University of Veszprém
Department of Computer Science

DIPLOMA THESIS

Control of nonlinear process systems based on model analysis

Tamás Schné

Supervisor: Prof. Katalin Hangos
Consultant: Dr. Gábor Szederkényi
2003

Nyilatkozat

Alulírott Schné Tamás, diplomázó hallgató, kijelentem, hogy a diplomadolgozatot a Veszprémi Egyetem Számítástudomány Alkalmazása tanszékén készítettem mérnök-informatikus diploma (master of engineering in information technology) megszerzése érdekében.

Kijelentem, hogy a diplomadolgozatban foglaltak saját munkám eredményei, és csak a megadott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem azt, hogy a diplomadolgozatban foglalt eredményeket a Veszprémi Egyetem, valamint a feladatot kiíró szervezeti egység saját céljaira szabadon felhasználhatja.

Veszprém, 2003. május 14.

Schné Tamás

Summary

Fermentation processes are widely used in process industries from pharmaceutical industry to biologic cleaning. It is easy to see that the effective and safety operation of these is very important which needs to be controlled continuously.

In my work I investigated some of the most important problems, that can occur in controlling such nonlinear systems. I used a simple fermentation process model for this, because it is easy to use but it has the most characteristic problems that can occur in the case of large and difficult systems, as well. I considered two basic operation modes: continuous and fed-batch reactors.

With the help of reachability distributions I studied whether the system would be controllable on the whole state-space when the input feed flow is constant but the substrate concentration changes in it. I also investigated the zero-dynamics of the process that means the case when the system output does not change but the value of the state variables is not necessarily constant. Finally, I considered the controllers that reliably stabilize the whole process. In the last section I describe the MAPLE realization of the reachability distribution calculating algorithm.

Keywords: continuous and fed-batch processes, reachability analysis, zero-dynamics, controllers.

Tartalmi összefoglaló

Fermentációs folyamatok az ipar számos területén előfordulnak, a gyógyszergyártástól kezdve az alkohol előallításán át a biológiai tisztításig mindenütt. Könnyen belátható, hogy ezek hatékony és biztonságos működése igen fontos feladat, amihez elengedhetetlen a folyamat állandó kézben tartása.

Dolgozatomban ilyen nemlineáris folyamatrendszerek irányítása során leggyakrabban felmerülő problémákat vizsgáltam meg. Ehhez egy fermentációs reaktor modelljét hívtam segítségül, mert egyszerűsége miatt könnyen kezelhető, de mégis magában hordozza mindazokat a lehetséges problémákat, amik egy nagy és bonyolult rendszer kapcsán előfordulhatnak. Két alapvető működési módot vizsgáltam meg: folyamatos és félfolyamatos működésű reaktorokat.

Az elérhetőségi disztribúció segítségével tanulmányoztam, hogy vajon irányítható lesz-e a rendszer a teljes állapottéren, ha a bemeneti tápoldat mennyisége állandó, de változik benne a tápanyag aránya. Továbbá ellenőriztem a folyamat zeró dinamikáját, vagyis azt az esetet, amikor a rendszer kimenetén nem mutatkozik változás, de az állapotváltozók értéke mégsem állandó. Végül ezek ismeretében kiválasztottam azokat a szabályzókat, amikkel megbízhatóan irányítható rendszert kapunk.

Az utolsó fejezetben az elérhatőségi disztribúciót szamító algoritmus MAPLE realizációját ismertetem.

Kulcsszavak: folyamatos és félfolyamatos működés, elérhetőségi analízis, zerodinamika, szabályzó tervezés.

Contents

1 Introduction

The design aim of process systems is to get energy and/or cost effective systems based on the principles of thermodynamics. It is well known that the dynamic behavior of such process systems is generally nonlinear and their control, based on locally linearized models, is usually difficult.

In this diploma work it is investigated how the methods of nonlinear model analysis (i.e. analysis of reachability and zero dynamics) can be used in designing stabilizing controllers of simple nonlinear process systems (continuous and fed-batch fermenters). These problems were investigated in many papers, e.g. control analysis of fed-batch fermentation processes in [10] or the analysis of zero-dynamics of nonlinear systems in [3]. The necessary definitions and notations are based on [6].

The problems I inspected in my diploma work were inspected in [9] as well, where the input was the feed flow rate. This work showed e.g. that the model of fed-batch fermenters can be transformed into a two variable one (minimal realization) for this input. Another result was that the performance of the feedback linearization controllers was better than the controllers based on locally linearized models.

This diploma work is organized as follows. For better understanding I briefly summarized the basic notions and notations in Chapter 2. The reachability analysis of the fermenters and the investigation of the zero-dynamics can be found in Chapter 3. In Chapter 4 static feedback controllers (pole-placement and LQ) and controllers based on feedback linearization (exact and input-output linearization) are designed. For choosing the appropriate one the stability region, time and input energy needed by the controller was compared to get an acceptable method to control the fermenter. In Chapter 5 a short description of a reachability distribution calculating program can be found.

2 Basic notions and notations

The dynamic properties of an open-loop system indicate control problems and difficulties and may provide guidelines for control structure design. This chapter summarizes the basic mathematical notions, notations and techniques for the analysis and control of nonlinear systems ([5],[6]).

2.1 State-space representation

The general form of state-space representation of a linear time-invariant (LTI) system is written in the following form:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.1}$$

with given initial condition $x(t_0) = x(0)$ and

$$x(t) \in \mathbb{R}^n, \quad u(t) \in \mathbb{R}^m, \quad y(t) \in \mathbb{R}^r$$

being vectors of finite dimensional spaces and

$$A \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m}, \quad C \in \mathbb{R}^{r \times n}, \quad D \in \mathbb{R}^{r \times m}$$

being matrices. In most cases the elements of matrix D are usually zeros.

Definition 2.1 *The state-space representation (SSR) of LTI systems is the quadruplet of the constant matrices (A, B, C, D). The dimension of a SSR is the dimension of the state vector: $dim x(t) = n$. The state space X is the set of all states:*

$$x(t) \in X, \quad dim\, X = n$$

In nonlinear case the matrices (A, B, C, D) can be time dependent, but the most general form is the following:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t), u(t)) \end{aligned} \tag{2.2}$$

The vectors x, u and y have the same dimensions as above.

A special case of this form is the so called *input-affine form* when the nonlinear functions f and h specialize to

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + \textstyle\sum_{i=1}^{m} g_i(x(t))u_i(t) \\ y_i(t) &= h_i(x(t)) \qquad\qquad\qquad\qquad i = 1, \ldots, p \end{aligned} \tag{2.3}$$

The state $x = (x_1, \ldots, x_n)$ is assumed to belong to an open set of X of $\mathbb{R}^n$. It is also assumed (if otherwise not stated) that the number of inputs equals the number of outputs i.e. $m = p$. f and g_i are *n-dimensional* vector fields, i.e. f, $g_i \in \mathbb{R}^n \to \mathbb{R}^n$.

2.2 Coordinates transformations

Transforming the coordinates in the state space is often very useful in order to highlight some properties of interest (e.g. reachability, observability etc.), or to show how certain control problems can be solved.

2.2.1 Linear change of coordinates

This kind of transformation corresponds to the substitution of the original n dimensional state vector x with a new vector z related to x by a transformation of the form

$$z = Tx \tag{2.4}$$

where T is a nonsingular $n \times n$ matrix. Accordingly, the original description of the linear system

$$\dot{x} = Ax + Bu \tag{2.5}$$

$$y = Cx \tag{2.6}$$

is replaced by a new description

$$\dot{z} = \bar{A}z + \bar{B}u \tag{2.7}$$

$$y = \bar{C}z \tag{2.8}$$

where

$$\bar{A} = TAT^{-1}, \quad \bar{B} = TB, \quad \bar{C} = CT^{-1}$$

2.2.2 Nonlinear change of coordinates

A nonlinear change of coordinates is written as

$$z = \Phi(x) \tag{2.9}$$

where Φ represents a $\mathbb{R}^n$-valued function of n variables, i.e.

$$\Phi(x) = \begin{bmatrix} \Phi_1(x) \\ \Phi_2(x) \\ \vdots \\ \Phi_n(x) \end{bmatrix} = \begin{bmatrix} \Phi_1(x_1, \ldots, x_n) \\ \Phi_2(x_1, \ldots, x_n) \\ \vdots \\ \Phi_n(x_1, \ldots, x_n) \end{bmatrix} \tag{2.10}$$

with the following properties

1. Φ is invertible, i.e. there exists a function Φ^{-1} such that $\Phi^{-1}(\Phi(x)) = x$ for all $x \in \mathbb{R}^n$.

2. Φ and Φ^{-1} are both smooth mappings, i.e. they have continuous partial derivatives of any order.

A transformation of this type is called a *global diffeomorphism* on $\mathbb{R}^n$.

Sometimes, a transformation having both these properties and defined for all x is difficult to find. Thus, in most cases one rather looks at transformations defined only in a neighborhood of a given point. A transformation of this type is called a *local diffeomorphism*. In order to check whether a given transformation is a local diffeomorphism or not, the following result is very useful.

Suppose Φ is a smooth function defined on some subset U of $\mathbb{R}^n$. Suppose the jacobian matrix of Φ is nonsingular at a point $x = x^0$. Then, on a suitable open subset U^0 of U, containing x^0, Φ defines a local diffeomorphism.

2.3 Lie-derivative

Let $\lambda \in \mathbb{R}^n \to \mathbb{R}$, $f \in \mathbb{R}^n \to \mathbb{R}^n$ be functions and $U = \mathrm{dom}(f) \cap \mathrm{dom}(\lambda) \subseteq \mathbb{R}^n$ an open set. *The Lie-derivative of λ along f:*

$$L_f\lambda(x) = \frac{\partial\lambda(x)}{\partial x} f(x) = \sum_{i=1}^{n} \frac{\partial\lambda(x)}{\partial x_i} f_i(x) = < d\lambda(x),\, f(x) > \tag{2.11}$$

It can be seen that the Lie-derivative is an element of $\mathbb{R}$. From this follows that the operation above can be used repeatedly:

$$L_g L_f\lambda(x) = \frac{\partial(L_f\lambda(x))}{\partial x} g(x) \tag{2.12}$$

$$L_f^k\lambda(x) = \frac{\partial(L_f^{k-1}\lambda(x))}{\partial x} f(x) \tag{2.13}$$

2.4 Lie-product

Let $f, g \in \mathbb{R}^n \to \mathbb{R}^n$ be real vector-valued functions and $U = \mathrm{dom}(f) \cap \mathrm{dom}(g) \subseteq \mathbb{R}^n$ an open set. *The Lie-product (bracket) of f and g:*

$$[f,\, g](x) = \frac{\partial g(x)}{\partial x} f(x) - \frac{\partial f(x)}{\partial x} g(x) \tag{2.14}$$

where

$$\frac{\partial g(x)}{\partial x} = \begin{bmatrix} \frac{\partial g_1(x)}{\partial x_1} & \frac{\partial g_1(x)}{\partial x_2} & \cdots & \frac{\partial g_1(x)}{\partial x_n} \\ \frac{\partial g_2(x)}{\partial x_1} & \frac{\partial g_2(x)}{\partial x_2} & \cdots & \frac{\partial g_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n(x)}{\partial x_1} & \frac{\partial g_n(x)}{\partial x_2} & \cdots & \frac{\partial g_n(x)}{\partial x_n} \end{bmatrix} \tag{2.15}$$

Naturally, $[f,\, g](x) \in \mathbb{R}^n$. The Lie-product of two functions has some important properties:

- **bilinearity over** $\mathbb{R}$

$$\text{Let } f_1,\, f_2,\, g_1,\, g_2 \in \mathbb{R}^n \to \mathbb{R}^n,\, r_1,\, r_2 \in \mathbb{R}$$

$$[r_1 f_1 + r_2 f_2,\, g_1] = r_1[f_1, g_1] + r_2[f_2, g_1] \tag{2.16}$$

$$[f_1,\, r_1 g_1 + r_2 g_2] = r_1[f_1, g_1] + r_2[f_1, g_2] \tag{2.17}$$

- **skew commutativity**

$$[f,\, g] = -[g,\, f] \tag{2.18}$$

- **Jacobi identity**

$$[f,\, [g,\, h]] + [g,\, [h,\, f]] + [h,\, [f,\, g]] = 0 \tag{2.19}$$

2.5 Reachability of input-affine systems

2.5.1 Controllability of linear systems

For better understanding we review the controllability of linear systems before analyzing the reachability of nonlinear ones. Consider the linear system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.20}$$

in which $x \in \mathbb{R}^n$ denotes the state vector $u \in \mathbb{R}^m$ the input vector and $y \in \mathbb{R}^r$ the output vector. The matrices (A,B,C) consist of real numbers and have proper dimensions.

Definition 2.2 *The linear system (2.20) is reachable if for all* $x_1,\, x_2 \in \mathbb{R}^n$ *there exists such an input* u *that drives the system from the initiate state* x_1 *to the final state* x_2 *in finite time.*

Reachability is a (global) system property in the linear case which is (or is not) valid on the entire state space.

Suppose that there exists a d-dimensional subspace V of $\mathbb{R}^n$ having the following property:

- V is invarinat under A, i.e. $Ax \in V$ for all $x \in V$.
- V contains the image (i.e. the range-space) of the matrix B, i.e. $Bu \in V$ for all $u \in \mathbb{R}^m$.

Then, after in appropriate linear coordinates transformation T in the state space the Eq. (2.20) can be decomposed in the form

$$\begin{aligned} \dot{\bar{x}}_1 &= \bar{A}_{11}\bar{x}_1 + \bar{A}_{12}\bar{x}_2 + \bar{B}_1 u \\ \dot{\bar{x}}_2 &= \bar{A}_{22}\bar{x}_2 \end{aligned} \tag{2.21}$$

where

$$\bar{x} = Tx = [\bar{x}_1 \;\; \bar{x}_2], \quad \bar{A} = TAT^{-1} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ 0 & \bar{A}_{22} \end{bmatrix}, \quad \bar{B} = TB = \begin{bmatrix} \bar{B}_1 \\ 0 \end{bmatrix}$$

and $\bar{x}_1 \in \mathbb{R}^d$, $\bar{x}_2 \in \mathbb{R}^{n-d}$.

Note that $n - d$ can be equal to 0. From the decomposition (2.21) we can see that the set of coordinates denoted by $\bar{x}_2$ does not depend on the input u but only on time.

It is well known from the theory of linear systems that the system (2.20) is controllable (i.e. $d = n$) if and only if

$$\operatorname{rank} \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} = n \tag{2.22}$$

2.5.2 Reachability of nonlinear systems

In this section we briefly summarize the most important results and methods of analyzing local reachability of nonlinear systems based on the book of Isidori [6].

Definition 2.3 *The system (2.3) is said to be locally reachable around a point $x_1 \in X$ if there exists such a neighborhood V of x_1 that the system can reach any state $x_2 \in V$ in finite time with an appropriate input u.*

Opposite to that reachability is a (global) system property in the linear case we can say only that it is generally locally valid in the case of nonlinear systems (i.e. not a system property). Therefore reachability will be used in the local and controllability in the global sense (i.e. a nonlinear system is called controllable if it is reachable around any point of its state space). Obviously, reachability and controllability are equivalent in the case of linear time-invariant systems.

Firstly, we extend the notion of reachability for nonlinear systems. An important concept for studying this property is the so-called *distribution:*

$$\Delta(x) = \text{span}\,\{f_1(x), \ldots, f_d(x)\}$$

where $f_1, \ldots, f_d \in \mathbb{R}^n \to \mathbb{R}^n$ are vector fields and $U = \text{dom}(f_1) = \ldots = \text{dom}(f_d) \subseteq \mathbb{R}^n$ an open set. A vector space is assigned to each point of U. This way, a distribution can be treated as a matrix such that the columns are $f_i,\ i = 1, \ldots, d$. Some operations follow from the definition of distributions:

$$(\Delta_1 + \Delta_2)(x) = \Delta_1(x) + \Delta_2(x) \tag{2.23}$$

$$(\Delta_1 \cap \Delta_2)(x) = \Delta_1(x) \cap \Delta_2(x) \tag{2.24}$$

Obviously, if the distribution Δ_1 and Δ_2 are spanned by the functions $(f_1, \ldots, f_n)$ and $(g_1, \ldots, g_m)$ respectively, then the distribution $(\Delta_1 + \Delta_2)$ is spanned by $(f_1, \ldots, f_n, g_1, \ldots, g_m)$.

Definition 2.4 *1. $f \in \mathbb{R}^n \to \mathbb{R}^n$ is a smooth vector field if it is partially derivable for infinite times and these derivatives are continuous functions.*

2. A distribution Δ is called smooth distribution if it is spanned by smooth vector fields.

3. Nonsingular distribution Δ defined on U:
$\exists d \in \mathbb{N}$ such that

$$dim(\Delta(x)) = d \quad \forall x \in U \tag{2.25}$$

4. x^0 is a regular point of a distribution Δ:
There exists a neighborhood U^0 of x^0 such that Δ is nonsingular on U^0.

5. Point of singularity:
Not a regular point.

6. f belongs to the distribution Δ ($f \in \Delta$):

$$f(x) \in \Delta(x) \quad \forall x \tag{2.26}$$

7. A distribution Δ_1 contains a distribution Δ_2 ($\Delta_2 \subseteq \Delta_1$):

$$\Delta_2(x) \subseteq \Delta_1(x) \quad \forall x \tag{2.27}$$

8. A distribution Δ is involutive:

$$f_1 \in \Delta,\ f_2 \in \Delta \Rightarrow [f_1,\ f_2] \in \Delta \tag{2.28}$$

9. A distribution Δ is invariant under the vector field f :

$$g \in \Delta \Rightarrow [f,\ g] \in \Delta \tag{2.29}$$

Lemma 2.1 *Let Δ be a nonsingular involutive distribution of dimension d and suppose that Δ is invariant under the vector field f. Then at each point x^0 there exist a neighborhood U^0 of x^0 and a coordinates transformation $z = \Phi(x)$ defined on U^0, in which the vector field f is represented by a vector of the form*

$$\bar{f}(z) = \begin{bmatrix} \bar{f}_1(z_1, \ldots, z_d, z_{d+1}, \ldots, z_n) \\ \vdots \\ \bar{f}_d(z_1, \ldots, z_d, z_{d+1}, \ldots, z_n) \\ \bar{f}_{d+1}(z_{d+1}, \ldots, z_n) \\ \vdots \\ \bar{f}_n(z_{d+1}, \ldots, z_n) \end{bmatrix} \tag{2.30}$$

It's not difficult to see that $\bar{f}(z)$ presents the coordinate-dependent nonlinear analogue of $\bar{A}$ in (2.21).

Theorem 2.1 *Let Δ be a nonsingular involutive distribution of dimension d and suppose that Δ is invariant under the vector fields $f, g_1 \ldots, g_m$. Moreover, assume that the distribution span$\{g_1, \ldots, g_m\}$ is contained in Δ. Then, for each point x^0 it is possible to find a neighborhood U^0 of x^0 and a local coordinates transformation $z = \Phi(x)$ defined on U^0 such that, in the new coordinates the system (2.3) is represented by equations of the form*

$$\dot{\zeta}_1 = f_1(\zeta_1, \zeta_2) + \sum_{i=1}^{m} g_i(\zeta_1, \zeta_2) u_i \tag{2.31}$$

$$\dot{\zeta}_2 = f_2(\zeta_2) \tag{2.32}$$

$$y_i = h_i(\zeta_1, \zeta_2) \tag{2.33}$$

where $\zeta_1 = (z_1, \ldots, z_d)$ and $\zeta_2 = (z_{d+1}, \ldots, z_n)$.

Suppose that the assumptions of Theorem 2.1 are satisfied, choose a point x^0 and set $x(0) = x^0$. For small values of t the state remains in U^0 and we can use Eqs. (2.31)-(2.33) to interpret the behavior of the system. From these, we can see that ζ_2 coordinates of $x(t)$ are not affected by the input. If we denote

by $x^0(T)$ the point of U^0 reached at time $t = T$ then it's clear that the set of points that can be reached at time T, starting from x^0, is a set of points whose ζ_2 coordinates are necessarily equal to the ζ_2 coordinates of $x^0(T)$. Roughly speaking, if we can find an appropriate Δ distribution and the local coordinates transformation $z = \Phi(x)$ then we can clearly identify the part of the system that behaves independently of the input in the neighborhood of x^0. It is also important to note that if the dimension of Δ is equal to n then the dimension of the vector ζ_2 is 0, which means that the input affects all state variables in a neighborhood of x^0 (the system is reachable in a neighborhood of x^0).

Lemma 2.2 *Let Δ be a given smooth distribution and $\tau_1, \ldots, \tau_q$ a given set of vector fields. The family of all distributions which are invariant under $\tau_1, \ldots, \tau_q$ and contain Δ has a minimal element, which is a smooth distribution.*

Notation: The smallest distribution that contains Δ and is invariant under the vector fields $g_1, \ldots, g_m$ will be denoted by

$$\langle\, g_1, \ldots, g_m \mid \Delta \rangle \tag{2.34}$$

2.5.3 Algorithm for generating the reachability distribution

As we will see in this subsection, the distribution (2.34) can be constructed algorithmically. The method was proposed by Isidori in 1989 [6]. The description of a MAPLE realization can be found in Chapter 5 and the program code in Appendix B.

1. *At the beginning we have the vector fields $(f = g_0, g_1, \ldots, g_m)$ and*

$$\Delta_0 = \text{span}\,\{\, g_1, \ldots, g_m \,\} \tag{2.35}$$

2. *Development of the reachability distribution*

$$\Delta_k = \Delta_{k-1} + \sum_{i=0}^{m} [g_i, \Delta_{k-1}] \tag{2.36}$$

Note that one term in the last sum $[g_i, \Delta_{k-1}]$ is computed using the functions $(\Phi_1, \ldots, \Phi_l)$ spanning the distribution Δ_{k-1}:

$$[g_i, \Delta_{k-1}] = \text{span}\,\{\, [g_i, \Phi_1], \ldots, [g_i, \Phi_l] \,\} \tag{2.37}$$

It is proved that Δ_k has the property:

$$\Delta_k \subset \langle\, g_0, \ldots, g_m \mid \Delta_0 \,\rangle \tag{2.38}$$

3. *Stopping condition*
   If $\exists k^*$ such that $\Delta_{k^*} = \Delta_{k^*+1}$, then

$$\Delta_{k^*} = \langle\, g_0, \ldots, g_m \mid \Delta_0 \,\rangle \tag{2.39}$$

The algorithm above exhibits some interesting properties. It starts with the distribution spanned by the input functions $g_i(x)$ of the original state equation. Thereafter it requires to compute the Lie-brackets ($[f(x), g_i(x)]$) of the functions $f(x)$ and $g_i(x)$ respectively. Then we further expand the distribution obtained in the previous step until the stopping condition occurs.

2.6 The zero dynamics

The basic definitions and notation are summarized in this section based on the work of Byrnes and Isidori [3]. The zero dynamics is an important concept that plays a role exactly similar to the zeros of the transfer function in a linear system.

Definition 2.5 *Consider the system (2.3) with constraints $y = 0$ that is*

$$\begin{aligned} \dot{x} &= f(x) + \textstyle\sum_{i=1}^{m} g_i(x)u_i \\ 0 &= h(x) \end{aligned} \tag{2.40}$$

The constrained system (2.40) is called the zero-output constrained dynamics, or briefly, the zero dynamics.

2.6.1 The zero dynamics of SISO nonlinear systems

Definition 2.6 *It is said that the system (2.3) with $m = 1$ and $p = 1$ has relative degree r at x^0 if $L_g L_f^k h(x) = 0$ for all x in a neighborhood of x^0 and all $k < r - 1$, and $L_g L_f^{r-1} h(x^0) \neq 0$.*

After a suitable coordinates transformation $z = \Phi(x)$ where $z_i = \phi_i(x) = L_f^{i-1} h(x)$ for $1 \leq i \leq r$ and $L_g \phi_i(x) = 0$ for $r + 1 \leq j \leq n$ the state-space model (2.3) with $m = 1$, $p = 1$ and relative degree r can be rewritten as

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{r-1} &= z_r \\ \dot{z}_r &= b(\xi, \eta) + a(\xi, \eta)u \\ \dot{\eta} &= q(\xi, \eta) \end{aligned} \tag{2.41}$$

where $\xi = [z_1 \ \dots \ z_r]^T$, $\eta = [z_{r+1} \ \dots \ z_n]^T$, $a(\xi,\eta) = L_g L_f^{r-1} h(\Phi^{-1}(\xi,\eta))$ and $b(\xi,\eta) = L_f^r h(\Phi^{-1}(\xi,\eta))$.

The *Problem of Zeroing the Output* is to find, if it exists, pairs consisting of an initial state x^* and input function u defined for all t in a neighborhood of $t = 0$, such that the corresponding output $y(t)$ of the system is identically zero for all t in a neighborhood of $t = 0$. For any fixed initial state x^* the input function u can be determined as follows. Let us set the output to be identically zero, then the system's behavior is governed by the differential equation

$$\dot{\eta}(t) = q(0, \eta(t)). \tag{2.42}$$

The dynamics (2.42) describes the internal behavior of the system when the output is forced to be zero and it is called the *zero dynamics*. The initial state of the system must be set to a value such that $\xi(0) = 0$, while $\eta(0) = \eta^0$ can be chosen arbitrarily. Furthermore, the input must be set as

$$u(t) = -\frac{b(0, \eta(t))}{a(0, \eta(t))} \tag{2.43}$$

where $\eta(t)$ denotes the solution of (2.42) with initial condition $\eta(0) = \eta^0$.

The investigation of the zero dynamics can be extremely useful when selecting the outputs to be controlled since the stabilization of an output with globally asymptotically stable zero dynamics implies the global asymptotic stability of the closed-loop system. This procedure will be illustrated on the example of a bio-reactor in section 3.4. Motivated by the notions of linear systems' theory, nonlinear systems with globally asymptotically stable zero dynamics are called *minimum-phase* systems.

3 Modelling and control analysis of fermentation processes

3.1 A short review of the operation of fermentation processes

Fermentation processes are widely used in the biochemical industries. Fermenters are bio-reactors used for producing some kind of biomass or the by-product of biomass growth. Baker's yeast, antibiotics and beer are typical products of fermentation processes. A fermenter is usually modelled as a continuous stirred tank reactor (CSTR) where the biomass grows in a liquid-phase environment. The schematics of a continuous fermenter is shown in fig. 3.1.

Two basic operation types of fermentation processes can be distinguished:

- In *continuous fermenters* the liquid volume in the reactor is held constant, which means that the inlet feed flow rate is equal to the output flow rate in each time instant (i.e. the product is continuously taken away from the bio-reactor).
- In *fed-batch fermentation processes* the substrate is gradually fed into the reactor, and the whole product is taken away at the end of the process (i.e. the output flow rate is zero during fermentation).

Although the nonlinear characteristics of continuous and fed-batch fermentation processes are similar, the control related problems are quite different in the two cases. However, the control goal is common in most cases: to stabilize the system in a wide neighborhood of the the operating point (or predefined trajectory) and to maximize biomass production.

3.2 Reachability of continuous fermentation processes

3.2.1 State-space model of the fermenter

In order to be able to focus on the key issues in the reachability analysis of bio-reactors, the simplest possible bio-reactor, a perfectly stirred continuous fermenter is chosen as a test case. Despite of its simplicity, it exhibits the key properties which make bio-reactors difficult to control.

Nonlinear state-space model

An isotherm nonlinear continuous fermenter is considered in this section with

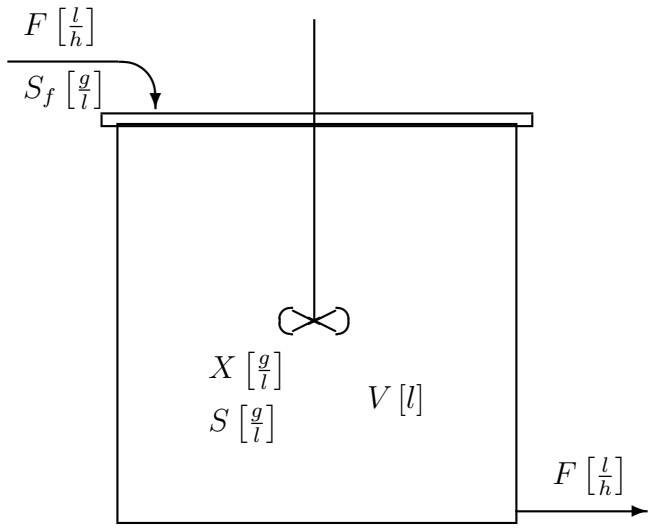

Figure 3.1: A simplified model for a continuous fermenter

constant volume V and constant physico-chemical properties. The dynamics of the process is given by the state-space model

$$\frac{dX}{dt} = \mu(S)X - \frac{XF}{V} \tag{3.1}$$

$$\frac{dS}{dt} = -\frac{\mu(S)X}{Y} - \frac{SF}{V} + \frac{FS_f}{V} \tag{3.2}$$

where

$$\mu(S) = \mu_{max}\frac{S}{K_2S^2 + S + K_1} \tag{3.3}$$

The first equation originates from the biomass component balance, while the second is from the substrate component mass balance. They are coupled by

X	biomass concentration		$\left[\frac{g}{l}\right]$
S	substrate concentration		$\left[\frac{g}{l}\right]$
S_f	substrate feed concentration		$\left[\frac{g}{l}\right]$
F	feed flow rate	3.2	$\left[\frac{l}{h}\right]$
V	volume	4	$[l]$
Y	yield coefficient	0.5	$[-]$
μ_{max}	kinetic parameter	1	$\left[\frac{1}{h}\right]$
K_1	kinetic parameter	0.03	$\left[\frac{g}{l}\right]$
K_2	kinetic parameter	0.5	$\left[\frac{l}{g}\right]$

Table 3.1: Variables and parameters of the fermentation process model

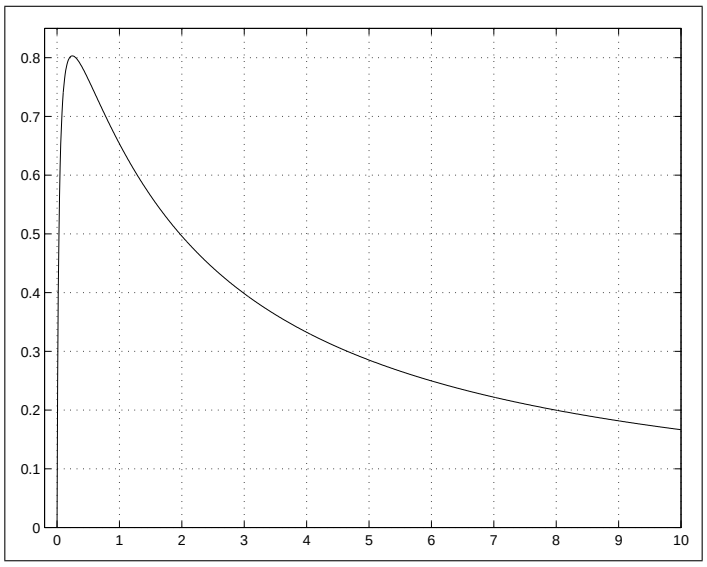

Figure 3.2: The nonlinear reaction rate function $\mu(S)$

the nonlinear reaction rate function $\mu(S)$ which is the main source of the nonlinearity and uncertainty in this simple model. The variables and parameters of the model together with their units and parameter values are given in Table 3.1 and taken from [7]. The above model can easily be written in standard input-affine form with the state vector $x = [X \;\; S]^T$. The substrate feed concentration is chosen as manipulable input variable $u = S_f$. The $u = F$ case was investigated in [9].

$$\dot{x} = f(x) + g(x)u \tag{3.4}$$

$$f(x) = \begin{bmatrix} \mu(S)X - \frac{XF}{V} \\ -\frac{\mu(S)X}{Y} - \frac{SF}{V} \end{bmatrix}, \; g(x) = \begin{bmatrix} 0 \\ \frac{F}{V} \end{bmatrix} \tag{3.5}$$

In controller design theory centered state vector and input variables are usually used. So the new state vector is $x = [\bar{X} \;\; \bar{S}]^T = [X - X_0 \;\; S - S_0]^T$, the input is $u = \bar{S}_f = S_f - S_{f_0}$ and $f(x)$ and $g(x)$ will have the following form:

$$f(x) = \begin{bmatrix} \mu(\bar{S} + S_0)(\bar{X} + X_0) - \frac{(\bar{X}+X_0)F}{V} \\ -\frac{\mu(\bar{S}+S_0)(\bar{X}+X_0)}{Y} - \frac{(\bar{S}+S_0)F}{V} \end{bmatrix}, \; g(x) = \begin{bmatrix} 0 \\ \frac{F}{V} \end{bmatrix} \tag{3.6}$$

with (X_0, S_0, S_{f_0}) being a steady-state operating point.

Calculation of the optimal operating point

At the optimal equilibrium state the biomass concentration in the outlet flow is at maximum, i.e. XF is maximal. It can be calculated from the nonlinear

model that this equilibrium point is at

$$S_0 = \frac{1}{2}\frac{-2K_1 + 2\sqrt{K_1^2 + S_{f_0}^2 K_1 K_2 + S_{f_0} K_1}}{S_{f_0} K_2 + 1} \tag{3.7}$$

$$X_0 = (S_{f_0} - S_0)Y \tag{3.8}$$

and the corresponding substrate feed concentration is

$$S_{f_0} = 10\frac{g}{l} \tag{3.9}$$

Substituting the parameter values from Table 3.1 gives

$$S_0 = 0.2187\frac{g}{l},\ X_0 = 4.8907\frac{g}{l} \tag{3.10}$$

Linearized model

In order to compare linear and nonlinear control techniques, the linearized version of the nonlinear state equation (3.4) and (3.6) is used around the steady-state point (X_0, S_0, S_{f_0}) above:

$$\dot{x} = Ax + Bu \tag{3.11}$$

where

$$A = \left[\frac{\partial f}{\partial x}\right]_{x=0} = \begin{bmatrix} 0 & 0.4004 \\ -1.6044 & -1.6031 \end{bmatrix},\quad B = g(0) = \begin{bmatrix} 0 \\ 0.8022 \end{bmatrix} \tag{3.12}$$

3.2.2 Controllability analysis

Controllability properties of the system play key role in designing controllers not only in the desired operating point but also for the entire operating region. Therefore, nonlinear analysis techniques are recommended to complement the usual analysis based on locally linearized models.

Analysis based on local linearization

With the help of the matrices A and B (see in Eq. 3.12) we can compute the Kalman-controllability matrix:

$$\mathcal{C} = \begin{bmatrix} 0 & 0.3212 \\ 0.8022 & -1.286 \end{bmatrix} \tag{3.13}$$

The determinant of this matrix is not zero, i.e. it is of full rank, so the system is controllable (in the linear sense) in a close neighborhood of the optimal

operating point. However, in two possible operating points $S^* = \sqrt{\frac{K_1}{K_2}}$ and $X^* = 0$ the controllability matrix has only rank 1, that is, the system is locally not controllable.

Nonlinear reachability analysis

After generating the *reachability distribution* based on the algorithm described in section 2.5.3, nonlinear reachability analysis is used for identifying the singular points of the state-space around which the control of the system is problematic or even impossible. The local reachability distribution is generated incrementally in two steps as follows. The initial distribution is

$$\Delta_0 = \text{span}\{g\} = \text{span}\left\{\begin{bmatrix} 0 \\ \frac{F}{V} \end{bmatrix}\right\} \tag{3.14}$$

This is extended by the Lie-bracket of f and g in the first step

$$\Delta_1 = \Delta_0 + \text{span}\{[f,g]\} = \text{span}\{g,[f,g]\} \tag{3.15}$$

$$\Delta_1 = \begin{bmatrix} 0 & -\frac{\mu'(S)XF}{V} \\ \frac{F}{V} & \frac{\mu'(S)XF}{YV} + \frac{F^2}{V^2} \end{bmatrix} \tag{3.16}$$

where $\mu'(S)$ denotes $\frac{\partial \mu(S)}{\partial S}$. It can be seen that Δ_1 has rank 2, it is of full rank. The second step gives the following:

$$\Delta_2 = \Delta_1 + [f,\Delta_1] + [g,\Delta_1] = \text{span}\{g,[f,g],[f,[f,g]],[g,[f,g]]\} \tag{3.17}$$

The distribution Δ_2 evaluated at a given point of the state-space can be treated as a 2×4 matrix. It is clear that the maximum rank of Δ_2 can be only 2, i.e. it cannot be of full rank and the algorithm stops at this point accordingly.

Singular points

At singular points ($X = 0$ and $\mu'(S) = 0$) Δ_1 has rank 1. We can determine these non-desired states by checking whether the determinant of the distribution is zero at a point. If it is, then we have to make sure of this is a steady-state point or not. $X = 0$ is trivial. It means that there is no biomass in the reactor and because the inlet flow doesn't contain biomass it is a steady-state point. The other singular point $\mu'(S) = 0$ holds if $S = \sqrt{\frac{K_1}{K_2}}$. In this case the substrate concentration is higher than its optimal value but can be changed by the input (S_f) before the biomass concentration would decrease to zero. We can say that this is not a steady-state point. The first singular point is "wash-out" state in a sense, as the biomass disappears from the reactor.

Therefore, this is an undesirable state.

Non-singular points

At any other point in the state-space including the desired operating point $[\bar{X} \quad \bar{S}]^T = [0 \quad 0]^T$ the reachability distribution has rank 2, which means that the system is reachable in a neighborhood of these points and we can apply state feedback controllers to stabilize the process.

3.3 Reachability of fed-batch fermentation processes

The reachability of a simple nonlinear fed-batch fermentation process model is investigated in this section. It is shown that the known difficulties of controlling such processes are primarily caused by the fact that the rank of the reachability distribution is always less than the number of state variables.

3.3.1 Problem statement

Bio-processes in general and fermentation processes in particular are difficult to model and to control even in the simplest cases. The dynamic state-space model of a fermenter is derived from first engineering principles which fix certain structural elements in the model. The state equations are derived from dynamic conservation balances of the overall mass, component masses and energy if applicable. The speciality of a fermentation model appears in the so called source function of these balances which is highly nonlinear and non-monotonous in nature.

The aim of this section is to use rigorous nonlinear analysis of a simple fed-batch fermenter model for analyzing the reachability properties and to relate them to the physico-chemical phenomena taking place in the reactor.

3.3.2 Nonlinear state-space model

The simplest dynamic model of a fed-batch fermenter consists of three conservation balances for the mass of the cells (e.g. yeast to be produced), that of the substrate (e.g. sugar which is consumed by the cells) and for the overall mass. Here we assume that the fermenter is operating under isotherm conditions, that is no energy balance is needed. The cell growth rate is described by a nonlinear static function μ.

Initially, a solution containing both substrate and cell is present in the fermenter. During the operation we feed a solution of substrate with a given

feed flow rate F to the reactor.

Under the above assumptions the nonlinear state-space model of the fermentation process (see Eq. 3.1) is extended by a new equation

$$\frac{dX}{dt} = \mu(S)X - \frac{XF}{V} \tag{3.18}$$

$$\frac{dS}{dt} = -\frac{\mu(S)X}{Y} - \frac{SF}{V} + \frac{FS_f}{V} \tag{3.19}$$

$$\frac{dV}{dt} = F \tag{3.20}$$

where

$$\tag{3.21}$$

$$\mu(S) = \mu_{max}\frac{S}{K_2S^2 + S + K_1} \tag{3.22}$$

The input-affine form with the state vector $x = [X \;\; S \;\; V]^T$ and the input $u = S_f$ is the following:

$$\dot{x} = f(x) + g(x)u \tag{3.23}$$

$$f(x) = \begin{bmatrix} \mu(S)X - \frac{XF}{V} \\ -\frac{\mu(S)X}{Y} - \frac{SF}{V} \\ F \end{bmatrix}, \; g(x) = \begin{bmatrix} 0 \\ \frac{F}{V} \\ 0 \end{bmatrix} \tag{3.24}$$

3.3.3 Reachability analysis

Investigating Eq. (3.24) we can see that the volume depends only on the feed flow rate F and is not influenced by the input or other state variables. From this follows that the system will not be controllable on the entire state-space. We get the same result by constructing the reachability distribution according to the algorithm described in section 2.5.3.

$$\Delta_0 = \text{span}\{g\} \tag{3.25}$$

$$\Delta_1 = \Delta_0 + \text{span}\{[f, g]\} = \text{span}\{g, [f, g]\} \tag{3.26}$$

$$\Delta_1 = \text{span}\left\{\begin{bmatrix} 0 & -\frac{\mu'(S)XF}{V} \\ \frac{F}{V} & \frac{\mu'(S)XF}{YV} \\ 0 & 0 \end{bmatrix}\right\} \tag{3.27}$$

It is easy to see that the rank of Δ_1 is at most 2. Because the third row of this distribution contains zero elements any other matrices we get in the following steps will have the same rank:

$$\Delta_2 = \Delta_1 + \text{span}\{[f, \Delta_1] + [g, \Delta_1]\} = \text{span}\{g, [f, g], [f, [f, g]], [g, [f, g]]\} \tag{3.28}$$

$$\Delta_2 = \text{span}\left\{ \begin{bmatrix} \delta_{11}(x) & \delta_{12}(x) & \delta_{13}(x) & \delta_{14}(x) \\ \delta_{21}(x) & -\frac{1}{Y}\delta_{12}(x) & -\frac{1}{Y}\delta_{13}(x) & -\frac{1}{Y}\delta_{14}(x) \\ 0 & 0 & 0 & 0 \end{bmatrix} \right\} \tag{3.29}$$

where

$$\delta_{11}(x) = 0 \tag{3.30}$$

$$\delta_{12}(x) = -\mu'(S)\frac{XF}{V} \tag{3.31}$$

$$\delta_{13}(x) = \frac{XF}{V}(\mu'(S)\frac{F}{V} - \mu'^2(S)\frac{X}{Y} + \mu(S)\mu''(S)\frac{X}{Y} + \mu''(S)\frac{SF}{V}) \tag{3.32}$$

$$\delta_{14}(x) = -\mu''(S)\frac{XF^2}{V^2} \tag{3.33}$$

$$\delta_{21}(x) = \frac{F}{V} \tag{3.34}$$

and $\mu'(S) = \frac{\partial \mu(S)}{\partial S}$.

3.3.4 Minimal realization of fed-batch fermentation processes

As mentioned earlier, it was found in [9] that the fed-batch processes was not controllable on the whole state-space for input $u = F$. To solve this problem it showed a coordinates transformation that resulted a minimal realization of the model, which was already reachable. The original input-affine form was:

$$f(x) = \begin{bmatrix} \mu(S)X \\ -\frac{1}{Y}\mu(S)X \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\mu_{max}SX}{K_1+S+K_2S^2} \\ -\frac{\mu_{max}SX}{(K_1+S+K_2S^2)Y} \\ 0 \end{bmatrix} \quad , \quad g(x) = \begin{bmatrix} -\frac{X}{V} \\ \frac{S_f-S}{V} \\ 1 \end{bmatrix} \tag{3.35}$$

The work [9] gave a function for which it was easy to check from the model equations (3.35) that

$$\gamma(x) = V\left(-\frac{1}{Y}X - S + S_f\right) \tag{3.36}$$

is constant in time under any input i.e. $\frac{d}{dt}\gamma = \frac{\partial \gamma}{\partial x}\dot{x} = 0$ (see e.g. [8] or [10] for a complete control Lie-algebraic derivation).

Using the calculated γ function, it was possible to give a minimal state space realization of fed-batch fermentation processes. Since the reachability hypersurface defined by γ is two-dimensional, the minimal realization will contain two state variables (i.e. the input-to-state behavior of the system can be described by two differential equations). It's clear from the above that

$$\gamma(x(t)) = -\frac{1}{Y}X(t)V(t) - (S(t)V(t) - S_fV(t)) = \tag{3.37}$$
$$= -\frac{1}{Y}X(0)V(0) - (S(0)V(0) - S_fV(0)) = \gamma(x(0)).$$

Therefore we can express the volume V from the above equation in the following way:

$$V = \frac{\gamma(x(0))}{-\frac{1}{Y}X + S_f - S}, \quad -\frac{1}{Y}X + S_f - S \neq 0 \tag{3.38}$$

and the minimal state space model reads

$$\dot{x} = f_{min}(x) + g_{min}(x)u, \tag{3.39}$$

where

$$x = \begin{bmatrix} X \\ S \end{bmatrix}, \; f_{min}(x) = \begin{bmatrix} \frac{\mu_{max}SX}{K_1+S+K_2S^2} \\ -\frac{\mu_{max}SX}{(K_1+S+K_2S^2)Y} \end{bmatrix}, \; g_{min}(x) = \begin{bmatrix} \frac{\frac{1}{Y}X^2+X(S-S_f)}{\gamma(x(0))} \\ \frac{(-\frac{1}{Y}X+S_f-S)(S_f-S)}{\gamma(x(0))} \end{bmatrix} \tag{3.40}$$

We can see by expressing V from γ that the structure of the reaction rate function in f_{min} remained unchanged. It's also important to note that the function g_{min} in the minimal realization (3.40) depends on the initial state of the system.

This result of [9] will be used in analyzing the zero dynamics of fed-batch processes.

3.4 The zero dynamics of continuous fermentation processes

In order to analyze zero dynamics as it is described in section 2.6, we need to extend the original nonlinear state equation (3.4) with a nonlinear output equation

$$y = h(x) \tag{3.41}$$

where y is the output variable and h is a given nonlinear function. The zero dynamics of an input-affine nonlinear system containing two state variables can be analyzed using a suitable nonlinear coordinates transformation $z = \Phi(x)$:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} y \\ \lambda(x) \end{bmatrix} \tag{3.42}$$

where $\lambda(x)$ is a solution of the following partial differential equation (PDE):

$$L_g\lambda(x) = 0 \tag{3.43}$$

where $L_g\lambda(x) = \frac{\partial\lambda}{\partial x}g(x)$ i.e.

$$\frac{\partial\lambda}{\partial\bar{X}}g_1(x) + \frac{\partial\lambda}{\partial\bar{S}}g_2(x) = 0 \tag{3.44}$$

With the help of Φ we get a system in which the dynamics depend on only one state variable. Thus, the analysis of the zero dynamics will be easier.

In the case of the simple fermenter the above equation is the following:

$$\frac{\partial \lambda}{\partial \bar{X}} 0 + \frac{\partial \lambda}{\partial \bar{S}} \frac{F}{V} = 0 \tag{3.45}$$

from which we get

$$\frac{\partial \lambda}{\partial \bar{S}} = 0 \tag{3.46}$$

This means that λ is independent of $\bar{S}$ and we can choose an arbitrary continuously differentiable function F of $\bar{X}$ for λ:

$$\lambda = F(\bar{X}) \tag{3.47}$$

Then we can use the simplest possible coordinates transformation $z = \Phi(x)$ in the following form:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} y \\ \bar{X} \end{bmatrix} \tag{3.48}$$

3.4.1 Selecting the substrate concentration as output

Choose a linear function of the substrate concentration as output:

$$z_1 = y = k_s \bar{S} \tag{3.49}$$

where k_s is an arbitrary positive constant. Then the equation $z = \Phi(x)$ has the following form:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} k_s \bar{S} \\ \bar{X} \end{bmatrix} \tag{3.50}$$

From $z_2 = \lambda(x)$ (see in Eq. 3.42) the zero dynamics in the transformed coordinates can be computed as

$$\dot{z}_2 = \dot{\lambda}(x) = \frac{\partial \lambda(x)}{\partial x} \dot{x} = \frac{\partial \lambda(x)}{\partial x} (f(x) + g(x)u) \tag{3.51}$$

which gives

$$\dot{z}_2 = L_f \lambda(x) + L_g \lambda(x) u = L_f \lambda(x) = L_f \lambda(\Phi^{-1}(z)) \tag{3.52}$$

since $L_g \lambda(x) = 0$ (see Eq. 3.43). The inverse transformation $x = \Phi^{-1}(z)$ is given by

$$\begin{bmatrix} \bar{X} \\ \bar{S} \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{z_1}{k_s} \end{bmatrix} \tag{3.53}$$

The above equation is constrained by $y = k_s\bar{S} = z_1 = 0$. Then the zero dynamics of the system is given by the differential equation

$$\dot{z}_2 = L_f\lambda(\Phi^{-1}(z_1, z_2)) = L_f\lambda(\Phi^{-1}(0, z_2)) = (z_2 + X_0)(\mu(S_0) - \frac{F}{V}) \tag{3.54}$$

Using the values in Table 3.1 and Eq. (3.10) we get $\dot{z}_2 = 0$ in the optimal operating point. This result is very important because the biomass concentration in the reactor will not change when the output is constant. However, in a small neighborhood of the optimal operating point this can be unstable depending on the value of $\frac{F}{V}$. If this ratio is smaller than $\mu(S_0)$ the zero dynamics is unstable. With the condition $\frac{F}{V} > \mu(S_0)$ we can design a feedback controller which can stabilize the system in a small neighborhood of the optimal operating point.

3.4.2 Selecting the biomass concentration as output

Choose a linear function of the biomass concentration as output:

$$z_1 = y = k_x\bar{X} \tag{3.55}$$

where k_x is an arbitrary positive constant, then the equation $z = \Phi(x)$ has the following form:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} k_x\bar{X} \\ \bar{X} \end{bmatrix} \tag{3.56}$$

In this case the system has relative degree 2 according to definition 2.6. From this follows that the equation $\dot{\eta} = q(\xi, \eta)$ does not exists (see in Eq. 2.41) which means there is no zero dynamics. In this very fortunate case if we manage to stabilize the biomass concentration either by a full state feedback or by an output feedback controller then the overall system will be stable.

3.4.3 Selecting the linear combination of the biomass and the substrate concentrations as output

In this case the output is the linear combination of the biomass and the substrate concentrations:

$$z_1 = y = k_x\bar{X} + k_s\bar{S} \tag{3.57}$$

where k_x and k_s are arbitrary positive constants, then the inverse transformation $x = \Phi^{-1}(z)$ is given by

$$\begin{bmatrix} \bar{X} \\ \bar{S} \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{z_1 - k_x z_2}{k_s} \end{bmatrix} \tag{3.58}$$

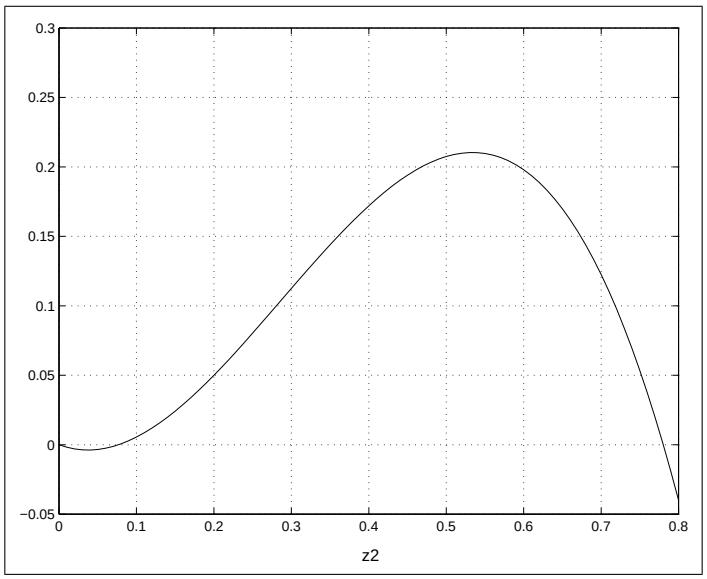

Figure 3.3: The numerator of the zero dynamics in the combined output case, $k_s = k_x = 1, F = 7\frac{l}{h}, X_0 = 0\frac{g}{l}, S_0 = 0\frac{g}{l}$. The equilibrium points are at $z_2 = 0\frac{g}{l},\ z_2 = 0.0768\frac{g}{l},\ z_2 = 0.7802\frac{g}{l}$.

Thus the zero dynamics in the transformed coordinates can be computed as

$$\dot{z}_2 = L_f\lambda(\Phi^{-1}(0, z_2)) = (z_2 + X_0)(\mu(-\frac{k_x}{k_s}z_2 + S_0) - \frac{F}{V}) \tag{3.59}$$

Where the denominator is 0 the zero dynamics has singular points:

$$z_2 = \frac{k_s}{2k_x}\frac{2K_2S_0 + 1 \pm \sqrt{1 - 4K_1K_2}}{K_2} \tag{3.60}$$

These are at $z_2 = 0.2491$ and $z_2 = 2.1882$ when the parameters k_x and k_s are supposed to be 1. The $\frac{k_s}{k_x}$ ratio determines the place of the singular points.

The equilibrium points of the zero dynamics are even more important. In this case when the output is the linear combination of the substrate and the biomass concentrations the equilibrium points are at

$$z_2 = -X_0 \tag{3.61}$$

and at

$$z_2 = \frac{k_s}{2k_x}\frac{-\mu_{max}V + F + 2FK_2S_0 \pm \sqrt{\mu_{max}^2V^2 - 2\mu_{max}VF + F^2 - 4F^2K_1K_2}}{FK_2} \tag{3.62}$$

These can be seen in Figure 3.3. At the first and at the third point the zero dynamics is stable and unstable at the second. The place of the equilibrium

points is determined by $\frac{k_s}{k_x}$ again. From a controller designer's point of view the stable zero dynamics are much to be desired: we can choose appropriate k_x and k_s to get a stable equilibrium point at the optimal operating point. When the output is zero then an output feedback controller will not sense any change and the system will stay at the desired point after all.

3.5 The zero dynamics of fed-batch fermentation processes

The coordinates transformation generator function Now again, we have two state variables in the minimal realization model (3.40). The PDE $\frac{\partial \lambda}{\partial x} g(x) = 0$ can be solved analytically for this case to obtain a function λ that satisfies the condition in (3.43)

$$\lambda(x) = \mathcal{F}\left(\ln\left(\frac{X}{S_f - S}\right)\right), \tag{3.63}$$

where $\mathcal{F}$ is an arbitrary function of the class C^1. Note that this solution requires the assumption $S_f - S > 0$ which always holds under physically meaningful operating and initial conditions. It's important to note that λ does not depend on γ containing the initial conditions.

3.5.1 Choosing the substrate concentration as output

Now we choose $z_1 = y = h(x) = S$. Because λ does not depend on γ, the following coordinates-transformation is valid for all initial states:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} S \\ \ln\left(\frac{X}{S_f - S}\right) \end{bmatrix} = \Psi(x) \tag{3.64}$$

This results in the inverse transformation:

$$\begin{bmatrix} X \\ S \end{bmatrix} = \begin{bmatrix} \exp(z_2)(S_f - z_1) \\ z_1 \end{bmatrix} = \Psi^{-1}(z). \tag{3.65}$$

Then the zero dynamics reads

$$\dot{z}_2 = \dot{\lambda} = \frac{\partial \lambda}{\partial x}\dot{x} = \frac{1}{X} f_1(x) + \frac{1}{S_f - S} f_2(x), \tag{3.66}$$

which gives

$$\dot{z}_2 = \mu(S)\left(1 - \frac{X}{(S_f - S)Y}\right) \tag{3.67}$$

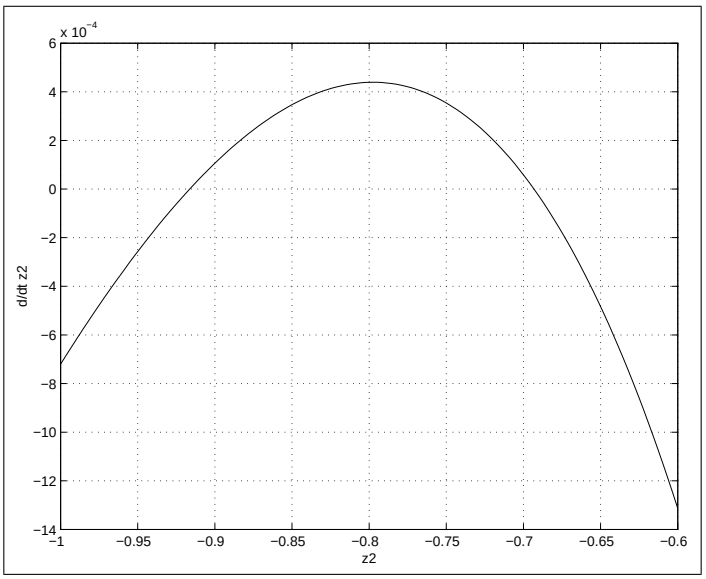

Figure 3.4: The zero dynamics in the transformed coordinates: fed-batch bioreactor, input: inlet feed flow rate, output: biomass concentration, $z_1 = 4\frac{g}{l}$

If we apply the inverse coordinates-transformation given by (3.65), the μ function remains in the expression and this gives

$$\dot{z}_2 = \mu(z_1)\left(1 - \frac{\exp(z_2)}{Y}\right) \tag{3.68}$$

It's easy to calculate that the equilibrium point of (3.68) is at $z_2^* = \ln(Y)$ independently of how z_1 (the substrate concentration) is set (if $z_1 > 0$). This means that if the substrate concentration is kept on any constant value (by manipulating the input feed flow rate), then the biomass concentration always converges to the corresponding equilibrium value on the $X - S$ plane independently of the reaction rate function μ.

3.5.2 Choosing the biomass concentration as output

In this case $z_1 = y = h(x) = X$ and the coordinates-transformation is:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} X \\ \ln\left(\frac{X}{S_f - S}\right) \end{bmatrix} = \Psi(x) \tag{3.69}$$

Its inverse transformation is in the following form:

$$\begin{bmatrix} X \\ S \end{bmatrix} = \begin{bmatrix} z_1 \\ \frac{\exp(z_2)S_f - z_1}{\exp(z_2)} \end{bmatrix} = \Psi^{-1}(z) \tag{3.70}$$

Then the zero dynamics in the new coordinates is:

$$\dot{z}_2 = \frac{\mu_{max}(\exp(z_2)S_f - z_1)\exp(z_2)(Y - \exp(z_2))}{(\exp(2z_2)(K_1 + S_f + K_2S_f^2) - \exp(z_2)(z_1 - 2K_2S_fz_1) + K_2z_1^2)Y} \quad (3.71)$$

The right hand side of eq. (3.71) is shown in Fig. 3.4 for a fixed value of $z_1 = 4\frac{g}{l}$. It is visible, that in this case the zero dynamics have two equilibria, one of which is independent of z_1 and is locally asymptotically stable ($z_2^* = \ln(Y) \approx -0.6931\frac{g}{l}$), and the other one is unstable. It means that a high gain feedback of the biomass concentration may move the biomass concentration itself out of the desirable range.

4 Controller design

Linear systems can be controlled easier than nonlinear ones. Many well known methods can be found in the literature (e.g. pole-placement, LQR controllers) [4]. However, most of the real-life systems are nonlinear and difficult to control. The most common way to solve this problem is either use linear techniques on locally linearized versions of nonlinear models or use model-based predictive control.

Fermentation processes in particular exhibit strong nonlinear characteristics and are known to be difficult to control. The investigated simple fermentation process is therefore used as a benchmark problem for advanced nonlinear analysis and control techniques.

Controllers of different type are designed and compared on the example of a simple fermenter near an optimal production operating point. Nonlinear analysis of controllability and zero dynamics presented in the previous chapter is used to investigate open-loop system properties, to explore the possible control difficulties and to design the system output to be used for control. Several type of controllers are tested including pole-placement and LQ controllers, and input-output linearization controllers.

4.1 Basis of stability analysis

It is well known that every feedback changes the stability properties of a system. This is the reason for using controllers on systems being unstable or having narrow stability region. The stability analysis used in this work is based on Lyapunov technique where the aim is to find a positive definite scalar-valued energy function *V(x)* that has negative definite time derivative in the whole operating region. Most often a general quadratic Lyapunov function is used in the form of

$$V(x) = x^T P x \tag{4.1}$$

with P being a positive definite symmetric quadratic matrix. This function is scalar-valued and positive definite everywhere. The stability region of an autonomous nonlinear system is then determined by the negative definiteness of its time derivative:

$$\frac{dV}{dt} = \frac{\partial V}{\partial x}\dot{x} = \frac{\partial V}{\partial x}\bar{f}(x) \tag{4.2}$$

where $\bar{f}(x) = f(x)$ in the open loop case (assuming zero input) and $\bar{f}(x) = f(x) + g(x)K(x)$ in the closed loop case where $K(x)$ is the static feedback.

The matrix P of the Lyapunov function is determined in the following way: after calculating the feedback gain K the autonomous system was linearized in the optimal operating point. This resulted matrix A. Then a positive definite symmetric matrix Q was chosen. After this the following equation was solved for P:

$$A^T P + PA = -Q \tag{4.3}$$

where $Q = I^{2\times 2}$. According to Lyapunov's criterion if A is a stability matrix, i.e $Re\{\lambda_i(A)\} < 0$ then P has to be a matrix with the necessary properties. This analysis cannot show the exact stability region only a subset of it.

4.2 Open loop system behavior

Before investigating different type of controllers and their performance we should examine the open loop system behavior. In the case of a simple continuous fermenter there are two state variables. Every variable has a minimum and a maximum value in the stability region of the system. Beyond these values the fermenter will be unstable which means the biomass concentration will decrease to zero and the substrate concentration will increase to S_{f_0}. However, from the nonlinearity the stability region will be a circle or an ellipse in the state-space and not a rectangle if one can believe. For centered state variables the minimum and maximum values are:

$$\begin{array}{rcl} \bar{X} & \in & [-0.03 \;\; \infty] \\ \bar{S} & \in & [-0.13 \;\; 0.1] \end{array} \tag{4.4}$$

An interesting value is the upper bound of $\bar{X}$. Infinity denotes that $\bar{X}$ can increase extremely high, but without growing the amount of substrate the biomass concentration will decrease to optimum.

4.3 Pole-placement controller

Pole-placement controllers are based on a very simple principle: when a linear system is stable then all of the eigenvalues of matrix A (see in Eq. 2.20) have strictly negative real parts, i.e. $Re\{\lambda_i(A)\} < 0, \;\forall i$. The eigenvalues are the poles of the system. If the system is not stable then we can compute a static feedback $u = -Kx$:

$$K = (\underline{\alpha} - \underline{a}) T_l^{-T} \mathcal{C}^{-1} \tag{4.5}$$

where $\underline{\alpha}$ is the new desired characteristic polynomial, $\underline{a}$ is the original characteristic polynomial, T_l denotes the Toeplitz matrix and finally $\mathcal{C}$ is the Kalman-

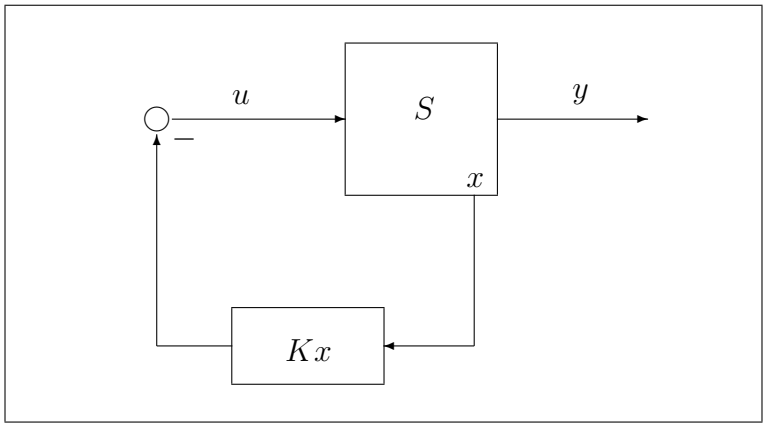

Figure 4.1: Control configuration of static linear feedback (pole-placement, LQ control)

controllability matrix (see Eq. 3.13). With this u the poles will have strictly negative real parts. The schematic control configuration of static linear feedback is shown in Figure 4.1.

Pole-placement controllers are based on full state feedback, which means they use the value of all state variables, so generally the feedback gain K does not contain zero values. Firstly, a controller like this is designed such that the poles of the linearized model of the closed-loop system are at $[-1 \ -2]^T$. The necessary feedback gain is

$$K = [4.2329 \ 1.7423] \tag{4.6}$$

For this feedback the stability region will be in the following rectangle:

$$\begin{array}{rcl} \bar{X} & \in & [-0.08 \ \ \infty] \\ \bar{S} & \in & [-0.2187 \ \ 0.07] \end{array} \tag{4.7}$$

Unfortunately, ∞ has not the same meaning as before. Simulations showed that starting the system from a value over the optimal biomass concentration it will not reach the optimum but a steady state value over it. The more deviation is to be controlled the more remaining error we get. For the substrate concentration the stability region is much wider. The controller can handle the case in which the substrate is missing from the reactor (e.g. at the start of the process). The Lyapunov function of pole-placement controller can be seen in Figure 7.1.

4.4 LQ controller

LQ-controllers are popular and widely used for process systems. They are known to stabilize any stabilizable linear time invariant system globally, that is over the entire state-space. This type of controller is designed for the locally linearized model of the process and minimizes the cost function

$$J(x(t), u(t)) = \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t))dt \tag{4.8}$$

where Q and R (the design parameters) are positive definite weighting matrices of appropriate dimensions. They can be called cost matrices, because Q is the punishment for deviating from the optimal operating point, and R means the input energy needed by the controller. The optimal input that minimizes the above functional is in the form of a linear full state feedback controller $u = -Kx$. The result for two different weighting matrix selections are investigated. By cheap control the deviation from the optimum really does not matter. We use small weights in contradiction to expensive control.

Cheap control

In this case the design parameters Q and R are selected to be $Q = I^{2\times 2}$ and $R = 1$. The resulting full state feedback gain is $K = [0.2361\ 0.2884]$. This gives the stability region in:

$$\begin{array}{rcl} \bar{X} & \in & [-0.04 \ \ \infty] \\ \bar{S} & \in & [-0.2 \ \ 0.09] \end{array} \tag{4.9}$$

At the upper bound of $\bar{X}$ ∞ denotes the same as before: the controller will set the biomass concentration toward to its optimal value but ever further as the deviation increases. The Lyapunov function of LQR cheap controller can be seen in Figure 7.2.

Expensive control

The weighing matrices were $Q = 10 \times I^{2\times 2}$ and $R = 1$. The full state feedback gain was $K = [1.7417\ 1.9683]$. Expensive control will have the same stability region as pole-placement controller, and the same behavior, as well.

$$\begin{array}{rcl} \bar{X} & \in & [-0.09 \ \ \infty] \\ \bar{S} & \in & [-0.2187 \ \ 0.07] \end{array} \tag{4.10}$$

The Lyapunov function of LQR expensive controller can be seen in Figure 7.3.

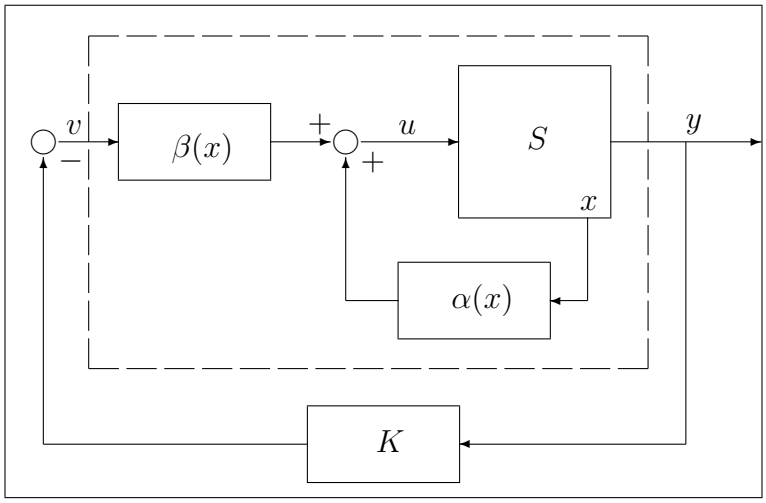

Figure 4.2: Control configuration of input-output linearization with stabilizing outer feedback.

4.5 Stabilization by feedback linearization

Applying feedback linearization, a nonlinear technique, the state variables are transformed into new coordinates z in which the system will have linear dynamics. Then, a simple linear static controller can be employed on the linearized system.

Exact linearization

Exact linearization can be done only when we find such an (artificial) output as the system's relative degree is equal to the number of the state variables according to definition 2.6. It means the system has no zero dynamics (see Eq. 2.41). The biomass concentration would be an appropriate choice (from section 3.4.2). This means we have found an output $\lambda(x) = \bar{X}$ which satisfies the conditions of exact linearization, that is a solution of [6]

$$L_g \lambda(x) = 0 \tag{4.11}$$

Note that the above equation is exactly the same as the equation (3.43) used for determining the coordinate transformation for analyzing zero dynamics of the system.

The components of state feedback $u = \alpha(x) + \beta(x)v$ for linearizing the system are calculated as

$$\alpha(x) \;=\; -\frac{L_f^2 \lambda(x)}{L_g L_f \lambda(x)} \tag{4.12}$$

$$\beta(x) = \frac{1}{L_g L_f \lambda(x)} \tag{4.13}$$

and the new coordinates are

$$z_1 = \lambda(x) = \bar{X} \tag{4.14}$$

$$z_2 = L_f \lambda(x) = \frac{\mu_{max}(\bar{S} + S_0)(\bar{X} + X_0)}{K_2(\bar{S} + S_0)^2 + \bar{S} + S_0 + K_1} - \frac{(\bar{X} + X_0)F}{V} \tag{4.15}$$

The state-space model in the new coordinates is

$$\dot{z}_1 = z_2 \tag{4.16}$$

$$\dot{z}_2 = v \tag{4.17}$$

which is linear and controllable. At first look the linearized model may seem quite simple but if we calculate $\alpha(x)$ and $\beta(x)$, needed by the linearizing state feedback, we can see that they are very complicated functions of x. Moreover, the second new coordinate z_2 depends on $\mu(S)$ which indicates that the coordinate transformation is sensitive with respect to uncertainties in the reaction rate expression. In fact, there is no use to apply exactly linearized model in this case because the feedback is hard to compute and evaluate numerically. The Lyapunov function of exact linearization controller can be seen in Figure 7.4.

Input-output linearization

Here we are looking for more simple and practically useful forms for linearizing the input-output behavior of the system. This method is like the former but here the relative degree is smaller than the number of state variables. This means there is zero dynamics which we calculated in section 3.4 for different outputs. It makes our task easier because we don't need to look for an artificial output $\lambda(x)$ to satisfy a strict condition and don't have to calculate so complex feedback. The static nonlinear full state feedback is calculated as

$$u = \alpha(x) + \beta(x)v = -\frac{L_f h(x)}{L_g h(x)} + \frac{1}{L_g h(x)} v \tag{4.18}$$

provided that $L_g h(x) \neq 0$ in a neighborhood of the operating point where v denotes the new reference input. As we will see the key point in designing such controllers is the selection output function $h(x)$ where the original nonlinear state equation (3.4) is extended by a nonlinear output equation $y = h(x)$ where y is the output variable. The control configuration of input-output linearization is shown in Figure 4.2.

Controlling the substrate concentration In this case the chosen output is $h(x) = \bar{S}$. The full state feedback is composed of the functions

$$\alpha(x) = -\frac{L_f h(x)}{L_g h(x)} \tag{4.19}$$

$$= (\bar{S} + S_0)\frac{\mu_{max}(\bar{X} + X_0)V + FYK_2(\bar{S} + S_0)^2 + FY(\bar{S} + S_0) + FYK_1}{(K_2(\bar{S} + S_0)^2 + \bar{S} + S_0 + K_1)YF} \tag{4.20}$$

$$\beta(x) = \frac{1}{L_g h(x)} = \frac{V}{F} \tag{4.21}$$

The new coordinates are

$$z_1 = \bar{S} \tag{4.22}$$

$$z_2 = -\frac{\mu_{max}(\bar{S} + S_0)(\bar{X} + X_0)}{(K_2(\bar{S} + S_0)^2 + \bar{S} + S_0 + K_1)Y} - \frac{(\bar{S} + S_0)F}{V} \tag{4.23}$$

The state-space model in the new coordinates is

$$\dot{z}_1 = v \tag{4.24}$$

$$\dot{z}_2 = (z_2 + X_0)(\mu(S_0) - \frac{F}{V}) = 0 \tag{4.25}$$

The outer loop for stabilizing the system is the following

$$v = -Kh(x) \tag{4.26}$$

where $K = [10\ 0]$ was applied. This kind of controller will have the following bounds for the stability region:

$$\begin{aligned} \bar{X} &\in [-0.04 \quad 0.09] \\ \bar{S} &\in [-0.03 \quad 0.08] \end{aligned} \tag{4.27}$$

These are very narrow limits for a relatively large feedback gain, but choosing a smaller gain will result narrower bounds. If the first component of K was over 10 then region would decrease and the accuracy, as well. The Lyapunov function of input-output controller controller with substrate concentration feedback can be seen in Figure 7.5.

Controlling the linear combination of the substrate and the biomass concentrations In this case $h(x) = k_x\bar{X} + k_s\bar{S}$ was chosen as output where the row vector $k = [k_x\ k_s]$ is calculated to get an equilibrium point at the optimal operating point. Then the functions α and β are given as

$$\alpha(x) = -\frac{\mu(\bar{X}+X_0)VY - (\bar{X}+X_0)FY - \mu(\bar{X}+X_0)V - (\bar{S}+S_0)FY}{YF} \tag{4.28}$$

$$\beta(x) = \frac{V}{F} \tag{4.29}$$

The transformed coordinates are

$$z_1 = k_x\bar{X} + k_s\bar{S} \tag{4.30}$$

$$z_2 = L_f h(x) = k_x f_1(x) + k_s f_2(x) \tag{4.31}$$

where $f_1(x)$ and $f_2(x)$ are the components of f. The state-space model in the new coordinates is

$$\dot{z}_1 = v \tag{4.32}$$

$$\dot{z}_2 = (z_2 + X_0)(\mu(-\frac{k_x}{k_s}z_2 + S_0) - \frac{F}{V}) \tag{4.33}$$

The value of $k = [k_x\ k_s]$ was [0.02 5.23] and in the outer loop a negative feedback with the gain $K = [10\ 10]$ was applied. This resulted the stability region in:

$$\begin{aligned} \bar{X} &\in [-0.35\ \ 0.25] \\ \bar{S} &\in [-0.2187\ \ 0.02] \end{aligned} \tag{4.34}$$

The feedback, which controls the linear combination of the biomass and the substrate concentrations as output, gives us a very wide stability region. In spite of this fortunate property, after all transient have settled the remaining error will be too much. The Lyapunov function of input-output controller with the linear combination of biomass and substrate concentrations feedback can be seen in Figure 7.6.

4.6 Selecting the appropriate controller

Firstly, controllers based on the local linearized model will be compared. This is based on investigating the stability region, time and the input energy needed by the controller but only for the biomass concentration because the main goal of the fermentation process is to maximize XF, where the input/output flow rate F is constant. The time needed by the controller is the time T when the biomass concentration reaches the 95% of its steady state value. The used input energy is calculated as $E = \sqrt{\int_0^T |u(t)|^2\, dt}$.

Pole-placement and LQR with expensive Q weighting matrix had the same stability region and showed the same behavior. However, the latter needed

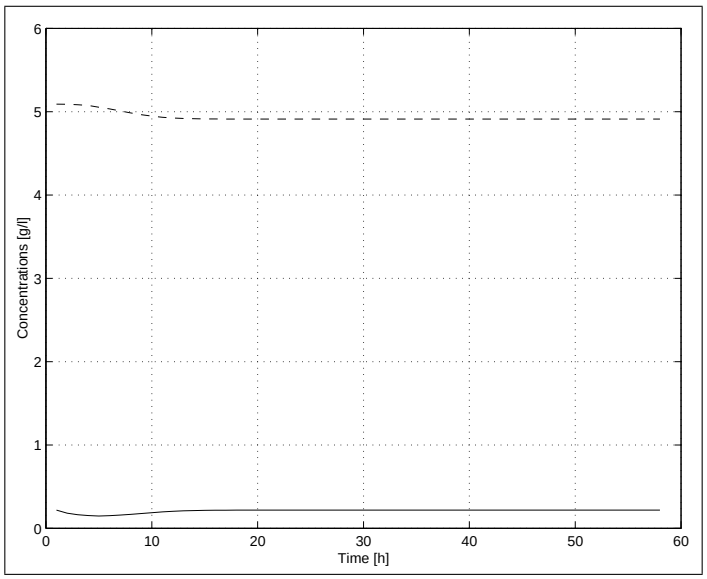

Figure 4.3: Biomass and substrate concentrations controlled by LQR cheap controller, $X(0) = 5.0907\frac{g}{l},\ S(0) = 0.2187\frac{g}{l}$

more time and energy. From simulations we get that cheap control is better than expensive LQR or pole-placement control. It drives the biomass concentration closer to its optimum than the other ones. Its accuracy is about ± 0.01 when $\bar{X} \in [-0.04\ 0.2]$ (see figure 4.3). The reason for this is the small gain in the feedback. Important disadvantages of the cheap control are the smaller stability bounds below zero and it needs more time and energy than the expensive LQR or the pole-placement controller.

From the comparison of controllers based on input-output feedback linearization follows that the linear combination of biomass and substrate concentrations seems to be a better choice as an output.

To get the appropriate controller an LQR cheap and a feedback linearization controller must be investigated. Although the latter has much wider stability region than the open loop system or any other fed back system, its remaining error for biomass concentration is greater than the LQR controller's, it can be form 0.03 up to 0.15. When the substrate concentration changes then the LQR controller needs less time to stabilize the system. The used energy is generally equal. The only advantage of the feedback linearization controller is the size of the stability region.

Conclusion: if one wanted to use a controller based on the some way linearized system, then the value of both state variables should be used and the best controller choice would be an LQR cheap controller.

5 Program description

In this chapter will be described how the MAPLE implementation of Isidori's algorithm (see section 2.5.3) works. The program structure is like in a high-level program language: often used program parts are written in functions which can be found before the main part. Some among them is the implementation of a MATLAB function [2]. The only reason for this was to help that, who wants to implement this MAPLE program in MATLAB. However, there is an important difference: MAPLE handles vectors in row vector form [1].

5.1 Function descriptions

Size: This is one of the implemented MATLAB functions. It calculates the size of the input vector or matrix. Firstly, it determines the number of rows by increasing the variable i until h is not an error message. If it is then i reached the maximum at that direction. The output vector is [*row column*].

Lie: The function *Lie* calculates the Lie-products described in section 2.4. From the definition of this product follows that the order of the input vector is important. It returns a vector *Lie_prod*.

Projector: It will choose a row from the distribution Δ. The input is the number of the row we want to project. This function is used at the calculation of Lie-products.

Compare: The selection of the old and the new Lie-products will be done by this function. It returns true when the input vectors are identical and false otherwise.

Zeros: This is the another MATLAB function implemented in MAPLE. It generates as long zero vector as large the input is. Zero vectors are used for marking the old Lie-products.

5.2 The main part of the program

At first we give the inputs to the program: the function set which contains only the name f and g. If g is a matrix then $g_1, \ldots, g_n$ should be written. Then we give the exact form of these vector functions. At last, we give the state variable x as a vector of $x_1, \ldots, x_n$.

After this, the program determines the number of the state variables, Δ_0 from $g_1, \ldots, g_n$ and the rank of Δ_0.

According to the second step at the algorithm (see Eq. 2.36) a new distribution is calculated. Firstly, the $[g_i, \Delta_{k-1}]$, $i = 0, \ldots, n$ Lie-products are calculated and stored in vector *product_ temp*. Because a distribution is a set of vector functions, an element can be only once in it. That's why the Lie-products which are already in Δ_k will be replaced by zero vectors. Each turn after the new Lie-products from $[g_i, \Delta_{k-1}]$ are selected, the program will gather them either into the beginning of *product_ temp* if $i = 1$ or after the last new Lie-product calculated in the previous turn $(i - 1)$.

When *product_ temp* contains all the new Lie-products which are not in Δ_{k-1}, the program builds a new matrix from them and from Δ_{k-1}. At first, it will copy the previous distribution into Δ_k and then will put the elements of *product_ temp* after it.

The third step of Isidori's algorithm (see Eq. 2.39) is implemented such a way that the program calculates the rank of Δ_k. If it is greater than the rank of Δ_{k-1} then this will be the new distribution and comes the next turn, otherwise Δ_{k-1} will be the result. At last, the program transpose Δ_{k-1} to get a form we know and use.

6 Conclusions

I inspected in this diploma work some of the problems, that can occur in controlling nonlinear fermentation processes: whether the system is controllable or not and how the state variables behave when the output does not change, i.e. the zero dynamics. Then, using these results I determined an appropriate controller for the system.

Firstly, I investigated the controllability of the locally linearized model of the continuous system because if this property does not hold then the original nonlinear system is probably not controllable, as well. Fortunately, I got favorable result. This method cannot be done in the case of fed-batch fermenter because there was no steady state point around which the linearization could be done. Calculating the reachability distribution for nonlinear systems showed that only the continuous process is controllable. After this, I calculated the singular points of the distribution which should be avoided during control. I got that this points are far from the optimal operating point.

Before designing and comparing different type of controllers I had to get to know how the system behaves when its output does not change so the controller does not sense anything. The analysis of zero dynamics resulted in two outputs for which the dynamics could be stable, and one output for which there was no zero dynamics. These results will be published in a European Control Conference (ECC) publication in 2003 [11].

Finally, I designed and simulated six different controllers: pole-placement, LQ cheap, LQ expensive, exact linearized controller and input-output linearized controller for two type of outputs. I determined the stability region of them, time and energy needed of control. Then I chose the one that reliably stabilizes the whole process. This was the LQ cheap controller.

In section 5 a short description of a MAPLE realization of Isidori's algorithm is written. This program is not only for fermentation processes, it works for any type of systems. Only the necessary inputs should be given.

7 Appendix A

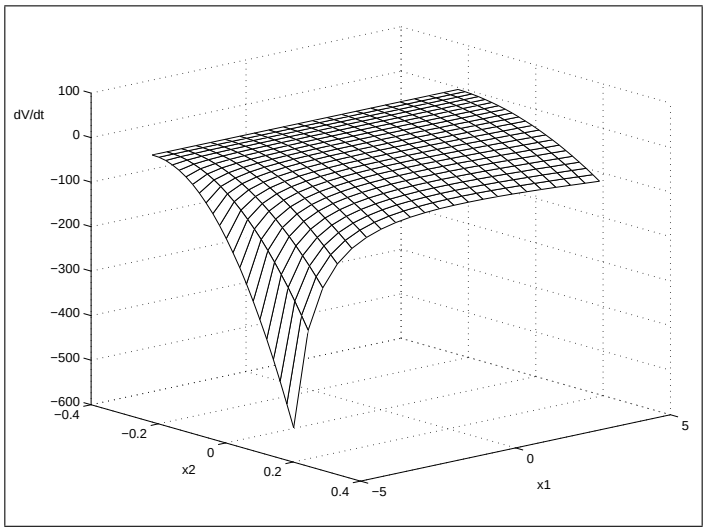

Figure 7.1: Time derivative of the Lyapunov function of the Pole-placement controller

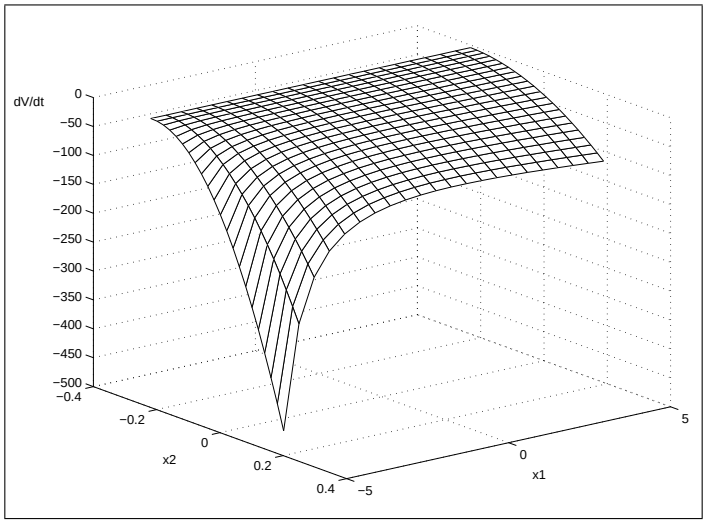

Figure 7.2: Time derivative of the Lyapunov function of the LQR cheap controller

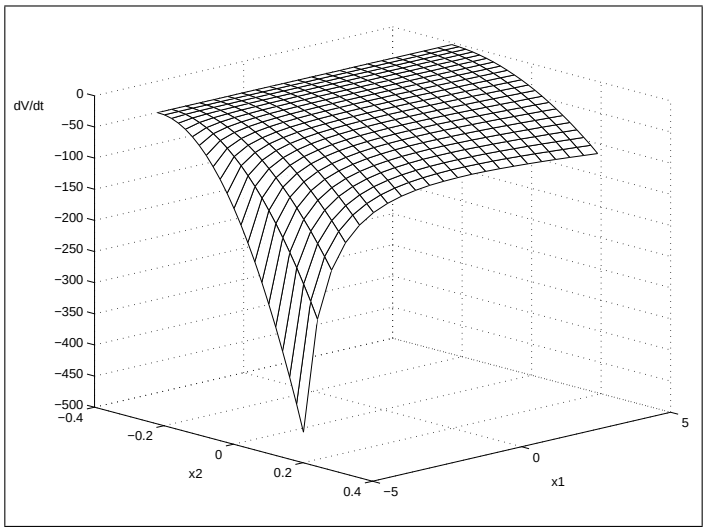

Figure 7.3: Time derivative of the Lyapunov function of the LQR expensive controller

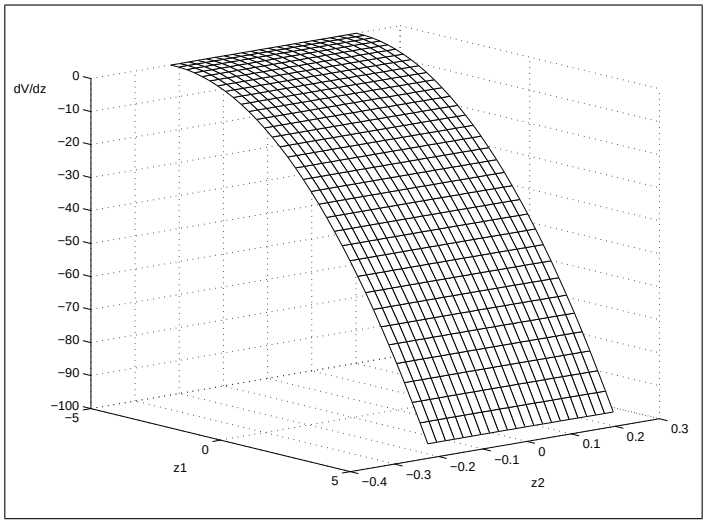

Figure 7.4: Time derivative of the Lyapunov function of the exact linearization controller

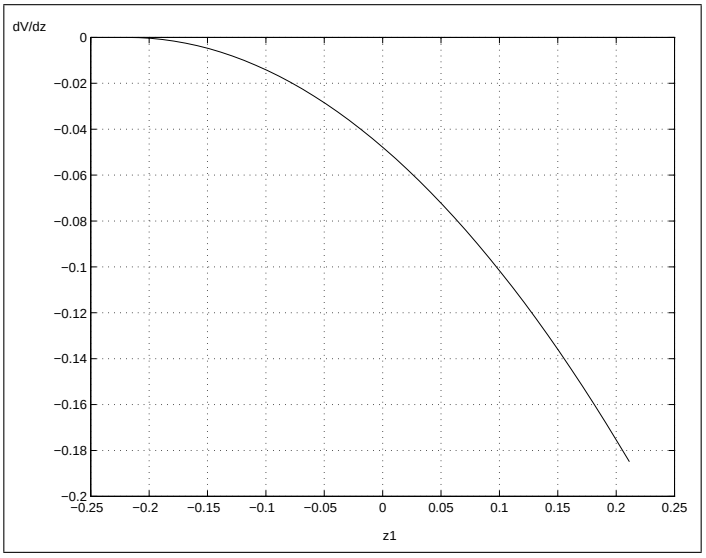

Figure 7.5: Time derivative of the Lyapunov function of the input-output controller with substrate feedback

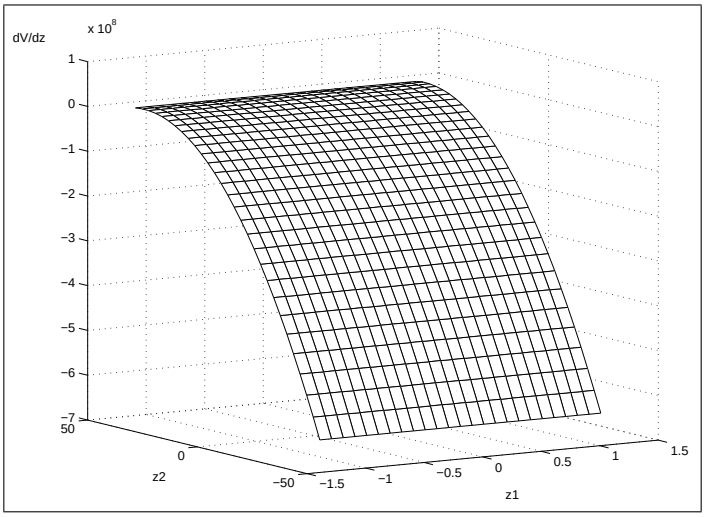

Figure 7.6: Time derivative of the Lyapunov function of the input-output controller with the linear combination of biomass and substrate concentration feedback

8 Appendix B

```
with(linalg);
%This function calculates the size of a vector or matrix.
size:=proc(a) local h,i,row,column;
   for i while h<>lasterror do
       h:=traperror(a[i,1]);
       if h=lasterror then
         if i=1 then
           row:=1;
         else
           row:=i-1;
         fi;
       fi;
   od;
   unassign('i','h');
   for i while h<>lasterror do
       if row=1 then
         h:=traperror(a[i]);
       else
         h:=traperror(a[1,i]);
       fi;
       if h=lasterror then
         column:=i-1;
         RETURN([row,column]);
       fi;
   od;
end:
%Lie-product calculator function
Lie:=proc(a,b) local Lie_prod,i; global var_num;
    Lie_prod:=evalm(multiply(jacobian(b,x),a)-multiply(jacobian(a,x),b));
    for i to var_num do
        Lie_prod[i]:=simplify(Lie_prod[i]);
    od;
    RETURN(Lie_prod);
end:
%A row-selector function. The input is the number of the row we want to select.
projector:=proc(a) local row,i; global Delta;
    row:=array(1..var_num,[]);
    for i to var_num do
        row[i]:=Delta[a,i];
    od;
    RETURN(row);
end:
%This function compares two vectors
```

```
compare:=proc(a,b) local i,k;
    k:=0;
    for i to size(a)[2] do
        if a[i]-b[i]=0 then
          print(i,a[i],b[i]);
          k:=k+1;
        fi;
    od;
    if k=size(a)[2] then
      RETURN(true);
    else
      RETURN(false);
    fi;
end:
%This function generates a vector with length a containing zeros
zeros:=proc(a) local i,b;
   b:=array(1..a);
   for i to a do
       b[i]:=0;
   od;
   RETURN(b);
end:
%Giving the inputs
function_set:=vector([f,g]);
f:=vector([-x1/(x2^ 2+2*x1+3),4*x1/(x2^ 2+2*x1+3)]);
g:=vector([-x1,-x2]);
x:=vector([x1,x2]);
var_num:=size(x)[2];
%Calculating Delta0
Delta:=array(1..size(function_set)[2]-1,1..var_num):
for i to size(Delta)[1] do
    for j to var_num do
        Delta[i,j]:=function_set[i+1][j];
    od;
od;
print(Delta);
old_rank:=rank(Delta);
may_end:=false;
%Calculating the other distributions
for p while may_end<>true do
    l:=1:
    %The vector product_temp will contain the Lie-products
    product_temp:=array(1..size(function_set)[2]*size(Delta)[1]):
    for i to size(function_set)[2] do
      k:=l:
```

```
  %We get an element from function_set and compute its Lie-products with Delta.
  %Exactly k pieces.
  for j to size(Delta)[1] do
      product_temp[k]:=Lie(function_set[i],projector(j));
      k:=k+1;
  od;
  %The Lie_products which are already in Delta will be zeroed. Product_temp will
  %contain zero vectors and Lie_products which are not in Delta
  for n from l to k-1 do
    for j to size(Delta)[1] do
      if compare(product_temp[n],projector(j)) then product_temp[n]:=zeros(var_num);
        j:=size(Delta)[1];
      fi;
    od;
  od;
  %The new Lie_products will be collected and will be put to the beginning of
  %product_temp. The variable l will point at the next position after this list.
  %When i reaches the end of function_set the product_temp will contain only
  %new Lie_products.
  for n from l to k-1 do
      if compare(product_temp[n],zeros(var_num)) then
      else
        product_temp[l]:=product_temp[n];
        l:=l+1;
      fi;
  od;
od;
%Generating a new distribution which size is the sum of the size of Delta and the
%number of the new Lie_products
Delta_temp:=array(1..size(Delta)[1]+l-1);
for i to size(Delta)[1] do
    Delta_temp[i]:=projector(i);
od;
for j to l-1 do
    Delta_temp[j+i-1]:=product_temp[j];
od;
%Delta_temp is a vector containing vectors but it's not a matrix. Delta_new will
%be a matrix.
Delta_new:=array(1..size(Delta)[1]+l-1,1..var_num):
%Giving value to m(S) so that Maple could compute rank
m(S):=mmax*S/(K2*S^ 2+S+K1):
for i to size(Delta_new)[1] do
    for j to var_num do
        Delta_new[i,j]:=Delta_temp[i][j];
    od;
```

```
    od;
    %Calculating the rank.
    new_rank:=rank(Delta_new);
    %To get a m(S) independent result we delete the value of m(S).
    unassign('m(S)');
    %Recomputing Delta_new for symbolic computing.
    for i to size(Delta_new)[1] do
        for j to var_num do
            Delta_new[i,j]:=Delta_temp[i][j];
        od;
    od;
    %Inspecting the ranks whether they changed or not.
    if old_rank=new_rank then
      may_end:=true;
    else
      Delta:=array(1..size(Delta)[1]+l-1,1..var_num):
      for i to size(Delta)[1] do
          for j to var_num do
              Delta[i,j]:=Delta_temp[i][j];
          od;
      od;
      old_rank:=new_rank;
    fi;
od;
%Transposing the result to get a known form of the distribution.
Delta:=transpose(Delta);
```

References

[1] *Maple V Release 5 Help*, 1997.

[2] *MATLAB 5.3 User's Guide*, 1999.

[3] C. I. Byrnes and A. Isidori. Asymptotic stabilization of minimum-phase nonlinear systems. *IEEE Transactions on Automatic Control*, AC-36:1122–1137, 1991.

[4] Katalin Hangos, József Bokor, and Miklós Gerzson. *Computer Controlled Systems*. University of Veszprém, Veszprém, 1994.

[5] Katalin Hangos, József Bokor, and Gábor Szederkényi. Analysis and control of nonlinear process systems. Technical report, Systems and Control Laboratory, SCL-1-2001, Budapest, MTA-SZTAKI, 2001.

[6] Alberto Isidori. *Nonlinear Control Systems*. Springer, Berlin, 1995.

[7] Ch. Kuhlmann, I. D. L. Bogle, and Z. S. Chalabi. Robust operation of fed batch fermenters. *Bioprocess Engineering*, 19:53–59, 1998.

[8] R. Mahadevan, S. K. Agrawal, and F. J. Doyle III. Differential flatness based nonlinear predictive control of fed-batch bioreactors. *Control Engineering Practice*, 9:889–899, 2001.

[9] Gábor Szederkényi. *Grey-box approach for the diagnosis, analysis and control of nonlinear process systems.* PhD thesis, University of Veszprém, 2002.

[10] G. Szederkényi, M. Kovács, and K. M. Hangos. Reachability of nonlinear fed-batch fermentation processes. *International Journal of Robust and Nonlinear Control*, 12:1109–1124, 2002.

[11] G. Szederkényi, T. Schné, and K. M. Hangos. Zero dynamics of continuous and fed-batch bioreactors. In *European Control Conference - ECC 2003*, 2003.