On the Controllability of Fermentation Systems

Christian Kuhlmann

A thesis submitted for the degree of Doctor of Philosophy of the University of London

Department of Chemical Engineering
University College London
London WC1E 7JE

March 1998

PAGINATION AS IN ORIGINAL

Contents

Abstract

This thesis concerns the controllability of fermentation processes. Fermentation processes are often described by unstructured process models. A control system can be used to reduce the effect of the uncertainties and disturbances.

A process is called controllable if a control system satisfying suitably defined control objectives can be found. Controllability measures based on linear process models are identified. The idealised control objective for perfect control allows fast evaluation of the controllability measures. These measures are applied to compare different designs of a continuous fermentation process by identifying the controllability properties of the process design.

The operational mode of fed batch fermentations is inherently dynamic. General control system design methods are not readily applicable to such systems. This work presents an approach for the design of robust controllers suitable for these processes. The control objective is to satisfy a set of robustness constraints for a given set of model uncertainties and disturbances.

The optimal operation and design problems are combined into a single optimal control problem. The controller design is integrated into the process design problem formulation. In this way the control system and the process are designed simultaneously. Different problem formulations are investigated. The proposed approach is demonstrated on complex fermentation models. The resulting operating strategies are controllable with respect to the aims of control.

List of Figures

Chapter 1

Introduction and Motivation

This thesis concerns the integration of design and control of fermentation processes. Fermentation processes can often be described by unstructured process models. The models can be used in model based design procedures.

The fermentation process is often followed by several downstream processes in order to recover the final product from the fermentation broth. The successful operation of downstream processing steps depends on the outcome of the fermentation process. Therefore it is important to design a fermentation process which can be controlled such that a successful fermentation which meets the requirements of the downstream processes can be guaranteed. An example is the washout problem of a continuous fermenter. The operating point for optimal productivity is often close to the wash-out state. If the process is designed without taking uncertainties or disturbances into account it can happen that the 'real' process operates at the wash-out state. Modifications carried out at a later stage to the process may be expensive and should be avoided if possible. Here a systematic approach is required to find out how far to 'back off' from the optimum in order to guarantee problem free operation of the process.

One approach to process design is to optimise its model with respect to an objective function representing the design objectives such as annual profit, minimal environmental impact or maximum amount of product in a given time.

If the dynamic aspects of a continuous process are ignored the process can be modeled by algebraic equations leading to a parameter optimisation problem. For fed batch processes the dynamics cannot be ignored since these processes are not operated at steady state. Due to the dynamic mode of operation, the optimisation problem is posed as an optimal control problem. The uncertainties in the parameter values and in the disturbances are often ignored in the optimisation problems. The process design is usually optimised for the nominal values of the parameters and disturbances. Possible variations in the values of the parameters which may be encountered during operation and the impact these have on the operational objectives are often not taken into account at the design stage.

Once a plant is commissioned a control system is added to reduce the effect of uncertainties and external disturbances. Some of the process states, the controlled variables, are measured or estimated from other variables. The values of these variables are read by a controller. The controller calculates appropriate values for some of the input variables which are selected as the manipulated variables. Such a scheme is called a feedback controller.

The control system is tuned in order to satisfy some control objective. In continuous process plants a common control objective is to move rapidly to the set points after a disturbance strikes the plant. The set points may be calculated on-line based on process measurements. The design engineer is interested in a control system which is capable of achieving the control objective. If there exists such a control system the design is called controllable. *Controllability* is defined here by the following statement:

A design is controllable if a control system which satisfies the control objective exists.

The term 'control objective' is not specified in more detail since different control objectives may be defined for specific problems. Depending on the application, not all controllers can be implemented e.g. due to hardware restrictions. For example, in plants processing flammable material pneumatic controllers are often considered to be the only option. In this case the plant should be controllable using pneumatic controllers only and it is not of much benefit to establish wether the plant is controllable using any other controller. In general, it is not necessary to restrict the controller to be a feedback controller. Some control objectives could be satisfied using open loop control without implementing a control system.

It is desirable that controllability is incorporated into the design procedure as early as possible in order to avoid failures once the plant is built. Mathematical tools have been developed to establish whether or not a design is controllable for specific control objectives. The tools focus mainly on continuous processes and are not easily adapted for dynamic processes. They are applied once the design is fixed. If the process is found not to be controllable the engineer has to iterate through process designs to produce an economically attractive design with better controllability properties. Without carrying out controllability analysis it is not always clear what the engineer has to change in order to achieve better controllability. Several approaches have been proposed which integrate controllability analysis directly into the design stage such that the final process is controllable.

Since fermentation processes are operated in continuous, batch and fed batch modes it is desirable to identify and develop suitable procedures which integrate controllability into the design procedure of such processes. The goal of this thesis is to present procedures which incorporate controllability into the design stage of fermentation processes, because current controllability analysis is not directly applicable.

The first part of this thesis reviews approaches which assess controllability. The modeling of fermentation processes is reviewed and appropriate models are identified. In the second chapter the mathematical foundations necessary for the design procedure proposed in the third chapter are explained. The third chapter presents a procedure for controller design which is suitable for dynamic processes. This approach is incorporated into an optimisation based design procedure. An algorithm to solve the integrated control and design problem is described and an alternative problem formulation is examined. Three case studies, presented in the fourth chapter, demonstrate how the approach incorporates controllability at the design stage of a fermentation process.

Chapter 2

Design and Controllability

This chapter reviews the area of design and controllability of fermentation systems. The first part concentrates on evaluation of controllability and how to integrate different evaluation methods into the design stage of a process. The second part of the chapter presents the modeling of fermentation systems using unstructured models. These models are widely used to describe the behavior of fermentation processes ([62]). The third part of this chapter reviews the area of controllability of fermentation systems. The fourth and last part of this chapter reviews optimal control techniques applied to fed batch fermentation as a method to design these processes.

2.1 Controllability

In this section different controllability analyses are reviewed. The work on controllability can be loosely classified into two, not necessarily distinct, classes. One class of methods aims to measure controllability of a given plant. These are called evaluation methods. The second class of methods integrates controllability aspects into the design procedure. It is necessary to have a measure of controllability before it can be integrated systematically into a design approach.

Other reviews concerning controllability can be found in [59, 60, 61, 67, 68, 85, 83, 84, 63].

Since plants rarely operate at the nominal design conditions or at the nominal steady states, it is important to include other aspects of plant operation into the design of the process. Traditionally, the plant engineer designs the process without considering the control structure. He/she designs the plant and the control engineer decides how to control the plant and how to choose the control inputs and the controlled outputs. Here the term design includes selecting a control structure, i.e. deciding which variables should be controlled outputs and which should be control inputs. The controlled plant should be able to reject disturbances and to be steered in an 'acceptable manner' between different operating points. These properties should be ensured in the presence of uncertainties in the model on which the design is based. A mathematical model of the process is needed to evaluate controllability and estimates of the likely disturbances and model uncertainties are also required.

Morari ([59]) reviewed the existing tools to assist in the design of the process and to judge the controllability of a process. He divided the tools mainly into three generations. The first generation of synthesis tools consisted of computer programs which did the routine calculations. The second generation automated the design procedure. This generation of computer programs was based on optimisation methods. It is possible to integrate dynamic aspects of the design into this procedure ([45]). The third generation of tools gives some insight into the process behavior and therefore gives the design engineer some understanding of why a certain design may fail to have an acceptable dynamic behavior. These tools are mainly analytical. They relate the process properties such as its dynamic and steady state behavior to the design and process model. A suitable tool to assess controllability should be able to quantify controllability so as to compare different designs. A controllability measure that aims to quantify how controllable a design is, will be called a 'quantitative concept'. A measure that does not quantify controllability, will be called a qualitative concept. The qualitative concepts only indicate that it is possible to control the process but not if it is easy or difficult to control.

Other kind of controllability measures are the structural controllability measures (e.g. [48]). Structural methods take only the structure of the process model into account. The numerical values of the process parameters are ignored. These methods are not discussed in detail here. They can be viewed as a part of the qualitative concepts.

Currently there are no general tools or systematic procedures for assessing controllability which can be applied to any process or any control objective (e.g. operating at a certain set point, start-up or shut down). The different approaches can only be applied to a limited class of processes. There are, for example, methods to test controllability either qualitatively or quantitatively if the process can be described by a linear model. This can often be done for continuous processes whose behaviour is sufficiently linear around an operating point. It is possible to quantify the controllability of such a process (e.g. [59, 84]). However none of these methods indicate if it is possible to realise a start up of the plant which leads to the desired working point. This is a shortcoming of the local methods which are based on the linearisation of the nonlinear process.

There are many definitions of the term controllability available in the literature. Engineers from different disciplines have different definitions of this term. The term controllability is often defined to suite particular objectives.

Morari ([60]) pointed out the first time the term controllability appeared in the literature. It was used by Ziegler and Nichols ([104]). They defined controllability as

the ability of the process to achieve and maintain the desired equilibrium value

This is a quite general definition which can be widely applied. Probably the best known definition of the term controllability arose from control engineering. Kalman [39] introduced the concept of controllability for a linear system.

A deeper insight into the controllability problem is obtained from the definitions given by Rosenbrock [75, 76]. In these definitions, the Kalman concept of controllability is included. Rosenbrock explained controllability as follows

In engineering practice, a system is called controllable if it is possible to achieve the specified aims of control, whatever these may be. By extension, the system is said to be more or less controllable according to the ease or difficulty of exerting control. These ideas of controllability have been embodied in a number of mathematical definitions, which however do not exhaust the possible meaning.
(Rosenbrock, 1970)

This definition indicates that controllability should be quantified, however how this can be done was not explained by Rosenbrock. He considered various qualitative measures.

2.1.1 Evaluation of Controllability without Disturbances

In this section analytical tools for assessing controllability without specifying any disturbances are described. These methods are limited to linear systems since it is difficult to investigate the controllability properties of nonlinear systems in general.

A linear process model can be obtained by either linearising the nonlinear process model using a Taylor series or by identifying a linear model by using experimental data. These models are only valid locally at the operating point.

2.1.1.1 Qualitative Concepts

In [75, 76] Kalman introduced a controllability concept which is called pointwise state controllability.

The system is pointwise-state controllable if, given any two states c_0 and c_1, there exists a time $t_1 > 0$ and a control u defined on $[0, t_1]$ which takes the state from $x(0) = c_0$ to $x(t_1) = c_1$.

This definition of controllability answers the question whether it is possible to construct an input $u(t)$ that steers the system from any state in the state space to any desired state in a finite time. There are no restrictions on how the desired state is reached or on the input. If a system is pointwise state controllable this does necessarily not mean that it is possible to maintain the system at the desired state; it only means that it is possible to move the system to this state. It is worth noting that if it is not possible to move a system to a desired state, then it is not possible to maintain the system at this state.

A concept closely related to pointwise-state controllability is that of output controllability. A definition of output-controllability can be found in [72].

A system is output controllable if there exists a control policy $u(t)$ which will steer the system from any given initial output state y_0 to any other desired output state y_d in a finite time.

If a system is output controllable it is possible to steer the system to any state in the output space. But this does not necessarily mean that is possible to find a control sequence that is able to maintain the output at this state. Pointwise state controllability and output controllability indicate only the possibility to steer all or a number of states to the desired states but it does not give any information on how this can be done or on the control sequence which achieves this.

Rosenbrock [75] introduced the term functional controllability. This term was also called functional reproducibility [12].

The system is functionally controllable if given any suitable vector y of output functions defined for $t > 0$, there exists a vector u of inputs defined for $t > 0$ which generates the output vector y from the initial condition $x(0) = 0$

If a system is functionally controllable it is possible to steer the system to any state in the output space and maintain it there. Therefore a system must be output controllable to be functionally controllable. Otherwise it would not be possible to bring the system to the desired steady state. This concept however restricts the type of outputs. The restriction is that the output function should satisfy certain smoothness conditions that are not of practical interest. This concept does not put any restriction or limitation on the input.

Another concept of controllability was defined by Rosenbrock [75] to deal with control problems which are related to nonminimum phase systems. It is an extension of functional controllability.

The system is controllable (l) if it is functional controllable and if in addition all the zeros of the transfer function matrix $G(s)$ lie in the open left plane.

Rosenbrock [76] explained the different definitions of controllability:

These different kinds of controllability are appropriate in different circumstances. Pointwise state controllability is appropriate in rocket guidance problems, and in some problems of batch processing or start-up or grade-change in industrial plants. Functional controllability is appropriate for the usual servo following problem of industrial control in which the elements of the vector y must approximate to constant desired output values. Controllability (l) ensures that difficulties associated with nonminimum phase response do not arise. It is important to note that pointwise state controllability is usually called just "controllability" in the literature. This leads to a presumption that it implies functional controllability, which is not true. Pointwise state controllability and functional controllability are distinct properties, and either can exist without the other.

A system can be pointwise state controllable and not functionally controllable and vice versa. If the system is pointwise state controllable this does not imply that the output states can be maintained at the desired state. If the output states can be steered to certain states and maintained there then the system is functional controllable but this does not mean that all the states of the system are controllable.

Skogestad [83] discussed the use of the term controllability

Unfortunately, in the 60's the term "controllability" became synonymous with the rather narrow concept of "state controllability" introduced by Kalman, and the term is still used in this restrictive manner in the system theory community. "State controllability" is the ability to bring a system from a given initial state to any final state (but with no regard to the quality of the response between these two states).

This concept is of interest for realizations and numerical calculations, but as long as we know that all unstable modes are both controllable and observable, it has little practical significance.

The above indicates that as long as the plant is stabilisable, pointwise state controllability is not of any interest. A plant is stabilisable means that there exists a controller which makes the unstable plant stable. Skogestad points out further that functional controllability is the more appropriate concept.

Russell and Perkins (1987) gave the following reasons why functional controllability has several advantages over state controllability.

- *Usually it is neither necessary nor practical to control and observe every state variable in the system. Consider for example a model of a distillation column which includes a number of states for each plate.*
- *State controllability does not guarantee that it is possible to independently specify arbitrary trajectories of the chosen set of output variables, whereas functional controllability does. This is important since the major goal of regulatory control is usually to maintain the plant at some steady state.*
- *Although the initial and final states are specified in the definition of state controllability, no conditions can be imposed on the trajectory between those states, or on the behavior of the trajectory after the final time.*

In connection to the definition of functional controllability Rosenbrock (1970) points out

In many industrial situations the aim of control is to make the output vector y of a plant take a certain form as a function of time. For example the regulator problem is to make $y_i(t) = v_i(t)$, where v_i are constant, or slowly changing, desired outputs.

From the above it becomes quite clear that most of the researchers cited agree that functional controllability is the most appropriate definition for the controllability assessment.

To determine the most suitable definition of controllability it is necessary link it to the objectives of a control system. One reason to control a bio-processing plant is to guarantee constant product quality. To satisfy this demand the control system should be able to keep certain states which can be identified as quality parameters (e.g. concentration) constant. These states are defined as outputs. As long as the resulting control system is stabilisable, functional controllability guarantees realisable control.

Another objective of a control system is to maintain the processing plant at optimal or near optimal operational conditions. This does not mean only that the outputs should be kept at their optimum levels but also that all other states determined by the optimisation scheme should be kept constant. A possible solution to this problem is to define all states as outputs and to apply functional controllability to the system. In this case, a necessary condition for functional controllability is pointwise state controllability and here the concept of output controllability and pointwise state controllability are the same. This is so because the state space and the output space are identical. Functional controllability ensures that the new steady state can be reached and maintained. This condition includes the less restrictive condition of output controllability which requires that the new steady state can be reached.

If the objective of the control system is to operate the plant safely, then functional controllability is sufficient, but is not necessary. To be sufficient, all critical states important for safety have to be defined as outputs. In this case functional controllability ensures that the plant operates safely if it is stabilisable.

A shortcoming of all these definitions is that they put no constraints or conditions on the outputs or inputs. The above definitions imply that the plant is controllable with respect to unbounded and unlimited inputs (e.g. speed of the actuator).

2.1.1.2 Quantitative Concepts

In the area of quantitative measures of controllability, a distinction is made between open loop and closed loop methods. Open loop methods use the model of the plant only and no controller or control structure is assumed to evaluate the controllability properties. For closed loop methods a controller is assumed and the controllability is evaluated for this controller-plant configuration.

Skogestad (1994) introduced a general definition of input-output controllability

The ability to achieve acceptable control performance, that is, to keep the outputs (y) within specified bounds from their set points (r), in spite of unknown variations in the plant (e.g. disturbances (d) and model perturbations) using available inputs (u) and available measurements.

This definition of controllability is suited for control systems for chemical processes. The term 'the ability to achieve acceptable control performance' indicates again that quantification of controllability is necessary.

One of the first systematic concepts to quantify controllability analytically was Morari [59]. He defined dynamic resilience to be

"(Dynamic) resilience" is the quality of the regulatory and the servo behavior which can be obtained for the plant by feedback.

Morari showed that dynamic resilience is related to the realisibility of the transfer function inverse. Russell and Perkins [77] related functional controllability to the term dynamic resilience. Dynamic resilience is an attempt to quantify functional controllability and investigate it in more depth. This new concept incorporates that of controllability (1) defined by Rosenbrock. Rosenbrock [75] showed that a system with a transfer function matrix $G(s)$ is functionally controllable if and only if $G(s)$ is nonsingular. This condition is equivalent to that given in the definition of functional controllability. If the transfer function matrix is defined as

$$G(s) = \frac{y(s)}{u(s)} \tag{2.1}$$

then the expression

$$u(s) = G^{-1}(s)y(s) \tag{2.2}$$

gives the input trajectories which generate the required output trajectories. The necessity of the non-singularity condition is obvious, since the input u can only be generated if $G^{-1}(s)$ can be computed which is only possible if $G(s)$ is nonsingular. There are features other than non-singularity that may cause problems when computing the required input, i.e. when inverting the transfer function matrix. Morari [59] identified three potential problems. These are

- nonminimum phase behavior i.e. right half plane zeros and time delays.
- physically bounded inputs
- plant/model mismatch

Computing u in the presence of right half plane zeros can result in unrealisable unbounded inputs i.e. unstable inverse transfer function matrices. The presence of time delays leads to predictive elements in the inverse transfer function that makes the on-line calculation (which is necessary for feedback control) of u impossible. As a measure of resiliency Morari introduced the integral square error of the output y. If the input computed for steering the output, y, to the desired output exceeds the physically realizable input, functional controllability can not be guaranteed anymore. Morari [59] showed how singular values can be used as a measure of the 'gain' of a Multi Input Multi Output (MIMO) system, and derived a bound on the disturbance magnitude for which the control input can be computed without saturating. The condition number is defined as the ratio of the maximum to minimum singular values of the transfer function matrix. For a plant to be controllable it is a desirable to have a condition number close to one and a large minimum singular value. A plant with a large condition number is said to be ill-conditioned. If the input is computed on the basis of an uncertain model, then it is not known if the process will reach the desired output. The condition number can be used to quantify the effect of how sensitive the design is to model uncertainties.

Hovd and Skogestad [36] reviewed the Relative Gain Array (RGA), introduced by Bristol [11] and the Performance Relative Gain Array (PRGA) and related them to controllability properties. Wolff and Skogestad [84] described tools available for controllability analysis.

The quantitative controllability concepts are basically extensions of functional controllability. This was shown by Russell and Perkins [77] who related functional controllability to dynamic resilience. In [84] it is pointed out that functional controllability can be quantified. Dynamic resilience quantifies functional controllability. These measures serve as indicators for the controllability of the process.

All the methods discussed so far are limited to linear process models. This is sufficient for a plant operating close to one steady state. For a plant operating at different steady states or for a plant that does not have a steady state (e.g. batch processes) these methods are not applicable.

The controllability measures give an idea of how difficult a certain design is to control. They are useful for ranking different designs with respect to their controllability. Using these measures designs with similar steady state economics can be compared. A problem arises when the designs have different steady state economics and different controllability properties. Is a design with good steady state economics that is difficult to control preferable to a design with less promising steady state economics but that is easier to control? The measures do not take into account the costs that are associated with the choice of output and manipulated variables. For example, if a big control action is required to steer a process to the set point this would indicate bad controllability. But if this control input is comparably cheap it might be preferred to a small but expensive control action. There is no method which translates these controllability measures directly into economical benefit.

2.1.2 Integrating Controllability into Process Design

Different approaches are reviewed here whose aim is to integrate controllability into the design procedure.

2.1.2.1 Process Flexibility

In this section methods are reviewed which incorporate parameter uncertainty into the design procedure. The design problem is formulated as an optimisation problem. Other reviews of optimal process design under uncertainty are in [25, 27].

Assume first that the process mainly operates at steady state and that the dynamic transients of the process between different steady states can be neglected. A model of the process at steady state is given by a set of nonlinear algebraic equations

$$h(x, u, d, p) = 0 \tag{2.3}$$

where x is the state vector of the system, u is the vector of control inputs which can be adjusted during operation, d describes the vector of design variables which remain constant once they are selected and p is a vector of uncertain parameters not known exactly in advance. P represents uncertainties in the process model and external disturbances. The set of equations Eq. 2.3 can be solved for a given vector of design variables d, control inputs u and uncertain parameters p

$$(u, d, p) \quad \Longrightarrow \quad \boxed{h(x, u, d, p) = 0} \quad \Longrightarrow \quad x \tag{2.4}$$

Thus $\dim(h) = \dim(x)$ and the states of the system can be seen as a function of the 'inputs' u, d and p

$$x = f(u, d, p) \tag{2.5}$$

implicitly defined by the model of the process. It is assumed here that the Jacobian

$$J = \frac{\partial h}{\partial x} \tag{2.6}$$

is invertible for the required inputs.

Now consider the following optimisation problem

$$\begin{aligned} \min_{x,u,d} \quad & C(x, u, d, p) \\ \text{s.t.} \quad & h(x, u, d, p) = 0 \\ & g(x, u, d, p) \leq 0 \end{aligned} \tag{2.7}$$

where $C(x, u, d, p)$ represents some objective function such as economic profit and $g(x, u, d, p) \leq 0$ is a set of design constraints to be satisfied by the process design. The parameter p is not known exactly and is assumed to be bounded in a set P

$$P = \{p | \underline{p} \leq p \leq \overline{p}\} \tag{2.8}$$

where $\underline{p}$ and $\overline{p}$ denote the lower and upper bounds of the parameter, respectively. Problem 2.7 is not well defined in this form since a different optimal solution of the problem corresponds to each value of the parameter vector p. If a specific value of p is assumed (e.g. a nominal value p_{nom}) this problem becomes a well defined optimisation problem and can be solved under appropriate assumptions. The optimal solution to this problem $(x^\star, u^\star, d^\star)$ guarantees optimal steady state operation and constraint satisfaction if the uncertain parameters p are at the values used for the optimisation. This is not guaranteed for parameters other than the ones selected for the optimisation and the solution is not optimal anymore. Depending on the problem it cannot, in general, be assumed that the plant operates close to the optimum even if the real parameters p are close to the ones used in the optimisation.

If we assume that the control variables cannot be adjusted during operation, the design problem under uncertainty can be reformulated as a deterministic problem. A control variable which cannot be adjusted during operation is equivalent to a design variable which is fixed over the whole time horizon of operation. Assume, for now, that there are no design constraints in the original problem and a probability distribution function for the uncertain parameter vector is available. Then the original objective function can be replaced by its expected value

$$\begin{aligned} \min_{x,d} \quad & E_{p \in P}\{C(x, d, p)\} \\ \text{s.t.} \quad & h(x, d, p) = 0 \end{aligned} \tag{2.9}$$

which can for practical purposes be approximated by the weighted sum

$$\begin{aligned} \min_{x_1,\dots,x_n,d} \quad & \textstyle\sum_{i=1}^{n} w_i C_i(x_i, d, p_i) \\ \text{s.t.} \quad & h_i(x_i, d, p_i) = 0, \quad i = 1, \dots, n \end{aligned} \tag{2.10}$$

where w_i are the respective weights. This approximation can be interpreted as a multiperiod design problem. The problem is to optimise a plant which is described by the model $h(\cdot) = 0$. This plant is operated at n time intervals. The i-th time interval lasts the relative time w_i and the vector of uncertain parameters in this period is p_i. The state vector in the i-th period is x_i.

Additionally it may be required that the design constraints hold for all possible realisations of the uncertain parameter vector. This results in the following optimisation problem

$$\begin{aligned} \min_{x,x_1,\dots,x_n,d} \quad & \textstyle\sum_{i=1}^{n} w_i C_i(x_i, d, p_i) \\ \text{s.t.} \quad & \left.\begin{aligned} h(x_i, d, p_i) = 0 \\ g(x_i, d, p_i) \le 0 \end{aligned}\right\} \quad i = 1, \dots, n \\ & \left.\begin{aligned} h(x, d, p) = 0 \\ g(x, d, p) \le 0 \end{aligned}\right\} \quad \forall p \in P \end{aligned} \tag{2.11}$$

This problem formulation ignores the fact that the control parameters can be adjusted during operation. Such a problem is called permanently feasible ([2, 21]) since the design constraints hold for all possible parameter combinations without adjustment of any of the process variables. This approach may lead to conservative results.

In order to achieve less conservative results it is possible to take the adjustment of the control variables into account in the problem formulation. The problem is then formulated as a two stage problem. In the first stage the design variables are selected (design stage). In the second stage (the operating stage) the control variables are adjusted to force feasible operation if possible. Mathematically this is equivalent to solving

$$
\begin{array}{lll}
\min\limits_{d} & \mathrm{E}_{p\in P}\{C^\star\} & \text{design stage} \\
\text{s.t.} & \left.\begin{array}{rl} C^\star = \min\limits_{x,u} & C(x,u,d,p) \\ \text{s.t.} & h(x,u,d,p) = 0 \\ & g(x,u,d,p) \leq 0 \end{array}\right\} & \text{operating stage}
\end{array}
\tag{2.12}
$$

where $C^\star$ is the optimal operation of the plant for a given parameter vector p. For some design parameters d it may not be possible to solve the sub-problem for the operating stage. In [64] a penalty term is added to the objective function in this case. Such an approach is suitable if some of the design constraints are soft constraints. Violation of soft constraints does not lead to a disaster which has to be avoided under all circumstances. For example not meeting product specifications leads to an economic penalty due to the loss of product which can be expressed as a penalty term in the objective function. However, finding an appropriate penalty for the violation of soft constraints is not straightforward in realistic situations. Haleman and Grossmann ([30]) defined a constraint which can be added to the problem to ensure that the process can be operated under all possible uncertainties. If this constraint is included in the problem formulation the penalty term for infeasible operation is avoided

$$
\forall p \in P \quad \{\exists u \quad (\forall j \in J \begin{bmatrix} h(x,u,d,p) = 0 \\ g_j(x,u,d,p) \leq 0 \end{bmatrix})\} \tag{2.13}
$$

where $g_j(x,u,d,p)$ is the j-th component of $g(x,u,d,p)$. This constraint, called the feasibility constraint, ensures that there exists a control variable, u, such that the design constraints are satisfied for all possible parameters p. In [30] it is shown that the feasibility constraint 2.13 is equivalent to

$$\begin{aligned} \max_p \min_{x,u} \max_j \quad & g_j(x,u,d,p) \leq 0 \\ \text{s.t.} \quad & h(x,u,d,p) = 0 \end{aligned} \tag{2.14}$$

The resulting two-stage problem with the feasibility constraint cannot be solved directly in this form. The first simplification is to discretise the uncertain parameter space and then to approximate the calculation of the expectation value by a weighted sum. The problem is then given by

$$\begin{aligned} \min_{x_1,\ldots,x_n,u_1,\ldots,u_n,d} \quad & \textstyle\sum_{i=1}^{n} w_i C_i(x_i,u_i,d,p_i) \\ \text{s.t.} \quad & \left.\begin{aligned} h(x_i,u_i,d,p_i) = 0 \\ g(x_i,u_i,d,p_i) \leq 0 \end{aligned}\right\} \quad i = 1,\ldots,n \\ & \begin{aligned} \max_p \min_{x,u} \max_j \quad & g_j(x,u,d,p) \leq 0 \\ \text{s.t.} \quad & h(x,u,d,p) = 0 \end{aligned} \end{aligned} \tag{2.15}$$

If there are no control variables this formulation reduces to Problem 2.11. In the two-stage formulation which includes the feasibility constraint, 2.13, the control variables are chosen such that the design constraints are satisfied at all times. In practice this is likely to be difficult to achieve since there is hardly enough information available to achieve this goal. Therefore it is more realistic to consider a realisable control system which can be adjusted.

One of the control objectives is to obtain offset free steady state control. This can be achieved in practice under appropriate assumptions if integral action is included in the controller. Instead of considering the control variables at the optimisation stage, it is then possible to integrate the control system into the process model. This can be achieved by adding the set points as additional equality constraints to the model ([97]). The control variables are turned into state variables which are determined through the set points. This is only true if there are as many control variables as set points (square plant). No control variables are present in this formulation and the flexibility problem becomes equivalent to the permanent feasible program. One level of optimisation drops out of the feasibility constraint in the flexibility problem. A different set of control inputs and controlled variables can be incorporated into this formulation when integer variables are included in the model. This yields a steady state controllable design where the controlled states and control inputs are already chosen.

Methods which design continuous flexible plants under uncertainty and use dynamic models are proposed in [97, 22, 3, 4, 57, 68]. Walsh & Perkins ([97]) reformulate the dynamic optimisation problem as a Non linear Programming (NLP) problem. Uncertainty and disturbances are specified as a bounded set. Structural changes, such as different control structures or different plant designs, are left to the design engineer and are not included in the optimisation problem. These structural changes are taken into account in [3, 4] and [57] by introducing integer variables, such that the final problem is a Mixed Integer Nonlinear Programming (MINLP) problem.

2.1.2.2 Trade Off between Process Design and Controllability

Multi-objective optimisation can be used to design controllable plants. In this formulation an optimisation problem with two objective functions is formulated. The first objective function is an economic design measure and the second objective function describes a controllability measure. In a multi-objective optimisation framework ([18]) these two measures can be traded off against each other and the designer can specify 'how much controllability' is necessary. In [25] a measure of flexibility is introduced as a second objective. In [65] the minimum singular value is proposed as a controllability measure. In [51] the RGA is proposed as a controllability measure and structural changes are taken into account in the plant model.

2.1.2.3 Design with Specified Disturbances

The design of a process which is controllable with respect to specific disturbances is similar to those in Section 2.1.2.1. The difference is that controllability is not evaluated for a continuous set of disturbances. Single disturbances are specified and the plant is designed for good controllability for these disturbances. In [82] PI controllers are implemented on the model and the control structure selection for a distillation column is included into the problem. In [10] a **M**odel **P**redictive **C**ontroller (MPC) is used to evaluate the control performance. The control objective is the **I**ntegral **S**quare **E**rror (ISE) and the objective function is augmented with the ISE and a penalty parameter. Thus bad control performance penalises the objective function. This approach is demonstrated on a continuous fermenter with a substrate recycle and the problem is solved for different values of the penalty parameter.

2.1.2.4 Risk Conscious Operation of Batch Plants

Terwiesch et al. ([92, 93]) propose an approach to design optimal profiles for batch processes. The optimal control problem is transformed into a NLP problem with uncertain parameters. Different objective functions are proposed to formulate the problem. As discussed earlier in the steady state situation the expectation value over the uncertain parameter space can be minimised. The uncertain parameter space is discretised. This approach does not contain the feasibility constraint. Therefore feasible operation can only be guaranteed for parameter values p_i contained in the discretised set.

2.1.3 Evaluation of Controllability with Disturbances

Some evaluation methods assess controllability after the process design is fixed. These methods are independent of other aspects of the design process and are widely applicable. Their strength is at the same time balanced by a weakness in that it is often unclear how these methods can be integrated into a certain design procedure. This is particularly important if it is found that the design is not controllable.

An approach which takes disturbances into account is described in [85, 84, 67]. Here the scaled control input required to reject the worst case disturbance is used as a measure of controllability. A linear system at steady state can be described by an input-output model in deviation variables

$$y = Gu + G_p p \tag{2.16}$$

where u denotes the control variables and p the disturbances. It is assumed that the bounds on the control input are symmetric and the available input is scaled such that

$$\underline{u} = -1 \quad \text{and} \quad \overline{u} = 1 \tag{2.17}$$

The disturbance is scaled, equivalently, as

$$\underline{p} = -1 \quad \text{and} \quad \overline{p} = 1 \tag{2.18}$$

A possible control objective is to require that the output stays within certain bounds. The model 2.16 is scaled such that these bounds are

$$\underline{y} = -1 \quad \text{and} \quad \overline{y} = 1 \tag{2.19}$$

Another common control objective is to achieve perfect control which implies that there is no steady state offset ($y = 0$). If the dynamic linear model is analyzed in the frequency domain, the same analysis can be done over all frequencies. Note that a control objective of zero offset is much too stringent at high frequencies and may not be achieved. The controllability problem at steady state with respect to disturbance rejection can be formulated as an optimisation problem to determine the worst case output over all disturbances. The problem to be solved is

$$\begin{aligned} Y_{min} = \max_{p} \min_{u} \quad & |y| \\ \text{s.t.} \quad & Gu + G_p p = y \\ & |d| \leq 1 \\ & |u| \leq 1 \end{aligned} \tag{2.20}$$

where $|\cdot|$ denotes a vector norm. If the solution to this problem satisfies the control objective (i.e. $Y_{min} = 0$), the process is steady state controllable, otherwise the process is steady state uncontrollable.

In order to measure how easy or difficult it is to control a process, the magnitude of the control input required to achieve perfect control for the worst case disturbance may be considered as a measure for controllability ([46, 98, 85]). This worst case control input is the solution of the following optimisation problem. If the control objective is $|y| \leq 1$, the problem is

$$\begin{aligned} U_{min} = \max_p \min_u \quad & |u| \\ \text{s.t.} \quad & Gu + G_p p = y \\ & |y| \leq 1 \\ & |d| \leq 1 \end{aligned} \tag{2.21}$$

If the control objective is perfect control the problem is

$$\begin{aligned} U_{min} = \max_p \min_u \quad & |u| \\ \text{s.t.} \quad & Gu + G_p p = 0 \\ & |d| \leq 1 \end{aligned} \tag{2.22}$$

Note that if perfect control is not possible, this problem is not feasible. A small U_{min} implies that a process is easily controlled since there is plenty of input to react to changes.

For a square plant (i.e. $\dim(y) = \dim(u)$) a more detailed analysis of the problem is possible since a unique solution exists for the control input. The control input which is required to achieve perfect control is

$$\begin{aligned} U_{min} = \max_p \quad & |u| = |G^{-1}G_p p| \\ \text{s.t.} \quad & |p| \leq 1 \end{aligned} \tag{2.23}$$

This determines the control input which is required to reject, perfectly, the worst disturbance. Since we are interested in the absolute value of the largest control input it is mathematically correct to choose the infinity-norm for u and d.

$$U_{min} = ||u||_\infty = \max_i ||(G^{-1}G_p)_i||_1 \tag{2.24}$$

where $\max_i ||(\cdot)_i||_1$ represents the largest row sum. If, instead, the two-norm is chosen for u and d and a single disturbance ($G_p = g_p$) is considered the input for the worst case disturbance ($d = 1$) is

$$||u||_2 = ||G^{-1}g_p||_2 \tag{2.25}$$

It is clear that these measures are indicators and are not absolute. It is possible that the required control input for perfect control is larger than one. Depending on the direction of the control input, it may not be realisable. This is demonstrated in Figure 2.1 where the two-norm is visualized for a two-dimensional control input. The dashed box indicates the infinity norm and the true bounds on the control variable. The circle indicates the two-norm. The control input u_b is not realizable. The control input u_a (which has the same length) is realisable. The advantage of using the two-norm is that a further analytical treatment of the problem is possible. By using the singular values of the transfer function, a bounded solution of this worst case control input can be derived ([67])

$$\frac{||g_p||_2}{\overline{\sigma}(G)} \leq ||u||_2 \leq \frac{||g_p||_2}{\underline{\sigma}(G)} \tag{2.26}$$

where $\overline{\sigma}(G)$ and $\underline{\sigma}(G)$ are the maximum and minimum singular values of G, respectively. The exact value for $||u||_2$ depends on the direction of the disturbance. Eq. 2.26 shows that a general requirement is to have $\underline{\sigma}(G)$ as large as possible in order to minimise the required magnitude of the control input. A condition which ensures that a disturbance can be rejected independent of the direction is $\underline{\sigma}(G) = 1$. This might be too pessimistic since such a disturbance may never occur.

In cases where particular disturbances are expected the performance can be evaluated in terms of the disturbance condition number. Normalising the magnitude of

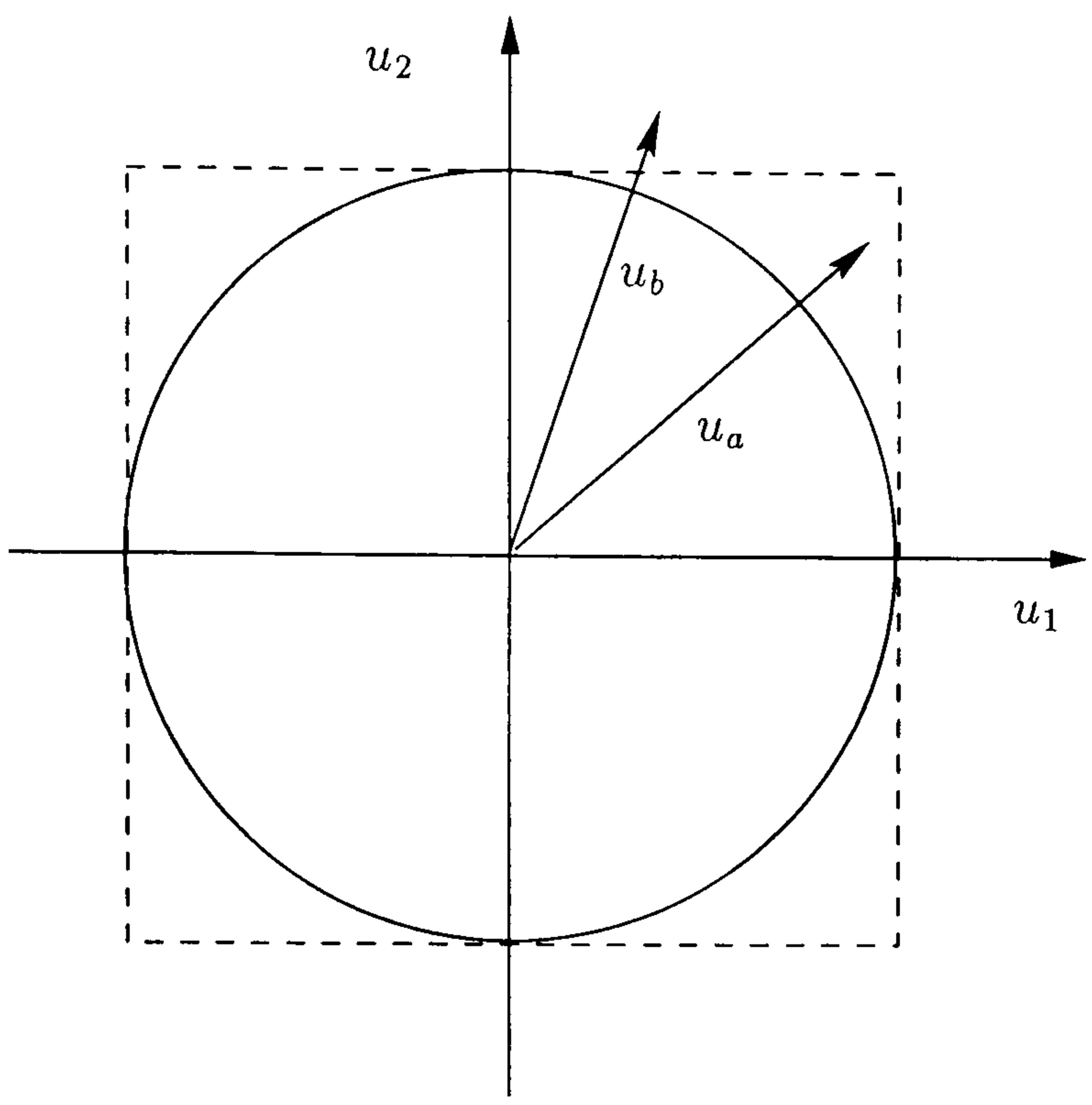

Figure 2.1: Two-norm and Infinity-norm of two control inputs.

the control input in Eq. 2.26 on the basis of the control magnitude and the disturbance coming in the best direction (this is the disturbance which produces the smallest control input) gives

$$1 \leq \gamma^* = \frac{\overline{\sigma}(G)\ ||u||_2}{||g_p||_2} \leq \gamma \tag{2.27}$$

where γ^* is the disturbance condition number and γ is the condition number defined as

$$\gamma(G) = \frac{\overline{\sigma}(G)}{\underline{\sigma}(G)} \tag{2.28}$$

A large γ^* means that the magnitude of the control input needed to reject this particular disturbance is larger than a control input needed to reject a disturbance in the best direction. A larger control input is more likely to exceed the bounds. The condition number γ provides an upper bound on this measure. This is one reason why a plant with a low condition number is desirable. But a large condition number γ does not imply that the disturbance condition number $\gamma^\star$ is necessarily large.

Other controllability measures based on linear process models have been proposed. A comprehensive overview of existing controllability tools based on linear process models can be found in [85]. Controllability analysis based on linearised process models has been shown to be a valuable tool for comparing the control performance of different process designs (e.g. [67]).

The feasibility constraint can be used to measure controllability (flexibility) of a given design with fixed design parameters $\hat{d}$. In this case, the steady state control objective is to satisfy all of the constraints $g(\cdot)$. It is assumed that the controller (operator) has perfect knowledge of all the states and is able to adjust the control inputs, u, perfectly according to the control objective. Satisfaction of the design constraints for a given control input $\hat{u}$ and parameter $\hat{p}$ can then be expressed as

$$\begin{array}{l} g(x,\hat{u},\hat{d},\hat{p}) \leq 0 \\ h(x,\hat{u},\hat{d},\hat{p}) = 0 \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \max\limits_{j} & g_j(x,\hat{u},\hat{d},\hat{p}) \leq 0 \\ \text{s.t.} & h(x,\hat{u},\hat{d},\hat{p}) = 0 \end{array} \qquad (2.29)$$

Since we assume that the controller can adjust the control inputs in an optimal manner in order to satisfy the constraints, the control variables are defined by

$$\begin{array}{ll} \min\limits_{u}\max\limits_{j} & g_j(x,u,\hat{d},\hat{p}) \leq 0 \\ \text{s.t.} & h(x,u,\hat{d},\hat{p}) = 0 \end{array} \qquad (2.30)$$

This condition should not only hold for the single parameter vector $\hat{p}$, but also for all parameter vectors contained in the set P. Mathematically, this can be expressed as

$$\left.\begin{array}{ll} \min\limits_{u,x}\max\limits_{j} & g_j(x,u,\hat{d},p) \leq 0 \\ \text{s.t.} & h(x,u,\hat{d},p) = 0 \end{array}\right\} \quad \forall p \in P \tag{2.31}$$

This condition is equivalent to the feasibility constraint. If it is fulfilled, the design $\hat{d}$ is steady state controllable.

A measure of how controllable (flexible) a plant with design parameters $\hat{d}$ is, is given by the maximum scaled uncertainty δ for which condition 2.31 holds. The set P is defined in this case by

$$P = \{p | p_{nom} - \delta(p_{nom} - \underline{p}) \leq p \leq p_{nom} + \delta(\overline{p} - p_{nom})\} \tag{2.32}$$

where p_{nom} is defined as

$$p_{nom} = \frac{\underline{p} + \overline{p}}{2} \tag{2.33}$$

The measure δ is the fractional range of the uncertainty the design $\hat{d}$ can tolerate while still being able to operate in the feasible region. Thus $\delta = 1$ indicates that the constraints can be satisfied for all uncertain parameters. A $\delta < 1$ indicates that the design $\hat{d}$ is not steady state controllable. Measures larger than unity indicate an easily controllable design since there are some safety margins. The measure δ for a given design $\hat{d}$ is found by solving

$$\begin{array}{lll} \max\limits_{\delta} & \delta & \\ \text{s.t.} & \max\limits_{p}\min\limits_{x,u}\max\limits_{j} & g_j(x,u,\hat{d},p) \leq 0 \\ & \qquad \text{s.t.} & h(x,u,\hat{d},p) = 0 \\ & p_{nom} - \delta(p_{nom} - \underline{p}) \leq p \leq p_{nom} + \delta(\overline{p} - p_{nom}) & \end{array} \tag{2.34}$$

2.1.3.1 Optimal Design of Batch Fermenters

In [87] an integrated approach is proposed to design optimal batch processes. An optimal control problem is solved to determine an optimal operating policy and the corresponding optimal state profile. Once these are established, different criteria are proposed to investigate the controllability of these profiles. The control objective is to track a subset of these optimal profiles. The criteria include invertibility of the model for all possible uncertainties. A safe operation of the process is ensured if it is possible to find a control input such that the states do not violate any safety constraints in the presence of non nominal operating conditions. It is not obvious how to modify the original problem formulation once it is found that the optimal profiles do not satisfy the safety criteria.

The invertibility criteria is equivalent to the perfect control criteria for continuous plants. Denote by $y_{sp}(t)$ the time-varying vector set point profile for the process outputs. The process is invertible if it is possible to find a control input to track these profiles under all uncertainties. For analytical solutions of invertibility the reader is referred to [41, 42]. Alternatively the inversion problem can be formulated as an optimal control problem [53]. This optimisation problem is essentially the feasibility constraint for a dynamic system. The control objectives are that the output tracks the set point perfectly $y_{sp}(t) = y$ and that the control bounds are not exceeded.

2.2 Modeling of Fermentation Systems

In this section models of fermentation systems will be presented. This review is not intended to be exhaustive. The intention is to explain the models used in this thesis.

For thorough reviews on models of fermentation systems the reader is referred to [62, 5].

2.2.1 The Tank Fermenter

In this section the unsteady state mass balances of a tank fermenter are derived. It is assumed that there is only a single species of microorganism growing in the tank and there is only one growth limiting substrate. The extension to multiple species is straightforward. Multiple substrates can be included in the model by using appropriate kinetic expressions for the growth rates.

The fermenter is shown in Fig. 2.2. The stream F_{in} flows into the fermenter.

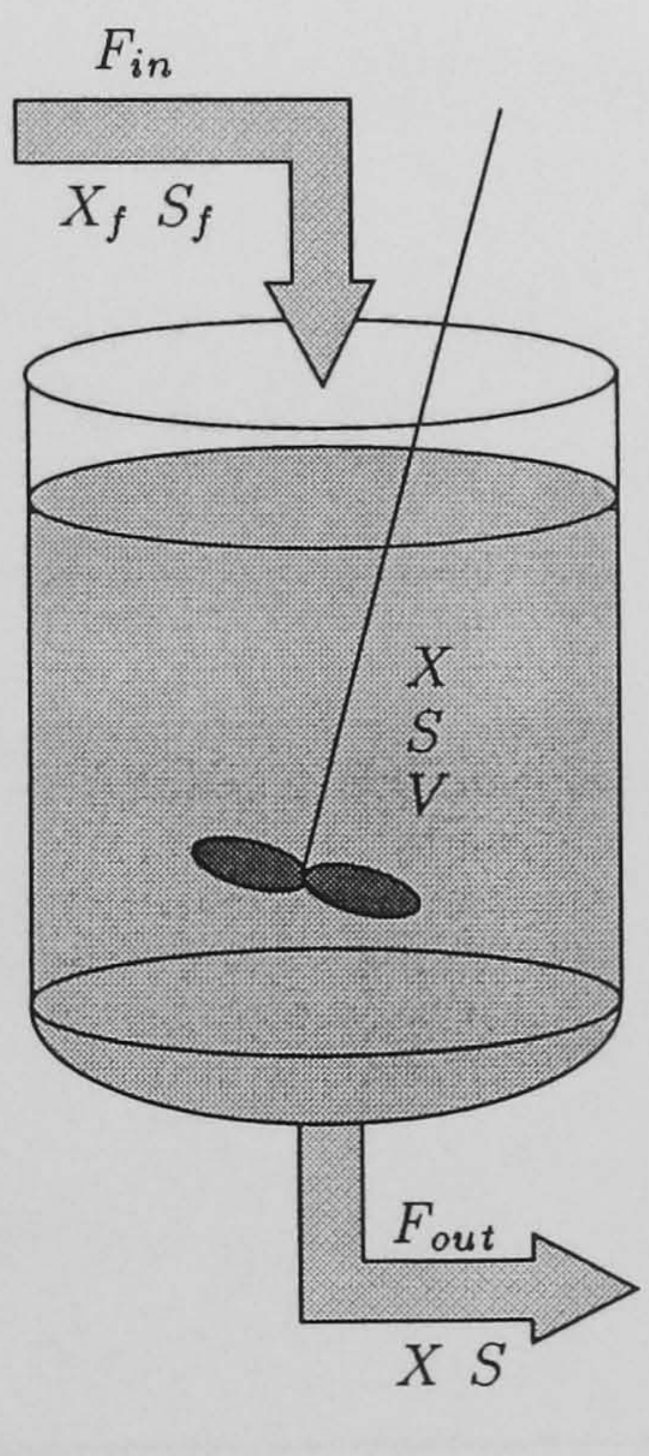

Figure 2.2: An ideal stirred tank

The cell concentration of this stream is X_f and the concentration of the growth limiting substrate in this stream is S_f. The stream leaving the fermenter is called F_{out}. It is assumed that the content is well mixed. Therefore the cell and substrate concentrations of the outlet stream are the same as in the fermenter. The model describing the dynamics of the total cell mass in the fermenter (XV), the total substrate (SV), the total product (PV) and the tank volume V is given by

$$\frac{d(XV)}{dt} = r_X V + X_f F_{in} - X F_{out} \quad (2.35)$$

$$\frac{d(SV)}{dt} = r_S V + S_f F_{in} - S F_{out} \quad (2.36)$$

$$\frac{d(PV)}{dt} = r_P V + P_f F_{in} - P F_{out} \quad (2.37)$$

$$\frac{dV}{dt} = F_{in} - F_{out} \quad (2.38)$$

and by applying the chain rule of differentiation, the model becomes

$$X\frac{dV}{dt} + \frac{dX}{dt}V = r_X V + X_f F_{in} - X F_{out} \quad (2.39)$$

$$S\frac{dV}{dt} + \frac{dS}{dt}V = r_S V + S_f F_{in} - S F_{out} \quad (2.40)$$

$$P\frac{dV}{dt} + \frac{dP}{dt}V = r_P V + P_f F_{in} - P F_{out} \quad (2.41)$$

$$\frac{dV}{dt} = F_{in} - F_{out} \quad (2.42)$$

Re-arranging these equations gives

$$\frac{dX}{dt} = r_X + \frac{X_f - X}{V} F_{in} \quad (2.43)$$

$$\frac{dS}{dt} = r_S + \frac{S_f - S}{V} F_{in} \quad (2.44)$$

$$\frac{dP}{dt} = r_P + \frac{P_f - P}{V} F_{in} \tag{2.45}$$

$$\frac{dV}{dt} = F_{in} - F_{out} \tag{2.46}$$

Reaction rate terms are derived in the following section.

2.2.2 Process Kinetics

In this section, the reaction terms in the model are presented. For the growth of cell mass in a pure batch process, the specific rate is

$$\frac{1}{X}\frac{dX}{dt} = \mu(X,S) \quad [h^{-1}] \tag{2.47}$$

and the corresponding rate expression

$$r_X = \mu(X,S)X \tag{2.48}$$

For the substrate consumption in a batch process, the specific rate is

$$\frac{1}{S}\frac{dS}{dt} = \rho(X,S) \quad [h^{-1}] \tag{2.49}$$

and the rate is

$$r_S = \rho(X,S) \tag{2.50}$$

A parameter of great importance in these models is the yield factor. This factor was originally defined by Monod (Reference in [62]) by the quotient

$$Y_{X/S} = -\frac{\Delta X}{\Delta S} = -\frac{X_t - X_0}{S_t - S_0} \approx \tag{2.51}$$

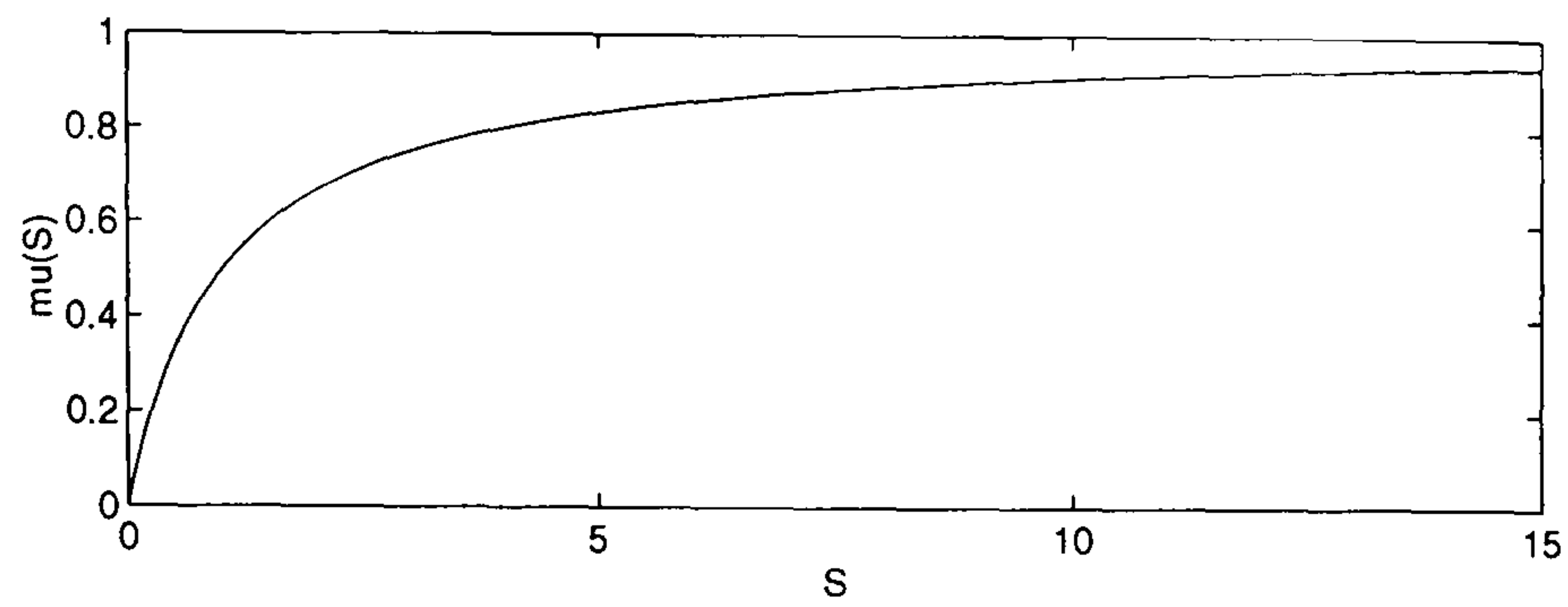

Figure 2.3: Monod kinetics

$$-\frac{dX/dt}{dS/dt} = -\frac{r_x}{r_S} \quad [\frac{\text{g cells formed}}{\text{g substrate used}}] \tag{2.52}$$

The assumption of a constant yield coefficient during fermentation is not realistic. In real fermentation the yield coefficient depends on various other parameters which either can not be modeled or their influence is often neglected. The following expression shows the influence of the growth rate on the yield factor if a maintenance term is included.

$$\frac{1}{Y_{X/S}} = \frac{1}{Y_{X/S}^{max}} + \frac{m_s}{\mu(x,S)} \tag{2.53}$$

This means that the cells are not using the substrate for growth only, a part of the substrate is utilised for maintenance.

Different expressions for the specific growth rate are reviewed in [62]. The most widely used growth rate expression is the Monod kinetics, shown in Fig. 2.3, where S is the growth limiting substrate concentration

$$\mu(S) = \mu_{max}\frac{S}{K_S + S} \tag{2.54}$$

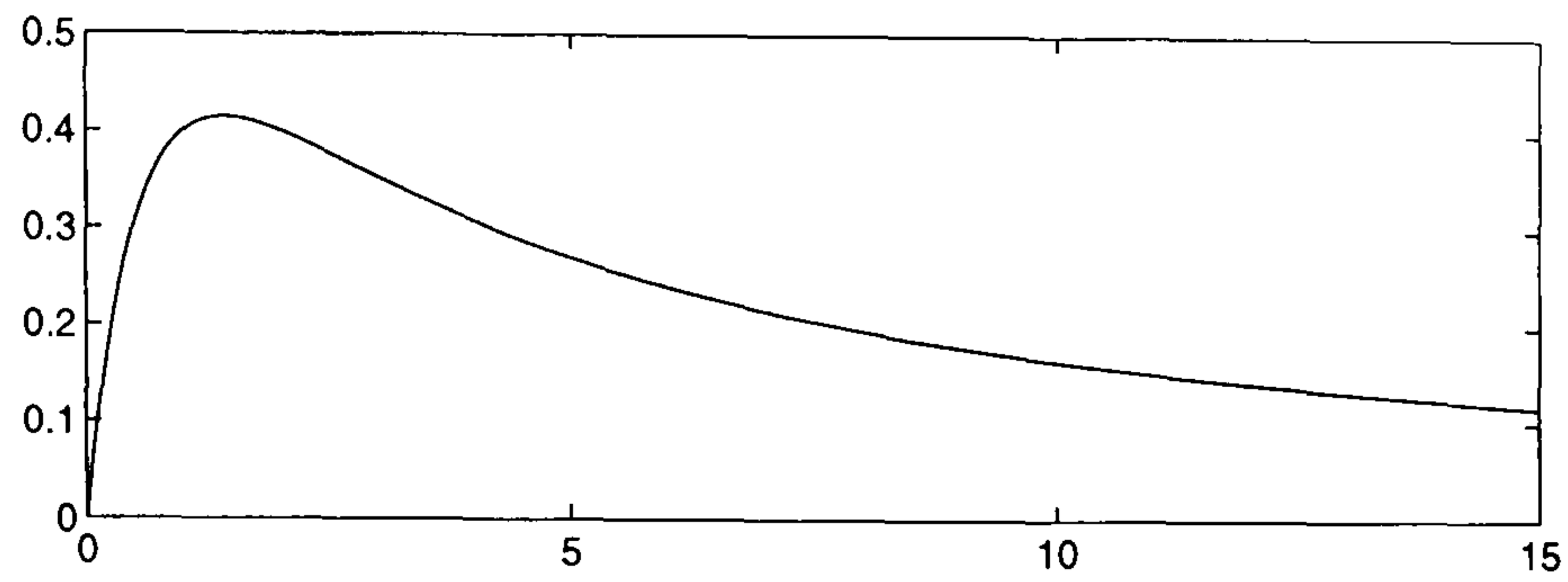

Figure 2.4: Substrate inhibited Monod kinetic

and K_S is the Monod constant. The Monod kinetics can be formally derived for enzyme kinetics (Michaelis Menten kinetics) in order to show the analogy between the varios mechanisms. The Monod kinetic can be modified to include substrate inhibition

$$\mu(S) = \mu_{max}\frac{S}{K_S + S + S^2/K_i} \tag{2.55}$$

where K_i is the inhibition constant. This is shown in Fig 2.4 Other modifications to include product inhibition are

$$\mu(S,P) = \mu_{max}\frac{S}{S + K_S}(1 - \frac{P}{K_P}) \tag{2.56}$$

where a high product concentration has a negative effect on the specific growth rate as shown in Fig 2.5.

2.3 Controllability of Fermentation Systems

The succesful operation of the downstream processes depends strongly on the performance of the fermentation in a biochemical plant. It is therefore important to

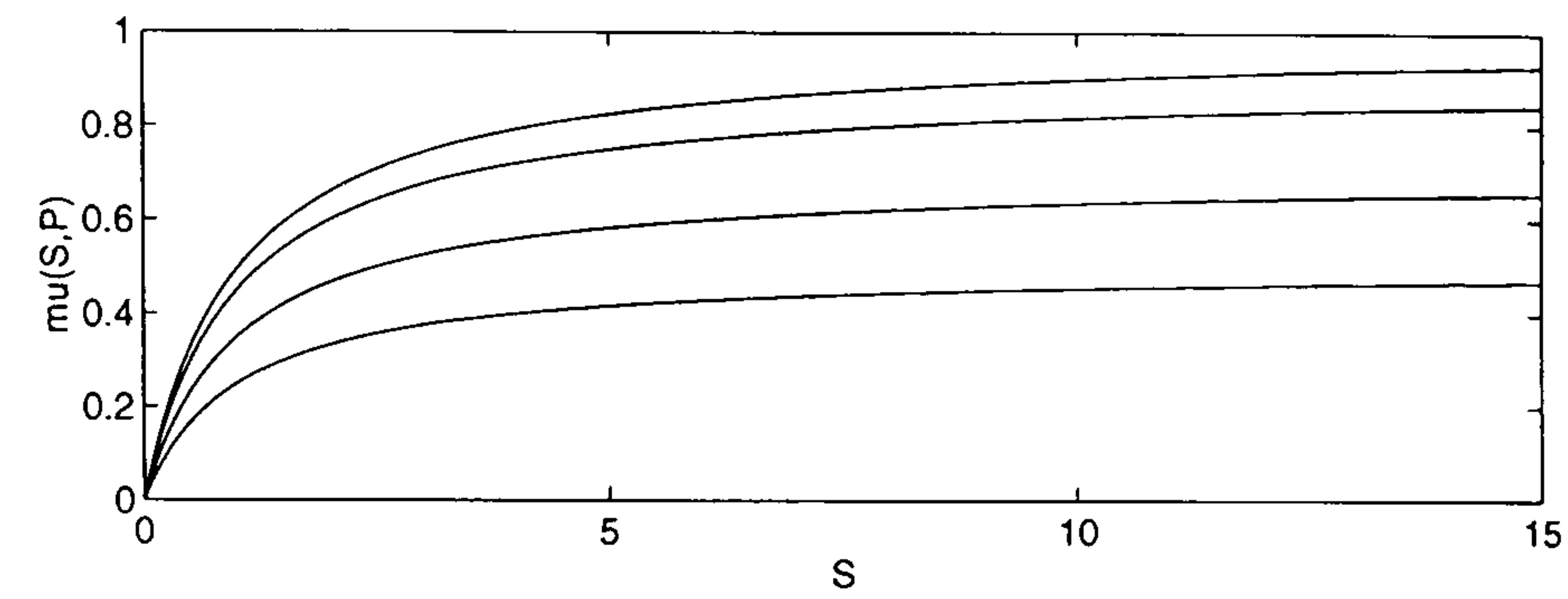

Figure 2.5: Product inhibited Monod kinetic

keep variations of the fermentation's states small. Some work evaluating the controllability of continuous fermentation processes has been published. In [1] various designs of continuous bioreactors were examined for their controllability properties based on steady state arguments. It is found that a better control performance of a bioreactor can be expected if a second feed stream, which does not contain any growth limiting substrate, is used as the manipulated control input. In [54] a comparison is made between using the dilution rate or the input substrate concentration as the manipulated variable in order to keep the cell concentration constant. The substrate concentration of the inlet flow yields better controllability properties than the dilution rate. This result is confirmed by others ([102]) using linearised transfer function models. In [10] a high concentration feed stream and a substrate recycle are added to a single continuous fermenter to design an optimal controllable continuous fermenter. In [103] different designs of a continuous bioreactor with a cross flow filtration recycle are studied.

2.4 Optimal Control of Fed Batch Fermenters

Optimal control of fed batch fermentation has been widely studied and different approaches for the solution of these problems have been proposed. In [35] the feed rate was used as a control variable and an analytic expression for the switching times between bang-bang control intervals and singular arcs was derived. For the singular arcs, a feedback law for the feed rate was obtained. General characteristics for the optimal feed rate profiles for different classes of fed batch fermentations were presented in [56], and in [47] these characteristics were used to establish a numerical procedure to obtain the optimal control profiles for the singular control problem. In [20] the control and state profiles were approximated with Lagrange polynomials and orthogonal collocation was used to transform the optimal control problem into a NLP problem. Discontinuities of the control profiles were taken into account via the introduction of a set of additional super-elements as decision variables. This formulation can deal with general constraints. In [55] a non-singular control approach was presented to solve fed batch optimisation problems. The feed substrate concentration and the temperature in the fermenter were used as control variables in [43] in order to obtain a non-singular control problem. The growth rate was assumed to be temperature dependent. To compute the optimal trajectories the Hamiltonian of the system was minimised. The system and adjoint equations were solved by orthogonal collocation and the constraints were introduced via a penalty function approach. In [78] the substrate concentration in the reactor was used as the control variable to derive a non-singular control problem. The optimal profile of the substrate concentration was then realised with a feedback controller which used the feed substrate concentration as the manipulated variable. A control vector parameterisation technique was used in [17] to solve the optimisation problem which was formulated in the context of a differential algebraic system of equations.

The problem takes into account general constraints on state and control variables. In [23] the optimal control problem was also formulated as a differential algebraic system and Kelly's transformation was used to derive a nonsingular control problem. Iterative dynamic programming was used in [50] to obtain the optimal feed rate profiles. General constraints can be included via a penalty function approach. Different optimisation methods were compared in [32] to obtain the optimal feed rate profiles. It was concluded that the problem is nonconvex with several local optima.

2.5 Conclusions

One of the most widely studied methods for controllability analysis is that based on linear process models. This method for evaluating controllability is applicable to processes which operate at steady state and for which the dynamic behavior can be adequately approximated by a linear process model around the steady state. The advantage of such a method is that it can be used on linear process models derived either from detailed nonlinear models or from linear dynamic shortcut models based on steady state models ([46, 86]). Thus at an early stage of the design process controllability tests can be introduced. Another advantage of open loop measures is that the control system does not need to be specified to determine a controllability measure. It is generally difficult to integrate these measures into a systematic design procedure [65, 51]. However, these tools are useful to screen different design alternatives to identify attractive process structures. This is demonstrated in chapter five.

Steady State controllability (or flexibility) can be taken into account by using the flexibility framework described above. The design framework aims to optimise the process performance under uncertainty according to an objective function. Control variables are used to guarantee feasible operation. The set points are added to the model and are determined by the optimisation. The control objective in this approach is to force feasible operation, i.e. to satisfy a set of inequality constraints. It is not obvious in this approach whether the process can be transfered from one steady state to another without constraint violation. This difficulty can be overcome by using dynamic models in the problem formulation. In order to incorporate the controller in the model it is not sufficient to include the set points into the problem formulation. When using dynamic models the control system should be specified, e.g PI controllers should be included into the problem formulation. This is due to the fact that although a control objective which requires no offset at all times is appropriate for operation at steady state, it is much too restrictive if the process operates dynamically. A dynamic model of the process is required and information about a suitable controller is also required. This approach is in principle applicable to batch processes if the design constraints represent the control objective.

In the multi-objective design approach a nonlinear model is combined with a controllability indicator which should be easily available. This is fulfilled in the methods proposed by [65, 51]. The measures are based on linear process models and therefore are not applicable for nonlinear processes operating in non-steady state modes.

Nonlinear dynamic models for fermentation systems are described in third part of this chapter. These models are low order models which describe the behavior of the process sufficiently well.

Different optimal control methods have been applied to unstructured fermentation models. These optimal control methods do not guarantee that the resulting operating strategy is a favorable one in the presence of uncertainties and disturbances. This shortcoming can be overcome if a control system which counteracts these uncertainties is incorporated into the optimal control method.

The next chapter presents methods to transform an optimal control problem into a parameter optimisation problem. In the third chapter the above flexibility approach is incorporated into the transformed problem formulation. In order to guarantee bounds on the outcome of the fermentation, a second worst case objective is formulated and the problem is solved in a multi-objective optimisation framework. Such an approach provides a systematic way to incorporate controllability aspects into the design procedure.

Chapter 3

Optimal Control

The design of a fed batch fermenter can be formulated as an optimal control problem. In this chapter the mathematical foundations necessary to solve optimal control problems are explained. **S**equential **Q**uadratic **P**rogramming (SQP) to solve static optimisation problems will be explained. The formulation of a general optimal control problem is introduced. Methods to reformulate optimal control problems as parameter optimisation problems are reviewed. The reformulated problem can be solved using any appropriate NLP solver. These methods usually parameterise the control variable $u(t)$ as a piecewise function of time, such as a constant function or a Lagrange polynomial. The various methods proposed in the literature differ mainly in how the **O**rdinary **D**ifferential **E**quations (ODEs) are formulated as constraints in the optimal control problem. The system of ODEs, Eq. 3.28, can be solved using an **I**nitial **V**alue **P**roblem (IVP) or **B**oundary **V**alue **P**roblem (BVP) solver in each optimisation iteration ([74, 91, 90]). Since the differential equations are solved to feasibility and sequentially in each optimisation iteration, this approach is called the *sequential feasible* approach. *Sequential feasibility* relates only to the solution of the ODEs and not to the optimisation iteration if, for example, an SQP type method is employed to solve the problem. A premature termination of the optimisation does *not* in general return a feasible iteration because there is no guarantee that all the inequalities are satisfied. There are SQP methods which produce feasible iterations (see [66]). In these methods the optimisation can be stopped before termination and a feasible point is obtained.

In the *simultaneous* approach, the ODEs are solved simultaneously with the optimisation problem by using, for example, a method of weighted residuals (e.g. orthogonal collocation) to approximate the state profiles([20, 6, 73, 88]).

Alternatively, as a hybrid between these two approaches, a multiple shooting method originally developed for solving BVPs can be employed to discretise the ODEs([69, 90, 44]). The ODEs are feasible in the shooting intervals at each iteration and the solution at the nodes converges simultaneously with the optimisation iterations.

Each of these methods has its own advantages and disadvantages and has successfully been applied to a number of process examples. Gradient calculations for the *sequential feasible* method are described in [74], [49] and [14]. These calculations can also be applied to the *hybrid* method. For the *simultaneous* approach, the gradients can be obtained directly from the expressions for the discretisation.

These methods will be used later to optimise fed-batch fermentations.

3.1 Sequential Quadratic Programming

SQP is not a single algorithm but rather a conceptual method from which numerous specific algorithms have evolved. In general the nonlinear programming (NLP) problem to be solved is

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0 \\ & g(x) \leq 0 \end{aligned} \tag{3.1}$$

The basic idea of SQP is to model the NLP problem at a given approximate solution x^k by a **Q**uadratic **P**rogramming (QP) subproblem and then to use the solution of this to construct a better approximation at x^{k+1}. An algorithm is said to be locally convergent if it converges to a solution starting from a point close to the solution. It is globally convergent if it converges to some local solution from any remote starting point. In order to achieve global convergence towards a solution, a means of monitoring progress is needed. In the unconstrained case the objective function can be used directly. In the constrained case progress can be monitored through the use of a merit function. The merit function is used to find an acceptable step towards the solution of the problem.

Most of the following analysis is carried out for equality constrained problems only

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0 \end{aligned} \tag{3.2}$$

Extensions to the inequality constrained problems are done whenever necessary. The first order optimality conditions (Karush-Kuhn-Tucker (KKT) conditions) for problem Eq. 3.2 are

$$\nabla_x \mathcal{L}(x^\star, \lambda^\star) = 0 \tag{3.3}$$

and

$$h(x^\star) = 0 \tag{3.4}$$

where the Lagrangian function $\mathcal{L}$ is defined as

$$\mathcal{L}(x, \lambda) = f(x) - h(x)^T \lambda \tag{3.5}$$

and the components of λ are the Lagrangian multipliers.

3.1.1 The Quadratic Subproblem

In an SQP algorithm, a QP sub-problem is solved at each iteration to determine a suitable search direction p. The QP sub-problem can be formulated in such a way as to define a Newton step in x and λ applied to the KKT conditions

$$\nabla_x f(x^\star) - \nabla_x h(x^\star)^T \lambda^\star = 0 \tag{3.6}$$

$$h(x^\star) = 0 \tag{3.7}$$

The set of linear equations which defines a Newton step at x^k in iteration k is then

$$\begin{pmatrix} \nabla_x^2 \mathcal{L}(x^k, \lambda^k) & -\nabla_x h(x^k)^T \\ \nabla_x h(x^k) & 0 \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} = \begin{pmatrix} -\nabla_x \mathcal{L}^k \\ -h(x^k) \end{pmatrix} \tag{3.8}$$

where $\nabla_x^2 \mathcal{L}(x^k, \lambda^k)$ is the Hessian of the Lagrangian function. The Hessian of the Lagrangian function can be approximated by a quasi Newton update B_k at the k-th iteration. Rearranging Eq. 3.8 yields the following set of linear equations which are solved to perform the Newton step

$$\begin{pmatrix} B_k & -\nabla_x h(x^k)^T \\ \nabla_x h(x^k) & 0 \end{pmatrix} \begin{pmatrix} p \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} -\nabla_x f(x^k) \\ -h(x^k) \end{pmatrix} \tag{3.9}$$

where p is defined as

$$p = x^{k+1} - x^k \tag{3.10}$$

The solution of problem 3.9 is equivalent to the solution of the following QP sub-problem

$$\begin{aligned} &\min_p \quad \tfrac{1}{2} p^T B_k p + \nabla_x f(x^k)^T p \\ &\text{s.t.} \quad \nabla h(x^k) + h(x^k) = 0 \end{aligned} \tag{3.11}$$

Solving the above sub-problem at each iteration to find the search direction is equivalent to applying Newton's method to the KKT conditions.

This QP sub-problem can be extended to cover inequality constrained optimisation problems by applying an active set strategy [24, 9] or an interior point method [37, 99].

For nonlinear constrained optimisation problems it cannot be guaranteed that the QP sub-problem has a solution. It is possible that the linearisation of the constraints at the current point results in an empty feasible region. In [7] an easily adaptable approach is described to deal with this situation. If the QP solver cannot find a feasible point, it often returns the sum of the constraint violations (infeasibilities) at a given point. If the feasibility tolerance of the QP solver (this is the tolerance associated with a constraint violation) is reset to this sum and the problem is re-solved then sub-problem must have a non-empty feasible region. In [7] it is shown that the resulting search direction is a descent direction for the merit function.

3.1.2 The Hessian Update

The Hessian of the Lagrangian function can be approximated using a quasi-Newton update. A common procedure for generating the Hessian approximation is to require that the *secant* condition

$$B_{k+1}(x^{k+1} - x^k) = \nabla\mathcal{L}(x^{k+1}, \lambda^{k+1}) - \nabla\mathcal{L}(x^k, \lambda^k) \tag{3.12}$$

is satisfied. Additionally, it is required that the actual approximation B_k is updated by

$$B_{k+1} = B_k + U_k \tag{3.13}$$

where U_k is a rank-one or rank-two update and that the update preserves symmetry (B_k symmetric $\Rightarrow$ B_{k+1} symmetric). One of the most successful rank-two updates having these features is the *Broyden-Fletcher-Goldfarb-Shanno (BFGS)* update [24].

$$B_{k+1} = B_k + \frac{yy^T}{s^Ty} - \frac{B_k ss^T B_k}{s^T B_k s} \tag{3.14}$$

where

$$s = x^{k+1} - x^k \tag{3.15}$$

and

$$y = \nabla\mathcal{L}(x^{k+1}, \lambda^{k+1}) - \nabla\mathcal{L}(x^k, \lambda^{k+1}) \tag{3.16}$$

A matrix B_{k+1} generated by Eq. 3.14 is positive definite if B_k is positive definite and the condition

$$y^T s > 0 \tag{3.17}$$

is satisfied. Positive definiteness of the Hessian approximation is ensured if Eq. 3.17 is satisfied for each update. Powell [71] suggested that it is possible to maintain the positive definiteness of the Hessian by modifying the update equation (Eq. 3.14) by replacing y in Eq. 3.16 with

$$\hat{y} = \theta y + (1 - \theta)B_k s \tag{3.18}$$

where $\theta \in (0, 1]$. With this modification it is always possible, for a suitable θ, to satisfy Eq. 3.17 although the secant condition, Eq. 3.12, is not satisfied anymore. Although no proof of local convergence has been found for the above Hessian approximation, the algorithms based on this procedure have been shown to be successful in practice ([9]).

3.1.3 The Merit Function

In order to achieve global convergence, a merit function is needed to compute an appropriate step length. Intuitively the merit function can be regarded as a function which balances progress in the objective function with progress in satisfying the constraints. The step length α is defined as

$$x^{k+1} = x^k + \alpha p \tag{3.19}$$

where $\alpha \in (0,1]$ and x_{k+1} is the next approximation of the solution. The merit function is then minimised with respect to α to compute the appropriate step length.

One of the first merit functions to be proposed was the non-differentiable penalty function, the l_1-merit function,

$$\phi(x) = f(x) + \rho|h(x)|_1 \tag{3.20}$$

where ρ is a positive constant. Global convergence can be shown with the above ϕ as a merit function ([31, 9]) and with the parameter ρ chosen such that

$$\rho = |\bar{\lambda}|_\infty + \epsilon \tag{3.21}$$

where $\bar{\lambda}$ are the Lagrangian multipliers from the corresponding QP sub-problem and $\epsilon > 0$. Unfortunately using this merit function can lead to small step sizes close to the solution [16]. To circumvent this problem Powell ([71]) proposed the following merit function instead

$$\phi(x) = f(x) + \sum_j \sigma_j |h_j(x)| \tag{3.22}$$

where the weights σ_j for the iterations $k = 1, 2, \ldots n$ are given by

$$\sigma_j^k = \min[\sigma_j^{k-1}, \frac{1}{2}(\bar{\lambda}_j^k + \sigma_j^{k-1})] \tag{3.23}$$

and their initial values by

$$\sigma_j^0 = \bar{\lambda}_j^0 \tag{3.24}$$

The weakness of this approach is that neither global nor local convergence can be proved and that it fails on certain examples ([16]). However, experience has shown that it is suitable to solve a large class of problems ([9]).

To overcome the problem of slow convergence a non-monotone line search procedure, the "watchdog" method, was proposed [16]. Here the step size is chosen to reduce an alternative merit function, e.g. the Lagrangian, every iteration and to reduce the original merit function, the l_1-merit function, every n iterations ($n \geq 2$). i

Other merit functions such as the augmented Lagrangian have been proposed (e.g. [7]) for which global convergence can be proved. However the adjustment of the penalty parameter is complicated and is not straightforward and, is therefore not pursued here any further.

Computing an appropriate step length along the search direction obtained from the QP sub-problem can be done by applying any method for unconstrained optimisation. An exact minimisation is usually not justified since it requires too many function evaluations. An approximate iterative minimisation based on a quadratic interpolation or simple backtracking ($\alpha_{i+1} = \alpha_i/2$), is sufficient until the Armijo inequality

$$\phi(\alpha_i) \leq \phi(0) + \xi\alpha\frac{d\phi}{d\alpha}(0) \tag{3.25}$$

is satisfied, where ξ is a small positive constant ($\xi \approx 0.1$).

The two merit functions described in this section can be extended to cater for inequality constrained problems by defining the functions

$$g_j(x)_+ = \max[0, g_j(x)] \tag{3.26}$$

and treating the terms as equalities in the merit function. The multipliers for these constraints result from the QP sub-problem by applying an active set strategy.

3.1.4 Termination

In order to establish if the algorithm converged to a local minimum within a certain tolerance, a termination criterion is needed. A common termination criterion which is related to KKT conditions is

$$|\nabla f(x^k)p| + \sum_{j=1}^{m_{eq}} |\bar{\lambda}_j h_j(x^k)| + \sum_{i=m_{eq}+1}^{m} |\bar{\lambda}_j g_j(x^k)| \leq \epsilon \tag{3.27}$$

where ϵ is an appropriate small positive number and m_{eq} and m denote the number of equality constraints and total number of constraints, respectively. Note that this termination criterion applies to *all* stationary points and not only to local minima. If one wants to ensure that a minimum has been located, a check involving second order information needs to be incorporated.

3.1.5 Outline of the Algorithm

The various steps of a basic SQP algorithm are outlined below:

step 1 Evaluate the objective function and gradients at the initial point. Choose an initial Hessian approximation (e.g. identity matrix).

step 2 Solve the current QP sub-problem.

step 3 Find an appropriate step length α via a line search procedure.

step 4 Check the termination criteria. If they are satisfied, terminate.

step 5 Evaluate the gradient at the new point and update the Hessian approximation.

step 6 Go to *step 2*.

3.2 The Continuous Optimal Control Problem

In the previous section a method to solve parameter optimisation problems is described. The design problem of a fed batch fermentation can be formulated as an optimal control problem. In this section we describe the general optimal control problem. This problem will be later reformulated such that it can be solved as a parameter optimisation problem. The control variables of the optimal control problems considered in this thesis are either the process input, for example a feed rate, the process duration or the process initial conditions. The process is modeled by a system of ODEs

$$\dot{x} = f(t, x, u) \tag{3.28}$$

where $x \in \mathrm{R}^n$ and $u \in \mathrm{R}^k$ represent the state and control variables, respectively. The time horizon $I = [t_0, t_f]$ may be assumed to be either fixed or determined by the optimisation. It is assumed that the function $f : I \times \mathrm{R}^n \times \mathrm{R}^k \rightarrow \mathrm{R}^n$ is such that a unique solution to Eq. 3.28 exists for any given initial conditions $x(t_0) = x_0$ and controls $u(t)$. A control strategy $u(t)$ and state trajectory $x(t)$ are called optimal if they satisfy the model ODEs and optimise the objective function

$$J = \Phi(x(t_f)) \tag{3.29}$$

which is given here in Mayer form. The initial conditions may be fixed or be determined by the optimisation. This form of the objective function may seem to be restrictive, but in fact more general forms can be transformed into it. The more general form

$$J = G(x(t_f)) + \int_{t_0}^{t_f} F(x, u)dt \tag{3.30}$$

can be transformed to that given in Eq. 3.29 by adding the equation

$$\dot{y} = F(x, u) \tag{3.31}$$

with the initial condition

$$y(t_0) = 0 \tag{3.32}$$

to the model ODEs. The objective function in Eq. 3.30 can now be written as

$$J = G(x(t_f)) + y(t_f) = \Phi(z(t_f)) \tag{3.33}$$

where $z = \begin{pmatrix} x \\ y \end{pmatrix}$. This and other transformations are shown in Table 3.1. The general **U**nconstrained **O**ptimal **C**ontrol **P**roblem (UOCP) is therefore defined as

$$\min_{x(t),u(t)} \quad \Phi(x(t_f))$$
$$\text{s.t.} \quad \dot{x} = f(t,x,u) \tag{3.34}$$

In most practical examples, constraints on the control variables are specified. The simplest constraints are of the form

$$\underline{u}(t) \leq u(t) \leq \overline{u}(t) \tag{3.35}$$

where $\underline{u}(t)$ and $\overline{u}(t)$ are the lower and upper time-varying bounds on the control variable, respectively. The solution of the optimisation problem has to satisfy these constraints. Simple time varying bounds on the state variables can also be given

$$\underline{x}(t) \leq x(t) \leq \overline{x}(t) \tag{3.36}$$

Other more complex constraints can be imposed on the solution as well. These constraints are often described by a set of inequalities which depends on the states, controls, and time

$$g(x,u,t) \leq 0 \tag{3.37}$$

This set of inequalities represents path and interior point constraints imposed on the state variables and controls. Path constraints are constraints which have to be satisfied over continuous intervals of time, this can be the whole time horizon or parts of it. Interior point constraints are constraints which have to be satisfied at certain instants of time only

$$g(x,u,t_i) \leq 0 \tag{3.38}$$

original formulation	transformed formulation
free end time min $\Phi(x(t_f))$ st $\dot{x} = f(x,u), \quad 0 \le t \le t_f$	fixed end time min $\Phi(x(1))$ st $\dot{x} = f(x,u)y, \quad 0 \le t \le 1$ $\dot{y} = 0$
non autonomous min $\Phi(x(t_f))$ st $\dot{x} = f(x,u,t), \quad 0 \le t \le t_f$	autonomous min $\Phi(x(t_f))$ st $\dot{x} = f(x,u,y), \quad 0 \le t \le t_f$ $\dot{y} = 1$ $y(0) = 0$
Bolza Problem min $\Phi(x(t_f)) + \int_0^{t_f} F(x,u,t)dt$ st $\dot{x} = f(x,u,t), \quad 0 \le t \le t_f$	Mayer Problem min $\Phi(x(t_f)) + y(t_f)$ st $\dot{x} = f(x,u,t)$ $\dot{y} = f_0(x,u,t), \quad 0 \le t \le t_f$ $y(0) = 0$
Lagrange Problem min $\int_0^{t_f} f_0(x,u,t)dt$ st $\dot{x} = f(x,u,t), \quad 0 \le t \le t_f$	Mayer Problem min $y(t_f)$ st $\dot{x} = f(x,u,t)$ $\dot{y} = f_0(x,u,t), \quad 0 \le t \le t_f$ $y(0) = 0$
Tschebyscheff Problem min $\max_{t\in[0,t_f]} f_0(x,u,t)$ st $\dot{x} = f(x,u,t), \quad 0 \le t \le t_f$	Mayer Problem min $y(t_f)$ st $\dot{x} = f(x,u,t), \quad 0 \le t \le t_f$ $\dot{y} = 0$ $f_0(x,u,y) - y \le 0$

Table 3.1: Transformation of different optimal control problems

A subclass of these interior point constraints involve boundary conditions only

$$g(x(t_0), x(t_f)) \leq 0 \tag{3.39}$$

A special case of this constraint is when all initial conditions are fixed

$$x(0) = x_0 \tag{3.40}$$

A subclass of the general path constraints are nonlinear constraints involving the control variable only

$$g(u, t) \leq 0 \tag{3.41}$$

The general constrained optimal control problem is then given by

$$\begin{array}{lrl} \min\limits_{x(t),u(t)} & \Phi(x(t_f)) & \\ \text{s.t.} & \dot{x} = f(t, x, u) & \text{model equations} \\ & \underline{u}(t) \leq u(t) \leq \overline{u}(t) & \text{simple control bounds} \\ & \underline{x}(t) \leq x(t) \leq \overline{x}(t) & \text{simple state bounds} \\ & g_1(x(t_0), x(t_{tf}) \leq 0 & \text{boundary conditions} \\ & g_2(x, u, t_i) \leq 0 & \text{interior point constraints} \\ & g_3(u, t) \leq 0 & \text{control path constraints} \\ & g_4(x, u, t) \leq 0 & \text{general path constraints} \end{array} \tag{3.42}$$

and it will be refered to as an OCP (**O**ptimal **C**ontrol **P**roblem). The solution of Problem 3.42 optimises the objective function and satisfies all the constraints of the problem. All the constraint formulations in Eq. 3.42 are special cases of the constraint formulation in Eq. 3.37. The overall general optimal control can be written in a more compact form as

$$\begin{aligned} \min_{x(t),u(t)} \quad & \Phi(x(t_f)) \\ \text{s.t.} \quad & \dot{x} = f(t,x,u) \\ & g(x,u,t) \leq 0 \end{aligned} \tag{3.43}$$

The optimality conditions of this problem are not given here since we do not attempt in this work to solve the problem exactly. Optimality conditions for general optimal control problems of this type can be found, for example, in [13].

If the general optimal control problem depends on a free vector of parameters $p \in \mathrm{R}^{n_p}$, these parameters can be treated as additional constant states. Each parameter p_i can be viewed as an additional state x_{n+i} for which a constraint is added to the problem to force this state to be constant over the whole time horizon. The initial value of this new state is determined by the optimisation. The constraint which forces p_i to be constant over the time horizon is

$$x_{n+1} = p_i \quad \forall t \in [t_0, t_f] \tag{3.44}$$

An equivalent formulation of this constraint is

$$\dot{x}_{n+i} = 0, \quad x(t_0) = p_i \tag{3.45}$$

This ODE is simply added to the model equations and the initial value of this state determines the parameter p_i.

In the above formulation it was assumed that the end time was fixed. The duration of a process is often an optimisation variable to be determined. Problems with free end time can be included in the above formulation by introducing an additional state variable and normalising the time in the ODE model. The extended system is given by

$$\begin{aligned} \frac{dx}{d\tau} &= f(\theta, x, u)(x_{n+1} - t_0) \\ \frac{dx_{n+1}}{d\tau} &= 0 \end{aligned} \tag{3.46}$$

where the original time has been replaced by

$$\theta(\tau) = t_0 + \tau(x_{n+1} - t_0), \quad \tau \in [0, 1] \tag{3.47}$$

and τ is the normalised time. This shows that a wide class of problems is covered in the general problem formulation.

3.3 Discretisation Schemes

In this section a number of tools for reformulating the continuous OCPs as NLP problems are outlined. The parameterisation of the control profile is described first, to be followed by the discretisation of ODEs using explicit and implicit Runge Kutta schemes. The implicit Runge Kutta scheme is shown to be equivalent to a collocation scheme.

3.3.1 Control Parameterisation

In order to reformulate an OCP as a NLP problem the control function $u(t)$ is defined by a finite number of parameters. For convenience it is often assumed that the continuous control variable profile $u(t)$ can be approximated by piecewise low order polynomials over time. The time horizon from t_0 to t_f is divided into m suitable sub-intervals

$$0 = t_0 \leq t_1 \leq ... \leq t_{m-1} \leq t_m = t_f \tag{3.48}$$

In each of these sub-intervals the control variable is approximated by a low order function. In the j-th interval the control function is defined as

$$u(t) = \varphi_j(q_j, t), \quad q_j \in \mathrm{R}^k, \quad t \in [t_j, t_{j+1}] \tag{3.49}$$

where q_j is a vector containing the parameters of this function in the j-th interval. In the simplest case a constant function is used for each time interval and the control parameterisation in the j-interval is then given by

$$\varphi_j(t) = q_j, \quad q_j \in \mathrm{R}^k \tag{3.50}$$

This is illustrated in Fig. 3.1. Another simple case is when a piecewise linear approximation is used

$$\varphi_j(t) = q_{j1} + \frac{t - t_j}{t_{j+1} - t_j}(q_{j2} - q_{j1}), \quad q_j = \begin{pmatrix} q_{j1} \\ q_{j2} \end{pmatrix} \in \mathrm{R}^{2k} \tag{3.51}$$

Here the control function in the j-th interval is a linear interpolation between the two end values q_{j1} and q_{j2} of the interval. Fig. 3.2 shows a piecewise linear approximation of the control variable. A more general parameterisation is given by approximating the control variable by a polynomial of degree $s - 1$ (in Langrange form) over each time interval. The control function in the j-th sub-interval is given in this case by

$$\varphi_j(t) = \sum_{i=1}^{s} q_{ji} \mathcal{L}_i^s(\tau_j), \quad q_j \in \mathrm{R}^{sk}, \quad \tau_j \in [0, 1] \tag{3.52}$$

where τ_j is the normalised time over the $j-th$ sub-interval

$$\tau_j = \frac{t - t_{j-1}}{t_j - t_{j-1}}, \quad t \in [t_{j-1}, t_j] \tag{3.53}$$

and

$$\mathcal{L}_i^s(\tau) = \prod_{i=1, l \neq i}^{m} \frac{\tau - c_l}{c_i - c_l} \tag{3.54}$$

This representation interpolates the control variable between the points

$$\varphi_j(c_i) = q_{ji}, \quad i = 1, \ldots, s-1 \tag{3.55}$$

by a polynomial of degree $s-1$. It is illustrated in Fig. 3.3 for a quadratic polynomial representation.

In general it is not necessary that all the control variables in an optimal control problem are parameterised in the same way. Each control function can be approximated by a separate parameterisation. In principle the parameterisation for a control variable could be different for each time interval. However, this would make the implementation less straight-forward.

There are two possible ways to enforce continuity of the control variables at the boundaries of the time elements. These are illustrated here for a scalar control function using a piecewise linear parameterisation. If the parameterisation of Eq. 3.51 is chosen then continuity at the interval boundaries can be enforced by adding the $m-1$ constraints

$$q_{j2} = q_{(j+1)1}, \quad j = 0, 1, \ldots, m-2 \tag{3.56}$$

to the optimisation problem.

Continuity at the boundaries can also be achieved by adding an ODE to the model equation for each control variable. The control variable is then defined on the interval j by

$$\dot{u} = \frac{q_{(j+1)1} - q_{j1}}{t_{j+1} - t_j} \tag{3.57}$$

and the initial condition is

$$\varphi(t_0) = q_{01} \tag{3.58}$$

The control profiles obtained with these two methods, shown in Fig. 3.4, are identical and the only difference is an implementation issue. They are two different representations of the same control parameterisation.

The parameters q_{ji} corresponding to the control variable parameterisation can be included in a NLP formulation. If we denote by q the vector containing all the control parameters q_{ij}

$$q = (q_{11}, q_{12}, \ldots, q_{21}, q_{22}, \ldots q_{km}) \tag{3.59}$$

then an **U**nconstrained OCP (UOCP) is approximated by the following optimisation problem

$$\begin{aligned} \min_{x(t),q} \quad & \Phi(x(t_f)) \\ \text{s.t.} \quad & \dot{x} = f(t, x, q) \end{aligned} \tag{3.60}$$

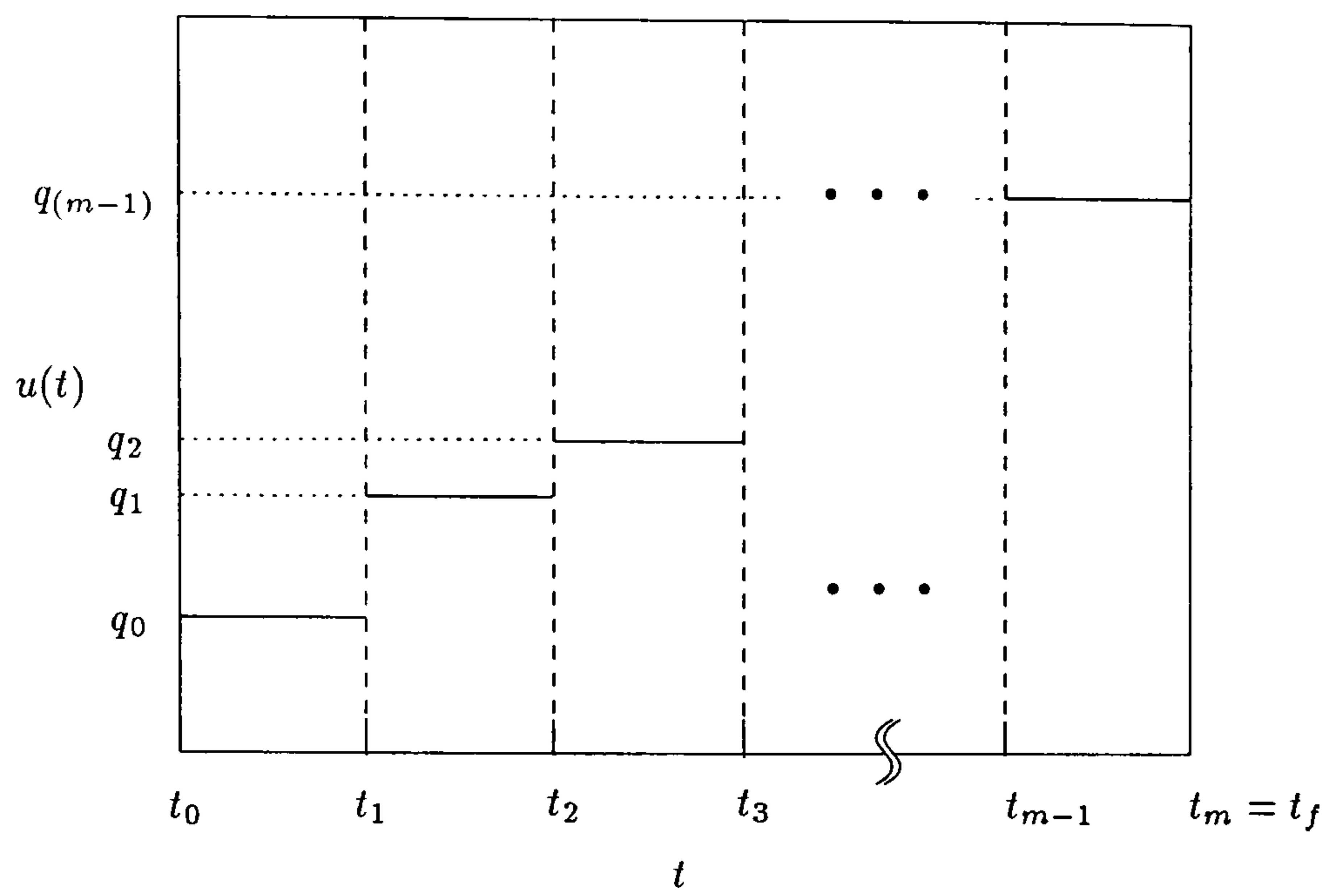

Figure 3.1: Piecewise-constant approximation of the control variable

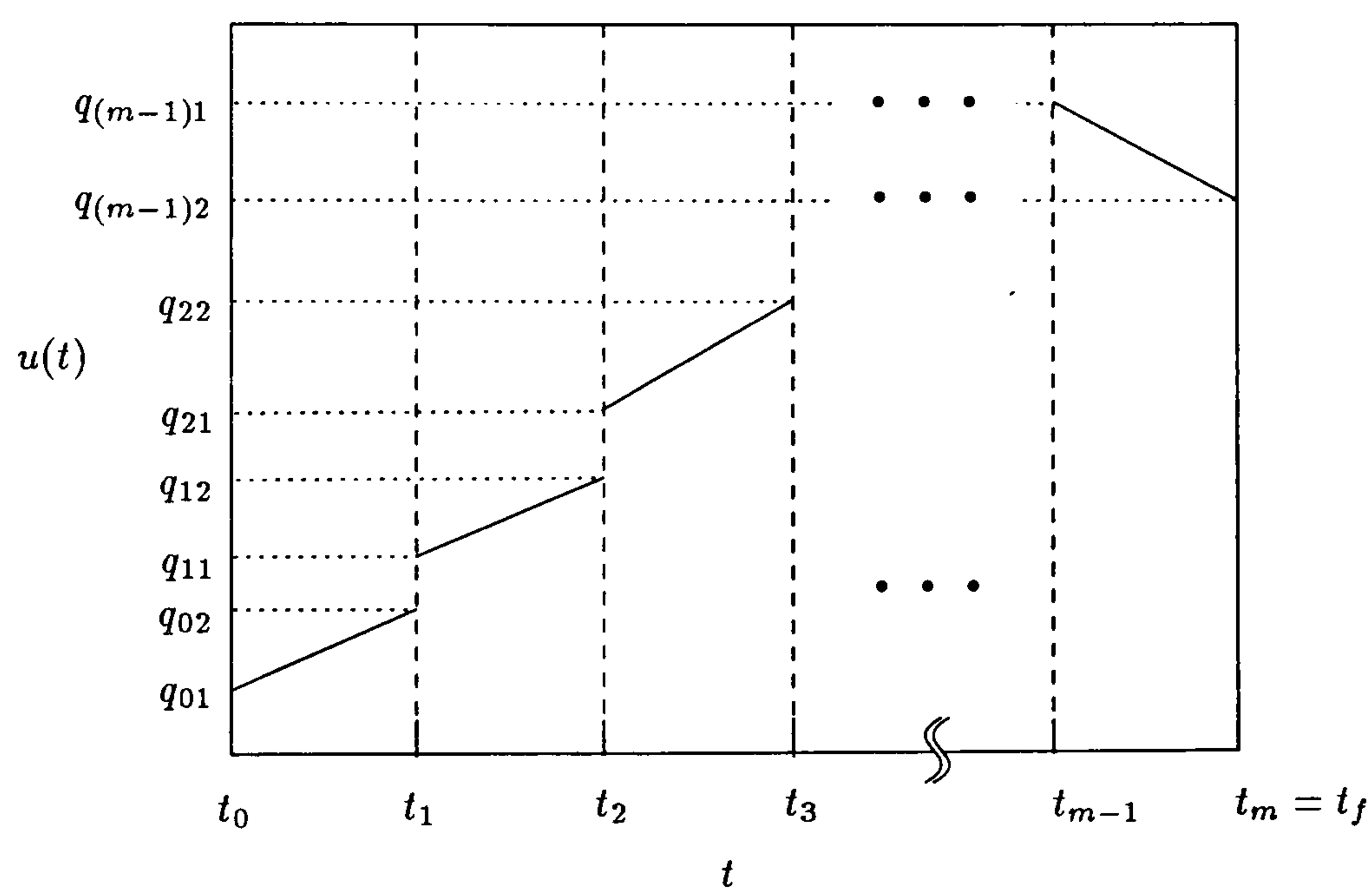

Figure 3.2: Piecewise-linear approximation of the control variable

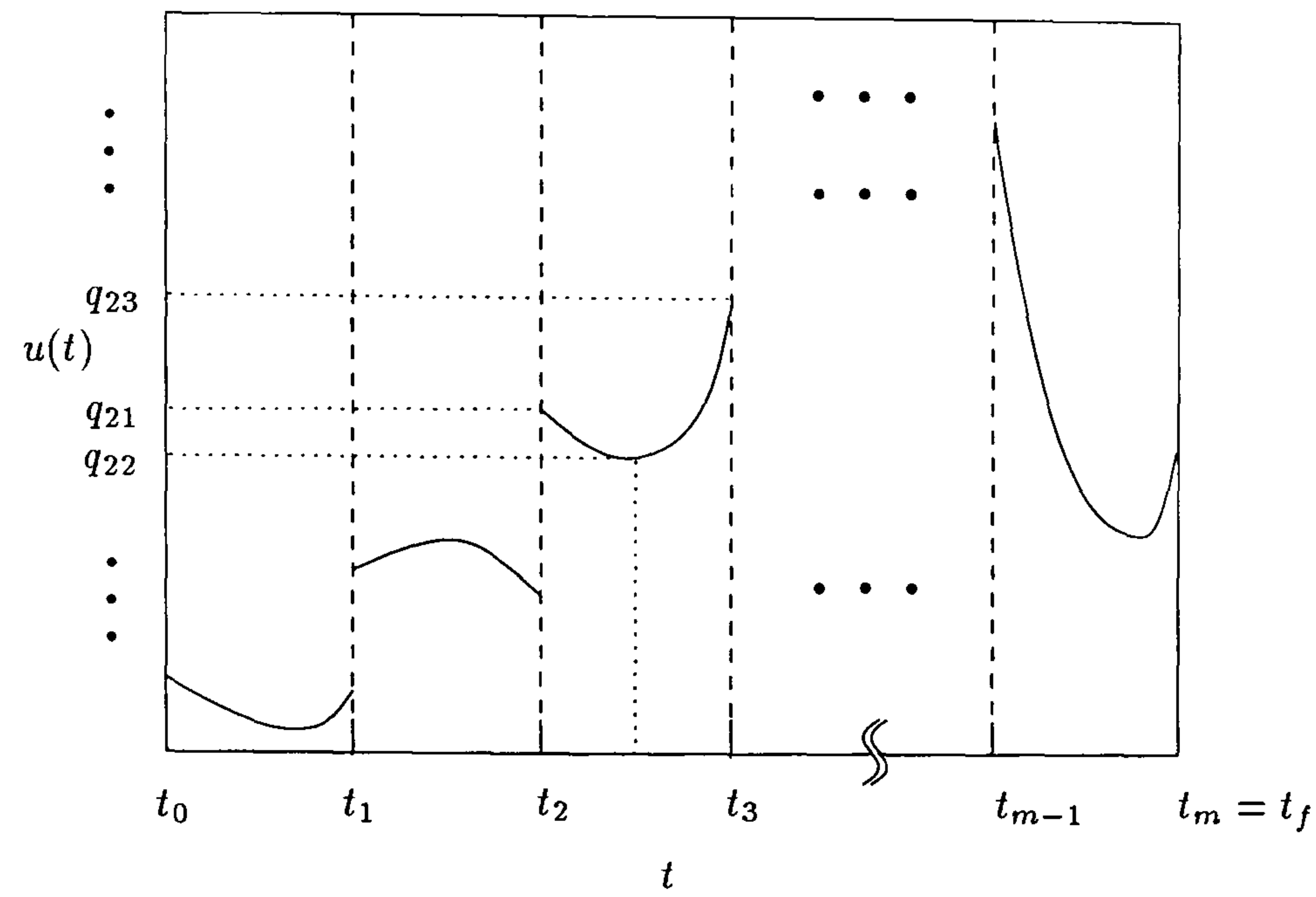

Figure 3.3: Piecewise-quadratic approximation of the control variable

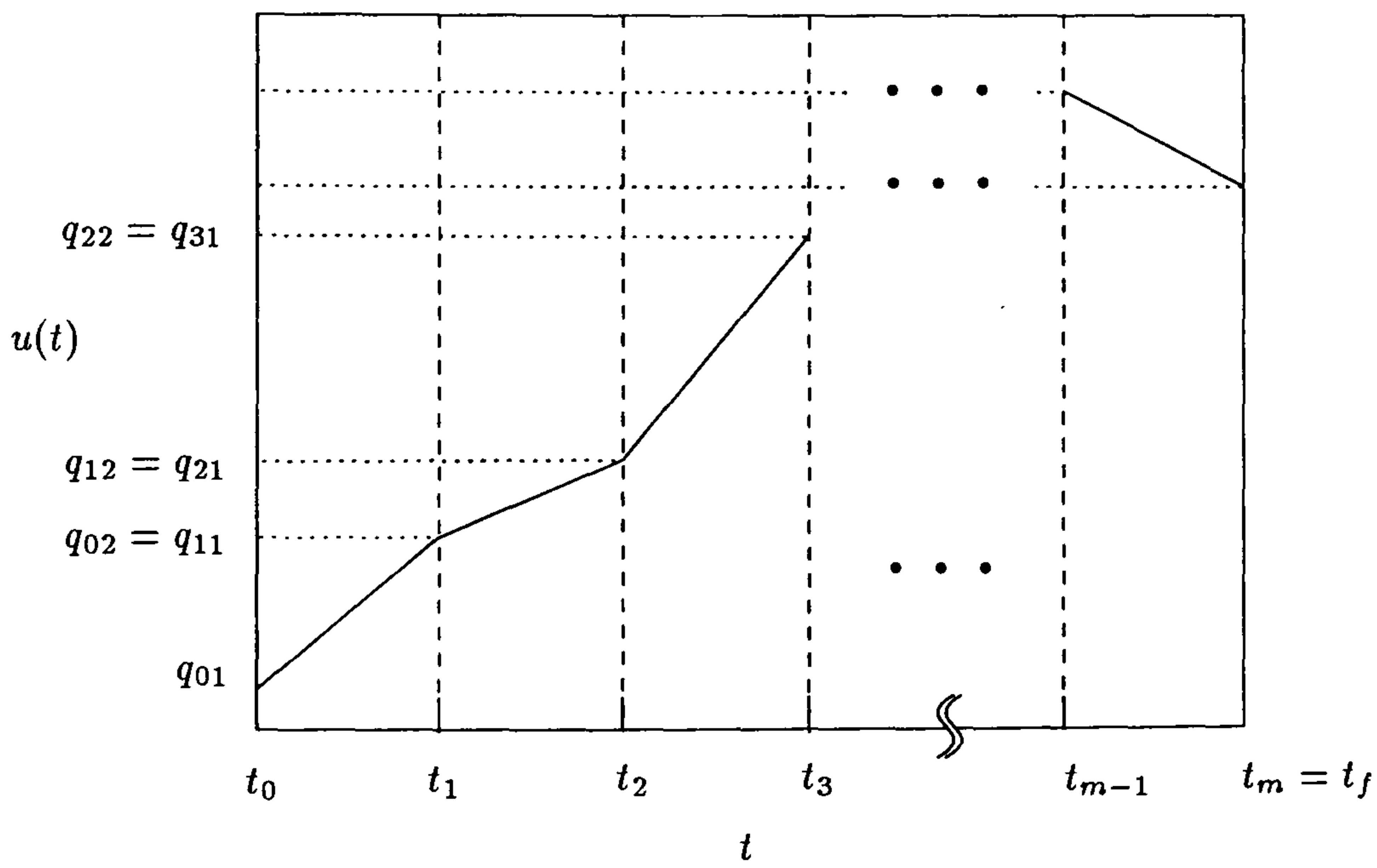

Figure 3.4: Piecewise-linear approximation of the control variable. The control variable is continuous at the interval boundaries

From an NLP point of view the optimisation of the continuous variable $u(t)$ represents an infinite dimensional problem. Using control parameterisation transforms the infinite dimensional control variable into a finite dimensional one. Although the continuous nature of the state $x(t)$ is retained, it will be shown that the UOCP can be converted into a NLP problem. The approximative solution method proposed here assumes *a priori* a certain structure of the control profile. This structure has to be chosen without knowing much about the profile of the 'true' solution. It is therefore likely that the solution is sub-optimal and it is difficult to estimate how much it deviates from the 'true' solution. Some methods have been proposed to estimate this deviation(e.g. [80]).

3.3.2 Discretisation Schemes for ODEs

The discretisation of ODEs using an **E**xplicit **R**unge **K**utta (ERK) scheme is first considered. It is shown that an **I**mplicit **R**unge **K**utta (IRK) method is equivalent to a collocation method. By this equivalence, it is possible to explain the differences between the multiple shooting and the direct discretisation approaches. We then describe methods to use this discretisation to solve IVPs and BVPs. The IRK approach is chosen here because it can be shown clearly how the same discretisation scheme can be employed for different solution procedures.

3.3.2.1 Explicit Runge Kutta Method

Here we present the set of equations which defines an ERK method of third order ([28]). For a given step size sequence the resulting equations approximate the solution of the ODE system. An ERK method of order 3 is defined by the following set of equations

$$k_1 = f(t_i, x_i, u(t_i)) \tag{3.61}$$
$$k_2 = f(t_i + h, x_i + hk_1, u(t_i + h)) \tag{3.62}$$
$$k_3 = f(t_0 + h/2, x_i + \frac{h}{4}(k_1 + k_2), u(t_i + h/2)) \tag{3.63}$$
$$x_{i+1} = x_i + \frac{h}{6}(k_1 + 4k_3 + k_2) \tag{3.64}$$

where x_i is the state at the current time t_i and h is the step size. x_{i+t} is the state at the time $t_{i+1} = t_i + h$. Appropriate error estimates and step size control mechanisms are given in [28].

3.3.2.2 Implicit Runge Kutta and Collocation Methods

Collocation methods are often used to discretise the differential equations. The resulting set of nonlinear equations is added as constraints to the NLP. Here the equivalence between a collocation method and an IRK method is shown. Based on this equivalence it is possible to outline the basic differences between the various approaches.

This section discusses some of the basic properties of collocation methods used to discretise ODEs. The time horizon $[t_0, t_f]$ is split into n intervals

$$t_0 < t_1 < \ldots < t_{n-1} < t_n = t_f \tag{3.65}$$

It is assumed that the solution of the ODEs, $x(t)$, can be approximated by a polynomial in each of these sub-intervals. It is required that each of these polynomials solves the ODE at s collocation points inside the interval. The s collocation points for the j-th interval are given by

$$t_j \leq t_j + c_1 h < \ldots < t_j + c_s h \leq t_{j+1} \tag{3.66}$$

where

$$h_j = t_{j+1} - t_j \tag{3.67}$$

The polynomial which approximates the solution on the j-th interval is defined by

$$\phi(t_j) = x(t_j) \tag{3.68}$$

$$\dot{\phi}(t_j + c_i h_j) = f(t_j + c_i h_j, \phi(t_j + c_i h_j)), \quad i = 1, 2, ..., s \tag{3.69}$$

The first equation ensures continuity between each of the intervals. The second equation ensures that the polynomial approximates the solution of the ODE at the collocation points $t_j + c_i h_j$. The solution at the end point of the j-interval is given by

$$x(t_{j+1}) = \phi(\tau_{j+1}) \tag{3.70}$$

These equations define an s-stage IRK method. $\dot{\phi}$ is a polynomial of degree $s-1$ (if ϕ is a polynomial of degree s) which interpolates s data points

$$f(t_j + c_i h_j, \phi(t_j + c_i h_j)), \quad i = 1, \ldots, s \tag{3.71}$$

in the j-interval. If k_i is defined as

$$k_i = \phi(t_j + c_i h_j), \quad i = 1, \ldots, s \tag{3.72}$$

the corresponding Lagrange interpolation polynomial for the j-th interval is given by

$$\dot{\phi}(t) = \sum_{l=1}^{s} k_l \mathcal{L}_l^s(\tau_l) \tag{3.73}$$

where τ is defined as

$$\tau_j = \frac{t - t_{j-1}}{t_j - t_{j-1}} \tag{3.74}$$

and

$$\mathcal{L}_l^s(\tau) = \prod_{i=1, i\neq l}^{s} \frac{\tau - c_i}{c_l - c_i} \tag{3.75}$$

Now if we integrate Eq. 3.73 with respect to t from t_j to $t_j + c_i h_j$, where $i = 1, \ldots, s$ and from t_j to t_{j+1}, we get

$$\phi(t_j + c_i h_j) = \phi(t_j) + h_j \sum_{l=1}^{s} \left(\int_0^{c_i} \mathcal{L}_l^s(r) dr \right) k_l, \quad i = 1, \ldots, s \tag{3.76}$$

$$\phi(t_j + h_j) = \phi(t_j) + h_j \sum_{l=1}^{s} \left(\int_0^{1} \mathcal{L}_l^s(r) dr \right) k_l \tag{3.77}$$

Since the integral expression in the sum depends on the collocation points, they can be determined once the collocation points are fixed. Setting

$$a_{il} = \int_0^{c_i} \mathcal{L}_l(r) dr \tag{3.78}$$

$$b_l = \int_0^1 \mathcal{L}_l(r)dr \tag{3.79}$$

gives the IRK scheme on the j-th interval

$$x_i = x_0 + h_j \sum_{l=1}^{s} a_{il} f(t_j + c_l h_j, x_l), \quad i = 1, \ldots, s \tag{3.80}$$

$$x_{s+1} = x_0 + h_j \sum_{l=1}^{s} b_j f(t_j + c_l h_j, x_l) \tag{3.81}$$

The solution of the nonlinear equations for all sub-intervals approximates the solution of the ODEs. A special error estimate for the IVP case is dealt with in the implementation section. The collocation points c_i and the parameters a_{ij} and b_j are conveniently expressed in an array format

$$\begin{array}{c|ccc} c1 & a_{11} & \ldots & a_{1s} \\ \vdots & & \ddots & \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array} \tag{3.82}$$

3.4 Discretisation of Optimal Control Problems

Here the reformulation of the general optimal control problem i as a NLP is discussed. In general the reformulations discussed here do not produce a solution which is exactly that of the original continuous problem. The solution of the reformulated problem can be viewed as an approximation on the solution of the original problem (it can never be better than the original solution, as long as the original constraints are satisfied). This is due to the fact that the control parameterisation confines $u(t)$

to a certain domain. From an engineering point of view the reformulated problem can sometimes be more relevant since for example the control profile of the continuous solution cannot be implemented in practice but has to be approximated by a staircase function. In such a case the reformulated problem represents reality better than the original solution which must be subsequently approximated in order to be implementable.

Different methods to reformulate path constraints in optimal control problems are also discussed. These reformulations can then be included in the NLP formulation of the optimal control problem.

In this section discretisation schemes of ODEs based on solution methods for IVPs and BVPs are described. These schemes are used to get a finite dimensional approximation of the state variables, x, which are then included in the NLP formulation.

It is worth noting again that a SQP type method can be viewed as applying Newton's method to the KKT conditions which include the equality constraints. In fact if no degree of freedom is left then a SQP reduces to Newton's method applied to the set of equality constraints[8]. We are aiming for a discretisation scheme where by Newton's method can be applied to a set of equations resulting from the discretisation. This set of equations can then be integrated into the NLP formulation of the optimal control problem to replace the ODEs.

The simplest discretisation (from an implementation point of view) is the so called single shooting approach. Here an IVP or BVP solver is used to solve the ODEs. A more complex approach is to apply a BVP technique to discretise the ODEs. An alternative approach to is to replace the ODEs by a set of nonlinear equations resulting from a discretisation scheme such as a collocation scheme.

3.4.1 Single Shooting Discretisation

In this feasible path approach, the ODEs are solved using any method suitable for solving IVPs. Since IVPs problems are a subclass of BVPs, a BVP solver can be employed as well ([90]). Given the initial conditions and the piecewise control approximation, $\varphi_j(t)$, the state variables $x(t)$ are determined uniquely. This approach minimises the objective function with respect to the optimisation variables $\{x_0, q\}$ and solves the system of ODEs for each objective function evaluation. The end values of the states appear in the NLP as functions of the optimisation variables

$$x_{tf} = x_{tf}(x_0, q) \tag{3.83}$$

These are then inserted into the objective function of the UOCP to obtain the SS-UOCP (**S**ingle **S**hooting **U**nconstrained **O**ptimal **C**ontrol **P**roblem)

$$\min_{x_0, q} \quad \Phi(x_0, q)) \tag{3.84}$$

which is an unconstrained optimisation problem. This approach is illustrated in Figure 3.5

3.4.2 Multiple Shooting Discretisation

In the multiple shooting discretisation the time horizon is divided into l suitable sub-intervals

$$0 = t_0 \leq t_1 \leq ... \leq t_{l-1} \leq t_l = t_f \tag{3.85}$$

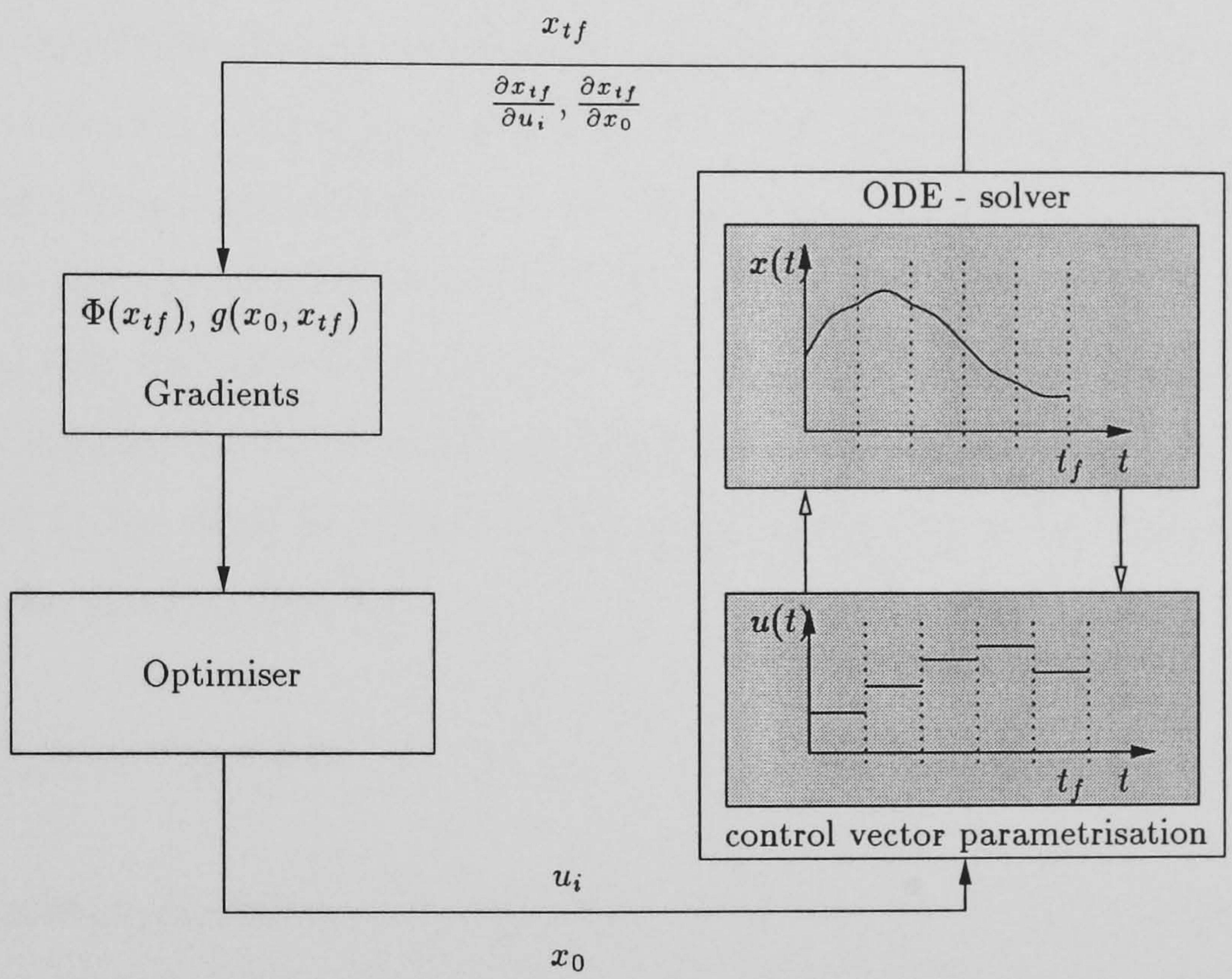

Figure 3.5: The Single Shooting Discretisation

For clarity of representation it is assumed that the sub-intervals for the control vector parameterisation are the same as those of the multiple shooting discretisation. In general these two interval grids do not need to be related but for implementation purposes it is reasonable to assume that the multiple shooting grid contains that of the control vector parameterisation. Deviations from the representation given here are straightforward. The ODEs are solved in each of the sub-intervals as in the single shooting case. The continuity between the sub-intervals is enforced through nonlinear equations which are added as constraints to the optimisation problem. The solution at the end point of the j-th interval $x^j(t_j)$ is implicitly defined through the ODEs if the initial values for this interval $x^j(t_{j-1})$ and the control parameters q_j are given. Any appropriate IVP solver can be employed to determine $x^j(t_j), j = 1, \dots, l$. In general this end value is not identical with the initial value for the next interval (which is an optimisation variable in this case) and therefore the following conditions are added to the NLP in order to enforce the continuity of the state equations between the shooting intervals

$$x^j(t_j) - x^{j+1}(t_j) = 0, \quad j = 1, \dots, l \tag{3.86}$$

The last of these conditions involves $x^{l+1}(t_l)$, which does not relate to an initial value for the next interval. This value can be used to evaluate the objective function

$$\Phi(x(t_f)) = \Phi(x^{l+1}(t_l)) \tag{3.87}$$

For simplicity of notation if the argument of the state variable is omitted this means that it relates to its initial value at this interval, e.g. x^j relates to the initial value of the state at the j-th interval $x^j(t_{j-1})$. A vector containing all the elements

$$\{x^j\}, \quad j = 1, \dots, l+1 \tag{3.88}$$

is denoted by x. It will be stated (if not clear from the context) if this x is the explicit continuous solution of the state equations $x(t)$. The optimisation variables in this case which are included into the NLP are the control parameters, the initial values of the states at each sub-interval and the final state value. These are given by the set

$$\{q, x\} \tag{3.89}$$

The reformulated problem which approximates the UOCP is MS-UOCP (**M**ultiple **S**hooting - **U**nconstrained **O**ptimal **C**ontrol **P**roblem)

$$\begin{aligned} \min_{x,q} \quad & \Phi(x(t_{l+1})) \\ \text{s.t.} \quad & x^j(t_j) - x^{j+1} = 0, \quad j = 1, \ldots, l \end{aligned} \tag{3.90}$$

The IVPs for each sub-interval are independent and can in principle be solved in parallel. The approach is illustrated in Figure 3.6.

3.4.3 Direct Discretisation

In the direct discretisation approach the state equations are discretised directly and the discretisation equations are included in the NLP probblem as nonlinear equality constraints. This approach can be explained using the single shooting method. In the single shooting method, an IVP solver is employed to solve the state equations. This IVP solver requires a number of integration steps where the length of the step is adapted in each function call in order to achieve a solution of the state equations within a certain tolerance. For each step a number of explicit or implicit discretisation equations, depending on the method chosen to solve the ODEs, have

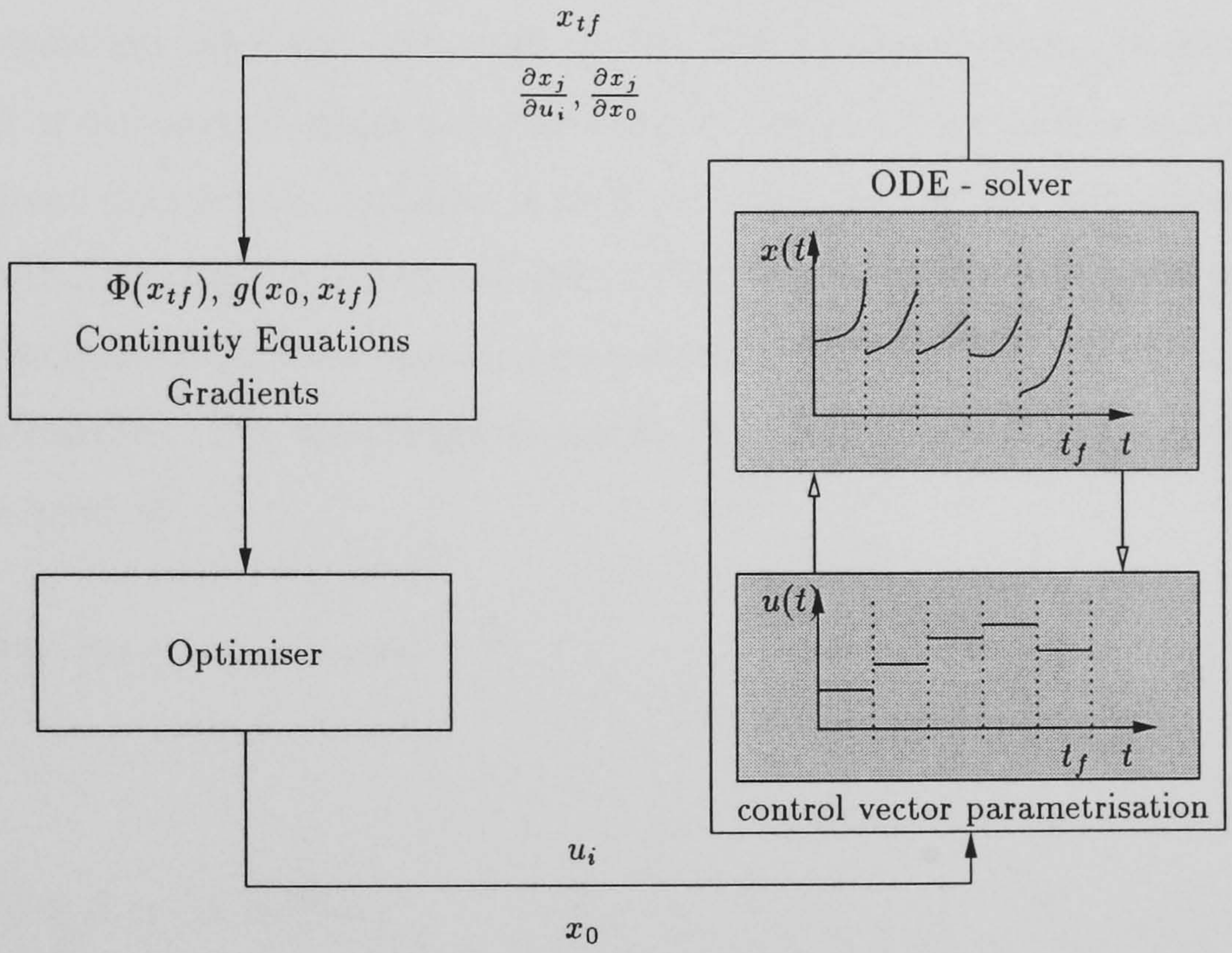

Figure 3.6: The Multiple-Shooting Discretisation

to be solved. If the step size sequence for the IVP solver is fixed in advance, this results in a fixed number of equations which have to be solved for each function call. These equations can be included as implicit equations directly as nonlinear constraints in the NLP and solved simultaneously with the optimisation. Here, as in the previous approach, the time horizon is split into l suitable sub-intervals as given in Eq. 3.85. In each of these intervals one integration step is applied. For notational convenience, it is again assumed that the grid from the control vector parameterisation coincides with that for the ODE discretisation. In general the time grid of the control vector parameterisation must be contained in the time grid of the direct discretisation scheme since it is an assumption of most discretisation methods that the right hand side of the ODEs must be continuously differentiable. This holds if an integration step is bigger than an interval from the control vector parameterisation. The discretisation results in a set of n nonlinear equations for each sub-interval

$$h_j(x_j, x_{j+c_1}, \ldots, x_{j+c_n}, q_j) = 0 \qquad (3.91)$$

where

$$0 \leq c_1 \leq \ldots \leq c_n = 1 \qquad (3.92)$$

Depending on the discretisation approach (and if $n > 1$) the variables

$$x_{j+c_2}, \ldots, x_{j+c_{n-1}} \qquad (3.93)$$

are not necessarily solutions of the state equations at the corresponding times in the interval. These equations are included as equality constraints in the NLP. A vector containing all the elements

$$\{x_j, x_{j+c_1}, \ldots, x_{j+c_n}\}, \quad j = 0, \ldots, l \qquad (3.94)$$

is denoted by x. The optimisation variables here are

$$\{x, q\} \tag{3.95}$$

and the UOCP is approximated by

$$\begin{aligned} \min_{x,q} \quad & \Phi(x(t_{l+1})) \\ \text{s.t.} \quad & h_j(x_j, x_{j+c_1}, \dots, x_{j+c_n}, q_j) = 0, \quad k = 0, \dots, l \end{aligned} \tag{3.96}$$

which will be called DD-UOCP (**D**irect **D**iscretisation - **U**nconstrained **O**ptimal **C**ontrol **P**roblem).

In this formulation, the step size sequence defined by the time intervals is fixed. This sequence may not give a solution with a specified accuracy even if this is possible with the same number of steps. For this reason the length of each interval is considered as an additional optimisation variable. A set of inequality constraints is added which bounds the error in each integration step. This approach can lead to an empty feasible region of the optimisation problem since it may not be possible to satisfy the same accuracy requirement for all time intervals. Alternatively, a set of equality constraints can be added which forces the integration error to be equal for all time steps. This can lead to very nonlinear and ill-conditioned NLP problems [81]. Another approach is to solve the problem for a given step sequence, in which a new problem with a refined grid is solved. Interpolation of the results from the previous problem can serve here as initial guesses. A series of problems can then be solved where the grid is refined between each problem.

3.5 Inequality Constraints

This section discusses different reformulations of the path constraints in OCP.

3.5.1 Interior Point Constraints

Interior point constraints can be included directly into the NLP formulation once the control profile and the ODEs are discretised. The value of the constraint is then evaluated at each function call using an integration routine or (in the case of the direct discretisation method) using the optimisation variables. For an efficient implementation it is useful if the discretised state values, say in the multiple shooting discretisation, can be used directly and the ODEs are not solved for each interior point constraint separately. In the single shooting approach this can be achieved by forcing the integration routine to output state values at all the times required for interior point constraint evaluations and not only at the final times of the integration interval. In the direct discretisation method, the discretised points can contain the interior ones, and so the constraints can be evaluated directly.

3.5.2 Path constraints

The general OCP formulation contains a set of path constraints. Path constraints can be active over intervals of time. They cannot be treated directly in a direct solution method and need to be transformed into a more suitable form. Various forms of transformations are discussed and the method used in this thesis is described. A path constraint of the form

$$g(x,u,t) \leq 0, \quad \forall t \in [t_0, t_f] \tag{3.97}$$

is known as a semi-infinite constraint. It is imposed on the solution over the continuous time interval from t_0 to t_f. From an NLP point of view, this formulation corresponds to an infinite number of constraints since it has to be satisfied continuously. Semi-infinite constraints are treated in another chapter in this thesis and in principle the methods discussed there can be applied here as well. There are basically three methods available to treat path constraints in an NLP framework

- via penalty terms in the objective function
- by converting them into end point constraints
- by replacing them by a finite number of inequality constraints

Each of these methods will be described below.

3.5.2.1 Penalty Methods

In this approach the objective function is augmented with a penalty term

$$\hat{\Phi} = \Phi + P \int_{t_0}^{t_f} \max(0, g(x, u, t))^2 dt \tag{3.98}$$

where P is a large positive number. If the constraint is to be satisfied exactly, $P \to \infty$ and therefore only an approximate satisfaction of the constraint as achieved. This approach may lead to numerical difficulties ([24]) and slow convergence ([52]), as often observed in penalty functions methods.

3.5.2.2 End Point Constraint Transformation

In this approach the path constraint is transformed to a single or multiple end point constraint, instead of being augmented to the objective function as in the penalty approach. The path constraint is transformed into a new set of constraints

$$y(0) = 0 \tag{3.99}$$

$$\dot{y} = max(0, g(x,u,t))^p \tag{3.100}$$

$$y(t_f) \leq 0 \tag{3.101}$$

where p is a positive integer. This transformation integrates the constraint violation up and replaces it with a single end constraint which can then be included into the NLP formulation. Setting p to 1 is sufficient but leads to non-differentiability at zero. First order differentiability is achieved for the exponent $p = 2$. A difficulty with this formulation is that the gradient of the active and inactive constraints is always zero at the solution. A simple way to overcome this difficulty is to relax the constraint in Eq. 3.101 to

$$y(t_f) \leq \varepsilon \tag{3.102}$$

where ε is a small positive scalar. It is worth pointing out that this reformulation is indeed a relaxation of the original constraint and the original constraint is not guaranteed to be satisfied exactly at the solution anymore. This is often acceptable in practical engineering problems. Another way to reformulate the constraint for pure state constraints of the form

$$g(x,t) \leq 0 \tag{3.103}$$

is described in [91]. In this smooth reformulation, the gradient of the constraint which is active at the solution is not zero. The path constraint is substituted by a new set of constraints

$$y(0) = \gamma \tag{3.104}$$

$$\dot{y} = \hat{g}(x,t) \tag{3.105}$$

$$y(t_f) \leq \gamma \tag{3.106}$$

where $\hat{g}(x,t)$ is given by

$$g = \begin{cases} g(x,t) & \text{if} \quad g(x,t) > \varepsilon \\ (-g(x,t) - \varepsilon)^2/4\varepsilon & \text{if} \quad -\varepsilon \leq g(x,t)\varepsilon \\ 0 & \text{if} \quad g(x,t) < \varepsilon \end{cases} \tag{3.107}$$

Here the constraint is satisfied exactly and the gradient of the active constraints at the solution is not zero. However the gradient of the inactive constraints at the solution is still zero.

3.5.2.3 Discretisation Method

In the discretisation method the path constraints are discretised so that they have to be satisfied only at certain times. Practically, this means that the path constraints are transformed into a finite number of interior point constraints which can readily be included in the optimisation problem as shown for interior point constraints.

3.5.2.4 Control Constraints

Inequality path constraints involving the control variable are a special case of the general path constraint. Control inequality constraints of the form

$$g(u,t) \leq 0 \tag{3.108}$$

which include the simple bound constraints can be directly included into the reformulated optimisation problem using the discretisation method. If the control variable is parameterised using Langrange polynomials the parameters represent the value of the control function $u(t)$ at the discrete time points. These parameters can be used to calculate the value of the reformulated constraints. In general it does not follow that the original constraints are satisfied if the transformed ones are satisfied. This is possible for some special cases only. Often the control constraints are simple bounds which are not time-varying. If these constraints are imposed on the control profile at the beginning and at the end of each control interval, and the control is parameterised by either a stair-case function or a linear function, then a satisfaction of the original constraints is guaranteed.

3.6 Gradient Evaluation

Any SQP type method for the solution of NLP problems requires the calculation of the objective function gradient and the constraint Jacobians in each iteration. Since some of the constrained functions of the OCP are defined implicitly by ODEs, an efficient method is needed to compute their derivatives with respect to the optimisation variables. The ODEs considered here are defined by

$$\dot{x} = f(x,q) \tag{3.109}$$

where the components of the vector x are the states of the system and q is a vector of model parameters. We need to compute derivatives with respect to the initial conditions, x_0, and the parameters, q_i. There are two main methods to compute these derivatives. They are explained below.

3.6.1 Numerical Differentiation

The simplest approach to compute the derivatives is to perturb one parameter at a time and then solve the ODEs repeatedly. The gradients are then obtained by finite differences ([74]). This scheme is called **E**xternal **N**umerical **D**ifferentiation (END) ([44]). It has some shortcomings. The output of an adaptive IVP solver is not differentiable with respect to the parameters and initial conditions ([28]). This is due to the step size control mechanism. If one of the inputs is varied, jumps of the order of the integration tolerance are expected in the output. They arise when a change in the input forces the integration routine to follow a different solution path. It is clear that using an integration routine as a black-box routine may produce very poor results except when high accuracies are used. The demand for high accuracy naturally results in long computing times. As a rule of thumb ([14, 44]), the tolerance TOL of the integration routine should be ε^2 where ε is the perturbation size.

This is illustrated on an example. The startup of a continuous fermenter is simulated using a third order explicit Runge-Kutta integration. The fermenter is described in Chapter 5. The sensitivity of the final value of the cell mass with respect to the initial substrate concentration is computed as described above for various initial substrate concentrations. The perturbation size was chosen equal to the integration tolerance. The result is displayed in Fig. 3.7, which shows the non-smoothness of the gradient. Since the simulation time is long enough to ensure that the unique

PAGINATION AS IN ORIGINAL

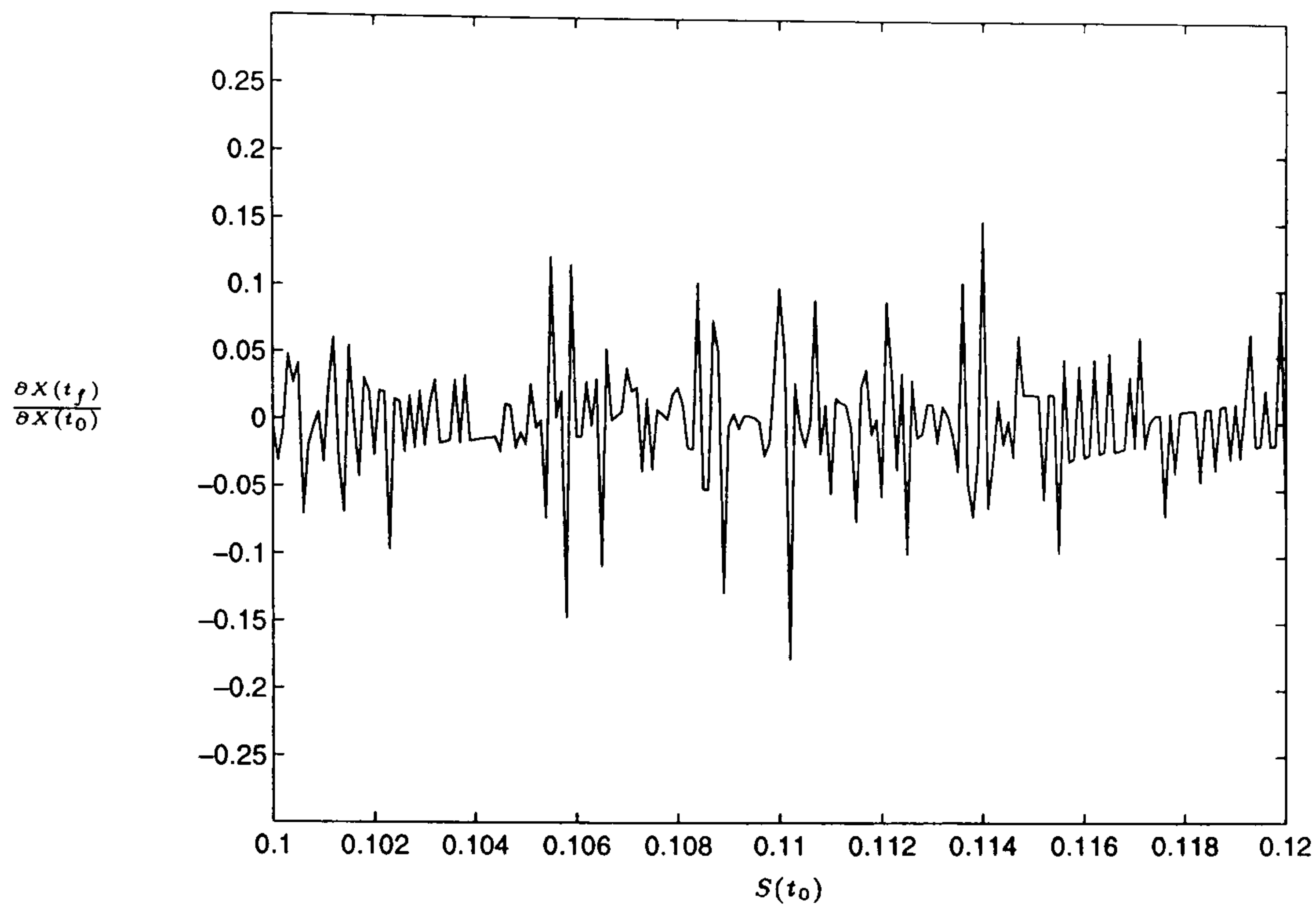

Figure 3.7: Derivative of the final cell mass with respect to the initial cell mass computed by using finite differences and an adaptive step size selection for each solution.

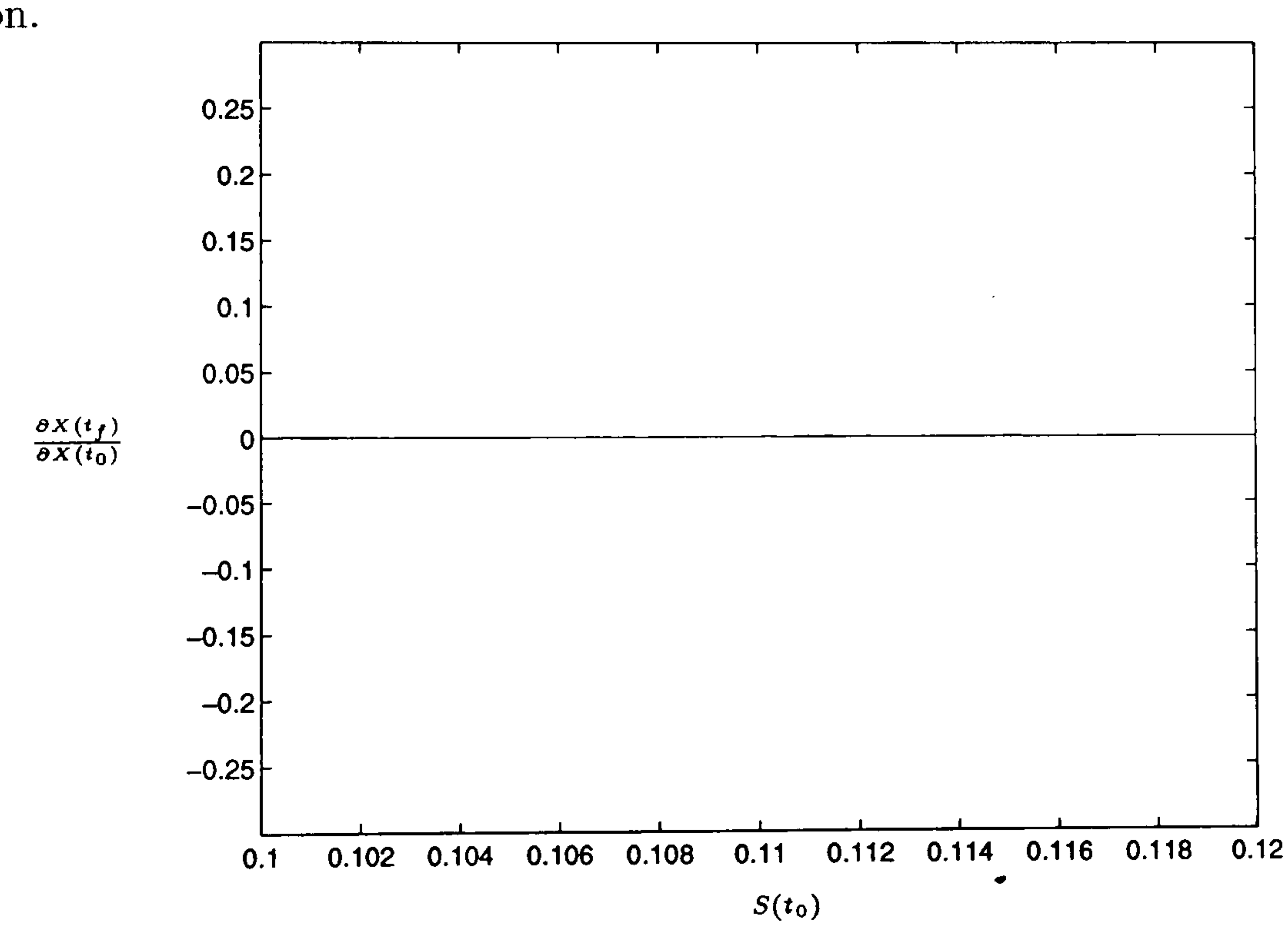

Figure 3.8: Derivative of the final cell mass with respect to the initial cell mass computed by using finite differences and the same step size sequence for all solutions.

extended state vector is defined and the new problem is given by

$$\begin{pmatrix} \dot{x} \\ z_1 \\ \vdots \\ z_n \\ z_{n+1} \\ \vdots \\ z_{n+n_p} \end{pmatrix} = \begin{matrix} f(t,x,q_i) \\ f(t,z_1,q_i) \\ \vdots \\ f(t,z_n,q_i) \\ f(t,z_{n+1},q_i+\varepsilon_{n+1}e_{n+1}) \\ \vdots \\ f(t,z,q_i+\varepsilon_{n+n_p}e_{n+n_p}) \end{matrix} \quad , \quad \begin{matrix} x(0)=x_0 \\ z_1(0)=x_0+\varepsilon e_1 \\ \vdots \\ z_n(0)=x_0+\varepsilon e_n \\ z_{n+1}(0)=x_0 \\ \vdots \\ z_{n+n_p}(0)=x_0 \end{matrix} \tag{3.110}$$

where n_p is the number of parameters and e_i is the cartesian basis vector. This problem can be solved by any appropriate IVP solver. If an implicit IVP solver is employed, special implementations can be tailored towards exploring the problem characteristics. The Jacobian of the original problem can be used as an approximation to that of the perturbed problem and the intermediate solutions of the original problem can serve as initial guesses for the perturbed problem. Using this strategy, the perturbation size ε can be chosen as TOL and the derivative is then obtained approximately with the integration tolerance TOL ([14]).

For some problems it may happen that the original and the perturbed solution differ significantly and the perturbed solution computed by the finite difference approximation is far from the original solution. In such a situation the computation should be stopped. This situation could be indicated for example by a large norm of the difference of the two solutions. The derivatives at the current time, t_j,

$$\frac{\partial x(t_j)}{\partial x_0} \quad \text{and} \quad \frac{\partial x(t_j)}{\partial q_i} \tag{3.111}$$

are computed via finite differences and then the computations are restarted. The initial values of the states for the perturbed problem are given by the original solution at the current time. At the end (t_f) the derivatives for the second part

$$\frac{\partial x(t_f)}{\partial x(t_j)} \quad \text{and} \quad \frac{\partial x(t_f)}{\partial q_i} \tag{3.112}$$

are computed via finite differences as well. The overall derivative can the be derived by applying the chain rule

$$\frac{\partial x(t_f)}{\partial x_0} = \frac{\partial x(t_f)}{\partial x(t_j)} \frac{\partial x(t_j)}{\partial x_0} \tag{3.113}$$

$$\frac{\partial x(t_f)}{\partial q_i} = \frac{\partial x(t_f)}{\partial x(t_j)} \frac{\partial x(t_j)}{\partial q_i} + \frac{\partial x(t)}{\partial q_i} \tag{3.114}$$

This approach can be applied as often as necessary.

3.6.2 The Sensitivity Equations

The problem of computing the derivatives of the solution of Eq. 3.109 with respect to the initial conditions $x(0)$ and with respect to the parameters q_i can be reduced to that of computing the derivatives with respect to the initial values only. This is done by adjoining n_p differential equations to the original set of ODEs and by assuming that the parameters q_i are constant functions of time

$$\begin{pmatrix} \dot{x} \\ q_i \end{pmatrix} = \begin{pmatrix} f(t,x,q_i) \\ 0 \end{pmatrix}, \quad \begin{array}{rcl} x(0) & = & x_0 \\ q_i(0) & = & q_i \end{array} \tag{3.115}$$

Now the problem is reduced to that of computing the derivatives with respect to the initial conditions of the transformed problem, which can be rewritten as

$$\dot{z} = h(t, z), \quad z(0) = z_0 \tag{3.116}$$

where z the state vector of the extended system and is defined by $z = (x^T q_i^T)^T$. Differentiating Eq 3.116 with respect to the initial conditions gives the following linear matrix differential equation which is known as the 'variational equation' ([28]) or 'sensitivity equation' ([74])

$$\dot{S} = \frac{\partial h(t, z)}{\partial z_0} S, \quad S(0) = I \tag{3.117}$$

where S is defined as

$$S(t) = \frac{\partial z(t)}{\partial z_0} \tag{3.118}$$

The solution $S(t)$ is the derivative of $z(t)$ with respect to the initial conditions at time t. Transforming this equation back gives the required derivatives with respect to the original initial conditions, $x(0)$, and the parameters q_i

$$\dot{S} = \begin{pmatrix} \frac{\partial f(t,x)}{\partial x} & \frac{\partial f(t,x)}{\partial q_i} \\ & \\ 0 & 0 \end{pmatrix} S, \quad S(0) = I \tag{3.119}$$

where S is given by

$$S = \begin{pmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial q_i} \\ \\ \frac{\partial q_i}{\partial x_0} & \frac{\partial q_i}{\partial q_i} \end{pmatrix} \tag{3.120}$$

Since $q_i(t)$ is independent of x_0 and $\frac{\partial q_i}{\partial q_i} = 1$, the sensitivity equations for the original system can be written as

$$\dot{S}_{x_0} = \frac{\partial f(t,x,q_i)}{\partial x} S_{x_0}, \quad S_{x_0}(0) = I \tag{3.121}$$

$$\dot{S}_{q_i} = \frac{\partial f(t,x,q_i)}{\partial x} S_{q_i} + \frac{\partial f(t,x,q_i)}{\partial q_i}, \quad S_{q_i}(0) = 0 \tag{3.122}$$

where S_{x_0} is a matrix which denotes the sensitivities with respect to the initial conditions and S_{q_i} is a vector which denotes the sensitivities with respect to the parameters q_i. If analytic expressions for f_x and f_{q_i} are known they can be used, otherwise a finite difference approximation is used

$$\frac{\partial f}{\partial x} = \frac{f(t,x+\varepsilon_j,q_i) - f(t,x,q_i)}{\varepsilon_j}, \quad j = 1,...,n \tag{3.123}$$

and

$$\frac{\partial f}{\partial q_i} = \frac{f(t,x,q_i+\varepsilon_j) - f(t,x,q_i)}{\varepsilon_j}, \quad j = n+1,...,n+n_p \tag{3.124}$$

and the sensitivity equations are solved simultaneously with the original system. The remaining question is how to choose the perturbation parameter. The advantage of this approach is that the perturbation parameter can be adapted during the integration. This step size selection is critical to the success of the finite difference approximation. In [24] it is suggested to perturb half the digits of x_i and q_i. For example, for the case of $\frac{\partial x}{\partial q_i}$,

$$\varepsilon_i = \sqrt{u}|q_i| \tag{3.125}$$

perturbes half the digits of q_i where u is the unit round-off error. This kind of strategy works provided x is not near zero. An important consideration for any incremental selection strategy is that it should vary linearly with x. The following strategy can therefore be used

$$\varepsilon_j = \sqrt{u}\ \max(|q_i|, ATOL) \tag{3.126}$$

where $ATOL$ is a user chosen constant to prevent computational difficulties when $|p_i|$ is close to zero.

3.7 Implementation Issues

Here the implementation of an IRK method at 'Radau2a' collocation points ([29]) is described. It closely follows the implementation described in [29].

The array defined by Eq. 3.82 represents the collocation points and the parameters and is given by

$$
\begin{array}{c|ccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4-\sqrt{6}}{10} & \frac{296-169\sqrt{6}}{1800} & \frac{88-7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\hline
 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9}
\end{array}
\qquad (3.127)
$$

In order to reduce round-off errors, it is preferable to work with the smaller quantities

$$Z_i = x_i - x_0 \qquad (3.128)$$

If the matrix $A = (a_{ij})$ of the Runge Kutta coefficients is nonsingular, which is the case in Eq. 3.127, then using the transformation of Eq. 3.128 gives the IRK scheme

$$\begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \end{pmatrix} = A \begin{pmatrix} hf(t_0 + c_1, x_0 + Z_1) \\ hf(t_0 + c_2, x_0 + Z_2) \\ hf(t_0 + c_3, x_0 + Z_3) \end{pmatrix} \qquad (3.129)$$

$$x(t_n + h) = x_0 + Z_3 \qquad (3.130)$$

If Newton's method is used to solve the set of nonlinear equations given in Eq. 3.129,

PAGINATION AS IN ORIGINAL

3.8 Conclusions

The first part of this chapter reviews the basic foundations of the SQP method and describes a basic algorithm. It is possible to construct SQP algorithms which exploit the structure of the optimal control problem to allow a more efficient implementation. With a targeted implementation it is possible to solve large scale problems more efficiently than with a general purpose algorithm.

Different methods to reformulate the optimal control problem as a parameter optimisation problem are described. We found that for the class of problems addressed in this work a multiple shooting discretisation leads to a more robust implementation than the direct and single shooting discretisation. The direct discretisation method causes problems when solving stiff ODE models. We observed stiff behavior in the model equations in certain situations. This caused problems when implementing the direct discretisation approach. An advantage of the multiple shooting implementation is that the model can easily be tested during the development with the integration method used later in the optimisation. Such tests are not straightforward with a direct discretisation approach. A general comparison of the methods can be found in [15]. In general the multiple shooting discretisation leads to longer computations since the ODEs are solved fully for each function call in contrast to the direct approach where the ODEs and optimisation are solved simultaneously.

Chapter 4

The Integration of Design and Control for Fermentation Systems

In this chapter the problem of optimal design and operation of fermentation processes will be studied. The first section demonstrates how the problem can be formulated as an optimal control problem. The second section defines the term 'robust design' and describes a computational approach to obtain such a robust design. The third section outlines an approach for the integration of design and robust control suitable for fermentation processes. Here the two separate approaches of optimal design and robust control are combined into a single mathematical program. In this way the design and the robust control problems can be solved simultaneously. The advantage of such an integrated design method is that the final design is optimal and at the same time robust control of the plant is made possible.

The proposed method produces a robust open loop design. An open loop design means that the design parameters are selected once only and remain fixed during operation. In order to allow some degree of on-line adjustment, the method is extended to incorporate feedback controllers. However, a disadvantage of this method is that the design and operation of the plant is optimal only with respect to some nominal parameters. This means that it is possible that the plant operates far away from the optimum while fulfilling the robustness requirements. In order to avoid such drawbacks, the problem is formulated differently. In this new formulation, the nominal design is traded off against robustness properties. This is done by optimising simultaneously the objective functions corresponding both to the nominal and to the worst case models in a multi-objective framework.

4.1 Optimal Design and Control of Fermentation Processes

In this section the optimal design and control of a fermentation process based on a nominal model will be discussed. This problem is formulated as an optimal control problem:

$$\begin{aligned} \min_{x,u} \quad & \Phi(x(t_f)) & \\ \text{s.t.} \quad & \dot{x} = f(x, u, p_{nom}) & \text{model equations} \\ & g(x, u, t) \leq 0 & \text{design constraints} \end{aligned} \tag{4.1}$$

The objective function is the design objective. The differential equations are the model equations where the uncertain parameters, p, are fixed at their nominal values p_{nom}. The inequality constraints are the design constraints which have to be fulfilled at the solution of the problem. The solution is optimal with respect to the nominal parameters only. This is an optimal control problem which can be solved using any of the methods described in Chapter 3.

4.2 Robust Design and Control of Fermentation Processes

The robust control problem is defined in terms of computing the controls which steer the states whilst satisfying a set of inequality constraints for a certain set of uncertain parameters. This is illustrated in Fig. 4.1 as that of steering a process state from

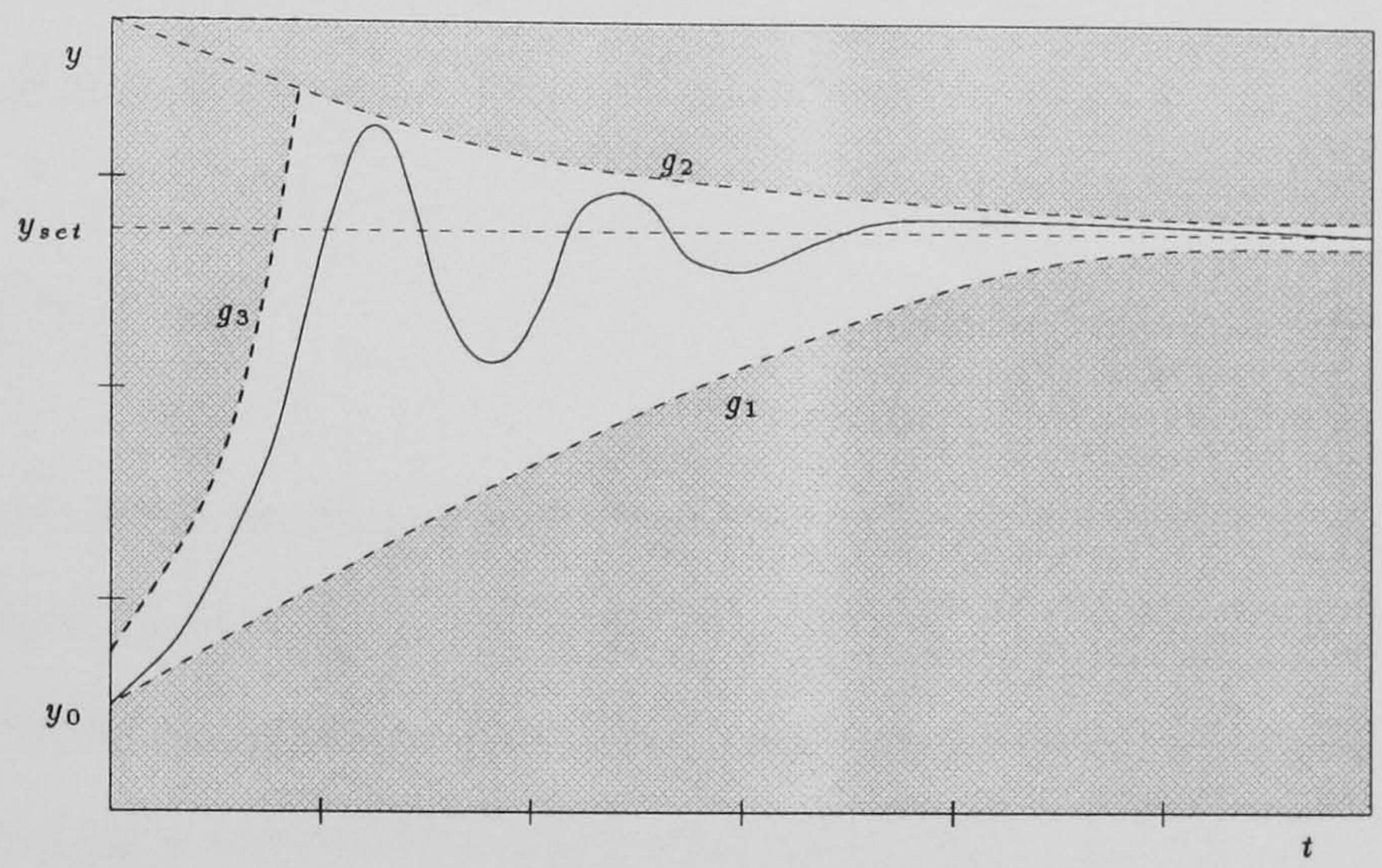

Figure 4.1: Moving the state y to y_{set} without violating the constraints g_1, g_2 and g_3.

an initial position y_0 to a new position y_{set} without violating the constraints g_1, g_2 and g_3. The problem is to determine a control trajectory which ensures the state does not violate any of these constraints during the transient period. The solution to this problem is robust if these constraints are not violated for all possible model uncertainties. This problem can be formulated in an optimal control framework. A similar problem formulation has been used for controller design in [70, 101, 100]. A complete description of an algorithm will be given in the next section. A more basic description is given below.

Prior to showing how to find a robust solution to the design and control problem, a mathematical definition of robustness is given first. A solution $(u^\star, x_0^\star)$ is called robust if it satisfies a set of robustness constraints with respect to a set of uncertain parameters

$$\left.\begin{array}{l} \dot{x} = f(x, u^\star, p) \\ g(x, u, t) \leq 0 \end{array}\right\} \forall p \in P \tag{4.2}$$

where the set P is defined as

$$P = \{p | \underline{p} \leq p \leq \overline{p}\} \tag{4.3}$$

This means that if the process is controlled by the input $u^\star$ starting from the initial condition $x_0^\star$, then none of the inequality constraints are violated irrespective of what value the uncertain parameter takes in the set P. The set P is static in the sense that the parameters p are uncertain but constant within this set over the entire time horizon. Time varying uncertainties can be incorporated into this formulation by the vector parameterisation approach. In this case, the time varying uncertainties are parameterised and bounded by box constraints. This representation is then included in the problem formulation.

The number of constraints in Eq. 4.2 is infinite since each value of the parameter vector p in the set P corresponds to a single set of constraints, and the set P is continuous. One approach to solve this type of problem is to approximate the feasible region, described implicitly by the infinite number of constraints, by a finite number of constraints. Algorithms based on this approach are sometimes called outer-approximation algorithms where the term 'outer' refers to the fact that the feasible region is approximated sequentially from outside the region. In these algorithms, $(u^*, x_0^*)_i$ at the i-th iteration is determined by solving the finite set of inequalities

$$\left.\begin{aligned} \dot{x} &= f(x,u,p) \\ g(x,u,t) &\leq 0 \end{aligned}\right\} \forall p \in P_i \subset P \tag{4.4}$$

where the set P_i contains only a finite number of elements. If the solution of this approximated problem is not a solution to the overall problem, the set P_i is updated according to

$$P_{i+1} = P_i \cup p_i \subset P \tag{4.5}$$

where p_i is the solution to

$$\max_{p \in P} \max_{t} g(x,u,t) \tag{4.6}$$

Problem 4.6 corresponds to the worst case constraint violation.

If

$$\max_{p \in P} \max_{t} g(x,u,t) \leq 0 \tag{4.7}$$

where the max operator is global, inequality 4.7 is satisfied $\forall p \in P$ and the algorithm terminates.

4.3 Robust Optimal Design and Control

In this section, robustness and optimal design and control methods are integrated to obtain combined robust optimal design methods. Problems 4.1 and 4.2 are combined into a single mathematical program. This is done by formulating Problem 4.2 as an additional set of constraints in Problem 4.1. The resulting robust optimisation problem can then be written as

$$\begin{aligned} &\min_{x,u} \quad \Phi(x(t_f)) && \\ &\text{s.t.} \quad \dot{x} = f(x,u,p_{nom}) && \text{model equations} \\ &\qquad\; g(x,u,t) \leq 0 && \text{design constraints} \\ &\qquad \left.\begin{array}{l} \dot{x}_p = f(x_p,u,p) \\ \phi(x_p,x,u,t) \leq 0 \end{array}\right\} \forall p \in P && \text{robustness constraints} \end{aligned} \tag{4.8}$$

where p_{nom} is the nominal value of the uncertain parameter vector. The set P of uncertain parameters is defined as

$$P = \{p | \underline{p} \leq p \leq \overline{p}\} \tag{4.9}$$

where $\underline{p}$ and $\overline{p}$ are the lower and upper bounds of the parameters respectively. This problem is called a **S**emi-**I**nfinite **O**ptimal **C**ontrol **P**roblem (SIOCP). It is called Semi-Infinite because the robustness constraints have to be satisfied for a continuous range of parameter values. This results in an infinite number of constraints which have to be satisfied at the solution. The solution to this problem is optimal in the sense that it optimises the nominal performance of the process and satisfies the robustness constraints for all parameter variations within the set P. If the actual parameters are different from p_{nom}, optimality is lost but robustness (feasibility) of the solution is maintained.

The reformulation of an optimal control problem as a NLP problem as described in Chapter 3 can be applied to the SIOCP problem to obtain an associated parameter optimisation problem which approximates the original problem. The discretised states are represented by the vector y and the control vector parameters and the initial conditions are represented by the vector q in the reformulated problem. An appropriate reformulation of the inequality constraints will be discussed later when a solution algorithm for this problem is described in detail. The reformulated problem is a **S**emi-**I**nfinite **P**rogramming (SIP) problem since it still has an infinite number of constraints relating to the robustness constraints. The resulting **D**iscretised **S**emi-**I**nfinite **O**ptimal **C**ontrol **P**roblem (DISIOCP) can then be written as

$$\begin{aligned} \min_{y,q} \quad & \Phi(y(t_f)) \\ \text{s.t.} \quad & h(y, q, p_{nom}) = 0 \\ & g(y, q, t) \leq 0 \\ & \left.\begin{aligned} & h(y_p, q, p) = 0 \\ & \phi(y_p, y, q, t) \leq 0 \end{aligned}\right\} \quad \forall p \in P \end{aligned} \tag{4.10}$$

where $h(\cdot) = 0$ are the discretised equations. Based on the particular method used to solve the ODEs, these equations represent for example continuity equations in the multiple shooting discretisation or the discrete equations in the direct discretisation. In the single shooting method these equations, ($h(\cdot) = 0$) are not present, however, the constraints remain to deal with the more general case.

Problem 4.10 cannot be solved directly using an SQP algorithm. An algorithm which uses SQP and solves a sequence of approximations to Problem 4.10 is presented in the next section.

4.3.1 A Solution Algorithm

In this section we describe the use of an SQP method to solve the SIP. The latter problem is an approximation of the SIOCP. The procedure is similar to that of finding a feasible point to an infinite number of constraints (the robustness problem) described in Section 4.2. The general method to solve a SIP is shown schematically in Figure 4.2. The original SIP problem is approximated by a NLP problem. The approximate problem can be solved using the standard SQP method in which the infinite number of constraints (in this case the robustness constraints) are replaced by a finite number of constraints. The approximation is further refined and the solution process progresses until the original problem is solved.

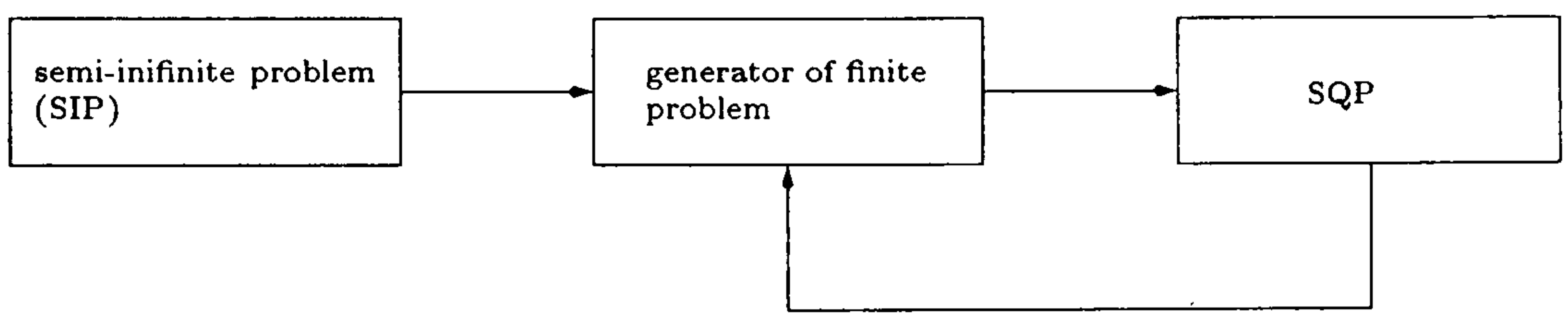

Figure 4.2: Solution method for SIP problems

Different variants of the above algorithm are discussed in [33]. The basic steps are described as follows

(a) $i = 0$. Start with an initial discretisation P_i of the continuous set P, $P_i \subset P$.

(b) Find a solution $(q^{\star})_i$ to DISIOCP corresponding to the discretised set P_i

(c) Find all the local constraint maximisers $p_1, \ldots, p_k$ as the solution to the problem [1]

$$\begin{aligned} \max_{p \in P} \max_{j} \max_{t} \quad & \phi_j(y_p, y, q^\star, t) \\ \text{s.t.} \quad & h(y, q^\star, p_{nom}) = 0 \\ & h(y_p, q^\star, p) = 0 \end{aligned} \tag{4.11}$$

and stop if

$$\begin{aligned} & h(y, q^\star, p_{nom}) = 0 \\ & h(y_p, q^\star, p_l) = 0 \\ & \phi(y_p, y, q^\star, t) \le 0 \end{aligned} \tag{4.12}$$

for $l = 1, \ldots, k$.

(d) Delete some of the elements in the set P_i. Add the maximisers $p_1, \ldots, p_k$ to the set P_i to obtain a new set P_{i+1} for the next iteration

$$P_{i+1} = P_i \cup \{p_1, \ldots, p_k\} \tag{4.13}$$

Instead of adding all the local maximisers to the set, only the global maximiser can be added. In this case it is sufficient in step (c) to find only the global maximiser and not all the local maximisers.

(e) Increment i: $i = i + 1$ and go to (b)

Each of the sub-problems which has to be solved in the individual steps is discussed below in more detail. It is important that all the sub-problems are considered in a consistent way. This means that the treatment of the path constraints of the optimal

[1]Note that a general path constraint $g_j(x, u, t) \le 0, \quad \forall t \in [0, t_f]$, is equivalent to the constraint $\max_{t \in [0, t_f]} g_j(x, u, t) \le 0$.

control problem in step (b) should be compatible with that of the objective function when searching for a constraint maximiser in step (c). This is because the original problem is solved approximately and the same approximation should be used in each step.

(a) Choice of an initial set P_1

An initial guess of the worst case uncertainty can often be obtained from an engineering insight into the problem. It may be useful to start with more than one element in the set. In some cases if the problem is initialised with a non-empty set P_1, it is is solved faster than initialising with a set P_1 which corresponds to the original nominal design problem.

(b) Solution of SIOCP

The SIOCP corresponding to a discretised set P_i is a standard optimal control problem which can be solved using a control vector parameterisation technique explained in Chapter 3. The ODEs can be discretised using one of the methods described in that chapter. The path constraints are discretised at certain points in time (t_i). The reasons behind this discretisation will be given in the next paragraph. The optimal input and initial conditions of this problem are given by $(q^\star)_i$.

(c) Computation of a constraint maximiser

The purpose of this step is to establish whether or not the solution $(q^\star)$ obtained in the previous step satisfies the robustness constraints for all possible parameters in the continuous set P and not only for those discretised parameters in the set P_i. We need to identify an appropriate reformulation for the path constraints. Possible reformulations were discussed in Chapter 3.

If the path constraints are reformulated as end point constraints, the gradients of all the constraints at the solution are zero. Computing a constraint maximiser is therefore not possible due to the zero gradients. The same situation holds for the other variants of this reformulation as discussed in Chapter 3. In this case, the gradients of the active constraints are non-zero at the solution but the gradients of the inactive constraints are zero. This means that for those inactive constraints no constraint maximiser can be found since no information about these constraints is available to the optimiser. The same arguments apply if a penalty approach is employed.

It can be concluded that combining path constraints into a single term (end point or penalty term) may be suitable in the context of solving the OCP but not to find constraint maximisers.

The above disadvantages do not apply in the discretisation approach. The discretisation approach is more suitable in this context, however, ts main disadvantages are amplified when solving SIOCP. It is possible using the above algorithm to find a solution to the SIOCP which satisfies the discretised robustness constraints $\forall p \in P_i$, however this does not imply that the original path constraints (the continuous robustness constraints) are satisfied.

In principle, it is possible to uses different reformulation in each of the steps. One needs to make sure that in step (b) when solving an OCP the constraint reformulation guarantees a stricter satisfaction of the path constraints than that in step (c) when searching for a constraint maximiser. Thus if in step (b) an exact satisfaction of the original set of constraints is achieved, then the discretisation approach can be applied in step (c). Since the end point constraint reformulation does not have good convergence properties ([52]), the discretisation approach is used in all steps of this algorithm.

If we apply the discretisation scheme for path constraints described in Chapter 3 to

Problem 4.11 the problem becomes

$$\begin{aligned} \max_{p\in P}\max_{j,y,y_p} \quad & \hat{\phi}_j(y_p, y, q^\star) \\ \text{s.t.} \quad & h(y, q^\star, p_{nom}) = 0 \\ & h(y_p, q^\star, p) = 0 \end{aligned} \tag{4.14}$$

where $\hat{\phi}$ is a vector containing all the discretised path constraints. The problem of finding the largest constraint violation over the whole time horizon then simplifies to that of finding the largest element in the vector of the discretised set of constraints. This is a *bilevel problem.* The objective function of the outer problem is implicitly defined by an inner optimisation problem. The solution of this problem is defined by the largest element of the vector of constraints. The most obvious method to solve this problem, as described by [26] for steady state-models and in [97, 3] for dynamic models, is to exchange the two max-operators and then circumvent the outer $\max_j$-problem by solving for each element in the constraint vector. Since the number of inequality constraints resulting from a discretisation of the path constraints can become rather large resulting in a large number of NLPs which have to be solved, a more efficient scheme would be desirable. In [97] it is argued that approximating Problem 4.14 by

$$\begin{aligned} \max_{y,y_p,\epsilon,p\in P} \quad & \epsilon \\ \text{s.t.} \quad & \log\left(\sum_{j=1}^{N} \exp\left(\frac{\hat{\phi}_j - \epsilon}{\nu_j}\right)\right) = 0 \\ & h(y, q^\star, p) = 0 \\ & h(y_p, q^\star, p) = 0 \end{aligned} \tag{4.15}$$

to avoid non-differentiabilities leads to a fewer number of NLPs. Here ν_j corresponds to the smallest significant constraint violation of $\hat{\phi}_j$.

In order to use one of the methods described in Chapter 3 to solve optimal control problem 4.14 it needs to be transformed in a more amenable form. It is transformed into a standard optimal control problem. The following problem is equivalent to Problem 4.14

$$\begin{aligned} \max_{p \in P} \min_{\delta, y, y_p} \quad & \delta \\ \text{s.t.} \quad & h(y, q^\star, p_{nom}) = 0 \\ & h(y_p, q^\star, p) = 0 \\ & \hat{\phi}(y_p, y, q^\star) \leq \delta \end{aligned} \tag{4.16}$$

where the index parameter j is avoided. This is still a bilevel problem with a possible non-differentiable inner problem. Problem 4.16 can be solved using a procedure based on the complete solution of the inner problem as proposed by [40], i.e. solving the problem in a feasible path approach. Such an approach is likely to fail. The difficulty here is that the function defined implicitly by the inner problem may not be continuously differentiable. Most optimisation algorithms are based on the assumption that the objective function is continuously differentiable. A stochastic optimisation algorithm to overcome this difficulty is used in [38] to solve the outer problem in the context of robust controller design.

Most of the methods proposed in the literature ([96]) to solve bilevel mathematical programs are based on replacing the inner optimisation problem by a set of constraints to the outer problem. These constraints define the optimality conditions of the inner problem. The resulting problem is a NLP problem which can be solved using any appropriate NLP solver. The KKT conditions of the inner problem are

$$h(y, q^\star, p_{nom}) = 0 \tag{4.17}$$

$$h(y_p, q^\star, p) = 0 \tag{4.18}$$

$$\hat{\phi}(y_p, y, q^*) \leq \delta \tag{4.19}$$

$$\sum_i \mu_i \frac{\partial h_i}{\partial y} + \sum_k \lambda_k \frac{\partial(\hat{\phi}_k - \delta)}{\partial y} = 0 \tag{4.20}$$

$$\sum_j \nu_j \frac{\partial h_j^p}{\partial y_p} + \sum_k \lambda_k \frac{\partial(\hat{\phi}_k - \delta)}{\partial y_p} = 0 \tag{4.21}$$

$$\sum_j \lambda_j = 1 \tag{4.22}$$

$$\lambda_j(\phi_j - \delta) = 0 \tag{4.23}$$

$$\lambda_j \geq 0 \tag{4.24}$$

Eq. 4.20 and Eq. 4.21 are the stationary conditions with respect to the state variables (y_p, y). These conditions can be ignored if it is assumed that the Jacobians of the equality constraints are nonsingular at the solution. Under this assumption it is always possible to find multipliers (μ, ν) such that these conditions are fulfilled. This is so because multipliers do not appear elsewhere in the set of Eqs. 4.17-4.24 and can be freely chosen to satisfy Eqs. 4.20 and 4.21.

The overall problem is now given by

$$\begin{aligned} \max_{\delta, y, y_p, \lambda, p \in P} \quad & \delta \\ \text{s.t.} \quad & h(y, q^*, p_{nom}) = 0 \\ & h(y_p, q^*, p) = 0 \\ & \hat{\phi}(y_p, y, q^*) \leq \delta \\ & \textstyle\sum_j \lambda_j = 1 \\ & \lambda_j(\phi_j - \delta) = 0 \\ & \lambda_j \geq 0 \end{aligned} \tag{4.25}$$

The complementary condition 4.23 can lead to computational difficulties when searching for an improved feasible direction at degenerate points ([19]). This condition defines the constraints which are active. It is taken into account in [19] and [26] where an active set methodology is developed which replaces this condition by constraints involving binary variables that indicate which constraints are active at the solution. An alternative method is presented in [18] where the complementary constraint is replaced by a penalty term in the objective function. Penalty function algorithms exchange a constrained optimisation problem with a sequence of unconstrained problems. Constraint violations are prevented (in the limit) by adding penalty terms to the objective function. The main advantage of such an approach is that the complementary condition (which is included in the penalty term) needs only to be satisfied in the limit when the solution of the modified problem converges to the solution of the original problem. The solution of the sequence of unconstrained problems circumvents the need for an active constraint strategy and avoids the combinatorial search problem at degenerate points ([19]). The main disadvantage of a penalty function approach is that it can lead to ill-conditioned sub-problems ([24].

A key advantage of the penalty function approach is that it relaxes the discrete nature of the complementary condition. In [19] it is proposed to relax the complementary conditions associated with the KKT criteria instead of including them in a penalty term in the objective function and to solve a series of sub-problems with a decreasing value of the relaxation constant r

$$-\lambda_j(\hat{\phi} - \delta) \leq r \tag{4.26}$$

is solved. By applying this strategy, the ill-conditioning of the sub-problems can be avoided. The overall problem can then be written as

$$\begin{aligned}
\max_{\delta, y, y_p, \lambda, p \in P} \quad & \delta \\
\text{s.t.} \quad & h(y, q^\star, p_{nom}) = 0 \\
& h(y_p, q^\star, p) = 0 \\
& \hat{\phi}(y_p, y, q^\star) \le \delta \\
& \textstyle\sum_j \lambda_j = 1 \\
& -\lambda_j(\phi_j - \delta) \le r \\
& \lambda_j \ge 0
\end{aligned} \tag{4.27}$$

whose solution converges to the solution of the original problem as $r \to 0$. Note here that the sign restrictions on $(\hat{\phi} - \delta)$ and λ mean that the product $-(\hat{\phi}_j - \delta)\lambda_j$ is bounded from below by zero. The solution of each sub-problem serves as an initial estimate of the solution of the next sub-problem with a decreased r.

As an example, consider the following bilevel optimisation problem where the state variables have been eliminated by solving the state equations

$$\begin{aligned}
\max_{p \in P} \min_{\delta} \quad & \delta \\
\text{s.t.} \quad & g_i \le \delta, \quad i = 1, 2, 3
\end{aligned} \tag{4.28}$$

The set P is defined by $P = \{p | 0 \le p \le 11\}$ and

$$g_1 = 0.75p - 0.75 \tag{4.29}$$

$$g_2 = -10p + 14 \tag{4.30}$$

$$g_3 = -0.5p + 3 \tag{4.31}$$

The solution of the inner problem as a function of p is shown in Fig. 4.3 by the solid line. The objective function defined implicitly by the inner problem is non-

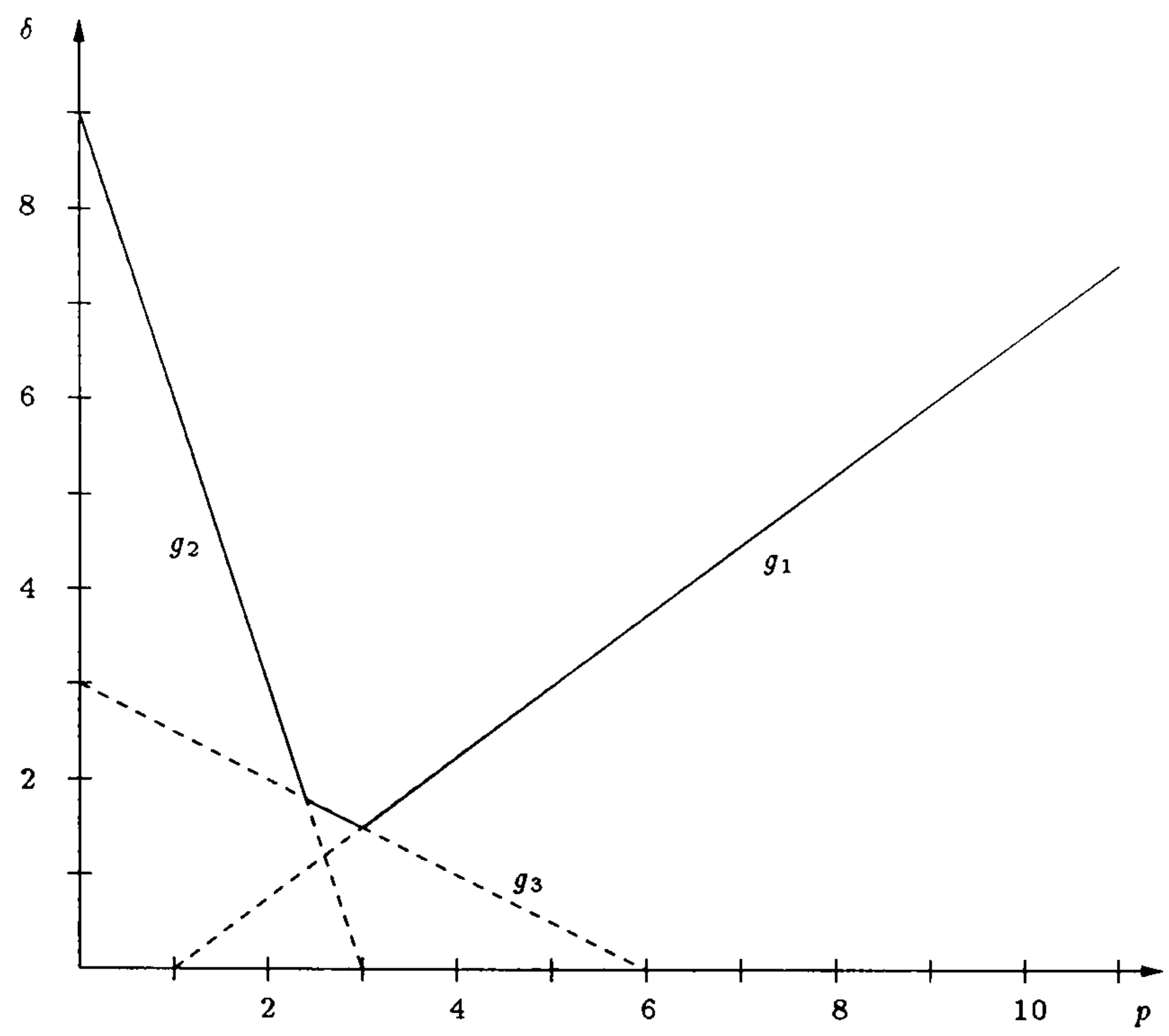

Figure 4.3: Constraints of a bilevel optimisation problem

differentiable. Replacing the inner optimisation problem by the relaxed KKT conditions gives the following NLP problem

$$\begin{aligned} \max_{p\in P,\delta} \quad & \delta \\ \text{s.t.} \quad & g_i \leq \delta, \quad i = 1,2,3 \\ & \textstyle\sum_i^3 \lambda_i = 1 \\ & -\lambda_i(g_i - \delta) \leq r, \quad i = 1,2,3 \end{aligned} \tag{4.32}$$

The original KKT conditions which define the inner optimisation problem require the knowledge of which constraints (dotted lines) are active for a given p. The feasible region of the original problem is shown by the solid line in Fig. 4.3. Figure 4.4 shows

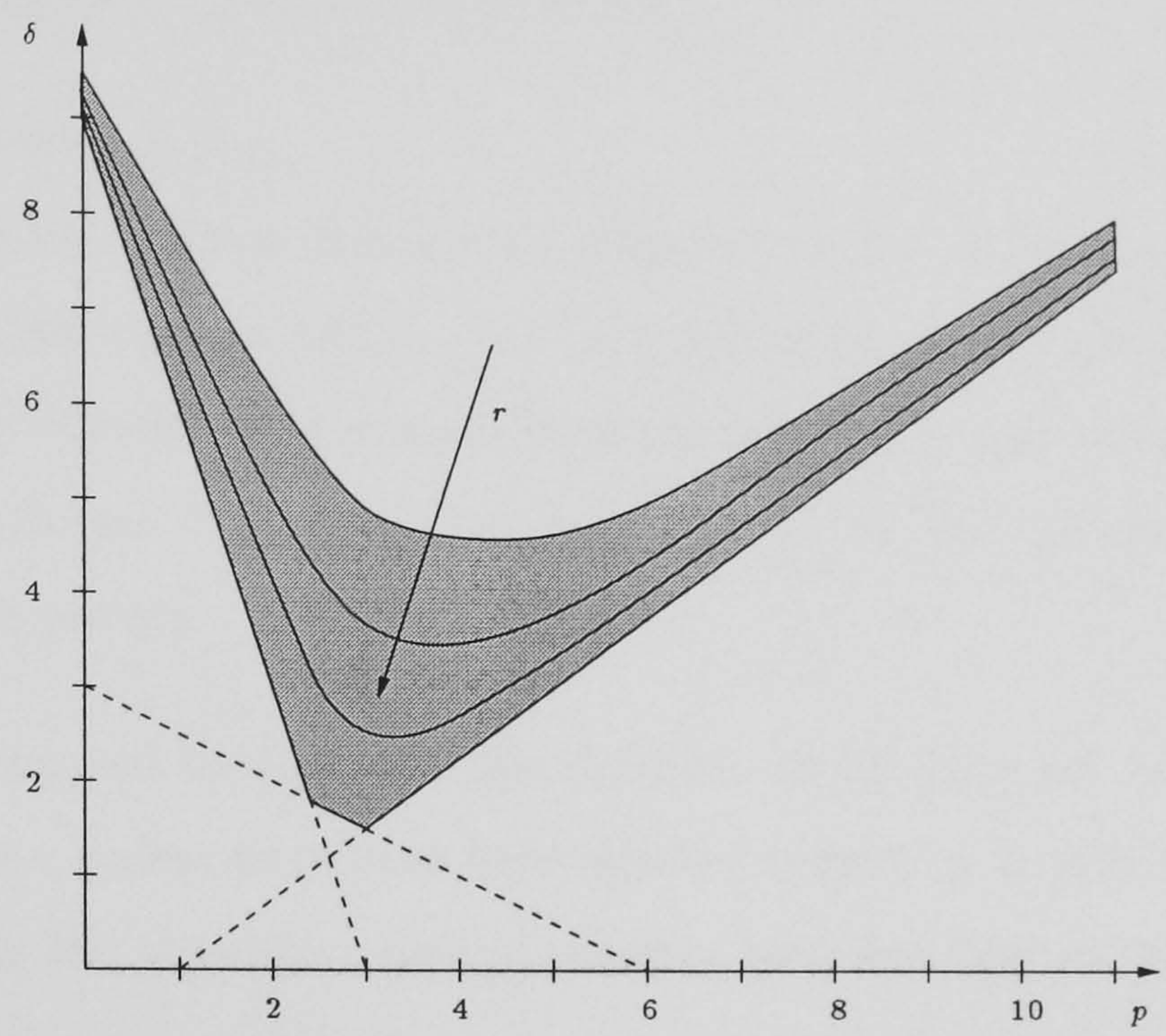

Figure 4.4: Feasible region of the relaxed bilevel optimisation problem

the feasible region (shaded area) of the optimisation problem defined by the relaxed KKT conditions projected into the δ, p-space. The arrow indicates decreasing values of r. The relaxation of the complementary condition gives a smoother representation of the inner optimisation problem.

It may happen that the algorithm for the overall problem terminates at a stationary point which is not the global maximum. This can be shown for the above example in Figure 4.4. There are two maxima for the problem at $p = 0$ and $p = 11$. In order to increase the chances of locating the global constraint maximiser, a local search is started from each vertex of the P- space. In this example, each of the initial points of the search is a solution to the problem since they are local maximisers.

The overall algorithm terminates if it cannot find a parameter p, within the set P, which corresponds to a constraint violation. The solution (q^*) defines a robust solution to the overall optimisation problem.

Selection of a new set P_{i+1}

This step includes removal of some old elements in the set P_i and addition of new elements to obtain the new set P_{i+1} for the next iteration. The most straightforward approach is not to delete any element from the old set and add the global or all local maximisers to the set P_i to obtain the new set P_{i+1}. In [33], the convergence of such an algorithm is proved.

In [97] it is proposed to delete all the elements which have not been active in the last n iterations, unless they have been deleted before. n is defined a priori. This strategy avoids the algorithm cycling between inclusion and rejection of the same constraint.

4.4 Robust Feedback Design

The previous sections described open-loop solutions for the integrated design and control problem. The design parameters and control profiles are determined once only and are not changed during process operation. This can lead to very conservative results since the solution of the problem is always constrained by the worst case scenario. An improvement to the objective function can be expected if the control profiles can be adjusted based on process measurements. In principle, different modes of control can be incorporated in the optimisation problem. In [95] number of methods for adjusting the controls are described:

(a) Optimal closed-loop control method: the control inputs are computed as feed back laws via dynamic programming.

(b) Open-loop control method without updating: the control inputs are based on initial information only.

(c) Open-loop control method with updating: same as (b) except that controls are recomputed as new information becomes available. This can be considered as an online optimiser scheme.

(d) Mean value approximation method: the closed loop control is computed with a fixed set of parameters (mean values). It can be used within a)-e).

(e) Best control law within a class: the feedback control is a parameterised class of control laws. One solves for the best control law within the specified class.

If method (a) is used, the best performance is expected. Unfortunately, it is not straight-forward to include a dynamic programming problem within an optimisation problem. Furthermore one is also faced with the 'curse of dimensionality' associated with dynamic programming. Method (b) applied to a fed batch fermenter results in an open loop problem which can be solved by methods discussed previously depending on the aim of control. This approach is an open loop approach. Method (c) is very similar to method (b). Instead of computing new control inputs at the beginning, controls are always recomputed at certain times. Such a scheme corresponds to a Model Predictive Control (MPC) scheme. Thus the overall robust design and control problem is a bilevel optimisation problem with multiple lower levels. Each level corresponds to a single MPC problem. If this problem is solved as a mathematical program by replacing the lower level problems by the corresponding KKT conditions, a very high dimensional problem results ([10]). Method (d) is faced with the same difficulties as the methods (a)-(c).

Method (e) converts the closed-loop problem into an open-loop one where the feedback loop is part of the model. The controller (e.g. PI, state feedback) is included into the model and the control parameters then act as design parameters of the process. Of course, the result is influenced by the type of controller used. This method can be incorporated readily into the process model once the feedback parameterisation is given. This approach has been proposed in [34] for the design of optimal feedback controllers.

A critical point in method (e) is how to impose bounds on the control input in the problem formulation. For the sake of argument assume that the control parameterisation in the model is given by

$$u(t) = f_c(x) \tag{4.33}$$

This description includes for example a state feedback controller. Imposing the bounds in the control variable by simply adding the path constraints

$$\underline{u} \leq u(t) \leq \overline{u} \tag{4.34}$$

may result in a compromise between periods where the constraint is active and when it is inactive. Such a constraint would imply that the control variable should never saturate. Since avoiding control saturation is usually not a control objective by itself, it should not affect the solution as long as it does not lead to the violation of any other constraint. It should be possible that control saturation can occur during operation! A more appropriate controller model for this case is the smoothed saturation function

$$u(t) = \begin{cases} \overline{u} & \text{if } \; f_c(t) > \overline{u} + \epsilon \\ \overline{u} - \frac{(f_c(t) - \overline{u} - \epsilon)^2}{4\epsilon} & \text{if } \; \overline{u} - \epsilon < f_c(t) \leq \overline{u} + \epsilon \\ f_c(t) & \text{if } \; \underline{u} + \epsilon \leq f_c(t) \leq \overline{u} - \epsilon \\ \underline{u} + \frac{(-f_c(t) + \underline{u} - \epsilon)^2}{4\epsilon} & \text{if } \; \underline{u} - \epsilon \leq f_c(t) < \overline{u} + \epsilon \\ \underline{u} & \text{if } \; f_c(t) < \underline{u} - \epsilon \end{cases} \tag{4.35}$$

where ϵ is a small positive scalar. The smoothing is necessary in order to avoid non differentiability in the optimisation problem.

4.5 A Multi-objective Problem Formulation

The robust design problem 4.8 is optimal with respect to the nominal model parameters. If these parameters take on values other than their nominal values, the process is not optimal anymore. Although it is guaranteed that the robustness constraints are satisfied for all allowed parameter variations, the design constraints can be violated even for small parameter perturbations. It is indeed possible that the process does not operate in a profitable way anymore. Special insight into the problem is therefore required to safeguard against this failure.

An alternative approach to design a process by optimising its nominal model is to optimise the worst case scenario and additionally to guarantee that the design constraints are satisfied for all possible parameter combinations.

$$\begin{aligned}\min_{u(t),x(t),t_f} \quad & \max_{p\in P} \Phi(x(t_f)) \\ \text{s.t.} \quad & \dot{x} = f(x,u,p) \\ & \left.\begin{aligned} \dot{x}_p = f(x_p,u,p) \\ g(x_p,u,t) \leq 0 \end{aligned}\right\} \quad \forall p \in P\end{aligned} \tag{4.36}$$

This problem formulation can however lead to unnecessarily conservative designs because the final solution is based only on the worst possible outcome which is probably not often encountered in practice. We have outlined so far two different objective functions. One which is too optimistic because it does not take any parameter variations into account, and one which is too pessimistic because it assumes that the parameters will always force the worst possible outcome. One way to overcome the problem of designing either a too optimistic or a too conservative process is to trade off these two objectives against each other in a multi-objective optimisation framework. In a multi-objective optimisation problem the aim is to find the optimum of two objective functions with the same constraints [18]. This problem results in the following multi-objective formulation.

$$\begin{aligned}\min_{u(t),x(t),t_f} \quad & \begin{pmatrix} \Phi(x(t_f)) \\ \max_{p\in P} \Phi(x(t_f)) \end{pmatrix} \\ \text{s.t.} \quad & \dot{x} = f(x,u,p) \\ & \left.\begin{aligned} \dot{x}_p = f(x_p,u,p) \\ g(x_p,u,t) \leq 0 \end{aligned}\right\} \quad \forall p \in P\end{aligned} \tag{4.37}$$

One approach to solve multi-objective optimisation problems is to turn one objective function into an ε-constraint which ensures that this objective does not exceed a certain value ε. The problem can now be solved for different values of ε in order to construct a trade off curve ([18]). In this formulation, the worst case objective is turned into a constraint. By noting that

$$\max_{p \in P} \Phi(x(t_f)) \leq \varepsilon \quad \Longleftarrow \quad \Phi(x(t_f)) \leq \varepsilon \quad \forall p \in P \tag{4.38}$$

(In case the max operator is global, the relation is true in both directions). Problem 4.37 can be written using the ε-constraint formulation as

$$\begin{aligned} \min_{u(t),x(t),t_f} \quad & \Phi(x(t_f)) \\ \text{s.t.} \quad & \dot{x} = f(x,u,p) \\ & \left.\begin{aligned} & \dot{x}_p = f(x_p,u,p) \\ & g(x_p,u,t) \leq 0 \\ & \Phi(x_p(t_f)) \leq \varepsilon \end{aligned}\right\} \quad \forall p \in P \end{aligned} \tag{4.39}$$

This problem has the same form as the nominal design problem with the robustness constraints (Problem 4.8) and therefore the same algorithm introduced in Section 4.3 can be used to solve it.

The above problem can be modified so that the second objective does not represent the worst case of the original objective function. Other objective functions relating to specific control objectives can be used as will be demonstrated in the case studies.

This problem can also be viewed in the framework presented in Section 4.3 where the nominal case is optimised and an additional set of robustness constraints is introduced. In Problem 4.39 the nominal case is also optimised. The design constraints are turned into robustness constraints such that they are satisfied for all possible parameter combinations. An additional robustness constraint is introduced which ensures that the objective function (or any other performance measure) is bounded by ε for all parameters p in P. Solving the problem for different bounds ε results in a set of Pareto optimal solutions ([18]) for the multi-objective optimisation problem.

4.6 Conclusions

The first part of this chapter presents a method to design a robust controller for a fed batch fermentation. A specific controller parameterisation is assumed and the controller parameters are included in the problem formulation. This problem formulation is incorporated into an optimal design procedure. As a result it is possible to design a fermentation process which is controllable with respect to the assumed controller. In order to guarantee acceptable performance a second objective function is introduced in the problem formulation which is based on a worst case scenario. This second objective function can be reformulated in a multiobjective optimisation framework.

Chapter 5

Case Studies

5.1 Controllability Analysis of a Continuous Fermentation

In this case study different designs of a continuous fermentation are compared with respect to their controllability properties. Four designs of a single cell producing fermenter are considered. These are common designs for continuous fermenters which can be found in the literature (e.g. [62]). *Design 1* (Figure 5.1a) is a single stream fermenter with one inlet and one outlet stream. The input flow rate and the feed substrate concentration are considered as possible manipulated inputs.

To enable manipulation of the overall feed concentration whilst maintaining a constant flow rate (as assumed in *Design 1*), two streams with different concentrations must be employed. Instead of using the feed concentration as a manipulated input, *Design 2* (Figure 5.1b) uses the flow rate of one of the inlet streams as the control input.

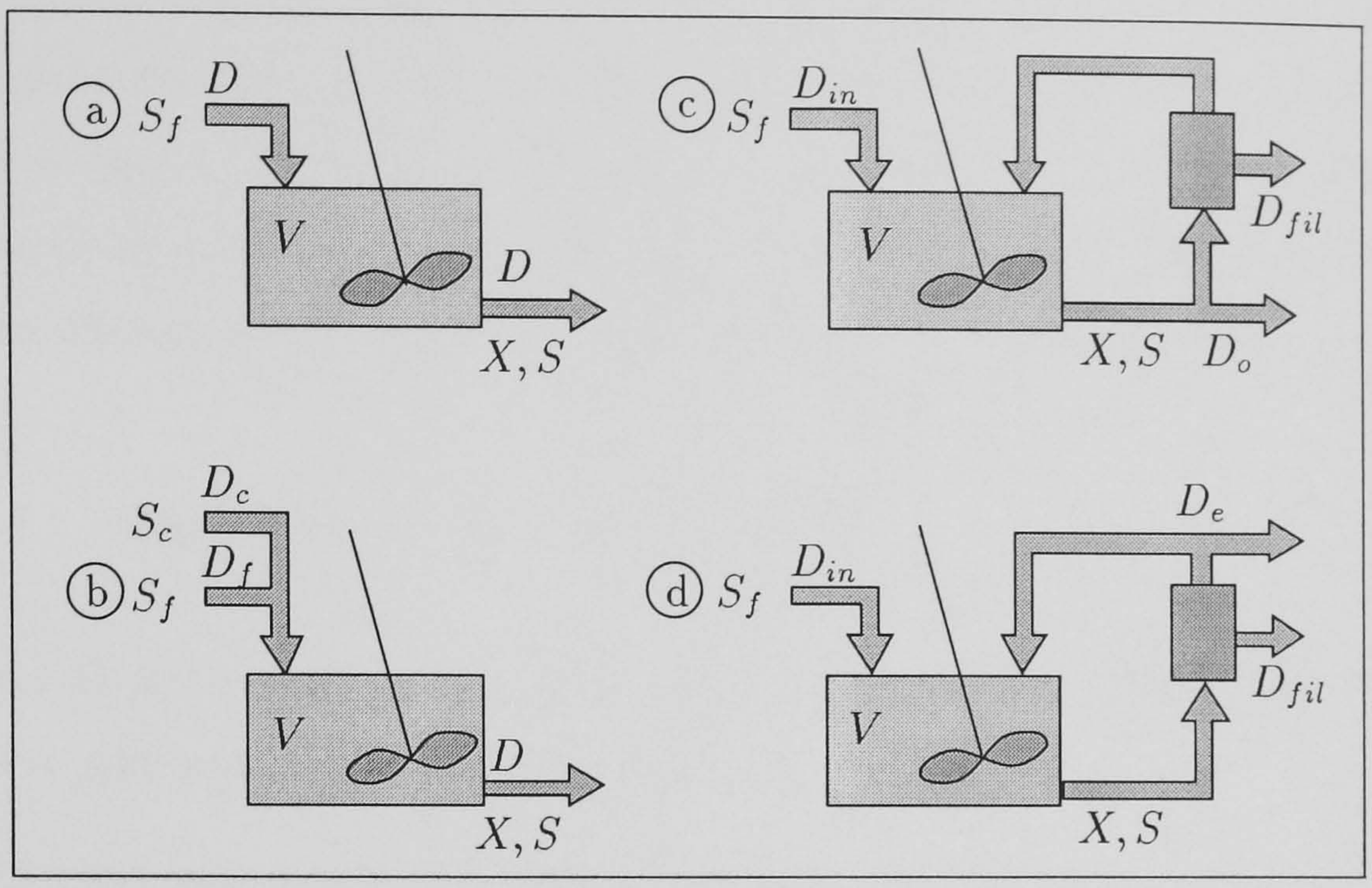

Figure 5.1: The four different fermenter designs: a) Design 1; b) Design 2; c) Design 3; d) Design 4

In *Designs 3* and *4* (Figures 5.1c & d) a recycle is introduced. The difference between the two designs is that in *Design 3* the product is withdrawn before the filter whereas in *Design 4* it is withdrawn after the filter. The inlet and the product outlet flow rates are considered as manipulated inputs under different recycles.

The nonlinear and the linearised models corresponding to the four designs are derived in the next section.

5.1.1 Nonlinear Dynamic Models

In all four designs the fermenter is described by an unstructured model. The specific growth rate is modeled by a Monod kinetic relationship

$$\mu(S) = \mu_{max} \frac{S}{K + S} \tag{5.1}$$

where S is the substrate concentration, μ_{max} is the maximum specific growth rate and K is the substrate saturation constant. For all reactors it is assumed that they have constant volume, their content is well mixed and the feed is sterile. The nominal Monod model parameters are $K = 0.05\ g/l$ and $\mu_{max} = 0.4\ 1/h$.

5.1.1.1 Single-Stream Fermenter: *Design 1*

Figure 5.1a shows a fermenter with one inlet and one outlet stream. The nonlinear model corresponding to this design is given by

$$\begin{aligned} \dot{X} &= \mu(S)X - XD \\ \dot{S} &= -\frac{\mu(S)}{Y}X + (S_f - S)D \end{aligned} \tag{5.2}$$

where X is the cell concentration. S_f is the feed substrate concentration, Y is the yield coefficient and D is the dilution rate. The nominal values of Y and S_f are taken as $0.4\ g/g$ and $1\ g/l$ respectively. The inlet flow rate or the inlet substrate concentration can be used as manipulated inputs.

The design objective is to maximise the profit per unit volume and unit time at steady state conditions. This objective can be formulated as maximising

$$Q = q_x DX - q_s DS_f \tag{5.3}$$

where q_x and q_s are the cost coefficients related to the product and the substrate, respectively. It can be shown that the optimal dilution rate which maximises the profit is

$$D_{opt} = \mu_{max}\left(1 - \sqrt{\frac{K}{S_f + K} + c\,\frac{S_f}{Y\mu_{max}^2(S_f + K)}}\right) \tag{5.4}$$

where $c = q_s/q_x$ is the ratio of the cost coefficients. Different sets of operating points are obtained by varying the cost coefficients. The first operating point ($OP1$) corresponds to $c = 0.035$, whereas the second operating point ($OP2$) corresponds to a negligible cost of feed substrate ($c = 0$). These two operating points fall into regions where it is common to operate fermenters. $OP1$ corresponds to an intermediate dilution rate and the dilution rate of $OP2$ is that which yields optimal productivity, however $OP2$ is close to the wash out.

The steady state values corresponding to the operating points $OP1$ and $OP2$ are shown in Table 5.1.

	X [g/l]	S [g/l]	D_{in} [l/h]
OP 1	0.381	0.047	0.1947
OP 2	0.328	0.179	0.3127

Table 5.1: Nominal operating points for *Design 1*

5.1.1.2 Two - Stream Fermenter: *Design 2*

Figure 5.1b shows a fermenter with two inlet streams D_c and D_f. The two inlet streams have different feed concentrations S_f and S_c respectively. The process is modeled by

$$\begin{aligned}\dot{X} &= \mu(S)X - X(D_f + D_c)\\ \dot{S} &= -\frac{\mu(S)}{Y}X + (S_f - S)D_f + (S_c - S)D_c\end{aligned} \tag{5.5}$$

The two streams are designed such that the mixed stream which enters the fermenter has the same flow rate and the same feed concentration as the inlet flow in *Design 1* at the two operating points $OP1$ and $OP2$. At nominal operating conditions, there is no difference between *Designs 1* and *2*. The dilution rate of the control flow D_c is made to satisfy certain bounds. The concentration of the control stream S_c is used as a design parameter and is varied from 0 to 200 % of the nominal feed concentration of *Design 1* at each of the two operating points.

5.1.1.3 Continuous Recycle Fermenter: *Designs 3 & 4*

Figures 5.1c & 5.1d show fermenters with recycle and filter units. The processes are modeled by

$$\begin{aligned} \dot{X} &= \mu(S)X - X\left((1-\frac{\epsilon-1}{\kappa-1})D_o + (\frac{\epsilon-1}{\kappa-1})D_{in} + (\frac{\epsilon}{\kappa} - \frac{\epsilon-1}{\kappa-1})D_e\right) \\ \dot{S} &= -\frac{\mu(S)}{Y}X - (S - S_{in})D_{in} \end{aligned} \tag{5.6}$$

where D_o is the dilution rate withdrawn before the filter and D_e is the dilution rate withdrawn after the filter. D_{in} is the dilution rate entering the fermenter, and D_{fil} is the stream leaving the filter. κ is the split fraction and ϵ the separation factor of the filter. The filter is assumed to be static and is modeled by algebraic equations. In both designs, $\epsilon = 1$ corresponds to a cell recycle with no biomass in the stream leaving the system in the filter and $\epsilon = 0$ corresponds to a pure substrate recycle. κ is taken as a design parameter. The cases of pure substrate and pure cell recycle will be investigated. The filter can be either of centrifugal type or a membrane type. The filter stream is employed to keep the volume in the fermenter constant. The nominal values of inlet flow rate and concentration at the two operating points are the same as those in *Design 1*. The operating points have the same conversion rate as *Design 1*. At steady state conditions, the filter stream is $D_{fil} = 0.5D_{in}$. In *Design 3* (Figure 5.1c) the product is withdrawn before the filter, i.e. $D_e = 0$, whereas in *Design 4* (Figure 5.1d) the product is withdrawn after the filter, i.e. $D_o = 0$.

5.1.2 Linearised Models

The nonlinear models are linearised at the two operating points and transformed to the Laplace domain to obtain the input-output transfer functions and disturbance-output transfer functions

$$y(s) = G(s)u(s) + \sum_i G_{d_i}(s)d_i(s) \tag{5.7}$$

where y is the controlled output, u is the control input and d is the vector of disturbances. $G(s)$ is the input-output transfer function and G_{d_i} is the disturbance transfer function corresponding to disturbance d_i. The output y is the cell mass concentration X. Disturbances are assumed to be due to uncertainties in the cell mass yield Y, maximum growth rate μ_{max}, inlet flow rate D_f and inlet substrate concentration S_f. The bounds on the inputs, disturbances and the allowed output deviation are given in Table 5.2. The input u, the output y and the disturbances d_i are scaled such that they are all in the range $[-1, 1]$.

5.1.3 Analysis

Based on the input-output and the disturbance-output transfer functions and the worst case disturbances, the controllability properties of the different designs can be analysed. The effects of the worst case disturbances and of the manipulated input

	Δu	Δd	Δy
Design 1	$\Delta D_{in} = 0.5 D_{in}$ $\Delta S_f = 0.25 S_f$	$\Delta D_{in} = 0.1 D_{in}$ $\Delta S_f = 0.1 S_f$ $\Delta \mu_{max} = 0.1 \mu_{max}$ $\Delta K = 0.1 K$ $\Delta Y = 0.1 Y$	$\Delta X = 0.1 X$
Design 2	$\Delta D_c = 0.25(D_f + D_c)$		
Design 3	$\Delta D_o = 0.25 D_o$ $\Delta D_{in} = 0.25 D_{in}$		
Design 4	$\Delta D_e = 0.25 D_e$		

Table 5.2: *Input, disturbance and output bounds for the different designs*

on the output are determined for each of the designs. The results are shown as frequency plots.

5.1.3.1 Design 1

The analysis is performed for two different control inputs. In *Case 1* the dilution rate is taken as the control input and in *Case 2* the feed concentration is employed as the control input to control the outlet cell concentration.

Figure 5.2 shows the frequency plots for the two control structures at the two operating points $OP1$ and $OP2$. Since perfect disturbance rejection at steady state conditions is desired, the gain of the transfer function at low frequencies has to be larger than the magnitude of the worst case disturbance.

Case 1: Using the dilution rate as control input (u=D)

At both operating points the effect of the dilution rate on the cell concentration is small. This corresponds to a small steady state gain which indicates controllability problems since the input is likely to saturate. In fact the worst case disturbance has

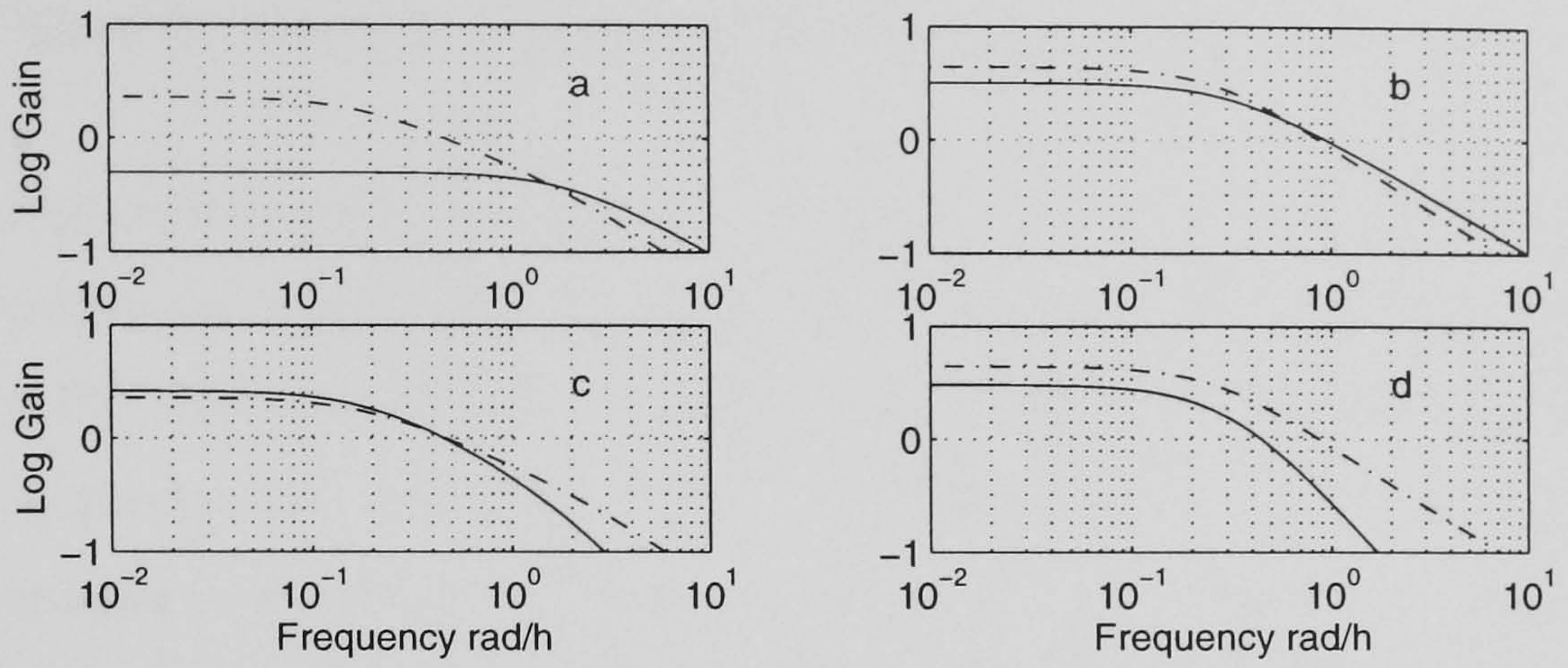

Figure 5.2: The input-output transfer function (full line) and the disturbance-output transfer function (dashed line) for *Design 1*
(a):$u = D$ at OP1; (b): $u = D$ at OP2
(c):$u = S_f$ at OP1; (d): $u = S_f$ at OP2

a larger effect on the output and one can therefore expect input saturation. Figure 5.2b shows that at operating point $OP2$, the steady state gain is larger than that at operating point $OP1$. However, a perfect rejection of the worst case disturbance is still not possible.

The results shown for the two operating points confirm other findings for these fermenters ([102]). For continuous fermenters with Monod kinetics, the steady state gain increases with increasing dilution rate. This indicates better controllability properties closer to the wash out.

The above analysis shows that *Design 1* is not controllable at either of the two operating points if the dilution rate is chosen to be the control input. This is due to a low steady state gain.

Case 2: Using feed substrate concentration as control input ($u = S_f$)

Figures 5.2c and 5.2d show the frequency plots for *Design 1* with the feed concentration as manipulated input. The system is not controllable at operating point $OP2$. The effect of the disturbances is too large and rejection of the disturbances at steady state conditions is not possible. However the effect of the disturbances is smaller at operating point $OP1$ where perfect rejection of the disturbances is possible at low frequencies. A good disturbance rejection can therefore be expected at this operating point if the feed concentration is chosen as the control input.

5.1.3.2 Design 2

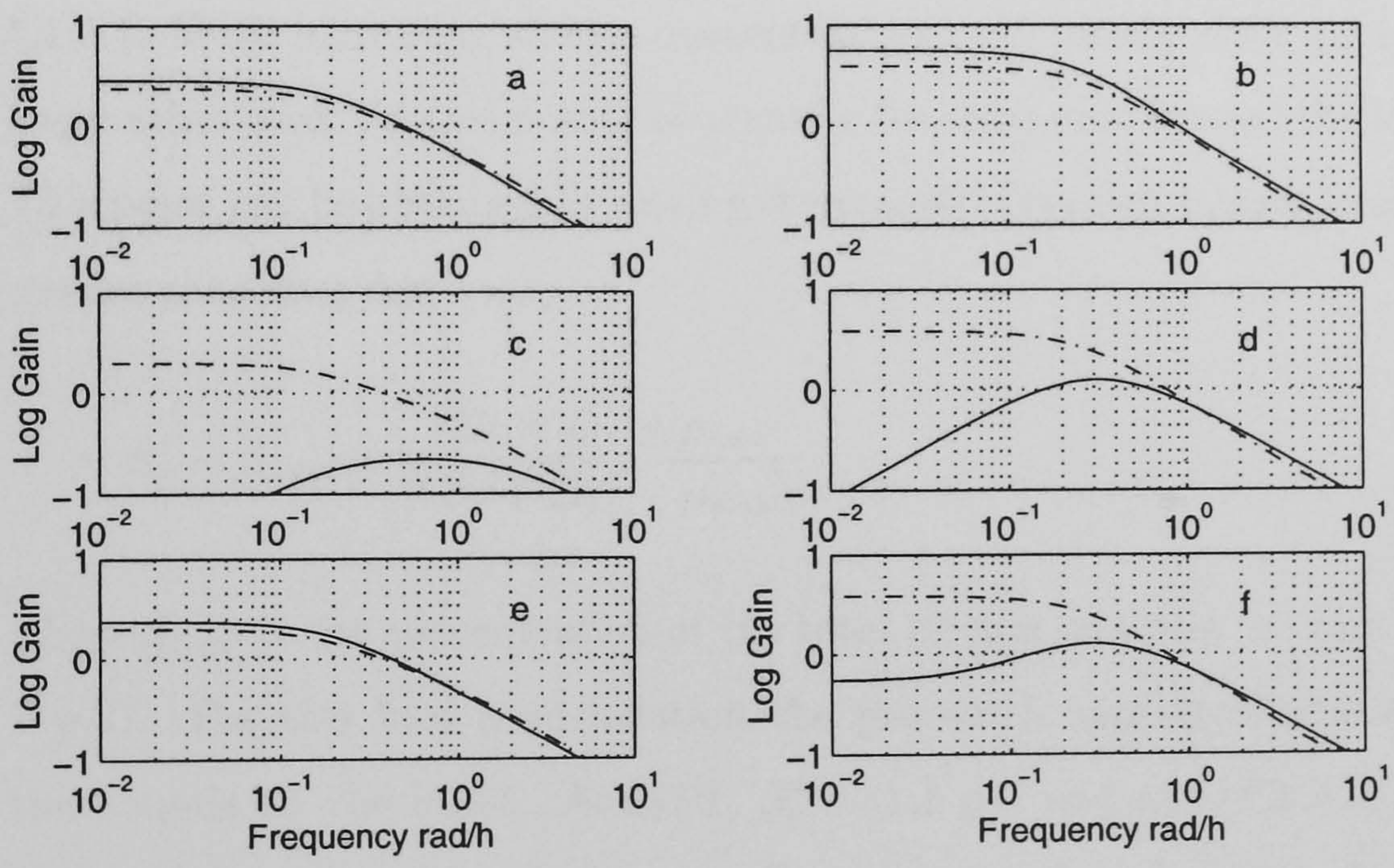

Figure 5.3: The input-output transfer function (full line) and disturbance-output transfer function (dashed line) for *Design 2* using D_c as the control input with different values of S_c.
(a): $S_c = 0$ at OP1, (b): $S_c = 0$ at OP2,
(c): $S_c = 1.1$ at OP1, (d): $S_c = 1.8$ at OP2,
(e): $S_c = 2$ at OP1, (f): $S_c = 2$ at OP2

In this design the dilution rate of the control stream is used as a control input ($u = D_c$). Figure 5.3 shows the frequency plots for different concentrations (S_c) of the control stream at the two operating points $OP1$ and $OP2$. For a control stream concentration of $S_c = 0$, the dilution rate D_c has a large effect on the output. This effect is larger than the effect of the disturbances at low frequencies at both operating points. Therefore the process is controllable at both operating points (Figure 5.3a and 5.3b). By increasing the concentration S_c, the gain of the input-output transfer function decreases. At intermediate values of S_c, the process is uncontrollable due to the very low transfer function gain. This is shown clearly in Figure 5.3c and 5.3d for operating points $OP1$ and $OP2$. The frequency plots corresponding to $S_c = 1$ are not shown here since this case is equivalent to *Design 1* with the dilution rate as a control input. This process is not controllable as was shown in Section 5.1.3.1. With increasing stream concentrations, the steady state gain decreases and approaches zero. At this point the transfer function zero crosses the imaginary axis. This point can be obtained by setting the transfer function to zero to give the control stream concentration

$$S_c^\star = S_{total} + \frac{(D_c + D_f)K\mu_{max}}{((D_c + D_f) - \mu_{max})^2} \tag{5.8}$$

where S_{total} is the concentration of the total stream entering the fermenter ($S_{total} = 1\ g/l$). For this feed concentration the process is uncontrollable independent of the bounds on the input. At $OP1$, $S_c^\star = 1.1\ g/l$ and at $OP2$ it is $S_c^\star = 1.8\ g/l$. The change in dilution from increasing the inlet flow and the change in growth from adding more substrate to the fermenter compensate each other at this feed concentration.

For stream concentrations higher than $S_c^\star$, a right-half plane zero appears in the transfer function. As S_c increases above $S_c^\star$ the zero moves away from the imaginary axis. The presence of a right half plane zero can have a significant effect on the control performance of the process. At the concentration where the zero crosses the imaginary axis the input has no effect on the output at steady state. The influence of the position of the transfer function zero is less pronounced the further away the zero is placed from the imaginary axis [59]. Figure 5.3e shows that for $S_c = 2$ the process is controllable at operating point $OP1$. At operating point $OP2$ the steady state gain is not large enough to make the process controllable.

5.1.3.3 Design 3

This design corresponds to a recycle fermenter with the product withdrawn before the filter. The outlet (D_o) and inlet (D_{in}) dilution rates are considered as control inputs.

Case 1: Using the outlet stream as control input ($u = D_o$)

Figures 5.4a and 5.4b show respectively the frequency plots for a cell recycle fermenter at the operating points OP1 and OP2. Perfect control is possible at all frequencies. Figures 5.4c to 5.4f show the frequency plots for the same fermenter at two different values of the splitting fraction κ. A low κ corresponds to a low recycle rate. For $\kappa = 0.5$ (Figure 5.4c and 5.4d) the system is not controllable at both operating points. The steady state gain increases with increasing κ and the system becomes controllable at $\kappa = 0.85$ (Figure 5.4e and 5.4f). This indicates that a cell recycle makes the fermenter more controllable. A substrate recycle makes the system controllable at high recycle rates.

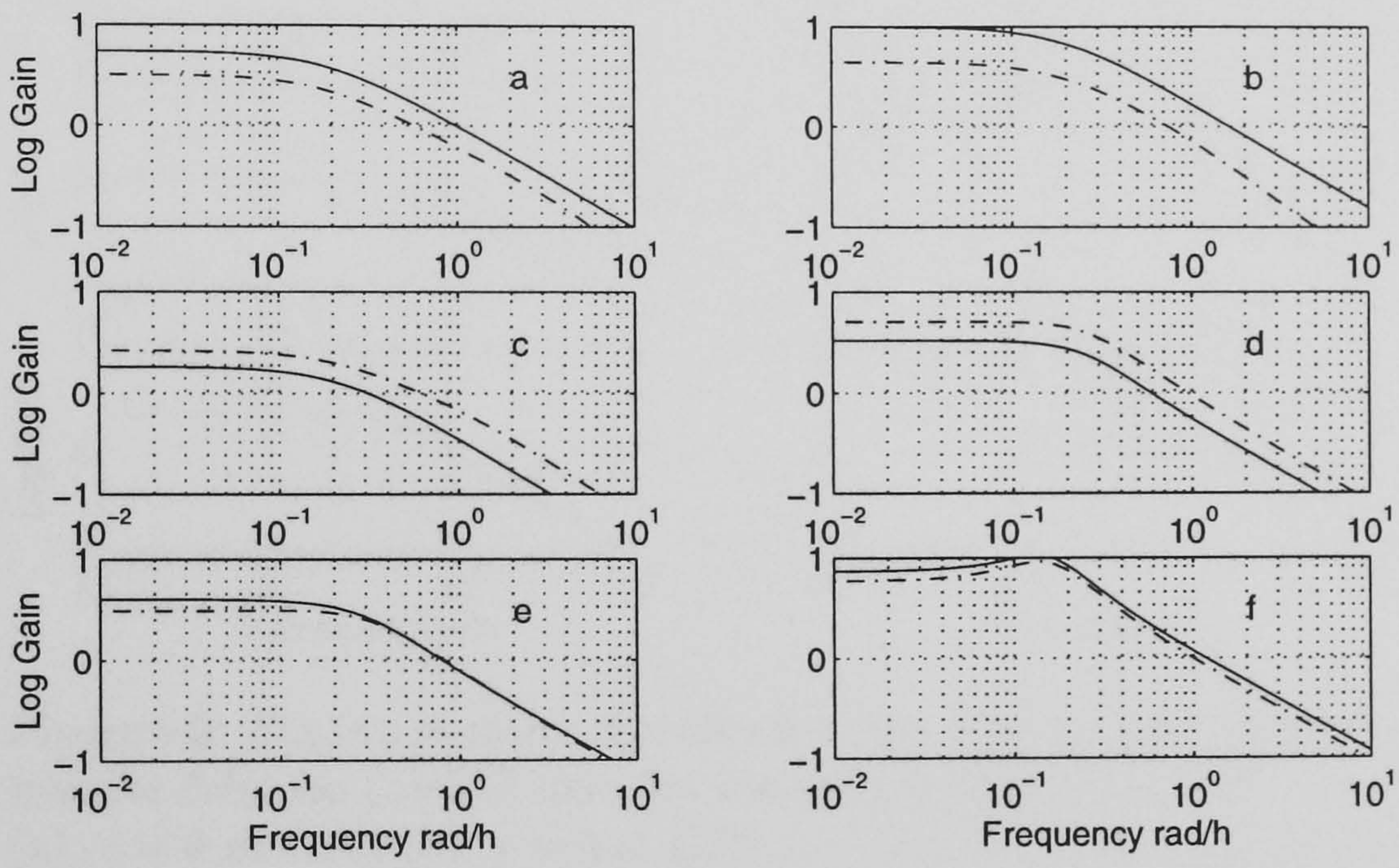

Figure 5.4: The input-output transfer function (full line) and the disturbance-output transfer function (dashed line) for *Design 3* with $u = D_o$
(a): $\epsilon = 1$ at OP1, (b): $\epsilon = 1$ at OP2,
(c): $\kappa = 0.5$ at OP1, (d): $\kappa = 0.5$ at OP2,
(e): $\kappa = 0.85$ at OP1, (f): $\kappa = 0.85$ at OP2

Case 2: Using the inlet flow rate as control input ($u = D_{in}$)

In this case, the input dilution rate D_{in} is used as the manipulated variable. The frequency plots are shown in Figure 5.5.

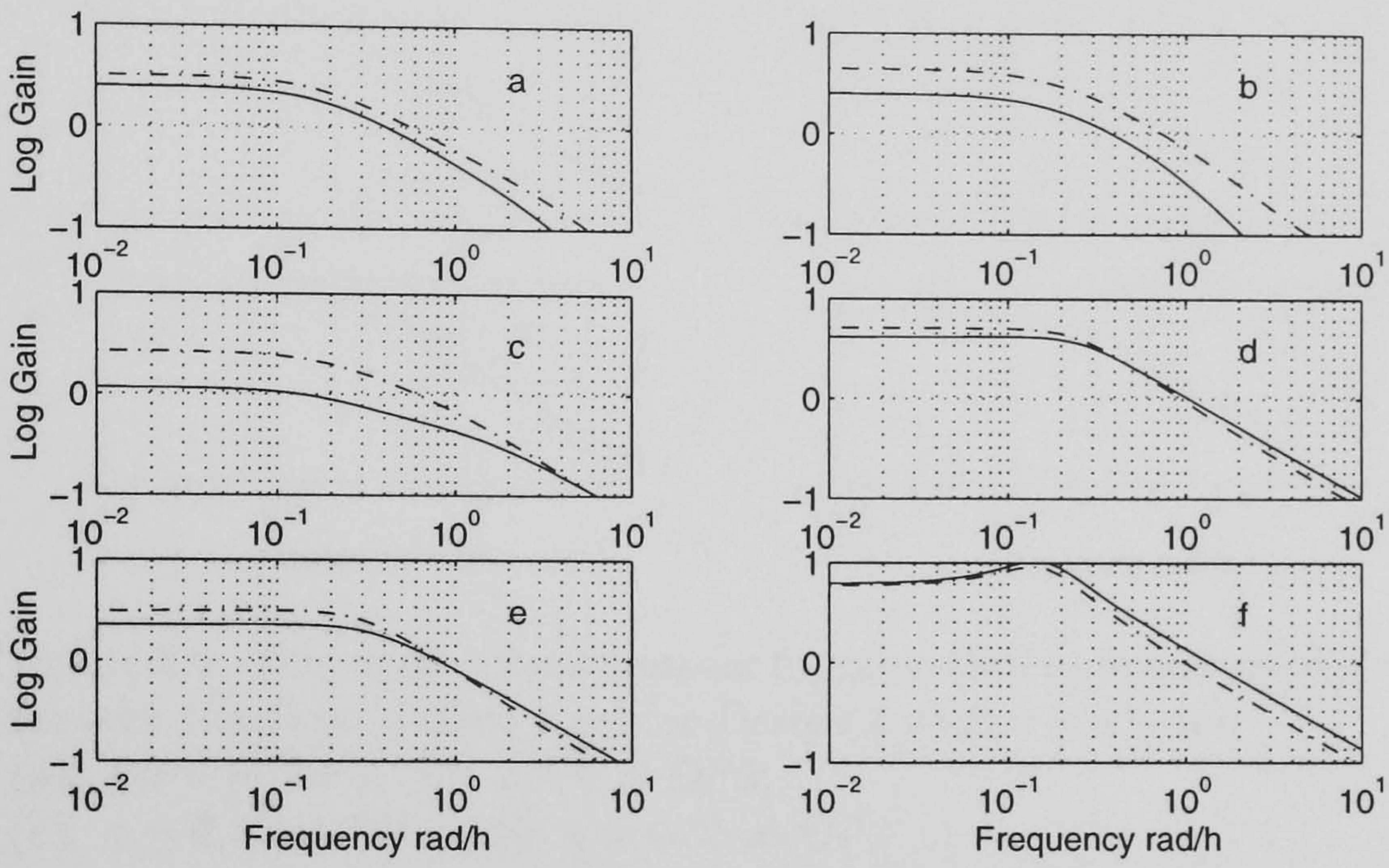

Figure 5.5: The input-output transfer function (full line) and the disturbance-output transfer function (dashed line) for *Design 3* with $u = D_{in}$ and
(a): $\epsilon = 0$ at OP1, (b): $\epsilon = 0$ at OP2,
(c): $\kappa = 0.5$ at OP1, (d): $\kappa = 0.5$ at OP2,
(e): $\kappa = 0.85$ at OP1, (f): $\kappa = 0.85$ at OP2

The effect of the input dilution rate on the cell concentration in the effluent is not large enough to guarantee disturbance rejection at steady state conditions independent of the operating point and the kind of recycle. The system becomes controllable only with a substrate recycle and high recycle rates (Figure 5.5f). All the other designs are uncontrollable (Figure 5.5a to Figure 5.5e).

5.1.3.4 Design 4

In this design the product is withdrawn after the filter and the output dilution rate of the filtered stream is taken as the control variable. Figures 5.6a and 5.6b show

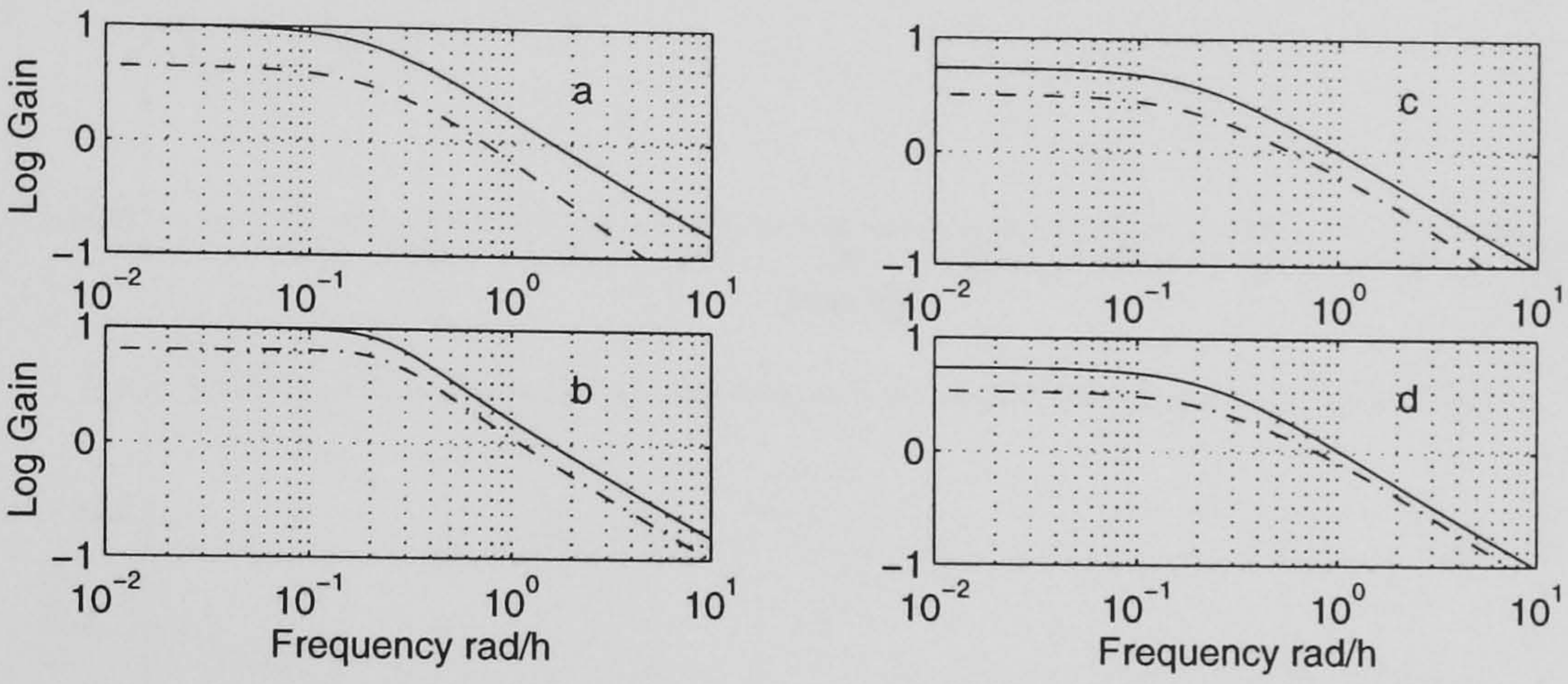

Figure 5.6: The input-output transfer function (full line) and the disturbance-output transfer function (dashed line) for *Design 4* with $u = D_e$ and
(a): $\epsilon = 0$ at OP1; (b): $\epsilon = 0$ at OP2;
(c): $\kappa = 0.85$ at OP1; (d): $\kappa = 0.85$ at OP2

the frequency plots with a cell recycle at the two operating points. The design is controllable in both cases. For a substrate recycle with intermediate recycle rates this design was found not to be controllable.

Figure 5.6c & 5.6d show the frequency plots corresponding to a high substrate recycle. At these operating points the design is controllable. In order to achieve the same conversion rate as *Design 1* and be controllable, the volume of this reactor has to be six times larger than that for *Design 1*.

5.1.4 Nonlinear Simulations

Simulations of the nonlinear models with simple PI controllers have been performed in order to confirm the results obtained from the controllability analysis. The simulations are for illustrative purposes only.

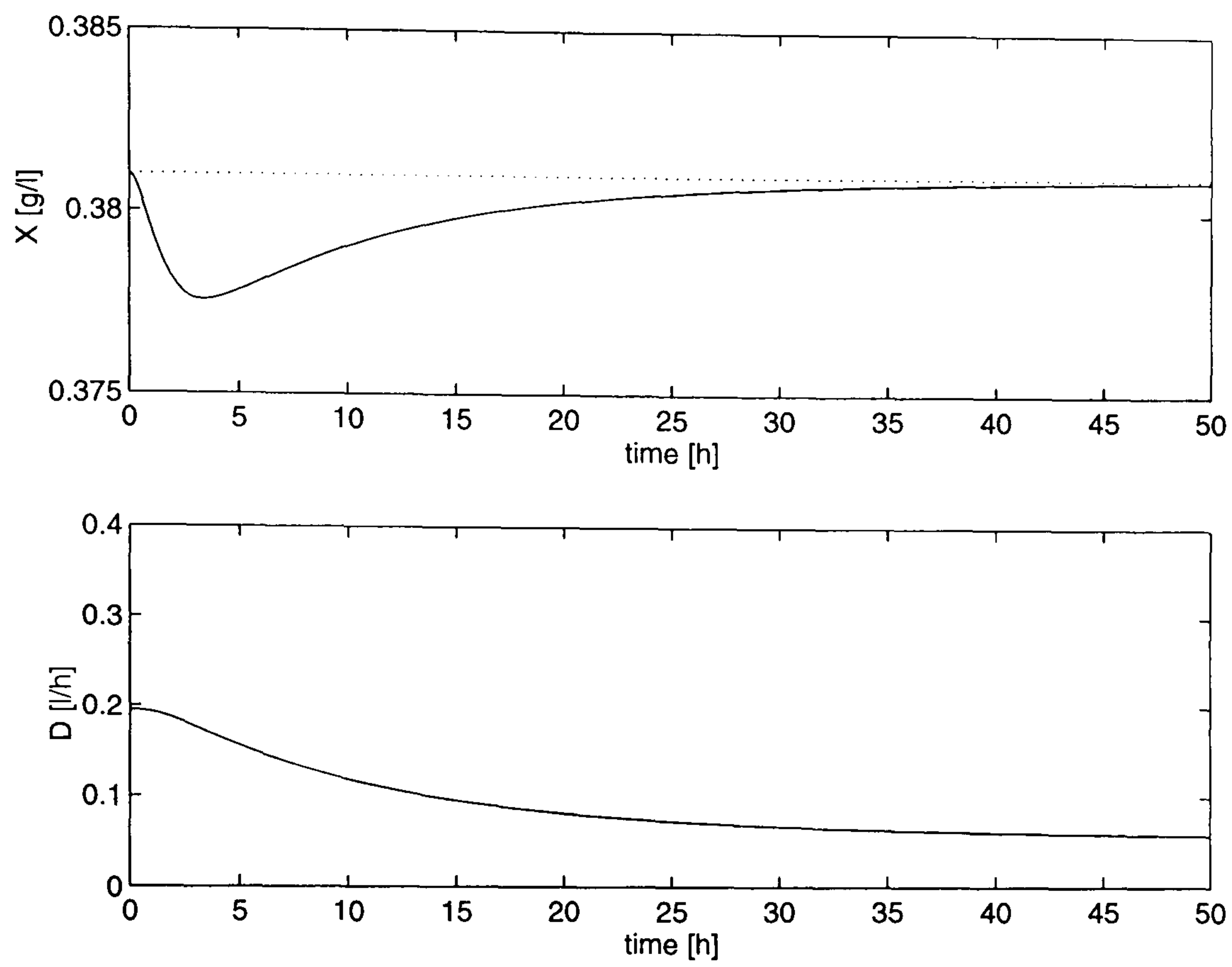

Figure 5.7: Nonlinear simulation of PI controlled fermenter based on *Design 1*. The dilution rate is the control input and a negative 4% step disturbance in Y is considered.

The above analysis showed that *Design 1* is not controllable if the dilution rate is chosen to be the control input. The bounds on the control input for this design were chosen to be twice as wide as those of the other designs. The poor controllability of this design is confirmed via a simulation of the controlled system (Figure 5.7). For a 4% step change in the yield factor Y, the control input exceeds its lower bound $(0.1 l/h)$. Note that this simulation corresponds to a single disturbance at half its maximum possible size. A small disturbance is chosen because it corresponds to a realisable input at steady state conditions. For larger disturbances, the control input saturates.

Figure 5.8 shows a simulation of a PI controlled fermenter based on *Design 2* with $S_c = 0$ subject to 10% step changes in all disturbances. The control system is able to reject these disturbances without exceeding the bounds on the control input. These findings correspond closely to the results obtained from the linearised process

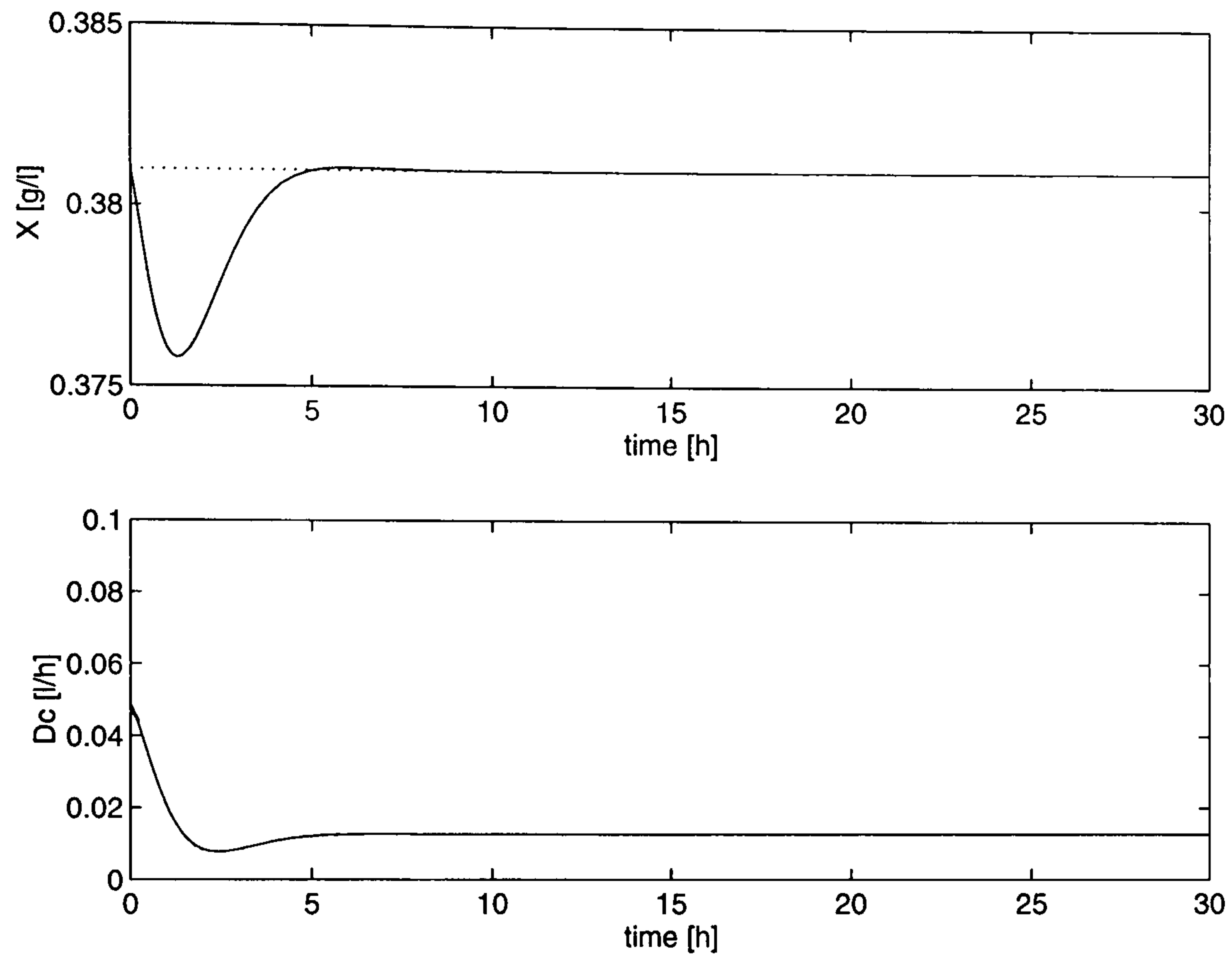

Figure 5.8: Nonlinear simulation of PI controlled fermenter based on *Design 2*. One stream is used as the control input and 10% step changes in all disturbances are considered.

models.

5.1.5 Conclusions

The above study shows that linear controllability analysis is a valuable tool in evaluating and comparing the controllability properties of different designs at an early stage of the design procedure.

A few common process designs of continuous fermenters are compared with respect to their controllability properties at two different operating points. The comparisons were made in terms of the control input required to reject the worst case disturbance for perfect control in the frequency domain. It can be seen that it is possible using frequency analysis to study the controllability properties of fermenters. Using the dilution rate as the control input to maintain the cell concentration at the desired level was found not to yield satisfactorily control performance. The control problems are related to input saturation. This was confirmed using a nonlinear simulation of the PI controlled process. Adding a second stream (which does not contain any growth limiting substrate) to the fermenter leads to superior control performance. This is predicted by linear analysis and confirmed with a nonlinear simulation. Recycles can have a positive effect on the controllability properties of the process. Cell recycles improve controllability if an outlet stream is used as the control input. For substrate recycles, high recycle rates are required to improve the controllability of the process. This, in general, is not desirable because it corresponds to larger reactor volumes.

5.2 Design of a Cell Producing Fed-Batch Fermentation

5.2.1 Nonlinear Dynamic Model

In this case study a cell producing fed batch fermenter as shown in Figure 5.9 is optimised. The nominal problem, i.e. without taking uncertainty into account, has been solved as an optimal control problem in [55]. The objective is to maximise the

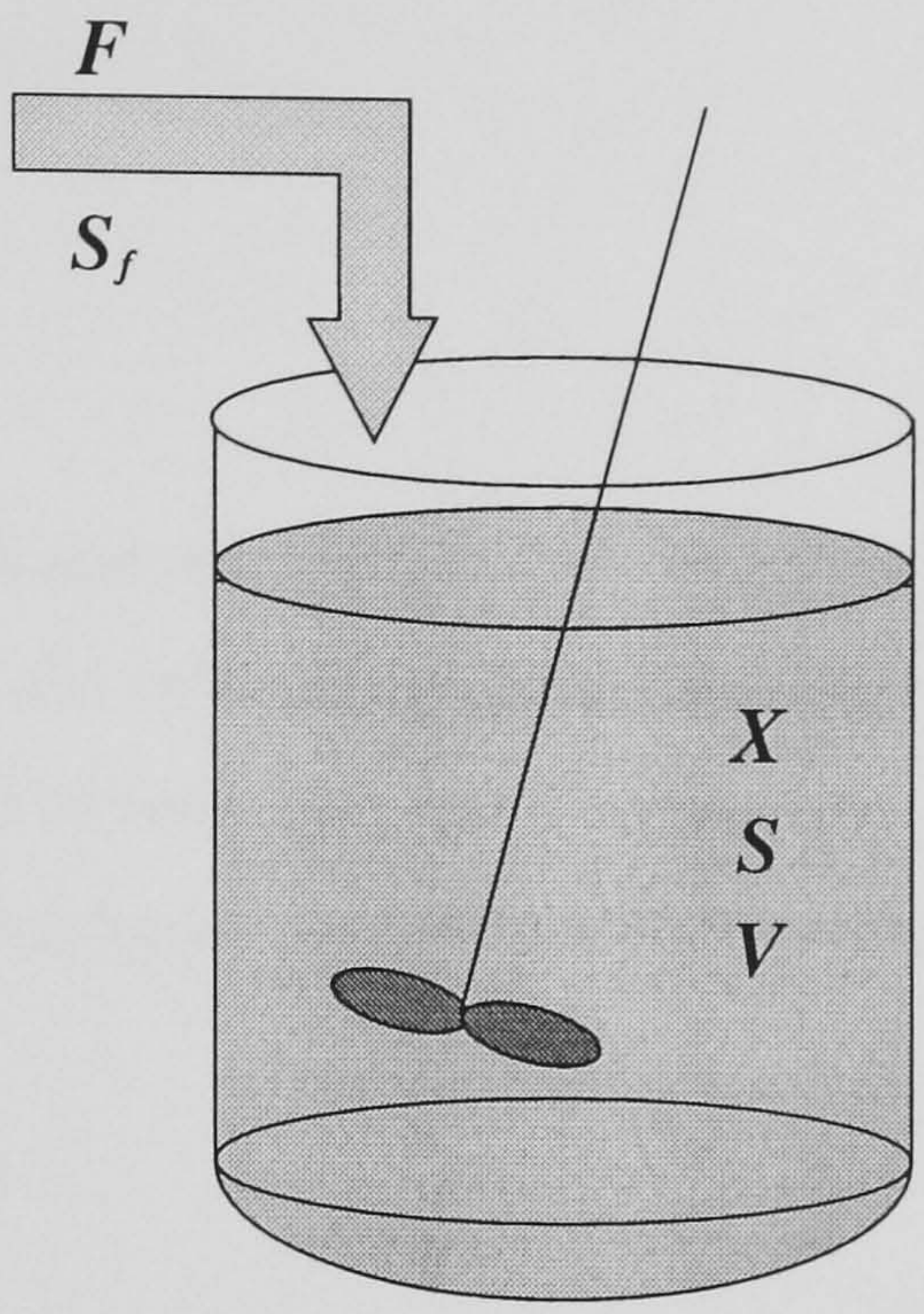

Figure 5.9: Cell producing fed batch fermenter

amount of product at the end of the fermentation

$$\Phi = X(t_f)V(t_f) \tag{5.9}$$

where X is the cell concentration, V the volume of the fermenter and t_f the final time of the fermentation is fixed to be $t_f = 3.8\ h$. The volume of the fermenter is assumed to be bounded by

$$V(t) \leq 5\ l \tag{5.10}$$

It is also assumed that the fermenter is fed by a single stream, its contents are well mixed and the feed is sterile. The unstructured model of this fermentation is given by

$$\dot{X} = \mu(S)X - \frac{X}{V}F$$

$$\dot{S} = -\frac{\mu(S)}{Y}X + \frac{(S_f - S)}{V}F \tag{5.11}$$

$$\dot{V} = F \tag{5.12}$$

where X and S are the cell and substrate concentrations, respectively. Y is the yield coefficient ($Y = 0.5$) and the influent substrate concentration S_f ($S_f = 10\ g/l$) is assumed to be constant. The feed flow rate F is assumed to be time varying and is considered to be the control input. Bounds on the flow rate are

$$0\ l/h \leq F(t) \leq 4\ l/h \tag{5.13}$$

The control input is parameterised using a piecewise constant function. It has been found through computer simulations that not much improvement is obtained using a more complex parameterisation. The growth rate is substrate inhibited and modeled by

$$\mu(S) = \mu_{max}\frac{S}{K + S + 0.5S^2} \tag{5.14}$$

The maximum growth rate $\mu_{max} = 1\ h^{-1}$ and the Monod constant $K = 0.03\ g/l$ are assumed to be uncertain by $\pm 10\%$.

5.2.2 Nominal Optimisation

Initially the model is optimised using the nominal parameters. This gives a total amount of cell mass $\Phi = 20.95\ g$ and a final product concentration of $X(t_f) = 4.19\ g/l$. This result serves as an upper bound on the objective function and can be used later to measure the difference between the nominal and the robust optimum

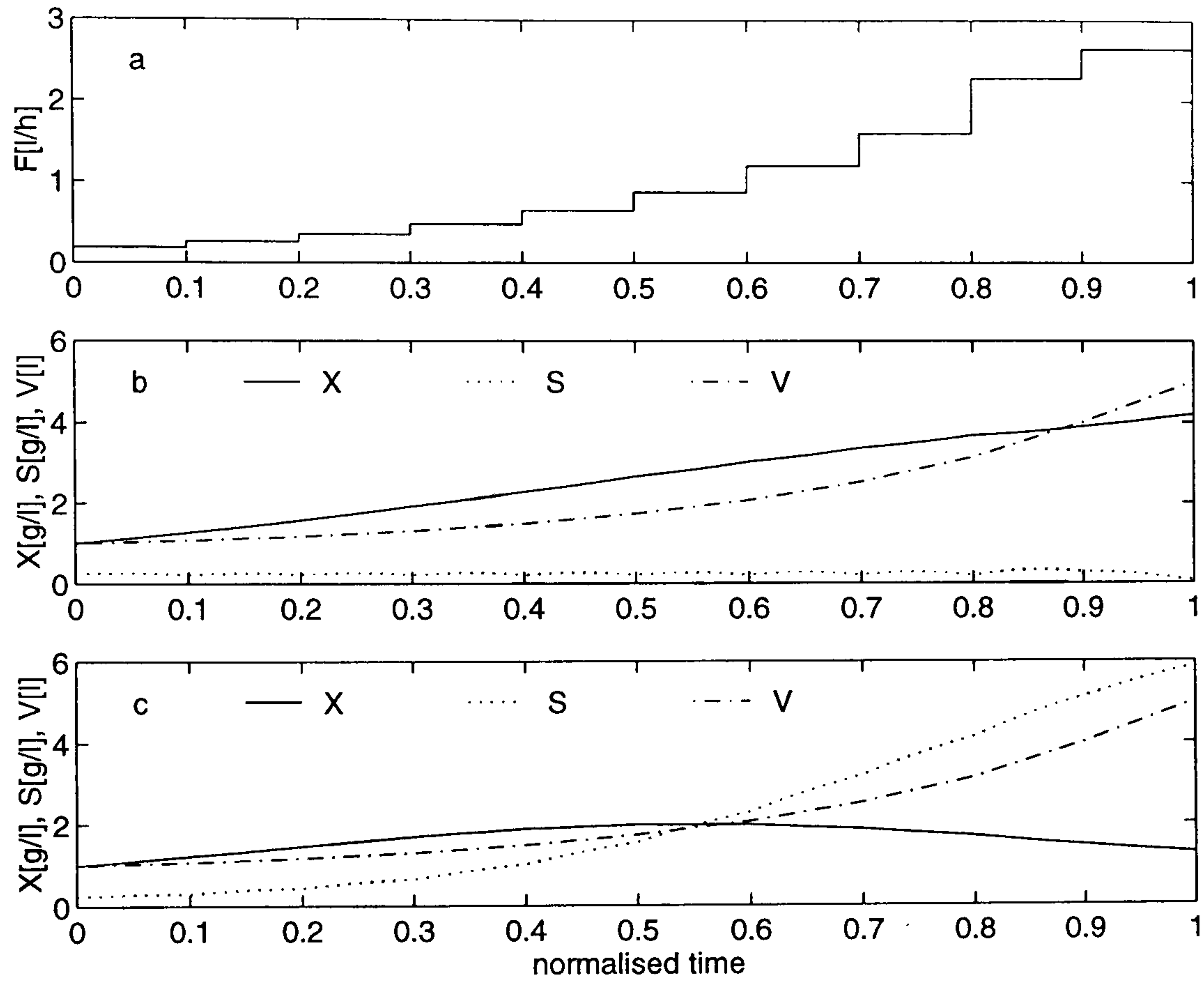

Figure 5.10: Optimal nominal solution: $X(t_f)V(t_f) = 20.95\ g$
(a):Feed rate input profile,
(b):State trajectories corresponding to the nominal parameters,
(c):State trajectories corresponding to values of $\mu_{max} = 0.9\ h^{-1}$ (90% of the nominal value) and $K = 0.033\ g/l$ (110% of the nominal value)

when uncertainty is taken into account. The optimal feed rate profile for the nominal problem is shown in Figure 5.10a. The corresponding state trajectories for the nominal parameter values are shown in Figure 5.10b. The optimal controller maintains the substrate concentration at a constant level which maximises the growth rate. The small oscillations are due to the staircase character of the control. In Figure 5.10c the effect of the parameter uncertainties is shown. The 10% change in the parameter values strongly influences the state profiles if the optimal feed rate based on the nominal parameters is applied. The final cell concentration ($X(t_f) = 1.28\ g/l$) is much less than in the nominal case. The total amount of product is only about 1/3 of the value predicted by the nominal optimisation.

5.2.3 Robust Optimisation

In order to avoid such failures and guarantee final product concentrations within certain bounds, an additional uncertainty constraint is introduced. This constraint restricts the final cell mass concentration to be within a certain neighbourhood of the nominal case for all parameter variations. This is expressed with the following set of constraints

$$\left.\begin{aligned} &\dot{x}_p = f(x_p, u, p) \\ &X_p(t_f) \le X(t_f) + 0.5\ g/l \\ &X_p(t_f) \ge X(t_f) - 0.5\ g/l \end{aligned}\right\} \forall p \in P \tag{5.15}$$

These constraints ensure that the final cell concentration will be within $\pm 0.5\ g/l$ bounds of the nominal solution at the final time. The robust optimal feed rate profile is shown in Figure 5.11a. Figure 5.11b shows the solution for the nominal

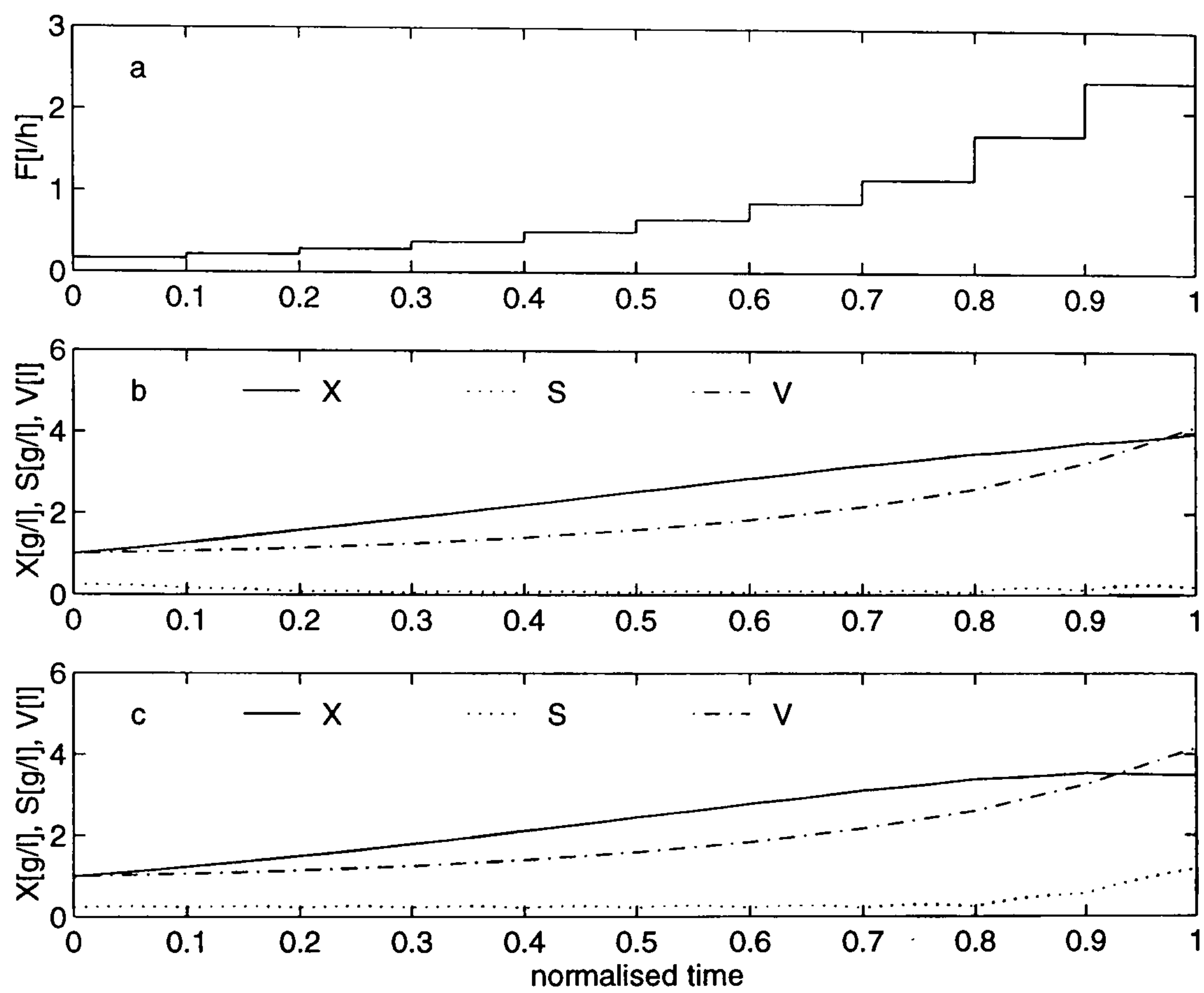

Figure 5.11: Robust optimal solution with uncertainty constraints Eq. 5.15: $X(t_f)V(t_f) = 16.51\ g$

(a):Feed rate input profile,

(b):State trajectories corresponding to the nominal parameters,

(c):State trajectories corresponding to values of $\mu_{max} = 0.9\ 1/h$ and $K = 0.033\ g/l$

parameter case. The trajectories corresponding to the parameter combination which maximizes the uncertainty constraint is shown in Figure 5.11c. It is shown that the effect of the uncertainty is much less than that corresponding to the solution without the uncertainty constraint. This is achieved at the expense of reducing the objective function of the nominal case to 16.51 g. As pointed out before, this is the price to pay to incorporate robustness.

5.2.4 Robust Optimisation with Feedback

In order to reduce the effect of the uncertainty and allow some degree of on-line adjustment, a linear constant gain state feedback controller is implemented on the model. The elements of the gain matrix are included in the optimisation problem as additional parameters. The control law is given by

$$\hat{F}(t) = F(t)_{nom} + k_1 \Delta X(t) + k_2 \Delta S(t) \tag{5.16}$$

where F_{nom} is the feed rate profile corresponding to the normal optimisation problem and ΔX and ΔS are the deviation from the corresponding nominal state trajectories. Since the input constraints in Eq. 5.13 ensure that the controller parameters are chosen such that input saturation is prohibited, a smooth saturation function as shown in Figure 5.13 was added to the problem

$$F(t) = \begin{cases} 4 & \text{if } \hat{F}(t) > 4 + \epsilon \\ 4 - \frac{(\hat{F}(t) - 4 - \epsilon)^2}{4\epsilon} & \text{if } 4 - \epsilon < \hat{F}(t) \leq 4 + \epsilon \\ \hat{F}(t) & \text{if } 0 \leq \hat{F}(t) \leq 4 \\ \frac{(-\hat{F}(t) - \epsilon)^2}{4\epsilon} & \text{if } -\epsilon \leq \hat{F}(t) < \epsilon \\ 0 & \text{if } \hat{F} < -\epsilon \end{cases} \tag{5.17}$$

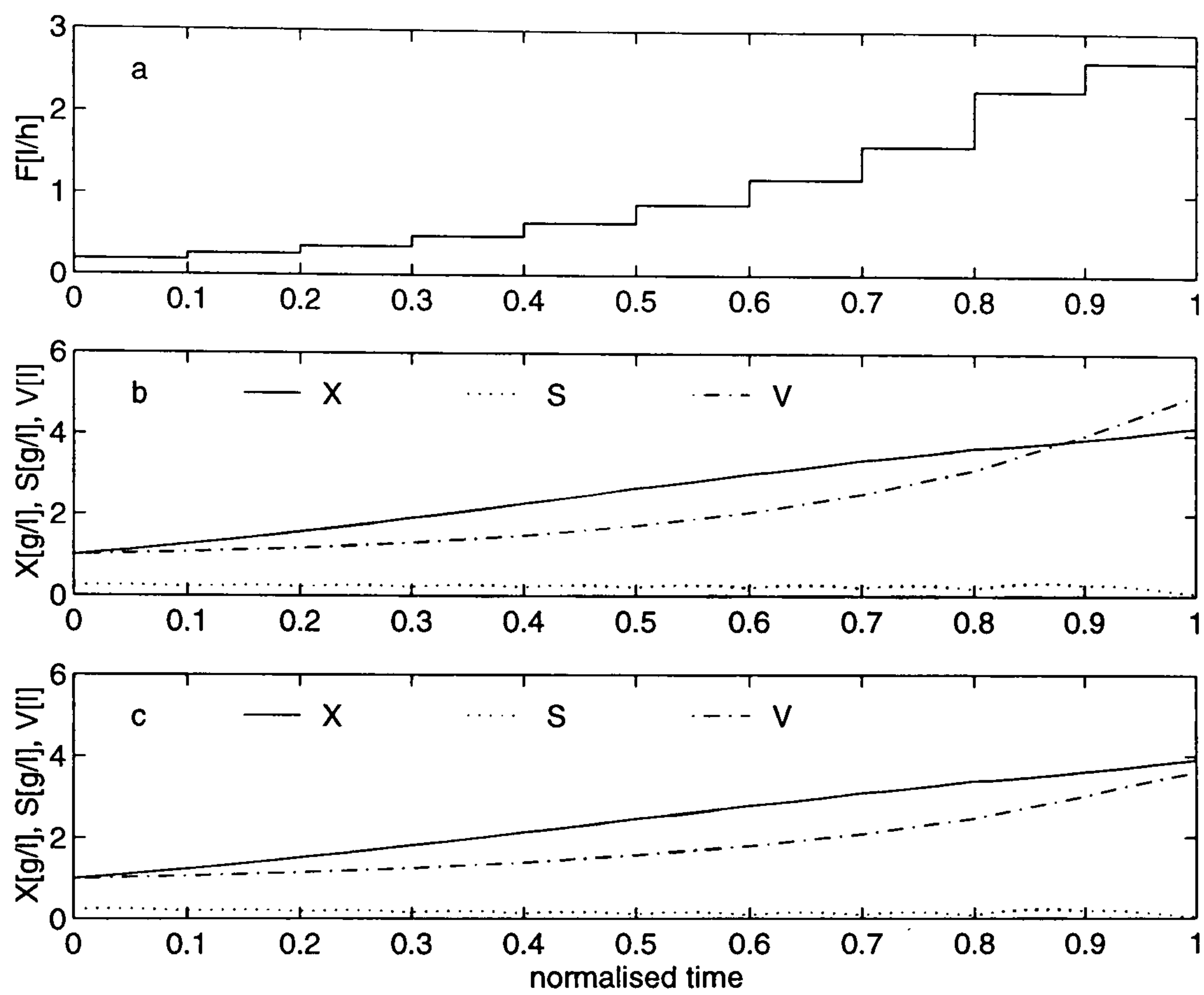

Figure 5.12: Optimal solution with uncertainty constraints Eq. 5.15 and feedback controller: $X(t_f)V(t_f) = 20.95\ g$
(a):Feed rate input profile,
(b):State trajectories for the nominal parameters (same as Fig. 5.10b),
(c):State trajectories for $\mu_{max} = 0.9\ 1/h$ and $K = 0.033\ g/l$

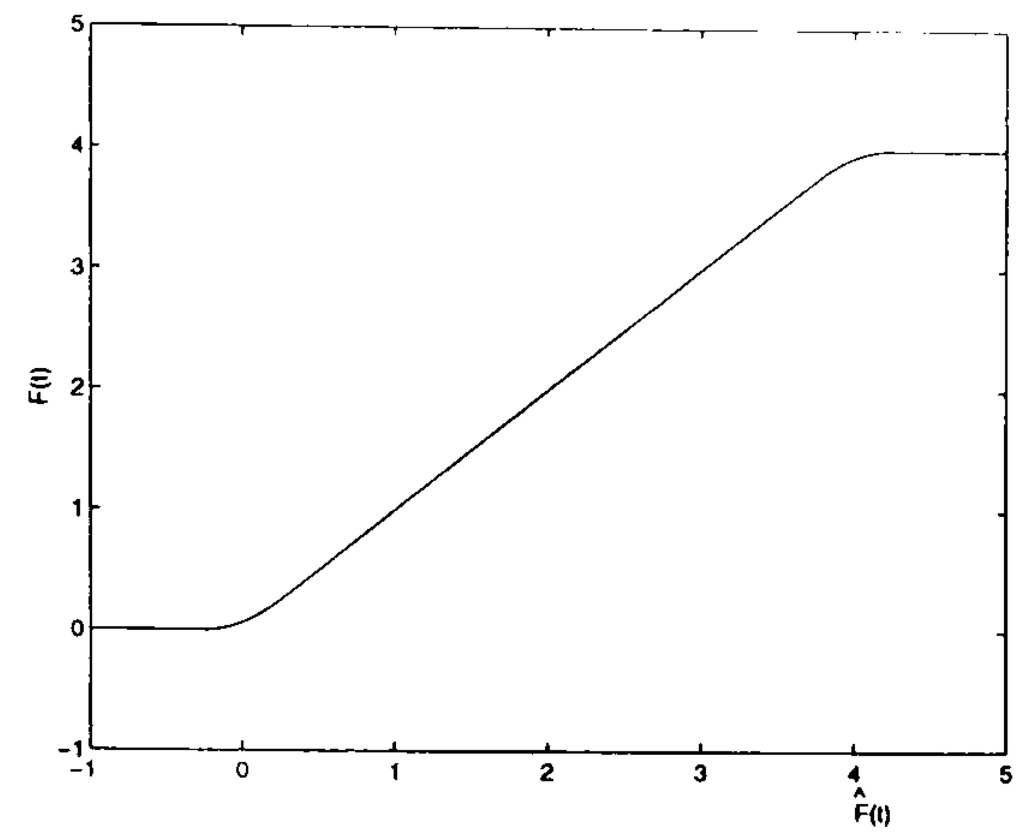

Figure 5.13: The smoothed saturation function Eq. 5.17

where ϵ is a small positive scalar. The smoothing is necessary in order to avoid non differentiability in the optimisation problem. This saturation function makes it possible for the input to saturate during operation which would not have been possible if, instead, constraint Eq. 5.13 was modified to bound $\hat{F}(t)$. Equation 5.13 is used to bound the flow rate $F(t)_{nom}$. The optimal input profile is shown in Figure 5.12a. It is identical to the input profile of the problem without any uncertainty constraints. The value of the objective function is $\Phi = 20.95\ g$ which is equal to the case without uncertainty constraints. The state trajectories corresponding to the nominal parameters case are shown in Figure 5.12b. For the worst case uncertainty the state trajectories are shown in Figure 5.12c. The uncertainty constraint is not active at the solution which implies that the final bounds on the cell concentration will be less than those required by the constraint. The cell concentration trajectories are quite close together. The controller shifted the effect of the uncertainty on the cell mass concentration to that on the volume.

5.2.5 Conclusions

The above study has shown that improvements can be made if model uncertainties are taken into account when optimising fed batch fermentation. The optimisation of the nominal model is compared to the optimisation taking parameter uncertainty into account. It is shown that a feedback controller based on process measurements or estimates can be used to reduce the effect of the uncertainty since it provides a degree of on-line adjustment. This approach has been shown to produce profiles which are more robust in the presence of uncertainty.

5.3 Design of a Competitive Fed-Batch Fermentation

5.3.1 Nonlinear Dynamic Model

In this case study the production of a metabolite produced by cells containing a recombinant plasmid is investigated. A major problem in the use of plasmids as recombinant vectors is the plasmid free cell generation and subsequent growth [79]. This phenomenon results in the loss of production of gene products encoded on the plasmid due to a wash out of the plasmid bearing cells in a continuous culture. Plasmid free cells are generated by plasmid loss due to improper segregation at cell division of the plasmid carrying cells or by growth. The plasmid carrying cells are competing for the same substrate with the plasmid free cells. The plasmid carrying cells excrete a metabolite into the media which is consumed by the plasmid free cells. These interactions are demonstrated in Fig. 5.14

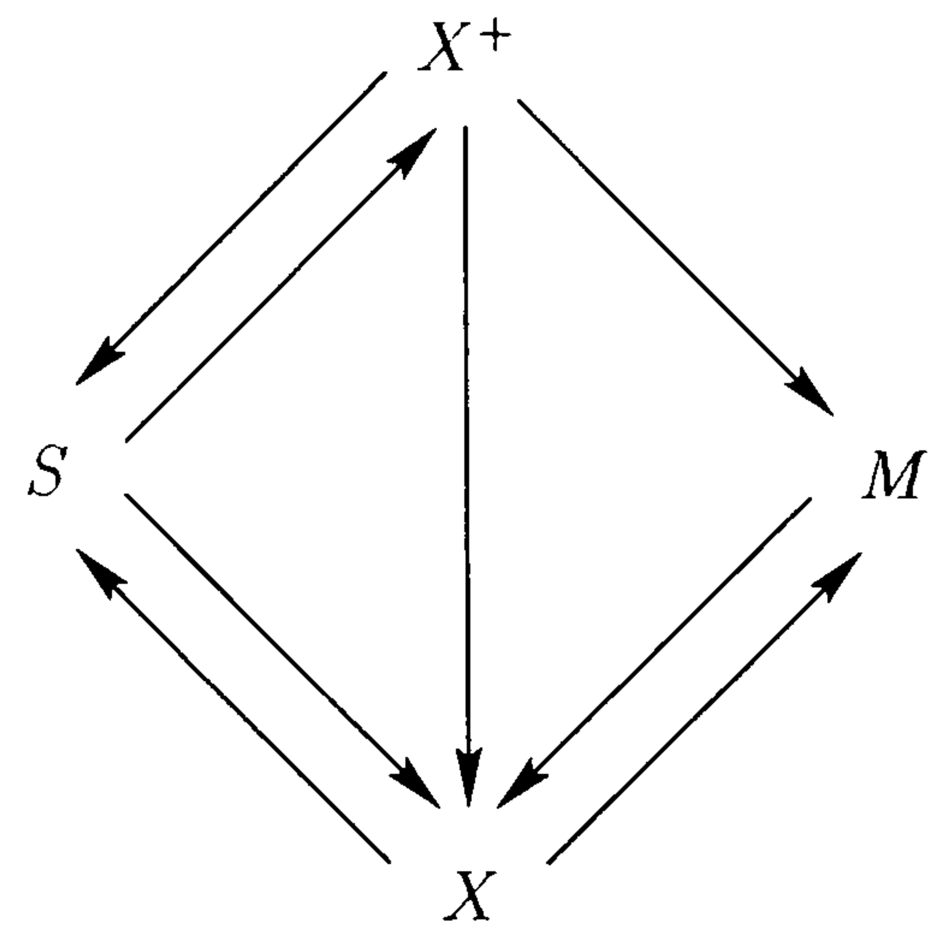

Figure 5.14: The interactions between the different components in the fermenter

In [79] an unstructured dynamic model for such a process is developed. It is demonstrated that the model predictions compare well with continuous and batch experiments. This model was modified here in order to operate in fed-batch mode. The model equations are

- The mass balance for the plasmid carrying cells

$$\dot{X}^{+} = (1-r)\mu^{+}(S)\, X^{+} - \frac{X^{+}}{V}F \tag{5.18}$$

- The mass balance for the plasmid free cells

$$\dot{X} = r\mu^{+}(S)\, X + \mu(S,M)\, X - \frac{X}{V}F \tag{5.19}$$

- The substrate mass balance

$$\dot{S} = -\frac{\mu^{+}(S)}{Y_S}X^{+} - \frac{\mu(S,M)}{Y_S}X + \frac{S_f - S}{V}F \tag{5.20}$$

- The metabolite mass balance

$$\dot{M} = -\frac{\mu(S,M)}{Y_M}X + k\mu^{+}(S)\, X^{+} - \frac{M}{V}F \tag{5.21}$$

- The overall mass balance

$$\dot{V} = F \tag{5.22}$$

where the growth rate of the plasmid carrying cells is modeled by a Monod kinetic

$$\mu^+(S) = \mu_{max}\frac{S}{K_s + S} \tag{5.23}$$

It is assumed that a constant fraction, r, of these cells looses the plasmid. The growth rate of the plasmid free cells is modeled as a dual Monod kinetic. Both the substrate, S, and the metabolite, M, are growth limiting substrate

$$\mu(S, M) = \mu_{max}\frac{S}{K_S + S}\frac{M}{K_M + M} \tag{5.24}$$

The substrate yield, Y_S, and metabolite yield , Y_M, are assumed to be constant over the operating range and Y_S is assumed to be the same for both cell types.

The optimisation objective is to maximise the amount of metabolite at the end of the fermentation time.

$$\Phi = M(t_f)V(t_f) \tag{5.25}$$

In order to avoid undesired side products which start building at high substrate concentrations the substrate concentration in the fermenter should not exceed $S = 6\ g/l$. This leads to the following path constraint for the optimisation

$$S(t) \leq 6\ g/l \quad \forall\, t\ \in\ [0, t_f] \tag{5.26}$$

r	0.14
μ_{max}	0.43 $1/h$
Y_S	0.13 g/g
Y_M	0.03 g/g
k	13 g/g
K_s	1.1 g/l
K_m	0.21 g/l
S_f	6.8 g/l

Table 5.3: The nominal model parameters

The volume of the fermenter is limited to $V_{max} = 5\ l$. The corresponding path constraint is

$$V(t) \leq 5\ l \quad \forall t \in [0, t_f] \tag{5.27}$$

The flow rate, F, of the feed is considered to be the control variable. Constraints on the flow rate are

$$0\ l/h \leq F(t) \leq 10\ l/h \tag{5.28}$$

The fermentation time is free within a four hours time horizon

$$0\ h \leq t_f \leq 4\ h \tag{5.29}$$

The nominal parameter values used in the model are given in Table 5.3. The upper and lower bounds of the uncertain parameters are shown in Table 5.4. The time horizon is divided into seven intervals, which gave good results in computational experiments. The control variable is parameterised to be constant in each of these intervals. A multiple shooting discretisation is used to discretise the ODE model.

p	$\underline{p}$	$\overline{p}$
r	0.1	0.2
μ_{max}	0.4 $1/h$	0.5 $1/h$
K_s	0.9 g/l	1.2 g/l

Table 5.4: The bounds on the uncertain parameters

The ODEs are solved by a third-order ERK in each interval and the gradients are obtained by IND. The path constraints are discretised on the same time grid. This approach can lead to violations of the original path constraints but the results were satisfactory.

5.3.2 Optimisation of the Nominal Model

Initially the nominal model is optimised. The nominal optimal input profile is shown in Fig. 5.15 and the corresponding nominal state profiles are given in Fig. 5.16. In the first time interval the feed-rate drives the substrate concentration to reach the substrate path constraint boundary at the end of the interval. In the subsequent time intervals, the flow-rate forces the substrate concentration to stay (approximately, due to the constraint discretisation and control parameterisation) at the substrate path constraint boundary until the volume path constraint boundary is reached. After this period no feed is added to the fermenter anymore. At the end of the fermentation, $t_f = 4\ h$, the total amount of metabolite is $\Phi = 17.44\ g$. For parameters other than the nominal ones the state profiles look different. The state profiles corresponding to the worst case scenario with respect to violating the substrate path constraint, is shown in Fig. 5.17. Here the substrate path constraint is violated.

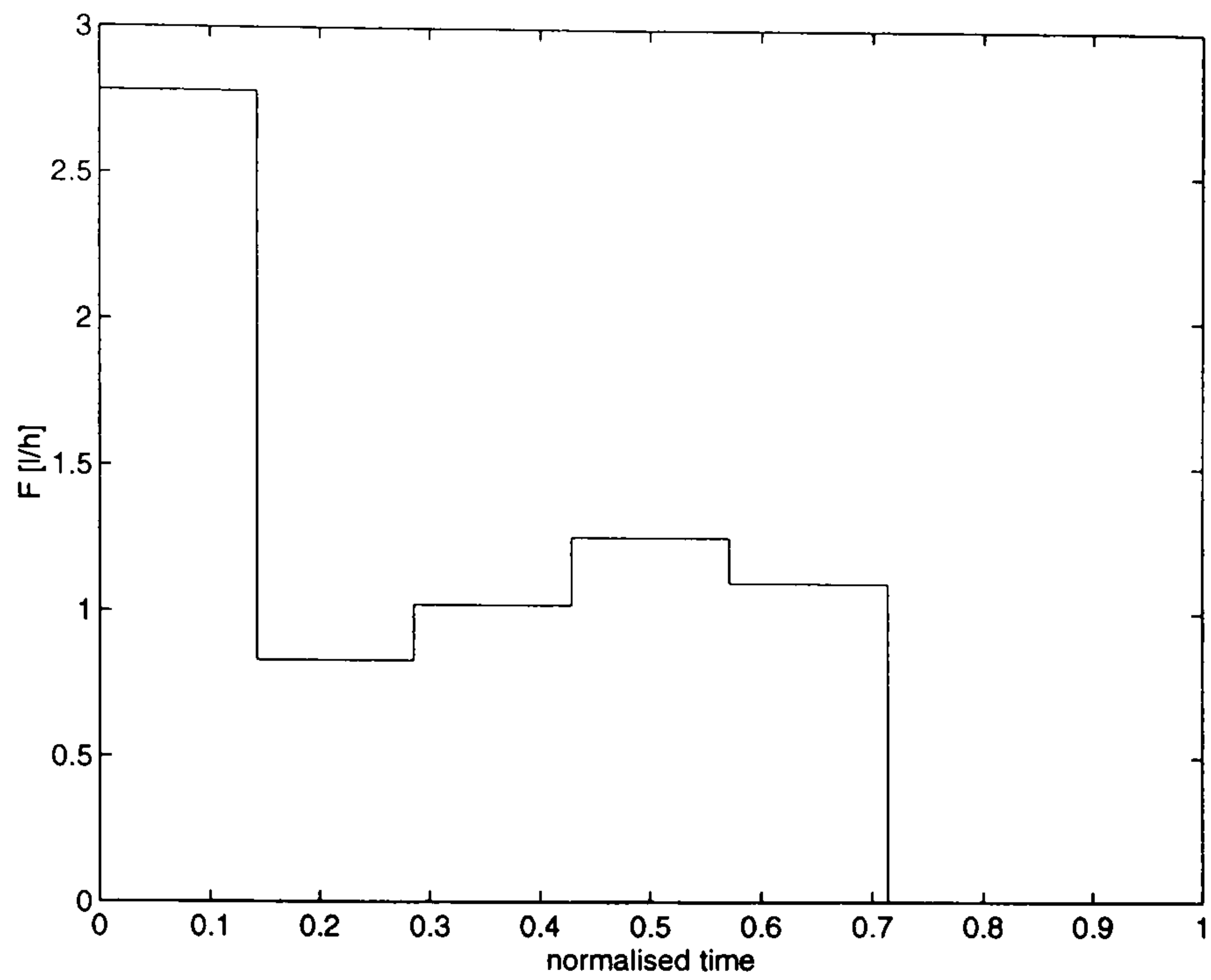

Figure 5.15: Optimal nominal input profile

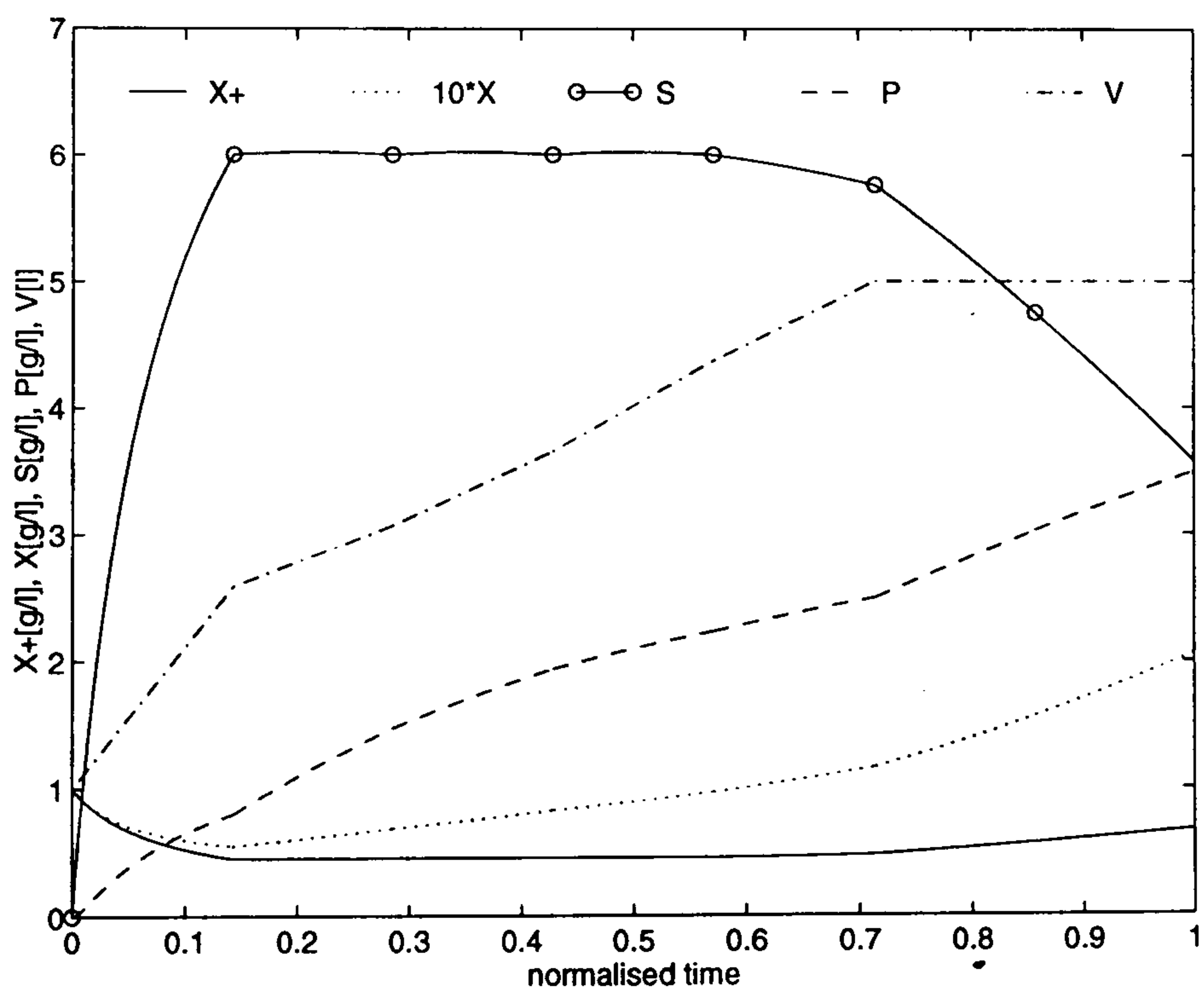

Figure 5.16: Optimal state profile

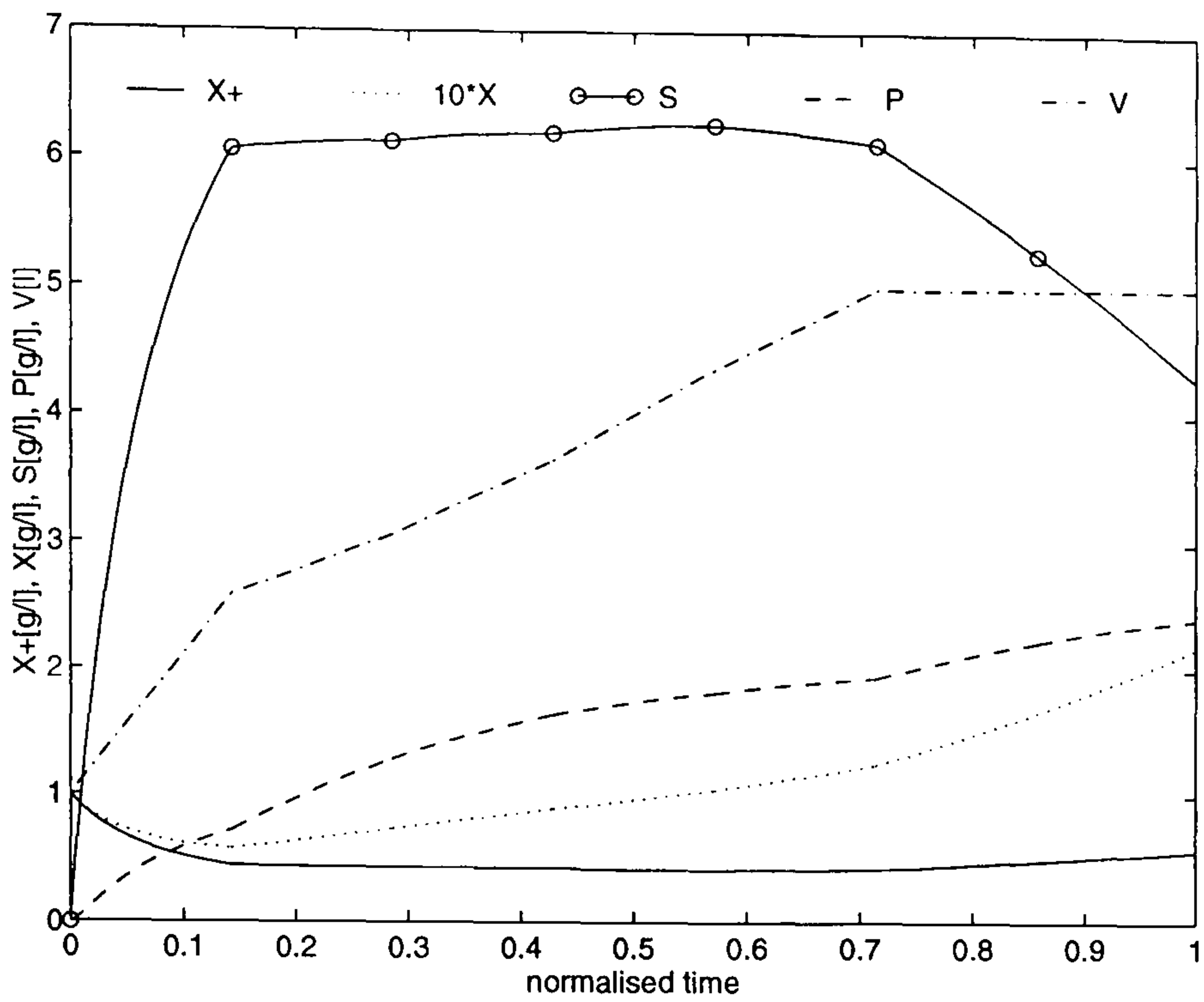

Figure 5.17: Worst case state profile

5.3.3 Robust optimisation

In order to avoid the violation of the path constraints for model parameters other than the nominal ones, the path constraints are included as uncertainty constraints in the optimisation problem. These constraints have to be satisfied for all possible parameter variations. The results from an open-loop robust optimisation run are shown here. The optimal robust input profile is shown in Fig. 5.18 and the corresponding state profile for the nominal parameter values is shown in Fig. 5.19. At the end of the fermentation, $t_f = 4\ h$, the total amount of metabolite is $M(t_f)V(t_f) = 17.32\ g$. The state profile corresponding to the worst case scenario with respect to violating the substrate path constraint is shown in Fig. 5.20. The path constraint is not violated in the worst case scenario.

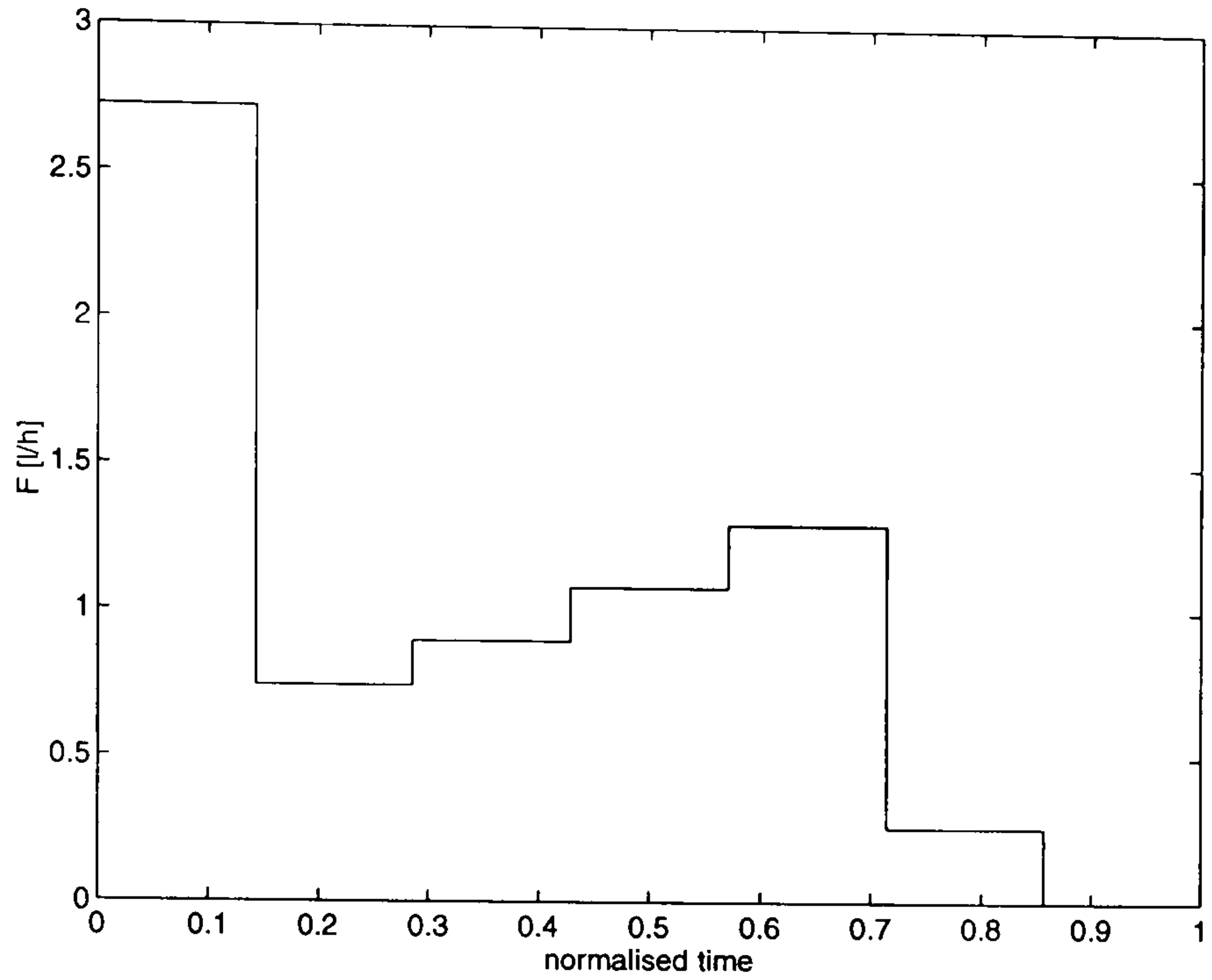

Figure 5.18: Robust input profile

5.3.4 Multi-Objective Optimisation

In this section two objective functions, $\Phi_1(t_f)$ and an objective corresponding to a worst case scenario $\max_{p \in P} \Phi_2(t_f)$, are traded off against each other. The first objective function is the nominal total amount of metabolite given by Eq. 5.25. The purpose of the second objective function is to avoid too low metabolite concentrations due to parameter uncertainty. The final metabolite concentration corresponding to the worst case scenario should be as close as possible to the nominal final metabolite concentration. This means that the difference between the nominal metabolite concentration,$M(t_f)$, and the worst case metabolite concentration, $\min_{p \in P} M_p(t_f)$, should be minimised. This difference can be expressed by

$$\Phi_2(t_f) = M(t_f) - \min_{p \in P} M_p(t_f) \tag{5.30}$$

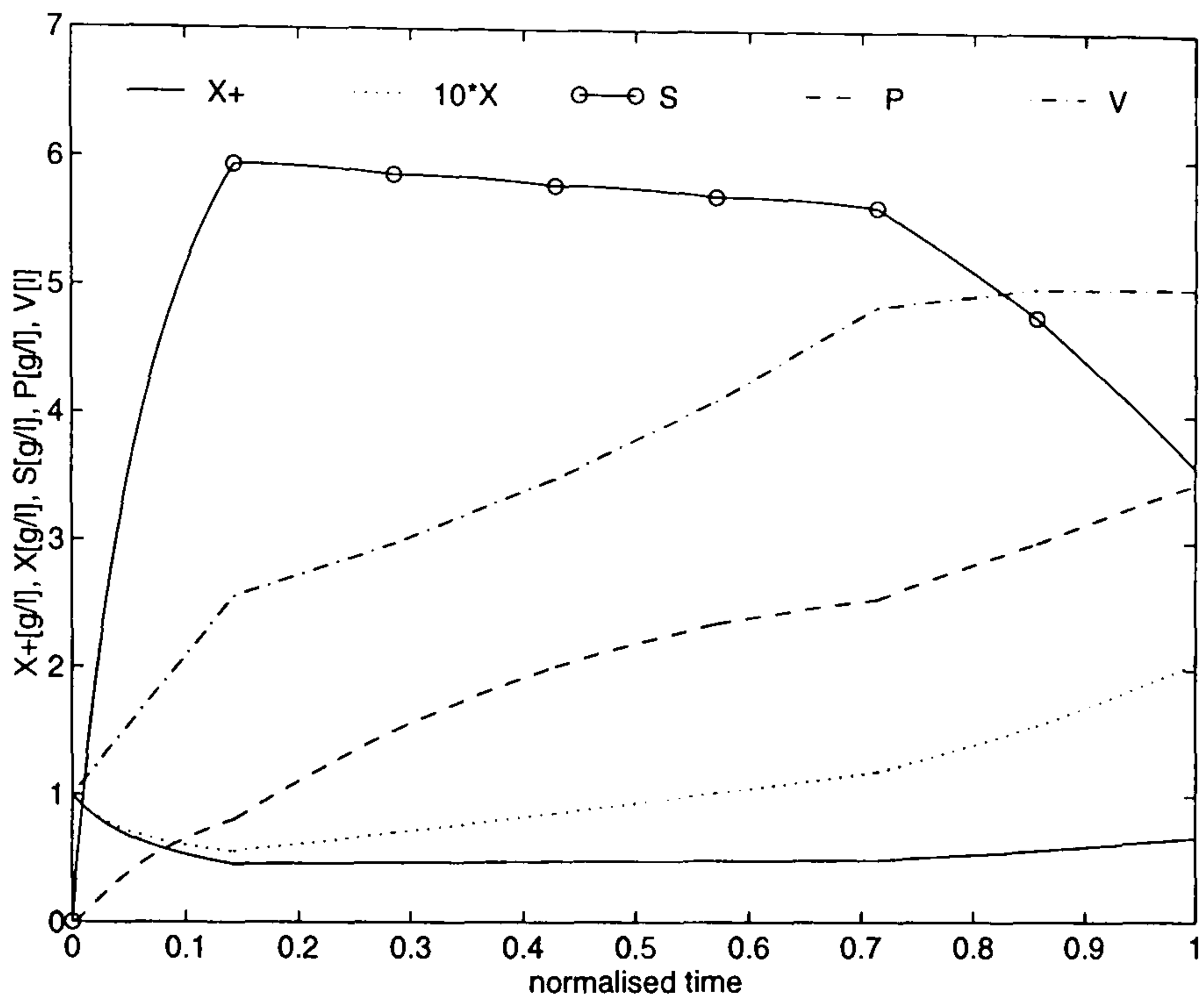

Figure 5.19: Optimal robust nominal profiles of the states

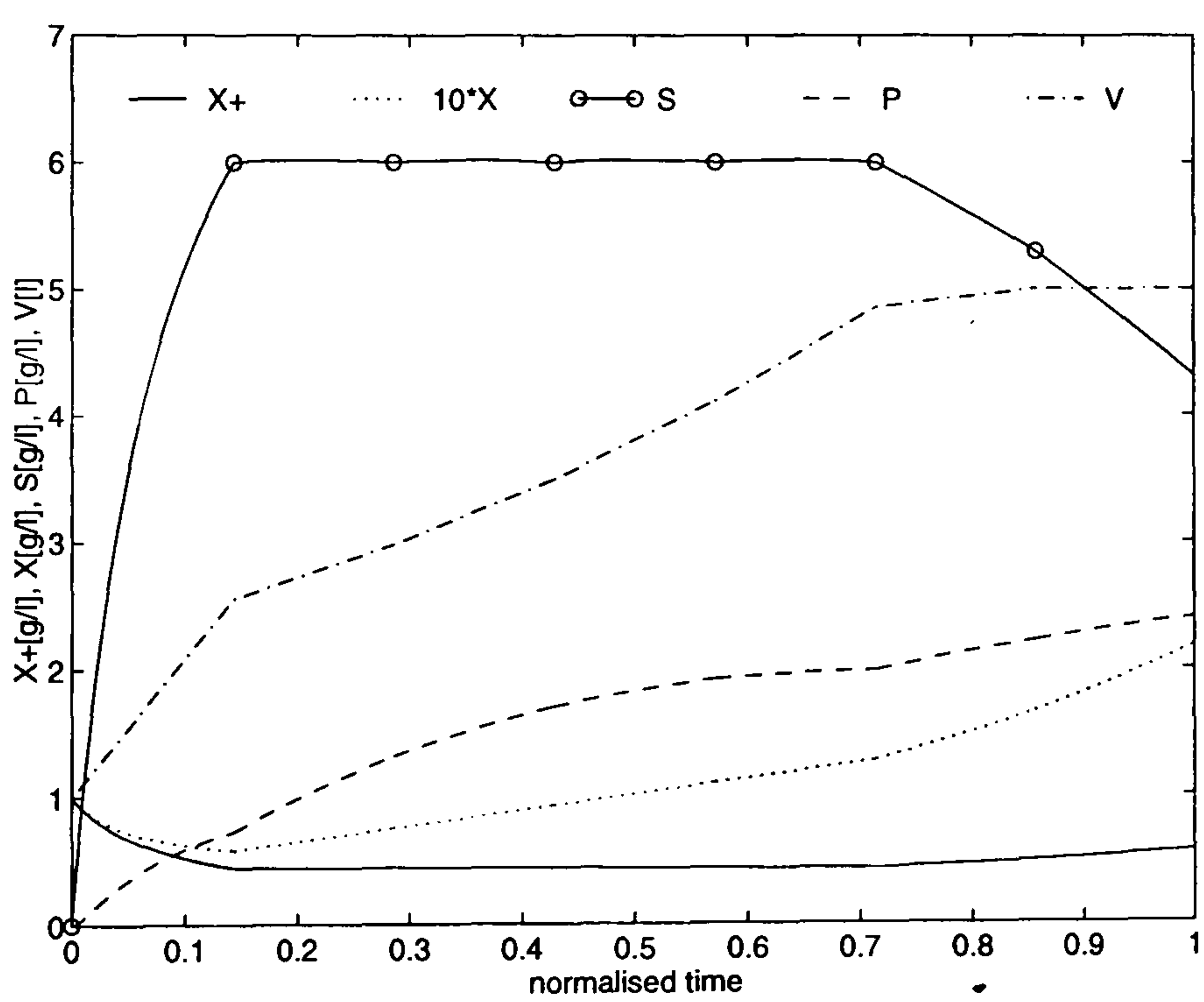

Figure 5.20: Robust worst case profiles of the states

which is equivalent to

$$\Phi_2(t_f) = \max_{p \in P} [M(t_f) - M_p(t_f)] \tag{5.31}$$

since the nominal metabolite concentration is independent of p. The objective function of the multi-objective optimisation problem is then given by

$$\Phi(t_f) = \begin{pmatrix} M(t_f)V(t_f) \\ \max_{p \in P} [M(t_f) - M_p(t_f)] \end{pmatrix} \tag{5.32}$$

Reformulating the second objective function as an ε-constraint

$$\max_{p \in P} [M(t_f) - M_p(t_f)] \leq \varepsilon \tag{5.33}$$

requires that an absolute difference between the nominal and the worst case concentrations is specified a priori . For small differences, this can lead to a small feasible region in the optimisation problem which makes the solution somewhat difficult. An alternative objective function, which expresses the same performance measure, is the worst case ratio of the nominal to the perturbed final metabolite concentration

$$\Phi_2 = \max_{p \in P} \frac{M(t_f)}{M_p(t_f)} \tag{5.34}$$

The corresponding ε-constraint is then

$$\max_{p \in P} \frac{M(t_f)}{M_p(t_f)} \leq \varepsilon \tag{5.35}$$

This constraint is equivalent to

$$\frac{M(t_f)}{M_p(t_f)} \leq \varepsilon \quad \forall p \in P \tag{5.36}$$

which can be rearranged to

$$\gamma M(t_f) - M_p(t_f) \leq 0 \quad \forall p \in P \tag{5.37}$$

where

$$\gamma = \frac{1}{\varepsilon} \tag{5.38}$$

This inequality can be interpreted as an additional uncertainty constraint which requires that the final worst case metabolite concentration is at least a fraction γ of the nominal metabolite concentration.

The trade-off curve between the two objective functions is shown by the solid line in Fig. 5.21. The nominal objective function is plotted against the worst case lower bound of the metabolite concentration given here as percentage of the nominal substrate concentration. For 'generous' bounds, the value of the first objective function is the same as that for the open-loop robust optimisation problem. Tighter bounds on the worst case final metabolite concentration lead to less product in the nominal case.

5.3.5 Multi-Objective Optimisation with Feedback

The dotted line in Fig. 5.21 gives the value of the nominal optimal product. This value serves as an upper bound for the robust optimisation. It is shown that for generous bounds (below $\approx$ 70%), the robust open loop optimisation yields results close to the nominal optimisation. It can therefore be concluded that a feedback control system cannot give a significantly better performance since there is not much scope for improvement. For tighter bounds on the worst case metabolite concentration, the robust open-loop profit decreases rapidly. A feedback controller can be implemented to improve the situation.

A feedback controller of the form

$$\hat{u} = K_p(M(t)_{nom} - M(t)) \tag{5.39}$$

and

$$u(t) = sat(\hat{u}) \tag{5.40}$$

where $sat(\cdot)$ is the saturation function given in Eq. 4.35, is incorporated into the model. The flow rate was chosen as a control input and the controller parameter, K_p, was included as an optimisation variable in the optimisation problem. The results of applying the feedback controller for different bounds are indicated by crosses in Fig. 5.21. The corresponding open-loop input profiles are shown in Fig. 5.22. Compared to the case without controller, the controller makes it possible for the robust nominal objective function to be equivalent to the nominal optimisation case. This holds even for tight bounds. Note that this result does not imply that the total amount of metabolite in the perturbed case is close to the nominal case. In order to achieve this, the worst case objective function has to be traded off against the nominal objective function.

5.3.6 Conclusions

In this case study it is shown that the robust optimisation approach can be incorporated in a multi-objective optimisation framework and the nominal objective function is traded off against a worst case objective function. The results have been

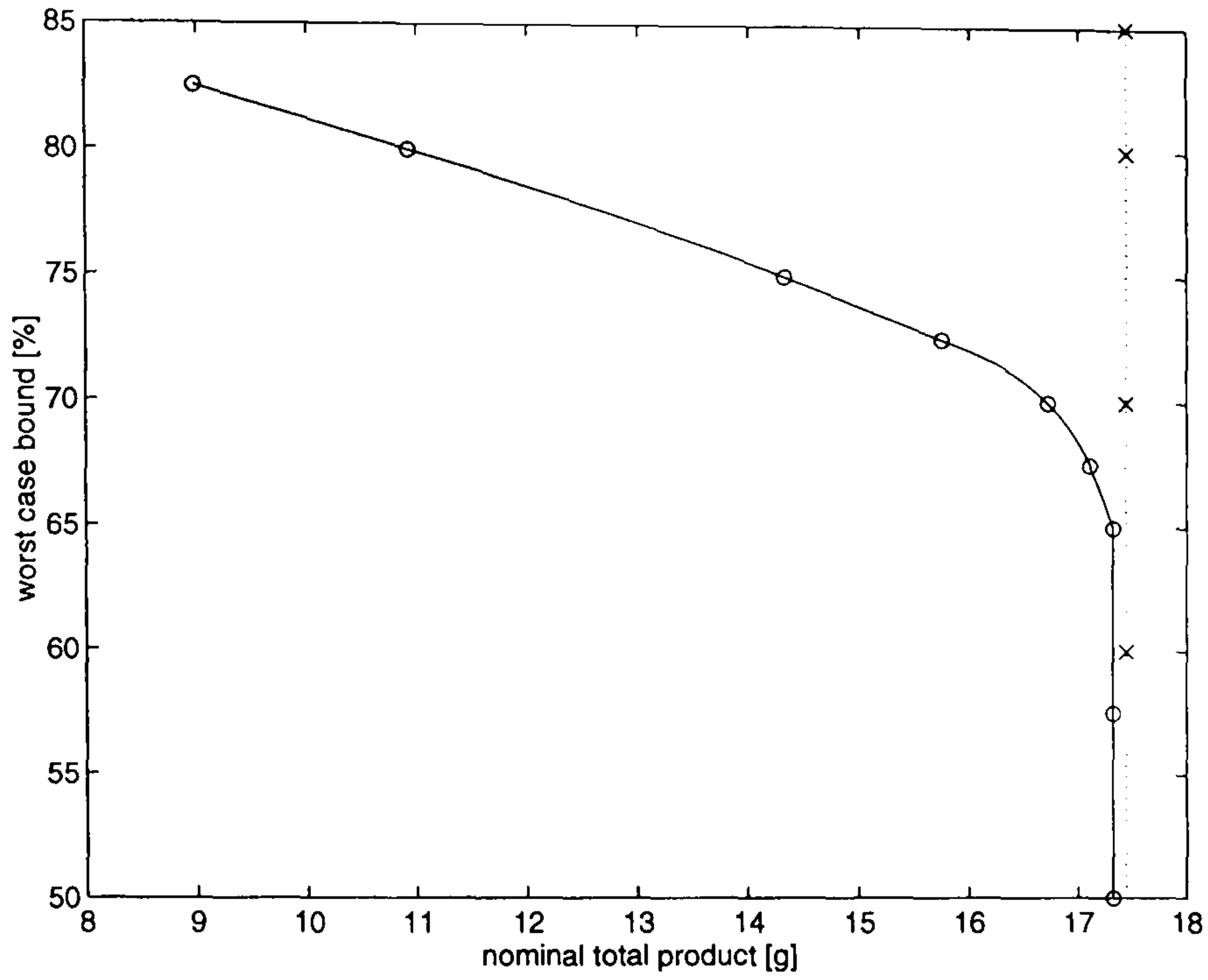

Figure 5.21: Trade-off curve between the controllability measure and the econimic measure for robust open-loop (full line) and feedback (dotted line) optimisation

compared to the nominal optimisation and robust optimisation ones. A feedback controller improves the optimisation results and does it significantly more if tighter bounds on the worst case performance are imposed.

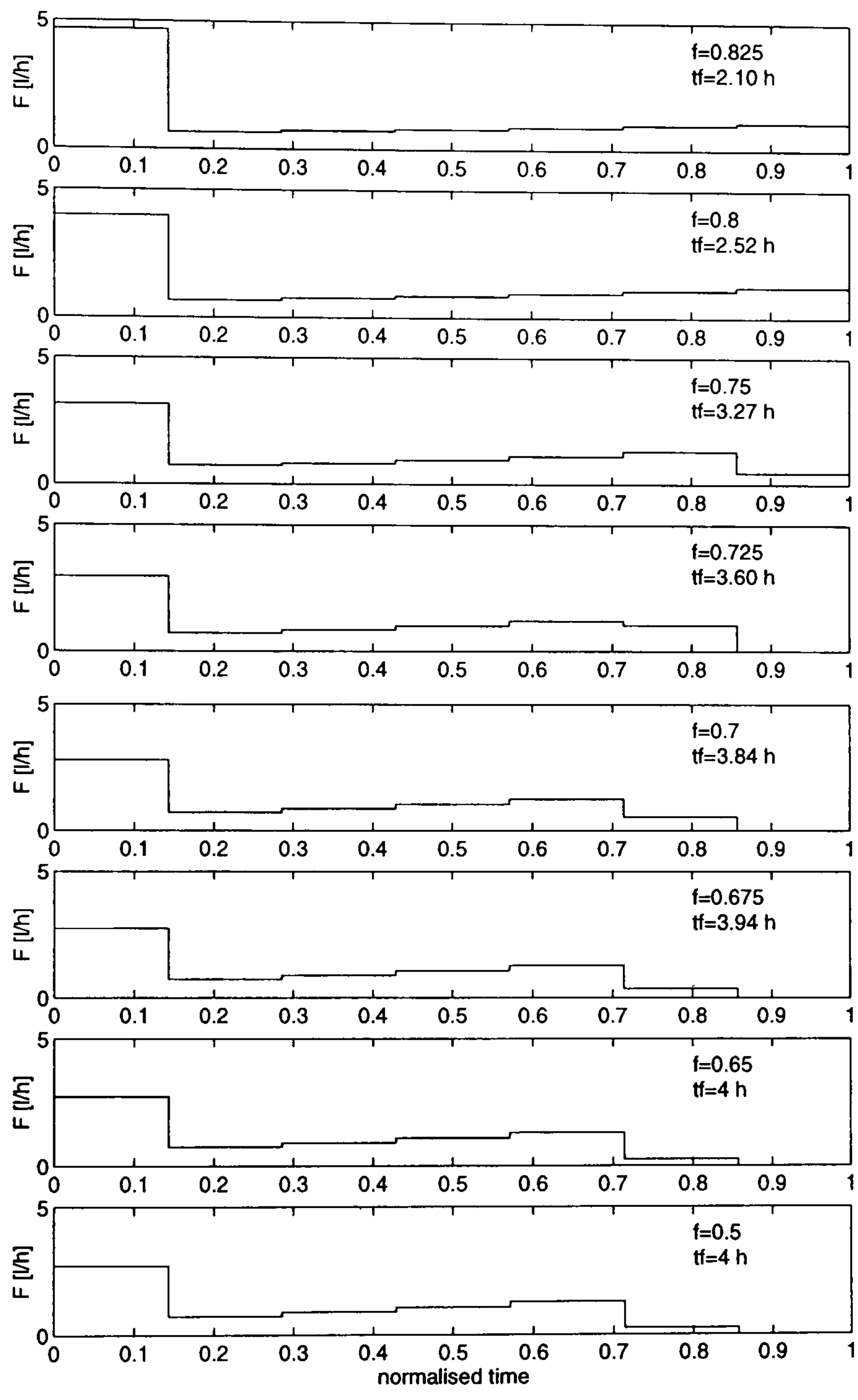

Figure 5.22: Open-loop input profiles

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis addresses the integration of controllability into the design procedure of fermentation processes. Chapter one gives an introduction to this area and motivates the work.

In chapter two methods for evaluating controllability of a given process design are reviewed. Some of these methods are based on linear process models and are suitable for continuous processes operating at steady state. These methods can be used to compare alternative process designs with respect to their controllability properties as was demonstrated in the first case study in chapter four. Four continuous fermenter designs were compared with respect to their controllability properties. These processes are often quoted in the literature and are designed such that they have similar steady state economics. From an economic point of view it is not always obvious which is the most attractive design for a fermentation process. It is shown that the structure of the process has an influence on the controllability of the fermentation. It is found that from a controllability point of view it is more advantageous to utilise the flow rate of a second feed stream containing no growth limiting substrate as a control input instead of manipulating the overall inlet flow rate.

It is not straightforward to incorporate controllability indicators into a systematic design procedure. Methods to design steady state controllable (flexible) processes have been proposed in the literature. Recently these methods have been extended to cover dynamical aspects of continuous processes. Fermentation processes are often operated in fed-batch mode and the above methods are not readily applicable.

Chapter three reviews and summarises methods to reformulate and solve optimal control problems by utilising NLP techniques. The control function is parameterised in order to convert the infinite dimensional optimisation problem into a finite dimensional description. Different methods to discretise the ODE model and to reformulate path constraints are discussed. The most suitable methods for optimising fed batch fermentations are identified. The identification of a suitable solution method is found to be critical to the successful solution of the optimisation problem.

In chapter four a method for the intergration of process design and control is proposed which is applicable to fed batch fermentations. The control system and the process are designed simultaneously. The problem of designing a fed-batch fermenter is formulated as an optimal control problem. The controller design problem is formulated as that of satisfying a set of constraints for a bounded set of uncertain parameters. The controller problem is added to the design problem of a fed-batch fermenter. In this way the controller model is included into the process model. The design parameters of the process as well as the controller parameters are determined simultaneously by the optimisation. An algorithm to solve this overall design and control problem is presented. In this formulation the nominal fermentation model is optimised and the control objective (expressed as a set of constraints) is guaranteed to be satisfied. However, it is still possible that the process performance is very sensitive to model uncertainty (e.g. the fermenter can still operate non-profitably in the presence of uncertainty). In order to avoid this scenario, a new problem formulation is proposed. A second objective function is introduced into the problem formulation which is based on a worst case scenario. This second objective function expresses the worst possible outcome with respect to a certain performance measure. The trade-off between this worst case performance and the original objective function are explored by optimising the two objective functions in a multi-objective optimisation framework. It is shown that the algorithm presented in chapter four can be applied to this problem by using the ϵ-constraint method to reformulate the problem. In this formulation, it is possible that the designer specifies an acceptable worst case performance and the fermentation can be optimised for this specification. This is demonstrated on two case studies. In both case studies it is shown that optimising a nominal model gives results which are sensitive to uncertainties in the process model. It is demonstrated that less sensitive results can be achieved with the proposed design method. It is shown also that the proposed method can be used to identify operating strategies where a control system can potentially improve the fermentation performance. This can be achieved without implementing a specific controller by constructing the trade off curve between the two objective functions as discussed above.

The major conclusion which comes out of this thesis is that the application of the controllability tools developed in this thesis at the design phase of fermentation systems have the potential to yield more controllable processes. Tools from the literature have been used and further developed to give improved controllability results for fermentation systems. The power of these tools has been demonstrated in different case studies and they provide a framework which has the potential to encourage engineers to apply these tools in practice. The design framework is based on unstructured fermentation models and it is assumed that such a model is available. Additionally it is assumed that an estimate of the model parameter uncertainties is available. These assumptions do not always apply in practice. A framework based on experimental data and including different model types is desirable as a future goal. However, for some fermentation systems these models are available and the proposed tools can be applied. The engineer can develop fermenter designs based on these models which satisfy controllability requirements and consider other objectives as economic measures as well. Considering only economic measures which do not take any operational aspects into account to design a fermentation process can lead to designs which are sensitive to uncertainties in the model on which the design is based. This can lead to process designs which are difficult to control. The tools proposed in this work can avoid these difficulties.

The presented robust optimisation technique has been successfully applied to fed batch fermentation processes. Various approaches to solve optimal control problems have been studied in order to identify the most suitable one for fermentation systems. This yielded an optimisation procedure which has promise in terms of shorter implementation times for 'real life' applications.

The multi-objective robust optimisation approach enables the engineer to study a possible trade off between the controllability measures and some economic measures in a systematic way. This avoids the usual and time consuming 'trial and error' approaches adopted in the past. There are certainly still some open questions which have not been addressed in this thesis and some of them are an outcome of this study. These are discussed in the next section.

The results obtained in this thesis are very encouraging and the next step is perhaps to apply them on an experimental pilot plant in order to demonstrate their applicability on a real process.

6.2 Future Work

The approach of integrated design and control procedure proposed in this thesis assumes that a certain type of controller is included in the plant model. As a next step, more general controller parameterisations should be investigated such that the final result is less constrained by the type of controller chosen. Including a MPC controller into the model formulation increases the size of the optimisation problem. The resulting optimisation problem is a multilevel problem. Investigation of efficient algorithms suitable to solve this type of problem may allow the use of this approach as an alternative to the one described in this thesis for the design of controllable processes.

The design procedure presented in this thesis assumes that the requirements of the downstream processing units, which have to be satisfied by the fermentation process are known. They are directly integrated into the problem formulation via the robustness constraints. Since these requirements are not always clear at this stage of the design process, it would be desirable to include the design of the downstream units into the problem formulation in order to remove any of these assumptions.

The robust design problem proposed here for the design of controllable fermenters is formulated as a semi-infinite optimal control problem. This problem is reformulated as a NLP problem. The resulting optimisation problem is large scale with many nonlinear constraints. Optimisation methods have been proposed in the literature [90, 44, 80] which explore the special structure of this type of optimal control problems. Methods which explore the special structure of multiperiod design problems have also been proposed [94]. It has been shown that interior point methods are capable of dealing effectively with a large number of inequality constraints [99]. Combining these ideas into a single algorithm should lead to an improved solution procedure for the class of problems proposed in this thesis.

The control structure selection could be included into the problem formulation by introducing integer variables. This leads to a mixed integer optimal control problem where solution procedures have recently been proposed ([58]).

Appendix A

Nomenclature

a Constant.

b Constant.

c_i Collocation point.

d Design parameter.

$f(\cdot)$ Objective function or dynamic model.

$g(\cdot)$ Design constraints.

$h(\cdot)$ Model of a system.

h Step size.

i, j Components or indices. $i = 1 \ldots n$.

k Runge Kutta parameters or indices.

n Dimensionality.

p	Uncertain parameter in a system or search direction.
q	Control parameter.
r	Reaction rate or relaxation constant.
t	Time.
u	Control input of a system.
x	The state variables. ($x \in \mathrm{R}^n$)
x_i	A component of x. ($x_i \in \mathrm{R}$)
x^*	Optimal solution.
y	Output of a system.
$\mathcal{L}(\lambda, \S)$	Lagrangian function.
R	The set of real numbers.
B	Quasi Newton update of the Hessian of the Lagrangian.
$C(\cdots)$	Objective function.
F_{in}	Feed stream into the reactor.
F_{out}	Stream out of the reactor.
$G(s)$	Transfer function in the Laplace domain.
$G_p(s)$	Disturbance transfer function.
J	Jacobian matrix
K	Constant in a specific rate.

P	Bounded set of uncertain parameters or product concentration.
S	Substrate concentration.
S_f	Feed substrate cocnentration.
X	Cell concentration.
Y	Yield factor.
Φ	Objective function.
$\underline{\sigma}(G)$	Minimum singular value of G.
$\overline{\sigma}(G)$	Maximum singular value of G.
γ	Condition number.
$\gamma^{\star}$	Disturbance condition number.
δ	Flexibility index.
ρ	Specific substrate consumption rate or penalty parameter.
μ_{max}	Maximum specific growth rate.
λ	Lagrangien multipliers.
σ	Weights in the merit function.
φ	Control parameterisation.
ϕ	Collocation polynomial or merit function.
ε	Constant.
μ	Specific growth rate or Lagrangian multipliers.
ν	Lagrangien multipliers.

Appendix B

Implementation Details

In this appendix algorithmic implementation details for the case studies 2 and 3 are given. First the implementation of the solution algorithm as presented in Chapter 3 and Chapter 4 is summarised. Then the case study specific implementation details (e.g. optimisation tolerances and integration accuracy) are elucidated together with a summary of the results.

A flow diagram of the overall solution algorithm is presented in Fig. B.1. The theoretical background with a discussion of the algorithm is given in Chapter 4.3.1.

In the initialisation step the parameters, as for example the integration tolerances for the multiple shooting discretisation, are set. The initial members of the discretised uncertainty space are selected (see Chapter 4.3.1 item (a)). The algorithm proceeds then by solving the discretised OCP (see Problem 4.10) for the initial selected set P_0. For all case studies the multiple shooting discretisation was used. When the optimisation terminates a problem to find the constraint maximiser for this solution is solved (see Chapter 4.3.1 item (c)). If this constraint maximiser satisfies all the constraints in the problem the algorithm terminates. In case the constraint

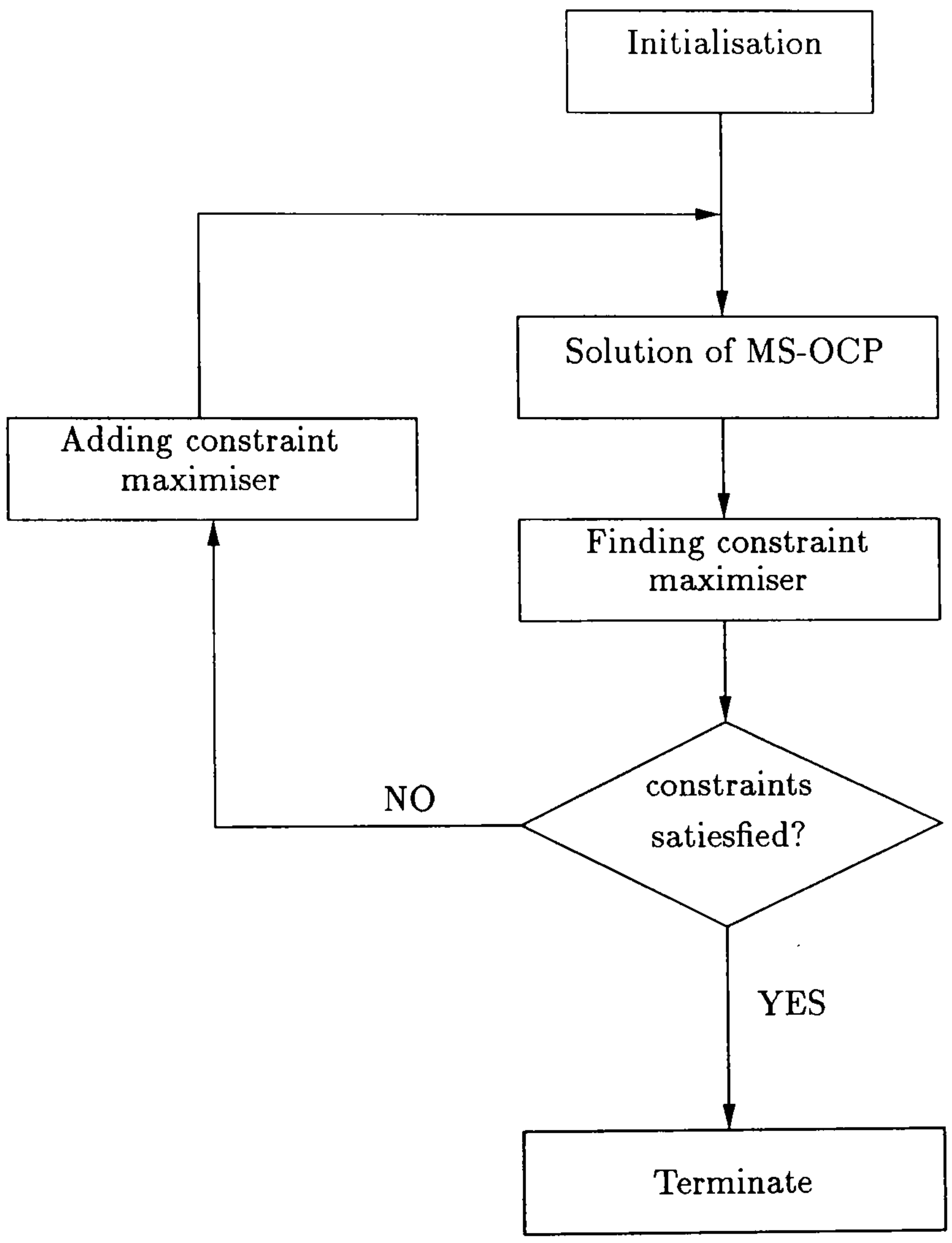

Figure B.1: Flow diagram of the overall solution algorithm.

maximiser violates one of the constraints, this constraint maximiser is used to update the current set P_i in order to form the set P_{i+1} and the algorithm goes back to solve the discretised SIOCP.

The different steps in the algorithm are as follows:

(a) *Initialisation:*

The tolerance for the termination criteria Eq. 3.27 of the SQP in step (*b*) and (*c*) is selected. The integration accuracy for the solution of the model equations in the multiple shooting intervals is set. The nominal model parameters and the lower and upper bound on the uncertain parameters have to be specified.

(b) *Solution of MS-OCP:*

The OCP is discretised by the multiple shooting method described in Chapter 3. The number of control intervals for the control vector parameterisation and the number of discretisation intervals for the model equations has to be selected in advance. The function for the control parameterisation has to be implemented in the model. The discretisation grid for the path constraints has to be implemented. The integration method and the method for gradient evaluation has to be implemented.

(c) *Finding the constraint maximiser:*

Here the constraint maximiser for the discretised OCP from the previous step has to be found.

(d) *Adding the constraint maximiser:*

Here the new set P is formed for the next iteration of the algorithm if the constraint maximiser does not satisfy all the constraints

B.1 Case Study 2

Here the information specific to case study 2 is given. The nominal optimisation problem is defined by the Eqs. 5.9 - 5.14. The robustness constraints are given by Eq. 5.15.

(a) *Initialisation:*

The nominal model parameters are shown in Table B.1, and the lower and

μ_{max}	1.00 $1/h$
Y	0.50 g/g
K	0.03 g/l

Table B.1: The nominal model parameters for case study 2

upper bound for the uncertain parameters are given in Table B.2. The termi-

p	$\underline{p}$	$\overline{p}$
μ_{max}	0.9 $1/h$	1.1 $1/h$
K	0.027 g/l	0.033 g/l

Table B.2: The bounds on the uncertain parameters for case study 2

nation tolerance for the SQP was chosen to be $\epsilon = 10^{-3}$ and the integration accuracy for the multiple shooting discretisation was set to $\epsilon = 10^{-4}$.

(b) *Solving MS-OCP:*

The number of control intervals was set to ten during the fermentation. The size of each control interval was chosen to be of equal size. The grids for the multiple shooting discretisation and the path constraint discretisation were identical with the control discretisation. The control function was approximated in each interval by a piecewise constant function. The model equations were solved by an IRK method as described in Chapter 3.7. The gradients were obtained by solving the sensitivity equations, Eqs. 3.121 and 3.122, simultaneously with the model equations. The Jacobian of the model equations was supplied analytically.

(d) *Adding the constraint maximiser:*
The constraint maximiser was added to the current set P_i to form the discretised uncertainty space P_{i+1} for the next iteration.

The different solutions of the case study are summarised in Table B.3. The parameter combination corresponding to all worst cases is $\mu_{max} = 0.91/h$ and $K = 0.033\ g/l$.

	$X_{nom}(t_f)V_{nom}(t_f)$	$X_{nom}(t_f)$	$X_{worstcase}(t_f)$
nominal optimisation	$20.95\ g$	$4.19\ g/l$	$1.28\ g/l$
robust optimisation	$16.51\ g$	$4.13\ g/l$	$3.69\ g/l$
feedback optimisation	$20.95\ g$	$4.19\ g/l$	$3.93\ g/l$

Table B.3: Solution summary for case study 2

B.2 Case Study 3

The nominal optimisation problem for case study 3 is defined by the Eqs. 5.18 to 5.29. The robustness constraints are all the inequality constraints, Eqs. 5.26 - 5.29, in the nominal optimisation problem.

r	0.14
μ_{max}	0.43 $1/h$
Y_S	0.13 g/g
Y_M	0.03 g/g
k	13 g/g
K_s	1.1 g/l
K_m	0.21 g/l
S_f	6.8 g/l

Table B.4: The nominal model parameters for case study 3.

p	$\underline{p}$	$\overline{p}$
r	0.1	0.2
μ_{max}	0.4 $1/h$	0.5 $1/h$
K_s	0.9 g/l	1.2 g/l

Table B.5: The bounds on the uncertain parameters for case study 3.

(a) *Initialisation:*

The nominal model parameters are shown in Table B.4, and the lower and upper bound for the uncertain parameters are given in Table B.5. The termination tolerance for the SQP was chosen to be $\epsilon = 10^{-3}$ and the integration accuracy for the multiple shooting discretisation was set to $\epsilon = 10^{-4}$.

(b) *Solving MS-OCP:*

The number of control intervals was set to seven during fermentation. The length of each control interval was chosen to be of equal size. The grids for the multiple shooting discretisation and the path constraint discretisation were identical with the control discretisation. The control function was approximated in each interval by a piecewise constant function. The model equations were solved using by a third-order ERK described in Chapter 3.3.2.1. The gradients were obtained by applying IND as explained in Chapter 3.6.1.

(d) *Adding the constraint maximiser:*

The constraint maximiser was added to the current set P_i to form the discretised uncertainty space P_{i+1} for the next iteration.

The different solutions of this case study are summarised in Figure 5.21 and given in Table B.6.

	$X_{nom}(t_f)V_{nom}(t_f)$	$M_{nom}(t_f)$	$\frac{M_{worst\ case}(t_f)}{M_{nom}(t_f)}$
nominal optimisation	17.44 g	3.49 g/l	
robust optimisation	17.32 g	3.46 g/l	
feedback optimisation	17.44 g	3.49 g/l	
mult.-obj. optimisation	17.32 g	3.46 g/l	≤ 0.65
mult.-obj. optimisation	17.11 g	3.42 g/l	0.675
mult.-obj. optimisation	16.72 g	3.35 g/l	0.7
mult.-obj. optimisation	15.76 g	3.15 g/l	0.725
mult.-obj. optimisation	14.33 g	2.87 g/l	0.75
mult.-obj. optimisation	10.92 g	3.19 g/l	0.8
mult.-obj. optimisation	8.98 g	3.06 g/l	0.825

Table B.6: Solution summary for case study 3

Bibliography

[1] **Agrawal, P.; Lim, H.C.:** Analysis of various control schemes for continuous bioreactors. Advances in Biochemical Engineering/Biotechnology 30 (1984) 62-90

[2] **Avriel, M., Wilde, D.J.:** Engineering design under uncertainty. I & EC Proc. Des. 8 (1969) 124-131

[3] **Bahri, P.A., Bandoni, J.A., Romagnoli, J.A.:** Integrated flexibility and controllability analysis in design of chemical processes. AIChE J. 43 (1997) 997-1015

[4] **Bahri, P.A., Bandoni, J.A., Romagnoli, J.A.:** Operability assessment in chemical plants. Comp. Chem. Eng. 20 (1996) S787-792

[5] **Baley, J.E., Ollis, D.F.:** Biochemical engineering fundamentals. McGraw-Hill (1986)

[6] **Biegler, L.T.:** Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Comp. Chem. Eng. 8 (1984) 243-248

[7] **Biegler, L.T., Cuthrell J.E.:** Improved infeasible path optimization for sequential quadratic modular simulators-II: The optimization algorithm. Comp. Chem. Eng. 9 (1985) 257-267

[8] **Biegler, L.T.:** Optimization strategies for complex process models. Adv. Chem. Eng. (1992) 197-257

[9] **Boggs, P.T., Tolle, J.W.:** Sequential Quadratic Programming. Acta Numerica (1996) 1 - 52

[10] **Brengel, D.D.; Seider W.D.:** Coordinated design and control optimisation of nonlinear processes. Comp. Chem. Eng. 16 (1992) 861-886

[11] **Bristol, E.H.:** On a new meassure of interaction for multivariable process control. IEEE Trans. Autom. Cont. 11 (1966) 186-193

[12] **Brockett, R.W., Mesarovic, M.D.:** The reproducibility of multivariable systems. J. Math. Anal. 11 (1965) 548-561

[13] **Bryson, A.E., Ho, Y.C.:** Applied optimal control Hemisphere Publishing Corporation, New York (1975)

[14] **Buchauer, O., Hiltmann, P., Kiehl, M.:** Sensitivity analysis of initial-value problems with application to shooting techniques. Numerische Mathematik 67 (1994) 151-159

[15] **Büskens, C:** Direkte Optimierungsmethoden zur numerischen Berechnung optimaler Steuerungen. Diplomarbeit (1993), Institut für numerische und instrumentelle Mathematik, Westfälische Wilhelms-Universität Münster

[16] **Chamberlain,R., Lemarechal C., Pedersen, H.C., Powell, M.J.D.:** The watchdog technique for forcing convergence in algorithms for constrained optimization. Math. Prog. 16 (1982) 1-17

[17] **Chen, C.T., Hwang, C.:** Optimal control computation for differential-algebraic process systems with general constraints. Chem. Eng. Com. 97 (1990) 9-26

[18] **Clark, P.A., Westerberg, A.W.:** Optimization for design problems having more than one objective. Comp. Chem. Eng. 7 (1983) 259-278

[19] **Clark, P.A., Westerberg, A.W. :** Bilevel programming for steady state chemical process design - 1. Fundamentals and algorithms. Comp. Chem. Eng. 14 (1990) 87-102

[20] **Cuthrell, J.E. Biegler, L.T.:** Simultaneous optimization and solution methods for batch reactors control profiles. Comp. Chem. Eng. 12 (1989) 49-62

[21] **Das, I.:** Robustness Optimization for Constrained, Nonlinear Programming Problems. Dept. of Computational & Applied Mathematics, Rice University, Houston, USA (1997)

[22] **Dimitriadis, V.D., Pistikopoulos, E.N.:** Flexibility analysis of dynamic processes. Ind. Eng. Chem. Res. 34 (1995) 4451-4462

[23] **Fu, P.C., Barford, J.P.:** Nonsingular optimal control for fed-batch fermentation processes with a differential algebraic system model. J. Proc.Cont. 3 (1993) 211-218

[24] **Gill, P.E., Murray, W., Wright, M.H.:** Practical Optimization. Academic Press (1981)

[25] **Grossmann, I.E.; Halemane, K.P.; Swaney, R.E.:** Optimization strategies for flexible chemical processes. Comp. Chem. Eng. 7 (1983) 439-462

[26] **Grossmann, I.E., Floudas, C.A. :** Active Constraint strategy for flexibility analysis in chemical processes. Comp.Chem. Eng. 6 (1987) 675-693

[27] **Grossmann, I.E., Straub, D.A.:** Recent developments in the evaluation and optimization of flexible chemical processes. Computer Oriented Process Engineering, Amsterdam (1991)

[28] **Hairer, E., Nørsett, S.P., Wanner G.:** Solving ordinary differential equations I. Springer Verlag (1987)

[29] **Hairer, E., Wanner G.:** Solving ordinary differential equations II. Springer Verlag (1991)

[30] **Halemane, K.P.;Grossmann, I.E.:** Optimal process design under uncertainty. AIChE J. 29 (1983) 425-433

[31] **Han, S.P.:** A globally convergent method for nonlinear programming. JOTA 22 (1977) 297-309

[32] **Hartig, F., Keil, F.J., Luus, R.:** Comparison of optimization methods for a fed-batch reactor. Hung. J. Ind. Chem. 23 (1995) 141-148

[33] **Hettich, R., Kortanek, K.O.:** Semi-infinite programming: Theory, methods and applications. SIAM Review 35 (1993) 380-429

[34] **Hicks G.A., Ray, W.H.:** Approximation methods for optimal control synthesis. Can. J. Chem. Eng. 49 (1971) 522-528

[35] **Hong, J.:** Optimal substrate feeding policy for a fed batch fermentation with substrate and product inhibition. Biotech. Bioeng. 28 (1986) 1421-1431

[36] **Hovd, M., Skogestad, S.:** Simple frequency-dependent tools for analysis of inherent control limitations Automatica 28 (1992) 989-996

1992

[37] **Jansen, B., Roos, C., Terlaky, T.**: A short survey on ten years interior point methods. Report 95-45 (1945) Faculty of Technical Mathematics and Informatics Delft University of Technology

[38] **Jayaraman, B., Holt, B.R.** : An optimization approach to robust nonlinear control design. Int. J. Control 59 (1994) 639-664

[39] **Kalman, R.E.**: Contributions to the theory of optimal control. Bul. Soc. Math. Mex. (1961) 102-119

[40] **Kolstad, C.D., Lasdon, L.S.**: Derivative evaluation and computational experience with large bilevel mathematical programs. JOTA 65 (1990) 485-499

[41] **Kravaris, C., Kantor, J.C.**: Geometric methods for nonlinear process-control . 1. Background Ind & Eng. Chem. Res. 29 (1990) 2295-2310

[42] **Kravaris, C., Kantor, J.C.**: Geometric methods for nonlinear process-control . 2. Controller Synthesis. Ind & Eng. Chem. Res. 29 (1990) 2310-2323

[43] **Kurtanjek, Ž**: Optimal nonsingular control of fed-batch fermentation. Biotech. Bioeng. 37 (1991) 814-823

[44] **Leineweber, D.B., Bock, H.G., Schloder, J.P., Gallitzendorfer, J.V., Schafer, A., Jansohn, P.**: A boundary value approach to the optimization of chemical processes described by DAE models. Report IWR, University of Heidelberg (1997)

[45] **Lenhoff, A.M., Morari, M.**: Design of resilient processing plants. I. process design under consideration of dynamic aspects. Chem. Eng. Sci. 37 (1982) 245-258

[46] **Lewin, D.R.:** A simple tool for disturbance resiliency diagnosis and feed forward control design. Comp. Chem. Eng. 20 (1996) 13-25

[47] **Lim, H.C., Tayeb, Y.J., Modak, J.M., Bonte, P.:** Computational algorithms for optimal feed rates for a class of fed-batch fermentation: Numerical Results for penicillin and cell mass production. Biotech. Bioeng. 28 (1986) 1408-1420

[48] **Lin, X., Newell, R.B.:** Output structural controllability condition for the synthesis of control-systems for chemical processes. Int. J. Syst. Sci. (1991) 107-132

[49] **Logsdon, J.S., Bielger, L.T.:** Decomposition strategies for large-scale dynamic optimization problems. Chem. Eng. Sci. 47 (1992) 851-864

[50] **Luus, R.:** Application of dynamic programming to differential algebraic process systems. Comput. Chem. Eng. 12 (1993) 373-377

[51] **Luyben, M.L., Floudas C.A.:** Analyzing the interaction of design and control – 1. A multiobjective framework and application to binary distillation synthesis. Comput. Chem. Eng. 18 (1994) 933-969

[52] **Machielsen, K.C.P.:** Numerical solution of optimal control problems with state constraints by sequential quadratic programming in function space. PhD thesis (1988), Department of Mathematics and Informatics, Technical University of Delft

[53] **McLellan, P.J.:** A differential-algebraic perspective on nonlinear controller design. Chem. Eng. Sci. 49 (1994) 1663-1679

[54] **Menawat, A.M.; Balachander, J.:** Alternate control structures for chemostat. AIChE J. 37 (1991) 302-306

[55] **Modak, J.M., Lim, H.C.:** Simple nonsingular control approach to fed batch fermentation optimization. Biotech. Bioeng. 33 (1989) 11-15

[56] **Modak, J.M., Lim, H.C., Tayeb, Y.J.:** General characteristics of optimal feed rate profiles for various fed-batch fermentation processes. Biotech. Bioeng. 28 (1986) 1396-1407

[57] **Mohideen, J., Perkins, J.D., Pistikopoulos, E.N.:** Optimal design of dynamic systems under uncertainty. AIChE J. 42 (1996) 2251-2272

[58] **Mohideen, J., Perkins, J.D., Pistikopoulos, E.N.:** Towards an efficient numerical procedure for mixed integer optimal control. Comp. Chem. Eng. 21 (1997) S457-S462

[59] **Morari, M.:** Design of resilient processing plants - III. Chem. Eng Sci. 38 (1983) 1881-1891

[60] **Morari, M.:** Effect of Design on the Controllability of Chemical Plants. IFAC Workshop (1992) London UK 3-16

[61] **Morari, M., Perkins, J.D.:** Controllability. AIChE Symposium (1995) USA

[62] **Moser, A.:** Bioprocess Technology. Springer Verlag 1988

[63] **Ogunnaike, B.A.; Ray, W.H.:** Process dynamics, modelling and control. Oxford University Press 1994

[64] **Pai, C.C.P., Hughes, R.:** Strategies for formulating and solving two-stage problems for process design under uncertainty. Comp. Chem. Eng. 11 (1987) 695-706

[65] **Palazoglu, A., Arkun, Y.** A multiobjective approach to design plants with robust dynamic operability characteristics. Comp. Chem Eng. 10 (1986) 567-575

[66] **Panier, E.R.,Tits, A.L.:** On Combining Feasibility, descent and superlinear convergence in inequality constraint optimization. Math. Prog. 59 (1993) 261-276

[67] **Perkins J.D.:** Interactions between process design and process control. IFAC DYCORD-'89, Maastricht, Netherlands (1989) 195-203

[68] **Perkins, J.D., Walsh S.P.K.:** Optimization as a tool for Design/Control integration. IFAC Workshop (1994) Baltimore USA 1-10

[69] **Petzold, L., Rosen, J.B., Gill, P.E., Laurent, O.J., Park, K.:** Numerical optimal control of parabolic PDEs using DASOPT. Num. Anal. Rep. NA-1 UCSD Department of Mathematics, University of Minnesota (1996)

[70] **Polak, E.** On the mathematical foundations of non differentiable optimization in engineering design. SIAM Review 29 (1987) 21-89

[71] **Powell, M.J.D.:** A fast algorithm for nonlinear constrained optimisation calculations. Numerical Analysis, Lecture Notes in Mathematics, Springer Verlag 630 (1978) 144-157

[72] **Ray, W.H.:** Advanced Process Control. McGraw-Hill Book Company (1981)

[73] **Renfro, J.G.; Morshedi, A.M.; Asbjornsen, O.A.:** Simultaneous optimization and solution of systems described by differential/algebraic equations. Comp. Chem. Eng. 11 (1987) 503-517

[74] **Rosen, O., Luus, R.:** Evaluation of gradients for piecewise constant optimal-control. Comp. Chem. Eng. 15 (1991) 273-281

[75] **Rosenbrock, H.H.:** State-space and multivariable control. Nelson, London (1970)

[76] **Rosenbrock, H.H.:** Computer-aided control system design. Academic press, London (1974)

[77] **Russell, L.W., Perkins, J.D.:** Towards a method for diagnosis of controllability and operability problems in chemical plants. Chem. Eng. Res. Des. 65 (1987) 453-461

[78] **San, K.Y., Stephanopoulos, G.:** Optimization of Fed-Batch Penicillin Fermentation: A case of singular optimal control with state Constraints. Biotech. Bioeng. 34 (1989) 72-78

[79] **Sardonini, C.A., DiBiasio D.:** A Model for growth a Saccharomyces cerevisiae containing a recombinant plasmid in selective media. Biotech. Bioeng. 29 (1987) 469-475

[80] **Schulz, V.H.:** Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite robots. PhD thesis (1996), University of Heidelberg

[81] **Schwartz, A.L.:** Theory and implementation of numerical methods based on Runge-Kutta integration for solving optimal control problems. PhD thesis(1996), University of California at Berkley

[82] **Schweiger, C.A., Floudas, C.A.** Interaction of Design and Control: Optimization with Dynamic Models. AIChE Meeting, Chicago (1996)

[83] **Skogestad, S.:** A procedure for SISO controllability analysis - with application to design of pH-processes. IFAC Workshop on IPDC, Baltimore, USA (1994) 23-28

[84] **Skogestad, S., Wolff,E.A.:** Controllability Measures for Disturbance Rejection IFAC Workshop on IPDC, London, UK (1992) 23-30

[85] **Skogestad, S.; Postlethwaite, I.:** Multivariable Feedback Control. Wiley & Son 1996

[86] **Skogestad, S.:** Dynamics and control of distillation columns - A tutorial introduction. Department of Chemical Engineering, Trondheim, Norway (1997)

[87] **Soroush, M., Kravaris, C.:** Optimal-design and operation of batch reactors .1. theoretical framework. Ind. Eng. Chem. Res. 32 (1993) 866-881

[88] **von Stryk, O., Burlisch, R.:** Direct and indirect methods for trajectory optimization. Ann. of Oper. Res. 37 (1992) 357-373

[89] **Swaney, R., Grossman, I.:** An index for operational flexibility in chemical process design .1. Formulation and theory. AIChE J. 31 (1985) 621-630

[90] **Tanarkit, P., Biegler, L.T.:** Stable decomposition for dynamic optimization. Ind. Eng. Chem. Res. 34 (1995) 1253-1266

[91] **Teo, K.L., Goh, C.J., Wong, K.H.:** A unified computational approach to optimal control problems. Longman Scientific & Technical (1991)

[92] **Terwiesch, P., Ravemark, D., Rippin, D.W.T.:** Risk-conscious operation of batch processes. IFAC-DYCORD-'95, Helsingor (1995) 409-414

[93] **Terwiesch, P., Agarwal, M., Rippin, D.W.T.:** Batch Unit optimization with imperfect modelling: a survey. J. Proc. Con. 4 (1994) 238-258

[94] **Varvarezos, D.K., Biegler, L.T., Grossmann, I.E.:** Multiperiod design optimization with SQP decomposition. Comp. Chem. Eng. 18 (1994) 579-595

[95] **Vidal, V.V.:** Notes on static and dynamic optimization. Technical University of Denmark (1984)

[96] **Vincente, L.N., Calamai, P.H.:** Bilevel and multilevel programming - A bibliography review. J. Glob. Opt. 5 (1994) 291-306

[97] **Walsh,S., Perkins, J.D.:** Operability and control in process synthesis and design. Adv. in Chem. Eng. 23 (1996) 301-401

[98] **Weitz, O.; Lewin, D.R.:** Dynamic Controllability and Resiliency Diagnosis Using Steady State Data. IFAC Workshop on IPDC, Baltimore, USA (1994) 11-16

[99] **Wright, S.J.:** Applying new optimization algorithms to model predictive control. Report (1993) Mathematics and Computer Science Division, Argonne National Laboratory

[100] **Zakian, V., Al-Naib, U.:** Design of dynamical systems and control systems by the methods of inequalitites. Proc. Inst. Elec. Eng. 120 (1973) 1421-1427

[101] **Zakian, V.:** New formulation of the method of inequalities. Proc. Inst. Elec. Eng. 126 (1979) 579-584

[102] **Zhao, Y.; Skogestad, S.:** A comparison of various control schemes for continuous bioreactors. IFAC Advanced Control of Chemical Processes, Kyoto, Japan (1994) 309-314

[103] **Zhao, Y.; Skogestad, S.:** Modelling and control of a continuous biorector with cross-flow filtration. CAB'6, Garmisch-Partenkirchen, Germany (1995)

[104] **Ziegler, J.G., Nichols, NB.:** Process lags in automatic-control circuits. Trans. A.S.M.E. 65 (1943) 433-444