Optimal Control for Nonlinear Systems

by

Peter Taraba

(Autoreferat for the third study degree
PhD in the branch 5.2.14 Automation and control)
Faculty of Electrical Engineering and Information technology
Slovak University of Technology in Bratislava
February 2013
Redmond, WA, USA

Dizertačná práca bola vypracovaná v externej forme doktorandského štúdia na Ústave riadenia a priemyselnej informatiky Fakutly elektortechiniky a informatiky Slovenskej technickej univerizty v Bratislave.

Predkladateľ	Ing. Peter Taraba 15803 Bear Creek Parkway E229 Redmond, WA, 98052, USA
Školiteľ	Doc. Ivan Sekaj, PhD. Ústav riadenia a priemyselnej informatiky Fakulta elektortechniky a informatiky Slovenská technická univerzita Bratislava Ilkovičova 3 812 19 Bratislava
Oponenti	....

Autoreferát bol rozoslaný dňa: .....

Obhajoba dizertačnej práce sa koná: ..... pred komisiou pre obhajobu dizertačnej práce v odbore doktorandského štúdia, vymenovanou predsedom spoločnej odborovej komisie dňa .....

číslo odboru: 5.12.14, odbor doktorandského štúdia: Automatizácia a riadenie

na Ústave riadenia a priemyselnej informatiky, Fakulty elektortechinky a informatiky, Slovenskej technickej univerzity, Ilkovičova 3, 812 19 Bratislava

Predseda spoločnej odborovej komisie Automatizácia na STU v Bratislave: Prof. Ing. Miroslav Fikar, DrSc.

Garant študijného programu Automatizácia a riadenie na FEI STU Prof. Ing. Peter Hubinský, PhD. Ústav riadenia a priemyselnej informatiky, STU FEI v Bratislave Ilkovičova 3, 812 19 Bratislava

TABLE OF CONTENTS

CHAPTER I

Introduction

Control theory is an interdisciplinary branch of engineering and mathematics. Due to the advancement of software and hardware (number of computations per second) capabilities, software engineering is becoming an important part of control theory by enabling computations which were not possible in the past. Engineering dates itself as far back as 1325. One of the most important concepts in control theory came in the 1890s when Alexander Lyapunov introduced the concept of stability theory. His concepts can be used even on nonlinear systems for finding control solutions which stabilize and control nonlinear systems to desired values, and his methods are used even today. The obstacle is to find the Lyapunov function for every single new problem one needs to solve. A few solutions using his concepts are described in [2], [3], [4] among many others. There are also a number of books describing the Lyapunov approach, for example [5]. Due to the work of Lev Pontryangin [6] and Richard Bellman, optimal control theory was popularized in the 1960s. The aim of this PhD thesis is to enable engineers to find optimal control solutions for nonlinear systems in a less time-consuming and more automatic manner than with previous approaches.

Finding an optimal control for a broad range of problems is not a simple task. Conventional control methods are based on model constructions. However, it may be difficult to construct a sufficiently accurate model or employ too many assumptions to solve a differential equation. As an example, one can mention predictive control using AR-Volterra models [9], which can be used to describe nonlinear problems, but the degree of Volterra models is increasing in order to sufficiently describe nonlinearity and dynamics and hence more difficult to use it afterwards for model predictive control (more computationly intense). There are currently many other methods which try to tackle this problem using a range of solutions (more 'brute force'), for example fuzzy inference control [1]. There is also [46] where authors of the paper partition a set of state space into simplicital cones and provide a piecewise affine control law which ensures feasibility and stability, but is also optimal with respect to LQR problems. Computational complexity of the algorithm presented in the thesis grows exponentially with the dimension, as it is also for piecewise linear quadratic optimal control [39]. However, as shown on examples, even with not completely refined meshes, good solutions can be found and we provide comparison of solutions for different mesh sizes.

Besides showing two dimensional example, we show that it is possible to find solution even for higher dimensional problems. The error estimation is very important for a lot of different numerical algorithms, as an example Finite Element Methods [25] can be mentioned. We introduce error estimation for the algorithm and for shown examples, we show how error estimation is decreasing with increasing number of mesh points.

In 2002, M. Dellnitz and O. Junge introduced Set Oriented Numerical methods for Dynamical Systems [14], which enabled studying of complicated temporal behavior of dynamical systems. These dynamical systems are described by ordinary differential equations, and hence share important similarities with control theory. Besides this paper, other papers on this concept [35] were introduced which solve optimal control problems by using software engineering.

In this autoreferat, we introduce Optimal Mesh Control in chapter II. In chapter III we summarize all the contributions of PhD thesis. Finally, in chapter IV, we state all the publications, references and patents author acquired during his PhD studies.

CHAPTER II

Optimal Control Mesh

Finding an optimal control for a broad range of problems is not a simple task. There are currently many methods which try to tackle this problem using a range of solutions. The closest ones to the algorithm suggested in this chapter are a set-oriented approach described in [48] and in [49], and a subdivision algorithm for optimal control [50]. In addition to these, there is [46] where authors of the paper partition a set of state space into simplicital cones and provide a piecewise affine control law which ensures feasibility and stability, but is also optimal with respect to LQR problems. For more references on this topic, see [50].

The main advantage of the subdivision algorithm for optimal control over a set-oriented approach is the ability to estimate when to stop increasing the mesh size and smaller foot-print of the final solution, because with the subdivision algorithm, solutions were found even for coarse divisions of state space. Moreover, a set-oriented approach has a need for sufficient partitioning (adaptive structure), which does not necessarily improve the quality of the final solution. The last advantage of the subdivision algorithm over a set-oriented approach is no need to convert from continuous to discrete model. With optimal control mesh, we keep the advantages subdivision algorithm had, and the computations needed for finding a solution are significantly faster than with subdivision algorithm. Even for three dimensional problems, the computations are faster than computations of two dimensional problems with the subdivision algorithm. In addition, this new algorithm introduces error estimation which can be used as an indicator when to stop increasing the mesh size.

Chapter 2.1 of this chapter describes the control problem we seek to address. Chapter 2.2 introduces the algorithm for finding optimal control mesh. Chapter 2.3 describes a problem that the algorithm can encounter on the border and a solution how to solve this problem. Chapter 2.4 shows extensions of the algorithm for robust applications and problems where only the subset of state space vector x is controlled. Chapters 2.5 and 2.6 show results for two dimensional problems (inverted pendulum and DC to DC converter [40]) and chapter 2.7 shows results for a three dimensional problem (CRS system [51]). Finally, conclusions are made in chapter 2.8.

2.1 Problem Formulation

Consider the problem of optimal stabilization of the continuous-time control system:

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

where $f : X \times U \to \mathbb{R}^N$ is continuous, one time differentiable, and it is assumed to be locally asymptotically controllable to the desired value $\check{x} \in X$, $x \in X \subset \mathbb{R}^N$ is the state of the system, X is a region of interest, $u \in U \subset \mathbb{R}^M$ is the control input, U is the compact region of admissible controls.

The goal is to construct an approximate optimal feedback $\hat{u} : x \to U$, such that time to converge to desired value $\check{x}$ will be minimal for any given point $x \in X$. This is similar to the energy function in [48]. The algorithm described in this chapter creates a mesh evenly distributed on the region of interest X and tries to assign to every point on the mesh its energy, which is the time needed to converge to the desired value, its optimal control value and error estimation for the energy. The algorithm starts with assigning $+\infty$ as energy for every point on the mesh, then assigns small energy values to the points closest to the desired value based on approximately how much time is needed to get to the desired value, and finally, the algorithm tries to spread the points which have finite energy further.

2.2 Algorithm Description

The algorithm suggested in this chapter is based on creating a mesh over region of interest X. As the mesh is getting smaller, function f can be better approximated by linear dependency locally on the mesh because of Taylor's theorem. Hence we are considering only functions f which are one time differentiable and continuous on the whole region of interest. We also assume the one time differentiability and continuity about the energy function on the region of interest, so that we can make an estimation of energies in the step 7 of the algorithm.

The main principle used in this algorithm is spreading through its neighbors, which are already able to converge to the desired value. Initially, there is a set S containing points around the desired value, then set $C \subset S$, which consists of points which for a certain control value u are directed closer to the desired value $\check{x}$. All the neighbor points of C will be included in set I, which is an active set of points which might have the ability to improve their energy function as one of their neighbors has improved its energy value. Besides that we also have set F which keeps track of the best energy value E and the best control value u for each point.

Steps of the algorithm:

1. This step of the algorithm creates a mesh such that distances between points are equal. If the region of interest is $X = [x_1^{min}, x_1^{max}] \times [x_2^{min}, x_2^{max}] \times \ldots \times$

$[x_N^{min}, x_N^{max}]$, then point on the mesh is defined as

$$p(i_1, i_2, \ldots, i_n) = \begin{pmatrix} x_1(i_1) \\ x_2(i_2) \\ \vdots \\ x_N(i_N) \end{pmatrix} = \begin{pmatrix} x_1^{min} + \frac{i_1-1}{M_1-1}(x_1^{max} - x_1^{min}) \\ x_2^{min} + \frac{i_2-1}{M_2-1}(x_2^{max} - x_2^{min}) \\ \vdots \\ x_N^{min} + \frac{i_N-1}{M_N-1}(x_N^{max} - x_N^{min}) \end{pmatrix},$$

where $M_1, M_2, \ldots, M_N$ are numbers of points on the mesh for different coordinates and $i_j \in \{1, \ldots, M_j\}$.

2. In this step, a finite set U of possible control values is created. The set U will be used in several steps of the algorithm.

3. All the points created in step 1 will be added to the set F which, in addition to the position of the point, also holds other information - optimal control value (which is not set initially) and the best energy value of the point, which is initially set as positive infinity as the worst case scenario (point is not able to convert to the desired value).

$$F = \{(p(i_1, \ldots, i_N), u(i_1, \ldots, i_N), E(i_1, \ldots, i_N), \varepsilon(i_1, \ldots, i_N)) : i_j \in \{1, \ldots, M_j\} \\ \forall j \in \{1, \ldots, N\}\},$$

where initially $E(i_1, \ldots, i_N) = +\infty$ and $u(i_1, \ldots, i_N)$ is not set. $\varepsilon(i_1, \ldots, i_N)$ is the error estimation of the energy value $E(i_1, \ldots, i_N)$, set initially also to $+\infty$. The way the error is estimated is described in section 5.

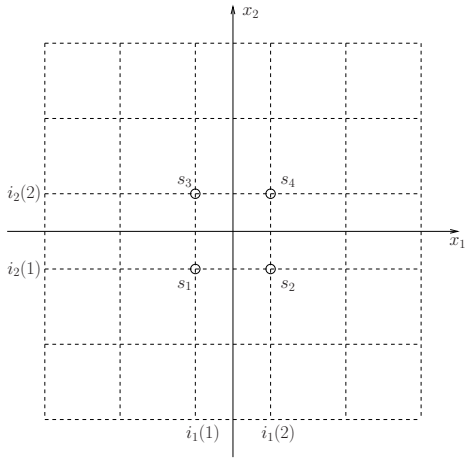
4. This step of the algorithm finds set S, which consists of the points closest to the desired value $\check{x}_j$. First, the algorithm finds L indexes i_j which are closest to the desired value for each coordinate

$$Z_j = \{i_j(1), \ldots, i_j(L) : |x_j(i_j(1)) - \check{x}_j| \leq \ldots \leq |x_j(i_j(L)) - \check{x}_j| \leq |x_j(k) - \check{x}_j| \\ \forall k \in \{1, \ldots, M_j\} \,/\, \{i_j(1), \ldots, i_j(L)\}\}$$

Then algorithm creates set S consisting of points closest to desired value through all the combinations Z_j.

$$S = \{p(k_1, k_2, \ldots, k_N) \ : \ k_j \in Z_j \ \forall j \in \{1, \ldots, N\}\}.$$

See fig. 2.1 for details (L is chosen 2 in the figure). For two dimensional problems, S will contain L^2 points, for N dimensional problems, the set S will contain L^N points.



Obr. 2.1: Mesh over the region of interest with initial points $S = \{s_1, s_2, s_3, s_4\}$, and indexes closest to the desired value for each coordinate $Z_1 = \{i_1(1), i_1(2)\}$ and $Z_2 = \{i_2(1), i_2(2)\}$.

5. This step of the algorithm tests if $\exists u \in U$ for point $p \in S$ such that the point can get closer to the desired value than it originally was. In case it is feasible, we will add this point to controllable set C of points which converge to the desired value $\check{x}$ and update point's energy and control value in set F.

   If p is the tested point and $f(p, u)$ is its direction for a control value $u \in U$ then the point can get closer to the desired value in case $\exists t > 0$ for cost function $J(p, u, t) = (p + t\ f(p, u) - \check{x})^T (p + t\ f(p, u) - \check{x})$ such that $J(t) < J(0)$. Optimal time $\hat{t}$ can be computed as

   $$\hat{t}(p, u) = \frac{f^T(p, u)(p - \check{x})}{f^T(p, u) f(p, u)}.$$

   The energy value for this point can be estimated as

   $$E(p, u) \approx \hat{t}(p, u)$$

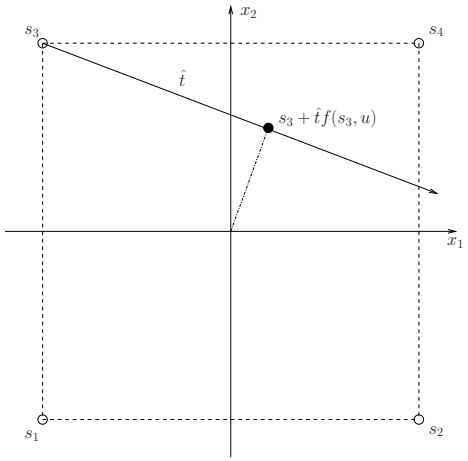
   Point p will be associated with optimal control value $\hat{u}(p) = \arg\min_{u \in U} E(p, u)$ and energy value $E(p, \hat{u}(p))$ in case $\hat{t}(p, \hat{u}(p)) > 0$.

   $$C = \left\{ (p, \hat{u}(p), E(p, \hat{u}(p))) : p \in S \ \wedge \ \hat{t}(p, \hat{u}(p)) > 0 \right\}$$

   See fig. 2.2, which displays this process for a two dimensional problem.

   For all the points in C we update their energy values and optimal control values also in set F.

6. In this step, we find the initial set I consisting of points which might have the ability to improve their energy value as points around them decreased their



Obr. 2.2: Testing if point s_3 can point closer to the desired value with new estimation of energy with value $E(p,u) \approx \hat{t}(p,u)$.

energy values. This will include all the points surrounding points in C. Let's define surrounding set of a point as

$$Sur(p(i_1, i_2, \ldots, i_N)) = \{p(j_1, j_2, \ldots, j_N) : |j_k - i_k| \leq 1 \text{ for } \forall k \in \{1, ..., M_k\}\} \\ / \{p(i_1, i_2, \ldots, i_N)\}.$$

In set I in addition to remembering which points have potential to improve their energy value, we also remember the energy value of the neighbor, which was changed. The reasoning behind this is to start spreading through points with lower energy values first. This has a huge impact on the performance of the algorithm in comparison with random order of points we adapt.

$$I = \{(E(i_1, \ldots, i_N), p(j_1, \ldots, j_n)) : p(i_1, \ldots, i_N) \in C \wedge \\ p(j_1, \ldots, j_n) \in Sur(p(i_1, \ldots, i_N))\}$$

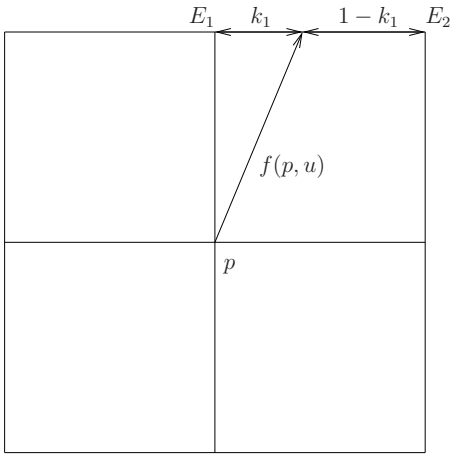
Duplicates of a point p in set I are not allowed and the algorithm remembers only the lowest energy value due to which point p can be improved.

7. Point p is selected from the set I with the lowest energy associated with it and this point is also removed from this set. For all the values $u \in U$, compute the time needed to get to the point between surrounding points of point p.

$$\tilde{t}(p,u) = \min_{i \in \{1,\ldots,N\}} \left| \frac{(x_i^{max} - x_i^{min})/(M_i - 1)}{f_i(p,u)} \right|$$

See fig. 2.3 for details. The new estimate of energy value for point p and control value u then can be written as

$$\tilde{E}(p,u) = \tilde{t}(p,u) + k_1 E_1 + (1 - k_1) E_2,$$



Obr. 2.3: Testing u for point p. Direction $f(p,u)$ points to a point in between of other two points on the mesh $(p+\tilde{t}(p,u)f(p,u))$, whose energy values can be used to approximate new energy value of p.

where E_1 and E_2 are energies associated with points in set F and k_1 and $1-k_1$ are relative distances towards these points from $(p+\tilde{t}(p,u)f(p,u))$. Value k_1 in different coordinates is computed following way:

$$d_i = \frac{p_i + \tilde{t}(p,u)f_i(p,u) - x_i^{min}}{x_i^{max} - x_i^{min}}(M_i - 1)$$

$$k_i = \lceil d_i \rceil - d_i$$

The new optimal control value for point p will be

$$\hat{u}(p) = \arg\min_{u \in U} \tilde{E}(p,u)$$

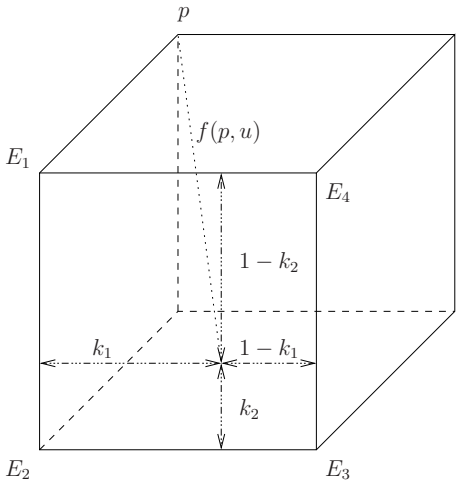
In case the new energy $\tilde{E}(p,\hat{u}(p))$ is smaller than the energy associated with point p in set F, we do the following:

- Update the energy and also control value for the point p in set F with $\hat{u}(p)$ and $\tilde{E}(p,\hat{u}(p))$
- Add surrounding points $Sur(p)$ of point p to set I, as these points might also improve their energy values and optimal control values as their neighbor information changed. In set I, associate these surrounding points with energy $\tilde{E}(p,\hat{u}(p))$ through which they might get improved. If these points already exist in I, update the energy associated with them only in case the value $\tilde{E}(p,\hat{u}(p))$ is lower than the one already associated with them.

8. If set I (set of points which might potentially improve their energy) is empty, the algorithm is done and the final solution is the set F. Otherwise go back to step 7.

Remark II.1. This is just an implementation detail we use for set I. The algorithm uses a list of points sorted by the energy values of their neighbor which recently updated its energy value and also a hash-table of points to the same energy value. This is done in case a point we need to add to I is already there and we just need to update the value of energy it is associated with in I. The combination of sorted list and hash-table significantly improves the algorithm performance.

Remark II.2. Estimation of energy values for three dimensional problems is shown in fig. 2.4.



Obr. 2.4: Testing u for point p. Direction $f(p,u)$ points to a point in between of other four points on the mesh $(p+\tilde{t}(p,u)f(p,u))$, whose energy values can be used to approximate a new energy value of p. This is now displayed for a three dimensional problem.

$$\begin{aligned}\tilde{E}(p,u) = \tilde{t}(p,u) &+ k_1\Big(k_2E_2 + (1-k_2)E_1\Big) \\ &+ (1-k_1)\Big(k_2E_3 + (1-k_2)E_4\Big).\end{aligned}$$

Similarly, this can be done even for more than three dimensional problems. One can use recursion to simplify the implementation of this part of the algorithm.

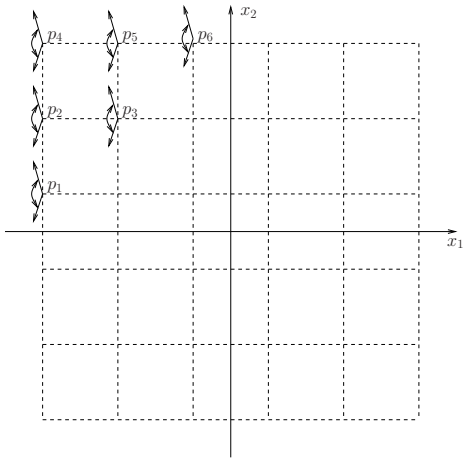
The evaluation function of found control function can be expressed as

$$J = \int ... \int E(p,\hat{u}) \approx \sum_i E(p_i,\hat{u})$$

over the region of interest and in fig. 2.9, 2.11 and 2.13 on the right side, one can see decreasing evaluation with increasing number of mesh points for three different examples.

2.3 Points Pointing Only Out of Bounds

In some cases (for example an inverted pendulum), one point on the border points only outside of region of interest and hence it's energy value cannot ever be updated. This would further cause other points around it not to reach any other energy value than ∞. In fig. 2.5 we display what happens for an inverted pendulum.



Obr. 2.5: Because p_1 points only outside of the region of interest, its energy value can't be ever updated and hence even point p_3 cannot ever update its value and finally because of p_3, point p_6 can't update its value either. This is happening due to the structure of the mesh rather than because of the example we run the algorithm on, as point p_6 can easily convert to the desired value through the region of interest for an inverted pendulum.

Due to this disadvantage of the algorithm, we approximate energy values outside of the box. See fig. 2.6 for details and following approximation of energy outside the box:

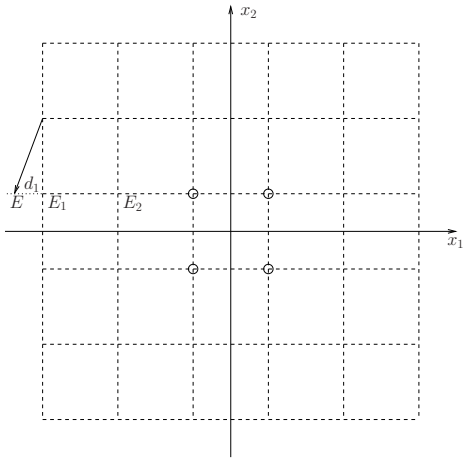
$$E \approx E_1 + (E_1 - E_2)(-d_1),$$

but only in case if $0 \geq d_1 \geq -1$.

2.4 Error Estimation

When computing energy value estimations, there are two sources of error. One comes from estimation $\tilde{t}(p, u)$ (ε_t), the second one from estimations of k_1 (ε_E) as both approximations count on constant behavior of vector $f(p, u)$. Due to this, we can estimate the error as the following:

$$\tilde{E}(p, u) = (\tilde{t}(p, u) \pm \varepsilon_t) + ((k_1 \pm z_1)(E_1 \pm \varepsilon_1) + (1 - k_1 \mp z_1)(E_2 \pm \varepsilon_2)),$$



Obr. 2.6: Estimation of energy E outside of the box based on values $E_1 < \infty$ and $E_2 < \infty$ inside of the box.

where ε_1 is the estimated error associated with the same point as energy E_1, the same holds for ε_2. The first error can be estimated as

$$\varepsilon_t = \frac{x_m^{max} - x_m^{min}}{M_m - 1} \left| \frac{1}{f_m(p,u)} - \frac{1}{f_m(p + \tilde{t}(p,u) f(p,u), u)} \right|,$$

where

$$m = \arg \min_{i \in \{1,\ldots,N\}} \left| \frac{(x_i^{max} - x_i^{min})/(M_i - 1)}{f_i(p,u)} \right|.$$

The second part of the error comes from the estimation of energy between the points and can be estimated as follows:

$$\varepsilon_E = |z_1(E_1 - E_2)| + k_1\varepsilon_1 + (1 - k_1)\varepsilon_2,$$

where

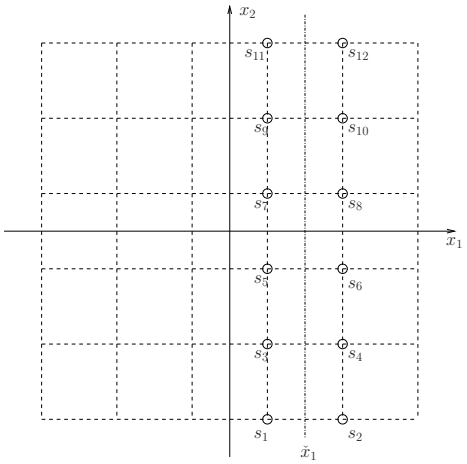
$$z_i = \left| \frac{\tilde{t}(p,u)(f_i(p,u) - f_i(p + \tilde{t}(p,u) f(p,u), u))}{x_i^{max} - x_i^{min}} (M_i - 1) \right|$$

for i-th coordinate.

2.5 Extensions of the algorithm

For certain problems, such as the DC to DC converter example, the desired value can be set not around origin 0, but around a value we try to converge to. Also, only one of the state space variables might be optimized and hence the initial set S in the algorithm might include more points. This is displayed in fig. 2.7.

This means, that if the algorithm is not supposed to optimize over variable x_2, then $Z_2 = \{1, \ldots, M_2\}$, containing all the indexes in its coordinate.



Obr. 2.7: This is how we extend set $S = \{s_1, \ldots, s_{12}\}$ for the algorithm if we optimize only over variable x_1 with the desired value $\check{x}_1$.

Mesh Size	50x50	75x75	100x100
ε_{max}	0.1341	0.0887	0.0711
Computation time	6s	13s	22s

Tabuľka 2.1: Results (maximum estimation error for a point on the mesh and computation time in seconds) for the inverted pendulum for different mesh sizes.

For a robust problem, where function f is dependent on a set of parameters q, function $f(p, u, q)$ is dependent on q and we have several sets of parameters Q, we estimate energy the following way:

$$\tilde{E}(p, u) = \sum_{q \in Q} (\tilde{t}(p, u, q) + k_1(q)E_1 + (1 - k_1(q))E_2).$$

2.6 Example 1

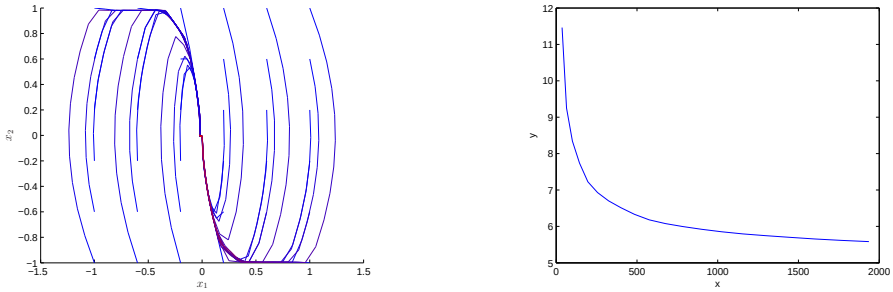
We use a single inverted pendulum to demonstrate the algorithm on a two-dimensional control problem. For simulating such a system, we use the following simplified equations:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \sin(x_1) + u$$

We use control boundaries $U = [-3, 3]$ (100 evenly distributed control values are used) and the region of interest $X = [-1, 1] \times [-1, 1]$. Results summary is in table 2.1. Trajectories of 6x6 points one can see in fig. 2.9.

Obr. 2.8: Inverted Pendulum. On the left, energy function $E(i_1, i_2)$ for a mesh 100x100. On the right, error estimation for meshes 100x100, 75x75, 50x50 from bottom to top. Error estimation decreases with increasing mesh size.



Obr. 2.9: Inverted Pendulum. (Left) On the region of interest 6x6 points were chosen and their convergence to origin can be seen. The trajectories are more blue towards time $t = 0$, and more red towards time $t = 3$. (Right) Decreasing evaluation function with increasing number of mesh points.

2.7 Example 2

The second example is DC to DC converter

$$\dot{x}_1 = 0.25(x_2 - I_L)$$

$$\dot{x}_2 = -x_1 - x_2 + u$$

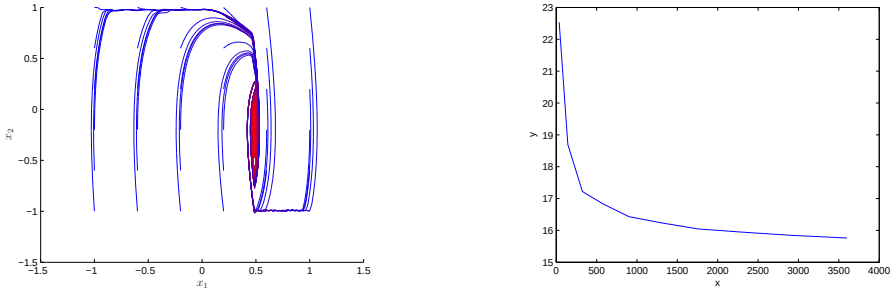
and the region of interest is $X = [-1, 1] \times [-1, 1]$ with control $U = [-1, 1]$ (11 evenly distributed control values on this interval is used). A robust solution is found using three different possible loads $I_L \in \{-0.2, 0.1, 0.3\}$. Result summary is in table 2.2. Trajectories of 6x6 points are displayed in fig 2.11.

2.8 Example 3

The last example is a three dimensional example in order to show that the algorithm is easy to use even on higher dimensional problems and that computations can be done in a very short time. It is convexed Reeds-Shepp (CRS) model.

$$\dot{x}_1 = u \cos(x_3)$$

Obr. 2.10: DC to DC converter. On the left, energy function $E(i_1, i_2)$ for a mesh 200x200. On the right, error estimation for meshes 100x100, 150x150, 200x200 from bottom to top. Error estimation decreases with increasing mesh size.



Obr. 2.11: DC to DC converter. (Left) On the region of interest 6x6 points were chosen and their convergence to desired value can be seen. The trajectories are more blue towards time $t = 0$, and more red towards time $t = 10$. (Right) Decreasing evaluation function with increasing number of mesh points.

$$\dot{x}_2 = u \sin(x_3)$$

$$\dot{x}_3 = v$$

The table 2.3 shows the summary results, where the error estimation decreases with higher mesh size and also shows computation times in seconds on a single core machine. In fig. 2.12 is displayed energy level and error estimation for a chosen x_3 close to desired value $\check{x}_3$, which is middle of region of interest. In fig. 2.13 are displayed trajectories for 4x4x4 points converging to origin.

2.9 Conclusion

We have shown a new approach for finding optimal control on a mesh, which is similar to a set-oriented approach and subdivision algorithm for optimal control. Computation times, shown on two and three dimensional examples, are better than computation times for the subdivision algorithm for optimal control. We have also shown error estimations for different mesh sizes, which adds value to the algorithm

Mesh Size	100x100	150x150	200x200
ε_{max}	0.4748	0.3130	0.2457
Computation time	6s	14s	29s

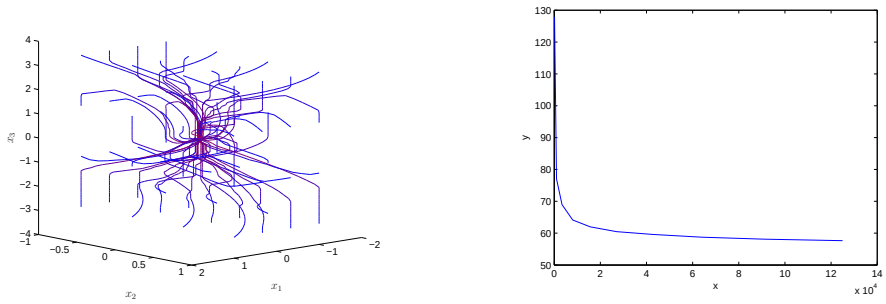
Tabuľka 2.2: Results (maximum estimation error for a point on the mesh and computation time in seconds) for the DC to DC converter for different mesh sizes.

Obr. 2.12: CSR model. On the left, energy function $E(i_1, i_2, 50)$ for a mesh 100x100x100 with chosen x_3 in the middle of region of interest. On the right, error estimation for mesh 100x100x100 with chosen x_3 in the middle of region of interest.

described in this chapter in comparison with the subdivision algorithm for optimal control. The optimal control mesh algorithm improves upon the set-oriented approach by having mesh consisting of points evenly distributed, while the set-oriented approach needs to adapt mesh in certain regions, which makes the memory foot-print of a solution larger. The software for the algorithm described in this chapter can be downloaded from [52].

Mesh Size	50x50x50	75x75x75	100x100x100
ε_{max}	0.1751	0.1084	0.0859
Computation time	21s	69s	159s

Tabuľka 2.3: Results (maximum estimation error for a point on the mesh and computation time in seconds) for the CSR model for different mesh sizes.



Obr. 2.13: CSR model. (Left) On the region of interest 4x4x4 points were chosen and their convergence to origin can be seen. The trajectories are more blue towards time $t = 0$, and more red towards time $t = 5$. (Right) Decreasing evaluation function with increasing number of mesh points.

CHAPTER III

Conclusion

In this chapter we summarize main contributions of this thesis. In chapter II, we described the subdivision method and box dimension of attractors of dynamical systems. The main contributions of the chapter II are

- Method, which approximates the box dimension of attractors. This method speeds up the convergence of the box dimension by using the information of several subdivision steps.
- With heuristic arguments we explain why the new method for box dimension approximation converges faster.
- Experiments on several examples (H‘enon, Lorenz, R¨ossler and Chua attractors) confirm previous two points.
- We show a counterexample, in which case the box dimension does not convert (Cantor set).

In chapter III, we introduce subdivision algorithm for finding optimal control. Main contributions of this chapter are

- Algorithm for finding optimal control in chosen region of interest, which does not require any strong assumptions for controlled problem nor model approximation of the controlled problem.
- Comparison with current method - set-oriented approach.
- Heuristic analysis of the method.
- Examples, including example for finding robust solution to control problem with unknown parameter.

Finally, in chapter IV, we introduced Optimal Control Mesh and the main contributions are

- Algorithm substantially faster than Subdivision Algorithm introduced in previous chapter.

- Error estimation for found solution as it is done similarly for other numerical algorithms such as Finite Elements Methods.
- Extended algorithm for finding robust solutions.
- Examples, which shown comparable results to algorithm introduced in chapter III as well in other papers (for example set-oriented approach).
- Higher dimensional solution for CSR model (three dimensions).
- Dependence of solution's energy value with decreasing mesh size (finer mesh). Examples have shown exponential dependance and based on that we can estimate what mesh size is sufficient for a good solution.

Obr. 3.1: Map of visitors of the website http://www.optirol.com.

The software Optirol, which is described in the chapter V of this thesis and uses algorithm introduced in chapter IV, can be downloaded from http://www.optirol.com [52]. The website http://optirol.com in one year of its life had visitors from 5 continents and more than 350 researchers downloaded it (fig. 3.1).

CHAPTER IV

Publications

List of publications, references, conference papers, patents and active presentations.

Publication 1

Taraba. P: Subdivision Algorithm For Optimal Control, Wiley, International Journal on Robust and Nonlinear Control (in press), January 2012

Publication 2

Taraba P.: Kneser-Ney smoothing with a correcting function for small data sets, IEEE Transactions on Audio, Speech and Language Processing, August 2007

Reference 2.1

NI. Chong-jia, LIU. Wen-ju, XU. Bo: Research on Large Vocabulary Continuous Speech Recognition System for Mandarin Chinese, Journal of Chinese Information Processing, 2009

Publication 3

Siegmund S., Taraba P.: Approximation of Box Dimension of Attractors Using the Subdivision Algorithm, Dynamical Systems, An International Journal, March 2006

Reference 3.1

Tearne, O.: Collapse of random attractors for dissipative SDEs, Thesis (Ph.D.)–University of Warwick, 2006

Reference 3.2

Taraba. P: Subdivision Algorithm For Optimal Control, Wiley, International Journal on Robust and Nonlinear Control, January 2012

Publication 4

Duc L.H., Ilchmann A., Siegmund S., Taraba P.: Asymptotic Stability of Linear Time-Varying Second-Order Differential Equations, The Quarterly of Applied Mathematics, Brown University, January 2006

Reference 4.1

M Porfiri, DJ Stilwell, EM Bollt: Synchronization in random weighted directed networks, Circuits and Systems I: Regular ..., 2008 - ieeexplore.ieee.org

Reference 4.2

A Pisano, E Usai: Contact force regulation in wire-actuated pantographs via variable structure control and frequency-domain techniques, International Journal of Control, 2008 - Taylor and Francis

Reference 4.3
J Sugie: Influence of anti-diagonals on the asymptotic stability for linear differential systems, Monatshefte für Mathematik, 2009 - Springer
Reference 4.4
A Berger, S Siegmund: Uniformly attracting solutions of nonautonomous differential equations, Nonlinear Analysis: Theory, Methods ..., 2008 - Elsevier
Reference 4.5
J Sugie: Global asymptotic stability for damped half-linear oscillators, Nonlinear Analysis: Theory, Methods and Applications, 2011 - Elsevier
Reference 4.6
J Rodriguez, MO Hongler: Networks of limit cycle oscillators with parametric learning capability, Recent Advances in Nonlinear Dynamics and ..., 2009 - Springer
Reference 4.7
L Berezansky, E Braverman: Nonoscillation and Stability of the Second Order Ordinary Differential Equations with a Damping Term, arXiv preprint arXiv: ..., 2008 - arxiv.org
Reference 4.8
M Onitsuka: Non-uniform asymptotic stability for the damped linear oscillator, Nonlinear Analysis: Theory, Methods and Applications, 2010 - Elsevier
Reference 4.9
A Delgado: Describing function of MOM systems Nanotechnology, (IEEE-NANO), 2011 11th IEEE ..., 2011 - ieeexplore.ieee.org
Reference 4.10
GR Hovhannisyan: Levinson theorem for two by two system, personal.kent.edu
Reference 4.11
S Hata, J Sugie: A necessary and sufficient condition for the global asymptotic stability of damped half-linear oscillators, Acta Mathematica Hungarica - Springer
Reference 4.12
J Sugie, T Shimadu, T Yamasaki: Global Asymptotic Stability for Oscillators with Superlinear Damping, Journal of Dynamics and Differential ..., 2012 - Springer
Reference 4.13
M Onitsuka: Uniform asymptotic stability for damped linear oscillators with variable parameters, Applied Mathematics and Computation, 2011 - Elsevier
Reference 4.14
M. Onitsuka: Asymptotic stability and uniform asymptotic stability for second-order linear differential equations with damping, 2008 - kurims.kyoto-u.ac.jp
Conference paper 1
Taraba P. : Optimal Control Mesh, International Conference on Modeling, Simulation and Control, San Francisco, October 2012
Conference paper 2
Taraba P., Kozak S. : MPC Control using AR Volterra Models, IEEE 11th Mediterranean Conference on Control and Automation MED'03, June 2003
Conference paper 3
Taraba P., Kozak S.: Model Predictive Control of chemical reactors, 14th International Conference on Process Control 2003, June 2003

Patent 1
Taraba P., Corradini G., Subert M. : Image Normalization For Computed Image Construction, Issued Patent: US7809211, October 2010
Active presentation 1
Talk 1: Self Tuning Controllers, Talk 2: Mathematical algorithms for detection of movement, IDSIA, Lugano, Switzerland, 09/26/2002
Active presentation 2
Approximation of Attractors Using the Subdivision Algorithm, Third International Workshop on Taylor Methods, Miami Beach, USA, 09/16-20/2004
Active presentation 3
Improving PID controllers using filters, Gesellschaft für Ang. Mathematik und Mechanik e.V. (GAMM 2005), Luxembourg, 03/28-31/2005
Active presentation 4
Kneser-Ney smoothing with a correcting function for small data sets, Microsoft Research, Redmond, USA, 02/22/2008

Literatúra

[1] Y. Gao, Uncertain inference control for balancing an inverted pendulum, Fuzzy Optimization and Decision Making, December 2012, Volume 11, Issue 4, pp 481-492

[2] H. Nijmeijer and H. Berghuis, *On Lyapunov Control of the Duffing Equation*, IEEE Transactions On Circuits And Systems - I Fundumental Theory And Application, Vol. 42, No. 8, August 1995.

[3] M. Mirrahimi, P. Rouchon, G. Turinici, *Lyapunov control of bilinear Schrödinger equations*, Automatica 41 (2005) 1987 - 1994.

[4] K. Beauchard, J.M. Coronb, M. Mirrahimi, P. Rouchon, *Implicit Lyapunov control of finite dimensional Schrödinger equations*, Systems & Control Letters 56 (2007) 388 - 395.

[5] H. K. Khalil. *Nonlinear Systems*. MacMillan, New York, 1992.

[6] L. S. Pontryagin, *The Mathematical Theory of Optimal Processes*, 1962.

[7] D. Richtmeyer, K.W. Morton, *Difference Methods for Initial Value Problems*, 2nd ed., Wiley, New York, 1967.

[8] K. Miller, *Moving finite elements*. II, SIAM Journal on Numerical Analysis, 1981.

[9] P. Taraba, S. Kozak, MPC control using AR-Volterra models, Proceedings of the 11th Mediterranean Conference on Control and Automation MED'03 , Rhodos, Greece, 2003

[10] K. Alligood, T.D. Sauer, J.A. Yorke, *Chaos. An introduction to dynamical systems. Textbooks in Mathematical Sciences.* Springer-Verlag, New York, 1997.

[11] B. Aulbach, M. Rasmussen, S. Siegmund, Invariant manifolds as pullback attractors of nonautonomous differential equations, submitted.

[12] M. Benedicks and L. Carleson, The dynamics of the Hénon map. *Ann. of Math.* **2** (1991), 73–169.

[13] M. Dellnitz, A. Hohmann, A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numer. Math.* **75** (1997), 293–317.

[14] M. Dellnitz, O. Junge, Set oriented numerical methods for dynamical systems. Handbook of dynamical systems, Vol. 2, 221–264, North-Holland, Amsterdam, 2002.

[15] M. Dellnitz, O. Schütze, S. Sertl, Finding zeros by multilevel subdivision techniques. *IMA Journal of Numerical Analysis* **22** (2002), 167–185.

[16] K.J. Falconer, *Fractal geometry: mathematical foundations and applications.* Chichester, Wiley, 1993.

[17] K. Gelfert, Maximum local Lyapunov dimension bounds the box dimension. Direct proof for invariant sets on Riemannian manifolds. *Z. Anal. Anwendungen* **22** (2003), 553–568.

[18] P. Grassberger, On the fractal dimension of the Hénon attractor. *Phys. Lett. A* **97** (1983), 224–226.

[19] M.Y. Li and J.S. Muldowney, Lower bounds for the Hausdorff dimension of attractors. *J. Dynam. Differential Equations* **7** (1995), 457–469.

[20] E.N. Lorenz, Deterministic non-periodic flow. *J. Atmospheric Sci.* **20** (1963), 130—141.

[21] T. Matsumoto, L.O. Chua and M. Komuro, The double scroll. *IEEE Trans. Circuits and Systems* **32** (1985), 797–818.

[22] Y.B. Pesin, *Dimension theory in dynamical systems. Contemporary views and applications.* Chicago Lectures in Mathematics. University of Chicago Press, Chicago, IL, 1997.

[23] O.E. Rössler, An equation for continuous chaos. *Phys. Lett.* **57** (1976), 397–398.

[24] C. Sparrow, *The Lorenz equations: bifurcations, chaos, and strange attractors.* Applied Mathematical Sciences, 41. Springer-Verlag, New York-Berlin, 1982.

[25] J.T. Oden, S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method, Computers & mathematics with applications, 2001 - Elsevier

[26] W. Tucker, The Lorenz attractor exists. *C. R. Acad. Sci. Paris Sér. I Math.* **328** (1999), 1197–1202.

[27] A. Isidori, Nonlinear Control Systems, third edition, Springer-Verlag, Berlin, 1995

[28] A. Fradkov, D. Hill, Exponential feedback passivity and stabilizability of nonlinear systems, Automatica 34:697-703, 1998

[29] E.D. Sontag, H.J. Sussmann, Further comments on the stabilizability of the angular velocity of a rigid body, Systems and Control Letters (12):437-442, 1988

[30] H.K. Khalil, Nonlinear Systems, Prentice Hall Inc., Englewood Cliffs, 1996

[31] J. D. Hedengren, T. F. Edgar, Approximate Nonlinear Model Predictive Control with In Situ Adaptive Tabulation, Computers and Chemical Engineering, Volume 32, pp. 706-714, 2008

[32] G. Pannocchia, S. J. Wright, J. B. Rawlings, Partial enumeration MPC: Robust stability results and application to an unstable CSTR. In DYCOPS, Leuven, Belgium, June 2010

[33] T.A. Johansen, A. Grancharova, Approximate explicit constrained linear model predictive control via orthogonal search tree. IEEE Transactions on Automatic Control 48 (5), 810–815, 2003

[34] B. Kosko, Neural Networks and Fuzzy Systems, Prentice Hall, Englewood Cliffs, New Jersey, 1992

[35] L. Grüne, O. Junge, A set oriented approach to optimal feedback stabilization, Systems and Control Letters, 54(2):169-180, 2005

[36] S. Siegmund, P. Taraba, Approximation of box dimension of attractors using the subdivision algorithm, Dynamical Systems 21, 1-24

[37] M. Dellnitz and A. Hohmann, A subdivision algorithm for the computation of unstable manifolds and global attractors, Numerische Mathematik 75 (1997) 293-317

[38] A. V. Roup and D. S. Bernstein, Adaptive Stabilization of a Class of Nonlinear Systems With Nonparametric Uncertainty, IEEE Transactions On Automatic Control, Vol. 46, No. 11, November 2001

[39] A. Rantzer, M. Johansson, Piecewise Linear Quadratic Optimal Control, IEEE Transactions On Automatic Control, Vol. 45, No. 4, April 2000

[40] B. Lincoln, A. Rantzer, Relaxing dynamic programming, IEEE Transactions On Automatic Control, Vol. 51, Issue 8, August 2006

[41] L. Grüne, O. Junge, Optimal stabilization of hybrid systems using a set oriented approach, Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Japan, pp. 2089-2093, 2006

[42] L. Grüne, O. Junge, Global optimal control of perturbed systems, Journal of Optimization Theory and Applications (JOTA) 136(2008), 411-429

[43] L. Grüne, O. Junge, Set oriented construction of globally optimal controllers, at-Automatisierungstechnik, 57(2009), 287-295

[44] L. Grüne, F. Müller, Global optimal control of quantized systems, Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems MTNS2010, Budapest, Hungary, 2010, 1231-1237

[45] P. O. Gutman, A linear programming regulator applied to hydroelectric reservoir level control, Automatica, 22(5), 1986, 533-541

[46] Alberto Bemporad, Manfred Morari, Vivek Dua, Efstratios N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica, 38, 2002, 3-20

[47] Francesco Borrellia, Mato Baotic, Alberto Bemporadb, Manfred Morari, Dynamic programming for constrained optimal control of discrete-time linear hybrid systems, Automatica, 41, 2005, 1709 – 1721

[48] L. Grüne, O. Junge, A set oriented approach to optimal feedback stabilization, Systems and Control Letters, 54(2):169-180, 2005

[49] L. Grüne, O. Junge, Optimal stabilization of hybrid systems using a set oriented approach, Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Japan, pp. 2089-2093, 2006

[50] P. Taraba, Subdivion Algorithm for Optimal Control, Wiley, International Journal on Robust and Nonlinear Control, 2012, doi: 10.1002/rnc.2801

[51] H.J. Sussmann, G. Tang, Shortest paths for the Reeds-Shepp Car: A worked out example of the use of geometric techniques in nonlinear optimal control, Technical Report No. SYNCON 91-10, Department of Mathematics, Rutgers University

[52] P. Taraba, Optirol, Available: http://www.optirol.com