Veszprémi Egyetem
Számítástudomány Alkalmazása Tanszék
Műszaki Informatika Szak

DIPLOMADOLGOZAT

Stability of Generalized Lotka-Volterra Systems

Magyar Attila

Témavezető: Dr. Hangos Katalin

2004

Nyilatkozat

Alulírott Magyar Attila, 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, 2004. május 13.

Magyar Attila

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Dr. Hangos Katalinnak a rengeteg jó tanácsért és támogatásért, melyekkel munkámat segítette.
Hálával tartozom családomnak, hogy tanulmányaim során mellettem álltak.
Köszönet illeti Szederkényi Gábort, akitől szintén sok segítséget kaptam, továbbá mindazokat, akik bármilyen módon elősegítették eme diplomadolgozat megszületését.

Summary

The subject of this diploma work is the stability analysis of the quasi-polynomial and Lotka-Volterra representations of process systems. My task is to analyze local, and global stability properties of the quasi-polynomial and Lotka-Volterra forms of a fermentation process with non-monotonous reaction kinetics, and compare the obtained results with the stability of the original fermenter.

In the course of transforming process systems into quasi-polynomial form, we introduce algebraic equations to the system and we obtain a system with greater dimension by differentiating them but the resulting model will be in quasi-polynomial (QP) form. In the course of embedding a QP system model into a Lotka-Volterra system (which is simply done by a matrix multiplication) we also raise the dimension of the state space with algebraic equations.

In the case of both quasi-polynomial and Lotka-Volterra system models, the Jacobian matrix can easily be expressed in terms of the system matrices. During the local stability analysis it turned out, that the eigenvalues of the Jacobian matrix are equal to the original process systems' ones in both cases, and the eigenvalues according to the algebraic equations are zeros. It is in accordance with the experience that the trajectories of QP and Lotka-Volterra systems composed from non quasi-polynomial process systems roam a lower dimension manifold of the state space.

For global stability analysis I used a known Lyapunov function family. Its parameters can be found by solving a linear matrix inequality. For Lotka-Volterra systems originated from non quasi-polynomial process systems it is a non strict linear matrix inequality which is not trivial to solve. I have proposed and tested two of such methods.

Keywords: *quasi-polynomial systems, Lotka-Volterra systems, stability analysis, linear matrix inequality.*

Tartalmi összefoglaló

Diplomamunkám témája nemlineáris folyamatrendszerek kvázipolinom, illetve Lotka-Volterra reprezentációjának stabilitásvizsgálata. Feladatom az volt, hogy egy nem monoton reakciókinetikájú fermentációs folyamat kvázipolinom, illetve Lotka-Volterra alakjának lokális, illetve globális stabilitási tulajdonságait vizsgáljam, és a kapott eredményeket összehasonlítsam az eredeti fermentor stabilitásával.
A folyamatmodellek kvázipolinom alakra transzformálása során algebrai egyenleteket adunk a rendszerhez, és ezeket differenciálva egy megnövekedett dimenziójú, de már kvázipolinom (QP) alakú rendszert kapunk. A QP rendszer modell Lotka-Volterra modellé alakítása folyamán (ami egy egyszerű mátrix-szorzással végezhető el) ugyancsak algebrai egyenletekkel növeljük az állapottér dimenzióját.
Mind kvázipolinom, mind Lotka-Volterra rendszerek esetén a Jakobi mátrixot egyszerűen ki lehet fejezni a rendszert leíró mátrixok segítségével. A lokális stabilitásvizsgálat során kiderült, hogy a Jakobi mátrix sajátértékei mindkét esetben megegyeznek az eredeti folyamatrendszer Jakobi mátrixának sajátértékeivel, az algebrai egyenleteknek megfelelő sajátértékek pedig nullák lesznek. Ez összhangban van azzal a tapasztalattal, hogy a nem kvázipolinom folyamatrendszerekből képzett QP, ill. Lotka-Volterra rendszerek trajektóriái az állapottérnek csak egy kisebb (az eredeti rendszer dimenziójával megegyező) dimenziójú sokaságot járják be.
A globális stabilitásvizsgálat során egy ismert Ljapunov-függvény családot használtam, ennek a paramétereit lineáris mátrix egyenlőtlenség megoldásával lehet meghatározni. Olyan Lotka-Volterra rendszerek esetén, melyeket egy nem kvázipolinom folyamatrendszerből hoztunk létre, ez egy nem szigorú lineáris mátrix egyenlőtlenséghez vezet, aminek a megoldására módszereket javasoltam.

Kulcsszavak: *kvázipolinom rendszer, Lotka-Volterra rendszer, stabilitásvizsgálat, lineáris mátrix egyenlőtlenség.*

Contents

List of Figures

1 Introduction

1.1 Motivation

The global stability analysis of general nonlinear systems needs special skills because the construction of a suitable Lyapunov function is far from being trivial. Therefore special canonical forms enabling to construct a Lyapunov function are of great theoretical and practical importance.

Lotka-Volterra (LV) and Generalized Lotka-Volterra (GLV) or quasi-polynomial (QP) models have proved to be one of the candidates for generally applicable canonical forms of nonlinear ordinary differential equation models because a wide class of such models can be represented in these forms [2]. Moreover, methods for investigating both local and global stability of Generalized Lotka-Volterra models have also been reported [4].

Lotka-Volterra systems [1] are widely used in certain areas of science e.g. in population biology. Structure and algebraic properties of generalized and classical Lotka-Volterra forms have been dealt with by Hernandez-Bermejo and Fairen in [1-2]. Sufficient conditions for the existence of Lyapunov functions for quasi-polynomial systems have been given by Figueiredo, Gléria and Rocha Filho [4].

The results related to Lotka-Volterra systems have been generalized and then used to describe and analyze nonlinear process systems [5]. The Lotka-Volterra equations are used as a unifying form because a large number of process systems can be transformed into it.

The objective of this work is to use quasi-polynomial and Lotka-Volterra representation of nonlinear process systems for local and global stability analysis.

1.2 Outline of this work

First of all basic concepts of quasi-polynomial (QP) and Lotka-Volterra (LV) systems will be presented in chapter 2. Definitions for input-affine QP models, linear matrix inequalities and zero dynamics are also presented in this chapter.

Chapter 3 deals with stability notions and stability properties. First, local and global stability analysis will be performed on both quasi-polynomial and Lotka-Volterra models. In section 3.2.3 and section 3.2.4 some computational problems of

the global stability analysis will be discussed: in order to find a Lyapunov function we have to solve a *linear matrix inequality*, this task can be accomplished e.g. with the MATLAB® LMI Control Toolbox [8], or with an other numerical algorithm [11]. Afterwards stability analysis of zero dynamics is investigated, and the final issue of this chapter will be the estimation of the stability region for quasi-polynomial and Lotka-Volterra systems.

In chapter 4 a case study of a simple fermentation process is presented with the help of which all the previously introduced techniques and methods are demonstrated.

2 Basic Definitions

Some basic notions on generalized and canonical Lotka-Volterra representation of input-affine nonlinear systems are briefly summarized in this section. Related notions and results on linear matrix inequalities and on zero dynamics are also described here.

2.1 Generalized Lotka-Volterra Systems

There are many unifying representations for nonlinear ODEs. With these methods both the structural simplicity and the dimension of the system increase. The canonical form of a unifying representation has been introduced by Brenig and Goriely [1], and called *generalized Lotka-Volterra (GLV) form, or quasi-polynomial (QP) form*:

$$\dot{x_i} = x_i\Big(\lambda_i + \sum_{j=1}^{m} A_{ij} \prod_{k=1}^{n} x_k^{B_{jk}}\Big), \tag{1}$$
$$i = 1, \ldots, n, \qquad m \geq n$$

where A and B are $n\ x\ m$, $m\ x\ n$ real matrices, $\lambda \in R^n$ vector, $x_i > 0, \quad i = 1, \ldots, n.$

2.1.1 Embedding into quasi-polynomial form

A set of nonlinear ODEs can be embedded to QP form if it satisfies two important requirements [2].

1. The set of nonlinear ODEs should be in the form:

$$\dot{x_s} = \sum_{i_{s1},\ldots,i_{sn},j_s} a_{i_{s1}\ldots i_{sn}j_s} x_1^{i_{s1}} \ldots x_n^{i_{sn}} f(\overline{x})^{j_s}, \tag{2}$$
$$x_s(t_0) = x_s^0, \qquad s = 1, \ldots, n$$

where

$$a_{i_{s1}\ldots i_{sn}j_s}, i_{s1}, \ldots, i_{sn}, j_s \in R, \quad s = 1, \ldots, n,$$

and $f(\overline{x})$ is some scalar valued function, which is not reducible to *quasi-monomial form* containing terms in the form of
$\prod_{k=1}^{n} x_k^{\Gamma_{jk}},\ j = 1, \ldots, m$ with Γ being a real matrix.

2. Furthermore, it is required that the partial derivatives of the system (2) fulfil:

$$\frac{\partial f}{\partial x_s} = \sum_{e_{s1},\ldots,e_{sn},e_s} b_{e_{s1}\ldots e_{sn}e_s} x_1^{e_{s1}} \ldots x_n^{e_{sn}} f(\overline{x})^{e_s} \tag{3}$$

where

$$b_{e_{s1}..e_{sn}e_s}, e_{s1}, \ldots, e_{sn}, e_s \in R, \quad s = 1, \ldots, n.$$

The embedding is performed by introducing some *new auxiliary variable*

$$y = f^q \prod_{s=1}^{n} x_s^{p_s}, \qquad q \neq 0. \tag{4}$$

With the above new variables we have introduced algebraic equations. Differentiating them, we get the quasi-polynomial form of the original equations (2):

$$\dot{x_s} = \left(x_s \sum_{i_{s1},\ldots,i_{sn},j_s} \left(a_{i_{s1}\ldots i_{sn}j_s} y^{j_s/q} \prod_{k=1}^{n} x_k^{i_{sk}-\delta_{sk}-j_s p_k/q} \right) \right), \qquad s = 1, \ldots, n \tag{5}$$

where $\delta_{sk} = 1$ if $s = k$ and 0 otherwise. In addition, a new quasi-polynomial ODE appears for the new variable y:

$$\begin{aligned} \dot{y} \;=\; & y \Bigg[\sum_{s=1}^{n} \Big(p_s x_s^{-1} \dot{x_s} + \\ & + \sum_{\substack{i_{s\alpha},j_s \\ e_{s\alpha},e_s}} a_{i_{s\alpha},j_s} b_{e_{s\alpha},e_s} q y^{(e_s+j_s-1)/q} \times \\ & \times \prod_{k=1}^{n} x_k^{i_{sk}+e_{sk}+(1-e_s-j_s)p_k/q} \Big) \Bigg], \qquad \alpha = 1, \ldots, n. \end{aligned} \tag{6}$$

It is important to observe that the embedding is not unique, because we can choose the parameters p_s and q in Eq. (4) in many different ways: the simplest is to choose $(p_s = 0,\ s = 1, ..., n;\ q = 1)$

2.2 Canonical Lotka-Volterra Systems

The set of quasi-polynomial models is closed under a special nonlinear transformation, the so-called quasi-monomial transformation [1].

2.2.1 Quasi-monomial transformations

A quasi-monomial transformation is defined as

$$x_i = \prod_{k=1}^{n} \hat{x}_k^{C_{ik}}, \qquad i = 1, \ldots, n \tag{7}$$

where C is an arbitrary invertible matrix. The GLV matrices modify to $\hat{B} = B \cdot C$, $\hat{A} = C^{-1} \cdot A$, $\hat{\lambda} = C^{-1} \cdot \lambda$, and the transformed set of equations will also be in GLV form. The above family of systems is split into classes of equivalence [2] according to the values of the products $M = B \cdot A$ and $\Lambda = B \cdot \lambda$.

2.2.2 LV form

The concept of Lotka-Volterra (LV) form is also based on quasi-monomial transformations. Quasi-monomial transforms define classes of equivalence, in which the products

$$M = B \cdot A \quad \text{and} \quad \Lambda = B \cdot \lambda$$

are the invariants of a class of equivalence. In addition, the quasi-monomials, which are exactly the LV variables, are also invariants of the equivalence classes.

Then the classical LV form gives the representative elements of these classes of equivalence. If $rank(B) = n$, then the set of ODEs in (1) can be embedded into the following m-dimensional set of equations, the so called Lotka-Volterra model:

$$\dot{z_\ell} = \lambda_{LV,\ell} z_\ell + z_\ell \sum_{i=1}^{m} A_{LV,\ell i} z_i, \qquad \ell = 1, \ldots, m \tag{8}$$

where $A_{LV} = B \cdot A$, $\lambda_{LV} = B \cdot \lambda$ and each z_ℓ represents a so called quasi-monomial:

$$\prod_{k=1}^{n} x_k^{B_{jk}}, \qquad j = 1, \ldots, m. \tag{9}$$

It is noticeable that the LV form is a special case of the GLV system form, where the exponent matrix B is equal to a unit matrix of size m ($B = I$), where m is the number of the different quasi-monomials of the GLV system.

2.3 Unambiguity of the LV form

As it was mentioned above we have a degree of freedom in the choice of the algebraic variable(s) (4) of the quasi-polynomial system.

It is a fundamental question whether the different QP models obtained by different algebraic variable introducing belong to the same GLV class of equivalence. It was shown in [1] that all these QP systems lead to a unique LV model. In this section it is shown that an LV model is only unique in a restricted sense: not only

the permutation of the variables is enabled, but also the re-scaling of the variables is permitted.

2.3.1 Permutation of LV variables

Let z be the vector containing the LV variables:

$$z = [z_1, z_2, \cdots, z_n] \tag{10}$$

Then multiplying z with a permutation matrix P (P can be transformed to unit matrix with row and column changes) represents variable swapping:

$$\hat{z} = P \cdot z, \tag{11}$$

Any permutation can be performed step-by-step i.e. only two variables are swapped at one step. The permutation matrix P that permutes the i-th and the j-th variables is as follows:

$$\hat{z} = P \cdot z = \begin{bmatrix} 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_i \\ \vdots \\ z_j \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_j \\ \vdots \\ z_i \\ \vdots \\ z_n \end{bmatrix} \tag{12}$$

Now the LV matrices A and λ have to follow the changes in the order of the variables, namely columns and rows i and j of matrix A and elements i and j of λ must be swapped.

Multiplying matrix A with P from the left:

$$P \cdot A =$$

$$= \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} & \dots & a_{1,i} & \dots & a_{1,j} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,1} & \dots & a_{i,i} & \dots & a_{i,j} & \dots & a_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{j,1} & \dots & a_{j,i} & \dots & a_{j,j} & \dots & a_{j,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,i} & \dots & a_{n,j} & \dots & a_{n,n} \end{bmatrix}$$

$$= \begin{bmatrix} a_{1,1} & \dots & a_{1,i} & \dots & a_{1,j} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{j,1} & \dots & a_{j,i} & \dots & a_{j,j} & \dots & a_{j,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,1} & \dots & a_{i,i} & \dots & a_{i,j} & \dots & a_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,i} & \dots & a_{n,j} & \dots & a_{n,n} \end{bmatrix}$$

is not enough because rows i and j are still in order so it is necessary to multiply A with P from the right, too:

$$\hat{A} = (P \cdot A) \cdot P =$$

$$= \begin{bmatrix} a_{1,1} & \dots & a_{1,i} & \dots & a_{1,j} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{j,1} & \dots & a_{j,i} & \dots & a_{j,j} & \dots & a_{j,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,1} & \dots & a_{i,i} & \dots & a_{i,j} & \dots & a_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,i} & \dots & a_{n,j} & \dots & a_{n,n} \end{bmatrix} \cdot \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{1,1} & \dots & a_{1,j} & \dots & a_{1,i} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{j,1} & \dots & a_{j,j} & \dots & a_{j,i} & \dots & a_{j,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,1} & \dots & a_{i,j} & \dots & a_{i,i} & \dots & a_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,j} & \dots & a_{n,i} & \dots & a_{n,n} \end{bmatrix}.$$

Similarly

$$\begin{aligned}\hat{\lambda} &= P\cdot\lambda = \\ &= \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_i \\ \vdots \\ \lambda_j \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_j \\ \vdots \\ \lambda_i \\ \vdots \\ \lambda_n \end{bmatrix}. \end{aligned} \tag{13}$$

2.3.2 Re-scaled LV variables

Let Q be a scaling matrix in the form $Q = \mathrm{diag}(q_1, q_2, \cdots, q_n)$ where $q_i \in R$. If we transform the original LV variables into the following set of variables:

$$\hat{z} = Q\cdot z = \begin{bmatrix} q_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & q_n \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} q_1 z_1 \\ \vdots \\ q_n z_n \end{bmatrix} \tag{14}$$

then the Lotka-Volterra coefficient matrix (matrix A) must be aligned to the scaling:

$$\dot{\hat{z}}_i = \hat{z}_i\Big(\lambda_i + \sum_{j=1}^{n} \frac{a_{ij}}{q_j}\hat{z}_j\Big), \quad i = 1, \dots, n \tag{15}$$

i.e. the i-th column of A must be re-scaled by q_i^{-1}:

$$\begin{bmatrix} \frac{a_{11}}{q_1} & \dots & \frac{a_{1n}}{q_n} \\ \vdots & \ddots & \vdots \\ \frac{a_{n1}}{q_1} & \dots & \frac{a_{nn}}{q_n} \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{q_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{q_n} \end{bmatrix} \tag{16}$$

which means

$$\hat{A} = A\cdot Q^{-1}, \qquad \hat{\lambda} = \lambda \tag{17}$$

being the LV system matrices.

Performing both permutation and multiplication will also keep LV form.

2.4 Input Affine QP Models

We are searching for an appropriate input affine model form:

$$\dot{x} = f(x) + \sum_{j=1}^{p} g_j(x)u_j,$$

where $f \in R^n \to R^n$, and $g_j \in R^n \to R^n$ are quasi-polynomial functions, and the input variable u is p-dimensional.

A rational form for QP model would be

$$\dot{x}_i = x_i\Big(\lambda_i + \sum_{j=1}^{m} A'_{ij} \prod_{k=1}^{n} x_k^{B_{jk}}\Big) + \sum_{l=1}^{p} x_i\Big(\mu_{li} + \sum_{j=1}^{m} C_{lij} \prod_{k=1}^{n} x_k^{B_{jk}}\Big)u_l, \quad i = 1, \ldots, n. \tag{18}$$

The above input affine form is the extension of the autonomous QP differential equation form if one regards u_i as a scalar parameter for every i. It can be shown that with any state feedback $u = H(x)$ the closed loop system remains in QP form if H is a quasi-polynomial function. Of course, the number of quasi-monomials (and accordingly the dimension of the LV system) of the closed loop system will be greater or equal, than the systems' one without feedback. For example, with a quasi-monomial feedback $u = v\prod_{k=1}^{n} x_k^{w_k}$, the number of quasi-monomials of the closed loop system will be $2m + 1$, in the worst case, where m is the number of quasi-monomials of the open loop system.

2.5 Zero dynamics

Zero dynamics for nonlinear systems is equivalent to the zeros of the transfer function in the case of linear systems.

The zero dynamics [12] of an input-affine model is the constrained system:

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x)u_i \\ 0 &= h(x) \end{aligned} \tag{19}$$

Consider the following SISO nonlinear input-affine system:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{20}$$

System (20) has *relative degree* r at point x_0 if

- $L_g L_f^k h(x) = 0$ for all x in a neighborhood of x_0 and $k < r-1$
- $L_g L_f^{r-1} h(x_0) \neq 0$

where $L_f h(x)$ denotes the so called *Lie-derivative* i.e.:

$$L_f h(x) = \frac{\partial h(x)}{\partial x} \cdot f(x)$$

Using a local coordinates transformation

$$z = \Phi(x)$$

where $z_i = \phi_i(x) = L_f^{i-1} h(x)$ for $1 \leq i \leq r$ and $L_g \phi_j(x) = 0$ for $r+1 \leq j \leq n$ the transformed system is

$$\begin{aligned} \dot{z_1} &= z_2 \\ \dot{z_2} &= z_3 \\ &\dots \\ \dot{z_{r-1}} &= z_r \\ \dot{z_r} &= b(\xi,\eta) + a(\xi,\eta)u \\ \dot{\eta} &= q(\xi,\eta) \end{aligned} \tag{21}$$

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 $b(\xi,\eta) = L_f^r h(\Phi^{-1}(\xi,\eta))$. If the output of the system (21) is identically zero then the system model

$$\dot{\eta} = q(0,\eta) \tag{22}$$

describes the internal behavior of system (21). Then for an initial state x^* for which

$$\xi(0) = 0, \quad \eta(0) = \eta_0 \text{ arbitrary},$$

the zeroing input is

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

If a system has asymptotically stable zero dynamics with respect to a given output, it means, that it is sufficient to stabilize this output and the closed loop system will be asymptotically stable.

2.6 Process systems in QP form

The state equation of a lumped process system contains additive terms corresponding to the different *mechanisms* taking place: convection, transfer and sources [7]. Based on this understanding, the input function g in the input-affine state equation

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

with x being the state and u being the input variable is always a *linear* function of the state vector x because of the properties of the convective terms it originates from.

The nonlinear state function $f(x)$ is broken down into a linear term originating from transfer and a general nonlinear term caused by the sources (other generation and consumption terms including chemical reactions, phase changes etc.) In most of the cases, the source term contains the following special nonlinearities:

- *reaction rate expressions*
  exponential type nonlinearities which account for the temperature dependence and polynomial expressions for the concentration dependence with terms in the form of
  $$e^{-\frac{c_1}{c_2 x_T}} \prod_i x_i^{\alpha_i}$$
- *global reaction rate expressions*
  rational function type nonlinear factors in terms of $f(x)$
  $$\frac{p_1(x)}{p_2(x)}$$
  where both p_1 and p_2 are polynomials of the state vector elements x_i.

Systems containing the above two kinds of nonlinearities can easily be transformed to QP form [2].

Another speciality of process systems is that the input function $g(x)$ is usually linear in x, or even may be just a constant function.

2.7 Linear Matrix Inequalities

Every *linear matrix inequality* is a convex constraint which can be expressed in the form

$$L(x) = L_0 + L_1 x_1 + \cdots + L_n x_n > 0 \tag{23}$$

where

- $L_0, L_1, \cdots, L_n$ are given symmetric matrices
- $(x_1, \cdots, x_n)^T \in R^n$ is the vector of decision variables

If we allow the equality in (23), then we talk about *non-strict LMI*. LMIs are solved with the so called *Ellipsoid algorithm* [9] which has polynomial complexity.
There are effective tools for solving LMIs. One of them is the MATLAB® LMI Control Toolbox [9] but there are other useful instruments, too [11].

3 Stability Properties

In this section local and global stability analysis is discussed for QP and LV systems. Henceforth it is assumed that x^* is a positive equilibrium point, i.e. $x^* \in int(R_+^n)$ in the QP case and similarly $z^* \in int(R_+^m)$ is a unique equilibrium point in the LV case. Afterwards in the last part of this chapter estimating of stability region will be considered.

3.1 Local Stability Analysis

Local stability analysis is performed around a fixed *steady-state operating point* x^* which is a stationary solution of the original nonlinear ODE.

3.1.1 Local stability analysis of quasi-polynomial models

Here the (non-trivial) stationary solution x^* is determined from the steady-state version of (1)

$$0 = x_i^* \Big(\lambda_i + \sum_{j=1}^{m} A_{ij} \prod_{k=1}^{n} x_k^{*B_{jk}} \Big), \qquad (24)$$
$$i = 1, \ldots, n, \qquad m \geq n$$

It is then easy to express the Jacobian matrix of a linearized QP system around a fixed point x^* in terms of A, B, and x^* [3]:

$$J = X^* \cdot A \cdot Q \cdot B \cdot X^{*-1}, \qquad (25)$$

where X^* and Q are the following diagonal matrices:

$$X^* = \begin{bmatrix} x_1^* & 0 & \ldots & 0 \\ 0 & x_2^* & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & x_n^* \end{bmatrix} \quad Q = \begin{bmatrix} q_1 & 0 & \ldots & 0 \\ 0 & q_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & q_m \end{bmatrix}$$

The components of q are the quasi-monomials of the system at the fixed point x^*.

$$q_j = \prod_{k=1}^{n} x_k^{*B_{jk}}, \qquad j = 1 \ldots m$$

3.1.2 Jacobian in the LV case

In the classical LV case the Jacobian matrix can be simplified as follows.

From the fact that the LV variables are the quasi-monomials of the system it follows that $Q_{LV} = X^*$, furthermore $B_{LV} = I$ (where I denotes the unit matrix). So the *Jacobian matrix for the LV system are*:

$$\begin{aligned} J_{LV} &= X^* \cdot A_{LV} \cdot Q_{LV} \cdot B_{LV} \cdot X^{*-1} = \\ &= X^* \cdot A_{LV} \cdot X^* \cdot I \cdot X^{*-1} = \\ &= X^* \cdot A_{LV} \cdot X^* \cdot X^{*-1} = \\ &= X^* \cdot A_{LV} \cdot I = \\ &= X^* \cdot A_{LV}. \end{aligned} \tag{26}$$

3.2 Global Stability Analysis

Global stability analysis can be performed by means of finding a Lyapunov function $V(x)$ which has the following special properties:

- $V : R^n \to R$
- $V(x) > 0$
- $\frac{dV(x)}{dt} = \frac{\partial V(x)}{\partial x} \cdot \frac{dx}{dt} < 0$

3.2.1 Global stability analysis of LV systems

For LV systems there is a well known Lyapunov function candidate family [4,5,6], which is in the form:

$$V(z) = \sum_{i=1}^{m} c_i \left(z_i - z_i^* - z_i^* \ln \frac{z_i}{z_i^*} \right), \tag{27}$$

$$c_i > 0, \quad i = 1 \dots m.$$

where $z^* = (z_1^*, \dots, z_m^*)^T$ is the unique positive equilibrium point of the system. The time derivative of the of the Lyapunov function (27) is:

$$\dot{V}(z) = \frac{\partial V(z)}{\partial z} \cdot \frac{dz}{dt} = \frac{1}{2}(z - z^*)(CA_{LV} + A_{LV}^T C)(z - z^*) \tag{28}$$

where $C = diag(c_1, \dots, c_m)$ and A_{LV} is the invariant characterizing the LV form. If $CA_{LV} + A_{LV}^T C$ is negative semi-definite then z^* is stable. If

$CA_{LV} + A_{LV}^T C$ is negative definite then z^* is globally asymptotically stable.
This way the check for global asymptotic stability requires to find a solution of the following linear matrix inequality:

$$\begin{aligned} CA_{LV} + A_{LV}^T C &< 0 \\ C &> 0 \end{aligned} \tag{29}$$

3.2.2 Global stability analysis of QP systems

The Lyapunov function (27) can be extended for GLV systems [4] by embedding the GLV system into LV form and using the Lotka-Volterra coefficient matrix $A_{LV} = B \cdot A$. However matrices A and B are usually not of rank m (Recall: m is the number of the *quasimonomials* of the QP system.) i.e. A_{LV} is rank-deficient in this case and it causes difficulties in solving the matrix inequality (29).

3.2.3 The use of LMIs to global stability analysis

In the course of global stability analysis of QP and LV systems it is necessary to solve the system of LMI (29) which is equivalent to the LMI

$$\begin{pmatrix} C & 0 \\ 0 & -CA_{LV}^T - A_{LV}C \end{pmatrix} > 0. \tag{30}$$

There are two factors which make the solving of the above LMI hard:

- Matrix C must be *diagonal* and positive definite. (Usually the constraint against this matrix is to be symmetric and positive definite.)
- If A_{LV} is not of full rank (and it is the case when one checks the global stability of a QP system), then there is no solution of the LMI (29) for the case of strict inequality. In addition, the widely used *Matlab LMI Control Toolbox* can not solve the non-strict LMI:

$$\begin{aligned} CA_{LV} + A_{LV}^T C &\leq 0 \\ C = \mathrm{diag}(c_1, \ldots, c_m) &> 0 \end{aligned} \tag{31}$$

The above problem of rank-deficient matrix A_{LV} can be solved by means of *singular perturbation*: a perturbation matrix εI is added to A_{LV} in the following way:

$$\tilde{A}_{LV} = A_{LV} + \varepsilon I, \qquad \varepsilon < 0, \qquad \varepsilon \to 0 \tag{32}$$

With this method the A_{LV} matrices of QP systems can be transformed to the standard, full rank definite form so that feasibility of LMI (29) can be examined for QP systems, too.

3.2.4 A numerical algorithm for solving the non-strict LMI case

The algorithm was introduced in [11], and it solves the non-strict LMI (31).
Let

$$c = [c_1, \ldots, c_n]^T \in R^n \quad \text{i.e.} \quad C = \text{diag}(c)$$

$$\phi_A(c) = \max_{v,\|v\|=1}\{v^T(A^TC + CA)v\} \qquad A \in R^{n\times n}, v \in R^n \tag{33}$$

$$g_A(c) = 2\hat{\Lambda}A\hat{v},$$

where $\hat{v}$ is the normalized eigenvector corresponding to the maximum eigenvalue of $A^TC + CA$, and $\hat{\Lambda} = \text{diag}(\hat{v})$.
Furthermore, let

$$\chi = \{c \mid 0 \le c_i \le 1, \quad i = 1, \ldots, n\}$$

If there exists some c, $\quad c_i > 0, \;\; i = 1, \ldots, n$ for which $\phi_A(c) \le 0$ then C is a solution for (33).
The algorithm is based on the following two facts:

1. $\phi_A(c) = \langle c, g_A(c)\rangle, \quad \forall c \in \chi$
2. $\phi_A(y) \ge \langle y, g_A(c)\rangle, \quad \forall y, c \in \chi$

The flowchart of the algorithm can be seen in Figure 1 on page 19.
The value of $\overline{c}^i$ can be determined by solving the following linear programming problem. Let

$$x = \left[s, c^T\right]^T, \quad s \in R, \quad c \in R^n \tag{34}$$

$$\min_x \begin{bmatrix} -1 & 0 & \ldots & 0 \end{bmatrix} x \tag{35}$$

subject to the constraints

$$\begin{aligned} &0 \le c_j \le 1, \quad j = 1, \ldots, n \\ &0 \le s \le 2\sqrt{n}\|A\|_2 \\ &s + (c - c^j)g_A(c^j) \le 0, \quad j = 1, \ldots, i \end{aligned} \tag{36}$$

If $\overline{x} = \left[\overline{s}, \overline{c}^T\right]$ is a solution for (35) with (36) then c can be chosen to be $\overline{c}^i$.

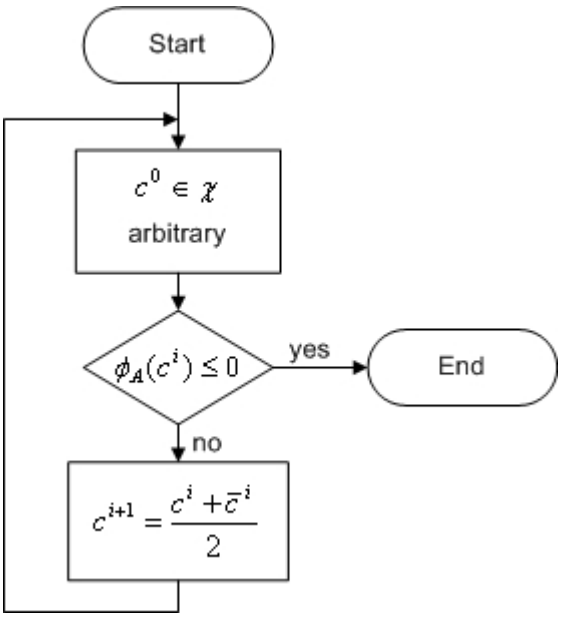

Figure 1: The flow diagram of the algorithm

3.3 Zero Dynamics and the Stability of Zero Dynamics for QP Systems

Embedding process systems into QP form unfortunately involves the loss of the input function's linear property mentioned in section 2.6. For this kind of input-affine QP systems the input function g is *linear in the quasi-monomials*. Hence the zero dynamics usually loses its QP structure. The stability analysis of this systems can be investigated by general methods [12].

With some specially chosen output function, however, zero dynamics is a quasi-polynomial system. In this case, tools discussed in sections 3.1.1 and 3.2.2 can be applied for the zero dynamics.

3.4 Estimating the Stability Region

If the system is not globally asymptotically stable, it would be useful to estimate the region around the equilibrium point in which the system is asymptotically stable. Various techniques exist for this purpose for example stability region estimating with quadratic Lyapunov functions[12], or contraction analysis[14].

In this section contraction analysis is discussed that uses the Jacobian of the QP and LV systems. A region of the state space is called a contraction region if the

Jacobian is uniformly negative definite in that region.

In what follows, the Jacobian matrix of quasi-polynomial and Lotka-Volterra systems is derived as a function of the state variables.

3.4.1 Jacobian Matrix for QP systems

Given the following general quasi-polynomial model:

$$\dot{x}_i = f_i(x) = x_i\Big(\lambda_i + \sum_{j=1}^{m} A_{ij} \prod_{k=1}^{n} x_k^{B_{jk}}\Big), \quad i = 1, \ldots, n. \tag{37}$$

Denote the Jacobian matrix of the system (37) at point x with

$$J_{QP}(x) = \begin{bmatrix} J_{11} & J_{12} & \cdots & J_{1n} \\ J_{21} & J_{22} & \ldots & J_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ J_{n1} & J_{n2} & \cdots & J_{nn} \end{bmatrix},$$

where

$$\begin{aligned} J_{il}(x) &= \frac{\partial f_i(x)}{\partial x_l} = x_i\Big(\sum_{j=1}^{m} A_{ij} B_{jl} \prod_{k=1}^{n} x_k^{B_{jk}}\Big) x_l^{-1} \\ J_{ii}(x) &= \frac{\partial f_i(x)}{\partial x_i} = \lambda_i + \sum_{j=1}^{m} A_{ij}(B_{ji}+1) \prod_{k=1}^{n} x_k^{B_{jk}} = \\ &= \Big(\sum_{j=1}^{m} A_{ij} B_{ji} \prod_{k=1}^{n} x_k^{B_{jk}}\Big) + \lambda_i + \sum_{j=1}^{m} A_{ij} \prod_{k=1}^{n} x_k^{B_{jk}}. \end{aligned}$$

Rewritten the above in matrix form, the Jacobian of system (37) is

$$J_{QP}(x) = X(x) \cdot A \cdot Q(x) \cdot B \cdot X(x)^{-1} + \Lambda + S(x), \tag{38}$$

where

$$\begin{aligned} X(x) &= diag(x_1, \ldots, x_n), \\ Q(x) &= diag(z_1, z_2, \ldots, z_m), \\ \Lambda &= diag(\lambda_1, \ldots, \lambda_n), \\ S(x) &= diag\Big(\sum_{j=1}^{m} A_{1j} z_j, \quad \sum_{j=1}^{m} A_{2j} z_j, \quad \ldots \quad \sum_{j=1}^{m} A_{nj} z_j\Big) \\ z_j &= \prod_{k=1}^{n} x_k^{B_{jk}}, \text{ a } j\text{-th monomial} \end{aligned}$$

Equation (38) is in accordance with the derived Jacobian for the equilibrium point of (37) [3]:

$$J_{QP}(x^*) = X(x^*) \cdot A \cdot Q(x^*) \cdot B \cdot X(x^*)^{-1},$$

since in a nonzero equilibrium point

$$\lambda_i = -\sum_{j=1}^{m} A_{ij} \prod_{k=1}^{n} x_k^{*^{B_{jk}}}$$

holds, which is

$$\Lambda = -S(x^*).$$

using the notations of (38).

3.4.2 Jacobian Matrix for LV systems

Consider the following Lotka-Volterra model:

$$\dot{z}_k = g_k(z) = z_k\Big(\lambda_{LV_k} \sum_{j=1}^{m} A_{LV_{ij}} z_j\Big), \quad k =, \ldots, m. \tag{39}$$

The Jacobian at point z

$$J_{LV}(z) = \begin{bmatrix} J_{11} & J_{12} & \cdots & J_{1m} \\ J_{21} & J_{22} & \ldots & J_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ J_{m1} & J_{m2} & \cdots & J_{mm} \end{bmatrix},$$

where

$$\begin{aligned} J_{kl}(z) &= \frac{\partial g_k(z)}{\partial z_l} = z_k A_{LV_{kl}} \\ J_{kk}(z) &= \frac{\partial g_k(z)}{\partial z_k} = z_k A_{LV_{kk}} + \lambda_{LV_k} + \sum_{j=1}^{m} A_{LV_{kj}} z_j. \end{aligned}$$

Rewritten the above in matrix form:

$$J_{LV}(z) = Q(z) \cdot A_{LV} + \Lambda_{LV} + S_{LV}(z), \tag{40}$$

using

$$\begin{aligned} z_j &= \prod_{k=1}^{n} x_k^{B_{jk}}, \text{ a } j\text{-th monomial} \\ Q(z) &= diag(z_1, z_2, \ldots, z_m), \\ \Lambda_{LV} &= diag(\lambda_{LV_1}, \ldots, \lambda_{LV_m}), \\ S_{LV}(z) &= diag\Big(\sum_{j=1}^{m} A_{LV_{1j}} z_j, \quad \sum_{j=1}^{m} A_{LV_{2j}} z_j, \quad \ldots \quad \sum_{j=1}^{m} A_{LV_{mj}} z_j \Big) \end{aligned}$$

It is also in accordance with the previously derived Jacobian for the equilibrium point of the Lotka-Volterra system (39) [5]:

$$J_{LV}(z^*) = Q(z^*) \cdot A_{LV},$$

because in the case of $z = z^*$

$$\lambda_k = -\sum_{j=1}^{m} A_{LV_{kj}} z_k^*$$

holds, i.e.

$$\Lambda_{LV} = -S_{LV}(z^*).$$

3.4.3 Contraction analysis

With the above presented Jacobian matrices it is possible to perform contraction analysis. A region Ω of the state space can be considered as stability region if the

Jacobian matrix of the system is uniformly negative definite for that region [14], i.e.

$$\frac{1}{2}\left(J^T(x) + J(x)\right) < 0, \qquad x \in \Omega, \tag{41}$$

where

$$J(x) = \frac{\partial f(x)}{\partial x}.$$

For Lotka-Volterra systems, inequality (41) is a linear matrix inequality because the Jacobian (40) is a linear function of z. Therefore, it can be managed with tools introduced in section 2.7. Furthermore, it is enough to find some corner points of the region, because of the convexity property of LMIs.

If the LV system were arisen from QP system by embedding, then the system trajectories would stay on a lower dimensional manifold of the state space because of the algebraic dependencies between the quasi-monomials. Therefore one has to take the intersection of the obtained region and the lower dimensional manifold determined by the monomial dependencies. The resulting region will give an estimation of the stability region.

In the case of QP systems (41) is unfortunately not a linear function of the QP state variables, so LMI tools cannot be used. But these QP systems can be transformed to (rank deficient) LV models, so they can be treated with the above tools .

4 Case Study of a Simple Fermentation Process

In what follows, the previously discussed techniques are applied to a simple process system, to a continuous fermentation process.

4.1 System Description

A fermentation process is a simple process system [7,12] with one perfectly stirred balance volume. In this *continuously stirred tank reactor* (seen in Figure 2 on page 25) a special simple fermentation reaction takes place. The reaction rate function is a nonlinear function (can be seen in Figure 3 on page 26). The physico-chemical properties are assumed to be constant.

The system has two state variables: the concentration of the biomass (cells of yeast) and the substrate concentration (e.g. sugar, which is consumed during the process). Initially the reactor contains a mixture with nonzero substrate and biomass concentration. The biomass growth is described by the non monotonous function μ (Figure 3).

The state space model of the fermentation process with nonlinear non-monotonous reaction rate function is investigated in the following form:

$$\begin{aligned} \dot{X} &= \mu(S)X - \tfrac{XF}{V} \\ \dot{S} &= -\tfrac{\mu(S)X}{Y} + \tfrac{(S_F - S)F}{V} \\ \mu(S) &= \mu_{max}\tfrac{S}{K_2S^2+S+K_1} \end{aligned} \tag{42}$$

Variables and their units of measure:

X: biomass concentration (state variable), $[g/l]$

S: substrate concentration (state variable), $[g/l]$

Constant parameters:

F: inlet feed flow rate, $[l/h]$

V: tank volume, $[l]$

S_F: inlet feed substrate concentration, $[g/l]$

μ_{max}: kinetic constant 1, $[-]$

K_1: kinetic constant 2, $[g/l]$

K_2: kinetic constant 3, $[l/g]$

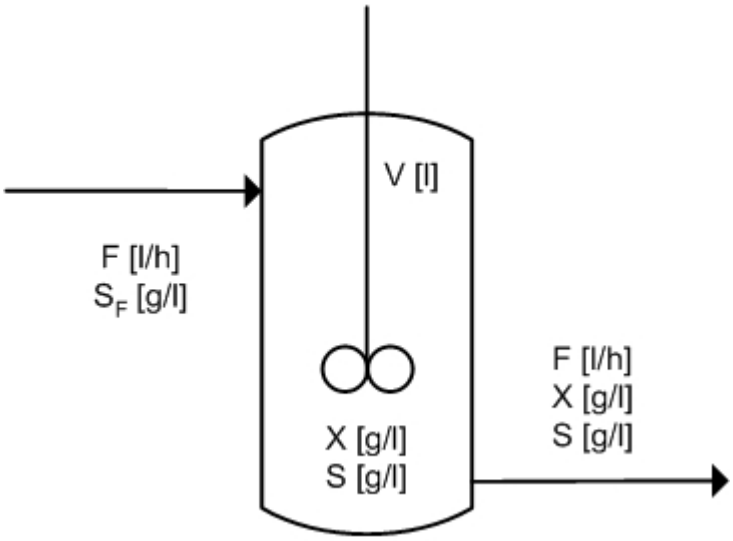

Figure 2: Fermentation process

Y: kinetic constant 4, [−]

Values of the parameters:

$$\begin{aligned} V &= 4\ l \\ S_F &= 10\ g/l \\ \mu_{max} &= 1\ [-] \\ K_1 &= 0.03\ g/l \\ K_2 &= 0.5\ l/g \\ Y &= 0.5\ [-] \end{aligned}$$

The investigated operating point will be the one for which the biomass concentration is maximal:

$$S_0 = \frac{1}{2}\frac{-2K_1 + 2\sqrt{K_1^2 + S_F^2 K_1 K_2 + S_F K_1}}{S_F K_2 + 1}$$

$$X_0 = (S_F - S_0)Y$$

$$F_0 = \mu(S_0)V$$

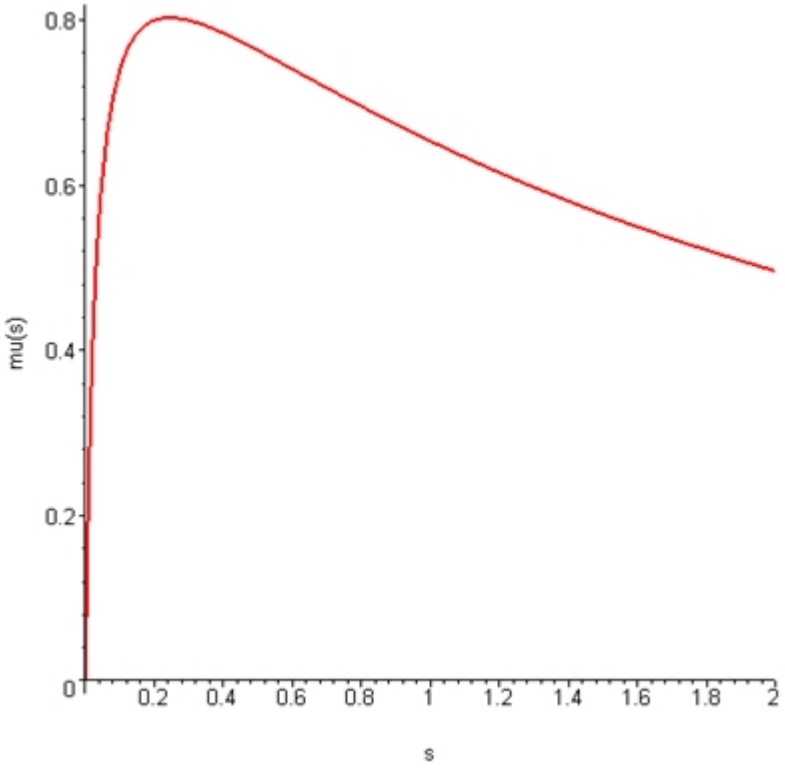

Figure 3: The graph of function $\mu(S)$

4.2 Transformation into Quasi-Polynomial Form

The method obtained in section 2.1.1 on page 5 is applied here because $\mu(S)$ is not a QP function.

The new algebraic variable is chosen to be:

$$Z = f(X,S) = \frac{1}{K_2 S^2 + S + K_1} \tag{43}$$

The QP system with the above choice is:

$$\begin{aligned}
\dot{X} &= X\left[\mu_{max} SZ - \frac{F}{V}\right] \\
\dot{S} &= S\left[-\frac{\mu_{max}}{Y} XZ + \frac{S_F F}{V} S^{-1} - \frac{F}{V}\right] \\
\dot{Z} &= Z\left[\frac{2\mu_{max} K_2}{Y} S^2 X Z^2 - \frac{2K_2 S_F F}{V} SZ + \frac{2K_2 F}{V} S^2 Z + \right. \\
&\quad \left. + \frac{\mu_{max}}{Y} SXZ^2 - \frac{S_F F}{V} Z + \frac{F}{V} SZ\right]
\end{aligned} \tag{44}$$

The quasi-monomials, i.e. the LV variables are:

$$\begin{aligned} q_1 &= SZ, \quad q_2 = XZ, \\ q_3 &= S^{-1}, \quad q_4 = S^2XZ^2, \\ q_5 &= S^2Z, \quad q_6 = SXZ^2, \\ q_7 &= Z. \end{aligned}$$

From the QP system equation it is easy to determine the QP invariants A, B, λ:

$$A = \begin{bmatrix} \mu_{max} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\mu_{max}}{Y} & \frac{S_F}{V}F & 0 & 0 & 0 & 0 \\ \frac{-2K_2S_F+1}{V}F & 0 & 0 & \frac{2\mu_{max}K_2}{Y} & \frac{2K_2}{V}F & \frac{\mu_{max}}{Y} & -\frac{S_F}{V}F \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 2 & 2 \\ 0 & 2 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}, \quad \lambda = \begin{bmatrix} -\frac{F}{V} \\ -\frac{F}{V} \\ 0 \end{bmatrix}$$

The numerical values of A and λ are

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 8.0223 & 0 & 0 & 0 & 0 \\ -7.2201 & 0 & 0 & 2 & 0.80223 & 2 & -8.0223 \end{bmatrix}, \quad \lambda = \begin{bmatrix} -0.80223 \\ -0.80223 \\ 0 \end{bmatrix}$$

4.2.1 Input-affine QP form

Using (18) and $u = F$, we get the following matrices for the input affine form:

$$A' = \begin{bmatrix} \mu_{max} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\mu_{max}}{Y} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2\mu_{max}K_2}{Y} & 0 & \frac{\mu_{max}}{Y} & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{S_F}{V} & 0 & 0 & 0 & 0 \\ \frac{-2K_2S_F+1}{V} & 0 & 0 & 0 & \frac{2K_2}{V} & 0 & -\frac{S_F}{V} \end{bmatrix}, \quad \lambda' = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mu = \begin{bmatrix} -\frac{1}{V} \\ -\frac{1}{V} \\ 0 \end{bmatrix}$$

4.3 Lotka-Volterra model

Computing the matrix product $B \cdot A$ for the above QP system results in the following Lotka-Volterra coefficient matrix A_{LV} :

$$\begin{bmatrix}
\frac{(-2K_2S_F+1)}{V}F & -\frac{\mu_{max}}{Y} & \frac{S_F}{V}F & \frac{2\mu_{max}K_2}{Y} & \frac{2K_2}{V}F & \frac{\mu_{max}}{Y} & -\frac{S_F}{V}F \\
\mu_{max} + \frac{(-2K_2S_F+1)}{V}F & 0 & 0 & \frac{2\mu_{max}K_2}{Y} & \frac{2K_2}{V}F & \frac{\mu_{max}}{Y} & -\frac{S_F}{V}F \\
0 & \frac{\mu_{max}}{Y} & -\frac{S_F}{V}F & 0 & 0 & 0 & 0 \\
\mu_{max} + \frac{2(-2K_2S_F+1)}{V}F & -\frac{2\mu_{max}}{Y} & \frac{2S_F}{V}F & \frac{4\mu_{max}K_2}{Y} & \frac{4K_2}{V}F & \frac{2\mu_{max}}{Y} & -\frac{2S_F}{V}F \\
\frac{(-2K_2S_F+1)}{V}F & -\frac{2\mu_{max}}{Y} & \frac{2S_F}{V}F & \frac{2\mu_{max}K_2}{Y} & \frac{2K_2}{V}F & \frac{\mu_{max}}{Y} & -\frac{S_F}{V}F \\
\mu_{max} + \frac{2(-2K_2S_F+1)}{V}F & -\frac{\mu_{max}}{Y} & \frac{S_F}{V}F & \frac{4\mu_{max}K_2}{Y} & \frac{4K_2}{V}F & \frac{2\mu_{max}}{Y} & -\frac{2S_F}{V}F \\
\frac{(-2K2Sf+1)}{V}F & 0 & 0 & \frac{2\mu_{max}K_2}{Y} & \frac{2K_2}{V}F & \frac{\mu_{max}}{Y} & -\frac{S_F}{V}F
\end{bmatrix}$$

With the numerical values substituted we get:

$$A_{LV} = \begin{bmatrix}
-7.2201 & -2 & 8.0223 & 2 & 0.80223 & 2 & -8.0223 \\
-6.2201 & 0 & 0 & 2 & 0.80223 & 2 & -8.0223 \\
0 & 2 & -8.0223 & 0 & 0 & 0 & 0 \\
-13.44 & -4 & 16.045 & 4 & 1.6045 & 4 & -16.045 \\
-7.2201 & -4 & 16.045 & 2 & 0.80223 & 2 & -8.0223 \\
-13.44 & -2 & 8.0223 & 4 & 1.6045 & 4 & -16.045 \\
-7.2201 & 0 & 0 & 2 & 0.80223 & 2 & -8.0223
\end{bmatrix}$$

4.4 Local Stability Analysis

Local stability analysis is performed around an equilibrium point of the system. First the original systems local stability will be investigated, afterwards the QP and LV representations will be examined.

4.4.1 Local stability of the original fermenter

The investigated operating point is

$$\begin{aligned} X_0 &= 4.8906 \\ S_0 &= 0.2187 \\ F_0 &= 3.2089 \end{aligned} \tag{45}$$

The Jacobian matrix is

$$J = \begin{bmatrix} 0 & 0.4004 \\ -1.6045 & -1.6031 \end{bmatrix},$$

with eigenvalues

$$\lambda_1 = -0.8009, \quad \lambda_2 = -0.8022.$$

So the original system's equilibrium point (45) is locally asymptotically stable.

4.4.2 Local stability of the quasi-polynomial system

The stability properties are investigated for the QP system (44). According to formula (25), using

$$X^* = \begin{bmatrix} 4.8906 & 0 & 0 \\ 0 & 0.2187 & 0 \\ 0 & 0 & 3.6682 \end{bmatrix} \tag{46}$$

and

$$Q = \begin{bmatrix} 0.8022 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 17.9397 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.5725 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.1475 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1754 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 14.3919 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.6682 \end{bmatrix}$$

the Jacobian matrix is

$$J_{QP} = \begin{bmatrix} 0 & 17.9399 & 1.0696 \\ -1.6045 & -36.6815 & -2.1392 \\ 26.3104 & 601.5186 & 35.0788 \end{bmatrix}$$

with eigenvalues

$$\lambda_1 = -0.8009, \quad \lambda_2 = -0.8022, \quad \lambda_3 = 0.000.$$

The additional zero eigenvalue comes from the algebraic equation of the new variable Z.

4.4.3 Local stability of the Lotka-Volterra system

From (26) the LV Jacobian matrix is as follows. Columns $1-4$ are

$$J_{LV}(1-4) = \begin{bmatrix} -5.7922 & -1.6045 & 6.4357 & 1.6045 \\ -111.5857 & 0 & 0 & 35.8792 \\ 0 & 9.1449 & -36.6818 & 0 \\ -42.3024 & -12.5899 & 50.4998 & 12.5899 \\ -1.2667 & -0.7018 & 2.8150 & 0.3509 \\ -193.4266 & -28.7834 & 115.4546 & 57.5668 \\ -26.4845 & 0 & 0 & 7.3364 \end{bmatrix}$$

Columns $5-7$:

$$J_{LV}(5-7) = \begin{bmatrix} 0.6436 & 1.6045 & -6.4357 \\ 14.3917 & 35.8792 & -143.9169 \\ 0 & 0 & 0 \\ 5.0500 & 12.5899 & -50.4998 \\ 0.1407 & 0.3509 & -1.4075 \\ 23.0909 & 57.5668 & -230.9092 \\ 2.9427 & 7.3364 & -29.4273 \end{bmatrix}$$

and the eigenvalues are

$$\lambda_1 = -0.8009, \quad \lambda_2 = -0.8022, \quad \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = \lambda_7 = 0.000.$$

4.4.4 Comparison of the local stability results

In the course of embedding the fermenter model (42) into QP and LV models, algebraic equations had been introduced and they had been differentiated (e.g. (43)). This way the original two differential equations were supplemented with quasi-differential equations.

In local stability analysis, this algebraic equations appear as *zero eigenvalues* of the Jacobian matrices and the non zero eigenvalues are equal to the original system's ones.

The algebraic equations mentioned above imply the effect that the system will roam a two dimensional manifold in the state space in both the QP and LV case . This two dimensional manifold can be seen in Figure 4 for the (three dimensional) QP system. Few trajectories reaching the equilibrium point (46) are illustrated, too.

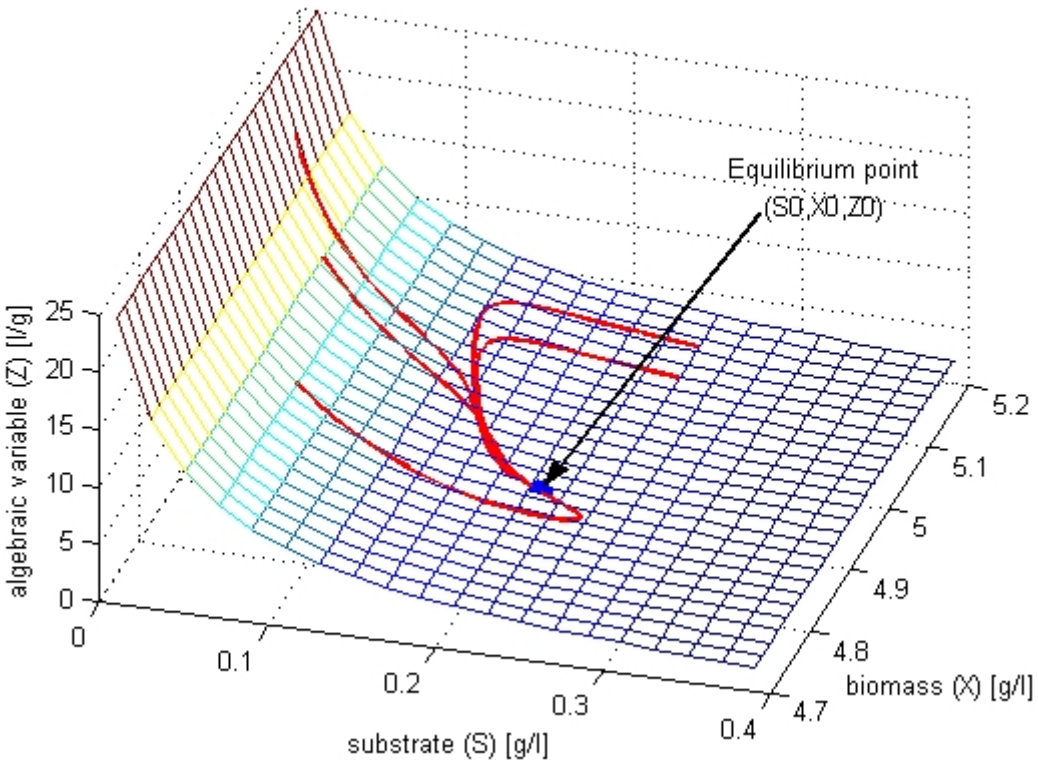

Figure 4: The two dimensional manifold and some trajectories of the QP system

4.5 Global Stability Analysis

As it was mentioned before, the Lyapunov function is the same for QP systems and their LV representation, so it is sufficient to investigate the global stability properties of the LV model.

Using *Matlab LMI Control Toolbox* the LMI (29) is infeasible even with singular perturbation (32).

There exists an other algorithm [11] specially for solving the non-strict LMI

$$\begin{aligned} CA_{LV} + A_{LV}^T C &\leq 0 \\ C &> 0 \end{aligned}$$

It was resulted in a parameter matrix

$$C = 10^{-10} \cdot \begin{bmatrix} 8.7068 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.131 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 31.018 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7587 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.1797 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.962 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9.5066 \end{bmatrix}$$

The elements of this matrix can be considered as zeros, i.e. the fermentation process (42) is not globally asymptotically stable.

It is not surprising, because the system has a so called *wash out* operating point beside the analyzed operating point (45). In this point the biomass concentration X decreases to zero, and the substrate concentration increases to S_F since there is no biomass consuming it.

4.6 Stability Region Analysis

As it was mentioned in section 3.4.3 on page 22, the algebraic dependencies between the quasi-monomials define a manifold in the 7 dimensional state space.

The algebraic dependencies between the monomials $q_1, q_2, q_3, q_4, q_5, q_6$, and q_7:

$$\begin{aligned} q_3 &= q_7 q_1^{-1} \\ q_4 &= q_2 q_1^2 q_7^{-1} \\ q_5 &= q_1^2 q_7^{-1} \\ q_6 &= q_1 q_2 \end{aligned} \tag{47}$$

i.e. the above algebraic equations define a 3 dimensional manifold in the 7 dimensional Lotka-Volterra state space. There is an additional algebraic equation (43), because the QP system was derived from a 2 dimensional process system.

From (41) and (40) the LMI for contraction analysis is in the form:

$$\begin{aligned} & \tfrac{1}{2}\left(J_{LV}^T(z) + J_{LV}(z)\right) = \\ = \; & \tfrac{1}{2}\left[(Q \cdot A_{LV} + \lambda_{LV} + S_{LV})^T + (Q \cdot A_{LV} + \lambda_{LV} + S_{LV})\right] = \\ = \; & \tfrac{1}{2}\left(A_{LV}^T \cdot Q + Q \cdot A_{LV}\right) + \lambda_{LV} + S_{LV} < 0, \end{aligned} \tag{48}$$

which can be written in the following form:

$$M_0 + M_1 q_1 + M_2 q_2 + + M_3 q_3 + M_4 q_4 + M_5 q_5 + M_6 q_6 + M_7 q_7 < 0 \tag{49}$$

where

$$M_0 = \begin{bmatrix} -0.80223 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.80223 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.80223 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.4067 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.6045 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.6045 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$M_1 = \begin{bmatrix} -14.44 & -1 & 4.0112 & 1 & 0.40112 & 1 & -4.0112 \\ -1 & -6.2201 & 0 & 0 & 0 & 0 & 0 \\ 4.0112 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -13.44 & 0 & 0 & 0 \\ 0.40112 & 0 & 0 & 0 & -7.2201 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -13.44 & 0 \\ -4.0112 & 0 & 0 & 0 & 0 & 0 & -7.2201 \end{bmatrix},$$

$$M_2 = \begin{bmatrix} -2 & -3.11 & 0 & 0 & 0 & 0 & 0 \\ -3.11 & 0 & 0 & 1 & 0.40112 & 1 & -4.0112 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -4 & 0 & 0 & 0 \\ 0 & 0.40112 & 0 & 0 & -4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -2 & 0 \\ 0 & -4.0112 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$M_3 = \begin{bmatrix} 8.0223 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -16.045 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 16.045 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 16.045 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8.0223 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$M_4 = \begin{bmatrix} 2 & 0 & 0 & -6.7201 & 0 & 0 & 0 \\ 0 & 2 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8.0223 & 0 & 0 & 0 \\ -6.7201 & -2 & 8.0223 & 8 & 0.80223 & 2 & -8.0223 \\ 0 & 0 & 0 & 0.80223 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 4 & 0 \\ 0 & 0 & 0 & -8.0223 & 0 & 0 & 2 \end{bmatrix},$$

$$M_5 = \begin{bmatrix} 0.80223 & 0 & 0 & 0 & -3.61 & 0 & 0 \\ 0 & 0.80223 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8.0223 & 0 & 0 \\ 0 & 0 & 0 & 1.6045 & 1 & 0 & 0 \\ -3.61 & -2 & 8.0223 & 1 & 1.6045 & 1 & -4.0112 \\ 0 & 0 & 0 & 0 & 1 & 1.6045 & 0 \\ 0 & 0 & 0 & 0 & -4.0112 & 0 & 0.80223 \end{bmatrix},$$

$$M_6 = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & -6.7201 & 0 \\ 0 & 2 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.0112 & 0 \\ 0 & 0 & 0 & 4 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0.80223 & 0 \\ -6.7201 & -1 & 4.0112 & 2 & 0.80223 & 8 & -8.0223 \\ 0 & 0 & 0 & 0 & 0 & -8.0223 & 2 \end{bmatrix},$$

$$M_7 = \begin{bmatrix} -8.0223 & 0 & 0 & 0 & 0 & 0 & -3.61 \\ 0 & -8.0223 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -16.045 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -8.0223 & 0 & 0.40112 \\ 0 & 0 & 0 & 0 & 0 & -16.045 & 1 \\ -3.61 & 0 & 0 & 1 & 0.40112 & 1 & -16.045 \end{bmatrix}.$$

The 7-dimensional convex region Φ defined by the following points is a feasibility region of LMI (49):

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
z_1	0.0916	20.1400	1.8772	1.8380	1.8606	1.9515	1.9269	0.8002	19934
z_2	0.1838	0.4059	7.0283	1.5788	1.5202	1.8725	1.1897	1.1	11114
z_3	0.1668	1.8993	1.4126	2.0417	1.3419	1.2565	1.1485	1	8800
z_4	0.0563	0.5681	0.2987	0.4824	2.1082	0.0955	0.1187	1	5825
z_5	0.0145	0.5712	0.5574	0.3949	0.1762	1.8150	0.1447	1.1	5811
z_6	0.1037	0.7107	0.2758	0.4932	0.2015	0.1140	3.0576	0.9	6494
z_7	0.1123	1.5302	1.9092	1.8569	1.8783	1.9635	1.9393	1801.7	16587

$$\Phi = \left\{ \sum_{i=1}^{9} \alpha_i P_i \quad \middle| \quad \sum_{i=1}^{9} \alpha_i = 1, \quad \alpha_i \in [0,1] \right\} \tag{50}$$

Taking the subset of Φ, in which the algebraic relationships (47) hold for the elements $\phi = [\phi_1, \phi_2, \ldots, \phi_7]^T$:

$$\begin{aligned} \phi_3 &= \frac{\sum_{i=1}^{9} \alpha_i P_{i7}}{\sum_{i=1}^{9} \alpha_i P_{i1}} \\ \phi_4 &= \frac{\left(\sum_{i=1}^{9} \alpha_i P_{i2}\right) \cdot \left(\sum_{i=1}^{9} \alpha_i P_{i1}\right)^2}{\sum_{i=1}^{9} \alpha_i P_{i7}} \\ \phi_5 &= \frac{\left(\sum_{i=1}^{9} \alpha_i P_{i1}\right)^2}{\sum_{i=1}^{9} \alpha_i P_{i7}} \\ \phi_6 &= \left(\sum_{i=1}^{9} \alpha_i P_{i1}\right) \cdot \left(\sum_{i=1}^{9} \alpha_i P_{i2}\right) \end{aligned} \tag{51}$$

The 2 dimensional region of the above 3 dimensional subset defined by the algebraic equation (43) is a contraction region Ω of the Lotka-Volterra system.

4.7 Zero dynamics and its stability analysis

The zero dynamics depend on the choice of the input and the output. The aim is to investigate the effect of the choice of the output on the zero dynamics . The input of the system is chosen to be the input flowrate, F. The output is selected to be the centered substrate concentration,

$$y = S - S_0$$

The zero-constrained output equation is

$$y = S - S_0 = 0, \tag{52}$$

i.e.

$$S = S_0.$$

From the above equations the constrained state equations are:

$$\begin{aligned}
\dot{X} &= X\left[\mu_{max} S_0 Z - \frac{F}{V}\right] \\
\dot{S} &= S_0\left[-\frac{\mu_{max}}{Y} XZ + \frac{S_F F}{V} S_0^{-1} - \frac{F}{V}\right] \\
\dot{Z} &= Z\left[\frac{F - 2K_2 S_F F}{V} S_0 Z + \frac{2\mu_{max} K_2}{Y} S_0^2 X Z^2 + \frac{2K_2 F}{V} S_0^2 Z + \right. \\
&\quad \left. + \frac{\mu_{max}}{Y} S_0 X Z^2 - \frac{S_F F}{V} Z\right]
\end{aligned} \tag{53}$$

The time derivative of the output is also zero, because $y \equiv 0$

$$\dot{y} = \dot{S} = 0,$$

i.e.

$$\begin{aligned}
0 &= S_0\left[-\frac{\mu_{max}}{Y} XZ + \frac{S_F F}{V} S_0^{-1} - \frac{F}{V}\right] \\
&= -\frac{\mu_{max} S_0}{Y} XZ + \frac{(S_F - S_0)F}{V}
\end{aligned}$$

From the above one can express the zeroing input as a function of the state variables:

$$F = \frac{\mu_{max} S_0}{Y} \cdot \frac{V}{S_F - S_0} \cdot XZ = \frac{\mu_{max} S_0 V}{Y(S_F - S_0)} XZ \tag{54}$$

Substituting it into (53), the zero dynamics transforms to

$$\begin{aligned}
\dot{X} &= X\left[\mu_{max}S_0 Z - \frac{\mu_{max}S_0}{Y(S_F - S_0)}XZ\right] \\
\dot{Z} &= Z\left[\frac{(1-2K_2S_F)\mu_{max}S_0^2}{Y(S_F-S_0)}XZ^2 + \frac{2\mu_{max}K_2S_0^2}{Y}XZ^2 + \frac{2K_2\mu_{max}S_0^3}{Y(S_F-S_0)}XZ^2 + \right. \\
&\quad \left. + \frac{\mu_{max}S_0}{Y}XZ^2 - \frac{\mu_{max}S_0S_F}{Y(S_F-S_0)}XZ^2\right] \qquad (55)
\end{aligned}$$

It can be seen that for this simple output function (52) the obtained closed loop zero dynamics system is in quasi-polynomial form. The quasi-monomials are:

$$Z, \quad XZ, \quad XZ^2$$

The system matrices A_0, B_0,and λ_0 are as follows

$$A_0 = \begin{bmatrix} \mu_{max}S_0 & -\frac{\mu_{max}S_0}{Y(S_F-S_0)} & 0 \\ & & \\ 0 & 0 & R \end{bmatrix} \quad B_0 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \quad \lambda_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

where

$$R = \frac{(1-2K_2S_F)\mu_{max}S_0^2}{Y(S_F-S_0)} + \frac{2\mu_{max}K_2S_0^2}{Y} + \frac{2K_2\mu_{max}S_0^3}{Y(S_F-S_0)} + \frac{\mu_{max}S_0}{Y} - \frac{\mu_{max}S_0S_F}{Y(S_F-S_0)}$$

The equilibrium point (X_0, Z_0) of system (55) is locally stable since the eigenvalues of its Jacobian matrix are

$$-0.8022, \quad 0$$

The system is globally asymptotically stable, too. The result of the linear matrix inequality (31) supplemented with singular perturbation (32) is the following diagonal matrix C containing the parameters of the Lyapunov function (27) in its diagonal.

$$C = \begin{bmatrix} 8.4341 \cdot 10^8 & 0 & 0 \\ 0 & 180.33 & 0 \\ 0 & 0 & 0.00061958 \end{bmatrix}$$

with

$$\varepsilon = -0.0000001$$

The global asymptotic stability of the zero dynamics means that it is sufficient to stabilize the centered substrate concentration and the whole system will show stability. In other words, state variable S would be a hopeful candidate for stabilizing purposes.

5 Conclusions and Future Work

Powerful and interesting methods for both local and global stability analysis of quasi-polynomial and Lotka-Volterra systems were discussed in this diploma work. It was shown that the class of LV models is closed under two similarity transformations: permutation and re-scaling.

It is apparent that the global stability analysis is far from being trivial in case of QP systems generated from non quasi-polynomial process systems. In this case the *linear matrix inequalities* degenerate to non-strict LMIs and there are very few tools to solve them.

A new method for solving these rank deficient linear matrix inequalities has been proposed that uses singular perturbation technique.

Unfortunately, zero dynamics loses the QP format except some special output functions. For these particular systems, the introduced methods can be used for stability analysis.

Further work includes the control of QP and LV systems by using the results of zero dynamics analysis. The controller design for QP models lead to the field of *bilinear matrix inequalities* [13].

6 Appendix

6.1 The Matlab script for solving the LMI (31)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Algorithm for the LMI A'C+CA<0 with singular perturbation
% Last modified: 2004.03.27
% Input: - epsilon for the perturbation value
%        - QP coefficient matrix A
%        - QP exponent matrix B
% Output: diagonal matrix C for the Lyapunov function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function solveLMI(epsilon,A,B)

%A=[mumax*s0 -(mumax*s0)/(Y*(Sf-s0)) 0;...
%0,0,((1-2*K2*Sf)*mumax*s0^2)/(Y*(Sf-s0))...
%+(2*mumax*K2*s0^2)/Y+(2*K2*s0^3*mumax)/(Y*(Sf-s0))...
%+(mumax*s0)/Y-(Sf*mumax*s0)/(Y*(Sf-s0))];

%B=[0 1;1 1;1 2]; lambda=[0;0]; Xs=[x0 0;0 y0]; Q=diag([y0 x0*y0
%x0*y0^2]); J=Xs*A*Q*B*inv(Xs); eig(J) eig(Q*B*A)

%pause

M=B*A;       % M = Lotka-Volterra coefficient matrix
%epsilon=-.0001;
M_hat=M+epsilon*eye(size(M,2));

%LMI solution
 setlmis([]);
 C=lmivar(1,[1,0; 1,0; 1,0]);   % C=diag(c1,c2,c3)

 %LMI #1:
```

```
lmiterm([1 1 1 C],1,M_hat);                % C*M_hat
lmiterm([1 1 1 C],M_hat',1);               % M_hat'*C


%LMI #2:
lmiterm([-2 1 1 C],1,1);                       % C>0

%solving LMIs:
LMISYS=getlmis;
[tmin,Pfeas] = feasp(LMISYS)
C_value=dec2mat(LMISYS,Pfeas,C)

epsilon
```

6.2 Algorithm for solving non-strict LMI

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Algorithm for A'C+CA<0 with a numerical algorithm
% Last modified: 2004.03.09
% Input: - number of iterations
%        - QP coefficient matrix A
%        - QP exponent matrix B
% Output: diagonal matrix C for the Lyapunov function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function numLMI(n,A,B);
global M;
no_iter=n;
M=B*A;
M_size=size(M,1);
a=zeros(no_iter,M_size);
%initial value
a0=0.5*ones(1,M_size);
```

```
a(1,:)=a0;
egy=phi(a0);
phi_m=zeros(no_iter,1);

        %main loop of the algorithm
for i=2:no_iter
   res=phi(a(i-1,:));
   phi_m(i-1)=res;
   %if the result is good
   if res<=0
      break
   end

   %Else jump into the calculations
   %parameters of the linear programming problem
   Aeq=[];
   Beq=[];
   %upper and lower bounds
   tmp1=2*sqrt(M_size)*norm(M);
   tmp2=ones(M_size,1);
   UB=[tmp1;tmp2];
   LB=zeros(M_size+1,1);
   %additional constraints
   tmp=zeros(M_size,1);
   f=[-1;tmp];
   Ac=zeros(i-1,M_size+1);
   Bc=zeros(i-1,1);
   for j=1:i-1
      tmp1=gM(a(j,:));
      tmp2=[1,tmp1'];
      Ac(j,:)=tmp2;
      Bc(j)=phi(a(j,:));
   end
```

```
   %call the lin.solver
   sol=LINPROG(f,Ac,Bc,Aeq,Beq,LB,UB);
   a_star=sol(2:M_size+1,:)
   a(i,:)=(a(i-1,:)+a_star')/2;
end

phi_m(i)=phi(a_star');

%Minimal error
min_index=1 phi_min=phi_m(1) for i=1:no_iter
   if phi_m(i)<phi_min
      phi_min=phi_m(i);
      min_index=i;
   end
end

%select the 'best' A
Af=diag(a(min_index,:)) %the result
```

6.2.1 Function gM

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Function gM()
% Last modified: 2004.03.09
% Input: - a
% Output: gM(a)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function out=gM(a) global M
%M=[-1 0 0 ;0 -2 0 ;0 0 -1.5 ];
A=diag(a); W=A*M+M'*A;

[V, D]=eig(W);
%Find the maximal eigenvalue
```

```
n=size(W,1); m=D(1,1); imax=1; for i=2:n
   if D(i,i)>m
      m=D(i,i);
      imax=i;
   end
end
%Get the corresponding eigenvector
v_max=V(:,imax);
Lambda=diag(v_max);
out=2*Lambda*M*v_max;
```

References

[1] B. Hernandez-Bermejo, V. Fairen: Lotka-Volterra Representation of General Nonlinear Systems, *Mathematical Biosciences*, **140**, 1-32 (1997)

[2] B. Hernandez-Bermejo, V. Fairen: Nonpolynomial Vector Fields Under the Lotka-Volterra Normal Form, *Physics Letters A*, **206**, 31-37 (1995)

[3] R. Diaz-Sierra, B. Hernandez-Bermejo, V. Fairen: Graph-theoretic Description of the Interplay Between Non-linearity and Connectivity in Biological Systems, *Mathematical Biosciences*, **156**, 229-253 (1999)

[4] A. Figueiredo, I. M. Gleria, T. M. Rocha Filho: Boundedness of Solutions and Lyapunov Functions in Quasi-polynomial Systems, *Physics Letters A*, **268**, 335-341 (2000)

[5] A. J. Magyar, K. M. Hangos: Lotka-Volterra Representation of Process Systems for Stability Analysis, *Process Control '03 International Conference*, Paper no. 121 (2003)

[6] B. Hernandez-Bermejo: Stability Conditions and Lyapunov Functions for Quasi-polynomial Systems, *Applied Mathematics Letters*, **15**, 25-28 (2002)

[7] K. M. Hangos, I.T. Cameron: *Process Modelling and Model Analysis*, Academic Press, London (2001)

[8] B. Pongrácz, G. Szederkényi, K. M. Hangos: The Effect of Algebraic Equations on the Stability of Process Systems Modelled by Differential Algebraic Equations, *Research Report SCL-003/2002*, Systems and Control Laboratory, Computer and Automation Research Institute HAS (2002) http://daedalus.scl.sztaki.hu

[9] P. Gahinet, A. Nemirovski, A. J. Laub, M. Chilali: LMI Control Toolbox User's Guide (1995)

[10] C. Scherer, S. Weiland: Lecture Notes DISC Course on Linear Matrix Inequalities in Control (1999)

[11] A. Figueiredo, I. M. Gleria, T. M. Rocha Filho: A numerical method for the stability analysis of quasi-polynomial vector fields, *Nonlinear Analysis*, **52**, 329-342 (2003)

[12] K. M. Hangos, J. Bokor, G. Szederkényi: *Analysis and Control of Nonlinear Process Systems*, Springer-Verlag, (2004)

[13] J. G. VanAntwerp, R. D. Braatz: A tutorial on linear and bilinear matrix inequalities, *Journal of Process Control*, **10**, 363-385 (2000)

[14] W. Lohmiller, J. E. Slotine: On Contraction Analysis for Non-linear Systems, *Automatica*, **34**, 683-696 (1998)