Institute of Information Engineering, Automation and Mathematics
Department of Information Engineering and Process Control
Faculty of Chemical and Food Technology
Slovak University of Technology in Bratislava
Radlinského 9, 812 37 Bratislava, Slovak Republic

Global Optimization of Processes

Bc. Radoslav Paulen

Diploma work

Supervisor:
prof. Dr. Ing. Miroslav Fikar, STU Bratislava
Consultant:
prof. M.A. Latifi, ENSIC Nancy

I want to express my sincere gratitude to my thesis supervisor and the head of the Institute of Information Engineering, Automation and Mathematics at Faculty of Chemical and Food Technology of Slovak University of Technology in Bratislava, prof. Dr. Ing. Miroslav Fikar, for given me the opportunity to became an Erasmus student at foreign university in Nancy (France) and his patient guidance throughout my studies. My big gratification and appreciation goes to my thesis consultant in Nancy, prof. M. Abderazzak Latifi from ENSIC–INPL, who instructed me on my research journey and gave me support I needed. Also I would like to express my thank to Ing. Michal Čižniar, for his guidance, comments, tireless support, and debates that inspired me a lot. I am grateful also to all the people, who helped me in any way throughout my studies and spent their precious time for me, whenever a I had some problem. Last, but not least, I would like to thank to members of my family for standing behind me and supporting my studies in many ways.

Bratislava, 2008
Radoslav Paulen

Abstract

This work deals with the problem of global optimization of processes (GOP). As a result of dynamic modeling of processes in chemical technology we usually obtain the set of differential equations. Optimization of these processes is often complicated by the presence of non-convexities. In recent years many methods were introduced, which solve global optimization problems. In this work deterministic spatial branch and bound optimization algorithm is used for finding the solution of GOP. Three techniques for computing bounds on variables participating in convex relaxation of original problem are proposed. Selected examples are solved for problems with processes described by ordinary differential parameter-dependent equations.

Abstrakt

Táto práca sa zaoberá problémom globálnej optimalizácie procesov. Výsledkom dynamického modelovania procesov chemickej technológie je obyčajne systém diferenciálnych rovníc. Optimalizácia týchto procesov je často komplikovaná prítomnosťou nekonvexností. V posledných rokoch bolo uvedených niekoľko metód, ktoré riešia problémy globálnej optimalizácie. V tejto práci je pre nájdenie riešenia globálnej optimalizácie procesov používaný optimalizačný algoritmus založený na priestorovej metóde vetiev a hraníc. Sú navrhnuté tri metódy pre výpočet hraníc premenných participujúcich na konvexnej relaxácii originálneho problému. Vyriešené sú vybrané príklady pre problémy s procesmi opísanými obyčajnými diferenciálnymi rovnicami závislými na parametroch.

Contents

1 Introduction **7**

2 Problem Statement **10**

2.1 Dynamic Process Models 10

2.2 Process Constraints 11

2.3 Objective Function 11

2.4 Dynamic Optimization Problem 12

3 Convex Relaxation **14**

3.1 Reformulation of NLP Problem 14

3.2 Convex Relaxation of Algebraic Functions 14

3.3 Bounds on Variables and Relaxation of Dynamic Information 17

3.3.1 Bounding the Solution of IVP 17

3.3.2 Convex Relaxation of Dynamic Information 20

3.3.3 Comparison of the Two Bounding Strategies 24

3.3.4 Convex Relaxation of the NLP 25

4 Global optimization algorithm **26**

5 Examples **29**

5.1 Example 1: A Simple Dynamic Optimization Problem 29

5.2 Example 2: A First–order Irreversible Liquid–phase Reaction 30

5.3 Example 3: Catalytic Cracking of Gas Oil 32

5.4 Example 4: A First–order Reversible Liquid–phase Reaction 34

6 Conclusion **38**

Bibliography **40**

List of Figures

2.1 Two–dimensional function with multiple optima 12

3.1 State variable trajectories for different values of p 18
3.2 Parameter independent bounds on the solution space. 19
3.3 Parameter independent and dependent bounds for $p = 0$ 21
3.4 Illustrative example of comparison of constant and affine bounds 25

5.1 Graphical results for example 2 32
5.2 Graphical results for example 3 34
5.3 Graphical results for example 4 36

Chapter 1

Introduction

One of the most fundamental principles in our world is the search for an optimal state. It begins in the microcosm where atoms in physics try to form bonds in order to minimize the energy of their electrons. When molecules form solid bodies during the process of freezing, they try to assume energy optimal crystal structures. These processes, of course, are not driven by any higher intention but purely result from the laws of physics.

Exactly the same goes for the biological principle of survival of the fittest, which, together with the biological evolution, leads to better adaptation of the species to their environment. Here, a local optimum is a well–adapted species that dominates all other creatures in its surroundings. In some manner mankind can be considered as the species which reached the point of global optimum as a fittest animal on earth. Human science is now so developed that the optimization problems became the part of daily life of every researcher, developer or designer in engineering, computational chemistry, finance and medicine amongst many other fields. In many practical situations, there are several possible actions, and the best one must be chosen. For example, the best design of an object, or the best control of a plant must be found. The set of possible actions is usually characterized by parameters $p = (p_1, \ldots, p_n)$, and the result of different actions (controls) is characterized by an objective function $f(p)$.

In some cases, the objective function describes losses or expenses; in such cases, the problem of finding the best action (design or control) can be described as the problem of global minimization, i.e., the problem of finding the values p for which the function $f(p)$ attains the smallest possible value. In other cases, the objective function describes gain; in such cases, the problem of finding the best action can be described as the problem of global maximization, i.e., the problem of finding the values p for which the function $f(p)$ attains the largest possible value. Similar problems arise in data processing, when we have a model characterized by several parameters which provide the best fit of the data, i.e., for which the discrepancy $f(p)$ between data and the model is the smallest possible. Due

to the fact that processes are usually described by set of differential equations, methods of dynamic optimization must be used to find (local) solution of optimization of these processes. The techniques utilized for solving dynamic optimization problems fall under two broad frameworks: variational methods and discretization methods.

The first technique, the variational approach, encompasses the classical methods of the calculus of variations and many of the modern methods of optimal control. These methods approach the problem in the original infinite dimensional space and attempt to determine stationary functions via the solution of the Euler–Lagrange equations. The variational approach for solving dynamic optimization problems is extremely attractive because by addressing the optimization problem in the infinite dimensional space, the problem can be solved in its original form without any mathematical transformations. In addition to the variational approach for solving dynamic optimization problems, another approach exists based on discretization. While discretization has the disadvantage that it is only an approximation of the infinite dimensional problem, it possesses the tremendous advantage that it transforms the original infinite dimensional problem into a problem defined at least partially in a finite space. Therefore, the problem can often be solved by standard nonlinear programming (NLP) methods. Discretization can be subdivided into two broad classifications known as simultaneous and sequential. The simultaneous method is a complete discretization of both state and control variables, often achieved via collocation (Tsang et al., 1975). While completely transforming a dynamic system into a system of algebraic equations eliminates the problem of optimizing in an infinite dimensional space, simultaneous discretization has the unfortunate side effect of generating a multitude of additional variables yielding large, unwieldy nonlinear programs (NLPs) that are often impractical to solve numerically.

Sequential discretization is usually achieved via control parametrization, (Brusch and Schappelle, 1973) in which the control variable profiles are approximated by a series of basis functions in terms of a finite set of real parameters. These parameters then become the decision variables in a dynamic embedded NLP. Function evaluations are provided to this NLP via numerical solution of a fully determined initial value problem (IVP), which is given by fixing the control profiles. This method has the advantages of yielding a relatively small NLP and exploiting the robustness and efficiency of modern IVP and sensitivity solvers.

For a number of years, researchers have known that dynamic optimization problems encountered in chemical engineering applications exhibit multiple local optima. This property, which can be attributed to non–convexity of the functions participating in most chemical engineering models, implies that standard local optimization methods will often yield suboptimal solutions to problems. Suboptimality can have direct economic, safety, and environmental impacts if a suboptimal operating policy is implemented on a real process.

With the rising performance of the computers present in last few years scientists were able to create techniques which led to solutions of problems of GOP (also called the non–convex dynamic optimization). There are many possible classifications of global optimization methods, but the simplest and most fundamental divides these methods in two categories: stochastic (also called probabilistic) and deterministic ones.

For the class of stochastic approaches, such as multistart, clustering methods, variable neighbourhood search, genetic algorithms, simulated annealing and few others, which are based on the random search technique, it was proved that they cannot guarantee the global optimality of the obtained solution. Algorithms based on the deterministic approach such as generalized Benders decomposition (Geoffrion (1972), Floudas and Visweswaran (1990) and Bagajewicz and Manousiouthakis (1991)), branch and bound (Soland (1971), Ryoo and Sahinidis (1995) and Adjiman et al. (1996)) and interval analysis (Ratschek and Rokne (1988), Vaidyanathan and El-Halwagi (1994) and Han et al. (1997)). guarantee the finite ϵ–convergence (convergence to the global optimum in finite computation steps for a given finite error tolerance) and the global optimality of the obtained solution. Therefore, although they take more computational effort, these methods are more interesting. Subclass branch and bound (BB) methods, which are based on the concept of relaxations, are the most utilized and most suitable for solving GOP problems.

Spatial Branch–and–Bound (sBB) algorithms are the extension of traditional BB algorithms to continuous solution spaces. They are termed "spatial" because they successively partition the Euclidean space where the problem is defined into smaller and smaller regions where the problem is solved recursively by generating converging sequences of upper and lower bounds to the objective function value.

In this work a deterministic sBB global optimization algorithm is used for GOP problems with set of first–order parameter dependent differential equations in the constraints. The main purpose of this work is to illustrate utilization of sBB global optimization algorithm and to develop successive ways to obtain bounds on variables participating on convex relaxation of original problem. Chapter 2 gives the mathematical formulation of the problem studied. It is a non–convex minimization problem with an IVP for a set of first–order parameter dependent differential equations in the constraints. Chapter 3 discusses convex relaxation of original non–convex dynamic optimization problem. Chapter 4 presents global optimization sBB algorithm. In Chapter 5 selected examples for parameter estimation problems are solved.

Chapter 2

Problem Statement

In this chapter we describe original non–convex dynamic optimization problem. Its solution gives an upper bound for sBB algorithm.

2.1 Dynamic Process Models

A process model is in general a set of equations. These equations determine existence of the process as they bond together inputs, properties, and variables which describe behavior and outputs of the process. All of these can be considered as functions of time in dynamic modeling yielding a set of differential equations. The processes considered in this work are described by the following set of first–order parameter dependent, typically nonlinear, differential equations

$$\dot{x}(t,p) = f\big(t, x(t,p), p\big) \qquad \forall t \in \mathcal{I} \equiv [t_0, t_{NP}] \tag{2.1}$$

where $t \in \mathcal{I} \subset \mathcal{R}$, denotes time as the independent variable and NP is the number of points considered additionally to the initial point t_0, $p \subset \mathcal{R}^r$ is the vector of parameters of the process, $x \subset \mathcal{R}^n$ stands for the state variables and $\dot{x} \subset \mathcal{R}^n$ are their derivatives with respect to t[1]. The function f is such that $f : \mathcal{I} \times \mathcal{R}^n \times \mathcal{R}^p \to \mathcal{R}^n$. The solution $x(t,p)$ of this set satisfies the initial condition

$$x(t_0, p) = x_0(p) \tag{2.2}$$

where the function x_0 is such that $x_0 : \mathcal{R}^r \to \mathcal{R}^n$.

[1]It is assumed for the rest of this work that a dot over a variable represents its derivative with respect to t.

2.2 Process Constraints

Constraints are mathematical functions, which, simply speaking, determine the domain of possible values of variables participated in process. Inequality constraints can be imposed at discrete time points, t_i. These are point constraints of the form

$$g_i\big(x(t_i,p),p\big) \leq 0 \qquad i = 0,1,\ldots,NP \tag{2.3}$$

where the functions $g_i, i = 0,1,\ldots,NP$, are such that $g_i : \mathcal{R}^n \times \mathcal{R}^r \rightarrow \mathcal{R}^{s_i}$. Of course any equality point constraint can be replaced by two inequality point constraints. Lower and upper bounds are imposed on the parameters p:

$$p^L \leq p \leq p^U \tag{2.4}$$

The impact of constraint functions on solution of optimization problem will be shown in next section.

2.3 Objective Function

The objective function for a dynamic optimization problem can be expressed in terms of the values of the state variables at discrete points and of the parameters

$$J\big(x(t_i,p),p;\, i = 0,1,\ldots,NP\big) \tag{2.5}$$

The function J is such that $J : \mathcal{R}^{n(NP+1)} \times \mathcal{R}^r \rightarrow \mathcal{R}$. Integral terms that may appear in the objective function can always be eliminated by introducing additional state variables and equations in the set of differential equations. Figure 2.1 shows a simple example of a two–dimensional function which exhibit multiple optima. The region of possible values of time–dependent variables is constrained by the simple bounds given on them. This region is denoted as X. Here we can see what is the effect of process constraints on optimization results.

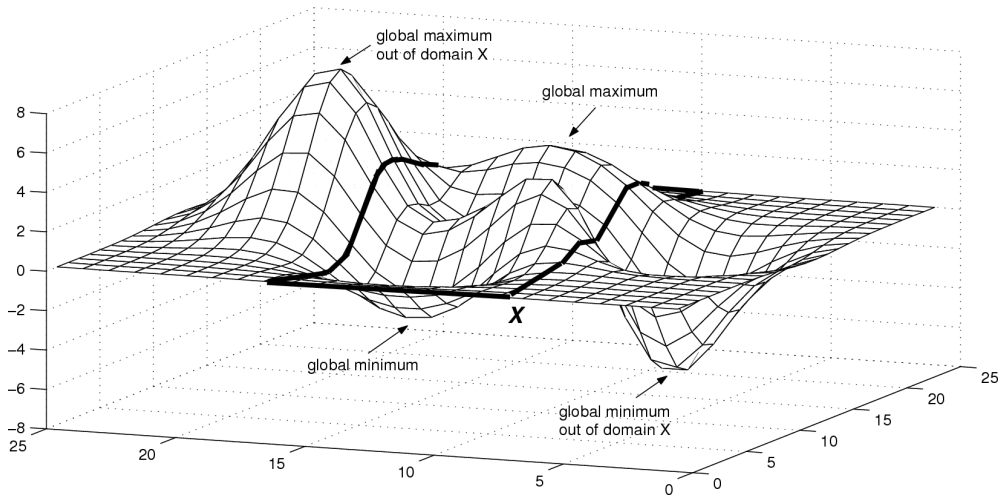


Figure 2.1: Two–dimensional function with multiple optima

2.4 Dynamic Optimization Problem

The formulation of the dynamic optimization problem studied is given by

$$\begin{aligned} \min_{p} \; & J\big(x(t_i,p),p;\, i=0,1,\ldots,NP\big) \\ s.t. \; & \dot{x}(t,p) = f\big(t,x(t,p),p\big) \qquad \forall t \in \mathcal{I} \\ & x(t_0,p) = x_0(p) \\ & g_i\big(x(t_i,p),p\big) \le 0 \qquad i=0,1,\ldots,NP \\ & p^L \le p \le p^U \end{aligned} \tag{2.6}$$

The minimization of the objective function (2.5) is considered subject to the dynamics of the system, described by IVP (2.1) and (2.2), the point constraints (2.3) and the bounds on the parameters (2.4). Systems with controls that depend on t can be transformed to this form using control parameterization (Vassiliadis et al., 1994a,b).

The following assumptions are made on the properties of the functions in (2.6):

- $J\big(x(t_i,p),p;\, i=0,1,\ldots,NP\big)$ is once continuously differentiable with respect to $x(t_i,p), i=0,1,\ldots,NP$ and p on $\mathcal{R}^{n(NP+1)} \times \mathcal{R}^r$.
- each element of $g_i\big(x(t_i,p),p\big), i=0,1,\ldots,NP$, is once continuously differentiable with respect to $x(t_i,p)$ and p on $\mathcal{R}^n \times \mathcal{R}^r$.
- each element of $f(t,x,p)$ is continuous with respect to t and once continuously differentiable with respect to x and p on $\mathcal{I} \times \mathcal{R}^n \times \mathcal{R}^r$.
- each element of $x_0(p)$ is once continuously differentiable with respect to p on $\mathcal{R}^r$.

- $f(t, x, p)$ satisfies a uniqueness condition on $\mathcal{I} \times \mathcal{R}^n \times \mathcal{R}^r$.

The sequential approach is used for the solution of this dynamic optimization problem. The gradients with respect to p can be evaluated using the parameter sensitivities. These are given from the solution of the sensitivity equations (Vassiliadis et al., 1994a,b). Due to the generally non–convex nature of the functions used in the formulation of the dynamic optimization problem, the solution obtained using the sequential approach and a standard gradient–based NLP technique, is a local optimum and therefore provides an upper bound for the global optimum solution.

Chapter 3

Convex Relaxation

As it was mentioned before, BB algorithms are operating with concept of relaxations. In this section we propose a possible convex relaxation of the non–convex dynamic optimization problem that was introduced in the previous chapter. The solution of this convex relaxation provides a lower bound for the global optimum of the non–convex problem.

3.1 Reformulation of NLP Problem

First, we the reformulate NLP problem (2.6) as

$$\begin{aligned} \min_{\hat{x},p} \; & J(\hat{x},p) \\ \text{s.t. } & g_i(\hat{x}_i,p) \leq 0 \qquad i = 0,1,\ldots,NP \\ & \hat{x}_i = x(t_i,p) \qquad i = 0,1,\ldots,NP \\ & p \in [p^L, p^U] \end{aligned} \tag{3.1}$$

where $\hat{x}$ is a vector of new added optimized variables and values of $x(t_i,p), i = 0,1,\ldots,NP$ are obtained from the solution of the IVP

$$\begin{aligned} & \dot{x} = f(t,x,p) \qquad \forall t \in \mathcal{I} \\ & x(t_0,p) = x_0(p) \end{aligned} \tag{3.2}$$

3.2 Convex Relaxation of Algebraic Functions

It is assumed that the functions J and g_{ij}, $i = 0,1,\ldots,NP$, $j = 1,2,\ldots,s_i$ can be decomposed into a sum of terms, where each term may be classified as linear (LT), convex (CT), bilinear (BT), trilinear (TT), fractional (FT), fractional trilinear (FTT) or univariate concave (UT).

Linear and convex terms do not require any transformation. For non–convex terms, except special types of non–convex terms, a convex underestimator is generated by adding a relaxation function to them (Adjiman et al., 1998b). The special types of non–convex terms (bilinear, trilinear, fractional, trilinear fractional, and univariate concave terms) can be replaced by very tight convex underestimators which are already known. The convex envelopes can be constructed by the following simple rules.

Underestimating Bilinear Terms

In the case of bilinear term xy, a tight convex lower bound over the domain $[x^L, x^U] \times [y^L, y^U]$ is obtained by introducing a new variable w_{BT} which replaces every occurence of xy in the problem and adding the following four linear inequality constraints

$$\begin{aligned} w_{BT} &\geq x^L y + x y^L - x^L y^L \\ w_{BT} &\geq x^U y + x y^U - x^U y^U \\ w_{BT} &\leq x^L y + x y^U - x^L y^U \\ w_{BT} &\leq x^U y + x y^L - x^U y^L \end{aligned} \tag{3.3}$$

Underestimating Trilinear Terms

Any trilinear term of the form xyz can be underestimated over the domain $[x^L, x^U] \times [y^L, y^U] \times [z^L, z^U]$ by introducing a new variable w_{TT} and boundig it by the following linear inequality constraints

$$\begin{aligned} w_{TT} &\geq x y^L z^L + x^L y z^L + x^L y^L z - 2 x^L y^L z^L \\ w_{TT} &\geq x y^U z^U + x^U y z^L + x^U y^L z - x^U y^L z^L - x^U y^U z^U \\ w_{TT} &\geq x y^L z^L + x^L y z^U + x^L y^U z - x^L y^U z^U - x^L y^L z^L \\ w_{TT} &\geq x y^U z^L + x^U y z^U + x^L y^U z - x^L y^U z^U - x^U y^U z^U \\ w_{TT} &\geq x y^L z^U + x^L y z^L + x^U y^L z - x^U y^L z^U - x^L y^L z^L \\ w_{TT} &\geq x y^L z^U + x^L y z^U + x^U y^U z - x^L y^L z^U - x^U y^U z U \\ w_{TT} &\geq x y^U z^L + x^U y z^L + x^L y^L z - x^U y^U z^L - x^L y^L z^L \\ w_{TT} &\geq x y^U z^U + x^U y z^U + x^U y^U z - 2 x^U y^U z^U \end{aligned} \tag{3.4}$$

Underestimating Fractional Terms

Fractional terms of the form $\frac{x}{y}$ are underestimated by introducing a new variable w_{FT} and two new inequality constraints which depend on the sign of the bounds on x

$$
\begin{aligned}
w_{FT} &\geq \begin{cases} \frac{x^L}{y} + \frac{x}{y^U} - \frac{x^L}{y^U} & \text{if } x^L \geq 0 \\ -\frac{x}{y^U} - \frac{x^L y}{y^L y^U} + \frac{x^L}{y^U} & \text{if } x^L < 0 \end{cases} \\
w_{FT} &\geq \begin{cases} \frac{x^U}{y} + \frac{x}{y^L} - \frac{x^U}{y^L} & \text{if } x^U \geq 0 \\ -\frac{x}{y^L} - \frac{x^U y}{y^L y^U} + \frac{x^U}{y^U} & \text{if } x^U < 0 \end{cases}
\end{aligned} \tag{3.5}
$$

Underestimating Fractional Trilinear Terms

For fractional trilinear term of the form $\frac{xy}{z}$, an underestimator is derived by introducing a new variable w_{FTT} by the following inequality constraints for $x^L, y^L, z^L \geq 0$:

$$
\begin{aligned}
w_{FTT} \geq & \frac{xy^L}{z^U} + \frac{x^L y}{z^U} + \frac{x^L y^L}{z} - 2\frac{x^L y^L}{z^U} \\
w_{FTT} \geq & \frac{xy^L}{z^U} + \frac{x^L y}{z^L} + \frac{x^L y^U}{z} - \frac{x^L y^U}{z^L} - \frac{x^L y^L}{z^U} \\
w_{FTT} \geq & \frac{xy^U}{z^L} + \frac{x^U y}{z^U} + \frac{x^U y^L}{z} - \frac{x^U y^L}{z^U} - \frac{x^U y^U}{z^L} \\
w_{FTT} \geq & \frac{xy^U}{z^U} + \frac{x^U y}{z^L} + \frac{x^L y^U}{z} - \frac{x^L y^U}{z^U} - \frac{x^U y^U}{z^L} \\
w_{FTT} \geq & \frac{xy^L}{z^U} + \frac{x^L y}{z^L} + \frac{x^U y^L}{z} - \frac{x^U y^L}{z^L} - \frac{x^L y^L}{z^U} \\
w_{FTT} \geq & \frac{xy^U}{z^U} + \frac{x^U y}{z^L} + \frac{x^L y^U}{z} - \frac{x^L y^U}{z^U} - \frac{x^U y^U}{z^L} \\
w_{FTT} \geq & \frac{xy^L}{z^U} + \frac{x^L y}{z^L} + \frac{x^U y^L}{z} - \frac{x^U y^L}{z^L} - \frac{x^L y^L}{z^U} \\
w_{FTT} \geq & \frac{xy^U}{z^L} + \frac{x^U y}{z^L} + \frac{x^U y^U}{z} - 2\frac{x^U y^U}{z^L}
\end{aligned} \tag{3.6}
$$

Underestimating Univariate Concave Terms

Univariate concave terms are trivially underestimated by their linearization at the lower bound of the variable range. Thus the convex envelope of the concave function $UT(x)$ over $[x^L, x^U]$ is the following linear function of x:

$$
UT(x^L) + \frac{UT(x^U) - UT(x^L)}{x^U - x^L}(x - x^L) \tag{3.7}
$$

An overall convex underestimator is given by the summation of the convex underestimators for each term in the function. Therefore, introduction of additional constraints is required for the special type of non–convex terms.

3.3 Bounds on Variables and Relaxation of Dynamic Information

It is very useful and in many cases essential to have bounds on variables, which are participating in optimization problem. This section discusses the possible techniques, which can be used to obtain bounds on these variables. For case of problem (3.1) bounds on parameters p are user–defined and bounds on variables $\hat{x}_i$ can depend on these parameters or just on bounds of these parameters, as it will be shown in next section. Within the generation of bounds on $\hat{x}_i$, which will definitely replace the presence of dynamic information in (3.1) relaxation of dynamic information will be formed.

3.3.1 Bounding the Solution of IVP

The dependence of convex relaxations on variable bounds is a common feature of deterministic global optimization algorithms. Since state variables appear in the non–convex objective function and constraints, a method for the derivation of rigorous bounds on these variables at point $t_i, i = 0, 1, \ldots, NP$, is needed. This issue can be resolved by generating bounds on the solution space of the dynamic system. Consider an example of the following IVP for a first–order parameter dependent differential equation

$$\begin{aligned} \dot{x} &= -x^2 + p \qquad \forall t \in [0, 1] \\ x(t = 0, p) &= 9 \end{aligned} \tag{3.8}$$

where $p \in [-5, 5]$. The solution $x(t, p)$ of this IVP for different values of p is shown in Fig. 3.1. As it can be seen, solutions of IVP for the upper bound and the lower bound of p give bounds on the trajectories. According to (Papamichail and Adjiman, 2002) this section introduces a systematic approach for the derivation of such bounds, applicable to IVPs for a system of generally nonlinear first–order parameter dependent differential equations. Notation $f(t, x, p) = f(t, x_k, x_{k-}, p)$, which is similar to (Papamichail and Adjiman, 2002) is used, where the following is considered: $x = (x_1, x_2, \ldots, x_n)^T$ and $x_{k-} = (x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)^T$.

Parameter Independent Bounds

Lower and upper parameter independent bounds can be determined for the solution $x(t, p)$ of IVP (3.2) such that $\underline{x}(t) \leq x(t, p) \leq \bar{x}(t) \quad \forall p \in [p^L, p^U] \quad \forall t \in \mathcal{I}$ where the inequalities are understood component–wise. Considering the assumptions and theorem given in Papamichail and Adjiman (2002) it can be assumed that, if f is continuous and satisfies a

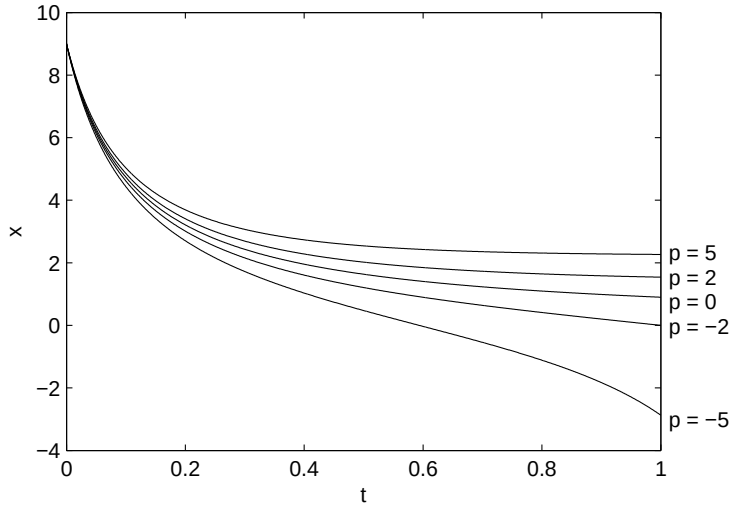


Figure 3.1: State variable trajectories for different values of p

uniqueness condition on $\mathcal{I}_0 \equiv (t_0, t_{NP}] \times \mathcal{R}^n \times [p^L, p^U]$, then the solution $\underline{x}(t)$ and $\bar{x}(t)$ of the following IVP satisfies

$$\begin{aligned}
\dot{\underline{x}}_k &= \inf f_k(t, \underline{x}_k, [\underline{x}_{k-}, \bar{x}_{k-}], [p^L, p^U]) \\
&\qquad\qquad \forall t \in \mathcal{I} \quad k = 1, 2, \ldots, n \\
\dot{\bar{x}}_k &= \sup f_k(t, \bar{x}_k, [\underline{x}_{k-}, \bar{x}_{k-}], [p^L, p^U]) \\
&\qquad\qquad \forall t \in \mathcal{I} \quad k = 1, 2, \ldots, n \\
\underline{x}(t_0) &= \inf x_0([p^L, p^U]) \\
\bar{x}(t_0) &= \sup x_0([p^L, p^U])
\end{aligned} \tag{3.9}$$

These IVPs provide a practical procedure to construct bounding trajectories for IVP (3.2) if the appropriate continuity and uniqueness conditions are satisfied. Natural interval extensions are used as inclusion functions (Nickel, 1986).

Example (3.8) which was already introduced at the beginning of this section is reconsidered here. Based on (3.9), IVPs whose solutions $\underline{x}(t)$ and $\bar{x}(t)$ can give bounds on the solution of IVP (3.8) $\forall v \in [-5, 5]$ are constructed. The subfunction is given by

$$\begin{aligned}
\dot{\underline{x}} &= -\underline{x}^2 - 5 \qquad \forall t \in [0, 1] \\
\underline{x}(0) &= 9
\end{aligned} \tag{3.10}$$

and the superfunction is given by

$$\begin{aligned}
\dot{\bar{x}} &= -\bar{x}^2 + 5 \qquad \forall t \in [0, 1] \\
\bar{x}(0) &= 9
\end{aligned} \tag{3.11}$$

The solutions of these bounding IVPs are shown in Fig. 3.2. They enclose the solution space of the original IVP for the parameter dependent differential equation.

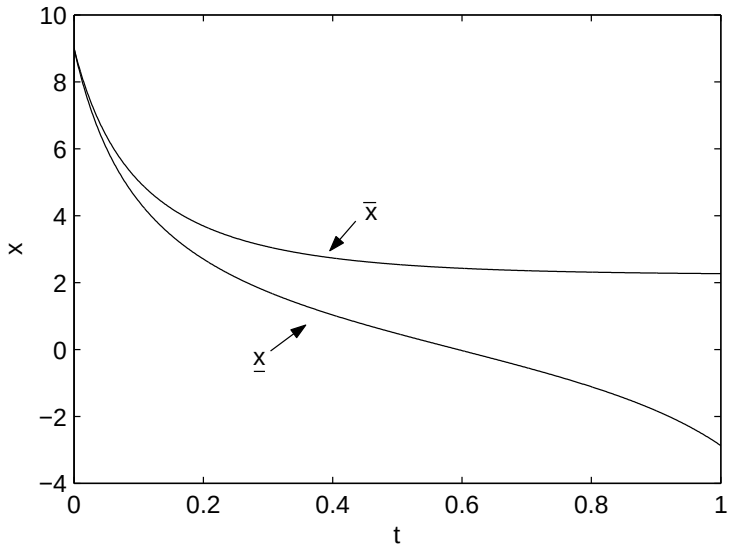


Figure 3.2: Parameter independent bounds on the solution space.

Parameter Dependent Bounds

Using the parameter independent bounds already presented, lower and upper parameter dependent bounds can be determined for the solution $x(t,p)$ of IVP (3.2) such that $\underline{\underline{x}}(t,p) \leq x(t,p) \leq \bar{\bar{x}}(t,p) \quad \forall p \in [p^L, p^U]\ \forall t \in \mathcal{I}$. For functions $\underline{f}(t,x,p)$ and $\bar{f}(t,x,p)$ we consider the same assumptions as for function $f(t,x,p)$, which participates in IVP (3.2). Following the theorems introduced by Papamichail and Adjiman (2004) it can be written

$$\begin{aligned} &\underline{f}(t,x,p) \leq f(t,x,p) && \forall x \in [\underline{x}(t), \bar{x}(t)] \\ &&& \forall p \in [p^L, p^U]\ \forall t \in \mathcal{I} \\ &\underline{\underline{x}}_0(p) \leq x_0(p) && \forall p \in [p^L, p^U] \end{aligned} \tag{3.12}$$

then the solution $x(t,p)$ of the IVP

$$\begin{aligned} &\dot{\underline{\underline{x}}} = \underline{f}(t, \underline{\underline{x}}, p) && \forall t \in \mathcal{I} \\ &\underline{\underline{x}}(t_0, p) = \underline{\underline{x}}_0(p) \end{aligned} \tag{3.13}$$

is such that

$$\underline{\underline{x}}(t,p) \leq x(t,p) \qquad \forall p \in [p^L, p^U]\ \forall t \in \mathcal{I} \tag{3.14}$$

where $x(t,p)$ is the solution of IVP (3.2). And in the same manner we can write

$$\begin{aligned} &\bar{f}(t,x,p) \geq f(t,x,p) && \forall x \in [\underline{x}(t), \bar{x}(t)] \\ &&& \forall p \in [p^L, p^U]\ \forall t \in \mathcal{I} \\ &\bar{\bar{x}}_0(p) \geq x_0(p) && \forall p \in [p^L, p^U] \end{aligned} \tag{3.15}$$

The solution $x(t,p)$ of the IVP

$$\begin{aligned} &\dot{\bar{\bar{x}}} = \bar{f}(t, \bar{\bar{x}}, p) && \forall t \in \mathcal{I} \\ &\bar{\bar{x}}(t_0, p) = \bar{\bar{x}}_0(p) \end{aligned} \tag{3.16}$$

is such that

$$\bar{\bar{x}}(t,p) \geq x(t,p) \qquad \forall p \in [p^L, p^U] \; \forall t \in \mathcal{I} \tag{3.17}$$

In what follows, parameter dependent bounds are constructed for the example already discussed above. The function $f(t,x,v) = -x^2 + p$ can be underestimated by the function $\underline{f}(t,x,p) = -(\underline{x} + \bar{x})\underline{\underline{x}} + \underline{x}\bar{x} + p \quad \forall x \in [\underline{x}, \bar{x}] \; \forall p \in [-5,5] \quad \forall t \in [0,1]$.

Previously mentioned solution $x(t,v)$ of the IVP

$$\begin{aligned} \dot{\underline{\underline{x}}} &= -(\underline{x} + \bar{x})\underline{\underline{x}} + \underline{x}\bar{x} + p \qquad \forall t \in [0,1] \\ \underline{\underline{x}}(0,p) &= 9 \end{aligned} \tag{3.18}$$

is such that

$$\underline{\underline{x}}(t,p) \leq x(t,p) \qquad \forall p \in [-5,5] \; \forall t \in [0,1] \tag{3.19}$$

where $\underline{x}(t)$ and $\bar{x}(t)$ are given from the solutions of IVPs (3.10) and (3.11) and $x(t,p)$ is the solution of IVP (3.8). Following previously proposed approach it can be shown that the solution $\bar{\bar{x}}_1(t,p)$ of the IVP

$$\begin{aligned} \dot{\bar{\bar{x}}}_1 &= -2\underline{x}\bar{\bar{x}}_1 + \underline{x}^2 + p \qquad \forall t \in [0,1] \\ \bar{\bar{x}}_1(0,p) &= 9 \end{aligned} \tag{3.20}$$

is such that

$$\bar{\bar{x}}_1(t,p) \geq x(t,p) \qquad \forall p \in [-5,5] \; \forall t \in [0,1] \tag{3.21}$$

and the solution $\bar{\bar{x}}_2(t,p)$ of the IVP

$$\begin{aligned} \dot{\bar{\bar{x}}}_2 &= -2\bar{x}\;\bar{\bar{x}}_2 + \bar{x}^2 + p \qquad \forall t \in [0,1] \\ \bar{\bar{x}}_2(0,p) &= 9 \end{aligned} \tag{3.22}$$

is such that

$$\bar{\bar{x}}_2(t,p) \geq x(t,p) \qquad \forall p \in [-5,5] \; \forall t \in [0,1] \tag{3.23}$$

All these parameter dependent bounds, for $p = 0$, together with the parameter independent ones are shown in Fig. 3.3. It can be observed that the second overestimator is tighter than the first one for all $t \in [0,1]$. Although the first overestimator crosses overestimator, constructed as the parameter independent upper bound, at $t = 0.8$, based on (3.14), it is still a valid overestimator.

3.3.2 Convex Relaxation of Dynamic Information

The set of equalities in (3.1) can be written as two sets of inequalities

$$\begin{aligned} \hat{x}_i - x(t_i,p) &\leq 0 \qquad i = 0,1,\ldots,NP \\ x(t_i,p) - \hat{x}_i &\leq 0 \qquad i = 0,1,\ldots,NP \end{aligned} \tag{3.24}$$

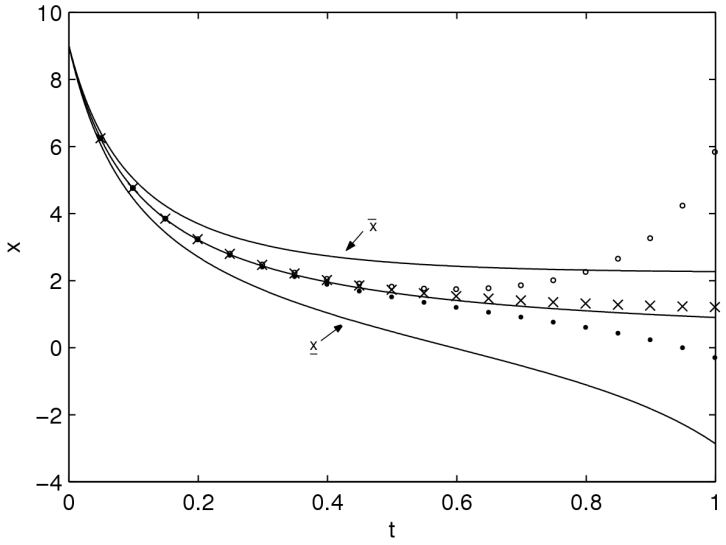


Figure 3.3: Parameter independent and dependent bounds for $p = 0$. The solution of $x(t, 0)$ (-), the underestimator (.), the first overestimator (o) and the second overestimator (x) are shown

Their relaxation is given by

$$\hat{x}_i + \breve{x}^-(t_i, p) \leq 0 \qquad i = 0, 1, \ldots, NP \tag{3.25}$$

$$\breve{x}(t_i, p) - \hat{x}_i \leq 0 \qquad i = 0, 1, \ldots, NP \tag{3.26}$$

where $\breve{x}$ denotes the convex underestimator of the specified function and $x^-(t_i, p) = -x(t_i, p)$. Thus, the function $\breve{x}(t_i, p)$ is a convex underestimator of $x(t_i, p)$ and the function $-\breve{x}^-(t_i, p)$ is a concave overestimator of $x(t_i, p)$. The generation of these under and overestimators is the most challenging step in the construction of the convex relaxation of the problem because no analytical form is available for $x(t_i, p)$. Next sections discuss developed strategies for obtaining these estimators.

Constant Bounds

The constant bounds are given by inequalities

$$\underline{x}(t_i) \leq \hat{x}_i \leq \bar{x}(t_i) \qquad i = 0, 1, \ldots, NP \tag{3.27}$$

These inequalities are valid convex underestimators and concave overestimators for $x(t_i, p)$ and therefore they can replace inequalities (3.25) and (3.26). These bounds do not depend on the parameters p themselves, but do depend on the bounds on p.

Affine Bounds

A tighter convex relaxation may be derived if parameter dependent under and overestimators can be constructed. This issue is tackled in this section, where affine functions of the parameters are obtained. If $\underline{f}(t,\underline{\underline{x}},p) = \underline{A}(t)\underline{\underline{x}} + \underline{B}(t)p + \underline{C}(t)$ and $\underline{\underline{x}}_0(p) = \underline{D}p + \underline{E}$, where $\underline{A}(t)$, $\underline{B}(t)$ and $\underline{C}(t)$ are continuous on $\mathcal{I}$, then from linear systems theory (Zadeh and Desoer, 1963) the solution of IVP (3.13) is given by

$$\underline{\underline{x}}(t,p) = \left\{\Phi(t,t_0)\underline{D} + \int_{t_0}^{t}\Phi(t,\tau)\underline{B}(\tau)d\tau\right\}p + \Phi(t,t_0)\underline{E} + \int_{t_0}^{t}\Phi(t,\tau)\underline{C}(\tau)d\tau \tag{3.28}$$

where $\Phi(t,t_0)$ is the transition matrix, which is the solution of the IVP

$$\begin{aligned}&\Phi(t,t_0) = \underline{A}(t)\Phi(t,t_0) \qquad \forall t\in\mathcal{I}\\ &\Phi(t_0,t_0) = I\end{aligned} \tag{3.29}$$

and I is the identity matrix. From (3.28), it is clear that $\underline{\underline{x}}(t,p)$ is an affine function of p of the form:

$$\underline{\underline{x}}(t,p) = \underline{M}(t)p + \underline{N}(t) \tag{3.30}$$

where $\underline{M}(t)$ is an $n\times r$ matrix and $\underline{N}(t)$ is an $n\times 1$ matrix. In the same manner, if there exist functions $\bar{f}(t,\bar{\bar{x}},p) = \bar{A}(t)x + \bar{B}(t)p + \bar{C}(t)$ and $\bar{\bar{x}}_0(p) = \bar{D}p + \bar{E}$, where $\bar{A}(t)$, $\bar{B}(t)$ and $\bar{C}(t)$ are continuous on I, then the solution of IVP (3.16) is of the form:

$$\bar{\bar{x}}(t,p) = \bar{M}(t)p + \bar{N}(t) \tag{3.31}$$

where $\bar{M}(t)$ is an $n\times r$ matrix and $\bar{N}(t)$ is an $n\times 1$ matrix. If the functions $\underline{f}$, $\underline{\underline{x}}_0$, $\bar{f}$ and $\bar{\bar{x}}_0$ defined above satisfy the conditions (3.12) and (3.15), then (3.30) and (3.31) provide parameter dependent bounds for the solution of IVP (3.2). Functions $\underline{f}$, $\underline{\underline{x}}_0$, $\bar{f}$ and $\bar{\bar{x}}_0$ that satisfy at least the inequality conditions can easily be constructed for dynamic systems with functions f and x_0 which can be decomposed into a sum of linear, bilinear, trilinear, univariate convex and univariate concave terms. The matrices $\underline{A}$ to $\underline{E}$ and $\bar{A}$ to $\bar{E}$ needed for the construction of these functions usually depend on the bounds p^L, p^U, $\underline{x}(t)$ and $\bar{x}(t)$. However, the functional form of $\underline{x}(t)$ and $\bar{x}(t)$ is not known and the matrices needed in (3.30) and (3.31) cannot be calculated analytically. The affine bounds for $t = t_i$ can be used for the convex underestimation of $x(t_i,p)$ and $x^-(t_i,p)$ over the domain $[p^L,p^U]\subset\mathcal{R}^r$

$$\begin{aligned}&\breve{x}(t_i,p) = \underline{M}(t_i)p + \underline{N}(t_i) \qquad i = 0,1,\ldots,NP\\ &\breve{x}^-(t_i,p) = -\bar{M}(t_i)p - \bar{N}(t_i) \qquad i = 0,1,\ldots,NP\end{aligned} \tag{3.32}$$

$\underline{M}(t_i)$ and $\underline{N}(t_i)$, $i = 0,1,\ldots,NP$ can be calculated from the solution of the linear system produced when (3.30) is applied for $t = t_i$ and for $r+1$ values of p chosen such that

$$\det\left(\begin{bmatrix} p_1 & p_2 & \ldots & p_{r+1}\\ 1 & 1 & \ldots 1 & \end{bmatrix}\right) \neq 0 \tag{3.33}$$

This system has the form $\underline{\underline{x}}(t_i, p_j) = \underline{M}(t_i)p_j + \underline{N}(t_i),\ j = 1, \ldots, r+1$. Condition (3.33) ensures that a unique solution exists. The values of $\underline{\underline{x}}(t_i, p_j),\ j = 1, \ldots, r+1$ are given from the numerical solution of IVP (3.13) coupled, when necessary, with IVP (3.9). In the same manner, $\bar{M}(t_i)$ and $\bar{N}(t_i),\ i = 0, 1, \ldots, NP$ can be calculated.

α–based Bounds

An alternative way to generate the underestimators needed has been proposed by Esposito and Floudas (2000a,b,c). Based on assumptions in Papamichail and Adjiman (2002), $x(t_i, p)$ is a twice continuously differentiable function of the parameters p on $\mathcal{R}^r$. This means that the α–based underestimators can be used for the convex underestimation of $x(t_i, p)$ and $x^-(t_i, p)$ over the domain $[p^L, p^U] \subset \mathcal{R}^r$

$$\breve{x}_k(t_i, p) = x_k(t_i, p) + \sum_{j=1}^{r} \alpha^+_{kij}(p^L_j - p_j)(p^U_j - p_j) \quad i = 0, 1, \ldots, NP \qquad (3.34)$$
$$k = 1, 2, \ldots, n$$

$$\breve{x}^-_k(t_i, p) = x^-_k(t_i, p) + \sum_{j=1}^{r} \alpha^-_{kij}(p^L_j - p_j)(p^U_j - p_j) \quad i = 0, 1, \ldots, NP \qquad (3.35)$$
$$k = 1, 2, \ldots, n$$

The difficulty in this approach is the calculation of the non–negative α^+_{kij} and α^-_{kij} parameters. There is no functional form available for the Hessian matrices in order to use interval calculations directly. As suggests Adjiman et al. (1998a) for underestimation of non–convex terms $f_{NT}(z)$, values of non–negative parameters α are calculated using scaled Gerschgorin method. This method requires the use of a symetric interval matrix $[H_{fNT}] = ([\underline{h}_{ij}, \bar{h}_{ij}])$ such that $[H_{fNT}] \ni H_{fNT}(z) = \nabla^2 f_{NT}(z),\ \forall z \in [z^L, z^U] \subset \mathcal{R}^m$. Then values of α can be calculated by the following formula

$$\alpha_i = \max\left\{0, -\frac{1}{2}\left(\underline{h}_{ii} - \sum_{j \neq i} |h|_{ij}\right)\right\} \quad i, j = 1, 2, \ldots, m \qquad (3.36)$$

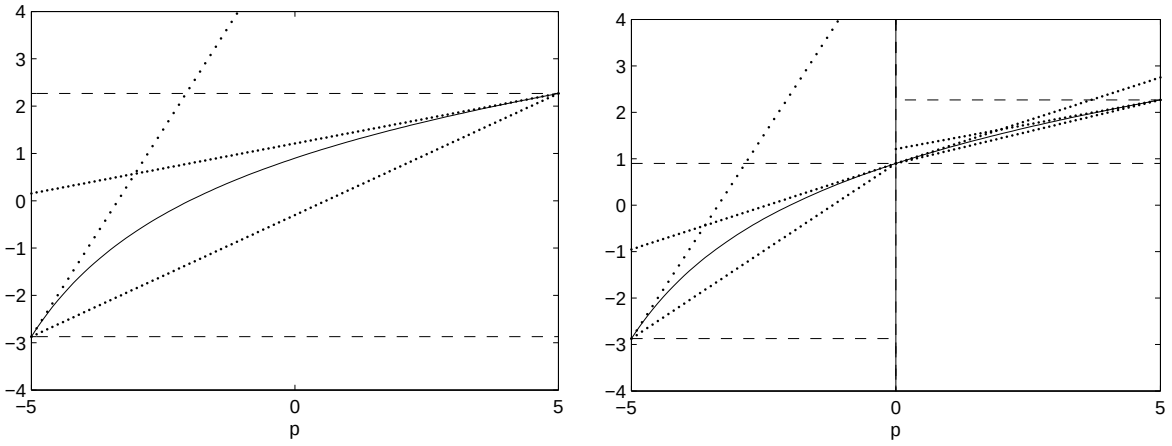
where $|h|_{ij} = \max\{|\underline{h}_{ij}|, |\bar{h}_{ij}|\}$. The interval matrix $[H_{fNT}]$ is calculated by applying natural interval extensions to the analytical expression for each second–order derivative of f_{NT}.

The calculation of the required interval matrices is given as $[H_{x_k(t_i)}] \ni H_{x_k(t_i)}(p) = \nabla^2 x_k(t_i, p), \forall p \in [p^L, p^U]$ and $[H^-_{x_k(t_i)}] = -[H_{x_k(t_i)}]$. Esposito and Floudas (2000a,b,c) proposed three methods based on sampling. Using this local information, the authors show that the ability of the algorithm to identify the global solution depends on the value of the parameters, which must be large enough for the lower bounding problem to have a unique

solution. As a result, they find that the number of sample points used affects the convexity of the underestimator. Thus, the method using interval calculations produces an interval matrix $[H^*]$, that may be an underestimation of the space of the Hessian matrices. This means that there may exist $p \in [p^L, p^U] : \nabla^2 x_k(t_i, p) = H_{x_k(t_i)}(p) \notin [H^*]$. A rigorous procedure is proposed in work of Papamichail and Adjiman (2004) for the calculation of the α^+_{kij} and α^-_{kij} parameters. The scaled Gerschgorin method put forward by Adjiman et al. (1998a) and formulas similar to (3.36) can be utilized again. The difficulties associated with the computation of valid Hessian matrices are resolved by constructing bounds using (3.9) for the IVP that is generated when the first and the second–order sensitivity equations are coupled with the original IVP (3.2). These bounds on the second–order derivatives can then be used to construct each element of the interval Hessian matrices needed.

3.3.3 Comparison of the Two Bounding Strategies

The solution $x(t, p)$ of IVP (3.8), for $t = 1$, is a concave function of the parameter p, as shown in Fig. 3.4(a) using a solid line. The constant and affine bounds methods proposed can be applied to construct valid convex relaxations on the domain [-5, 5]. The constant bounds for the whole range of parameters are given from the solution of IVPs (3.10) and (3.11) and are shown using the dashed lines. The affine bounds are shown using the dotted lines. They are given from (3.30) and (3.30), where the matrices needed are calculated from the solution of the linear system produced when these equations are applied for $t = 1$ and for $p = p^L$ and $p = p^U$. The values of the state variables are given from the numerical solution at $t = 1$ for $p = p^L$ and $p = p^U$ of IVPs (3.18)–(3.22) coupled, when necessary, with IVPs (3.10) and (3.11). In Fig. 3.4(b), the domain of p is divided into two subdomains and the two strategies are applied again. The affine underestimator is tighter than the constant lower bound and one of the two affine overestimators is tighter than the constant upper bound. Although the second affine overestimator is not that tight for the whole range of p, it reduces the convex space even more. Since the constant bounds are generated at no extra cost when the affine bounds are used, the relaxation strategies used in practice always involve the constant bounds, with or without the affine bounds.



(a) a (b) b

Figure 3.4: Over and underestimators for the solution of IVP (3.8) for $t = 1$. The solution $x(1, p)$ (-),the constant bounds(—) and the affine bounds($\cdots$) are shown.(a)One region: $p \in [-5, 5]$.(b)Two regions: $p \in [-5, 0]$ and $p \in [0, 5]$

3.3.4 Convex Relaxation of the NLP

After underestimating the objective function and overestimating the feasible region, the convex relaxation of the NLP problem (3.1) is given by

$$\begin{aligned} &\min_{\hat{x},p,w} \breve{J}(\hat{x}, p, w) \\ &\text{s.t. } \breve{g}_i(\hat{x}_i, p, w) \leq 0 \qquad i = 0, 1, \ldots, NP \\ &\quad \underline{x}(t_i) \leq \hat{x}_i \leq \bar{x}(t_i) \qquad i = 0, 1, \ldots, NP \\ &\quad \mathcal{C}(\hat{x}, p, w) \leq 0 \\ &\quad p \in [p^L, p^U] \end{aligned} \tag{3.37}$$

where $\breve{J}$ denotes the convex underestimator of the specified function, $\mathcal{C}$ denotes the set of additional constraints arising from the convex relaxation of non–convex terms of special types and w denotes the vector of new variables introduced by this relaxation. If the affine or α–based bounds are additionally used for the convex relaxation of the set of equality constraints then the following constraints can be added to the above formulation:

$$\hat{x}_i + \breve{x}^-(t_i, p) \leq 0 \qquad i = 0, 1, \ldots, NP \tag{3.38}$$

$$\breve{x}(t_i, p) - \hat{x}_i \leq 0 \qquad i = 0, 1, \ldots, NP \tag{3.39}$$

where $\breve{x}(t_i, p)$ and $\breve{x}^-(t_i, p)$ are given either from (3.32) for the case of utilizing affine bounds or from (3.34) and (3.35) for the case of α–based bounds.

Chapter 4

Global optimization algorithm

In this chapter global optimization algorithm is presented, as it was introduced in Papamichail and Adjiman (2004). After constructing the convex relaxation of the original NLP problem, a deterministic spatial BB global optimization algorithm, which follows the one by Horst and Tuy (1990), can be used in order to obtain the global minimum within an optimality margin. A relative optimality margin, ϵ_r, and a maximum number of iterations, $MaxIter$ are user–defined.

Step 1. ***Initialization***

- Set the upper bound on the objective function, $J^u := +\infty$.
- Initialize the iteration counter, $Iter := 0$.
- Initialize a list of subregions $\mathcal{L}$ to an empty list, $\mathcal{L} := \emptyset$.
- Initialize a region $\mathcal{R}$ to the region covering the full domain of variables p, $\mathcal{R} := [p^L, p^U]$.

Step 2. ***Upper bound***

- Solve the original NLP problem with bounds on p given by $\mathcal{R}$.
- If a feasible solution $p_{\mathcal{R}}$ is obtained with objective function $J^u_{\mathcal{R}}$, then set the best feasible solution $p^* := p_{\mathcal{R}}$ and $J^u := J^u_{\mathcal{R}}$.

Step 3. ***Lower bound***

- Obtain bounds on the differential variables.

- If affine bounds (α–based bounds) can be constructed and are additionally used for the convex relaxation of the set of equality constraints, then obtain the necessary matrices (bounds on the second–order sensitivities).
- Form the convex relaxation of the problem for $\mathcal{R}$ and solve it.
- If a feasible solution $p^*_{\mathcal{R}}$ is obtained for $\mathcal{R}$ with objective function $J^l_{\mathcal{R}}$, then add $\mathcal{R}$ to the list $\mathcal{L}$ together with $J^l_{\mathcal{R}}$ and $p^*_{\mathcal{R}}$.

Step 4. ***Subregion selection***

- If the list $\mathcal{L}$ is empty, then the problem is infeasible. Terminate.
- Otherwise set the region $\mathcal{R}$ to the region from the list $\mathcal{L}$ with the lowest lower bound, $\mathcal{R} := \arg\min_{\mathcal{L}_i \in \mathcal{L}} J^l_{\mathcal{L}_i}$.
- Remove $\mathcal{R}$ from the list $\mathcal{L}$.

Step 5. ***Checking for convergence***

- If $(J^u - J^l_R)/|J^l_R| \leq \epsilon_r$, then the solution is p^* with an objective function J^u. Terminate.
- If $Iter = MaxIter$, then terminate and report $(J^u - J^l_R)/|J^l_R|$.
- Otherwise increase the iteration counter by one: $Iter := Iter + 1$.

Step 6. ***Branching within R***

- Apply the least reduced axis rule on region $\mathcal{R}$ to choose a variable on which to branch and generate two new subregions $\mathcal{R}_1$ and $\mathcal{R}_2$ which are a partition of $\mathcal{R}$.

Step 7. ***Upper bound for each region***

- For $i = 1, 2$, solve the original NLP problem with bounds on p given by $\mathcal{R}_i$.
- For $i = 1, 2$, if a feasible solution $p_{\mathcal{R}_i}$ is obtained with objective function $J^u_{\mathcal{R}_i} < J^u$, then update the best feasible solution found so far $p^* := p_{\mathcal{R}_i}$, set $J^u := J^u_{\mathcal{R}_i}$ and remove from the list $\mathcal{L}$ all subregions $\mathcal{R}'$ such that $J^l_{\mathcal{R}'} > J^u$.

Step 8. ***Lower bound for each region***

- Obtain bounds on the differential variables.
- If affine bounds (α–based bounds) can be constructed and are additionally used for the convex relaxation of the set of equality constraints, then obtain the necessary matrices (bounds on the second–order sensitivities).
- Form the convex relaxation of the problem for each subregion $\mathcal{R}_1$ and $\mathcal{R}_2$ and solve it.
- For $i = 1, 2$, if a feasible solution $p^*_{\mathcal{R}_i}$ is obtained for $\mathcal{R}_i$ with objective function $J^l_{\mathcal{R}_i}$, then:
  - If affine or α–based bounds are used and $J^l_{\mathcal{R}_i} < J^l_{\mathcal{R}}$, then set $J^l_{\mathcal{R}_i} := J^l_{\mathcal{R}}$.
  - If $J^l_{\mathcal{R}_i} \leq J^u$, then add $\mathcal{R}_i$ to the list $\mathcal{L}$ together with $J^l_{\mathcal{R}_i}$ and $p^*_{\mathcal{R}_i}$.
- Go to Step 4.

In order to reduce the computational expense arising from the repeated solution of local dynamic optimization problems, the upper bound generation (Step 7) does not have to be applied at every iteration of the algorithm. This does not affect the ability of the algorithm to identify the global solution. In the BB algorithm of Horst and Tuy (1990) if the relaxed problem is feasible for a region, then it has to be at least as tight as the relaxation at its parent node to ensure that the bounding operation is improving. This is true when only constant bounds are used for the relaxation of the set of equality constraints because of their theoretical properties (Papamichail and Adjiman, 2002). A step to enforce this requirement was included in the algorithm for the case in which affine or α–based bounds are used additionally to the constant bounds (Step 8, 5fth bullet point).

Chapter 5

Examples

The global optimization algorithm presented in Chapter 4 was implemented using MATLAB 6.5. To obtain the bounds on variables we used the constant bounds approach (see sections 3.3.1 and 3.3.2). This technique is relatively simple, but suitable for the examples solved in this diploma work. Solution of NLP problems was found using function *fmincon*. It is an implementation of a general NLP solver, provided by the Optimization Toolbox, uses either a subspace trust region method, based on the interior–reflective Newton method, or a sequential quadratic programming method. The MATLAB function *ode45* was used for the integration of IVPs. It is an implementation of a Runge–Kutta method based on the Dormand–Prince pair. The interval calculations needed were performed explicitly using interval arithmetic. Interval arithmetic computations are showed for each example. First example is a simple dynamic optimization problem. The next three examples are parameter estimation problems in chemical kinetics modeling. All the case studies were solved on a Dell workstation (3 GHz Intel Pentium 4 CPU, 1GB RAM).

5.1 Example 1: A Simple Dynamic Optimization Problem

This example is a problem with one optimization parameter p defined in Chapter 3 by equations (3.8). This problem has two local minima. Its formulation is given by

$$\begin{aligned} \min_{p} \ & -x(t=1,p)^2 \\ \text{s.t. } & \dot{x} = -x^2 + p \qquad \forall t \in [0,1] \\ & x(t=0,p) = 9 \\ & -5 \leq v \leq 5 \end{aligned} \tag{5.1}$$

Applying the procedure defined by (3.9) leads to the expressions

$$\begin{aligned} \dot{\underline{x}} &= \inf(-\underline{x}^2 + [p^L, p^U]) \quad \forall t \in [0,1] \\ \underline{x}_0 &= \inf x_0([p^L, p^U]) \end{aligned} \tag{5.2}$$

$$\begin{aligned} \dot{\bar{x}} &= \sup(-\bar{x}^2 + [p^L, p^U]) \quad \forall t \in [0,1] \\ \bar{x}_0 &= \sup x_0([p^L, p^U]) \end{aligned} \tag{5.3}$$

which using the interval arithmetic calculation results in two bounding IVPs

$$\begin{aligned} \dot{\underline{x}} &= -\underline{x}^2 - 5 \quad \forall t \in [0,1] \\ \underline{x}_0 &= 9 \end{aligned} \tag{5.4}$$

$$\begin{aligned} \dot{\bar{x}} &= -\bar{x}^2 + 5 \quad \forall t \in [0,1] \\ \bar{x}(t_0) &= 9 \end{aligned} \tag{5.5}$$

Solutions of these ODEs represent a convex underestimator and concave overestimator of the relaxed problem solution space.

The global optimization algorithm converged with the relative convergence criterion ϵ_r set to 1×10^{-7}. The global optimum parameter found was $p = -5$ and the value of the objective function for the global optimum parameter was equal to -8.2290. However, it is a very simple problem and only 2 iterations were needed for convergence. The solution time was 1 second of CPU time.

5.2 Example 2: A First–order Irreversible Liquid–phase Reaction

The second example is a parameter estimation problem with two parameters and two differential equations as the constraints. It was published in Esposito and Floudas (2000b) as well as in Papamichail and Adjiman (2002). It involves a first–order irreversible isothermal liquid–phase chain reaction.

$$\mathrm{A} \overset{k_1}{\rightarrow} \mathrm{B} \overset{k_2}{\rightarrow} \mathrm{C}$$

The problem can be formulated as follows:

$$\begin{aligned}
\min_{k_1,k_2} \; & \sum_{j=1}^{10}\sum_{i=1}^{2}(x_i(t=t_j,k_1,k_2)-x_i^{exp}(t_j))^2 \\
\text{s.t. } & \dot{x}_1 = -k_1 x_1 \qquad \forall t\in[0,1] \\
& \dot{x}_2 = k_1 x_1 - k_2\, x_2 \qquad \forall t\in[0,1] \\
& x_1(t=0,k_1,k_2)=1 \\
& x_2(t=0,k_1,k_2)=0 \\
& 0\le k_1\le 10 \\
& 0\le k_2\le 10
\end{aligned} \tag{5.6}$$

where x_1 and x_2 are the mole fractions of components A and B, respectively. k_1 and k_2 are the rate constants of the first and second reaction, respectively. $x_i^{exp}(t_j)$ is the experimental point for the state variable i at time t_j. The points used are taken from Esposito and Floudas (2000b).

Applying the procedure defined by (3.9) led to the expressions

$$\begin{aligned}
&\dot{\underline{x}}_1 = \inf(-[k_1^L,k_1^U]\times\underline{x}_1) \quad \forall t\in[0,1] \\
&\underline{x}_{10} = \inf x_{10}([p^L,p^U]) \\
&\dot{\underline{x}}_2 = \inf([k_1^L,k_1^U]\times[\underline{x}_1,\bar{x}_1]-[k_2^L,k_2^U]\times\underline{x}_2) \quad \forall t\in[0,1] \\
&\underline{x}_{20} = \inf x_{20}([p^L,p^U])
\end{aligned} \tag{5.7}$$

$$\begin{aligned}
&\dot{\bar{x}}_1 = \sup(-[k_1^L,k_1^U]\times\bar{x}_1) \quad \forall t\in[0,1] \\
&\bar{x}_{10} = \sup x_{10}([p^L,p^U]) \\
&\dot{\bar{x}}_2 = \sup([k_1^L,k_1^U]\times[\underline{x}_1,\bar{x}_1]-[k_2^L,k_2^U]\times\bar{x}_2) \quad \forall t\in[0,1] \\
&\bar{x}_{20} = \sup x_{20}([p^L,p^U])
\end{aligned} \tag{5.8}$$

which using the interval arithmetic calculation results in four bounding IVPs

$$\begin{aligned}
&\dot{\underline{x}}_1 = -k_1^L\underline{x}_1 \quad \forall t\in[0,1] \\
&\underline{x}_{10} = 1 \\
&\dot{\underline{x}}_2 = k_1^L\underline{x}_1 - k_2^U\underline{x}_2 \quad \forall t\in[0,1] \\
&\underline{x}_{20} = 0
\end{aligned} \tag{5.9}$$

$$\begin{aligned}
&\dot{\bar{x}}_1 = -k_1^U\bar{x}_1 \quad \forall t\in[0,1] \\
&\bar{x}_{10} = 1 \\
&\dot{\bar{x}}_2 = k_1^U\bar{x}_1 - k_2^L\bar{x}_2 \quad \forall t\in[0,1] \\
&\bar{x}_{20} = 0
\end{aligned} \tag{5.10}$$

Solutions of these ODEs represent a convex underestimator and concave overestimator of the relaxed problem solution space.

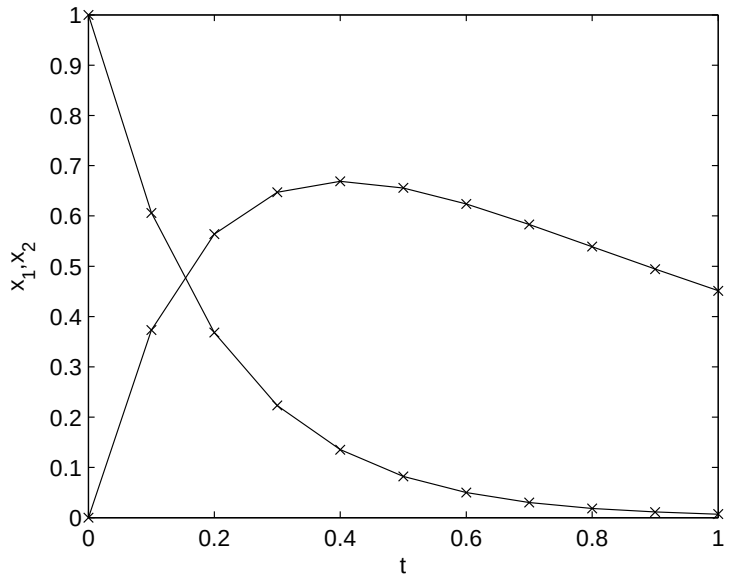


Figure 5.1: Experimental points and state variable trajectories for the globally optimal parameters in Example 2

The global optimization algorithm converged with the relative convergence criterion ϵ_r set to 1×10^{-2}. The global optimum parameter found was $k_1 = 5.0035$ and $k_2 = 1.0000$ and the value of the objective function for the global optimum parameter was equal to 1.1856×10^{-6}. 3436 iterations were necessary for convergence of the algorithm in 2632 seconds. The experimental points and trajectories of state variables for global optimum are showed in Fig. 5.1. The upper bound calculation was performed once every 100 iterations.

5.3 Example 3: Catalytic Cracking of Gas Oil

This example is a parameter estimation problem with three parameters and two differential equations in the constraints. It appears in Esposito and Floudas (2000b) and Papamichail and Adjiman (2004). It involves an overall reaction of catalytic cracking of gas oil (A) to gasoline (Q) and other products (S):

$$\begin{aligned} \mathrm{A} &\xrightarrow{k_1} \mathrm{Q} \\ \mathrm{Q} &\xrightarrow{k_2} \mathrm{S} \\ \mathrm{A} &\xrightarrow{k_3} \mathrm{S} \end{aligned}$$

The problem can be formulated as follows:

$$\begin{aligned}
\min_{k_1,k_2,k_3} \ & \sum_{j=1}^{20}\sum_{i=1}^{2}(x_i(t=t_j,k_1,k_2,k_3)-x_i^{exp}(t_j))^2 \\
\text{s.t. } & \dot{x}_1 = -(k_1+k_3)x_1^2 \qquad \forall t\in[0,0.95] \\
& \dot{x}_2 = k_1x_1^2 - k_2x_2 \qquad \forall t\in[0,0.95] \\
& x_1(t=0,k_1,k_2,k_3)=1 \\
& x_2(t=0,k_1,k_2,k_3)=0 \\
& 0\le k_1\le 20 \\
& 0\le k_2\le 20 \\
& 0\le k_3\le 20
\end{aligned} \tag{5.11}$$

where x_1 and x_2 are the mole fractions of components A and Q, respectively. k_1, k_2 and k_3 are the rate constants of the respective reactions. $x_i(t_j)$ is the experimental point for the state variable i at time t_j. The points used are again taken from Esposito and Floudas (2000b).

Applying the procedure defined by (3.9) results in expressions

$$\begin{aligned}
&\dot{\underline{x}}_1 = \inf(-[k_1^L+k_3^L,k_1^U+k_3^U]\times\underline{x}_1^2) \quad \forall t\in[0,1] \\
&\underline{x}_{10} = \inf x_{10}([p^L,p^U]) \\
&\dot{\underline{x}}_2 = \inf([k_1^L,k_1^U]\times[\underline{x}_1,\bar{x}_1]\times[\underline{x}_1,\bar{x}_1]-[k_2^L,k_2^U]\times\underline{x}_2) \quad \forall t\in[0,1] \\
&\underline{x}_{20} = \inf x_{20}([p^L,p^U])
\end{aligned} \tag{5.12}$$

$$\begin{aligned}
&\dot{\bar{x}}_1 = \sup(-[k_1^L+k_3^L,k_1^U+k_3^U]\times\bar{x}_1^2) \quad \forall t\in[0,1] \\
&\bar{x}_{10} = \sup x_{10}([p^L,p^U]) \\
&\dot{\bar{x}}_2 = \sup([k_1^L,k_1^U]\times[\underline{x}_1,\bar{x}_1]\times[\underline{x}_1,\bar{x}_1]-[k_2^L,k_2^U]\times\bar{x}_2) \quad \forall t\in[0,1] \\
&\bar{x}_{20} = \sup x_{20}([p^L,p^U])
\end{aligned} \tag{5.13}$$

which using the interval arithmetic calculation results in following bounding IVPs

$$\begin{aligned}
&\dot{\underline{x}}_1 = -(k_1^U+k_3^U)\underline{x}_1^2 \quad \forall t\in[0,1] \\
&\underline{x}_{10} = 1 \\
&\dot{\underline{x}}_2 = k_1^L\underline{x}_1^2 - k_2^U\underline{x}_2 \quad \forall t\in[0,1] \\
&\underline{x}_{20} = 0
\end{aligned} \tag{5.14}$$

$$\begin{aligned}
&\dot{\bar{x}}_1 = -(k_1^L+k_3^L)\bar{x}_1^2 \quad \forall t\in[0,1] \\
&\bar{x}_{10} = 1 \\
&\dot{\bar{x}}_2 = k_1^U\bar{x}_1^2 - k_2^L\underline{x}_2 \quad \forall t\in[0,1] \\
&\bar{x}_{20} = 0
\end{aligned} \tag{5.15}$$

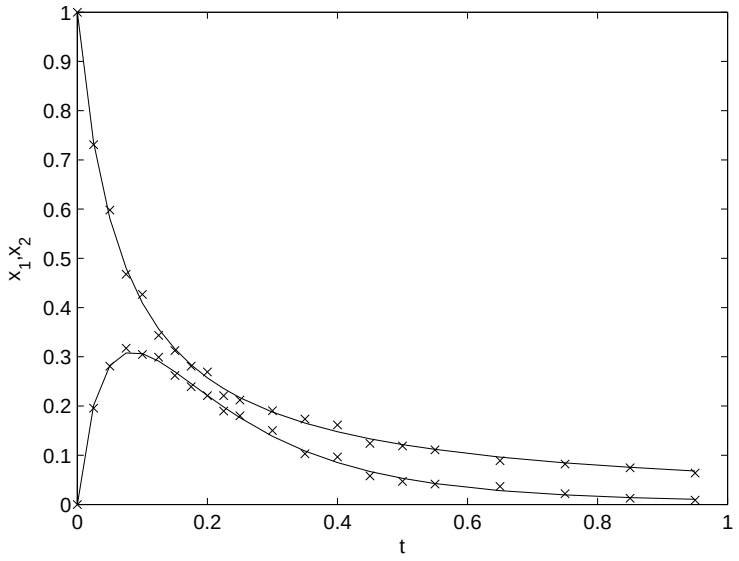


Figure 5.2: Experimental points and state variable trajectories for the globally optimal parameters in Example 3

Solutions of these ODEs represent a convex underestimator and concave overestimator of the relaxed problem solution space.

The globally optimal parameters are $k_1 = 12.2111$, $k_2 = 7.9764$, and $k_3 = 2.2259$ with the corresponding value of the objective function equal to 2.655×10^{-3}. The experimental points and the state variable trajectories for the global optimum are shown in Fig. 5.2. Algorithm converged after 8497 iterations and 13637 seconds of CPU time. The upper bound calculation was performed once every 100 iterations.

5.4 Example 4: A First–order Reversible Liquid–phase Reaction

The fourth example is a parameter estimation problem with four parameters and three differential equations. It appears in Esposito and Floudas (2000b). It involves a first–order reversible isothermal liquid–phase chain reaction.

$$\mathrm{A} \underset{k_2}{\overset{k_1}{\rightleftarrows}} \mathrm{B} \underset{k_4}{\overset{k_3}{\rightleftarrows}} \mathrm{C}$$

The problem can be formulated as follows:

$$
\begin{aligned}
\min_{k_1,k_2,k_3,k_4} \; & \sum_{j=1}^{20}\sum_{i=1}^{3}(x_i(t=t_j,k_1,k_2,k_3,k_4)-x_i^{exp}(t_j))^2 \\
\text{s.t. } & \dot{x}_1 = -k_1x_1 + k_2x_2 \qquad \forall t\in[0,1] \\
& \dot{x}_2 = k_1x_1 - (k_2+k_3)x_2 + k_4x_3 \qquad \forall t\in[0,1] \\
& \dot{x}_3 = k_3x_2 - k_4x_3 \qquad \forall t\in[0,1] \\
& x_1(t=0,k_1,k_2,k_3,k_4) = 1 \\
& x_2(t=0,k_1,k_2,k_3,k_4) = 0 \\
& x_3(t=0,k_1,k_2,k_3,k_4) = 0 \\
& 0\le k_1\le 10 \\
& 0\le k_2\le 10 \\
& 0\le k_3\le 10 \\
& 0\le k_4\le 10
\end{aligned}
\tag{5.16}
$$

where x_1, x_2 and x_3 are the mole fractions of components A, B and C, respectively. k_1, k_2 and k_3 are the rate constants of the first and second reaction, respectively. $x_i(t_j)$ is the experimental point for the state variable i at time t_j. The points used are taken from Esposito and Floudas (2000b).

Applying the procedure defined by (3.9) leads to the expressions

$$
\begin{aligned}
&\dot{\underline{x}}_1 = \inf(-[k_1^L,k_1^U]\times\underline{x}_1 + [k_2^L,k_2^U]\times[\underline{x}_2,\bar{x}_2]) \quad \forall t\in[0,1] \\
&\underline{x}_{10} = \inf x_{10}([p^L,p^U]) \\
&\dot{\underline{x}}_2 = \inf([k_1^L,k_1^U]\times[\underline{x}_1,\bar{x}_1] - [k_2^L+k_3^L,k_2^U+k_3^U]\times\underline{x}_2 + [k_4^L,k_4^U]\times[\underline{x}_3,\bar{x}_3]) \quad \forall t\in[0,1] \\
&\underline{x}_{20} = \inf x_{20}([p^L,p^U]) \\
&\dot{\underline{x}}_3 = \inf([k_3^L,k_3^U]\times[\underline{x}_2,\bar{x}_2] - [k_4^L,k_4^U]\times\underline{x}_3) \quad \forall t\in[0,1] \\
&\underline{x}_{30} = \inf x_{30}([p^L,p^U])
\end{aligned}
\tag{5.17}
$$

$$
\begin{aligned}
&\dot{\bar{x}}_1 = \sup(-[k_1^L,k_1^U]\times\bar{x}_1 + [k_2^L,k_2^U]\times[\underline{x}_2,\bar{x}_2]) \quad \forall t\in[0,1] \\
&\bar{x}_{10} = \sup x_{10}([p^L,p^U]) \\
&\dot{\bar{x}}_2 = \sup([k_1^L,k_1^U]\times[\underline{x}_1,\bar{x}_1] - [k_2^L+k_3^L,k_2^U+k_3^U]\times\bar{x}_2 + [k_4^L,k_4^U]\times[\underline{x}_3,\bar{x}_3]) \quad \forall t\in[0,1] \\
&\bar{x}_{20} = \sup x_{20}([p^L,p^U]) \\
&\dot{\bar{x}}_3 = \sup([k_3^L,k_3^U]\times[\underline{x}_2,\bar{x}_2] - [k_4^L,k_4^U]\times\bar{x}_3) \quad \forall t\in[0,1] \\
&\bar{x}_{30} = \sup x_{30}([p^L,p^U])
\end{aligned}
\tag{5.18}
$$

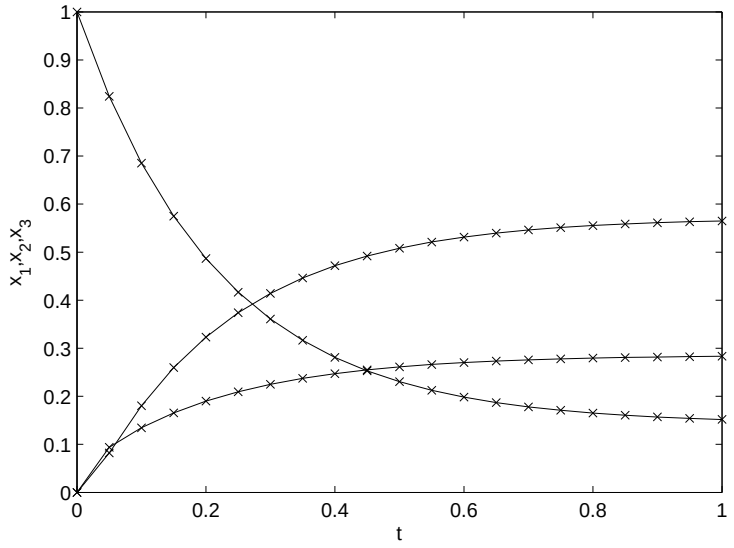


Figure 5.3: Experimental points and state variable trajectories for the globally optimal parameters in Example 4

which using the interval arithmetic calculation results in six bounding IVPs

$$\begin{aligned}
&\dot{\underline{x}}_1 = -k_1^U \underline{x}_1 + k_2^U \underline{x}_2 \quad \forall t \in [0,1] \\
&\underline{x}_{10} = 1 \\
&\dot{\underline{x}}_2 = k_1^L \underline{x}_1 - (k_2^U + k_3^U)\underline{x}_2 + k_4^L \underline{x}_3 \quad \forall t \in [0,1] \\
&\underline{x}_{20} = 0 \\
&\dot{\underline{x}}_3 = k_3^L \underline{x}_2 - k_4^U \underline{x}_3 \quad \forall t \in [0,1] \\
&\underline{x}_{30} = 0
\end{aligned} \tag{5.19}$$

$$\begin{aligned}
&\dot{\bar{x}}_1 = -k_1^L \bar{x}_1 + k_2^U \bar{x}_2 \quad \forall t \in [0,1] \\
&\bar{x}_{10} = 1 \\
&\dot{\bar{x}}_2 = k_1^U \underline{x}_1 - (k_2^L + k_3^L)\bar{x}_2 + k_4^U \bar{x}_3 \quad \forall t \in [0,1] \\
&\bar{x}_{20} = 0 \\
&\dot{\bar{x}}_3 = k_3^U \bar{x}_2 - k_4^L \bar{x}_3 \quad \forall t \in [0,1] \\
&\bar{x}_{30} = 0
\end{aligned} \tag{5.20}$$

Solutions of these ODEs represent a convex underestimator and concave overestimator of the relaxed problem solution space.

The experimental points and trajectories of state variables for global optimum are shown in Fig. 5.3.

Algorithm converged within the relative convergence criterion ϵ_r set to 1×10^{-2} global optimization. Global optimum found was with parameter values $k_1 = 3.9990$, $k_2 = 1.9981$, $k_3 = 40.0000$ $k_4 = 20.0007$, and the value of the objective function for the global optimum parameter was equal to 1.1856×10^{-6}. Algorithm converged after 44600 iterations and

142380 seconds of CPU time. The upper bound calculation was performed once every 100 iterations.

Chapter 6

Conclusion

As it was mentioned before, modeling of processes (dynamic systems) leads to a dynamic model consisting of set of differential equations. Considered processes are described by first–order parameter dependent, typically nonlinear, differential equations. Local optimum solutions of these systems are found by means of dynamic optimization using gradient–based methods, while dynamic optimization problem is formulated as a non–convex NLP problem.

Main purpose of this work was to present a global optimization algorithm suitable for solving GOP problems. A deterministic sBB global optimization algorithm was introduced, discussed and implemented for the dynamic optimization problem. Local solutions, produced using the sequential approach, were used as an upper bound on the global minimum of the objective function value. Lower bounds were provided from the solution of a convex relaxation of the problem on subregions considered in the BB algorithm. This convex relaxation was achieved after defining a convex underestimation of the objective function and a convex overestimation of the feasible region. Algebraic functions were underestimated using well–known techniques.

Three procedures were proposed for the convex relaxation of the dynamic information, namely constant, affine a α–based bounds. Constant and affine bounds are operating with theory of bounding solutions of IVPs. It was shown that affine bounds technique is useable only in case that righthand side of model differential equations can be decomposed into sum of non-fractional terms. Alpha–based bounds approach is based on the overestimation of the space of solutions of the parameter dependent ordinary differential equations(ODE) system that results when the original system is coupled with the first and the second–order sensitivity equations.

We implemented the proposed algorithm and used it to solve selected examples relevant to chemical engineering. The principle of constant bounds is very useful and also quite simple. Therefore we focused on this approach in the work. Explicit interval arithmetic

calculations are used and solution of IVP (3.9) is performed once at each node of BB tree. Results show that the method of constant bounds for larger problems results in enormous rise of the number of iterations, and of course CPU time, needed to obtain the global optimum. When affine estimators are used additionally to the constant bounds there is an increase in the size of the integrated ODE systems. However, the comparison of bounding strategies suggests that the algorithm can converge in far fewer iterations than in the case of using the constant bounds alone. That is the reason why further work should be dedicated to focus on larger systems with utilization of combinations of the proposed bounding strategies.

Bibliography

C. S. Adjiman, I. P. Androulakis, C. D. Maranas, and C. A. Floudas. A global optimization method, αBB for process design. *Computers and Chemical Engineering*, 20:419–424, 1996.

C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. A global optimization method, αBB, for general twice-differentiable constrained NLPs - II. Implementation and computational results. *Computers and Chemical Engineering*, 22(9):1159–1179, 1998a.

C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs - I. Theoretical advances. *Computers and Chemical Engineering*, 22(9):1137–1158, 1998b.

M. Bagajewicz and V. Manousiouthakis. On the generalized benders decomposition. *Computers and Chemical Engineering*, 15:691–700, 1991.

R. G. Brusch and R. H. Schappelle. Solution of highly constrained optimal control problems using nonlinear programming. *AIAA Journal*, 11(2):135–136, 1973.

W. Esposito and C. Floudas. Deterministic global optimization in nonlinear optimal control problems. *Journal of Global Optimization*, 17:97–126, 2000a.

W. Esposito and C. Floudas. Global optimization for the parameter estimation of differential-algebraic systems. *Industrial and Engineering Chemistry Research*, 39:1291–1310, 2000b.

W. Esposito and C. Floudas. Global optimization of nonconvex problems with differential-algebraic constraints. *European Symposium on Computer Aided Process Engineering.*, 10: 73–78, 2000c.

C.A. Floudas and V. Visweswaran. A global optimization algorithm (gop) for certain classes of non-convex nlps–i theory. *Computers and Chemical Engineering*, 14:1397–1417, 1990.

A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.

J. R. Han, V. Manousiouthakis, and S. H. Choi. Global optimization of chemical processes using the interval analysis. *Korean Journal of Chemical Engineering*, 14:270–276, 1997.

R. Horst and H. Tuy. *Global optimization. Deterministic approaches (3rd ed.).* Springer Verlag, Berlin, Germany, 1990.

K. L. Nickel. Using interval methods for numerical solution of ODE's. *ZAMM*, 66(11): 513–523, 1986.

I. Papamichail and C. S. Adjiman. A rigorous global optimization algorithm for problems with ordinary differential equations. *Journal of Global Optimization*, 24:1–33, 2002.

I. Papamichail and C. S. Adjiman. Global optimization of dynamic systems. *Computers and Chemical Engineering*, 28:403–415, 2004.

H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization.* Ellis Horwood, Chichester, England, 1988.

H. S. Ryoo and N. V. Sahinidis. Global optimization of nonconvex nlps and minlps with applications in process design. *Computers and Chemical Engineering*, 19:551–566, 1995.

R. M. Soland. An algorithm for separable nonconvex programming problems II: Nonconvex constraints. *Management Science*, 17:759–773, 1971.

T. H. Tsang, D. M. Himmelblau, and T. F. Edgar. Optimal control via collocation and non-linear programming. *Int. J. Control*, 21(5):763–768, 1975.

R. Vaidyanathan and M. El-Halwagi. Global optimization of nonconvex nonlinear programs via interval analysis. *Computers and Chemical Engineering*, 18:889–897, 1994.

V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. 1. problems without path constraints. *Industrial and Engineering Chemistry Research*, 33(9):2111–2122, 1994a.

V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. 2. problems with path constraints. *Industrial and Engineering Chemistry Research*, 33(9):2123–2133, 1994b.

L. A. Zadeh and C. A. Desoer. *Linear system theory: The state space approach. Series in system science.* McGraw-Hill, New York, USA, 1963.