measurements of the room temperature of room i and the heat flow into room j, as shown in Eq. (11.4).

$$K_{ij} = \frac{T_i(0)}{Q_j(0)} \quad (11.4)$$

where $i, j \in [1, 4]$.

The **time constants $\mathbf{d_{ij}}$** were estimated using the time domain representation of (11.2). Assuming a step at the input, i.e. $Q_j(s) = \frac{M_j}{s}$, the temperature in room i for a step of size M_j in the heat flow is

$$T_i(s) = \frac{K_{ij}}{1 + d_{ij}s} \frac{M_j}{s}$$

The time domain representation of this is

$$T_i(t) = K_{ij}M_j \left(1 - e^{-\frac{t}{d_{ij}}}\right)$$

The determination of the time constant d_{ij} in the above equation was inspired by a method in [Astrom and Hagglund, 2006]. The time when the output signal $T_i(t)$ reaches 63% and the value of the output signal at that time were recorded. Denoting the time where $T_i(t)$ reaches 63% by t_{63} and the value of $T_i(t)$ at that time by $0.63 \cdot T_i(\infty)$, and substituting this into the above time domain representation gives

$$0.63 \cdot T_i(\infty) = K_{ij}M_j \left(1 - e^{-\frac{t_{63}}{d_{ij}}}\right)$$

Thus the time constant was calculated as shown in Eq. 11.5.

$$d_{ij} = -t_{63} \left[\ln\left(1 - \frac{0.63 \cdot T_i(\infty)}{K_{ij}M_j}\right)\right]^{-1} \quad (11.5)$$

The step size M_j is known from the measurement data, as well as the value $0.63 \cdot T_i(\infty)$ for 63% of the final value of the step response. The time t_{63} can be read from the step-response data, and the gain K_{ij} has been calculated previously.

Therefore, to estimate the first-order models, first the static gains K_{ij} are calculated using the steady-state values of the rooms temperatures $T_i(0)$ and the heat flows into the rooms $Q_j(0)$. Then, using the height of the step in Q_j, M_j, the time where the rooms temperature reaches 63% of its final steady-state value, and the value of the output signal at that time, the time constant for the model is calculated with Eq. (11.5).

Estimation Results

The parameters of the first order models estimated as described in Section 11.2 are shown in table 11.1 for four adjacent rooms. It can be seen in the estimated time constants, that the heat dynamics of the adjacent rooms in both cases are quite slow processes. As mentioned previously, there are two time constants describing the temperature dynamics. A fast time constant corresponds the heating of the room air when heat flow is added to a room. A slow time constant represents the heating of building elements, such as walls and floors, which also contributes to the heating of the room air. Here, the slower time constant was estimated.

Table 11.1 Parameters for the first order model estimated for four adjacent rooms. $\mathbf{Q = 3000}$ and $\mathbf{T_{out} = -20°C}$.

	K [10^{-3} °C/W]	**d [h]**		**K [10^{-3} °C/W]**	**d [h]**
$\mathbf{G_{11}}$	10.4	2.2	$\mathbf{G_{31}}$	2.6	10.5
$\mathbf{G_{12}}$	2.6	10.5	$\mathbf{G_{32}}$	1.3	17.2
$\mathbf{G_{13}}$	2.6	10.5	$\mathbf{G_{33}}$	10.4	2.2
$\mathbf{G_{14}}$	1.3	17.2	$\mathbf{G_{34}}$	2.6	10.5
$\mathbf{G_{21}}$	2.6	10.5	$\mathbf{G_{41}}$	1.3	17.2
$\mathbf{G_{22}}$	10.4	2.2	$\mathbf{G_{42}}$	2.6	10.5
$\mathbf{G_{23}}$	1.3	17.2	$\mathbf{G_{43}}$	2.6	10.5
$\mathbf{G_{24}}$	2.6	10.5	$\mathbf{G_{44}}$	10.4	2.2

It can be seen in table 11.1, that for the G_{ij} with $i = j$, the estimated time constant is smaller than for the G_{ij} with $i \neq j$. This means that the room, to which the heat flow is added, is heated up faster than the neighboring rooms. The gains estimated are larger for G_{ij} with $i = j$ than for G_{ij} with $i \neq j$. Thus, the heat flow has the largest effect on the room it is added to. The effect on neighboring rooms is less. This can also be seen in the estimation data in Fig. 11.3.

To evaluate the estimated first order models, the input data are applied to the estimated first-order models. The output data resulting from this are compared to the recorded output data. For the four adjacent rooms, the identification results are shown in Fig. 11.4.

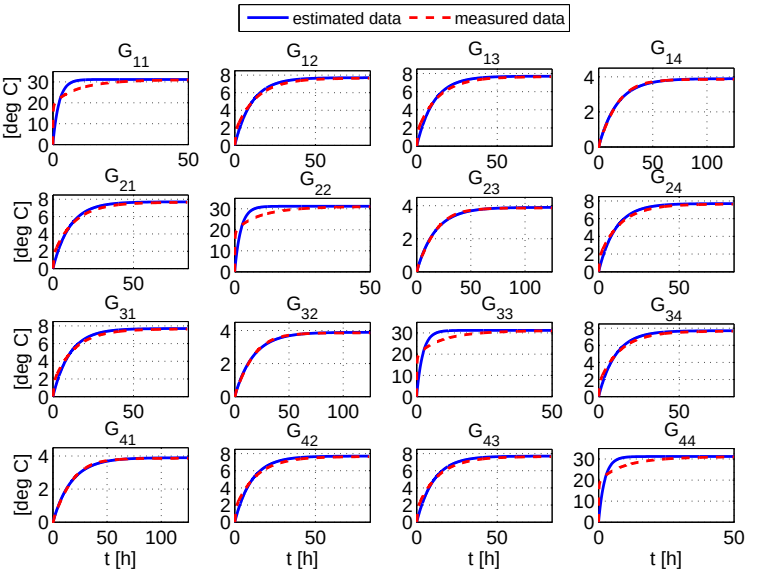

Figure 11.4 Identification results of first order models for four rooms. Output of identified model (red dashed line) and measured temperature (blue solid line). The outside temperature was -20°C. For the models on the main diagonal of this figure, the fit was ca. 30%, and for the other models ca. 85% to 95%.

The correctness of the estimated models was evaluated by calculating the goodness of fit between the output data generated with the estimated models and the recorded room temperature data. This resulted in an approximate fit of 30% for the G_{ij} where $i = j$, and an approximate fit of 85% to 95% for the remaining G_{ij}. In Fig. 11.4 it can be seen as well, that the first-order models fit better for the G_{ij} where $i \neq j$ than for those where $i = j$. This is due to the two time constants in the heat transfer dynamics, as discussed previously. Here, the first-order models were estimated to capture the slower dynamics (compare to Section 11.2 and Fig. 11.3). This results in correct steady-state values, but especially for the transfer functions G_{ij} with $i = j$ the transient dynamics are inaccurate. For the transfer functions G_{ij} with $i \neq j$, there is a smaller inaccuracy. To increase the room temperature of an adjacent room, the wall temperatures have to increase, which is connected

to the slower of the two time constants. Hence, for adjacent rooms, the faster time constant is less pronounced and a first-order model describes the temperature dynamics of the adjacent rooms better than the dynamics of the room, which the heat flow is added to.

11.3 Decoupled PI Control

The temperature dynamics of the adjacent rooms are coupled due to the interconnecting walls. This can be seen in the estimation data in Fig. 11.3 for the four adjacent rooms. An increased heat flow into one of the rooms also increases the temperature in adjacent rooms. The extent of the influence is of course depending on the building materials of the walls. In badly isolated houses the influence can be significant.

Many times, the room temperature would be regulated locally for each room. Then, a change of reference temperature in one of the rooms is expected to have an influence on the temperature in adjacent rooms.

The purpose of a decoupling network is to decrease this effect. When the reference value for one of the system outputs is changed, the controller will adjust the corresponding input to drive the output to its new desired value. If the system is coupled, that change in input signal will also have an effect on the other output signals, which is not desired. With the help of the decoupling network, the effect of the change in input signal on these other output signals can be estimated. With this estimate, the decoupling network can help the PI controllers regulating the other output signals to counteract, so that the level of the other output signals does not change.

This section first describes how decoupling is done in case of the adjacent rooms and how it is implemented with the Modelica Buildings Library models. Furthermore, simulation results comparing the PI controlled room temperature with and without decoupling network are presented.

Inverted Decoupling for Adjacent Rooms

To decouple the temperature dynamics of the adjacent room, inverted decoupling as described in Section 10.2 is used.

Here, the heat flows into the rooms are the input signals for the system of four adjacent rooms, and the room temperatures are the output signals. The transfer functions describing the dynamics of this system are given in Eq. (11.1).

The decoupling network, as in Eq. (10.4), for the case of the four adjacent rooms is shown in Eq. (11.6). The heat flow signals calculated by the PI controllers are denoted by $\bar{Q}_i,\ i = 1, ..., 4$. The heat flow signals after manipulation by the decoupling network, which are the input signals to the rooms, are denoted by $Q_i,\ i = 1, ..., 4$. Comparing this to Section 10.2, the

$\bar{Q}_i$ here correspond to the c_i in Section 10.2, and the Q_i correspond to the u_i in Section 10.2.

$$\begin{aligned}
Q_1(s) &= \bar{Q}_1(s) - Q_2(s)\frac{G_{12}(s)}{G_{11}(s)} - Q_3(s)\frac{G_{13}(s)}{G_{11}(s)} - Q_4(s)\frac{G_{14}(s)}{G_{11}(s)} \\
Q_2(s) &= \bar{Q}_2(s) - Q_1(s)\frac{G_{21}(s)}{G_{22}(s)} - Q_3(s)\frac{G_{23}(s)}{G_{22}(s)} - Q_4(s)\frac{G_{24}(s)}{G_{22}(s)} \\
Q_3(s) &= \bar{Q}_3(s) - Q_1(s)\frac{G_{31}(s)}{G_{33}(s)} - Q_2(s)\frac{G_{32}(s)}{G_{33}(s)} - Q_4(s)\frac{G_{34}(s)}{G_{33}(s)} \\
Q_4(s) &= \bar{Q}_4(s) - Q_1(s)\frac{G_{41}(s)}{G_{44}(s)} - Q_2(s)\frac{G_{42}(s)}{G_{44}(s)} - Q_3(s)\frac{G_{43}(s)}{G_{33}(s)}
\end{aligned} \tag{11.6}$$

To cope with constraints on the allowed heat flow $Q_i,\ i = 1, ..., 4$, an anti-windup strategy was used for the PI controllers used with the decoupling network. Because of the structure of inverted decoupling, an anti-windup method as employed for single PID controllers could be applied directly [Gagnon et al., 1998].

Implementation The decoupling method presented in this section was implemented with the Modelica language into the model of four rooms presented in Section 11.1. This implementation was done using transfer functions.

For each of the rooms, a PI controller was designed to regulate the room temperature of one room. The control signal was the heat flow into the respective room. The PI controllers were tuned using the Ziegler-Nichols method to locally regulate the room temperature. The decoupling network in Eq. (11.6) was then added between the PI controllers and the system to be controlled, i.e., the four adjacent rooms (compare also to Section 10.2).

Simulation Setup To test the decoupling network, the reference for the room temperature was changed for one of the rooms, and the room temperatures in all the rooms were recorded. This was done both with and without the use of the decoupling network.

At the beginning of the simulation, all room temperatures were set to 25 °C. After a steady-state has been reached, the reference temperature for room 1 was changed to 16 °C. The outside temperature was set to a constant value of −20 °C.

11.4 Simulation Results

The results simulating the four adjacent rooms are shown in Fig. 11.5. The blue solid line shows the simulation results without using a decoupling network, i.e., just local PI controller for each room. The red dashed line shows the simulation results with a decoupling network.

In Fig. 11.5, the temperatures in rooms 1, 2, 3 and 4 are shown in the upper left panel, the upper right panel, the third panel on the left and the third panel on the right, respectively. The heat flow into the rooms, as determined by the PI controllers, are shown in the second panel on the left side, the second panel on the right side, the lower left panel and the lower right panel, respectively. The time is shown in hours as well.

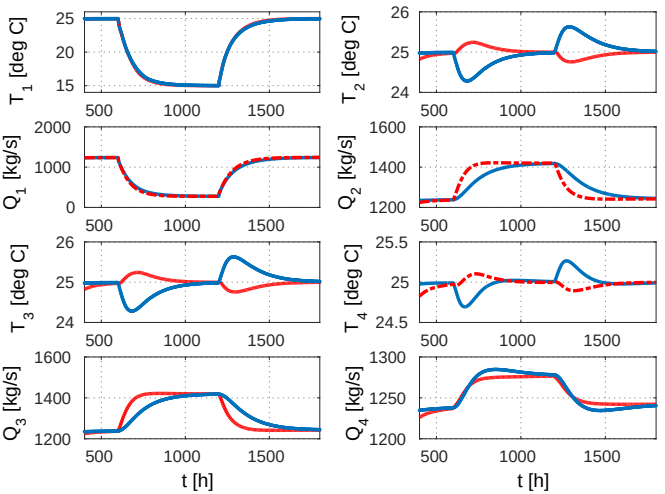

Figure 11.5 In Modelica: PI control without decoupling (blue solid line) and with inverted decoupling (red dashed line). With the decoupled PI, the effect of the coupled temperature dynamics is decreased. The control signals Q_i, $i = 1, ..., 4$, react sooner when decoupling is used.

It can be seen that without the decoupling network, the temperature of the adjacent rooms deviates from its desired value with 0.5 to 1 °C. It takes some hours until the PI controller in the respective room has compensated for the disturbance and driven the temperature back to its desired value. Using the decoupling network, the deviation in the room temperature is smaller. The room temperature deviation was 0.7 °C without a decoupling network, and 0.3 °C with a decoupling network.

The remaining deviation of the room temperature of adjacent rooms,

despite a decoupling network, is due to the fact that the first-order models used in the decoupling transfer functions do not describe the temperature dynamics very accurate. First-order models were chosen due to simplicity of the identification process. However, as can be seen in Fig. 11.4, there are some significant deviations between the output data generated using the estimated model and the recorded output data, especially for G_{ij} where $i = j$. This is because the temperature dynamics are governed by two different time constants, as described in Section 11.2. Using second-order models instead of first-order models, would describe the temperature dynamics more accurately and could lead to a better decoupling.

Using a decoupling network, the heat flow into the rooms is increased earlier compared to the case when no decoupling network is used. The decoupling network consists of feed-forward transfer functions representing the temperature dynamics of the adjacent rooms. With this, the influence of the disturbance from the temperature change occurring in a neighboring room is counteracted earlier than when just a PI controller would be able to see it.

12

Decoupling Control in a Ventilation System

The application of inverted decoupling to a ventilation system is described in this chapter.

The coupling here is from the mass flow rate of the air in the ventilation system to the room temperature. If there is a disturbance in the CO_2 concentration, for example from people entering the room, the mass flow rate needs to be increased in order to keep the CO_2 concentration at an acceptable value. Especially during winter months in northern Europe, this can lead to a drop in room temperature, which is not desired.

The goal is to decrease this deviation of the room temperature by using inverted decoupling in addition to a PI controller. The room temperature is to be regulated using the temperature of the ventilation air. Since the room temperature gets influenced by the mass flow rate of the air in the ventilation system, but the CO_2 concentration does not get influenced by the temperature of the ventilation air, the coupling is one-directional. Hence, only one of the two decoupling transfer functions usually present in inverted decoupling for 2-by-2 systems is used. With this, the decoupling becomes a feed-forward transfer function from the mass flow rate of the air in the ventilation system to the temperature of the air in the ventilation system.

This chapter starts with a description of the room and its ventilation system, which are used for experiments. Then, a dynamical model for the temperature and CO_2 dynamics is presented. This model was used for simulation studies, which were carried out prior to experiments. After that, the estimation of first-order models needed for the decoupling transfer function is described. Then, the decoupling transfer function, as well as the discretization of the decoupling transfer function and the used PI controller are explained. Next follows a description of the simulation studies done in Matlab. Then, the experiments performed in a real room are described.

12.1 The KTH Test-bed

Experiments are performed using a test-bed located at the Royal Institute of Technology in Stockholm, Sweden. The test-bed is installed in the basement of a multi-stories university buildings.

One of the rooms and its ventilation system are used to perform the experiments. This room has a size of 81 m^2. The walls are constructed of a heavyweight concrete structure. The south-east facing wall is an outside wall, with ca. 2.5 m^2 of window area. During the university's semester time, the room is used for student laboratories.

There are sensors measuring temperature, CO_2 and humidity at different locations in the room. Temperature sensors are found at each wall of the room, and in the center of the room under the ceiling. Here, the temperature sensor in the center of the room is used.

A work-station computer located in the room is connected to the test-bed. This computer has Matlab installed, where control algorithms to be tested in experiments can be implemented. The implementation using Matlab is done using an object-oriented approach. There are already classes implementing the connection to the test-bed, i.e., the measurement of data from the sensors installed in the rooms and the application of control signals to actuators. Other classes implemented are, for example, classes to handle time measurements and one for a general controller. The class for a general controller is used as a base-class for all control algorithms. For example, this class handles the input and output signals for a controller and provides the possibility for bounds on input and output signals. A control algorithm to be tested is then implemented in the context of this already existing implementation, using the general controller class as a base-class.

The ventilation system

The ventilation system for the room is a variable air volume (VAV) ventilation system, where the air flow rate depends on the CO_2 concentration in the room. A scheme depicting the ventilation system is shown in Figure 12.1. Fresh air from the outside is first processed by a heat exchanger using exhaust air from the room. Then, the air flows through a heating and a cooling system. Having a temperature of approximately 20°C, the air then passes a duct to be distributed into the room. A part of this air, ca. 70%, is delivered into the room directly. The remaining ca. 30% can be cooled by means of chilled water. The flow of this chilled water is regulated with a chilled water valve, determining how much the air gets cooled down. Note that, since the air in the ventilation system as delivered to the room is around 20°C, the ventilation system can be used for both cooling and heating.

The actuators available in the ventilation system are two air dampers,

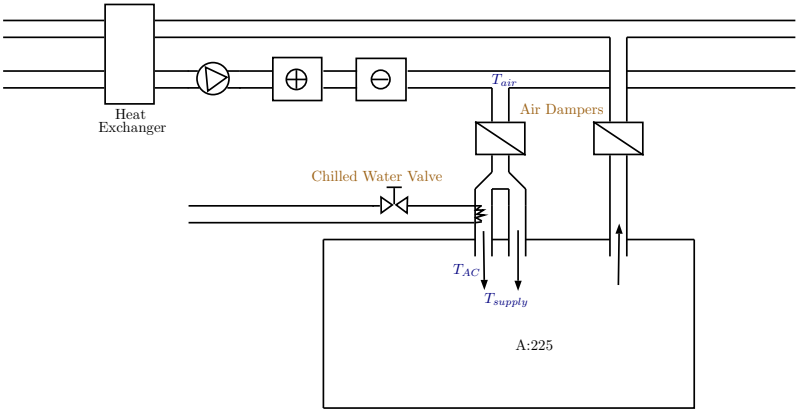

Figure 12.1 A scheme of the ventilation system installed in the room of the KTH test-bed. The chilled water valve and the air dampers are the actuators, which can be used. Here, the supply air temperature $T_{\mathbf{supply}}$ and the mass flow rate of the air in the ventilation system were used as control signals.

and a chilled water valve. The air dampers regulate the rate of the airflow into and out of the room, and the chilled water valve regulates the flow of cold water used to cool down a part of the air entering the room.

The air dampers are always set to an equal opening percentage, so that the pressure in the room is not changed. The mass flow rate $\dot{m}_v$ into the room is given by the valve opening percentage of the dampers through the static map shows in Eq. (12.1).

$$\dot{m}_v = V_o \cdot p_1 - p_2 \tag{12.1}$$

The constants are $p_1 = 0.002$, $p_2 = 0.1$ and V_o is the valve opening percentage.

Because of the architecture of the ventilation system, the chilled water can cool down only 30% of the air entering the room. The temperature of the part of the air cooled down by the chilled water is denoted by T_{AC} and the temperature of the air in the ventilation system by T_{air}. The temperature of the total air entering the room, T_{supply}, is hence calculated by Eq. (12.2).

$$T_{\text{supply}} = 0.3 \cdot T_{\text{AC}} + 0.7 \cdot T_{\text{air}}. \tag{12.2}$$

There are limitations on the temperatures T_{AC}, T_{air} and T_{supply}. By opening the chilled water valve to 100%, T_{AC} can be lowered to $16°C$. By closing the chilled water valve completely, T_{AC} can be increased to $21°C$. Hence, the limits are $T_{\text{AC}} \in [16\ 21]°C$. Following from Eq. (12.2), and $T_{\text{air}} = 22.5$, the limits for the supply air temperature are $T_{\text{supply}} \in [20.5\ 22.05]°C$.

From measurements of T_{AC} and T_{supply}, and the help of Eq. (12.2) it could be seen that T_{air} during the experiments done was around $22.5°C$. From Eq. (12.1) it follows that for air dampers open between 0% and 100%, the air flow rate of the air entering the room from the ventilation system varies in the range of $\dot{m}_V \in [0.1002\ 0.34]$.

The low-level controller

Of the air entering the room from the ventilation system, ca 30% can be cooled down, as mentioned above. The cooling down is done using cold water. The flow rate of this cold water determines how much the ventilation air is cooled down. This flow rate of cold water is regulated using the chilled water valve. A low-level PI controller determines the opening percentage of this valve needed to reach a certain AC temperature T_{AC}. This PID controller was already present in the test-bed system, and was not modified for this thesis. The block diagram in Fig. 12.2 shows how the low-level PID controller regulates the AC temperature T_{AC} based on the desired AC temperature $T_{\text{AC,ref}}$ using the valve opening percentage $V_{o,AC}$.

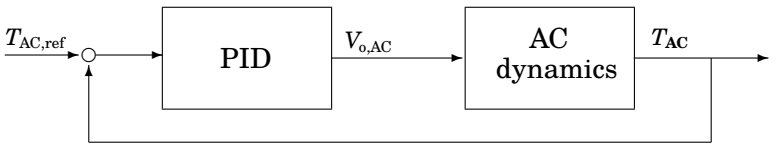

Figure 12.2 A block diagram of the low-level PID installed in the ventilation system to control the AC temperature, using the opening position of the chilled water valve $V_{\text{o,AC}}$ as a control signal.

The decoupled PI controller proposed in this thesis was designed to use the supply air temperature T_{supply} as a control signal. Hence, the desired AC temperature $T_{\text{AC,ref}}$ needs to be deduced from a desired supply air temperature, provided by the decoupled PI controller. Also, both the CO_2 concentration and the room temperature are influenced by the air flow rate $\dot{m}_v$ in the ventilation system. Hence, the system to be controlled by the decoupled PI controller is shown in Figure 12.3.

It has the desired supply air temperature and the air flow rate in the ventilation system as input signals, and the rooms temperature and the CO_2 concentration as outputs. The desired supply air temperature is recalculated into the desired AC temperature using Eq. (12.2). The low-level PI controller then adjusts the valve position of the chilled water valve such that the desired AC temperature, and thus the desired supply air temperature, is reached. The air flow rate is recalculated into the respective valve opening positions for the air dampers using the static map in Eq. (12.1).

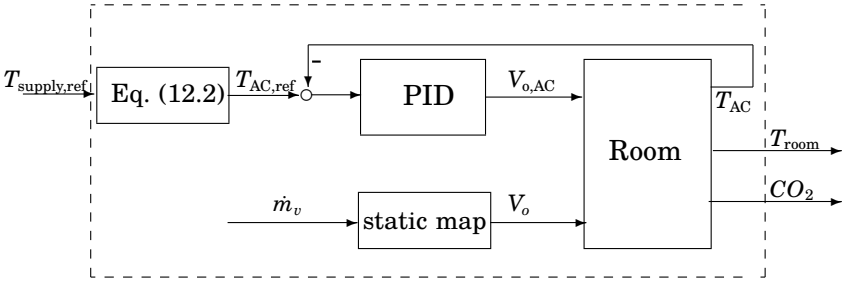

Figure 12.3 A block diagram of the system to be controlled. This system includes the temperature and CO_2 dynamics of the room, the low-level PID controller for the AC temperature, a static map transforming the mass flow rate $\dot{m}_v$ of the ventilation air into the opening percentage of the air damper valves V_o and Eq. (12.2) to recalculate a reference supply air temperature $T_{\text{supply,ref}}$ into a reference AC temperature $T_{\text{AC,ref}}$

In the following sections, including simulation studies and experiments, it is assumed that the low-level PID controller regulates the AC temperature perfectly.

12.2 Modeling the KTH Test-Bed

Before experiments are performed on the test-bed, the proposed decoupled PI controller was tested in Matlab and Simulink. The dynamical model used for simulation can be found in [Scotton, 2012; Parisio et al., 2013; Parisio et al., 2014a], where the authors developed the model based on simplified general building physics modeling of the test-bed room. A brief description of this model is given in this section.

For the model it is assumed that the thermal zone is well mixed, that there is no air infiltration and that thermal effects of vapor production are neglected.

The energy balance for the temperature of the room is as shown in Eq. (12.3).

$$m_a c_{pa} \frac{dT_{\text{room}}}{dt} = Q_{\text{vent}} + \sum_j Q_{\text{wall},j} + \sum_j Q_{\text{win},j} + Q_{\text{occu}} + Q_{\text{rad}} \tag{12.3}$$

where the heat flow from the ventilation system is Q_{vent}, the heat flow between wall j and the room node is $Q_{\text{wall},j}$, the heat flow between window j and the room node is $Q_{\text{win},j}$, the heat flow produced by internal gains like occupants, equipment or lightning is Q_{occu} and the heat flow produced by the radiators in the room is Q_{rad}.

With this, the temperature dynamics at a node in the center of the room are described by Eq. (12.4).

$$\begin{aligned}\frac{dT_{\text{room}}}{dt} &= \frac{\dot{m}_v}{m_a}\left(T_{\text{supply}} - T_{\text{room}}\right) + \sum_j \frac{h_i A^j_{\text{wall}}}{m_a c_{pa}}\left(T^j_{\text{wall,i}} - T_{\text{room}}\right) \\ &+ \sum_j \frac{1}{R^j_{\text{win}} m_a c_{pa}}\left(T_{\text{out}} - T_{\text{room}}\right) + \frac{c}{m_a c_{pa}} N_{\text{people}} \\ &+ \frac{1}{m_a c_{pa}} \sum_j G^j A^j_{\text{win}} I^j + \frac{A_{\text{rad}} h_{\text{rad}}}{m_a c_{pa}}\left(T_{\text{mean,rad}} - T_{\text{room}}\right)\end{aligned} \tag{12.4}$$

The room temperature T_{room} depends on the ambient temperature T_{out}, on the temperatures of the insides of the walls $T^j_{\text{wall,i}}$, the radiation I^j through the windows, the air flow $\dot{m}_v$ and the air temperature T_{supply} of the air from the ventilation system, on the mean radiator temperature $T_{\text{mean,rad}}$ and the number of people N_{people} in the room.

All walls are modeled as a network of two capacitors and three resistors. The temperature at the inside of wall j is calculated as an energy balance between the outdoor and the indoor surface of that wall, see Eq. (12.5).

$$\begin{aligned}\frac{dT^j_{\text{wall,o}}}{dt} &= \frac{2h_o A^j_{\text{wall}}}{C^j}\left(T_{\text{out}} - T^j_{\text{all,o}}\right) + \frac{2}{R^j_{\text{wall}} C^j}\left(T^j_{\text{all,i}} - T^j_{\text{all,o}}\right) \\ \frac{dT^j_{\text{wall,i}}}{dt} &= \frac{2h_i A^j_{\text{wall}}}{C^j}\left(T_{\text{room}} - T^j_{\text{all,i}}\right) + \frac{2}{R^j_{\text{wall}} C^j}\left(T^j_{\text{all,o}} - T^j_{\text{all,i}}\right)\end{aligned} \tag{12.5}$$

Here, R^j_{wall} is the thermal resistance of the j-th wall and C^j the thermal capacitance of the j-th wall.

The model describing the CO_2 concentration in the room is modeled as depicted in Eq. (12.6).

$$V\frac{dC_{\text{CO2}}}{dt} = \left(\dot{m}_v C_{\text{CO2,i}} + g_{\text{CO2}} N_{\text{people}}\right) - \dot{m}_v C_{\text{CO2}} \tag{12.6}$$

The CO_2 concentration in the room is denoted by C_{CO2}, the CO_2 concentration of the air entering the room from the ventilation system by $C_{\text{CO2,i}}$ and the number of people by N_{people}.

Equations (12.4), (12.5) and (12.6) constitute the simulation model. Note that this model is bi-linear, since the input signals $\dot{m}_v$ and T_{supply} are multiplied.

The model equations were implemented into Matlab by means of S-functions, and used in Simulink as a simulation model.

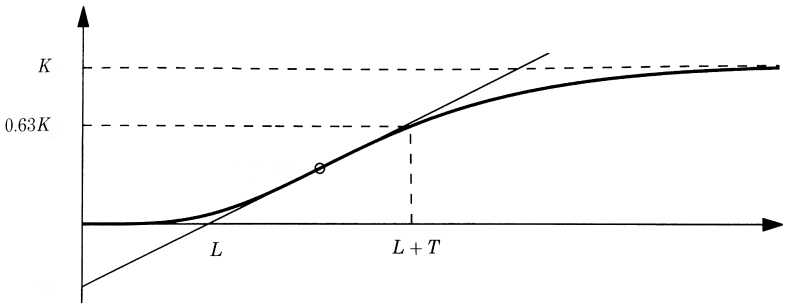

Figure 12.4 A sketch of an example step response of a process. The sketch illustrates how the parameters K, L, T of the FOTD model in Eq. (12.7) are estimated. Source: [Astrom and Hagglund, 2006]

12.3 Estimation of Simple Models

First-order models describing the dynamics of the system shown in Figure 12.3 were estimated from step response data measured either using the simulation model or on the test-bed. These models were then used to form the decoupling transfer functions.

To estimate the first-order models, a method to estimate FOTD (first-order-time-delay) models from [Astrom and Hagglund, 2006] was used. From the data of a step-response, the static gain K, the apparent time delay L and the average residence time T_{ar} were determined. The apparent time delay is the time where the steepest tangent on the step-response crosses the steady-state value, which the systems output had before the step. The average residence time is the time where the step response reaches 63% of its final value, it is also $T_{ar} = T + L$ (see also Figure 12.4). The resulting FOTD model is as shown in Equation (12.7).

$$G(s) = \frac{K}{1 + sT} e^{-sL} \tag{12.7}$$

Here, the FOTD model was approximated as a first order model without a time delay.

12.4 Decoupling for the Ventilation System

A decoupling transfer function was used here to decrease the coupling between the mass flow rate of air from the ventilation system and the room temperature. An increase in the mass flow rate of the ventilation air, for example because the CO_2 concentration needs to be decreased, has

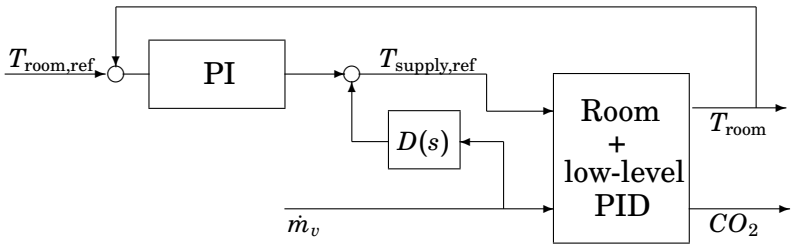

Figure 12.5 A block diagram showing the set-up for the decoupled PI controller. The block named "Room + low-level PID" is the system shown in Fig. 12.3. The PI controller controls the room temperature T_{room} by determining a reference supply air temperature for the low-level PID controller $T_{\text{supply,ref}}$. The decoupling transfer function is $D(s)$.

an influence on the room temperature. In the winter season, when the outside air temperature is significantly cooler than the room temperature, an increase in the mass flow rate will decrease the room temperature, which is not desired. However, a change in supply air temperature does not have an influence on the CO_2 concentration. This can also be seen in Equations (12.4), (12.5) and (12.6).

Hence, one-directional decoupling was used here. One-directional decoupling means that only one of the two decoupling transfer functions in a 2-by-2 system was used. The decoupling network $D(s)$ used for the ventilation system is shown in Figure 12.5. The decoupling transfer function takes the mass flow rate $\dot{m}_v$ as an input and its output is added to the supply air temperature $T_{\text{supply,ref}}$ determined by the PI controller.

Note that there was no controller for the CO_2 concentration. Since the experiments on the test-bed were performed during student holiday time, there was very little disturbance on the CO_2 concentration. A large enough CO_2 concentration disturbance would have been needed to get a change in mass flow rate $\dot{m}_v$ large enough to see the effect of the decoupling transfer function. Hence, the mass flow rate was changed manually instead of being the output of a CO_2 concentration controller.

With G_{11} being the transfer function from the supply air temperature T_{supply} to the room temperature T_{room}, and G_{12} being the transfer function from the mass flow rate $\dot{m}_v$ to the room temperature T_{supply}, the decoupling transfer function is as shown in Eq (12.8). For details on the decoupling transfer function see Section 10.2.

$$D(s) = \frac{G_{11}(s)}{G_{12}(s)} \tag{12.8}$$

The first-order models estimated as described in Section 12.3 were used for $G_{11}(s)$ and $G_{12}(s)$.

12.5 Discrete PI Controller and Decoupling Network

For implementation into the KTH test-bed system, a discrete controller was used. The decoupling network was discretized using backwards differences.

With a first-order model for the transfer functions $G_{11}(s)$ and $G_{12}(s)$, the decoupling transfer function $D(s)$ as shown in Eq. (12.8) has a form as shown in Eq. (12.9).

$$D(s) = \frac{b_1 s + b_0}{a_1 s + a_0} \tag{12.9}$$

Denoting the input signal of the decoupling network by u_2, and the output signal by c_D, the time domain representation of the decoupling transfer function in Eq. (12.9) is

$$a_1 \dot{c}_D(t) + c_D(t) = b_1 \dot{u}_2(t) + b_0 u_2(t)$$

To discretize the above equation, backwards differences were used for the derivatives.

$$\dot{c}_D(t) \approx \frac{c_d(k) - c_D(k-1)}{h}$$
$$\dot{u}_2(t) \approx \frac{u_2(k) - u_2(k-1)}{h}$$

Discretization with sampling time h, using the finite differences above, results in the discrete decoupling transfer function shown in Eq. 12.10.

$$c_D(k) = \frac{b_1 + b_0 h}{a_1 + a_0 h} u_2(k) - \frac{b_1}{a_1 + a_0 h} u_2(k-1) + \frac{a_1}{a_1 + a_0 h} c_D(k-1) \tag{12.10}$$

The output c_D of the decoupling transfer function gets subtracted from the supply air temperature T_{supply} determined by the temperature PI controller. The input u_2 to the decoupling transfer function is here the mass flow rate $\dot{m}_v$.

The discrete time temperature PI controller [Åström and Wittenmark, 1997] implemented is shown in Eq. 12.11. The variable to be controlled was the room temperature $T_{\text{room}}(k)$ and the output of the controller is the supply air temperature $T_{\text{supply,ref}}(k)$. The sampling time is denoted by h.

$$\begin{aligned} e(k) &= T_{\text{room,ref}}(k) - T_{\text{room}}(k) \\ T_{\text{supply,ref}}(k) &= K_p e(k) + K_i e_I(k) + T_{room,ref}(k) \\ e_I(k+1) &= e_I(k) + e(k) \cdot h \end{aligned} \tag{12.11}$$

The reason, that the reference temperature to the room $T_{\text{room,ref}}(k)$ was added to the PI controller was to drive the room temperature to its reference value faster.

12.6 Simulations in Matlab

The decoupling PI controller described in Section 12.4 was tested in a simulation study using the model described in Section 12.2. The simulations were done in Matlab/Simulink, where the simulation model was implemented as an S-function and the decoupled PI controller as a Matlab function in the discrete version from Section 12.5.

In this section, first the estimation of first-order models from step-response data is described. Then the setup and the results of the simulations are given.

Model Estimation

First-order models describing the dynamics from the supply air temperature T_{supply} to the room temperature T_{room} and from the air flow rate $\dot{m}_v$ were estimated using step-response tests as described in Section 12.3. These models were used to form the decoupling network transfer function.

The data recorded at the simulation model and used to estimate the first-order models is shown in Fig. 12.6 and Fig. 12.7.

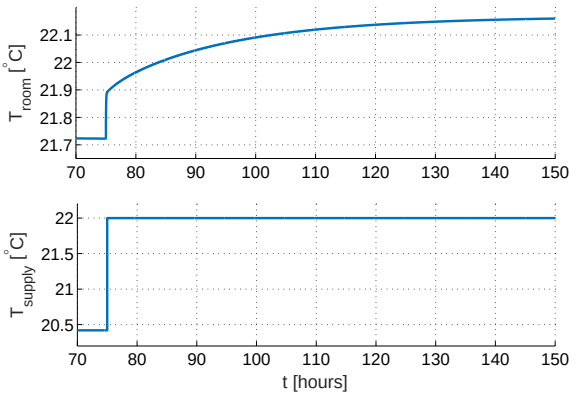

Figure 12.6 Step-response data measured at the Matlab simulation model to estimate G_{11}.

The data in Fig. 12.6 were used to estimate the dynamics from the supply air temperature to the room temperature, which was captured by the first order transfer function $G_{11}(s)$. For this, the air flow rate was kept constant at $\dot{m}_v = 0.2$ [kg/s]. A step from 20.42 °C to 22 °C was applied to the supply air temperature T_{supply}. The room temperature was recorded.

The data in Fig. 12.7 were used to estimate the dynamics from the air flow rate to the room temperature, captured by the first order transfer

function $G_{12}(s)$. This time, the supply air temperature was kept constant at $T_{\text{supply}} = 20.42$ °C. A step from 0.12 to 0.33 [kg/s] was applied to the air flow rate $\dot{m}_v$. The room temperature was recorded.

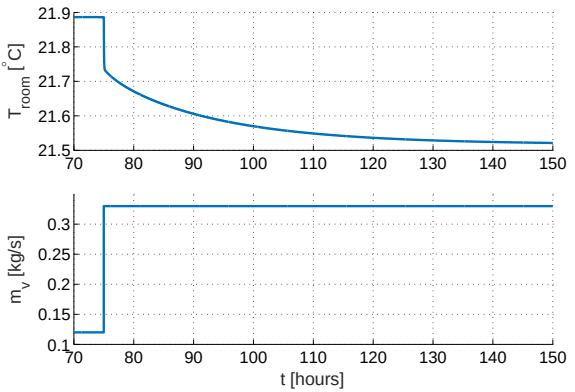

Figure 12.7 Step-response data measured at the Matlab simulation model to estimate G_{12}.

Since the first-order models G_{11} and G_{12} are supposed to describe the change in room temperature that happens after a change in the respective input signals air flow rate or supply air temperature, the data were prepared for estimation by setting the time, when the step in the respective input signal happened, to time zero. Furthermore, the offsets at the new zero time for input and output data were removed.

It can be seen in Figures 12.6 and 12.7, that there are two different time constants governing the dynamics. After approximately 150 hours a steady-state is reached in the step-response. This corresponds to the long time constant, which covers the temperature dynamics of building envelope elements like walls, the floor or the ceiling. The fast time constant, in the size of 25 minutes, is visible in the figures by an initial fast rise of the step-response. This corresponds to the temperature dynamics of the air in the room.

Using the method described in Section 12.3, G_{11} and G_{12} were estimated to be

$$G_{11} = \frac{0.11}{0.04134s + 1} \tag{12.12}$$

$$G_{12} = \frac{-0.7533}{0.03903s + 1} \tag{12.13}$$

Using the decoupling transfer function in Eq. (12.8) and the estimated

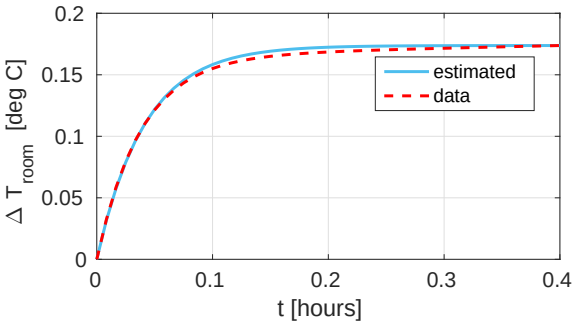

Figure 12.8 Comparison of the model output from the estimated G_{11} to the measured simulation data.

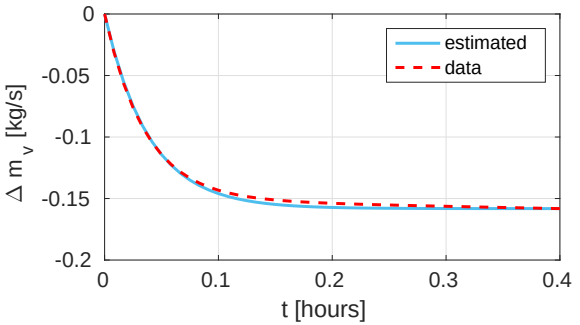

Figure 12.9 Comparison the model output from the estimated G_{12} to the measured simulation data

models above lead to the following transfer function for decoupling:

$$D = \frac{0.0166s + 0.4211}{0.004131s + 0.1072} \tag{12.14}$$

The step input signals shown in Fig. 12.6 and Fig. 12.7 were applied on G_{11} and G_{12}, and the resulting outputs were compared to the recorded output data. This comparison is shown in Fig. 12.8 and Fig. 12.9 for G_{11} and G_{12}, respectively. In both cases, a goodness of fit of 93% could be reached, using the normalized root mean square error as a measure.

Note that here, the first-order models were estimate to fit the faster dynamics in the step-response data, capturing the temperature dynamics of the rooms air.

Simulation Setup

For simulation purposes, an outside temperature of 21.9 °C was assumed, since the experiments on the test-bed were carried out during the summer season. Also the temperature in the adjacent rooms was assumed to be at 21.9 °C. The reference for the room temperature was set to 21.9 °C. The outside CO_2 concentration was assumed to be 400 ppm. The number of occupants was set to $N_{people} = 1$. The simulation time was 4 hours.

The limitations on the supply air temperature and the air flow rate present in the KTH test-bed were taken into account in the simulation. Hence, it was $T_{\text{supply}} \in [20.5, 22.05]^{\circ}C$ and $\dot{m}_v \in [0.1, 0.34]$ [kg/s]. The parameters for the PI controller were set to $K_p = 15.32$; and $K_i = 302.4$;. These were determined using the Ziegler Nichols tuning method in Matlab.

Initially, the air flow rate $\dot{m}_v$ was set to 0.12 [kg/s]. The PI controller then regulates the room temperature to its reference value using the supply air temperature T_{supply}. After 2 hours, the air flow rate is increased to 0.33 [kg/s]. The PI controller can now re-adjust the supply air temperature, so that the room temperature follows its reference.

The idea with the decoupling network is, that the room temperature will deviate less from its reference after the step change in the air flow rate using the decoupling network compared to without it.

Simulation Results

The results of the simulation are shown in Fig. 12.10. The first panel shows the room temperature, the second panel shows the supply air temperature of the ventilation system and the third panel the air flow rate of the air in the ventilation system. The red solid lines are the simulation results with the decoupling network, and the blue dash-dotted lines are the results without using the decoupling network.

Without the decoupling network, the room temperature decreases approximately $0.06^{\circ}C$, before the PI controller changes the supply air temperature and brings the room temperature back to its reference value. Using the decoupling network with the PI controller, it can be seen that the room temperature returns to its reference value sooner, and that the deviation from the reference value is less. The deviation from the room temperature is now reduced to $0.01^{\circ}C$. Looking at the control signal T_{supply}, it can be seen that with the decoupling network the control signal reacts slightly faster than without using the decoupling network. Even though without the decoupling network, the deviation from the reference value of the room temperature is small, a significant decrease in deviation from the reference value can be observed when using a decoupling transfer function with the PI controller.

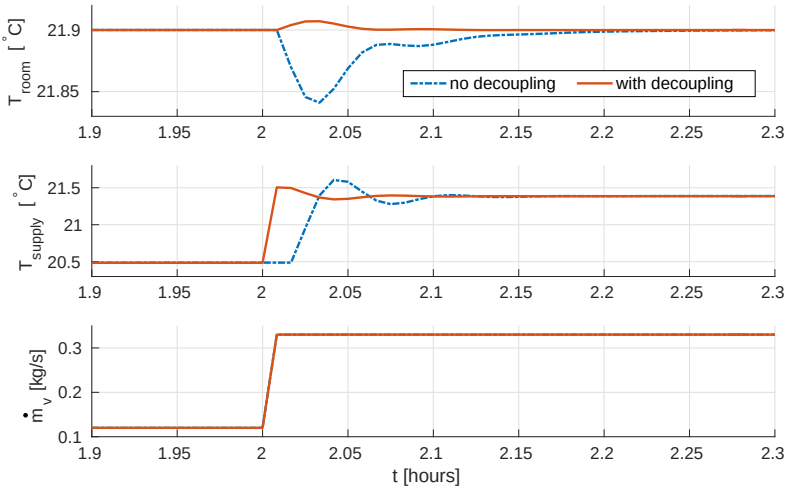

Figure 12.10 Simulation of the KTH test-bed with and without a decoupling network. Blue: no decoupling network; red: with a decoupling network. After a step in the mass flow rate $\dot{m}_v$, the room temperature T_{room} deviates less from its reference temperature with using a decoupling network compared to when not using a decoupling network. Using the decoupling network, the control signal T_{supply} reacts earlier than without a decoupling network.

12.7 Experiment using the KTH test-bed

This section describes how the decoupled PI controller was tested on the test-bed at KTH described 12.1. As in the case with the simulation model, first-order models describing the dynamics of the system to be decoupled were estimated at first. Then these models were used to form the transfer function for the decoupling network. This decoupling network and a PI controller were then implemented into the object-oriented Matlab environment, which was installed for testing new control algorithms for the test-bed. Using this, the proposed decoupled PI controller was tested and compared to the case without a decoupling network.

The Estimated Models

The first-order models used for the decoupling transfer function were estimated from step-response data. The estimation was done as described in Section 12.3. The data was collected during the summer season in June 2015.

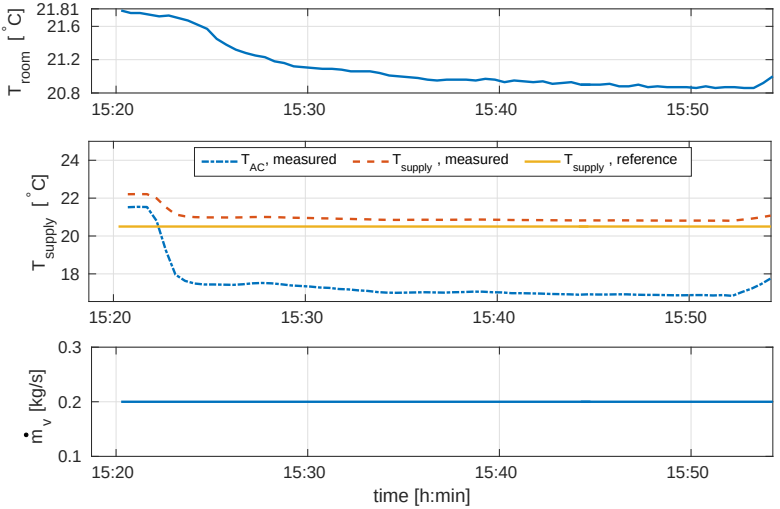

Figure 12.11 Data collected to estimate a first-order model from $T_{\text{supply,ref}}$ to T_{room}. The first panel shows the measured room temperature. The second panel shows the reference supply air temperature $T_{\text{supply,ref}}$ (solid line), the actual measured supply air temperature T_{supply} (dashed line), and the measured AC temperature T_{AC} (dash-dotted line). $T_{\text{supply,ref}}$ is a step from 22 °C to 20.5 °C. The first data point was not measured.

For estimation of the model G_{11} from the supply air temperature to the room temperature, the data shown in Fig. 12.11 was used. A step from 22 °C to 20.5 °C was applied to the reference of the internal PI controller, i.e. to $T_{\text{supply,ref}}$. The air flow rate was kept constant. Note that the first data point where $T_{\text{supply,ref}} = 22°C$ was not measured, and can thus not be shown in this figure. Fig. 12.11 shows the room temperature in the first panel and the mass flow rate in the third panel. The second panel shows $T_{\text{supply,ref}}$, which is given as a reference to the internal PI controller, and also the actual, measured supply air temperature and the AC temperature T_{AC}. During the time the measurements were taken, the outside temperature was in the range of 19.6 °C and 20.1 °C.

The data used to estimate the model G_{12} from the air flow rate $\dot{m}_v$ to the room temperature is shown in Fig. 12.12. The room temperature is shown in the first panel, and the mass flow rate in the third. The second panel shows the reference supply air temperature to the internal PI controller, the measured supply air temperature and the AC temperature. Here, the supply air temperature was held constant, while the air flow rate was increased from 0.12 to 0.33 [kg/s]. The outside temperature was in the range of 18 °C

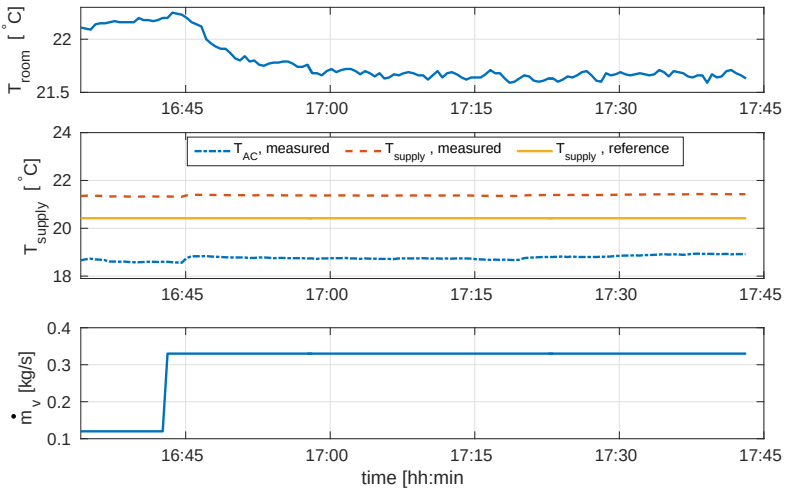

Figure 12.12 Data collected to estimate a first-order model from $\dot{m}_v$ to T_{room}. The first panel shows the measured room temperature. The second panel shows the reference supply air temperature $T_{\text{supply,ref}}$ (solid line), the actual measured supply air temperature T_{supply} (dashed line), and the measured AC temperature T_{AC} (dash-dotted line). The air flow rate $\dot{m}_v$ is a step from 0.12 to 0.33 [kg/s]

to 20.5 °C.

As was done for the tests with the simulation model, the data are prepared for estimation by setting the time, when the step in the respective input signal happens, to time zero and removing the offsets at the new zero for input and output data.

By using the method described in Section 12.3, first-order plus time delay (FOTD) models were estimated for G_{11} and G_{12} to fit the data shown in Fig. 12.11 and 12.12. This was done, since the data seemed to indicate that a time delay was present.

The FOTD models estimated for G_{11} and G_{12} from the data are shown in Eq. (12.15) and Eq. (12.16), respectively. Note that the unit for the time constants is hours in all following equations.

$$G_{11}(s) = \frac{0.6067}{1 + s \cdot 0.11} e^{-s \cdot 0.03} \tag{12.15}$$

$$G_{12} = \frac{-2.57}{0.12 \cdot s + 1} \cdot e^{-0.04 \cdot s} \tag{12.16}$$

Approximating the FOTD models by a first order model without time

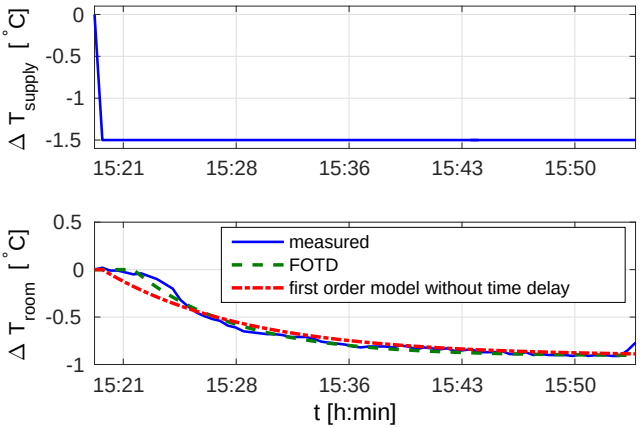

Figure 12.13 Comparison of the estimated FOTD model and the first order model without time delay for G_{11} to the measured room temperature T_{room}. The supply air temperature T_{supply} is decreased by $1.5°C$.

delay gives the models in Eq. 12.17 and Eq. (12.18) for G_{11} and G_{12}, respectively.

$$\bar{G}_{11}(s) = \frac{0.61}{1 + s \cdot 0.15} \tag{12.17}$$

$$\bar{G}_{12} = \frac{-2.57}{0.13 \cdot s + 1} \tag{12.18}$$

To test the estimated models, the input data from the estimation data sets in Fig. 12.11 and Fig. 12.12 were applied to G_{11} and G_{12}, respectively. The output data produced by these transfer functions was then compared by the measured output data. Figure 12.13 shows the output data generated by the estimated FOTD and first-order with no time delay models compared to the measured output data for G_{11}, and Fig. 12.14 shows the same for G_{12}.

As seen in figures 12.13 and 12.14, the first-order models without a time delay resulted in a fit comparable to that of the FOTD models. Hence, the first order models without time delay were chosen to be used for the decoupling network.

The reason that the models with the time-delay do not lead to the better fit than the models without is due to the placement of the used temperature sensor in comparison to the air outlet of the ventilation system. The sensor used for temperature measurements is placed in the center of the room under the ceiling, in close proximity to the air outlet of the ventilation system. Hence, the influence of any change in supply air temperature or air

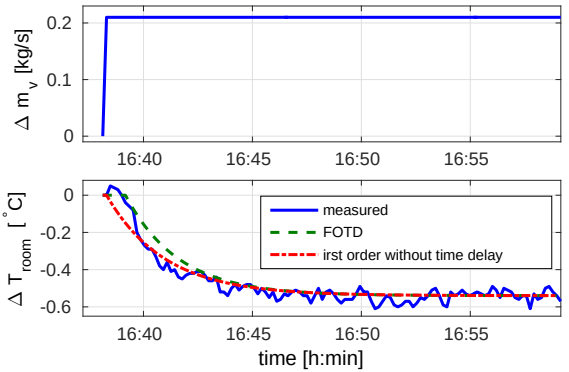

Figure 12.14 Comparison of the estimated FOTD model and the first order model without time delay for G_{12} to the measured room temperature T_{room}. The supply air temperature T_{supply} is decreased by $1.5°C$.

flow rate on the measured room temperature has a negligible small time delay.

Using the first-order models resulted in the decoupling transfer function as shown in Eq. (12.19).

$$D(s) = \frac{-0.39 \cdot s - 2.57}{0.08 \cdot s + 0.61} \tag{12.19}$$

Experiment Setup and Results

The experiments to test the decoupled PI controller on the test-bed were carried out in July 2015, during summer season. During that time, the outside temperature was in the range of 17.4 and 19.6 °C.

Using the estimated first-order models, the parameters of the PI controller for G_{11} were tuned using the Ziegler-Nichols tuning rules in Matlab. The resulting parameters used in the tests were $K_P = 2.12$ and $K_i = 0.004$. The reference temperature for the room was set to 21.9 °C.

To test the effect of the decoupling network, the air flow rate was increased manually from 0.12 to 0.33 $[kg/m^3]$. The resulting room temperature was then recorded and observed. For evaluation purposes, the AC temperature T_{AC} and the CO_2 concentration were recorded as well.

Figure 12.15 shows the control signals $\dot{m}_v$ and T_{supply} and the measured room temperature T_{room} and CO_2 concentration. The room temerature is shown in the upper left panel. The lower left panel shows the supply air temperature as determined by the PI controller, which regulates the room temperature. The air flow rate $\dot{m}_v$ is shown in the lower right panel. The upper right panel shows the CO_2 concentration. This is not used in the

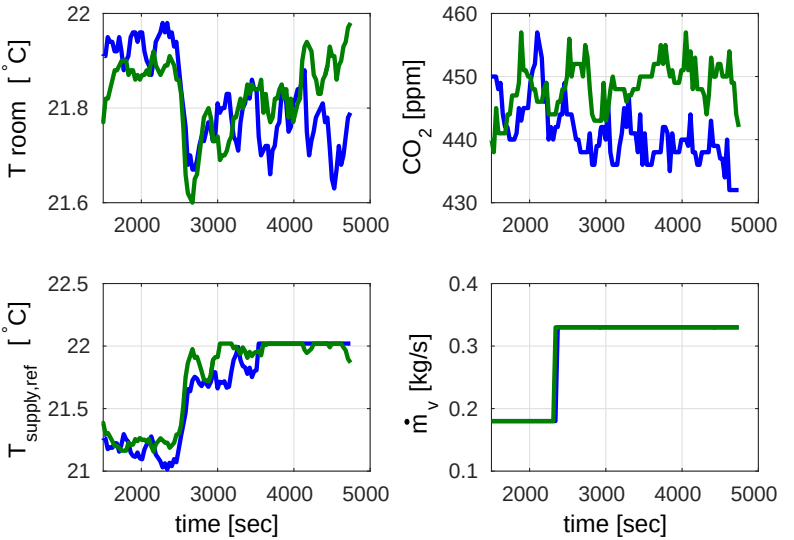

Figure 12.15 Comparing the decoupled PI controller with the case without a decoupling network. The data was recorded in July 2015 on the KTH testbed. The upper left panel shows the measured room temperature. The lower left panel shows the supply air temperature determined by the controller. The upper right panel shows the measured CO_2 level. The lower right panel shows the mass flow rate of ventilation air. In the case where the decoupling transfer function was used, the room temperature returned to its reference temperature faster as when no decoupling was used.

experiments, but shown here for the sake of completeness. In all panels, the green line shows the case with decoupling, and the blue line shows the case without decoupling.

For both cases, with and without decoupling, the room temperature was initially at approximately $21.9°C$. When the air flow rate was increased manually, the room temperature dropped about $0.2°C$, and the supply air temperature was increased by the PI controller to drive the room temperature back to its reference value of $21.9°C$. It seems from these results, that with the decoupling network, the room temperature returns to its reference value quicker than without a decoupling network. These results would however have to be tested by more experiments. The supply air temperature is at its maximum saturation level during the second half of the experiments. Also, the drop in room temperature is less than one degree Celsius. With an improved experiment setup, leading to a larger drop in room temperature, and avoiding saturation, better results could be achieved.

13

Discussion

Inverted decoupling was applied to two different examples in building automation to reduce interactions between some of these variables. The first example was four rooms arranged in a square with coupled temperature dynamics. The purpose was to reduce the effects of these couplings through inverted decoupling. The second example was the regulation of the room temperature using an air conditioning system, when the air flow rate in the air conditioning system was changed. This is a one-directional coupling, where the change of air flow rate influences the room temperature. This influence was to be decreased using inverted decoupling.

In both of the examples, inverted decoupling could decrease the effects of interactions in simulation studies using non-linear simulation models, even though only first-order models were used for the decoupling transfer functions. These models were estimated using step-response data, either recorded from the simulation models or measured at the KTH test-bed.

[Shen et al., 2010] and [Garrido et al., 2011] also applied a decoupling method to an example from building estimation, finding that decoupling could reduce or even cancel out present interactions. In these papers, the rooms temperature of four rooms was regulated using variable air volume dampers in an air conditioning system, which lead to cross-couplings. [Garrido et al., 2011] used inverted decoupling as well, having a system of first-order plus time delay models for simulations. [Shen et al., 2010] tested a decoupling method based on equivalent transfer function methods on a lab-size process, but the used decoupling method is not using PID controller.

However, decoupling control is not very commonly found in buildings automation. Instead, it is mainly used in process control and chemical engineering. Inverted decoupling is often used for distillation processes [Seborg et al., 2010]. Searching with the search engine Scopus [Elsevier, 2016] for "inverted decoupling" gave 24 search results in English language. Most of these article were in journals within chemical engineering, and none of them in building automation.

In this thesis, inverted decoupling was used to reduce interactions for the two examples described above. Even though interactions could be decreased, they were not canceled out completely. The reason for this is that the process to be controlled, and the used simulation models, were not linear. In the case fo the adjacent rooms, the simulation model was a complex model based on multiple physical domains, taken from the Modelica Buildings Library. For the ventilation system at KTH, the simulation model was bi-linear, and experiments were performed on a real test-bed. The models used for the transfer functions of the decoupling network on the other hand were chosen to be first order models.

When using models of different orders for all the transfer functions in the decoupling network, one has to take realizability into consideration, so that none of the transfer functions in the decoupling network has more zeros than poles or an unrealistic time delay. In [Garrido et al., 2011], an extended version of inverted decoupling was presented. With this approach, there is more flexibility for the transfer functions used for the decoupling network. On the other hand, the configuration of the decoupling transfer function needs to be changed and adjusted to the model describing the process dynamics. Using the same number of poles and zeros for all transfer functions estimated for the system's dynamics, the problem of realizability can be circumvented. Here, the choice of first-order models is motivated by the idea of having as simple models as possible, and to have a simple way of identifying model parameters. In this thesis, the first-order models were estimated using step-response tests. However, it is possible to estimate firs-order models using auto-tuning methods, originally developed to tune PID controllers [Berner et al., 2014]. However, realizability can be achieved with higher order models as well.

In fact, the step-response data measured for both the adjacent rooms and the simulations of the KTH test-bed indicated that there are two time constants governing the dynamics of the systems. In Fig. 11.4 for example, it can be seen that the step-response in made up of one part corresponding to a faster time constant, and another part corresponding of a slower time constant of several hours. For the KTH ventilation system simulations it is similar, compare to figures 12.6 and 12.7. In both cases, the slow time constant corresponds to the temperature dynamics of the building envelope, consisting of walls, ceiling and floor. The faster time constant corresponds to the heating of the air inside the room. The first-order models only take into account one of these time constants. In the case of the four adjacent rooms, the models were estimated to model the dynamics of the slower time constant. This was because the temperature dynamics of the building envelope was important for the coupling of the temperature dynamics between the rooms. For the KTH ventilation system, the first-order models were estimated to use the faster time constant, since the

heating of the room air was considered.

Using models of higher order, which more accurately describe the process dynamics might improve the decoupling. In the case here, estimating second order models taking into account both the slow and the fast time constant present in the step-response would improve the goodness of fit of the model compared to the measured output data, and hence lead to better decoupling. When all parts of the multi-variable system are estimated through second-order transfer functions, realizability is still given.

In [Seborg et al., 2010] it is stated, that decoupling methods are not very useful, since there are no models accurate enough to describe a process. However, the simulation results in this thesis show that it is possible to decrease interactions in a non-linear process by using first-order models for the decoupling transfer functions. Using second-order models instead, the decoupling could even be improved further.

For both the adjacent rooms and the KTH ventilation system, the coupling of the temperature dynamics did not lead to a very large change in temperature dynamics. For the adjacent rooms for example, the change in room temperature following a change in heat flow into a neighboring room was $0.7°C$. Nevertheless, using the decoupling transfer functions could decrease the effects of the coupling in the simulation studies.

For the experiments done on the KTH test-bed, the results showed that the room temperature returned to its reference value slightly faster using the decoupling network, than when decoupling is not used. The size of the deviation from the reference temperature is however the same for both cases. One reason might be that the deviation of the room temperature is only $0.2°C$. One reason for this is that the experiments were done during the summer season, when the outside temperature was between $16°C$ and $20°C$. Moreover, the temperature of the air in the ventilation system is in the range of approximately $20°C$ to $22°C$, which gives little flexibility to have larger deviation of the room temperature by just increasing the flow rate of the air. Also, the temperature sensor used is in close proximity to the air outlet of the ventilation system, so that a change in supply air temperature has an almost immediate effect on the measured room temperature. Another issue, as can be seen in Fig. 12.15, was that the supply air temperature determined by the controller and the decoupling network was saturated for almost half the experiment time. This is the same period when the room temperature deviates between the cases with and without decoupling. Furthermore, it was assumed for the experiments done that the low-level PI controller regulating the AC temperature T_{AC} is regulating perfectly, i.e., that $T_{AC,ref} = T_{AC}$. As seen in for example in Fig. 12.11, the reference given to the low-level controller is not reached. For the estimation data in Fig 12.11, there is a remaining error of approximately $0.8°C$, which is more than the deviation of the room temperature from its reference. Despite

that, the simulation study for the KTH test-bed showed a reduction in the interactions using inverted decoupling. An improvement of the experiment setup can be valuable to show the effect of decoupling even in a real setting.

As mentioned above, decoupling methods are not very common in building automation research. The most popular method in the research community to coordinate different variables is MPC. MPC has the advantage of, e.g., taking into account weather predictions, occupant schedules and other disturbances. However, MPC needs a complex model of higher than first order to describe building dynamics, which needs to be identified for every specific building. Furthermore, the necessary change in the set-up of a building with an existing controller, maybe additional sensors, add to the cost of installing such a system on an existing building.

In this thesis, the idea was to use a decoupling scheme in order to upgrade an already existing control structure with PID controller and rule-based supervisory control. An advantage of inverted decoupling, as compared to ideal or simplified decoupling is that the same controller as for the SISO process, including anti-windup, can be used [Gagnon et al., 1998].

To summarize, inverted decoupling was used for two different applications in building automation. Using first-order models to describe the process dynamics of non-linear simulation models, inverted decoupling could reduce interactions between the control variables.

The simplicity of the implementation and design of inverted decoupling makes it attractive to combine with already existing PI controllers on buildings components and other low-level controllers, making it easier for a supervisory controller to coordinate all the different parts of a building automation system.

14

Conclusion and Future Work

Inverted decoupling was applied to two different examples in the area of temperature control in buildings in the second part of the thesis.

The first example was a nonlinear model of four adjacent rooms with coupled temperature dynamics. The simulation model was implemented using the Modelica Buildings Library, which gives a complex model based on multiple physical domains. In this example, the goal was to eliminate the couplings in temperature dynamics between the rooms using the inverted decoupling method. Since the simulation model was approximated by simple first order models, perfect elimination was not possible. However, despite the very simple first-order approximation, the effect of the couplings could be reduced using inverted decoupling.

The second example concerned the control of room temperature using the air flow rate and the air temperature in a ventilation system. The coupling in this example was from the change in air-flow rate to the room temperature. The room temperature was to be regulated using the temperature of the air in the ventilation system. However, a change in the flow-rate of this air had an influence on the room temperature as well. As in the previous example, the coupling could not be taken away completely. However, in a simulation study the coupling from the air flow rate to the room temperature could be reduced using inverted decoupling.

Future Work

Considering the results in this part of the thesis, the possibility to use inverted decoupling in building automation seems promising. There are several directions in which the research can be continued. Some of these directions are as follows:

Models of Second Order: Fast and Slow Dynamics As discussed in Section 11.2 for example, the step-response test data showed that there are two different time constants involved in the dynamics. This seems reasonable, since the air in a room gets heated up much quicker than the building envelope, when heat is added to it. To cover both the slow and the fast dynamics, models of the following structure could be estimated instead of first-order models:

$$G = \frac{K_1}{1 + sT_1} + \frac{K_2}{1 + sT_2} \tag{14.1}$$

The fast dynamics are represented by the time constant T_1, and the slow dynamics are represented by the time constant T_2. This model has one zero and two poles. The estimation of the model parameters could be done with the step-response data used in this thesis, by considering it as two super-imposed step responses.

As an example, a second order transfer function as in Eq. (14.1) was estimated for G_{11} in Chapter 11, i.e., for the dynamics from the heat flow into room 1 to the temperature in room 1 for the four adjacent rooms. The second order model in Eq. (14.1) is a sum of two first-order models. The estimation method for first-order models described in Chapter 11 was therefore used. Figure 14.1 shows the recorded output data (blue solid line), the output data generated by the first order model estimated in Chapter 11 (greed dashed) and the output data generated by the second-order model according to Eq. (14.1) (red dash-dotted). It can be seen that the second-order models has a much better fit compared to the first-order model.

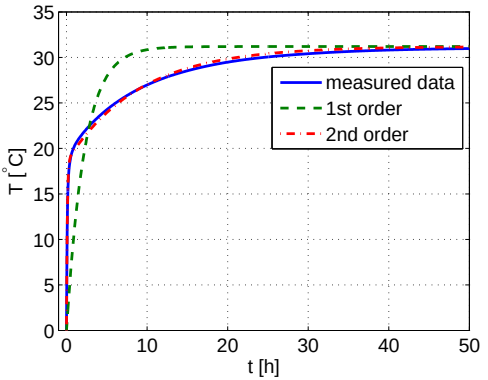

Figure 14.1 Comparison of a first-order and second-order model estimated for G_{11} in Chapter 11. It can be seen, that the second-order model has a better fit with the measured data than the first-order model.

Because of the improved model fit, it is expected that the decoupling

would be improved using second-order models instead of first-order models to describe the temperature dynamics.

Using Eq. (14.1) for all transfer functions $G_{ij},\ i,j = 1,...,4,$ in Eq. (10.4), the decoupling network's transfer functions are made up of third order transfer functions having three poles and three zeros. The decoupling network is thus realizable. Looking at Fig. 11.4, it appears that the use of second order models is most crucial for $G_{ii},\ i = 1,...,4$. When using second order models as in Eq. (14.1) above for the $G_{ii},\ i = 1,...,4,$ in the decoupling network in Eq. (10.4), and first order models for all other transfer functions $G_{ij},\ i,j = 1,...,4,\ i \neq j,$, the decoupling network is made up from second order transfer functions with two zeros and two poles, which is realizable as well. With this, better decoupling effect is expected.

Alternative Experiment Setups for the Ventilation System An update of the experiment setup for the ventilation system in the KTH test-bed could improve the results concerning the effect of the inverted decoupling method.

As seen in the experiment results in Fig. 12.15, the control signal $T_{\text{supply,ref}}$ reached its upper saturation level after approximately half the simulation time. An adjustment of initial values of all used signals, so that the control signal stays within its boundaries, will make the results clearer.

The deviation of the room temperature from its reference value is only ca. $0.2°C$. An increase in this deviation when no decoupling is used can make the effect of the decoupling network more visible. One way to achieve that could be to use a sensor at a different location in the room. In the room used for experiments in KTH test-bed, there are sensors at all walls in addition to the temperature sensor in the center of the room. In this way, a temperature sensor further away from the air outlet of the ventilation system, leading to slower dynamics, could help increase the deviation of the room temperature.

Furthermore, using the temperature of the ventilation air for heating limits the possibilities for heating, since it is in the range o approximately $20°C$ to $22°C$. Performing experiments during winter season, instead of summer season, enables the use of radiators for heating.

Improving the Modelica Simulation Model The simulations for the adjacent rooms are done using the Modelica Buildings Library version 1.6. The room model available in this version of the library was a completely mixed room. A room model that also includes the air flow in the room is available in newer version of the Buildings Library.

Furthermore, the integration of a heating system, for example radiators, in the simulation model would be interesting to make the simulations even more realistic.

Decoupling of Several Rooms in a Row In this thesis, four adjacent rooms were used to test the inverted decoupling method. This leads to a 4-by-4 decoupling matrix shown in Eq. (10.4). For larger buildings with more rooms, the decoupling system will get more complex, with communications needed between all the rooms. However, when the temperature is changed in one of the rooms, the effect of this on the temperature in rooms further away will be neglectable.

Considering a row of more than two rooms as an example, a change of room temperature in the first room of the row will have an insignificant effect on, e.g., the 5th room of the row. Probably, a significant effect would be measured only on the neighboring second room. In this case, the decoupling matrix $D(s)$ in Eq. (10.4) would be a banded matrix, with the entries on the secondary diagonal being the only non-zero ones. It would be interesting investigating how well the decoupling works with this setting. This would have the advantage of easier implementation for larger buildings.

Model Predictive Control Model predictive control is viewed to be the most suitable control method for the control climate and energy in buildings. It would be interesting to implement Model Predictive Control into the Modelica model of the four adjacent rooms, in order to compare it to decoupled PI controllers proposed in this thesis.

Bibliography

Afram, A. and F. Janabi-Sharifi (2014). "Theory and applications of hvac control systems–a review of model predictive control (mpc)". *Building and Environment* **72**, pp. 343–355.

Astrom, K. J. and T. Hagglund (2006). *Advanced PID control*. Isa.

Åström, K. J. and B. Wittenmark (1997). *Computer Controlled Systems*. Prentice Hall.

Berner, J., K. J. Åström, and T. Hägglund (2014). "Towards a new generation of relay autotuners". In: *19th IFAC World Congress*. Cape Town, South Africa.

Chen, P. and W. Zhang (2007). "Improvement on an inverted decoupling technique for a class of stable linear multivariable processes". *ISA transactions* **46**:2, pp. 199–210.

Dassault System (2016). *Dymola*. `http://www.3ds.com/products-services/catia/products/dymola`. Online, accessed March 2016.

Doukas, H., K. D. Patlitzianas, K. Iatropoulos, and J. Psarras (2007). "Intelligent building energy management system using rule sets". *Building and environment* **42**:10, pp. 3562–3569.

Dounis, A. I. and C. Caraiscos (2009). "Advanced control systems engineering for energy and comfort management in a building environment - a review". *Renewable and Sustainable Energy Reviews* **13**:6, pp. 1246–1261.

Elsevier (2016). *Scopus*. `https://www.scopus.com/`. Online, accessed May 2016.

Gagnon, E., A. Pomerleau, and A. Desbiens (1998). "Simplified, ideal or inverted decoupling?" *ISA Transactions* **37**:4, pp. 265–276.

Garrido, J., F. Vázquez, F. Morilla, and T. Hägglund (2011). "Practical advantages of inverted decoupling". *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* **225**:7, pp. 977–992.

Hazyuk, I., C. Ghiaus, and D. Penhouet (2012). “Optimal temperature control of intermittently heated buildings using model predictive control: part ii–control algorithm”. *Building and Environment* **51**, pp. 388–394.

Holmberg, F. (2001). “Implementation of a pid controller for building automation”. *MSc Theses*.

Jensen, L. (1976). “Digital reglering av klimatprocesser”. *Report TFRT*.

Jensen, L. (1978). *Digital Reglering av Klimatprocesser (Digital Control of Climate Processes)*. PhD thesis TFRT-1014. Department of Automatic Control, Lund University, Sweden.

Kolokotsa, D., K. Niachou, V. Geros, K. Kalaitzakis, G. Stavrakakis, and M. Santamouris (2005). “Implementation of an integrated indoor environment and energy management system”. *Energy and Buildings* **37**:1, pp. 93–99.

Lamoudi, M. Y., M. Alamir, and P. Béguery (2012). “Model predictive control for energy management in buildings. part 1: zone model predictive control”. In: *IFAC Conference on Nonlinear Model Predictive Control 2012 (NMPC’12)*, pp. 21–26.

Lawrence Berkeley National Laboratory (2016a). *Modelica Buildings Library*. `http://simulationresearch.lbl.gov/modelica`. Online, accessed March 2016.

Lawrence Berkeley National Laboratory (2016b). *Modelica Buildings Library: online model documentation*. `http://simulationresearch.lbl.gov/modelica/`. Online, accessed March 2016.

Liu, T., W. Zhang, and F. Gao (2007). “Analytical decoupling control strategy using a unity feedback control structure for mimo processes with time delays”. *Journal of Process Control* **17**:2, pp. 173–186.

Ma, Y., A. Kelman, A. Daly, and F. Borrelli (2012). “Predictive control for energy efficient buildings with thermal storage: modeling, stimulation, and experiments”. *Control Systems, IEEE* **32**:1, pp. 44–64.

Mathews, E., D. Arndt, C. Piani, and E. Van Heerden (2000). “Developing cost efficient control strategies to ensure optimal energy use and sufficient indoor comfort”. *Applied Energy* **66**:2, pp. 135–159.

Moroşan, P.-D., R. Bourdais, D. Dumur, and J. Buisson (2010). “Building temperature regulation using a distributed model predictive control”. *Energy and Buildings* **42**:9, pp. 1445–1452.

Oldewurtel, F., A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari (2012). “Use of model predictive control and weather forecasts for energy efficient building climate control”. *Energy and Buildings* **45**, pp. 15–27.

Paris, B., J. Eynard, S. Grieu, T. Talbert, and M. Polit (2010). "Heating control schemes for energy management in buildings". *Energy and Buildings* **42**:10, pp. 1908–1917.

Parisio, A., L. Fabietti, M. Molinari, D. Varagnolo, and K. H. Johansson (2014a). "Control of hvac systems via scenario-based explicit mpc". In: *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, pp. 5201–5207.

Parisio, A., M. Molinari, D. Varagnolo, and K. H. Johansson (2013). "A scenario-based predictive control approach to building hvac management systems". In: *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*. IEEE, pp. 428–435.

Parisio, A., D. Varagnolo, M. Molinari, G. Pattarello, L. Fabietti, and K. H. Johansson (2014b). "Implementation of a scenario-based mpc for hvac systems: an experimental case study". In: *Preprints of the 19th World Congress, The International Federation of Automatic Control*. Vol. 10, p. 10.

Peffer, T., M. Pritoni, A. Meier, C. Aragon, and D. Perry (2011). "How people use thermostats in homes: a review". *Building and Environment* **46**:12, pp. 2529–2541.

Salsbury, T. I. (2005a). "A survey of control technologies in the building automation industry". In: *16th IFAC World Congress*.

Salsbury, T. I. (2005b). "A survey of control technologies in the building automation industry". In: *Proc IFAC World Congress*.

Scotton, F. (2012). "Modeling and identification for hvac systems".

Seborg, D. E., D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III (2010). *Process dynamics and control*. John Wiley & Sons.

Shen, Y., W.-J. Cai, and S. Li (2010). "Normalized decoupling control for high-dimensional mimo processes for application in room temperature control hvac systems". *Control Engineering Practice* **18**:6, pp. 652–664.

The Modelica Association (2016). *Modelica*. `https://www.modelica.org/`. Online, accessed March 2016.

Waller, M., J. B. Waller, and K. V. Waller (2003). "Decoupling revisited". *Industrial & engineering chemistry research* **42**:20, pp. 4575–4577.

Wang, S. (2010). *Intelligent buildings and building automation*. Routledge.

Wetter, M., W. Zuo, and T. Nouidui (2011a). "Modeling of heat transfer in rooms in the modelica buildings library". *Buildings" Library," in Proceedings of Building Simulation*.

Wetter, M., W. Zuo, T. Nouidui, and X. Pang (2011b). "Modelica buildings library". *Buildings Library, in Proceedings of Building Simulation*.

Wetter, M., W. Zuo, T. S. Nouidui, and X. Pang (2014). "Modelica buildings library". *Journal of Building Performance Simulation* **7**:4, pp. 253–270.

Wong, A. C. W. and A. T. P. So (1997). "Building automation in the 21st century". In: *Advances in Power System Control, Operation and Management, APSCOM-97. Fourth International Conference on*. Vol. 2, pp. 819–824.

Zhou, P., G. Huang, and Z. Li (2014). "Demand-based temperature control of large-scale rooms aided by wireless sensor network: energy saving potential analysis". *Energy and Buildings* **68**, pp. 532–540.

A

Convexity of the Cost Function

One of the advantages of the cost function (5.1) over, e.g., the asymmetric cost presented in [Kovatchev et al., 2000] is that it is convex.The convexity of $L(y(k))$ in Eq. (5.1) with respect to $y(k)$ can be shown as follows. First, it is shown that $L_1 = \max\{(d - y(k))^3, 0\}$ is convex in $y(k)$. It is $L_1 = \max\{(d - y(k))^3, 0\} = (\max\{(d - y(k)), 0\})^3$. Since $\max\{(d - y(k)), 0\}$ is the point wise maximum of two affine functions, it is convex. The function x^3 is convex and non-decreasing as long as $x \geq 0$. The composition of these two convex functions to $L_1 = (\max\{(d - y(k)), 0\})^3$, gives a convex function [Boyd and Vandenberghe, 2004]. Note that $\max\{(d - y(k)), 0\}$ is always positive. The second part of $L(y(k))$, which is $L_2 = a \cdot y(k) + b$, is an affine function and thus convex. The weighted sum of L_1 and L_2 gives $L(y(k)) = L_2 + cL_1$, and is convex since L_1 and L_2 are convex and $c \geq 0$ [Boyd and Vandenberghe, 2004].

B

Simulation Results Part I

The following pages contain additional results for the first part of this thesis. In Chapter 6, simulation results for one virtual patient were presented. The figures here show the simulation results for the all three virtual patients.

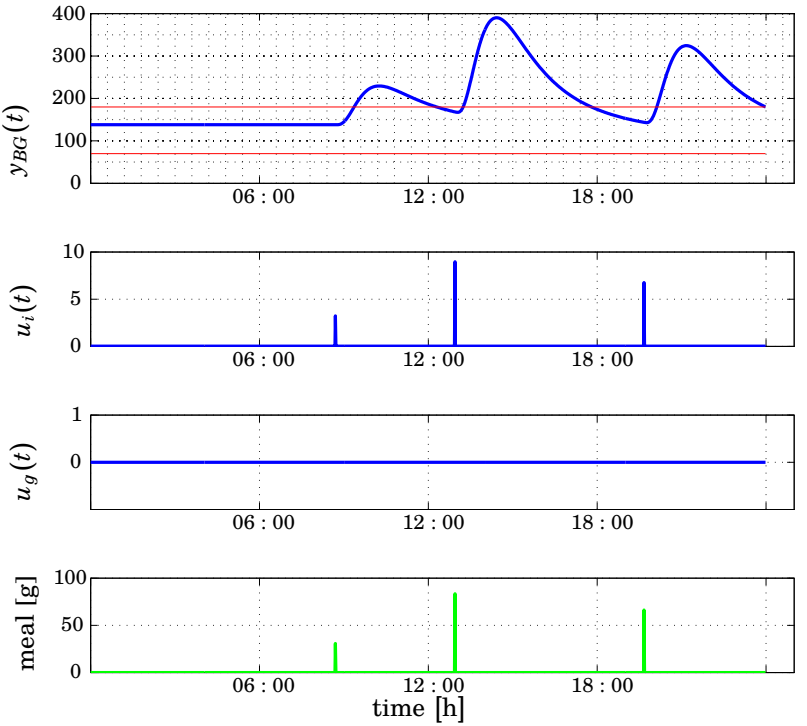

Figure B.1 Simulation results for Virtual Patient 1 with Bolus Calculator.

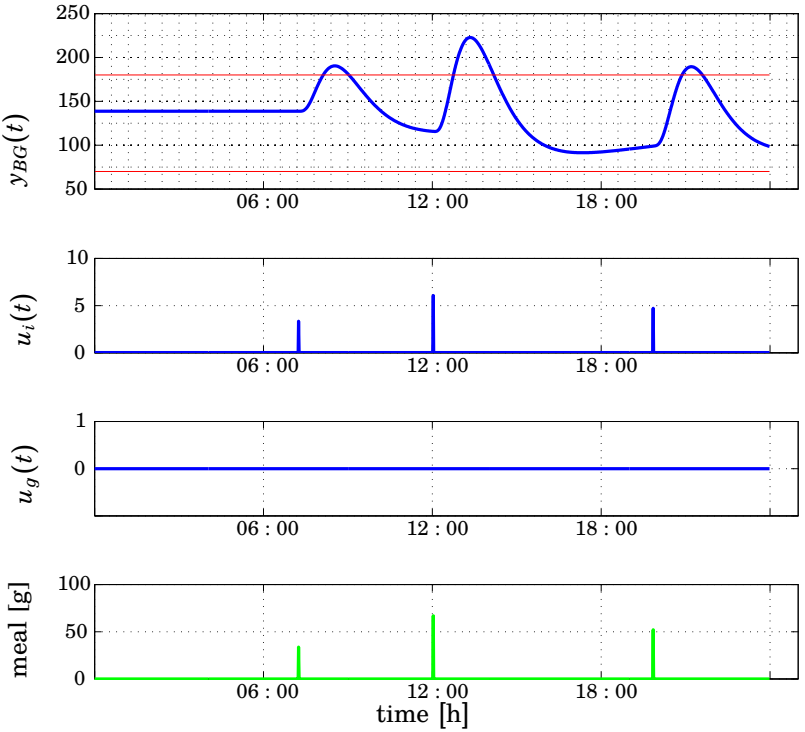

Figure B.2 Simulation results for Virtual Patient 2 with Bolus Calculator.

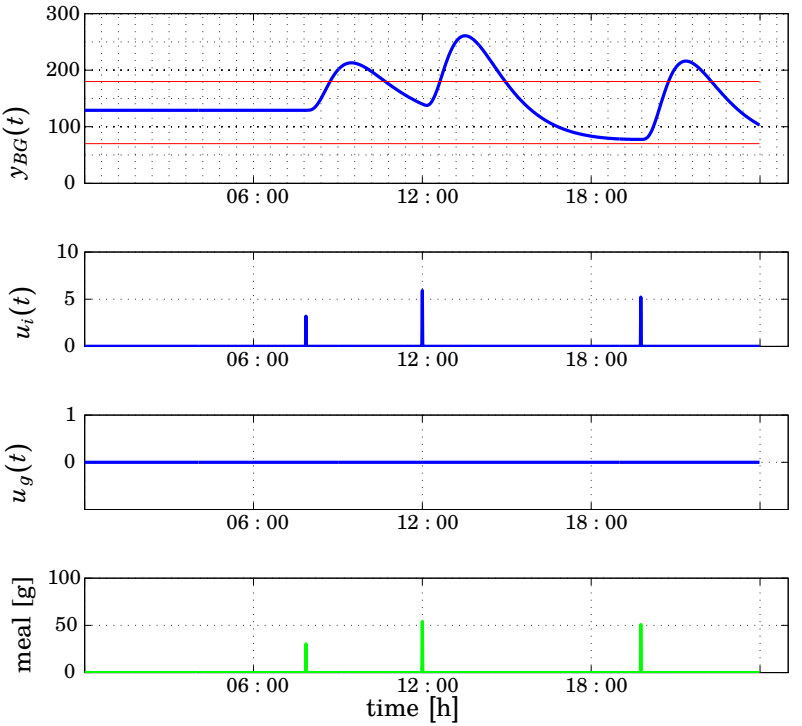

Figure B.3 Simulation results for Virtual Patient 3 with Bolus Calculator.

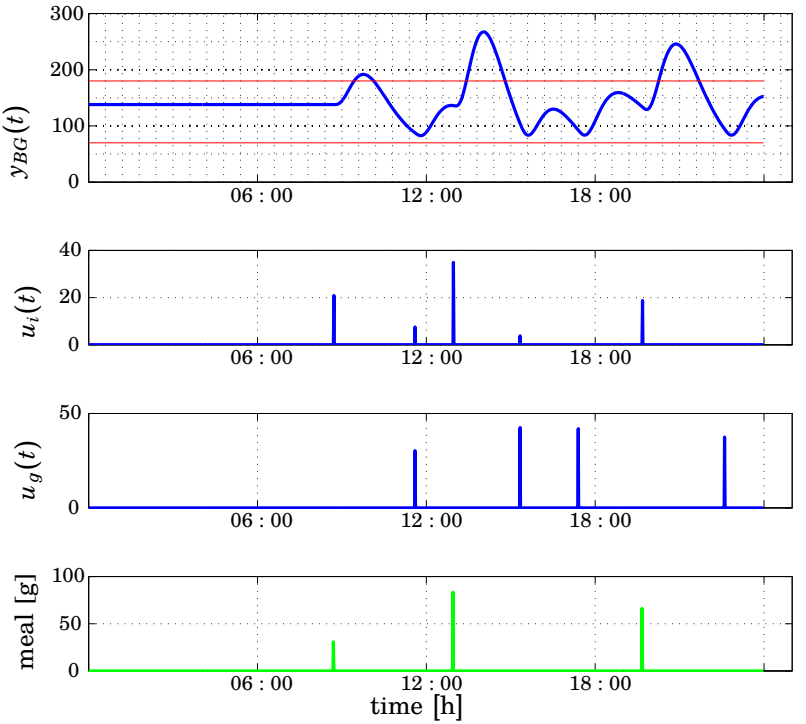

Figure B.4 Simulation results for Virtual Patient 1 with the optimization-based controller, tuned for good control.

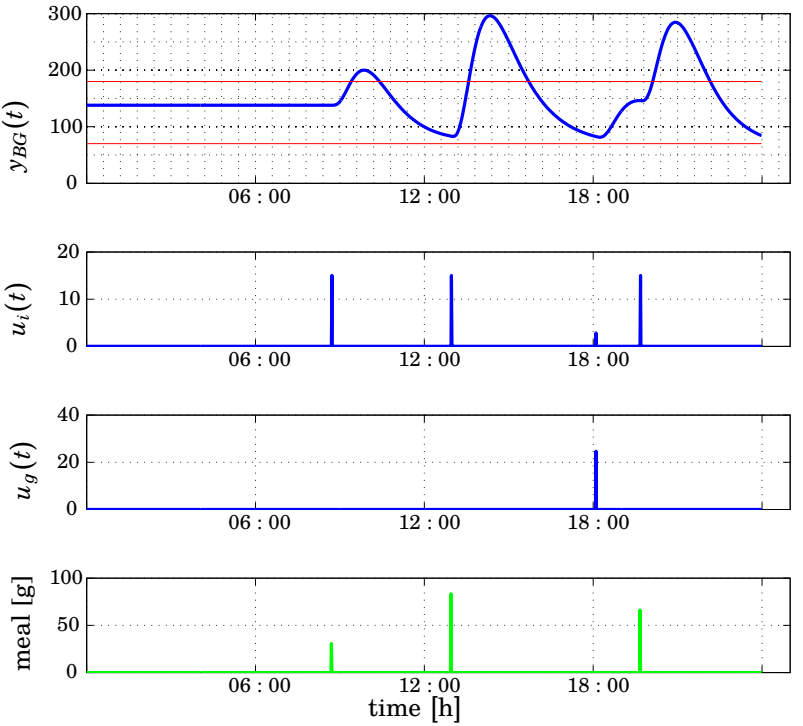

Figure B.5 Simulation results for Virtual Patient 1 with the optimization-based controller, tuned for less insulin and glucose dose advices

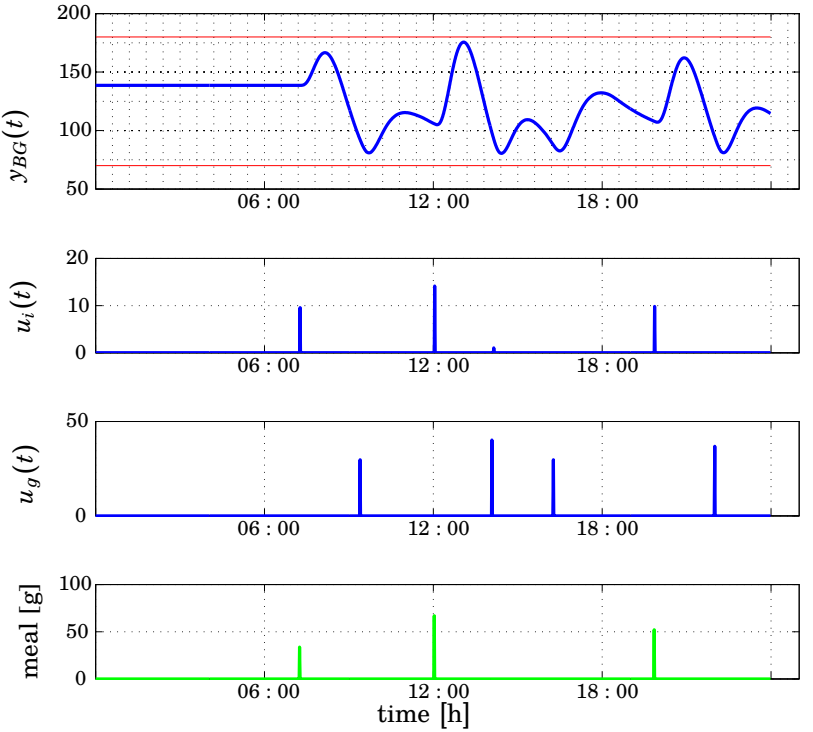

Figure B.6 Simulation results for Virtual Patient 2 with the optimization-based controller, tuned for good control.

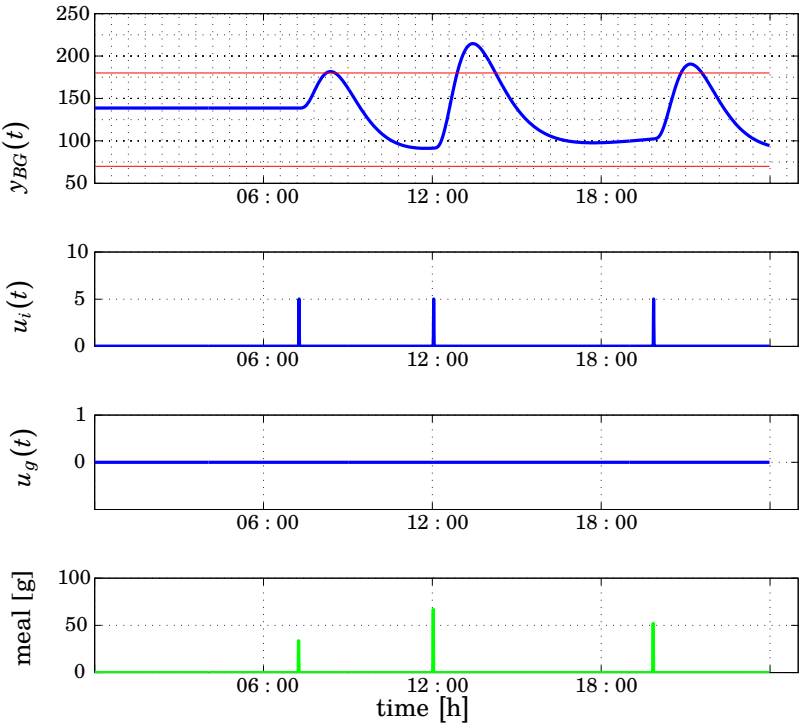

Figure B.7 Simulation results for Virtual Patient 2 with the optimization-based controller, tuned for less insulin and glucose dose advice.

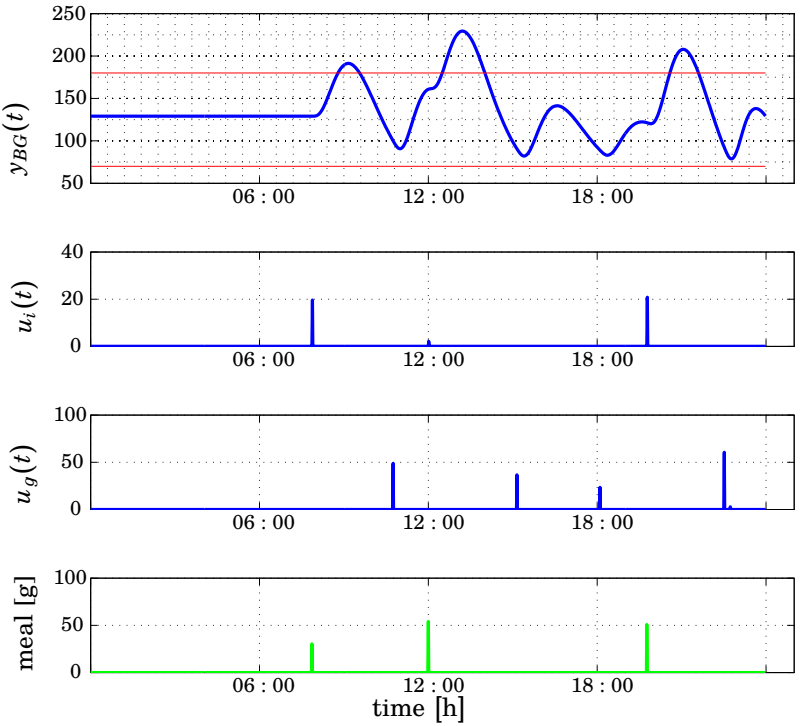

Figure B.8 Simulation results for Virtual Patient 3 with the optimization-based controller, tuned for good control.

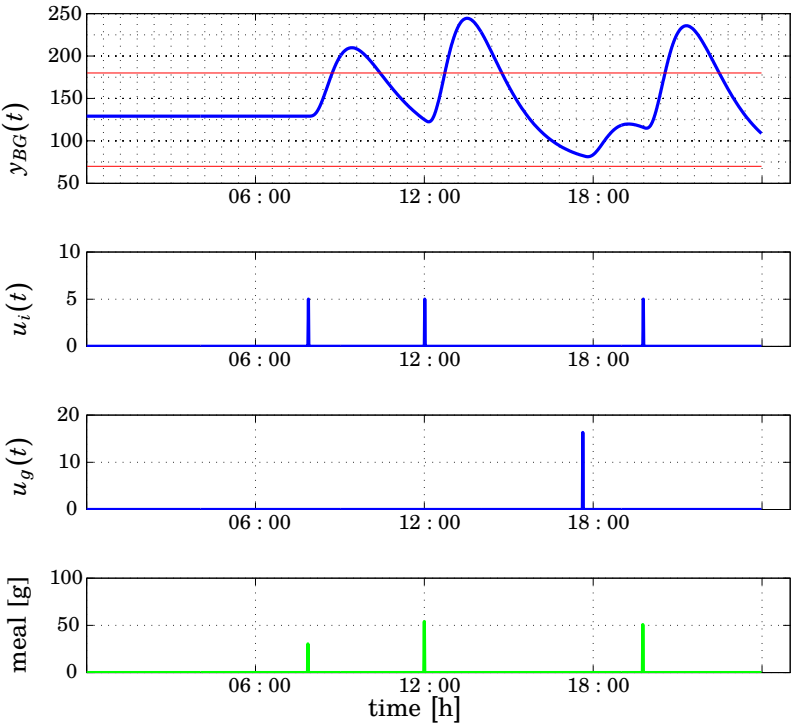

Figure B.9 Simulation results for Virtual Patient 3 with the optimization-based controller,tuned for less insulin and glucose dose advice.