The Optimization Algorithm rFSQP with Application to Nonlinear Model Predictive Control of Grate Sintering

Thesis by

Frode Martinsen

Submitted in partial fulfillment of the requirements for the degree of

Doktor Ingeniør

Department of Engineering Cybernetics
Norwegian University of Science and Technology
N-7491 Trondheim, Norway
August 2001

Report 2001:9-W

URN:NBN:no-1287

Preface

This thesis has been submitted to the Norwegian University of Science and Technology (NTNU) in partial fulfillment of the requirements for the degree of Doktor Ingeniør.

The doctoral project has been accomplished at the Department of engineering cybernetics, NTNU, in cooperation with Elkem Sauda, Norway, and the Department of Chemical Engineering at the Carnegie Mellon University. The Elkem ASA plant at Sauda is now a part of Eramet Norway AS, a subsidiary of the Eramet Group. My supervisors have been professor Bjarne A. Foss and associate professor Tor A. Johansen. One semester of the doctoral project was spent at the Department of Chemical Engineering at Carnegie Mellon University under the supervision of professor Lorenz T. Biegler. The work has been supported by the Norwegian Research Council (NFR) grant # 119314/221.

Acknowledgments

During the work on this thesis, many people have been involved in guidance, discussions, solving practical problems, and motivating me. I would like to mention some of them here.

First, sincere thanks to professor Bjarne A. Foss for initiating the project, accepting me as a doctoral student, and for his optimistic attitude, inspiration, and support throughout the work. He also established the contact and opened up for the cooperation with professor Lorenz T. Biegler at the Carnegie Mellon University.

Due to professor Lorenz T. Biegler's well-known experience in optimization, he taught me virtually all there is to know about practical optimization. He included me in his work-group as one of his students and shared with me from his long scientific experience. He also provided access to his optimization codes which has been of paramount importance in my work. His hospitality and friendly nature have made him a highly regarded mentor. A particular thanks to Ph.D. student Andreas Wächter from professor Biegler's group for his hospitality during my stay at Carnegie Mellon University, and for always answering my questions by email during the last two years.

I would also like to thank associate professor Tor Arne Johansen, for always taking an interest in discussing some of the mathematical details.

I am grateful for receiving the opportunity of working on industrial problems within the frame of a doctoral degree. I acknowledge the cooperative

spirit and helpfulness that I have met within Elkem ASA, and I especially wish to thank Stein Wasbø and Ragnar Tronstad for their valuable support and cooperation. Personnel at the sintering plant, Marianne Brandett, Bente Baugstø, Leif-Idar Rossemyr[1], Sigurd Simonsen and Gunnar Mørkesdal all provided essential assistance during the measurement campaigns. Morten Raanes at SINTEF Materials Technology performed the SEM analysis documented in section 2.6.3, while Tone Anzjøn prepared the samples for these tests.

A good working environment including colleagues, supervisors, secretaries and lab personnel has been an important factor for me to succeed with this project. I also thank the members of the process control group at NTNU, led by professor Bjarne A. Foss, for an inspiring and social environment. In particular, Geir Stian Landsverk who has been sharing office with me during the last year and a-half is duely thanked for his insightful comments on the sintering process and for (always) accepting offers to waste time in non-academic dialogue. Geir Stian has also checked the thesis for typograhpical errors etc. My first office-mate, my friend Vidar Sørhus, is also thanked. Without him I would never have started on this project.

My parents and family are thanked for believing in me. My thanks to Ida, Kent and "beste" for their patience. My wife, Anne Berit, has always encouraged me and believed in me, and for her patience, I love and respect her.

The last three years of this work, my son, Magnus, has been my greatest source of inspiration. Thank you.

Frode Martinsen

Trondheim, August, 2001

[1]Stein Wasbø, Ragnar Tronstad, Bente Baugstø and Leif-Idar Rossemyr are presently with Eramet Norway AS.

Summary

This thesis contributes to the research on optimization algorithms for nonlinear programming, and to the application of such algorithms to nonlinear model predictive control.

Regarding the contribution to research on algorithms for nonlinear programming, a novel algorithm is put forward with a complete theory for global and local convergence. This is the main contribution of the thesis. The algorithm, named `rFSQP`, is a **r**educed Hessian **F**easible **S**equential **Q**uadratic **P**rogramming method. It remains feasible with respect to nonlinear inequalities at all `SQP` iterations, but nonlinear equality constraints are treated as in general reduced Hessian `SQP` methods. The `rFSQP` algorithm is implemented in `MATLAB` and tested on a number of small scale problems with encouraging results. However, the algorithm is designed for large scale problems with few degrees of freedom. Some preliminary testing of the algorithm on large scale problems are investigated.

The thesis also contributes to the understanding of the relation between sequential and simultaneous reduced gradient methods, and to the understanding of the relation between discretization methods for dynamical systems and the choice of optimization algorithms.

The thesis also contributes to model based control approaches of grate sintering. Grate sintering is a complex metallurgical process, where melting of solids and fast gas dynamics give rise to stiff process models, i.e. the "time constants" of the system differ by many decades in magnitude. Hence, application of real-time optimization methods like nonlinear model predictive control to the grate sintering process is challenging. The thesis gives a framework for implementing nonlinear model based control of grate sintering by giving a control objective, a nonlinear model and choosing an appropriate discretization scheme. The thesis gives a reduced order model which is less computationally demanding. Data from industrial experiments are used to adapt the model and to assess the control objective.

Contents

URN:NBN:no-1287

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Optimality is a natural phenomenon which has engaged scientists in various guises for centuries. The following quote illustrates its generality and importance:

> "Since the fabric of the universe is most perfect, and is the work of a most wise Creator, nothing whatsoever takes place in the universe in which some form of maximum and minimum does not appear."
>
> Leonhard Euler, 1744[1]

A creek always taking the steepest path downhill is an example of minimization of energy. Consider another example; imagine a blind person, call him Mr. Iterate, on top of the mountain Besseggen[2] and ask him to find his way down on his own. By careful steps he explores the downhill path, going one step at a time, adapting his step lengths to the terrain and possibly using his memory to correct zig-zagging. Eventually, he arrives at the saddle point between the two lakes, and after testing for further descent directions, declares that he is now at the lowest point. Of course, if he strayed, say 5 meters, towards any of the two lakes he would have sensed a descent direction and continued his path. However, from his point of view there

[1] Restated from Troutman (1996), p. 339.

[2] A famous mountain hike in Norway crosses this ridge which has a saddle point between a grey and a green colored lake on each side of the trek between two peaks. It is commonly believed that Ibsen's character Peer Gynt jumped off this ridge on his buck.

is nothing to gain by moving away from his present position. The terrain seems to be flat, i.e. he has stopped at a *local* solution.

Although there exists a considerable body of optimization examples that can be handled by pen-and-paper calculations, computerized solutions of optimality problems have emerged during the last few decades as a powerful tool for solving larger and harder optimization problems.

A computer program for solving optimization problems commonly iterates the problem, and tries to improve on the present solution. If the problem is nonlinear but smooth and analytic (exists and is differentiable), a common approach is to linearize and solve simpler subproblems. The solutions to these subproblems become search directions, and it is then customary to moderate the step lengths to compensate for the error in the linearization. Linearization only provides local information and extrapolating information too far can be hazardous.

Returning to Mr. Iterate, this could be the way he chooses where to place his next step; he perturbs ("linearizes") the terrain in front to find the steepest descent and moderates his step length if the terrain is rugged or very steep. Note that Mr. Iterate does not have a look-ahead property, but he has a memory. Using his memory he can speed up his descent if the steepest descent path tend to zig-zag. I.e. if he experiences zig-zagging he can bend his step directions towards what seems to be the historical average direction towards the minimum. Needless to say, he would not be pleased if the terrain suddenly revealed a discontinuous vertical wall.

If there are fences in the terrain and it is required that the optimal point should be within the fences, Mr. Iterate may consider searching within the fences, or to cross them and search for the lowest topological point on the outside, but keeping in mind that he should return to the inside for the final point. Perhaps it is reasonable that he would make only conservative strolls outside the fences, and hesitate straying too far away from them?

If there are stronger requirements, e.g. that the optimal point should lie *on* a fence or a trek, Mr. Iterate could be forced to follow the trek or to stray from it. If there are many treks it could then be time consuming to trace them all.

Summarizing the various ways and reasonings of Mr. Iterate in an algorithm and implementing this in a computer program is precisely what is undertaken in this thesis. The algorithm shall be a generic and effective set of rules that applies to all problems falling within a specified set of assumptions. The proposed algorithm, termed `rFSQP`, is designed to solve nonlinear optimization problems with thousands of variables subject to nonlinear constraints. Specifically, `rFSQP` is designed to always remain within

(e.g. feasible) all in-/outside fences (inequalities), but may stray from "on"-fences (equalities) away from the final point, i.e. it is an inequality feasible algorithm for nonlinear optimization problems[3].

Like Mr. Iterate the algorithm will converge to a locally optimal point, and close to the optimal point the convergence will be sufficiently fast. The computer program is tested on a number of small sample problems, and on two larger problems. The first large problem behaves nicely, i.e. it changes by relatively moderate rates in all directions. The second problem is more challenging; it is the case of grate sintering. The dynamical model of this problem has strong nonlinearities and the time constants are separated by several decades. Both problems are examples of nonlinear model predictive control (`NMPC`).

1.2 Contributions

The main contribution of this thesis is the optimization algorithm `rFSQP`. A complete convergence analysis is given, considering both global and local properties, and the algorithm is implemented and tested on a number of problems.

The main case is grate sintering. For this case a reduced order model is developed from models available in the literature. Industrial experiments including special measurements were conducted, and the data was used for model adaption and assessing the control objective.

Various implementation issues concerning the interplay between discretization and optimization nonlinear `MPC` are explored. This provides insight into how a continuous time model must be discretized to allow optimization.

Appendix D gives a generic approach to representing hyperbolic `PDE`'s as multi-models. Due to numerical difficulties the approach has not been pursued further.

1.3 Outline

This chapter has presented the motivation for the present work, what is believed to be its main contribution, and placed it in a broader context. More detailed background, including references, are given in the introductions to chapter 2 and 3.

[3] Such problems belong to the class of nonlinear programming problems, which is a subclass of mathematical programming problems.

Chapter 2 presents the grate sintering case, with a nonlinear `PDE` model, control objective, industrial experiments and data analysis.

Chapter 3 presents the optimization algorithm `rFSQP` with global and local convergence analysis, implementation details and numerical results on a selection of small sample problems.

Chapter 4 applies the optimization algorithm `rFSQP` to two different nonlinear model predictive control (`NMPC`) examples of different complexity. The second example is the grate sintering example.

Chapter 5 ends the thesis and gives its conclusions.

Appendices A to C provide some additional details for the various chapters.

Appendix D is a reprint of the paper Martinsen, Johansen, and Foss (1999).

The notation is consistent within each chapter, but to conform to the common notation within the sintering and optimization literature, respectively, the notation is not consistent between different chapters. The notation for chapter 2 is given in section 2.1.

Chapter 2

Sintering

This chapter considers the metallurgical process of grate sintering. The purpose is to develop and assess a dynamic model suitable for modelbased control. The model is adapted to industrial data from experiments conducted at the sintering plant at Sauda in south-west Norway. The process at Sauda is batch-wise sintering of manganese ore, but much of the discussion, and in particular the model, is equally relevant for travelling grate sintering. It should be noted that manganese is in many aspects similar to iron which is most frequently considered in the literature.

This chapter starts with some background on sintering in section 2.2, i.e. the process is described with some comments on the sintering plant as a whole. The underlying principles governing the quality and production rate of sintering are reviewed. In section 2.3 the model is documented and discussed, while the industrial experiments and measurements are described in section 2.4. Model adaption to the industrial data and model validation are adressed in section 2.5, while data reconciliation and analysis is considered in section 2.6. Some concluding remarks and a summary of the contributions of this chapter follows in section 2.7. The notation used in this chapter is summarized in section 2.1. Parts of this chapter have been published in Martinsen, Johansen, and Foss (1999), which is reprinted in appendix D.

2.1 Notation for chapter 2

Arabic letters

a_i - polynomial coefficient [-]

A_b - specific surface area $[m^2/m^3]$

A_f - frequency factor $[m/s\sqrt{K}]$

c_p - specific heat capacities of solid $[J/kgK]$ and gas $[J/molK]$

d_p - average particle diameter $[m]$

D_{ON} - axial gas dispersion coefficient (O_2-N_2) $[m^2/s]$

E - activation energy $[kJ/mol]$

G - mass flow rate of gas used in equation (2.2)$[m^2t/h]$

G_s - gas volume per sinter mass used in equation (2.2) and (2.3) $[m^3/t]$

F_i - liquid fraction

h_c - convective heat transfer coefficient $[J/m^2sK]$

h_m - mass transfer coefficient $[m/s]$

ΔH - heat of reaction $[J/mol]$

k_{th} - thermal conductivity of gas $[J/mK]$

k_r - chemical reaction rate $[m/s]$

K_r - overall combustion rate constant $[m/s]$

L - height of bed $[m]$

L_f - latent heat of fusion $[kJ/mol]$

L_v - latent heat of evaporation $[J/kg]$

m_i - exponent [-]

M_i - molecular weight $[kg/mol]$

n_C - number of coke particles per unit bed volume $[1/m^3]$

n_i - molar flow $[mol/m^2s]$

p_i, Δp - (partial) pressure, differential pressure $[Pa]$

$P_{l,t}^{Voice}$ - Laminar or turbulent Voice gas permeability of bed (units vary according to exponent m_1 in equation (2.1))

r_i - reaction rate $[mol/m^3 s]$

R - universal gas constant 8.314 $[J/molK]$

$\tilde{R}_R$, R_R - reaction rate for water $[mol/m^3 s]$

s_i - stoichiometric constant [-]

$T_{s,g}$ - temperature, solid or gas $[K]$

T_{fu} - incipient melting temperature of solids $[K]$

T_w - wet-bulb temperature $[K]$

v - gas velocity $[m/s]$

v_0 - gas velocity as referred to the empty bed used in equation (2.2) and (2.3) $[m/h]$

v_w - heat wave velocity used in equation (2.3) $[m/h]$

W - actual moisture content $(= x_{H_2O(l)})$ $[kg/m^3]$

W_{cr} - critical moisture content $[kg/m^3]$

x_i - component concentration of solids $[kg/m^3]$ or gas $[mol/m^3]$

Greek letters

ε - void fraction $[-]$

φ - scaling factor for the Sherwood relation [-]

κ_g - adiabatic constant $[-]$

μ - viscosity $[kg/ms]$

ρ - density $[kg/m^3]$

ω - specific humidity [-]

Dimensionless numbers

Re - Reynolds number: $Re = \frac{d_p G}{\mu}$

Sh - Sherwood number: $Sh = \frac{h_m d_p}{D_{ON}}$

Sc - Schmid number: $Sc = \frac{\mu}{\rho D_{ON}}$

Nu - Nusselt number: $Nu = \frac{h_c d_p}{k}$

Pr - Prandtl number: $Pr = \frac{c_p \mu}{k}$

2.2 Background

The main purpose of sintering is to convert weakly-bounded granules into a partially fused porous sinter cake suitable for feeding a furnace. In the furnace the sintered ore is reduced with carbon. Sintering is an agglomeration process in which fine graded materials are partially fused into larger lumps by heating the charge through coke combustion. Fusion occurs when the solid charge particles undergo re-crystallization across the old grain boundaries, and possibly by simultaneous softening and partial melting. Sintering is a complex process involving flow of gas through a packed bed, heat and mass transfer between gas and solids, heterogeneous chemical reactions, and melting of solids. The heat for sintering of oxidic ores like Fe_xO_y and Mn_xO_y is provided by heat exchange between gas and solids and by combustion of coke. Typically the temperature must be raised to the range of $1000^oC - 1400^oC$. Since only approximately 5% (weight) of coke is needed to raise the temperature to this level, the process is generally considered to be economic in terms of energy. Water is added to micro-agglomerize finer granules by the capillary forces of water. Micro-agglomeration increases the gas permeability of the bed allowing a larger gasflow, which in turn improves sintering conditions. The critical water content is typically below 8% (weight). Typical granule sizes of the incoming material are in the range $3mm$ to $6mm$. A large fraction of finer grades can cause low gas permeability of the bed, while coarser grades can give poor micro-agglomeration. Both cases give poor sintering conditions.

Sintering plants are commonly located close to a furnace, since, due to its low mechanical strength, sinter is ill-suited for transportation and extensive handling. The mechanical strength of the sinter can be increased by adding more coke and thereby increasing the sintering temperature. Increased sintering temperature allows a larger fraction of the solids to melt

thereby giving a stronger sinter. However, this also gives a glassy surface of the sinter, which impairs its reducibility since the effective surface area is reduced. Hence, there is a compromise between mechanical strength and reducibility which is mainly controlled by the coke weight percent.

The metallurgical process of sintering prepares the ore to form suitable feed for a (blast) furnace. Granulated ore and coke are mixed, moistened with water and micro-pelletized to form the charge. The charge is loaded onto a grate and levelled to form a bed which is ignited by a gas-fuelled ignition hood. A heat wave and coke combustion zone travels down through the bed under the influence of a suction pressure. Hot gas from the combustion zone passes through moist charge deeper in the bed where water evaporates. The process can be divided into five subsequent zones; heat exchange, fusion, combustion, drying and overmoist charge. This is illustrated in figure 2.1. A number of operations, such as feed mixing, feed charging, crushing of the produced sinter cake, screening and recycling of fines are needed. A simplified outline of the overall sintering plant is shown in figure 2.2. The sintering plant as a whole is only considered in section 2.4 where the experiments conducted at the plant are described. The dynamic model in section 2.3 is concerned with the sintering process itself, i.e. the process taking place inside the sintering pan.

The industrial plant at Sauda produces manganese alloys, such as ferromanganese ($FeMn$) and silicomanganese ($SiMn$) from manganese ore in electric furnaces. $FeMn$ is typically used as an addition in the steel industry to produce certain steel qualities. Such steel qualities are used in rail-way tracks, wear-plates, etc. Because of the high reduction temperature the electric furnace is competitive with the blast furnace, especially for high purity qualities where the carbon content of $FeMn$ must be low. $FeMn$-alloys with low and medium carbon (LC/MC) content are produced in a subsequent refining process (`MOR`). Limestone, CaO, is commonly added in ironmaking to adjust the basicity defined by the weight-ratio of $(CaO + MgO)/(SiO2 + Al2O3)$. Limestone is not used at the sintering plant in Sauda, and is not included in the model. For a discussion of basicity in the context of manganese reduction, see Rosenqvist (1983), p. 357. General properties of the sintering process are described in Schluter and Bitsianes (1962).

2.2.1 Process goals

For the sintering process the overall goal is to produce sinter at a prescribed quality and rate at the lowest possible cost. The process dependent outputs

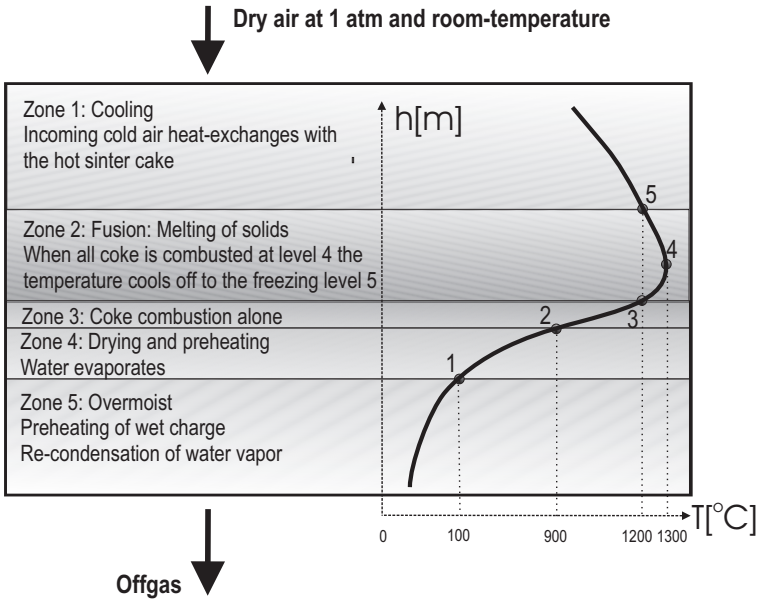

Figure 2.1: *Zones 1-5 in the sintering bed.* The figure shows a snapshot of a vertical slice of the sintering bed approximately mid-way through the process. The five zones are described to the left, with the corresponding temperature profile to the right. As time goes by the temperature profile proceeds down through the bed, i.e. at a time earlier than the snap-shot in the figure the temperature profile is shifted upwards, and at later times the profile is shifted downwards relative to the one shown in the figure. The offgas is led through pipes to a cyclone, and the pressure below the pan is less than 1atm. At the time instance shown in the figure cool air from the surroundings is being sucked into zone 1 where it heat-exchanges with the hot sinter cake. I.e. the air is preheated, and the sinter is cooled. At level 5 the temperature is 1200^oC, which is the approximate freezing point of the sinter. In zone 2 the temperature is above the freezing point, and the solids are partly melted depending on the heat available for fusion. At level 4 all the coke in the bed has been combusted. Hence, above this point there are no sources of energy and the available energy between level 4 and 5 is being used for fusion and heat-exchange. At level 3 the heat from coke combustion has raised the temperature in the bed to the melting point of the charge. From level 3 to 4 coke is combusted and fusion commences. At level 2 the ignition temperature of coke is reached, and the heat from coke combustion quickly raises the temperature. In zone 4, hot gas from zone 3 pre-heats the charge, and evaporates water. Water vapor is being transported down through the bed to zone 5 where it may recondensate. At level 1 the temperature of the charge is at the boiling point of water, and above this level water evaporates.

are the quality in terms of mechanical strength and reducibility (Dawson 1993), and the production rate.

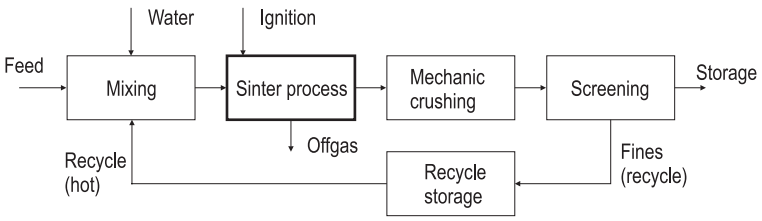

Figure 2.2: *Simplified sintering plant.* Feed, consisting of a mix of various ores, coke and usually lime stone, enters to the left. The composition of the feed is known, but not the moisture content. The feed is mixed with water and (hot) recycle of undersized particles. The adhesive capillary forces of water gives micro-agglomeration of the finer particles. Only fine particles are micro-agglomerized, and large fractions of coarser particles are usually undesirable. The temperature of the recycle can be above the boiling point of water, hence evaporation of water can make it difficult to mix in the optimal water content. The mix of ore, coke, water and recycle is then charged onto the sintering bed which is supported underneath by a grate. A suction pressure is established and air is drawn down through the bed. The sintering process commences when ignition is applied to the top of the bed. The coke in the top layer of the bed is ignited and the sintering process proceeds as described in figure 2.1. After completion of the sintering process the produced sinter cake is crushed into manageable lumps by a mechanical device. The sinter is then screened, and the fines are recycled, while coarser sinter is stored or fed to the furnace.

Production rate

The productivity is quantified by the stationary Voice gas permeability P^{Voice} (Voice, Brooks, and Gledhill 1953) in the laminar and turbulent flow regimes:

$$\begin{aligned} P_l^{Voice} &= v\left(\tfrac{L}{\Delta p}\right)^{1.0} = \tfrac{1}{150\mu}\cdot\tfrac{\varepsilon^3}{(1-\varepsilon)^2}\cdot d_p^2 \\ P_t^{Voice} &= v\left(\tfrac{L}{\Delta p}\right)^{0.5} = \sqrt{\tfrac{1}{1.75\rho}}\cdot\left(\tfrac{\varepsilon^3}{1-\varepsilon}\right)^{0.5}\cdot d_p^{0.5} \end{aligned} \tag{2.1}$$

The productivity relation thus becomes (Olsen 1997):

$$G = \frac{v}{G_s} = \frac{P^{Voice}}{G_s}\cdot\left(\frac{\Delta p}{L}\right)^{m_1} \tag{2.2}$$

where $m_1 = 0.5$ for turbulent flow and $m_1 = 1$ for laminar flow. The mass air flow G then serves as an on-line measure of production rate. Note that when significant melting occurs, the solids becomes a continuous media and the particle size d_p is not a meaningful parameter. Dawson (1993) suggests to select d_p as the minimum measured sieve fraction of incoming ores in the case

of significant melting. The process variable influencing on production rate is mainly the permeability P of the bed which in turn is influenced mainly by water content. Water is added to micro-agglomerize finer particles by the capillary forces of water. This increases the average particle diameter and the void fraction of the bed if the water content is kept below a critical value W_{cr}. A large air flow is promoted by a large permeability which in turn forces the heat wave to travel faster through the sintering bed, thus giving a shorter batch duration and consequently an increased production rate.

Utilizing the Voice permeability P^{Voice} to estimate production rate is impractical since it does not account for the amount of recycle in the plant. In the present work, P^{Voice} is discarded and the production rate is estimated from the calculated recycle. The recycle stream serves as an indirect on-line measure of mechanical sinter quality: Poor mechanical quality will give increased recycle rate, which in turn reduces the production rate.

Quality

The sinter quality is determined by the amplitude and shape of the heat wave. Increased coke content increases the maximum sintering temperature, $T_{s,max}$ (Venkataramana, Gupta, Kapur, and Ramachandran 1998), but to achieve high reducibility $T_{s,max}$ should not be too high (Toda, Senzaki, Isozaki, and Kato 1984). In addition, proper ignition is necessary to establish initial conditions for sintering (Dash and Rose 1977). As discussed above there is an optimal water content yielding the highest bed permeability. Hence, there is an optimum depending both on the coke (Toda et al. 1984) and water content (Hinkley, Waters, O'Dea, and Litster 1994). Due to the large (hot) recycle and time delays present, the process is considered difficult to control, with quality and production rates being hard to predict (Cumming and Thurlby 1990).

Control objective

To summarize, the control objective should balance an optimal temperature profile, while considering quality, against minimizing sintering batch time. The sintering batch time is reduced by an increased air mass flow G since the heat wave velocity v_w is increased by an increasing air flow. The vertical velocity of the travelling combustion zone is assumed to follow the linear

relationship (Olsen 1997), p.30:

$$v_w = \frac{v_0}{G_s \rho_b} \tag{2.3}$$

where v_0 $[m/h]$ is the gas velocity, $\rho_b = (1-\varepsilon)\rho_s$ $[t/m^3]$ is the bulk density of the bed and G_s $[m^3/t]$ is the gas volume per sinter mass.

The optimal temperature profile can be quantified by integrating the part of the solid temperature that has values above the fusion temperature T_{fu}. The fusion temperature T_{fu} is defined by the liquidus curves of the ore composition. Partial melting of the solids occur when the temperature is raised above T_{fu}. Fusion is discussed in more detail in section 2.3.1.

The temperature profile at various levels inside the bed should be evenly distributed during the batch to give equal sintering conditions in throughout the whole sintering bed. A measure for the temperature profile at each spatial level is

$$\tau_z \equiv \begin{cases} \int_0^{t_{end}} (T_{s,z} - T_{fu})dt & \text{if } T_{s,z} \geq T_{fu} \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

where the subscript z emphasizes the spatial distribution of the temperature profile and t_{end} is the batch duration time. Since the fusion temperature T_{fu} is uncertain and will vary with varying ore composition, a "smooth" switch is suggested to approximate the switch caused by equation (2.4). The smoothing function is chosen as the sigmoid function

$$\sigma(T_s) = \frac{1}{1 + e^{-k(T_s - T_{fu})}} \tag{2.5}$$

This is plotted with $T_{fu} = 1200^o C$ and $k = 0.1$ in figure 2.3

Hence, the control objective is to maximize an objective ϕ subject to the nonlinear inequality constraint imposed by quality as discussed in section 2.2.1. This is expressed formally as

$$\begin{aligned} \max_{T_s} \quad & \phi(T_s) = \int_0^L w(z) \int_0^{t_{end}} \sigma(T_s)(T_s - T_{fu})dtdz \\ s.t. \quad & \psi_z(T_s) = \int_{t=0}^{t_s} \sigma(T_s)(T_s - T_{fu})dt \leq q \\ & \text{additional constraints} \end{aligned} \tag{2.6}$$

where the parameter q remains to be selected. Observe that ψ_z is a vector function since T_s is spatially distributed. Hence, ϕ must be evaluated at

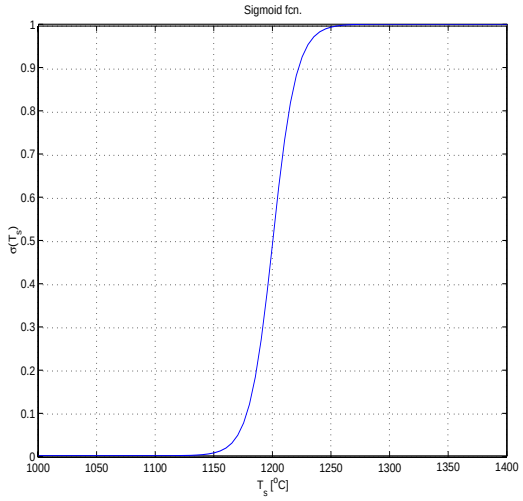

Figure 2.3: *Sigmoid function.* The figure shows the sigmoid function σ with parameters $T_{fu} = 1200^oC$ and $k = 0.1$. The sigmoid function is used as a smooth switch in the control objective.

each spatial position in the sintering bed, see appendix A.1 for details. The weighting function $w(z)$ is introduced to give varying contributions to the objective at different spatial levels. The parameter q is not chosen as a function (of T_s) since the uncertainty is assumed to be captured by the smoothing inherent in the sigmoid function. The validity of the objective function ϕ is assessed in section 2.6.3. The parameter k in σ can be tuned down to reduce the negative contribution following from $T_s < T_{fu}$.

2.2.2 Control objectives

There are few reported results on control of the sintering process. Kim and Kwon (1998) considers a linear `MPC` scheme designed to control the burn-through point of travelling grate sintering, using an identified input/output model. To the best of the authors knowledge, this is the only reference on model predictive control of the sintering process reported in the literature.

Across multiple batches

Controlling the coke and water content as discussed above can only be done across several batches, since the composition inside each sintering pan cannot be altered once it has been charged. The experiments documented in section

2.4 make some preliminary investigations of the relationship between the coke and water inputs, and the quality and production rate outputs. Inside the sintering pan, the coke and water contents have a negative bias due to the mixing of unmeasured (hot) recycle into the measured fresh feed. Note that the accuracy of the weights measuring the fresh feed may vary considerably. Manipulated inputs are the added coke, ignition energy, water and air flow rate, while the feed is regarded as a disturbance.

During one batch

The model and the control strategy considered in the following only considers the process taking place inside the sintering pan, i.e. the coke and water concentrations are not considered as control inputs. The only control action that influences the sintering process as the batch proceeds is the gas velocity. The gas velocity is controlled by adjusting the choke valve in the offgas-pipe, thus altering the differential pressure drop across the bed. We also assume that the there is sufficient ignition in the sense that the duration and quality of ignition allows the sintering process to start at the top of the bed. Below a qualitative assessment of the gas velocity as a control action is outlined, while a quantitative discussion is given in section 2.3.4.

The heat exchange properties of the initial raw charge is much better than the heat exchange properties of the sintered material. I.e., the heat transfer properties are altered by melting since the surface area is reduced, and the heat capacity of sintered material is altered due to the change in chemical composition. To compensate for this, a large amount of excess air is used in the sintering process, and it is not expected that the O_2 concentration will be rate limiting in coke combustion. There is an upper bound on the gas velocity, since too large gas velocities can cause the bed to collapse giving poor sintering conditions. The upper limit is dependent upon the permeability P^{Voice} as discussed in section 2.2.1, since a larger permeability allows a larger gasflow without collapsing the bed.

The large difference in heat exchange properties of charge and sinter causes the fusion zone to widen as the batch proceeds, giving different sintering conditions at the various layers of the bed. Commonly this is discussed in terms of "matching" of the combustion heat wave and the heat exchange wave in the literature. Adjusting the gas velocity, based on a model and the measured offgas quantities, then controls the width of the fusion zone at the various layers of the bed. This conjecture is investigated by ballistic simulations in section 2.5, and nonlinear `MPC` based on this control strategy is implemented in chapter 4. Preliminary experiments were conducted to

investigate this conjecture at the industrial plant, see section 2.4.3. The results of these experiments are not conclusive but the hypothesis cannot be rejected on the basis of these experiments.

Note that by assuming a constant velocity of the heat wave as it passes through the bed, simpler control schemes not depending on real-time optimization and a complex process model may be considered. This has not been considered in the present work, since it is outside the scope of the thesis which focuses on optimization and nonlinear MPC.

To summarize; the control objective is to maintain the same heat wave shape at all layers in the sintering bed. Informally the objective then is to balance the heat wave profile spatially for product quality, while simultaneously maximizing the gas velocity for production rate. These issues are revisited a number of times throughout this chapter.

2.3 Modeling

Several models of the sintering process are presented in the literature (Muchi and Higuchi 1972), (Dash and Rose 1977), (Hoislbauer and Jaquemar 1983), (Kasai, Yagi, and Omori 1984), (Cumming and Thurlby 1990), (Patisson, Bellot, Ablitzer, Marlière, Dulcy, and Steiler 1991), (Nath, Da Silva, and Chakraborti 1997), (Venkataramana, Gupta, Kapur, and Ramachandran 1998). These models are presented as nonlinear PDE's, and mainly focus on reproducing important process quantities. Noting that nonlinear PDE models are difficult to implement in a control strategy, we seek to exploit the underlying structure of the sintering process to develop a simplified model which later on can be utilized to develop a MPC strategy for the sintering process.

The model in this section should be a *control relevant* model suitable for MPC. Reproduction of the internal states is important since a driving hypothesis is that synchronization of the model and process through state estimation is beneficial in an industrial implementation. If the states cannot be measured and estimation is hard, it is likely that the MPC algorithm performs poorly.

This section will emphasize the development of the control relevant model, and the control algorithm itself is to be investigated in chapter 4. The global PDE model discussed in section 2.3.1 is compiled from cited references. Modelling assumptions are summarized in section 2.3.2.

2.3.1 Global PDE model

A global PDE model is understood as a nonlinear PDE model describing the whole sintering bed as in figure 2.1 without explicit consideration of zones. In this section a model valid for the sintering bed is compiled from cited literature. A detailed review of the various physical and empirical relations used in the literature is included.

The following states are included in the model

$$x = \left[T_s, T_g, x_C, x_{H_2O(l)}, x_{O_2}, x_{N_2}, x_{CO_2}, x_{H_2O(v)}\right]$$

i.e. temperature of solids and gas, coke concentration in solid, liquid water content[1] and gas composition including water vapor. The gas velocity and pressure drop are not included as states in the model, see section 2.3.4 for a discussion. The hyperbolic PDE's constituting the model are, for the mass balance of gas (Patisson et al. 1991)

$$\begin{aligned}
\varepsilon\frac{\partial x_{O_2}}{\partial t} + v\frac{\partial x_{O_2}}{\partial x} &= -r_{O_2} \\
\varepsilon\frac{\partial x_{CO_2}}{\partial t} + v\frac{\partial x_{CO_2}}{\partial x} &= r_{O_2} \\
\varepsilon\frac{\partial x_{H_2O(v)}}{\partial t} + v\frac{\partial x_{H_2O(v)}}{\partial x} &= r_{H_2O} \\
\varepsilon\frac{\partial \rho_g}{\partial t} + v\frac{\partial \rho_g}{\partial x} &= M_C r_{O_2} + M_{H_2O} r_{H_2O}
\end{aligned}$$

for the mass balance of solids (Patisson et al. 1991),

$$\begin{aligned}
\frac{\partial \rho_C}{\partial t} &= -M_C r_{O_2} \\
\frac{\partial \rho_{H_2O(l)}}{\partial t} &= -M_{H_2O} r_{H_2O} \\
\frac{\partial \rho_s}{\partial t} &= -M_C r_{O_2} - M_{H_2O} r_{H_2O}
\end{aligned}$$

and for the energy balance (Patisson et al. 1991)

$$\begin{aligned}
\varepsilon\frac{\partial T_g}{\partial t} + v\frac{\partial T_g}{\partial x} &= k_1(T_s - T_g) \\
\frac{\partial T_s}{\partial t} &= k_2(T_g - T_s) + k_3 g(T_s)
\end{aligned}$$

$k_1 = \frac{A_b h_c}{\rho_g c_{p,g}}$, $k_2 = \frac{A_b h_c}{\rho_s c_{p,s}}$ and $k_3 = \frac{1}{\rho_s c_{p,s}}$ are aggregated temperature dependent parameters. The gas is assumed to be ideal and we assume plug-flow. The solid states are assumed not to move, i.e. $\partial T_s/\partial x = 0$ etc.

Kinetic parameters are only considered for coke combustion, fusion and drying in the model. The chemical reactions considered are

$$\begin{aligned}
C + O_2 &\rightarrow CO_2 \\
H_2O_{(l)} &\leftrightarrow H_2O_{(v)}
\end{aligned}$$

[1] Liquid and solids are lumped in one phase.

The kinetic relations r_i

$$\begin{aligned} g(T_s) &= (-\Delta H_r) r_{O_2} - L_v(T_s) M_{H_2O} r_{H_2O} - L_f r_f \\ r_{O_2}(T_s) &= s_{O_2} R_C(T_s) \\ r_{H_2O}(T_s) &= s_{H_2O} R_w(T_s) \\ r_f(T_s) &= s_f R_f(T_s) \end{aligned}$$

are discussed in the following subsections. The kinetic parameters of coke combustion r_{O_2}, fusion and solidification of solids r_f and condensation r_{H_2O} are not known in detail. The kinetic model of coke combustion is discussed in Parker and Hottel (1936) and Muchi and Higuchi (1972) assuming the reaction $C + O_2 \rightarrow CO_2$. The kinetics of fusion of solids is described by empirical schemes based on slag diagrams (Patisson et al. 1991) or linear schemes based on process experience (Cumming and Thurlby 1990). The kinetics of condensation of water is derived from laboratory tests (Patisson et al. 1990) or by heuristics and experience (Dash and Rose 1977), (Zou, Huang, Yang, and Chen 1995). The heat of coke combustion is released to the solid phase, see discussion in Cumming and Thurlby (1990). Limestone is not utilized in the industrial plant, and is not included in the model. We continue by specifying the model parameters in the next subsections.

Introductory relations

Some introductory relations are derived. Subscript b refers to bulk sizes, s refers to spherical particle while p refers to (non-spherical) particle. Recalling that $A_s = \pi d_s^2$ and $V_s = \frac{\pi}{6} d_s^3$, resembles $\frac{A_s}{V_s} = \frac{6}{d_s}$ and $\frac{A_s}{m_s} = \frac{6}{d_s \rho_s} = \frac{A_s/V_s}{\rho_s}$, where $m_s = V_s \rho_s$. The void fraction ε is defined by:

$$1 - \varepsilon = \frac{\text{volume of solids}}{\text{volume of bed}} = \frac{m/\rho_b}{AL} = 1 - \frac{\rho_b}{\rho_a} \tag{2.7}$$

where ρ_b is the bulk density of the bed and ρ_a is the granule apparent density (see Hinkley, Waters, and Litster (1994) for details). The volume occupied by solid (spheres) in the total volume is $V_b = (1 - \varepsilon)V$. This implies for spherical particles that $\frac{A_b}{V} = \frac{A_b}{V_b/(1-\varepsilon)} = (1 - \varepsilon)\frac{A_b}{V_b} = \frac{6(1-\varepsilon)}{d_s}$. For a non-spherical particle of the same density as a sphere occupying the same volume ($\rho_s = \rho_p = \rho$ and $V_s = V_p$), define the mean particle diameter as $d_p = \frac{6m}{A_p \rho} = \frac{6V_s}{A_p}$. Then define the form factor ϕ_f as the ratio between the surface area of a sphere and the surface area of a particle occupying the same volume:

$$\phi_f = \frac{A_s}{A_p} = \frac{d_p A_s}{6V_s} = \frac{d_p}{d_s} \leq 1$$

i.e. $A_p = A_s/\phi_f = \frac{\pi d_s^2}{\phi_f} \geq A_s$. Finally, the reaction rate per unit surface area is given by $r_{A_b} = -\frac{dn/dt}{A_b}$, and per unit mass and unit volume by $r_m = \frac{6 r_{A_b}}{d_p \rho}$ and $r_V = \frac{6(1-\varepsilon) r_{A_b}}{d_p}$, respectively. Typical values of void fraction and form factor of the present materials (prior to sintering) are $\varepsilon \in [0.4 - 0.6]$ and $\phi_f \approx 0.75$, see Rosenqvist (1983), p. 143. A_b is in the order 2000 $[m^2/m^3]$.

The harmonic mean diameter of the charge particles is calculated from mesh analysis of the raw charge as (Hinkley, Waters, and Litster 1994):

$$\frac{1}{d_p} = \frac{f_1}{d_1} + \frac{f_2}{d_2} + \cdots + \frac{f_n}{d_n} \tag{2.8}$$

where f_n is the fraction of particles between two sieve sizes with a mean diameter d_n.

Empirical relations

Parameter uncertainties are present in the global models, since essential parameters typically are determined from empirical formulas valid only under idealized conditions. In industrial sintering processes the formation of cracks and channels leads to areas where air passes through without interacting with the mass in the sinter bed. In particular, the mass, h_m, and heat, h_c, transfer coefficients are calculated from the Nusselt and Sherwood numbers. Empirical relations for Sh and Nu are stated in equations (2.10) and (2.9) (Wakao and Kaguei 1982):

$$Nu = \frac{h_c d_p}{k_{th}} = \frac{1}{\varepsilon}\left(2 + 1.1 Pr^{1/3} Re^{0.6}\right) \tag{2.9}$$

$$Sh = \frac{h_m d_p}{D_{ON}} = \frac{1}{\varepsilon}\left(2 + 1.1 Sc^{1/3} Re^{0.6}\right) \tag{2.10}$$

valid for an idealized bed with homogeneous packing. The Reynolds, Schmid and Prandtl numbers are given by

$$\begin{aligned} Re &= \frac{\rho_g v d_p}{\mu} \\ Sc &= \frac{\mu}{\rho_g D_{ON}} \\ Pr &= 0.7 \end{aligned}$$

where the value $Pr = 0.7$ holds for diatomic gases[2]. In an industrial bed the gas flowing through channels and large cracks does not interact with the solid, and the values estimated from the empirical relations for an idealized bed will deviate from the actual values. Various heuristics are utilized to overcome this in the literature, i.e. altering the constants of the empirical relations (Dash and Rose 1977), (Hoislbauer and Jaquemar 1983), (Nath et al. 1997), and introducing a scaling factor (Cumming and Thurlby 1990), (Patisson et al. 1991). According to the discussion above the gas fraction passing through possible large cracks and channels in the sinter cake does not contribute to the mass transfer and should not be included when calculating h_m from Sh. Therefore a factor φ is introduced to compensate the Sherwood relation (Schluter and Bitsianes 1962):

$$h_m = \varphi \frac{D_{ON}}{d_p \varepsilon} \left(2 + 1.1 Sc^{1/3} Re^{0.6}\right) \tag{2.11}$$

The Nusselt relation is scaled by the same factor φ.

The heat capacity of the solid and the void fractions will change in a complicated way as sintering proceeds. The specific heat capacity of sintered material for Fe-sinter is given by (Rose and Dash 1979)

$$c_{p,s} = 753 + 24 \cdot 10^{-3} T_s \quad [J/kgK] \tag{2.12}$$

As seen from figure 2.4 this linear approximation is suitable for iron ore. For manganese ore the situation is different, and the relation for $c_{p,s}$ used in the model is the dashed curve in the right part of figure 2.4. This relation was obtained as a linear combination of the data for the three Mn-oxides. The linear combination was chosen from plant data where the ore contained 47.0% MnO_2, 25.5% Mn_2O_3 and 27.5% Mn_3O_4.

This model of $c_{p,s}$ does not include other elements contained in the ore, and it does not reflect changes caused by fusion, chemical reactions and thermal decomposition of the Mn-oxides.

The temperature dependence of $c_{p,g}$ is modelled as:

$$\frac{c_{p,g}}{R} = a_1 + a_2 T + a_3 T^2 + a_4 T^3 + a_5 T^4 \quad [J/kgK] \tag{2.13}$$

where a_i values are specified in Moran and Shapiro (1993), p. 680. For ideal gases $c_{p,g} - c_{v,g} = R$ gives a similar expression for $c_{v,g}$. The temperature

[2] A diatomic gas contains two atoms in its molecules. O_2 is a diatomic gas, while $H_2O_{(v)}$ has three atoms in its molecules. A small error is introduced by this, but since $H_2O_{(v)}$ is not present in the combustion zone, the error is assumed to be of minor importance.

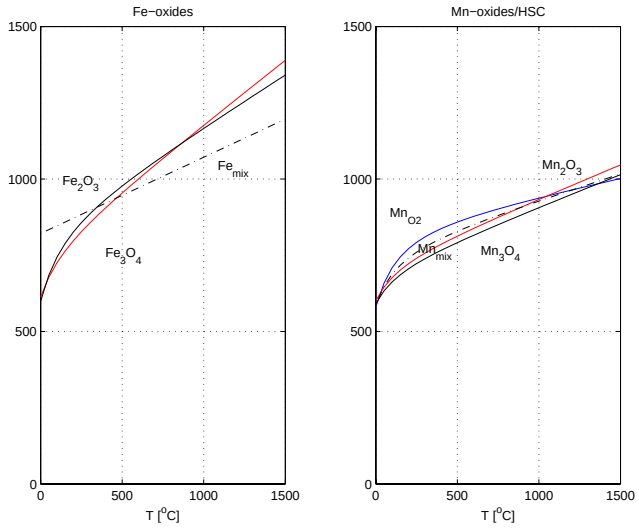

Figure 2.4: *Specific heat capacities of Mn-oxides.* The figure shows specific heat capacities of Fe-oxides (left) and Mn-oxides (right). The dashed line Fe_{mix} to the left is the straight line given by the linear approximation (2.12), while the dashed line Mn_{mix} is computed as a linear combination of the three Mn-oxides shown. Observe that the compound heat capacity of Mn-oxides is not well approximated by a straight line. Data for the Fe-oxides are taken from Perry and Green (1984), p.3-131/2, while the data for the Mn-oxides are taken from `HSC` (Roine 1997).

dependence of the viscosity μ is modelled by Sutherland's formula (White 1999), p. 771:

$$\mu = \mu_0 \left| \frac{T_g}{T_0} \right|^{3/2} \left| \frac{113 + 273.1}{113 + T_g} \right|^{m_2} \quad [kg/ms] \tag{2.14}$$

where $\mu_0 = 1.72 \cdot 10^{-5}$ [kg/ms] and $m_2 \approx 0.9$. The temperature dependency of the axial gas dispersion coefficient D_{ON} is modelled as:

$$D_{ON} = D^0_{ON} \left| \frac{T_g}{273} \right|^{m_3} \quad [m^2/s]$$

where $m_3 \approx 1.5$. The thermal conductivity of gas k_{th} is estimated from the Prandtl number:

$$k_{th} = \frac{\mu c_{p,g}}{Pr^{m_4}}$$

where $m_4 \approx 10.7$ is determined by adapting the computed k_{th} to tabulated data from Perry and Green (1984).

Coke combustion and overall heat of reaction

A number of different reactions occur during sintering of manganese ore. The equilibrium diagram in figure 2.5 shows the possible reactions at given operating conditions. These reactions contribute to the overall heat of reaction, ΔH_r. Since only coke combustion is considered to be of major importance with respect to the dynamic properties, the kinetics of reduction of Mn-ore is neglected.

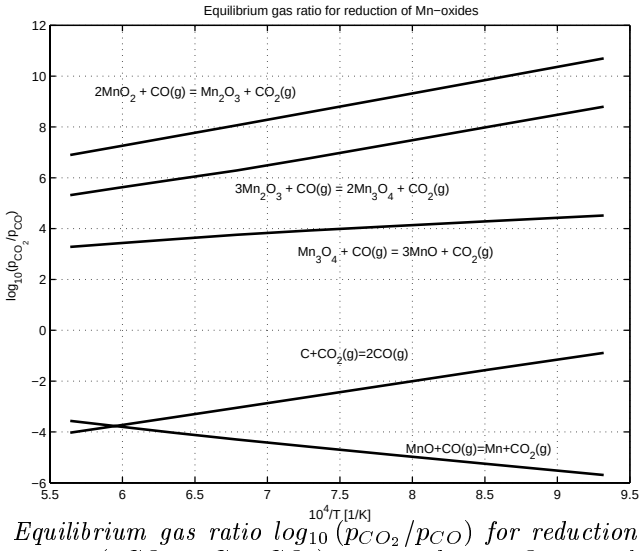

Figure 2.5: *Equilibrium gas ratio* $log_{10}(p_{CO_2}/p_{CO})$ *for reduction of Mn-oxides.* The Boudouard-line ($2CO \leftrightarrow C + CO_2$) crosses the $MnO - Mn$ line at approximately 1400^oC. The measurements conducted at the Sauda plant show that the temperature in the sintering bed is nominally below this point. Hence, reduction to Mn is not likely to occur. The slopes of the other oxide-components slants upward, and reduction in the presence of CO will occur endothermic.

In a (reducing) atmosphere of CO the following endothermic reactions are observed in figure 2.5:

$$\begin{aligned} MnO_2 + \tfrac{1}{2}CO &\leftrightarrow \tfrac{1}{2}Mn_2O_3 + \tfrac{1}{2}CO_2, & \Delta H_{r,1} &\approx -100 \text{ kJ/mol C} \\ \tfrac{1}{2}Mn_2O_3 + \tfrac{1}{6}CO &\leftrightarrow \tfrac{1}{3}Mn_3O_4 + \tfrac{1}{6}CO_2, & \Delta H_{r,2} &\approx -30 \text{ kJ/mol C} \\ \tfrac{1}{3}Mn_3O_4 + \tfrac{1}{3}CO &\leftrightarrow MnO + \tfrac{1}{3}CO_2, & \Delta H_{r,3} &\approx -22 \text{ kJ/mol C} \end{aligned}$$

where the heat of reaction is taken as the average over the temperature range of interest. These reactions move to the right during combustion and fusion, cf. figure 2.1. However, due to the large air excess during sintering, only a small amount of CO is actually present. According to Olsen (1997) the offgas ratio $\alpha = \frac{CO}{CO+CO_2}$ subsequent to combustion can become as high

as $0.25 - 0.30$. I.e. we have $\log_{10}(\frac{p_{CO_2}}{p_{CO}}) \approx \log_{10}(3) \approx 0.47$ showing by inspection of figure 2.5 that the three reactions above are possible. The Boudouard-reaction:

$$2CO \leftrightarrow C + CO_2,\ \Delta H^o_{298} = -172.5 \text{ kJ/mol C}$$

which is strongly endothermal, prevents further reduction by CO. In addition thermal dissociation at $p_{O_2} = 1atm$ occurs:

$$500^oC : MnO_2 \leftrightarrow \frac{1}{2}Mn_2O_3 + \frac{2}{12}O_2, \qquad \Delta H^o_{298} = 41.8 \text{ kJ/mol C}$$
$$800^oC : \frac{1}{2}Mn_2O_3 \leftrightarrow \frac{1}{3}Mn_3O_4 + \frac{1}{6}O_2, \qquad \Delta H^o_{298} = 16.3 \text{ kJ/mol C}$$

Expressing $X_{CO} = \frac{\alpha}{1+\alpha}X_{CO_2}$ and assuming that coal is present as pure carbon, combustion is expressed as:

$$C+(1+\alpha)O_2+3.76(1+\alpha)N_2+L \leftrightarrow (1-\alpha)CO_2+\alpha CO+3.76(1+\alpha)N_2+L$$

where excess air L consisting of 79% N_2 and 21% O_2 is utilized. Choosing $\alpha = 0.25$ gives a heat of reaction $\Delta H_{r,c} = -335$ kJ/mol C ≈ -28 MJ/kg C. The heat of combustion for coal is experimentally determined to approximately $\Delta \hat{H}_{r,c} \approx -35MJ$/kg C (Olsen 1997), i.e. $7MJ/kg$ more than indicated by the enthalpy of reaction stated above. This is attributed to volatile components present in industrial coal. The overall heat of reaction used in the model is $\Delta H_r = \Delta \hat{H}_{r,c} - \Delta H_{r,1} - \Delta H_{r,2} - \Delta H_{r,3} \approx -270\ kJ/mol\ C$ which is close the reported values for Fe-sintering in the literature. Note that thermal dissociation is not included since it is completed before coke combustion proceeds.

The reaction rate of coke combustion is important due to the large temperature gradients present in grate sintering. Coke combustion is a heterogenous reaction, and the overall combustion rate is controlled by two physical phenomena; the chemical combustion rate and the gas transport rate to the individual coke particle. The combustion rate for a coke particle is governed by (Parker and Hottel 1936), (Muchi and Higuchi 1972)

$$k_r = A_f \cdot e^{-E/RT_s}\sqrt{\tau_s} \cdot x_{O_2} \tag{2.15}$$

with the frequency factor $A_f = 6.53 \cdot 10^5\ m/s\sqrt{K}$ and $E = 185\ kJ/mol$. τ_s is the gas temperature at the solid surface. This relationship is extensively used in the literature, but it assumes the first order reaction $C + O_2 \rightarrow CO_2$ ($\alpha =$

0) and that combustion commences on the coke particle surfaces (Muchi and Higuchi 1972). Both of these assumptions are inconsistent with industrial experience. Still, equation (2.15) is used in the present model assuming $\tau_s = T_s$. The activation energy, however, is reduced to approximately 70 kJ/mol to fit the model to the measured data. By using the ideal gas law (2.15) can be written in terms of p_{O_2} instead of x_{O_2}.

Since the transport of reactants and products to and from the coke particle surface can be rate limiting at elevated temperatures, the overall reaction rate, including both chemical and transportation phenomena, is modelled by the relation (Muchi and Higuchi 1972):

$$K_r = \frac{k_r h_m}{k_r + h_m}$$

where h_m $[m/s]$ is calculated from equation (2.11). Pore diffusion D_{pd} may be included similarly according to

$$\frac{1}{K_r} = \frac{1}{k_r} + \frac{1}{h_m} + \frac{1}{D_{pd}}$$

This is not included in the present model. This gives the overall combustion rate of coke:

$$R_r = 4\pi \left(d_{p,C}/2\right)^2 n_C K_r x_{O_2} \tag{2.16}$$

where n_C $[1/m^3]$ is the number of coke particles per unit bed volume and $d_{p,C}$ is the average coke particle diameter.

Fusion

Particle growth due to fusion is modelled according to Dash and Rose (1977) as

$$d_{p,f} = d_p(1 + k_f L_f)$$

where k_f is a constant to be determined experimentally and L_f is the enthalpy of fusion (melting). If significant melting does not occur, the scheme of Dawson (1993) where smaller particles attaches to larger particles is assumed to be prevailing. This is termed heterogeneous texture, i.e. the mean particle size of sintered particles is in the region of the larger sieve sizes of the charge. However, if significant melting occurs the situation will be completely altered, with formation of larger continuous blocks which again

forms a homogeneous texture. This is assumed to correspond to a texture of very fine particles, with particle sizes in the region of the finest mesh sizes present in the charge. The transition between these two regimes is controlled by the sintering temperature (Dawson 1993). Particle growth is not included in the present model.

Fusion itself is complex and is modelled by the following equations (Patisson et al. 1991):

$$\begin{aligned} r_f &= \rho_s \tfrac{dF_i}{dT_s} \cdot \tfrac{\partial T_s}{\partial t} \\ F_1 &= (1 - x_h)\left[a_0 + a_1(T_s - T_{df}) + a_2(T_s - T_{df})^2 + a_3(T_s - T_{df})^3\right] \\ F_2 &= F_m \tfrac{T_s - T_{fs}}{T_{s,m} - T_{fs}} \end{aligned}$$

where the function F_1 denotes the liquid fraction during melting, x_h is the hematite fraction of the ternary system $FeO - MnO - SiO_2$, T_{df} is the incipient melting temperature, and a_i are dependent on the basicity index and can be found from a liquidus surface diagram, see Verein Deutscher Eisenhüttenleute (VDEh) (1995), figure 3.286. During solidification, composition is assumed to be constant and the simplified relation F_2 is adopted where F_m is the liquid fraction at $T_{s,max}$ and T_{fs} is the temperature at the end of solidification.

Using the liquidus surface diagram is impractical and the present model only considers linear schemes based on process experience (Cumming and Thurlby 1990). The present model implements a quadratic approximation where the parameters were tuned to fit the data.

$$\begin{aligned} r_f &= (\hat{a}_1 \cdot (T_s - T_{df}) + \hat{a}_2 \cdot (T_s - T_{df})^2) \cdot \rho_s \quad [mol/s] \\ L_f &= 255 \quad [kJ/mol] \end{aligned} \tag{2.17}$$

with $T_{df} = 1250^o C$ and different values of $\hat{a}_i$ for melting and freezing following the scheme of Nath, Da Silva, and Chakraborti (1997). This simplified model gives a kink in the temperature profile when the model switches from melting to freezing. Since these parameters were selected without reference to liquidus diagrams, the model is conceptual, and refinements should be considered since the sinter quality is governed by the shape of the temperature profile. Note that fusion has a significant influence on the falling edge of the temperature profile.

Water

Water evaporates and condensates according to

$$H_2O_{(l)} \leftrightarrow H_2O_{(v)} \tag{2.18}$$

The drying model is governed by (Patisson, Bellot, and Ablitzer 1990)

$$\begin{aligned}
\tilde{R}_R &= R_M \cdot P(W_r) \\
P(W_r) &= 1-(1-W_r)(1-1.796W_r+1.0593W_r^2) \\
R_M(T_w) &= \frac{A_b h_c}{M_{H_2O}\Delta H_v(T_w)} \\
R_R(T_s) &= \frac{A_b k_{th}}{RT_g}\left(p_{v,sat}(T_s)-p_{H_2O}\right) \\
p_{v,sat}(T_s) &= exp\left(25.541-\frac{5211}{T_s}\right) \\
k_{th} &= \frac{h_c T_g}{3.155 p_g \sqrt{(1-0.24x_{lg})\left(1+\frac{x_{lg}}{7}\right)}(1-x_{lg})}
\end{aligned} \tag{2.19}$$

where $W_r = W/W_{cr}$ and the given coefficients of $P(W_r)$ determined from laboratory tests (Patisson, Bellot, and Ablitzer 1990) are valid for a Fe-charge. Coefficients for Mn must be determined experimentally[3]. The wet-bulb temperature T_w can be calculated by solving a nonlinear equation (Patisson et al. 1990). The present approach uses the approximation (Rose and Dash 1979)

$$T_w = 293.4 + 324.6 \cdot W - 594.1 \cdot W^2 + 292.1 \cdot W^3$$

which is based on tabulated data from Perry and Green (1984). x_{lg} is the logarithmic mean of the molar fraction of vapor in the bulk gas and at the saturated surface:

$$x_{lg} = \frac{x_{v,g} - x_{v,s}}{ln\frac{x_{v,g}}{x_{v,s}}}$$

Since $x_{v,s}$ is unknown the present model implements $x_{lg} = 0.5 \cdot x_{v,g}$. The heat transfer coefficient h_c for the water-vapor system may differ from the overall heat transfer coefficient from equation (2.9). This is not considered in the present model. If $R_R > 0$ and $W \geq W_{cr}$ (falling rate) $r_{H_2O} = \tilde{R}_R$. If $R_R > 0$ and $W < W_{cr}$ (constant rate) $r_{H_2O} = R_R$. The heat capacity of the moist offgas is modelled by (Perry and Green 1984), p. 12-3:

$$c_p = 0.24 + 0.45\omega$$

where $\omega = 0.622\frac{p}{p-p_{H_2O}}$. Simplified schemes for R_R are found in Nath, Da Silva, and Chakraborti (1997) and Zou, Huang, Yang, and Chen (1995). The latent heat of evaporation is approximated by (Patisson et al. 1990)

$$L_v(T_w) = 3.1563 \cdot 10^6 - 2396.6 \cdot T_w \;\; [J/kg]$$

[3] The present model is implemented with the Fe-coefficients.

2.3.2 Modelling assumptions

In this section the main assumptions on which the global `PDE` model is based, are summarized.

i. *The total pressure p is assumed to drop linearly down through the bed. The gas velocity v is considered as a control action and is assumed not to be distributed. I.e. all levels experience the same velocity at any given time instance.*

ii. *The manganese ore is treated as one substance, i.e. various oxidation levels of manganese are not considered. Likewise other substances contained in the ore are neglected. More specifically, the various oxidation levels of manganese are treated as inerts, and only influence on the specific heat capacity of the ore.*

iii. *Complete coke combustion is assumed, hence CO is not generated. The gas N_2 is considered to be inert.*

iv. *Gas dynamics are assumed to be plug-flow. The gas dynamics are approximated by first order finite differences in the spatial dimension.*

v. *Radially homogenous mixing is assumed. Adding another dimension should include Fourier's law to allow conduction in the solid phase. Including radial conduction is not considered to be of major importance in a control relevant model. Channeling effects may have a significant influence on the sintering process, and taking this into account is a possible extension to the present model. The present model only considers this through the scaling of the Sherwood relation (2.10).*

vi. *Solid particles have homogenous density and temperature. Coke particles are porous, but the void fraction of coke is considered as part of the gas phase. Small particle sizes gives uniform temperature inside particles, due to high thermal conductivity.*

vii. *Fusion (i.e. sintering) is modelled by a polynomial approximation. The polynomial was selected by adaption to measurements.*

viii. *Coke combustion and water evaporation/condensation are the only chemical reactions considered. Coke combustion generates the heat necessary for sintering/fusion to occur, and its kinetic parameters control especially the leading edge of the temperature front. A water evaporation model based on the literature is implemented. Other*

reactions influence on the overall energy requirements, but are expected to have little influence on the dynamic characteristics. Decomposition of limestone is not modelled, but can be included without altering the structure.

ix. *Thermodynamic relations are approximated by polynomials.*

x. *Slump and shrinkage are neglected. The sinter bed at Sauda is observed to shrink about 5cm during a batch. This alters the heat exchange volume above the fusion zone, cf. figure 2.1. A small reduction in the cooling zone is expected to have minor influence on the conditions inside the fusion zone. Slump is only relevant for travelling grate sintering.*

2.3.3 Comparison with existing models

There are three good models presented in the literature; (Hoislbauer and Jaquemar 1983), (Cumming and Thurlby 1990) and (Patisson, Bellot, Ablitzer, Marlière, Dulcy, and Steiler 1991). The model by Hoislbauer and Jaquemar (1983) has the best numerical qualities, while the model by Cumming and Thurlby (1990) is the most extensive. Note that the paper by Hoislbauer and Jaquemar (1983) focuses on numerical properties, and that the model is comparable in complexity to the one given by Patisson et al. (1991). Cumming's model gives no data for the numerous chemical reactions considered. Hence, reproducing the results is hard. The numerical simulations are reasonable but seem to be based on simplifying assumptions not fully documented. The present model is based on Patisson et al. (1991). Their model includes the basic `PDE` model restated in section 2.3.1, the fusion, coke and water kinetics considered herein. Hence, the model by Patisson et al. (1991) includes the most important physical phenomena, and gives data for the water kinetics. The fusion data is not given, and the present model uses a simplified polynomial approximation. No models in the literature gives gas velocity simulations, and all use Ergun's relation. Ergun's relation is discussed in section 2.3.4 below.

2.3.4 Computation of the control input v

As discussed in section 2.2.2 the gas velocity is considered as a control input. The simplest approach, which is the method implemented in chapter 4, is to consider v directly as a control input. In an industrial implementation

the control actuator will typically be a valve disc in the outlet gas pipe. Manipulating the valve disc will alter the pressure drop across the bed which is a measured quantity. Therefore it is necessary to model the relationship between the pressure drop and gas velocity, and include gas velocity as a state in the model. The gas velocity model is derived below as

$$\frac{dv}{dx} = \frac{RT_g}{p}\sum_{i,j}\hat{r}_{i,j} + \frac{v}{T_g}\cdot\frac{dT_g}{dx} - \frac{v}{p}\cdot\frac{dp}{dx} \tag{2.20}$$

The ideal gas law with molar flow n_i in $[mol/m^2s]$ and apparent velocity v in $[m/s]$ is

$$n_i = \frac{p_i v}{RT_g}$$

Differentiation gives

$$\frac{dn_i}{dx} = \frac{p_i}{RT_g}\cdot\frac{dv}{dx} + \frac{v}{RT_g}\cdot\frac{dp_i}{dx} - \frac{p_i v}{RT_g^2}\cdot\frac{dT_g}{dx}$$

assuming $v > 0$ and reorganizing

$$\frac{dp_i}{dx} = \frac{RT_g}{v}\cdot\frac{dn_i}{dx} - \frac{p_i}{v}\cdot\frac{dv}{dx} + \frac{p_i}{T_g}\cdot\frac{dT_g}{dx}$$

Using

$$\frac{dn_i}{dx} = s_i r_i$$

where s_i is a stoichiometric constant gives

$$\frac{dp_i}{dx} = \frac{RT_g}{v}\cdot s_i r_i - \frac{p_i}{v}\cdot\frac{dv}{dx} + \frac{p_i}{T_g}\cdot\frac{dT_g}{dx}$$

The total molar flow is

$$\frac{dn_t}{dx} = \sum_i s_i r_i$$

and

$$v = \frac{n_t RT_g}{p}$$

Differentiation and using the expression for n_t gives

$$\begin{aligned}\frac{dv}{dx} &= \frac{RT_g}{p}\cdot\frac{dn_t}{dx} + \frac{n_t R}{p}\cdot\frac{dT_g}{dx} - \frac{RT_g n_t}{p^2}\cdot\frac{dp}{dx} \\ &= \frac{RT_g}{p}\cdot\sum_i s_i r_i + \frac{v}{T_g}\cdot\frac{dT_g}{dx} - \frac{v}{p}\cdot\frac{dp}{dx}\end{aligned} \tag{2.21}$$

The term $\sum_{i,j} \hat{r}_{i,j}$ in equation (2.20) occurs from the matrix-vector product $S \cdot r$ when there are more components present. Assuming infinite fast gas dynamics a mechanistic energy balance yields the pressure drop as the isothermal Ergun's relation:

$$\frac{dp}{dx} = 150\frac{\mu v(1-\varepsilon)^2}{(\phi_f d_s)^2 \varepsilon^3} + 1.75\frac{\rho v^2(1-\varepsilon)}{\phi_f d_s \varepsilon^3} \tag{2.22}$$

For small particle sizes, low fluid velocity and high fluid viscosity, the flow will be laminar (viscous). For larger particle sizes, high fluid velocity and less viscous fluids, the flow will be turbulent. In grate sintering, the flow is a priori expected to vary continuously from laminar to turbulent flow as the sintering batch proceeds. The first term in Ergun's formula refers to the viscous and the second to the turbulent pressure drop.

Given the measured pressure drop across the bed the velocity is calculated by equation (2.21) under some simplifying hypotheses discussed next. Considering dp/dx as a control input (i.e. a linear decay from top to bottom, the bottom pressure being a manipulated variable), assuming known (constants) ε and $\phi_f d_p$, and computing μ by equation (2.14). Hence, all terms of equation (2.21) are defined and dv/dx can be integrated to give the state v. However, assuming a linearly decaying pressure drop across the bed is not physical since it neither accounts for the distributed temperature profile, nor evaporation of water which produces additional gas molecules. In addition equation (2.22) is derived for isothermal conditions, while the temperature in the sintering bed is not constant. Since both the density and viscosity of the gas are temperature dependent, the extended Ergun relation of Wonchala and Wynnyckyj (1987) should be considered. Calculation of the pressure profile dp/dx is also dependent upon the parameters ε and $d_p = \phi_f d_s$, which are dependent on the degree of melting.

As outlined in Rosenqvist (1983), p. 146, Ergun's relation allows calculation of ε and d_p if two sets of measurements are available. Here dp/dx is the pressure drop per unit height of the bed, μ and ρ are viscosity and density of the fluid, and v is the *superficial* velocity of the fluid, i.e. velocity referred to the empty bed. Laboratory tests for accurate measurement of ε and d_p are suggested in Hinkley, Waters, and Litster (1994) and Hinkley, Waters, O'Dea, and Litster (1994). Hinkley, Waters, and Litster (1994) also suggests that other constants should be utilized in the Ergun relation when wet charge is considered. The average particle diameter, d_p, will not have a physical interpretation once melting in the fusion zone has occurred, see the discussion on fusion in section 2.3.1. Still, all cited models utilize some equivalent particle diameter when melting has occurred.

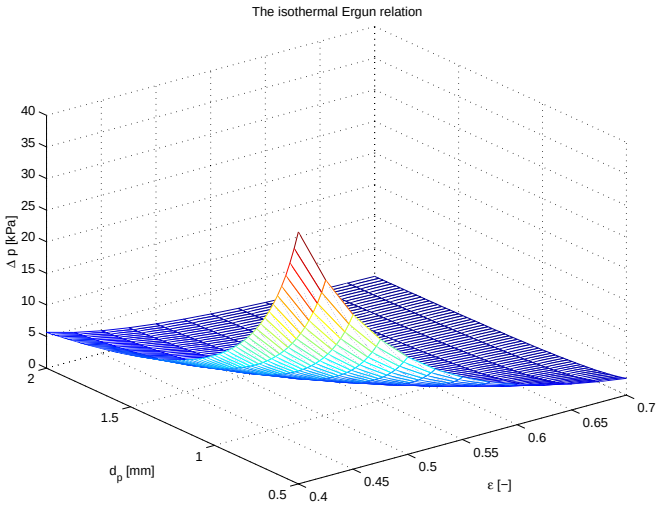

Figure 2.6: *Ergun's relation.* The figure shows Ergun's relation for $\mu = 1.7e-5 Pa\,s$, $\rho = 1.29 kg/m^2\,s$, $L = 0.4m$, $v = 1m/s$. The nonlinearity in Ergun's relation is considerable especially for small void fractions, ε, and small particle sizes, d_p. Noting that since the measurements presented in section 2.4 shows a typical pressure drop of 5kPa, the nominal values of ε and d_p cannot be too low. $\Delta p = 5kPa$ and $\varepsilon = 0.4$ implies a $d_p = 2.2mm$, while $\varepsilon = 0.6$ implies a $d_p = 0.6mm$.

Figure 2.6 shows the nonlinearity of Ergun's isothermal relation as a function of ε and d_p. As the discussion in the figure caption indicates the Ergun relation is moderately nonlinear in the region of interest. Using constant values for ε and d_p is reasonable as an initial approximation in a control relevant model. A model for ε and d_p should comply to the following heuristics (Olsen 1997)

- ε_{ch} remains unchanged during the batch
- ε_f is altered due to the particle growth caused by fusion. The fusion zone has the lowest permeability and thus has the lowest void fraction.
- ε_{hex} is the largest.

where $\varepsilon_f < \varepsilon_{ch} < \varepsilon_{hex}$ are the void fractions in the *fusion* (zone 2 and 3), *charge* (zone 4 and 5) and *heat exchange* (zone 1) as referred to figure 2.1. The literature only give qualitative models for void fraction and particle growth (Cumming and Thurlby 1990).

A simplified approach compared to including v as a state and applying Ergun's relation is to compute v directly from the quadratic equation

(2.22) in v for a specified dp/dx. This approach should be validated against the complete model outlined above. Due to the added complexity of the extended Ergun relation and the uncertainty in modelling ε and d_p, the velocity v itself is considered as a control action in the following.

2.4 Experiments and measurement campaigns

Experiments conducted at the sintering plant in Sauda, Norway, were initially designed to collect data which could be used to obtain the following goals:

- Identification of an input-output model.
- Adaption and validation of the nonlinear `PDE` model.
- Verification of the connection between charge composition, process operation and product quality.
- Provide realistic feedback (process simulator) for testing nonlinear `MPC`.

In addition thorough knowledge of the sintering process was gained. The first goal was not achieved, since the unknown recycle led to large disturbances. Hence, the information content in the measured output was dominated by disturbances. The measurements allowed calculation of the recycle which showed that the recycle was much larger than expected. Initially the recycle was assumed to be in the order 20-30% with small variations. In section 2.6 the actual recycle is calculated to vary in the range 0-53%.

In addition the process output "production rate" was not well-defined in terms of on-line measurements at the time the experiments were conducted. Model adaption and validation has been performed by visual comparison of the measurements against the simulations. This was an iterative procedure where model parameters were manually updated, the model simulated, and the result compared to the measurements, see figures 2.13 and 2.15 below. The relationship between in- and outputs has been verified by the experiments, and this motivated a mathematical formulation of a control objective suitable for use in nonlinear `MPC`. Nonlinear `MPC` is documented in chapter 4. The measurements were not used directly as process simulator, but the model was adapted to the data. Hence, the model is a realistic representation of the industrial process.

In section 2.4.1 follows a description of the experiment setup, including a more detailed description of the plant and the available instrumentation

relevant for the experiments. The measurements conducted at the plant are documented in section 2.4.2. Data analysis is documented in section 2.6.

2.4.1 Plant and experiment description

The industrial sintering plant under investigation is located in Sauda, Norway and was owned by the Elkem ASA company at the time the experiments were conducted. It is now owned by Eramet Norway AS, a subsidiary of the Eramet Group. Two independent measurements campaigns were carried out. The first in March 1998, and the second during the Spring of 1999.

- The first campaign investigated the effect of altering the gasflow through the bed without altering the charge composition, see section 2.4.3.
- The second campaign investigated the effect of altering the charge composition without altering the gasflow. These experiments occupy the bulk of the remaining part of this chapter, in particular section 2.4.2. Note that some of the measurements documented in section 2.4.2 were available during the first measurement campaign as well.

The plant layout together with relevant instrumentation is described in figures 2.7 and 2.8. The sintering plant is part of a larger plant for electrical reduction to $FeMn$ and $SiMn$, and the sintering plant is designed to provide suitable feed for the electrical furnaces at the plant. The sintering plant in Sauda is a batch-wise plant with six parallel Greenawalt-pans.

To allow the set-point change in coke composition to propagate into the pan, each experiment was designed as outlined in figure 2.9. Letting each experiment span three consecutive batches was based on the assumption of 20-30% recycle, i.e. accepting a bias in the coke composition, and compensating for this once the actual recycle has been calculated. Since each experiment spans 3 consecutive batches, it is relevant to investigate the relationship between any two consecutive batches. From the difference in fresh feed between two consecutive batches given in table 2.1, an empirical probability distribution function F was calculated, see figure 2.10. Based on the third column of table 2.1 a bootstrap estimate (Efron and Tibshirani 1986) was computed. 1000 new observation series was generated. Note that since there are only 11 observations the variance estimates are uncertain. The expectation, standard deviation and `BC` (bias-corrected) confidence intervals are given in table 2.2.

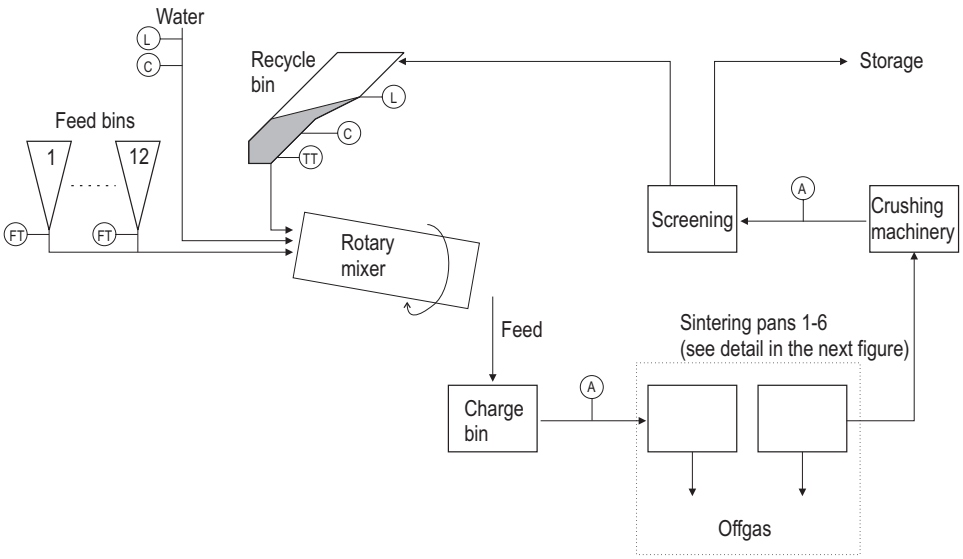

Figure 2.7: *Detailed sintering plant.* The figure shows the details of the sintering plant that are important for the experiments, together with the available instrumentation. Starting from the upper left there are 12 feed bins containing coke, Mn-ore and various scrap from the plant. The mass-flow from each bin is measured automatically (FT). Added water is manually controlled (C) and the water flow is measured manually by a flotameter (L). The amount of recycle is unknown, as indicated by the irregular and unknown shape of the recycle bin. The recycle level is measured manually by lowering a mechanical device into the bin (L). The recycle output is controlled from the control room by adjusting the set-point of a vibrating device (C). The recycle temperature is measured automatically by a sensor attached to the outer wall (point-welding) of the recycle bin (TT). Inputs are mixed in a rotary mixer and stored in the charge bin. The feed is charged into the six sintering pans. The moisture content of the feed is measured manually by a drying test of a sample taken from a pan (A). Details of the instrumentation of the sintering pans inside the dotted box are shown in figure 2.8. After completion of the sintering process the produced sinter cake is crushed into manageable lumps by a mechanical chrushing device. The sinter is then screened, and the fines ($d_p < 3mm$) are recycled, while coarser sinter is stored or fed to the furnace. Manual samples of the produced sinter are taken by shuffling approximately 40-50kg of sinter into containers before screening (A).

2.4.2 Measurements at plant: Campaign 2

This section summarizes the measurement campaign carried out at the sintering plant in Sauda during the Spring of 1999. The measurements logged during each experiment are shown for experiment 1 in figure 2.11 and A.1 in appendix A.2. The sampling period was 5 seconds. In addition to the

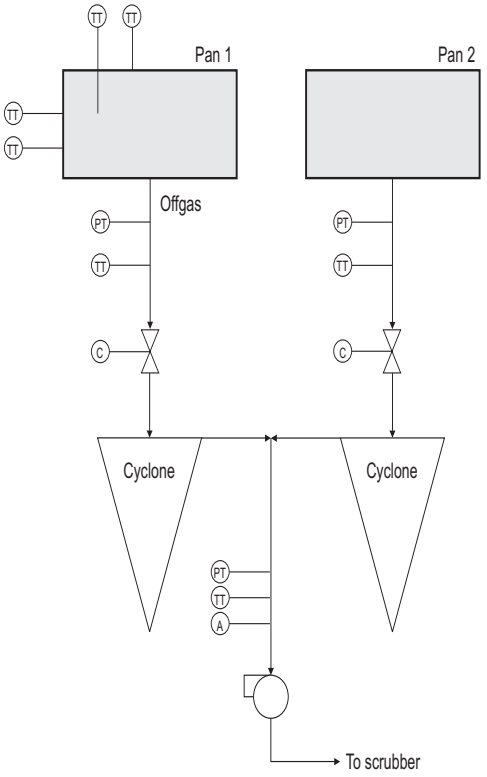

Figure 2.8: *Instrumentation of sintering pan.* The figure shows the details of the instrumentation of the sintering pan that were available for the experiments. The figure shows the details of the block inside the dotted box in figure 2.7. The experiments were conducted in pan no. 1 of the six pans. Temperature measurements on the top and $15cm$ down into the pan by K- and S-elements respectively, in addition to sensors (point-welding) on the outer wall at depths $15cm$ and $30cm$ were logged (TT). The offgas temperature and differential pressure were logged for both pan 1 and 2 (TT) and (PT). The offgas-flow is controlled from the control room by a chocking valve (C). This is operated according to a prescribed recipe. The pressure after the cyclones was logged (PT), with a pitot-element (Pt-100) intended for flow-measurements. The gas flow must be compensated for temperature, measured by (TT) at the same location. Just below this point a thin pipe was inserted allowing for gas-samples to be taken (A). A fan was attached to this tube, and moisture and composition tests were taken by sucking out offgas.

measurements shown in the figure mass-flows from the feed-bins and the wall-temperature of the recycle bin were logged. Various manual samples such as the recycle level, the water content of the charge and offgas and sinter were also taken. Some details of the curves in figure 2.11 deserve special attention. In the first subfigure the solid temperature and ignition gas temperature in pan 1 are shown. The first batch starts at approximately 17 minutes as seen from the ignition gas temperature (dashed curve). Just before 20 minutes the ignition hood is removed and the S-element (cf. caption of figure 2.8) is inserted into a ceramic tube placed into the bed prior to ignition. At approximately 53 minutes the S-element is removed and pan 1 is fetched to be emptied into the crushing machinery. This pattern is repeated for each batch.

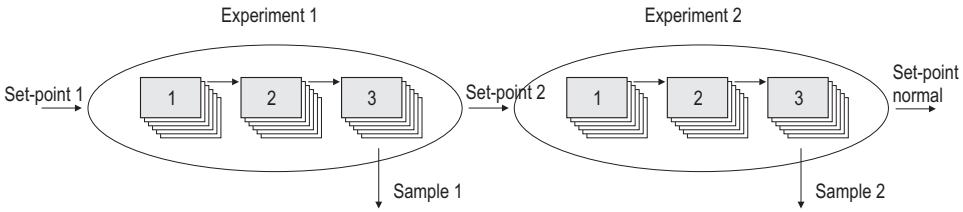

Figure 2.9: *Batchwise experiments.* The figure shows how the batchwise plant influences on the conducted experiments. The horizontal axis of the figure is time, and one batch lasts approximately one hour. To the left, experiment 1 starts by setting the feed set-point to the desired value. Due to the hold-up of the plant, the feed charged into the pans will not be influenced by the set-point change until approximately batch 2 or 3 (depending on the amount of recycle which is unknown). Note that each batch (labeled 1, 2, 3) consists of six pans as indicated on the figure by the six overlapping boxes. After batch 3 is finished, pan 1 is emptied into the crushing machinery and a sample is taken by shuffling a large amount of sinter into a container. Now experiment 1 is completed, and experiment 2 commences by setting the feed set-point to its desired value. Note that the set-point for experiment 2 may be altered earlier than indicated due to the plant hold-up. Since the recycle is unknown a degree of conservatism wrt. set-point alterations was introduced to guarantee that the feed composition in the pan in batch 3 actually was as desired.

In subfigure 2 it is observed that the batch in pan 2 starts about 5 minutes prior to the batch in pan 1. This is due to the sequential nature of the batch cycle at the plant, and was not altered during the experiments. The pressure does not vary substantially during the batch, despite the generation of gas molecules caused by evaporation. This indicates that the gas velocity increases during the batch. The pressure difference between pan 1 and 2 is discussed in the figure caption. No large leakages in the pipes were observed.

Towards the end of each batch the offgas temperature is expected to approach the gas temperature in the combustion zone. Subfigure 3 shows considerably lower offgas temperatures than the solid temperature shown in subfigure 1.

The slowly falling offgas temperature during the first part of a batch is attributed to a change from laminar to turbulent flow due to the lower differential pressure during the initial phase of the batch, combined with the large difference in heat-capacities between the solid and gas phase of the packed-bed system. I.e. initially the Reynolds-number is low giving poor heat-exchange and the hot ignition gas is transported through the pan without loosing to much heat to the solid phase. Once the pressure difference is increased, the Reynolds-number increases and the heat-exchange between the two phases is increased. Hence, the offgas temperature is expected to

Experiment	Fresh feed [t]		
	s_2	s_3	Δs
1	28.1	22.4	-5.7
2	24.3	21.0	-3.3
3	19.7	17.3	-2.4
4	-	14.9	-
5	16.6	20.0	3.4
6	25.8	29.0	3.2
7	15.0	26.1	11.1
8	27.6	21.1	-6.5
9	25.1	26.4	1.3
10	-	16.5	-
11	27.6	17.4	-10.2
12	28.5	20.2	-8.3
13	18.4	31.2	12.8

Table 2.1: *Measured fresh feed from the feed bins.* The table shows measured fresh feed from the feed bins for series 2 and 3 for all experiments. The last column is used to estimate the variance between consequtive series shown in figure 2.10. Due to logging errors for the second series of experiment 4 and 10 they are not included in the variance estimates.

	Non-parametric	Parametric
E{F}	-0.304	-0.381
SD	2.28	2.08
BC_{lower}	-3.88	-3.74
BC_{upper}	3.73	3.21

Table 2.2: *Bootstrap estimates.* The table gives the computed expectation, standard deviation and BC (bias corrected) confidence intervals for the bootstrap estimates shown in figure 2.10.

fall abruptly. As the figure shows it falls smoothly, and this is attributed to the difference in heat capacity.

2.4.3 Control action experiments: Campaign 1

This section contains a summary of an unpublished report (Martinsen 1998b) which summarizes measurements taken at the sintering plant at the Elkem (now owned by Eramet) plant in Sauda from 3rd to 5th of March 1998.

Air is sucked through the six sintering pans by means of three fans,

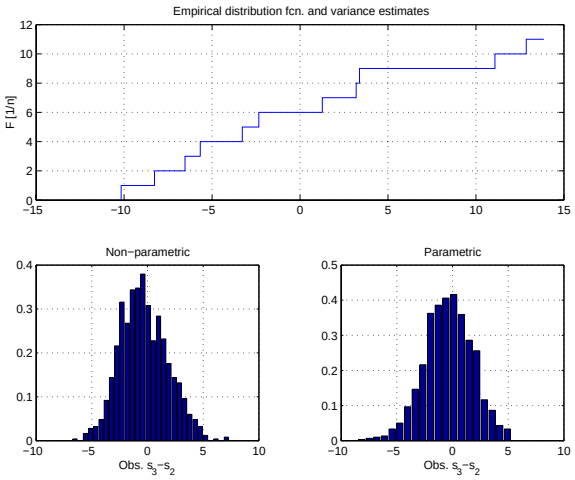

Figure 2.10: *Bootstrap estimates.* The figure shows the empirical distribution function (upper part) formed by sorting the data in the third column of table 2.1. The estimates in the lower part shows that the parametric (right part) normal distribution estimate is well suited for estimating the variance. This is concluded due to the similarity with the non-parametric estimate (left part). The normal distribution generated 1000 new datasets from the distribution $X^* \sim N(\bar{X}_n, Y_n)$ where $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, $Y_n = \frac{1}{n}\sum_{i=1}^{n} X_i^2$ and $n = 11$. The estimates show that the variance is considerable, and it cannot be expected that two consequtive batches are well correlated. I.e. large variations in fresh feed caused by large variations in recycle, makes it difficult to predict the charge composition in each batch.

each fan being connected to two pans. This produces a differential pressure between atmosphere and the pipes connecting the pans to the fan inlet. Temperature and pressure are measured inside these pipes for each of the pans, hereafter denoted off-gas (bottom) temperature, $T_{i,bot}$, and pressure, $p_{i,bot}$ where i indexes pans 1 to 6, see figure 2.8. Gas-flow through the pans is controlled by means of valve discs for each pan. These valve discs were operated manually during the experiments to obtain increased gas-flow as compared to nominal operation. The effect of increased gas-flow on sintering time gives insight into how the disc valve should be operated if considered as an actuator. Possible negative side-effects of increased gas-flow are mainly the possibility of bed-collapse. The sampling period was preset to 5 seconds, and the sintering batch time was set to 38 minutes. The ignition time for each pan was set to 150 seconds. Figure 2.12 shows temperature measurements in pan 1 logged with two different pressure drops across the bed. As seen from the figure the batch with increased pressure

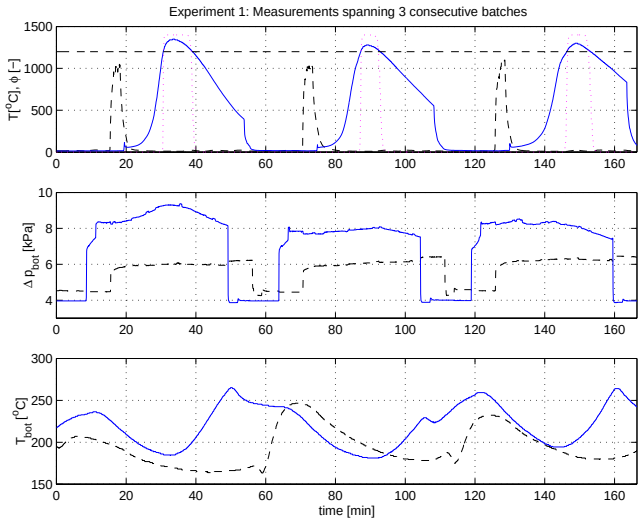

Figure 2.11: *Experiment 1: Measurements.* The figure shows the logged measurements during all three batches included in experiment 1. Subfigure 1 shows the temperature measured inside the pan $15cm$ down by a S-element (solid), and the temperature measured on top during ignition by a K-element (dashed). The dashed line at 1200^oC indicates the fusion temperature T_{fu}. The dotted curves show the scaled value of the objective function $\phi(T_s)$. Subfigure 2 shows the differential pressure measured beneath pan 1 (solid) and pan 2 (dashed). The large difference in level between the two parallell pans can be due to instrumentation errors, valve malfunction, channeling inside the pans or leakages. The third subfigure shows the temperature measured beneath pan 1 (solid) and pan 2 (dashed).

drop rises faster to its peak value, but due to measurement noise the results are uncertain. In these experiments temperature profiles were measured by K-elements inserted directly into the bed. K-elements are not designed for temperatures above 1250^oC, and the measured values may deviate from the actual values. The non-smoothness of the solid temperature profile in figure 2.12 is attributed to breakage, since at approximately 41 minutes no further measured values were logged. The four "steps" towards the end of the measured $p_{1,bot}$ in figure 2.12 are caused by couplings between the different fans, possibly through the electric power supply or the common scrubber following the three fans. The four steps correspond to batch completion of pan 6, 5, 4 and 3 respectively. Pan 2 was not included in the batch cycle during this experiment. Such steps did not appear during the second measurement campaign carried out a year later, see section 2.4.2.

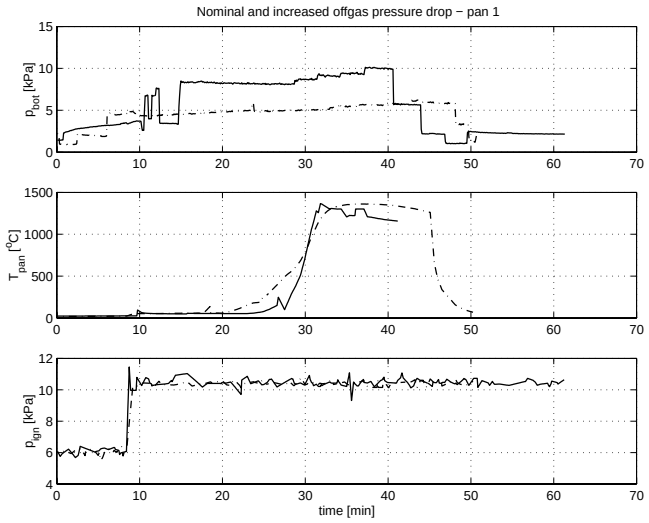

Figure 2.12: *Control action experiments.* The figure shows the logged measurements of one batch duration in pan 1 for two consecutive batches. Dashed-dotted lines in all three parts shows measurements taken during the first batch with p_{bot} at its nominal value. Solid lines in all three parts shows measurements taken during the second batch with p_{bot} at an increased value. The upper part shows the measured pressure drops across the bed. The spikes from approximately 10-12 minutes indicates that the actuator was manually adjusted to the desired value by trial and error. The middle part shows temperature profile measurements 20 cm down into the bed, while the lower part shows the ignition gas pressure. The lower part is included for syncronization issues; when p_{ign} increases at approximately 9 minutes, the ignition is shut off and the K-elements are inserted into the charge. Ignition starts at approximately 6 minutes. The abrupt termination of the solid temperature profile indicates a broken K-element, and the remaining samples has been removed. The non-smooth corner in the dashed-dotted temperature profile at approximately 45 minutes was caused by manual removal of the K-element at the end of the batch.

Another run (not shown) in a neighboring pan showed that although the actuator was set at its maximum value, the pressure drop did not increase. Despite this, the difference in measured temperature profile were significant. Hence, the results reported from the experiments are not conclusive in the sense that it cannot be judged whether the differences in measured temperature profiles are caused by the pressure drops or by other factors. From figure 2.10 consecutive batches cannot be expected to be well correlated, mainly due to varying recycle and channeling effects. Further investigations into the relation between the pressure drop and temperature profile require

further experiments, preferably on laboratory equipment were disturbances are easier to handle.

Apart from the deviation case documented below and in figure 2.12, the plant was in nominal operation during the experiments. The production rate was logged to be above average with a low recycle, and a high production rate. As a consequence, the measurements are representative to the nominal operation mode. By nominal operation it is understood that the plant is operating without large deviations from the average operating conditions observed throughout a year. The measurements taken with increased suction have a larger air-flow through the pans (not nominal operating mode). These experiments were conducted in order to investigate whether increased gas-flow would lead to faster sintering batch times without collapsing of sintering beds or other undesired effects. The measurements are presented after filtering and interpolation to reduce measurement noise.

The results do show that increasing the nominal gasflow will not necessarily increase the sintering batch time. The present nominal value coincides with the maximum allowable value when all pans are operated simultaneously. This maximum value is upward limited by the fan capacity. Hence, pan 2 was excluded from the experiments shown in figure 2.12 to allow an increased pressure drop.

There are significant differences between pan-positions that should be investigated thoroughly. In practice the current actuators for adjusting the disk valve openings at the plant are not suited for automatic control. Hence, the actuators and possibly the disc valves must be replaced, and the fan capacity increased if automatic control of gas-flow is to be implemented.

2.5 Model adaption and validation

This section discusses implementation issues and documents how the model was adapted to data by comparing simulations and measurements. Two models are considered. In section 2.5.1 a full model is considered, where "full" indicates that all the states are included in the simulations. In section 2.5.2 a reduced order model is considered which includes a subset of the states of the full model. In section 2.5.3 some preliminary simulations illustrate the use of gas velocity as a control input to the reduced model. In section 2.5.4 the model is compared to data which it has not been adapted to.

2.5.1 Full PDE model

The model from section 2.3.1 was implemented in `MATLAB` and solved with `ode15s`, see figure 2.13. `ode15s` is an implicit solver of varying order using adaptive step lengths. It uses finite differences in solving the nonlinear equations. The spatial derivatives were approximated to the first order with 10 spatial elements. The simulated model was tuned by manually

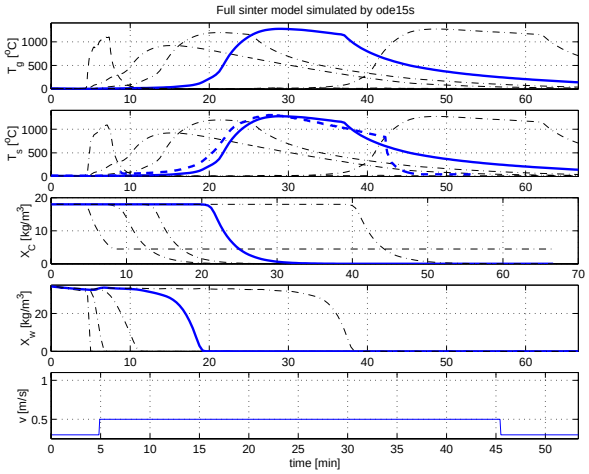

Figure 2.13: *Simulation - full model.* The figure shows simulations of the model with all states included. The upper part shows gas temperatures (dashed-dotted) at depths 4, 8, 16 (boldface solid) and 40cm, along with the ignition profile (dashed). The second part shows solid temperature profiles (dashed-dotted) at the same depths along with the measured profile (dashed boldface) at 15cm. The third part shows coke concentration profiles at the same depths, while the fourth part shows water concentration profiles at the same depths. The lower part shows the velocity (spatially uniform).

adapting the activation energy E in equation (2.15) and visually comparing the boldface curves in the second subfigure of figure 2.13. Observe the "kink" in the temperature profiles caused by switching the fusion model from "melting" to "freezing". This kink is not desired, and future refinements of the model should include a more realistic model of fusion. The kink in the measured profile is caused by manual removal of the thermo-element from the sintering bed, see figure 2.11 for further details. Note that the velocity profile is spatially uniform in the sense that the velocity at any time instant was as shown in figure 2.13 in all spatial elements. Figure 2.13 shows that the solid temperature profiles in the upper layers are narrow, while the

temperature profiles in the lower layers are wide. The figure also shows that the gas temperature profiles are almost equal to the solid temperature profiles. This does not comply with the measurements shown in figure 2.11, and further model refinements are needed. Towards the end of the batch, air passes through a bed almost completely converted to sinter with poor heat exchange properties, hence the air will not receive much heat from the sinter. The model does not reflect the different heat exchange capacities of wet charge and sintered material.

Since the model does not produce a physical gas temperature, the model must be refined if industrial implementation of model based control is to be considered. A physical gas temperature is essential for a control relevant model since the offgas temperature should be utilized for state estimation in an industrial implementation. If the gas temperature is considerably lower than the model temperature, other mechanisms may be important for transporting heat down through the bed, i.e. conduction by Fourier's law and radiation. The adaption of the activation energy, together with the discrepancy in gas temperatures shows that further refinements are needed in an industrial implementation.

The models by Hoislbauer and Jaquemar (1983) and Patisson et al. (1991) are both simulated with finite element methods. Using the implicit solver `ode15s` resulted in acceptable simulation times but gave inexact gas concentrations (a "ripple" is introduced by the plug-flow assumption), see figure 2.14. I.e. the accuracy of the gas profiles was improved by increased spatial resolution, but this did not influence markedly on neither the temperature profiles nor the solid coke and water profiles. Hence, since the inaccuracy in gas concentrations has little influence on the important process states, water was excluded from the model and gas was considered as dry air in order to reduce the computational complexity.

2.5.2 Reduced model

The state vector after eliminating water and the gas components becomes

$$x_{red} = [T_g, T_s, X_C]$$

The gas phase was lumped into dry air under the assumption that there is always excess air so that coke combustion is never limited by oxygen concentration. Water was excluded under the assumption that it does not have significant influence on the shape of the heat profile at elevated temperatures. The water does influence on the overall energy balance, and this influences on the temperature profiles in lower layers due to the distributed

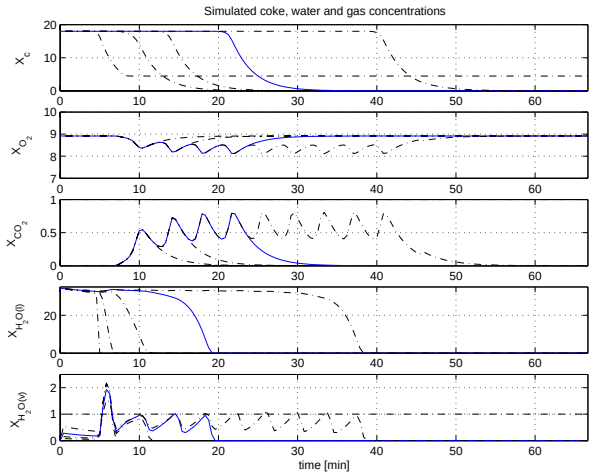

Figure 2.14: *Gas concentrations - full model.* The figure shows simulations of the model with all states included (the full model). Subfigure 1 and 3 shows coke and water (respectively) concentration profiles reprinted from figure 2.13. Subfigure 2, 3 and 5 shows the ripple in the simulated gas profiles. Solid coke and water have units kg/m^3 while gas components have units mol/m^3. The ripple in gas composition is caused by limited spatial resolution combined with the plug-flow assumption.

nature of the process. Still the reduced model can be tuned to fit the measurements. The activation energy of the reduced model was tuned to be $E_r = 74kJ/mol$, while the value for the full model is $E = 70.9kJ/mol$. The activation energy is increased slightly to make coke combustion of the reduced model harder to start. This is as expected since no energy is needed to evaporate water. The resulting reduced model after tuning was simulated with `ode15s` as shown in figure 2.15.

The computational time on a `COMPAQ Deskpro` with a Pentium II processor and 128Mb memory for simulating the full model as shown in figure 2.13 was approximately 5 minutes, while it was approximately 20 seconds for the model shown in figure 2.15. The condition number of the Jacobian of the reduced model at the initial point $x_{red} = [14^oC, 14^oC, 18kg/m^3]$ (with uniform spatial distribution) is in the order 10^{14}, i.e. the system is very stiff. Attempts to solve both the full and reduced model with explicit and modified Euler, Runge-Kutta 4 and implicit Euler were unsuccessful. Due to the ill-conditioning and the limited accuracy of implicit Euler, very short steplengths lead to excessive computational times. Simulating with the level 2 implicit Runge-Kutta method Lobatto IIIC allowed increasing the step

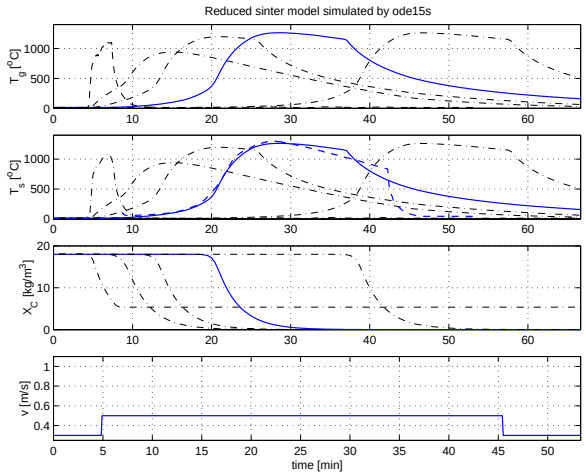

Figure 2.15: *Simulation - constant v.* The figure shows simulations with the reduced model. The upper part shows gas temperatures (dashed-dotted) at depths 4, 8, 16 (solid) and 40cm, along with the ignition profile (dashed). The second part shows solid temperature profiles at the same depths along with the measured profile (dashed) at 15cm. The third part shows coke concentration profiles at the same depths, while the lower part shows the velocity (spatially uniform).

length to $h = 1s$. The total simulation time with Lobatto IIIC was approximately 10 hours. Adaptive step lengths did not reduce the simulation time for Lobatto IIIC. These issues influence on the `MPC` implementation and are revisited in chapter 4.

2.5.3 Introductory control action simulations

The suggested control scheme from section 2.2.2 is investigated by simulating the reduced model with two different velocity inputs as shown in figure 2.15 and 2.16. As is seen by comparison of figure 2.15 and 2.16, the temperature profiles are influenced by the velocity profile. The modified velocity profile gives a more evenly distributed temperature profile at different levels, but the sintering time is prolonged. I.e. a control objective should balance sintering time against a uniformly shaped temperature profile.

The full order model exhibited similar responses as the reduced model when exposed to the modified velocity profile. The spatial temperature profiles for the full model at selected time instances are shown in figure 2.17. Observe again the more even profile and prolonged sintering time in the lower part. Hence, maximizing ϕ alone from equation (2.5) may

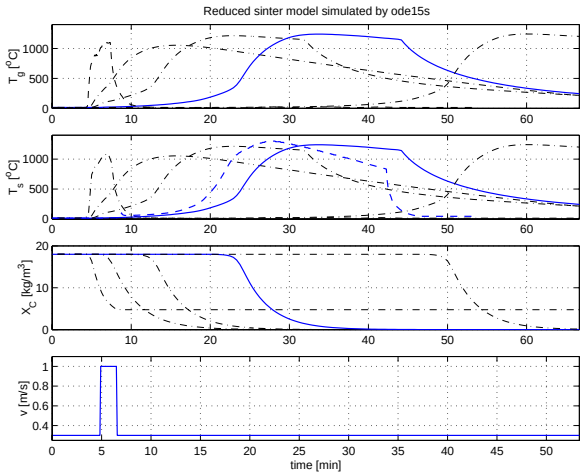

Figure 2.16: *Simulation - profiled v.* The figure shows simulations with the reduced model. The depths and profiles are as in figure 2.15. Observe the altered velocity profile, and the influence this has on the temperature profiles as compared to figure 2.15.

lead to prolonged sintering times, and the control problem (2.6) should be augmented to penalize the batch duration time t_{end}. The present model is implemented with a fixed batch time t_{end}, which may lead to impaired solutions. If this turns out to be a problem, the objective can be augmented as discussed above as a remedy.

2.5.4 Model validation

The reduced model is compared to the measurements of the first campaign. In figure 2.18 the dashed pressure measurements from figure 2.12 are repeated (dashed). The model is seen to fit well to the new data. Observe that the measured pressure profile is used in the model to generate a spatial pressure profile as discussed in section 2.3.4. This pressure profile then allows direct calculation of the gas density $\rho_g(T_g, p)$ by the ideal gas law. The velocity is scaled to reflect the level difference in the figure, since the pressure and velocity are not linked in the present model. Note that the scaling discussed in the figure caption can be done by using Ergun's relation, equation (2.22). I.e. solving Ergun's relation for $p_1 = 5kPa$ and $p_2 = 6kPa$ for a given set of parameters[4] gives approximately the same ratio 6/5.

[4] $\mu = 1.7 \cdot 10^{-5} kg/ms$, $\rho_g = 1.29 kg/m^3$, $d_p = 2mm$, $\varepsilon = 0.4$ and $\phi_f = 0.43$.

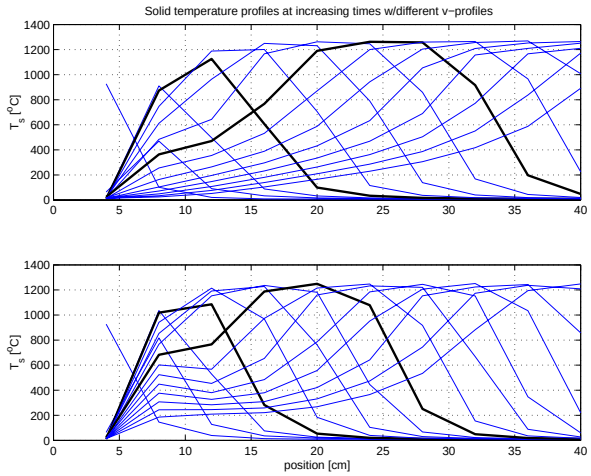

Figure 2.17: *Simulation - T_s-profiles.* The figure shows simulations with the full model. The abscissa is position inside the bed with 0cm being the top and 40cm being the bottom. The figure shows solid temperature profiles at time instances 5.8, 10, 14.2, 18.3 (boldface), 22.5, 26.7, 30.8, 35 (boldface), 39.2, 43.3, 47.5, 51.7, 55.8, 60 and 64.2 minutes. The upper part shows profiles obtained with the nominal velocity profile from figure 2.15, while the lower part shows profiles obtained with the velocity profile shown in figure 2.16.

The simplified fusion model does not allow the simulated temperature to increase significantly above 1250^oC. Validating against measurements for other experiments is hard since the temperature profile is poorly correlated with the coke input, see figure 2.23 in section 2.6 for details.

2.6 Data analysis

The data from the experiments documented in section 2.4 are analyzed in this section. The aim of this section is a verification of the control strategy suggested in section 2.2.2. In particular, the validity of the suggested control objective (2.6) is assessed. By conventional experiment planning methodology (Dougherty 1990), section 14, the input (feed composition) was perturbed in two dimensions (water and coke) and the output responses (quality and production rate) were measured. Since the perturbations were not conducted directly in the sintering pan, the hold-up of the recycle and charge bins forced a degree of conservatism in the experiment design, cf. figure 2.9.

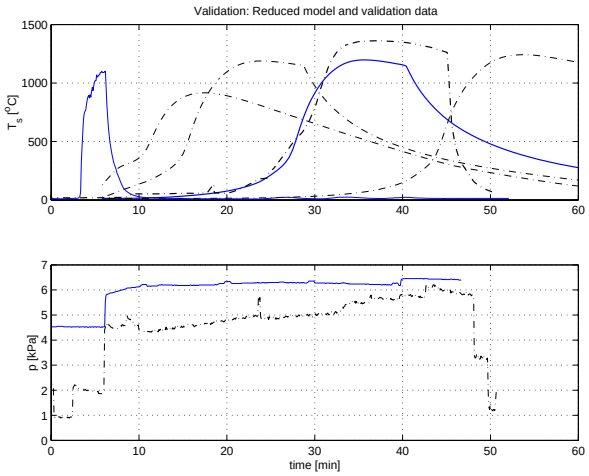

Figure 2.18: *Validation of model.* The figure shows simulations with the reduced model and validation data. The upper part shows the ignition profile from figure 2.11, batch 3, and simulations of reduced model with this ignition profile. The measured profile (dashed-dotted boldface) is the one shown in figure 2.12. The lower curve shows the measured pressure drop for experiment 1, batch 3, (solid) which was used to tune the model along with the one from figure 2.12 (dashed). The approximate ratio between these is 6/5, hence, the simulated profile in the upper part was generated with the velocity profile from figure 2.15 scaled by 6/5, i.e. 0.42m/s. The solid simulated profile is plotted for the level corresponding to 20cm.

The coke content in the pan is not directly measurable. In particular, assuming a constant recycle will give a constant bias of coke content, while a varying recycle will give a varying bias. Figure 2.19 shows the set points and calculated real values of each experiment labeled 1 through 13. The figure clearly shows that the locations of the real operating points marked by ($\square$) are not well correlated to the set-points labeled by (o), and identification of an input-output model based on the experiments was abandoned due to this. The experiments were designed to have $\pm 2\%$ deviation from normal operating values in coke and water concentrations. The water content in the pan was measured by drying a small sample of charge, and measuring the weight loss caused by evaporation. The coke content in the pan was calculated by calculating the recycle, see details in section 2.6.1.

The production rate is calculated by a mass balance in section 2.6.1, and the qualities reducibility and mechanical strength are analyzed in section 2.6.2. The objective function suggested in equation (2.6) is assessed in

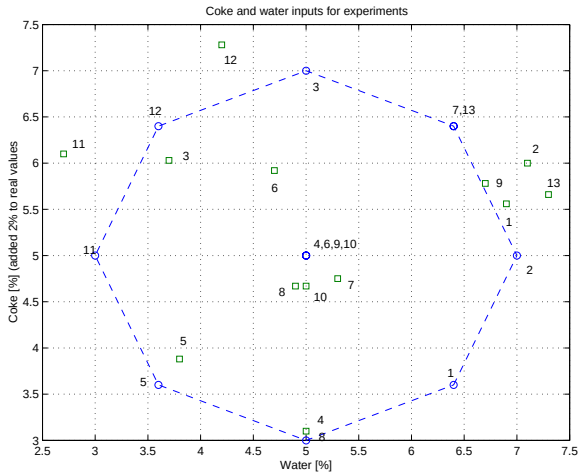

Figure 2.19: *Set-points for experiments.* The figure shows the set-points (o) and real values (□) for coke and water concentrations for experiment 1-13. Note that a 2% bias is added to the real coke values. The real coke values were calculated by compensating for the recycle as described in the text, while the real water values are measured. The dashed circle only serves to emphasize the positions of the set-points at a circle with radius 2 and centered at the normal operating point.

section 2.6.3.

2.6.1 Productivity

A realistic measure of production rate should weight the recycle and mass gas-flow. The recycle is calculated from a stationary mass balance given by the difference between the measured output from the feed bins 1-12 in figure 2.7 and the calculated maximum production from the six Greenawalt-pans. An uncertainty is introduced by how the total mass of fresh feed (i.e. from the feed bins) is calculated from the measurements: The output from the feed bins is logged, but, due to the plant automation system, it is uncertain which parts of the logged time-series to include in the calculation. I.e. feed is added from the feed bins at irregular intervals based on level measurements and various time-stamps in the automation system. Close investigation of the batch-cycles was done to determine which parts to include in the calculation of the total mass of fresh feed. Then, the fresh feed was calculated by integrating the time-series from each feed bin. In addition to this, the calibration of the weights in the sintering plant was uncertain. Hence, the calculated real values for the set-points given in figure 2.19 are uncertain.

The total mass of the six pans is calculated by calculating ρ_{ch} from the measured output from the feed-bins and multiplying by the volume:

$$m_{ch} = 6(1-\varepsilon) \cdot \rho_{ch} \cdot L \cdot A$$

where $L = 0.4$ $[m]$ is the height and $A = 3^2$ $[m^2]$ is the cross-sectional area of each pan. The void fraction ε is unknown, but can be measured if the density of solids is known. The bulk solid density ρ_{ch} is computed from the logged feed-ratio and the measured moisture content. Since the coke and water fractions are small and do not vary much, assuming $\rho_{ch} \approx 3600$ $[kg/m^3]$ will not introduce large errors. The value $\varepsilon = 0.3$ was computed by assuming a 100% production rate for the "best" experiment (no. 13), and computing the equivalent void fraction as $\varepsilon = 1 - m_{meas}/m_{max}$ where m_{meas} is measured feed, and m_{max} is the value achieved for m_{ch} with $\varepsilon = 0$. Using $\varepsilon = 0.3$ in calculating the production rates gave the results shown in table 2.3. As can be seen from the table, the recycle is larger than expected and the variations are profound. Keeping in mind the large variance of consecutive batches from figure 2.10, the calculated productivity indicate that it is unlikely that the coke concentration in the pan will follow the set-point due to the large variations in the recycle.

2.6.2 Quality tests

The quality in terms of mechanical strength and reducibility was measured off-line by taking product samples from each of the thirteen experiments described in section 2.4.2. The mechanical strength was measured by the standard test equipment used at the plant. 12.50kg of sinter of fraction $d_p \in [10, 16]$mm was tumbled according to a preset program. The weight fractions ¿10mm, 1-10mm and ¡1mm after tumbling were measured. The results are given in table 2.4. The reducibility of the sinter samples were measured by reduction in a CO-atmosphere in the laboratory scale retort shown in figure 2.20. Sinter lumps were loaded into the retort, and the retort was hung from a scale and placed in an electric heater. The retort was purged with the inert gas N_2 and preheated to 1000^oC. Then N_2 was replaced by CO and reactions similar to $Mn_{x_1}O_{y_1} + CO \rightarrow Mn_{x_2}O_{y_2} + CO_2$ gives a weight loss since oxygen from the charge is transported out of the retort. The weightloss for all 13 experiments were calculated in grams weight loss between 10% and 90% reduction from start to total weight loss. The reducibility experiments were conducted for both 10-16mm and 5-10mm fractions. The two different

Experiment	Production rate (calculated)	
	Prod.rate [%]	Coke [%]
1	71.1	3.56
2	66.7	4.00
3	54.1	4.03
4	46.8	1.10
5	62.6	1.88
6	90.9	3.92
7	82.2	2.75
8	66.4	2.67
9	83.7	3.78
10	52.0	2.67
11	54.1	4.10
12	63.1	5.28
13	100.0	3.66

Table 2.3: *Production rates.* The table gives the calculated production rates for the experiments. The table shows calculated production rates for experiments 1-13. The production rate varies between a maximum of 100% at experiment 13 and a minimum of 46.8% at experiment 4. The recycle, which is obtained by subtracting the production rate from 100 varies between 53.2% and 0%. Note that 100% production rate at experiment 13 was achieved under the hypothesis of a void fraction $\varepsilon = 0.3$. This void fraction is a lower bound on the void fraction for experiment 13. The actual void fractions were not measured. The coke concentration numbers were computed by a mass balance adding the measured feed and calculated recycle and assuming no coke in the recycle.

fractions gave almost equal reducibility measures. The reducibilities are summarized in table 2.5.

By assuming a linear dependence between the function τ_z from equation (2.4) and strength and reducibility, the `MATLAB NAG` toolbox function `g02caf` was used for linear regression. The heuristic rule that a certain minimum strength is required while simultaneously maintaining a high reducibility was implemented by weighting the two linear regression functions as shown in figure 2.21.

The computed quality model is conceptual since no data exists that allow verification, and there are no quantitative quality requirements given at the plant. Note that the choice of the weighting functions controls the shape of the quality model, hence different quality models can be formed by altering the weights. The quality model is founded on the arguable assumption that a linear regression is adequate. However, including heuristics by weighting

Experiment	Fractions [kg]		
	+10mm	1-10mm	-1mm
1	6.42	5.27	0.72
2	7.45	4.28	0.65
3	6.63	5.17	0.61
4	7.55	4.24	0.59
5	6.96	4.73	0.65
6	7.00	4.76	0.65
7	7.95	3.87	0.56
8	6.34	5.25	0.69
9	8.08	3.73	0.67
10	7.91	3.94	0.57
11	7.82	4.00	0.64
12	7.23	4.57	0.63
13	6.67	4.61	0.59

Table 2.4: *Mechanical strength of sinter.* The table gives the mechanical strength of produced sinter for experiments 1-13. The mechanical strength quality is defined as the normalized +10mm fraction. The numbers were measured by a tumbler test using the regular equipment at the sintering plant.

functions can still be applied even if the underlying functions are changed. The standard errors of the adapted parameters in the linear regressions allows assessment of the standard error of the quality model as is shown in figure 2.22.

The second column in table 2.6 gives the normalized values of τ_z for all experiments. This figure gives a direct measure of the conditions inside the bed during sintering. As discussed in section 2.2 the quality of the produced sinter is governed by the temperature profile in the bed. The third column gives the values of the objective ψ_z[5] from equation (2.6) as calculated from the data. The fourth and fifth column show the calculated integral of the sigmoid $\int \sigma_z dt$ from equation (2.5) and $T_{z,\max}$ from the same data.

The identified quality model do show that the results from the experiments conform to the heuristics reported in the literature as discussed in section 2.2.1. The relation between the inputs coke and water and the outputs production rate, strength and reducibility are shown in figure 2.23.

The linear regression models shows that τ_z increases (slightly) as expected with the coke content. The inverse proportionality towards water

[5]The level z being the position of the S-element, i.e. 15cm down into the bed.

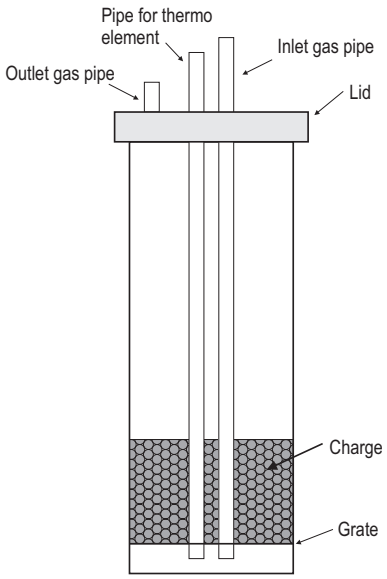

Figure 2.20: *Retort used for reduction experiment.* The figure shows the laboratory scale equipment used for reduction experiments. 350g of sinter lumps are loaded into the retort, and the retort with sinter is hung from a scale and pre-heated to 1000^oC. During preheating the retort is purged with N_2. When the temperature reaches 1000^oC N_2 is replaced by a CO gasflow. In the reducing CO atmosphere oxygen is removed from the sinter and transported out with the gas giving a weight loss which is logged on the scale. The internal diameter of the retort is 5.3cm, and the outer diameter is 5.7cm. The grate is placed 1cm above the bottom, and the height from the bottom to the lid is 33cm. About 5cm charge is used. The retort is made of an $FeCrAl$-alloy and it is magnetic up to 950^oC. At 1000^oC the magnetic influence is low, but appears as noise on the logged measurements.

content is attributed to the competing properties of water: An optimal water content W^* yields a higher permeability allowing the heat wave to travel faster through the bed, giving shorter sintering batch times. Increasing the water content beyond W^* does not increase permeability, and additional energy is needed for evaporation, resulting in poorer sintering conditions. Hence a linear regression model is not well suited for analyzing water content with any output.

The production rate vs. coke content varies as expected, while strength and reducibility are almost flat. This is why the "intermediate" quality model based on τ_z shown in figure 2.21 was suggested. "Intermediate" hints to the logical "chain" that a higher coke content implies a higher integral τ_z yielding increased strength and decreased reducibility. The 90% confidence intervals are wide for all regression models, indicating that input-output identification cannot be expected to give good results. Nonlinear identification is not considered due to the known uncertainties in the experiments.

Experiment	Reducibility (+10mm fraction)		
	Δm [g]	time [min]	$\Delta \dot{m}$ [g/min]
1	22.27	119.5	0.41
2	20.05	147.5	0.43
3	19.91	143.5	0.34
4	20.06	95.5	0.36
5	21.98	150.5	0.54
6	21.56	119.0	0.41
7	18.96	98.5	0.40
8	22.54	114.5	0.43
9	23.00	95.0	0.45
10	19.71	99.5	0.43
11	21.74	122.0	0.46
12	19.98	88.0	0.49
13	21.52	125.0	0.44

Table 2.5: *Reducibility of produced sinter.* The table gives the measured reducibility of produced sinter for experiments 1-13. The quality reducibility is defined as the number Δm in the second column. The reducibility experiments were conducted on laboratory equipment located at the metallurgical department at NTNU in Trondheim, Norway.

2.6.3 Assessment of the objective function

From the discussion above, it is apparent that the validity of the objective function ϕ suggested in equation (2.5) can be evaluated by comparing it to τ_z. The large variations in integrated sintering temperatures shown in the first column of table 2.6 suggest that a more detailed metallurgical analysis of the produced sinter can be used to verify the data on which the quality model from figure 2.21 is based on. Figure 2.25, 2.26 and 2.27 show microprobe images of sinter samples taken from experiment 1, 7 and 11 respectively. From the first column of table 2.6 these experiments correspond to the second smallest, maximum and (approximate) average of integrated sintering temperature. Experiment 1 was chosen in favor of experiment 5 (the minimum) since experiment 1 has been shown with in greater detail in the preceding sections. The pictures were taken by a scanning electron microscope (`SEM`) (`EPMA` - Electron Probe Micro Analyzer) with 4 wavelength-dispersive spectrometers and an energy-dispersive spectrometer, capable of taking both area and point measurements. The figures show (row-wise) a grey scale overview, Al, Mn, Ca, O, Si, Fe and K plots for the same area of

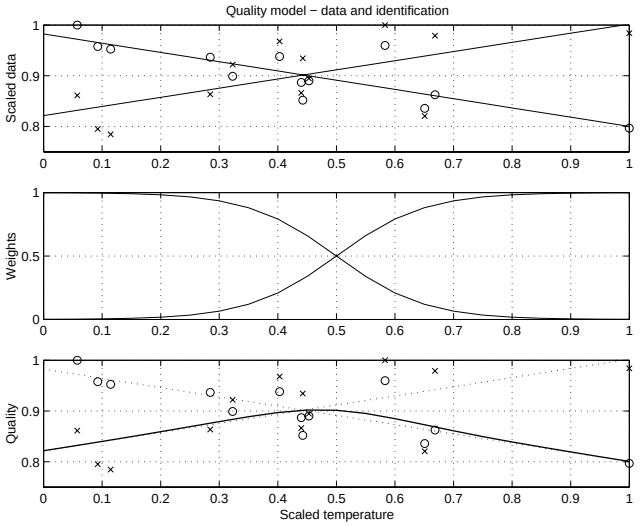

Figure 2.21: *Quality model.* The figure shows the normalized reducibility ('o') and strength ('x') data along with the linear regression models (upper part). The weighting functions (middle part) were chosen somewhat arbitrarily, but according to the hypothesis in section 2.2.1. I.e. to balance the competing objectives that the temperature should be high to give good strength but not too high to give good reducibility. The quality model (lower part) is formed as the convex combination of the two linear regression models using the weighting functions. The dotted lines in the lower subfigure are reprints of the linear regression models from the upper subfigure.

each of the three samples. The color coding along with the bars to the right gives the relative content of each component. Note the coarser resolution used for experiment 1.

The grey scale plots show that for experiment 1 the light gray phase (containing mainly Mn- and Fe-oxides) has sharp corners indicating that the amount of melting is minor. Similarly for experiment 7 the corners of the light gray phase are rounded and neighboring crystals grow together and extensive melting and recrystallization are prevailing. The average case in experiment 11 shows an intermediate stage where the smaller crystals in the light gray phase have rounded corners, while the larger crystals have sharp corners. Figures 2.25-2.27 support the calculated integral temperatures shown in table 2.5 and it can be concluded that the linear regression models in figure 2.21 are based on reasonable data. Observe that the results are based on point samples in a distributed media, and local effects such as

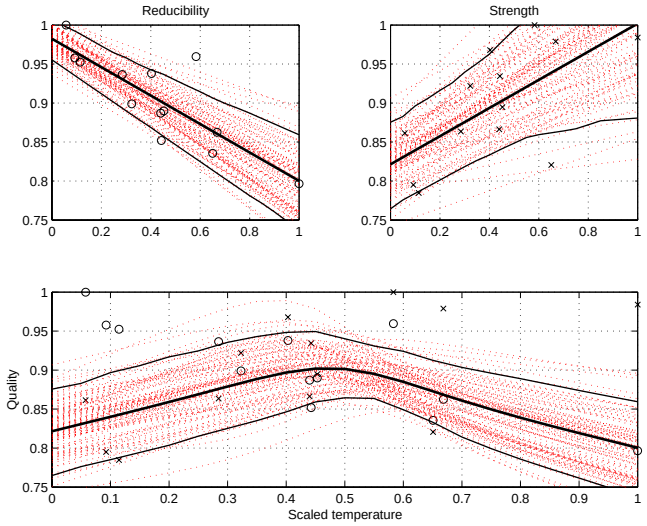

Figure 2.22: *Standard deviation of quality model.* The figure shows the computed standard deviation of the quality model. The standard errors of the parameters of the linear regression models were used to generate 100 new models (upper part) from a normal distribution (Monte-Carlo simulation). 100 new quality models were formed by the same procedure as shown in figure 2.21. The 95% standard deviations were found by counting the points between 5 and 95 as shown by the solid lines. The bold face solid curves are reprints from figure 2.21.

channeling, where/when the sinter samples are taken etc. must be kept in mind. Despite these limitations, the results conform well to the hypothesis in section 2.2.1. I.e. the competing objectives that the temperature should be high to give good strength but not too high to give good reducibility are reproduced. Note that figure 2.25 and 2.26 corresponds to experiments 1 and 7 with coke content as shown in table 2.3. Along with the first subfigure in figure 2.23, this indicates that the calculated real setpoints as shown in figure 2.19 are wrong. I.e., since the locations of the calculated real set points are questionable, the hypothesis that increased coke content increases the sintering temperature cannot be assessed by the data.

From figures 2.25-2.27 and table 2.6 it is observed that τ_z is well suited as an objective. Since τ_z is non-smooth, the objective function ψ_z from equation (2.6) was suggested. In figure 2.24 the correlation between ψ_z and τ_z is shown. The objective function ψ_z is seen to fit well to the integral. Observe from the lower part of figure 2.24 that the use $T_{\max}$ gives an inappropriate objective function since it is not well correlated to τ_z. Note that

Experiment	τ_z [-]	$\psi_{z=15cm}(T_s)/1000$ [-]	$\int \sigma_{z=15cm}(T_s)$ [-]	T_{max} [^oC]
1	0.09	4.86	81.7	1300
2	0.32	17.21	121.0	1430
3	0.65	34.74	269.5	1410
4	0.44	23.61	139.3	1450
5	0.06	3.05	38.3	1330
6	0.44	23.46	154.0	1450
7	1.00	53.48	312.8	1450
8	0.11	6.04	100.3	1290
9	0.58	31.16	263.3	1340
10	0.67	35.74	245.8	1430
11	0.40	21.50	168.4	1420
12	0.45	21.50	168.4	1430
13	0.28	15.15	183.7	1390

Table 2.6: *Assessment of the objective function.* The table assesses the validity of the objective function for experiments 1-13. The first column shows the normalized integral of the measured temperature $T_s > T_{fu}$ for all experiments. The second column shows the value of the objective function $\phi(T_s)$ from equation (2.6) integrated over the time series for each experiment. The third column shows similar data for the integral of the sigmoid function $\sigma(T_s)$ from equation (2.5). The last column shows the achieved maximum temperature for each experiment.

$T_{\max}$ was scaled to the interval $[0, 1]$ by choosing its lower value as the zero point and its upper value as one and scaling the intermediate values accordingly. Note also that using $\int \sigma(T_s)dt$ alone as an objective is not as good as using ψ_z, since σ doesn't pay extra credit for solid temperature profiles with temperatures above, say 1250^oC, cf. figure 2.3.

It is concluded that ψ_z is well correlated to τ_z, and then the overall objective ϕ from equation (2.6) is a valid objective since this is a weighted integral of ψ_z in the spatial dimension.

The parameter q from equation (2.6) can be selected by inspecting the second column of table 2.6. The mean value $q = \bar{\phi} = 22.42 \cdot 10^3$ may serve as an initial choice[6].

[6]The actual value of q used in the implementation must be compensated for varying sampling period. The values in table 2.6 were obtained by direct summation over the time-series of the output from the `MATLAB` function `logsig`(T_s). Hence, the sampling periods of 5 seconds is inherent in table 2.6.

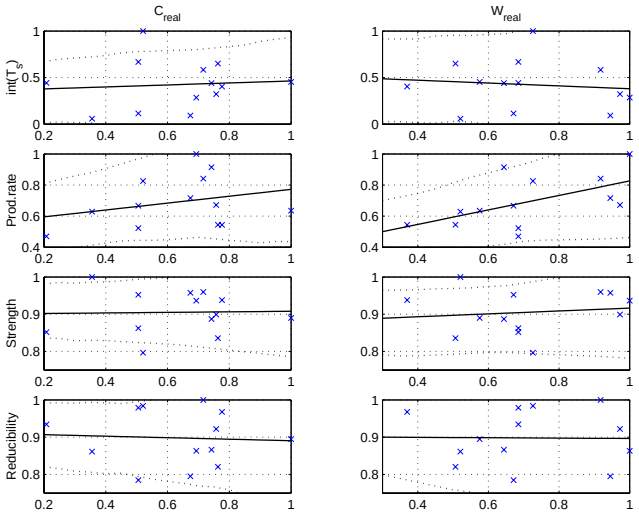

Figure 2.23: *In- and output data.* The figure shows the datapoints ('x') along with linear regression models (solid lines) of various output variables plotted against the normalized calculated coke content (first column) and normalized measured water content (second column). The first row shows the normalized integral $\int(T_s > T_{fu})$ for all experiments. The second row shows the normalized calculated production rate, the third row shows the normalized measured quality strength and the last row shows the normalized measured quality reducibility. The standard errors of the parameters of the linear regression models were used to generate 100 new models from a normal distribution (Monte-Carlo simulation). The 90% standard deviations (dotted lines) were found by counting the points between 10 and 90 (similar to figure 2.22).

2.7 Conclusions

A dynamic model of grate sintering is gathered from the literature, adapted to data, and simulated in `MATLAB`. A reduced order model including only the solid and gas temperatures and coke concentration was suggested to reduce the computational complexity. Extensive experiments were conducted to validate the model and the hypothesis on which the control strategy is founded. A control strategy was suggested, and simulations support the hypothesis discussed in sections 2.2.1 and 2.2.2.

The model is implemented using `MATLAB`'s `ode15s` giving reasonable computational times, but inaccurate gas dynamics. The inaccurate gas dynamics has little influence on the temperature profiles. Since the gas dynamics has little influence on the temperature profiles, the gas states along with the

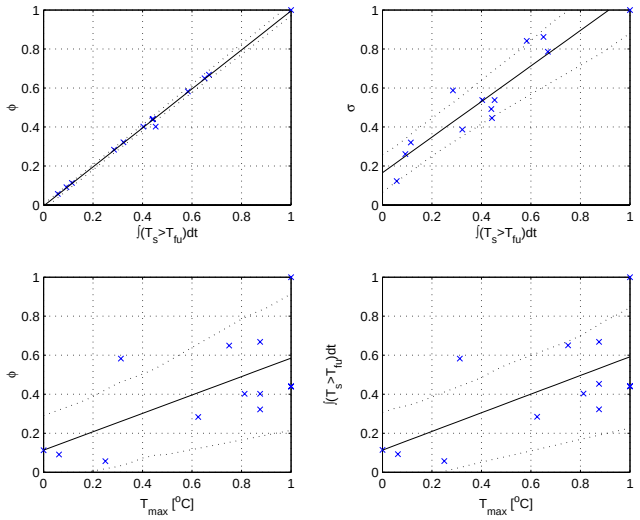

Figure 2.24: *Assessment of the objective function.* The figure shows the datapoints ('x') along with linear regression models (solid lines) of the normalized objective $\phi(T_s)$ from equation (2.6) plotted against the normalized temperature integral (lower left) and measured maximum temperature (lower left). The normalized sigmoid function $\sigma(T_s)$ from equation (2.5) is plotted against the normalized temperature integral (upper right). The normalized temperature integral is also plotted against the measured maximum temperature (lower right). The standard errors of the parameters of the linear regression models were used to generate 100 new models from a normal distribution (Monte-Carlo simulation). The 95% standard deviations (dotted lines) were found by counting the points between 5 and 95 (similar to figure 2.22).

water is excluded from the model in the reduced model. The reduced order model is considerably faster to simulate; the simulation time drop from 5 minutes to approximately 20 seconds per batch. The reduced model maintains approximately the same level of accuracy as the full model. The gas temperature profiles do not comply with the measurements, and the activation energy of coke combustion has been extensively tuned to adapt the model to the data. The solid temperature profiles are satisfactory reproduced by the model. The model is good in the sense that it is intended as a control relevant model, and it is essentially the solid temperature profiles that govern the process outputs quality and production rate.

The gas velocity is suggested as a control input, and its viability has been documented by the experiments. However, an industrial implementa-

tion using pressure as control input must include the velocity as a state. In addition to correcting the erroneous gas temperature and activation energy, including velocity as a state and improving the fusion model are essential for an industrial implementation. This is left as future work. Despite these shortcomings, the principal properties are maintained and model based control with the present model is sound. An improved model is expected to fit into the same control framework as the present model.

The main contribution of this chapter is the reduced order model which along with the implementation gives short simulation times. The model is adapted to industrial data, and a control objective formally stated in equation (2.6) is proposed. Industrial experiments explore and support the viability of the proposed objective.

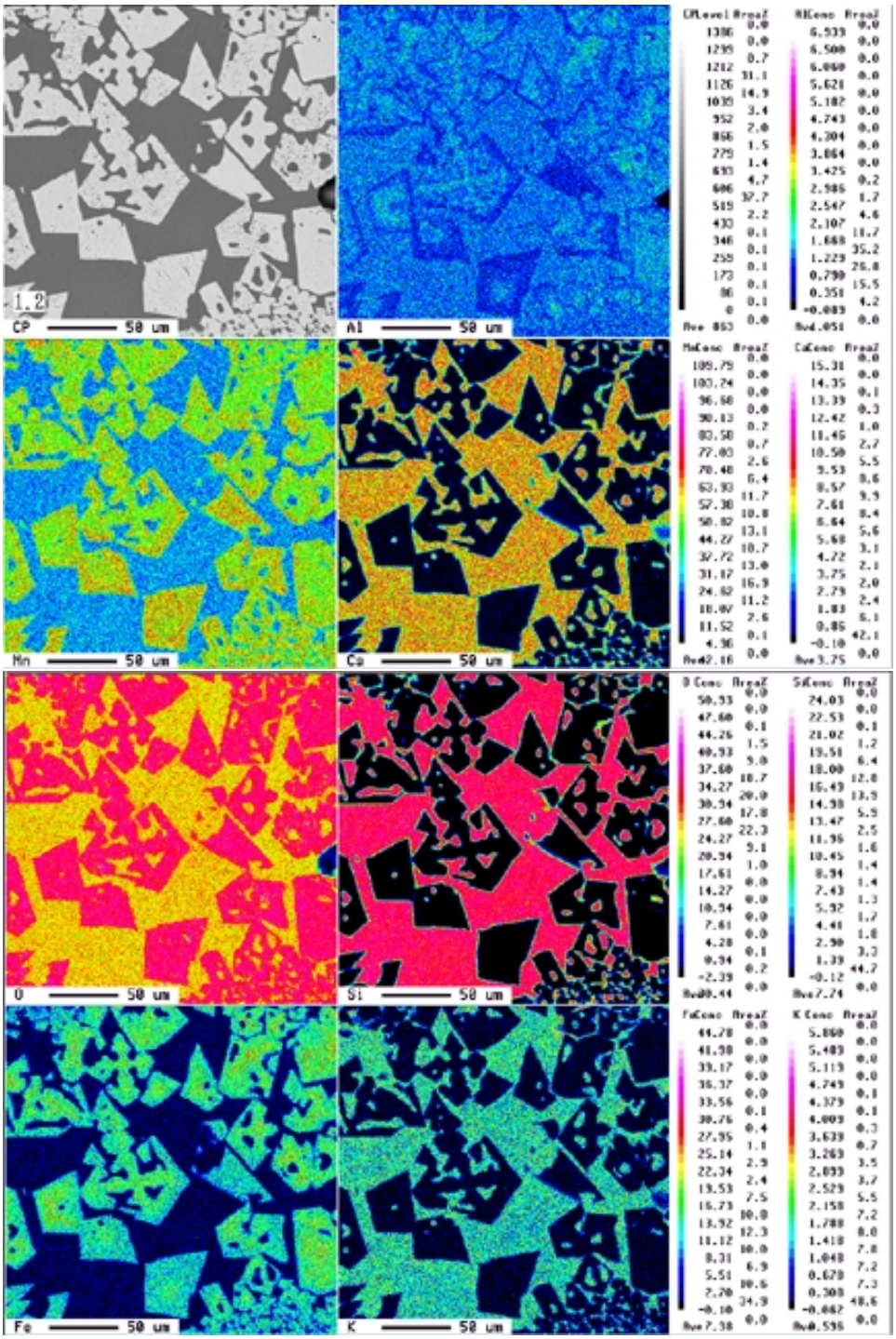

Figure 2.25: *SEM exp. 1.* The figure shows SEM (scanning electron microscope) plots for a sample taken from experiment 1.

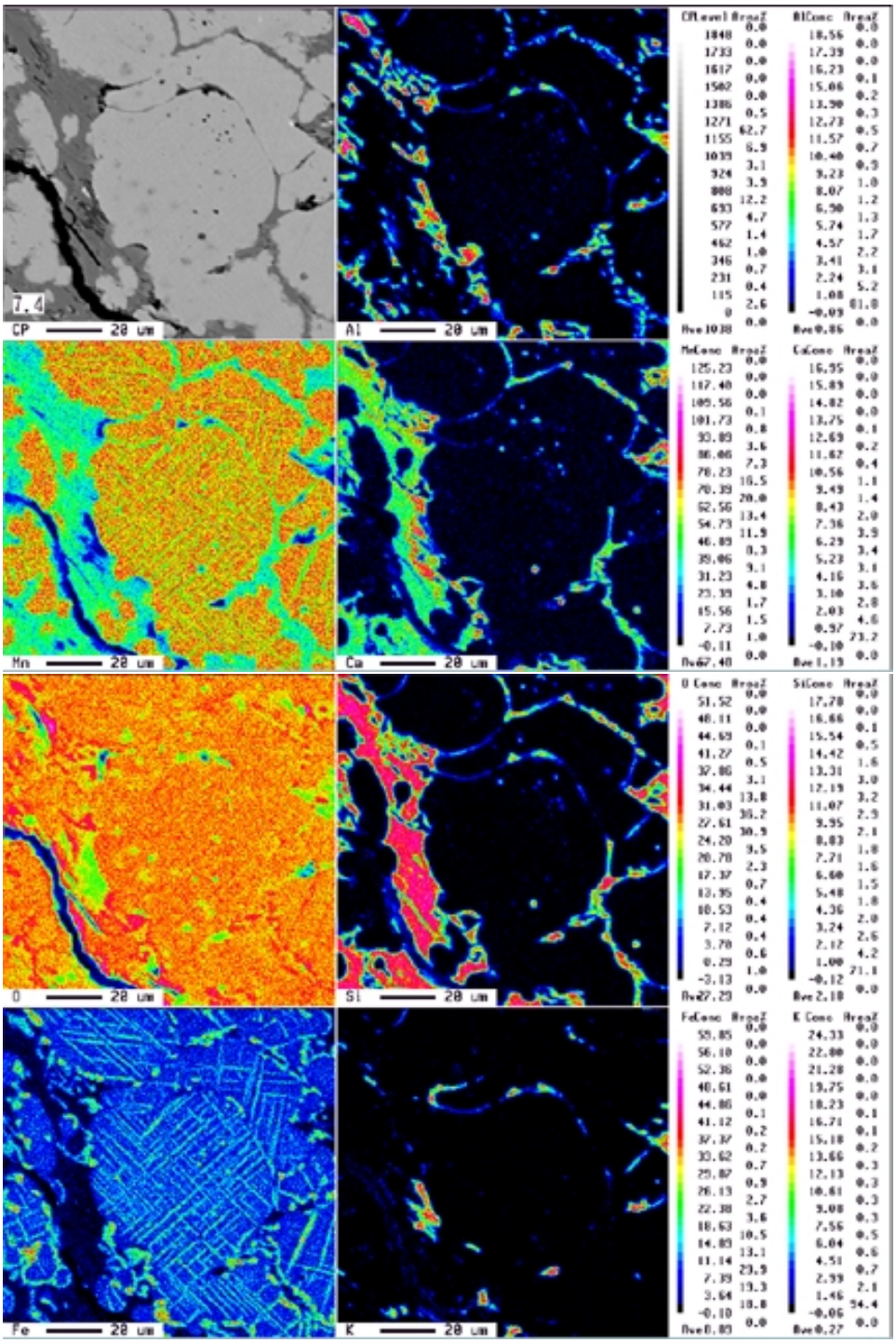

Figure 2.26: *SEM exp. 7.* The figure shows SEM plots for a sample taken from experiment 7.

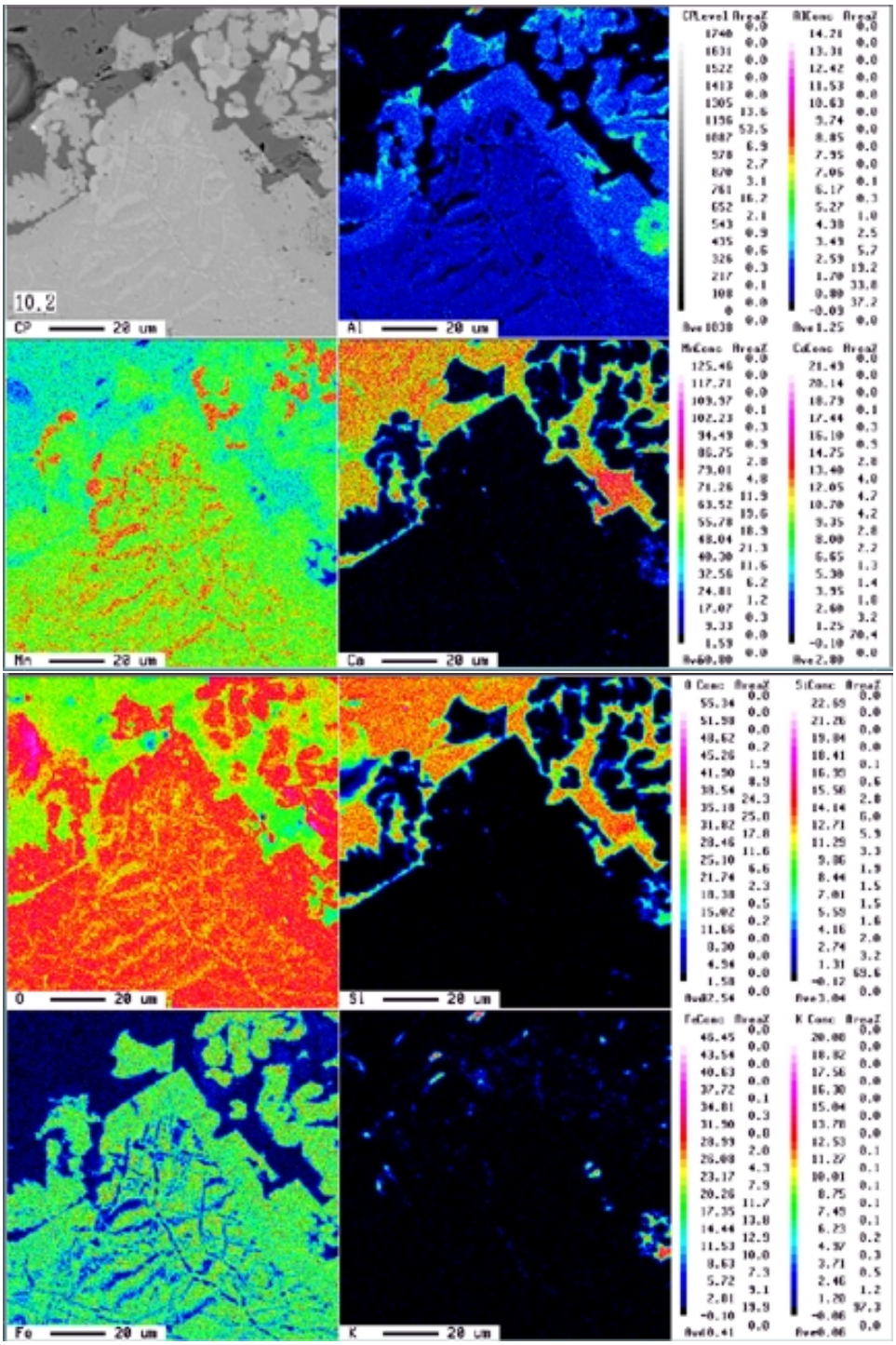

Figure 2.27: *SEM exp. 10.* The figure shows SEM plots for a sample taken from experiment 10.

Chapter 3

rFSQP - a feasible SQP method

This section considers nonlinear programming. By that we understand that we are seeking local solutions to smooth nonlinearly constrained optimization problems. We do not consider global, stochastic, or any form of discontinuous mathematical programming. Neither do we consider linear, quadratic or convex programming although quadratic programming is an important part in solving the subproblems occurring in nonlinear programming. Our main focus is on maintaining feasibility with respect to inequalities in reduced Hessian `SQP` methods. We start in section 3.1 with some general background relevant for comparison of various implementations of nonlinear programming codes. Then we continue with an introduction in section 3.2 pertaining to specific issues relevant to the suggested algorithm. In section 3.3 we suggest the algorithm `rFSQP` which is a feasible reduced Hessian `SQP` method. Then the convergence properties of `rFSQP` is analyzed in section 3.4, and we present some simulation results in section 3.5. We summarize this chapter in section 3.6.

Sections 3.2 through 3.4 and parts of section 3.5 have been submitted for possible publication in `SIAM` Journal on Optimization (Martinsen and Foss 2001).

3.1 Background

Our interest in nonlinear programming is due to our interest in nonlinear `MPC` which rely heavily on real-time optimization. Nominal stability results for nonlinear `MPC` relies on nominal feasibility of the model constraint and

feasibility of some other stabilizing inequality constraints at each iteration of the nonlinear MPC algorithm. Few results are available on robust stability of nonlinear MPC, but such results should not rely on nominal feasibility of the model constraint. The model constraint is an equality constraint, and currently the only efficient method available for maintaining feasibility with respect to equality constraints is by evolution. This is also known as the sequential method, single shooting or the Newton method. This method perform a simulation over the MPC horizon to find a solution to the nonlinear equalities, see chapter 4 for further discussions and references. See Ascher, Mattheij, and Russell (1995), sections 4.1 and 4.6.2, for an exposition of single shooting.

Simulating over the horizon to solve a nonlinear equality constraint like $f(x, u)$ where x is dependent on u results in an optimization problem in the reduced space of u which typically is known as the control input. However, if there are nonlinear inequality constraints on the original nonlinear MPC problem, it is not trivial to resolve these in the reduced problem since they must be transformed through a nonlinear map from the space of (x, u) into the reduced space of u. This chapter focus on optimization issues alone, and for now we leave the application issues until chapter 4.

This section continues by outlining some basic optimization principles needed to facilitate a comparison between various approaches to nonlinear programming. Then other work on reduced Hessian and feasible methods is summarized. We conclude this section with a summary of the most popular nonlinear programming codes with particular references to where numerical results can be found.

3.1.1 Background on optimization

This section contains brief descriptions of optimization methods used for nonlinear programming. Since this is mainly text-book material, we do not go into detail, but provide a few relevant references. The text-books by Gill, Murray, and Wright (1981) and Nocedal and Wright (1999) have been the main sources of information, but the books by Fletcher (1987) and Dennis and Schnabel (1996) have also been used to some extent.

General approaches

Note that the items below do not provide a mutually exclusive list. As an example an SQP method may be implemented with a reduced gradient method for step-computation. Likewise, the augmented Lagrangian method

is more a question of choice of merit function, which is listed below in the section on low-level implementation issues. The list is therefore intended as a compilation of "buzz-words" with a few comments and references to further reading.

`Reduced gradient methods` There are a number of approaches that result in variations of reduced gradient methods. These methods separate the variables into dependent and independent variables. The dependent are resolved from the equalities, while the independent ones are optimized subject to remaining constraints. The original reduced gradient method is `GRG` which seeks to remain feasible with respect to equalities, but also reduced Hessian `SQP` methods are reduced gradient methods. Reduced Hessian methods does not remain feasible with respect to equalities. Also interior methods can use decomposition strategies for step-computation. Even the sequential method frequently used in `MPC` results in a reduced gradient method. See Gill et al. (1981) for an introduction to reduced gradient methods.

`Augmented Lagrangian methods` replaced the earlier quadratic penalty methods by introducing explicit multiplier estimates in the merit function, thereby reducing ill-conditioning. The multipliers of the subproblem must converge to the optimal multipliers of the nonlinear programming problem for the algorithm to converge. In particular, it cannot converge faster than the multipliers, i.e. a first order estimate of the multipliers is not sufficient for superlinear convergence which is needed in practical algorithms. Elaborate multiplier schemes are therefore needed, and with such schemes augmented Lagrangian methods show excellent local convergence properties (noting that the merit function is smooth and that the full Newton step will be accepted locally). Global convergence is somewhat impaired by the strong dependence on the multiplier estimates which may be poor far from the solution. See Gill et al. (1981), pp. 225-233, Nocedal and Wright (1999), pp. 513-523, for introductions and Bertsekas (1982) for a deeper exposition of augmented Lagrangian methods.

`Projected Lagrangian methods` uses a model of the nonlinear program for step computation. Two variants `SLC` and `SQP`, exist of projected Lagrangian methods. Sequential linearly constrained (`SLC`) programming uses a nonlinear objective with linearized constraints. Sequential quadratic programming (`SQP`) uses a quadratic model of the Lagrangian with linearized constraints. The workload of each subproblem

is therefore lower in SQP, but at the expense of the need for a merit function to measure overall progress. Fewer function evaluations are typically observed in SQP than in SLC, since the subproblems in SLC must be solved to great accuracy. SQP methods may be implemented with incomplete solutions of the subproblems, see Murray and Prieto (1995).

Interior point methods for nonlinear programming typically use slacks for inequalities, and then put the slacks in a logarithmic barrier term. Then the slacks are kept positive, but since the inequalities are solved as equalities (with slacks), there is no feasibility guarantee for the inequalities. To avoid ill-conditioning, the slacks should not approach zero away from the solution, since this would make the barrier term exceedingly large. This is accomplished by computing a modified Newton direction, see Wright (1997b) and Nocedal and Wright (1999), pp. 510-512 for further details. Note that even though the iteration count usually is low for interior methods, the workload of each iteration may be high. Originally interior methods with barriers were suggested by Frich (1954), section 12, and further investigated by Fiacco and McCormick (1990), but they were considered inferior to SQP methods due to the ill-conditioning until Karmarkar (1984) published his algorithm for linear programming. See Martinsen (1998a) for a historical overview of interior methods. Currently primal-dual interior methods for nonlinear programming are being implemented with considerable success, see the section on overview of existing methods below.

Feasibility

We illustrate the difference between infeasible and feasible steps by a conceptual example in figure 3.1:

The infeasible method to the left uses the usual decomposition into null- and range-space steps. The qp-subproblem generates a step $Z_k p_{Z,k}$ in the null-space, while the range-space step $Y_k p_{Y,k}$ is computed from the constraint linearizations at x_k. The vector sum $d_k = Z_k p_{Z,k} + Y_k p_{Y,k}$ gives the next trial-point x_{k+1} indicated on the figure. To ensure convergence from remote starting points a line-search on an exact penalty function is performed along $d_k = x_{k+1} - x_k$ giving the actual next iterate as $x_{k+1} = x_k + \alpha d_k$. We observe that poor scaling and ill-conditioning may unbalance the null- and range-space steps which in turn alters the composite direction d_k which in turn

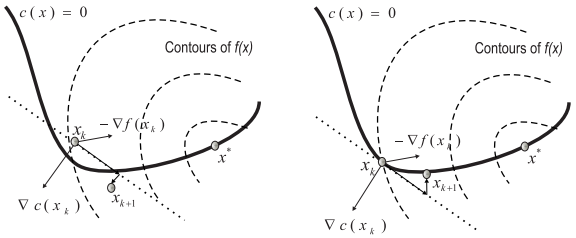

Figure 3.1: *Conceptual comparison between an infeasible SQP (left) and a feasible GRG (right)method.* The dotted line through x_k is the null-space of the equality constraint $c(x)$ at x_k. The range-space is orthogonal to the null-space. The optimum of the constrained objective $f(x)$ is located at x^*.

may cause small steps in the line-search, especially far from the optimum x^*. Note that the line through x_k is evaluated at an infeasible point, and that its slope need not be a "parallel" displacement of the tangent-plane through the point given by projecting x_k onto c. It is even possible that the next iterate x_{k+1} is located at a point where $c(x)$ is undefined. The feasible GRG method to the right alternates between solving a SLC-problem for the free variables (i.e. reduced gradient), and evaluating the model at trial points generated by a modification of the search direction to step back to the equality constraint. Note that the step back onto $c(x) = 0$ does not lie in the range-space in general.

Low level implementation issues

It is important to realize that the actual performance of an algorithm depends heavily on the implementation. That is, some successful codes do not have a sound theoretical foundation, and some that do have poor results. In general, even the very same code can lead to very different performance when compiled with different compilers (or options) or operating systems, due to small differences in rounding, orders in addition etc. For a code, there may be that for some examples, where on one machine the method works fine, and on another just fails; and in particular in ill-conditioned problems the number of iterations can easily differ[1].

`Step computation` by solving the subproblems can be done in the full space of all variables, or by decomposition strategies where dependent variables are eliminated and a dense `qp` is solved for the dependent vari-

[1] Personal cummunication from Andreas Wächter.

ables. Conjugate gradients, possibly projected, may also be considered. The time spent in the `qp` can be reduced if an incomplete solution can be incorporated in the overall algorithm, see Murray and Prieto (1995) for further details.

`Line search/trust-regions` The line search computes a direction in a subproblem and then determines step length by reducing the step length according to the Wolfe conditions applied to a penalty function. Usually only the first Wolfe condition is tested, and the `BFGS` update is skipped if the step length is too short. An implementation of the full Wolfe conditions is provided by Moré and Thuente (1994). Trust regions determine the step length first based on the relative success of previous iterations, and modifies the subproblem for step computation with the additional constraint $\|d_k\| \leq \Delta$ where d_k is the step to be computed and Δ is the (current) trust-region radius. The trust region approach is more involved than the line search, but is better suited to deal with indefiniteness of the model-Hessian. See Nocedal and Wright (1999), sections 3 and 4 for further details.

`Hessian` Exact second derivatives of the objective and constraint functions can be incorporated if the method is able to deal with indefinite Hessians. In favor of second derivatives are sparsity, accuracy and a speed-up due to improved curvature information, i.e. a quadratic convergence rate may be obtained. Counter-acting this is the need for handling a possibly indefinite Hessian, and the cost of producing second derivatives. Two major alternatives for dealing with indefiniteness exist; adding a multiple of the identity to the diagonal or using a trust-region that limits the step-size. Adding a multiple of the identity gives a convex model and can be done by perturbed Cholesky factorization. This gives a model-Hessian that deviates from the original Hessian. Trust-regions, on the other hand, limit the step-size in case of an unbounded solution which may occur with indefinite Hessians. A step to the boundary of the trust-region in the computed direction is then expected to "bypass" saddle points etc., and the iteration can proceed from the new point. Both approaches have been implemented with success in production codes. `BFGS` approximations, on the other hand, lead to a dense positive definite model-Hessian, but the convergence rate is only superlinear. In addition the quality of the model-Hessian may be poor during earlier iterations. In large-scale methods limited memory `BFGS` is used to reduce the storage requirement. See Murray

(1997) for further details and references.

`Penalty` functions, or merit functions, are commonly used to provide global convergence results. Exact penalty functions have minimizers that coincide with the minimizers of the nonlinear program for finite values of the penalty weight. Inexact penalty functions like barrier-functions and the quadratic penalty function, which are smooth in relevant regions, does not have minimizers that coincide with the minimizers of the nonlinear program unless the penalty parameter is infinity. Infinite values of the penalty parameter give ill-conditioning, which is undesirable. Augmented Lagrangian methods are exact penalty functions that are smooth but depend strongly on the multiplier estimates as discussed above. The l_1-penalty function is exact but non-smooth like any norm-based penalty function would be. The l_1-function is less dependent upon the multiplier estimates, but needs complicated line search mechanisms (a watch-dog is frequently used) to accept the full Newton step locally. Hence, augmented Lagrangian methods have good local convergence properties, but global convergence is hard to establish. On the other hand, l_1-methods have better global convergence properties, but local convergence needs sophisticated line search mechanisms to be superlinearly convergent. See Fletcher (1987), chapters 12 and 14 for further details.

`Numerical linear algebra` is essential for the actual performance of any algorithm. On large-scale problems exploiting sparsity is essential. Avoiding unnecessary factorizations by directly updating the factorization itself is common. The choice of pivoting strategies is important for rank-deficient systems, see Golub and Van Loan (1996) for further details.

Other work on reduced Hessian methods

The method of Biegler et al. (1995) maintains a dense approximation of the reduced Hessian using quasi-Newton updates. In Tjoa and Biegler (1991) inequality constraints are slacked, and the slacks are resolved using a dense `qp` algorithm. These methods are designed for problems dominated by equality constraints with few degrees of freedom. Betts and Frank (1994) use second derivatives in combination with a sparse `qp` solver and sparse `KKT` factorizations and is not limited to problems with few degrees of freedom. Coleman and Conn (1982b), (1982a) consider inequalities, but need to resort to finite differences for the reduced Hessian to converge which can be costly. `SNOPT`

uses a full limited memory Hessian but computes the step by a null-space approach involving the reduced (i.e. projected) Hessian. A number of further references on reduced Hessian methods is given and discussed by Biegler et al. (1995).

Other work on feasible methods

By feasible methods we understand methods that remain feasible with respect to all inequalities at all intermediate iterates. These methods may still evaluate the problem functions outside the feasible region for example in the line search. We note that any method can easily be made feasible with respect to linear inequalities, so our discussion implicitly assumes the presence of nonlinear inequalities.

Mayne and Polak (1976) suggested a modification that would make any interior point method feasible. Their method were adapted to `SQP` methods by Tits and co-workers in a series of papers; Panier and Tits (1987), (1993), Lawrence and Tits (1996), (2000). Their algorithm `FSQP` has emerged as a reliable code which is commercially available. Interior methods can also be designed as feasible methods by including the inequalities and not the slacks in the barrier function. This approach has been pursued by Gay, Overton, and Wright (1998), but leads to complex algebraic manipulations in the derivation of the algorithm. Recently Byrd, Nocedal, and Waltz (2000) have suggested a feasible interior method using slack resetting with good results.

In optimal control feasibility of equality constraints has been approached by evolution of the dynamic model constraint over the relevant time-span. See Pytlak (1999) for further references. We return to these issues in chapter 4.

Overview of other large scale methods

The methods described in the following are a selection of established optimization codes. The selection is not exhaustive, and the description of each method is only of limited depth. The interested reader will need to resort to the cited references for a deeper presentation of each method.

`GRG` Until the mid-eighties generalized reduced-gradient (`GRG`) methods were recognized as the most successful methods for nonlinear programming. `GRG` methods try to remain feasible with respect to equalities by a feasibility restoration phase following the step computation. The feasibility restoration phase needs to evaluate the constraint functions at possibly

many trial points which can be costly. GRG methods therefore tend to perform poorly on problems with severe nonlinearities in the equality constraints, and when initialized far from the optimum. To remedy this practical implementations do not require feasibility at early iterations, and thus become more related to projected Lagrangian methods. For a description of a general GRG method see Gill et al. (1981), pp. 220-223. See Edgar and Himmelblau (1989), p. 362, for a comparison of GRG with other methods available in the mid-eighties. Further references are provided in the references above. Today the most successful GRG implementation is CONOPT by Drud (1985).

MINOS is a SLC method which solves a linearly constrained subproblem with a nonlinear objective. Although SLC methods are commonly agreed to be inferior to SQP methods due to the excessive number of function evaluations, MINOS by Murtagh and Saunders (1995) remains a successful implementation. For a description of a general SLC method see Gill et al. (1981), p.234-236, and Nocedal and Wright (1999), pp. 523-525. For an assessment of the performance of MINOS see Bondarenko et al. (1999). MINOS and CONOPT are designed for large problems with a modest number of degrees of freedom. MINOS does not implement globalization mechanisms like merit functions.

LANCELOT is a sequential augmented Lagrangian method. LANCELOT by Conn, Gould, and Toint (1992) is designed for large problems with many degrees of freedom. See Bondarenko et al. (1999) for numerical results.

SNOPT by Gill, Murray, and Saunders (1997) is an active-set SQP algorithm using a modified augmented Lagrangian merit function with line search. Global convergence results are difficult to establish because the merit function is treated as a function of the variables, slacks and multipliers and is not minimized by a solution point, i.e. it is not an exact penalty method. The convergence theory for SNOPT is similar to the properties of NPSOL addressed in Gill, Murray, Saunders, and Wright (1992). SNOPT uses a limited memory convex quasi-Newton approximation to the Hessian of the Lagrangian. It uses a reduced Hessian approach in solving the qp subproblems. See Gill et al. (1997), Bondarenko et al. (1999) and Morales et al. (2001) for numerical evaluations. SNOPT is expected to outperform MINOS on problems with highly nonlinear constraints.

`filterSQP` by Fletcher and Leyffer (1998) is a trust-region `SQP` method using second derivatives of the objective and constraints, and it solves the subproblem by an active-set method for indefinite quadratic programming. If a direction of negative curvature is detected it takes a step to the trust-region boundary in this direction. `filterSQP` uses a "filter" mechanism that eliminates the need for a merit function. See Morales et al. (2001) for numerical evaluations. Convergence results are an ongoing research.

`LOQO` by Vanderbei and Shanno (1999) implements a line search primal-dual interior method. It uses second derivatives of the objective and constraint functions, and slacks for inequality constraints. It uses the quadratic inexact penalty function, hence global convergence results are difficult to obtain. It uses a variant of modified Cholesky factorization to handle indefinite subsystems. See Morales et al. (2001) for numerical evaluations.

`KNITRO` is a trust-region primal-dual interior-method using second derivatives. Unlike most interior methods it computes the steps in a `SQP`-like manor that approximately minimize the barrier function. It uses projected conjugate gradients (`CG`) to handle indefiniteness. I.e. when the `CG`-iteration breaks down due to indefiniteness, it takes a step to the trust-region boundary in the computed direction. It uses a non-differentiable merit function for global convergence. `KNITRO` is described in Byrd, Hribar, and Nocedal (1999), and analyzed in Byrd, Liu, and Nocedal (1998) and Byrd, Gilbert, and Nocedal (2000). See Morales et al. (2001) for numerical evaluations.

3.2 Introduction

In this section we suggest a feasible reduced Hessian `SQP` method. The method implements the l_1-merit function and `BFGS` updates for the reduced Hessian. The method is feasible with respect to inequality constraints. Since inequalities are the main problem in reduced Hessian methods, see Nocedal and Wright (1999) p.551, our method is aimed at continuous large-scale nonlinearly constrained optimization problems with few degrees of freedom. In particular we expect our method to be efficient for problems dominated by nonlinear equality constraints, with relatively few nonlinear inequalities. Note that any `SQP` method will be feasible with respect to affine inequalities and simple bounds. The method extends `FSQP`' of Tits and co-workers Panier

and Tits (1987), (1993), Lawrence and Tits (1996), (2000) to the reduced Hessian setting. The work of Tits and co-workers extends the work of Mayne and Polak (1976).

The local convergence analysis for the present algorithm simply states that after the correct active set has been identified, the problem is equality constrained and two-step superlinear convergence follows as in Byrd and Nocedal (1991) or one-step superlinear convergence as in Biegler, Nocedal, and Schmid (1995).

In the following the NLP

$$\begin{array}{ll} \min\limits_{x_k} & f(x_k) \\ s.t. & c_{\mathcal{E}}(x_k) = 0 \\ & c_{\mathcal{I}}(x_k) \leq 0 \end{array} \tag{3.1}$$

is considered with objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, equality constraints $c_{\mathcal{E}} : \mathbb{R} \rightarrow \mathbb{R}^{p_1}$ and inequality constraints $c_{\mathcal{I}} : \mathbb{R} \rightarrow \mathbb{R}^{p_2}$. n is the number of variables, p_1 is the number of nonlinear equality constraints and p_2 is the number of nonlinear inequality constraints. Subscript k refers to iteration index. We design our algorithm for the case $p_1 >> p_2 > 0$, which corresponds to a large number of nonlinear equality constraints and few nonlinear inequality constraints. We associate two sets with the NLP (3.1):

$$\begin{aligned} X &= \{x_k \in \mathbb{R}^n \mid c_{\mathcal{E}}(x_k) = 0,\ c_{\mathcal{I}}(x_k) \leq 0\} \\ X_I &= \{x_k \in \mathbb{R}^n \mid c_{\mathcal{I}}(x_k) \leq 0\} \end{aligned}$$

with $X \subseteq X_I$. A point $x_k \in X_I$ satisfies the inequality constraints but not necessarily the equality constraints. Points in X are feasible, while points in X_I are (at least) feasible with respect to inequalities. We will generate iterates $x_k \in X_I$ and a solution $x^* \in X$. The Lagrangian $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \rightarrow \mathbb{R}$ associated with (3.1) is defined by

$$\mathcal{L}(x_k, \lambda_{\mathcal{E},k}, \lambda_{\mathcal{I},k}) \triangleq f(x_k) + \sum_{i=1}^{p_1} \lambda^i_{\mathcal{E},k} c^i_{\mathcal{E}}(x_k) + \sum_{i=1}^{p_2} \lambda^i_{\mathcal{I},k} c^i_{\mathcal{I}}(x_k) \tag{3.2}$$

where the real-valued vectors $\lambda_{\mathcal{E},k}$ and $\lambda_{\mathcal{I},k}$ are the Lagrange multipliers.

Practical experience with reduced Hessian methods indicate that they require a good starting point to converge. This is expressed in assumption 4.1 in Biegler et al. (1995), i.e. that the sequence of iterates $\{x_k\}$ is contained in a convex set D. In the present method this assumption can be satisfied by a suitable choice of the feasible set X_I with respect to the inequalities.

The prize to pay for this is the need for an initial point $x_0 \in X_I$. Problems dominated by nonlinear equality constraints, possibly with nonlinear inequality constraints, occur in optimal control problems such as nonlinear MPC. Nonlinear inequalities may occur if there are quality-related constraints and quality is a nonlinear function of the problem variables, i.e. such as $q = q(x^*)$ and $q(x_k) \leq 0$. Solving this with q as a slack variable cannot guarantee $q(x_k) \leq 0$ away from the solution x^*. Initialization in nonlinear MPC may involve a shifted solution from the previous MPC-iteration, hence a feasible initial point is a reasonable assumption, see Allgöwer, Badgwell, Qin, Rawlings, and Wright (1999) and the references therein.

General benefits of feasible iterates are summarized in Lawrence and Tits (2000), and we restate them here with some comments pertaining to our algorithm.

- The qp subproblems may become infeasible if the constraint linearizations are inconsistent. This can happen since we include equality constraints in addition to inequalities.

- The objective cannot be used as merit function when equality constraints are present. We use the l_1-merit function for global convergence.

- Keeping the iterates inside a feasible region can ensure that the problem functions always are evaluated at points where they are defined. If the feasible region is defined by linear constraints and simple bounds, this can be guaranteed for any SQP-method. In our method a Maratos correction term is computed by evaluating the constraint functions at the point $x_k + d_k$, where the vector d_k is a step. The point $x_k + d_k$ may lie outside the set X_I unless X_I is defined by linear constraints and simple bounds. See section 3.3.2, equation (3.32) for details. In addition the line search may evaluate the problem functions at points where inequality constraints are infeasible. These considerations applies to FSQP' as well.

- Meaningful exploration of trade-offs between design alternatives.

- The optimization algorithm will have iterates that are feasible with respect to inequalities. If terminated prior to convergence the non-optimal solution may be utilized.

The usual SQP direction finding qp associated with (3.1) is

$$\begin{array}{ll} \min\limits_{d_k^0} & \frac{1}{2}\left\langle d_k^0, W(x_k)d_k^0\right\rangle + \left\langle g_k, d_k^0\right\rangle \\ s.t. & c_{\mathcal{E}}^i(x_k) + \left\langle \nabla c_{\mathcal{E}}^i(x_k), d_k^0\right\rangle = 0 \quad i = \{1,\ldots,p_1\} \\ & c_{\mathcal{I}}^i(x_k) + \left\langle \nabla c_{\mathcal{I}}^i(x_k), d_k^0\right\rangle \leq 0 \quad i = \{1,\ldots,p_2\} \end{array} \tag{3.3}$$

where we introduce the objective gradient $g_k = \nabla f_k$ and $W(x_k)$ is the Hessian $\nabla^2\mathcal{L}_k$ of the Lagrangian or an approximation to it. The qp (3.3) produces the ordinary SQP direction d_k^0. Superscript 0 refers to ordinary SQP directions throughout this chapter. The rFSQP algorithm tilts the ordinary SQP direction and produces a direction d_k.

A feasible direction for $x_k \in X_I$ is a direction $d_k^0 \in \mathbb{R}^n$ generated by (3.3) such that there exists an $\bar{\alpha} > 0$ satisfying

$$\left(x_k + \alpha d_k^0\right) \in X_I \ \ \forall \alpha \in [0, \bar{\alpha}] \tag{3.4}$$

Define the index-set of ϵ-active inequality constraints as

$$J^\epsilon(x_k) \triangleq \left\{i \mid -\epsilon < c_{\mathcal{I}}^i(x_k) \leq 0\right\} \tag{3.5}$$

for a small $\epsilon > 0$. Observe that (3.3) allows $d_k^0 \in span\left\{\nabla c_{\mathcal{I}}^i(x_k) \mid c_{\mathcal{I}}^i(x_k) = 0\right\}^\perp$ which may violate (3.4). Define

$$A(x_k) = \left[\nabla c_{\mathcal{E}}^i(x_k)\ \nabla c_{\mathcal{I}}^j(x_k) \mid \forall i \in \{1,\ldots,p_1\}, \forall j \in J^\epsilon(x_k)\right] \tag{3.6}$$

To the best of our knowledge only Coleman and Conn (1982b), (1982a) give theoretical results for inequality constrained reduced Hessian methods. Tjoa and Biegler (1991) include simple bounds, and use slacks for the inequalities. In addition there exists a successful implementation, SNOPT, by Gill et al. (1997) with promising results (see Bondarenko et al. (1999) for results). We observe that varying dimension of the working set matrix will present additional challenges when inequality constraints are present in reduced Hessian methods, since BFGS updates do not generally allow varying dimensions. We discuss this issue as part of our implementation, but do not intend to resolve this issue since we take the view that since we are able to identify the correct active set at a finite but potentially large iteration index k_0 the problem is equality constrained thereafter. Hence, the tools of Byrd and Nocedal (1989) can be applied to handle the case of infinitely many BFGS updates.

This section continues with a description of the algorithm in section 3.3, then global and local convergence analysis are provided in section 3.4. Some implementation issues and numerical results are discussed in section 3.5, and some concluding remarks are provided in section 3.6.

3.3 The rFSQP algorithm

We adapt the feasibility mechanisms from FSQP' (Lawrence and Tits 2000) to the reduced Hessian setting. Observe that neither the equality constraints nor the reduced gradients we wish to investigate are present in FSQP'. We make the assumptions:

Assumption 3.1 *$X \neq \emptyset$.*

Assumption 3.2 *The functions $f : \mathbb{R}^n \to \mathbb{R}$, $c^i_{\mathcal{E}} : \mathbb{R}^n \to \mathbb{R}$ and $c^i_{\mathcal{I}} : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable on X_I for all i.*

Assumption 3.3 *The set $\{\nabla c^i_{\mathcal{E}}(x^*) \mid \forall i \in \{1, \ldots, p_1\}\} \cup \{\nabla c^i_{\mathcal{I}}(x^*) \mid i \in J^{*,\epsilon}\}$ is linearly independent (LICQ).*

Assumption 3.4 *Strict complementary slackness holds at x^*, i.e. $c^i_{\mathcal{I}}(x^*)\lambda^{*,i}_{\mathcal{I}} = 0$ for $i = \{1 \ldots p_2\}$ and if $c^i_{\mathcal{I}}(x^*) = 0$ then $\lambda^{*,i}_{\mathcal{I}} > 0$.*

Under assumptions 3.1-3.4 and first and second order KKT-conditions, it is well known (see, for example Fiacco and McCormick (1990), Fletcher (1987)) that x^* is an isolated local constrained minimizer of the NLP (3.1) with unique multipliers. We assume that at least one such isolated local constrained minimizer of (3.1) exists. The LICQ-assumption guarantees that the constraint linearizations are consistent at x^*. Note that we make a stronger assumption related to the LICQ-assumption in section 3.4.2, assumption 3.6. Recovering from infeasibility resulting from inconsistency is considered an implementation issue, see Marazzi and Nocedal (2000) for an exposition and Gill et al. (1997) for an example strategy.

3.3.1 The feasibility mechanism

We restate the essential features of FSQP' (Lawrence and Tits 2000). In FSQP' the qp (3.3) is perturbed as

$$\begin{array}{ll} \min\limits_{d_k,\gamma_k} & \frac{1}{2}\langle d_k, W_k d_k\rangle + \gamma_k \\ s.t. & \langle g_k, d_k\rangle \leq \gamma_k \\ & c^i_{\mathcal{E}}(x_k) + \langle \nabla c^i_{\mathcal{E}}(x_k), d_k\rangle = 0 \qquad i = \{1, \ldots, p_1\} \\ & c^i_{\mathcal{I}}(x_k) + \langle \nabla c^i_{\mathcal{I}}(x_k), d_k\rangle \leq \gamma_k \eta_k \quad i = \{1, \ldots, p_2\} \end{array} \tag{3.7}$$

with real scalars γ_k and $\eta_k \geq 0$, and where we have added equality constraints. The tilting parameter η_k controls the amount of tilting. Large values emphasize feasibility, while small values emphasize descent. Lawrence

and Tits (1996) handle equality constraints by rewriting them as doubly bounded inequalities, while we keep them as equalities. Problem (3.7) motivates the derivation of our reduced gradient mechanism in section 3.3.2. Note that $sign(\gamma_k) = -sign(\eta_k)$ must be inferred to yield feasibility of the inequality constraints.

Lawrence and Tits (1996) update η_k based on the following least-squares problem. Given an estimate J_k^E of the active set, `FSQP'` computes an estimate d_k^E of the `SQP`-direction d_k^0 by solving the `qp` (3.8).

$$\begin{array}{ll} \min\limits_{d_k^E} & \frac{1}{2}\left\langle d_k^E, W_k d_k^E \right\rangle + \left\langle g_k, d_k^E \right\rangle \\ s.t. & c_{\mathcal{E}}^i(x_k) + \left\langle \nabla c_{\mathcal{E}}^i(x_k), d_k^E \right\rangle = 0, \quad i = \{1, \ldots, p_1\} \\ & c_{\mathcal{I}}^i(x_k) + \left\langle \nabla c_{\mathcal{I}}^i(x_k), d_k^E \right\rangle = 0, \qquad i \in J_k^E \end{array} \tag{3.8}$$

This estimate is used to update the parameter η_k to provide adequate tilting into the feasible set, and in `FSQP'` this is cheaper to compute than d_k^0 itself since (3.8) is a least-squares problem.

The present algorithm solves `qp`'s in the reduced space, so the need for the estimate d_k^E to reduce the computational workload is less important. Hence, we simply compute the `SQP`-direction. The notation d_k^E is retained for ease of exposition and to allow direct comparison between `FSQP'` and `rFSQP`. Showing that $d_k^E = d_k^0$ for k sufficiently large is a simple observation, but it is still necessary to show that

$$\eta_k = \mathcal{O}(\left\| d_k^E \right\|^2) \tag{3.9}$$

(with a suitable order constant C_k^η) is sufficient to guarantee global convergence and local superlinear convergence.

In `FSQP'` a Maratos effect avoidance scheme is implemented near the solution by

$$\begin{array}{ll} \min\limits_{d_k^C} & \frac{1}{2}\left\langle d_k + d_k^C, W_k(d_k + d_k^C) \right\rangle + \left\langle g_k, d_k + d_k^C \right\rangle \\ s.t. & c_{\mathcal{E}}^i(x_k + d_k) + \left\langle \nabla c_{\mathcal{E}}^i(x_k), d_k^C \right\rangle = 0, \qquad i = \{1, \ldots, p_1\} \\ & c_{\mathcal{I}}^i(x_k + d_k) + \left\langle \nabla c_{\mathcal{I}}^i(x_k), d_k^C \right\rangle = -\left\| d_k \right\|^\kappa, \qquad i \in J_k^\epsilon \end{array} \tag{3.10}$$

In (3.10) d_k^C denotes the correction to the direction d_k, and κ is a scalar in the range $(2, 3)$. However, `rFSQP` is implemented with an l_1-penalty function, and we do not expect (3.10) to remedy the well-known Maratos effect associated with this. Instead, to promote local convergence analysis, we assume that a step-length of one is accepted near the optimum, and work through the local analysis in view of this. This assumption is also made by Biegler

et al. (1995), and is motivated by that the Maratos effect can be handled by implementing a non-monotone line-search like the watch-dog method of Chamberlain, Powell, Lemarechal, and Pedersen (1982). Other alternatives are listed in Nocedal and Wright (1999), section 18.11. Summarizing, the correction d_k^C from (3.10) is needed to guarantee that the feasibility condition $c_{\mathcal{I}}^i(x_k + d_k + d_k^C) \leq 0$ with a unit step-length is satisfied. I.e. that the watch-dog method can be implemented without interfering with feasibility of inequality constraints.

We require that the sequence $\{d_k\}$ possesses the following properties:

P1 $d_k = 0$ if x_k is a `KKT` point for (3.1).

P2 $D\phi_\nu(x_k, d_k) < 0$ if x_k is not a `KKT` point for (3.1) (the l_1-merit function ϕ_ν is to be defined in (3.11)). The directional derivative operator D is used due to the non-derivative property of l_1-merit functions.

P3 $\left\langle \nabla c_{\mathcal{I}}^i(x_k), d_k \right\rangle < 0$ for all $i \in J_k^\epsilon$ if x_k is not a `KKT` point for (3.1).

P4 $d_k = d_k^0 + \mathcal{O}(\left\| d_k^0 \right\|^2)$.

Observe that P2 differs from P2 in Lawrence and Tits (2000) since we need a descent in the l_1-merit function. Next the reduced gradient method and the algorithm will be outlined.

3.3.2 The feasible reduced Hessian method (rFSQP)

The algorithm is essentially a restatement of algorithm II by Biegler et al. (1995) with some extensions to allow for the additional inequality constraint and the parameters γ_k and η_k in equation (3.7). Note that the dimension of A_k generally will change when the active set J_k^ϵ changes. The next iterate is computed as $x_{k+1} = x_k + \alpha_k d_k + \alpha_k^2 d_k^C$ where α_k is a step-length parameter chosen to reduce the value of the constrained l_1-merit function

$$\begin{aligned} \phi_\nu(x_k) &= f(x_k) + \nu_k \left\| c_{\mathcal{E}}(x_k) \right\|_1 \\ s.t.\ \ & c_{\mathcal{I}}(x_k) \leq 0 \end{aligned} \tag{3.11}$$

with penalty parameter ν_k. Including a term $\nu_k \max\{0, c_{\mathcal{I},k}\}$ is oblivious by a strictly feasible search direction. Observe that (3.11) differs from the merit functions used in both Biegler et al. (1995) and Lawrence and Tits (2000).

Define the nonsingular matrix $[Y_k\ Z_k]$ where $Y_k \in \mathbb{R}^{n \times m_k}$ and $Z_k \in \mathbb{R}^{n \times (n - m_k)}$. Y_k is a basis for the range space $\mathcal{R}(A_k^T)$ of the Jacobian A_k

from equation (3.6), while Z_k is a basis for the null space $\mathcal{N}(A_k^T)$. The dependence of m on k where $m_k = p_1 + |J_k^\epsilon|$ allows adaption to varying size of the working set matrix A_k. Here $|J_k^\epsilon|$ denotes the length, i.e. number of elements in the vector J_k^ϵ. The null-space of A_k^T is the tangent space of the active constraints, and since Z_k is a basis for the null-space of A_k^T we have

$$A_k^T Z_k = 0 \tag{3.12}$$

We choose the variable-reduction factorization, see Gill et al. (1981), p.221, $A(x_k)^T = [C(x_k)\ N(x_k)]$ where

$$Z(x_k) = \begin{bmatrix} -C(x_k)^{-1}N(x_k) \\ I \end{bmatrix}, \quad Y(x_k) = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{3.13}$$

More details on the factorization are found in Xie and Byrd (1999). The solution to (3.7) can then be separated as

$$d_k = Y_k p_{Y,k} + Z_k p_{Z,k} \tag{3.14}$$

where the vectors $p_{Y,k} \in \mathbb{R}^{m_k}$ and $p_{Z,k} \in \mathbb{R}^{(n-m_k)}$. By (3.12), the range-space step becomes

$$p_{Y,k} = -(A_k^T Y_k)^{-1} c_k \tag{3.15}$$

where

$$c_k \triangleq \begin{bmatrix} c_{\mathcal{E}}(x_k) \\ \tilde{c}_{\mathcal{I}}(x_k) \end{bmatrix} \quad \text{and} \quad \tilde{c}_{\mathcal{I}}(x_k) \triangleq \left\{ c_{\mathcal{I}}^i(x_k) - \gamma_{k-1}\eta_k \mid i \in J_k^\epsilon \right\} \tag{3.16}$$

Since γ_k is unavailable at this stage γ_{k-1} is used. Convergence of the sequence $\{\gamma_k\}$ is established in section 3.4, hence, using γ_{k-1} is sufficient. Then $d_k = -Y_k(A_k^T Y_k)^{-1} c_k + Z_k p_{Z,k}$. Observe that $Y_k(A_k^T Y_k)^{-1}$ is a right inverse of A_k^T and that (3.15) is a particular solution to the linearized equations. To compute d_k, solve for $p_{Z,k}$ and γ_k from (derived in section 3.4)

$$\begin{aligned} B_k p_{Z,k} &= -Z_k^T g_k - \zeta_k w_k \\ \left\langle Z_k^T g_k + \zeta_k w_k, p_{Z,k} \right\rangle &\leq \gamma_k \end{aligned} \tag{3.17}$$

In (3.17) the cross-term approximation $Z_k^T W_k Y_k p_{Y,k} = \zeta_k w_k$ is introduced where ζ_k is a scalar damping parameter and w_k is an approximation to the rectangular matrix-vector product $Z_k^T W_k Y_k p_{Y,k}$. Assuming positive definiteness of the reduced Hessian approximation B_k, equation (3.17) gives a unique $p_{Z,k}$ while there is freedom in choosing γ_k. This might have been

anticipated, since augmenting the problem (3.7) with γ_k gives a semidefinite Hessian.

Since the inactive constraint linearizations are excluded, $p_{Z,k}$ is computed by an `EQP` strategy, see Gill et al. (1981), p. 244, Fletcher (1987), sections 11.2 and 11.3, and Nocedal and Wright (1999), p. 534. The main motivation for choosing the `EQP` strategy is that it allows use of theorem 3.1 from Byrd and Nocedal (1989) restated below (note that $Z_k^T \nabla c_{\mathcal{I},k}^i \neq 0$ when $i \notin J_k^\epsilon$ in general). The active inequalities are treated as equalities and are included in c_k in (3.15).

The Lagrange multipliers are estimated from

$$\lambda_k \triangleq [\lambda_{\mathcal{E},k}\ \lambda_{\mathcal{I},k}] = -(Y_k^T A_k)^{-1} Y_k^T g_k \tag{3.18}$$

where $\lambda_{\mathcal{I},k} = \left\{ \lambda_{\mathcal{I},k}^i \mid i \in J_k^\epsilon \right\}$. This is a first-order estimate of the form $\min \|A_k \lambda_k + g_k\|$, obtained by observing that $(Y_k^T A_k)^{-1} Y_k^T$ is a left-inverse to A_k. I.e. any $g_k \in \mathcal{R}(A_k)$ is compatible with (3.18), and we cannot assume that the estimated multipliers coincide with the `NLP`-multipliers away from a solution x^*, see Gill et al. (1981), p. 248. By assumption 3.3 (`LICQ`), we have that λ^* exists and is unique. We choose the multipliers $\lambda_{\mathcal{I},k}^i = 0$, $i \notin J_k^\epsilon$. The update-rule for ζ_k follows by solving

$$\begin{gathered} \hat{\zeta}_k \left[2 \cos \theta_k^{BFGS}(p_{Z,k}) \left| g_k^T Z_k w_k \right| + w_k^T B_k^{-1} Z_k^T g_k + \hat{\zeta}_k w_k^T B_k^{-1} w_k \right] = \rho \|c_k\|_1 \\ \zeta_k = \min \left\{ 1, \hat{\zeta}_k \right\} \end{gathered} \tag{3.19}$$

for a constant $\rho > 0$. $\cos \theta_k^{BFGS}$ is defined in equation (3.38), and measures the angle between $p_{Z,k}$ and $B_k p_{Z,k}$. The null-space component $Z_k p_{Z,k}$ must make an acute angle with the projection of $-g_k$ onto the null-space to give descent. For the moment neglecting the cross-term w_k, we have from (3.17) that $B_k p_{Z,k} = -Z_k^T g_k$. Byrd and Nocedal (1991) derive the $\cos \theta_k^{BFGS}$ relation for reduced Hessian methods. Note the dependence $\zeta_k(p_{Z,k})$ through $\cos \theta_k^{BFGS}(p_{Z,k})$, which is not available at this stage since ζ_k must be computed before (3.17). Therefore the actual implementation is done as in Biegler et al. (2000). The update rules for the damping factor $\zeta_k \in (0, 1]$ and the penalty parameter ν_k

$$\nu_k = \begin{cases} \nu_{k-1} & \text{if } \nu_{k-1} \geq \|\lambda_k\|_\infty + 2\rho \\ \|\lambda_k\|_\infty + 3\rho & \text{otherwise} \end{cases} \tag{3.20}$$

are derived in lemma 3.3, section 3.4.2 and follows Biegler et al. (1995) closely.

As in Biegler et al. (1995) we approximate the cross term $Z_k^T W_k Y_k$ by a rectangular matrix S_k. Since $W_{k+1} = \nabla^2_{xx}\mathcal{L}(x_{k+1}, \lambda_{k+1})$, the approximation

$$Z_k^T W_{k+1}(x_{k+1} - x_k) \approx Z_k^T \left[\nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1})\right] \tag{3.21}$$

holds when x_{k+1} is close to x_k. I.e., in section 3.4.3, lemma 3.11, we prove that $J_k^\epsilon = J^*$ for all k sufficiently large. If $Z(x_k)$ is locally Lipschitz continuous the approximation (3.21) holds locally, and the comments in Nocedal and Wright (1999), p. 567, applies. This continuity assumption is also made in Biegler et al. (1995). The error in the approximation is of order $\mathcal{O}(\|d_k\|^2)$. The matrix S_{k+1} is required to satisfy the secant relation

$$S_{k+1}(x_{k+1} - x_k) = Z_k^T \left[\nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1})\right]$$

and comparing with (3.21) we have $S_{k+1} \approx Z_k^T W_{k+1}$. Hence, computing the full matrix $Z_k^T W_k Y_k$ is avoided as in Biegler et al. (1995) by approximating S_k by Broyden's method, (see Dennis and Schnabel (1996), p. 170):

$$S_{k+1} = S_k + \frac{(\bar{y}_k - S_k \bar{s}_k)\, \bar{s}_k^T}{\bar{s}_k^T \bar{s}_k} \tag{3.22}$$

where

$$\begin{aligned} \bar{y}_k &= Z_k^T \left[\nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1})\right] \\ \bar{s}_k &= x_{k+1} - x_k \end{aligned} \tag{3.23}$$

and computing

$$\begin{aligned} w_k &= S_k Y_k p_{Y,k} \\ \bar{w}_k &= \alpha_k S_{k+1} Y_k p_{Y,k} \end{aligned} \tag{3.24}$$

Since the Broyden approximations may become unbounded they are safeguarded as in Biegler et al. (1995). I.e. choose a constant Γ and define

$$w_k = \begin{cases} w_k & \text{if } \|w_k\| \le \frac{\Gamma}{\|p_{Y,k}\|^{1/2}} \|p_{Y,k}\| \\ w_k \frac{\Gamma \|p_{Y,k}\|^{1/2}}{\|w_k\|} & \text{otherwise} \end{cases} \tag{3.25}$$

Choose a sequence of positive numbers $\{\gamma_k^w\}$ such that $\sum_{k=1}^\infty \gamma_k^w < \infty$, and set

$$\bar{w}_k = \begin{cases} \bar{w}_k & \text{if } \|\bar{w}_k\| \le \frac{\alpha_k \|p_{Y,k}\|}{\gamma_k^w} \\ \bar{w}_k \frac{\alpha_k \|p_{Y,k}\|}{\gamma_k^w \|\bar{w}_k\|} & \text{otherwise} \end{cases} \tag{3.26}$$

However, if the BFGS update criterion (to be discussed) is not satisfied, w_k is recalculated by finite differences:

$$\begin{aligned} w_k &= Z_k^T \left[\nabla_x \mathcal{L}(x_k + Y_k p_{Y,k}, \lambda_k) - \nabla_x \mathcal{L}(x_k, \lambda_k)\right] \\ \bar{w}_k &= Z_k^T \left[\nabla_x \mathcal{L}(x_k + \alpha_k Y_k p_{Y,k}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1})\right] \end{aligned} \tag{3.27}$$

which require reevaluation of the constraint gradients at two extra points. Biegler et al. (1995) argues that finite difference updating of w_k and $\bar{w}_k$ does not occur too often. However, with inequality constraints and changes in the active set finite difference updating is likely to be required more often since the conditions for application of the Broyden-update, see (3.36) below, may be harder to satisfy when adding constraints to the active set. The cross term approximations w_k and $\bar{w}_k$ does not influence on the global convergence properties. Hence, an efficient implementation may only calculate them if the algorithm seems to approach a solution to speed up the local convergence rate.

The auxiliary subproblems (3.8) and (3.10) from FSQP' must be adopted before the algorithm is summarized. Since d_k^C is computed with J_k^ϵ and d_k^E is computed with J_{k+1}^ϵ, no extra factorizations of A_k is needed to compute d_k^C and d_k^E. (Since the algorithm can compute d_k^E after the new updates f_{k+1}, g_{k+1}, c_{k+1}, A_{k+1}, Y_{k+1}, Z_{k+1} and B_{k+1} are available, the next active set is available to us at this stage, so we can actually compute the SQP-direction directly (without estimating J_k^E as in FSQP')).

Consider the null-space approach for (3.8)

$$d_{k+1}^E = Y_{k+1} p_{Y,k+1}^E + Z_{k+1} p_{Z,k+1}^E \tag{3.28}$$

where $p_{Y,k+1}^E \in \mathbb{R}^{m_{k+1}}$ and $p_{Z,k+1}^E \in \mathbb{R}^{(n-m_{k+1})}$ are computed from

$$p_{Y,k+1}^E = -(A_{k+1}^T Y_{k+1})^{-1} c_{k+1}^E \tag{3.29}$$

where $c_{k+1}^E \triangleq \begin{bmatrix} c_{\mathcal{E}}(x_{k+1}) \\ \tilde{c}_{\mathcal{I}}(x_{k+1}) \end{bmatrix}$ with $\tilde{c}_{\mathcal{I}}(x_{k+1}) \triangleq \left\{c_{\mathcal{I}}^i(x_{k+1}) \mid i \in J_{k+1}^\epsilon\right\}$, and the unconstrained (observe that (3.8) only has equality constraints)

$$\min_{p_{Z,k+1}^E} \quad \left\langle Z_{k+1}^T g_{k+1} + \zeta_{k+1}^E w_{k+1}^E, p_{Z,k+1}^E \right\rangle + \tfrac{1}{2} \left\langle p_{Z,k+1}^E, B_{k+1} p_{Z,k+1}^E \right\rangle$$

The approximation $\zeta_{k+1}^E w_{k+1}^E$ is computed in the same manner as $\zeta_{k+1} w_{k+1}$. Assuming a positive definite B_{k+1} this gives

$$B_{k+1} p_{Z,k+1}^E = -\left(Z_{k+1}^T g_{k+1} + \zeta_{k+1}^E w_{k+1}^E\right) \tag{3.30}$$

The Maratos correction term d_{k+1}^C has a similar decomposition, observing that only equality constraints are present in (3.10). Construct the right-hand side $c_k^C = \begin{bmatrix} c_{\mathcal{E}}(x_k + d_k) \\ \tilde{c}_{\mathcal{I}}(x_k + d_k) \end{bmatrix}$ with $\tilde{c}_{\mathcal{I}}(x_k+d_k) = \left\{c_{\mathcal{I}}^i(x_k + d_k) \ + \|d_k\|^\kappa \mid i \in J_k^\epsilon\right\}$. Compute

$$d_k^C = Y_k p_{Y,k}^C + Z_k p_{Z,k}^C \tag{3.31}$$

where $p_{Y,k}^C \in \mathbb{R}^{m_k}$ and $p_{Z,k}^C \in \mathbb{R}^{(n-m_k)}$,

$$p_{Y,k}^C = -(A_k^T Y_k)^{-1} c_k^C \tag{3.32}$$

and the unconstrained

$$\min_{p_{Z,k}^C} \ \left\langle Z_k^T g_k + \zeta_k(w_k + w_k^C) + B_k p_{Z,k}, p_{Z,k}^C \right\rangle + \tfrac{1}{2} \left\langle p_{Z,k}^C, B_k p_{Z,k}^C \right\rangle$$

giving (assuming B_k positive definite)

$$B_k p_{Z,k}^C = -\left(Z_k^T g_k + \zeta_k(w_k + w_k^C) + B_k p_{Z,k}\right) \tag{3.33}$$

The approximation w_k^C is computed by the same method as w_k was computed (Broyden or finite differences). Note that equations (3.15, 3.29, 3.32) requires solving a set of linear equations with changing right-hand sides which can be done efficiently, since the same factorizations apply. However, this strategy requires evaluation of the constraint functions at three different points in addition to the finite difference updates for w_k and $\bar{w}_k$ above, which may be costly.

3.3.3 The BFGS update scheme

An updating scheme for the reduced Hessian $B_k = Z_k^T W_k Z_k$ allowing for changes in the active set is discussed next. It is well known that the `BFGS` formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{3.34}$$

will give a positive definite update $B_{k+1} \in \mathbb{R}^m \times \mathbb{R}^m$ provided that B_k is positive definite and that $s_k^T y_k > 0$. Define

$$\begin{aligned} s_k &= \alpha_k p_{Z,k} \\ y_k &= Z_k^T \left[\nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1})\right] - \bar{w}_k \end{aligned} \tag{3.35}$$

When the working-set changes from iteration k to $k+1$ the dimensions of the secant vectors s_k and y_k will not be compatible with the dimension of B_k. In `rFSQP` B_{k+1} is re-initialized to I when the working set changes. To the best of our knowledge, `BFGS` updates for reduced Hessian methods with inequality constraints are not available in the literature. (The updating scheme in `SNOPT` by Gill et al. (1997) updates the sparse but full Hessian W_k, but applies `LUSOL` by Gill, Murray, Saunders, and Wright (1987) to allow for adding and deletion of rows and columns in the working-set matrix A_k, from which Z_k is formed. Then the reduced Hessian can be produced by computing $B_k = Z_k^T W_k Z_k$.) This topic is also addressed in `MINOS` (Murtagh and Saunders 1995).

A similar scheme allows varying dimensions in the Broyden update (3.22).

In addition the `BFGS` update criterion I of Biegler et al. (1995) is implemented. I.e. choose a constant $\gamma_{fd} > 0$ and a sequence of positive numbers $\{\gamma_k^w\}$ such that $\sum_{k=1}^{\infty} \gamma_k^w < \infty$ (this is the same sequence as in (3.26)). Compute s_k and y_k by the secant relation (3.35).

- If $\bar{w}_k$ is computed by Broyden's method (3.24, 3.26), and if both $s_k^T y_k > 0$ and

$$\|p_{Y,k}\| \leq (\gamma_k^w)^2 \, \|p_{Z,k}\| \tag{3.36}$$

  hold at iteration k, then update B_k by the `BFGS` formula (3.34). Otherwise, set $B_{k+1} = B_k$.

- If $\bar{w}_k$ is computed by finite differences (3.27), and if both $s_k^T y_k > 0$ and

$$\|p_{Y,k}\| \leq \gamma_{fd} \, \|p_{Z,k}\| \, / \sigma_k^{1/2} \tag{3.37}$$

  hold at iteration k where $\sigma_k = \max\{\|e_k\|, \|e_{k+1}\|\}$, $e_k = x_k - x^*$. Then update B_k by the `BFGS` formula (3.34). Otherwise, set $B_{k+1} = B_k$.

The dependence $\sigma(x^*)$ is circumvented as in Biegler et al. (1995), noting that the `KKT` condition measure must be modified as in equation (3.44) defined in section 3.4. The main benefit of the scheme is that it allows application of the following theorem from Byrd and Nocedal (1989). Define

$$\cos\theta_k^{BFGS} = \frac{s_k^T B_k s_k}{\|s_k\| \, \|B_k s_k\|} \tag{3.38}$$

Theorem 3.1 *Byrd and Nocedal (1989): Let $\{B_k\}$ be generated by the* `BFGS` *scheme above, where for all $k \geq 1$, $s_k \neq 0$ and*

$$\begin{array}{rcl} \frac{y_k^T s_k}{s_k^T s_k} & \geq & m_1 > 0 \\ \frac{\|y_k\|^2}{y_k^T s_k} & \leq & M \end{array} \tag{3.39}$$

Then there exist constants β_1, β_2 and $\beta_3 > 0$ such that, for any $k \geq 1$, the relations

$$\cos\theta_j^{BFGS} \geq \beta_1$$

$$\beta_2 \leq \frac{\|B_j s_j\|}{\|s_j\|} \leq \beta_3$$

hold for at least $\lceil \frac{1}{2}k \rceil$ values of $j \in [1,k]$.

Since the updates are skipped for the remaining iterates; $B_{k+1} = B_k$, all matrices in $\{B_k\}$ are characterized by the theorem. This theorem applies to the present update scheme since we are able to identify the correct active set at a finite index k_0, see section 3.4.3, lemma 3.11.

Since adding rows and columns will only take place a finite number of times, the condition number will be finite at index k_0. After k_0 theorem 3.1 applies directly. In practice, finite precision numerics can impair the finite conditioning, and some testing and/or restarting procedure must be implemented. The definition of good iterates follow as in Biegler et al. (1995).

3.3.4 The rFSQP algorithm

This is based on algorithm `FSQP`' in Lawrence and Tits (2000) and algorithm `II` in Biegler et al. (1995).

Algorithm 3.1 *`rFSQP`*

i. Choose constants $\theta \in (0, \frac{1}{2})$, $\kappa \in (2,3)$, $\epsilon_l > 0$, $0 < \underline{C}^\eta < \bar{C}^\eta$, $\bar{D} > 0$, $0 < \tau < \tau' < 1$, $\Gamma > 0$, $\rho > 0$, $\gamma_{fd} > 0$ for (3.25, 3.37), $\gamma_{-1} < 0$ for (3.15), $\eta_0 > 0$ for (3.16) and $C_0^\eta \in [\underline{C}^\eta, \bar{C}^\eta]$ for (3.9). Select a summable sequence of positive numbers $\{\gamma_k^w\}$ for (3.26, 3.36). Choose a bound $\epsilon > 0$ for the active set identification.

Set $k = 0$ and select a starting point $x_0 \in X_I$, set the initial active set $J_0^\epsilon = \{i \mid -\epsilon < c_{\mathcal{I}}^i(x_0) \leq 0\}$. Choose the initial penalty parameter ν_o, a $(n-m) \times (n-m)$ symmetric and positive definite matrix B_0 and a $(n-m) \times n$ starting matrix S_0 for the Broyden approximation.

ii. *Evaluate f_0, g_0, c_0 and A_0. Compute Y_0 and Z_0 from (3.13).*

iii. *Set findiff=false. Solve for the range space step $p_{Y,0}$ from (3.15).*

iv. *Compute an approximation w_0 by Broyden's method (3.24,3.25)*

v. *Compute the damping parameter $\zeta_0 \in (0, 1]$ from (3.19) and compute the tilted null-space step $p_{Z,0}$ and γ_0 from (3.17).*

vi. *If (3.37) is satisfied, but (3.36) is not satisfied, set findiff=true and recompute w_0 by finite differences (3.27), with σ_0 replaced by $\|KKT\|$ from (3.44). Further, recompute ζ_0 from (3.19) and recompute $p_{Z,0}$ and γ_0 from (3.47).*

vii. *MAIN LOOP: Define the tilted SQP direction by $d_k = Y_k p_{Y,k} + Z_k p_{Z,k}$ from (3.14). If $d_k = 0$ STOP.*

viii. *Compute the Maratos correction term d_k^C from (3.31,3.32,3.33) if it exists and satisfies $\|d_k^C\| \leq \|d_k\|$. Otherwise set $d_k^C = 0$. Set $\alpha_k = 1$.*

ix. *Arc search. Test the constrained Armijo condition*

$$\begin{array}{l} \phi_{\nu_k}(x_k + \alpha_k d_k + \alpha_k^2 d_k^C) \leq \phi_{\nu_k}(x_k) + \theta \alpha_k D\phi_{\nu_k}(x_k, d_k) \\ s.t.\ \ c_{\mathcal{I}}^i(x_k + \alpha_k d_k + \alpha_k^2 d_k^C) \leq 0, \quad i = \{1, \ldots, p_2\} \end{array} \tag{3.40}$$

x. *If (3.40) is not satisfied, choose a new $\alpha_k \in [\tau\alpha_k, \tau'\alpha_k]$ and go to step (ix), otherwise set $x_{k+1} = x_k + \alpha_k d_k + \alpha_k^2 d_k^C$, and define $J_{k+1}^\epsilon = \{i \mid -\epsilon < c_{\mathcal{I}}^i(x_{k+1}) \leq 0\}$ from the last evaluation of $c_{\mathcal{I}}$ in (3.40).*

xi. *Evaluate f_{k+1}, g_{k+1}, c_{k+1} and A_{k+1}. Compute Y_{k+1} and Z_{k+1} from (3.13).*

xii. *Compute the Lagrange multiplier estimates from (3.18) for $i \in J_{k+1}^\epsilon$, set $\lambda_{\mathcal{I},k+1}^i = 0$ for $i \notin J_{k+1}^\epsilon$. Update the weight ν_{k+1} of the merit function from (3.20).*

xiii. *Update S_{k+1} from (3.22, 3.23). If findiff=false, calculate $\bar{w}_k$ by Broyden's method (3.24, 3.26), otherwise calculate $\bar{w}_k$ by finite differences (3.27).*

xiv. Compute s_k and y_k from (3.35). If $s_k^T y_k \leq 0$ or if [findiff=true and (3.37) is not satisfied] or if [findiff=false and (3.36) is not satisfied] set $B_{k+1} = B_k$. Otherwise compute B_{k+1} by the `BFGS` *update scheme in section 3.3.3.*

xv. Set findiff=false. Compute $p_{Y,k}^E$ from (3.29). Compute an approximation w_{k+1} by Broyden's method (3.24,3.25), and $\zeta_{k+1} \in (0,1]$ from (3.19). Compute $p_{Z,k}^E$ from (3.30).

xvi. If (3.37) is satisfied, but (3.36) is not satisfied, set findiff=true and recompute w_{k+1} by finite differences (3.27), with σ_{k+1} replaced by $\|$`KKT`$\|$ from (3.44). Further, recompute ζ_{k+1} from (3.19) and recompute $p_{Z,k}^E$ from (3.30).

xvii. Select $C_{k+1}^\eta \in \left[\underline{C}^\eta, \bar{C}^\eta\right]$.
If ($\|d_k\| < \epsilon_l$) then, compute d_{k+1}^E from (3.28). Then
if $\left\|d_{k+1}^E\right\| \leq \bar{D}$ then set $\eta_{k+1} \leftarrow C_{k+1}^\eta \left\|d_{k+1}^E\right\|^2$
else set $\eta_{k+1} \leftarrow C_{k+1}^\eta \|d_k\|^2$
else set $\eta_{k+1} \leftarrow C_{k+1}^\eta \epsilon_l^2$

xviii. Compute $p_{Y,k+1}$ from (3.15). Compute $p_{Z,k+1}$ and γ_{k+1} from (3.17) with w_{k+1} and ζ_{k+1} computed in step (xv,xvi). If findiff=false and if (3.37) is satisfied, but (3.36) is not satisfied, set findiff=true and recompute w_{k+1} by finite differences (3.27), with σ_{k+1} replaced by $\|$`KKT`$\|$ from (3.44). Further, recompute ζ_{k+1} from (3.19) and recompute $p_{Z,k+1}$ and γ_{k+1} from (3.17) with the new w_{k+1}.

xix. Set $k \leftarrow k+1$ and go to step (vii)

3.4 Global and local convergence

This section is based on Lawrence and Tits (2000) and Biegler et al. (1995). The first subsection summarizes `KKT` conditions. Subsection 3.4.2 and 3.4.3 are concerned with global convergence, i.e. that the algorithm will converge from any starting point $x_0 \in X_I$, and local convergence, i.e. the convergence rate when initialized within a sufficiently small neighborhood of a local minimizer x^*. Global convergence is based on descent, while local convergence is based on Newton's method.

Previous work on inequality constrained reduced Hessian `SQP` methods include Coleman and Conn (1982b), (1982a), and Fletcher and co-workers,

see Fletcher (1987), p. 317. In particular Byrd and Nocedal (1991) and Biegler et al. (1995) only consider equality constraints, while SNOPT by Gill et al. (1997) includes nonlinear inequality constraints. The convergence properties of SNOPT are similar to the properties of NPSOL addressed in Gill, Murray, Saunders, and Wright (1992).

The local convergence analysis in section 3.4.3 is based on Biegler et al. (1995). I.e. by proving that the optimal active set is identified, i.e. $J_k^\epsilon = J^{*,\epsilon}$ for all k sufficiently large. A complicating factor is the updating rule for the reduced Hessian $Z_k^T W_k Z_k$ with varying dimension of Z_k, see section 3.3.3. Still, our main focus remains on feasibility mechanisms and not on approximations to the reduced Hessian. Changes in the active set might be handled if we accept refactorization of the null-space matrix Z_k of the working set matrix A_k $(\neq A_{k-1})$ at each major iteration. The effort needed for this can be reduced by application of LUSOL, see Gill et al. (1997), Gill et al. (1987) for further comments. In the present algorithm the choice of the parameter ϵ in equation (3.5) has significant influence on the practical performance. Choosing a large value for ϵ causes the algorithm to stay well away from inequalities, hence refactorizations are less frequent than what could be anticipated. On the other hand, a large ϵ-value causes the algorithm to approach inequalities conservatively. Large ϵ-values therefore seem to give the algorithm a practical behavior more related to interior methods than to conventional SQP methods.

Observe that eliminating variables, by considering the equality constraints only, cannot lead to superlinear convergence, since the Hessian of the Lagrangian

$$\nabla^2 \mathcal{L}^* = \nabla^2 f(x^*) + \sum_{i=1}^{p_1} \lambda_{\mathcal{E}}^{*,i} \nabla^2 c_{\mathcal{E}}^i(x^*) + \sum_{i \in J^{*,\epsilon}} \lambda_{\mathcal{I}}^{*,i} \nabla^2 c_{\mathcal{I}}^i(x^*) \tag{3.41}$$

also depends on the active inequality constraints (note the influence of strict complementary slackness, assumption 3.4). Superlinear convergence can only be achieved if the reduced Hessian approximation scheme B_k converges to the projection of $\nabla^2 \mathcal{L}^*$ in the tangent space of active constraints at x^*.

Define the projection onto the tangent space of equality constraints and active inequality constraints by

$$\mathcal{P}(x_k) = I - A(x_k) \left(A(x_k)^T A(x_k) \right)^{-1} A(x_k)^T$$

The corresponding projection onto the range space of $A(x_k)$ is written

$$\mathcal{Q}(x_k) = I - \mathcal{P}(x_k)$$

Denote $\mathcal{P}(x_k) = \mathcal{P}_k$ and $\mathcal{Q}(x_k) = \mathcal{Q}_k$. From Boggs and Tolle (1995), theorem 3.5, two-step superlinear convergence of $x_k \to x^*$ requires

$$\lim_{k\to\infty} \frac{\left\| \mathcal{P}_k \left(W_k - \nabla^2 \mathcal{L}^* \right) \mathcal{P}_k \left(x_{k+1} - x_k \right) \right\|}{\| x_{k+1} - x_k \|} = 0$$

with superlinear convergence occurring for tangential convergence, i.e. if

$$\lim_{k\to\infty} \frac{\mathcal{Q}_k \left(x_{k+1} - x_k \right)}{\| x_{k+1} - x_k \|} = 0$$

I.e. if there are active inequality constraints at x^* we will not be able to achieve superlinear two-step convergence by excluding inequality constraints in the matrix $A(x_k)$. In addition, the conditions for theorem 3.5 in Boggs and Tolle (1995) in the reduced Hessian case requires that the null-space matrices Z_k are smooth, i.e. that

$$\| Z(x_k) - Z(x^*) \| = \mathcal{O} \left(\| x_k - x^* \| \right) \tag{3.42}$$

We prove in the following that for k large enough that $J_k^\epsilon = J^{*,\epsilon}$, hence (3.42) can be achieved near the solution x^* by using the `QR`-factorization or updating the factors of this, see Nocedal and Wright (1999), p. 567, for further comments. Alternative implementations from Biegler et al. (2000) and Xie and Byrd (1999) for the decomposition (3.13) below have not been shown to satisfy (3.42). Note that close to the solution refactorization is rarely needed once the optimal active set has been identified. Hence, this may not be an important issue in practice.

3.4.1 KKT conditions

The `KKT`-conditions of the original `NLP` (3.1) are

$$\begin{aligned} g(x^*) + \sum_{i=1}^{p_1} \lambda_{\mathcal{E}}^{*,i} \nabla c_{\mathcal{E}}^i(x^*) + \sum_{i=1}^{p_2} \lambda_{\mathcal{I}}^{*,i} \nabla c_{\mathcal{I}}^i(x^*) = 0 \\ c_{\mathcal{E}}^i(x^*) = 0, \quad i = \{1, \ldots, p_1\} \\ c_{\mathcal{I}}^i(x^*) \leq 0, \quad i = \{1, \ldots, p_2\} \\ \lambda_{\mathcal{I}}^{*,i} c_{\mathcal{I}}^i(x^*) = 0 \;\; and \;\; \lambda_{\mathcal{I}}^{*,i} \geq 0, \;\; i = \{1, \ldots, p_2\} \end{aligned} \tag{3.43}$$

A measure of the error $\sigma_k = \max\{\|e_k\|, \|e_{k+1}\|\}$, $e_k = x_k - x^*$ used in equation (3.37) must be specified without knowledge of x^* to be practical. Denote the pseudo-inverse of Z^T as $Z^{+,T}$, then $Z^{+,T} Z^T = I$. The first line of the `KKT` conditions (3.43) is restated as

$$Z_*^{+,T} Z_*^T (g_* + A_* \lambda_*) = Z_*^{+,T} Z_*^T g_* = 0$$

implying $Z_*^T g_* = 0$. Then a distance measure of (3.43) can be stated as

$$\|KKT\| \triangleq \left\| Z_k^T g_k \right\| + \|c_{\mathcal{E},k}\| + \|\lambda_{\mathcal{I},k} c_{\mathcal{I},k}\| + \|\max\{0, c_{\mathcal{I},k}\}\| + \|\min\{0, \lambda_{\mathcal{I},k}\}\| \tag{3.44}$$

We need to establish the relationship between the SQP direction d_k^0 from (3.3) and the feasible direction $d_k = Y p_{Y,k} + Z p_{Z,k}$ from (3.15, 3.47), when implemented as a reduced gradient method. It is well known that when W_k is positive definite and $X \neq \emptyset$ that d_k^0 (unique due to convexity) is a KKT point for (3.3) *iff* $\exists(\lambda_{\mathcal{E},k}^0, \lambda_{\mathcal{I},k}^0)$ such that

$$\begin{aligned} W_k d_k^0 + g_k + \sum_{i=1}^{p_1} \lambda_{\mathcal{E},k}^{0,i} \nabla c_{\mathcal{E}}^i(x_k) + \sum_{i=1}^{p_2} \lambda_{\mathcal{I},k}^{0,i} \nabla c_{\mathcal{I}}^i(x_k) = 0 \\ c_{\mathcal{E}}^i(x_k) + \left\langle \nabla c_{\mathcal{E},k}^i, d_k^0 \right\rangle = 0, \quad i = \{1, \ldots, p_1\} \\ c_{\mathcal{I}}^i(x_k) + \left\langle \nabla c_{\mathcal{I},k}^i, d_k^0 \right\rangle \leq 0, \quad i = \{1, \ldots, p_2\} \\ \lambda_{\mathcal{I},k}^{0,i} \left(c_{\mathcal{I}}^i(x_k) + \left\langle \nabla c_{\mathcal{I},k}^i, d_k^0 \right\rangle \right) = 0 \;\; and \;\; \lambda_{\mathcal{I},k}^{0,i} \geq 0, \;\; i = \{1, \ldots, p_2\} \end{aligned} \tag{3.45}$$

The KKT-conditions for the reduced gradient problem in Biegler et al. (1995) subject to inequality constraints and treated as an EQP are

$$Z_k^T W_k Z_k p_{Z,k}^0 + \left(Z_k^T g_k + \zeta_k w_k \right) = 0 \tag{3.46}$$

The $p_{Z,k}$ and γ_k relations in (3.17) are motivated by considering the following reduced qp

$$\begin{array}{cc} \min\limits_{p_{Z,k}, \gamma_k} & \frac{1}{2} \langle p_{Z,k}, B_k p_{Z,k} \rangle + \gamma_k \\ s.t. & \left\langle Z_k^T g_k + \zeta_k w_k, p_{Z,k} \right\rangle \leq \gamma_k \end{array} \tag{3.47}$$

The KKT-conditions for this are

$$\begin{aligned} Z_k^T W_k Z_k p_{Z,k} + \mu_k \left(Z_k^T g_k + \zeta_k w_k \right) = 0 \\ \mu_k = 1 \\ \left\langle Z_k^T g_k + \zeta_k w_k, p_{Z,k} \right\rangle \leq \gamma_k \\ \mu_k \left(\left\langle Z_k^T g_k + \zeta_k w_k, p_{Z,k} \right\rangle - \gamma_k \right) = 0 \;\; and \;\; \mu_k \geq 0 \end{aligned} \tag{3.48}$$

These conditions are always consistent since $(p_{Z,k}, \gamma_k) = (0, 0)$ satisfies the constraints, and always gives a unique solution due to convexity, see lemma 3.1 below. This guarantees the existence of the associated qp-multiplier μ_k. Assuming $Z_k^T W_k Z_k$ positive definite, bounded μ_k, and assumption 3.2, $p_{Z,k}$ and γ_k can be computed from (3.17) directly with $Z_k^T W_k Z_k = B_k$.

3.4.2 Global convergence

It is established in the following that a sequence $\{x_k\}$ generated by the algorithm has accumulation points that are KKT points of the original NLP. We start with some modifications of some of the lemmas in Lawrence and Tits (2000) to incorporate the equality constraints, range-null space decomposition and reduced Hessian. These lemmas state that solving the problem (3.15, 3.47) resemble solving the original NLP with $x_k \in X_I \; \forall k$ and $x^* \in X$. Note that most of the results in the following represent minor modifications to the results in Lawrence and Tits (2000), but that the multipliers are handled differently since we use the estimate (3.18) and not the qp-multipliers. We maintain the same structure as in Lawrence and Tits (2000) to facilitate comparison.

Assume in the sequel a positive definite and bounded reduced Hessian approximation $B_k = Z_k^T W_k Z_k$;

Assumption 3.5 $K_1 \left\| p_{Z,k} \right\|^2 \leq \langle p_{Z,k}, B_k p_{Z,k} \rangle \leq K_2 \left\| p_{Z,k} \right\|^2, \; \forall p_{Z,k} \in \mathbb{R}^{(n-m_k)}$, *$K_1$ and K_2 strictly positive.*

This assumption is introduced for ease of exposition, since it is satisfied for the BFGS update rule we use, see discussion in section 3.3.3 in conjunction with theorem 5.6 in Biegler et al. (1995).

Assumption 3.6 *Assume that there exists positive constants γ_a and β_0 such that*

$$\left\| Y(x_k) \left(A^T(x_k) Y(x_k) \right)^{-1} \right\| \leq \gamma_a, \quad \left\| Z(x_k) \right\| \leq \beta_0 \tag{3.49}$$

for all $x_k \in X_I$. This implies $\left\| Y_k p_{Y,k} \right\| \leq \gamma_a \left\| c_k \right\|$. Assume also that (3.39) holds for all $k \geq 1$ when B_k is updated, and that there exist a constant $\kappa_c > 0$ such that for all k

$$\left\| w_k \right\| \leq \kappa_c \left\| c_k \right\|^{1/2} \tag{3.50}$$

This is satisfied for both finite differences and Broyden updating, see Biegler et al. (1995), section 4, for a discussion. Note that assumption 3.6 is related to assumption 3.3 (LICQ) since $A_k^T Y_k$ must have linearly independent columns to be invertible. The implementation of rFSQP handles this by using a feature of the LU-decomposition routine MA28AD. I.e. linaerly dependent rows are included into the null-space which guarantees full rank of $A_k^T Y_k$.

Lemma 3.1 *Suppose assumptions 3.1-3.6 hold. Then, since B_k is symmetric positive definite, if $x_k \in X_I$, then $p_{Z,k}$ is well-defined and $(p_{Z,k}, \gamma_k)$ is the unique KKT point of (3.47). Furthermore, $p_{Z,k}$ is bounded over compact subsets of $X_I \times \mathcal{P} \times \mathbb{R}^+$, where $\mathcal{P}$ is the set of symmetric positive definite $n - m_k \times n - m_k$ matrices and $\mathbb{R}^+$ the set of nonnegative reals. In addition $d_k = Y_k p_{Y,k} + Z_k p_{Z,k}$ exists and is unique and bounded over $X_I \times \mathcal{R} \times \mathcal{P} \times \mathbb{R}^+$ where $\mathcal{R}$ is the set of nonsingular $m_k \times m_k$ matrices defined by $A_k^T Y_k$ and the decomposition of A_k in (3.13).*

Proof*:* Observe that $(p_{Z,k}, \gamma_k) = (0, 0)$ always satisfies the inequality constraint of (3.47). Boundedness of $p_{Z,k}$ and γ_k is a consequence of the Frank-Wolfe theorem (Frank and Wolfe 1956), (Perold 1980) and assumption 5: "Every quadratic program whose objective function is bounded below on the feasible set has a finite global minimizer." Uniqueness follows from assumption 3.5 (convexity). I.e. $(p_{Z,k}, \gamma_k)$ is well-defined. By theorem 6 in Fiacco and McCormick (1990) the solution is continous. The multiplier μ_k is obviously bounded. Due to non-singularity of $(A_k^T Y_k)$ the range-space step $p_{Y,k}$ is well-defined, unique, continous and bounded for all bounded c_k which are bounded on the compact the set X_I by assumption 3.2. Then $d_k = Y_k p_{Y,k} + Z_k p_{Z,k}$ exists and is continous, unique and bounded as well. □

This extends lemma 1 of Lawrence and Tits (2000) to include equality constraints, partitioning of d_k in (3.14) and the reduced Hessian B_k and establishes existence, uniqueness, continuity and boundedness of $d_k = Y_k p_{Y,k} + Z_k p_{Z,k}$. The lemma characterizes the solution of the problem (3.47), which is utilized to characterize the solution of (3.17) in lemma 3.2.

Lemma 3.2 *Suppose assumptions 3.1-3.6 hold. Then, given B_k symmetric positive definite and $\eta_k \geq 0$*

i) $\gamma_k \leq 0 \; \forall x_k \in X_I$. *Moreover,* $\gamma_k = 0$ *iff* $p_{Z,k} = 0$

ii) $d_k = Y_k p_{Y,k} + Z_k p_{Z,k} = 0$ *iff* x_k *is a* **KKT** *point for the* **NLP** *(3.1), where γ_k and $p_{Z,k}$ are defined by equation (3.47), $p_{Y,k}$ is defined by (3.15) and d_k by (3.14).*

Proof: To prove *i)* observe that since $(p_{Z,k}, \gamma_k) = (0, 0)$ is always feasible for (3.47) the optimal value is non-positive. Since B_k is positive definite we have $\langle p_{Z,k}, B_k p_{Z,k} \rangle \geq 0$, i.e. $\gamma_k \leq 0$. If $p_{Z,k} = 0$ the inequality of (3.47) gives $\gamma_k = 0$ by squeezing. Finally suppose $\gamma_k = 0$, then for $x_k \in X_I$, $B_k \succ 0$ we observe that since $p_{Z,k} = 0$ is feasible it is also optimal due to the

non-negativeness of the objective. Convexity gives uniqueness of $p_{Z,k} = 0$. To prove *ii*) suppose that $d_k = 0$, from (3.14) this solution is satisfied by choosing $Y_k p_{Y,k} = -Z_k p_{Z,k}$. In detail, (3.13) gives

$$\begin{bmatrix} I \\ 0 \end{bmatrix} p_{Y,k} = - \begin{bmatrix} -C_k^{-1} N_k \\ I \end{bmatrix} p_{Z,k}$$

The last $n - m_k$ rows implies $p_{Z,k} = 0$, which inserted in the first m_k rows implies $p_{Y,k} = 0$. We have

$$d_k = 0 \Longleftrightarrow (p_{Y,k}, p_{Z,k}) = (0,0) \tag{3.51}$$

If A_k is factorized by `QR`-factorization the relation (3.51) follows from the direct sum $span(A_k) = \mathcal{R}(A_k^T) \oplus \mathcal{N}(A_k)$. When $(p_{Y,k}, p_{Z,k}, \gamma_k) = (0,0,0)$ then by (3.48) there exist a scalar multiplier $\mu_k \geq 0$ such that

$$\begin{gathered} \mu_k \left(Z_k^T g_k + \zeta_k w_k \right) = 0 \\ \mu_k = 1 \end{gathered} \tag{3.52}$$

Thus, since by the selection in step (x) of algorithm `rFSQP` $c_{\mathcal{I}}^i(x_k) = 0$ $\forall i \in J_k^\epsilon$, the last line of the optimality conditions (3.52) become $\lambda_{\mathcal{I},k}^T c_{\mathcal{I}}(x_k) = 0$ since λ_k's estimated by (3.18) are bounded, and we choose $\lambda_{\mathcal{I},k}^i = 0$, $\forall i \notin J_k^\epsilon$. Also $p_{Y,k} = 0$ implies $w_k = 0$ if either Broyden's method or finite differences is used. The first line of (3.52) then becomes $Z_k^T g_k = 0$. We have $\left\| Z_k^T g_k \right\| = \left\| Z_k^T (g_k + A_k \lambda_k) \right\| = 0$. Hence, the `KKT`-conditions (3.43) are satisfied by following the derivation of (3.44). To prove the converse part of *ii*) note that if x_k is a `KKT` point for the `NLP`, then the `NLP`'s `KKT`-conditions (3.43) and (3.48) show that $(p_{Y,k}, p_{Z,k}, \gamma_k) = (0,0,0)$ is a `KKT` point of the reduced gradient problem. Assumption 3.3 implies uniqueness of λ^*. □

This adapts lemma 2 of Lawrence and Tits (2000) to our setting. The lemma is concerned with problem (3.47). We observe that the `KKT`-conditions (3.48) imply (3.17), hence statements about the solution to (3.47), are equally valid for $p_{Z,k}$ and γ_k computed by (3.17).

The next two lemmas establish that the line search is well defined.

Lemma 3.3 *Suppose assumptions 3.1-3.6 hold. Suppose $x_k \in X_I$ is not a `KKT` point for the `NLP`, B_k is symmetric positive definite and $\eta_k > 0$. Then*

i) $D\phi_\nu(x_k, d_k) < 0$, *and*

ii) $\left\langle \nabla c_{\mathcal{I}}^i(x_k), d_k \right\rangle < 0$, *for all* $i \in J_k^\epsilon$

iii) $\phi_\nu(x_j) - \phi_\nu(x_{j+1}) \geq \gamma_\nu \left[\left\| Z_j^T g_j \right\|^2 + \|c_j\|_1 \right]$ *for the good iterates* j *and constant* ν_j *in (3.11).*

This lemma is modeled after section 3.4 and lemma 4.1 of Biegler et al. (1995), and replaces lemma 3 in Lawrence and Tits (2000). The proof is almost identical to the proof in Biegler et al. (1995), and is given in appendix B for completeness. Item *iii)* guarantees the existence of a non-zero steplength for non-stationary points. The use of Taylor's theorem for the continous non-differentiable merit function locally is justified by observing that existence of the directional derivative in a point implies existence in a small "neighbourhood" in the relevant direction of the point by continuity. This is also assumed implicitly in Biegler et al. (1995). A related alternative to this is to apply the mean-value theorem locally. The lemma allows direct application of theorem 4.2 in Biegler et al. (1995).

From the proof in appendix B we get the usual update rule for the penalty parameter ν_k

$$\nu_k = \begin{cases} \nu_{k-1} & \text{if } \nu_{k-1} \geq \|\lambda_k\|_\infty + 2\rho \\ \|\lambda_k\|_\infty + 3\rho & \text{otherwise} \end{cases} \tag{3.53}$$

The update-rule for ζ_k follows by choosing $\zeta_k = \min\left\{1, \hat{\zeta}_k\right\}$ where $\hat{\zeta}_k$ is defined by solving equation (B.4) with equality.

Lemma 3.4 *Suppose assumptions 3.1-3.6 hold. Then, if* $\eta_k = 0$, x_k *is a* `KKT` *point for the* `NLP` *(3.1) and the algorithm will stop in step (vii) at iteration* k*. On the other hand, whenever the algorithm does not stop in step (vii), the line search is well defined, i.e. step (ix) is satisfied for a step* $\alpha_{k_j} > 0$ *for some finite* k_j*.*

Proof: Suppose that $\eta_k = 0$. Then $k > 0$ and, by step (xvii), either $d_k^E = 0$, or $d_{k-1} = 0$. The latter cannot hold due to step (vii). If $d_k^E = 0$ equation (3.30) with $(p_{Y,k}^E, p_{Z,k}^E) = (0,0)$ implies $\left\| Z_{k+1}^T g_{k+1} \right\| = 0$. Lemma 3.3 *ii)* gives feasibility of $c_{\mathcal{I}}^i(x_k)$. For all $i \in J_k^\epsilon$ we have $c_{\mathcal{I}}^i(x_k) = 0$, and for $i \notin J_k^\epsilon$ we are free to choose $\lambda_k^{E,i} = 0$, hence $\lambda_k^E c_{\mathcal{I}}(x_k) = 0$. We have $c_{\mathcal{E}}^i(x_k) = 0$ from (3.29) whenever $p_{Y,k}^E = 0$. Since the `KKT` measure (3.44) is zero we have that x_k is a `KKT` point for the `NLP` with multipliers $\lambda^{*,i}$ from (3.18) for $i \in J_{k-1}^\epsilon$ and $\lambda^{*,i} = 0$ for $i \notin J_{k-1}^\epsilon$. By lemma 3.2 item *ii)* $d_{k+1} = 0$ and the algorithm will stop in step (vii). The first claim is proved. Also, we have established that $\eta_k > 0$ whenever step (ix) is reached. Lemma 3.3 (descent and feasibility) and assumption 3.2 (continuity) gives the second claim. □

This is equivalent to lemma 4 in Lawrence and Tits (2000). Lemma 3.3 and 3.4 show that the algorithm is well defined. Lemma 3.2 shows that if the algorithm generates a finite sequence terminating at a point x_N, then x_N is a KKT point for the NLP (3.1). We now turn to consider infinite sequences $\{x_k\}$, i.e. that $\eta_k > 0$ for all k by lemma 3.4.

Lemma 3.5 *Suppose assumptions 3.1-3.6 hold. Then the sequence* $\{\eta_k\}$ *generated by algorithm* **rFSQP** *is bounded. Further, the sequence* $\{d_k\}$ *is bounded on subsequences on which* $\{x_k\}$ *is bounded.*

Proof: The first claim follows from step (xvii) of algorithm rFSQP. When η_k is bounded, so is d_k by lemma 3.1 under assumption 3.5 on compact subsets $X_I \times \mathcal{R} \times \mathcal{P} \times \mathbb{R}^+$. □

This is lemma 5 of Lawrence and Tits (2000), and neither the lemma nor the proof is affected by the reduced gradient approach. We restate proof since it is very short.

Lemma 3.6 *Suppose assumptions 3.1-3.6 hold. Suppose* $\mathcal{K}$ *is an infinite index set such that* $x_k \overset{k\in\mathcal{K}}{\to} x^* \in X$, $\{\eta_k\}$ *is bounded on* $\mathcal{K}$, *and* $d_k \overset{k\in\mathcal{K}}{\to} 0$. *Then* $J_k^\epsilon \subseteq J^{*,\epsilon}$, *for all* $k \in \mathcal{K}$, k *sufficiently large and the multiplier sequence* $\{\lambda_k\}$ *is bounded on* $\mathcal{K}$. *Further, given any accumulation point* $\eta^* \geq 0$ *of* $\{\eta_k\}_{k\in\mathcal{K}}$, $(0,0)$ *is the unique solution of (3.17) when* $p_{Y,k} = 0$.

Proof: From (3.51) $p_{Y,k} \overset{k\in\mathcal{K}}{\to} 0$, $p_{Z,k} \overset{k\in\mathcal{K}}{\to} 0$. We have $\gamma_k \overset{k\in\mathcal{K}}{\to} 0$ by squeezing in (3.17). Arguing as in Lawrence and Tits (2000) we have for an $i' \notin J^{*,\epsilon}$ and a $\delta_{i'} > 0$ that

$$c_{\mathcal{I}}^{i'}(x_k) + \left\langle Y_k^T \nabla c_{\mathcal{I},k}^{i'}, p_{Y,k} \right\rangle - \gamma_{k-1}\eta_k \leq -\frac{\delta_{i'}}{2} < 0$$

I.e. $i' \notin J_k^\epsilon$ for all $k \in \mathcal{K}$ and k sufficiently large, hence $J_k^\epsilon \subseteq J^{*,\epsilon}$ is proved. Boundedness of $\{\lambda_k\}$ follows from (3.18). The third claim follows by taking limits in (3.17) with $p_{Y,k} = 0$ proving that $(0,0)$ is the unique solution. □

This is equivalent to lemma 6 of Lawrence and Tits (2000).

Lemma 3.7 *Suppose assumptions 3.1-3.6 hold. Then, if* $\mathcal{K}$ *is an infinite index set such that* $d_k \overset{k\in\mathcal{K}}{\to} 0$, *all accumulation points of* $\{x_k\}_{k\in\mathcal{K}}$ *are* **KKT** *points for the* **NLP** *(3.1).*

Proof: Suppose $\mathcal{K}' \subseteq \mathcal{K}$ is an infinite index set on which $x_k \overset{k\in\mathcal{K}'}{\to} x^* \in X$. In view of assumption 3.5 and lemma 3.5, assume, without loss of generality

that $B_k \overset{k\in\mathcal{K}'}{\to} Z^{*,T}W^*Z^*$, a positive definite matrix, and $\eta_k \overset{k\in\mathcal{K}'}{\to} \eta^* \geq 0$. In view of lemma 3.6, $(p_{Y,k}, p_{Z,k}, \gamma_k) = (0,0,0)$ is the unique solution to (3.15, 3.47), which is a KKT point satisfying $\|\mathtt{KKT}\| = 0$ in (3.44) by lemma 3.2. □

This is equivalent to lemma 7 of Lawrence and Tits (2000). The proof is unaltered and follows from lemma 3.2 and 3.6 herein. Note that requiring $B_k \overset{k\in\mathcal{K}'}{\to} Z^{*,T}W^*Z^*$ is less strict than requiring $W_k \overset{k\in\mathcal{K}'}{\to} W^*$.

Theorem 3.2 *Under assumptions 3.1-3.6, algorithm* ***rFSQP*** *generates a sequence* $\{x_k\}$ *for which all accumulation points are* ***KKT*** *points for the* ***NLP*** *(3.1).*

Proof: Suppose $\mathcal{K}$ is an infinite index set on which $x_k \overset{k\in\mathcal{K}}{\to} x^*$. In view of lemma 3.5 and assumption 3.5 we may assume without loss of generality that $d_k \overset{k\in\mathcal{K}}{\to} 0$, $\eta_k \overset{k\in\mathcal{K}}{\to} \eta^* \geq 0$ and $B_k \overset{k\in\mathcal{K}}{\to} Z^{*,T}W^*Z^*$. If $\eta^* = 0$ step (xvii) of algorithm rFSQP implies the existence of an infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ such that either $d_k^E \overset{k\in\mathcal{K}'}{\to} 0$ for all $k \in \mathcal{K}'$, or $d_{k-1} \overset{k\in\mathcal{K}'}{\to} 0$. If $d_{k-1} \overset{k\in\mathcal{K}'}{\to} 0$, $x_{k-1} \overset{k\in\mathcal{K}'}{\to} x^*$, since $|x_k - x_{k-1}| \leq 2\,|d_{k-1}| \overset{k\in\mathcal{K}'}{\to} 0$. Lemma 3.7 implies that x^* is a KKT point. When $d_k^E \overset{k\in\mathcal{K}'}{\to} 0$ from step (xvii) and $J_k^\epsilon \subseteq J^{*,\epsilon}$ for all $k \in \mathcal{K}'$ from lemma 3.6. Boundedness of $\{\lambda_k^E\}$ follows since we are free to choose them as the estimates (3.18). Taking limits in (3.29, 3.30) we conclude from $\left\| Z_k^T g_k \right\| \overset{k\in\mathcal{K}'}{\to} 0$, $\|c_k\|_1 \overset{k\in\mathcal{K}'}{\to} 0$ and (3.44), that x^* is a KKT point for the NLP.

For $\eta^* > 0$ lemma 3.3 applies and the claim follows from theorem 4.2 in Biegler et al. (1995). I.e. bounded $\{\|\lambda_k\|\}_{k\in\mathcal{K}}$ implies that there exist a k_0 such that ν_k is constant for $k > k_0$ by the update rule (3.53). Consider $k > k_0$ and using lemma 3.3 *iii*)

$$\begin{aligned}\phi_\nu(x_{k_0}) - \phi_\nu(x_{k+1}) &= \sum_{j=k_0}^{k} \left(\phi_\nu(x_j) - \phi_\nu(x_{j+1})\right) \\ &\geq \gamma_\nu \sum_{j\in\mathcal{J}\cap[k_0,k]} \left[\left\| Z_j^T g_j \right\|^2 + \|c_j\|_1\right]\end{aligned}$$

where $\mathcal{J}$ is the set of good iterates as defined by Biegler et al. (1995). By descent from lemma 3.3 *i*) and assumption 3.2 ϕ_ν is bounded below and the sum is finite. The multiplier estimate (3.18) at x^* and assumption 3.3 implies $\|\mathtt{KKT}\| = 0$ in (3.44). □

This is equivalent to theorem 1 of Lawrence and Tits (2000), and the first part concerning the case $\eta^* = 0$ is restated for the sake of completeness.

3.4.3 Local convergence

The local convergence section in Lawrence and Tits (2000) is mainly concerned with restating global convergence under stronger 2nd order conditions, and proving that the FSQP' direction $d_k + d_k^C$ converges to the SQP direction d_k^0. Once this is established a step-length of 1 is accepted locally and 2-step superlinear convergence follows by a standard result. On the other hand, the local convergence analysis of Biegler et al. (1995) is mainly concerned with establishing 1-step superlinear convergence. This is possible due to the approximation terms w_k and $\bar{w}_k$, and a detailed order analysis. Note that the Maratos correction term d_k^C in FSQP' does not handle the Maratos effect for the l_1-merit function used in the present algorithm. As in Biegler et al. (1995) we implement a standard watch-dog strategy to handle the Maratos effect, and prove in proposition 3.1 that the constraint in (3.40) does not prevent a step-size of one to be accepted for a large enough k.

The regularity assumptions are strengthened;

Assumption 3.7 *The functions $f, c_{\mathcal{E}}^i, c_{\mathcal{I}}^i : \mathbb{R}^{\aleph} \to \mathbb{R}$, for all i are three times continuously differentiable on X_I, and their Hessians are Lipschitz continuous in a neighborhood of x^*.*

Definition 3.1 *A point x^* satisfies second order sufficiency conditions with strict complementary slackness for the NLP (3.1) if there exist a multiplier vector $\lambda^* \in \mathbb{R}^{\aleph+\aleph}$ such that;*

- *The pair (x^*, λ^*) satisfies (3.43), i.e. x^* is a KKT point for the NLP (3.1).*
- $\lambda^{*,i} > 0 \ \forall i \in J^{*,\epsilon}$
- $Z_*^T \nabla_{xx}^2 \mathcal{L}\ (x^*, \lambda^*)\, Z_*$ *is positive definite on the subspace*

$$\left\{\nabla c_{\mathcal{E}}^i(x^*) \mid \forall i \in \{1, \ldots, p_1\}\right\} \cup \left\{\nabla c_{\mathcal{I}}^i(x) \mid i \in J^{*,\epsilon}\right\}$$

In section 3.4.2, we proved that every accumulation point of $\{x_k\}$ is a KKT point of the NLP (3.1). We define the reduced Hessian of the Lagrangian function

$$G_k = Z_k^T \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k) Z_k$$

To prove that the sequence $\{x_k\}$ converges to a KKT point x^*, we need the following assumptions.

Assumption 3.8 *The sequence $\{x_k\}$ has an accumulation point x^* which satisfies the second order sufficiency conditions with strict complementary slackness.*

Assumption 3.9 *Biegler et al. (1995) (assumption 5.1): The point x^* is a local minimizer for the* `NLP` *(3.1), at which the following conditions hold:*
i) For all $q \in R^{n-m}$, $q \neq 0$, we have $q^T G^* q > 0$.
ii) There exists constants γ_a, β_0 and γ_c such that, for all x_k in a neighborhood of x^*,

$$\left\| Y(x_k) \left[A(x_k)^T Y(x_k) \right]^{-1} \right\| \leq \gamma_a$$
$$\| Z(x_k) \| \leq \beta_0$$
$$\left\| [Y(x_k)\ Z(x_k)]^{-1} \right\| \leq \gamma_c$$

(see assumption 3.6 as well.)
iii) $Z(x_k)$ and $\lambda(x_k)$ are Lipschitz continuous in a neighborhood of x^*, i.e. there exist constants γ_λ and γ_Z such that

$$\begin{aligned} \| \lambda(x) - \lambda(z) \| &\leq \gamma_\lambda \| x - z \| \\ \| Z(x) - Z(z) \| &\leq \gamma_Z \| x - z \| \end{aligned}$$

for all x, z near x^*.

Item numbers *i*)-*iii*) follow item numbers *iii*)-*v*) in Biegler et al. (1995). Assumption 3.9 *v*) requires that we are able to identify the correct active set $J^{*,\epsilon}$ for a finite k_0. Then, for a $k_1 \geq k_0$, for all $k \geq k_1$ the assumption requires that any possible refactorization of $A(x_k)$ will result in a continuous $Z(x_k)$. This assumption can be satisfied if no further factorizations are needed for all $k \geq k_1$. This holds locally by continuity, i.e. a finite condition number of $A(x^*)$ will give a finite condition number of $A_k^T Y_k$ in a neighborhood of x^*. Then there is no need to refactorize $A(x_k)$. If the algorithm is implemented with refactorization at each iteration, the assumption is violated unless `QR`-factorizations are employed.

Assumption 3.10 *Biegler et al. (1995) (assumption 5.2): The line search has the property that, for all large k, $\phi_\nu((1-\vartheta)x_k - \vartheta x_{k+1}) \leq \phi_\nu(x_k)$ for all $\vartheta \in [0,1]$. In other words, x_{k+1} is in the connected component of the level set $\{x \mid \phi_\nu(x) \leq \phi_\nu(x_k)\}$ that contains x_k.*

As in Biegler et al. (1995) assumption 3.10 is only likely to hold locally with practical line search methods. The presence of inequalities does not influence on assumption 3.10 since we are always feasible with respect to them in algorithm `rFSQP`. Thus assumption 3.10 applies for all $k \geq k_1$, and we can assume that lemma 4.1 and 4.2 of Byrd and Nocedal (1991) (lemma 4.1 in Xie and Byrd (1999)) hold in the level sets defined by assumption 3.10.

Lemma 3.8 *Suppose assumptions 3.1 and 3.3-3.10 hold. Then the entire sequence generated by algorithm* `rFSQP` *converges to a point* x^* *satisfying definition 3.1.*

Proof: Note that points x^* satisfying definition 3.1 are isolated by theorem 2.1 in Robinson (1974) under assumption 3.3. I.e. there exists a ball $\mathcal{B}(x^*, \varepsilon)$ containing x^* in its interior as the only `KKT` point. From assumption 3.8 the sequence has such an accumulation point x^*. By descent from lemma 3.3 i) and assumption 3.7 ϕ_ν is bounded below, then we have that the sequence $\{\phi_\nu\}_k$ is bounded below. By the Armijo condition (3.40) we have that $\{\alpha_k D\phi_{\nu_k}(x_k, d_k)\} \to 0$, giving

$$\{\alpha_k \langle g_k, d_k\rangle - \alpha_k \nu_k \|c_k\|_1\} \to 0$$

where $\nu_k > 0$ by (3.53). If $g_k = 0$ and $c_k = 0$, we are at a `KKT` point according to (3.44) and there is nothing to prove. Assume $g_k \neq 0$, then since $c_k \neq 0$ implies $d_k \neq 0$ by (3.15), it suffices to consider the first part, i.e. we have $\{\alpha_k \langle g_k, d_k\rangle\} \to 0$, implying $\{\alpha_k d_k\} \to 0$ since $g_k \neq 0$. We have

$$\begin{aligned} |x_{k+1} - x_k| = & \left|\alpha_k d_k + \alpha_k^2 d_k^C\right| \\ \leq & \ 2\alpha_k |d_k| \end{aligned}$$

by step (viii) and $0 < \alpha_k \leq 1$. Obviously $|x_{k+1} - x_k| \to 0$ for large enough k. I.e. $\|x_{k+1} - x_k\| = \|x_{k+1} - x^* + x^* - x_k\| \leq \|x_{k+1} - x^*\| + \|x_k - x^*\| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$, and we cannot leave $\mathcal{B}(x^*, \varepsilon)$ without creating another clusterpoint and hence another `KKT` point in $\mathcal{B}(x^*, \varepsilon)$. □

This is equivalent to lemma 8 of Lawrence and Tits (2000). Observe that the lemma is concerned with the case where the algorithm `rFSQP` never stops in step (vii). The proof follows Panier and Tits (1987), (1993).

The next lemmas guarantee that the sequence $\left\{d_k + d_k^C\right\}$ approach the sequence $\left\{d_k^0\right\}$ of ordinary `SQP` directions sufficiently fast, where $d_k^0 = \left(p_{Y,k}^0, p_{Z,k}^0\right)$ is the solution to the unperturbed problem (3.46) with $p_{Z,k}^0 = p_{Z,k}$.

Lemma 3.9 *Suppose assumptions 3.1 and 3.3-3.10 hold. Assume that the sequence* $\{x_k\}$ *converges to a* `KKT` *point* x^*. *Then* $d_k^0 \to 0$ *and* $\lambda_k^0 \to \lambda^*$.

Proof: At x^* we have from (3.43) that $Z_*^T g_* = 0$, $c_{\mathcal{I}}^i(x^*) \leq 0$, $\lambda_{\mathcal{I}}^{*,i} c_{\mathcal{I}}^i(x^*) = 0$ and $\lambda_{\mathcal{I}}^{*,i} \geq 0$, $i = \{1, \ldots, p_2\}$. Since $c_{\mathcal{E}}(x^*) = 0$ and $c_{\mathcal{I}}^i(x^*) = 0$ $\forall i \in J^{*,\epsilon}$ continuity and (3.15) with $c_k \triangleq \left[\begin{array}{cc} c_{\mathcal{E}}^T(x_k) & \tilde{c}_{\mathcal{I}}^T(x_k) \end{array}\right]^T$ and $\tilde{c}_{\mathcal{I}}(x_k) \triangleq \left\{c_{\mathcal{I}}^i(x_k) \mid i \in J_k^\epsilon\right\}$, imply $p_{Y,k}^0 \to 0$ implying $w_k \to 0$. Assumptions 3.2, 3.5

and 3.6 gives strict convexity and boundedness of (3.46), implying $p^0_{Z,k} \to 0$ by (3.44). Evidently all multipliers estimated by (3.18) will converge since $(Y_k^T A_k)^{-1} Y_k^T$ is a left-inverse of A_k, and $g^* = -A^*\lambda^*$ by (3.43), then assumption 3.3 and 3.8 gives a unique λ^* satisfying strict complementary slackness. Assumption 3.2 implies that `LICQ` holds in a small neighborhood of x^*, and $\lambda^0_k \to \lambda^*$ follows. □

This is equivalent to lemma 9 of Lawrence and Tits (2000) modified to allow for the decomposition. The proof partly follows Panier and Tits (1987). If $\left\|d^E_{k+1}\right\| > 0$ and $\|d_k\| > 0$, then we know from step (xvii) that $\eta_k > 0$. If $\|d_k\| = 0$ the algorithm will stop in step (vii). Comparing (3.29, 3.30) with (3.44), we observe that $\left\|d^E_{k+1}\right\| = 0$ can only occur at `KKT` points, and by lemma 3.2, the algorithm would have stopped at step (vii). Hence, $\eta_k > 0$ can be assumed for all $x_k \neq x^*$.

Lemma 3.10 *Suppose assumptions 3.1 and 3.3-3.10 hold. Then, if $\mathcal{K}$ is a subsequence on which $\{\eta_k\}$ converges, say to $\eta^* \geq 0$, then $d_k \to 0$ and $\gamma_k \to 0$.*

Proof: We prove that $(d_k, \gamma_k) \overset{k\in\mathcal{K}}{\to} (0,0)$ by contradiction. I.e. suppose that on some infinite index set $\mathcal{K}' \subseteq \mathcal{K}$ that (d_k, γ_k) is bounded away from zero. Assumption 3.5 allows the assumption $B_k \overset{k\in\mathcal{K}'}{\to} B^*$ without loss of generality. When assumption 3.8 holds at x^* lemma 3.2 *ii*) and (3.51) imply $(p^*_Y, p^*_Z) = (0,0)$. Assumption 3.7 implies for any $x_k \overset{k\in\mathcal{K}'}{\to} x^*$ that $p_{Z,k} \overset{k\in\mathcal{K}'}{\to} 0$, $\gamma_k \overset{k\in\mathcal{K}'}{\to} 0$. Clearly $p_{Y,k} \overset{k\in\mathcal{K}'}{\to} 0$ by (3.15) since $\|c(x_k) - c(x^*)\| \overset{k\in\mathcal{K}'}{\to} 0$ by continuity, and the contradiction proves the claim.

Now suppose $p_{Z,k} \not\to 0$. Since $\{B_k\}$ and $\{\eta_k\}$ are bounded, there exists an infinite index-set $\mathcal{K}$ on which $\{B_k\}$ and $\{\eta_k\}$ converge and $p_{Z,k}$ is bounded away from zero. This contradicts $p_{Z,k} \overset{k\in\mathcal{K}'}{\to} 0$ established above, i.e. $p_{Z,k} \to 0$. Then $\gamma_k \to 0$ by squeezing in (3.47). A similar argument gives $p_{Y,k} \to 0$, hence $d_k \to 0$ by (3.14). □

This is equivalent to lemma 12 of Lawrence and Tits (2000) modified to allow for the decomposition. This lemma is included to show that assumption 3.8 does not inhibit convergence. The next lemma shows that we eventually identify the correct active set.

Lemma 3.11 *Suppose assumptions 3.1 and 3.3-3.10 hold. Then, for all k sufficiently large $J^\epsilon_k = J^{*,\epsilon}$.*

Proof: Since $\{\eta_k\}$ is bounded and lemma 3.10 implies $(d_k, \gamma_k) \to (0,0)$, lemma 3.6 implies $J^\epsilon_k \subseteq J^{*,\epsilon}$ for all k sufficiently large. Choose an $i' \in J^{*,\epsilon}$.

By assumption 3.7 we have $c^{i'}_{\mathcal{I},k} \to c^{i'}_{\mathcal{I},*}$. Hence $J^{\epsilon}_k \to J^{*,\epsilon}$ asymptotically. Then for any $\epsilon > 0$ we will have $\left| c^{i'}_{\mathcal{I},k} \right| > -\epsilon$ for all k large enough. Then, since $x_k \in X_I \ \forall k$ and i' was arbitrarily, $J^{\epsilon}_k = J^{*,\epsilon}$ for all k large enough. □

This is equivalent to lemma 13 of Lawrence and Tits (2000) modified to allow for the decomposition. In `FSQP'` the `qp`-multipliers are used to identify the correct active set as done by Powell (1978). In `rFSQP` we use ϵ-active constraints J^{ϵ}_k, and prove by continuity of c_k that asymptotically $J^{\epsilon}_k \to J^{*,\epsilon}$. Then J^{ϵ}_k will be stable for all large enough k. Note that the complementary slackness assumption was not utilized in the proof. Since there exists a finite k_0 such that $J^{\epsilon}_k = J^{*,\epsilon}$ for all $k \geq k_0$ we have an equality constrained problem. Identification of the optimal active set allows us to treat the problem as equality constrained locally (see Powell (1978)), and the local analysis will follow directly from Biegler et al. (1995) and Byrd and Nocedal (1991), provided that our tilted direction approach the ordinary `SQP` direction d^0_k sufficiently fast, and that the constrained line-search accepts a unit step-length. These provisions are proved in the following.

Identification of the correct active set in full-space methods is important since it says that the `qp` actually solves the problem in the correct tangent space. When we solve for $p_{Y,k}$ from (3.15) we always satisfy this, provided that we pick the correct active set in step (x) of algorithm `rFSQP`. Thus, a pre-assigned active set strategy depends on accurate identification of the correct active set near the solution. Choosing the ϵ-active set J^{ϵ}_{k+1} from (3.5) in step (x) of algorithm `rFSQP`, identification of the optimal active set occurs when the algorithm converges to a `KKT` point. The bound $\epsilon > 0$ can be chosen as the nonlinear inequality feasibility tolerance *featol* and is typically a small number.

Lemma 3.12 *Suppose assumptions 3.1 and 3.3-3.10 hold. For all k sufficiently large, d^E_k is uniquely defined and $d^E_k = d^0_k$.*

Proof: Since (3.29) with the associated c^E_{k+1} equals equation (12) in Biegler et al. (1995), augmented with the additional active inequalities, $p^E_{Y,k} = p^0_{Y,k}$ follows. The estimate $p^E_{Z,k}$ is defined as the unique solution of (3.30), where uniqueness follows from assumption 3.5. Then $p^E_{Z,k} = p^0_{Z,k}$ follows since (3.30) is identical to (3.46). Note that ζ^E_{k+1} and w^E_{k+1} are computed by the same relations as ζ_{k+1} and w_{k+1}, see steps (xv,xvi) in algorithm `rFSQP`. Uniqueness and $d^E_k = d^0_k$ then follow from (3.28). □

This is equivalent to lemma 15 of Lawrence and Tits (2000) modified to allow for the equality constraints. The lemma shows that d^E_k is a "good"

estimate of the SQP direction d_k^0, which is obvious since in fact it is the SQP-direction.

Lemma 3.13 *Suppose assumptions 3.1 and 3.3-3.10 hold. Then,* $\eta_k \to 0$.

Proof: This follows from step xvii of algorithm rFSQP since by lemma 3.10 and 3.12 $\{d_k\}$ and $\left\{d_k^E\right\}$ both converge to 0. □

This is lemma 16 of Lawrence and Tits (2000). The proof is unchanged. Note that the lemma establishes that η_k actually converges, a property that has been assumed so far.

We now consider the approximation error between d_k, d_k^C and d_k^0.

Lemma 3.14 *Suppose assumptions 3.1 and 3.3-3.10 hold. Then,*
i) $\eta_k = \mathcal{O}\left(\left\|d_k^0\right\|^2\right)$
ii) $p_{Y,k} = p_{Y,k}^0 + \mathcal{O}\left(\left\|d_k^0\right\|^2\right)$ *and* $p_{Z,k} = p_{Z,k}^0$
iii) $d_k = d_k^0 + \mathcal{O}\left(\left\|d_k^0\right\|^2\right)$
iv) $\gamma_k = \mathcal{O}\left(\left\|d_k^0\right\|\right)$

Proof: $\eta_k = \mathcal{O}\left(\left\|d_k^0\right\|^2\right)$ follows without modifications from Lawrence and Tits (2000), i.e. by step (xvii) of algorithm rFSQP: Lemma 3.12 gives existence and uniqueness of $d_k^E = d_k^0$. Lemma 3.10 and 3.9 ensure that step (xvii) of algorithm rFSQP chooses $\eta_k = C_k^\eta \left\|d_k^E\right\|^2$ for all k sufficiently large, thus item *i*) follows.

From equation (3.17) and (3.46) $p_{Z,k} = p_{Z,k}^0$. From (3.15) and the associated c_k we have

$$\begin{aligned}(A_k^T Y_k) p_{Y,k} = \quad -c_k &= -\begin{bmatrix} c_{\mathcal{E},k} \\ c_{\mathcal{I},k} \end{bmatrix} - \begin{bmatrix} 0 \\ \gamma_{k-1}\eta_k e_{\left|J_k^\epsilon\right|} \end{bmatrix} \\ &= -c_k^0 - \begin{bmatrix} 0 \\ \gamma_{k-1}\eta_k e_{\left|J_k^\epsilon\right|} \end{bmatrix}\end{aligned}$$

where $e_{\left|J_k^\epsilon\right|}$ is a vector of ones of length $\left|J_k^\epsilon\right|$. We have $p_{Y,k} = p_{Y,k}^0 + \mathcal{O}(\left\|d_k^0\right\|^2)$, since $\gamma_{k-1} < 0$ away from KKT points by lemma 3.2. By item *ii*) $d_k = Y_k p_{Y,k}^0 + Z_k p_{Z,k}^0 + \mathcal{O}(\left\|d_k^0\right\|^2) = d_k^0 + \mathcal{O}(\left\|d_k^0\right\|^2)$ for all k sufficiently large. From inequality (3.47) we have $\left\langle Z_k^T g_k + \zeta_k w_k, p_{Z,k}\right\rangle \leq \gamma_k \leq 0$. By lemma 5.8 in Biegler et al. (1995), $w_k = \mathcal{O}(p_{Y,k})$, $\gamma_k \leq 0$, assumption 3.2, and using

$c \left\| d_k^0 \right\|^2 \leq d(\left\| p_{Y,k}^0 \right\|^2 + \left\| p_{Z,k}^0 \right\| \left\| p_{Y,k}^0 \right\| + \left\| p_{Z,k}^0 \right\|^2)$, we have from item *ii*) that

$$\begin{aligned} |\gamma_k| \leq & \quad c_1 \left\| p_{Z,k} \right\| + c_2 \left\| p_{Z,k} \right\| \left\| p_{Y,k} \right\| \\ \leq & \quad c_1 \left\| p_{Z,k}^0 \right\| + c_3 \left\| p_{Z,k}^0 \right\| (\left\| p_{Y,k}^0 \right\| + \left\| p_{Y,k}^0 \right\|^2 + \left\| p_{Z,k}^0 \right\| \left\| p_{Y,k}^0 \right\| + \left\| p_{Z,k}^0 \right\|^2) \\ = & \quad c_1 \left\| p_{Z,k}^0 \right\| + c_3 \left\| p_{Z,k}^0 \right\| \left\| p_{Y,k}^0 \right\| + c_3 \left\| p_{Z,k}^0 \right\| \left\| p_{Y,k}^0 \right\|^2 + c_3 \left\| p_{Z,k}^0 \right\|^2 \left\| p_{Y,k}^0 \right\| \\ + & \quad c_3 \left\| p_{Z,k}^0 \right\|^3 \end{aligned}$$

Now, since for $a, b \geq 0$

$$\begin{aligned} (a+b) \geq & \quad a \\ (a+b)^2 \geq & \quad 2ab \\ (a+b)^3 \geq & \quad a^2 b + ab^2 + b^3 \end{aligned}$$

we have, writing $(a+b) = (\left\| p_{Z,k}^0 \right\| + \left\| p_{Y,k}^0 \right\|)$

$$|\gamma_k| \leq \quad c_1(\left\| p_{Z,k}^0 \right\| + \left\| p_{Y,k}^0 \right\|) + c_4(\left\| p_{Z,k}^0 \right\| + \left\| p_{Y,k}^0 \right\|)^2 + c_5(\left\| p_{Z,k}^0 \right\| + \left\| p_{Y,k}^0 \right\|)^3$$

then $\gamma_k = \mathcal{O}\left(\left\| p_{Z,k}^0 \right\| + \left\| p_{Y,k}^0 \right\|\right) = \mathcal{O}\left(\left\| d_k^0 \right\|\right)$ by (3.51). □

This is equivalent to lemma 17 of Lawrence and Tits (2000) modified to allow for the decomposition.

Lemma 3.15 *Suppose assumptions 3.1 and 3.3-3.10 hold. Then* $d_k^C = \mathcal{O}\left(\left\| d_k^0 \right\|^2\right)$.

Proof: Let $\tilde{c}_{\mathcal{I},k} = \left[c_{\mathcal{I}}^i(x_k + d_k) + \|d_k\|^\kappa \mid i \in J^{*,\epsilon}\right]^T$. Then, for $t^i \in (0,1)$

$$\tilde{c}_{\mathcal{I},k} = c_{\mathcal{I}}^i(x_k) + \left\langle \nabla c_{\mathcal{I}}^i(x_k), d_k \right\rangle + \frac{1}{2} \left\langle d_k, \nabla^2 c_{\mathcal{I}}^i(x_k + t^i d_k) d_k \right\rangle + \|d_k\|^\kappa$$

when $d_k = Y_k p_{Y,k} + Z_k p_{Z,k}$, we have by (3.15), (3.16), $Z_k^T \nabla c_{\mathcal{I},k}^i = 0 \ \forall i \in J_k^\epsilon$ and lemma 3.11 for $i \in J^{*,\epsilon}$ and k sufficiently large

$$= \eta_k \gamma_{k-1} + \mathcal{O}(\|d_k\|^2) + \|d_k\|^\kappa$$

and from lemma 3.14

$$\begin{aligned} &= \mathcal{O}(\left\| d_k^0 \right\|^2)\mathcal{O}(\left\| d_k^0 \right\|) + \mathcal{O}(\|d_k\|^2) + \mathcal{O}(\|d_k\|^\kappa) \\ &= \mathcal{O}(\left\| d_k^0 \right\|^2) \end{aligned}$$

since $\kappa > 2$. We have from (3.32), assumption 3.6 and the Broyden safeguards (3.25, 3.26) (see also lemma 5.8 in Biegler et al. (1995)) that $p^C_{Y,k} = \mathcal{O}([c_{\mathcal{E}}(x_k + d_k) \;\; \tilde{c}_{\mathcal{I},k}]) = \mathcal{O}(\left\|d^0_k\right\|^2)$. We have from (3.33) that $B_k \left(p^C_{Z,k} + p_{Z,k}\right) = -\left(Z^T_k g_k + \zeta_k(w_k + w^C_k)\right)$, using (3.17)

$$B_k p^C_{Z,k} = -\zeta_k w^C_k$$

We have by assumption 3.5

$$p^C_{Z,k} = \mathcal{O}(p^C_{Y,k}) = \mathcal{O}(\left\|d^0_k\right\|^2)$$

and the claim follows from (3.31). □

This is equivalent to lemma 18 of Lawrence and Tits (2000) modified to allow for the decomposition.

Proposition 3.1 *Suppose assumptions 3.1 and 3.3-3.10 hold. Then a steplength $\alpha_k = 1$ satisfies the feasibility requirement in the line-search (3.40) of algorithm* ***rFSQP*** *for all k sufficiently large.*

Proof: Consider $\alpha_k = 1$, then $x_{k+1} = x_k + d_k + d^C_k$. We need to prove $c^i_{\mathcal{I}}(x_{k+1}) \leq 0$. The definitions

$$c(x_k)=\begin{bmatrix} c^i_{\mathcal{E}}(x_k)\ \forall i \in \{1,\ldots,p_1\} \\ c^i_{\mathcal{I}}(x_k) - \gamma_{k-1}\eta_k,\ i \in J^\epsilon_k \end{bmatrix}, \quad c^C(x_k)=\begin{bmatrix} c^i_{\mathcal{E}}(x_k + d_k)\ \forall i \in \{1,\ldots,p_1\} \\ c^i_{\mathcal{I}}(x_k + d_k) + \|d_k\|^\kappa\ ,\ i \in J^\epsilon_k \end{bmatrix}$$

allows

$$c(x_{k+1}) = c(x_k + d_k) + \left\langle A(x_k + d_k), d^C_k \right\rangle + \mathcal{O}\left(\left\|d^C_k\right\|^2\right)$$

expanding and using lemma 3.15 gives

$$\begin{aligned} &= c(x_k + d_k) + \left\langle A(x_k), d^C_k \right\rangle + \left\langle d_k, \nabla^2 c(x_k) d^C_k \right\rangle \\ &+ \mathcal{O}\left(\|d_k\|^2\right) d^C_k + \mathcal{O}\left(\left\|d^0_k\right\|^4\right) \end{aligned}$$

denoting $A(x_k) = A_k$ and using lemmas 3.14 and 3.15 give

$$= c(x_k + d_k) + \left\langle A_k, Y_k p^C_{Y,k} \right\rangle + \left\langle A_k, Z_k p^C_{Z,k} \right\rangle + \mathcal{O}\left(\left\|d^0_k\right\|^3\right)$$

inserting $p^C_{Y,k} = -(A^T_k Y_k)^{-1} c^C_k$ from (3.32) and using $A^T_k Z_k = 0$ gives

$$= c(x_k + d_k) - A^T_k Y_k (A^T_k Y_k)^{-1} c^C(x_k) + \mathcal{O}\left(\left\|d^0_k\right\|^3\right)$$

finally the above definitions of $c(x_k)$ and $c^C(x_k)$ result in

$$= - \begin{bmatrix} 0 \\ \gamma_{k-1}\eta_k + \|d_k\|^\kappa \end{bmatrix} + \mathcal{O}\left(\left\|d_k^0\right\|^3\right)$$

By lemma 3.14 $\gamma_{k-1}\eta_k = \mathcal{O}(\left\|d_k^0\right\|^3)$, then $\kappa < 3$ gives $c_{\mathcal{I}}^i(x_{k+1}) < 0,\ i \in J_k^\epsilon$. □

This is equivalent to the first part of proposition 1 of Lawrence and Tits (2000). Note that we do not prove descent, since we are working with a l_1-penalty function. We observe by $A_k^T Z_k = 0$ in the proof above that $p_{Z,k}^C$ only serves to minimize $\left\|d_k + d_k^C\right\|_B$ i.e. that the correction d_k^C does not tilt the overall step d_k unnecessarily away from the `SQP` direction.

Theorem 3.3 *Suppose assumptions 3.1 and 3.3-3.10 hold. Then algorithm* ***rFSQP*** *generates a sequence $\{x_k\}$ which converges 1-step superlinearly to x^*.*

Proof: The proof follows from Biegler et al. (1995) since the algorithm produces a convergent sequence of iterates satisfying

$$x_{k+1} - x_k = d_k^0 + \mathcal{O}(\left\|d_k^0\right\|^2)$$

Note that by lemma 3.11 the local analysis is an equality constrained problem and then lemma 3.3 and assumption 3.9 and 3.10 gives theorem 5.4 of Biegler et al. (1995). Then theorem 5.6 of Biegler et al. (1995) follows and assumption 3.5 is satisfied by our Hessian update scheme. $\zeta_k = 1$ is allowed in equation (3.19) for large k since

$$\hat{\zeta}_k \left[2\cos\theta_k^{BFGS} \left|g_k^T Z_k w_k\right| + w_k^T B_k^{-1} Z_k^T g_k + \hat{\zeta}_k w_k^T B_k^{-1} w_k\right] \leq \rho \left\|c_k\right\|_1$$

The last inequality will be satisfied with $\zeta_k = 1$ for large enough k (see Biegler et al. (1995), section 6). Lemma 6.2 and 6.3 in Biegler et al. (1995) then gives the claim. □

As in Biegler et al. (1995) removing the approximation term w_k will give a two-step superlinearly convergent algorithm. This will reduce the workload of each iteration, and may therefore give a better algorithm in practice. See Biegler et al. (2000) for numerical comparisons of this issue.

3.5 Implementation and results

We discuss implementation and test the algorithm on selected problems from the Hock-Schittkowski (Hock and Schittkowski 1981) test-set. The results

presented in this section illustrate some of the properties of rFSQP, but do not expose it to its intended use; large-scale optimization of problems dominated by equality constraints with a few nonlinear inequality constraints. Two such cases are investigated in chapter 4.

3.5.1 Implementation details

Some essential details are discussed next. The algorithm was implemented in MATLAB 6.0 (R12) and Visual FORTRAN 6.5[2] developing studio, with mex-gateways to some FORTRAN routines.

Most implementation issues follow the same strategy as described in Biegler et al. (2000) and Lawrence and Tits (2000), but are somewhat simplified by vectorization in MATLAB. We treat linear constraints $A_{\mathcal{I},L}x_k \leq b_{\mathcal{I},L}$ and $A_{\mathcal{E},L}x_k = b_{\mathcal{E},L}$ and simple bounds, $L \leq x_k \leq U$, by solving a qp for $p_{Z,k}$ and γ_k. This requires an explicit Z_k. Hence, the "adjoint" approach for computing products like $Z_k^T g_k$ without forming an explicit Z_k is not implemented. Note that simple bounds and linear constraints are not included in the Jacobian matrix A_k in equation (3.6) which defines the matrix Z_k. The qp solved when utilizing the specific form of Z_k and Y_k is

$$\begin{array}{cc} \min\limits_{p_{Z,k},\gamma_k} & \frac{1}{2}\langle p_{Z,k}, B_k p_{Z,k}\rangle + \gamma_k \\ s.t. & \left\langle Z_k^T g_k + \zeta_k w_k, p_{Z,k}\right\rangle \leq \gamma_k \\ & \left\langle Z_k^T A_{\mathcal{I},L}^T, p_{Z,k}\right\rangle \leq b_{\mathcal{I},L} - A_{\mathcal{I},L}x_k - C_k p_{Y,k} \\ & \left\langle Z_k^T A_{\mathcal{E},L}^T, p_{Z,k}\right\rangle = b_{\mathcal{E},L} - A_{\mathcal{E},L}x_k - C_k p_{Y,k} \\ & L - x_k - Y_k p_{Y,k} \leq Z_k p_{Z,k} \leq U - x_k - Y_k p_{Y,k} \\ & \left\langle Z_k^T \nabla c_{\mathcal{I},k}^{i,T}, p_Z\right\rangle - \gamma_k \eta_k \leq -c_{\mathcal{I}}^i(x_k) - \left\langle Y^T \nabla c_{\mathcal{I},k}^{i,T}, p_{Y,k}\right\rangle, \quad i \notin J_k^\epsilon \end{array} \tag{3.54}$$

where the last line includes the inactive inequality constraints not included in the computation of $p_{Y,k}$. I.e. we do not implement the EQP strategy which was assumed during the analysis to facilitate use of theorem 3.1.

Similar qp's, but without γ_k, are solved for $p_{Z,k}^E$ and $p_{Z,k}^C$ as well, i.e. 3 reduced size qp's are solved at each iteration. The clipping of d_k^C and d_k^E occurring in Lawrence and Tits (2000) is not needed, since all bounds are accounted for in the qp's. If a qp was declared infeasible, we set $p_{Z,k}$ to zero and cut $d_k = Y_k p_{Y,k}$ so as to satisfy bounds. This is only introduced to

[2] MATLAB is a licensed product from The Mathworks Ltd. Visual FORTRAN 6.5 is a licensed product from COMPAQ.

recover from inconsistent constraint linearizations away from the solution. Alternatives are described by Gill et al. (1997) (elastic programming). We use the qp-solver `e04naf` from the `NAG` toolbox. The `NAG`-solver only accepts dense Jacobians, and should be replaced by a sparse solver in a production code[3].

The update rule for η_k is vectorized, and the update rule for C_k^η is as in `FSQP`:

i. If the full step $\alpha_k = 1$ was accepted set $C_{k+1}^\eta = C_k^\eta$.

ii. If $\alpha_k < 1$ and no constraints were infeasible at any trial points, set $C_{k+1}^{\eta,i} = \max\left\{\frac{C_k^{\eta,i}}{\delta}, \underline{C}^\eta\right\}$ (decrease $C_{k+1}^{\eta,i}$).

iii. If $\alpha_k < 1$ and some constraints were infeasible at some trial points, set $C_{k+1}^{\eta,i} = \min\left\{\delta C_k^{\eta,i}, \bar{C}^\eta\right\}$ (increase $C_{k+1}^{\eta,i}$) for the constraints that were infeasible at some trialpoints, and set $C_{k+1}^{\eta,i} = C_k^{\eta,i}$ for the rest.

All parameter values are as given in Biegler et al. (2000), and Lawrence and Tits (2000) except for the tolerance ϵ_f controlling the η_k-updates. We use the default value $\epsilon_f = 1e-3$. The parameter ϵ controlling the inclusion in the active set J_k^ϵ is critical for the performance of `rFSQP`. We use the default value $\epsilon = 1$. To speed up the algorithm, it may be advantageous to start with a large value, say $\epsilon = 1$, and decrease it down to $k_\epsilon * \mathit{featol}$, where *featol* is the feasibility tolerance for the nonlinear inequalities and $k_\epsilon > 1$, during the progress of the algorithm. This is problem dependent though, and the results reported are for constant values of ϵ. The value $\epsilon = 1$ was found by trial and error. A large value of ϵ is beneficial for problems where the solution x^* does not lie on active inequality constraints. For problems where the solution does lie on active inequality constraints, large values of ϵ can cause slow convergence, since the algorithm may approach the solution conservatively.

Basis selection in `MATLAB` can be done by `LU`-decomposition using partial pivoting given $n > m_k$ provided $rank(A_k) = m_k$. If $rank(A_k) < m_k$ the system is not guaranteed to have a solution, and this can cause numerical problems. To deal with rank-deficient problems row and column pivoting are necessary, and we use the Harwell routine `MA28` (Harwell Laboratory

[3]In the present implementation of `rFSQP` the constraint Jacobian is converted to a dense matrix by issuing the `MATLAB` command `full` just before the call to `e04naf`. Still this works better than using `MATLAB`'s `quadprog` which accepts sparse Jacobians, but does not handle poorly conditioned Jacobians appropriately.

1995) through a mex-gateway for this purpose. This routine allows reuse of a previous pivoting sequence provided the same non-zero structure is maintained. Hence, complete refactorization is rarely needed (this is similar to the approach of Biegler et al. (2000)), except when a change in the active set occurs.

To asses the relative level of sophistication of our implementation we implemented the algorithm of Biegler et al. (2000) which we term rSQP for brevity. In fact rFSQP is implemented as an extension to this algorithm with some important modifications. Our implementation of the algorithm by Biegler and co-workers was sensitive to the results from the qp-solver on some of the harder problems. Since the MATLAB qp-solver seems to produce inaccurate solutions for badly conditioned problems, we implemented rSQP with the qp-solver QPKWIK obtained with kind permission from Professor Lorenz T. Biegler. QPKWIK is also used in Biegler's implementation. Since QPKWIK needs the inverse Cholesky-factors of the Hessian as input, we also replaced the BFGS-updating scheme with Biegler's updating scheme for the inverse Cholesky-factors (which makes use of the qp-multipliers). Since the Hessian of the implemented qp (3.54) is semi-definite, the inverse Cholesky factors are undefined and use of QPKWIK in rFSQP is therefore not trivial. However, since rFSQP worked well with both MATLAB's quadprog and NAG's e04naf in combination with a standard BFGS updating scheme we did not consider using QPKWIK in rFSQP. Clearly this puts a bias on the comparison of the two algorithms, and we only include the results on rSQP to give an idea of the level of sophistication of rFSQP. Some of the results reported below give unexpectedly poor results for rSQP and we expect that this is a result of our implementation.

The line search is identical in both implementations apart from the feasibility check in rFSQP, and follows as in Dennis and Schnabel (1996) modified to allow for checking of constraint violations. The constrained line search tests for feasibility prior to descent, but does not re-order constraints as in Lawrence and Tits (2000). I.e. vectorization in MATLAB allows a simultaneous test, while a FORTRAN program must perform this test in a loop. In the present version of rFSQP the qp-multipliers are not used other than for updating the penalty parameter ν.

When constraints are removed from the active set, we re-initialize the corresponding reduced Hessian to I. The current implementation does not implement a watch-dog strategy, and we do not handle a rank-deficient Jacobian. The algorithm detects this and stops, but this does not occur on any of the test-problems. From table 4 in Biegler et al. (2000) we note that the cross-terms w_k and $\bar{w}_k$ have relatively little influence on the

iteration counts for the selected problems from the Hock-Schittkowski test-set. Therefore we implemented both algorithms with Broyden corrections only. Then, since we never resort to finite differences the iteration counts and gradient evaluation counts reported below are identical for both `rSQP` and `rFSQP`.

3.5.2 Results

We give preliminary test results for a selection from the Hock-Schittkowski test set for *our* implementation of `rSQP` by Biegler et al. (2000), and `rFSQP` in table 3.1. The results for `rSQP` are included to illustrate the level of sophistication of the implementation of `rFSQP`. In `rSQP` nonlinear inequalities were implemented as equalities with slacks.

On hs33 both implementations converge to $\tilde{x}^* = (0, 0, 2)$ with objective value $f^* = -4$ (which is also reported in table 6.4 of Gill et al. (1997)). The value $\tilde{x}^*$ is the same value as returned by the `NAG` routine `e04ucf`. On hs34 `rFSQP` performs poorly, and needs $\epsilon = 0.1$ to converge. On hs93 `rFSQP` attracts very slowly to x^*, and no `BFGS`-updates occurs. On hs111 both algorithms converge to a point $\tilde{x}^* \neq x^*$ given by Hock and Schittkowski (1981). The value $\tilde{x}^*$ is the same value as returned by the `NAG` routine `e04ucf`. In hs112 there is an error in the problem, and the problem was solved with the modification stated in the `CUTE` (Bongartz, Conn, Gould, and Toint 1995) test set. We expect that the difference between `rSQP` and `rFSQP` on hs111 is caused by the different choices of Hessian update schemes in combination with different `qp`-solvers.

The `rSQP`-results indicate the inadequate level sophistication of the code which we based `rFSQP` on. I.e. it solves the simpler problems with iteration and function evaluation counts comparable to Biegler et al. (2000), but experiences problems especially on hs100 and hs111 and fails on hs84 and hs117. The `rFSQP`-results show that on some problems the iteration count increases (as expected), when the `SQP`-direction is tilted. In other cases the performance is improved, since feasible iterates may remain in a "nicer" region than the pure `SQP` iterates. This may be beneficial during later iterations. Observe that the workload of each iteration of `rFSQP` is larger than for `rSQP` since 3 subproblems must be solved instead of one.

Comparing to the results reported for `FSQP`' by Lawrence and Tits (1996), Lawrence and Tits (2000) show that `FSQP`' performs better on most of the problems reported in table 3.1. In particular, `rFSQP` performs poorly on hs12, hs34, hs43, hs84, hs93 and hs117. Comparing with `SNOPT` it is observed that `rFSQP` experiences problems on hs34, hs43, hs66, hs80, hs84, hs93,

Problem	rSQP			rFSQP			
	IT	NF	NG	IT	NO/NC	NG	ϵ
hs1	28	45	28	28	45/45	28	(-)
hs12	22	22	22	15	33/48	15	(def)
hs29	14	23	14	10	18/27	10	(def)
hs30	17	17	17	9	11/19	9	(def)
hs31	9	16	9	12	30/41	12	(def)
hs33	5	5	5	5	6/10	5	(def)
hs34	10	11	10	34	86/119	34	0.1
hs43	28	94	28	27	68/94	27	(def)
hs66	8	9	8	10	12/21	10	(def)
hs80	11	11	11	11	11	11	(-)
hs81	11	11	11	11	11	11	(-)
hs84	fail			23	45/67	23	(def)
hs93	37	77	37	fail			
hs99	10	15	10	16	79/79	16	(def)
hs100	63	169	63	30	101/130	30	(def)
hs111	136	361	136	67	128/128	67	(-)
hs112	24	57	24	30	68/68	30	(-)
hs113	41	62	41	15	29/43	15	(def)
hs117	fail			32	38/69	32	(def)

Table 3.1: *rFSQP on the Hock-Schittkowski test set.* The table gives the results on the Hock-Schittkowski (Hock and Schittkowski 1981) test set. rSQP=our implementation of the algorithm of Biegler et al. (2000), rFSQP=the present algorithm, IT=iterations, NF=function evaluations, NG=gradient evaluations, NO=objective evaluations, and NC=constraint evaluations. For rSQP NO=NC≡NF. Since finite differences are turned off we have IT=NG. ϵ is the active set tolerance for rFSQP. The default value ϵ=1 is marked with (def). For equality constrained problems ϵ is irrelevant and marked with (-).

hs100 and hs117. Hence, further testing is necessary to evaluate the possible advantages of the `rFSQP` algorithm over the algorithms of Lawrence and Tits (2000), Biegler, Nocedal, and Schmid (1995) and Gill, Murray, and Saunders (1997). It may be that the reported results can be improved by improving the implementation of `rFSQP`. Hence, the poor results need not necessarily be a result of the method `rFSQP`, but rather a consequence of the implementation.

We experienced that the `rFSQP` algorithm is sensitive to the threshold value ϵ for J_k^ϵ on certain problems. On equality constrained problems ϵ is irrelevant. A large value like $\epsilon = 1$ causes the algorithm to stay well inside the interior of the feasible region, and this reduces the number of changes in the active set, which reduces the frequency of complete re-factorization of the constraint Jacobian to find a new pivoting strategy. However, this causes the algorithm to approach inequalities conservatively, which may cause slow convergence on problems where the optimum is located on active nonlinear inequalities. The observed behavior of the algorithm is thus more related to interior methods[4] than to "working set" `SQP` methods. Still it is implemented and analyzed as a `SQP` method, and in particular larger counts for function evaluations are expected on large scale problems, as is usual for `SQP` methods.

There are a number of issues to be addressed to improve the present implementation. It should be tested on large-scale problems, the Hessian updates should be improved, possibly by exploiting the `qp`-multipliers. 2nd derivative Hessians may be introduced, for example by partial separability of the Lagrangian combined with perturbed Cholesky factorization or with trust-regions combined with conjugate-gradient steps. Other line searches and mechanisms for global convergence should be implemented, in particular how to implement a watch-dog in combination with the constrained line search. Finite differences and rank-deficient Jacobians should be handled. Hot-starting the `qp`-solver by re-use of the last active set, and sparse linear algebra `qp`-solvers could be used. A production code should also be independent of the interpreted `MATLAB` environment.

To illustrate the workings of `rFSQP` we compare it with the iterations of `rSQP` on hs12 from the Hock-Schittkowski test-set and a modification of hi3 in Himmelblau (1972), p. 394. The modification is that we multiply the objective by 1000 and alter the second constraint. Note that hi3 is quite

[4]Some feasible interior methods for nonlinear programming has appeared recently; Forsgren and Gill (1998) and Byrd, Nocedal, and Waltz (2000).

similar to hs59. We restate these problems for completeness.

$$\begin{array}{ll} \min\limits_{x} & .5x_1^2 + x_2^2 - x_1x_2 - 7x_1 - 7x_2 \\ s.t. & 25 - 4x_1^2 - x_2^2 \geq 0 \end{array} \tag{3.55}$$

$$\begin{array}{ll} \min\limits_{x} & 1000 \cdot f(a, x, x^2, x^3, x^4, \exp(x)) \\ s.t. & 700 - x_1x_2 \leq 0 \\ & \frac{x_1^2}{125} - x_2 \leq 0 \\ & 5(x_1 - 55) - (x_2 - 50)^2 \leq 0 \\ & 0 \leq x_1 \leq 75 \\ & 0 \leq x_2 \leq 65 \end{array} \tag{3.56}$$

In hs12, equation (3.55), the initial point is $x_0 = (0, 0)$ while the solution is $x^* = (2, 3)$. In hi3, equation (3.56), the original initial point is infeasible, so we start from $x_0 = (70, 40)$ instead. The solution is $x^* = (75, 65)$. The iterations are plotted with objective contours and constraints as equalities in figures 3.2 and 3.3. The feasible regions are evident from the bold face lines which are the borders of the inequalities. The iteration sequences for

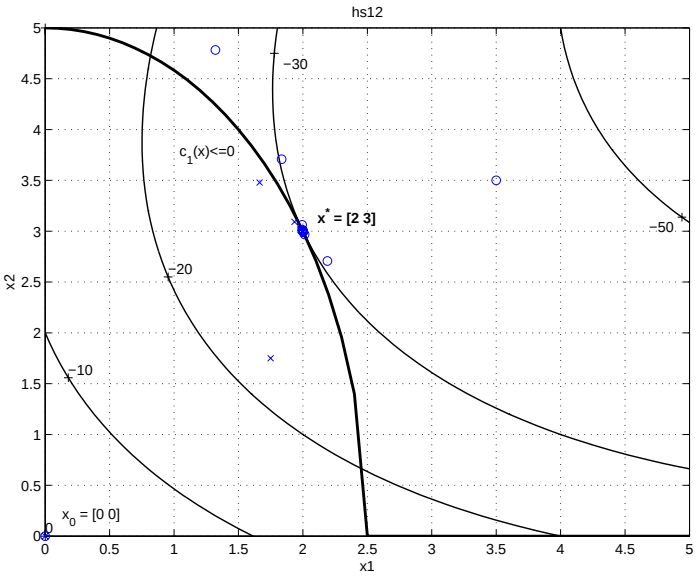

Figure 3.2: *Iterations for* `rSQP` *(o) and* `rFSQP` *(×) on hs12 from the Hock-Schittkowski test set.*

`rSQP` and `rFSQP` on hs12 are given in table 3.2. The iteration sequences for `rSQP` and `rFSQP` on hi3 are given in table 3.3.

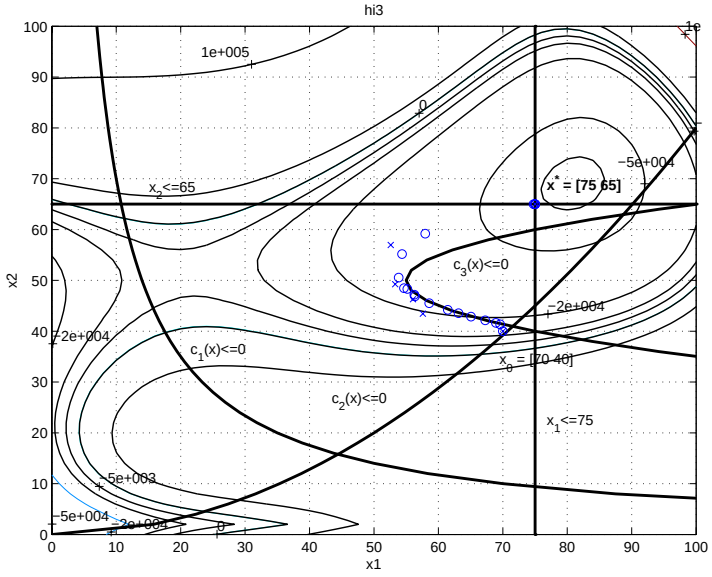

Figure 3.3: *Iterations for* `rSQP` *(o) and* `rFSQP` *(×) on hi3 from Himmelblau (1972).*

We observe from figure 3.2 that `rSQP` produces infeasible iterates, while `rFSQP` remain feasible. The effect of the constrained line search is observed in iteration 1 where both methods generated the same step $d_1 = (3.5, 3.5)$. In `rSQP` the full step was accepted since this gives a reduction in the merit function, but in `rFSQP` the full step gives infeasibility and the next trial value $\alpha = 0.5$ is tested. $\alpha = 0.5$ is accepted since it is feasible and gives a decrease in the merit function. In Biegler et al. (1995) one-step superlinear convergence was proved with finite differences for the correction terms w_k and $\bar{w}_k$. Since we only implement Broyden corrections we expect 2-step superlinear convergence, i.e. that we get one extra correct figure every 2nd iteration (for large k). The results for hs12 shows that this is nearly obtained for both algorithms. We have $\alpha_{k,\texttt{rSQP}} = 1$ for all k and $\alpha_{k,\texttt{rFSQP}} = 0.5$ for all k on hs12. Improvements in performance can be expected by implementing a watch-dog for `rFSQP` to allow unit step-lengths for large k and including finite differences. The quality of finite difference corrections depends on good multiplier estimates, and using the `qp` multipliers should be considered.

On hs12 one could anticipate that reducing $\epsilon = 1$ down towards *featol*$= 10^{-5}$ would improve the result for `rFSQP`. It turned out that `rFSQP` was

Iteration no.	rSQP		rFSQP	
	x_1	x_2	x_1	x_2
0	0	0	0	0
1	3.5000	3.5000	1.7500	1.7500
2	3.5000	3.5000	1.6638	3.4783
3	2.1956	6.1283	1.9345	3.0917
4	1.3217	4.7830	1.9811	3.0133
5	1.8346	3.7076	1.9934	2.9994
6	2.1901	2.7060	1.9973	2.9980
7	1.9942	3.0615	1.9988	2.9985
8	2.0128	2.9674	1.9995	2.9991
9	1.9946	3.0149	1.9997324	2.9995665
10	2.0033	2.9915	1.9998641	2.9997889
11	1.9983	3.0045	1.9999316	2.9998958
12	2.0009	2.9975	1.9999657	2.9999482
13	1.9995	3.0013	1.9999828	2.9999741
14	2.0003	2.9993	1.9999914	2.9999871
15	1.9998548	3.0003874	1.9999957	2.9999935
16	2.0000787	2.9997903		
17	1.9999575	3.0001133		
18	2.0000230	2.9999387		
19	1.9999876	3.0000331		
20	2.0000067	2.9999821		
21	1.9999964	3.0000097		
22	2.0000020	2.9999948		

Table 3.2: *rFSQP on hs12.* The table gives the iterations for rSQP and rFSQP on hs12 from the Hock-Schittkowski test set.

insensitive to ϵ on hs12, and the same result was achieved for all trial values $\epsilon \in [10^{-5}, 10]$. However, increasing the parameter δ used in updating the parameter C_k^η has an effect. C_k^η in turn influence on the updating of η_k through step xvii in the rFSQP algorithm. Increasing δ from 2 to > 2.5 reduced the iteration count from 15 to 12 iterations on hs12.

Figure 3.3 illustrates the practical performance of rFSQP in the face of a nonconvex feasible region. Observe that rFSQP tends to stay far away from the inequality boundary during the early iterations, which in this case is beneficial. In hs12 where the optimum is located on a nonlinear inequality boundary, rFSQP is seen to be somewhat conservative. On hi3 rSQP also

Iteration no.	rSQP		rFSQP	
	x_1	x_2	x_1	x_2
0	70	40	70	40
1	69.5840	41.3540	57.5890	43.4455
2	68.9159	41.6532	56.0615	46.2790
3	67.2837	42.1434	53.2885	49.2451
4	65.0578	42.8566	52.6044	56.9472
5	63.1240	43.5463	75	65
6	61.4623	44.2100		
7	58.5788	45.5074		
8	56.3748	46.8613		
9	54.6501	48.4721		
10	56.2571	47.1791		
11	55.1728	48.2882		
12	53.8286	50.5531		
13	54.3550	55.1590		
14	57.9557	59.1733		
15	74.7619	65		
16	75	65		

Table 3.3: *rFSQP on hi3.* The table gives the iterations for rSQP and rFSQP on hi3 from Himmelblau (1972).

walks round the third inequality constraint in equation (3.56). The first direction for $(x_1, x_2) = (70, 40)$ is $d_1 = (-0.4160, 1.3540)$. (The direction for the third slack is zero, with $s_3(0) = 0$.) The full step is accepted, but does not enter or cross the feasible region in this case. Note that not all α_k's are equal to one for rSQP in this example. The number of function evaluations NF is 32 for rSQP and NO/NC is 5/10 for rFSQP. For hi3 we have $\alpha_{k,\texttt{rFSQP}} = 1$ for all k.

3.6 Conclusions

rFSQP is a reduced Hessian SQP method that remains feasible with respect to inequality constraints at all iterates. rFSQP is designed for large scale problems dominated by equality constraints, but with a few nonlinear inequality constraints of which we wish to remain feasible. It can be argued that feasibility of inequalities can give general benefits as discussed in section 3.2, which motivate the extension to large scale methods like reduced Hes-

sian methods. The benefit of feasible iterates must be traded off against the main benefit of SQP methods; the iterates need not be feasible, see Boggs and Tolle (1995) (feasibility can be very expensive to achieve in the presence of nonlinear constraints). However, we are only concerned with feasibility of inequalities, and the trade-off is thus shifted in favor of rFSQP, although of course no general recommendation can be provided.

The preliminary results in section 3.5.2, show that on some problems rSQP performs better, while on other problems rFSQP perform better. As could have been anticipated there is no general recommendation to be made about the method of choice for a given problem. Comparing the results of rSQP with the results of Biegler et al. (2000) shows that there is a significant potential for improvements in rFSQP as well. Hence, the testing is not conclusive. Given the complexity of the algorithm, further testing and refinements are needed to show any clear advantages of the proposed algorithm.

Acknowledgement 3.1 *The author expresses his gratitude to Professor Lorenz T. Biegler and Andreas Wächter at Carnegie Mellon University for their insightful comments concerning various implementation issues and for providing access to their implementation of the algorithm published in Biegler et al. (2000).*

Chapter 4

NMPC

Nonlinear model predictive control (NMPC) is a control strategy where application of nonlinear optimization methods is essential. The motivation for this chapter is to combine the model from chapter 2 and the optimization algorithm from chapter 3 in a NMPC framework. This chapter is application oriented, and contributes to the practical knowledge of implementation of NMPC. Parts of this chapter has been submitted for possible publication to the 15th IFAC World Congress on Automatic Control (Martinsen and Biegler 2002).

Optimal control problems generally lead to boundary value problems (BVP), since the adjoint equations must be integrated backwards in time (Vinter 2000), (Lewis and Syrmos 1995). Since NMPC is a special case of optimal control problems, it is expected that numerical methods suitable for BVP's should be adapted in NMPC as well. The most popular numerical methods for solving BVP's are (finite) element methods, possibly in combination with orthogonal collocation (Finlayson 1980). Multiple shooting has also been applied to BVP's with success, see Ascher, Mattheij, and Russell (1995) for details on multiple shooting and orthogonal collocation. However, it is important to observe that batch-processes may be formulated as initial value problems (IVP), since stabilizing end-point constraints are irrelevant.

The chapter focuses on application of SQP optimization algorithms in NMPC, but emphasizes that appropriate model discretization is essential for the performance. A rule of thumb is that if it is impossible to integrate a model with a certain method, the method cannot be used in NMPC. In particular, since the sinter model from chapter 2 could not be integrated by explicit methods, it should not be represented with explicit methods in NMPC.

We discuss the differences and similarities between feasible and infeasible path methods, sequential and simultaneous methods and reduced and full space methods. We informally assess the suitable choices between these various strategies by applying them to two cases of different complexity.

The chapter starts with some background in section 4.1. In section 4.2 a conceptual comparison between single shooting and reduced Hessian methods is provided. Simulation results follow in section 4.3. Discussion and conclusions to the chapter follow in sections 4.4 and 4.5.

4.1 Introduction

This chapter is concerned with practical issues related to optimization within nonlinear model predictive control (NMPC), which is one of the major obstacles to be overcome in industrial applications of NMPC (Qin and Badgwell 2000). We consider different approaches to optimization, such as feasible vs. infeasible methods, sequential vs. simultaneous methods, reduced vs. full space methods and dense vs. sparse algebra methods, and apply these methods to two cases; a CSTR with first order reaction and to the grate sintering from chapter 2.

The theory of optimization algorithms is not dependent on how the equality constraints are formed. For instance in optimal control, and in particular in the special case of nonlinear model predictive control (NMPC), much concern is put into discretization schemes for the nonlinear equality constraints. These equality constraints result from a continuous-time nonlinear dynamical system repeated over a time horizon P. Three major variants are usually considered to handle unstable modes; orthogonal collocation, multiple shooting and single shooting (possibly with a variable grid). Ascher, Mattheij, and Russell (1995) discuss the general benefits of these approaches, and Barclay, Gill, and Rosen (1998) discusses this in conjunction with SQP algorithms. Vassiliadis (1993) addresses the dynamic optimization of general DAE systems. These approaches seek to find formulations of the equality constraints that are less hard to satisfy, while simultaneously reducing the discretization error.

Nonlinear inequality constraints may be introduced for stability purposes in NMPC (Scokaert, Mayne, and Rawlings 1999), (Chen and Allgöwer 1998b), (1998a). These references extends earlier work (Li and Biegler 1988), (Li and Biegler 1989), (Li, Biegler, Economou, and Morari 1990), (de Oliveira and Biegler 1995). These references are concerned with nominal stability. Hence, termination prior to convergence of the optimizer cannot guarantee

stability unless the equality constraints are satisfied.

The immediate answer to this is single shooting; i.e. always solving the model. Single shooting algorithms progress towards a solution by iterating between solving the model and solving a reduced size optimization problem. Due to this, single shooting is said to be a sequential method. Single shooting produces a reduced gradient problem in the free variables to be solved at each NMPC iteration. Maintaining feasibility of nonlinear inequalities involving dependent variables can then be obtained by use of rFSQP from chapter 3 or FSQP' of Lawrence and Tits (2000). This may be costly if evaluation of the problem functions is costly, e.g. if an implicit discretization scheme must be applied. In addition, according to Ascher et al. (1995), section 4.1 and 4.6.2, single shooting lacks robustness when applied to unstable systems. This problem can be somewhat alleviated by using an adaptive step length (Baker and Polak 1994), (Pytlak and Vinter 1998), (Pytlak and Vinter 1999), (Pytlak 1999).

To solve optimization problems with stabilizing endpoint constraints simultaneous methods must be applied. End-point constraints make the problem a two-point boundary value problem (TPBVP) which in general cannot be resolved with single shooting. Simultaneous methods do not solve the model at each iteration. Instead a simultaneous search for a model solution and optimal point is carried out. Multiple-shooting and orthogonal collocation, possibly on finite elements (Finlayson 1980), are the most widely used simultaneous methods. Since simultaneous methods do not solve the model at each iteration, they cannot guarantee stability in the nominal stability setting of dual-mode or quasi-infinite horizon NMPC if terminated prior to convergence. Note that the results reported in Bock, Diehl, Schlöder, Allgöwer, Findeisen, and Nagy (2000) show that termination prior to convergence in multiple shooting may be viable for some applications. Decomposition strategies for orthogonal collocation on finite elements have been considered by Cervantes and Biegler (2000) and Biegler, Cervantes, and Wächter (2001).

Initialization near the optimal point can happen frequently in real-time applications like model predictive control and receding horizon estimation (see Allgöwer et al. (1999)). In NMPC an optimization problem is solved repeatedly with (usually) small changes from one problem to the next. These are real-time applications where termination prior to convergence may be necessary, and an improved but non-optimal feasible solution is favorable if early termination is required. The only method that can be terminated prior to convergence is single shooting. As shown in section 4.3 below, choosing an appropriate optimization algorithm may have significant influence on

the overall computation time, in particular for processes where implicit discretization schemes are necessary.

Observe that for finite time batch-processes stability in the sense of Lyapunov[1] is irrelevant, provided the process does not have finite escape time. Hence, the grate sintering case is formulated as a `NMPC` problem without endpoint constraints, and it is not established that the objective is a Lyapunov function for the `NMPC` problem. As a consequence, `TPBVP` techniques are not needed to solve this problem. Due to the stiffness of the model the implicit Lobatto IIIC scheme was adapted in the solution.

The choice between active set vs. interior-point methods is also important in `NMPC`. Bartlett, Wächter, and Biegler (2000) discuss this issue both on the `NLP` and the `qp` level. The results in section 4.3 were achieved with the `qp` solver `e04naf` from the `MATLAB NAG` toolbox. This is an active set solver, where the constraint Jacobian must be presented as a dense matrix.

Define the superscript notation $z^k = \{z_k\}_{k\subset\mathbb{N}}$ for an entity z indexed by k. The nonlinear `MPC` problem with $\dim(u_k) = n_u$ and $\dim(x_k) = n_x$

$$\begin{array}{ll} \min_{x,u;k} & \frac{1}{2}\left(\sum_{k=0}^{P-1}\|x_{k+1}\|_Q^2 + \sum_{k=0}^{M}\|u_k\|_R^2\right) \\ \textbf{s.t.} & c_{\mathcal{E},k} = c_{\mathcal{E}}(x_k,u_k) = f(x_k,u_k) - x_{k+1} = 0, \quad k = 0,\cdots P-1 \\ & x^k \in \mathbb{X}\times\cdots\times\mathbb{X} \\ & u^k \in \mathbb{U}\times\cdots\times\mathbb{U} \end{array} \tag{4.1}$$

with $M < P$ is considered in this paper. The equality constraints in (4.1) have been formed by assuming an explicit discretization scheme. Endpoint constraints or augmentation of the objective may be included to guarantee nominal stability of the `MPC` algorithm, see Mayne, Rawlings, Rao, and Scokaert (2000). Observe that reference tracking and non-zero set-points can be handled in this framework with minor modifications. We assume sufficiently smooth 1st principles state-space models with measured states, analytic 1st order derivatives and that

$$(\mathcal{X},\mathcal{U}) = (\mathbb{X}\times\cdots\times\mathbb{X}, \mathbb{U}\times\cdots\times\mathbb{U})$$

can be described by bounds.

[1] For example, cf. Khalil (1996), theorem 3.1.

4.2 Optimization methods

First we consider SQP in general. The MPC problem (4.1) can be restated as a general nonlinear programming (NLP) problem:

$$\begin{array}{ll} \min_x & f(x) \\ \textbf{s.t.} & c_{\mathcal{E}}(x) = 0 \\ & c_{\mathcal{I}}(x) \leq 0 \end{array} \tag{4.2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $c_{\mathcal{E}} : \mathbb{R}^n \to \mathbb{R}^m$ and $c_{\mathcal{I}} : \mathbb{R}^n \to \mathbb{R}^p$ where $n = n_x P + n_u M$, $m = n_x(P+1)$ and $p = 2n$ (assuming upper and lower bounds on (x_k, u_k) over the horizons P and M). The Jacobian matrix of the equality constraints is denoted $A_k^T = A(x_k)^T = [\nabla c_{\mathcal{E}}^1(x_k), \nabla c_{\mathcal{E}}^2(x_k), \cdots, \nabla c_{\mathcal{E}}^m(x_k)]$ where $c_{\mathcal{E}}^i(x_k)$ is the i-th component of the vector $c(x_k)$. The matrix $G(x_k)$ made up of A_k^T and the gradients of the active inequality constraints is assumed to have full column rank. The null-space of $G(x_k)^T$ defines the tangent space to the equality and active inequality constraints at x_k. Denote $H\mathcal{L}_k$ the Hessian of the Lagrangian function $\mathcal{L}(x_k, \lambda_{\mathcal{E},k}, \lambda_{\mathcal{I},k}) = f(x_k) + \lambda_{\mathcal{E},k}^T c_{\mathcal{E}}(x_k) + \lambda_{\mathcal{I},k}^T c_{\mathcal{I}}(x_k)$ where $\lambda_{\mathcal{E},k}$ and $\lambda_{\mathcal{I},k}$ are the multiplier vectors. We assume strong second order sufficient conditions, i.e. that x^* is an isolated minimum of the NLP (4.2) and that $\lambda_{\mathcal{E}}^*$ and $\lambda_{\mathcal{I}}^*$ are unique. The strong second order conditions are:

A1 The 1st order necessary conditions hold, i.e. $\exists\ \lambda_{\mathcal{I}}^* \geq 0,\ \lambda_{\mathcal{E}}^*$ (optimal multipliers) s.t. $\nabla\mathcal{L}^* = \nabla f(x^*) + \nabla c_{\mathcal{E}}(x^*)\lambda_{\mathcal{E}}^* + \nabla c_{\mathcal{I}}(x^*)\lambda_{\mathcal{I}}^* = 0$.

A2 The columns of $G(x^*)$ are linearly independent.

A3 Strict complementary slackness hold, i.e. $c_{\mathcal{I}}^i(x^*)\lambda_{\mathcal{I}}^{i,*} = 0,\ i = 1, \cdots, p$ and if $c_{\mathcal{I}}^i(x^*) = 0$ then $\lambda_{\mathcal{I}}^{i,*} > 0$.

A4 The Hessian of the Lagrangian function with respect to x is positive definite on the null space of $G(x^*)^T$, i.e. $d^T H\mathcal{L}^*\ d > 0 \forall d \neq 0$ s.t. $G(x^*)^T d = 0$.

The quadratic subproblem to be solved at each iteration of the SQP algorithm becomes:

$$\begin{array}{ll} \min_{d_k} & \nabla f(x_k)^T d_k + \frac{1}{2} d_k^T B_k d_k \\ \textbf{s.t.} & \nabla c_{\mathcal{E}}(x_k)^T d_k + c_{\mathcal{E}}(x_k) = 0 \\ & \nabla c_{\mathcal{I}}(x_k)^T d_k + c_{\mathcal{I}}(x_k) \leq 0 \end{array} \tag{4.3}$$

where $B_k \succ 0$ usually is an approximation to $H\mathcal{L}_k$. The inequality constraints of this `qp` can be resolved either by an active-set strategy or an interior-point strategy. Interior-point methods usually introduce slacks $s \geq 0$ to the inequality constraints s.t. $c_{\mathcal{I}}(x) + s = 0$, and augment the objective; $f(x) + \sigma \sum_{i=1}^{p} \log s_i$. This leads to ill-conditioning when $s \to 0$ and consequently modified Newton methods must be applied to ensure rapid convergence, see Wright (1997b) for further details. Both active-set and interior-point methods can be implemented with sparse matrix solvers, but this seems to be easier to implement for the latter approach (Wright 1997a).

If B_k is uniformly positive definite on the null space $\mathcal{N}(G_k^T)$, then the solution d_k of problem (4.3) forms a strong descent direction for an exact penalty function ϕ_ν. The minimum of ϕ_ν corresponds to a solution of problem (4.2) (Fletcher 1987). Therefore a step size α_k along d_k can be found which is bounded away from zero yielding convergence. Convergence from remote starting points is not guaranteed by this since the linearized constraints might lead to infeasible subproblems. This can be resolved by resorting to elastic programming giving "good" infeasible points, see e.g. Gill, Murray, and Saunders (1997). We have argued that the basic `SQP`-method will converge to stationary (`KKT`) points, but special provisions must be made to avoid convergence to maximum and saddle points. This usually involve exact second order derivatives and trust-region methods, see Biegler (2000) for an introduction. Further details on `SQP` can be found in Boggs and Tolle (1995). The basic `SQP`-algorithm with quasi-Newton updates is summarized as:

Algorithm 4.1 *SQP*

i. Guess x_0, set $B_0 = I$.

ii. At x_k evaluate $f(x_k)$, $c_{\mathcal{E}}(x_k)$, $c_{\mathcal{I}}(x_k)$, $\nabla f(x_k)$, $\nabla c_{\mathcal{E}}(x_k)$ and $\nabla c_{\mathcal{I}}(x_k)$.

iii. If a certain curvature condition is satisfied update B_k by a BFGS formula.

iv. Solve (4.3) for d_k.

v. Test for convergence.

vi. Find a step-size α_k s.t. $0 < \alpha_k \leq 1$ and $\phi_\nu(x_k + \alpha_k d_k) < \phi_\nu(x_k)$.

vii. Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$ and go to ii.

4.2.1 Reduced gradient methods

The qp (4.3) can be resolved in the full space of free and independent variables, or in the reduced space of free variables by a suitable elimination of variables. Elimination of variables exploits that if $n_x P - n_u M >> n_u M$ and that $n_u M$ is small, the reduced subproblem for the null-space step will be small (but dense). In the full space the sparsity of both the Hessian (which commonly requires analytic Hessians) and the Jacobian can be exploited to yield fast solutions (Rao, Wright, and Rawlings 1998). This section shows that a reduced gradient approach can be derived by following two different strategies. The first uses a sequential approach, see e.g. de Oliveira and Biegler (1995), while the second follows the simultaneous null-space approach (Nocedal and Wright 1999), (Biegler, Nocedal, and Schmid 1995). We only outline the central issues necessary to promote a qualitative comparison of the methods.

Sequential approach (sSQP)

By iterating the model over the horizon P, the transformation $x^k = \Psi(x_0, u^k)$ allows the equivalent form

$$\begin{array}{ll} \min_{u^k} & f_k^u(u^k) \\ \textbf{s.t.} & \Psi^k(x_0, u^k) \in \mathcal{X} \\ & u^k \in \mathcal{U} \end{array} \tag{4.4}$$

The transformation $\Psi(\cdot)$ for linearized and discretized systems is essentially a projection onto the subspace $\mathcal{U}$. The sequential approach solves the model at each iteration, i.e. the qp-subproblem is solved following a feasible path strategy. The KKT conditions for problem (4.1) are (temporarily assuming inactive bound constraints on x^k and u^k without loss of generality)

$$\begin{aligned} \nabla_{x^k} f_k + \nabla_{x^k} c_{\mathcal{E}}^{k,T} \lambda_{\mathcal{E}}^k &= 0 \\ \nabla_{u^k} f_k + \nabla_{u^k} c_{\mathcal{E}}^{k,T} \lambda_{\mathcal{E}}^k &= 0 \\ c_{\mathcal{E}}^k &= 0 \end{aligned}$$

Details on the partial derivatives are given in appendix C.4. We eliminate $\lambda_{\mathcal{E}}^k = -\left(\nabla_{x^k} c_{\mathcal{E}}^k\right)^{-T} \nabla_{x^k} f_k$, define $S_k^T = -\nabla_{u^k} c_{\mathcal{E}}^{k,T} \left(\nabla_{x^k} c_{\mathcal{E}}^k\right)^{-T}$ and get the reformulated KKT conditions

$$\begin{aligned} \nabla_{u^k} f_k + S_k^T \nabla_{x^k} f_k &= 0 \\ c_{\mathcal{E}}^k &= 0 \end{aligned}$$

Note that the above result also can be derived by considering the total differential of $c_{\mathcal{E}}^k = 0$, which is $dc_{\mathcal{E}}^k = \nabla_{x^k} c_{\mathcal{E}}^{k,T} dx^k + \nabla_{u^k} c_{\mathcal{E}}^{k,T} du^k = 0$, and define $S_k^T = \left(\frac{dx^k}{du^k}\right)^T = -\nabla_{u^k} c_{\mathcal{E}}^{k,T} \left(\nabla_{x^k} c_{\mathcal{E}}^k\right)^{-T}$ (hence the notion of sensitivity[2]). In the view of problem (4.4), consider the objective and model derivatives with respect to u^k

$$\frac{df_k}{du^k} = \nabla_{u^k} f_k + \left(\frac{dx^k}{du^k}\right)^T \nabla_{x^k} f_k = \nabla_{u^k} f_k + S_k^T \nabla_{x^k} f_k$$
$$\frac{dc_{\mathcal{E}}^k}{du^k} = \nabla_{u^k} c_{\mathcal{E}}^k + \nabla_{x^k} c_{\mathcal{E}}^k S_k$$

Note that the sensitivity matrix gives search directions $d_k^x = S_k d_k^u$, hence $\frac{dc_{\mathcal{E}}^k}{du^k} d_k^u = (\nabla c_{\mathcal{E}}^k)^T d_k$. Also note that by defining $Z_k^T = [S_k^T \;\; I]$ we have $Z_k^T \nabla f_k^T = 0$. Linearizing the reformulated KKT conditions with respect to u^k gives the Newton system

$$\left[\begin{array}{c} Z_k^T \nabla^2 f_k^T + \nabla f_k \nabla Z_k^T \\ \nabla_{u^k} c_{\mathcal{E}}^k \end{array}\right] d_k^u = -\left[\begin{array}{c} \nabla_{u^k} f_k + S_k^T \nabla_{x^k} f_k \\ c_{\mathcal{E}}^k \end{array}\right]$$

We observe that the following qp in $d_k^u \in \mathbb{R}^{n_u M}$ has the same KKT conditions (except for the term $\nabla f_k \nabla Z_k^T$)

$$\begin{array}{c} \min_{d_k^u \in \mathbb{R}^{n_u M}} \left(\nabla_{u^k} f_k^T + S_k \nabla_{x^k} f_k^T\right) d_k^u + \frac{1}{2} (d_k^u)^T \left(S_k^T \nabla_{x^k}^2 f_k S_k + \nabla_{u^k}^2 f_k\right) d_k^u \\ s.t. \quad c_{\mathcal{E}}^k + \frac{dc_{\mathcal{E}}^k}{du^k} d_k^u = 0 \end{array} \tag{4.5}$$

The bound constraints in d_k^x are restated as

$$\left[\begin{array}{c} S_k \\ -S_k \end{array}\right] d_k^u \leq \left[\begin{array}{c} x_U^k - x^k \\ x^k - x_L^k \end{array}\right]$$

The algorithm evolves the model to get $x^k = \Psi(x_0, u^k)$. Then the algorithm solves the qp (4.5) for d_k^u, giving u^{k+1} which again is used to evolve the model giving x^{k+1} and so on.

Neglecting the second order derivatives of the model, $\nabla f_k \nabla S_k^T$, sacrifices the quadratic convergence of Newtons method, but it gives a positive definite Hessian. This is also known as the Gauss-Newton method, which will deteriorate to linear convergence if the projected contributions of the model

[2]This definition of S gives the same result as that of de Oliveira (1994), but is faster to compute in MATLAB which benefits from vectorization. Note that the inversion is not necessary to numerically solve for S from $AS = B$.

are significant (Biegler 2000). Since the Hessian $H\mathcal{L}^*$ is positive definite, this means that we will have linear convergence (possibly to a saddle or maximum point of the original system) of a convex and small system due to the assumption on $n_u M$. We also note that we will have quadratic convergence of (4.5) near the solution, if there are no active bounds at the solution. We observe from (4.4) that it is not meaningful to consider full-space sequential methods.

Reduced Hessian approach (rSQP)

In the null-space method a decomposition is applied to the KKT conditions to eliminate variables. Consider the SQP subproblem which at an iterate k generates a search direction d_k by solving

$$\begin{array}{c} \min_{d_k \in \mathbb{R}^{(n_x P + n_u M)}} \; g_k^T d_k + \frac{1}{2} d_k^T W\,(x_k, u_k)\, d_k \\ s.t.\;\; c_{\mathcal{E}}(x,u;k) + A(x,u;k)^T d_k = 0 \\ x^k \in \mathcal{X} \\ u^k \in \mathcal{U} \end{array} \tag{4.6}$$

Here $\nabla f_k = g_k$, W_k (usually) is a positive definite approximation of $H\mathcal{L}_k$. Where $H\mathcal{L}_k$ is the Hessian of the Lagrangian function $\mathcal{L}(x^k, u^k, \lambda_{\mathcal{E}}^k) = f_k(x^k, u^k) + \lambda_{\mathcal{E}}^{k,T} c_{\mathcal{E}}^k(x^k, u^k)$ evaluated at $(x^*, u^*, \lambda_{\mathcal{E}}^*)$. $A(x,u;k) = \nabla c_{\mathcal{E}}^k$ denotes the constraint Jacobian. The next iterate is computed as $(x_{k+1}, u_{k+1}) = (x_k, u_k) + \alpha_k (d_k^x, d_k^u)$ where α_k is the step length parameter chosen to reduce a suitable merit function. Denote $A(x,u;k) = A_k$ etc.

We partition $(x^k, u^k) \in \mathbb{R}^{(n_x P + n_u M)}$ into state and control variables through the basis given by a nonsingular matrix $[Y_k \;\; Z_k]$. This allows the representation of the search vector as $d_k = Y_k p_{Y,k} + Z_k p_{Z,k}$. Assume that Z_k is a basis for $\mathcal{N}(A_k^T)$, i.e. $A_k^T Z_k = 0$, and that Y_k is a basis for $\mathcal{R}(A_k^T)$. Hence, we decompose d_k into a range and null-space component. The model constraint from problem (4.6) can now be rewritten as $c_{\mathcal{E}}^k + A_k^T Y_k p_{Y,k} = 0$. Since $[Y_k \;\; Z_k]$ is non-singular, assuming full column rank of A_k leads to $p_{Y,k} = -\left(A_k^T Y_k\right)^{-1} c_{\mathcal{E}}^k$ which gives us $d_k = -Y_k \left(A_k^T Y_k\right)^{-1} c_{\mathcal{E}}^k + Z_k p_{Z,k}$. We arrive at the reduced size SQP subproblem (considering $p_{Y,k}$ as a constant)

$$\begin{array}{c} \min_{p_{Z,k} \in \mathbb{R}^{n_u M}} \; (Z_k^T g_k + Z_k^T W_k Y_k p_{Y,k})^T p_{Z,k} + \frac{1}{2} p_{Z,k}^T Z_k^T W_k Z_k p_{Z,k} \\ x^k \in \mathbb{X} \times \cdots \times \mathbb{X} \\ u^k \in \mathbb{U} \times \cdots \times \mathbb{U} \end{array} \tag{4.7}$$

The choice of Y_k and Z_k is motivated by the partitioning into dependent and free variables. Biegler et al. (1995) argues that the partitioning $A_k^T =$

$[C_k \quad N_k]$ with

$$Z_k = \begin{bmatrix} -C_k^{-1}N_k \\ I \end{bmatrix} \qquad Y_k = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{4.8}$$

should be utilized, assuming a non-singular C_k and $A_k^T Z_k = 0$. If we choose the natural partitioning $[C_k \quad N_k] = \left[\left(\frac{\partial c_\mathcal{E}^k}{\partial x^k}\right)^T \quad \left(\frac{\partial c_\mathcal{E}^k}{\partial u^k}\right)^T\right]$ arising from linearization with respect to (x^k, u^k), this leads to the following relations

$$Z_k = \begin{bmatrix} -C_k^{-1}N_k \\ I \end{bmatrix} = \begin{bmatrix} S_k \\ I \end{bmatrix}$$
$$d_k^x = p_{Y,k} + S_k p_{Z,k}$$
$$d_k^u = p_{Z,k}$$

Summarizing, the search direction in `rSQP` has an added range-space component $p_{Y,k}$ which is not present in the `sSQP` method. Inserting $p_{Y,k} = 0$ into (4.7) and comparing with (4.6), we observe that the search directions for the `sSQP` approach coincide with the `rSQP` approach for the choice

$$W_k = \begin{bmatrix} \nabla^2_{xx} f_k & \\ & \nabla^2_{uu} f_k \end{bmatrix}$$

and with $[C_k \;\; N_k]$ selected from the linearization. The step $p_{Z,k}$ is in the null space of A_k, i.e. it is tangential to the constraints, while $p_{Y,k}$ is in the range space of A_k. Then, informally $p_{Z,k}$ aims at reducing the objective while $p_{Y,k}$ searches for feasibility. Since the `sSQP` method is a feasible path method, $p_{Y,k}$ is not needed.

In comparing the two approaches we observe that the sequential approach maintains feasibility of all iterates, while `rSQP` searches for feasibility and optimality simultaneously. In addition the sequential method solves the model at each iteration, while `rSQP` only solves the model once. We also observe that the sequential method does not utilize `BFGS` updates, while this must be utilized in `rSQP` since we do not assume available second order derivatives. Recall that we neglected the cross-term involving second order model derivatives in equation (4.5). We also note that the sequential approach only handles initial value problems (`IVP`), i.e. $x_P \in \Omega_x$ cannot be guaranteed since it implements a shooting strategy in evolving the model over the horizon. The endpoint constraint changes the problem into a boundary value problem (`BVP`) which must be handled by simultaneous strategies. Therefore the sequential approach is limited to open-loop stable

and non-stiff systems. In fact, the stability of the algorithm requires an infinite prediction horizon which is intractable unless the step length can be increased to infinity, see de Oliveira (1994) and Chen and Allgöwer (1998a) for details. We also observe that the reduction of the size of the `qp` to be solved, introduces an additional cost of calculating S_k (`sSQP`) and Z_k (`rSQP`) respectively.

Additional equality (or active inequality) constraints reduce the available degrees of freedom, and consequently a manual decomposition of the Jacobian $A_k = [C_k \; N_k]$ as indicated in section 4.3 is not recommended. There is a number of commercial routines available for both dense and sparse algebra that can be applied. Decomposition strategies for sparse matrices are implemented in the Harwell subroutine libraries `MA28` and `MA48` (Harwell Laboratory 1995). We observe that in addition to assuming stability, we must assume non-singularity of $\nabla_{x^k} c_{\mathcal{E}}^k$ and C_k. Note that these are sparse lower triangular matrices of order $n_x P$. Hence, solving for S_k (`sSQP`) and Z_k (`rSQP`) is cheap i.e. of order $O\left((n_x P)^p\right)$, with $p = 2$ instead of $p = 3$ due to sparsity. Since the matrix is both sparse and lower triangular $p \in [1, 2]$ could be approached by a proper strategy. Recall that the efficiency of reduced gradient algorithms relies on the assumption that $n_x P - n_u M >> n_u M$ and that $n_u M$ is small. Alternatives to computing the Jacobian by analytic partial derivatives is by perturbation or by forward differences. This will have a significant impact on the computational demands, see the results in section 4.3.2.

4.3 Simulations

We implemented `NMPC` on two cases with three different optimization methods. The first is a basic full space `SQP` method following algorithm 4.1. The second is the reduced Hessian method `rFSQP,` cf. algorithm 3.1, and the third is the sequential method (`sSQP`). The first case is a `CSTR` while the second case is grate sintering from chapter 2. The `CSTR` example was thoroughly explored by application of various discretization methods and finite difference approximations to the Jacobian. For the `CSTR` there are no nonlinear inequality constraints, and `rFSQP` switches to nonlinear equality constrained mode for this case, i.e. it is equivalent to a `rSQP` method for this case.

4.3.1 Implementation issues

The basic SQP full-space method was implemented with the common l_1-penalty function. The rFSQP algorithm uses sparse linear algebra and, since rFSQP always remain feasible with respect to inequalities, uses an l_1-penalty function without penalization of inequality constraints. sSQP implemented an l_1-penalty function without penalization of equality constraints, since sSQP always remain feasible with respect to equalities. The different penalty functions are

$$\begin{gathered}\phi_{basic}(x_k) = f(x_k) + \nu_k \left\| c_{\mathcal{E}}(x_k) \right\|_1 + \nu_k \max\left\{ c_{\mathcal{I}}(x_k), 0 \right\} \\ \phi_{rFSQP}(x_k) = f(x_k) + \nu_k \left\| c_{\mathcal{E}}(x_k) \right\|_1 \\ \phi_{sSQP}(x_k) = f(x_k) + \nu_k \max\left\{ c_{\mathcal{I}}(x), 0 \right\}\end{gathered}$$

where $c_{\mathcal{E}}(x_k)$ is nonlinear equality constraints, and $c_{\mathcal{I}}(x_k)$ is nonlinear inequality constraints. The line search for all methods is backtracking line search (Dennis and Schnabel 1996).

The update rule for the penalty parameter is (Biegler, Nocedal, Schmid, and Ternet 2000):

$$\nu_{k+1} = \max([1.001 + \left\| \lambda_{k+1} \right\|_\infty, (3\nu_k + \left\| \lambda_{k+1} \right\|_\infty)/4, \nu_0])$$

where λ_{k+1} is an estimate of the Lagrange multipliers.

The relaxed convergence criteria from Gill, Murray, and Wright (1981), section 8.2.3, was implemented with tolerance 10^{-5} for the basic SQP method and sSQP. In rFSQP the algorithm stops whenever a certain KKT measure is decreased below the tolerance 10^{-5}. Hence, the rFSQP algorithm does not implement the relaxed termination criteria of Gill et al. (1981). The implementation of rFSQP is generally more carefully performed than the basic SQP and sSQP methods. Hence, the relaxed termination criteria used in basic SQP and sSQP partly compensates for a rudimentary implementation. However, as the discussion in section 4.2 indicates, the sSQP method may show linear convergence in certain circumstances, and relaxed termination criteria can therefore be of crucial importance in production codes as well. For the CSTR case the model was discretized with explicit and implicit Euler, Lobatto IIIC and ordinary Runge-Kutta 4. The Jacobian matrices with almost block-diagonal (ABD) structure for the selected discretization methods are given in appendix C. The CSTR case was implemented with both analytic Jacobian and finite difference approximations of the Jacobian. Finite differences were considered for both the full Jacobian (a dense matrix) and the elements along the block diagonal (a sparse matrix). In sSQP analytic and finite difference Jacobians were implemented as documented in

appendix C.4. I.e. the sensitivity matrix $\mathcal{S}$ was approximated directly by finite difference perturbations of the simulator. The `CSTR` case was investigated with different sampling rates and prediction and move horizons. For the sinter case only `sSQP` using `MATLAB`'s `ode15s` and `rFSQP` using Lobatto IIIC were investigated.

4.3.2 Case1: CSTR

Case 1 is the following isothermal `CSTR` with 1st order reaction from Matsuura and Kato (1967) also investigated by de Oliveira (1994)

$$\begin{aligned} \frac{dx_1}{dt} &= u_1 + u_2 - k_1\sqrt{x_1} \\ \frac{dx_2}{dt} &= (C_{B_1} - x_2)\frac{u_1}{x_1} + (C_{B_2} - x_2)\frac{u_2}{x_1} - \frac{k_2 x_2}{(1+x_2)^2} \end{aligned} \tag{4.9}$$

with parameter values $k_1 = 0.2$, $k_2 = 1$, $C_{B_1} = 24.9$ and $C_{B_2} = 0.1$. For $(u_1, u_2) = (1, 1)$ the `CSTR` has three equilibrium points at $x_1 = 100$, $x_2 \in (0.633, 2.72, 7.07)$, with the middle equilibrium point being unstable, and the others stable. The system (4.9) was discretized with a time step h, prediction horizon P, move horizon M and simulated for N_{MPC} samples, i.e. the `NMPC` problem is repeatedly solved N_{MPC} times. At time step 10 the process experiences a +50% step in C_{B_1} which is seen by the `NMPC` algorithm through the feedback only. We choose $Q = 10I_{n_x}$ and $R = I_{n_u}$ in equation (4.1) and penalize deviation from stationary values. I.e. the control objective is to keep the states and controls at their initial values $x = (100, 0.633)$ and $u = (1, 1)$.

We require that the physical bounds $(x_i, u_i) \geq 0$ are maintained over the horizons. The `SQP`-algorithms were initialized with the output from the previous call for each `NMPC` iteration. Note that integral action is not implemented. This is justified by the fact that we are mainly interested in comparing the optimization methods, and we expect that introducing integral action will not influence on this comparison. A representative simulation result is shown in figure (4.1). The process was simulated by `MATLAB`'s `ode45` in all cases.

Note that with the given initial conditions and the given disturbance, this case is pathological with respect to inequality constraints, i.e. the active set is empty and unaltered throughout the horizon P. The control inputs in figure 4.1 are nearly constant after a short transient subsequent to the parameter disturbance. Observing this typical behavior of the `NMPC` methods, we searched for the optimal control among those at set point until the step appears and constant thereafter. This produced the contours of figure (4.2). The figure caption provides a more detailed description on how the contours

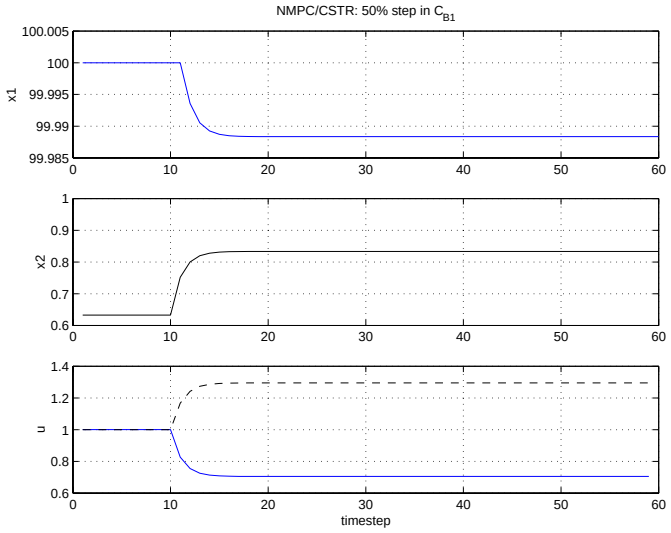

Figure 4.1: *NMPC of CSTR*. The figure shows typical results for `NMPC` of the `CSTR` case with $h = 1$. The various discretization methods produced nearly identical results. In the lower subfigure the solid curve is for u_1 and the dashed curve is for u_2. The step in C_{B_1} enters at time step 10, and this drives the state x_2 away from the equilibrium. If the control inputs are kept at the equilibrium input $[1, 1]$, the system will settle at an equilibrium point at $x = (100, 15.94)$.

were generated. Observe the steep contours on the figure indicating that even minor deviations will influence on the result. Observe also that the contours were generated for the class of constant controls described above, and do not match the contours for the different `NMPC` implementations. The contours are elongated, i.e. the system is poorly conditioned.

The computations were implemented in `MATLAB` with the `qp`-routines available as `mex/dll`-files on a Compaq Deskpro EN / Pentium II / 450MHz / 128Mb RAM running Windows NT4.0. Computational results are shown in tables 4.1-4.3 The results in tables 4.1-4.3 are summarized in the following conclusions:

- `sSQP` is sensitive to the choice between implicit and explicit integration methods, while both basic `SQP` and `r(F)SQP` is insensitive to this.
- Finite difference approximations of the full Jacobian in basic `SQP` and `r(F)SQP` should be avoided.
- `r(F)SQP` is superior for large number of variables.

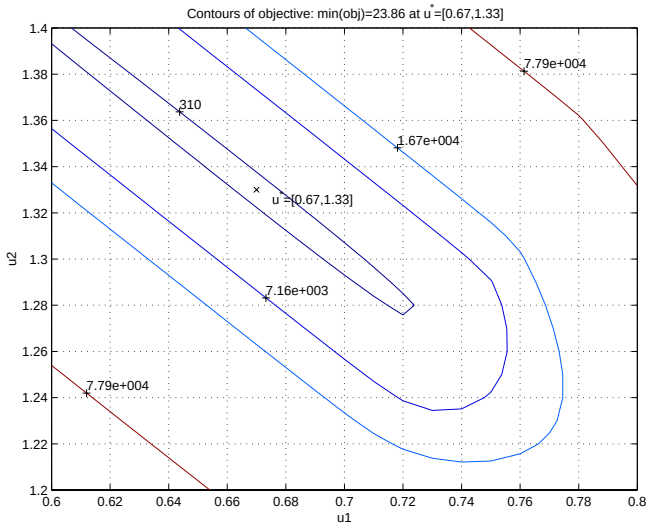

Figure 4.2: *Contours of NMPC of CSTR.* Contours of constant control. The control u was fixed at the stationary value $u = [1, 1]$ until the step in C_{B_1} entered. Then u was set to another constant value for remaining part of the horizon. The objective minimum 23.6 at $(u_1, u_2) = (0.67, 1.33)$ is marked by a $\times$. Note that the optimum is located at the floor of a narrow valley with elongated contours. The condition number is $9 \cdot 10^{10}$.

In the list above the notation `r(F)SQP` emphasizes that in this case, `rFSQP` is equivalent to `rSQP`. Note that a consequence of the first item is that simultaneous `SQP` methods can have fewer variables when implemented with implicit discretization methods. I.e., the step length h can be increased beyond the stability limit of explicit methods (but not beyond reasonable accuracy). Since the basic `SQP` and `sSQP` methods are approximately similar in implementation complexity, `sSQP` should be chosen when explicit discretization schemes suffice. In the face of more challenging processes reduced Hessian methods are superior provided that the assumption that there are few degrees of freedom continue to hold. This assumption commonly holds in `NMPC`. Note that none of the problems reported in tables 4.1-4.3 could be solved by the `MATLAB NAG` toolbox routine `e04ucf`. The `CSTR` example has also been solved by de Oliveira (1994), but there the focus is on viability of the `sSQP` method and no comparisons to other approaches are given.

As a preparation for section 4.3.4, some additional simulations were performed. The consequence of terminating `sSQP` early (after one `SQP` iteration) was investigated, since it is of interest to see whether this significantly im-

Discretization method	Horizons $h/P/M$	# vars. tot/dep/free	Jacobian analytic/fd1/fd2	Results	
				Obj	CPU time
Explicit Euler	2/6/5	22/12/10	analytic	33.55	5.9s
Explicit Euler	1/12/10	44/24/20	analytic	28.33	19s
Implicit Euler	2/6/5	22/12/10	analytic	31.41	9.5s
Implicit Euler	1/12/10	44/24/20	analytic	27.85	45s
Lobatto IIIC	2/6/5	34/24/10	analytic	32.37	7.3s
Lobatto IIIC	2/6/5	34/24/10	fd1	32.35	22s
Lobatto IIIC	2/6/5	34/24/10	fd2	32.35	25s
Lobatto IIIC	1/12/10	68/48/20	analytic	28.07	35s
Lobatto IIIC	1/12/10	68/48/20	fd1	28.06	96s
Lobatto IIIC	1/12/10	68/48/20	fd2	28.06	222s
RK4	2/6/5	70/60/10	analytic	32.40	14.1s
RK4	2/6/5	70/60/10	fd1	32.39	30s
RK4	2/6/5	70/60/10	fd2	32.40	117s
RK4	1/12/10	140/120/20	analytic	28.05	119s
RK4	1/12/10	140/120/20	fd1	28.04	338s
RK4	1/12/10	140/120/20	fd2	28.05	1136s

Table 4.1: *Nonlinear MPC on a CSTR: Basic SQP.* The table shows results for the basic SQP method. $h/P/M$ are the sampling time, prediction and move horizons. The # vars. tot/dep/free are the total, dependent and free number of variables. Note that there are no active inequality constraints in this case. The Jacobian is either analytic or approximated by finite differences. In mode fd1 only elements along the block diagonal were approximated, while in mode fd2 the full (dense) Jacobian was approximated. Obj is the objective value measured by summing the actual prosess outputs and implemented control actions over the NMPC horizon N_{MPC}. CPU time is the time measured by MATLAB's cputime command from start to end of the main NMPC loop.

pairs the performance. On the CSTR case terminating after 1 SQP iteration gave nearly identical results as compared to figure 4.1 and table 4.1. For the CSTR case the sSQP algorithm always terminates after 3-4 iterations. The similarity between early termination and normal termination is explained by that the first step gets very close to the optimum, and the following steps only provide minor improvements to the first step. However, early termination is problem dependent and for the sintering case other results are achieved, see the discussion of figure 4.5 and 4.6 in section 4.3.4.

Discretization method	Horizons $h/P/M$	# vars. tot/dep/free	Jacobian analytic/fd	Results	
				Obj	CPU time
Explicit Euler	2/6/5	22/12/10	analytic	33.18	4.4s
Explicit Euler	2/6/5	22/12/10	fd	33.54	5.7s
Explicit Euler	1/12/10	44/24/20	analytic	28.15	13.1s
Explicit Euler	1/12/10	44/24/20	fd	28.33	19.4s
Implicit Euler	2/6/5	22/12/10	analytic	31.41	33s
Implicit Euler	2/6/5	22/12/10	fd	31.41	131s
Implicit Euler	1/12/10	44/24/20	analytic	27.85	94s
Implicit Euler	1/12/10	44/24/20	fd	27.83	745s
Lobatto IIIC	2/6/5	34/24/10	analytic	32.37	49s
Lobatto IIIC	2/6/5	34/24/10	fd	32.36	192s
Lobatto IIIC	1/12/10	68/48/20	analytic	28.07	118s
Lobatto IIIC	1/12/10	68/48/20	fd	28.07	1121s
RK4	2/6/5	70/60/10	analytic	32.40	4.3s
RK4	2/6/5	70/60/10	fd	32.40	10.0s
RK4	1/12/10	140/120/20	analytic	28.05	12.2s
RK4	1/12/10	140/120/20	fd	28.05	58s
ode45	2/6/5	22/12/10	fd	23.03	36s
ode45	1/12/10	44/24/20	fd	28.49	149s

Table 4.2: *Nonlinear MPC on a CSTR: sSQP.* The table shows results for the `sSQP` method. The results for ode45 with $h = 2$ showed some ripple that were caused by the relaxed termination criteria. See table 4.1 for explanation of symbols.

4.3.3 Large scale applications of `rFSQP`

The `rFSQP` algorithm was tested on (pathological) large problems by decreasing the sampling time h to give very long prediction horizons in the `CSTR` case. The move horizon was limited to 20 in these simulations in order to limit the size of the constraint Jacobian for the `qp`[3].

The `rFSQP` algorithm was tested on the `CSTR` case with 40000 dependent and 20 free variables. This was done by choosing a very small step length and adjusting the prediction and move horizons accordingly. The task manager image in figure 4.3 shows that the system is running close to its limits. In particular, the lack of physical memory is critical. Attempts to run `rFSQP` with 100000 variables failed with an "unhandled exception" message from Windows.

[3]The `NAG` toolbox routine `e04naf` needs the Jacobian as a dense matrix.

Discretization method	Horizons $h/P/M$	# vars. tot/dep/free	Jacobian analytic/fd1/fd2	Results	
				Obj	CPU time
Explicit Euler	2/6/5	22/12/10	analytic	33.57	6.1s
Explicit Euler	1/12/10	44/24/10	analytic	28.27	13.4s
Implicit Euler	2/6/5	22/12/10	analytic	31.43	6.2s
Implicit Euler	1/12/10	44/24/10	analytic	27.99	12.8s
Lobatto IIIC	2/6/5	34/24/10	analytic	32.37	7.1s
Lobatto IIIC	2/6/5	34/24/10	fd1	32.35	11.5s
Lobatto IIIC	2/6/5	34/24/10	fd2	32.35	17.3s
Lobatto IIIC	1/12/10	68/48/20	analytic	28.07	17.2s
Lobatto IIIC	1/12/10	68/48/20	fd1	28.05	39s
Lobatto IIIC	1/12/10	68/48/20	fd2	28.06	95s
RK4	2/6/5	70/60/10	analytic	32.40	9.1s
RK4	2/6/5	70/60/10	fd1	32.39	19s
RK4	2/6/5	70/60/10	fd2	32.40	71s
RK4	1/12/10	140/120/20	analytic	28.05	23s
RK4	1/12/10	140/120/20	fd1	28.04	58s
RK4	1/12/10	140/120/20	fd2	28.05	781s

Table 4.3: *Nonlinear MPC on a CSTR: rFSQP.* The table shows results for the `rFSQP` method. See table 4.1 for explanation of symbols.

The text below is a copy of the text output on screen during the first 90 iterations. The algorithm was interrupted after this, and the results are plotted in figure 4.4. The results show that the algorithm works well, but machine limitations limits application to large scale problems.

The `rFSQP` algorithm has an option that allows direct call of the subroutine `MA28BD` if a pivoting sequence from `MA28AD` is given. This reduces the computational burden in `NMPC` where the `SQP` algorithm is called repeatedly, since complete refactorization is not necessary at each call from the outer `NMPC` algorithm. From the printout below it is observed that each `SQP` iteration in the beginning takes about 45 seconds. This includes initialization, `LU` decomposition by `MA28BD`, calculation of the range and null space steps and a convergence test. However, disk swapping adds considerably to computational times and with more physical memory, the computational time will decrease.

It is important to note that inadequate memory handling in `MATLAB` causes memory leaks in the program, i.e. when `MATLAB` enters a subfunction it allocates memory to the internal variables. Upon exit, the variables are

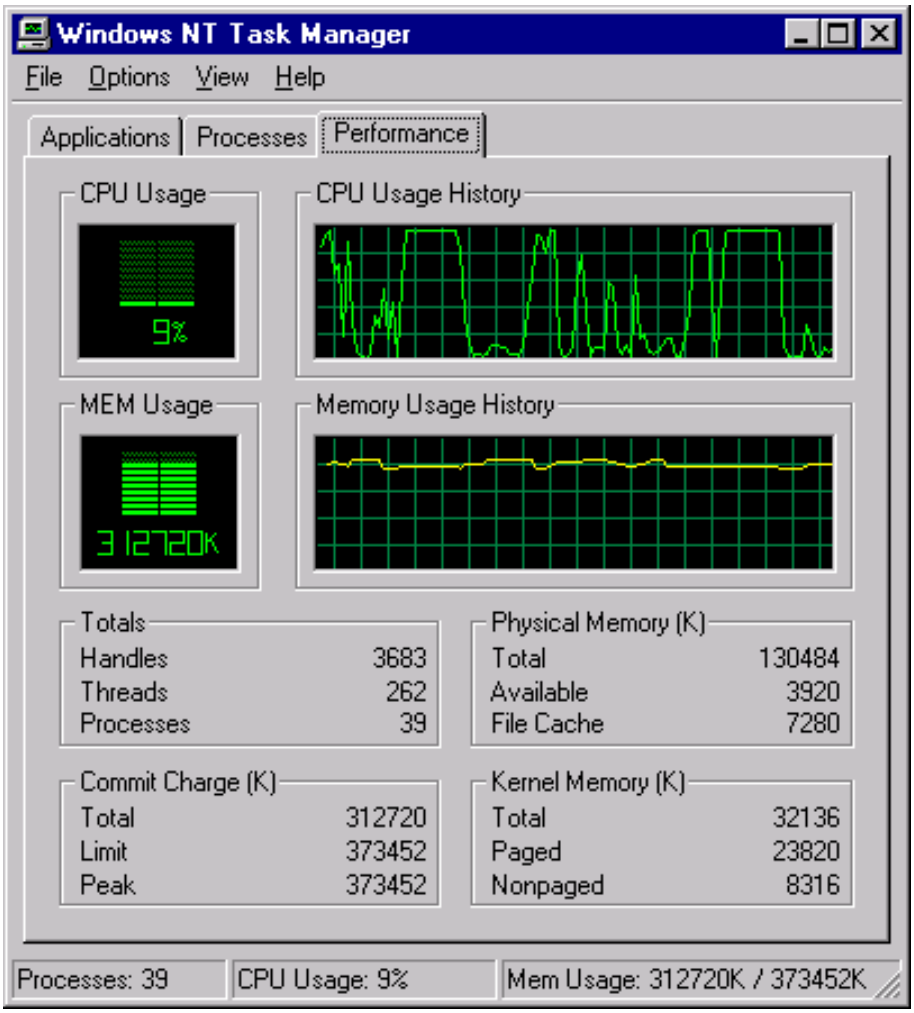

Figure 4.3: *Task manager crop image.* The image shows the task manager when running `rFSQP` on the `CSTR` case with 40000 dependent variables. As can be seen all physical memory is occupied, and the page file is almost fully occupied. The `CPU` load is very low due to exessive disk swapping.

deleted but memory does not seem to be completely deallocated. Hence, each time `MATLAB` calls a subfunction, it leaks memory up to the size of the local variables. To the best of the authors knowledge, the only solution to this problem is to make all variables global. In `rFSQP` all matrix variables are sparse and global, but vectors are not global. Note that if a subroutine allocates 10 vectors of 100000 variables at each call, `MATLAB` will leak up to

2Mb (in double real) at each call.

Printout from screen:

```
************************************************
* NMPC/CSTR: RK4                               *
* Frode Martinsen, 28/06/01                    *
************************************************
RK4 w/rFSQP (ANALYTIC=1)
Total nos. of vars.:  ntot=40020 (dependent=40000, free=20)
MPC: It:5/11999 (CTOT=3.3e+002s, CTnow=45s.  Last iSQP=1)
MPC: It:10/11999 (CTOT=5.4e+002s, CTnow=41s.  Last iSQP=1)
STEP =1
MPC: It:15/11999 (CTOT=2.6e+003s, CTnow=3.1e+002s.  Last iSQP=4)
MPC: It:20/11999 (CTOT=4.3e+003s, CTnow=3.5e+002s.  Last iSQP=4)
MPC: It:25/11999 (CTOT=8.9e+003s, CTnow=1.1e+003s.  Last iSQP=4)
MPC: It:30/11999 (CTOT=1.4e+004s, CTnow=1.3e+003s.  Last iSQP=4)
MPC: It:35/11999 (CTOT=1.6e+004s, CTnow=3.7e+002s.  Last iSQP=4)
MPC: It:40/11999 (CTOT=1.8e+004s, CTnow=5.1e+002s.  Last iSQP=4)
MPC: It:45/11999 (CTOT=2e+004s, CTnow=4e+002s.  Last iSQP=4)
MPC: It:50/11999 (CTOT=2.2e+004s, CTnow=3.6e+002s.  Last iSQP=4)
MPC: It:55/11999 (CTOT=2.4e+004s, CTnow=3.5e+002s.  Last iSQP=4)
MPC: It:60/11999 (CTOT=2.6e+004s, CTnow=3.3e+002s.  Last iSQP=4)
MPC: It:65/11999 (CTOT=2.8e+004s, CTnow=3.7e+002s.  Last iSQP=4)
MPC: It:70/11999 (CTOT=2.9e+004s, CTnow=3.4e+002s.  Last iSQP=4)
MPC: It:75/11999 (CTOT=3.1e+004s, CTnow=3.9e+002s.  Last iSQP=4)
MPC: It:80/11999 (CTOT=3.3e+004s, CTnow=3.7e+002s.  Last iSQP=4)
MPC: It:85/11999 (CTOT=3.5e+004s, CTnow=4.1e+002s.  Last iSQP=4)
MPC: It:90/11999 (CTOT=3.7e+004s, CTnow=5.7e+002s.  Last iSQP=4)
```

4.3.4 Case2: Grate sintering

The sintering process from chapter 2 is a batch process. Hence, stability of the nonlinear MPC algorithm with the objective (2.6) is not essential to the performance of the algorithm. The limited accuracy of the model, cf. section 2.7, indicates that analyzing nominal stability of the nonlinear MPC algorithm is of little value. Hence, checking whether the suggested objective (2.6) is a Lyapunov function is not considered. Due to the problem defined

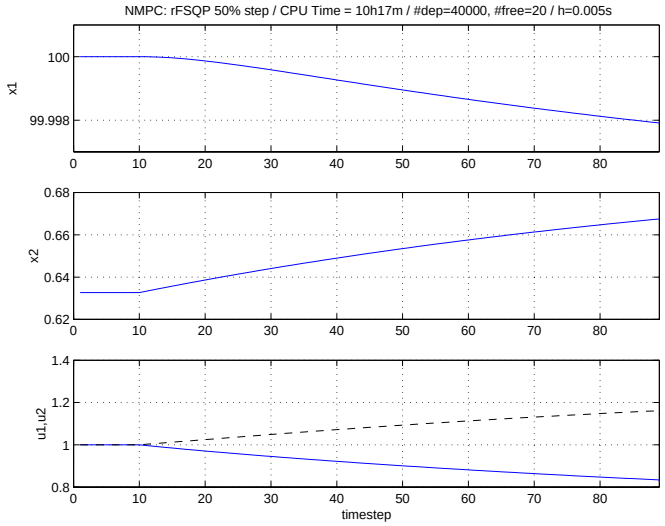

Figure 4.4: *rFSQP on large scale CSTR.* The figure shows the results from rFSQP after interruption at NMPC iteration 90. Comparing with figure 4.1, the sampling period $h = 0.005$ used in this figure implies that 200 time steps here corresponds to 1 time step in figure 4.1. I.e. the 80 datapoints from the step enters in this figure correponds to about a half time step in figure 4.1. Hence, the state and control values cannot be visually distinguished in the two figures, and the results are concluded to be good. Note that the step enters at time step 10 in both figures which is not the same physical time. Physical time is given by $h \cdot timestep$.

in equation (2.6), this is a large scale problem with a nonlinear inequality constraint and a large number of equality constraints.

The reduced model defined in section 2.5.2 was simulated with the implicit MATLAB solver ode15s in section 2.5. Attempts to simulate the model with explicit Runge-Kutta of 4th order (RK4) failed due to numerical instabilities (recall the condition number of $\sim 10^{14}$ for the constraint Jacobian $\frac{\partial f}{\partial x}$ at the initial point). In nonlinear MPC with rSQP and the model discretized by RK4, the algorithm spent about 90 seconds in forming the initial $[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial u}]$ of size $P \cdot 5n_x \times (P \cdot 5n_x + M \cdot n_u) = 11600 \times 11640 \approx 270 Mbyte$ (as a sparse matrix!) with $n_x = 29$ and $n_u = 1$. The horizons $P = 80$ and $M = 40$ were selected to give a prediction horizon of 40 minutes with a sampling rate $h = 30s$. Note that the reduced Hessian is of size 40×40, and approximating this by a dense BFGS matrix presents no limitations.

The algorithm spent about 130 seconds in LU-decomposition inside MA28AD[4].

[4]The PC used had only 128Mbyte memory. Virtual memory on the disk was in use,

The computed steps for the first SQP-iteration was in the order $\|p_Y\| \sim 10^{28}$ and $\|p_Z\| \sim 10^{24}$ indicating a large error introduced by inadequate discretization. With the use of Lobatto IIIC, the sintering process could be simulated with a step length $h = 1$. With a 45 minute prediction horizon this leads to a problem with approximately 2*78300 dependent variables[5]. A sparse Jacobian for this case requires about 3.2Mb using double real format.

Single shooting (sSQP) for the reduced sinter model did not converge. Lack of robustness of single shooting is well known, see (Barclay, Gill, and Rosen 1998) and (Ascher, Mattheij, and Russell 1995). The results presented for sSQP in figure 4.5 and 4.6 were obtained by limiting the number of SQP iterations to 5 and 1 respectively. The line search was aborted after 5 iterations in both cases. Note that aborting the line search was done in order to reduce the number of model evolutions to obtain x^k (superscript notation defined above equation (4.1)). Aborting the line search will not guarantee that the step taken gives descent. The number of control breaks were limited to 3 by control action blocking. The sensitivity matrix then has three columns, each corresponding to one control break. Perturbing ode15s to get $\mathcal{S}$ is done by three evolutions, which took about 75 seconds. Note that $\mathcal{S}$ is a narrow and dense lower trapezoidal matrix. The computational times given in the figures are longer than the batch time, and the approach cannot be implemented in the industrial plant. The objective value in figure 4.5 (see text on top) is lower than the number given in figure 4.6.

4.4 Discussion

In this chapter we implemented different MPC strategies on two cases and investigated the computational load and quality. From tables 4.1-4.3 we observe that among the different NMPC methods, the rSQP method seems preferable in view of computational time. In NMPC computational time is limited, and feasibility of intermediate iterates is essential for stability (Mayne, Rawlings, Rao, and Scokaert 2000), (Scokaert, Mayne, and Rawlings 1999). SQP is an adaptive subproblem, i.e. its computational time is not deterministic. Consequently, in NMPC feasible path SQP methods are preferred since they allow termination prior to convergence (Mayne 1997). Such methods must solve the model constraints at each SQP iteration, which may be time consuming if the model is represented with an implicit discretization scheme. Hence, sSQP becomes computationally demanding if implicit dis-

and part of the given times could be due to disk swapping.

[5]The 2* follows from the additional set of variables in Lobatto IIIC.

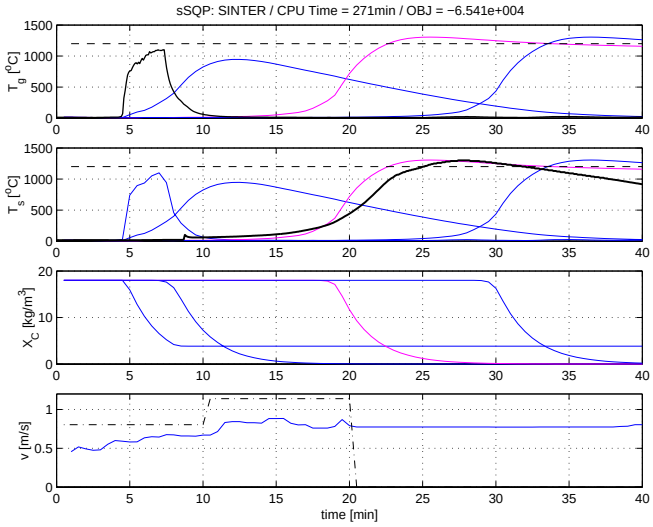

Figure 4.5: *NMPC of reduced sinter model with sSQP, MAXIT=5.* The figure shows the results applying `sSQP` to the reduced sinter model. The full sinter model was used as process simulator. The upper part shows the gas temperatures. The dashed line indicates the fusion temperature. The boldface curve is the measured ignition temperature, while the solid curves show the computed profiles at spatial discretization levels $n = 2, 5, 10$ where $n = 10$ is the bottom level. The second subfigure shows the solid temperature profiles at levels $n = 1, 2, 5, 10$ along with the measured temperature profile (boldface) at 15cm depth, cf. figure 2.11. The dashed line shows the fusion limit. The third subfigure shows the computed coke concentrations at levels $n = 1, 2, 5, 10$. The last subfigure shows the computed control input v (solid). The dashed-dotted line shows the control profile u_k computed for the last iteration of the `NMPC` algorithm, and is only included to show the effect of control action blocking. I.e. the time scale is not relevant for this profile.

cretization methods are applied, whereas `rSQP` preforms equally well regardless of whether implicit or explicit discretization scheme is applied. Both methods require that the selected discretization scheme is appropriate, i.e. if the model cannot be simulated with a given method, it cannot be expected that the optimization algorithms perform well either.

Feasibility with respect to inequality constraints is easier to achieve. The `FSQP` method of Tits and co-workers (Lawrence and Tits 2000), (Lawrence and Tits 1996) and the `rFSQP` method from chapter 3 maintain feasibility with respect to inequality constraints, and *asymptotic* feasibility with respect to nonlinear equality constraints by combining these with an exact penalty function and an arc search. The feasible path `sSQP` method investigated

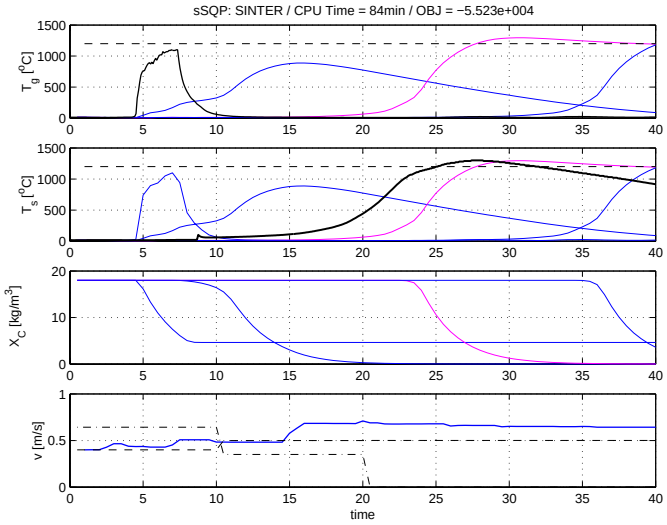

Figure 4.6: *NMPC of reduced sinter model with sSQP, MAXIT=1.* The figure shows the result applying `sSQP` to the reduced sinter model with the maximum number of `SQP` iterations limited to `MAXIT`=1. The subfigures are as in figure 4.5. In the last subfigure the dashed line is computed control at the first `NMPC` iterate. Only the first value is implemented, i.e. the time scale is irrelevant.

in section 4.2.1 does not seem to perform well in the presence of strong nonlinearities. This is contradictory to the needs for `NMPC`; problems with strong nonlinearities and trajectory tracking (Qin and Badgwell 2000).

The large scale example investigated in section 4.3.3 shows that application of `rFSQP` in the grate sintering case is not computationally feasible with the presently available computing equipment. Hence, this has not been solved. Grate sintering is a complex process with pronounced nonlinearities and where the time constants of the system differ by many decades in magnitude. Hence, application of real-time optimization methods like nonlinear model predictive control to the grate sintering process is challenging.

The thesis still contributes to the application of model based control to grate sintering. The thesis gives a framework for implementing nonlinear model based control of grate sintering by giving a control objective. Further, a nonlinear model along with an appropriate discretization scheme is presented. To complete the course of `NMPC` for grate sintering two directions may be investigated; providing a better discretization method or acquiring a better computer. Reimplementing `rFSQP` in `C++` or `FORTRAN` may also be

considered.

4.5 Conclusion

The practical considerations discussed in this chapter explore the choices an engineer must take if he wants to implement `NMPC` at a given process. The interplay between discretization methods and optimization algorithms has been investigated informally.

First the engineer must select an appropriate discretization scheme. If he chooses an explicit discretization scheme, he can choose between sequential or simultaneous optimization methods. Sequential methods are easy to implement, while simultaneous methods are harder to implement. This applies in particular to reduced Hessian methods which may be quite sophisticated. If implicit discretization methods must be applied, the performance of `sSQP` deteriorates while `rSQP` does not degrade.

Chapter 5

Conclusion

5.1 Conclusions to the thesis

Grate sintering

A reduced order model of the grate sintering process was developed from models available in the literature. Both the full model and the reduced model were simulated with the `MATLAB` function `ode15s`, which is an implicit variable step length solver of varying order. Industrial experiments including special measurements were conducted, and the data was used for model adaption and assessing the control objective. The models were adapted by visual comparison to data collected from industrial experiments. Data analysis supports the proposed control objective. The reduced model is less computationally demanding, and reproduces the important process states satisfactorily.

A control objective for use in nonlinear `MPC` was suggested and assessed by data analysis based on industrial data.

In conclusion, the fusion model needs to be refined, the gas velocity must be included as a state and pressure must be used as control input before the model can be used in an industrial model based control setting. Improved numerical schemes must be investigated if the model is to be applied in real-time optimization. The reduced model provides an adequate basis on which to continue.

rFSQP

The optimization algorithm is described, and global and local convergence results are given. The algorithm was implemented in `MATLAB` and tested

on small scale problems. The algorithm is a reduced Hessian method and the algorithm is expected to be efficient for large scale problems with few degrees of freedom.

The algorithm worked well, and it is concluded that the present `MATLAB` implementation serves as a good alternative for small to medium scale problems with, say, up to 10000 variables. A number of possible algorithmic extensions to the algorithm may be implemented and tested on the existing platform. In particular, analytic Hessians and trust-region extensions should be investigated. A watch-dog algorithm should also be implemented.

NMPC

A qualitative comparison between sequential and simultaneous reduced gradient methods is given and illuminates the similarities and differences between these approaches. The comparison was supplemented with simulations on a simple nonlinear `MPC` case, where the interplay between choices among various optimization methods and discretization methods was investigated.

The `rFSQP` algorithm was tested on a simple large scale problem with promising results. However, the computing environment limits testing on very large scale problems with number of variables in the range from 100000 and upwards. Hence, the algorithm was not tested on the grate sintering case which has about 150000 variables when discretized with Lobatto IIIC.

The sequential method `sSQP` was tested on the sintering case with termination prior to convergence. The results are modestly encouraging, since the computational load is excessive. It is concluded that simultaneous optimization algorithms are well suited for problems where explicit discretization methods are inadequate.

The grate sintering case should be implemented with discretization schemes allowing fewer variables, or alternatively ran on a more powerful computer.

`rFSQP` should be reimplemented in `C++` or `FORTRAN` to allow for large scale applications. Further testing on large scale applications, in particular problems with nonlinear inequalities, should be performed.

Bibliography

Allgöwer, F., T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright (1999). Nonlinear Predictive Control and Moving Horizon Estimation–An Introductory Overview. In P. M. Frank (Ed.), *Advances in Control: Highlights of ECC'99*. Springer.

Ascher, U. M., R. M. M. Mattheij, and R. D. Russell (1995). *Numerical solution of boundary value problems for ordinary differential equations*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM). Classics in Applied Mathematics.

Baker, T. E. and E. Polak (1994). On the optimal control of systems described by evolution equations. *SIAM J. Control Optim. 32*(1), 224–260.

Barclay, A., P. E. Gill, and J. B. Rosen (1998). SQP methods and their application to numerical optimal control. In *Variational calculus, optimal control and applications (Trassenheide, 1996)*, pp. 207–222. Basel: Birkhäuser.

Bartlett, R. A., A. Wächter, and L. T. Biegler (2000). Active set vs. interior point strategies for model predictive control. In J. Zhu (Ed.), *Proceedings of the American Control Conference (ACC 2000), Chicago, IL., 2000*, pp. 4229–4233.

Bertsekas, D. P. (1982). *Constrained optimization and Lagrange multiplier methods*. New York: Academic Press Inc. [Harcourt Brace Jovanovich Publishers].

Betts, J. T. and P. D. Frank (1994). A sparse nonlinear optimization algorithm. *J. Optim. Theory Appl. 82*(3), 519–541.

Biegler, L. T. (2000). Efficient solution of dynamic optimization and NMPC problems. In *Nonlinear model predictive control (Ascona, 1998)*, pp. 219–243. Basel: Birkhäuser.

Biegler, L. T., A. M. Cervantes, and A. Wächter (2001). Advances in simultaneous strategies for dynamic process optimization. Technical report, Carnegie Mellon University. Submitted for possible publication.

Biegler, L. T., J. Nocedal, and C. Schmid (1995). A reduced Hessian method for large-scale constrained optimization. *SIAM J. Optim. 5*(2), 314–347.

Biegler, L. T., J. Nocedal, C. Schmid, and D. Ternet (2000). Numerical experience with a reduced Hessian method for large-scale constrained optimization. *Comp. Opt. and Appl. 15*, 45–67.

Bock, H. G., M. M. Diehl, J. P. Schlöder, F. Allgöwer, R. Findeisen, and Z. Nagy (2000). Real-time optimization and nonlinear predictive control of processes governed by differential-algebraic equations. In L. T. Biegler, A. Brambilla, and C. Scali (Eds.), *Prepints: International Symposium on Advanced Control of Chemical Processes (ADCHEM 2000), Pisa, Italy, 2000*, pp. 695–703.

Boggs, P. T. and J. W. Tolle (1995). Sequential quadratic programming. In *Acta numerica*, pp. 1–51. Cambridge: Cambridge Univ. Press.

Bondarenko, A. S., D. M. Bortz, and J. J. Moré (1999). COPS: Large-scale nonlinearly constrained optimization problems. Technical report, Argonne National Laboratory, Argonne, IL. ANL/MCS-TM-237.

Bongartz, I., A. R. Conn, N. I. M. Gould, and P. L. Toint (1995). CUTE: Constrained and unconstrained testing environment. *ACM Trans. Math. Software 21*, 123–160.

Byrd, R. H., J. C. Gilbert, and J. Nocedal (2000). A trust region method based on interior point techniques for nonlinear programming. *Math. Program. 89*(1, Ser. A), 149–185.

Byrd, R. H., M. E. Hribar, and J. Nocedal (1999). An interior point algorithm for large-scale nonlinear programming. *SIAM J. Optim. 9*(4), 877–900 (electronic). Dedicated to John E. Dennis, Jr., on his 60th birthday.

Byrd, R. H., G. Liu, and J. Nocedal (1998). On the local behaviour of an interior point method for nonlinear programming. In *Numerical analysis 1997 (Dundee)*, pp. 37–56. Harlow: Longman.

Byrd, R. H. and J. Nocedal (1989). A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM J. Numer. Anal. 26*(3), 727–739.

Byrd, R. H. and J. Nocedal (1991). An analysis of reduced Hessian methods for constrained optimization. *Math. Programming 49*(3 (Ser. A)), 285–323.

Byrd, R. H., J. Nocedal, and R. A. Waltz (2000). Feasible interior methods using slacks for nonlinear optimization. Technical report, Optimization Technology Center, Northwestern University. OTC 2000/11.

Cervantes, A. M. and L. T. Biegler (2000). A stable elemental decomposition for dynamic process optimization. *J. Comput. Appl. Math. 120*(1-2), 41–57. SQP-based direct discretization methods for practical optimal control problems.

Chamberlain, R. M., M. J. D. Powell, C. Lemarechal, and H. C. Pedersen (1982). The watchdog technique for forcing convergence in algorithms for constrained optimization. *Math. Programming Stud.* (16), 1–17. Algorithms for constrained minimization of smooth nonlinear functions.

Chen, H. and F. Allgöwer (1998a). A computationally attractive nonlinear model predictive control scheme with guaranteed stability for stable systems. *J. Proc. Cont. 8*(5-6), 475–485.

Chen, H. and F. Allgöwer (1998b). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica J. IFAC 34*(10), 1205–1217.

Coleman, T. F. and A. R. Conn (1982a). Nonlinear programming via an exact penalty function: asymptotic analysis. *Math. Programming 24*(2), 123–136.

Coleman, T. F. and A. R. Conn (1982b). Nonlinear programming via an exact penalty function: global analysis. *Math. Programming 24*(2), 137–161.

Conn, A. R., N. I. M. Gould, and P. L. Toint (1992). *LANCELOT*. Berlin: Springer-Verlag. A Fortran package for large-scale nonlinear optimization (release A).

Cumming, M. J. and J. A. Thurlby (1990). Developments in modelling and simulation of iron ore sintering. *Ironmak. Steelmak. 17*(4), 245–254.

Dash, I. and E. Rose (1977). An analytical study of processes occurring in an iron ore sinter bed. In F. Lancaster (Ed.), *Automation in Mining, Mineral and Metal Processing*, IFAC Proc., pp. 649–659.

Dawson, P. R. (1993). Part 2 Research studies on sintering and sinter quality. *Ironmak. Steelmak. 20*(2), 137–143.

de Oliveira, N. M. C. (1994). *Newton-type algorithms for nonlinear constrained chemical process control*. Ph. D. thesis, Carnegie Mellon University, Pittsburgh, PA.

de Oliveira, N. M. C. and L. T. Biegler (1995). An extension of Newton-type algorithms for nonlinear process control. *Automatica J. IFAC 31*(2), 281–286.

Dennis, Jr., J. E. and R. B. Schnabel (1996). *Numerical methods for unconstrained optimization and nonlinear equations*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM). Corrected reprint of the 1983 original.

Dougherty, E. R. (1990). *Probability and Statistics for the Engineering, Computing and Physical Sciences*. Englewood Cliffs, NJ: Prentice Hall.

Drud, A. (1985). CONOPT: a GRG code for large sparse dynamic nonlinear optimization problems. *Math. Programming 31*(2), 153–191.

Edgar, T. F. and D. M. Himmelblau (1989). *Optimization of chemical processes*. New York, NY: McGraw-Hill.

Efron, B. and R. Tibshirani (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statist. Sci. 1*(1), 54–77. With a comment by J. A. Hartigan and a rejoinder by the authors.

Fiacco, A. V. and G. P. McCormick (1990). *Nonlinear programming* (Second ed.). Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM). Sequential unconstrained minimization techniques.

Finlayson, B. A. (1980). *Nonlinear analysis in chemical engineering*. Chemical engineering. New York, NY: McGraw-Hill.

Fletcher, R. (1987). *Practical methods of optimization* (Second ed.). Chichester: John Wiley & Sons Ltd.

Fletcher, R. and S. Leyffer (1998). Nonlinear programming without a penalty function. Technical report, Department of Mathematics and Computer Science. The University of Dundee. NA/171.

Forsgren, A. and P. E. Gill (1998). Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. Optim. 8*(4), 1132–1152 (electronic).

Frank, M. and P. Wolfe (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly 3*, 95–110.

Frich, R. (1954). General principles of the double gradient method. In *Principles of linear programming*, Chapter 12. Oslo: Inst. of Economics, Univ. of Oslo.

Gay, D. M., M. L. Overton, and M. H. Wright (1998). A primal-dual interior method for nonconvex nonlinear programming. In Y. Yuan (Ed.), *Advances in Nonlinear Programming*, pp. 31–56. Kluwer.

Gill, P. E., W. Murray, and M. A. Saunders (1997). SNOPT: An SQP algorithm for large-scale constrained optimization. Technical report, Dept of EESOR, Stanford University. Report SOL 97-3.

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright (1987). Maintaining *LU* factors of a general sparse matrix. *Linear Algebra Appl. 88/89*, 239–270.

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright (1992). Some theoretical properties of an augmented Lagrangian merit function. In P. Pardalos (Ed.), *Advances in Optimization and Parallel Computing*, pp. 101–128. North Holland.

Gill, P. E., W. Murray, and M. H. Wright (1981). *Practical optimization.* London: Academic Press Inc. [Harcourt Brace Jovanovich Publishers].

Golub, G. H. and C. F. Van Loan (1996). *Matrix computations* (Third ed.). Baltimore, MD: Johns Hopkins University Press.

Harwell Laboratory (1995). *Harwell Subroutine Library Specifications (Release 12).* Oxfordshire, UK.: AEA Technology.

Himmelblau, D. M. (1972). *Applied Nonlinear Programming.* New York, NY: McGraw-Hill.

Hinkley, J., A. Waters, and J. Litster (1994). An Investigation of Preignition Air Flow in Ferrous Sintering. *Int.J.Miner.Proc. 42*(1-2), 37–52.

Hinkley, J., A. Waters, D. O'Dea, and J. Litster (1994). Voidage of Ferrous Sinter Beds: New Measurement Technique and Dependence on Feed Characteristics. *Int.J.Miner.Proc. 41*(1-2), 53–69.

Hock, W. and K. Schittkowski (1981). *Test examples for nonlinear programming codes.* Berlin: Springer-Verlag.

Hoislbauer, F. and C. Jaquemar (1983). Mathematical modelling of heat and mass transfer in iron ore packed bed industrial processes. In R. Lewis, K. Morgan, and B. Schrefler (Eds.), *Num. Met. Heat Transf.*, Volume II, Chapter 19, pp. 485–510. Wiley.

Karmarkar, N. (1984). A new polynominal time algorithm for linear programming. *Combinatorica 4*(4), 373–395.

Kasai, E., J.-i. Yagi, and Y. Omori (1984). A mathematical model of sintering process considering melt-formation and solidification phenomena. In *Proc. Ironmak. Conf.*, Volume 43, pp. 241–249. AIME.

Khalil, H. K. (1996). *Nonlinear systems* (Second ed.). Upper Saddle River, NJ: Prentice Hall.

Kim, Y. and W. Kwon (1998). An application of min-max generalized predictive control to sintering processes. *Control Engn. Practice 6*(8), 999–1007.

Lawrence, C. T. and A. L. Tits (1996). Nonlinear equality constraints in feasible sequential quadratic programming. *Optimization Methods and Software 6*(4), 265–282.

Lawrence, C. T. and A. L. Tits (2000). A computationally efficient feasible sequential quadratic programming algorithm. Technical report, Institute for Systems Research, University of Maryland. TR 98-46, Submitted for possible publication in SIAM J. Optim.

Lewis, F. L. and V. L. Syrmos (1995). *Optimal control* (Second ed.). New York, NY: John Wiley.

Li, W. C. and L. T. Biegler (1988). Process control strategies for constrained nonlinear systems. *Ind. Eng. Chem. Res. 27*, 1421–1433.

Li, W. C. and L. T. Biegler (1989). Multistep, Newton-type control strategies for constrained nonlinear processes. *Chem. Eng. Res. Des. 67*, 562–577.

Li, W. C., L. T. Biegler, C. G. Economou, and M. Morari (1990). A constrained pseudo-Newton control strategy for nonlinear systems. *Computers Chem. Engng. 14*(4/5), 451–468.

Marazzi, M. and J. Nocedal (2000). Feasibility control in nonlinear optimization. Technical report, Optimization Technology Center, Northwestern University, Evanston Il. OTC 2000/04. To appear in Foundations of Computational Mathematics, Cambridge University Press.

Martinsen, F. (1998a). Interior-point methods. In E. W. Jacobsen (Ed.), *Literature for the 8th Nordic Process Control Workshop, Stockholm, Sweden, 1998*, pp. 350–369.

Martinsen, F. (1998b). Temperature profile measurements. Technical report, Dep. of Engneering Cybernetics, NTNU. Unpublished report.

Martinsen, F. and L. T. Biegler (2002). Application of optimization in nonlinear MPC. In *Proceeedings of the 15th IFAC World Congress on Automatic Control*, Barcelona. Submitted.

Martinsen, F. and B. A. Foss (2001). A feasible reduced Hessian SQP method. Technical report, Department of Engineering Cybernetics, NTNU, Norway. ITK-2001-5-W. Submitted for possible publication in SIAM Journal on Optimization.

Martinsen, F., T. A. Johansen, and B. A. Foss (1999). A control relevant dynamic model of grate sintering. In *CCA'99, Kona, HI.* IEEE.

Matsuura, T. and M. Kato (1967). Concentration stability of the isothermal reactor. *Chem. Eng. Sci. 22*, 171–184.

Mayne, D. Q. (1997). Nonlinear model predictive control : An assessment. In J. C. Kantor, C. E. Garcia, and B. Carnahan (Eds.), *CPC-V : Proceedings of the Fifth International Conference on Chemical Process Control, Tahoe City, CA., 1996*, AIChE symposium series ; no. 316, pp. 217–231. CACHE.

Mayne, D. Q. and E. Polak (1976). Feasible directions algorithms for optimization problems with equality and inequality constraints. *Math. Programming 11*, 67–80.

Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. Scokaert (2000). Constrained model predictive control: Stability and optimality. *Automatica J. IFAC 36*(6), 789–814.

Morales, J. L., J. Nocedal, R. A. Waltz, G. Liu, and J.-P. Goux (2001). Assessing the potential of interior methods for nonlinear optimization. Technical report, Optimization Technology Center, Northwestern University. OTC 2001/x.

Moran, M. M. and H. N. Shapiro (1993). *Fundamentals of Engineering Thermodynamics* (Second ed.). New York, NY.: Wiley.

Moré, J. J. and D. J. Thuente (1994). Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Software 20*(3), 286–307.

Muchi, I. and J. Higuchi (1972). Theoretical analysis of sintering operation. *Trans. Iron Steel Inst. Jpn. 12*, 54–63.

Murray, W. (1997). Sequential quadratic programming methods for large-scale problems. *Comput. Optim. Appl. 7*(1), 127–142. Computational issues in high performance software for nonlinear optimization (Capri, 1995).

Murray, W. and F. J. Prieto (1995). A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM J. Optim. 5*(3), 590–640.

Murtagh, B. and M. A. Saunders (1995). MINOS 5.4 User's guide. Technical report, Dept of EESOR, Stanford University. Report SOL 83-20R. Revised 1995.

Nath, N. K., A. J. Da Silva, and N. Chakraborti (1997). Dynamic process modelling of iron ore sintering. *Steel Research 68*(7), 285–292.

Nocedal, J. and S. J. Wright (1999). *Numerical optimization*. New York: Springer-Verlag.

Olsen, S. E. (1997). Jernfremstilling. Kompendium, Metallurgisk institutt, NTNU.

Panier, E. R. and A. L. Tits (1987). A superlinearly convergent feasible method for the solution of inequality constrained optimization problems. *SIAM J. Control Optim. 25*(4), 934–950.

Panier, E. R. and A. L. Tits (1993). On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Math. Programming 59*(2, Ser. A), 261–276.

Parker, A. S. and H. C. Hottel (1936). Combustion rate of carbon. *Ind. Eng. Chem. 28*, 1334–1341.

Patisson, F., J. P. Bellot, and D. Ablitzer (1990). Study of moisture transfer during the strand sintering process. *Metall. Trans. 21B*, 37–47.

Patisson, F., J. P. Bellot, D. Ablitzer, E. Marlière, C. Dulcy, and J. M. Steiler (1991). Mathematical modelling of iron ore sintering process. *Ironmak. Steelmak. 18*(2), 89–95.

Perold, A. (1980). A generalization of the Frank-Wolfe theorem. *Math. Programming 18*, 215–227.

Perry, R. H. and D. Green (1984). *Perry's Chemical Engineers' Handbook* (6 ed.). McGraw-Hill.

Powell, M. J. D. (1978). The convergence of variable metric methods for nonlinearly constrained optimization calculations. In O. Mangasarian, R. Meyer, and S. Robinson (Eds.), *Nonlinear Programming 3*, pp. 27–63. New York: Academic Press.

Pytlak, R. (1999). *Numerical methods for optimal control problems with state constraints*. Lecture notes in mathematics: 1707. Germany: Springer-Verlag.

Pytlak, R. and R. B. Vinter (1998). A feasible directions algorithm for optimal control problems with state and control constraints: convergence analysis. *SIAM J. Control Optim. 36*(6), 1999–2019 (electronic).

Pytlak, R. and R. B. Vinter (1999). Feasible direction algorithm for optimal control problems with state and control constraints: implementation. *J. Optim. Theory Appl. 101*(3), 623–649.

Qin, S. J. and T. A. Badgwell (2000). An overview of nonlinear model predictive control applications. In F. Allgöwer and A. Zheng (Eds.), *Nonlinear model predictive control (Ascona, 1998)*, pp. 128–145. Birkhäuser.

Rao, C. V., S. J. Wright, and J. B. Rawlings (1998). Application of interior-point methods to model predictive control. *J. Optim. Theory Appl. 99*(3), 723–757.

Rawlings, J. B. (2000). Tutorial overview of model predictive control. *Control Systems Magazine 20*(3), 38–52.

Robinson, S. M. (1974). Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms. *Math. Programming 7*, 1–16.

Roine, A. (1997). Outokumpu HSC Chemistry for Windows. User's Guide, v.3.0, Outokumpu Research Oy, Finland.

Rose, E. and I. Dash (1979). An analytical study of factors affecting gas flow in sintering. *Ind. Eng. Chem. Process Des. Dev. 18*(1), 67–72.

Rosenqvist, T. (1983). *Principles of Extractive Metallurgy* (2 ed.). McGraw-Hill.

Schluter, R. and G. Bitsianes (1962). The combustion zone in the iron ore sintering process. In W. Knepper (Ed.), *Agglomeration*, pp. 585–639.

Scokaert, P. O. M., D. Q. Mayne, and J. B. Rawlings (1999, March). Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Auto. Cont 44*(3), 648–654.

Tjoa, I.-B. and L. T. Biegler (1991). Simultaneous solution and optimization strategies for parameter-estimation of differential-algebraic equation systems. *Ind. Eng. Chem. Res. 30*(2), 376–385.

Toda, H., T. Senzaki, S. Isozaki, and K. Kato (1984). Relationship between heat pattern in sintering bed and sinter properties. *Trans. Iron Steel Inst. Jpn. 24*, 187–196.

Troutman, J. L. (1996). *Variational calculus and optimal control* (Second ed.). New York: Springer-Verlag. With the assistance of William Hrusa, Optimization with elementary convexity.

Vanderbei, R. J. and D. F. Shanno (1999). An interior-point algorithm for nonconvex nonlinear programming. *Comput. Optim. Appl. 13*(1-3), 231–252. Computational optimization—a tribute to Olvi Mangasarian, Part II.

Vassiliadis, V. (1993). *Computational solution of dynamic optimization problems with general differential-algebraic constraints.* Ph. D. thesis, University of London, U.K.

Venkataramana, R., S. S. Gupta, P. C. Kapur, and N. Ramachandran (1998). Mathematical modelling and simulation of the iron ore sintering process. *Tata Search*, 25–30.

Verein Deutscher Eisenhüttenleute (VDEh) (Ed.) (1995). *Slag Atlas* (second ed.). Stahleisen, Düsseldorf.

Vinter, R. (2000). *Optimal control.* Boston, MA: Birkhäuser Boston Inc.

Voice, E. W., S. H. Brooks, and P. K. Gledhill (1953). The permeability of sinter beds. *JISI 174*, 136–139.

Wakao, N. and S. Kaguei (1982). *Heat and Mass Transfer in Packed Beds*, Volume 1 of *Topics in Chemical Engineering.* Gordon and Breach Science Publishers.

White, F. M. (1999). *Fluid Mechanics* (Fourth ed.). Mechanical engineering series. New York, NY: McGraw-Hill.

Wonchala, E. and J. Wynnyckyj (1987). Nonisothermal flow of gases through packed beds. *Metall. Trans. B 18B*, 279–280.

Wright, S. J. (1997a). Applying new optimization algorithms to model predictive control. In J. C. Kantor, C. E. Garcia, and B. Carnahan (Eds.), *CPC-V : Proceedings of the Fifth International Conference on Chemical Process Control, Tahoe City, CA., 1996*, Volume 93, pp. 147–155. CACHE.

Wright, S. J. (1997b). *Primal-dual interior-point methods.* Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

Xie, Y. F. and R. H. Byrd (1999). Practical update criteria for reduced Hessian SQP: global analysis. *SIAM J. Optim. 9*(3), 578–604.

Zou, Z., T. Huang, X. Yang, and J. Chen (1995). Mathematical model and computer simulation of moisture transfer process during sintering. *Trans. NFsoc 5*(1), 15–20.

Appendix A

Appendices to chapter 2

A.1 Sinter objective

The objective from equation (2.6) is restated

$$\begin{aligned}\phi &= \int_0^L w(z) \int_0^{t_{end}} \frac{1}{1+e^{k_1(T_s-T_{fu})}}(T_s - T_{fu})dtdz \\ \psi_z &= \int_{t=0}^{t_{end}} \sigma(T_s)(T_s - T_{fu})dt\end{aligned}$$

The double integral is approximated by a sum over the prediction horizon for each spatial element j. Choosing $P \cdot h = t_s$ where h is the sampling time and t_{end} is the sintering batch time. The integrals are approximated by summation:

$$\begin{aligned}\phi &= \sum_{j=1}^{n} w_j \sum_{k=1}^{P} \frac{1}{1+e^{k_1(T^j_{s,k}-T_{fu})}}(T^j_{s,k} - T_{fu}) \\ \psi_j &= \sum_{k=1}^{P} f^j_k(T_s) = \sum_{k=1}^{P} \frac{1}{1+e^{k_1(T^j_{s,k}-T_{fu})}}(T^j_{s,k} - T_{fu}) \quad for\ j = 1, \ldots, n\end{aligned}$$

where n is the spatial resolution and w_j are weights. The kernel f_k for a given j is considered.

$$f_k = \frac{(T_{s,k} - T_{fu})}{1+e^{k_1(T_{s,k}-T_{fu})}}$$

The gradient is

$$g_k = \frac{\partial f_k}{\partial T_{s,k}} = -\frac{k_1(T_{s,k} - T_{fu})e^{k_1(T_{s,k}-T_{fu})}}{\left(1 + e^{k_1(T_{s,k}-T_{fu})}\right)^2} + \frac{1}{1 + e^{k_1(T_{s,k}-T_{fu})}}$$

The Hessian is

$$H_k = \frac{\partial g_k}{\partial T_{s,k}} = \frac{2k_1^2(T_{s,k}-T_{fu})\left(e^{k_1(T_{s,k}-T_{fu})}\right)^2}{\left(1+e^{k_1(T_{s,k}-T_{fu})}\right)^3} - \frac{2k_1 e^{k_1(T_{s,k}-T_{fu})}}{\left(1+e^{k_1(T_{s,k}-T_{fu})}\right)^2} - \frac{k_1^2(T_{s,k}-T_{fu})e^{k_1(T_{s,k}-T_{fu})}}{\left(1+e^{k_1(T_{s,k}-T_{fu})}\right)^2}$$

The variable vector for the reduced sinter model with n spatial stages is $x = [x_1^T, x_2^T, \ldots, x_P^T, u_{0,}^T \ldots, u_M^T]^T$ where

$$x_1 = [T_{g,1}^2, \ldots, T_{g,1}^n, T_{s,1}^1, \ldots, T_{s,1}^j, \ldots, T_{s,1}^n, x_{C,1}^1, \ldots, x_{C,1}^n]^T$$

where the subscript is the time index k and the superscript is the spatial index j. The gradient of ψ_{zj} is formed as $\nabla_{x_1}\psi_j = [0, \ldots, 0, g_1^j, 0, \ldots, 0]^T$ which gives $\nabla_x\psi_j = [0, \ldots, 0, g_1^j, 0, \ldots, 0, g_2^j, 0, \ldots, 0, g_P^j, 0, \ldots, 0]^T$. The objective gradient is

$$\begin{aligned}\nabla_x\phi = \sum_{j=1}^{n} w_j\nabla_x\psi_j = \quad & [0, \ldots, 0, w_1 g_1^1, w_2 g_1^2, \ldots, w_n g_1^n, 0, \ldots \\ & \ldots, 0, w_1 g_2^1, w_2 g_2^2, \ldots, w_n g_2^n, 0, \ldots \\ & \vdots \\ & \ldots, 0, w_1 g_P^1, w_2 g_P^2, \ldots, w_n g_P^n, 0, \ldots, 0]^T\end{aligned}$$

finally

$$\nabla\phi = [\nabla_x\phi^T, \nabla_u\phi^T]^T = [\nabla_x\phi^T, 0, \ldots, 0]^T$$

The objective Hessian is formed as a block diagonal matrix with $\nabla^2\phi_j = w_j H_{k,j}$, $k = 1, \ldots, P$ along the block diagonal. Observe that there are no couplings across spatial discretization boundaries or time indexes.

A.2 Pitot measurements

The pitot measurements are shown in figure A.1. These measurements give an estimate of the gas velocity in the pipe following the cyclones, see figure 2.8. The gas velocity in the sintering pans is upward limited by the gas velocity shown in the figure. Through consideration of the pressure drops across the two pans connected in parallell an estimate of the gasflow through the pan can be achieved. This relies on identification of cracks, channeling effects and pipe leaks. The gas volume flow can be calculated from the pitot measurements. This allows further adaption of the gasflow predicted by the model. This is left as future work.

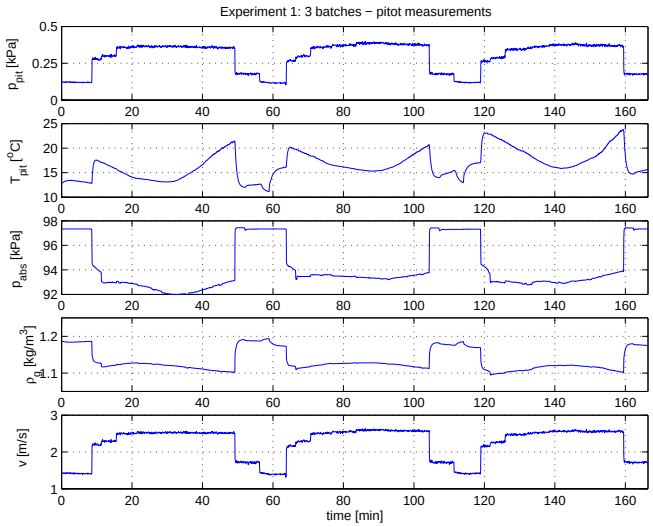

Figure A.1: *Experiment 1: Pitot measurements.* The figure shows measurements placed after the cyclones shown in figure 2.8 during all three batches included in experiment 1. Subfigure 1 shows the pitot pressure measured by a dP-cell connected to a pitot sensor. Subfigure 2 shows the gas temperature measured at the same spot. The third subfigure shows the absolute pressure which is calculated by $p_{atm} - p_{2,bot}$ where $p_{2,bot}$ is the solid curve in figure 2.11, subfigure 2. Subfigure 4 shows gas density calculated from the ideal gas law by assuming dry air and using p_{abs} from subfigure 3. Subfigure 5 shows the calculated gas velocity using the relation $v = \sqrt{\frac{2 \cdot 9.8 \cdot p_{pit}}{\rho_g}}$.

Appendix B

Appendix to chapter 3

B.1 Proof of lemma 3.3

The lemma is restated for ease of reading:

Lemma B.1 *Suppose assumptions 3.1-3.6 hold. Suppose $x_k \in X_I$ is not a* KKT *point for the* NLP*, B_k is symmetric positive definite and $\eta_k > 0$. Then*

i) $D\phi_\nu(x_k, d_k) < 0$, *and*

ii) $\left\langle \nabla c^i_{\mathcal{I}}(x_k), d_k \right\rangle < 0$, *for all* $i \in J^\epsilon_k$

iii) $\phi_\nu(x_j) - \phi_\nu(x_{j+1}) \geq \gamma_\nu \left[\left\| Z_j^T g_j \right\|^2 + \|c_j\|_1 \right]$ *for the good iterates j and constant ν_j in (3.11).*

Proof: The descent property follows by an appropriate choice of the penalty parameter ν in (3.11). The directional derivative is

$$\begin{aligned} D\phi_\nu(x_k, d_k) = &\ \langle g_k, d_k \rangle - \nu_k \|c_k\|_1 \\ = &\ \langle g_k, Y_k p_{Y,k} \rangle + \left\langle Z_k^T g_k, p_{Z,k} \right\rangle - \nu_k \|c_k\|_1 \end{aligned}$$

From (3.15) and (3.18) $\langle g_k, Y_k p_{Y,k} \rangle = \langle \lambda_k, c_k \rangle$. Recalling $\langle \lambda_k, c_k \rangle \leq \|\lambda_k\|_\infty \|c_k\|_1$ and by choosing $\nu_k \geq \|\lambda_k\|_\infty + 2\rho$ for a constant $\rho > 0$, we get

$$\begin{aligned} D\phi_\nu(x_k, d_k) \leq &\ \left\langle Z_k^T g_k, p_{Z,k} \right\rangle - 2\rho \|c_k\|_1 \\ = &\ \left\langle Z_k^T g_k + \zeta_k w_k, p_{Z,k} \right\rangle - \langle \zeta_k w_k, p_{Z,k} \rangle - 2\rho \|c_k\|_1 \end{aligned} \tag{B.1}$$

We have $B_k p_{Z,k} = -\left(\zeta_k w_k + Z_k^T g_k\right)$ from (3.17). Considering $B_k s_k = \alpha_k B_k p_{Z,k}$ and equation (3.38) as in Biegler et al. (1995), section 3.4,

$$\cos\theta_k^{BFGS} = \frac{s_k^T B_k s_k}{\|s_k\| \|B_k s_k\|} = -\frac{\left\langle Z_k^T g_k + \zeta_k w_k, p_{Z,k} \right\rangle}{\left\| Z_k^T g_k + \zeta_k w_k \right\| \|p_{Z,k}\|}$$

is well-defined. We have that

$$D\phi_\nu(x_k, d_k) \leq -\left\|Z_k^T g_k + \zeta_k w_k\right\| \|p_{Z,k}\| \cos\theta_k^{BFGS} - \zeta_k \langle w_k, p_{Z,k}\rangle - 2\rho \|c_k\|_1$$

holds for all k. This is equation (68) in Biegler et al. (1995), and the derivation for the good iterates indexed by j follows

$$\frac{\|s_k\|}{\|B_k s_k\|} = \frac{\|p_{Z,k}\|}{\left\|Z_k^T g_k + \zeta_k w_k\right\|}$$

$$\frac{1}{\beta_3}\left\|Z_j^T g_j + \zeta_j w_j\right\| \leq \|p_{Z,j}\| \leq \frac{1}{\beta_2}\left\|Z_j^T g_j + \zeta_j w_j\right\| \tag{B.2}$$

and the assumption $\beta_3 > 1$ is valid since β_3 is an upper bound. We get

$$D\phi_\nu(x_k, d_k) \leq -\frac{\beta_1}{\beta_3}\left\|Z_j^T g_j\right\|^2 + \left[2\zeta_j \cos\theta_j^{BFGS} \left|g_j^T Z_j w_j\right| - \zeta_j w_j^T p_{Z,j}\right] - 2\rho \|c_j\|_1$$

Choosing

$$2\zeta_j \cos\theta_j^{BFGS} \left|g_j^T Z_j w_j\right| - \zeta_j w_j^T p_{Z,j} \leq \rho \|c_j\|_1$$

gives for all good iterates

$$D\phi_\nu(x_k, d_k) \leq -\frac{\beta_1}{\beta_3}\left\|Z_j^T g_j\right\|^2 - \rho \|c_j\|_1 \tag{B.3}$$

which is zero only at KKT points according to (3.44). As in Biegler et al. (1995) we get the update rule (for all iterates)

$$\zeta_k \left[2\cos\theta_k^{BFGS} \left|g_k^T Z_k w_k\right| + w_k^T B_k^{-1} Z_k^T g_k + \zeta_k w_k^T B_k^{-1} w_k\right] \leq \rho \|c_k\|_1 \tag{B.4}$$

This inequality is clearly consistent for a sufficiently small and positive ζ_k, hence $0 < \zeta_k \leq 1$ is feasible and gives descent. For the non-good iterates the angle θ_k^{BFGS} is $\pi/2$ so $-\zeta_k w_k^T p_{Z,k} \leq \rho \|c_k\|_1$ holds, giving

$$D\phi_\nu(x_k, d_k) \leq -\left\|Z_k^T g_k + \zeta_k w_k\right\| \|p_{Z,k}\| \cos\theta_k^{BFGS} - \rho \|c_k\|_1$$

which is nonpositive, and zero only at KKT points since $c_k = 0$ implies $w_k = 0$. The first claim is proved.

To prove item *ii*), strict feasibility, from lemma 3.2 *i*) we have $\gamma_{k-1} < 0$ otherwise by lemma 3.2 *ii*) and equation (3.51), the algorithm would have

stopped in step (vii). Then, since $Z_k^T \nabla c^i_{\mathcal{I},k} = 0$, $i \in J_k^\epsilon$ we have by (3.15) and the associated definition of c_k

$$\left\langle \nabla c^i_{\mathcal{I}}(x_k), d_k \right\rangle = \left\langle Y_k^T \nabla c^i_{\mathcal{I}}(x_k), p_{Y,k} \right\rangle = \eta_k \gamma_{k-1} < 0$$

for $i \in J_k^\epsilon$.

To establish the third claim we follow the derivation of lemma 4.1 of Biegler et al. (1995). From (B.3) we have

$$D\phi_\nu(x_k, d_k) \leq -b_2 \left[\left\| Z_j^T g_j \right\|^2 + \left\| c_j \right\|_1 \right]$$

where $b_2 = \{\beta_1/\beta_3, \rho\}$. Continuing as in Biegler et al. (1995) we establish

$$\left\| d_j \right\|^2 \leq 3\beta_0^2 \left\| p_Z^j \right\|^2 + 3\gamma_a^2 \left\| c_j \right\|^2$$

where $\left\| c_j \right\|^2 \leq \sup_{x_j \in X_I} \left\| c(x_j) \right\| \left\| c_j \right\|_1$, since $\left\| \cdot \right\| \leq \left\| \cdot \right\|_1$. From (B.2), the assumption $\left\| w_k \right\| \leq \kappa_c \left\| c_k \right\|^{1/2}$, $\left\| \cdot \right\| \leq \left\| \cdot \right\|_1$, $\zeta_j \leq 1$ and $a^2 + 2ab + b^2 \leq 3a^2 + 3b^2$ for $a, b \geq 0$ we have

$$\left\| p_{Z,j} \right\|^2 \leq \frac{3}{\beta_2^2} \left[\left\| Z_j^T g_j \right\|^2 + \kappa_c^2 \left\| c_j \right\|_1 \right]$$

giving

$$\begin{aligned} \left\| d_j \right\|^2 \leq & \quad -\tfrac{9\beta_0^2}{\beta_2^2} \left\| p_{Z,j} \right\| + \left(\tfrac{9\beta_0^2}{\beta_2^2} \kappa_c^2 + 3\gamma_a^2 \sup_{x_j \in X_I} \left\| c(x_j) \right\| \right) \left\| c_j \right\|_1 \\ \leq & \quad b_3 \left[\left\| Z_j^T g_j \right\|^2 + \left\| c_j \right\|_1 \right] \end{aligned}$$

where

$$b_3 = \max \left\{ \frac{9\beta_0^2}{\beta_2^2}, \frac{9\beta_0^2}{\beta_2^2} \kappa_c^2 + 3\gamma_a^2 \sup_{x_j \in X_I} \left\| c(x_j) \right\| \right\}$$

The non-zero lower bound on $\tilde{\alpha} > \frac{(1-\theta_j^{BFGS}) b_2}{b_1 b_3}$ holds and the Armijo condition then holds for $\alpha_j = \tau \tilde{\alpha}$. Then $-D\phi_\nu(x_j, d_j) \geq b_2 \left(\left\| Z_j^T g_j \right\|^2 + \left\| c_j \right\|_1 \right)$ in equation (3.40) gives

$$\phi_\nu(x_j) - \phi_\nu(x_{j+1}) \geq \quad \gamma_\nu \left[\left\| Z_j^T g_j \right\|^2 + \left\| c_j \right\|_1 \right]$$

where

$$\gamma_\nu = \theta_j^{BFGS} b_2 \min\left\{1, \frac{(1-\theta_j^{BFGS})\tau b_2}{b_1 b_3}\right\}$$

where min is only needed to handle an $\alpha_j = 1$. $\square$

Appendix C

Appendices to chapter 4

The almost block diagonal (`ABD`) form used in nonlinear `MPC` in chapter 4 is derived in detail below. The `ABD` form depends on the discretization method. General `ABD` forms are given for explicit Euler, Runge-Kutta of 4th order (`RK4`) and for Lobatto IIIC. The `ABD` forms are given for a general nonlinear model $\dot{x} = f(x, u)$ with prediction horizon N and move horizon M. Only the model constraint over the horizon is considered, but additional constraints can be fitted into the same scheme. Note that for distributed systems the Jacobian $\nabla_{x,u} f(x, u)$ has a sparse structure that should be exploited. This is not considered in the following, since it is problem dependent.

C.1 Explicit Euler

Explicit Euler gives the discretization scheme

$$x_{k+1} = x_k + hf(x_k, u_k)$$

where h is the time-step. Introduce

$$g(x_k, u_k) = -(x_{k+1} - x_k) + hf(x_k, u_k), \quad k = 0, \ldots, N-1$$

Define the variable vector

$$\tilde{x} = [\ x_1^T \quad \cdots \quad x_N^T \quad u_0^T \quad \cdots \quad u_M^T\ ]^T$$

Newtons method for solving this nonlinear equation is given by

$$J(\tilde{x})p_k = -g(\tilde{x}), \quad k = 0, \ldots, N-1 \tag{C.1}$$

where $J(\tilde{x}) = \nabla_{\tilde{x}} g(\tilde{x})$ with the vector function

$$g(\tilde{x}) = \begin{bmatrix} -(x_1 - x_0) + hf(x_0, u_0) \\ -(x_2 - x_1) + hf(x_1, u_1) \\ \vdots \\ -(x_{N-1} - x_{N-1}) + hf(x_{N-1}, u_M) \end{bmatrix} = \begin{bmatrix} g_1(x_1, u_0) \\ g_2(x_2, x_1, u_0) \\ \vdots \\ g_N(x_N, x_{N-1}, u_M) \end{bmatrix} \tag{C.2}$$

The Jacobian is defined by

$$J(\tilde{x}) = \nabla_{\tilde{x}} g(\tilde{x}) = \begin{bmatrix} \nabla_{\tilde{x}} g_1(\tilde{x}) \\ \nabla_{\tilde{x}} g_2(\tilde{x}) \\ \vdots \\ \nabla_{\tilde{x}} g_N(\tilde{x}) \end{bmatrix} =$$

$$\begin{bmatrix} \nabla_{x_1} g_1 & \nabla_{x_2} g_1 & \cdots & \nabla_{x_N} g_1 & \nabla_{u_0} g_1 & \cdots & \nabla_{u_M} g_1 \\ \nabla_{x_1} g_2 & \nabla_{x_2} g_2 & \cdots & \nabla_{x_N} g_2 & \nabla_{u_0} g_2 & \cdots & \nabla_{u_M} g_2 \\ \vdots & & \ddots & & & \ddots & \\ \nabla_{x_1} g_N & \nabla_{x_2} g_N & \cdots & \nabla_{x_N} g_N & \nabla_{u_0} g_N & \cdots & \nabla_{u_M} g_N \end{bmatrix}$$

which by defining

$$\begin{aligned} \nabla_x f|_k &= A_k \\ \nabla_u f|_k &= B_k \end{aligned}$$

gives

$$J(\tilde{x}) = \begin{bmatrix} -I & & & & & & & hB_0 & & & \\ (I+hA_1) & -I & & & & & & & hB_1 & & \\ & \ddots & \ddots & & & & & & & \ddots & \\ & & (I+hA_M) & -I & & & & & & & hB_M \\ & & & \ddots & \ddots & & & & & & \vdots \\ & & & & & (I+hA_{N-1}) & -I & & & & hB_M \end{bmatrix} \tag{C.3}$$

The nonlinear system (C.1) is now defined by substituting in (C.2) and (C.3). The Jacobian is sparse, a property that can be utilized by sparse optimization algorithms.

C.2 Runga-Kutta 4

`RK4` for the system $\dot{x} = f(x,u)$ is defined by the discretization scheme

$$\begin{aligned} x_{k+1} &= x_k + \tfrac{1}{6}(y_1 + 2y_2 + 2y_3 + y_4) \\ y_1 &= hf(x_k, u_k) \\ y_2 &= hf(x_k + \tfrac{y_1}{2}, u_k) \\ y_3 &= hf(x_k + \tfrac{y_2}{2}, u_k) \\ y_4 &= hf(x_k + y_3, u_k) \end{aligned}$$

Observe that the property $u_k = u_k + \frac{y_1}{2} = u_k + \frac{y_2}{2} = u_k + y_3$ has been utilized. Define the variable vector

$$\tilde{x} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & x_1 & y_{21} & \cdots & y_{24} & x_2 & \cdots & y_{N4} & x_N & u_0 & \cdots & u_M \end{bmatrix}$$

and the vector function

$$g(\tilde{x}) = \begin{bmatrix} -y_{11}+hf(x_0,u_0) \\ -y_{12}+hf(x_0+\frac{y_{11}}{2},u_0) \\ -y_{13}+hf(x_0+\frac{y_{12}}{2},u_0) \\ -y_{14}+hf(x_0+y_{13},u_0) \\ -(x_1-x_0)+\frac{1}{6}(y_{11}+2y_{12}+2y_{13}+y_{14}) \\ -y_{21}+hf(x_1,u_1) \\ -y_{22}+hf(x_1+\frac{y_{21}}{2},u_1) \\ -y_{23}+hf(x_1+\frac{y_{22}}{2},u_1) \\ -y_{24}+hf(x_1+y_{23},u_1) \\ -(x_2-x_1)+\frac{1}{6}(y_{21}+2y_{22}+2y_{23}+y_{24}) \\ \vdots \end{bmatrix} = \begin{bmatrix} g_{11}(y_{11},u_0) \\ g_{12}(y_{11},y_{12},u_0) \\ g_{13}(y_{12},y_{13},u_0) \\ g_{14}(y_{13},y_{14},u_0) \\ g_{15}(y_{11},y_{12},y_{13},y_{14},x_1) \\ g_{21}(x_1,y_{21},u_1) \\ g_{22}(x_1,y_{21},y_{22},u_1) \\ g_{23}(x_1,y_{22},y_{23},u_1) \\ g_{24}(x_1,y_{23},y_{24},u_1) \\ g_{25}(x_1,y_{21},y_{22},y_{23},y_{24},x_2) \\ \vdots \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix}$$

where the last vector emphasizes that the initial pattern is repeated over the horizon N. The Jacobian matrix is defined by

$$J(\tilde{x}) = J(y,x,u) = \begin{bmatrix} \nabla_{\tilde{x}} g_1(\tilde{x}) \\ \nabla_{\tilde{x}} g_2(\tilde{x}) \\ \vdots \\ \nabla_{\tilde{x}} g_N(\tilde{x}) \end{bmatrix} =$$

$$\begin{bmatrix} \nabla_{y_{11}} g_{11} & \cdots & \nabla_{y_{14}} g_{11} & \nabla_{x_1} g_{11} & \cdots & \nabla_{y_{N4}} g_{11} & \nabla_{x_N} g_{11} & \nabla_{u_0} g_{11} & \cdots & \nabla_{u_M} g_{11} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \nabla_{y_{11}} g_{15} & \cdots & \nabla_{y_{14}} g_{15} & \nabla_{x_1} g_{15} & \cdots & \nabla_{y_{N4}} g_{15} & \nabla_{x_N} g_{15} & \nabla_{u_0} g_{15} & \cdots & \nabla_{u_M} g_{15} \\ \nabla_{y_{11}} g_{21} & \cdots & \nabla_{y_{14}} g_{21} & \nabla_{x_1} g_{21} & \cdots & \nabla_{y_{N4}} g_{21} & \nabla_{x_N} g_{21} & \nabla_{u_0} g_{21} & \cdots & \nabla_{u_M} g_{21} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \nabla_{y_{11}} g_{N5} & \cdots & \nabla_{y_{14}} g_{N5} & \nabla_{x_1} g_{N5} & \cdots & \nabla_{y_{N4}} g_{N5} & \nabla_{x_N} g_{N5} & \nabla_{u_0} g_{N5} & \cdots & \nabla_{u_M} g_{N5} \end{bmatrix}$$

By defining

$$\begin{aligned} \nabla_x f|_{x_k,u_k} &= A_k^0 & \nabla_u f|_k &= B_k^0 \\ \nabla_x f|_{x_k+\frac{y_1}{2},u_k} &= A_k^1 & \nabla_u f|_{x_k+\frac{y_1}{2},u_k} &= B_k^1 \\ \nabla_x f|_{x_k+\frac{y_2}{2},u_k} &= A_k^2 & \nabla_u f|_{x_k+\frac{y_2}{2},u_k} &= B_k^2 \\ \nabla_x f|_{x_k+y_3,u_k} &= A_k^3 & \nabla_u f|_{x_k+y_3,u_k} &= B_k^3 \end{aligned}$$

the Jacobian for `RK4` becomes

$$
J(y,u) =
\left[\begin{array}{cccccccccccccccccccc}
-I & & & & & & & & & & & & & & & & hB_0^0 & & & \\
hA_0^1 & -I & & & & & & & & & & & & & & & hB_0^1 & & & \\
 & \frac{h}{2}A_0^2 & -I & & & & & & & & & & & & & & hB_0^2 & & & \\
 & & \frac{h}{2}A_0^3 & -I & & & & & & & & & & & & & hB_0^3 & & & \\
\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & -I & & & & & & & & & & & & 0 & & & \\
 & & & & hA_1^0 & -I & & & & & & & & & & & & hB_1^0 & & \\
 & & & & hA_1^1 & \frac{h}{2}A_1^1 & -I & & & & & & & & & & & hB_1^1 & & \\
 & & & & hA_1^2 & & \frac{h}{2}A_1^2 & -I & & & & & & & & & & hB_1^2 & & \\
 & & & & hA_1^3 & & & hA_1^3 & -I & & & & & & & & & hB_1^3 & & \\
 & & & & I & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & -I & & & & & & & & 0 & & \\
 & & & & & & & & & \ddots & \ddots & & & & & & & & \ddots & \\
 & & & & & & & & & & hA_{N-1}^0 & -I & & & & & & & & hB_M^0 \\
 & & & & & & & & & & hA_{N-1}^1 & \frac{h}{2}A_N^1 & -I & & & & & & & hB_M^1 \\
 & & & & & & & & & & hA_{N-1}^2 & & \frac{h}{2}A_N^2 & -I & & & & & & hB_M^2 \\
 & & & & & & & & & & hA_{N-1}^3 & & & hA_N^3 & -I & & & & & hB_M^3 \\
 & & & & & & & & & & I & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & -I & & & & 0
\end{array}\right]
$$

Comparing the variable vectors for explicit Euler and `RK4` reveals that the number of variables is considerably larger in `RK4`. Assuming $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ the number of variables in explicit Euler is $\theta_{euler} = n_x N + n_u M$. In `RK4` the dimension of y_i equals the dimension of x giving $\theta_{RK4} = 5n_x N + n_u M$. Observe that the degrees of freedom $n_u M$ is unaltered by the choice of discretization schemes. Assuming $n_u M \ll n_x N$ shows that `RK4` has approximately 5 times the number of variables as compared to explicit Euler. Hence, the increased order of accuracy of `RK4` comes at the expense of a larger optimization problem. Note that the calculations above are somewhat counteracted by the enlarged stability region of `RK4` allowing longer step-lengths.

Considering the sintering case from section 2.3 with the state vector $x = [T_g, T_s, X_C]$ of dimension $n_x = (n_z - 1) + 2n_z = 29$ if the spatial dimension is $n_z = 10$. Choosing a batch time of 40 minutes and a sampling interval of 30 seconds gives a prediction horizon of $N = 80$. The control action has dimension $n_u = 1$ and the move horizon is chosen as $M = 40$. This gives

$$\begin{aligned} \theta_{euler} &= n_x N + n_u M = 29 \cdot 80 + 1 \cdot 40 = 2360 \\ \theta_{RK4} &= 5n_x N + n_u M = 5 \cdot 29 \cdot 80 + 1 \cdot 40 = 11640 \end{aligned}$$

Note that due to the stiffness of the system choosing the sampling interval as 30 seconds leaves both explicit Euler and `RK4` numerically unstable.

C.3 Lobatto IIIC

Lobatto IIIC is an A-stable implicit Runge-Kutta method with second order accuracy. The Butcher table is

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cc} 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Since the last row of A equals b^T Lobatto IIIC is a `FSAL` method. The discretization scheme for a time-invariant system can be written as

$$\begin{bmatrix} \frac{1}{h}(X_1 + X_2 - 2x_{k-1}) - f(X_1, U) \\ \frac{1}{h}(-X_1 + X_2) - f(X_2, U) \end{bmatrix} = 0$$

and setting $x_k = X_2$. Choosing the variable vector

$$\tilde{x} = \begin{bmatrix} X_{1,1} & X_{1,2} & X_{2,1} & X_{2,2} & \cdots & X_{N,1} & X_{N,2} & U_1 & \cdots & U_M \end{bmatrix}$$

where the first index is the time index over the prediction/move horizons gives the following right-hand side:

$$g(\tilde{x}) = \begin{bmatrix} \frac{1}{h}(X_{1,1} + X_{1,2} - 2x_0) - f(X_{1,1}, U_1) \\ \frac{1}{h}(-X_{1,1} + X_{1,2}) - f(X_{1,2}, U_1) \\ \frac{1}{h}(X_{2,1} + X_{2,2} - 2X_{1,2}) - f(X_{2,1}, U_2) \\ \frac{1}{h}(-X_{2,1} + X_{2,2}) - f(X_{2,2}, U_2) \\ \vdots \\ \frac{1}{h}(X_{N,1} + X_{N,2} - 2X_{N-1,2}) - f(X_{N,1}, U_M) \\ \frac{1}{h}(-X_{N,1} + X_{N,2}) - f(X_{N,2}, U_M) \end{bmatrix} = 0$$

Observe that x_{k-1} has been replaced by $X_{k-1,2}$ throughout the horizon. By defining

$$\begin{aligned} \nabla_X f|_{X_{k,1},U_k} &= A_k^1 \quad \nabla_U f|_{X_{k,1},U_k} = B_k^1 \\ \nabla_X f|_{X_{k,2},U_k} &= A_k^2 \quad \nabla_U f|_{X_{k,2},U_k} = B_k^2 \end{aligned}$$

the constraint Jacobian becomes:

$$
J(\tilde{x}) = \begin{bmatrix}
\frac{I}{h}+A_1^1 & \frac{I}{h} & & & & & & & & & & & & & & & B_1^1 & B_M^1 \\
-\frac{I}{h} & \frac{I}{h}+A_1^2 & & & & & & & & & & & & & & & B_1^2 & B_M^2 \\
 & -\frac{2I}{h} & \frac{I}{h}+A_2^1 & \frac{I}{h} & & & & & & & & & & & & & B_2^1 & \vdots \\
 & & -\frac{I}{h} & \frac{I}{h}+A_2^2 & \ddots & & & & & & & & & & & & B_2^2 & B_M^1 \\
 & & & -\frac{2I}{h} & \ddots & \ddots & & & & & & & & & & & \ddots & B_M^2 \\
 & & & & \ddots & & -\frac{2I}{h} & \frac{I}{h} & & & & & & & & & & \\
 & & & & & \ddots & \frac{I}{h}+A_M^1 & -\frac{I}{h} & \frac{I}{h} & & & & & & & & & \\
 & & & & & & -\frac{I}{h} & \frac{I}{h}+A_M^2 & \ddots & & & & & & & & & \\
 & & & & & & & \ddots & \ddots & & & & & & & & & \\
 & & & & & & & & & -\frac{2I}{h} & \frac{I}{h}+A_N^1 & \frac{I}{h} & & & & & & \\
 & & & & & & & & & & -\frac{I}{h} & \frac{I}{h}+A_N^2 & & & & & &
\end{bmatrix}
$$

For the grate sintering case with prediction data etc. as discussed for `RK4`, but with a sampling time of $h = 1s$, a dense Jacobian will require about 12.2Gb in double real, while a sparse Jacobian will require about 1.6Mb of memory.

C.4 sSQP

The sensitivity matrix $\mathcal{S}$ for the sequential method (de Oliveira 1994), pp.19-20, is considered in the following. Given the system

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= c(x) \end{aligned}$$

Define

$$\begin{aligned} \nabla_x f|_k &= A_k \\ \nabla_u f|_k &= B_k \\ \nabla_x c|_k &= C_k \end{aligned}$$

We have

$$\begin{aligned} \frac{\partial y_{k+1}}{\partial u_k} &= \frac{\partial y_{k+1}}{\partial x_{k+1}} \cdot \frac{\partial x_{k+1}}{\partial u_k} = C_{k+1}B_k \\ \frac{\partial y_{k+2}}{\partial u_k} &= \frac{\partial y_{k+2}}{\partial x_{k+2}} \cdot \frac{\partial x_{k+2}}{\partial x_{k+1}} \cdot \frac{\partial x_{k+1}}{\partial u_k} = C_{k+2}A_{k+1}B_k \\ \frac{\partial y_{k+2}}{\partial u_{k+1}} &= \frac{\partial y_{k+2}}{\partial x_{k+2}} \cdot \frac{\partial x_{k+2}}{\partial u_{k+1}} \cdot \frac{\partial x_{k+1}}{\partial u_k} = C_{k+2}B_{k+1} \end{aligned}$$

which gives the general case

$$S_{ij} = \begin{cases} 0 & if \;\; i > j \\ C_{k+i}B_{k+j-1} & if \;\; i = j \\ C_{k+i}(\prod_{m=1}^{i-j} A_{k+i-m})B_{k+j-1} & otherwise \end{cases}$$

The sensitivity matrix for a prediction horizon P and move horizon M becomes (with $k = 0$ and $C_i = I$ for all i)

$$\mathcal{S} = \begin{bmatrix} B_0 & & & & \\ A_1B_0 & B_1 & & & \\ A_2A_1B_0 & A_2B_1 & B_2 & & \\ \vdots & & \ddots & \ddots & \\ \prod_{m=1}^{M} A_mB_0 & \cdots & & A_MB_{M-1} & B_M \\ \prod_{m=1}^{M+1} A_mB_0 & \cdots & & A_MB_{M-1} & A_{M+1}B_M + B_{M+1} \\ \vdots & & & \vdots & \vdots \\ \prod_{m=1}^{N-1} A_mB_0 & \cdots & & A_{N-1}B_{M-1} & \tilde{B} \end{bmatrix} \tag{C.4}$$

where

$$\tilde{B} = \prod_{m=M+1}^{N-1} A_mB_M + \prod_{m=M+2}^{N-1} A_mB_{M+1} + \cdots + A_{N-1}B_{N-1} + B_N$$

Considering the lower leftmost element of $\mathcal{S}$ it is observed that this element corresponds to the observability matrix of the linearized system. Complete observability then suffices to guarantee full rank of $\mathcal{S}$ since all other elements are truncated versions of the lower left element. Full rank of $\mathcal{S}$ implies observability of all (unstable) modes, and the system may be stabilized by a suitable strategy. Note that pinning down unstable modes by an end-point constraint $x_N^u = 0$ is susceptible to numerical roundoff errors (Rawlings 2000). The stability proof of the sequential method is only valid for open loop stable systems (de Oliveira and Biegler 1995).

If the discretization scheme is known beforehand (i.e. if solvers like `MATLAB`'s `ode15s` are not used), an analytic Jacobian can be implemented without the need for extensive for-loops as would be the case if the elements of $\mathcal{S}$ were to be constructed explicitly. Note that by perturbation of inputs to the simulator, the Jacobian constructed by finite differences in the simulator outputs gives the sensitivity matrix directly. I.e. perturbing each control u_i over the horizon M gives the variations in each output $y_{i,j}$ for $i = 1, \ldots, n_y$ and $j = 1, \ldots, N$. This results in n_uM calls to the simulator. If the process can be stably discretized with explicit Euler or `RK4` considerable savings can be achieved. Note that the grate sintering case from chapter 2 cannot be stably discretized with explicit methods, while such methods work well with

the `CSTR`-example. We consider explicit Euler only in the following, but observe that the approach fits equally well with all (explicit and implicit) discretization schemes.

For explicit Euler the right hand side function is given by equation (C.2) and the Jacobian is given by equation (C.3). Partition the Jacobian from equation (C.3) such that the parts for the states and the controls are separated in matrices A and B as below:

$$A = \begin{bmatrix} -I & & & & & & & \\ (I+hA_1) & -I & & & & & & \\ & \ddots & \ddots & & & & & \\ & & & (I+hA_M) & -I & & & \\ & & & & \ddots & \ddots & & \\ & & & & & & (I+hA_{N-1}) & -I \end{bmatrix}$$

and

$$B = \begin{bmatrix} hB_0 & & & \\ & hB_1 & & \\ & & \ddots & \\ & & & hB_M \\ & & & \vdots \\ & & & hB_M \end{bmatrix}$$

The sensitivity matrix from chapter 4 is given by $\mathcal{S} = -(\nabla_{x^k} h^k)^{-1} \nabla_{u^k} h^k = -A^{-1}B$ with A and B given above in the explicit Euler case. Since

$$A^{-1} = \begin{bmatrix} -I & & & & \\ -\tilde{A}_1 & -I & & & \\ -\tilde{A}_2\tilde{A}_1 & -\tilde{A}_2 & -I & & \\ \vdots & & \ddots & \ddots & \\ -\prod_{m=1}^{N-1} \tilde{A}_m & \cdots & & -\tilde{A}_{N-1} & -I \end{bmatrix}$$

where $\tilde{A}_j = (I + hA_j)$, the sensitivity formed by $\mathcal{S} = -A^{-1}B$ is given by equation (C.4). Observe that partitioning of the Jacobian into appropriate submatrices A and B continue to hold for the other discretization methods as well. Hence, analytic sensitivity matrices can easily be formed for `RK4` and Lobatto IIIC as well.

This gives two alternative methods for forming the sensitivity matrix. If analytic derivatives are know the last approach should be chosen. If

the discretization scheme is given (and not hidden inside the solver as in `ode15s` or `HYSYS`) the latter approach may be chosen (with matrices A and B adapted to the discretization scheme), even if analytic derivatives are not known. This is of particular interest with implicit discretization methods where the model can be hard to solve, since a simple evolution does not suffice. This is seen by the fact that perturbing the "derivative" function $f(\cdot)$ in $\dot{x} = f(x, u)$ in x and u to get A_k and B_k and composing A and B from these may be cheaper than simulating the system to get $\mathcal{S}$ directly. If the discretization scheme is hidden inside the simulator numerical perturbation to get $\mathcal{S}$ directly is the only option.

For large scale systems it is essential that the matrix A is formed directly as a sparse matrix. In the grate sintering case with approximately 10000 variables the storage of a full A would require $10000^2 \times 2 = 200Mb$ in double precision, and about $2 \cdot 10000 \times 2 = 40Mb$ (depending on the non-zero structure of the A_k's) in sparse form. The sensitivity matrix $\mathcal{S}$ (and B) has much smaller dimension if $n_u M$ is small. In the grate sintering case $\mathcal{S}$ has dimension $10000 \cdot 1$ since there is only one control. The dimension of $\mathcal{S}$ is the same as the dimension of B.

Appendix D

Reprint of the CCA-paper

This is a reprint of the paper Martinsen, Johansen, and Foss (1999). The issues in this paper was not pursued further during the progress of the doctoral work. The paper suggests a generic multi-model for hyperbolic `PDE`'s. The main drawback is that the switching between zone-boundaries causes problems when the system is being integrated by conventional solvers like `MATLAB`'s `ode15s`. The results in the paper were obtained by significant de-tuning of the model parameters to allow integration by explicit schemes. The approach is thus viable for `PDE`'s that can be integrated by explicit solvers, but extensions to stiff and unstable systems are left as future work.

A control relevant dynamic model of grate sintering

Frode Martinsen, Bjarne A. Foss and Tor Arne Johansen
Norwegian University of Science and Technology
Department of Engineering Cybernetics
N-7491 Trondheim, NORWAY
fax: + 47-73594399 email: frodem@itk.ntnu.no
www.itk.ntnu.no/groups/process_control/

Keywords: Sintering, dynamic model, multi-model, simulation.

Abstract

In this paper a control relevant nonlinear dynamic model of grate sintering is presented. The model is designed for control purposes for use in future model predictive control (MPC) strategies. A multi-model approach is utilized where the model is presented as a convex combination of locally affine models. The model performance is compared to global models listed in the literature by simulations and by comparison to industrial plant data.

1 Introduction

The metallurgical process of sintering prepares the (iron) ore to form suitable feed for the blast furnace. Granulated ore and coke are mixed, moistened with water and micro-pelletized to form the charge. The charge is loaded onto a grate and leveled to form a bed which is ignited by a gas-fueled ignition hood. A heat wave and coke combustion zone travels down through the bed under the influence of a suction pressure. Hot gas from the combustion zone passes through moist charge deeper in the bed where water evaporates. The process can be divided in five subsequent zones; heat exchange, fusion, combustion, drying and over-moist charge. This is illustrated in figure 2 a). The main purpose of sintering is to convert weakly-bounded granules into a partially fused porous sinter cake suitable for feeding to the blast furnace. Sintering is a complex process involving flow of gas through a packed bed, heat and mass transfer between gas and solids, heterogeneous chemical reactions and melting of solids.

Several models of the sintering process is presented in the literature [1], [2], [3], [4], [5], [6], [7], [8]. These models are presented as nonlinear PDE's, and mainly focus on reproducing important process quantities. There are few reported results on model based control of the sintering process. Kwon et al. [9] uses a linear MPC scheme with an identified input/output model to control the burn-through point of traveling grate sintering. The reported real-time experiments and simulations demonstrates that the chosen MPC algorithm performs well. In the present work, the problem of controlling production rate and quality is directly addressed by identifying a model from designed experiments. As argued below there is an economic criteria restricted by constraints involved, and investigating MPC as a means to control grate sintering is motivated. We seek to exploit the underlying structure of the sintering process to develop a structured model which later on can be utilized to develop a robust control strategy for the sintering process. In the present approach the global nonlinear PDE model is approximated by a convex combination of locally linear or affine models. The local (zone) models are interconnected by both boundary values and propagation of zone positions as sintering proceeds. The interconnection of the zones are handled by the multi-modeling techniques of Johansen and Foss [10]. This method gives smooth interpolation of zones and is chosen since exact zone boundaries are uncertain and overlap.

Early attempts to model the sintering process divided the process into zones [1]. This approach was later abandoned [2]. The present approach re-investigates this approach, but the motivation for this differs from earlier works. The model in this paper aims at a *control relevant* model, not a detailed first principles model. In a control relevant model reproducing the internal states is not as important as reproducing the outputs. This becomes important in a state space oriented MPC strategy due to the need for either measuring or estimating the next internal state value x_{k+1}. If x_{k+1} cannot be measured and estimation is hard, it is likely that the MPC algorithm performs poorly. In such a case an input/output oriented MPC strategy might perform better, despite the model being less accurate in reproducing the internal states.

The present paper will emphasize the development of the control relevant model, and the control algorithm itself is to be investigated in a later paper.

2 Model

A model can be derived either from physical knowledge, identified from data or as a combination of these. A control relevant model is tailored to the selected control strategy, i.e. the control strategy influence on what information the model should produce. For the sintering process the overall goal is to produce sinter at prescribed quality and rate at the lowest possible cost. Manipulated inputs are coke, water, ignition energy and air flow, while the feed is regarded as a disturbance. The process dependent outputs are quality in terms of mechanical strength and reducibility [11], and production rate. The amplitude and shape of the heat wave determines the quality: Increased coke content increases the maximum sintering temperature, $T_{s,max}$ [8], but to achieve high reducibility $T_{s,max}$ should not be too high [12]. In addition proper ignition is necessary to establish the initial conditions for sintering [2]. The process variable influencing on production rate is mainly the permeability of the bed which in turn is influenced mainly by water content. There is an optimal water content yielding the highest bed permeability, i.e. there is an optimum in both coke [12] and water content [13] which can be formulated in a MPC criterion. Bounds on $T_{s,max}$ and minimum ignition energy are formulated as constraints in the MPC formulation. Due to the large (hot) recycle and time delays present the process is considered difficult to control, with quality and production rates being hard to predict [5].

The recycle stream is an important process variable serving as an on-line measure of mechanical sinter quality: Poor mechanical quality will give increased recycle rate, which in turn reduces the production rate. Reducibility can be quantified from the temperature profile in the fusion zone [12]. Since modeling the sinter quality is difficult, it is identified from experimental data not modeled explicitly at this stage. The stationary equations of the Voice permeability P in the laminar and turbulent flow regimes are respectively:

$$P_l = v\left(\frac{L}{\Delta p}\right)^{1.0} = \frac{1}{150\mu}\cdot\frac{\varepsilon^3}{(1-\varepsilon)^2}\cdot d_p^2$$
$$P_t = v\left(\frac{L}{\Delta p}\right)^{0.5} = \sqrt{\frac{1}{1.75\rho}}\cdot\left(\frac{\varepsilon^3}{1-\varepsilon}\right)^{0.5}\cdot d_p^{0.5}$$

The productivity relation thus becomes:

$$G = \frac{v}{G_s} = \frac{P}{G_s}\cdot\left(\frac{\Delta p}{L}\right)^n \qquad (1)$$

where $n = 0.5$ for turbulent flow and $n = 1$ for laminar flow. The mass air flow G serves as an on-line measure of production rate. Note that when significant melting occurs d_p becomes a conceptual parameter.

The discussion above on in- and outputs is represented graphically in figure 1.

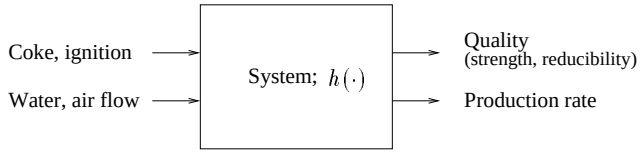

Figure 1: *System description.* The sinter quality is influenced mainly by coke content and ignition temperature, i.e. $h_q = h_q(x_C, T_{ign})$, while production rate is influenced mainly by water content and air flow, i.e. $h_p = h_p(x_{H_2O(l)}, G)$. Reduced mechanical quality increases recycle, which in turn decreases the production rate, i.e. the system is not decoupled.

A mechanistic model based on models reported in literature is presented in section 2.1, and a state-space multimodel is suggested in section 2.2. The models considered in this paper does not incorporate radial distributions, and the industrial process considered is batch-wise sintering of manganese ore in Greenawalt pans.

2.1 Mechanistic model

The general PDE model is stated in appendix A. Parameter uncertainties are present in the global models, since essential parameters typically are determined from empirical formulas valid only under idealized conditions. In industrial sintering processes the formation of cracks and channels leads to areas where air passes through without interacting with the mass in the sinter bed. In particular, the mass, h_m, and heat, h_c, transfer coefficients are calculated from the Nusselt and Sherwood numbers. These number are again calculated from the empirical relations [14]:

$$Nu = \frac{h_c d_p}{k} = \frac{1}{\varepsilon}\left(2 + 1.1Pr^{1/3}Re^{0.6}\right) \qquad (2)$$
$$Sh = \frac{h_m d_p}{D_{ON}} = \frac{1}{\varepsilon}\left(2 + 1.1Sc^{1/3}Re^{0.6}\right) \qquad (3)$$

valid for an idealized bed with homogeneous packing. In an industrial bed the gas flowing through channels and large cracks are not interacting with the solid, and the values estimated from the empirical relations for an idealized bed will deviate from the actual values. Various heuristics are utilized to overcome this in the literature, i.e. altering the constants of the empirical relations [2], [3], [7], and introducing a scaling factor [5], [6].

The kinetic parameters of coke combustion, fusion and solidification of solids and condensation are also not known in detail. The kinetic model of coke combustion is discussed in [1] assuming the reaction $C + O_2 \rightarrow CO_2$. The kinetics of fusion of solids is described by empirical schemes based on slag diagrams [6] or linear schemes based on process experience [5]. The kinetics of condensation of water is derived from laboratory tests [15] or

by heuristics and experience [2]. In addition the heat capacity of the solid and the void fraction will change in a complicated way as sintering proceeds. The average particle diameter, d_p, will not have a physical interpretation once melting in the fusion zone has occurred. Still all cited models utilize some equivalent particle diameter when melting has occurred.

Assuming infinite fast gas dynamics a mechanistic energy balance yields the pressure drop as the isothermal Ergun's relation:

$$\frac{\partial p}{\partial z} = 150\frac{\mu v(1-\varepsilon)^2}{d_p^2\varepsilon^3} + 1.75\frac{\rho v^2(1-\varepsilon)}{d_p\varepsilon^3} \qquad (4)$$

The pressure profile $\frac{\partial p}{\partial z}$ is highly nonlinear in ε and d_p, and estimating them by use of the Ergun relation will lead to uncertain values. Consequently the models available in literature uses heuristic estimates of the profiles $\frac{\partial \varepsilon}{\partial z}$ and $\frac{\partial d_p}{\partial z}$. All these phenomena gives rise to a model with many uncertain parameters, which must be accounted for when applying model based control to the sintering process.

2.2 State-space multi-model

The bed is divided in a fixed number n_z of vertical layers named elements Δz_i. Each element is allocated a model type $M_{z,i}$, $i \in 1, \cdots, 5$. These zone models are models of the natural zone partitioning depicted in figure 2 a). The allocation is based on process knowledge, i.e. as the state of the element as a function of either T_s and/or x_C, $x_{H_2O(l)}$. Once the model in an element has changed from $M_{z,i}$ to $M_{z,i-1}$, it can never change back to $M_{z,i}$.

As an example the transition from $M_{z,4}$ to $M_{z,3}$ is controlled by the ignition temperature of coke $T_{C,ign} \approx 900-1050^oC$, i.e. in an element Δz_i with solid temperature in this range, both $M_{z,4}$ and $M_{z,3}$ is calculated, and the resulting states become a weighted sum of the two models. Similar heuristics controls the transition between other zone models. The void fraction profile $\frac{\partial \varepsilon}{\partial z}$ is estimated by combining the weighting functions, the measured pressure and the Ergun relation in equation (4). A graphical illustration of the modeling concept and weighting functions is presented in figure 2 b). Note that a transition spans more than one element in general. In addition the same technique can be applied independent of the model type, linear vs. nonlinear etc., allocated to the individual elements. However, if the structure of the zone models differs explicit handling of initial conditions must be done to prevent inconsistencies. In the present model the model structure inside each zone is identical (linear or affine), and such initialization problems does not occur.

Viewed as a discretization of a system of PDE's, the described multi-modeling technique is related to the finite element method (FEM)[1]. Extensions to FEM is that in the present approach interpolation is done in the state-space as well as in time and the spatial domain.

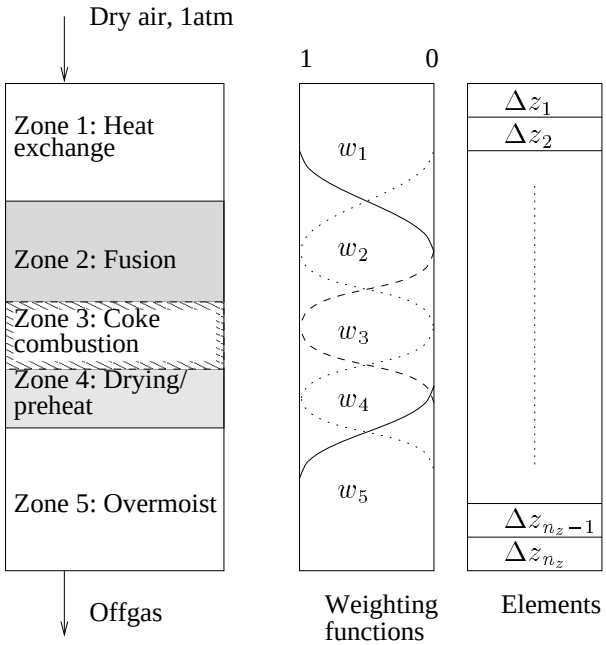

Figure 2: *Sintering zones.* **a)** The sinter bed is composed of five zones modeled separately. The zone widths and positions vary during the batch. The figure shows a typical situation at a time instance midway through the batch. The width and temperature of the fusion zone determines the sinter quality. **b)** Zone interconnection is handled by multi-modeling techniques. The elements Δz_i to the right have stationary vertical positions throughout the batch, but are allocated different models $M_{z,i}$ depending on the value of the weighting functions w_i which are moving as the batch proceeds. In transition regions the new model state in the element is computed as a weighted sum of the neighboring model types.

3 Simulations

Simulations of the global and multi-model approaches are compared to measurements taken on an industrial plant. The plant data and simulations are shown in figure 3.

The temperature dependent aggregated parameters of the mechanistic model were manually tuned against the plant data. Similarly, the parameters associated with each of the five multi-models were tuned against the

[1] FEM is applied in [6].

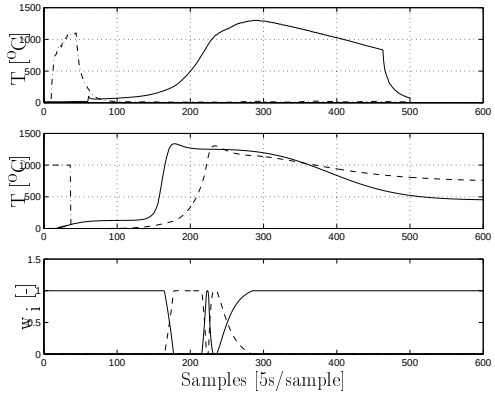

Figure 3: *Process data, simulations and weighting functions.* The upper part shows measured ignition temperature (dashed) and solid temperature measured at 15cm depth. The measured value has its peak value 1300^oC at sample 285. The sudden fall at sample 463 is caused by removing the S-element from the sintering pan. The middle part shows simulated temperatures at 15 cm depth, together with the applied ignition temperature (dashed, samples 1-36). The mechanistic model (solid line) has its peak value 1337^oC at sample 179, while the multi-model (dashed line) has its peak value 1305^oC at sample 233. The lower part shows the values of the weighting functions w_5 to w_1 from left to right. Note the resemblance to figure 2 and the simulated multi-model temperature.

plant data. Since the multi-models are valid in different operating ranges as defined by the weighting functions, tuning the multi-models was easier than tuning the mechanistic model. The interpolation functions are linear (non-smooth), with support and localization selected by process knowledge.

4 Discussion

Noting that simplified kinetics of fusion and water evaporation/condensation have been utilized, some improvements of the mechanistic model are expected upon introducing the results of Patisson et al. [6], [15]. The model does not incorporate radial spatial inhomogeneities, which are known to appear regularly in industrial applications in form of channels, cracks and inhomogeneous filling and mixing. Such inhomogeneities causes radial temperature and concentration gradients, in addition to the axial gradients modeled herein. Future work on mechanistic modeling should incorporate radial gradients, possibly with additional simplifying assumptions.

Manual tuning of parameters in a non-linear, coupled distributed system is non-optimal, and a suitable identification scheme should be implemented. Manual tuning of the parameters of the (affine) multi-models is again non-optimal, and work is in progress to implement augmented Kalman filtering in each operating regime. The observability of the (distributed) system is yet to be investigated. Tuning has only been performed for one set-point $x_{c,0}, x_{w,0}$ and is not expected to be robust to set-point variations. The affine multi-model uncertainty problem can be adressed by the method of Slupphaug and Foss [16].

The mechanistic model was simulated by applying Newton iterations to an implicit finite difference scheme giving unconditional stability. The stability of the multi-model simulation scheme has not been analyzed. The linear weighting functions can be replaced by smooth functions, and the elements Δz_i of the multi-model can be decoupled from the spatial discretization resolution of the implicit difference scheme. The relation between FEM and the proposed state-space multi-model algorithm should be pursued further. A multi-model uncertainty class should be defined and identified. The control algorithm itself is to be investigated in a later paper.

The simulations showed that the performance of the simpler model gave satisfactory results as compared to the mechanistic model. However, an improved mechanistic model can give better simulation performance. The proposed multi-model algorithm reveals an interesting relation between multi-modeling techniques and finite-element methods which suggests a direction for future research. This also suggests that stability issues of the proposed multi-model algorithm can be analyzed with methods previously applied to FEM. Finally, multi-models have so far been extensively investigated for data driven models, while the present case is motivated by acknowledging that models published in the literature have important parameter uncertainties.

A Mechanistic model

The following states are included in the model $x = \left[T_s, T_g, x_c, x_w, x_{O_2}, x_{N_2}, x_{CO_2}, x_{H_2O(v)}\right]$; i.e. temperature of solids and gas, coke concentration in solid, liquid water content[2] and gas composition including water vapor. Void fraction and particle size are estimated from

[2] Liquid and solids are lumped in one phase.

measured values, the pressure profile and (4).

$$\frac{\partial T_s}{\partial t} + k_1 (T_s - T_g) = g(T_s)$$
$$\frac{\partial T_g}{\partial t} + v\kappa_g \frac{\partial T_g}{\partial z} + k_2 (T_g - T_s) = 0$$
$$\frac{\partial x_c}{\partial t} = -M_C R_r(T_s)$$
$$\frac{\partial x_w}{\partial t} = -M_{H_2O} r_4$$
$$\frac{\partial x_j}{\partial t} + v\frac{\partial x_j}{\partial z} = r_j$$

where k_1, k_2 and k_3 are aggregated temperature dependent parameters and

$$g(T_s) = [k_3 R_r(T_s)\Delta H_r - k_3 R_f(T_s)\Delta H_f - k_3 r_{H_2O}\Delta H_v(T_s)]$$
$$r_1(T_s) = -R_r(T_s)$$
$$r_2(T_s) = 0$$
$$r_3(T_s) = R_r(T_s)$$
$$r_4(T_s) = r_{H_2O(v)}$$

Relations for $R_r(T_s)$ are given by [1], $R_f(T_s)\Delta H_f$ and r_{H_2O} by [7]. Kinetic parameters is only considered for coke combustion, fusion and drying in the model. Shrinkage and slump is not included. The heat of coke combustion is released to the solid phase, see discussion in [5]. Limestone is not utilised in the industrial plant, and is not included in the model.

B Notation

Parameters:

ε - void fraction

d_p - average particle diameter

h_m - mass transfer coefficient

h_c - heat transfer coefficient

k - thermal conductivity of gas

v - gas velocity

μ - viscosity

ρ - density

D_{ON} - axial gas dispersion coefficient (O_2-N_2)

G - mass flow rate of gas

G_s - gas volume per sinter mass

L - height of bed

p - pressure

κ_g - adiabatic constant

ΔH - heat of reaction

c_p - specific heat capacity

Dimensionsless numbers:

Re - Reynolds number: $Re = \frac{d_p G}{\mu}$

Sh - Sherwood number: $Sh = \frac{h_m d_p}{D_{ON}}$

Sc - Schmid number: $Sc = \frac{\mu}{\rho D_{ON}}$

Nu - Nusselt number: $Nu = \frac{h_c d_p}{k}$

Pr - Prandtl number: $Pr = \frac{c_p \mu}{k}$

References

[1] I. Muchi and J. Higuchi, "Theoretical Analysis of Sintering Operation," *Trans. Iron Steel Inst. Jpn.*, vol. 12, pp. 54–63, 1972.

[2] I. Dash and E. Rose, "An Analytical Study of Processes Occurring in an Iron Ore Sinter Bed," in *Automation in Mining, Mineral and Metal Processing* (F. Lancaster, ed.), IFAC Proc., pp. 649–659, 1977.

[3] F. Hoislbauer and C. Jaquemar, "Mathematical Modelling of Heat and Mass Transfer in Iron Ore Packed Bed Industrial Processes," in *Num. Met. Heat Transf.* (R. Lewis, K. Morgan, and B. Schrefler, eds.), vol. II, ch. 19, pp. 485–510, Wiley, 1983.

[4] E. Kasai, J. Yagi, and Y. Omori, "A mathematical model of sintering process considering melt-formation and solidification phenomena," in *Proc. Ironmak. Conf.*, vol. 43, pp. 241–249, AIME, 1984.

[5] M. Cumming and J. Thurlby, "Developments in Modelling and Simulation of Iron Ore Sintering," *Ironmak. Steelmak.*, vol. 17, no. 4, pp. 245–254, 1990.

[6] F. Patisson, J. Bellot, D. Ablitzer, E. Marlière, C. Dulcy, and J. Steiler, "Mathematical Modelling of Iron Ore Sintering Process," *Ironmak. Steelmak.*, vol. 18, no. 2, pp. 89–95, 1991.

[7] N. Nath, A. Da Silva, and N. Chakraborti, "Dynamic Process Modelling of Iron Ore Sintering," *Steel Research*, vol. 68, no. 7, pp. 285–292, 1997.

[8] R. Venkataramana, S. Gupta, P. Kapur, and N. Ramachandran, "Mathematical Modelling and Simulation of the Iron Ore Sintering Process," *Tata Search*, pp. 25–30, 1998.

[9] W. Kwon, Y. Kim, S. Lee, and K.-N. Paek, "Event-based modeling and control for the burnthrough point in sintering processes," *IEEE Trans. Contr. Sys. Tech.*, vol. 7, no. 1, pp. 31–41, 1999.

[10] T. A. Johansen and B. A. Foss, "Operating regime based process modelling and identification," *Computers and Chemical Engineering*, vol. 21, pp. 159–176, 1997.

[11] P. Dawson, "Part 2 Research Studies on Sintering and Sinter Quality," *Ironmak. Steelmak.*, vol. 20, no. 2, pp. 137–143, 1993.

[12] H. Toda, T. Senzaki, S. Isozaki, and K. Kato, "Relationship Between Heat Pattern in Sintering Bed and Sinter Properties," *Trans. Iron Steel Inst. Jpn.*, vol. 24, pp. 187–196, 1984.

[13] J. Hinkley, A. Waters, D. O'Dea, and J. Litster, "Voidage of Ferrous Sinter Beds: New Measurement Technique and Dependence on Feed Characteristics," *Int.J.Miner.Proc.*, vol. 41, no. 1-2, pp. 53–69, 1994.

[14] N. Wakao and S. Kaguei, *Heat and Mass Transfer in Packed Beds*, vol. 1 of *Topics in Chemical Engineering*. Gordon and Breach Science Publishers, 1982.

[15] F. Patisson, J. Bellot, and D. Ablitzer, "Study of Moisture Transfer During the Strand Sintering Process," *Metall. Trans.*, vol. 21B, pp. 37–47, 1990.

[16] O. Slupphaug and B. A. Foss, "Quadratic Stabilization of Discrete-Time Uncertain Nonlinear Multi-Model Systems using Piecewise Affine State-Feedback," *Int. J. of Control*, 1999. To appear.