Czech Technical University in Prague
Faculty of Electrical Engineering

Doctoral Thesis

August 2018 **Matej Pčolka**

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

Numerical Aspects of Optimal Control Design for Nonlinear Systems

Doctoral Thesis

Matej Pčolka

Prague, August 2018

Programme: Electrical Engineering and Information Technology (P 2612)
Branch of study: Control Engineering and Robotics (2612V042)

Supervisor: Prof. RNDr. Sergej Čelikovský, CSc.

Thesis Supervisor:
Prof. RNDr. Sergej Čelikovský, CSc.
Department of Control Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo náměstí 13
121 35 Prague 2
Czech Republic

Aut inveniam viam aut faciam.

I shall either *find* a way or *make* one.

– Hannibal

This work is dedicated to my parents and Evinka.

Abstract

In this thesis, numerical aspects of optimal control design for nonlinear systems are studied. The issues it tackles are inspired by questions that arise when designing a nonlinear model predictive controller and a need of responding to numerous phenomena that can be encountered during its operation.

The first part of the thesis deals with choice of optimization step for the underlying local optimization routine. Various step length choices are investigated with respect to both their convergence properties and the optimality of the resulting solution and, moreover, expressions for determining search step length making use of information about either the evolution of the optimization criterion value or the distance from the considered constraints are provided.

In the second part, a task of specifying an appropriate optimization horizon is discussed with three special cases elaborated in detail. At first, two approaches for measurement feedback introduction are compared. The second case focuses on efficiency maximization, considers the horizon itself to be an optimizable variable and provides two algorithms tailored to this task. In the last case, three class-specific adaptive horizon formulas for nonlinear model predictive controller decreasing its computational/memory demands are proposed.

Presence of discontinuities in the optimal control task and robustification of the optimization algorithm against the corresponding negative effects constitute the topic of the third part of the thesis. This part can be further split into two more specialized ones—in the first of them, an algorithm based on a mid-processing iteration suitable for discrete-valued input optimization is introduced while the second one contributes with an algorithm capable of handling optimal control tasks including hybrid systems and non-smooth/discontinuous objective function.

The last part of the thesis elaborates a comprehensive framework for overall optimization of production processes. The developed algorithm employs a rigorous re-parameterization procedure and expresses the input profile as a parametric function of continuous time, which enables to reduce the cardinality of the set of the optimized input profile parameters. In addition to the commonly considered manipulated variables, the proposed optimization routine adjusts also the initial conditions of the operated process to exploit its full production potential.

Keywords:

Optimal control of nonlinear systems, nonlinear model predictive control, optimization algorithms, quantized input optimization, control of hybrid systems, production-process optimization.

Abstrakt

V tejto práci sú študované numerické aspekty optimálneho riadenia nelineárnych systémov. Problémy, ktoré rieši, sú inšpirované otázkami, ktoré vyvstávajú pri návrhu nelineárneho prediktívneho regulátora založeného na modeli, a potrebou reagovať na javy, s ktorými je možné sa stretnúť počas jeho prevádzky.

Prvá časť práce sa zaoberá výberom optimalizačného kroku príslušnej optimalizačnej rutiny. Sú vyšetrované rôzne voľby dĺžky kroku s ohľadom na ich konvergenčné vlastnosti i optimalitu výsledného riešenia a navyše sú poskytnuté aj výrazy pre určenie dĺžky kroku využívajúce informáciu buď o vývoji hodnoty optimalizačného kritéria alebo o vzdialenosti od uvažovaných obmedzení.

V druhej časti je diskutovaná úloha špecifikovania vhodného optimalizačného horizontu a tri špeciálne prípady sú rozpracované do detailov. Najprv sú porovnané dva prístupy k zavedeniu spätnej väzby od meraní. Druhý prípad sa zameriava na maximalizáciu efektívnosti, uvažuje horizont ako optimalizovateľnú premennú a poskytuje dva algoritmy šité na mieru tejto úlohe. V poslednom prípade sú pre určitú triedu navrhnuté tri formuly pre adaptívne nastavenie horizontu pre nelineárny prediktívny regulátor založený na modeli znižujúce jeho výpočtové/pamäťové nároky.

Prítomnosť nespojitostí v úlohe optimálneho riadenia a robustifikácia optimalizačného algoritmu voči odpovedajúcim negatívnym efektom predstavujú tému tretej časti práce. Táto časť môže byť ďalej rozdelená na dve viac špecializované: v prvej z nich je predstavený algoritmus založený na medzispracujúcej iterácii vhodný pre optimalizáciu vstupov s diskrétnymi hodnotami, zatiaľ čo druhá prispieva algoritmom schopným poradiť si s úlohami optimálneho riadenia obsahujúcimi hybridné systémy a nehladké/nespojité účelové funkcie.

Posledná časť práce vypracováva komplexný rámec pre celkovú optimalizáciu produkčných procesov. Vyvinutý algoritmus využíva rigoróznu reparametrizačnú procedúru a vyjadruje vstupný profil ako parametrickú funkciu spojitého času, čo umožňuje zredukovať mohutnosť množiny optimalizovaných parametrov vstupného profilu. Ako prídavok ku bežne uvažovanej optimalizácii manipulovaných premenných upravuje navrhnutá optimalizačná rutina aj počiatočné podmienky ovládaného procesu s cieľom využiť jeho úplný produkčný potenciál.

Kľúčové slová:

Optimálne riadenie nelineárnych systémov, nelineárne prediktívne riadenie založené na modeli, optimalizačné algoritmy, optimalizácia kvantovaných vstupov, riadenie hybridných systémov, optimalizácia produkčných procesov.

Acknowledgements

First of all, I would like to express my gratitude to Sergej Čelikovský, my supervisor, for his enduring willingness to discuss and direct my ideas, for his priceless suggestions, fruitful discourses and his support and advice he has provided me with.

I would like to thank Rush Robinett for being my mentor at Michigan Tech in the true sense of the word and for showing me the big picture ("life is an optimization task—define your cost function and optimization strategy and when you're not satisfied with what you've got, update *both* the strategy *and* the cost function and try again").

Many thanks go to Michael Šebek for enabling me to take part in all those conferences that have broadened my horizons and also to the department staff, Petra Stehlíková, Helena Doležílková, Jaroslava Nováková and Lenka Jelínková, for helping with all the paperwork.

I must not omit my colleagues at KU Leuven—Lieve Helsen, Stefan Antonov, Maarten Sourbron, Clara Verhelst, Jan Hoogmartens and Wout Parys—for being an affable companion and for creating a research-friendly environment during my stay. I spent some time also at Michigan Tech and without Mahdi Shahbakhti, Meysam Razmara, Gordon Parker and Nina Mahmoudian, it would have been just three empty months filled with snow. With you, folks, it was a wonderful and highly inspiring stay (filled with snow anyway).

I would like to acknowledge: Martin Štellmach for being a *blood brother* of mine, a truly staunch friend willing to find a minute or two (or an hour or two) for a refreshing chat every time I came back home (and for rehearsing for all those sold-up concerts that have never taken place); Jaroslav Kříž for his inexhaustible pull-up-, music-, linguistic- and prank-enthusiasm; Jakub Šimánek for spending a significant part of our PhD journey with us and for sharing his thesis template; and Ondřej Bruna for the upper-class dinner in Barcelona, for the escape games he organized during the months we were supposed to spend writing our theses, for the enriching meditative evening debates and for the unforgettable quotes ("highly educated yeasts" and "they're eating ——— !" will accompany me forever).

I am deeply grateful to my family as well. Thank you parents for your warm-hearted attitude, care and love (even though sometimes "I've broken a shovel. . . ") and thank you for supporting me on my life path. Also, thank you for watching all the 1000-and-1 episodes of my never-ending photodocumentaries and for discussing so many control-related matters no parents have probably discussed yet. Dominika, Paulinka und Alek, thank you for creating an oasis of relax and joy in the town of pouring (?) and for providing a 24/7 *worldwide* (!) medical consulting and petrol patrol service.

Above all, I would like to thank Eva Žáčeková. Wherever one goes, the most important thing is who one travels with. Thank you for those many inventions that are yet to be patented (e.g. litmus for micro-animals detection), thank you for travelling 5 continents with me (still, I need to see a wild kiwi bird before retirement), thank you for jumping out of a plane with me, thank you for listening to all my "scientific" speculations. All in all, thank you for sharing both the good and bad, thank you for *everything*!

Also, I would like to thank *you*, The Reader going to dive into this thesis. Thank you.

Declaration

I hereby declare that this dissertation thesis is my own personal effort and I have carefully cited all the literature sources I had used.

Prague, 20. 8. 2018

Prehlasujem, že som svoju dizertačnú prácu vypracoval samostatne a v predloženej práci dôsledne citoval použitú literatúru.

V Praze, 20. 8. 2018

Contents

Chapter 1

Introduction and Motivation

Since the very earliest ages, people have been trying to solve everyday problems where more than one option could come into account to achieve their own benefit—prehistoric groups of hunters/gatherers had to choose the place of their current stay according to the availability of the food and the risk of being attacked by another group, army commanders needed to create sophisticated strategies to defeat the enemy troops with as little own casualties as possible, farmers had to decide growing of which crop could bring the best profit. Especially important are these problems and solving of them in the field of engineering and economics—how shall an investor react to the stock market situation or what is the most effective trajectory of a spacecraft to reach the Moon?

The aforementioned and many other tasks have much in common—if the problem is well posed and feasible, there exists at least one choice that leads to the best achievable result with respect to the desired objective that can be expressed as a more or less complicated function of the decision variables. Moreover, these tasks need to be solved with respect to many constraints that can also be formulated in a mathematical form either as equalities, inequalities or other expressions. Way of solving such problems has attracted many scientists and a number of effective approaches have been developed and formed into a very wide area of *mathematical optimization*.

Coming out of the mathematical apparatus developed by Fermat, Lagrange, Newton and Gauss, one of the first theoretical achievements in this field were made by George B. Dantzig and Leonid Kantorovich as soon as during the first half of the 20-th century. Shortly after the World War II., Richard E. Bellman invented *Bellman equation* while Lev Semenovich Pontryagin formulated his maximum principle which both became cornerstones of the *optimal control theory*. Since then, generations of mathematicians such

as Karush, Kuhn, Tucker, Nemirovski, Shor and others have successfully addressed many open problems in the optimal control theory and helped developing methods appropriate for solving the optimization tasks.

Both in the area of industrial control and in the field of systems and control theory, optimal control approaches are getting more and more popular. Essentially, it could be said that if some problem *can* be formulated as an optimization task and the available hardware allows for it, then the problem also *is* solved as an optimization task using the available optimization techniques. Out of all branches of optimal control, Model Predictive Control (MPC) is currently the most attractive one and it has become almost a synonym for *applied optimal control*.

Early beginnings of this control branch can be dated back to late '80s when an idea of *generalized predictive control* was introduced [Clarke et al., 1987a], [Clarke et al., 1987b]. Soon, linear version of the MPC considering linear dynamics of the controlled system and quadratic cost function penalizing sum of squares of control actions and reference tracking errors over a prediction horizon with respect to equality/inequality constraints imposed upon the inputs/states/outputs was introduced [Muske and Rawlings, 1993], [Maciejowski and Huzmezan, 1997], [Morari and H Lee, 1999], [Camacho and Bordons, 2004]. Shortly after that, the MPC has become one of the most frequently employed representative of the optimal control applied in industrial practice with solidly growing number of applications [Qin and Badgwell, 2003].

The need to push the number of online calculations as low as possible has led to the concept of *explicit* MPC [Bemporad et al., 2002]. The main idea consists in offline precalculation of the optimal control actions for the expected set of the state values and parameterization of these inputs by the corresponding state values, which enables that only simple table search is performed online. As one can note, the precalculation can be either highly time-consuming or even impossible due to the curse of dimensionality for larger systems, which is the main bottleneck of this concept.

While earlier, the linear MPC was the only version of the predictive control that was computationally tractable in real operation, with the development of computational resources and optimization techniques, more involved branches of predictive control have been studied. For a wide variety of industrial applications, the area of *nonlinear* MPC [Diehl et al., 2009], [Mayne, 2014] can be particularly attractive. Here, it should be noted that although much effort has been spent in this area, the currently most advanced methods almost always introduce certain kind of approximation and linearization and having

performed these simplifications, methods of quadratic or linear programming are employed. In this way, a quick and computationally nondemanding solution of the optimization problem can be obtained, however, it can be expected that these approximations bring some inevitable inaccuracies into the optimal control design which can degrade the controller performance. Thanks to relocating the main computational power from the "online" world to the "offline" one, explicit MPC for nonlinear systems (usually also considering inputs quantization) has been studied with considerable interest [Grancharova and Johansen, 2009], [Grancharova and Johansen, 2012], [Chakrabarty et al., 2017]—however, the burden for any problems except of those of smaller size is usually huge since the precalculation requires that a mixed-integer nonlinear programming task be solved. A number of researchers have focused on the implementational aspects of the optimization as well and *automatic code generation* [Mattingley and Boyd, 2009], [Mattingley et al., 2010], [Houska et al., 2011] has become very intensively used. Although these concepts are impressively successful in general and the obtained codes are quickly processed and solved, their contribution can be seen more in the area of implementation.

This thesis concentrates on practically motivated issues arising when a nonlinear model predictive controller using certain local optimization routine is operated in real world environment and focuses on both the development of new algorithms with extended/enhanced functionality *and* on the improvement of the computational/memory effectiveness. Trying to summarize all the mentioned issues, they can be expressed by a single question: *"how well will it (the controller) do?"* This neatly compact and interestingly embracing question sooner or later boils down to more specific ones such as *"how good control performance will be achieved?"*, *"how robust against certain phenomena will the controller be?"*, *"how demanding will the deployment/use of the controller be?"*, *"how quickly will the controller calculate input actions?"*, *"how much computational power and memory will it need?"* and uncountably many others—and it usually turns out the particular question is more or less related to certain numerical aspect of the optimal controller. Therefore, this thesis tries to provide algorithms yielding satisfactory answers to (some of) these questions, namely it deals with the choice of the search step length for the underlying optimization routine, develops algorithms robustified against influence of discontinuities (input quantization and discontinuous dynamics/discontinuous cost criterion are addressed), studies how to choose/find optimization horizon that would be beneficial for achieving more attractive control performance and decreasing computational/memory complexity and last but not least, it introduces an algorithm optimizing the production process as a whole including

minimization of the complexity of the solution and optimization of the initial conditions of the process. The algorithms are verified by the means of *in silico* experiments and compared with other possible alternatives.

1.1 Structure of the Thesis

This doctoral thesis takes the format of a *thesis by publication*, thereby it presents publications relevant to the topic of the thesis. This thesis format is approved by the Dean of Faculty of Electrical Engineering by the *Directive for dissertation theses defense, Article 1.*

The rest of this thesis is organized as follows. In Chapter 3, the choice of search step for local optimization routines is studied and several formulas for calculation of adaptive search step are provided. Chapter 4 focuses on the optimization horizon and its influence on both the observed control performance and the computational/memory demands of the optimal controller designed to optimize over the chosen horizon. In the corresponding subchapters, algorithms for adaptive horizon calculation are introduced. In Chapter 5, algorithms dealing with optimal control problems involving discontinuous phenomena—namely manipulated input quantization and discontinuous dynamics/optimization cost function—are presented. A novel approach to overall production-process optimization combining reparameterization of the input profile and joint parametric optimization of the input profile parameters and process initial conditions is proposed in Chapter 6. The thesis is concluded by Chapter 7 where the contributions of the thesis are summarized and potential future work is outlined.

Chapter 2

Contributions

The research conducted by the author summarized in this thesis was strongly motivated by the effort to provide solutions for chosen real-world problems related to or directly resulting from nonlinear nature of many systems/tasks a control engineer might encounter and certain phenomena accompanying process of optimal control design in such cases.

As such, the contributions of this thesis can be divided into the following parts corresponding to the main phenomena they are devoted to:

1. **Choice of optimization step**

   For non-convex optimization tasks, the choice of search step size for the underlying optimization routine is a highly delicate task since it affects not only the computational burden and the convergence speed but also the resulting control performance quality. In this thesis, various alternatives (namely constant steps, steps based on quadratic approximation and those performing exact line search) are studied, tested on a set of *in silico* experiments and compared. Moreover, several adaptive formulas (logarithmic, approximating second difference of Hamiltonian and exploiting sigmoid function of optimization constraints) for calculation of search step length are proposed. More details about this partial contribution are provided in Chapter 3 and publications of the author referenced therein.

2. **Optimization horizon**

   Majority of applications of optimal control (either for linear or nonlinear systems) consider the optimization horizon to be constant and chosen a priori. As shown later in this thesis, this can lead to either compromising on control performance compared

with the actually achievable potential or to unnecessarily high time/memory consumption of the designed governor. Chapter 4 of this thesis elaborates on the choice of optimization/prediction horizon with the emphasize put on *i)* feedback introduction, *ii)* optimality in case of effectiveness maximization and *iii)* reduction of the computational/memory complexity with certain behavior guarantees.

3. **Robustification against discontinuity effects**

   Extensive use of digital control subsystems and a persistent effort to decrease the communication traffic gives rise to a situation that the manipulated inputs can be only discrete-valued or belong to a finite values set. Seemingly completely different is the problem of controlling systems with discontinuous dynamics such that a piecewise continuous cost criterion is minimized. The common denominator of both cases is the presence of certain discontinuity complicating the process of optimal controller design and either leading to a mixed-integer nonlinear programming task formulation or use of various approximations/assumptions. In this thesis, two enhanced variants of the Hamiltonian-based gradient optimization method are provided. The first of them is tailored to discrete-valued inputs optimization, its computational time consumption is almost the same as that of the continuous-valued optimization and it considerably suppresses the initially observed oscillations of the input profiles. The second one deals with the task of controlling nonlinear systems with hybrid dynamics minimizing piecewise continuous criterion, provides noteworthy improvement in control performance compared with the approximated solution and requires only a fraction of computational load of the approaches solving mixed-integer nonlinear programming task. The contributions to this particular subtopic are presented in Chapter 5.

4. **Production-process optimization**

   Chapter 6 introduces a novel comprehensive algorithm embracing the optimization of the production processes as a whole. Consisting of several stages, during one of them it performs a reparameterization of the input profile in (continuous) time enabling reduction of the cardinality of the set of optimizable input profile parameters and thus also decrease of the computational demands. Additionally, a joint parametric optimization of the process initial conditions and input profile parameters executed during another stage enables the algorithm to exploit the full available potential of the production process and brings substantial gain compared with alternative strategies.

Chapter 3

Choice of Optimization Step

The first part of this thesis focuses on the choice of the optimization step and its influence on the convergence of the optimization routine and on the quality of the obtained solution.

Very often, gradient-based philosophy is used to solve the problems of optimal control design for nonlinear systems and the resulting nonlinear programming tasks. At the very beginning, it should be mentioned that these tasks might be—and usually also *are*—non-convex. Although a sophisticated choice of the search step length might contribute to achieving an attractive and satisfactory behavior in such situations, only very little guarantees can be provided, if any. Nevertheless, freeing oneself from these sorrowful visions, a number of rules for choosing the length of the search step can be come up with. The very simplest choice is a constant step with the biggest advantage of not presenting any further computational burden on the optimizer. Unfortunately, the corresponding convergence might be slightly questionable. Another option is to use diminishing steps that gradually decrease with iterations. They possess good theoretical convergence properties, but must be handled carefully to prevent the optimizer from converging to non-stationary points. Last to be mentioned are various rules for successive reduction of the step size based on decreasing the step by a chosen factor until cost function decrease is achieved. Further rules, formulas and nice discussions on this topic can be found in [Himmelblau, 1972], [Bertsekas, 1999], [Zhou et al., 2006], [Yuan, 2008], [Bazaraa et al., 2013] or in [Kuhn and Tucker, 2014] and other works mentioned therein.

In [A.1], an example of optimal control of penicillin production was considered as proving ground to test and compare several variants of the gradient optimization search step choice. Being the very first and simplest of them with search steps remaining constant over the whole process of optimization, a family of *fixed* search steps containing several mem-

bers chosen from a certain range of values was inspected. As another option, a *parabola-minimizing* steps family was evaluated coming out of an assumption that having computed the gradient, the optimization criterion in that direction could be expressed as a quadratic function of the step length. Last of all, an exhaustive line search over a vector of steps was executed when applying members of *general-curve-minimizing* family with the steps being spaced either linearly or logarithmically and in both cases, a quasi-convexity assumption could be added as well. The comparison carried out in [A.1] revealed that the best convergence rate was provided by parabola-minimizing steps, yet, this family was also sensitive to meta-parameters setting and for some of its members, the convergence was *not* achieved. Moreover, the values of the cost function were the lowest among all the inspected families. On the other hand, the individuals executing exact line search (i.e. members of the general-curve-minimizing family) were the most successful in search for the optimal value since on average, their achieved cost function values were the highest, however, this was ransomed at the price of slowest convergence.

Although the most detailed discussion about the size of the search step was given in [A.1], careful search step choices had to be made in all other publications of the author of this thesis. For the problems that did not require a more advanced solution, constant search step was employed [A.8], [A.9], [A.10], [A.11]. Here, it should be mentioned that the last of them represents also another contribution of this thesis and is provided in Chapter 4.1. Solving more involved optimization tasks, several alternatives represented by adaptive search step formulas were employed. In these situations, the step size was required to be *i)* smaller for bigger gradients (to prevent jumping out of steep but narrow "valleys") and also *ii)* larger for tiny gradients (to whiz through "plateaus" in the optimization space). [A.5] and [A.6] used search step being proportional to inverse of the second difference of the Hamiltonian with respect to the optimized input profile. To eliminate potential numerical issues, the search steps were projected on a predefined interval of admissible values. In [A.2] and [A.3], another adaptive search step approach was introduced and successfully applied—in this case, the search step was calculated using a logarithm of the cost function decrease reached in the previous iteration. Since [A.2] contributes to a different part of this thesis as well, its full text is provided in Chapter 5.1. Likewise, [A.3] studies also another subtopic covered in Chapter 5.2 and can be found therein. Last but not least, another original alternative search step length choice was proposed in [A.13]. Here, it should be mentioned that although not dealing exactly with optimal control for *nonlinear* systems, the solved optimization task was highly nonlinear due to the nature of the optimization

criterion and supplementary constraints and therefore, the two optimization tasks can be identified to be *essentially the same*. The search step choice itself was based on a sigmoid function—hyperbolic tangent in particular—of the distance from the boundary of the feasible region and offered an elegant way to handle constraints. Unlike the traditional interior point methods where the gradient of the constraint is exploited to "push" the solution away from the boundary, this approach only slows down (and eventually stops) the progress in those directions that are likely to cause constraints violation.

As already mentioned, the most detailed discussion related to this part of the thesis was presented in a dedicated section in [A.1] following on the next page. For the second-difference-approximation adaptive formulas, the readers are warmly referred to [A.5] and [A.6], the logarithm-based search step expressions can be found in [A.2] in Chapter 5.1 and in [A.3] in Chapter 5.2, and for the constraints-dependent search step, [A.13] should be consulted.

Bioprocess Biosyst Eng (2014) 37:71–81
DOI 10.1007/s00449-013-0929-z

ORIGINAL PAPER

Numerical aspects of optimal control of penicillin production

Matej Pčolka · Sergej Čelikovský

Received: 29 September 2012 / Accepted: 12 February 2013 / Published online: 20 March 2013

Abstract Since their discovery, fermentation processes have gone along not only with the industrial beverages production and breweries, but since the times of Alexander Fleming, they have become a crucial part of the health care due to antibiotics production. However, complicated dynamics and strong nonlinearities cause that the production with the use of linear control methods achieves only suboptimal yields. From the variety of nonlinear approaches, gradient method has proved the ability to handle these issues—nevertheless, its potential in the field of fermentation processes has not been revealed completely. This paper describes constant vaporization control strategy based on a double-input optimization approach with a successful reduction to a single-input optimization task. To accomplish this, model structure used in the previous work is modified so that it corresponds with the new optimization strategy. Furthermore, choice of search step is explored and various alternatives are evaluated and compared.

Keywords Optimal control · Nonlinear systems · Fermentation process · Gradient method optimization · Antibiotics production

M. Pčolka (✉)
Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo náměstí 13, 121 35 Prage 2, Czech Republic
e-mail: pcolkmat@fel.cvut.cz

S. Čelikovský
Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Prague 8, Czech Republic
e-mail: celikovs@utia.cas.cz

Introduction

Rapid increase of the industrial productivity of antibiotics that might be witnessed during the last few decades is basically owed to a massive improvement of production technologies rather than to a sophisticated control background. As a consequence, only suboptimal operation manners have been involved with final product concentrations deep below maximum reachable values. The important next step is to consider and carefully analyze all advantages and disadvantages of the used control strategy. A wide variety of ways how to operate the input feed flow (which influences the formation of the final product especially by the amount of the substrate nutrient supplied to system through it) has been discussed in literature so far. As an initial attempt, one can consider indirect feedback methods for nutrient feeding based on pH or dissolved oxygen measurements [1]—the substrate concentration is then maintained at predetermined setpoint by either a simple open-loop controller [2] or an on/off [3] or a PID type controller, followed by fuzzy approaches which appeared in the 1990s [4] and have been revitalized at the beginning of the millennium [5]. However, the most impressive results have been reached using model predictive control (MPC) approach. Several studies describing the MPC control of bioprocess, in general [6–10], and penicillin production, in particular [11, 12], can be found. The main drawback of this method is the fact that it is usually performed either with an approximately or exactly linearized mathematical model of the controlled process. Approximate linearization performed at certain operating point [7] can be invalid for operating points far away from the original one (and it is known that the operating points range varies a lot during the cultivation). Moreover, no stability assumptions can be made for closed-loop control

based on the approximate models obtained at each step and even one unstable model obtained by approximate linearization can degrade the MPC performance vastly. Exact linearization blows all these problems away—unfortunately, in the area of fermentation processes, the existence of exact linearization is rather rare and occasional. Therefore, a proper alternative is needed—gradient descent method which has already proved encouraging results in various research areas [13–15] is a strong candidate as it can handle even a nonlinear process model very effectively. The crucial points for this model based method are the availability of a mathematical model describing the biochemical process and determination of an adequate cost functional to be optimized.

The aim of this paper is to continue the previous efforts of the authors in [16], where they consider the single-input model with the input being the feed flow. The first attempts to consider a non-nutrient input and use different strategies for the double-input model were presented in [17], while the comparison of both the so-called quasi-double-input and the true-double-input strategies was performed in [18]. Based on that, the present paper goes deeper into the strategy used in [17] and studies further its efficiency both from the biotechnological and the numerical point of view. In the current paper, gradient search step choice is discussed and three alternative families are provided: (i) fixed step family, (ii) parabola-minimizing step family, and (iii) general-curve-minimizing step family. Each of the mentioned families contains more members whose results are later compared with respect to the following criteria: optimality, iterations-to-converge and time-to-converge.

The paper is organized as follows: "Model of the fermentation process" introduces nonlinear dynamical model of the fermentation process which is used for the optimization purposes. The penicillin cultivation is chosen to represent the fermentation processes, modification of the previously used model (which is crucial for the use of new control strategy introduced later in this paper) is explained. In "Optimal control design", the optimization issues of the final product concentration maximization including the constraints specification are formulated. The gradient method is introduced, its theoretical background is clarified and having done the necessary problem order reduction, the used control strategy is proposed. In "Optimization results", results of constant vaporization strategy are presented, compared to those obtained using strategy presented earlier and discussed. "Choice of gradient search step" introduces search step families and brings a brief description of particular family members, while "Results with enhanced search step choice" summarizes the results of the numerical experiments for different step choices and comments upon them. "Conclusions" concludes the paper.

Model of the fermentation process

Let us consider a penicillin cultivation [12, 16, 19] described by the following model:

$$\begin{aligned}
\frac{\mathrm{d}V}{\mathrm{d}t} &= u - V\lambda\left(\mathrm{e}^{w\frac{T_{\mathrm{opt}}-T_{\mathrm{f}}}{T_{\mathrm{b}}-T_{\mathrm{f}}}}-1\right),\\
\frac{\mathrm{d}C_{\mathrm{X}}}{\mathrm{d}t} &= (\mu - K_{\mathrm{D}})C_{\mathrm{X}} - \frac{\mathrm{d}V}{\mathrm{d}t}\frac{C_{\mathrm{X}}}{V},\\
\frac{\mathrm{d}C_{\mathrm{S}}}{\mathrm{d}t} &= -\sigma C_{\mathrm{X}} + \frac{C_{\mathrm{S,in}}u}{V} - \frac{\mathrm{d}V}{\mathrm{d}t}\frac{C_{\mathrm{S}}}{V},\\
\frac{\mathrm{d}C_{\mathrm{P}}}{\mathrm{d}t} &= \pi C_{\mathrm{X}} - K_{\mathrm{H}}C_{\mathrm{P}} - \frac{\mathrm{d}V}{\mathrm{d}t}\frac{C_{\mathrm{P}}}{V}.
\end{aligned} \tag{1}$$

Here, V (l) refers to cultivation broth volume, C_{X} (gl^{-1}) represents biomass concentration, C_{S} (gl^{-1}) stands for the limiting substrate concentration (let us consider carbon to be the limiting substrate), and C_{P} (gl^{-1}) represents the final product (penicillin) concentration. The substrate feed flow rate u (lh^{-1}) is the operated input.

Parameters λ (h^{-1}), and w (−) are specific vaporization constants, T_{opt} (K) represents empirically obtained optimal operational temperature (see [20]), and T_{f} (K) and T_{b} (K) refer to the freezing and the boiling temperature of the broth, respectively, which are considered to be the same as those of the water [12].

A simple constant term K_{D} (h^{-1}) models biomass death kinetics, while the total of specific biomass growth rate μ (h^{-1}) and the specific production rate π (h^{-1}) weighted by biomass-on-substrate yield coefficient $Y_{\mathrm{X/S}}$ and the product-on-substrate yield coefficient $Y_{\mathrm{P/S}}$ gives specific substrate consumption rate σ (h^{-1}):

$$\sigma = Y_{\mathrm{X/S}}^{-1}\mu + Y_{\mathrm{P/S}}^{-1}\pi.$$

In this paper, Contois kinetics of the biomass growth [21] and Haldane kinetics [22] of the product formation are considered, which results in the following expressions for the μ and π:

$$\mu = \mu_{\max}\frac{C_{\mathrm{S}}}{K_{\mathrm{X}}C_{\mathrm{X}} + C_{\mathrm{S}}},\ \pi = \pi_{\max}\frac{C_{\mathrm{S}}}{K_{\mathrm{P}} + C_{\mathrm{S}} + C_{\mathrm{S}}^2/K_{\mathrm{I}}}, \tag{2}$$

where $\mu_{\max}$ (h^{-1}), $\pi_{\max}$ (h^{-1}) are the maximum specific growth and production rates, K_{X} (−) is the Contois saturation constant, K_{P} (gl^{-1}) is product formation saturation constant and K_{I} (gl^{-1}) is inhibition constant for product formation.

Input substrate concentration $C_{\mathrm{S,in}}$ (gl^{-1}) reflects the effect of the input flow u on the substrate concentration C_{S}. Finally, penicillin hydrolysis is modeled by a degradation

constant K_H (h^{-1}). At this point, more interested readers are referred to [19] and [12] where the model is described in more detail.

Now, let us assume that the cultivation volume V can not only be increased by exogenous input u, but a volume withdrawal can be performed as well. This requires a new input variable to be introduced and the volume differential equation changes into:

$$\frac{dV}{dt} = u_1 - V\lambda\left(e^{w\frac{T_{opt}-T_f}{T_b-T_f}} - 1\right) - u_2, \tag{3}$$

where u_1 corresponds with the old input variable, and u_2 (lh^{-1}) stands for volume withdrawal. The resulting scheme of the cultivation tank is shown in Fig. 1.

From the practical point of view, the introduction of the volume withdrawal brings several advantages—firstly, the engineer can control the tank volume and the tank overflow can be prevented. In industrial practice, this is often the main reason of introducing the effluent flow; however, the withdrawal is usually controlled ad hoc (if the volume reaches chosen level, certain part of the broth is withdrawn). The approach presented in this paper and described in detail in the following section takes the volume withdrawal directly into account and exploits it in favor of final product concentration maximization. Secondly, all the state variables are present in the withdrawn broth which can be exploited for state variables measurements (which in many cases of cultivations is performed manually) and cultivation analysis.

This little change of volume differential, however, effects the differential equations of other state variables as well. Let us remind that the state variables are in form of concentration which cannot be increased nor decreased by volume withdrawal. Therefore, terms including volume differential should be modified as follows: $dV/dt \rightarrow dV/dt + u_2$.

For the needs of optimization and to follow the conventional notation, let us rewrite the extended model (1) into the ordinary form using $x^T = [x_1, \ldots, x_4] = [V, C_X, C_S, C_P]$ and $u^T = [u_1, u_2]$:

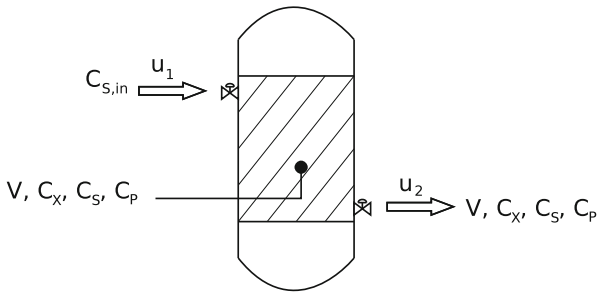

Fig. 1 Scheme of the cultivation tank

$$\begin{aligned}
\dot{x}_1 &= u_1 - \lambda(e^{w\frac{T_{opt}-T_f}{T_b-T_f}} - 1)x_1 - u_2,\\
\dot{x}_2 &= \left(\mu_{max}\frac{x_3}{K_X x_2 + x_3} - K_D\right)x_2\\
&\quad - \left(u_1 - \lambda\left(e^{w\frac{T_{opt}-T_f}{T_b-T_f}} - 1\right)x_1\right)\frac{x_2}{x_1},\\
\dot{x}_3 &= -\left(\frac{\mu_{max}}{Y_{X/S}}\frac{x_3}{K_X x_2 + x_3} + \frac{\pi_{max}}{Y_{P/S}}\frac{x_3}{K_P + x_3 + x_3^2/K_I}\right)x_2\\
&\quad + \frac{C_{S,in}u_1}{x_1} - \left(u_1 - \lambda\left(e^{w\frac{T_{opt}-T_f}{T_b-T_f}} - 1\right)x_1\right)\frac{x_3}{x_1},\\
\dot{x}_4 &= \pi_{max}\frac{x_3}{K_P + x_3 + x_3^2/K_I}x_2 - K_H x_4\\
&\quad - \left(u_1 - \lambda\left(e^{w\frac{T_{opt}-T_f}{T_b-T_f}} - 1\right)x_1\right)\frac{x_4}{x_1}.
\end{aligned} \tag{4}$$

For the better comprehension of the model, let us introduce a brief description of the model and an explanation of the phenomena typical for the penicillin cultivation.

The first differential equation describes the change of the volume profile. The increase of the volume happens due to the presence of the first input u_1 and it is decreased either applying the second input u_2 or in a very natural way due to the vaporization expressed by the vaporization term.

Like every other living organism, the biomass reproduces and dies—in the model (4), the increase of the biomass concentration is described by the reproduction term represented by biomass growth rate μ, while the decrease of biomass concentration follows from biomass death modeled by a constant death rate K_D. Except of these physiological ways of growth and decay, the biomass concentration is affected in a slightly "artificial" way—thanks to the vaporization, the biomass concentration increases while the volume increase (caused by the feed poured into the tank) leads to its decrease.

The "fuel" (essential nutrient) which is consumed by the "driving engine" of the whole bioprocess (biomass) is represented by the substrate concentration. It is crucial not only for keeping biomass alive but also for the product formation. Both of these consumption phenomena are described by the third state differential equation including the yield constants $Y_{X/S}$ and $Y_{P/S}$. Moreover, dual impact of input feed flow on substrate concentration can be observed—the input feed flow increases the substrate concentration (via qualitative constant of the feed $C_{S,in}$) and, on the other hand, the level of the substrate concentration in the broth is decreased due to dilution. Following from the concentration character of the third state variable, the vaporization term is included in its differential equation as well.

Product concentration (the most attractive variable from the industrial point of view) is increased at production rate π which, however, is completely different from the growth rate μ. This reflects different phases of the microorganisms

life—at certain phase, either the biomass reproduction or the product formation is preferred. Hydrolysis of the penicillin is modeled by constant term K_H, while the last two terms are related to the concentration nature of the fourth state variable.

The model (4) is adapted to the newly proposed constant vaporization strategy described in the next section, and it is further used in the optimization procedure as the representant of the controlled system behavior.

Optimal control design

In [16], an optimal feeding strategy coming out of a projected gradient method has been introduced. Theoretical complication given by the state dependence of the input saturation has been successfully addressed, and assumption on sufficiently large cultivation tank volume has been made. However, in industrial application, the cultivation tank may be filled up with a such large initial volume that application of the computed input feed flow rate leads to the tank overflow in short horizon. A perspective-offering solution to this problem has been tackled in the previous section. Here, we propose a way to operate the second input which can bring interesting results improvement.

Constant vaporization strategy

Applying another exogenous input u_2, one can avoid tank overflow, yet another problem occurs. A thoughtful reader has surely already noticed that having introduced two input variables, the first differential equation of mathematical description of the system does not comply with physical laws. It can be shown that at certain point the volume can reach zero value and further withdrawal can theoretically cause negative volume, which is physically impossible. One way of avoiding this is to set a dynamical constraint on the second input u_2 which ensures that at the point of zero volume V, the withdrawal does not exceed the inlet flow. However, looking at the issue from the engineering point of view, it is not either convenient to decrease the volume below certain too low value, as the final product amount equals to concentration C_P multiplied by the volume V.

Let us introduce an idea leading to a strategy solving the sketched negative volume difficulty. It consists in an assumption that the second input u_2 is used to compensate the effect of the first input u_1 on the volume V. From the first differential equation of model (4), it is obvious that the volume is affected by feed flow rate u_1, volume withdrawal rate u_2 and by natural vaporization described by the middle term. The key idea of the constant vaporization strategy is that we require the volume to be just naturally vaporizing without any other dynamical response to the exogenous signals—from the first differential equation of the model (4) which (in agreement with the requirement for constant vaporization of the broth) should be equal to vaporization term only, the second input u_2 can be calculated directly as $u_2 = u_1$, which results in $\dot{x}_1 = -K_{vap}x_1$ where K_{vap} is the overall vaporization constant, $K_{vap} = \lambda(\exp(w\frac{T_{opt}-T_f}{T_b-T_f}) - 1)$. Let us note that the sketched idea of double-input problem simplification is also described in [18] where except of quasi-double-input strategies, the true-double-input strategies performing complete optimization with both inlet and outlet flow inputs are described and compared.

Following the above mentioned strategy, the inlet flow stays the only optimization variable and the system description of the original process given by (4) changes into the following model with reduced input set:

$$\begin{aligned}
\dot{\xi}_1 &= -K_{vap}\xi_1,\\
\dot{\xi}_2 &= \left(\mu_{max}\frac{\xi_3}{K_X\xi_2+\xi_3} - K_D\right)\xi_2 - \left(\upsilon - K_{vap}\xi_1\right)\frac{\xi_2}{\xi_1},\\
\dot{\xi}_3 &= -\left(\frac{\mu_{max}}{Y_{X/S}}\frac{\xi_3}{K_X\xi_2+\xi_3} + \frac{\pi_{max}}{Y_{P/S}}\frac{\xi_3}{K_P+\xi_3+\xi_3^2/K_I}\right)\xi_2\\
&\quad + \frac{C_{S,in}\upsilon}{\xi_1} - \left(\upsilon - K_{vap}\xi_1\right)\frac{\xi_3}{\xi_1},\\
\dot{\xi}_4 &= \pi_{max}\frac{\xi_3}{K_P+\xi_3+\xi_3^2/K_I}\xi_2 - K_H\xi_4 - \left(\upsilon - K_{vap}\xi_1\right)\frac{\xi_4}{\xi_1}.
\end{aligned} \tag{5}$$

State vector ξ corresponds to the original state vector, $\xi = [\xi_1, \ldots, \xi_4]^T = [V, C_X, C_S, C_P]^T$ while υ represents inlet feed flow rate.

Model (5) supplemented by the corresponding model parameters (Table 1) provides an engineer with a tool to

Table 1 Model parameters

Parameter	Value
μ_{max}	0.11
π_{max}	0.004
K_P	0.1
$Y_{X/S}$	0.47
K_D	0.0136
K_X	0.06
K_H	0.01
$Y_{P/S}$	1.2
$C_{S,in}$	500
K_{vap}	6.23×10^{-4}
K_I	0.1
T_{opt}	298
T_f	273
T_b	373

design the optimal control minimizing a properly chosen criterion.

Optimization task formulation

From the optimization point of view, penicillin production optimization can be viewed as fixed initial state, free time interval and free final state issue. Without any loss of generality and due to upper cultivation duration constraint, let us now consider multiple optimization routines with fixed time intervals of length $t_{\text{end}} \in \{200, 300, 400, 500\}$. This helps us to simplify the optimization procedure and avoid difficulties with general time interval solution.

For the optimization purpose, the objective functional reflecting the optimization effort needs to be formulated mathematically. From the industrial point of view, two quite antagonistic goals can be chosen—both the quantity (represented by the amount of the product) and the quality of the final product (represented by its concentration in the cultivation broth) can be desired to be maximized. It can be intuitively seen that following only one of these optimization directions, two extremes are reached neither of which is preferable. Maximization of product amount (without any concentration check) can end up with an extremely large volume containing only a very low level of penicillin concentration, while maximization of concentration (without any volume limitation) can lead to a highly concentrated tiny-volumed broth. Without doubt, the quality of the final product is the factor affecting the duration of the product post-processing and subsequently also the efficiency of the whole industrial process crucially—the more concentrated the broth is, the shorter post-processing procedure is needed and, therefore (assuming a very common situation in the industrial practice with multiple cultivation tanks, but only a limited number of post-processing machines available), the cultivation can be repeated more frequently which can positively influence the overall productivity. Moreover, the maximization of the product concentration is very often directly requested in the industrial practice—if some product amount is guaranteed, the industrial companies are usually interested in obtaining the product of highest possible quality. Taking the high-quality-product requirements into account, concentration maximization (also considered in [12, 23–25] and many other papers) is chosen to be the preferred optimization criterion. However, it has been already mentioned that the absence of volume limitation can results in a small volume—these issues are in detail handled in [18] where in the true-double-input cases, the volume constraints are applied. In the case considered in this paper, the effect of vaporization phenomena is not critical enough to degrade the control performance and therefore, no volume limitations need to be considered.

As the main goal is to maximize the final product concentration, the following criterion in the Mayer form is formulated:

$$\mathcal{J} = -\xi_4(t_{\text{end}}), \tag{6}$$

where $\mathcal{J}$ denotes the criterion for the constant vaporization strategy.

Regarding state optimization constraints, it can be shown that the model (5) satisfies physical constraints (state variables nonnegativity) and no further attention is necessary to be paid to low state constraints. Moreover, constant vaporization strategy eliminates the need for upper volume constraint handling. Thus, input saturation constraint $0 \le \upsilon \le U_{\max}$ and input piecewise constant character $\mathrm{d}\upsilon/\mathrm{d}t = 0$ for $ml \le t < (m+1)l$, $m = 0, 1, \ldots$ (accomplished by sampling of the inputs υ with sampling period $l = 4$ h) are the only static constraints related to this optimization task.

Having properly defined the system equations, the input constraints and the objective functional, the optimization problem for $t \in [t_0, t_{\text{end}}]$ (without any loss of generality, let us consider $t_0 = 0$ h) can be summarized:

$$\upsilon^*(t) = \arg\min_{\upsilon(t)} \mathcal{J}(\xi(t)) \tag{7}$$

such that the following constraints hold:

$$\begin{aligned} \dot{\xi}(t) &= f(\xi(t), \upsilon(t)), \\ \xi(t_0) &= \xi_0, \\ 0 \le \upsilon(t) &\le U_{\max}. \end{aligned} \tag{8}$$

Here, $f(\xi(t), \upsilon(t))$ refers to the model (5). The values of ξ_0 and $U_{\max}$ are summarized in Table 2.

Nonlinear gradient method

This method belongs to the family of the optimal control methods [26]. For the problem stated by (7) and the constraints given in the form of (8), the optimal input υ^* is searched iteratively. First of all, the initial input vector υ_0 is estimated (in our case, zero vectors have been chosen). Then, the following procedure is applied:

$$\upsilon^*_{k+1} = \upsilon^*_k - \alpha \frac{\partial \mathcal{J}}{\partial \upsilon}, \tag{9}$$

Table 2 Optimization constraints

Parameter	Value
$\xi_{1,0}$	To be specified
$\xi_{2,0}$	1.5
$\xi_{3,0}$	6
$\xi_{4,0}$	0
$U_{\max}$	0.05

where $k = 0, 1, 2, \ldots$ is the number of the iteration, and α is the search step parameter whose choice is described later. Here it should be noted that direct calculation of $\partial\mathcal{J}/\partial\upsilon$ is quite complicated due to the fact that ξ_4 depends on υ via a differential equation. Therefore, let us rather introduce Hamiltonian $\mathcal{H}$ in this general form:

$$\mathcal{H} = L + p^{\mathrm{T}} f. \tag{10}$$

Here, L represents the integral penalty of the optimized criterion, f refers to the model (5), and p is the adjoint state vector solved back in time. As our criterion $\mathcal{J}$ does not contain the integral penalty, the Hamiltonian turns into the following form:

$$\mathcal{H} = p^{\mathrm{T}} f. \tag{11}$$

To compute the gradient of (6) with respect to $\upsilon(t)$, set first:

$$\begin{aligned} -\dot{p} &= \frac{\partial \mathcal{H}}{\partial \xi}, \\ \dot{\xi} &= \frac{\partial \mathcal{H}}{\partial p}, \\ \xi(t_0) &= \xi_0, \\ p(t_{\mathrm{end}}) &= -\left(\frac{\mathrm{d}\phi}{\mathrm{d}\xi}|_{t=t_{\mathrm{end}}}\right), \end{aligned} \tag{12}$$

where ϕ is the terminal term of the optimization criterion. In our case, $\phi = -\xi_4(t_{\mathrm{end}})$ from which it follows $p(t_{\mathrm{end}}) = [0, 0, 0, 1]^{\mathrm{T}}$. It can be shown (mathematically rigorous proof is beyond the scope of this paper) that $\partial\mathcal{J}/\partial\upsilon = -\partial\mathcal{H}/\partial\upsilon$—thus, gradient $\partial\mathcal{H}/\partial\upsilon$ can be used in iterative procedure (9), which changes into:

$$\upsilon^*_{k+1} = \upsilon^*_k + \alpha \frac{\partial \mathcal{H}}{\partial \upsilon}. \tag{13}$$

At this moment, a constant search step parameter has been chosen $\alpha = 0.002$. Examination of another step choices is provided in the following section. Input saturation constraint is handled by mapping the iterated input vectors υ_{k+1} on an admissible input sets $\Upsilon_{\mathrm{admiss}} = \{\upsilon, 0 \le \upsilon \le U_{\max}\}$ by a simple saturation. Requirement of piecewise constant nature of the input υ is satisfied by sampling with sampling period $l = 4$ h.

The iterative procedure described by (13) terminates at the moment when the improvement obtained at the $(k + 1)$-st iteration is less than a chosen tolerance compared to the k-th optimization iteration result.

Optimization results

In this section, results obtained by the constant vaporization strategy are presented and compared to those obtained by the original one-input gradient method optimization (CG) presented in [16] and constant volume strategy which instead of natural vaporization keeps the volume constant. The latter one is described in [17] and [18]. The optimization results have been simulated with the penicillin cultivation model in MATLAB environment.

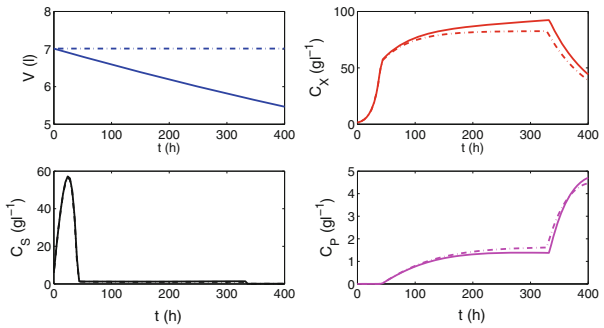

Fig. 2 Process variable profiles—comparison (*dashdot* CVol, *full* CVap)

Strategy results comparison

First, the constant vaporization (CVap) strategy has been tested on simulations with initial volume $V_0 = 7$ l and compared to the constant volume (CVol) strategy. Figure 2 shows very satisfactory cultivation results and reveals a slight superiority of the CVap strategy. It is due to the fact that the effect of input feed flow is inversely proportional to the actual amount of the broth in tank. While the effect of the input feed flow is always the same with the constant volume strategy, with the constant vaporization its positive influence improves as the volume decreases with time. Next, from the picture, it is obvious that the cultivation period that contributes to the final product concentration $C_{\mathrm{P}}(t_{\mathrm{end}})$ the most takes approximately the last 75 h. A rapid product concentration increase can be observed during this period; however, the biomass concentration decreases badly. This has a simple biological explanation—as can be seen from the characters of both μ and π (see Eq. 2), increasing one of them, the second one decreases, which corresponds to the fact that either the biomass population growth or the penicillin production is being preferred at the very same time.

Volume dependency

Next, the CVol and CVap strategies have been tested on multiple simulations with various initial volume V_0. Initial volume conditions have been chosen as linearly increasing, $V_0(1) = 7 + 3k$, $k \in \{0, 1, \ldots, 10\}$.

Looking at the Fig. 3, it can be seen that with increasing V_0, final product concentration $C_{\mathrm{P}}(t_{\mathrm{end}})$ decreases for both the strategies which can be (once again) explained as the

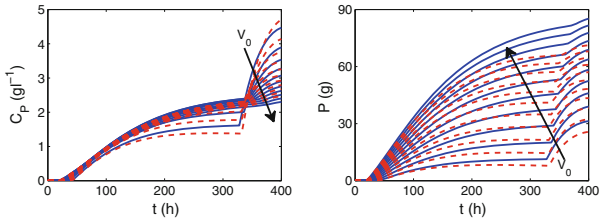

Fig. 3 Volume-dependency of C_P and P profiles (*blue full* CVol, *red dashed* CVap)

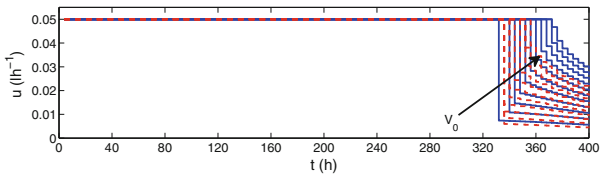

Fig. 4 Volume-dependency of input profiles (*blue full* CVol, *red dashed* CVap)

consequence of the inverse proportional effect of the actual initial volume. On the other hand, the total amount of product P increases with initial volume V_0 increase. This is due to the fact that the total product amount P is proportional not only to product concentration C_P but also to broth volume, $P = C_P V$. From this point of view, the constant volume strategy is able to obtain better results as the volume is held constant—with the constant vaporization, the volume decreases steadily and, thus, the total amount of product at t_{end} is lower than with the CVol strategy. Nevertheless, comparing the quality of the cultivation in the sense of the product concentration, CVap strategy is the more succesful candidate.

Yet, another interesting tendency is to be observed from Fig. 4—it is a convergency of input profiles to a high-saturation-valued vector with V_0 increase. From technological point of view, this is caused by the increase of V_0/U_{max} ratio—the higher the volume is, the more feed is needed to keep the whole system developing and the higher the V_0/U_{max} ratio is, the longer a high-saturated input must be applied.

To inspect the effect of various V_0 in more details, another set of simulations has been performed; however, with a constant ratio $V_0/U_{max} = 7/0.05$. The initial volume V_0 has been set linearly growing as in the previous simulation set.

Figure 5 shows that holding the V_0/U_{max} ratio fixed, C_P profiles aggravation (namely the final product concentration decay) for CVol strategy is not as drastic as in the previous case and, moreover, the C_P profile for CVap strategy does not change at all. However, this is to be expected as with fixed V_0/U_{max} ratio and the same initial concentrations, the system parameters does not change at

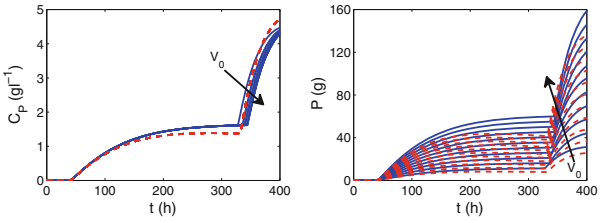

Fig. 5 Volume-dependency of C_P and P profiles, fixed V_0/U_{max} ratio (*blue full* CVol, *red dashed* CVap)

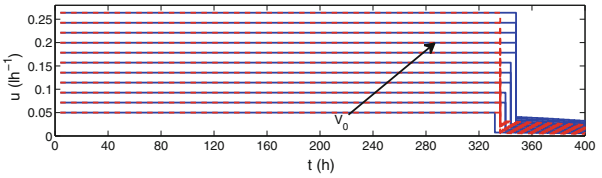

Fig. 6 Volume-dependency of input profile, fixed V_0/U_{max} ratio (*blue full* CVol, *red dashed* CVap)

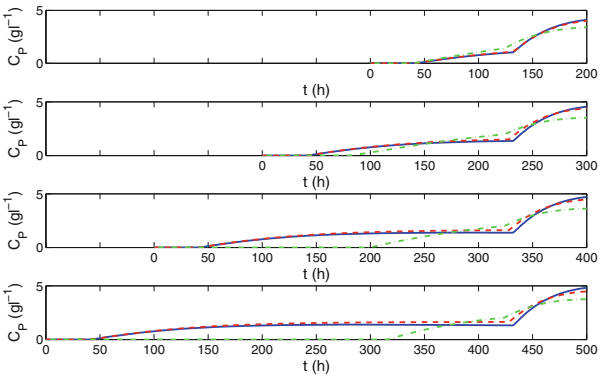

Fig. 7 C_P profiles for various cultivation lengths, comparison (*green dashdot* CG, *red dashed* CVol, *blue full* CVap

all and the system with higher V_0 is an exact scale-up of the lower V_0 one. The scale-up claim is supported by the Fig. 6, where input profiles for various V_0 are shown and it is obvious that the dynamical character of the CVap input profile remains the same and the vectors are multiplied by the V_0/U_{max} ratio.

Cultivation length dependency

As has already been mentioned, cultivation length is considered to be constant, yet it can be chosen from a set {200,300,400,500} h. Figure 7 compares cultivation with classical gradient method (presented in [16]) to the CVol and CVap strategy, respectively. For every chosen cultivation length, it is obvious that the CVap strategy achieves better results than the other ones and the product concentrations at the final time $C_P(t_{end})$ are higher.

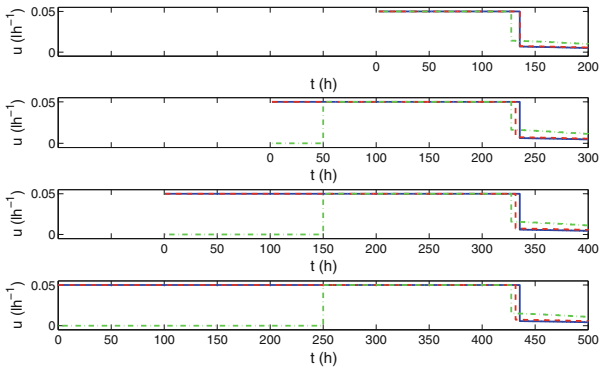

Fig. 8 Input profiles for various cultivation lengths, comparison (*green dashdot* CG, *red dashed* CVol, *blue full* CVap)

Looking at the Fig. 8, convergence of input profiles to a certain "superprofile" can be seen. Similar kind of convergence has already been mentioned in [15] as well. However, although the profiles are stable backward in time, they do not settle down at the same value (here, we assume settling down in negative march of time). In negative time, CG method settles down on a zero value, while CVol and CVap strategies obtained the input profiles settle down on upper saturation. The fact that volume is held constant (constantly decreasing, respectively) by the second virtual input (virtual due to the fact that it is not considered in optimization) and it cannot dynamically aggravate the product concentration profile enables to deliver more feed into the cultivation tank without negative effect of volume increase and, thus, with better fed biomass population, the product concentrations obtained at the end of the cultivation are higher.

Choice of gradient search step

In the previous sections, constant search step α has been assumed. Although gradient methods are able to find the (closest local) optimum, in industrial practice the quality of the solution (product concentration at final time) is of similar importance level as the promptness of the optimization algorithm (which is nothing but the convergence property). Convergence speed of the gradient method is directly related to the choice of the step with which the descent is performed—in this section, various families of gradient search steps with the effort to find the best one are investigated and evaluated. Let us remark that this evaluation can be performed and generalized for other strategies as well.

I *Fixed step family (FSF)*—the first and the simplest family contains search steps α which are constant over the whole duration of optimization. These steps are chosen as $\alpha_q = q \times 10^{-4}$, $q \in \{1, 2, \ldots, 10\}$. Fixed search step family members are then denoted as Fα_q, e.g., F1e-3 stands for fixed step approach with $\alpha = 1 \times 10^{-3}$.

II *Parabola-minimizing family (PMF)*—this family unifies approaches looking for step α_k as a minimum of parabola. The main idea is that (having computed the gradient $\partial\mathcal{H}_k/\partial v$ at the k-th iteration of the procedure described by (13)) the value of optimization criterion at this iteration $\mathcal{J}_k$ is assumed to be a quadratic function of the step α,

$$\mathcal{J}_k(\alpha) = K_{2,k}\alpha^2 + K_{1,k}\alpha + K_{0,k}. \tag{14}$$

Under this assumption, the minimum of this parabola can be found analytically choosing three different steps $[\alpha_a, \alpha_b, \alpha_c]$, computing the corresponding values $[\mathcal{J}_k|_{\alpha=\alpha_a}, \mathcal{J}_k|_{\alpha=\alpha_b}, \mathcal{J}_k|_{\alpha=\alpha_c}]$ for input vectors $v_{k-1} + \alpha_a\partial\mathcal{H}_k/\partial v$, $v_{k-1} + \alpha_b\partial\mathcal{H}_k/\partial v$ and $v_{k-1} + \alpha_c\partial\mathcal{H}_k/\partial v$ and determining coefficients $K_{0,k}$, $K_{1,k}$, $K_{2,k}$. As the parabolic approximation of the criterion $\mathcal{J}_k$ might be inaccurate in certain cases, we consider a set of triplets $[\alpha_a, \alpha_b, \alpha_c]$ at which the criterion is evaluated as follows:

$$\begin{aligned}&[\alpha_a, \alpha_b, \alpha_c] = [0 \times \alpha_{bas}, 1 \times \alpha_{bas}, 2 \times \alpha_{bas}],\\ &\alpha_{bas} = \frac{q}{2} \times 10^{-4}, q \in \{1, 2, \ldots, 8\}.\end{aligned} \tag{15}$$

Let us note that with $\alpha_a = 0$, $\mathcal{J}_k|_{\alpha=\alpha_a} = \mathcal{J}_{k-1}$. With this choice, one third of computational effort can be spared. The search step α_k which is then applied at the k-th iteration is computed as

$$\alpha_k = \arg\min(K_{2,k}\alpha^2 + K_{1,k}\alpha + K_{0,k}). \tag{16}$$

From now on, members of parabola-minimizing step family are denoted as Pmq, e.g., Pm7 represents approach where $\alpha_{bas} = \frac{7}{2} \times 10^{-4}$.

III *General-curve-minimizing family (GCMF)*—approaches performing exhaustive line search are grouped in this family. Two different sub-branches are explored: (1) a sub-branch considering linear distances between the search steps, (2) and the one performing brute-force search on logarithmically spaced vector of steps. The members of the first sub-branch are denoted as GmLin$s$ with s being the spacing of the linearly increasing vector of the examined search steps, $\alpha_{k,j} = j \times s$, $j \in \{1, 2, \ldots, j_{\max}\}$, $0 < \alpha_{k,j} \leq 1$. Here, $j_{\max}$ is the value of $1/s$ truncated to zero decimal digits,

$$j_{\max} = trunc(1/s, 0) = \lfloor 1/s \rfloor. \tag{17}$$

In most cases, a quasi-convexity of $\mathcal{J}$ as a function of α is assumed, it means that if for certain $\alpha_{k,j}$ from the explored vector $\mathcal{J}_{k,j} > \mathcal{J}_{k,j-1}$, the line search at that α-search

iteration is terminated and the applicable step for the gradient-optimization iteration is chosen as $\alpha_k = \alpha_{k,j-1}$. Such approaches are then denoted GmLin*s*QC to remark the quasi-convexity assumption, e.g., GmLin5e-4QC performs exact line search through a vector of 2,000 steps α with linear spacing $s = 5 \times 10^{-4}$ under assumption that $\mathcal{J}(\alpha)$ is quasi-convex. The second sub-branch tries to reduce the number of searched steps and it involves approaches searching through logarithmically spaced vector of steps α. Regarding this sub-branch, two approaches are considered: GmLog13 exploits vector of 13 values from 1×10^{-4} to 1 with logarithmic spacing, while GmLog13QC takes the same vector into consideration and adds quasi-convexity assumption.

Results with enhanced search step choice

In this section, results of the three-step choice families described in the previous section are presented.

Table 3 brings comparison of optimality $\mathcal{J}(\mathrm{gl}^{-1})$, number of iterations-to-converge ItC (−) and time-to-converge TtC (min) for the inspected families and their member approaches. "NA" value means that the convergence has not been achieved.

Table 3 Results and computational demands comparison

Family	Member	$\mathcal{J}$	ItC	TtC
FSF	F1e-4	4.71	48,287	707
	F2e-4	4.71	24,137	336
	F3e-4	4.71	16,092	224
	F4e-4	4.71	12,070	167
	F5e-4	4.71	9,651	133
	F6e-4	4.71	8,040	110
	F7e-4	4.71	6,890	94
	F8e-4	NA	NA	NA
	F9e-4	NA	NA	NA
	F1e-3	NA	NA	NA
PMF	Pm1	4.70	275	7
	Pm2	4.69	366	10
	Pm3	4.70	428	11
	Pm4	4.70	453	12
	Pm5	4.70	470	12
	Pm6	4.70	726	19
	Pm7	NA	NA	NA
	Pm8	NA	NA	NA
GCMF	GmLin2e-4QC	4.74	886	121
	GmLin3e-4QC	4.76	2,055	108
	GmLin4e-4QC	4.76	1,187	73
	GmLin5e-4QC	NA	NA	NA
	GmLin4e-4	4.76	1,016	16,185
	GmLog13QC	4.76	1,410	52
	GmLog13	4.76	1,297	124

Regarding the FSF, it can be seen that with increase of the search step, ItC value decreases and so does TtC. This can be expected as with greater steps α, one can await faster convergence rate as the gradient method moves quicker towards the supposed minimum. A situation which often occurs when using gradient method can be observed for steps $\alpha \geq 8 \times 10^{-4}$—from this value, very large search steps destroy the convergence properties of gradient search which is known to be susceptible to the oversized step choice. Here it could be noted that fixed step family is the most computationally demanding from the three search steps families with sovereignly highest ItC values. This is the price to be paid for the fact that the gradient search with sufficiently small search step guarantees convergence to the closest local minimum. Its local-minimum-convergence is the next disadvantage—as can be seen later, this can be overcome exploring larger part of step-space.

On the other hand, the PMF converges usually extremely fast compared to the other families. However, with increasing distance between the examined α_a, α_b and α_c, the approximation is less and less accurate which reflects in ItC increase and for very large α_{bas}, the converge is not reached.

Looking at the $\mathcal{J}$-column, GCMF achieves the most superior results. Also this family demonstrates that larger steps bring faster convergence to a small environment of the minimum but with very large search steps, convergence is not guaranteed. An interesting insight offers comparison of GmLin4e-4QC and GmLin4e-4 approaches—as they both achieve the same value of the final product concentration, the quasi-convexity assumption is proved right. Yet, the difference between TtC values is enormous. Although GmLin4e-4 spares approximately 150 iterations, it is clear that the most of the α-search iterations performed at every optimization iteration k are redundant. The same is supported by the approaches using logarithmically-spaced α-vectors. Moreover (as is shown in Fig. 9, where the steps α_k which are finally applied at particular iteration k are depicted), the steps which are most attractive from the convergence point of view are quite small and large steps

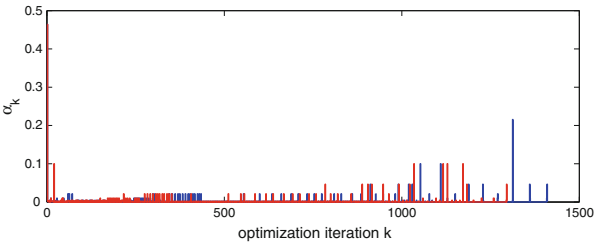

Fig. 9 α_k evolution (*blue* GmLog13QC, *red* GmLog13)

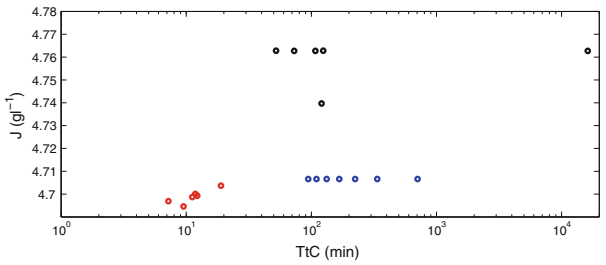

Fig. 10 Step choice families comparison (*blue* FSF, *red* PMF, *black* GCMF)

α_k occur rather rarely. This is the consequence of nonlinear dynamics of the system which turns the optimization task into a nonconvex one and the υ-gradient of the Hamiltonian $\partial\mathcal{H}/\partial\upsilon$ which is used in the best direction search is usually only locally valid. Therefore, it is more convenient to distribute the examined search steps with higher intensity tightly around the current point in the explored optimization variables space while including a few outliers, the convergence gets faster.

Comparison of $\mathcal{J}$ and TtC for the successful members of particular step choice families can be seen in Fig. 10.

Conclusion

This paper follows the patterns suggested in previous publications of the authors—a model of the controlled system involving the second input variable is derived, a neat way of problem reduction to a single-input optimization is performed and thanks to this, a successful constant vaporization control strategy is introduced. The necessary model adaptation is performed so that it comports with the strategy requirements. Results are verified on a set of numerical experiments and discussed in detail.

The comparison obtained by verifying the constant vaporization strategy on a set of numerical simulations and confronting it with the previously introduced methods can be summarized as very encouraging—CVap strategy achieves better results which suggest its possible industrial use. A "superprofile" convergence (observed in earlier publications) occurs in this case as well, and this supports the claim that it is a property of the optimization issues where the fixed time, fixed initial condition and free terminal condition are considered.

Next, various ways of gradient search step choice are suggested, explained and compared. The most time-sparing group of approaches appears to be the PMF which approximates the cost criterion by a parabola while regarding the final product concentration value, the most successful is the GCMF. As a tradeoff between optimality and time consumption, logarithmically spaced vector of search steps can be chosen—it combines fast convergence rate and higher obtained final product concentration and adding the quasi-convexity assumption, it seems to be the fair choice for further utilization within the optimization routine.

Acknowledgment This research has been supported by the Czech Science Foundation through the grant no. 13-20433S.

References

1. Lee S (1996) High cell-density culture of *Escherichia coli*. Trends biotechnol 14(3):98–105
2. Gregory M, Turner C (1993) Open-loop control of specific growth rate in fed-batch cultures of recombinant *E. coli*. Biotechnol Techn 7(12):889–894
3. Suzuki T, Yamane T, Shimizu S (1990) Phenomenological background and some preliminary trials of automated substrate supply in pH-stat modal fed-batch culture using a setpoint of high limit. J Ferment Bioengineer 69(5):292–297
4. Siimes T, Linko P, von Numers C, Nakajima M, Endo I (1995) Real-time fuzzy-knowledge-based control of Baker's yeast production. Biotechnol bioengineer 45(2):135–143
5. Horiuchi J (2002) Fuzzy modeling and control of biological processes. J biosci bioengineer 94(6):574–578
6. Zhu G, Zamamiri A, Henson M, Hjortsø M (2000) Model predictive control of continuous yeast bioreactors using cell population balance models. Chem Engineer Sci 55(24):6155–6167
7. Azimzadeh F, Galan O, Romagnoli J (2001) On-line optimal trajectory control for a fermentation process using multi-linear models. Comput Chem Engineer 25(1):15–26
8. Alford J (2006) Bioprocess control: Advances and challenges. Comput Chem Engineer 30(10):1464–1475
9. Nagy Z (2007) Model based control of a yeast fermentation bioreactor using optimally designed artificial neural networks. Chem Engineer J 127(1):95–109
10. Alvarez M, Stocks S, Jørgensen S (2009) Bioprocess modelling for learning model predictive control (l-mpc). In: Computational intelligence techniques for bioprocess modelling, supervision and control, pp 237–280
11. Zhang H, Lennox B (2004) Integrated condition monitoring and control of fed-batch fermentation processes. J Process Contr 14(1): 41–50
12. Ashoori A, Moshiri B, Khaki-Sedigh A, Bakhtiari M (2009) Optimal control of a nonlinear fed-batch fermentation process using model predictive approach. J Process Contr 19(7):1162–1173
13. Dai Y, Yuan Y (2000) A nonlinear conjugate gradient method with a strong global convergence property. SIAM J Optim 10(1):177–182
14. Birgin E, Martínez J (2001) A spectral conjugate gradient method for unconstrained optimization. Appl Math optim 43(2):117–128
15. Čelikovský S, Papáček Š, Cervantes-Herrera A, Ruiz-Leon J (2010) Singular perturbation based solution to optimal microalgal growth problem and its infinite time horizon analysis. IEEE Trans Automat Contr 55(3):767–772
16. Pčolka M, Čelikovský S (2012) Gradient method optimization of penicillin production. In: 24th Chinese Control and Decision Conference (CCDC), 2012. IEEE, pp 74–79
17. Pčolka M, Čelikovský S (2012) Gradient method optimization of penicillin production: New strategies. In: 20th Mediterranean Conference on Control & Automation (MED), 2012. IEEE, pp 1235–1240
18. Pčolka M, Čelikovský S (2012) Multiple-Input Cultivation Model Based Optimization of Penicillin Production. In: 51st IEEE Conference on Decision and Control, Hawaii, USA

19. Van Impe J, Bastin G (2002) Optimal adaptive control of fed-batch fermentation processes with multiple substrates. In: Control Applications, 1993., Second IEEE Conference on. IEEE, pp 469–474
20. Constantinides A, Spencer J, Gaden Jr E (1970) Optimization of batch fermentation processes. II. Optimum temperature profiles for batch penicillin fermentations. Biotechnol Bioengineer 12(6): 1081–1098
21. Contois D (1959) Kinetics of bacterial growth: relationship between population density and specific growth rate of continuous cultures. Microbiology 21(1):40
22. Briggs G, Haldane J (1925) A note on the kinetics of enzyme action. Biochem j 19(2):338
23. Álvarez L, García J, Urrego D Control of a fedbatch bioprocess using nonlinear model predictive control.
24. Azimzadeh F, Galán O, Barford J, Romagnoli J (1999) On-line optimization control for a time-varying process using multiple models: A laboratory scale fermenter application. Comput Chem Engineer 23:S235–S239
25. Foss B, Johansen T, Sørensen A (1995) Nonlinear predictive control using local models–applied to a batch fermentation process. Contr Engineer Practice 3(3):389–396
26. Bryson A, Ho Y (1969) Applied optimal control. Blaisdell, New York

Chapter 4

Optimization Horizon

This chapter addresses the choice of optimization horizon and its influence on several aspects of optimal control tasks involving nonlinear systems; namely, the following cases are considered: *i)* choice of optimization horizon with respect to *feedback introduction*; *ii)* horizon optimization aiming at *efficiency maximization*; and *iii)* choice of optimization horizon *reducing computational/memory demands*. Each of these cases is introduced and studied separately in the following subchapters.

4.1 Feedbackization

In this part, the choice of optimization horizon in combination with feedback introduction is addressed. A standard widespread control engineering practice [Kwon and Han, 2006], [Mattingley et al., 2011], [Allgöwer and Zheng, 2012] is to follow the *receding horizon principle*[1], i.e. gather output measurements and disturbance predictions, solve the optimization task over the optimization horizon, apply the first input sample to the controlled system, wait for the new measurements/predictions, shift the optimization horizon by the period that has elapsed since the previous data arrival and repeat the whole procedure again. Here, it is important to remark that the optimization horizon is chosen a priori and remains fixed over the whole "lifetime" of the controller. Dealing with systems operated over an infinite horizon where the optimal control task is not related to any particular time instant, this common choice is appropriate, however, in case of batch-operated systems—e.g. chemical processes or bioprocesses—where the optimal control task focuses

[1] In fact, *receding horizon control* is often used as a synonym of *model predictive control* or *applied optimal control*.

on a specific moment (usually the end of operating period), the receding horizon principle does not correspond with the actual control goal.

Overcoming this issue, [A.11] introduces a control scheme employing a *shrinking horizon*, where—unlike the receding horizon paradigm—the optimization horizon shrinks by the time interval that has elapsed since the last data arrival and thus, the terminal part of the optimization criterion always corresponds to the very same actual time instant. Considering penicillin production maximization task, detailed evaluations and comparisons of receding horizon optimal control (RHOC) and shrinking horizon optimal control (SHOC) show that for a broad range of numbers of optimized input samples over the chosen optimization period (i.e. for a variety of sampling periods), SHOC yields almost the same results while using RHOC, decrease of the number of optimized input samples usually goes hand in hand with decrease of the achieved optimization criterion value. Next, influence of gradual tightening of the computational restrictions was inspected as well. While SHOC is almost immune against lowering the number of the iterations allowed, this is definitely not the case of the commonly used RHOC for which the results differ significantly for various restrictions and in certain cases, the more limited variants achieve paradoxically far better results than those with looser restraints. In addition, [A.11] provides also a hybrid approach offering a trade-off between the control performance and memory demands.

The publication mentioned above is presented in the original formatting and is provided on the next page et seq.

2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

Enhancement of Practical Applicability of Optimal Control of a Nonlinear Process

Matej Pčolka[1] and Sergej Čelikovský[1,2]

Abstract— **This paper makes a step towards practical applicability of the optimal control for industrial penicillin production. Using the nonlinear gradient method as the key optimization tool, two ways of measurement feedback incorporation into the optimization procedure are proposed. Firstly, the receding horizon approach (whose linear variant is widely spreading in the field of operation of various industrial processes) is investigated considering different lengths of optimization horizon. Secondly, the shrinking horizon approach inspired by the character of the solved task with terminal criterion is examined. In order to make the latter comparable to the receding horizon approach, various sampling periods of the input signal are considered. Utilization of the nonlinear continuous time model of the controlled process clearly distinguishes this paper from the earlier publications. The behavior of both approaches is tested on a set of numerical experiments with the focus on performance under constrained computational resources. The obtained results demonstrate the superiority of shrinking horizon approach and its strong computational restriction resistance.**

I. INTRODUCTION

Shortly after the discovery of the penicillin in 1930s, the discoveries of the next members of antibiotics family have begun to appear with an increasing frequency and the production of various types of antibiotics have experienced unprecedented boom. Over more than 80 years, technology of antibiotics production has undergone rapid development during which the control approaches have changed from rather obscure heuristic thumb-sucking methods to serious attempts based on solid fundamentals of control theory. The earliest open-loop strategies [1] have been gradually replaced by closed-loop approaches exploiting either on/off [2] or well-known PID schemes. Fuzzy approaches explored intensively in the '90s [3] have brought welcomed automated alternative to the operator-performed feeding based on empirical experiences and they lived to see a resurrection at the beginning of the new millennium [4]. Yet, one of the most impressive results have been obtained by advanced control strategy called Model Predictive Control (MPC) [5]. It exploits mathematical model of the controlled system and performs optimization trying to minimize chosen criterion while satisfying known constraints. The main drawback of the classical MPC is the condition of discrete linear model of the controlled system. Being used to control essentially nonlinear process (and the dynamics which appears during antibiotics production *is* undoubtedly strongly nonlinear), the model can be obtained by one of the two linearization ways. The simpler one, approximate linearization, is performed at certain operating point - however, its validity can be guaranteed only in a small region of the linearization point. In order to improve the usability of this method, multiple-step approximate linearization should be performed. However, besides the fact that it can be prohibitively time consuming, neither the optimality nor the stability of the controller using such set of models can be proved. The more sophisticated linearization method, exact linearization, avoids these problems - unfortunately, in the field of bioprocesses, the exactly linearizable processes are rather rare and occasional. Therefore, in order to fulfill the optimality requirement, proper nonlinear optimization method should be used. Based on successful applications referred in [6]–[8], nonlinear gradient method as a well-developed optimization method has been chosen for this paper. Trying to get near to real industrial utilization, some way of the "feedbackization" needs to be found. In this paper, we propose two such ways. First one (Receding Horizon Optimal Control) is inspired by the way how the feedback incorporation into the optimization routine is ensured in the case of classical MPC. The second one (Shrinking Horizon Optimal Control) takes advantage of the fact that the antibiotics production is a fixed-time task and performs optimization over a horizon which is gradually shrunk by the sampling period of the input discretization. In order to compare the performance of the proposed methods, a set of horizon lengths have been tried in the case of Receding Horizon Optimal Control while various sampling periods have been examined in the case of the case of Shrinking Horizon Optimal Control. To test the practical applicability of the proposed methods, both of them have been computed under gradually tightening computational resources constraints. The obtained results suggest that the Shrinking Horizon approach can provide an engineer with a better and stronger tool in industrial optimization than the Receding Horizon approach as it is much more resistant to the computational resources cut-down and achieves far better final product concentrations.

The paper is organized as follows: in Section II, model of the cultivation of penicillin G is introduced and briefly explained. Section III brings optimization routine description (in this role, nonlinear gradient method is used). Two "feedbackization" approaches are proposed and discussed. In Section IV, results of the numerical experiments for both methods are presented and summarized. Receding Horizon

[1]Department of Control Engineering (DCE), Faculty of Electrical Engineering (FEE) of Czech Technical University (CTU) in Prague, Technická 2, 166 27 Praha 6, Czech Republic

[2]Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8, Czech Republic

978-1-4799-0178-4/$31.00 ©2013 AACC

strategy is tested with various lengths of the optimization horizon while the effect of different sampling on results of Shrinking Horizon approach is examined. Practical usability is discussed coming out of another set of numerical experiments with gradual increase of severity of computational restrictions. Section V summarizes the work, sketches the future work and concludes the paper.

II. PENICILLIN CULTIVATION MODEL

Cultivation of the penicillin G [9], [5] is considered in this paper. For this kind of cultivation, several phenomena are fundamental.

Biomass concentration C_X represents the living microorganisms which consume essential nutrient enabling both the biomass reproduction (at growth rate μ) and product formation (happening at specific product rate π). Except of biomass reproduction, the concentration of biomass is affected by a constant death rate K_D. Concentration of essential nutrient C_S (here, carbon is considered) can be complemented pouring the feed at certain feed flow rate into the tank - this is also the way how the volume level V is increased while its natural decrease is caused by the vaporization phenomena. Last important phenomena is the hydrolysis degradation of penicillin (penicillin concentration is usually denoted as C_P) taking place at constant rate K_H. All the mentioned "concentration variables" (C_X, C_S and C_P) are influenced by the dilution phenomena (the volume increase automatically decreases the concentration of certain variable).

The dynamical behavior of this bioprocess can be summarized into a compact set of ordinary differential equations as follows:

$$\begin{aligned}
\dot{x}_1 &= u - K_{vap}x_1,\\
\dot{x}_2 &= \left(\mu_{max}\frac{x_3}{K_X x_2 + x_3} - K_D\right)x_2 - \left(\frac{u}{x_1} - K_{vap}\right)x_2,\\
\dot{x}_3 &= -\left(\frac{\mu_{max}}{Y_{X/S}}\frac{x_3}{K_X x_2 + x_3} + \frac{\pi_{max}}{Y_{P/S}}\frac{x_3}{K_P + x_3 + x_3^2/K_I}\right)x_2 + \frac{C_{S,in}u}{x_1} - \left(\frac{u}{x_1} - K_{vap}\right)x_3,\\
\dot{x}_4 &= \pi_{max}\frac{x_3}{K_P + x_3 + x_3^2/K_I}x_2 - K_H x_4 - \left(\frac{u}{x_1} - K_{vap}\right)x_4.
\end{aligned} \tag{1}$$

In this description, states $x = [x_1, x_2, x_3, x_4]^{\mathrm{T}}$ correspond to $[V, C_X, C_S, C_P]$ and input u represents feed flow rate. Model (1) is further exploited as both the optimization model and simulation test-bed for the results evaluation.

Values of the system parameters can be found in Table I. Interested readers looking for a more detailed description are referred to [5], [6], [9].

TABLE I
MODEL PARAMETERS.

Parameter	Value	Parameter	Value
μ_{max}	0.11	$Y_{P/S}$	1.2
π_{max}	0.004	$C_{S,in}$	500
K_P	0.1	K_{vap}	6.23×10^{-4}
$Y_{X/S}$	0.47	K_I	0.1
K_D	0.0136	K_X	0.06
K_H	0.01		

III. "FEEDBACKIZATION" OF THE OPTIMAL CONTROL

In this paper, we come out of the optimization task introduced in [6] - however, the idea is considerably extended by the "feedbackization" which is explained later in this section.

A. Problem formulation

The usual objective of industrial production of penicillin is to obtain as high concentration of the penicillin at the end of the cultivation as possible. Although it can be expected that increasing the cultivation period, the result shall improve, the cultivation length cannot exceed certain maximal value - thus, without loss of objectivity, from now on we will consider that the cultivation length is fixed, $T = 400\,\mathrm{h}$.

In industrial factories, the value of the input feed flow is often operated by a human operator. In order to consider input profile $u(t)$ applicable, it shall be piecewise constant with shortest acceptable sampling period $l_{\mathrm{low}} = 4\,\mathrm{h}$.

Making aforementioned assumptions and taking all the technical requirements into consideration, we can formulate the optimization task as minimization of criterion $\mathcal{J}$,

$$\mathcal{J} = -x_4(T), \tag{2}$$

while satisfying the following constraints:

$$\begin{aligned}
&\dot{x} = f(x,u), \quad x(0) = x_0 = (x_{1,0}, \ldots, x_{4,0})^{\mathrm{T}},\\
&0 \le u \le u_{max},\\
&\dot{u} = 0 \quad \text{for} \quad ml \le t < (m+1)l,\ m = 0, 1, \ldots.
\end{aligned} \tag{3}$$

Here, $f(x,u)$ represents model (1). Optimization constraints and initial values of the states are listed in Tab. II.

B. Optimization background

As the candidate for the optimization routine, nonlinear gradient method belonging to optimal control methods family [10] has been chosen. The key idea of the gradient method is to find the optimal input u^* using its initial estimate u_0^* iteratively as follows:

$$u_{k+1}^* = u_k^* + \alpha\frac{\partial \mathcal{H}}{\partial u} \tag{4}$$

with constant search step $\alpha = 0.001$. If at any iteration i such a situation occurs that $\mathcal{J}_i = \mathcal{J}_{i-1}$, the procedure is terminated. Hamiltonian $\mathcal{H}$ whose partial derivative with respect to u occurs in Eq. (4) is constructed as

$$\mathcal{H} = L + p^{\mathrm{T}} f, \tag{5}$$

where L is the integral term of the minimization criterion $\mathcal{J}$ (in our case, $L = 0$), f refers to model (1) and p is the adjoint state vector with dynamics

$$-\dot{p} = \frac{\partial \mathcal{H}}{\partial x}, \quad p(T) = -\left(\frac{\mathrm{d}\phi}{\mathrm{dx}}|_{t=T}\right). \tag{6}$$

Here, ϕ is the terminal term of the criterion $\mathcal{J}$. The optimization task then turns into minimization of criterion (5) with combined constraints (3) and (6).

Convergence properties of the used method are discussed in [11] while more details on the used method can be found in [6].

C. Receding horizon approach

The first way to incorporate the feedback into the optimization routine is inspired by the idea of the well-known classical linear MPC - it is the Receding Horizon Optimal Control (RHOC) approach. Although there is a big amount of literature discussing the properties of the receding horizon approach, basically they address either linear discrete [12], [13], nonlinear discrete [14], nonlinear linearized [15] or nonlinear discretized models of the controlled system [16]. In this paper, we consider continuous nonlinear model of the system which strongly distinguishes the current paper from those above mentioned.

As the idea is quite straightforward, we provide only a brief description. Algorithmically, the RHOC approach can be formulated as follows:

1) measure current values of the states, denote them x_0;
2) find u^* minimizing criterion $\mathcal{J} = -x_4(T_P)$, $T_P < T$, using mathematical model (1) with the initial conditions x_0 and with respect to constraints given by (3);
3) apply the first l hours of u^* and shift to new time $t_{new} = t_{old} + l$;
4) if the shifted time $t_{new} < T$, go to 1), else finish.

Here, T_P denotes optimization horizon which remains constant for each sub-run of the RHOC procedure. Such an approach incorporates the current measurements of states x into the optimization routine - it can be simply shown that at every iteration of this algorithm, the optimized input u^* is a function of current measurement x_0. As the optimization tool, nonlinear gradient method described in the previous subsection is exploited. Stability and convergence properties of this approach can be found in [12]–[14] and others. Let us emphasize, that in contrast with the up-to-date literature, here we consider continuous nonlinear model of the cultivation which inherently *is* continuous and nonlinear. In such case, discretization can (except of discretization error introduction) make the state description cumbersome and blur the phenomena affecting its dynamics while linearization can aggravate the optimality of the controller performance.

TABLE II
OPTIMIZATION CONSTRAINTS.

Parameter	Value
$x_{1,0}$	7.012
$x_{2,0}$	1.5
$x_{3,0}$	6
$x_{4,0}$	0
u_{max}	0.05

D. Shrinking horizon approach

Motivation for the Shrinking Horizon Optimal Control (SHOC) approach is the fact that the penicillin production optimization can be formulated as fixed time task with terminal criterion only. Under such conditions, the RHOC approach can be modified in the following way:

1) set $T_P = T$;
2) measure current values of the states, denote them x_0;
3) find u^* minimizing criterion $\mathcal{J} = -x_4(T_P)$ using mathematical model (1) with the initial conditions x_0 and with respect to constraints given by (3);
4) apply the first l hours of u^*, shift to new time $t_{new} = t_{old} + l$ and decrease the optimization horizon T_P to $T_{P,new} = T_{P,old} - l$;
5) if the shifted time $t_{new} < T$, go to 2), else finish.

The key difference between the SHOC and the RHOC approaches is that while in the case of the RHOC approach, the criterion minimization is performed over constant horizon T_P which is being shifted gradually, the SHOC approach in fact always minimizes the criterion with respect to the end of the cultivation T. In other words, the optimization performed at time $t = 0\,\mathrm{h}$ considers T-hours optimization horizon, the optimization performed at $t = 1 \times l\,\mathrm{h}$ optimizes over horizon with length $T_P = (T - 1 \times l)$ hours and so on. Although it can appear to be only a negligible "cosmetic" modification, the effect can be crucial as will be shown in the next section. It can be expected that when optimizing a criterion which shall be minimized at given terminal time T, the SHOC approach is able to fulfils the requirements in a better way than the RHOC one as it focuses exactly on the terminal time T.

IV. RESULTS

In this section, results of a set of numerical experiments demonstrating the behavior of the proposed RHOC and SHOC approaches are presented.

The RHOC approach has been tested with different lengths of optimization horizon $T_P \in \{320, 200, 100, 80, 40\}\,\mathrm{h}$, each of the chosen horizons corresponds to P input samples of the input feed flow u, $P \in \{80, 50, 25, 20, 10\}$. Regarding the SHOC approach, effect of different input sampling has been explored - the sampling period has been chosen from a predetermined set $l \in \{4, 5, 8, 16, 20\}\,\mathrm{h}$, which reflects on the maximum number of input samples n_s which are at disposal at the beginning of cultivation, $n_s \in \{100, 80, 50, 25, 20\}$. As the real time proceeds, these numbers gradually decrease by one until the last sub-run, where only one input sample is

optimized. Let us remark, that the RHOC approach always considers that input is piece-wise constant over $4\,\mathrm{h}$. In both cases, the basic tested scenario has been the one with unconstrained computational resources.

Further, computational resources cut-down has been simulated. Regarding the number of iterations which can be performed at the beginning of each sub-run with the current state measurements of the algorithms mentioned in Subsec. III-C and III-D, constraints have been gradually set to $300, 100, 50$ and 25 iterations allowed.

Another very often constraint in industrial practice is restriction imposed on duration of the optimization. This fact is taken into consideration by terminating the sub-run optimization routine if the cumulative sum of the durations of the performed iterations exceeds $60\,\mathrm{s}$.

Fig. 1 and 2 show performance of the SHOC approach in unconstrained case of $n_s = 50$ ($l = 8\,\mathrm{h}$). In the pictures, t_{opt} corresponds to the optimizer internal time at which the current sub-run optimization is computed while t_{real} refers to the real time. Each input profile computed at certain sub-run and the corresponding predicted product concentration profile are plotted with a unique color which differs them from the other sub-run profiles. It is not a surprise that all sub-run profiles are only "shrunk children" of the first sub-run profile (this is common for both the u and x_4 profiles) and it can be shown that in case that no unmodeled phenomena (either noise, unknown/unpredicted external error signals or omitted dynamics) occur and no computational restrictions are imposed, j-th sub-run input profile is equal to the last $T-(j-1)\times l$ hours of the profile obtained at the first sub-run.

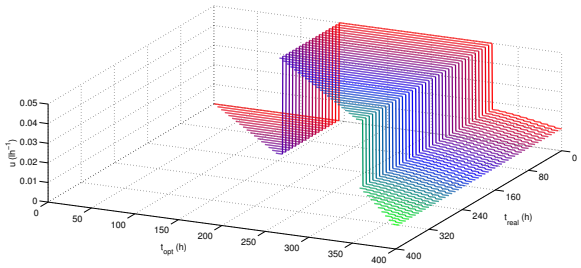

Fig. 1. Evolution of u profile, SHOC (unconstrained), $n_s = 50$.

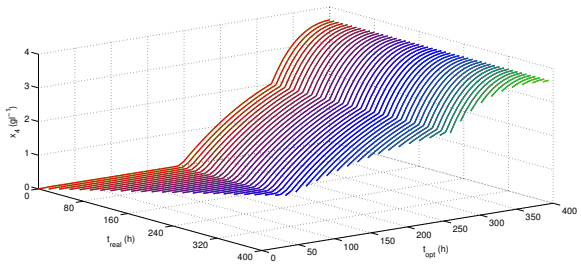

Fig. 2. Evolution of x_4 profile, SHOC (unconstrained), $n_s = 50$.

Looking at Fig. 3 and 4, performance degradation caused by computational resources constraints (maximum allowed number of iteration for each sub-run has been set to 25) can be noticed. It is obvious that the assumption on profile shrinking does not hold under these restrictions, yet the product concentration at time T is only negligibly lower than the results obtained in the unconstrained case. Realizing that 25 iterations can be a really too severe condition even in the industrial practice, the SHOC approach can be regarded as enough restriction-resistant.

Numerical results of all executed SHOC simulations can be found in Tab. III and they clearly show that even with lower number of input samples at disposal, the SHOC approach is able to achieve excellent results with negligible deviation from those obtained using higher number of input samples. Although not recognizable from the values given in Tab. III, it has been witnessed that in unconstrained case of $l = 5\,\mathrm{h}$, slightly better results (however, the difference is less than 10^{-2}) have been obtained than in the unconstrained case of $l = 4\,\mathrm{h}$. This suggests that the results are sensible not only to the chosen sampling period but also to the exact time position of the switching point between low and high input saturation and the non-saturated subsequence of the input profile.

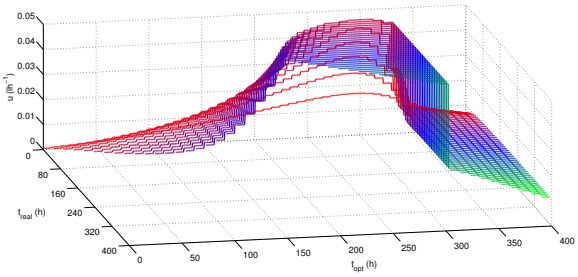

Fig. 3. Evolution of u profile, SHOC (25 iterations allowed), $n_s = 50$.

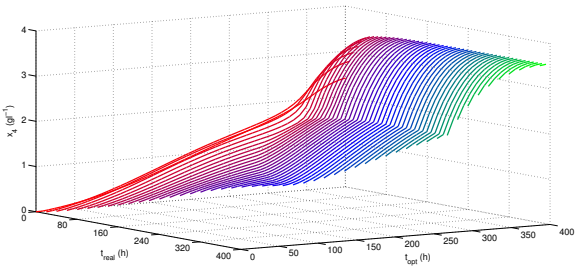

Fig. 4. Evolution of x_4 profile, SHOC (25 iterations allowed), $n_s = 50$.

Looking at Fig. 5, it can be observed how does the input profile evolve in case that the unconstrained RHOC approach with optimization horizon $T_P = 200\,\mathrm{h}$ is used. For the sake of lucidity, optimizer time axis is restricted to $400\,\mathrm{h}$ although it is obvious that the sub-run optimization performed at $t_{real} = 300\,\mathrm{h}$ considers optimization horizon $\langle 300, 500\rangle\,\mathrm{h}$. In this case, no shrinking occurs and rather shifting tendency is visible which is caused by the fact that RHOC always shifts optimization horizon in time.

TABLE III

SHRINKING HORIZON, $x_4|_{t=T}$ (gl^{-1}).

l (h)	n_s (-)	Restrictions					
		—	300 it	100 it	50 it	25 it	60 s
4	100	3.62	3.62	3.62	3.62	3.55	3.62
5	80	3.62	3.62	3.62	3.55	3.54	3.62
8	50	3.62	3.62	3.55	3.54	3.53	3.58
16	25	3.55	3.55	3.54	3.52	3.50	3.55
20	20	3.54	3.54	3.53	3.51	3.47	3.54

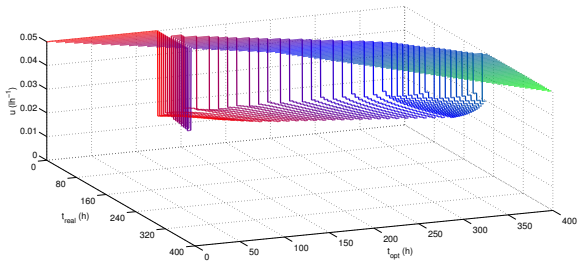

Fig. 5. Evolution of u profile, RHOC (unconstrained), $P = 50$.

Fig. 6 brings interesting insight into evolution of product concentration profile. As - in this particular case - the RHOC approach optimizes over 200 h, the input is operated so that the x_4 value is maximized at the end of that horizon - this leads to such a situation that the optimizer performing optimization routine e.g. at time $t_{real} = 360$ h tries to maximize penicillin production which is to be reached at $t_{real} + T_P = 560$ h instead of maximization at time $T = 400$ h. Due to this fact and the dynamics of penicillin cultivation (for which it has been shown in [6] and [7] that the optimal input profile causes product concentration profile which is quite flat and low at the beginning of cultivation which by the way results from the secondary-metabolism-character of the penicillin cultivation), RHOC generally obtains much lower concentration (see Tab. IV) at final time T than SHOC which always "keeps in mind" optimization at final time T^*. Constrained case (with 25 being the maximum number of iteration) is depicted in Fig. 7 and 8.

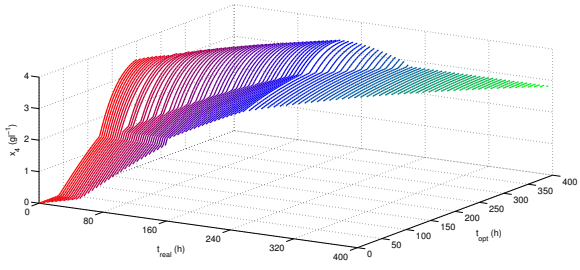

Fig. 6. Evolution of x_4 profile, RHOC (unconstrained), $P = 50$.

Tab. IV summarizes the RHOC simulations and reveals and interesting paradox - with $T_P = 320$ h, the optimizations

* Here, a comparison of the results obtained during the SHOC and RHOC simulations for which it holds that $n_s = P$ seems to be a fair choice.

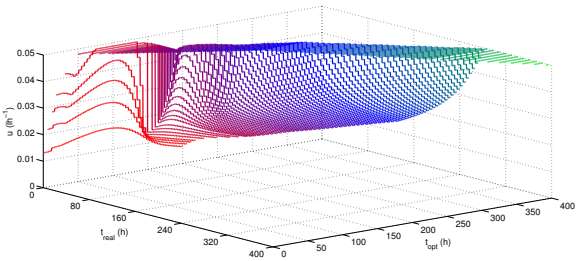

Fig. 7. Evolution of u profile, RHOC (25 iterations allowed), $P = 50$.

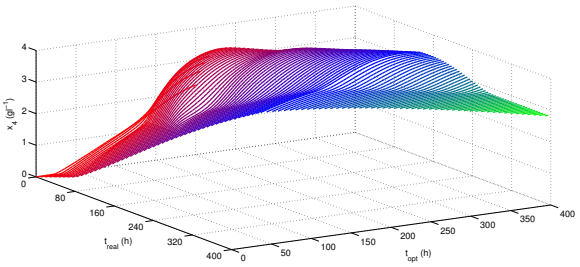

Fig. 8. Evolution of x_4 profile, RHOC (25 iterations allowed), $P = 50$.

restricted to less iterations at each sub-run achieve far better results than those with less strict restrictions although one would expect that the longer the optimization horizon is, the better results can be obtained. This paradox is the next consequence of the use of receding horizon in this particular case - when combined with "too long" optimization horizon T_P, it leads to a situation that the optimizer focuses on the end of optimized horizon which can lie far beyond the important time point T. Under that conditions (optimization horizon is large), it can be better for the optimizer not to converge to the intended sub-run optimum and apply sub-optimal input sequence (which is demonstrated by the first row of Tab. IV). On the other hand, the table shows also the negative impact of too short optimization horizon as for every optimization horizon (except of $T_P = 320$ h) and computational restriction it holds that the lower the horizon is, the lower concentration is reached.

TABLE IV

RECEDING HORIZON, $x_4|_{t=T}$ (gl^{-1}).

T_P (h)	P (-)	Restrictions					
		—	300 it	100 it	50 it	25 it	60 s
320	80	0.47	1.26	2.50	2.60	2.63	2.52
200	50	2.81	2.82	2.82	2.82	2.81	2.82
100	25	2.79	2.77	2.72	2.62	2.65	2.77
80	20	2.60	2.56	2.60	2.60	2.60	2.56
40	10	2.25	2.25	2.24	2.23	2.22	2.25

To sum the comparison up, Fig. 9 and 10 show the applied input profiles and resulting closed loop state profiles for chosen cases (full - unconstrained SHOC with $n_s = 50$, dash - 25 iterations SHOC with $n_s = 50$, dashdot - unconstrained RHOC with $P = 50$, dot - 25 iterations RHOC with $P = 50$).

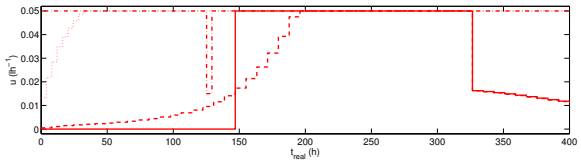

Fig. 9. Applied input profiles.

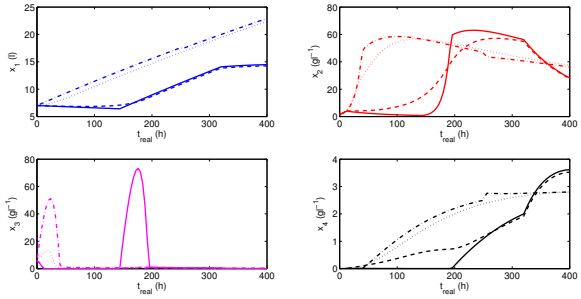

Fig. 10. Closed-loop state profiles.

Trying to overcome the gap between SHOC and RHOC, a hybrid approach has been tested. The key idea is that the algorithm starts as an RHOC one with optimization horizon T_P while the last T_P hours are optimized using the SHOC approach. Let us mention that the input profile is always sampled with sampling period $l = 4\,\mathrm{h}$. The results summarized in Tab. V show that the hybrid approach brings radical improvement (up to almost 300% in the case of $T_P = 320\,\mathrm{h}$) compared with the original RHOC approach. Although it reaches generally lower values of product concentration than the SHOC one, it can be used in cases where both short sampling period and low number of input samples (e.g. from the memory reasons) are required.

TABLE V
COMBINATION OF SH AND RH, $x_4|_{t=T}\,(\mathrm{gl}^{-1})$.

T_P (h)	P (-)	Restrictions					
		—	300 it	100 it	50 it	25 it	60 s
320	80	3.51	3.47	3.46	3.44	3.35	3.45
200	50	3.34	3.34	3.33	3.32	3.30	3.33
100	25	3.36	3.35	3.32	3.22	3.11	3.35
80	20	3.30	3.18	3.09	3.02	2.93	3.20
40	10	2.30	2.30	2.30	2.28	2.26	2.30

V. CONCLUSION

In this paper, two ways of "feedbackization" (RHOC and SHOC) have been proposed and tested with various numbers of input samples at disposal. Moreover, every setting has been examined under several computational restrictions.

A. Results summary

The results clearly show the superiority of the SHOC approach and its resistance against coarse input sampling and computational restriction resistance. An attempt to improve the performance of the RHOC approach - a hybrid approach - has been suggested with satisfactory increase of the final product concentration. Although the SHOC approach achieves better results than the hybrid one, the latter one can be preferred in specific cases e.g. when the size of input samples storage is constrained.

B. Future Work

Although there is place for testing the proposed approaches in combination with two-input strategies (presented in [7] and [8]), more important is that both approaches shall be tested under presence of measurement and/or process noise and external error signals. Moreover, as suggested, the effect of non-coherent input sampling and possibility of time-optimizing procedure shall be explored.

ACKNOWLEDGMENT

This research has been supported by the Czech Science Foundation through the grant no. 13-20433S.

REFERENCES

[1] M. Gregory and C. Turner, "Open-loop control of specific growth rate in fed-batch cultures of recombinant E. coli," *Biotechnology Techniques*, vol. 7, no. 12, pp. 889–894, 1993.
[2] T. Suzuki, T. Yamane, and S. Shimizu, "Phenomenological background and some preliminary trials of automated substrate supply in pH-stat modal fed-batch culture using a setpoint of high limit," *Journal of Fermentation and Bioengineering*, vol. 69, no. 5, pp. 292–297, 1990.
[3] T. Siimes, P. Linko, C. von Numers, M. Nakajima, and I. Endo, "Real-time fuzzy-knowledge-based control of Baker's yeast production," *Biotechnology and bioengineering*, vol. 45, no. 2, pp. 135–143, 1995.
[4] J. Horiuchi, "Fuzzy modeling and control of biological processes," *Journal of bioscience and bioengineering*, vol. 94, no. 6, pp. 574–578, 2002.
[5] A. Ashoori, B. Moshiri, A. Khaki-Sedigh, and M. Bakhtiari, "Optimal control of a nonlinear fed-batch fermentation process using model predictive approach," *Journal of Process Control*, vol. 19, no. 7, pp. 1162–1173, 2009.
[6] M. Pčolka and S. Čelikovský, "Gradient method optimization of penicillin production," in *24th Chinese Control and Decision Conference (CCDC), 2012*, pp. 74–79, IEEE, 2012.
[7] M. Pčolka and S. Čelikovský, "Gradient method optimization of penicillin production: New strategies," in *20th Mediterranean Conference on Control & Automation (MED), 2012*, pp. 1235–1240, IEEE, 2012.
[8] M. Pčolka and S. Čelikovský, "Multiple-Input Cultivation Model Based Optimization of Penicillin Production," in *51st IEEE Conference on Decision and Control, Hawaii, USA*, 2012 (accepted).
[9] J. van Impe and G. Bastin, "Optimal adaptive control of fed-batch fermentation processes with multiple substrates," in *Second IEEE Conference on Control Applications, 1993*, pp. 469–474, IEEE, 1993.
[10] A. Bryson and Y. Ho, "Applied optimal control," *New York: Blaisdell*, 1969.
[11] M. Shetty, *Nonlinear programming*. Wiley Online Library, 1993.
[12] J. Lee, W. Hyun Kwon, and J. Choi, "On stability of constrained receding horizon control with finite terminal weighting matrix," *Automatica*, vol. 34, no. 12, pp. 1607–1612, 1998.
[13] J. Rawlings and K. Muske, "The stability of constrained receding horizon control," *Automatic Control, IEEE Transactions on*, vol. 38, no. 10, pp. 1512–1516, 1993.
[14] G. Nicolao, L. Magni, and R. Scattolini, "Stability and robustness of nonlinear receding horizon control," *Nonlinear model predictive control*, pp. 3–22, 2000.
[15] E. Meadows, M. Henson, J. Eaton, and J. Rawlings, "Receding horizon control and discontinuous state feedback stabilization," *International Journal of Control*, vol. 62, no. 5, pp. 1217–1229, 1995.
[16] T. Ohtsuka, "A continuation/gmres method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, 2004.

4.2 Time Efficiency Maximization

A commonly encountered situation is that the controlled system is expected to operate at the highest achievable time efficiency. The motivation for this requirement comes from the fact that due to the nonlinear nature of the underlying production processes, the yields do not scale linearly with the length of the operating period. Generally speaking, increase of the production period turns into higher achieved yields at its end, however, these yields might be not worth waiting from certain point of time. Instead of increasing the productivity by just prolonging the duration of particular batches, it is desired to reach highest achievable production yields while shortening the operating period of the process. This enables that the production process can be "restarted" very often and over a chosen reference period, the total yields grow. In such cases, the operating/optimization period should also be included in the set of optimizable variables to enable its proper manipulation. However, majority of the commonly used approaches [Ashoori et al., 2009], [Cougnon et al., 2011], [Santos et al., 2012], [Abdollahi and Dubljevic, 2012], [Ghouali et al., 2015], [Chang et al., 2016], [Anilkumar et al., 2018] choose the optimization period a priori not reflecting the dynamics of the system and treat the mentioned optimization tasks as fixed-terminal-time ones, which completely eliminates any opportunity to influence the length of the process operating period.

In [A.15], two algorithms optimizing also the batch duration and thus addressing the time efficiency maximization tasks appropriately were provided. The first of them exploits numerical gradient of the terminal criterion with respect to the optimization horizon and manipulates the horizon directly. In order to ensure that the input profile is sampled coherently and a chosen number of optimized input samples is preserved, the sampling period is recalculated corresponding to the operating horizon. Tackling the given task differently, the second algorithm introduces an alternative time scaling and an auxiliary virtual input corresponding to the time scale and then exploits Hamiltonian-based gradient optimization such that both the original and the auxiliary input are updated in an iterative manner. It should be mentioned that this way, the time duration of particular input samples is optimized. The correct physical sense of the obtained solution is guaranteed by enforcing the auxiliary input to be box-constrained. To prevent the algorithms from converting to nearly-zero operating period or from exceeding the maximal allowed batch length, additional constraints on terminal product amount and terminal time were added in both cases. A comparison of the commonly used fixed-horizon approach and the two horizon-optimizing algorithms was provided in [A.15] with process of penicillin production

serving as a test bed. A single-batch evaluation revealed that only 2.5-times higher amount of the terminal product can be gathered at the price of stretching the operating period 4.5-times, which demonstrates the nonlinearity of the relation between the terminal time and the terminal product and justifies development of horizon-optimizing algorithms. This is also supported by an *in silico* experiment considering a one-month reference period where about 80 % efficiency improvement was observed for both horizon-optimizing algorithms. Here, a slight superiority of the second one can be stated apparently resulting from the latter algorithm having greater optimization freedom—since optimizing the duration of each input sample separately—at disposal.

The aforementioned publication is presented in the original formatting starting on the next page.

2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

Algorithms for Nonlinear Predictive Control Maximizing Penicillin Production Efficiency

Matej Pčolka[1], Sergej Čelikovský[1,2]

***Abstract*— In this paper, maximization of efficiency of the penicillin production is addressed. To overcome the issues caused by the terminal time being an optimizable parameter, two adaptations of the Hamiltonian-based gradient method are proposed. The first algorithm incorporates the terminal time directly into the set of optimized variables while the second one makes use of a time scaling with optimizable time-dependent time scale being a virtual input. Both algorithms are verified on a set of numerical experiments and the obtained results show significant improvement of the penicillin production efficiency gained by their use, which suggests their exploitation within the industrial nonlinear predictive controllers for bioprocess optimization.**

I. Introduction

Bioprocess control can be regarded as a very attractive and at the same time also very challenging area of control engineering. Due to inherently nonlinear character of the dynamics of majority of the bioprocesses, the bioprocesses have been used as test-beds for verification and validation of many control approaches ranging from the earliest open-loop strategies [1], through closed-loop approaches exploiting either on/off [2] or traditional PID schemes, fuzzy approaches [3], [4], [5] to the latest trends represented by the model predictive control [6], [7], [8], [9].

Although providing very promising results, the up-to-date process-control applications of model predictive control usually consider optimization over fixed pre-defined prediction horizon. While the optimization over the fixed horizon might be reasonable in the areas of building climate control, vehicular control, robotics, etc., the efficiency of the production with respect to the elapsed time might play the key role in the area of process control. Therefore, it is highly advantageous to formulate the optimization criterion as maximization of production normalized with respect to the duration of the production reflecting thus the maximization of the productiveness. Such formulation contains explicit dependence on the terminal time which then becomes an optimizable parameter.

Majority of the currently available optimization algorithms exploited within predictive controllers focus on optimization problems with fixed initial state, free terminal state and fixed time interval. The free-time-interval optimization problems usually appear in time-optimal control tasks [10], [11] which frequently occur in the areas of robotics or vehicular control and lead to use of switching curves. However, this approach is not applicable in case of maximization of the production efficiency. Moreover, the free-time-interval optimization problems such as productivity maximization have not been satisfactorily addressed in the area of bioprocess control yet. Therefore, the current paper focuses on formulation and solving of the optimization problem where the terminal time is free but constrained. In order to accomplish this, the existing methodology for the optimization over fixed time interval is adapted and two algorithms for nonlinear predictive control maximizing the production efficiency are developed. Their performance is validated on a series of numerical experiments representing maximization of the penicillin productiveness. Penicillin cultivation is one of the industrially best-known bioprocesses [12], [6], [13] and since it possesses many features and phenomena that typically occur in systems biology, it is often considered as a suitable representative of the bioprocesses. The obtained results show significant increase of the efficiency of the penicillin production that results from the use of the newly proposed algorithms.

The paper is organized as follows: Sec. II introduces the problem to be solved. The dynamics of the controlled process is described and the control objective and constraints are formulated. In Sec. III, two algorithms addressing the formulated optimization problem are presented. In case of the first algorithm, the set of optimized variables is extended and the gradient search for the optimal terminal time is performed. On the other hand, the second alternative introduces a time scaling of the system dynamics description and handles the optimization problem thanks to a virtual input representing the time scale. The results of the two proposed methods are presented in Sec. IV and are compared with the results of optimization with fixed time interval. Sec. V concludes the paper.

II. Problem Formulation

In this Section, both the system description and control requirement specification are provided and the resulting optimization task is formulated.

A. System dynamics

The biomass represented by its concentration C_X (gl^{-1}) in the cultivation broth with volume V (l) is the driving force of the whole cultivation. Consuming the essential nutrient which is fed into the cultivation tank at feed flow rate u (lh^{-1}) containing certain input nutrient concentration $C_{S,in}$, it ensures its own reproduction counteracting the natural death taking place at biomass death rate K_D and, moreover, it produces the secondary metabolite – penicillin. The term *secondary metabolism* refers to the fact that the biomass growth rate μ and penicillin production rate π are rather different,

$$\mu = \mu_{max} \frac{C_S}{K_X C_X + C_S},$$

[1]Department of Control Engineering, Faculty of Electrical Engineering of Czech Technical University in Prague, Technická 2, 166 27 Praha 6, Czech Republic

[2]Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8, Czech Republic

978-1-4673-8682-1/$31.00 ©2016 AACC

$$\pi = \pi_{max} \frac{C_S}{K_P + C_S + C_S^2/K_I}, \tag{1}$$

and reach their maxima at different concentrations of the nutrient C_S (gl^{-1}). The industrially most important variable is the penicillin represented by the penicillin concentration C_P (gl^{-1}) which is being produced at production rate π thanks to the metabolism of the biomass and decreases at constant rate K_H due to the natural hydrolysis. All concentrations – concentration of biomass C_X, concentration of essential nutrient C_S and penicillin concentration C_P – are negatively affected by the dilution of the cultivation broth caused by the feed flow rate u increasing the broth volume V. Last phenomenon to mention is the natural vaporization of the cultivation broth.

Considering states $x = [x_1, x_2, x_3, x_4]^{\mathrm{T}}$ corresponding to $[V, C_X, C_S, C_P]^{\mathrm{T}}$ and the manipulated input u, the dynamics of the cultivation is mathematically expressed as follows:

$$\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= u - K_{vap} x_1, \\
\frac{\mathrm{d}x_2}{\mathrm{d}t} &= \left(\mu_{max} \frac{x_3}{K_X x_2 + x_3} - K_D \right) x_2 - \left(\frac{u}{x_1} - K_{vap} \right) x_2, \\
\frac{\mathrm{d}x_3}{\mathrm{d}t} &= -\left(\frac{\mu_{max}}{Y_{X/S}} \frac{x_3}{K_X x_2 + x_3} + \frac{\pi_{max}}{Y_{P/S}} \frac{x_3}{K_P + x_3 + x_3^2/K_I} \right) x_2 + \frac{C_{S,in} u}{x_1} - \left(\frac{u}{x_1} - K_{vap} \right) x_3, \\
\frac{\mathrm{d}x_4}{\mathrm{d}t} &= \pi_{max} \frac{x_3}{K_P + x_3 + x_3^2/K_I} x_2 - K_H x_4 - \left(\frac{u}{x_1} - K_{vap} \right) x_4.
\end{aligned} \tag{2}$$

The model parameters are listed in Tab. I. For further information about the penicillin cultivation and modeling, see [6], [12], [14] and references therein.

TABLE I
PARAMETERS OF THE MODEL.

Parameter	Value	Parameter	Value
μ_{max}	0.11	$Y_{P/S}$	1.2
π_{max}	0.004	$C_{S,in}$	500
K_P	0.1	K_{vap}	6.23×10^{-4}
$Y_{X/S}$	0.47	K_I	0.1
K_D	0.0136	K_X	0.06
K_H	0.01		

B. Control objectives

Typically, the model predictive controller employed in the process control focuses on maximization of the final product of the controlled process that can be gathered at the pre-defined terminal time T_F. In case of penicillin cultivation studied in the current paper, this effort corresponds to maximization of amount of penicillin at terminal time T_F. Since the particular states represent concentrations of the "bioprocess variables", the optimization task to be solved by the predictive controller can be formulated as follows: given the initial state $x(0) = x_0$, find the optimal input feed flow rate u that minimizes criterion

$$J = -C_P V \,|_{t=T_F} = -x_4(T_F) x_1(T_F) \tag{3}$$

with respect to system dynamics (2) and input constraints

$$0 \leq u \leq u_{max}. \tag{4}$$

The above formulated optimization task with minimization criterion (3) is a problem with fixed initial state, free terminal state and fixed terminal time and can be comfortably solved by the available optimization techniques [15], [14].

Here it should be noticed that the terminal time T_F needs to be chosen very carefully in order to exploit the production capacities in the most economical way. Therefore, it might be more beneficial to maximize the *efficiency* of the penicillin production instead of performing just pure penicillin maximization at the terminal time. This adaptation changes the minimization criterion (3) into

$$J = -\frac{C_P V}{T_F} \,|_{t=T_F} = -\frac{x_4(T_F) x_1(T_F)}{T_F}, \tag{5}$$

where the terminal time is constrained,

$$0 < T_F \leq T_{F,max}. \tag{6}$$

In order to have the problem well-posed, let us partially redefine the criterion (5) such that in case that $x_4(T_F) = 0$, also $J = 0$. To reflect the practical aspects of the penicillin production, certain minimal amount of the penicillin at terminal time is required,

$$P_{min} \leq x_4(T_F) x_1(T_F). \tag{7}$$

The newly formulated criterion (5) with constraints (4) and (7), however, turns the optimization task into a more difficult fixed-initial-state, free-terminal-state and free-terminal-time problem. Although tasks with free terminal time have been studied in control theory for a long time, the available literature usually deals with time-optimal control problems [10], [16]. However, the typical time-optimal control approach based on switching curves is not applicable for the above formulated task. In order to successfully address the problem of productiveness maximization, suitable optimization algorithms are designed in the following Section.

III. OPTIMIZATION ALGORITHMS

Both proposed algorithms come out of the classical Hamiltonian-based gradient method [15], [16]. For a system with

dynamics

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t), u(t)) \tag{8}$$

and minimization criterion

$$J = \int_0^{T_F} L(x,u)\mathrm{d}t + \phi(x(T_F)), \tag{9}$$

the Hamiltonian is created as follows:

$$H = L + \lambda^{\mathrm{T}} f(x,u),$$

where L is the integral term of the minimization criterion (9), $f(x,u)$ represents the dynamics of the controlled system and λ is the co-state vector. Having obtained the state profile x by integrating the differential equations of the system dynamics (8) with fixed initial condition x_0 and considering input profile u^{l-1} from the previous iteration, the co-state vector is obtained as a solution of the backward co-state dynamics differential equation

$$-\dot{\lambda} = \partial H/\partial x,$$

where the terminal condition is specified as

$$\lambda(T_F) = \partial\phi/\partial x|_{t=T_F}.$$

Here, ϕ is the terminal part of the minimization criterion (9).

In order to deal with the state constraints, they are relaxed by introducing an auxiliary variable(s) c_v representing the constraint violation and the minimization of c_v is then incorporated into the minimization criterion. In case considered in this paper, state constraint (7) is handled as follows: at first, let us define

$$c_v = \max(0, P_{min} - x_4(T_F)x_1(T_F)). \tag{10}$$

Then, the updated criterion J is summarized as

$$J = -\frac{x_4(T_F)x_1(T_F)}{T_F} + \omega c_v^2, \tag{11}$$

where ω is a user-defined weighting parameter.

Starting from an initial guess of the optimal input profile u^0, the iterative gradient search for the input profile minimizing the criterion J is performed as

$$u^l = u^{l-1} - \alpha\frac{\partial H}{\partial u}, \tag{12}$$

where α is the gradient step length. In order to simplify the implementation, the input profile is coherently sampled with a pre-defined sampling period T_{samp}. Satisfaction of input constraints $u_{min} \le u \le u_{max}$ is enforced by projecting the calculated profile u^l on the admissible interval $\langle u_{min}, u_{max}\rangle$. The whole procedure is terminated if $|J(u^l) - J(u^{l-1})| < \epsilon$ for some reasonably chosen tolerance $\epsilon > 0$.

A. Algorithm 1

If the terminal time T_F is not fixed and can be manipulated, the gradient search for optimal u can be extended as follows. In order to ensure $\delta J = 0$ and thus reach the extremum of J, it is necessary that $\partial J/\partial T_F = 0$. If this condition is not satisfied, the terminal time T_F can be iteratively adapted following the negative direction of the gradient $\partial J/\partial T_F$,

$$T_F^l = T_F^{l-1} - \alpha_T\frac{\partial J}{\partial T_F} \tag{13}$$

with chosen step length α_T and projecting the calculated T_F^l on $(0, T_{F,max}\rangle$. In case that $\{u, T_F\}$ are the optimizable variables, combined use of (12) and (13) results in minimization of the criterion J. Since analytical calculation of $\partial J/\partial T_F$ is usually difficult, numerical approximation

$$\frac{\Delta J}{\Delta T_F} \approx \frac{\partial J}{\partial T_F} \tag{14}$$

with suitable ΔT_F is to be used in practical implementation. As already mentioned, the original gradient method considers that the inlet feed flow rate is coherently sampled. Therefore, the sampling period T_{samp} is re-calculated according to current T_F^l at each iteration l such that fixed number N of input samples is preserved,

$$T_{samp}^l = \frac{T_F^l}{N}.$$

The algorithm (in further text referred to as FRTETO1 – **FR**ee **TE**rminal **T**ime **O**ptimization 1) is terminated if $|J(u^l, T_F^l) - J(u^{l-1}, T_F^{l-1})| < \varepsilon$, $\varepsilon > 0$.

B. Algorithm 2

The second algorithm FRTETO2 (**FR**ee **TE**rminal **T**ime **O**ptimization 2) representing an alternative to FRTETO1 relies on time scaling where the time scale is considered as a manipulated variable.

The main idea can be explained as follows. Consider a dynamical system with states $x(t)$ and inputs $u(t)$ with dynamics described by (8). At first, let us extend the state vector with an auxiliary state variable (x_5 in the particular case presented in this paper) representing the elapsed time,

$$\frac{\mathrm{d}x_5}{\mathrm{d}t} = 1,\ x_5(0) = 0. \tag{15}$$

It is obvious that $x_5(T_F) = T_F$ and therefore, the terminal value of the auxiliary state x_5 can be used in criterion instead of the length of the optimization time interval T_F. To satisfy the constraints imposed on the terminal time (6), the relaxation is performed and an additional "c_v variable" is introduced,

$$c_{v,T} = \max(0, T_{F,max} - x_5(T_F)). \tag{16}$$

In the specific case considered in this paper, (5) can be reformulated as

$$J = -\frac{x_4(T_F)x_1(T_F)}{x_5(T_F)} + \omega_1 c_v^2 + \omega_2 c_{v,T}^2, \tag{17}$$

where ω_1, ω_2 are user-defined weights. Choosing ω_1, ω_2 large enough, arbitrarily precise satisfaction of the original state constraints can be achieved given that the optimization task with the original constraints was feasible.

This adaptation eliminates the explicit dependence of J on value of T_F. However, in the current situation the terminal value of x_5 is not manipulated and therefore, the terminal time T_F can not be optimized. To overcome this, let us perform a time-scaling with a scaled time τ such that

$$\mathrm{d}t = \tilde{u}(\tau)\,\mathrm{d}\tau \tag{18}$$

with $\tilde{u}$ being the time scale. In order to preserve physical sense, let us require $\tilde{u} \in \langle 0, \tilde{u}_{max} \rangle$, $\tilde{u}_{max} \geq 1$. It is obvious that

$$t = \int_0^\tau \tilde{u}(s)\mathrm{d}s = t(\tau) \tag{19}$$

and

$$\bar{x}(\tau) \equiv x(t), \;\; \bar{u}(\tau) \equiv u(t). \tag{20}$$

By straightforward derivation and combination of (8), (18) and (20), the system description in scaled time τ can be obtained as follows:

$$\frac{\mathrm{d}\bar{x}}{\mathrm{d}\tau} = \frac{\mathrm{d}\bar{x}(t(\tau))}{\mathrm{d}\tau} = \frac{\mathrm{d}x}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}\tau} = \tilde{u}(\tau) f(\bar{x}(\tau), \bar{u}(\tau)) \tag{21}$$

and from (15), it follows that

$$\frac{\mathrm{d}\bar{x}_5}{\mathrm{d}\tau} = \tilde{u}, \;\; \bar{x}_5(0) = 0. \tag{22}$$

Then, considering time scale $\tilde{u}$ to be the auxiliary manipulated variable, a fixed-terminal-time optimization task in scaled time τ on time interval $\langle 0, \tau_F \rangle$ is equivalent to the free-terminal-time problem in time t on interval $\langle 0, T_F \rangle$, $0 \leq T_F \leq \tilde{u}_{max}\tau_F$ with the terminal time

$$T_F = \int_0^{\tau_F} \tilde{u}(s)\mathrm{d}s. \tag{23}$$

Furthermore, it can be shown that the coherently sampled solution of the optimization task in scaled time τ

$$\{\bar{u}^*, \tilde{u}^*\} = \arg\max J(\bar{x}, \bar{u})$$

with constant sampling period τ_{samp} is equivalent to a piecewise constant solution of

$$\{u^*\} = \arg\max J(x, u)$$

with non-coherent sampling $T_{samp}(t) \equiv \tilde{u}(\tau)\tau_{samp}$. Thanks to this, the free-terminal-time optimization problem in t can be re-formulated using the time scaling, solved as the fixed-terminal-time optimization task in τ using the above described Hamiltonian-based gradient method and then the optimal inlet feed flow rate is re-scaled to the original time t according to the optimal scaling $\tilde{u}^*$.

IV. Results

In this section, the results of the FRTETO1 and FRTETO2 algorithms are presented. To provide a more comprehensive comparison and demonstrate the improvement of the efficiency of the penicillin production gained by the use of the FRTETO1 and FRTETO2 algorithms, the results of the optimization with fixed terminal time are presented as well. The scenario used to evaluate the performance of all compared alternatives is considered as follows: the productivity of the penicillin during one month shall be maximized ($T_{F,max} = 720\,\mathrm{h}$). The optimization constraints and state initial conditions are listed in Tab. II.

The commonly used option is to choose $T_F = T_{F,max} = 720\,\mathrm{h}$ and find inlet feed flow rate that minimizes criterion (3) – this option is in further text referred to as FITETO (**FI**xed **TE**rminal **T**ime **O**ptimization). In this case, the sampling period of the input u was chosen as $T_{samp} = 4\,\mathrm{h}$. Since the terminal time is fixed such the constraint (6) is always satisfied, this constraint does not need to be considered. The interested readers are warmly referred to [14], [17] for further information.

TABLE II
Constraints.

Parameter	Value	Parameter	Value
$x_{1,0}$	7.012	u_{max}	0.05
$x_{2,0}$	1.5	$\tilde{u}_{max}$	5
$x_{3,0}$	6	P_{min}	30
$x_{4,0}$	0		

On the other hand, the more appropriate way of addressing such task might be as follows: at first, find the time interval $\langle 0, T_F^* \rangle$, $0 \leq T_F^* \leq T_{F,max}$, and the corresponding inlet feed flow rate such that (5) is minimized and then perform as many cultivations with duration T_F^* as possible such that the overall sum of durations of all cultivations does not exceed $T_{F,max} = 720\,\mathrm{h}$. In other words, the cultivation over the time interval $\langle 0, T_F^* \rangle$ is repeated C-times where

$$C = \max\{c \,|\, c \in \mathbb{N}^+;\, T_{F,max} \geq c \times T_F^*\}.$$

To compare the performance of all alternatives, the following set of evaluators is introduced. At first, the achieved efficiency of the production E (gh^{-1}) is inspected,

$$E = \frac{C_P(T_F)V(T_F)}{T_F}.$$

Furthermore, the amount of penicillin P_1 (g),

$$P_1 = C_P(T_F)V(T_F)$$

produced during one cultivation and the length of one cultivation T_F are provided. Moreover, C (–) corresponding to the number of the *completed* cultivations (each of length T_F) that fit into the time interval $\langle 0, T_{F,max} \rangle$ is evaluated together with the total duration T_{tot} (h) of all completed cultivations

$$T_{tot} = C \times T_F$$

and the total penicillin P_{tot} (g) produced during C completed cultivations

$$P_{tot} = C \times P_1.$$

The numerical results for $T_{F,max} = 720$ are provided in Tab. III.

TABLE III
Comparison of the algorithms, $T_{F,max} = 720$.

	E	P_1	T_F	C	T_{tot}	P_{tot}
FITETO	12.7×10^{-2}	91.1	720	1	720	91.9
FRTETO1	22.8×10^{-2}	40.2	176.6	4	706.4	160.8
FRTETO2	23.1×10^{-2}	36.6	158.4	4	633.6	146.4

Looking at Tab. III, a quite interesting thing can be observed. While the efficiency of the production E is almost the same for both FRTETO1 and FRTETO2 algorithm (the difference is about 1 %), the obtained one-cultivation yield P_1 and the calculated length of one cultivation T_F differ significantly. This can be owed to the different optimization approach – FRTETO1 sets the sampling period equally for

all input profile samples while FRTETO2 optimizes the duration of each sampling instance separately. The fact that FRTETO2 has more degrees of freedom when performing the optimization leads to slightly higher efficiency E achieved by the FRTETO2. This also demonstrates the nonlinear dependence of the achieved penicillin amount on the length of the cultivation period – to increase the amount of the produced penicillin by almost 10 % from 36.6 g (FRTETO2) to 40.2 g (FRTETO1), more than 11 % increase of the cultivation length is necessary (158.4 h for FRTETO2 vs. 176.6 h for FRTETO2). This phenomenon becomes much more significant for longer cultivation periods – during one cultivation with fixed length of 720 h (4.5-times longer than in case of FRTETO2 algorithm), 91.9 g of penicillin is produced which is only 2.5-times more than in case of FRTETO2 algorithm. As a result, the productiveness drops from more than 23.1×10^{-2} gh^{-1} (FRTETO2) to only 12.7×10^{-2} gh^{-1} (FITETO). The significance and efficiency improvement reached by use of the newly proposed algorithms are indisputable.

From Tab. III, it can be also observed that using the FRTETO1 algorithm, higher P_{tot} is achieved. This is due to the fact, that although the efficiency is slightly lower for FRTETO1, 4 longer cultivations of FRTETO1 fit into the $T_{F,max} = 720$ h and since more penicillin is produced during each of them, also the resulting amount of penicillin P_{tot} is higher than in case of FRTETO2. Here, let us remark that for calculation of P_{tot}, only *fully completed* cultivations are taken into account. However, decreasing the $T_{F,max}$ to 700 h, the situation changes – this can be seen from Tab. IV. With the lower $T_{F,max}$, only 3 longer FRTETO1 cultivations fit into it unlike 4 shorter FRTETO2 cultivations. As a result, FRTETO2 is able to finish more complete cultivations and therefore, it also achieves higher P_{tot}.

Furthermore, comparing FITETO results from Tab. III and Tab. IV, it can be seen that although P_{tot} is lower for $T_{F,max} = 700$, the penicillin is being produced more effectively during the shorter time period. Based on this, it can be concluded that the longer periods obviously lead to more penicillin produced, however, the amount of penicillin gathered during the period of time by which the terminal time is increased is usually not worth waiting.

TABLE IV
COMPARISON OF THE ALGORITHMS, $T_{F,max} = 700$.

	E	P_1	T_F	C	T_{tot}	P_{tot}
FITETO	12.9×10^{-2}	90.3	700	1	700	90.3
FRTETO1	22.8×10^{-2}	40.2	176.6	3	529.8	120.6
FRTETO2	23.1×10^{-2}	36.6	158.4	4	633.6	146.4

Fig. 1 shows the penicillin profiles normalized with respect to the terminal time T_F as a function of normalized time t/T_F. The red line shows the FITETO with $T_F = 720$ h, the profile of FITETO with $T_F = 700$ h is depicted in blue color, the green line represents FRTETO1 algorithm and the black line shows profile obtained by FRTETO2 algorithm. Let us note that on the time axis, $t/T_F = 1$ refers to the end of the cultivation with the particular algorithm. It can be seen that the newly proposed algorithms are roughly twice as effective in maximization of the penicillin productiveness as the fixed-interval algorithms.

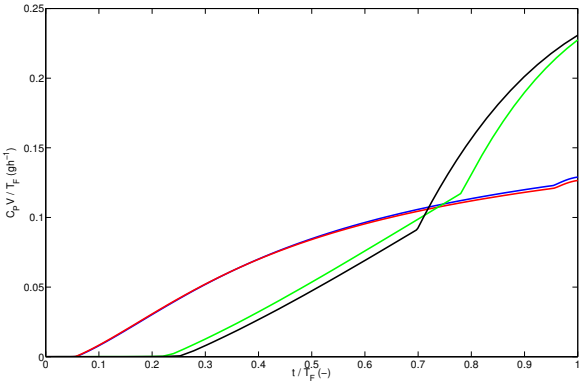

Fig. 1. Normalized penicillin production.

Fig. 2 brings comparison of the relative penicillin production $C_P(t)V(t)/t$. The coloring is the same as in case of Fig. 1. Looking at the picture, it can be seen that while the FRTETO1 and FRTETO2 profiles are monotonically increasing and reach their maxima at the end of the profile which corresponds to the fact that the relative penicillin production at *terminal time* is maximized, the maxima of the FITETO profiles occur somewhere close to the one third of the cultivation period. It is obvious that also in case of FITETO algorithm, it would be much more advantageous to terminate the cultivation much earlier instead of waiting till the end of cultivation $T_{F,max}$. Based on the observed profiles for FITETO algorithm, a naive approach that would improve its performance could be formulated: maximize the amount of penicillin at the end of the cultivation $T_{F,max}$ and when the relative penicillin production $C_P(t)V(t)/t$ starts decreasing, terminate the cultivation. Apparently, the premature termination of the cultivation would spare much time consumed by the FITETO algorithm while leading to higher penicillin production efficiency.

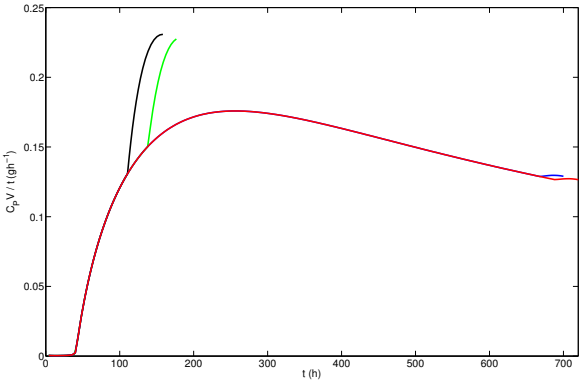

Fig. 2. Relative penicillin production.

In order to provide the readers with an information about the feed flow rates applied by the particular algorithm, Fig. 3 shows the u^* input profiles. An indication of a superprofile

convergence that frequently occurs in bioprocess control [14], [17], [18] can be observed in this case as well – the almost linearly decreasing part of the input profile is gradually shifted towards the end of cultivation and it is made steeper for the increasing T_F. Inspecting the input profile for FRTETO2 (the black line), a very short and rapid decrease from the upper to the lower boundary shortly after $t = 100\,\mathrm{h}$ can be observed. This phenomenon has not been reported in the previous works due to the fact that they considered coherent sampling with constant sampling period which was chosen too long for the sudden drop of the inlet feed flow rate to appear. Since FRTETO2 optimizes also the duration of the particular input samples, such drop with very short duration can occur.

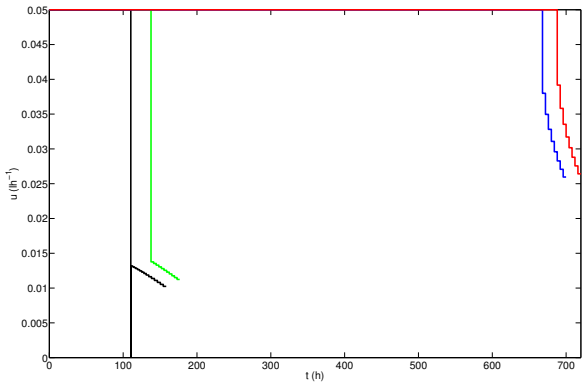

Fig. 3. Input profiles.

To complete the visualization of the obtained results, Fig. 4 shows the total penicillin production for $T_{F,max} = 700\,\mathrm{h}$. The green, black and blue lines represent total penicillin profiles for FRTETO1, FRTETO2 and FITETO algorithm. The dashed lines mark the *completed* cultivations performed by the FRTETO1 and FRTETO2 algorithms while the markers indicate the achieved T_{tot} and P_{tot} values.

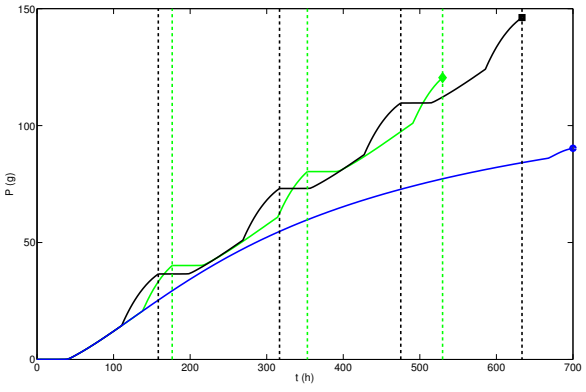

Fig. 4. Total penicillin production ($T_{F,max} = 700\,\mathrm{h}$).

V. Conclusion

In this paper, two algorithms suitable for nonlinear predictive controller maximizing the efficiency of the penicillin production were presented. Both of them treat the cultivation period as a free parameter in addition to the traditionally optimized inlet feed flow rate. The first alternative manipulates the length of the cultivation directly and re-calculates the sampling period of the inlet feed flow rate accordingly. The second one exploits a time scaling with the time-dependent scale being the virtual input, which leads to the length of particular sampling instances being also optimized. The encouraging results show efficiency improvement of almost $82\,\%$ which demonstrates the usefulness of the proposed algorithms and suggests their exploitation by the nonlinear predictive controllers in the industrial practice.

Acknowledgement

This research has been supported by the Czech Science Foundation through the grant no. 13-20433S.

References

[1] M. Gregory and C. Turner, "Open-loop control of specific growth rate in fed-batch cultures of recombinant E. coli," *Biotechnology Techniques*, vol. 7, no. 12, pp. 889–894, 1993.

[2] T. Suzuki, T. Yamane, and S. Shimizu, "Phenomenological background and some preliminary trials of automated substrate supply in pH-stat modal fed-batch culture using a setpoint of high limit," *Journal of Fermentation and Bioengineering*, vol. 69, no. 5, pp. 292–297, 1990.

[3] T. Siimes, P. Linko, C. von Numers, M. Nakajima, and I. Endo, "Real-time fuzzy-knowledge-based control of Baker's yeast production," *Biotechnology and bioengineering*, vol. 45, no. 2, pp. 135–143, 1995.

[4] J. Horiuchi, "Fuzzy modeling and control of biological processes," *Journal of bioscience and bioengineering*, vol. 94, no. 6, pp. 574–578, 2002.

[5] B. Cosenza and M. Galluzzo, "Nonlinear fuzzy control of a fed-batch reactor for penicillin production," *Computers & Chemical Engineering*, vol. 36, pp. 273–281, 2012.

[6] A. Ashoori, B. Moshiri, A. Khaki-Sedigh, and M. Bakhtiari, "Optimal control of a nonlinear fed-batch fermentation process using model predictive approach," *Journal of Process Control*, vol. 19, no. 7, pp. 1162–1173, 2009.

[7] M. Pcolka and S. Celikovsky, "On nonlinear continuous-time optimal control of penicillin cultivation," in *Control Conference (ECC), 2013 European*, pp. 4442–4447, IEEE, 2013.

[8] J. Wan, O. Marjanovic, and B. Lennox, "Disturbance rejection for the control of batch end-product quality using latent variable models," *Journal of Process Control*, vol. 22, no. 3, pp. 643–652, 2012.

[9] S. Craven, J. Whelan, and B. Glennon, "Glucose concentration control of a fed-batch mammalian cell bioprocess using a nonlinear model predictive controller," *Journal of Process Control*, vol. 24, no. 4, pp. 344–357, 2014.

[10] H. Hermes and J. P. LaSalle, "Functional analysis and time optimal control," tech. rep., DTIC Document, 1971.

[11] J. E. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The international journal of robotics research*, vol. 4, no. 3, pp. 3–17, 1985.

[12] J. van Impe and G. Bastin, "Optimal adaptive control of fed-batch fermentation processes with multiple substrates," in *Second IEEE Conference on Control Applications, 1993*, pp. 469–474, IEEE, 1993.

[13] F. Logist, B. Houska, M. Diehl, and J. F. Van Impe, "Robust multi-objective optimal control of uncertain (bio) chemical processes," *Chemical Engineering Science*, vol. 66, no. 20, pp. 4670–4682, 2011.

[14] M. Pcolka and S. Celikovsky, "Gradient method optimization of penicillin production," in *Control and Decision Conference (CCDC), 2012 24th Chinese*, pp. 74–79, IEEE, 2012.

[15] A. Bryson and Y. Ho, "Applied optimal control," *New York: Blaisdell*, 1969.

[16] K. Zhou, J. C. Doyle, K. Glover, *et al.*, *Robust and optimal control*, vol. 40. Prentice Hall New Jersey, 1996.

[17] M. Pčolka and S. Čelikovský, "Multiple-input cultivation model based optimization of penicillin production," in *IEEE 51st Annual Conference on Decision and Control (CDC), 2012*, pp. 7075–7080, IEEE, 2012.

[18] S. Celikovsky, S. Papacek, A. Cervantes-Herrera, and J. Ruiz-Leon, "Singular perturbation based solution to optimal microalgal growth problem and its infinite time horizon analysis," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 767–772, 2010.

4.3 Computational/Memory Demand Reduction

The last part of this chapter focuses on the choice of optimization horizon aiming at reduction of both the computational and memory burden. In application-oriented works, most of such efforts are implemented based on heuristic principles by setting the horizon to be constant having found it as a trade-off between the performance expectations and the computational load of the corresponding optimal controller. Nevertheless, for certain class of the controlled systems, an adaptive online rule for the choice of the prediction horizon can be derived. Then, the optimization horizon can be updated in active operation based on the general technical/physical constraints imposed on the system and exploiting actual measurements of the output variables and the current influences of the environment.

In [A.3], a racing car optimal control problem was studied. In particular, a (hybrid nonlinear) predictive controller[2] with *adaptive optimization horizon* was designed such that the lap-time was minimized and at the same time, safety requirements given by the track width were respected. Speaking about this particular task, the motivation for adaptive optimization horizon turns out to be very intuitive: driving at higher speeds requires longer prediction horizons and vice versa. In the mentioned work, several optimization horizon formulas were provided (see Sec. III. C of [A.3]), all of them employing the current velocity measurements. Following the first method, the horizon is calculated as a linear function of the vehicle speed. Although very gentle and straightforward in terms of effort needed to calculate the horizon, this choice requires the linear coefficient to be tuned based on additional information about the particular track to achieve satisfactory behavior. The other two (logarithmic) formulas adopt a different philosophy, make additional use of the internal model of the dynamics of the controlled vehicle used also by the optimization routine and calculate the horizon long enough to bring the velocity from its current value to a predefined range with these ranges being their main differentiating element. Here, it should be remarked that this model-based calculation takes the current parameters of the system dynamics, the environmental aspects (e.g. road slipperiness) and also the input action constraints into account. Combining this with a suitable velocity interval choice, preservation of the maneuverability properties and satisfaction of the safety requirements can be guaranteed; the corresponding theoretical derivations are also provided in Sec. III.C of [A.3]. An interesting consequence is that a predictive controller with logarithmic horizon is able to stabilize the vehicle with respect to the racing track given that an optimal

[2]The design of the underlying optimization algorithm enabling to handle tasks involving hybrid system dynamics and/or piecewise continuous cost criterion is presented in Chapter 5.2 of this thesis.

controller with infinite horizon is capable of satisfying this requirement. Numerical experiments and comparisons presented in the mentioned paper demonstrate that while also the linear-horizon formula can balance performance and computational/memory demands and outperforms the constant-horizon controller, only the logarithmic-horizon approaches are Pareto-optimal with respect to multiple evaluative criteria. Moreover, a detailed sensitivity analysis reveals satisfactory robustness of the proposed control scheme using either of the last two adaptive horizon formulas against various parameter perturbations.

Since the publication [A.3] mentioned above combines two contributions of this thesis, its full text is presented later in Chapter 5.2.

Chapter 5

Robustification against Discontinuity Effects

In this chapter, robustification against effects resulting from presence of various discontinuities is studied.

In general, this class of problems can be divided into two subclasses: *i*) tasks dealing with optimization of discrete-valued variables, i.e. *quantized inputs optimization*, and *ii*) tasks involving systems with piecewise continuous dynamics and/or piecewise continuous cost criterion, i.e. *discontinuous dynamics/discontinuous cost optimization*. Each of these subclasses has its own specifics and peculiarities and since the approaches addressing and solving them developed within this thesis differ substantially, they are discussed in separate subchapters.

5.1 Quantized Inputs Optimization

Optimization considering quantized inputs is a rather delicate task and when discussing rigorous approaches, mainly linear systems have been considered so far [Picasso et al., 2003], [Muller et al., 2011], [Aguilera and Quevedo, 2013]. Although also for nonlinear systems, some theoretical conclusions might be drawn [Rawlings and Risbeck, 2017], not much about the actual way *how exactly* the problem can be solved is provided in the theoretically-oriented papers. Therefore, for the subclass of the problems involving optimization of discrete-valued inputs for *nonlinear* systems, the most common approach is to employ a posteriori quantization. Thanks to its simplicity, this naive method is highly popular especially among industrial practice control engineers since it does not require

any adaptations of the optimization routine itself, can be handled by a hardware solution and thus it preserves the original complexity and calculation time demands of the resulting controller. On the other hand, it suffers from several substantial drawbacks, namely: *i)* the a posteriori quantization introduces undesirable chattering of the optimized input leading to increased actuator wear and decreased life span and, moreover, this chattering can often "leak into" other (even continuous-valued) control signals, and *ii)* excluding the post-processing from the optimization routine aggravates the negative effects the input quantization might have on the control quality and results in significant performance degradation compared with the continuous-valued model predictive controller. A more appropriate but at the same time also a far more complicated approach is to formulate the discrete-valued input optimization task as a mixed-integer nonlinear programming one and employ a suitable mixed-integer solver [Bemporad, 2003], [Chakrabarty et al., 2017]. This choice suppresses both of the mentioned drawbacks, on the other hand, the computational and memory demands even for low number of quantization levels are several times higher than those of the original continuous-valued problem and sooner or later, the curse of dimensionality fully develops, which might be devastating for applicability of such approach. Therefore, practical deployment of online control systems based on the mixed-integer nonlinear programming approach still remains mostly hypothetical and so far, it has not seen much exploitation. To bring mixed-integer-based approaches closer to real life, explicit solutions can be used [Grancharova and Johansen, 2009], [Grancharova and Johansen, 2012], [Chakrabarty et al., 2017], however, the computational/memory burden even for moderate prediction horizons and moderate number of input/state variables are huge. A nicely straightforward but rather exotic approach is to evaluate all the possible variants and choose the one minimizing the cost function [Cortes et al., 2012], [Rodriguez et al., 2013].

This thesis introduces a robustified optimization algorithm relying on a regularly executed mid-processing optimization iteration. Its main philosophy can be explained as follows: at first, the optimized input profile from the previous optimization iteration is projected on the discrete set of admissible input values. Then, this profile is extended with the already applied input samples and the whole sequence is filtered using a low-pass filter with a suitable characteristics. Last of all, the last P samples of the sequence are extracted, projected on the admissible set of input values and subsequently used for the next optimization iteration. Using the mid-processing iteration, the information about the input quantization is incorporated directly into the optimization routine. Moreover,

the high-frequency portion of the optimized input signal is reduced and thanks to that, both the undesirable oscillatory behavior and control performance degradation are remedied. Furthermore, the calculation time is independent of the number of input quantization levels and is nearly the same as that of the original continuous-valued task, which distinguishes the proposed algorithm from the mixed-integer programming approach where the computational time rises very steeply. The robustified algorithm was presented in [A.14] and in Sec. 4 and 5 of [A.2] and in both cases, the obtained results were compared with those of the naive approach exploiting the a posteriori quantification while as the baseline, the continuous-valued nonlinear predictive controller was considered. In the first case summarized in [A.14] dealing with a fast-dynamics system, the median of the control performance degradations observed for a set of $2 - 21$ quantization levels dropped from 81 % for the naive a posteriori quantization to 6 % for the newly proposed robustified algorithm while the control signal was $3 - 24\times$ less oscillatory with the new algorithm than with the naive approach. Similar results reported in [A.2] were achieved when controlling a slow-dynamics system: the control performance aggravation of up to almost 30 % (naive approach) was remedied to at most 17 % (mid-processing-iteration-based algorithm) while the oscillations of both the discrete- and continuous-valued input profiles occurring for the naive approach were significantly suppressed with use of the newly proposed robustified optimization algorithm.

The publications mentioned above are presented in the original formatting and are provided on the next page et seq. Let us remind that while the whole publication [A.14] focuses on the quantized inputs optimization, primarily Sec. 4 and 5 of [A.2] address this task.

2015 IEEE Conference on Control Applications (CCA)
Part of 2015 IEEE Multi-Conference on Systems and Control
September 21-23, 2015. Sydney, Australia

Quantized Nonlinear Model Predictive Control for a Building

Matej Pčolka[1], Eva Žáčeková[1], Rush Robinett[2], Sergej Čelikovský[1,3], Michael Šebek[1]

Abstract— **In this paper, the task of quantized nonlinear predictive control is addressed. In such case, values of some inputs can be from a continuous interval while for the others, it is required that the optimized values belong to a countable set of discrete values. Instead of very straightforward a posteriori quantization, an alternative algorithm is developed incorporating the quantization aspects directly into the optimization routine. The newly proposed quaNPC algorithm is tested on an example of building temperature control. The results for a broad range of number of quantization steps show that (unlike the naive a posteriori quantization) the quaNPC is able to maintain the control performance close to the performance of the original continuous-valued nonlinear predictive controller and at the same time it significantly decreases the undesirable oscillations of the discrete-valued input.**

I. INTRODUCTION

The last decades have brought significant rise of use of modern control methods such as Model Predictive Control (MPC). Increasing popularity can be witnessed not only amongst academicians but also in the community of process control engineers [1], [2], [3], [4]. Furthermore, thanks to its considerable potential for simultaneous energy saving and thermal comfort satisfaction, MPC has become popular also in rather conservative areas such as building climate control. However, although current literature offers not only large amount of simulation studies [5], [6], [7], [8], [9] but also several interesting real MPC applications for building climate control [10], [11], some practical aspects and questions remain open. One of the most challenging tasks is optimal control considering discrete-valued input actions.

Sometimes the actuator constraints result in a situation that part of the manipulated variables might not be set with infinite accuracy and rather than from a continuous interval, they can be chosen only from some countable set of discrete values. The simplest and the most straightforward way to handle such an issue is to use traditional continuous-valued optimization and (when converged) project the calculated optimal input sequence on the desired discrete-valued admissible set. Although simple indeed (and thus often used in the industrial practice), the *a posteriori* quantization usually causes undesirable oscillations of the discrete-valued input which contribute to actuator abrasion, increase of actuator stress and failures and reduction of the actuator service lifetime. Moreover, optimality of the a posteriori quantized solution obtained by continuous-valued optimization can not be guaranteed as long as the optimization does not take the quantization into account. Therefore, the a posteriori quantization might deteriorate the control performance badly.

[1]Department of Control Engineering, Faculty of Electrical Engineering of Czech Technical University in Prague, Technická 2, 166 27 Praha 6, Czech Republic

[2]Department of Mechanical Engineering–Engineering Mechanics, Michigan Technological University, 1400 Townsend Drive, Houghton, Michigan, United States

[3]Institute of Information Theory and Automation, Czech Academy of Sciences, Pod Vodárenskou věží 4, 182 08 Praha 8, Czech Republic

A very high-performance approach is to formulate the optimization task as a mixed integer optimization problem. However, solution of such complicatedly formulated task is usually time consuming with steeply rising complexity [12], [13] and requires massive computational power since the mixed-integer programming problems are known to be NP-hard [14]. Although there are several solvers able to handle nonlinear systems and nonlinear optimization criteria, majority of the reliable ones are not free for industrial use. Further information can be found in [15].

In the current paper, we focus on development of an alternative to the already mentioned approaches. We exploit continuous-valued optimization and make a suitable adaptation of the internal optimization procedure such that the samples of the resulting input sequence belong to the chosen set of discrete values. Instead of making use of time-consuming and computationally demanding nonlinear mixed integer programming, we propose a *mid-processing* procedure that both results in quantized optimized input sequence and is also computationally modest. The functionality of the proposed quaNPC algorithm is demonstrated using an example of zone temperature control for a building in Houghton, Michigan, and its performance is compared with both the original continuous-valued nonlinear MPC and the a posteriori quantized version.

This paper is organized as follows. In Sec. II, the controlled system is presented including the model description, evaluative criterion and imposed constraints specification. Sec. III provides a detailed description of the newly proposed optimization procedure. The performance of the newly proposed quaNPC algorithm is examined in Sec. IV. The comparison with the original nonlinear MPC and the a posteriori quantized nonlinear MPC (apqNMPC) shows significant reduction of the oscillations compared with the a posteriori quantized version while keeping the control performance close to that of the original continuous-valued NMPC. Sec. V concludes the paper.

II. BUILDING OF INTEREST

In this Section, one particular zone in the Lakeshore Building of Michigan Technological University (Houghton, Michigan) serving as the testbed for the proposed quaNPC algorithm is described. Besides the model, also optimization criterion and constraints for the manipulated variables are provided.

A. Description

In the current paper, a single zone which is a part of Michigan Technological University is considered. Having size of $10.2 \times 8.5 \times 2.7\,\mathrm{m}$ and being equipped with an air-handling unit system, it represents the typical office zone very reliably. As can be seen from the Fig. 1 (the investigated zone is the middle one), the zone is surrounded by the corridor and the neighboring zones from three sides and thus has one exterior wall. The temperatures of the neighboring zones,

978-1-4799-7787-1/15/$31.00 ©2015 IEEE

corridor and ambient temperature are all measured and are considered as disturbances. Let us note that since for the ambient temperature, weather forecast is available with sufficient prediction horizon, these predictions are incorporated directly into the predictive control algorithm. The temperatures of neighboring zones and corridor are not predicted and therefore are only included as currently measurable (but nonpredictable) disturbances. For the investigated zone itself, measurements from two temperature sensors (S_{1a} and S_{1b} in Fig. 1) – such configuration is exploited in the modeling and validation part since the MPC internal model predicts their average, however, a more detailed validation test-bed models temperatures measured by the two sensors separately. For the sake of completeness, Fig. 1 shows locations of the sensors of temperatures in the neighboring zones/corridor (S_2, S_3 and S_4) and the ambient temperature (S_5) as well.

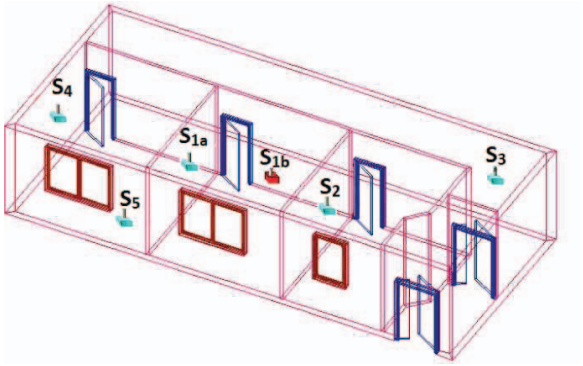

Fig. 1. Scheme of the building (taken from [16]).

Regarding the optimizable variables, the following three quantities can be manipulated: supply air temperature T_S, supply air mass flow rate $\dot{m}$ and indoor/outdoor air mixing ratio δ. The last one plays an important role in providing the fresh air during the working hours.

For better clarity, the performance of the air handling unit (AHU) can be explained as follows. At first, the air mixture is created as a combination of the air from the controlled zone with temperature T_Z and the fresh outside air with temperature T_O where δ specifies the mixing ratio. The temperature of the air mixture T_m is then given as

$$T_m = \delta T_Z + (1-\delta)T_O. \tag{1}$$

If necessary, the mixed air can be heated to the required supply air temperature $T_m \rightarrow T_S$, while cooling is not allowed. Afterwards, the supply air is sent to the zone at the supply air mass flow rate $\dot{m}$. Let us note that the output of the supply air fan is quantized with certain quantization step Q, therefore the fan can produce only several levels of mass flow rate rather than an arbitrary value from a continuous interval.

B. Mathematical models

In order to obtain a reliable comparison of all examined alternatives (apqNMPC, continuous-valued NMPC and quaNPC), two models with slightly different structures were identified from the available data. In the following descriptions, T_Z stands for zone temperature, $T_D = [T_E, T_C, T_W, T_O]^{\mathrm{T}}$ represent the disturbance temperatures corresponding to the temperatures of the east and west room (T_E and T_W), corridor (T_C) and outside environment (T_O) and T_S and $\dot{m}$ denote the supply air temperature and mass flow rate.

Structure I with the simpler structure serves as the internal model for the predictive controllers. The internal-model structure is considered as follows:

$$T_{Z,k+1} = \sum_{i=0}^{n_a-1} a_i T_{Z,k-i} + \sum_{q=1}^{4} \sum_{i=0}^{n_D^q-1} b_i^q T_{D,k-i}^q + c\dot{m}(T_{S,k} - T_{Z,k}). \tag{2}$$

with n_a and n_D^q being the numbers of the lagged state and q-th disturbance values. Here, $n_a = 2$ and $n_D^q = [1,\ 1,\ 1,\ 2]$ were chosen. Regarding the data pre-processing, T_Z was calculated as average of T_{Z1} and T_{Z2} measured by the two sensors (S_{1a} and S_1b).

The performance of the controllers was validated considering Structure II with the following more involved structure:

$$\begin{aligned} T_{Z1,k+1} &= a_1 T_{Z1,k} + a_2 T_{Z1,k} + \sum_{q=1}^{4} \sum_{i=0}^{n_{D1}^q-1} b_{i,1}^q T_{D,k-i}^q \\ &\quad + c\dot{m}(T_{S,k} - T_{Z1,k}) \\ T_{Z2,k+1} &= b_1 T_{Z1,k} + b_2 T_{Z2,k} + \sum_{q=1}^{4} \sum_{i=0}^{n_{D2}^q-1} b_{i,2}^q T_{D,k-i}^q. \end{aligned} \tag{3}$$

In this structure, the measured temperatures T_{Z1} and T_{Z2} are modelled separately. In this case, $n_{D1}^q = [1,\ 1,\ 1,\ 1]$ and $n_{D2}^q = [1,\ 1,\ 1,\ 0]$ were considered.

The parameters $\{a_i, b_i^q\, c\}$ of the Structure I and $\{a_{\{1,2\}}, b_{\{1,2\}}, b_{i,\{1,2\}}^q, c\}$ of the Structure II, respectively, were identified using MPC relevant identification method [17]. Since this method performs multi-step output prediction errors minimization, it is especially suitable for identification for MPC and was already succesfully used in case of identification of building models [18].

Figure 2 shows comparison of the identified models.

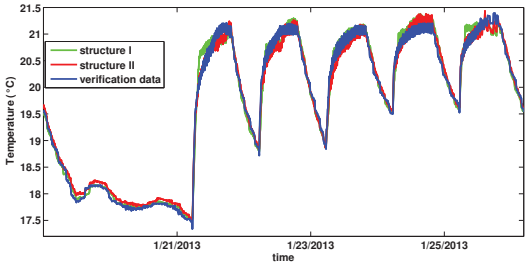

Fig. 2. Structure I. vs. Structure II.

C. Control requirements

To control the room temperature, nonlinear model predictive controller is exploited for which the minimization criterion directly corresponds with the control specifications as follows.

The most important part of the temperature control is thermal comfort satisfaction. In this case, the thermally

comfortable situation is specified such that $T_Z \geq \underline{T}_Z$. As usual in case of office buildings, the lower threshold for the room temperature is given as

$$\underline{T}_Z = \begin{cases} 22^\circ\,\mathrm{C} & \text{from } 8\,\mathrm{a.m.} \text{ to } 6\,\mathrm{p.m.}, \\ 20^\circ\,\mathrm{C} & \text{otherwise.} \end{cases} \quad (4)$$

Then, the cost criterion minimization term corresponding to the underheating is defined as comfort violation

$$CV = \max(0, \underline{T}_Z - T_Z). \quad (5)$$

To provide fresh air during the working hours, the following restrictions on the maximal mixing ratio δ and minimal supply air mass flow rate $\dot{m}$ are imposed:

$$\{\overline{\delta}, \underline{\dot{m}}\} = \begin{cases} \{0.7,\, 0.1\} & \text{from } 7:30 \text{ a.m. to } 6\,\mathrm{p.m.}, \\ \{1,\, 0\} & \text{otherwise.} \end{cases} \quad (6)$$

Obviously, certain minimal air recirculation with some predefined outdoor air content is required during the working hours. No further requirements for the air quality are provided.

The second very important aspect of the temperature control is its economical part. As can be seen from the description of the AHU performance provided in the previous Subsection, there are two procedures that consume energy: heating the mixed air from temperature T_m to T_S and sending the air into the room at desired $\dot{m}$. The first one is then expressed as

$$E_H \propto \dot{m}(T_S - T_m) \quad (7)$$

while the fan power is expressed as

$$E_F \propto \dot{m}^3. \quad (8)$$

The overall evaluative criterion is obtained as a weighted linear combination of (5), (7) and (8) summed over chosen period P,

$$J = \sum^{P} \omega_1 \dot{m}^3 + \omega_2 \dot{m}(T_S - T_m) + \omega_3 CV. \quad (9)$$

In this paper, $\omega_1 = 0.5$, $\omega_2 = 1$ and $\omega_3 = 30$ are used.

To complete the control requirements specification, manipulated variables constraints need to be provided:

$$\begin{aligned} T_S &\in \langle T_m,\, 30^\circ C \rangle, \\ \dot{m} &\in \dot{M}_{\mathrm{adm}}, \\ \delta &\in \langle 0,\, \overline{\delta} \rangle. \end{aligned} \quad (10)$$

It should be noted that in accordance with the AHU description provided earlier, $\dot{M}_{\mathrm{adm}}$ is a set of discrete values defined as

$$\dot{M}_{\mathrm{adm}} = \{\dot{m}_{\mathrm{a}} | \dot{m}_{\mathrm{a}} \in \langle \underline{\dot{m}}, 0.52 \rangle, \dot{m}_{\mathrm{a}} = a \times Q,\, a \in \mathbb{Z}\}, \quad (11)$$

where Q specifies the quantization step of the fan.

III. OPTIMIZATION

Having defined the cost criterion J over chosen prediction horizon P with J generally given as

$$J = \sum_{k=1}^{P} L(x_k, u_{c,k}, u_{q,k}), \quad (12)$$

the related optimization task can be summarized as follows:

$$\text{find } \{u_c^*, u_q^*\} = \arg\min J \quad (13)$$

such that

$$\begin{aligned} x_{k+1} &= f(x_k, u_{c,k}, u_{q,k}) \\ u_{c,k} &\in \langle \underline{u_c}, \overline{u_c} \rangle \\ u_{q,k} &\in U_{\mathrm{adm}}, \end{aligned} \quad (14)$$

where u_c are the continuous-valued inputs, u_q are quantized (discrete-valued) inputs and $f(x_k, u_{c,k}, u_{q,k})$ represents the dynamics of the controlled system. Here, the admissible set of quantized input values $U_{q,\mathrm{adm}}$ is defined as

$$U_{q,\mathrm{adm}} = \{u_{q,\mathrm{adm}} | u_{q,\mathrm{adm}} \in \langle \underline{u_q}, \overline{u_q} \rangle, u_{q,\mathrm{adm}} = m \times Q\}, \quad (15)$$

where $Q \in \mathbb{R}^+$ is the chosen quantization step and $m \in \mathbb{Z}$ represents multiple of the quantization step. Without loss of generality, it can be assumed that for both $\underline{u_q}$ and $\overline{u_q}$, there exist such $a,\, b \in \mathbb{Z}$ that

$$\underline{u_q} = a \times Q,\ \overline{u_q} = b \times Q.$$

If the original constraints for u_q do not satisfy this condition, the original admissible input interval $\langle \underline{u_q}, \overline{u_q} \rangle$ can be contracted to $\langle \underline{u_{q,n}}, \overline{u_{q,n}} \rangle$, where

$$\underline{u_{q,n}} = \min\{u_{q,n} = a \times Q | u_{q,n} \geq \underline{u_q},\, a \in \mathbb{Z}\}$$

and

$$\overline{u_{q,n}} = \max\{u_{q,n} = b \times Q | u_{q,n} \leq \overline{u_q},\, b \in \mathbb{Z}\}.$$

To solve the task (13) with respect to the constraints (14–15), adaptation of the well-known Hamiltonian-based gradient optimization method is proposed.

The original gradient method [19], [20] can be described as follows: starting from the initial guess of the optimal input sequence u^0, the following iterative search is performed:

$$u^l = u^{l-1} - \alpha \frac{\partial H}{\partial u}, \quad (16)$$

where l specifies the number of iteration of the gradient algorithm. The Hamiltonian H is created as

$$H = L(x_k, u_{c,k}, u_{q,k}) + \lambda_{k+1}^{\mathrm{T}} f(x_k, u_{c,k}, u_{q,k}) \quad (17)$$

where λ is a costate vector with backwards dynamics

$$\lambda_k = \frac{\partial H}{\partial x}(x_k, u_{c,k}, u_{q,k}, \lambda_{k+1}) \quad (18)$$

and terminal condition

$$\lambda_P = \frac{\partial J}{\partial x}|_P. \quad (19)$$

At each iteration l, the obtained input profile u^l is projected on the admissible input interval $\langle \underline{u}, \overline{u} \rangle$, which enforces satisfaction of input constraints. The procedure (16) is terminated if $|J(u^l) - J(u^{l-1})| \leq \epsilon$ for a properly chosen sensitivity $\epsilon > 0$.

It is obvious that the above described procedure results in continuous-valued sequence $\{u_c, \hat{u}_q\}$. To obtain discrete-valued input sequence u_q, additional postprocessing can be performed. The most straightforward postprocessing consists in rounding the obtained continuous-valued input sequence $\hat{u}_q$ away from zero to the nearest multiple of the quantization step,

$$u_q = Q \cdot \mathrm{round}\left(\frac{\hat{u}_q}{Q}\right), \quad (20)$$

where $\mathrm{round}(\cdot) = \mathrm{sgn}(\cdot)\lceil |\cdot| \rceil$. The main disadvantage of such naive postprocessing is that it often leads to high-frequency oscillations of the input signals. Such oscillations can be

ineffective from the economical point of view and moreover might result in increase of actuator abrasion and decrease of its lifetime.

To overcome these issues, the adaptation of the original Hamiltonian-based method is proposed in the sequel. The key part of the adaptation is to include a regular *mid-processing* each $\mathbf{I}$-th iteration, $\mathbf{I} \in \mathbb{N}^+$. The *mid-processing iteration* is then each iteration l for which $l = m \times \mathbf{I}$, $m \in \mathbb{N}^+$.

During the *mid-processing iteration*, the following procedure is performed: at first, continuous-valued sequence $\hat{u}_q^l$ is rounded towards the nearest multiple of Q to obtain $u_{q,\circ}^l$,

$$u_{q,\circ}^l = Q \cdot \text{round}\left(\frac{\hat{u}_q^l}{Q}\right). \tag{21}$$

Then, sequence of n_f past and P future input samples $\overleftrightarrow{u_q}^l = [\overleftarrow{u_q}, u_{q,\circ}^l]$ is created. Here,

$$\overleftarrow{u_q} = [u_{q,k-n_f}, u_{q,k-(n_f-1)}, \ldots, u_{q,k-2}, u_{q,k-1}] \tag{22}$$

represents the last n_f samples that have been applied to the system. Then, $\overleftrightarrow{u_q}^l$ is filtered with a user-defined low-pass filter with order n_f suppressing the undesired high frequencies in the resulting sequence $\overleftrightarrow{u_{q,\text{filt}}}^l$. Last of all, the filtered sequence $\overleftrightarrow{u_{q,\text{filt}}}^l$ is quantized,

$$\overleftrightarrow{u_{q,\text{filt},\circ}}^l = Q \cdot \text{round}\left(\frac{\overleftrightarrow{u_{q,\text{filt}}}^l}{Q}\right) \tag{23}$$

and the last P-samples subsequence is extracted and used for the next iteration,

$$u_q^l = [\overleftrightarrow{u_{q,\text{filt},\circ,k}}^l, \overleftrightarrow{u_{q,\text{filt},\circ,k+1}}^l, \ldots, \overleftrightarrow{u_{q,\text{filt},\circ,k+P}}^l]. \tag{24}$$

The overall optimization procedure is then summarized as follows:

Algorithm quaNPC

1) using the given initial condition x_0 and inputs from the previous iteration $\{u_c^{l-1}, u_q^{l-1}\}$, obtain state trajectories $X = [x_0, x_1, \ldots, x_P]$ such that $x_{k+1} = f(x_k, u_{c,k}^{l-1}, u_{q,k}^{l-1})$;
2) using the state trajectories X and inputs $\{u_c^{l-1}, u_q^{l-1}\}$, obtain the co-state trajectory $\Lambda = [\lambda_0, \lambda_1, \ldots, \lambda_P]$ with terminal condition (19) and dynamics (18);
3) calculate gradient $\partial H/\partial u$ and perform gradient step (16), obtain u_c^l and $\hat{u}_q^l$;
4) **if** $\mod(l, \mathbf{I}) = 0$
   then perform the *mid-processing*:
   i) quantize $\hat{u}_q^l$ according to (21) with chosen Q, obtain $u_{q,\circ}^l$,
   ii) create sequence $\overleftrightarrow{u_q}^l = [\overleftarrow{u_q}, u_{q,\circ}^l]$,
   iii) filter $\overleftrightarrow{u_q}^l$ using a low-pass filter with chosen characteristics and order n_f, obtain $\overleftrightarrow{u_{q,\text{filt}}}^l$,
   iv) quantize $\overleftrightarrow{u_{q,\text{filt}}}^l$ with chosen Q, obtain $\overleftrightarrow{u_{q,\text{filt},\circ}}^l$,
   v) obtain u_q^l as the last P samples of $\overleftrightarrow{u_{q,\text{filt},\circ}}^l$;
   else $u_q^l = \hat{u}_q^l$;
5) project the sequences u_c^l and u_q^l on the admissible intervals $\langle \underline{u_c}, \overline{u_c} \rangle$ and $\langle \underline{u_q}, \overline{u_q} \rangle$;
6) **if** $|J(u^l) - J(u^{l-\mathbf{I}})| \leq \epsilon$
   then terminate,
   else $l = l + 1$, repeat from 1).

The performance of the quaNPC algorithm is inspected in the following Section where a comparison with the naive postprocessing and the original continuous-valued nonlinear MPC is provided.

IV. RESULTS

To examine all three alternatives (original continuous-valued NMPC, apqNMPC exploiting the a posteriori quantization and the newly proposed quaNPC), a numerical experiment corresponding to a two-week period was performed. In this case, $Q = \overline{m}/4$ was chosen which results in a mass flow rate profile with 5 admissible quantization levels.

Fig. 3, 4 and 5 show zone temperature, supply air mass flow rate and supply air temperature profiles for the three inspected variants. Since there are no other requirements for the air quality than just the highest allowed indoor/outdoor air mixing ratio $\overline{\delta}$ and given that the ambient temperature is always lower than the indoor temperature (which in Michigan holds fairly well), it is obvious that the optimal value of mixing ratio is always $\delta^* = \overline{\delta}$. Therefore, figure showing profiles of δ is unnecessary.

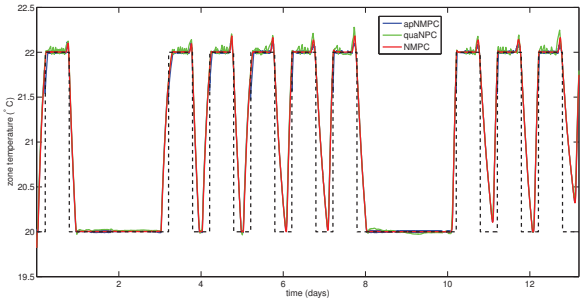

Fig. 3. Zone temperature.

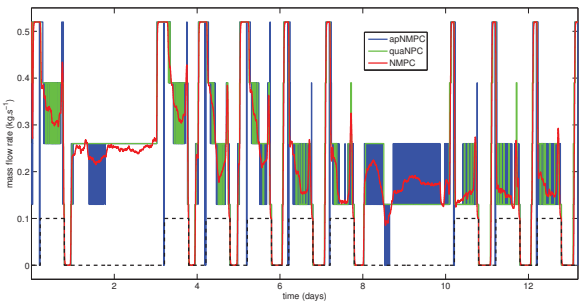

Fig. 4. Supply air mass flow rate.

While the thermal comfort performance is fairly similar and can be considered satisfactory for all three controllers, significant contrasts can be observed by inspecting manipulation of the input profiles. Although mass flow rate profile (see Fig. 4) for quaNPC is rather flat with only occasional oscillations between two neighboring quantization steps, apqNMPC mass flow rate profile oscillates much more. Fig. 6 shows a detailed picture of what seems to be more oscillatory part of the quaNPC mass flow rate. Exploring the zoomed part it can be seen that although some increase of oscillations occurs

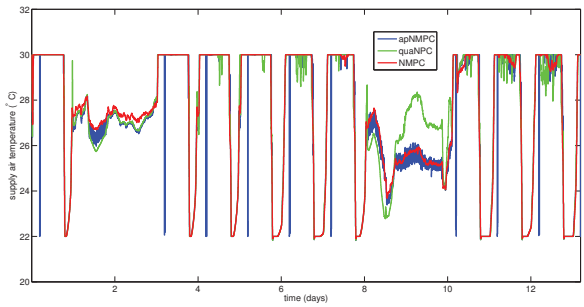

Fig. 5. Supply air temperature.

indeed for quaNPC algorithm, the number of oscillations is still negligible compared with the apqNMPC. This is supported also by Fig. 7, where frequency spectra of the mass flow rate profiles for all three controllers are depicted. Here, f_N is the normalized frequency.

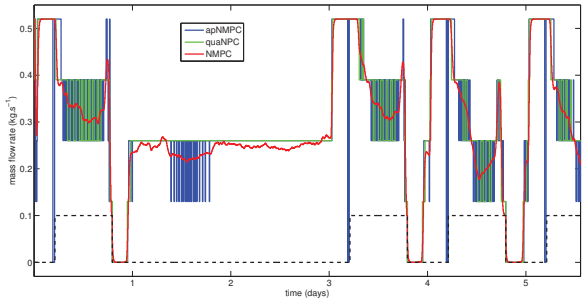

Fig. 6. Supply air mass flow rate – detail.

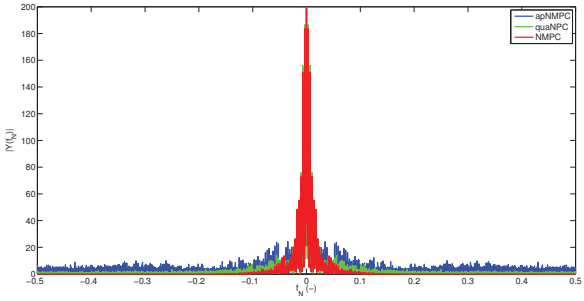

Fig. 7. Frequency spectra of mass flow rate profiles.

Since the controllers calculate not only the supply air mass flow rate $\dot{m}$, it might be interesting to look at the behavior of the other calculated input profiles. Looking at Fig. 5, it is evident that a posteriori quantization increase oscillations not only in the mass flow rate profile but in supply air temperature profile as well. On the other hand, according to Fig. 4, 5 and 7, it can be concluded that the efforts to decrease high-frequency content of the input signal by making use of quaNPC was successful and it positively affect not only the quantized variable but the other optimized variables as well.

TABLE I

TWO-WEEK COMPARISON OF THE CONTROLLERS.

N_q	$I_{J,r}$(%) apqNMPC	$I_{J,r}$(%) quaNPC	RSN (–)
2	369	68	24.4
4	95	17	6.3
6	86	7	5.0
8	76	5	4.0
10	69	2	3.7
21	59	0	2.8

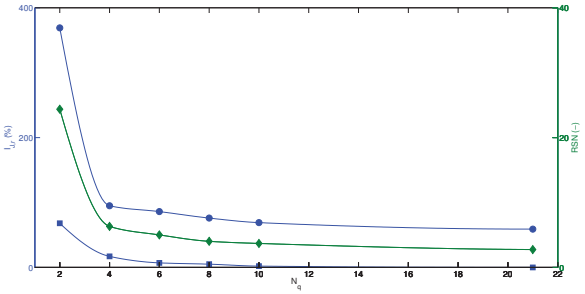

Fig. 8. Two-week comparison, multiple N_q (● - $I_{J,r}$ of apqNMPC, ■ - $I_{J,r}$ of quaNPC, ♦ - RSN).

In Tab. I, the numerical comparison of the controllers is provided for various different numbers N_q of considered quantization levels. For sake of completeness, also one very high value of N_q was inspected. As already indicated, the main objective of proposing a new algorithm was to both reduce the undesired oscillations of the discrete-valued input and keep the performance close to the continuous-valued predictive controller. Therefore, two evaluators are proposed as follows. $I_{J,r}$ represents the relative increase of the cost criterion normalized with respect to the original continuous-valued NMPC while RSN specifies the ratio of the apqNMPC U-D mass flow rate switchings versus quaNPC U-D mass flow rate switchings. For this purpose, the *U-D* (up-and-down) *switching* is defined as number of sampling periods at which two consecutive differences of the discrete-valued input have opposite signs. For better illustration, the numerical values from Tab. I are graphically presented in Fig. 8.

The more complete comparison brought by Tab. I and Fig. 8 demonstrates what was already indicated by the experiment with 5-valued mass flow rate. Just pure a posteriori quantization deteriorates control performance significantly. For the finest considered quantization with 21-valued input, the quaNPC algorithm is able to ensure performance which is equivalent to the original continuous-valued NMPC in the sense of the cost criterion value, while the value of the evaluative criterion for the apqNMPC is still almost 60 % higher than for the original NMPC. Decreasing the number of the admissible input values N_q of the quantized input, the contrast between the apqNMPC and quaNPC is even more significant. With the most coarse quantizing $N_q = 2$, increase of cost criterion of quaNPC is 68 % which is about the same as achieved by apqNMPC with $N_q = 10$. On the other hand, apqNMPC with coarsest quantization causes cost criterion to be almost 370 % higher than for the continuous-valued NMPC.

The values of RSN listed in Tab. I prove that the reduction of the oscillatory behavior of the quantized input was success-

fully accomplished for other number of quantization steps as well. As can be expected, the relative reduction decreases with finer quantization but also in case of the finest one ($N_q = 21$), the mass flow rate calculated by quaNPC oscillates almost 3 times less than the one provided by apqNMPC. The trend sketched in Fig. 8 shows that when reducing the number of admissible values of supply air mass flow rate, the ratio between apqNMPC oscillations and quaNPC oscillations rises nearly exponentially from almost 3 ($N_q = 21$) to more than 24 ($N_q = 2$) which clearly demonstrates that (with less quantization steps) quaNPC is much more considerate of the actuators than apqNMPC.

For the practical applications, also time consumption of the optimization procedure is important. The apqNMPC approach exploits the same optimization procedure as in case of original continuous-valued NMPC and since the a posteriori quantization can be performed using a hardware solution, this alternative does not bring any increase of computational time compared to the original NMPC. On the other hand, quaNPC procedure increases the duration of the calculations by $0.25\,\mathrm{s}$ which corresponds to 19% of time consumed by the continuous-valued NMPC. Here it should be remarked that this percentual increase is constant and is independent of number of quantization steps which is not the case for mixed integer nonlinear programming tasks where the time needed to find the solution often grows exponentially.

Combination of attractive performance approaching the continuous-valued NMPC standard, more careful manipulation of the discrete-valued inputs and nondemanding implementation and calculations makes quaNPC proposed in the current paper a suitable candidate for the application in industrial practice.

V. CONCLUSION

In the current paper, the task of nonlinear model predictive control with both continuous-valued and discrete-valued inputs was handled. An alternative to the commonly used a posteriori quantization of the optimized input values was proposed. Rather than using a completely independent and separate post-processing of the calculated values, the quantization aspects are incorporated directly into the developed quaNPC optimization algorithm.

The comparison of the quaNPC algorithm with the a-posteriori-quantizing approach shows significant reduction of quantized-input oscillations for quaNPC demonstrated by both lower number of U-D switchings (quaNPC input profiles oscillate $2.8 - 24.4$ times less than apqNMPC profiles for the examined range of numbers of quantization steps) and input frequency spectrum with lower content of higher frequencies. Regarding the cost criterion, application of quaNPC results in much smaller deviations from the performance of the continuous-valued NMPC than just naive a posteriori quantization. From certain reasonable number of quantization steps $N_q = 6$, the quaNPC performance deviates only marginally from the NMPC performance ($7\,\%$ and less; $0\,\%$ degradation for the finest tested quantization) while performance of apqNMPC is much worse even up to a very fine quantization (control performance degradation of $59 - 86\,\%$). The limit case $N_q = 2$ (basically an on-off control) shows that although the a posteriori quantization is absolutely unsatisfactory from the control performance point of view, the quaNPC algorithm is able to keep the control performance within admissible limits. The comparison therefore proves that quaNPC is a promising candidate for the utilization in the area of industrial control.

VI. ACKNOWLEDGEMENT

The authors would like to thank Meysam Razmara, Mahdi Shahbakhti (Michigan Technical University, Houghton) and Mehdi Maasoumy (University of California, Berkeley) for providing the authors with the data measured in the Lakeshore building and for their support when discussing issues related to the paper. This research has been supported by the Czech Science Foundation through the grants no. 13-20433S and 13-12726J.

REFERENCES

[1] S. Del Favero, D. Bruttomesso, F. Di Palma, G. Lanzola, R. Visentin, A. Filippi, R. Scotton, C. Toffanin, M. Messori, S. Scarpellini *et al.*, "First use of model predictive control in outpatient wearable artificial pancreas," *Diabetes care*, vol. 37, no. 5, pp. 1212–1215, 2014.

[2] B. Hredzak, V. G. Agelidis, and M. Jang, "A model predictive control system for a hybrid battery-ultracapacitor power source," *Power Electronics, IEEE Transactions on*, vol. 29, no. 3, pp. 1469–1479, 2014.

[3] D. Schlipf, D. J. Schlipf, and M. Kühn, "Nonlinear model predictive control of wind turbines using lidar," *Wind Energy*, vol. 16, no. 7, pp. 1107–1129, 2013.

[4] M. A. Perez, J. Rodriguez, E. J. Fuentes, and F. Kammerer, "Predictive control of ac–ac modular multilevel converters," *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 7, pp. 2832–2839, 2012.

[5] M. Sourbron, C. Verhelst, and L. Helsen, "Building models for model predictive control of office buildings with concrete core activation," *Journal of building performance simulation*, vol. 6, no. 3, pp. 175–198, 2013.

[6] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, "Use of model predictive control and weather forecasts for energy efficient building climate control," *Energy and Buildings*, vol. 45, pp. 15–27, 2012.

[7] M. Wallace, R. McBride, S. Aumi, P. Mhaskar, J. House, and T. Salsbury, "Energy efficient model predictive building temperature control," *Chemical Engineering Science*, vol. 69, no. 1, pp. 45–58, 2012.

[8] S. Yuan and R. Perez, "Multiple-zone ventilation and temperature control of a single-duct vav system using model predictive strategy," *Energy and buildings*, vol. 38, no. 10, pp. 1248–1261, 2006.

[9] H. Karlsson and C.-E. Hagentoft, "Application of model based predictive control for water-based floor heating in low energy residential buildings," *Building and environment*, vol. 46, no. 3, pp. 556–569, 2011.

[10] M. Kummert, P. André, and J. Nicolas, "Optimal heating control in a passive solar commercial building," *Solar Energy*, vol. 69, pp. 103–116, 2001.

[11] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," *Control Systems Technology, IEEE Transactions on*, vol. 20, no. 3, pp. 796–803, 2012.

[12] S. Pancanti, L. Leonardi, L. Pallottino, and A. Bicchi, "Optimal control of quantized input systems," in *Hybrid Systems: Computation and Control*. Springer, 2002, pp. 351–363.

[13] T. Geyer, M. Larsson, and M. Morari, "Hybrid emergency voltage control in power systems," in *Proceedings of the European Control Conference 2003*, 2003.

[14] H. W. Lenstra Jr, "Integer programming with a fixed number of variables," *Mathematics of operations research*, vol. 8, no. 4, pp. 538–548, 1983.

[15] M. R. Bussieck and S. Vigerske, "Minlp solver software," *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[16] M. Maasoumy, M. Razmara, M. Shahbakhti, and A. S. Vincentelli, "Handling model uncertainty in model predictive control for energy efficient buildings," *Energy and Buildings*, vol. 77, pp. 377–392, 2014.

[17] D. Laurí, J. Salcedo, S. Garcia-Nieto, and M. Martínez, "Model predictive control relevant identification: multiple input multiple output against multiple input single output," *Control Theory & Applications, IET*, vol. 4, no. 9, pp. 1756–1766, 2010.

[18] E. Žáčeková, Z. Váňa, and J. Cigler, "Towards the real-life implementation of mpc for an office building: Identification issues," *Applied Energy*, vol. 135, pp. 53–62, 2014.

[19] K. Zhou, J. C. Doyle, K. Glover *et al.*, *Robust and optimal control*. Prentice Hall New Jersey, 1996, vol. 40.

[20] A. E. Bryson, *Applied optimal control: optimization, estimation, and control*. Taylor and Francis, 1975.

Control Engineering Practice 53 (2016) 124–138

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Bridging the gap between the linear and nonlinear predictive control: Adaptations for efficient building climate control

Matej Pčolka [a,*], Eva Žáčeková [a,*], Rush Robinett [b], Sergej Čelikovský [c], Michael Šebek [a]

[a] Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Praha 6, Czech Republic
[b] Mechanical Engineering-Engineering Mechanics, Michigan Technological University, United States
[c] Institute of Information Theory and Automation, Czech Academy of Sciences, Czech Republic

ARTICLE INFO

Article history:
Received 7 June 2015
Received in revised form
23 January 2016
Accepted 24 January 2016
Available online 5 February 2016

Keywords:
Model predictive control
Identification for control
Building climate control

ABSTRACT

The linear model predictive control which is frequently used for building climate control benefits from the fact that the resulting optimization task is convex (thus easily and quickly solvable). On the other hand, the nonlinear model predictive control enables the use of a more detailed nonlinear model and it takes advantage of the fact that it addresses the optimization task more directly, however, it requires a more computationally complex algorithm for solving the non-convex optimization problem. In this paper, the gap between the linear and the nonlinear one is bridged by introducing a predictive controller with linear time-dependent model. Making use of linear time-dependent model of the building, the newly proposed controller obtains predictions which are closer to reality than those of linear time invariant model, however, the computational complexity is still kept low since the optimization task remains convex. The concept of linear time-dependent predictive controller is verified on a set of numerical experiments performed using a high fidelity model created in a building simulation environment and compared to the previously mentioned alternatives. Furthermore, the model for the nonlinear variant is identified using an adaptation of the existing model predictive control relevant identification method and the optimization algorithm for the nonlinear predictive controller is adapted such that it can handle also restrictions on discrete-valued nature of the manipulated variables. The presented comparisons show that the current adaptations lead to more efficient building climate control.

1. Introduction

Presently, energy savings and reduction of energy consumption in buildings are some of the most challenging issues facing the engineering community. The reason is straightforward and the numbers speak for themselves – up to 40% of the total energy consumption can be owed to the building sector (Perez-Lombard, Ortiz, & Pout, 2008). More than half of this 40% is consumed by various building heating/cooling systems. Therefore, the recent significant emphasis on the energy savings in this area is right on target and can be observed in recent years. For example, the strategy of the European Union called "20–20–20" (European Economic & Social Committee, 2005) should be mentioned. Intended to be followed by all of Europe through the year 2020, this strategy aims at 20% reduction of the use of primary energy sources and production of the greenhouse gas emissions, and the renewable energy sources are expected to provide 20% of the consumed energy. With the clearly evident need for savings in the area of the building climate control, improvements can be found when considering the latest control techniques.

Model Predictive Control (MPC) is one of the most promising candidates for an energetically efficient control strategy (Pčolka, Žáčeková, Robinett, Čelikovský, & Šebek, 2014a, 2014b). This was also demonstrated within the framework of the Opticontrol project. One research team at ETH Zurich (Switzerland) showed via numerous simulations that using MPC instead of the classical control strategies achieves more than 16% savings (Gyalistras & Gwerder, 2010; Oldewurtel et al., 2010) depending on the building type. If one considers real operational conditions, these savings can be even higher when the MPC is modified appropriately for the conditions. This was shown by teams from Prague (Prívara, Široký, Ferkl, & Cigler, 2011; Žáčeková & Prívara, 2012 and UC Berkeley (Ma, Kelman, Daly, & Borrelli, 2012) where the actual cost savings were even better than the theoretical expectations (27% and 25% reduction of the energy consumption, respectively).

However, MPC suffers from several drawbacks including the complexity of the optimization routine and the need for a reliable mathematical model of the building. In order to be feasible and

* Corresponding authors.
E-mail addresses: matej.pcolka@fel.cvut.cz (M. Pčolka), eva.zacekova@fel.cvut.cz (E. Žáčeková), rdrobine@mtu.edu (R. Robinett), celikovs@utia.cas.cz (S. Čelikovský), sebekm1@fel.cvut.cz (M. Šebek).

http://dx.doi.org/10.1016/j.conengprac.2016.01.007
0967-0661/

computable, simplified formulations are often considered. Moreover, linear models are usually assumed and exploited by the optimizer. Therefore, in the majority of the MPC applications, the overall task is formulated as a linear/convex optimization problem easily solvable by the commonly available solvers for quadratic or semidefinite programming (Verhelst, Degrauwe, Logist, Van Impe, & Helsen, 2012; Prívara et al., 2011). Although being computationally favorable and able to find the global minimum in case of the convex formulation of the optimization task, their disadvantage is that they do not enable minimization of the nonlinear/nonconvex cost criteria and therefore, only certain approximation of the real cost paid for the control is optimized. Moreover, they resort to the optimization of either the setpoints or the energy delivered to the heating/cooling system while leaving all its distribution to the suboptimal low-level controllers which can lead to a significant loss of the optimality gained by the MPC.

In several recent works, the effort to take the nonlinearities (caused either by the dynamical behavior of the building or by the control requirements formulation) into account within the optimization task can be found (Ma et al., 2012, 2011). In this paper, we discuss both possibilities for the zone temperature control (the linear and the nonlinear MPC) and moreover, we bridge the two banks of the gap between the nonlinear and the linear variant of the MPC by introducing linear model that changes in time. Such model can describe the building dynamics in a more reliable and flexible way than the original linear model while it still keeps the low complexity of the optimization task (since with the linear model, the optimization task to be solved remains convex). The way of obtaining a time-varying model is described and the results of the linear predictive controller with linear model that changes in time are compared with the results of the original (linear and nonlinear) MPCs.

It should be mentioned that a good predictive controller relies on a good system dynamics predictor and therefore, we focus on the identification of such reliable multi-step predictors as well. The MPC employs optimization over certain given prediction horizon and this fact should be taken into consideration also in the design of the identification procedure. Unlike the commonly used identification methods (PEM, Ljung, 1999) which provide models that are able to predict well only over short horizons, the methods based on minimization of multi-step prediction errors (MRI – model predictive control relevant identification, Laurí, Salcedo, Garcia-Nieto, & Martínez, 2010) offer models with more attractive prediction properties. Therefore, we exploit the MRI for identification of both linear and nonlinear models. While several published works deal with application of MRI for estimation of parameters of linear models (Chi, Fei, Zhao, Zhao, & Liang, 2014; Shook, Mohtadi, & Shah, 1991; Zhao, Zhu, & Patwardhan, 2014), no extension, to the best knowledge of the authors of this paper, has been provided for estimation of parameters of *nonlinear* models. Moreover, even the linear version of MRI in the literature is usually validated only on simple artificial examples. On the other hand, this paper presents application of both the linear and the newly proposed nonlinear MRI versions on much more complex and realistic example of building model identification.

Furthermore, a very important practical aspect of the building temperature control is addressed in this work as well. In real-life building applications, water pumps are a crucial part of the actuators used to manipulate the optimized input variables. These water pumps possess nonlinear output dynamics where the amount of mass flow rate which can be provided by the pump is often quantized. Therefore, the achievable water mass flow rates belong to a countable set of discrete values rather than to a continuous interval. The appropriately designed control algorithm should take this information properly into account. This can be performed in several ways: (1) mixed-integer programming techniques can be employed, (2) additional postprocessing after the calculation of the optimal inputs can be applied, or (3) the (originally continuous-valued) optimization procedure itself can be adapted such that discrete-valued input profiles are obtained.

First of all, the mixed-integer programming approach is the most suitable one in case that one of the manipulated variables should belong to countable set of discrete values. However, the mixed-integer programming problems are known to be NP-hard (Bussieck & Vigerske, 2010; Lenstra, 1983; Pancanti, Leonardi, Pallottino, & Bicchi, 2002) and their solution using mixed-integer programming solvers requires massive computational power. Furthermore, the majority of reliable currently available mixed-integer solvers able to handle nonlinear system description/nonlinear optimization criterion are not free for industrial use. Since the computational burden caused by solving the mixed-integer programming task is huge and it is in direct opposite to the extensive effort to simplify the control schemes and systems used in buildings, this direction is not suitable. Instead of formulating the building temperature control problem as a mixed-integer programming task, the other two mentioned options (additional postprocessing and adaptation of the continuous-valued optimization procedure) are elaborated in the current paper.

The paper is organized as follows: Section 2 illustrates the problem of the building climate control on a simple example. Both the building and the heat delivery system description are provided. Furthermore, control performance criterion, comfort requirements and restrictions are introduced. In Section 3, the models supplying predictions to the model-based controllers are described. The nonlinear model is derived in Section 3.1 based on the thermodynamics while for the linear model, the assumed simplifications are presented in Section 3.2. The linear time-varying model is presented in Section 3.3. A new approach to estimating parameters of the nonlinear model with respect to the multi-step prediction error minimization criterion proposed in Section 3.4. Two alternative versions of this approach are presented which are some of the main contributions of this paper. All models are verified on the data set obtained from TRNSYS environment and their results are discussed. Section 4 describes the controllers including the low level re-calculation (for the linear MPC) and the nonlinear optimization routine (for the nonlinear MPC). In order to address the discrete-valued nature of part of the considered actuators, the nonlinear MPC optimization routine is changed in two ways: either a naive additional post-processing is employed or the *mid-processing* iteration (which is another main contribution of this paper) is incorporated into the routine. In Section 5, building behaviors of all proposed controllers are investigated and their results are presented and examined. Section 6 draws conclusion of the paper.

2. Problem formulation

In this section, the description of the building, constraints and the evaluative performance criterion are formulated.

2.1. Building of interest

The building under our investigation is a simple medium weight one-zone building modeled in the TRNSYS16 (University of Wisconsin-Madison, 1979) environment, which is a high fidelity simulation software package widely accepted by the civil engineering community as a reliable tool for simulating the building behavior.

The building considered in this paper is a medium sized one with a size of $5 \times 5 \times 3$ m and a single-glazed window (3.75 m^2) placed in the south-oriented wall. The Heating, Ventilation and Air

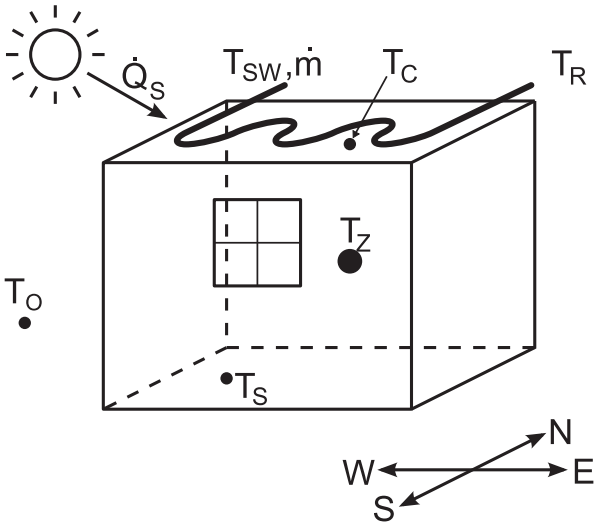

Fig. 1. A scheme of the modeled building.

Conditioning (HVAC) system used in the building is of the so-called active layer type. Technically, the HVAC system consists of TABS (thermally activated building system) – a set of metal pipes encapsulated into the ceiling distributing the supply water which then enables thermal exchange with the concrete core of the modeled building consequently heating the air in the room. This configuration corresponds to the commonly used building heating system in the Czech Republic. Ambient environmental conditions (ambient temperature, ambient air relative humidity, solar radiation intensity and others) are simulated using TRNSYS Type15 with the yearly weather profile corresponding to Prague, Czech Republic.

Fig. 1 shows a sketch of the building HVAC system configuration, the "building" variables and the environment variables. Regarding the building inner variables, four of them are considered to be available – zone temperature T_Z, ceiling temperature T_C, temperature of the return water T_R and temperature of the south-oriented wall T_S. From the environmental influences, solar radiation $\dot{Q}_S$ and outside-air temperature T_O are taken into account as disturbances while the supply water temperature T_{SW} and the mass flow rate of the supply water $\dot{m}$ are the controlled input variables. The TRNSYS model in this configuration offers a good numerical test-bed to compare the control approaches, and the results obtained with this model can be generalized without any loss of objectivity.

The next step is to describe the heat distribution system. In the application presented in this paper, the configuration of the heating system as shown in Fig. 2 is considered. Clearly, the storage tank plays a key role as the sole heat supplier in this system. In fact, having obtained the requirements for the supply water temperature T_{SW} and the supply water mass flow rate $\dot{m}$, these two values are "mixed" using the return water with the temperature T_R flowing into the building inlet pipe through the side-pipe at the mass flow rate $\dot{m}_S$ and the water from the storage tank which is kept at certain constant value T_{St} (in this paper, $T_{St} = 60\,^{\circ}\mathrm{C}$ is considered) and can be withdrawn from the tank at mass flow rate $\dot{m}_{St}$. Based on this, the following set of equations can be written for the upper three-way valve:

$$\begin{aligned} \dot{m}T_{SW} &= \dot{m}_{St}T_{St} + \dot{m}_S T_R \\ \dot{m} &= \dot{m}_{St} + \dot{m}_S. \end{aligned} \tag{1}$$

which can be further rewritten into an expression for the calculation of the storage water mass flow rate,

$$\dot{m}_{St} = \dot{m}\frac{(T_{SW} - T_R)}{(T_{St} - T_R)}. \tag{2}$$

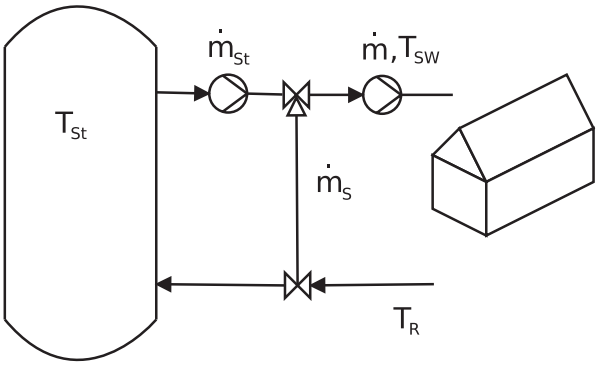

Fig. 2. A scheme of heat distribution system.

Having the return water temperature values at disposal and extracting the storage water with the temperature of T_{St} at the mass flow rate $\dot{m}_{St}$, both the supply water temperature and supply water mass flow rate related to the heating requirements can be achieved.

2.2. Control performance requirements

Considering the building climate control, one of the most important tasks is to ensure the required thermal comfort which is specified by a pre-defined admissible range of temperatures related to the way of use of the building (office building, factory, residential building, etc.). Under the weather conditions of middle Europe with quite low average temperatures where heating is required for more than half of year, the thermal comfort satisfaction requirement can be further simplified such that the zone temperature is bounded only from below. Since an office building with regular time schedule is considered, the lowest admissible zone temperature $T_Z^{min}(t)$ whose violation will be penalized is defined as a function of working hours as

$$T_Z^{min}(t) = \begin{cases} 22\,^{\circ}\mathrm{C} & \text{from 8 a. m. to 6 p. m.,} \\ 20\,^{\circ}\mathrm{C} & \text{otherwise.} \end{cases} \tag{3}$$

Then, the thermal comfort violation is expressed as

$$CV(t) = \max(0, T_Z^{min}(t) - T_Z(t)). \tag{4}$$

Besides the comfort violation $CV(t)$, the price paid for the operation of the building is penalized in the cost criterion as well. Coming out of the considered structure of the building and its energy supply system, the monetary cost includes the price for the consumed hot water and the electricity needed to operate the two water pumps. While the hot water price P_W is considered constant (see Table 1), the electricity price $P_E(t)$ which applies to the operation of the supply and storage water pumps is piece-wise constant and similar to the lowest admissible zone temperature

Table 1
List of the specific parameters.

T_Z^{min} (°C)	22/20	P_W (–)	2.6199
HT (€/kWh)	0.1168	α_0 (–)	9
LT (€/kWh)	0.0502	α_1 (–)	9.25×10^{-3}
T_{St} (°C)	60	α_2 (–)	1.875×10^{-6}
$[\underline{\dot{m}}, \overline{\dot{m}}]$	[15,60]	ΔT (°C)	5

profile, it depends on the working hours as follows:

$$P_E(t) = \begin{cases} HT & \text{from 8 a. m. to 6 p. m.,} \\ LT & \text{otherwise.} \end{cases} \quad (5)$$

In order to bring the presented case study closer to reality, the values of high and low tariff (*HT* and *LT*) have been chosen in accordance with the real prices approved by the Regulatory Office for Network Industries of Slovak Republic (R.O. for Network Industries, 2011). The exact values of *HT* and *LT* in €/kWh are listed in Table 1.

Thus, the overall performance criterion over a time interval $\langle t_1, t_2 \rangle$ is formulated as

$$J = \int_{t_1}^{t_2} \omega CV(t)\,\mathrm{d}t + \int_{t_1}^{t_2} \left(P_E(t)(P_C(\dot{m}) + P_C(\dot{m}_{St})) + P_W \dot{m}_{St}\right)\mathrm{d}t. \quad (6)$$

Here, ω is the virtual price for the comfort violation $CV(t)$ which is defined by Eq. (4) and $P_W \dot{m}_{St}$ represents the cost paid for the consumed hot water. Time-varying electricity price is expressed as a function of time by Eq. (5) and the power consumptions of the water pumps corresponding to $\dot{m}$ and $\dot{m}_{St}$ can be calculated as a quadratic function of the particular mass flow rate,

$$P_C(\dot{m}) = \alpha_0 + \alpha_1 \dot{m} + \alpha_2 \dot{m}^2,$$
$$P_C(\dot{m}_{St}) = \alpha_0 + \alpha_1 \dot{m}_{St} + \alpha_2 \dot{m}_{St}^2. \quad (7)$$

The parameters $\alpha_{0,1,2}$ are listed in Table 1.

Let us note that since the criterion (6) specifies the control requirements for the control of a building in a very compact form, all considered controllers will be evaluated and compared according to this criterion.

2.3. Constraints

In order to ensure proper functionality of the heat distribution system depicted in Fig. 2, the following technical constraints imposed on the manipulated variables need to be taken into account.

First of all, the constraints on mass flow rates which can be achieved by both the supply water pump and storage water tank pump need to be respected. The upper bound of the mass flow rates is given by the maximal power of the considered pumps. Technically, the lower bound on the supply water mass flow rate $\underline{\dot{m}}$ and storage tank mass flow rate $\underline{\dot{m}}_{St}$ is zero, however, the supply water pump is required to always maintain some nonzero supply water mass flow rate. To prevent the supply water pump from damage resulting from water overpressure potentially caused by the storage tank pump, the storage tank mass flow rate must never exceed the supply water mass flow rate. Due to this, the mass flow rate of the supply water and the storage tank mass flow rate are bound together by the relation $\dot{m}_{St} \leq \dot{m}$. The last mass flow rate constraint results from a common feature of the water pumps that are very often multi-valued and cannot set the mass flow rate with arbitrarily small sensitivity. Therefore, the mass flow rate values must belong to a countable admissible set of discrete values.

The second group of constraints is imposed on the supply water temperature. Since the storage tank is the only source of hot water and no additional heater that could increase the water temperature to values higher than T_{St} is considered, it is obvious that the highest required supply water temperature must be lower than or equal to storage water temperature. However, the heat losses caused by the transportation of the storage water should be also reflected and therefore, it is more realistic to consider the upper constraint for the supply water temperature to be several degrees lower than the storage water temperature. Last of all, let us note that a situation which requires a value of T_{SW} to be lower than the return water temperature T_R would mean negative storage water mass flow rate $\dot{m}_{St}$, which can not be practically realized. On the other hand, it is also obvious that such T_{SW} requirement really cannot be satisfied as only the hot water storage is considered in this configuration. With no cold water storage neither water chiller provided, the temperature of the supply water cannot be decreased below the return water temperature and the active cooling mode is not allowed.

Since the storage water mass flow rate is not an independent variable and is uniquely given by the supply water mass flow rate $\dot{m}$ and supply water temperature T_{SW}, the constraints for storage water mass flow rate can be omitted. To sum up, the above mentioned technical constraints are mathematically formulated as follows:

$$\underline{\dot{m}} \leq \dot{m} \leq \overline{\dot{m}}$$
$$\dot{m} \in \dot{M}_{\text{adm}} = \{\dot{m}_a | \dot{m}_a = a \times q_{st},\ a \in \mathbb{Z}\},$$
$$\max\{T_R, \underline{T}_{SW}\} \leq T_{SW} \leq T_{St} - \Delta T. \quad (8)$$

Parameters $\underline{\dot{m}}$, $\overline{\dot{m}}$ and ΔT are provided in Table 1. Several different values of quantization steps q_{st} were considered in this work and their exact values are specified later.

3. Modeling and identification

In this section, the derivation of models for the particular variants of the MPC (being one of the crucial part of the whole control approach) is described and explained. A special emphasis is put on explanation and description of Model Predictive Control Relevant Identification (MRI) approach, the identification procedure providing mathematical models with good prediction behavior on wider range of prediction horizons.

3.1. Nonlinear model (NM)

In the current paper, the methodology that is widely used for modeling of heat transfer effects in buildings (ASHRAE, 2009; Barták, 2010; Lienhard, 2013) is followed. As explained in the dedicated literature, several physical phenomena need to be considered to obtain an appropriate structure reliably describing the building behavior. The most crucial aspects influencing the thermodynamics within the inspected zone are:

1. *Convection from walls*: This phenomenon occurs when fluid (in this case the zone air) moves along the body (wall) with different surface temperature. It affects both the heated wall T_C and the unheated wall T_S and the zone temperature T_Z. Derived from the well known Newton's cooling law, the heat flux $q_{W,\text{conv}}$ caused by convection can be expressed as

   $$q_{W,\text{conv}} = h_{W,\text{conv}}(T_W - T_Z).$$

   In this expression, $h_{W,\text{conv}}$ denotes the convection heat transfer coefficient and T_W refers to temperature of one of the considered walls, i.e. T_C or T_S.
   In case that the fluid is externally forced to move, the convection heat transfer coefficient $h_{W,\text{conv}}$ is independent of the temperature difference $T_W - T_Z$. However, in case that the fluid motion is caused solely by buoyant forces arisen from different temperatures of the fluid and the body (and thus temperature-dependent density of the fluid) and the gravitational effects, the convection heat transfer coefficient $h_{W,\text{conv}}$ is expressed as a function of this temperature difference (ASHRAE, 2009; Lienhard, 2013). A common and empirically proven choice is to express the convection heat transfer coefficient $h_{W,\text{conv}}$ as a weak function of the temperature difference $\Delta T = T_W - T_Z$, typically $h_{W,\text{conv}} \propto |\Delta T^{1/4}|$ or $h_{W,\text{conv}} \propto |\Delta T^{1/3}|$ (Lienhard, 2013; Zmrhal & Drkal, 2006). Based on the technical specification of

the building examined in this paper (absence of the ventilation fan), the forced convection is neglected and the convection heat transfer coefficient is in accordance with ASHRAE (2009), Barták (2010), and Lienhard (2013) modeled as

$$h_{W,\mathrm{conv}} = \bar{h}_{W,\mathrm{conv}} \left| T_W - T_Z \right|^{\frac{1}{3}},$$

where $\bar{h}_{W,\mathrm{conv}}$ accounts also for influence of the surface area of the convecting wall $A_{\mathrm{W,conv}}$. Then, the convection heat flux $q_{W,\mathrm{conv}}$ from particular wall can be summarized as

$$q_{W,\mathrm{conv}} = \bar{h}_W \left| T_W - T_Z \right|^{\frac{1}{3}} (T_W - T_Z). \tag{9}$$

2. *Mutual interactions of the walls*: Out of the three possible heat transfer phenomena – conduction, radiation and convection –, the first two might apply when inspecting the mutual interactions between the considered walls (ASHRAE, 2009; Lienhard, 2013). Conduction heat flux q_{cond} occurs due to the presence of common edges and vertices of the walls and being the simpler one, it is expressed by a formula resembling Newton's cooling law (Balmer, 2010; Lienhard, 2013),

$$q_{W,\mathrm{cond}} = h_{\mathrm{cond}}(T_S - T_C).$$

Here, the conduction heat transfer coefficient $h_{W,\mathrm{cond}}$ is proportional to the surface area of the walls and inversely proportional to the distance between the points at which the temperatures T_C and T_S are provided.
Regarding the radiation, the well known Stefan–Boltzmann law applies:

$$q_{\mathrm{rad}} = h_{\mathrm{rad}}(T_S^4 - T_C^4),$$

with h_{rad} embracing (besides the effect of the Stefan–Boltzmann constant) various influences such as *view factor* between the two irradiating objects, emissivity/absorptivity and the surface area (Balmer, 2010). In case that the temperature difference between the two objects is relatively small (which holds true also for the heated and unheated wall temperatures), radiation heat flux q_{rad} can be with sufficient accuracy approximated by a linear function of the temperature difference,

$$q_{\mathrm{rad}} \approx \bar{q}_{\mathrm{rad}} = \bar{h}_{\mathrm{rad}}(T_S - T_C)$$

and the joint conduction/radiation heat flux can be then expressed as

$$q_{\mathrm{cd,rd}} = q_{W,\mathrm{cond}} + \bar{q}_{\mathrm{rad}} = h_{\mathrm{cd,rd}}(T_S - T_C). \tag{10}$$

3. *Effects of ambient environment*: Here, influences of solar radiation and ambient temperature are considered. The values of the first of them (solar radiation) are provided in terms of the corresponding heat flux and therefore, no further derivations are necessary, $q_{\mathrm{sol}} = \dot{Q}_S$. The latter one is assumed to be "measured" on the outer surface of the unheated wall and is assumed to vary only negligibly across the wall surface. Then, the heat flux resulting from the different inner and outer surface temperatures of the wall is described in terms of conduction through the wall as

$$q_O = h_{O,\mathrm{cond}}(T_O - T_S). \tag{11}$$

Since the heated wall contains metal piping filled with hot supply water, the effect of the ambient temperature T_O on the temperature T_C of its inner surface is neglected.
Due to the presence of the window and possible associated gaps and interstices, the ambient temperature is assumed to directly influence the zone temperature according to the following expression:

$$q_{O,Z} = h_{O,Z}(T_O - T_Z), \tag{12}$$

where the heat transfer coefficient $h_{O,Z}$ reflects all the above mentioned window-related leakage effects.

4. *Thermal energy supplied by the manipulated variables*: In the currently presented case, this energy is provided by the hot supply water of the temperature T_{SW} circulating at mass flow rate $\dot{m}$ in the metal piping encapsulated in the concrete core of the building. The thermal energy that is transferred from the supply water into the concrete core can be quantified as follows:

$$q_{\mathrm{in}} = c_w \dot{m}(T_{SW} - T_R). \tag{13}$$

Furthermore, based on the low thermal resistivity of the metals, it is assumed that the metal piping in which the water circulates has temperature T_P only negligibly different from the return water, $T_P \approx T_R$. Therefore, the return water temperature can be used for expression of the conductive heat transfer from the concrete core to the heated wall surface,

$$q_{R,\mathrm{cond}} = h_{R,\mathrm{cond}}(T_R - T_C), \tag{14}$$

with the heat transfer coefficient $h_{R,\mathrm{cond}}$ covering the effects of the different piping and wall materials and the distance from the water piping to the heated wall surface.

Based on this, thermodynamics of each of the considered inner variables of the building can be summarized:

- dynamics of the zone temperature T_Z is positively influenced by the convection from both considered walls and the heat flux coming from the ambient environment. Furthermore, the zone temperature is also increased due to the presence of solar radiation entering the room directly through the window,

$$\frac{\mathrm{d}T_Z}{\mathrm{d}t} \propto q_{C,\mathrm{conv}}, \quad \frac{\mathrm{d}T_Z}{\mathrm{d}t} \propto q_{S,\mathrm{conv}}, \quad \frac{\mathrm{d}T_Z}{\mathrm{d}t} \propto q_{O,Z}, \quad \frac{\mathrm{d}T_Z}{\mathrm{d}t} \propto q_{\mathrm{sol}}. \tag{15}$$

- heated wall surface temperature T_C is decreased by the amount of heat that is transferred into the zone air via convection while it is increased by the heat resulting from mutual interaction with the unheated wall and also by the heat transferred from heated supply water piping,

$$\frac{\mathrm{d}T_C}{\mathrm{d}t} \propto -q_{C,\mathrm{conv}}, \quad \frac{\mathrm{d}T_C}{\mathrm{d}t} \propto q_{\mathrm{cd,rd}}, \quad \frac{\mathrm{d}T_C}{\mathrm{d}t} \propto q_{R,\mathrm{cond}}. \tag{16}$$

- similar to the heated wall, the unheated wall is cooled down by the convection into the zone air. Moreover, the unheated wall surface temperature T_S decreases due to the thermal exchange with the heated wall while it is increased due to the effects of the ambient environment (ambient temperature T_O and solar radiation q_{sol}),

$$\frac{\mathrm{d}T_S}{\mathrm{d}t} \propto -q_{S,\mathrm{conv}}, \quad \frac{\mathrm{d}T_S}{\mathrm{d}t} \propto -q_{\mathrm{cd,rd}}, \quad \frac{\mathrm{d}T_S}{\mathrm{d}t} \propto q_O, \quad \frac{\mathrm{d}T_S}{\mathrm{d}t} \propto q_{\mathrm{sol}}. \tag{17}$$

- finally, the return water temperature T_R is affected by the supplied thermal energy and further heat transfer with the surface of the heated wall,

$$\frac{\mathrm{d}T_R}{\mathrm{d}t} \propto -q_{R,\mathrm{cond}}, \quad \frac{\mathrm{d}T_R}{\mathrm{d}t} \propto q_{\mathrm{in}} \tag{18}$$

For further use in a mathematical model, all the building inner variables are considered as the state variables of the mathematical model of the building thermodynamics, $x = [T_Z, T_C, T_S, T_R]$. Moreover, inputs $u = [T_{SW}, \dot{m}]$ stand for the manipulated variables being supply water temperature and the mass flow rate of the supply water and $d = [T_O, q_{\mathrm{sol}}]$ correspond to the predictable

disturbances, namely the temperature of the ambient environment and the solar radiation. Then, the above mentioned phenomena described by Eqs. (15)–(18) are captured by the following set of differential equations:

$$\begin{aligned}
\dot{x}_1 &= \overline{p}_1\left|x_2 - x_1\right|^{\frac{1}{3}}(x_2 - x_1) + \overline{p}_2\left|x_3 - x_1\right|^{\frac{1}{3}}(x_3 - x_1) + \overline{p}_3(d_1 - x_1) + \overline{p}_4 d_2\\
\dot{x}_2 &= -\overline{p}_5\left|x_2 - x_1\right|^{\frac{1}{3}}(x_2 - x_1) + \overline{p}_6(x_3 - x_2) + \overline{p}_7(x_4 - x_2)\\
\dot{x}_3 &= -\overline{p}_8\left|x_3 - x_1\right|^{\frac{1}{3}}(x_3 - x_1) - \overline{p}_9(x_3 - x_2) + \overline{p}_{10}(d_1 - x_3) + \overline{p}_{11} d_2\\
\dot{x}_4 &= -\overline{p}_{12}(x_4 - x_2) + \overline{p}_{13} u_2(u_1 - x_4).
\end{aligned} \tag{19}$$

To ensure admissible computational complexity of the predictive controller exploiting the nonlinear model, the structure (19) was discretized using Euler discretization method considering fixed a priori known sampling time t_s (Stetter, 1973). In this paper, t_s=15 min is considered. The discretization procedure results in a series of difference equations expressing the one-step predictions of the system behavior,

$$\begin{aligned}
x_{1,k+1} &= x_{1,k} + p_1\left|x_{2,k} - x_{1,k}\right|^{\frac{1}{3}}(x_{2,k} - x_{1,k}) + p_2\left|x_{3,k} - x_{1,k}\right|^{\frac{1}{3}}(x_{3,k} - x_{1,k})\\
&\quad + p_3(d_{1,k} - x_{1,k}) + p_4 d_{2,k}\\
x_{2,k+1} &= x_{2,k} - p_5\left|x_{2,k} - x_{1,k}\right|^{\frac{1}{3}}(x_{2,k} - x_{1,k}) + p_6(x_{3,k} - x_{2,k}) + p_7(x_{4,k} - x_{2,k})\\
x_{3,k+1} &= x_{3,k} - p_8\left|x_{3,k} - x_{1,k}\right|^{\frac{1}{3}}(x_{3,k} - x_{1,k}) - p_9(x_{3,k} - x_{2,k})\\
&\quad + p_{10}(d_{1,k} - x_{3,k}) + p_{11} d_{2,k}\\
x_{4,k+1} &= x_{4,k} - p_{12}(x_{4,k} - x_{2,k}) + p_{13} u_{2,k}(u_{1,k} - x_{4,k}),
\end{aligned} \tag{20}$$

which are more suitable for implementation of the predictive controller than the continuous-time model (19). To obtain estimates of the parameters p of the discretized structure (20), MRI approach (whose explanation is provided later in this Section) belonging to advanced identification techniques was employed.[1]

3.2. Linear model (LM)

In order to simplify the model (19), let us adopt the assumption that the cubic roots of the temperature differences related to the heat convection are constant over the whole range of the operating points of the building. This simplifies the nonlinear terms as follows:

$$p|x_i - x_j|(x_i - x_j)^{\frac{1}{3}} \approx a(x_i - x_j). \tag{21}$$

Furthermore, $q_{\text{in}} = c_w \dot{m}(T_{SW} - T_R)$ is assumed to be the control input instead of the pair $\dot{m}$ and T_{SW}. Based on these assumptions, the linear version of the model Eq. (20) can be summarized as a discrete-time state space model as follows:

$$x_{k+1} = Ax_k + Bu_k + B_d d_k \tag{22}$$

with the state matrices having the following structure:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & 0 \\ a_5 & a_6 & a_8 & a_7 \\ a_9 & a_{10} & a_{11} & 0 \\ 0 & a_{12} & 0 & a_{13} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b \end{bmatrix}, \quad B_d = \begin{bmatrix} b_{d1} & b_{d2} \\ 0 & 0 \\ b_{d3} & b_{d4} \\ 0 & 0 \end{bmatrix}. \tag{23}$$

In this model, state and disturbance variables correspond to the previously mentioned ones and $u = q_{\text{in}}$ refers to the optimized input. The sampling period of the system has been chosen as t_s = 15 min. The model parameters a, b, b_d have been estimated by a multistep prediction error minimization procedure (MRI). For further details on this method, the readers are referred to Žáčeková & Prívara (2012).

3.3. Switched linearly approximated model (SLM)

The main idea of this approach is that for a combination of inputs u, disturbances d and state variables x, a linear time-varying approximation of model (20) can be found by replacing particular nonlinearities with time-varying terms. In case of a building, this approach is even more natural and expected as the nonlinear mathematical description of the building contains terms depending on the differences between two state variables, namely $p\left|x_i - x_j\right|^{\frac{1}{3}}(x_i - x_j)$ which are likely to vary much less than the temperatures themselves. As an opposite to the linear models described earlier where the nonlinear terms are linearized "before the identification" and having the gathered data at disposal, parameters of linear time invariant model are estimated considering the purely linear character of the model, in this case, the nonlinear model is identified off-line and using its parameters, the nonlinearities are continuously approximated on-line depending on the actual values of the chosen auxiliary variables which leads to a time-varying linear model.

In order to get rid of the nonlinear terms coupling the states, let us propose an approximation procedure based on the auxiliary variables as follows.

Let us introduce two auxiliary variables, $\delta_{x_{1,2},k}$ and $\delta_{x_{1,3},k}$ defined such that

$$\begin{aligned}
\delta_{x_{1,2},k} &= \sqrt[3]{|x_{2,k_m} - x_{1,k_m}|}\\
\delta_{x_{1,3},k} &= \sqrt[3]{|x_{3,k_m} - x_{1,k_m}|},
\end{aligned} \tag{24}$$

where $k \geq k_m$ refers to discrete time and k_m indicates the time instant when the last available values of the state variables arrived. The derived model shall predict the behavior of the building over certain prediction horizon during which no current values of the state variables are available. Therefore, at each "measurement" time instant, the values of $\delta_{x_{1,2},k}$ and $\delta_{x_{1,3},k}$ are calculated and they are used by the optimizer over the whole prediction horizon. The necessity of realizing the difference between the *real-life time* (in which the model is time-varying) and the *internal time of the optimizer* (in which the model stays constant over the prediction horizon) is obvious.

Then, the nonlinear terms appearing in the model Eq. (20) can be approximated as

$$\begin{aligned}
\sqrt[3]{|x_2 - x_1|}(x_2 - x_1) &\approx \delta_{x_{1,2},k}(x_{k_m})(x_2 - x_1),\\
\sqrt[3]{|x_3 - x_1|}(x_3 - x_1) &\approx \delta_{x_{1,3},k}(x_{k_m})(x_3 - x_1)
\end{aligned} \tag{25}$$

for all $k \geq k_m$. Here, the expressions $\delta_{x_{1,2},k}(x_{k_m})$, $\delta_{x_{1,3},k}(x_{k_m})$ are used to emphasize the fact that the values of auxiliary variables depend only on the last available state values.

The bilinear term in the last differential equation is (similar to the previous approaches) considered as the new controlled input q_{in} while the vector of disturbances d remains unchanged. The linearized difference equations can be now summarized as:

$$x_{k+1} = A_{app}(x_{k_m})x_k + B_{app}u_k + B_d d_k, \tag{26}$$

where

[1] Let us note that the parameters p_i of the discrete time model (20), $i \in \{1, 2, \ldots, 13\}$, differ from the parameters $\overline{p}_i$ of the continuous time model (19) since they incorporate also the effect of the chosen sampling period t_s.

$$A_{app}(x_{k_m}) = \begin{bmatrix} 1-(\tilde{p}_1+\tilde{p}_2+p_3) & \tilde{p}_1 & \tilde{p}_2 & 0 \\ \tilde{p}_5 & 1-(\tilde{p}_5+p_6+p_7) & p_6 & p_7 \\ \tilde{p}_8 & p_9 & 1-(\tilde{p}_8+p_9+p_{10}) & 0 \\ 0 & p_{12} & 0 & 1-p_{12} \end{bmatrix}, \quad (27)$$

with

$$\tilde{p}_1 = p_1\delta_{x_{1,2},k}, \quad \tilde{p}_2 = p_2\delta_{x_{1,3},k},$$
$$\tilde{p}_5 = p_5\delta_{x_{1,2},k}, \quad \tilde{p}_8 = p_8\delta_{x_{1,3},k} \quad (28)$$

and

$$B_{app} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ p_{13} \end{bmatrix}, \quad B_d = \begin{bmatrix} p_3 & p_4 \\ 0 & 0 \\ p_{10} & p_{11} \\ 0 & 0 \end{bmatrix}. \quad (29)$$

At this point, the whole algorithm of obtaining the linear approximated model of the building can be summarized.

At each discrete sample $k = k_m$, the values of the state variables x are provided and the auxiliary variables $\delta_{x_{1,2},k}$, $\delta_{x_{1,3},k}$, are evaluated according to Eq. (24). Making use of the calculated auxiliary variables, a linear discrete-time model (26) of the building is created with the corresponding matrices. This approximated model is used until the new state values arrive, which means that at each discrete time sample, a new model is approximated and used by the optimizer over the following prediction horizon $k \in \{1, 2, \ldots, P\}$ of the *internal time of the optimizer*.

The readers interested in theoretical properties of the linear MPC exploiting model belonging to widely used family of linear time-/parameter-varying models (which SLM also belongs to) are warmly referred to Falcone, Borrelli, Tseng, Asgari, & Hrovat (2008) where the stability and feasibility of such formulation are discussed in detail. It should be noticed that one of the crucial assumption is that on constancy of the model over the prediction horizon, which is satisfied also by the SLM model and therefore, the results obtained in Falcone et al. (2008) hold also for the case of LMPC with SLM model.

3.4. MRI identification for nonlinear models

Having the model structures at disposal, it is necessary to estimate the parameters of these structures from the available input/output data. Since the obtained models are expected to be used by the predictive controllers as system dynamics predictors, this fact needs to be taken into account as early as at the point of choosing of the identification procedure. Instead of classical identification methods performing minimization of one-step prediction error (the so-called prediction error methods or PEMs Ljung, 2007), advanced approach focusing directly on minimization of multi-step prediction error is exploited since it provides models with better long-term prediction performance which is highly requested when considering use of the model with MPC. The objective is to find such parameters of the given model structure which minimize the multi-step prediction error (Laurí et al., 2010) over the whole prediction horizon,

$$J_{MRI} = \sum_{k=0}^{N-P}\sum_{i=1}^{P}\left[y_{k+i} - \hat{y}_{k+i|k}\right]^2, \quad (30)$$

where $\hat{y}_{k+i|k}$ is the i-step output prediction constructed from data up to time k, N corresponds to the number of samples and P stands for prediction horizon considered for identification. In case of linear model structures which is also the case of structure (22), several reliable approaches can be found. Therefore, one particular algorithm that has already been successfully used for building model parameters identification (interested readers are referred to Žáčeková & Prívara, 2012) will be used also in this paper to estimate the parameters of the linear structure (22).

When talking about identification of models with nonlinear structure performing minimization of (30), no methods of solving of the arisen problem can be found in the available literature according to authors' best knowledge. The proposed extension of the MRI identification methods (Žáčeková & Prívara, 2012) for nonlinear systems is described in the following text.

Without any loss of generality, let us assume nonlinear systems where the multi-step predictor $\hat{y}_{k+i|k}$ can be formulated in the following way:

$$\hat{y}_{k+i|k} = Z_{L,k+i}\hat{\theta}_L + Z_{NL,k+i}\hat{\theta}_{NL}, \quad i \in 1, 2, \ldots, P, \quad (31)$$

where $Z_{L,k+i} = [u_{k+i-n_d} \ldots u_{k+i-n_b}\ y_{k+i-1} y_{k+i-n_a}]$ and $\hat{\theta}_L = [\hat{b}_{n_d} \ldots \hat{b}_{n_b}\ \hat{a}_1 \ldots \hat{a}_{n_a}]$ are regression matrix and the vector of unknown parameters describing the linear part of the model dynamics, respectively. n_a denotes the number of past outputs in the regressor, n_b is the number of inputs in the regressor and n_d represents their delay compared to the outputs. The nonlinear part of the system dynamics is described by $\hat{\theta}_{NL} = [\hat{\theta}_1\ \hat{\theta}_2 \ldots \hat{\theta}_n]^T$ with n being the number of identified parameters and $Z_{NL,k+i} = [f_1(\cdot)\ f_2(\cdot) \ldots f_n(\cdot)]$.In general, $f_i(\cdot)$ are functions of $u_{k+i-1}, \ldots, u_{k+i-n_{b,NL}}$ and $y_{k+i-1}, \ldots, y_{k+i-n_{a,NL}}$ with parameters $n_{a,NL}$ specifying the number of past outputs in the nonlinear dynamics and $n_{b,NL}$ representing the number of inputs in the nonlinear structure.

It is important to note that not every output contained in regression matrices $Z_{L,k+i}$ and $Z_{NL,k+i}$ is available at time k, thus the multi-step predictions $\hat{y}_{k+i|k}$ must be obtained recursively by applying i-times the expression $\hat{y}_{k+1|k} = Z_{L,k+1}\hat{\theta}_L + Z_{NL,k+1}\hat{\theta}_{NL}$ with initial conditions y_k. Now, the estimate of matrix of parameters $\hat{\theta}$ can be obtained as a solution of the following optimization task:

$$[\hat{\theta}_L, \hat{\theta}_{NL}]^* = \arg\min_{[\theta_L,\theta_{NL}]} \sum_{i=1}^{P}\sum_{k=0}^{N-i} [y_{k+i} - Z_{L,k+i}\theta_L - Z_{NL,k+i}\theta_{NL}]^2$$

subject to :

$$\theta_L \in \theta_L(\mathcal{S}_L), \quad \theta_{NL} \in \theta_{NL}(\mathcal{S}_{NL}) \quad (32)$$

where $\mathcal{S}_L$ and $\mathcal{S}_{NL}$ correspond to the sets of all admissible estimated parameters. These constraints enable the user to incorporate certain a priori information into the identification procedure, for example to ensure that certain parameters are nonnegative or lie in a constrained interval, etc. In the currently presented case, two different methods of obtaining of $\hat{y}_{k+i|k}$ were exploited:

- *variant A* – for computing of $\hat{y}_{k+i|k}$, the output predictions are used only for recursive calculation of Z_L and for calculation of Z_{NL}, the available output data are exploited. In such case, the optimization task (32) is *polynomial* in parameters and can be solved employing standard solver for nonlinear programming. This is certain kind of approximation where the nonlinear part of the system dynamics is basically identified just in sense of minimization of one-step prediction error while the linear part is still identified with respect to the multi-step prediction error minimization criterion.
- *variant B* – for computing of $\hat{y}_{k+i|k}$, the output predictions are used for recursive calculation of Z_L as well as Z_{NL}. In this case, the parameters of both the linear and nonlinear part of the system dynamics are searched such that the multi-step prediction errors are minimized. It should be noted that in this case, the optimization task (32) is again a nonlinear programming problem, however, it might not be only polynomial in the estimated parameters any more.

3.5. Identification results

Making use of the above mentioned identification procedures, the parameters of all model structures presented in the current Section were identified from the available identification data set.

At first, the comparison of the nonlinear models obtained using variant A and variant B of the nonlinear MRI identification (denoted as nMRIa and nMRIb, respectively) is presented in Fig. 3. For identification purposes, prediction horizon $P=20$ samples was considered which with sampling period $t_s=15$ min corresponds to duration of 5 h. It can be argued that the prediction horizon is shorter than the real prediction horizon of the predictive controllers (in the current application, the predictive controllers perform optimization calculations over 12 h corresponding to 48 samples), however, in Žáčeková, Váňa, & Cigler (2014) and Gopaluni, Patwardhan, & Shah (2004) it was shown that from certain prediction horizon threshold, the increase of the identification prediction horizon can lead to degradation of the performance of the obtained model.

It is obvious that both obtained nonlinear models fit the verification data very well also on longer verification interval with slight superiority of the model identified making use of nMRIb. nMRIa variant provides model with performance which is only slightly worse than that of the model obtained by (seemingly) more computationally demanding variant nMRIb. It is true that within the nMRIb, a more general nonlinear programming task needs to be solved (which is undoubtedly more computationally demanding than just solving of polynomially nonlinear programming performed within nMRIa), however, the overall optimization which is solved within nMRIb takes less computational time than optimization performed within nMRIa. Although one iteration of nMRIb is slower (due to solving of the more general optimization problem), on the other hand less iterations are needed to converge to the solution of the optimization problem. This can be explained such that the task formulated within nMRIb brings the chosen nonlinear structure closer to reality and thus also to the verification data – this of course holds well only in case that a reasonable model structure was chosen. Therefore, it might be more advantageous to choose identification of nonlinear model in variant nMRIb which can be ultimately faster and provides a more accurate and reliable model. Based on this, the model obtained by nMRIb was chosen to be used with the nonlinear predictive controller in the role of the system dynamics predictor.

Now, the graphical and numerical comparison of all above described models follow. Since the models are intended to be used with the MPC, one of their most important features is the ability to provide reasonable predictions over the whole prediction horizon. In this paper, the prediction horizon $T_P = 12$ h is considered which with 15-min sampling corresponds to $P=48$ samples. Let us remind that in the role of the nonlinear model, nMRIb was chosen.

Fig. 4 shows several weeks of comparison of the models which are used for the building behavior predictions with the linear time invariant (LM model), linear time-varying (SLM model) and nonlinear MPC (NM model). At each discrete time sample ($t_s=15$ min), 12-h predictions are calculated based on the provided state values. All the predictions of the models are plotted together with the verification data.

Looking at Fig. 4, it is clear that while the NM behavior constraints the quality of the prediction behavior from above with the smallest deviations from the verification data and the LM behavior exhibits the highest prediction errors, the performance of the performance of the time-varying model is somewhere in the middle between these two "limit" cases. The most obvious are the differences in the behavior when looking at the 200-th and the 300-th hour of the comparison. While the absolute value of prediction errors for the off-line identified linear model reaches up to 2 °C, the error obviously decreases through the switched linearly approximated time-varying model down to the nonlinear model which provides the predictions with the least prediction error out of the three compared models, which in turn justifies the use of the predictive controller with the more complex nonlinear model.

In order to compare the models in a more complete way, the statistical comparison of the models is provided in Table 2. The length of the evaluated period was nearly 3 months. In the table, LM specifies the linear model, SLM stands for the switched linearly approximated model and NM represents the nonlinear model. For each model, ε_{av} being the average prediction error over the whole 12-h prediction horizon and the maximum prediction error ε_{max} over the prediction horizon are inspected.

The table clearly demonstrates that the most reliable predictions are provided by the NM model. However, this is not a surprise as this model takes the whole dynamics of the building into account including the nonlinearities. On the other hand, it can be seen that considering the linear time-dependent model, the quality of the predictions fairly improves compared to the linear time invariant model. With SLM model, the reduction of ε_{av} is almost 40% and the reduction of ε_{max} is nearly 37%.

4. Model predictive control

In this section, the considered MPC variants are briefly explained and the optimization routines used to solve the corresponding optimization problems are presented. At the end of this Section, the quantized nonlinear predictive control algorithm is proposed.

4.1. Linear MPC

The control requirements which have been chosen for the linear MPC to be satisfied (minimization of both the thermal comfort

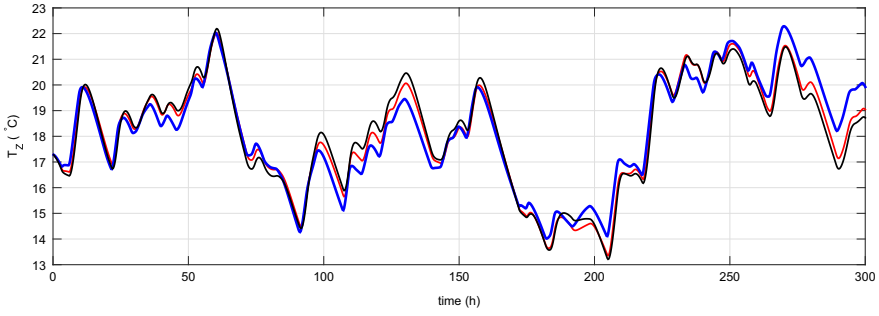

Fig. 3. Comparison of nMRIa (——) and nMRIb (——) models with the verification data (——).

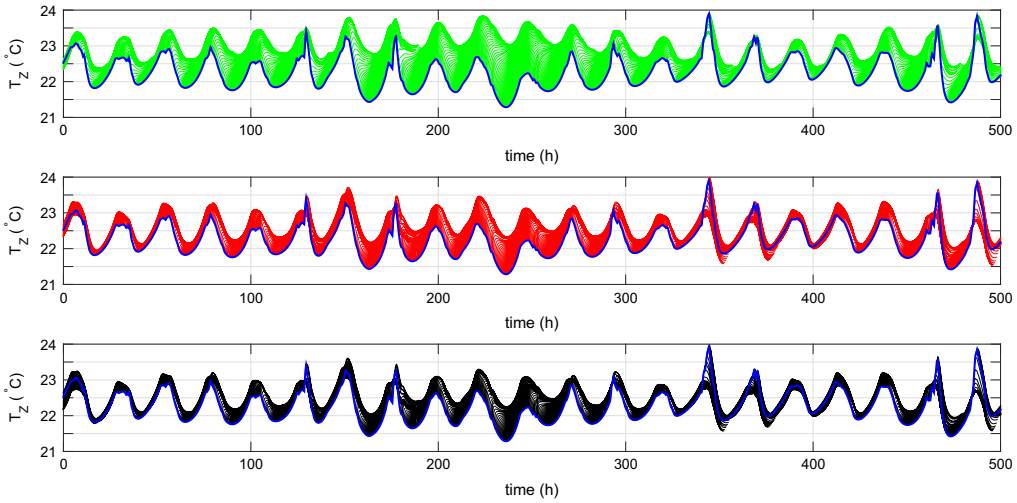

Fig. 4. Comparison of T_Z predictions of the LM (——), SLM (——) and NM (——) models with the verification data (——).

Table 2
Statistical comparison of the models.

	LM	SLM	NM
ε_{av} (°C)	0.57	0.34	0.30
ε_{max} (°C)	1.89	1.20	1.08

violation and the energy consumption) can be mathematically summarized as follows:

$$J_{MPC,k} = \sum_{i=1}^{P} W_{1,\mathbf{p}}(k+i) \| q_{\text{in},k+i} \|_{\mathbf{p}} + \sum_{i=1}^{P} W_{2,\mathbf{p}} \| CV_{k+i} \|_{\mathbf{p}} \tag{33}$$

$$\text{s. t.:} \quad \text{linear dynamics (22)}$$
$$0 \le q_{\text{in},k+i} \le \bar{q}_{\text{in},k+i}, \quad i = 1, \ldots, P$$
$$\hat{T}_{Z,k+i|k} \ge T_{Z,k+i}^{min} - CV_{k+i}.$$

This formulation considers a combination of linear and quadratic penalization indicated by the index $\mathbf{p} \in \{1, 2\}$ which enables us to shape the penalization criterion conveniently. Time-varying weighting matrices W reflecting the time dependence of the electricity tariffs and prediction horizon P stand for the tuning parameters of the controller. Comfort violation is calculated based on the difference between the zone temperature prediction $\hat{T}_Z$ and its lowest acceptable bound T_Z^{min} and the hard constraints are relaxed employing an auxiliary variable CV. Exact values of the optimization problem settings can be found in Table 3.

As the linear version of MPC optimizes supplied heat q_{in}, a post-processing procedure is needed to obtain the particular values of T_{SW} and $\dot{m}$ which correspond to the true control inputs of the thermally activated building system (TABS). This straightforward postprocessing holds the mass flow rate fixed $\dot{m} = \dot{m}_{pp}$ and it calculates the supply water command as $T_{SW} = q_{\text{in}}/\dot{m}c_w + T_R$. Should the calculated supply water command be higher than $\overline{T}_{SW}$, $T_{SW} = \overline{T}_{SW}$ is set and the mass flow rate command is calculated as $\dot{m} = q_{\text{in}}/c(T_{SW} - T_R)$. If the heating effort is lower than a threshold value $q_{\text{in},tr}$, the TABS manipulated variables are set to $T_{SW} = T_R$ and $\dot{m} = \underline{\dot{m}}$. The settings of the post-processing procedure are listed in Table 3.

Table 3
Table of controller parameters.

$W_{1,1}$ (high tariff)	0.01	$W_{1,2}$ (high tariff)	1.6
$W_{1,1}$ (low tariff)	0.005	$W_{1,2}$ (low tariff)	0.8
$W_{2,1}$	2×10^6	$\bar{q}_{\text{in}}$	90×10^4
$W_{2,2}$	10^4	$q_{\text{in},tr}$	700
$\underline{T}_{SW}$	20	$\overline{T}_{SW}$	50
P	48	$\dot{m}_{pp}$	20

4.2. Nonlinear MPC

Thanks to the use of nonlinear programming optimization method, the nonlinear MPC can exploit the more reliable nonlinear discrete-time state-space description of the building behavior and address directly the minimization of the evaluative criterion (6). To obtain computationally tractable solution, also the criterion (6) needs to be discretized in time. This results in the following nonlinear MPC cost criterion:

$$J_{NMPC,k} = \frac{1}{t_s}\sum_{i=1}^{P} \omega CV_{k+i} + \frac{1}{t_s}\sum_{i=1}^{P} \left(P_{E,k+i}(P_C(u_{2,k+i}) + P_C(\bar{u}_{3,k+i}))\right) + \frac{1}{t_s}\sum_{i=1}^{P} P_W \bar{u}_{3,k+i}, \tag{34}$$

where t_s represents the chosen constant sampling period, P stands for the prediction horizon, $P_C(\cdot)$ corresponds to Eq. (7) and $\bar{u}_3$ represents a virtual input which corresponds to the storage water mass flow rate $\dot{m}_{St}$,

$$\bar{u}_3 = u_2 \frac{u_1 - x_4}{T_{St} - x_4}.$$

The obtained optimal profiles u_1, u_2 are required to satisfy the technical limitations which are formulated as box constraints,

$$\max\{\underline{T}_{SW}, x_4\} \le u_{1,k} \le \overline{T}_{SW},$$
$$\underline{\dot{m}} \le u_{2,k} \le \overline{\dot{m}}. \tag{35}$$

Last but not least, the dynamics of the building must not be violated which is represented by the satisfaction of the model dynamics (20).

In the role of the optimization routine, gradient optimization algorithm (Zhou, Doyle, & Glover, 1996; Bryson, & Ho, 1975) with variable step length is employed. This approach is able to address optimization problems in the following form:

$$\text{minimize } J = \sum_{i=1}^{P} L(x_i, u_i) + \phi(x_P, u_P)$$
$$\text{such that } u_i \in \langle \underline{u}, \overline{u} \rangle. \tag{36}$$

To find the solution of (36), the following idea is employed: starting from an initial estimate of the optimal input profile u^0, the opposite direction of the gradient of the minimization cost criterion is iteratively followed until convergence to the optimal input vector,

$$u^l = u^{l-1} - \alpha^l \frac{\partial J}{\partial u}. \tag{37}$$

Here, l represents the iteration of the gradient algorithm and α^l is the step length at l-th iteration.

To obtain computationally tractable solution of this optimization task, the Hamiltonian

$$\mathcal{H} = L_k + \lambda_{k+1}^{\mathrm{T}} f(x_k, u_k) \tag{38}$$

is created. Here, L_k is the integral or in discrete-time case the *summation* part of the criterion J, $f(x_k, u_k)$ is the vector field representing the dynamics of the controlled system and λ is the so-called co-state vector with the backwards dynamics

$$\lambda_k = \frac{\partial H}{\partial x}(x_k, u_k, \lambda_{k+1}) \tag{39}$$

and the terminal condition

$$\lambda_P = \left.\frac{\partial J}{\partial x}\right|_P. \tag{40}$$

It can be shown that the gradients of both the cost criterion J and the Hamiltonian H with respect to the input vector u are equal, $\partial J/\partial u = \partial H/\partial u$, and therefore, the iterative search (37) turns into

$$u^l = u^{l-1} - \alpha^l \frac{\partial H}{\partial u}. \tag{41}$$

To satisfy the input constraints, the input profile u^l is at each iteration projected on the admissible input interval $\langle \underline{u}, \overline{u} \rangle$. The iterative search (41) is used until convergence which is usually defined as

$$|J(u^l) - J(u^{l-1})| \leq \epsilon \tag{42}$$

with some reasonably chosen nonnegative tolerance $\epsilon > 0$.

As can be expected, the search step length α significantly influences the convergence properties of the algorithm. In order to provide smooth and uniform convergence to the optimum, α should be small in case that the cost criterion J decreases rapidly and it should increase in case that the change of the cost criterion $|J(u^l) - J(u^{l-1})|$ is small. To satisfy these requirements, the following formula for the search step length is proposed:

$$\alpha^l = -\beta \log(\gamma \Delta J^l). \tag{43}$$

Here, $\Delta J^l = |J(u^l) - J(u^{l-1})|$ is the change of the cost function value and $\beta > 0$, $\gamma > 0$ are some suitably chosen constants. Last of all, the step length α^l is constrained at each gradient algorithm iteration,

$$\underline{\alpha} \leq \alpha^l \leq \overline{\alpha}. \tag{44}$$

Parameters $\underline{\alpha} > 0$ and $\overline{\alpha} > 0$ are together with β and γ considered to be the tuning parameters of the presented optimization algorithm.

4.3. Quantized MPC

As was mentioned earlier, the mass flow rate should belong to the admissible set of discrete values $\dot{M}_{\mathrm{adm}}$. In case of the linear MPCs which calculate optimal amount of energy that should be delivered into the zone and subsequently perform the postprocessing to obtain the values of mass flow rate and supply water temperature, the discrete-valued nature of the mass flow rate can be very straightforwardly taken into account. However, the situation is more complicated in case of nonlinear MPC. As already mentioned in the Introductory Section, two ways how to obtain discrete-valued mass flow rate sequence are considered in this work.

The first of them consists in use of additional postprocessing which is performed after the continuous-valued optimization is finished. The most straightforward postprocessing routine is pure rounding of the obtained continuous-valued mass flow rate sequence u_2 away from zero to the nearest multiple of the quantization step,

$$u_{2,q} = q_{st} \cdot \mathrm{round}\left(\frac{u_2}{q_{st}}\right), \tag{45}$$

with $\mathrm{round}(\cdot) = \mathrm{sgn}(\cdot)\lceil|\cdot|\rceil$. Major advantage of this approach is its simplicity – the a posteriori quantization can be performed by a hardware component and therefore, no increase of the computational complexity occurs. However, it can be expected that such naive approach significantly degrades the control performance of the original controller since the fact that the manipulated variable will be quantized a posteriori is not taken into account in the used optimization routine.

This drawback is solved by the adaptation of the original Hamiltonian-based method representing the second way of achieving that discrete-valued mass flow rate profile is obtained. Here, a regular *mid-processing* iteration is performed each $\mathbf{I}$-th iteration of the gradient search. Thanks to this, the information about the discrete-valued nature of one of the manipulated variables is incorporated into the optimization procedure and the optimality of the original continuous-valued optimization technique is preserved.

The *mid-processing* is performed at particular iterations $l = m \times \mathbf{I}$, $m \in \mathbb{N}^+$ after the gradient step is made and it can be described as follows: first of all, the quantized mass flow rate sequence $u_{2,\bigcirc}^l$ is obtained by projecting the continuous-valued mass flow rate vector $\hat{u}_2^l$ on the admissible set $\dot{M}_{\mathrm{adm}}$ given by (8) with respect to the chosen quantization step q_{st},

$$u_{2,\bigcirc}^l = q_{st} \cdot \mathrm{round}\left(\frac{\hat{u}_2^l}{q_{st}}\right). \tag{46}$$

These P predicted quantized mass flow rate samples are connected with n_f past mass flow rate samples $\overleftarrow{u_2} = [u_{2,k-n_f}, u_{2,k-n_f-1}, \ldots, u_{2,k-1}]$ with k representing the current time step, and vector $\overleftrightarrow{U_2} = [\overleftarrow{u_2}, u_{2,\bigcirc}^l]$ is received. The vector $\overleftrightarrow{U_2}$ represents all mass flow rate samples that will have been applied into the system until time $k + P$ and have influence on the frequency properties of the manipulated variable u_2 .

Then, $\overleftrightarrow{U_2}$ is filtered with a suitably defined low-pass filter with order n_f which helps us to suppress the undesired high frequencies and decrease oscillations in the last P-sample subvector representing the currently optimized input sequence. This P-sample subvector is extracted and after quantization and projection on its admissible range is used for the next iteration of the gradient search.

The overall control algorithm is then summarized as follows:

Algorithm agqNPC

1. obtain current values of the state variables $x_{curr,k}$
2. consider input profiles from the previous iteration

$\{u_1^{l-1}, u_2^{l-1}\}$ and obtain state trajectories $X = [x_0, x_1, \ldots, x_P]$ according to the model (20) with $x_0 = x_{curr,k}$;

3. according to the co-state dynamics (39), obtain the co-state trajectory $\Lambda = [\lambda_0, \lambda_1, \ldots, \lambda_P]$ with terminal condition (40);
4. calculate gradients $\partial H/\partial u_1$, $\partial H/\partial u_2$, and perform gradient step (41), obtain ${u_1}^l$ and $\hat{u}_2^l$;
5. **if** $\mathrm{mod}(l, \mathbf{I}) = 0$
   then perform the *mid-processing*:
   (i) quantize mass flow rate $\hat{u}_2^l$ according to (46) with chosen q_{st}, obtain $u_{2,\bigcirc}^l$,
   (ii) create sequence $\overleftrightarrow{U}_2^{\,l} = [\overleftrightarrow{u}_2, u_{2,\bigcirc}^l]$,
   (iii) filter $\overleftrightarrow{U}_2^{\,l}$ using a low-pass filter of order n_f with the chosen characteristics, obtain $\overleftrightarrow{U}_{2,\mathrm{filt}}^{\,l}$,
   (iv) quantize $\overleftrightarrow{U}_{2,\mathrm{filt}}^{\,l}$ with chosen q_{st}, obtain $\overleftrightarrow{U}_{2,\mathrm{filt},\bigcirc}^{\,l}$,
   (v) extract ${u_2}^l$ as the last P samples of $\overleftrightarrow{U}_{2,\mathrm{filt},\bigcirc}^{\,l}$;
   else $u_2^l = \hat{u}_2^l$;
6. project the sequences ${u_1}^l$ and ${u_2}^l$ on the admissible intervals $\langle \max\{\underline{T}_{SW}, x_4\}, \overline{T}_{SW} \rangle$ and $\langle \underline{\dot{m}}, \overline{\dot{m}} \rangle$;
7. **if** $|J(\{u_1^l, u_2^l\}) - J(\{u_1^{l-\mathbf{I}}, u_1^{l-\mathbf{I}}\})| \leq \epsilon$
   then terminate,
   else $l = l + 1$, repeat from (2);
8. apply the first sample of the calculated input profiles into the system, in the next time instance repeat from (1).

The performance of both the naive a posteriori quantization and the algorithm employing the *mid-processing* iteration is verified in the following section. In order to provide a better comparison, the results of the original continuous-valued nonlinear MPC are provided together with the results of the linear versions of predictive controller.

5. Results

First of all, visual comparison of the thermal comfort performance is presented in Fig. 5.

Fig. 5 shows the zone temperature profiles over a 6-day period for the linear predictive controllers with LM and SLM and the nonlinear continuous-valued predictive controller. From this figure, it can be seen that all controllers are carefully tuned to achieve satisfactory thermal comfort performance since all of them are able to satisfy the room temperature requirements and maintain the zone temperature within the admissible zone above the zone temperature threshold. This feature is very crucial since a controller that does not fulfill the thermal comfort requirements and violates the zone temperature threshold significantly is literally useless for building temperature control. Out of all considered controllers, the nonlinear MPC (NMPC) exhibits the most superior performance – it satisfies the required thermal comfort keeping the zone temperature within the admissible range and on the other hand, it obviously does not waste too much energy keeping the zone temperature just as high above the threshold as needed. This result could have been expected as the NMPC combines the model with the best prediction performance out of the considered set and it also directly addresses the minimization of the optimization criterion corresponding to the ultimate evaluative performance criterion (6).

Fig. 6 provides the second part of the visual comparison – it depicts the monetary cost that is being paid for the control at each time instance.

All profiles exhibit sinusoidal-like trends – this is caused by the consideration of time-varying price of the electricity. The higher parts of the profiles correspond to low-tariff hours while the lower parts match the non-working hours with cheap electricity. Also from this figure, the monetarily more economical nature of the NMPC can be observed. The NMPC spares significant amount of expenses compared to its linear counterparts. This superiority comes from the use of more precise nonlinear model and it is of course caused also by the nonlinear cost function of the NMPC which directly corresponds to the amount of money that is paid for the control. It can be also seen that the SLM model which is closer to the nonlinear one enables also the controller with approximated cost function to achieve better economical performance than the original linear model. For further illustration, the cumulative sum of the monetary cost of the control is depicted in Fig. 7. The provided profiles are normalized with respect to the total price TP_{LM} that is paid by the linear MPC with the ordinary time-invariant linear model.

The statistical comparison of the energy consumption can be found in Table 4. *TP* expresses the overall price paid for zone temperature control. Moreover, the particular energy consumptions normalized with respect to the consumption of the linear MPC using the ordinary off-line identified linear model are expressed. Furthermore, also the comparison of the average computational time T_{av} and the maximum computational time T_{max} per discrete time instance is provided.

The superiority of the NMPC is demonstrated once again. It can be seen that although the comparison of the identified models was very optimistic in the case of linear time-dependent model versus the linear time-invariant one, the resulting effect of the good

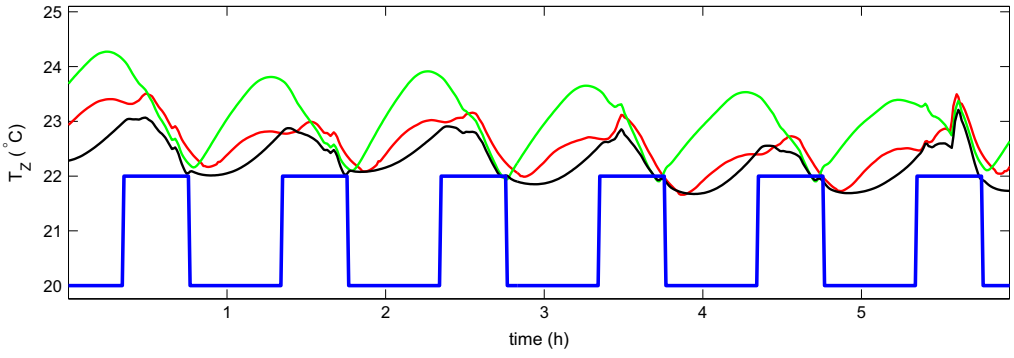

Fig. 5. Zone temperature control (— – linear MPC with LM, — – linear MPC with SLM, — – nonlinear continuous-valued MPC with NM, — – T_Z^{min}).

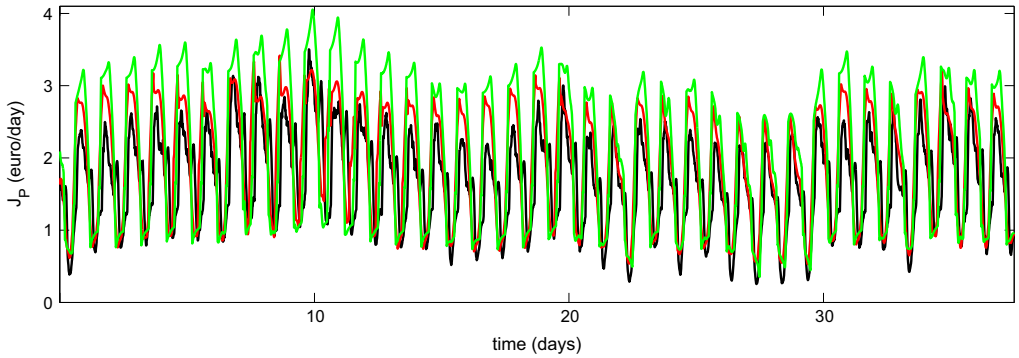

Fig. 6. Overal price of the control effort (— – linear MPC with LM, — – linear MPC with SLM, — – nonlinear continuous-valued MPC with NM).

model on the overall monetary cost of the control is not so attractive. This can be simply explained by the fact that although the good predictor is crucial for the proper functioning of the MPC (either linear or nonlinear), so is the properly chosen optimization criterion. Based on this observation, in the building climate control, the need for the use of nonlinear MPCs which are able to address the task of the real-life price minimization in a direct way instead of using certain approximation is obvious. However, one more aspect needs to be taken into account when choosing the controller type – its computational complexity. Table 4 shows two factors related to the computational demands of the particular control strategy: T_{av} being the average computational time needed for the calculation of the optimal input and T_{max} corresponding to the maximum calculation time. Let us mention that this calculation time includes also the time needed to obtain the model which (as will be shown) might contribute considerably to the overall calculation time. The comparison is evaluated depending on the type of the model which is used by the optimizer. The simplest controller being the LMPC with LM needs the shortest time to calculate the optimal input. As this variant does not consume any time to obtain the model and the same optimizer is used also by second member of the family of the linear MPCs (the controller with SLM model), one can get a very good insight into how long does it take to obtain the SLM model for the predictions. As the SLM variant performs the approximation of the nonlinear model at each sampling instant, the increase of the average computational time is understandable. Although in case of the LMPC with SLM, the average calculation time is longer than in case of the LMPC with LM, this is compensated by the better control performance.

Let us summarize the performance of the particular variants. Regarding the control performance and the energy consumption, the NMPC is the best candidate for the real-life application. On the other hand, the LMPC with the simplest off-line identified model is able to provide the fastest calculation of the optimal input sequence. Looking for a trade-off between the optimality and the time complexity, the presented time-varying approach exploiting SLM model is able to bridge the gap between these two and therefore, it stands for a promising candidate for the real-life application especially in case of large buildings complexes where it can be expected that the nonlinear optimization task can take too long to be solved.

Table 4
Comparison of the energy consumption and computational complexity.

	LM	SLM	NM
TP	83.7	77.0	66.8
TP/TP_{LM} (%)	100	92	80
T_{av} (s)	0.81	0.93	4.41
T_{max} (s)	1.20	1.49	6.21

Since one of the main objectives of this paper was to adapt the nonlinear MPC such that it provided discrete-valued mass flow rate profile, let us present a comparison of the performance of the following alternatives – the naive a posteriori quantization that is referred to as nqNPC and the adaptation of the gradient algorithm named agqNPC are compared with the continuous-valued NMPC from the previous comparison. At first, the situation with 7 admissible values for mass flow rate was considered. All three compared controllers (continuous-valued NMPC, nqNPC and agqNPC) were tuned to achieve approximately the same thermal comfort and therefore, only the economical part of the criterion might be focused on. At first, the calculated mass flow rate profiles are presented in Fig. 8.

Based on the visual comparison, it can be expected that the

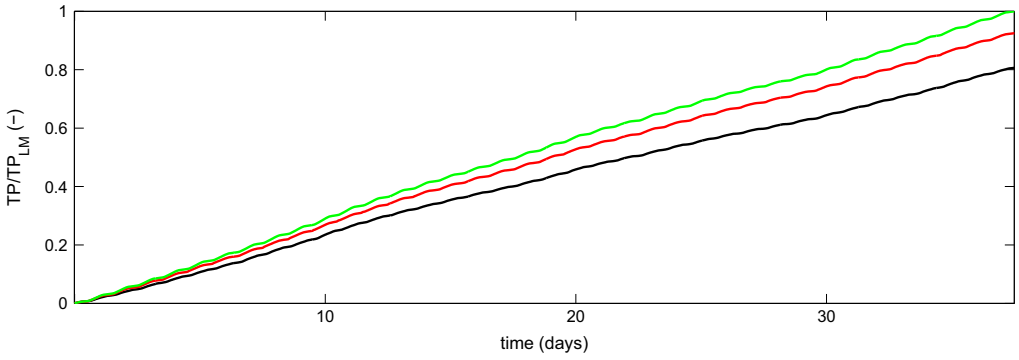

Fig. 7. Normalized cumulative price of the control effort (— – controller with LM , — – controller with SLM, — – controller with NM).

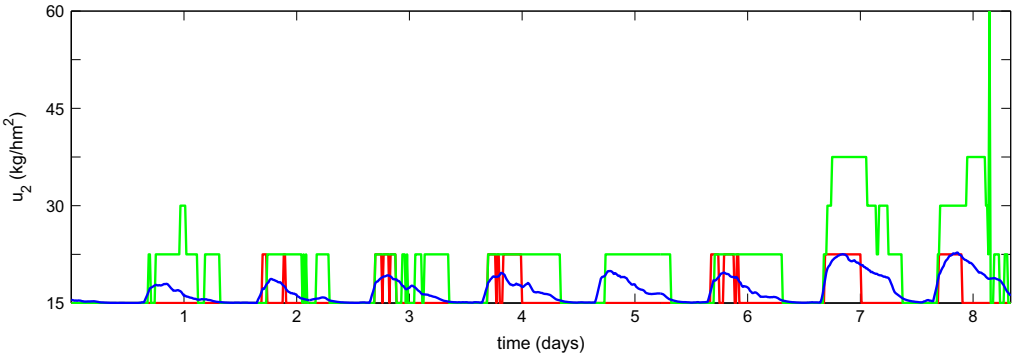

Fig. 8. Mass flow rates, $N_{qst} = 7$ (— – continuous-valued NMPC, — – nqNPC, — – agqNPC).

nqNPC pays the most for the operation of the building. On the other hand, the agqNPC with advanced handling of the quantization phenomena behaves more similar to the original continuous-valued NMPC. This demonstrates the fact that while within the agqNPC, the mid-processing iteration enables us to adapt the calculation of the mass flow rate *inside* the optimization procedure and take the quantization into account, the naive quantization does not provide such possibility and therefore, significant part of the optimality is lost. Moreover, a posteriori quantization obviously leads to more oscillatory profiles which stands for another drawback of such approach. Since the mass flow rate is not the only manipulated variable, it might be interesting to inspect how much affected is u_1 by the quantization of u_2. Such comparison is provided in Fig. 9 where the profiles of supply water temperature applied by the inspected controllers are shown.

Comparing Figs. 8 and 9, a waterbed effect of the quantization can be observed since the quantization of one manipulated variable causes oscillatory performance that "leaks" into the other manipulated variable profile. The situation might seem a little bit paradoxically – although the mass flow rate is the manipulated variable that is quantized, the other manipulated variable also strongly oscillates when comparing the quantized version with the original continuous-valued version of the controller. This is more significant in case of the nqNPC where the oscillations of the supply water temperature are much more aggressive than the oscillations of the mass flow rate. This can be explained by the fact that while the quantization of the mass flow rate projects the values belonging to particular interval to the same quantized value, no such "damping" applies to the supply water temperature and therefore, its oscillations fully develop.

The last part of the comparison is the numerical evaluation of the economical aspects of the control under the quantization conditions provided in Table 5. Besides the total control cost (denoted as NMPC, nqNPC and agqNPC according to the evaluated control algorithm) shown in euros, also percentage increases of energy consumption normalized with respect to the consumption achieved by continuous-valued NMPC are provided (in Table 5, the increases are referred to as EI_{nqNPC} and EI_{agqNPC}, respectively). To obtain a more reliable comparison, situations with 3 up to 8 quantization steps N_{qst} were compared. The range of quantization levels $N_{qst} \in \{3, \ldots, 8\}$ was chosen based on the actual market research – it turned out that none of the currently available water pumps offers use of more than 8 pre-programmed different speeds/mass flow rates and therefore, values of N_{qst} higher than 8 were not considered. On the other hand, the theory of optimal bang-bang (2-valued) control is nearly as mature and elaborated as the optimal control theory itself (Anderson & Moore, 1971; Kaya & Noakes, 1996; Ledzewicz & Schättler, 2002; Wonham & Johnson, 1964) – therefore the optimization problem with 2-valued valve was omitted and the lowest number of quantization levels was chosen as $\underline{N}_{qst} = 3$.

Inspecting Table 5, it is obvious that the increase of the quantization steps N_{qst} leads to decrease of the cost paid for the control – this holds for both the naive and advanced quantization handling. However, a considerable difference can be observed in the actual value of the control cost increase. While for the naive quantization algorithm nqNPC the control cost can be increased by as high portion as 28%, the control cost increase never exceeds 17% with the use of advanced agqNPC algorithm. The difference can be nicely illustrated on an example of $N_{qst} = 4$ steps. The advanced quantization algorithm agqNPC consumes only about 10% more energy than the continuous-valued NMPC while the naive quantization algorithm nqNPC cost increase is nearly twice as high – moreover, even with $N_{qst} = 6$ quantization steps, the nqNPC algorithm achieves worse control cost. The difference between the two algorithms turns

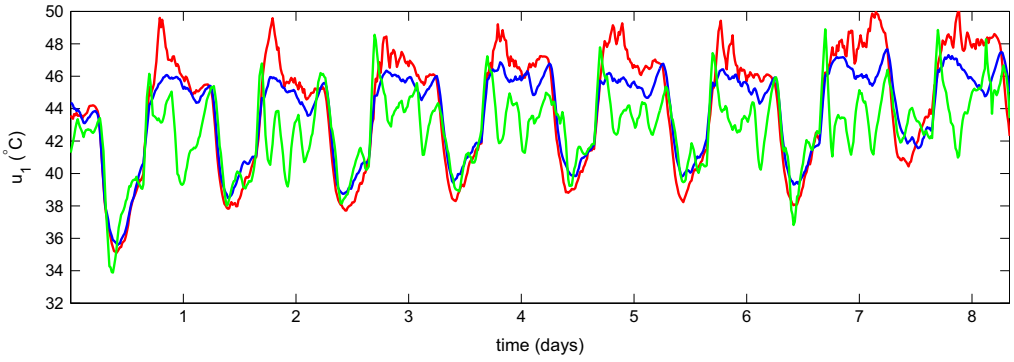

Fig. 9. Supply water temperatures, $N_{qst} = 7$ (— – continuous-valued NMPC, — – nqNPC, — – agqNPC).

Table 5
Comparison of the energy cost.

N_{qst}	NMPC	nqNPC	EI_{nqNPC}	aggNPC	EI_{aggNPC}
3	66.8	85.6	28.1	78.1	17.0
4	66.8	80.2	20.1	73.6	10.2
5	66.8	76.1	13.9	72.5	8.5
6	66.8	75.2	12.6	69.0	3.3
7	66.8	72.4	8.3	67.6	1.2
8	66.8	67.5	1.0	67.1	0.4

insignificant only for the highest number of quantization steps $N_{qst} = 8$. However, although the control cost increase might not be significant, the difference in handling the oscillatory effects should not be forgotten as documented in Fig. 10 which clearly shows that the high-frequency portion of both the mass flow rate and the supply water temperature signals is decreased by the aggNPC and brought closer to the continuous-valued NMPC.

Last but not least, the computational complexity should be mentioned. Since the naive quantization algorithm nqNPC involves only a post-processing procedure to handle the quantization, virtually no computational time increase compared with the continuous-valued NMPC is observed. In case of the advanced quantization algorithm aggNPC, the *mid-processing* iteration causes a constant average increase of the computational complexity $\Delta_{ct} = 0.28$ s representing about 6% of the average computational time of the continuous-valued NMPC. Here, it should be highlighted that the computational complexity increase introduced by the use of the aggNPC is *independent* of the number of quantization levels N_{qst} which strongly distinguishes it from the commonly used mixed-integer programming methods where the computational time rises very steeply even when using massive computational power (Causa et al., 2008; Geyer, Larsson, & Morari, 2003; Lenstra, 1983; Pancanti et al., 2002).

Given the combination of less oscillatory and more economical performance (compared with the naive quantization) and constant trifling time complexity increase, it can be concluded that the aggNPC is the better and more attractive choice for the industrial application of control with discrete-valued input variables.

6. Conclusion

The task of advanced building climate control was formulated, several ways how to solve it using model based predictive control paradigm (namely linear MPC with ordinary linear model, nonlinear MPC and linear MPC with time-varying linear model) were presented and chosen practically oriented aspects were discussed in this paper.

Since the modeling and the estimation of the unknown parameters of the model structures is crucial for proper functionality of the predictive controller, the first part of the paper was devoted to the related problems. MRI method that is known to be the appropriate choice for the identification for predictive controllers with linear model was used for identification of the linear model structure and furthermore, it was adapted for use in case of nonlinear model structures. With both presented variants of the nonlinear MRI algorithm, models with good prediction properties were obtained. Furthermore, a bridge between the nonlinear and linear model structure was introduced by a switched linear approximated model (SLM). All identified models (linear model, nonlinear model and SLM model) were tested on a series of verification data and the achieved results certified them for use within the MPC scheme.

The next part of the paper covers the design of the predictive controller. The algorithm for both linear and the nonlinear MPC were provided. According to the specifications of the control systems presented in the Introductory Section, one of the manipulated variables might not be set with infinite resolution. Therefore, certain adaptations of the predictive controllers were necessary. For the linear MPCs, the adaptation consisted only in change of admissible post-processing values for mass flow rate and therefore, it was not discussed in the paper. However, the adaptation of nonlinear MPC was more delicate. Out of the three possible options (use of mixed-integer programming, naive a posteriori quantization and inclusion of mid-processing iteration into the optimization routine), the first one was abandoned due to its high computational requirements. While the naive a posteriori quantization represents only another post-processing procedure, the last option with the mid-processing iteration of the optimization algorithm adapts the original continuous-valued optimization and incorporates the information about the quantization directly into the optimization routine.

All the presented controllers were compared with respect to the pre-defined evaluative criterion based on the real-life requirements and costs. The results demonstrate that although the nonlinear continuous-valued predictive controller addresses the minimization of the given evaluative criterion in the best way, it was also quite time consuming. Therefore, the linear MPC exploiting the SLM model can be regarded as a reasonable trade-off between the optimality of the solution and the time complexity of the underlying optimization, especially in case of huge centrally-controlled building complexes where the complexity of the optimization task can be very high.

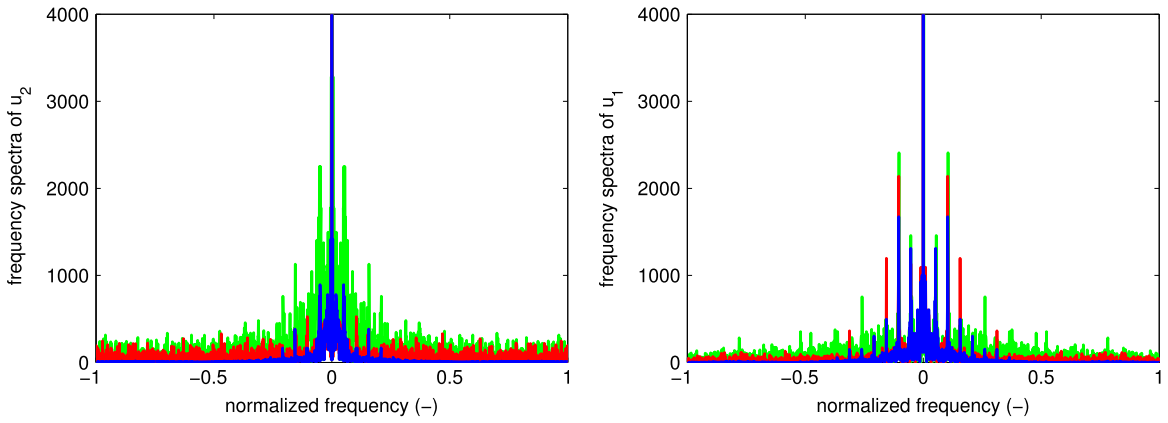

Fig. 10. Frequency spectra of the optimized variables, $N_{qst} = 8$ (— – continuous-valued NMPC, — – nqNPC, — – aggNPC).

The other part of the comparison was focused on evaluation of the performance of the nonlinear controllers under the restrictions on the discrete-valued nature of the mass flow rate. The provided comparison shows that the advanced of the inspected methods agqNPC is not only able to keep the economical aspects of the control closer to standard of the original continuous-valued controllers but also helps us to reduce the oscillations of the manipulated variables for the cost of a negligible *constant* time complexity increase. Therefore, it can be suggested that the advanced agqNPC algorithm be used in practice instead of naive but commonly frequently used a posteriori quantization.

Regarding the future work, it would be interesting to examine the effect of incorporation of the persistent excitation condition into the predictive controller procedure. Based on the available literature, if the persistent excitation condition is included, more informative data are obtained which then turns into a better ability to estimate the model parameters accurately. The suggested procedure should be compared with the advanced Kalman filtering algorithms such as Extended or Unscented Kalman filtering. Moreover, a procedure for the model parameter update should be designed for the nonlinear model. Last but not least, based on the performed numerical experiments the authors suggest the strategies be tested on a building in real operation.

Acknowledgment

This research has been supported by the Czech Science Foundation through the Grant nos. 13-20433S and 13-12726J.

References

Anderson, B. D. & Moore, J. B. (1971). *Linear optimal control* (Vol. 197). Englewood Cliffs: Prentice-Hall.

ASHRAE. (2009). *ASHRAE-handbook fundamentals*. NE Atlanta, GA: American society of heating, refrigerating and air-conditioning engineers, Inc.

Balmer, R. T. (2010). *Modern engineering thermodynamics-textbook with tables booklet*. Academic Press, Oxford, UK.

Barták, M. (2010). *Úvod do přenosových jevu pro inteligentní budovy*. Prague, Czech Republic: CTU Prague.

Bryson, A. E., & Ho, Y. C. (1975). *Applied optimal control: optimization, estimation, and control*. Taylor and Francis, Oxford, UK.

Bussieck, M. R. & Vigerske, S. (2010). MINLP solver software. In *Wiley encyclopedia of operations research and management science*.

Causa, J., Karer, G., Núnez, A., Sáez, D., Škrjanc, I., & Zupančič, B. (2008). Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor. *Computers & Chemical Engineering*, *32*(12), 3254–3263.

Chi, Q., Fei, Z., Zhao, Z., Zhao, L., & Liang, J. (2014). A model predictive control approach with relevant identification in dynamic PLS framework. *Control Engineering Practice*, *22*, 181–193.

European Economic and Social Committee. (2005). S.M. act: Communication from the commission to the council, The European parliament, The European economic and social committee and the committee of the regions.

Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., & Hrovat, D. (2008). Linear time-varying model predictive control and its application to active steering systems: *Stability analysis and experimental validation*. *International Journal of Robust and Nonlinear Control*, *18*(8).

Geyer, T., Larsson, M., & Morari, M. (2003). Hybrid emergency voltage control in power systems. In *Proceedings of the European control conference*, 2003.

Gopaluni, R., Patwardhan, R., & Shah, S. (2004). MPC relevant identification—Tuning the noise model. *Journal of Process Control*, *14*(6), 699–714.

Gyalistras, D., & Gwerder, M. (2010). Use of weather and occupancy forecasts for optimal building climate control (OptiControl): Two years progress report. Terrestrial Systems Ecology ETH Zurich, Switzerland and Building Technologies Division, Siemens Switzerland Ltd., Zug, Switzerland.

Kaya, C. Y., & Noakes, J. L. (1996). Computations and time-optimal controls. *Optimal Control Applications and Methods*, *17*(3), 171–185.

Laurí, D., Salcedo, J., Garcia-Nieto, S., & Martínez, M. (2010). Model predictive control relevant identification: *Multiple input multiple output against multiple input single output*. *Control Theory & Applications, IET*, *4*(9).

Ledzewicz, U., & Schättler, H. (2002). Optimal bang–bang controls for a two-compartment model in cancer chemotherapy. *Journal of Optimization Theory and Applications*, *114*(3), 609–637.

Lenstra, H. W., Jr. (1983). Integer programming with a fixed number of variables. *Mathematics of Operations Research*, *8*(4), 538–548.

Lienhard, J. H. (2013). *A heat transfer textbook*. Courier Corporation, Cambridge, Massachusetts, USA.

Ljung, L. (1999). *System identification*. Wiley Online Library.

Ljung, L. (2007). System identification toolbox for use with {MATLAB}.

Ma, J., Qin, S. J., Li, B., & Salsbury, T. (2011). Economic model predictive control for building energy systems. In *2011 IEEE PES innovative smart grid technologies (ISGT)* (pp. 1–6). IEEE, Piscataway, NJ, USA.

Ma, Y., Kelman, A., Daly, A., & Borrelli, F. (2012). Predictive control for energy efficient buildings with thermal storage: *Modeling, stimulation, and experiments*. *IEEE Control Systems, Magazine*, *32*(1), 44–64.

Oldewurtel, F., Gyalistras, D., Gwerder, M., Jones, C., Parisio, A., Stauch, V., et al. (2010). Increasing energy efficiency in building climate control using weather forecasts and model predictive control. In *10th REHVA world congress clima* (pp. 9–12).

Pancanti, S., Leonardi, L., Pallottino, L., & Bicchi, A. (2002). Optimal control of quantized input systems. In *Hybrid systems: computation and control* (pp. 351–363). Springer, Berlin Heidelberg, Germany.

Perez-Lombard, L., Ortiz, J., & Pout, C. (2008). A review on buildings energy consumption information. *Energy and Buildings*, *40*(3), 394–398.

Prívara, S., Široký, J., Ferkl, L., & Cigler, J. (2011). Model predictive control of a building heating system: *The first experience*. *Energy and Buildings*, *43*(2), 564–572.

Pčolka, M., Žáčeková, E., Robinett, R., Čelikovský, S., & Šebek, M. (2014). Economical nonlinear model predictive control for building climate control. In *American control conference (ACC), 2014* (pp. 418–423). IEEE, Piscataway, NJ, USA.

Pčolka, M., Žáčeková, E., Robinett, R., Čelikovský, S., & Šebek, M. (2014). From linear to nonlinear model predictive control of a building. In *IFAC world congress 2014* (Vol. 19, pp. 587–592).

R.O. for Network Industries. (2011). Approved electricity tariffs for the household consumers for 2011 (online). ⟨http://www.urso.gov.sk/doc/dokumenty/PorovnanieMaxCienEEpreDomacnisti2011.pdf⟩, accessed: 03/11/2013.

Shook, D. S., Mohtadi, C., & Shah, S. L. (1991). Identification for long-range predictive control. In *IEEE proceedings D (Control theory and applications)* (Vol. 138, pp. 75–84). IET, Piscataway, NJ, USA.

Stetter, H. J. (1973). *Analysis of discretization methods for ordinary differential equations* (Vol. 23). Springer, Berlin Heidelberg, Germany.

University of Wisconsin-Madison, Solar Energy Laboratory, & Klein, S. A. (1979). *TRNSYS, a transient system simulation program*. Solar Energy Laboratory, University of Wisconsin–Madison.

Verhelst, C., Degrauwe, D., Logist, F., Van-Impe, J., & Helsen, L. (2012). Multi-objective optimal control of an air-to-water heat pump for residential heating. *Building Simulation*, *5*, 281–291.

Wonham, W. M., & Johnson, C. (1964). Optimal bang–bang control with quadratic performance index. *Journal of Fluids Engineering*, *86*(1), 107–115.

Žáčeková, E., & Prívara, S. (2012). Control relevant identification and predictive control of a building. In *2012 24th Chinese control and decision conference (CCDC)* (pp. 246–251). IEEE, Piscataway, NJ, USA.

Žáčeková, E., Váňa, Z., & Cigler, J. (2014). Towards the real-life implementation of MPC for an office building: *Identification issues*. *Applied Energy*, *135*, 53–62.

Zhao, J., Zhu, Y., & Patwardhan, R. (2014). Some notes on MPC relevant identification. In *American control conference (ACC), 2014* (pp. 3680–3685). IEEE, Piscataway, NJ, USA.

Zhou, K., Doyle, J. C., Glover, K., (1996). *Robust and optimal control* (Vol. 40). New Jersey: Prentice Hall, Upper Saddle River, New Jersey, USA.

Zmrhal, V., & Drkal, F. (2006). The influence of heat convection on room air temperature. In *Proceedings of 17th air-conditioning and ventilation conference*.

5.2 Discontinuous Dynamics/Discontinuous Cost Optimization

Control of hybrid systems poses a considerable number of challenges [Sanfelice, 2013]. Thus, for the optimization tasks dealing with systems with discontinuous dynamics and/or piecewise continuous cost criteria, the common implementation consists in approximating the discontinuous dynamics/optimization cost near the current operating point by a (nonlinear) function corresponding to the last measured state/disturbance values and the last applied inputs. This way, the discontinuities are eliminated since the operating-point approximation is used over the whole prediction horizon and the resulting optimization task can be solved by ordinary (non)linear programming solvers. This approximation is performed at each discrete time instant *prior to* optimizing the input variables and thus it can be referred to as *a priori switching* MPC (APS-MPC). This naive and computationally cheap approach works fairly well for slow systems with sufficiently "wide" piecewise continuous intervals occurring in the description of the system dynamics/optimization criterion. This, however, compromises the quality of the achieved control performance and, moreover, such simplifying assumption might not be valid for a substantial class of control tasks. Alternatively, there are also methods that avoid use of approximations and solve the discontinuous task as is, nevertheless, they either *i)* rely on knowledge of the future dynamics switches over the optimization horizon [Xu and Antsaklis, 2000], [Giua et al., 2001], [Xu and Antsaklis, 2004], [Mhaskar et al., 2005], [Wu et al., 2006], which is a knowledge that is only very rarely at disposal; or *ii)* perform a possibly time consuming pre-processing and solve the original task by dividing the non-smooth problem into multiple smooth problems with additional constraints [Xu et al., 2015], which further increases the computational demands; or *iii)* restrict to only two possible sub-dynamics [Bengea and DeCarlo, 2005]; or *iv)* deal with piecewise affine/mixed logical and dynamical systems considering only linear set of sub-dynamics [Camacho et al., 2010], [Alexis et al., 2011]. As already mentioned in the previous section, it is also possible to formulate and solve the task by means of mixed-integer nonlinear programming techniques, but substantial growth of computational demands should be anticipated.

Being another part of this thesis, a novel optimization algorithm (HaSH-NPC) addressing these issues was developed and presented in [A.3]. The underlying idea is as follows: consider an optimization task involving a system with discontinuous dynamics being a function of system states, manipulated/disturbance inputs and a discrete-valued *dynamics-*

switcher which indicates the particular sub-dynamics consisting in minimization of a cost criterion defined as a piecewise continuous function of the states, inputs and a discrete-valued *criterion-switcher* identifying the particular sub-criterion. Next, a piecewise continuous Hamiltonian can be derived as a function of the system states, inputs/disturbances, adjoint states and a *hamiltonian-switcher*, an auxiliary variable mapping the dynamics- and criterion-switcher to a set of all possible Hamiltonian relations for the piecewise continuous optimization problem. Then, during the online iterative calculations, the derivatives of the piecewise continuous Hamiltonian obtained employing the dynamics-, criterion- and hamiltonian-switcher profiles evaluated along the input trajectories corresponding to the previous iteration are used to update the optimized input sequences; this iterative procedure is applied until a predefined convergence criteria are satisfied. An extension of the HaSH-NPC algorithm was also provided consisting in deriving formulas for adaptive choice of the optimization horizon with a detailed description being presented in Chapter 4.3. The performance of HaSH-NPC algorithm was compared with that of APS-MPC and validated on an example of a vehicle control considering hybrid race car dynamics and a piecewise continuous cost criterion. It was demonstrated that even with shorter constant prediction horizon, HaSH-NPC outperforms the commonly used approach in both the performance index maximization and safety requirements satisfaction. Moreover, a comparison with solutions obtained with a commercial mixed-integer nonlinear programming solver was presented in the mentioned paper and it was shown that while having comparable performance, the mixed-integer solver consumes incomparably more time than the HaSH-NPC algorithm. Last but not least, a very detailed sensitivity analysis was also performed and the results confirmed solid robustness against parameters/dynamics modelling mismatches.

The publication mentioned above is presented in the original formatting and is provided on the next page et seq.

Toward a Smart Car: Hybrid Nonlinear Predictive Controller With Adaptive Horizon

Matej Pčolka, Eva Žáčeková, Sergej Čelikovský, *Senior Member, IEEE,* and Michael Šebek, *Senior Member, IEEE*

***Abstract*— This paper focuses on the development of an optimization algorithm for car motion predictive control that addresses both hybrid car dynamics and hybrid minimization criterion. Instead of solving computationally demanding nonlinear mixed-integer programming task or approximating the hybrid dynamics/criterion, the *Hamiltonian-switching hybrid nonlinear predictive control* algorithm developed in this paper incorporates the information about hybridity directly into the optimization routine. To decrease the time complexity, several adaptive prediction horizon approaches are proposed, and for some of them, it is shown that they preserve maneuverability-related properties of the car. All developed alternatives are verified on an example of a motion control of a racing car and compared with the approximation-based nonlinear predictive control and a commercial product. Moreover, a sensitivity analysis examining robustness of the algorithm is included as well.**

***Index Terms*— Autonomous vehicles, hybrid systems, nonlinear model predictive control (MPC), optimization, vehicle control.**

I. Introduction

AUTOMOTIVE industry is one of the most dynamic engineering branches. Recently, huge progress toward an autonomous car has been witnessed [1]–[5], and out of the control methods able to replace a human driver, the model predictive control (MPC) is the most perspective one.

The most frequent variant is the linear MPC [4], [6], [7]. Although computationally simple, simplifications of the nonlinear dynamics/criterion provide only suboptimal performance. Some works present nonlinear MPC, however, they usually focus only on steering control [1], [8], [9]. The nonlinear MPC proposed in this paper manipulates steering, acceleration *and* braking and both satisfies the safety constraints and maximizes the performance indicators.

Complications arise from the strongly nonlinear sideslipping effects. One option is to model them by a steering coefficient being a piecewise continuous function of the forward velocity yielding a hybrid car dynamics model. Its attractiveness consists in replacing one complicated nonlinear function with a series of simpler subfunctions, as exploited in aerospace applications, chemical or electrical engineering [10]–[13].

Usually, mixed-integer programming (MIP) is exploited to handle the hybrid optimal control problems [14]–[16]; however, the nonlinear MIP tasks are NP-hard with exponentially growing time/computational demands [17], [18]. Some works [19], [20] propose alternatives, however, they either rely on restrictive assumptions such as *a priori* knowledge of the subdynamics sequence or perform a possibly time-consuming preprocessing. In this paper, we develop an algorithm requiring no such knowledge that avoids complex preprocessing, exploits *Hamiltonian-switcher* and solves the given optimization task directly as an ordinary nonlinear-programming task.

The computational burden is one of the weaknesses of the optimization-based approaches. Although dividing the "global" control task into smaller pieces and using decentralized approach [21] decreases the computational complexity, the price is the loss of optimality. However, since the complexity of the optimization task depends on the length of the optimization horizon, it can be reduced using adaptive horizon. In this paper, several alternatives are proposed with certain safety guarantees.

This paper is organized as follows. Section II introduces the vehicle behavior description, control requirements and constraints. Section III deals with the mathematical aspects of the problems the newly proposed algorithm focuses at, formulates a novel Hamiltonian-switcher-based algorithm and explains adaptive prediction horizon approaches. In Section IV, the results obtained from the numerical experiments are presented. Section V inspects the robustness of the proposed control algorithm with respect to parameter perturbations. Section VI concludes this paper.

II. Car Motion Modeling, Objectives, and Constraints

In the role of the test-bed system, a racing car with a hybrid steering coefficient was chosen.

A. Car Modeling

Car dynamics modeling is a highly delicate task, since the real car behavior is influenced by many factors, which

Manuscript received May 31, 2017; accepted August 14, 2017. Manuscript received in final form August 26, 2017. This work was supported in part by the Czech Science Foundation through the Research Grant under Grant 17-04682S and in part by the European Regional Development Fund through the Project Rob4Ind4.0 under Grant CZ.02.1.01/0.0/0.0/15_003/0000470. Recommended by Associate Editor J. Sun. *(Corresponding author: Matej Pčolka.)*

M. Pčolka and E. Žáčeková are with the Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, 121 35 Prague, Czech Republic (e-mail: pcolkmat@fel.cvut.cz; zacekeva@fel.cvut.cz).

S. Čelikovský is with the Institute of Information Theory and Automation, Czech Academy of Sciences, 182 08 Prague, Czech Republic (e-mail: celikovs@utia.cas.cz).

M. Šebek is with the Department of Control Engineering, Faculty of Electrical Engineering , Czech Technical University in Prague, 121 35 Prague, Czech Republic, and also with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, 121 35 Prague, Czech Republic (e-mail: sebekm1@fel.cvut.cz; sebekm1@ciirc.cvut.cz).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2017.2747504

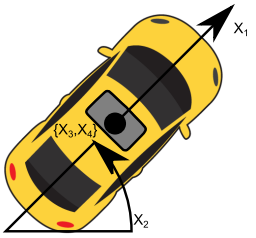

Fig. 1. State variables.

1) are constant (car mass, size and wheelbase), through those that 2) vary slightly/slowly (e.g., road inclination) up to those that 3) are highly nonlinear/stochastic [aerodynamic (im)perfections and their influence, car/road technical conditions and others].

In the literature, two main modeling branches are followed: kinematic (nonholonomic) modeling [22] and dynamic modeling [23]. While the dynamic modeling provides accurate models useful for simulation and analysis, the kinematic (nonholonomic) models are simpler and have low computational requirements, which is attractive for model-based control systems. On the other hand, they do not capture more complicated behavior, such as sideslipping. In this paper, this is overcome by a hybrid coefficient that models sideslipping as a decrease of the steering effectiveness.

The car dynamics is considered as follows:

$$\begin{aligned} x_{1,k+1} &= (p_1 - p_2 B_k) x_{1,k} + p_3 D_k, \\ x_{2,k+1} &= x_{2,k} + p_4 \alpha(x_{1,k}) \tan(S_k) x_{1,k}, \\ x_{3,k+1} &= x_{3,k} + p_5 \cos(x_{2,k}) x_{1,k}, \\ x_{4,k+1} &= x_{4,k} + p_5 \sin(x_{2,k}) x_{1,k}, \end{aligned} \tag{1}$$

where the forward speed $x_1\,(\mathrm{ms}^{-1})$, vehicle orientation $x_2\,(\mathrm{rad})$ and its x- and y-position $\{x_3,\, x_4\}\,(\mathrm{m})$ represent the state vector $x = [x_1, x_2, x_3, x_4]^{\mathrm{T}}$. For visualization, see Fig. 1. Regarding the manipulated variables $u = [D,\, B,\, S]^{\mathrm{T}}$, they correspond to normalized acceleration force $D\,(-)$, normalized braking force $B\,(-)$ and steering angle $S\,(\mathrm{rad})$.

p_1 expresses how much the car velocity is preserved in the no-gas-no-braking case and meaningful values are those close to 1. Driving a noninclined road with a tarmac surface, p_1 is typically slightly lower than 1 mainly due to ubiquitous friction and air resistance. Lower sub-1 values are caused by driving a rougher terrain (increased friction), uphill driving (effect of gravitational force) or aerodynamic imperfections (increased drag coefficient), while slightly super-1 values indicate downhill driving. In this paper, $p_1 = 0.999$.

p_2 represents the braking effect. Since the braking deceleration can vary from 4.5 up to almost $9.8\,\mathrm{ms}^{-2}$, with the sampling period of 0.1 s and $p_1 = 0.999$, p_2 can range from slightly less than 0.02 to slightly more than 0.04 depending on the velocity, vehicle/road conditions and properties (wet/icy road, bald tires and mass distribution). $p_2 = 0.03$ used in model (1) yields a 100-to-0 kph braking distance of around 85 m, a reasonable value for rather unpaved surfaces.

TABLE I
SYSTEM PARAMETERS

Parameter	Value	Parameter	Value
$p_1\,(-)$	99.9×10^{-2}	$a_1\,(\mathrm{m}^{-1}\mathrm{s})$	-4.5×10^{-2}
$p_2\,(-)$	3×10^{-2}	$a_2\,(-)$	181×10^{-2}
$p_3\,(\mathrm{ms}^{-1})$	35×10^{-2}	$a_3\,(-)$	1913.22×10^{-1}
$p_4\,(\mathrm{rad\,m}^{-1}\mathrm{s})$	36.36×10^{-3}	$a_4\,(\mathrm{m}^{-1}\mathrm{s})$	-19.15×10^{-2}
$p_5\,(\mathrm{s})$	1×10^{-1}	$\{\underline{D}, \overline{D}\}\,(-)$	$\{0, 1\}$
$v_1\,(\mathrm{ms}^{-1})$	18	$\{\underline{B}, \overline{B}\}\,(-)$	$\{0, 1\}$
$v_2\,(\mathrm{ms}^{-1})$	35	$\{\underline{S}, \overline{S}\}\,(\mathrm{rad})$	$\{-\pi/6, \pi/6\}$

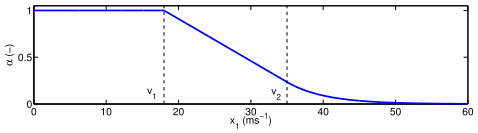

Fig. 2. Hybrid steering coefficient $\alpha(x_1)$.

The acceleration modeled by p_3 is affected by factors similar to those influencing p_1 and p_2. $p_3 = 0.35\,\mathrm{ms}^{-1}$ chosen here corresponds to 0-to-100 kph time around 8.5 s.

p_4 reflecting the influence of the car velocity and steering command on the car orientation is obtained as a product of the sampling period and the reciprocal of the wheelbase of the vehicle. In this paper, $p_4 = 36.36 \times 10^{-3}$ is assumed, which corresponds to a wheelbase of 2.75 m.

As mentioned, nonholonomic models describe the vehicle dynamics with sufficient accuracy at lower speeds and considering perfect adherence, however, they do not reflect sideslipping effects. In this paper, the sideslipping is interpreted as a decrease of steering effectiveness and is modeled by a piecewise continuous coefficient $\alpha(x_1)$ that equals 1 at lower speeds $x_1 \leq v_1$, decreases linearly between v_1 and v_2, and decays exponentially at speeds $x_1 > v_2$:

$$\alpha(x_1) = \begin{cases} \alpha_1(x_1) = 1 & 0 \leq x_1 \leq v_1, \\ \alpha_2(x_1) = a_1 x_1 + a_2 & v_1 < x_1 \leq v_2, \\ \alpha_3(x_1) = a_3 \exp(a_4 x_1) & v_2 < x_1. \end{cases} \tag{2}$$

For graphical interpretation of $\alpha(x_1)$, see Fig. 2. The parameters $\{p_1,\ p_2,\ p_3,\ p_4,\ p_5\}$, $\{v_{1,2}\}$ and $\{a_1, a_2, a_3, a_4\}$ of (1) and (2) are provided in Table I. Further information on car dynamics and modeling can be found in [22]–[25].

B. Objectives and Constraints

In automobile racing, the lap time is usually minimized. This can be transformed into speed x_1 maximization, which then stands for the performance part of the overall criterion.

The second aspect of the same (if not even greater) importance is the safety, which turns into a requirement that the car stays on the track with a predefined width W. As usual in car racing, some predefined tolerance Δ_r is admitted.

The only technical constraints are those imposed on the manipulated variables D, B and S:

$$\underline{D} \leq D \leq \overline{D}, \quad \underline{B} \leq B \leq \overline{B}, \quad \underline{S} \leq S \leq \overline{S}. \tag{3}$$

The numerical values of $\{\underline{D}, \overline{D}\}$, $\{\underline{B}, \overline{B}\}$ and $\{\underline{S}, \overline{S}\}$ can be found in Table I.

III. Controller Design

In this section, a novel optimization algorithm for hybrid nonlinear predictive control is proposed and its application to the investigated task is explained.

A. Hybrid Nonlinear Predictive Control Algorithm

Let us introduce a general description of a discrete-time system with switched dynamics as follows:

$$x_{k+1} = F(x_k, u_k, \mathfrak{s}_{d,k}) \tag{4}$$

where the *dynamics switcher* $\mathfrak{s}_{d,k} = \mathcal{S}_d(x_k, u_k)$ indicating the current system dynamics is obtained by a mapping $\mathcal{S}_d : \mathbb{R}^{n+m} \mapsto \{1, 2, \ldots, N_d\}$. Here, n and m are the dimensions of states x and inputs u, and $N_d \in \mathbb{N}^+$ is the number of switched dynamics. Moreover, let

$$\begin{aligned} F(x_k, u_k, 1) &= f_1(x_k, u_k), \\ F(x_k, u_k, 2) &= f_2(x_k, u_k), \\ &\vdots \\ F(x_k, u_k, N_d) &= f_{N_d}(x_k, u_k), \end{aligned} \tag{5}$$

where $f_{\mathfrak{s}_{d,k}}(x_k, u_k)$ expresses the particular subdynamics.

The hybrid optimization criterion $\mathcal{J}$ minimized at each time k is considered in the following form:

$$\mathcal{J} = \sum_{i=k+1}^{k+P} L(x_i, u_i, \mathfrak{s}_{c,i}) \tag{6}$$

with prediction horizon $P \in \mathbb{N}^+$. Next, assume that the function L can be expressed as

$$\begin{aligned} L(x_k, u_k, 1) &= l_1(x_k, u_k), \\ L(x_k, u_k, 2) &= l_2(x_k, u_k), \\ &\vdots \\ L(x_k, u_k, N_c) &= l_{N_c}(x_k, u_k), \end{aligned} \tag{7}$$

with $l_{\mathfrak{s}_{c,k}}(x_k, u_k)$ being the particular subcriterion term. Here, the *criterion switcher* $\mathfrak{s}_{c,k} = \mathcal{S}_c(x_k, u_k)$ is obtained by a mapping $\mathcal{S}_c : \mathbb{R}^{n+m} \mapsto \{1, 2, \ldots, N_c\}$, where $N_c \in \mathbb{N}^+$ is the number of the hybrid parts of the cost criterion term L.

Then, the optimization task is summarized as follows.

For given initial condition x^-, find

$$u^* = \arg\min \mathcal{J}(x, u, \mathfrak{s}_d, \mathfrak{s}_c) \tag{8}$$

with respect to

$$\begin{aligned} x_{k+1} &= F(x_k, u_k, \mathfrak{s}_{d,k}), \\ u_{\min} &\le u \le u_{\max}, \\ \mathfrak{s}_{d,k} &= \mathcal{S}_d(x_k, u_k) \in \{1, 2, \ldots, N_d\}, \\ \mathfrak{s}_{c,k} &= \mathcal{S}_c(x_k, u_k) \in \{1, 2, \ldots, N_c\}. \end{aligned} \tag{9}$$

The common implementation [further referred to as *a priori switching MPC* (APS-MPC) algorithm] performed at each sampling instance k is as described by Algorithm 1.

The APS-MPC approach eliminates the hybridity by evaluating $\mathcal{S}_d$ and $\mathcal{S}_c$ prior to solving the optimization task and assuming $\mathfrak{s}_d$ and $\mathfrak{s}_c$ constant over the whole P, which enables use of the standard NLP solvers instead of more demanding MINLP in step 2 of APS-MPC.

Algorithm 1 APS-MPC

1) according to the last measured states x^- and the last applied inputs u^-, evaluate the dynamics switcher $\mathfrak{s}_d = \mathcal{S}_d(x^-, u^-)$ and the criterion switcher $\mathfrak{s}_c = \mathcal{S}_c(x^-, u^-)$;
2) find u^* minimizing $\mathcal{J} = \sum_{i=k+1}^{k+P} l_{\mathfrak{s}_c}(x_i, u_i)$ such that $x_{k+1} = f_{\mathfrak{s}_d}(x_k, u_k)$ with the state initial conditions x^- and $u_{\min} \le u \le u_{\max}$;
3) apply the first sample of the optimized input u^* into the system, wait for the next measurement, repeat from 1).

While for simple tasks, potential issues with the validity of the approximation are not crucial, a suitable alternative needs to be found for less trivial cases. Here, one such alternative—*Hamiltonian-switching hybrid nonlinear predictive control* (HaSH-NPC) algorithm—is derived as an adaptation of the Hamiltonian-based gradient method [26], [27].

The original gradient algorithm makes use of the Hamiltonian $H(x, u, \lambda) = \lambda_{k+1}^{\mathrm{T}} F(x_k, u_k) + L(x_k, u_k)$. In the hybrid case with system dynamics (5) and criterion term (7), it can be derived that

$$\mathcal{H}(x, u, \lambda) = \begin{cases} \lambda_{k+1}^{\mathrm{T}} f_1(x_k, u_k) + l_1(x_k, u_k) & \text{iff } \mathfrak{s}_d = 1,\ \mathfrak{s}_c = 1, \\ \lambda_{k+1}^{\mathrm{T}} f_1(x_k, u_k) + l_2(x_k, u_k) & \text{iff } \mathfrak{s}_d = 1,\ \mathfrak{s}_c = 2, \\ \vdots & \\ \lambda_{k+1}^{\mathrm{T}} f_1(x_k, u_k) + l_{N_c}(x_k, u_k) & \text{iff } \mathfrak{s}_d = 1,\ \mathfrak{s}_c = N_c, \\ \lambda_{k+1}^{\mathrm{T}} f_2(x_k, u_k) + l_1(x_k, u_k) & \text{iff } \mathfrak{s}_d = 2,\ \mathfrak{s}_c = 1, \\ \vdots & \\ \lambda_{k+1}^{\mathrm{T}} f_{N_d}(x_k, u_k) + l_{N_c}(x_k, u_k) & \text{iff } \mathfrak{s}_d = N_d,\ \mathfrak{s}_c = N_c. \end{cases}$$

To make the above description more compact, let us introduce a *Hamiltonian-switcher* $\mathfrak{s}_h = \mathcal{S}_h(\mathfrak{s}_d, \mathfrak{s}_c)$

$$\mathcal{S}_h : \{1, 2, \ldots, N_d\} \times \{1, 2, \ldots, N_c\} \mapsto \{1, 2, \ldots, N_h\}. \tag{10}$$

Here, $N_h \in \mathbb{N}^+$ corresponds to the number of all possible Hamiltonian relations for the hybrid optimization problem. The mapping $\mathcal{S}_h$ can be with advantage chosen as

$$\mathcal{S}_h(\mathfrak{s}_d, \mathfrak{s}_c) = (\mathfrak{s}_d - 1)N_c + \mathfrak{s}_c. \tag{11}$$

Then, the hybrid-problem Hamiltonian can be constructed as

$$\mathcal{H}(x, u, \lambda, \mathfrak{s}_h) = H_{\mathfrak{s}_h}(x, u, \lambda) = \lambda_{k+1}^{\mathrm{T}} f_{\mathfrak{s}_d}(x_k, u_k) + l_{\mathfrak{s}_c}(x_k, u_k). \tag{12}$$

Instead of approximating the hybridity, the HaSH-NPC algorithm handles the problem correctly as described by Algorithm 2.

Let us note that the search step length choice is a highly complicated and still open question. While computationally least demanding, constant search steps often provide poor convergence. On the other hand, search steps obtained by a line search usually yield best convergence, however, their, calculation might be prohibitively time-consuming. A fair tradeoff

Algorithm 2 HaSH-NPC

Obtain the state measurements x^- and initial input profile estimate U^0; then iteratively repeat:

1) use the initial condition $x_0 = x^-$ corresponding to the currently measured states and input profile from the previous iteration U^{l-1} to obtain the state trajectories $X = [x_0, x_1, \ldots, x_P]$ according to (4); store the dynamics-switcher profile $S_d = [\mathfrak{s}_{d,1}, \mathfrak{s}_{d,2}, \ldots, \mathfrak{s}_{d,P}]$;
2) evaluate the criterion-switcher mapping $\mathcal{S}_c(X, U^{l-q})$ and obtain the criterion-switcher profile $S_c = [\mathfrak{s}_{c,1}, \mathfrak{s}_{c,2}, \ldots, \mathfrak{s}_{c,P}]$;
3) according to mapping (11), obtain hamiltonian-switcher profile $S_h = [\mathfrak{s}_{h,1}, \mathfrak{s}_{h,2}, \ldots, \mathfrak{s}_{h,P}]$;
4) create the piece-wise continuous Hamiltonian $\mathcal{H}(x, u, \lambda, \mathfrak{s}_h)$ according to (12), calculate its derivatives with respect to x and u;
5) evaluate the co-state backward dynamics
$$\lambda_k = \frac{\partial \mathcal{H}}{\partial x}(x_k, u_k^{l-1}, \lambda_{k+1}, \mathfrak{s}_{h,k}) = \frac{\partial \mathcal{H}_{\mathfrak{s}_{h,k}}}{\partial x}(x_k, u_k^{l-1}, \lambda_{k+1})$$
with $\lambda_P = \partial J / \partial x|_P$; obtain the co-state trajectory $\Lambda = [\lambda_0, \lambda_1, \ldots, \lambda_P]$;
6) evaluate the gradient $\partial \mathcal{H}/\partial u$, obtain input profile U^l by performing the gradient step as follows:
$$\begin{aligned} U^l &= U^{l-1} - \alpha_l \frac{\partial \mathcal{H}}{\partial u}(X, U^{l-1}, \Lambda, S_h), \\ &= U^{l-1} - \alpha_l \frac{\partial \mathcal{H}_{S_h}}{\partial u}(X, U^{l-1}, \Lambda); \end{aligned} \tag{13}$$
7) project U^l on the admissible interval $\langle u_{\min}, u_{\max} \rangle$;
8) **if** $\|U^l - U^{\{*,l-1\}}\| \leq \epsilon_1 \vee \left|J(U^l) - J(U^{l-1})\right| \leq \epsilon_2$, $\epsilon_1 > 0$, $\epsilon_2 > 0$, where
$$U^{\{*,l-1\}} = \arg\min(\min\{J(U^0), J(U^1), \ldots, J(U^{l-1})\})$$
then terminate; apply the first sample of $U^{\{*,l\}}$ into the system, wait for the new measurements, go to 1);
else $l = l + 1$, repeat from 1).

is offered by the heuristically chosen cost-function-dependent search steps that 1) are small (and prevent oscillations) if the cost criterion decreases rapidly and 2) increase (and speed up the convergence) if the cost criterion change is small. Moreover, their computational demands are negligible, since they can be expressed analytically. In this paper, the search step length α_l is considered as a function of the cost criterion value decrease $\Delta J_{l-1} = |J(U^{l-1}) - J(U^{l-2})|$ as follows:

$$\alpha_l = \beta \max(\overline{\alpha}, \min(\underline{\alpha}, -\log_{10}(\Delta J_{l-1}))), \tag{14}$$

where $\beta \gg 0$ and $\overline{\alpha} > \underline{\alpha} \gg 0$ shape and constrain the step.

B. Control Design

As indicated, the performance part of the criterion minimized over prediction horizon $P \in \mathbb{N}^+$ is expressed as

$$J_p = \sum_{i=k+1}^{k+P} -x_{1,i}. \tag{15}$$

TABLE II
CASE STUDY PARAMETERS

Parameter	Value	Parameter	Value
$\overline{R}$ (m)	3.5	ω_2 (−)	100
Δ_r (m)	0.5	ω_3 (−)	2
ω_1 (−)	7		

The satisfaction of the safety requirements can be accomplished in several ways. The first option is to track the central line given by $\{x_{\text{cent},k}, y_{\text{cent},k}\}$, which, however, disables speed optimization. Rather than that, keeping the x- and y-position within admissible limits is more advantageous. To handle this, a new state x_5 (m) representing the total driven distance is introduced, and the model (1) is extended as follows:

$$\begin{aligned} x_{\{1,\ldots,4\},k+1} &\mathrel{\widehat{=}} (1), \\ x_{5,k+1} &= x_{5,k} + p_5 x_{1,k}. \end{aligned} \tag{16}$$

As in [28], [29], feasibility issues are eliminated introducing relaxed safety part of the criterion formulated as follows:

$$J_s = \sum_{i=k+1}^{k+P} L(x_{3,i}, x_{4,i}, \mathbf{C}_X(x_{5,i}), \mathbf{C}_Y(x_{5,i})), \tag{17}$$

where

$$L = \begin{cases} 0 & r_i < \overline{R}, \\ |r_i - \overline{R}| & \overline{R} \leq r_i < \overline{R} + \Delta_r, \\ \omega_3 (r_i - \overline{R})^2 & \overline{R} + \Delta_r \leq r_i. \end{cases} \tag{18}$$

Here,

$$r_i = \sqrt{(x_{3,i} - \mathbf{C}_X(x_{5,i}))^2 + (x_{4,i} - \mathbf{C}_Y(x_{5,i}))^2} \tag{19}$$

represents the distance of the car from the central line $[\mathbf{C}_X, \mathbf{C}_Y]$, $\overline{R} = W/2$ is the half-width of the track, Δ_r is the considered tolerance, and ω_3 is a weighting parameter.

Having specified a set of discrete points $\{x_{\text{cent}}, y_{\text{cent}}\}$ lying on the central line and the corresponding driven distances $\{d_{\text{cent}}\}$ and exploiting spline interpolation techniques [30], functions $\mathbf{C}_X(x_5)$ and $\mathbf{C}_Y(x_5)$ can be obtained as $\mathbf{C}_X \approx x_{\text{cent}}(d_{\text{cent}})$, $\mathbf{C}_Y \approx y_{\text{cent}}(d_{\text{cent}})$, and then directly incorporated into the cost criterion (20).

To avoid simultaneous use of gas and brake, additional minimization term $D_k B_k$ is considered. The overall criterion for the predictive controller is then formulated as

$$\mathcal{J}_k = \omega_1 J_p + \omega_2 J_s + \sum_{i=k}^{P} D_i B_i. \tag{20}$$

J_p and J_s correspond to (15) and (17), respectively, and ω_1 and ω_2 are user-defined weights. The values of $\omega_{\{1,2,3\}}$, $\overline{R}$ and Δ_r are listed in Table II. Last of all, let us note that the solution is required to respect the hybrid dynamics (16) with $N_c = 3$ and $N_d = 3$ and satisfy constraints (3).

C. Adaptive Prediction Horizon

Prediction horizon is one of the key parameters specifying the tradeoff between computational complexity and optimality. The idea of adaptive prediction horizon comes in very naturally in case of car motion control—intuitively, the higher the velocity is, the longer horizon is needed to handle the car satisfactorily and respect the track constraints. In this paper, three adaptive prediction horizon approaches are considered.

1) Linear Adaptive Horizon ($\theta - P$ Approach): The prediction horizon is calculated using $[\theta x_1^-]$ being the nearest integer to a θ-multiple of x_1^-

$$P = \max(1, [\theta x_1^-]), \tag{21}$$

where $\theta > 0$ is a tuning parameter. Despite simple calculation, the choice of the parameter θ is tricky and depends on the current track—combination of long straight parts where the car velocity increases rapidly and short sharp curves demanding intensive braking requires higher θ, while presence of only low-curvature passages might also allow for lower θ. Absence of such information degrades the control performance. This shortcoming is eliminated by the more advanced alternatives for adaptive prediction horizon.

2) Nominal Logarithmic Adaptive Horizon (nom-$\log - P$ Approach): In this case, the horizon P is calculated as

$$P = \begin{cases} 1 & x_1^- \le v_1, \\ 1 + \left\lceil \log_{\overline{p}} \left(\frac{v_1}{x_1^-} \right) \right\rceil & v_1 < x_1^-, \end{cases} \tag{22}$$

where $\overline{p} = p_1 - p_2 \overline{B} < 1$ represents the velocity dynamics coefficient with maximum braking and minimum acceleration. Here, $\lceil \epsilon \rceil$ denotes the smallest integer not less than ϵ. Now, let us define the *nominal car dynamics* as dynamics (1) with $x_1 \le v_1$, i.e., $\alpha(x_1) = 1$, and let us specify the *preservation of nominal maneuverability* as the physical capability of the car to drive over a trajectory that is realizable by the nominal car dynamics. Then, the following statement can be made.

Theorem 1. *Consider a vehicle with dynamics (16) with $\overline{p} = p_1 - p_2 \overline{B} < 1$ and $\underline{D} = 0$. Let us assume that given initial conditions x^-, an optimal controller OC^∞ with prediction horizon $P = \infty$ with respect to criterion (20) and constraints (3) results in $\sup(r_k) \le \overline{R} + \Delta_r$. Then, given the same initial conditions x^-, an optimal controller $OC^\star$ with prediction horizon $P^\star$ calculated according to (22) preserves nominal maneuverability and also leads to $\sup(r_k) \le \overline{R} + \Delta_r$.*

Proof: The only difference between the nominal car dynamics and the dynamics of the real car is caused by the fact that $\alpha(x_{1,k}) < 1 \Leftrightarrow x_{1,k} > v_1$. Assuming an optimal controller, it can be expected that given enough information about the upcoming trajectory (represented by infinite prediction horizon $P = \infty$), the controller decreases the velocity $x_{1,k}$, such that $\alpha(x_{1,k}) = 1$ when necessary. Here, it should be noted that infinite prediction horizon, in fact, collapses to a horizon of such a finite length that the whole track is covered. A controller with shorter P (not covering the whole track) is able to ensure such a decrease only in case that P is large enough to bring x_1 from $x_{1,0} = x_1^-$ to $x_{1,P} < v_1$ with $D_k \equiv \underline{D}$, $B_k \equiv \overline{B}$. Directly substituting $D_k \equiv \underline{D}$, $B_k \equiv \overline{B}$ and $P^\star$ calculated according to (22) into the dynamics of the car (1), the nonlinear velocity dynamics turns into a linear one,

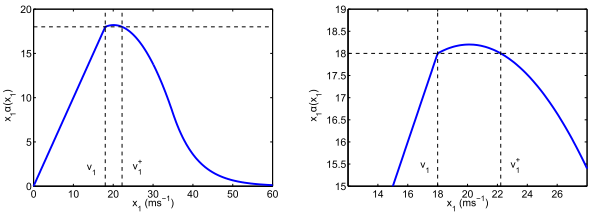

Fig. 3. $p_M(x_1)$ and supernominal maneuverability range $\langle 0, v_1^+ \rangle$, detail.

$$\overline{x}_{1,k+1} = \overline{p} \overline{x}_{1,k}, \tag{23}$$

with $\overline{p} = p_1 - p_2 \overline{B} < 1$. Then, $\overline{x}_{1,P^\star} = \overline{p}^{P^\star} x_1^-$. Obviously, $x_1^- \le v_1$ yields $P^\star = 1$, $\overline{x}_{1,P^\star - 1} \le v_1$, and $\alpha(\overline{x}_{1,P^\star - 1}) = 1$. For $x_1^- > v_1$, the substitution leads to

$$\overline{x}_{1,P^\star} = \left(\overline{p}^{\left(1 + \left\lceil \log_{\overline{p}} \left(\frac{v_1}{x_1^-} \right) \right\rceil \right)} \right) x_1^- \le \overline{p} \, \frac{v_1}{x_1^-} x_1^- < v_1. \tag{24}$$

Furthermore,

$$\overline{x}_{1,P^\star - 1} \le \frac{v_1}{x_1^-} x_1^- \le v_1 \Leftrightarrow \alpha(\overline{x}_{1,P^\star - 1}) = 1, \tag{25}$$

which means that $x_{2,P^\star}$ evolves from $x_{2,P^\star - 1}$ according to the nominal car dynamics. Therefore, $P^\star$ calculated according to (22) provides enough information to preserve nominal maneuverability. □

3) Supernominal Logarithmic Adaptive Horizon (S-nom-$\log - P$ Approach): Inspecting the dynamics (1), it can be seen that the function $\tan(S)$ is multiplied not only by $\alpha(x_1)$ but by the *maneuverability product* $p_M(x_1) = x_1 \alpha(x_1)$ that specifies the resulting efficiency of the steering S. Looking at Fig. 3, it is obvious that the value of p_M can be even higher than $v_1 \alpha(v_1)$. Let us call such values *supernominal values of* p_M, and let us define the *supernominal maneuverability range* as

$$\langle 0, v_1^+ \rangle = \{ x_1 |\, p_M(x_1) \ge v_1 \alpha(v_1) \ \vee \ \alpha(x_1) = 1 \}.$$

The value of v_1^+ can be found as the higher solution of the equation $p_M(x_1) = v_1 \alpha(v_1)$. Let us also specify the *preservation of supernominal maneuverability* as the capability of achieving that $x_{2,P-1}$ evolves to $x_{2,P}$ with either supernominal p_M or nominal $\alpha = 1$.

Now, let us have P calculated based on x_1^- as follows:

$$P = \begin{cases} 1 & x_1^- \le v_1^+, \\ 1 + \left\lceil \log_{\overline{p}} \left(\frac{v_1^+}{x_1^-} \right) \right\rceil & v_1^+ < x_1^-, \end{cases} \tag{26}$$

and formulate the following statement.

Theorem 2. *Consider a vehicle with dynamics (16) with $\overline{p} = p_1 - p_2 \overline{B} < 1$ and $\underline{D} = 0$. Let us assume that given initial conditions x^-, an optimal controller OC^∞ with prediction horizon $P = \infty$ with respect to criterion (20) and constraints (3) results in $\sup(r_k) \le \overline{R} + \Delta_r$. Then, given the same initial conditions x^-, an optimal controller*

$OC^{s\star}$ with prediction horizon $P^{s\star}$ calculated according to (26) preserves supernominal maneuverability and also leads to $\sup(r_k) \leq \overline{R} + \Delta_r$. *Furthermore, the following holds for the average value of prediction horizon exploited by controllers $OC^{s\star}$ and $OC^{\star}$*:

$$\text{mean}(P^{s\star}|OC^{s\star}) \leq \text{mean}(P^{\star}|OC^{\star}).$$

Proof: The first part of the proof is similar to the previous case and consists in direct substitution of $D_k \equiv \underline{D}$, $B_k \equiv \overline{B}$, and $P^{s\star}$ calculated by (26) into (1). Having accomplished this, it can be shown that with $x_1^- > v_1^+$ and $P^{s\star}$,

$$\overline{x}_{1,P^{s\star}-1} = \left(\overline{p}^{\left\lceil \log_{\overline{p}}\left(\frac{v_1^+}{x_1^-}\right)\right\rceil}\right) x_1^- \leq \frac{v_1^+}{x_1^-} x_1^- \leq v_1^+, \qquad (27)$$

i.e., the velocity can be decreased from $x_1^- > v_1^+$ such that the supernominal maneuverability range is reached at $k = P^{s\star} - 1$ and $x_{2,P^{s\star}}$ can evolve from $x_{2,P^{s\star}-1}$ with supernominal p_M.

The second part of the proof comes from the comparison of expressions (22) and (26)—since $v_1 \leq v_1^+$, $P^{s\star}(x_1^-)$ calculated according to (26) is not higher than $P^{\star}(x_1^-)$ calculated by (22) for any value of x_1^-. □

Remark. Considering r as an additional system output and $\mathbb{R}_{\text{adm}} = \langle 0, \overline{R} + \Delta_r \rangle$ as admissible set for r, a controller can be found stabilizable if it ensures that $r_k \in \mathbb{R}_{\text{adm}}$ $\forall k \geq 0$ iff $r_0 \in \mathbb{R}_{\text{adm}}$ or $\lim_{k\to\infty} r_k = r_a \in \mathbb{R}_{\text{adm}}$ iff $r_0 \notin \mathbb{R}_{\text{adm}}$. Then, it can be deduced that starting from initial conditions $r_0 \in \mathbb{R}_{\text{adm}}$, a suitably tuned optimal controller with the proposed adaptive prediction horizon is able to keep r within the admissible bounds given that this is achievable by the nominal car, i.e. $\mathbb{R}_{\text{adm}} = \langle 0, \overline{R} + \Delta_r \rangle$ is forward invariant with the proposed predictive controller and the adaptive predictive horizon. This covers the first part of the stability requirements. Their second part is covered by incorporating the track violation into the criterion (20). Since the controller makes control moves in the direction of negative gradient of the cost function, choosing suitable weights makes the nonzero safety part of the criterion decrease gradually from time $k-1$ to k, i.e., $J_{s,k} \leq J_{s,k-1}$. Therefore, if $r_0 \notin \mathbb{R}_{\text{adm}}$, the controller produces a series of control moves u_k such that $\lim_{k\to\infty} r_k = r_a \in \mathbb{R}_{\text{adm}}$, which covers the second part of the stability requirements. As such, Theorems I and II and their proofs guarantee the recursive feasibility when using the nom-log-P and S-nom-log-P prediction horizons.

IV. Results

Several numerical experiments were performed on different tracks to examine the performance of all presented alternatives. Their results are presented in this section.

A. APS-MPC Versus HaSH-NPC Comparison

At first, the hybrid predictive control algorithms were tested on Track 1 with the nominal prediction horizon $P = 25$ samples. Fig. 4 shows the behavior of the car on the track.

Looking at Fig. 4, it seems that both algorithms respect the safety constraints satisfactorily. However, more details are

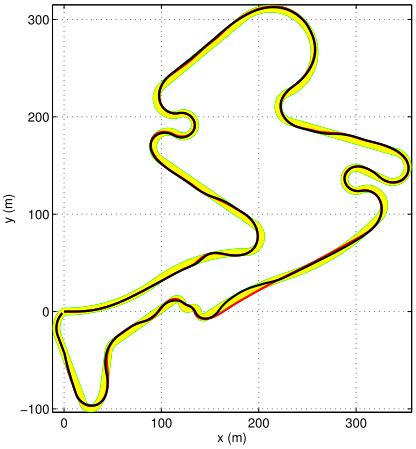

Fig. 4. Track 1. Black line: HaSH-NPC. Red line: APS-MPC.

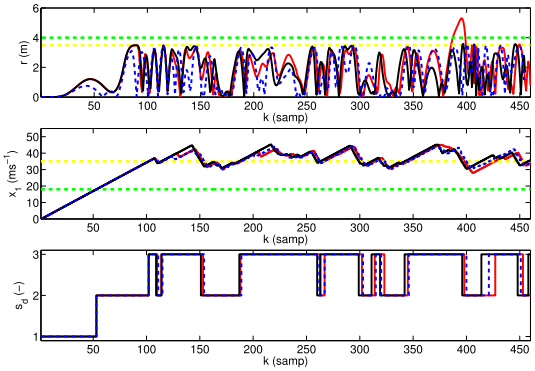

Fig. 5. Track 1—r, x_1, and s_d. Black line: HaSH-NPC with $P = 25$. Red line: APS-NPC with $P = 25$. Blue dashed line: APS-NPC with $P = 30$.

provided by the topmost subfigure of Fig. 5, where the distance of the car from the central line is shown. The yellow line $r = \overline{R}$ indicates the inner zero-penalized part of the track, while the green line $r = \overline{R} + \Delta_r$ indicates the transition between the linear penalization and quadratic penalization.

It can be seen that while the HaSH-NPC algorithm (represented by black solid line) almost never allows the car to leave the inner zero-penalized part of the track $r < \overline{R}$ and very safely satisfies the condition $r < \overline{R} + \Delta_r$, APS-MPC (represented by red solid line) working with the approximated description of the optimization problem happens to violate even the additional tolerance on the distance from the central line. Such a significant track violation can eventually bring the car to a point at which it is not able to return back to track and continue racing any more. This negative effect can be eliminated considering APS-MPC with increased prediction horizon $P = 30$ (represented by the blue dashed line). Although the use of longer prediction horizon complies with the expectations and helps to keep the car on the track, increase of the computational time can also be expected.

Fig. 5 also shows the x_1 profiles for the three above-mentioned variants (middle subfigure). Since the velocity determines the current system subdynamics, the velocities v_1 and v_2 of expression (2) are indicated by the green and yellow

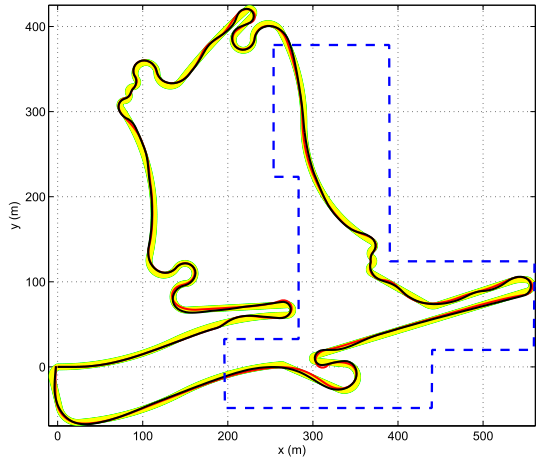

Fig. 6. Track 2. Black line: HaSH-NPC. Red line: APS-MPC.

dashed lines, and also the dynamics-switcher profiles $\mathfrak{s}_c$ are provided (see the bottommost subfigure).

It is obvious that the assumption on *a priori* known sequence of the system subdynamics cannot hold in this case, and therefore, the approaches relying on it could not be used. Given that the velocity (and thus the dynamics switcher) profiles are quite similar for the three depicted alternatives, while the track satisfaction differs significantly for HaSH-NPC versus APS-MPC with $P = 25$, it can be concluded that with equal prediction horizon, the HaSH-NPC handles the switching dynamics in a more appropriate way.

To obtain a more reliable comparison, another set of numerical experiments with the longer and more complicated Track 2 was performed. Track 2 and the behavior of the car with the two hybrid predictive control algorithms are presented in Fig. 6. Also in this case, nominal $P = 25$ was used.

In this case, the difference between the performances of the two algorithms is more significant. While the HaSH-NPC handles the complex track as well as the simpler one, certain problems in keeping the car on the track can be observed in case of APS-MPC. This is demonstrated by Fig. 7, where several details of the track are provided. Especially when driving at limit speed and cornering, the APS-MPC with nominal prediction horizon sometimes happens to get out of the track. Fig. 8 shows the velocity profiles and distances from the central line in one such situation in more detail. In the first subfigure, black and red lines represent the distance r reached by HaSH-NPC and APS-MPC at particular distance driven from the start d. Dashed lines mark $r = \overline{R}$ (yellow line) and $r = \overline{R} + \Delta_r$ (green line). In the second subfigure, black and red lines show velocity reached by HaSH-NPC/APS-MPC, and green and yellow dashed lines mark v_1 and v_2, respectively.

From Fig. 8, it can be seen that due to the approximation, APS-MPC does not decrease the speed sufficiently when cornering. This is not the case of HaSH-NPC, which acts appropriately and successfully satisfies the track tolerance. By increasing prediction horizon to $P = 45$ samples in case

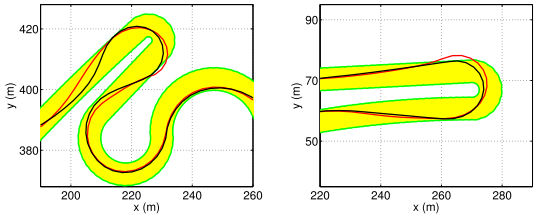

Fig. 7. Track 2. Black line: HaSH-NPC. Red line: APS-MPC.

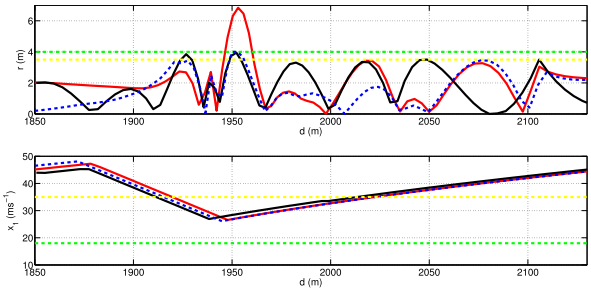

Fig. 8. Track 2—r and x_1. Black line: HaSH-NPC with $P = 25$. Red line: APS-NPC with $P = 25$. Blue dashed line: APS-NPC with $P = 30$.

TABLE III
COMPARISON OF HYBRID PREDICTIVE CONTROL ALGORITHMS

		$\overline{x}_1$ (ms^{-1})	$\overline{TV}$ (m)	$d_{\Delta_r V}$ (m)
T1	HaSH-NPC	33.4	0.1	0
	APS-MPC$_{25}$	33.0	1.8	34
	APS-MPC$_{30}$	33.2	0.1	0
T2	HaSH-NPC	34.7	0.4	0
	APS-MPC$_{25}$	34.2	3.3	42
	APS-MPC$_{35}$	34.4	0.9	10
	APS-MPC$_{45}$	34.6	0.5	0

of APS-MPC (blue dashed line), even the algorithm working with approximation achieves satisfactory performance.

The results of all experiments with constant prediction horizon are summarized in Table III, where T1 and T2 indicate the particular track. Several numerical evaluators were chosen as follows to provide a comprehensive comparison. As the first evaluator, the average velocity $\overline{x}_1$ was considered. The second evaluator $\overline{TV} = \max\{TV_k\}$ corresponds to the maximal track violation $TV_k = \max(0, r_k - \overline{R})$, where r_k is defined by (19). Let us remind the tolerance for the track violation $\Delta_r = 0.5$ m. The last evaluator $d_{\Delta_r V}$ represents the distance driven by the car when violating even the track tolerance ($r_k > \Delta_r$). For HaSH-NPC, prediction horizon $P = 25$ samples was considered, while in case of the other algorithm, the prediction horizon is indicated by the subscript (e.g., APS-MPC$_{35}$ means APS-MPC with horizon $P = 35$ samples).

Inspecting Table III, it can be seen that all algorithms achieve similar average velocities on particular track with slight superiority of HaSH-NPC results and (as expected), increase of prediction horizon results in increase of $\overline{x}_1$ in case of APS-NPC algorithm. However, a big difference can be noticed comparing $\overline{TV}$ and $d_{\Delta_r V}$. On Track 1, the HaSH-NPC algorithm never violates the track tolerance (see $\overline{TV} = 0.1 <$

TABLE IV
COMPARISON OF ADAPTIVE PREDICTIVE HORIZON APPROACHES

		$\overline{x}_1$ (ms^{-1})	SP	$\overline{P}$ (−)	max(P) (−)	E (−)
θ-P	$\theta = 0.2$	30.7	✗	6.6	11	–
	$\theta = 0.4$	32.7	✗	13.6	20	–
	$\theta = 0.6$	34.6	✓	20.6	31	1.68
	$\theta = 0.8$	34.7	✓	27.3	38	1.27
	$\theta = 1.0$	34.9	✓	33.3	49	1.05
nom-log-P		34.9	✓	22.4	36	1.56
S-nom-log-P		34.7	✓	17.8	29	1.95
c-HaSH-NPC		34.7	✓	25	25	1.39

Δ_r and $d_{\Delta_r V} = 0$). On the other hand, APS-MPC with $P = 25$ drives 34 m violating the track by more than 0.5 m with a maximal violation of 1.8 m. To make APS-MPC achieve acceptable performance the prediction horizon needs to be increased to $P = 30$ samples.

The situation is even more significant in case of the more complicated Track 2. Although $\overline{TV}$ of HaSH-NPC rises to 0.4 m, it stays within the defined track tolerance Δ_r with P as low as 25 samples. The APS-MPC algorithm, however, is not able to achieve desirable track-satisfaction performance even with $P = 35$ for which it still violates the track by up to 0.9 m. The satisfaction of the track tolerance is achieved with as long predictions as $P = 45$ samples. The poorer behavior of the APS-MPC algorithm is caused mainly by the approximation of the hybrid dynamics/cost criterion. MPC is a model-based controller, and therefore, neglecting/approximating the system dynamics in a significant way comes hand in hand with performance degradation. On the other hand, increasing P can remedy these negative effects, since more time is provided to take the corrective action. This explains why APS-MPC is outperformed by HaSH-NPC with equal prediction horizons and why also APS-MPC can satisfy the safety requirements with increased P.

B. Adaptive Prediction Horizon Approaches Comparison

To inspect the performance of the adaptive prediction horizon approaches, only HaSH-NPC algorithm was evaluated. Track 2 was considered because of its more complicated shape and a need for more aggressive car handling and maneuvering.

To obtain an illustrative and reliable comparison of different approaches, several evaluators were inspected. The first one was the average achieved velocity $\overline{x}_1$ representing the performance part of the criterion, while the safety part SP of the criterion, $\max(r_k - \overline{R}) \leq \Delta_r$, was evaluated binarily (✓—passed and ✗—failed). As the computational complexity and efficiency markers, average prediction horizon $\overline{P}$ considered by particular controller and "efficiency ratio" $E = \overline{x}_1/\overline{P}$ were evaluated as well. The results are summarized in Table IV. For the sake of completeness, results of algorithm with constant prediction horizon denoted as c-HaSH-NPC are provided as well.

At first, θ-P approach was tested with $\theta \in \{0.2, 0.4, 0.6, 0.8, 1\}$. It can be seen that while all θ-P variants with $\theta \geq 0.6$ passed the safety requirements, those with $\theta \geq 0.8$ might not be regarded as competitive due to their excessive computational complexity demonstrated by $\overline{P} \geq 27.3$. This comes hand-in-hand with a decrease of the efficiency ratio E, which degrades from $E = 1.68$ (for $\theta = 0.6$) to as low as $E = 1.05$ (for $\theta = 1.0$). The efficiency ratio for variants that did not pass the safety requirements was not evaluated.

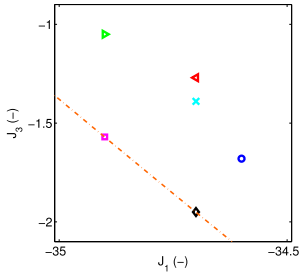

Fig. 9. Pareto optimality (○—0.6-P, ◁—0.8-P, ▷—1.0-P, ×—c-HaSH-NPC, □—nom-log-P, ◇—S-nom-log-P, and ·–·—Pareto frontier).

Unlike the θ-P variants, nom-log-P and S-nom-log-P approaches provide both safety constraints satisfaction and attractive performance with high computational efficiency. While nom-log-P approach achieves the highest $\overline{x}_1$, the S-nom-log-P approach is clearly the most computationally efficient with $E = 1.95$.

Considering multiple evaluative criteria J_i, $i \in \{1, \ldots, n_i\}$ and a set of solutions $\mathbf{X}$, solution $\tilde{x} \in \mathbf{X}$ is said to *dominate* solution $\check{x} \in \mathbf{X}$ iff $J_i(\tilde{x}) \leq J_i(\check{x})$ for all i and at least for one $j \in \{1, \ldots, n_i\}$, $J_j(\tilde{x}) < J_j(\check{x})$. A solution $\hat{x} \in \mathbf{X}$ is said to be *Pareto optimal* iff it is not dominated by any other solution $\check{x} \in \mathbf{X}$. Further details on Pareto optimality might be found in [31]–[33] and references therein.

With three evaluative criteria $J_1 = -\overline{x}_1$, $J_2 = \overline{P}$ and $J_3 = -E$, it can be shown that out of all alternatives that passed the safety requirements, only nom-log-P and S-nom-log-P approaches are not dominated by any other solution and thus are *Pareto optimal*. This is demonstrated in Fig. 9—the *Pareto frontier* [31] comprising the Pareto optimal solutions consists exclusively of logarithm-based (nom-log-P and S-nom-log-P) approaches.

Fig. 10 shows the tradeoff between the efficiency E and the safety-requirements satisfaction by depicting P_k (horizon at time k) as a function of $x_{1,k}$. The S-nom-log-P approach splits the approaches into two groups—those lying completely above the S-nom-log-P-profile are safety-acceptable yet efficiently suboptimal while those that "undercrawl" it significantly are in turn more efficient but might be safety-unacceptable.

The overview is completed by a comparison with a commercially available MINLP solver provided in Table V. In this role, *ga* function implementing genetic algorithm being part of MATLAB Global Optimization Toolbox was employed with three different settings denoted as ga_1, ga_2, and ga_3. CTR (−) expressing the ratio between the average computational time of *ga* and HaSH-NPC was evaluated as well.

This comparison shows the main advantage of the HaSH-NPC algorithm against the MINLP solvers—elimination of high computational complexity. While with the computation-

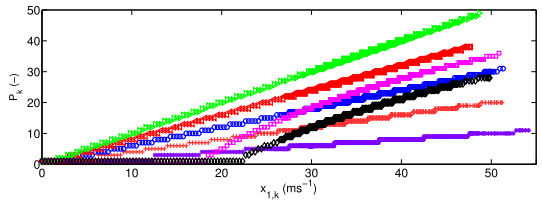

Fig. 10. P_k as a function of $x_{1,k}$ (∗—0.2-P, +—0.4-P, ○—0.6-P, ◁—0.8-P, ▷—1.0-P, □—nom-log-P, and ◇—S-nom-log-P).

TABLE V
MINLP SOLVER RESULTS

	nom-log-P			S-nom-log-P		
	ga_1	ga_2	ga_3	ga_1	ga_2	ga_3
$\overline{x}_1$ (ms^{-1})	33.0	33.8	34.4	32.9	33.5	34.3
$\overline{\text{TV}}$ (m)	0.3	0.2	0.2	0.3	0.2	0.2
CTR (−)	42.5	150.0	1063.8	41.7	152.1	1069.9

TABLE VI
p_1 SENSITIVITY ANALYSIS

p_1	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
0.99	38.1	0.2	37.6	0.2
0.999	**38.4**	**0.2**	**38.0**	**0.2**
0.99999	38.3	0.2	37.9	0.2
1.001	38.2	0.2	37.8	0.2
1.002	38.0	0.2	37.6	0.3
1.003	37.8	0.2	37.3	0.3
1.004	37.7	0.3	37.2	0.4

ally least demanding settings (which still consumes about 40× more time), the *ga* velocity performance is about 5% worse, the best achieved *ga* solution comparable with the HaSH-NPC one requires more than 1000× longer computations.

V. SENSITIVITY ANALYSIS

To evaluate the robustness of the proposed control algorithms, a detailed scenario-based sensitivity analysis was performed exploiting a subsection of the second track encircled in Fig. 6 by the blue dashed line. In each of the analyzed cases, *nom*-log-P and *S-nom*-log-P approaches were tested and $\overline{x}_1$ and $\overline{\text{TV}}$ were evaluated. Out of the model parameters, p_5 was excluded from the sensitivity analysis. The other "p" parameters (i.e., p_{1-4}) were perturbed separately, while the α-formula (i.e., a_{1-4} and $v_{1,2}$ parameters) was changed as a whole. Let us note that the MPC model parameters correspond to Table I unless otherwise stated, and for completeness, the unperturbed cases are presented in the tables in blue.

A. p_1 Sensitivity Analysis

Six values ranging from 0.99 to 1.004 were chosen, which can be interpreted as uphill/rough terrain driving, noninclined road driving, and downhill driving. The results are presented in Table VI.

From the $\overline{\text{TV}}$ results it can be seen that none of the tested values causes safety problems, which means that the algorithm is sufficiently robust against the p_1 mismatch.

TABLE VII
p_2 SENSITIVITY ANALYSIS

p_2	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
0.04	37.9	0.1	37.6	0.1
0.03	**38.4**	**0.2**	**38.0**	**0.2**
0.027	37.5	0.2	36.6	0.2
0.025	36.0	0.3	34.6	0.3
0.024	33.7	0.3	32.0	0.4
0.02	35.3	0.4	33.3	0.4
0.015	33.2	0.5	31.4	0.5
0.01	29.3	0.5	27.6	0.5
0.008	29.9	0.1	29.3	0.1
0.007	29.8	0.1	29.2	0.1
0.006	29.9	0.2	29.2	0.2
0.008	33.1	0.2	32.7	0.2
0.007	31.7	0.2	31.2	0.2
0.006	30.2	0.2	29.8	0.2

B. p_2 Sensitivity Analysis

Robustness against p_2 perturbation was tested on a set of ten scenarios with p_2 ranging from 0.04 to 0.006 to cover both the situations where the braking effectiveness is underestimated and those where the braking effect decreases (which can happen due to rain or snow) and MPC overestimates it. The results are listed in Table VII.

Inspecting the results obtained for $p_2 = 0.04$ to 0.024 (the first subtable), it can be seen that both variants are successful without any adaptations. Decreasing p_2 to 0.02, S-nom-log-P violated Δ_r, and therefore, the following workaround was proposed. Calculation of prediction horizon was performed considering "worst case guess" $p_{2,\text{wg}} = 0.01$, while for the MPC model itself, the original $p_2 = 0.03$ was used. Basically, only the prediction horizon was increased, while the dynamics remained the same. This was successfully tested for $p_2 = 0.02$ to 0.01 (see the second subtable). For $p_2 = 0.008$, Δ_r was again violated, and another workaround consisting in use of "worst case guess" for both the prediction horizon calculation and the MPC model was implemented with $p_{2,\text{wg}} = 0.005$. The usefulness of this adaptation is demonstrated by the results presented in the third subtable. Last of all, a p_2 estimator was designed according to (1) using x_1, D, and B measurements, and as the parameter for the optimizer, moving average calculated from p_2 estimates over the last ten samples was used. These results presented in the last subtable show that while already the original "no-estimator" algorithm had satisfied the safety requirements, the optimality in the sense of x_1 improved with the estimator.

C. p_3 Sensitivity Analysis

Regarding the p_3 parameter, eight values ranging from 0.2 to 0.7 corresponding to 0–100 kph acceleration times of 4–15 s were used. The results can be found in Table VIII.

The influence of the p_3 parameter perturbation on the performance is in some sense proportional to the p_3 perturbation—p_3 decrease/increase results in decrease/increase of both $\overline{x}_1$ and $\overline{\text{TV}}$, nevertheless,

TABLE VIII
p_3 SENSITIVITY ANALYSIS

p_3	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
0.20	36.0	0.1	35.7	0.1
0.25	36.9	0.1	36.7	0.1
0.30	37.6	0.1	37.2	0.2
0.325	38.2	0.2	37.7	0.2
0.35	**38.4**	**0.2**	**38.0**	**0.2**
0.40	38.6	0.2	38.3	0.3
0.50	39.3	0.3	38.7	0.3
0.60	39.7	0.3	39.2	0.4
0.70	40.0	0.3	39.6	0.4

TABLE IX
p_4 SENSITIVITY ANALYSIS

$p_4 \times 10^3$	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
33.33	36.5	0.4	35.8/35.6	0.6/0.4
34.48	37.5	0.4	37.0	0.5
35.71	38.2	0.3	37.7	0.4
36.36	**38.4**	**0.2**	**38.0**	**0.2**
37.04	38.3	0.2	37.9	0.2
38.46	38.1	0.1	37.7	0.1
40.00	37.9	0.1	37.3	0.1
41.67	37.7	0.0	37.1	0.1

the satisfaction of safety requirements remains unharmed for the whole inspected range.

D. p_4 Sensitivity Analysis

The p_4-perturbation robustness of the control algorithm was verified on a series of seven numerical experiments, where p_4 varied from 33.33×10^{-3} to 41.67×10^{-3}. Such values can be interpreted as wheelbase ranging from 2.4 to 3 m, which covers the vast majority of the race cars. The obtained results are shown in Table IX.

While the *nom*-log-P approach handles all evaluated p_4 values without violating the track constraints, the *S-nom*-log-P approach encounters difficulties with the lowest p_4 values representing a car with 3-m wheelbase. The difference between the two approaches is in their prediction horizon—although both algorithms optimize considering the same dynamics, slightly longer prediction horizon of *nom*-log-P approach provides it with enough time to take the corrective action. This weakness of the *S-nom*-log-P approach can be remedied by increasing the safety penalty ω_2 from 200 to 500. The results of such a configuration are presented in Table IX in the corresponding row after the slash mark. As can be seen, the unacceptable track violation was successfully eliminated.

Inspecting $\overline{x}_1$, it can be concluded that both understeering and oversteering lead to $\overline{x}_1$ decrease.

E. α Sensitivity Analysis

The α expression was perturbed as a whole to preserve monotonicity of the coefficient. These perturbations mean that multiple parameters were changed at time. Therefore, the perturbed coefficients are plotted instead of exact numerical perturbations of the particular parameters.

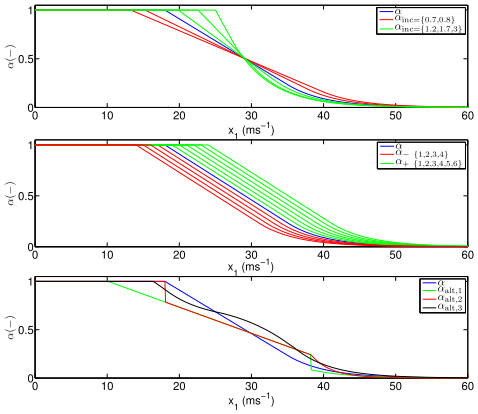

Fig. 11. Perturbations of α.

TABLE X
α SENSITIVITY ANALYSIS

α	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
$\alpha_{\text{inc}=0.7}$	31.9	0.4	30.0	0.4
$\alpha_{\text{inc}=0.8}$	34.1	0.3	32.1	0.3
$\alpha_{\text{inc}=1.0}$	**38.4**	**0.2**	**38.0**	**0.2**
$\alpha_{\text{inc}=1.2}$	35.2	0.2	33.5	0.2
$\alpha_{\text{inc}=1.7}$	34.4	0.3	33.1	0.3
$\alpha_{\text{inc}=3.0}$	34.1	0.3	32.4	0.4
α_{+6}	32.2	0.3	31.3	0.4
α_{+5}	32.3	0.2	31.3	0.3
α_{+4}	32.9	0.2	32.0	0.3
α_{+3}	33.3	0.1	32.3	0.2
α_{+2}	33.5	0.0	33.4	0.2
α_{+1}	34.5	0.1	34.2	0.1
α_0	**38.4**	**0.2**	**38.0**	**0.2**
α_{-1}	35.8	0.3	34.9	0.3
α_{-2}	31.5	0.4	30.5	0.4
α_{-3}	30.7	0.4	29.5	0.5
α_{-4}	30.1	0.5	28.5	0.5
$\alpha_{\text{alt},1}$	34.3	0.4	32.6	0.4
$\alpha_{\text{alt},2}$	35.2	0.3	34.2	0.4
$\alpha_{\text{alt},3}$	36.3	0.2	35.4	0.3

At first, the inclination of α varied from 0.7 up to 3. These perturbations are in Fig. 11 denoted as $\alpha_{\text{inc}=i}$ (i stands for the inclination). Next, α was "shifted" by -4 up to $+6\,\text{ms}^{-1}$ as shown in Fig. 11, where the corresponding profiles are denoted as α_s (s is the velocity shift). Following the results from Section V-D, $\omega_2 = 500$ was used for *S-nom*-log-P with $\alpha_{\{-2,-3,-4\}}$. Three cases were added (see $\alpha_{\text{alt},1}$, $\alpha_{\text{alt},2}$ and $\alpha_{\text{alt},3}$ in Fig. 11). The results can be found in Table X.

Table X demonstrates that although inaccurate α expression degrades $\overline{x}_1$, no significant interventions are needed for the algorithms to keep the car on the track with $\overline{\text{TV}} \leq \Delta_r$. It should be noted that this also holds for the inspected cases where not only the parameters of the α-expression were perturbed, but even completely different mathematical functions (higher powers of x_1, their reciprocals and logarithms) were used, which is represented by $\alpha_{\text{alt},1}$, $\alpha_{\text{alt},2}$ and $\alpha_{\text{alt},3}$.

TABLE XI
PERFORMANCE ENHANCEMENT WITH PARAMETER ESTIMATORS

	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
no estimator	30.3	1.2	29.2	1.5
estimator	33.1	0.2	32.8	0.2
ideal case	33.3	0.2	33.0	0.2

F. Performance Enhancement

It can be expected that the performance might further improve with an estimator providing regular parameters corrections. To verify this, $p_{2,\text{real}} = 0.01$ and $\alpha_{\text{real}} = \alpha_{\text{alt},1}$ were used in the real system dynamics. The other parameters were kept at their original values, since either their influence was insignificant or they are not expected to be misestimated. The x_1 and x_2 measurements were corrupted by white noises with the variances of $\sigma_1 = 0.5$ and $\sigma_2 = 35 \times 10^{-3}$. Estimates of p_2 were obtained as described in Section V-B. Regarding α coefficient, its current value was regularly estimated as well, and the obtained $\{x_1, \alpha(x_1)\}$ pairs were used for recursive approximation of $\alpha(x_1)$ expression. These estimates were used for both *nom*-log-P and *S-nom*-log-P algorithms. Table XI presents the results achieved without and with parameter estimator and those obtained with perfect knowledge of the system parameters. The results demonstrate that even though the original MPC parameters might be inaccurate, their continuous estimation can change the performance from *unsatisfactory* to almost equivalent to the *ideal case*.

VI. CONCLUSION

In this paper, the HaSH-NPC being a new hybrid nonlinear MPC algorithm for vehicular control was designed. Unlike the commonly used solution approximating the optimization problem (APS-MPC), the HaSH-NPC handles the hybridity in the system dynamics/cost criterion directly exploiting an auxiliary variable—the *Hamiltonian-switcher*. The HaSH-NPC algorithm was verified on an example of a race car with hybrid dynamics considering hybrid cost criterion. The results show a very attractive performance of the HaSH-NPC, which even with short prediction horizon outperforms the APS-MPC algorithm.

The second part of this paper focused on adaptive prediction horizons. Linear and logarithm-based prediction horizon approaches were proposed. Their results show that while also linear prediction horizons improve the computational burden when compared with the constant prediction horizon, they might not be able to provide acceptable safety-requirements satisfaction. This is overcome by the logarithm-based approaches, which are also shown to be *Pareto optimal* with respect to multiple evaluative criteria. Additional comparison with a commercially available MINLP solver provided the same prediction horizons demonstrates that the HaSH-NPC requires only a fraction of MINLP solver computational time with comparable performance.

In the last part, the results of a detailed sensitivity analysis were presented demonstrating the robustness of the HaSH-NPC with respect to various system parameters perturbations. Several performance enhancements that can further improve the robustness and the overall functionality of the algorithm were also proposed.

The results encourage practical use of the algorithms that provide a "recipe" for computationally effective nonlinear MPC for the automotive area.

REFERENCES

[1] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.

[2] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 163–168.

[3] J. Funke *et al.*, "Up to the limits: Autonomous Audi TTS," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2012, pp. 541–547.

[4] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1258–1269, Jul. 2013.

[5] H. Martínez-Barberá and D. Herrero-Pérez, "Multilayer distributed intelligent control of an autonomous car," *Transp. Res. C, Emerg. Technol.*, vol. 39, pp. 94–112, Feb. 2014.

[6] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *Proc. 46th IEEE Conf. Decision Control*, Dec. 2007, pp. 2980–2985.

[7] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 862–875, 2008.

[8] T. Keviczky, P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat, "Predictive control approach to autonomous vehicle steering," in *Proc. IEEE Amer. Control Conf.*, Jun. 2006, pp. 1–6.

[9] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *Proc. IEEE Amer. Control Conf.*, Jun. 2008, pp. 3719–3724.

[10] M. H. L. Hounjet and J. J. Meijer, "Evaluation of elastomechanical and aerodynamic data transfer methods for non-planar configurations in computational aeroelastic analysis," Nat. Aerosp. Lab., Amsterdam, The Netherlands, Tech. Rep. NLR-TP-95690 U, 1995.

[11] T. Geyer, M. Larsson, and M. Morari, "Hybrid emergency voltage control in power systems," in *Proc. Eur. Control Conf.*, 2003, pp. 1893–1898.

[12] Q. A. Acton, *Issues in Aerospace and Defense Research and Application: 2013 Edition*. Atlanta, GA, USA: ScholarlyEditions, 2012. [Online]. Available: https://books.google.cz/books?id=B4O0hukUBt8C

[13] L. Zhang, S. Wang, H. R. Karimi, and A. Jasra, "Robust finite-time control of switched linear systems and application to a class of servomechanism systems," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 5, pp. 2476–2485, Oct. 2015.

[14] O. Stursberg and S. Engell, "Optimal control of switched continuous systems using mixed-integer programming," in *Proc. 15th IFAC World Congr. Autom. Control*, Barcelona, Spain, 2002, pp. 433–438.

[15] M. Morari, M. Baotic, and F. Borrelli, "Hybrid systems modeling and control," *Eur. J. Control*, vol. 9, nos. 2–3, pp. 177–189, 2003.

[16] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. Eur. Control Conf.*, vol. 1. 2001, pp. 2603–2608.

[17] H. W. Lenstra, Jr., "Integer programming with a fixed number of variables," *Math. Oper. Res.*, vol. 8, no. 4, pp. 538–548, 1983.

[18] L. A. Wolsey, "Mixed integer programming," in *Wiley Encyclopedia of Computer Science and Engineering*. Hoboken, NJ, USA: Wiley, 2008.

[19] X. Xu and P. J. Antsaklis, "Results and perspectives on computational methods for optimal control of switched systems," in *Hybrid Systems: Computation and Control*. New York, NY, USA: Springer, 2003, pp. 540–555.

[20] S. Xu, S. E. Li, K. Deng, S. Li, and B. Cheng, "A unified pseudospectral computational framework for optimal control of road vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1499–1510, Aug. 2015.
[21] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, "Robust tube-based decentralized nonlinear model predictive control of an autonomous tractor-trailer system," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 1, pp. 447–456, Feb. 2015.
[22] R. Pepy, A. Lambert, and H. Mounier, "Path planning using a dynamic vehicle model," in *Proc. 2nd IEEE Inf. Commun. Technol. (ICTTA)*, vol. 1. Apr. 2006, pp. 781–786.
[23] R. N. Jazar, *Vehicle Dynamics: Theory and Application*. New York, NY, USA: Springer, 2008.
[24] H. Pacejka, *Tire and Vehicle Dynamics*. Amsterdam, The Netherlands: Elsevier, 2005.
[25] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics*, vol. 400. Warrendale, PA, USA: SAE International, 1995.
[26] A. E. Bryson, *Applied Optimal Control: Optimization, Estimation, and Control*. New York, NY, USA: Taylor & Francis, 1975.
[27] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, vol. 40. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.
[28] C. R. He, H. Maurer, and G. Orosz, "Fuel consumption optimization of heavy-duty vehicles with grade, wind, and traffic information," *J. Comput. Nonlinear Dyn.*, vol. 11, no. 6, p. 061011, 2016.
[29] H. Maurer and S. Pickenhain, "Second-order sufficient conditions for control problems with mixed control-state constraints," *J. Optim. Theory Appl.*, vol. 86, no. 3, pp. 649–667, 1995.
[30] C. de Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer-Verlag, 1978.
[31] Y. Censor, "Pareto optimality in multiobjective problems," *Appl. Math. Optim.*, vol. 4, no. 1, pp. 41–59, 1977.
[32] P. M. Pardalos, A. Migdalas, and L. Pitsoulis, Eds., *Pareto Optimality, Game Theory and Equilibria*, vol. 17. New York, NY, USA: Springer, 2008.
[33] W. Stadler, Ed., *Multicriteria Optimization in Engineering and in the Sciences*, vol. 37. New York, NY, USA: Springer, 2013.

Matej Pčolka received the bachelor's degree in cybernetics and measurement and the master's degree (Hons.) in aircraft instrument and control systems from Czech Technical University in Prague, Prague, Czech Republic, in 2009 and 2011, respectively, where he is currently pursuing the Ph.D. degree.

He was a Visiting Scholar with Katholieke Universiteit Leuven, Leuven Belgium, in 2012, and Michigan Technological University, Houghton, MI, USA, in 2013. His current research interests include numerical optimization, constrained optimal control, and nonlinear model predictive control with application in the area of bioprocesses, building climate control, and autonomous vehicle control.

Eva Žáčeková received the bachelor's degree in cybernetics and measurement and the master's degree (with honors) in systems and control from Czech Technical University in Prague, Prague, Czech Republic, in 2009 and 2011, respectively, where she is currently pursuing the Ph.D. degree.

She was a Visiting Scholar with Katholieke Universiteit Leuven, Leuven, Belgium, in 2012, and University of California, Berkeley, CA, USA, in 2013. Her current research interests include system identification, control relevant identification, persistent excitation, and dual model predictive control with application in the area of building climate control and autonomous vehicle control.

Sergej Čelikovský (SM'03) received the M.Sc. degree in applied mathematics from the Optimal Control Department, Moscow State University, Moscow, Russia, in 1984, and the Ph.D. degree in technical cybernetics from the Czechoslovak Academy of Sciences, Prague, Czech Republic, in 1988.

He was a Visiting Researcher with the University of Twente, Enschede, The Netherlands, in 1996, the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong, in 1998, and CINVESTAV, Instituto Politécnico Nacional, Mexico City, Mexico, from 1998 to 2000. He is currently a Research Fellow and the Head of the Department of Control Theory, Institute of Information Theory and Automation, Czech Academy of Sciences, and a Full Professor with the Department of Control Engineering, Czech Technical University in Prague, Prague. He is the coauthor of one book and two book chapters, 57 papers in the journals reported in WoS, and over 100 papers in the international conference proceeding. His current research interests include nonlinear systems, exact feedback linearization, stabilization, chaotic systems, observers, and output feedback control, underactuated mechanical systems, and walking robots.

Michael Šebek (SM'91) received the M.Sc. (Hons.) in electrical engineering from Czech Technical University in Prague, Prague, Czech Republic, and the Ph.D. and D.Sc. degrees in control theory from the Czech Academy of Sciences, Prague.

He is currently the Head of the Control Engineering Department, Czech Technical University in Prague, and the Head of the Cyber-Physical Systems Department, Czech Institute for Informatics, Cybernetics, and Robotics in Prague. He is also the CEO of PolyX Ltd., and a Producer of the Polynomial Toolbox for MATLAB. He held several visiting positions abroad, including the University of Twente, Enschede, The Netherlands, and ETH Zurich, Zürich, Switzerland. He has published well over 200 research papers, developed several commercial software packages, and has led numerous research projects. His current research interests include linear systems and control, networked systems, and vehicle formations.

Dr. Šebek was a Council Member, International Federation of Automatic Control. In 2016, he received the Werner von Siemens Best Educator Prize. He was the General Chair of the 16th World Congress.

Chapter 6

Production-process Optimization

In this chapter, overall optimization of production processes is addressed; in this context, production-process optimization refers to joint optimization of both the initial conditions of a production process and its manipulated variables such that an extremum of a chosen evaluative criterion is attained while respecting given constraints imposed on the optimized variables, i.e. both the control variables and the process initial conditions are assumed to be bounded. As a thoughtful reader might have already discovered, this approach can be particularly useful for a broad class of applications in the process control area, in business or in the area of mathematical economics.

In the state-of-the-art literature, such general optimization philosophy is somewhat overlooked. It should be remarked that optimization of the manipulated variables has gradually become a common practice across various control engineering areas. In such case, however, the initial conditions are considered to be fixed and immutable, which might be a reasonable postulate for a wide variety of control tasks ranging from building climate control [Ma et al., 2012] to flight control [Alexis et al., 2012] and others, nevertheless, when following this paradigm in case of process control [Mears et al., 2017], this is equivalent to giving up a significant part of the production potential of the controlled process. A search for more suitable initial conditions is performed rather rarely [Karthikeyan et al., 1996], [Jeong et al., 2010], [Kim et al., 2012] and the available works rely on statistical evaluation of a quite large number of real-life experiments with various initial settings, which is both extremely time consuming and economically ineffective—if not directly inadmissible—and, moreover, the fact that also the input profile can be optimized is omitted in such cases.

Another effect characterizing real applications in the area of process control—and bio-process control in particular—is that the output measurements are very irregularly sampled

due to a complicated laboratory evaluation of the withdrawn samples. This collides with the widely adopted approach (i.e. combination of discretized system dynamics model and subsequent optimization of discretized input profiles) and the endeavour to fully and immediately exploit the information contained in measurements would apparently lead to decrease of the sampling period yielding steep growth of the computational demands. To forestall the curse of dimensionality, a constant sampling period is usually chosen as a compromise sacrificing the optimality in favor of computational bearability. A more appropriate alternative might be a suitable parameterization of the input profile working in continuous time with several noteworthy attempts on bioprocess yields maximization [Liu, 2009], [Gong et al., 2011], [Liu et al., 2012], [Liu et al., 2013], [Liu and Gong, 2014]. In these works, the input was allowed to be at most quadratic function of time and the parameters of the a priori chosen input expression in combination with switching times were optimized.

In [A.4], a novel algorithm for overall production process optimization[1] going significantly beyond the state of the art and standing for one of the main contributions of this thesis was proposed with the following highlights. First of all, a very general set of basis functions was considered meaning that the optimized input was admitted to be truly *whichever* (eventually piecewise continuous) parameterized function of continuous time. Out of them, the most computationally effective parameterization was obtained solving a rigorously defined combinatorial optimization task providing certain control performance guarantees and finally, the corresponding parameters were optimized together with the process initial conditions enabling the full potential of the production process to be exploited. Going into detail, the algorithm for joint input profile and initial conditions optimization (JIPICO) works in three stages: during stage I, a sampled-input optimization with fixed process initial conditions maximizing a user-defined criterion is performed providing an initial estimate of the optimal input profile. Using a set of parametric basis functions, this estimate is reparameterized by a stage II optimization-based procedure during which the input profile complexity is reduced while a chosen performance degradation threshold is satisfied. To address this task rigorously, *influence* (effective reduction of cardinality of the optimized parameters set resulting from elimination of particular parameter) and *performance deterioration* (degradation of the evaluative criterion observed having excluded par-

[1] This algorithm was motivated by a successful attempt published in [A.12] where the input profile had been reparameterized and subsequent parametric optimization had yielded promising results. Nevertheless, the reparameterization had been performed using an ad hoc principle and the process initial conditions had not been optimized.

ticular parameter) corresponding to each optimized input-profile parameter are exploited. Then, the parameters are labeled as either *expendable* or *indispensable* and based on the mutual exclusivity of these two sets, a simpler optimization task consisting in maximization of the *influence* only over the *expendable* parameters set is formulated and solved. It can be shown that the original reparameterization and this simpler optimization are complementary to each other and therefore, the parameters of the reparameterized input profile can be obtained as the relative complement of the solution of the simpler optimization in the initial parameters set. Finally, stage III consists in parametric optimization of both the process initial conditions and input profile parameters and to speed-up the convergence, spline interpolation of the criterion values corresponding to perturbed parameters is exploited when constructing the numerical gradient.

A case study presented in [A.4] tested the JIPICO algorithm on an example of penicillin production and compared it with three other alternatives, namely *i)* sampled-input optimization with fixed initial conditions (HGO) corresponding to the commonly used approach, *ii)* input profile reparameterization and parametric optimization with fixed initial conditions (IPRO) and *iii)* optimization of initial conditions combined with sampled-input optimization (ICO). While both IPRO and ICO brought some improvement compared with HGO (3 and 28 %, respectively), a much higher gain of more than 60 % was obtained with JIPICO algorithm; evaluating a richer set of various cultivation length, JIPICO provided improvement of $46 - 64$ % with HGO being the benchmark. It is worth noting that even for the second shortest inspected period, JIPICO achieved better result than HGO with the longest one. The comparison further revealed that the yield per optimized input profile parameter (a measure of efficiency of the used strategy) is significantly higher with JIPICO algorithm than with the commonly used approach and mostly keeps growing with increase of the cultivation length.

Starting on the following page, the publication [A.4] is presented in the original formatting.

Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 354 (2017) 8529–8551

www.elsevier.com/locate/jfranklin

Production-process optimization algorithm: Application to fed-batch bioprocess

Matej Pčolka[a,*], Sergej Čelikovský[a,b]

[a] *Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo náměstí 13, Prague 121 35, Czechia*

[b] *Institute of Information Theory and Automation of the Czech Academy of Sciences, Pod Vodárenskou věží 4, Prague 182 08, Czechia*

Received 11 November 2016; received in revised form 19 July 2017; accepted 14 October 2017
Available online 6 November 2017

Abstract

This paper presents a computationally tractable algorithm focusing on overall optimization of a production process. The proposed algorithm embraces both the input profile and the state initial conditions optimization and consists of three stages: (i) optimization of the input profile with constant initial conditions, (ii) reduction of the input profile complexity and (iii) joint optimization of the input profile parameters and state initial conditions. The newly proposed algorithm is compared with several alternatives on a series of numerical experiments representing penicillin cultivation process. As a part of the evaluation, a broader range of optimization periods is considered and not only the criterion but also the complexity of the provided input profiles is inspected. The obtained encouraging results show the superiority of the newly proposed solution and demonstrate the usefulness of the joint-optimization algorithm.

1. Introduction

Process industry (and bioprocess industry in particular) is a highly interesting application area of the optimization theory and many recent works have been devoted to optimal ma-

* Corresponding author.
E-mail addresses: pcolkmat@fel.cvut.cz, matej.pcolka@gmail.com (M. Pčolka), celikovs@utia.cas.cz (S. Čelikovský).

https://doi.org/10.1016/j.jfranklin.2017.10.012
0016-0032/

nipulation of the process inputs (see e.g. [1–8] and references therein). However, unlike the branches where the initial conditions are given by the current measurements, also the initial state of the process can be usually manipulated within some range in process industry. "Better" choice of the initial conditions can improve the achieved results by as much as several tens of per cent [9]. Even despite the strong potential, works on initial conditions optimization are rather rare and are mostly based on statistical methods [9–11]. This requires exhaustive number of real-life experiments with different initial conditions, which can be economically unbearable. Furthermore, only the effect of initial conditions is inspected and the influence of the input variables is ignored. The model-based optimization algorithm proposed in this paper takes *both* these influences into account and its performance is illustrated on the penicillin production case study. Besides the (bio)process control, such approach can be exploited in business, mathematical economics and elsewhere.

Another specific property of process control is that the processes very often involve non-coherently measured/actuated systems operating over long horizons. With commonly used discrete time models/discretized inputs [2,5,8,12–14], the dimension of the optimization task rises steeply as the sampling period decreases. To evade the curse of dimensionality, a trade-off between the optimality of the discretized solution and the memory and computational demands of the optimization would need to be sought using the traditional approaches and apparently, the optimality would be sacrificed to the acceptable complexity of the optimization. However, an input-profile re-parametrization performed as a part of the algorithm proposed in this paper effectively decreases the number of the optimized input profile parameters while keeping the optimization working in continuous time. Moreover, the re-parametrization enables to handle irregularly available state measurements, since the re-calculation of the continuous-time input profile might be performed at whichever time. Here, several works focusing on maximization of yields of a fed-batch fermentation process can be mentioned. In [15], the control function was approximated by a piecewise constant function with fixed switching times and only the constant levels were optimized. In [16], the control function parametrization was extended to incorporate also linear and quadratic function of time and the coefficients of their linear combinations were optimized, however, the switching times were again expected to be known a priori. On the other hand, in [17], the feed rate was assumed to be constant or zero and the switching times were optimized, while in [18], also the terminal time was manipulated. The most complex tasks were solved in [19] and [20], where the control function was parametrized as a piecewise constant function of time and both the levels and the switching times were optimized. Although the time-scaling transform introduced in [17,19,20] enables to optimize also the switching times, the optimized feed flow rate remains a piecewise-constant function of time. On the other hand, the algorithm proposed in this paper enables the optimized input to be *truly* whichever parametrized continuous-time function. While already optimization of the switching times brings considerable improvement, not restricting to the piecewise-constant functions only but allowing also for a broader class, even more flexibility is offered and improved results might be obtained while the computational complexity is decreased. To sum up, the currently available works suffer from the following drawbacks: (i) the control function parametrization is determined ad hoc, and (ii) the process initial conditions are given a priori and are not optimized. In some sense, the newly proposed algorithm might be regarded as a significant extension of these works since it optimizes over much broader family of the basis functions and the choice of these functions is obtained as a result of a well-defined combinatorial optimization, and in addition, it optimizes also the state initial conditions. The resulting joint optimization algorithm consists of three stages: in stage I,

the initial guess of the optimized input profile is obtained via sampled-data optimization. In stage II, the initial input profile guess is re-parametrized in continuous time using a rigorous optimization-based procedure. In stage III, the state initial conditions and the newly obtained input profile parameters are jointly optimized by a parametric optimization. The combination of all these adaptations leads to significant improvement of the fermentation yields (46–64%), as demonstrated by the presented case study.

The paper is structured as follows: Section 2 presents preliminaries for the proposed algorithm including the problem formulation and the previous solution which serves as a springboard for the new approach. The proposed solution itself is provided in Section 3 where a three-stage optimization algorithm is composed and all stages are described in detail and discussed. Section 4 illustrates the proposed approach using a penicillin production optimization case-study. The results of the newly proposed approach are compared with the original algorithm presented in the previous work and the achieved improvement demonstrates the contribution of both the newly introduced optimization of the re-parametrized input profile and state initial condition optimization. Section 5 concludes the paper.

2. Preliminaries

In this section, the optimization tasks to be dealt with are specified and the motivation for the newly proposed algorithm is provided.

2.1. Problem formulation

In many applications, the ultimate goal is optimization of particular state function at the end of certain given period. For such tasks, the following minimization criterion can be used:

$$\mathcal{J}(u, x_0) = \mathbf{F}(x(T_\mathrm{F})), \tag{1}$$

where $\mathbf{F}$ is the chosen function of process states $x = [x^1, x^2, \ldots, x^n]^\mathrm{T}$, $n \in \mathbb{N}^+$, and T_F is the given end time of the optimization period. Moreover, the solution must respect the following set of constraints:

$$\begin{aligned} &\dot{x} = f(x, u), \quad x(0) = x_0, \\ &x_{0,\min} \le x_0 \le x_{0,\max}, \;\; x_0^i \in \mathbf{X}_{0,\mathrm{adm}}^i, \;\; u \in \mathcal{U}, \end{aligned} \tag{2}$$

where $\mathcal{U}$ is the class of all admissible input functions being all measurable on $[0, T_F]$ such that $u(t) \in [u_{\min}, u_{\max}]$, $\forall t \in [0, T_F]$, $u_{\min} \in \mathbb{R}$, $u_{\max} \in \mathbb{R}$, $u_{\min} \le u_{\max}$. Here, $f(x, u)$ represents the process dynamics and $\{x_{0,\min}, x_{0,\max}\}$ and $\{u_{\min}, u_{\max}\}$ specify the acceptable intervals for initial states x_0 and inputs u. Furthermore, some initial states x_0^i might be required to belong to an admissible set of discrete values $\mathbf{X}_{0,\mathrm{adm}}^i$. Although theoretically, $\mathbf{X}_{0,\mathrm{adm}}^i$ might be whichever user-defined set of discrete values, the most probable formulation is the one with a fixed resolution step, $\mathbf{X}_{0,\mathrm{adm}}^i = \{q_i R_i | q_i \in \mathbb{Z}\}$, with $R_i \in \mathbb{R}^+$ being the fixed resolution of setting the ith initial state.

Finally, the optimization task is summarized as:

$$\begin{aligned} &\text{find } \{u^*(t), x_0^*\} = \arg \min_{u(t),\, x_0} \mathcal{J}(u, x_0) \\ &\text{subject to constraints (2).} \end{aligned} \tag{3}$$

2.2. Sampled Hamiltonian-based projected gradient method

Standard way of solving the production-process optimization task is to simplify the optimization problem by setting the state initial condition x_0 to be fixed and considering only the input u as the optimizable variable [2,5,12,21,22]. Then, the task (3) can be treated as a fixed initial state, fixed time interval and free terminal state optimal control problem. To find the input u minimizing Eq. (1), the well-known gradient approach is often performed on $\mathcal{U}_{T_{\text{samp}}}$ admissible class of T_{samp}-sampled functions defined as follows.

Definition 1. For the given sampling period $T_{\text{samp}} \in \mathbb{R}^+$ and the constraints $u_{\min} \in \mathbb{R}$, $u_{\max} \in \mathbb{R}$, $u_{\min} \leq u_{\max}$, $\mathcal{U}_{T_{\text{samp}}}$ is an admissible class of measurable T_{samp}-sampled functions such that

$$\mathcal{U}_{T_{\text{samp}}} = \{u(t) \equiv \overline{u}(k),\ \forall t \in [(k-1)T_{\text{samp}}, kT_{\text{samp}});\ u_{\min} \leq \overline{u}(k) \leq u_{\max};\ k \in \mathbb{N}^+\}.$$

The key idea is to start from an initial approximation u_0 of the optimal input and follow the direction of the negative gradient of the cost criterion $\mathcal{J}$:

$$u_{l+1} = u_l - \alpha_l \frac{\partial \mathcal{J}}{\partial u}, \tag{4}$$

where l specifies the iteration number and α_l defines the step length for the gradient search [23].

To compute the gradient of $\mathcal{J}$, define first the Hamiltonian $\mathcal{H}$ as follows:

$$\mathcal{H} = \lambda^{\mathrm{T}} f(x, u), \tag{5}$$

where $f(x, u)$ refers to model of the process dynamics and λ is the adjoint state vector having the following dynamics:

$$\frac{\mathrm{d}\lambda}{\mathrm{d}t} = -\frac{\partial \mathcal{H}}{\partial x} \tag{6}$$

with terminal condition

$$\lambda(T_{\mathrm{F}}) = -\left.\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}x}\right|_{t=T_{\mathrm{F}}}. \tag{7}$$

Defined in the above way, Hamiltonian Eqs. (5)–(7) both incorporate the information about the criterion $\mathcal{J}$ and capture the dynamics of the controlled system. In particular, using the Hamiltonian Eqs. (5)–(7), the gradient of $\mathcal{J}$ can be computed as follows:

$$\frac{\partial \mathcal{J}}{\partial u} = \frac{\partial \mathcal{H}}{\partial u}. \tag{8}$$

As a consequence, the iterative search (4) can be performed as follows:

$$u_{l+1} = u_l - \alpha \frac{\partial \mathcal{H}}{\partial u} \tag{9}$$

where Eqs. (5)–(9) are to be used at each step. After each search step, the calculated profile u_{l+1} is projected on $\mathcal{U}_{T_{\text{samp}}}$. The iterative procedure is repeated until $|\mathcal{J}(u_{i+1}) - \mathcal{J}(u_i)| < \epsilon$ for some suitably selected $\epsilon > 0$.

More precisely, each iteration l is executed as follows: at first, the input profile u_l is used to obtain the state x profiles by integrating the equation $\dot{x} = f(x, u)$ with state initial condition x_0 over time interval [0, T_{F}]. Then, the terminal condition (7) for the adjoint state vector is obtained by evaluating $-\mathrm{d}\mathbf{F}/\mathrm{d}x$ at $t = T_{\mathrm{F}}$. This terminal condition $\lambda(T_{\mathrm{F}})$ is used together

with input u_l and state profiles x to integrate Eq. (6) backward in time. After that, all the adjoint state vector λ, state vector x and input u_l are exploited to evaluate the gradient of the Hamiltonian $\partial\mathcal{H}/\partial u$ which is then used to update the input profile from u_l to u_{l+1} according to Eq. (9) and after the projection on $\mathcal{U}_{T_{\text{samp}}}$, the new iteration $(l+1)$ starts. Considering the sampling period T_{samp}, the resulting optimal input is then represented as a vector of optimal input samples, $\overline{\mathbf{u}} = \{\overline{u}(k) \mid k = 1, 2, \ldots, P\}$, where k corresponds to the sampling instant and $T_F = P \times T_{\text{samp}}$. More details can be found in [23,24].

This approach suffers from a severe drawback—in case that the sampling period decreases, the complexity of the optimization grows. In the previous work [25], an alternative consisting in re-parametrization of the input profile was proposed to overcome this issue. However, the re-parametrization was not performed in a systematic way. In the current paper, the methodology to derive the input profile re-parametrization is studied in detail and a formalized approach is presented.

The computations described by Eqs. (5)–(9) are applicable only if the state initial condition x_0 is given and there is no straightforward way how to adapt the iterative schemes Eqs. (5)–(9) when x_0 is free. Note that with the free state initial condition, even the determination of the extremals using the Pontryagin's maximum principle turns into a difficult two-point boundary problem [26] which needs to be solved for both the states and the adjoint states.

Therefore, another purpose of this paper is to propose an alternative way how to treat free process initial condition—this will be described as a part of complex algorithm given in the sequel.

3. Three-stage optimization procedure

As already mentioned, this paper presents an algorithm incorporating both the optimal selection of the process initial conditions and manipulated variables and lowering complexity/computational demands caused by dense sampling by introducing the input profile re-parametrization. This algorithm is given by the following three-stage procedure:

Algorithm 1 Joint input profile and initial conditions optimization (JIPICO)**.**

Inputs: cost criterion $\mathcal{J}$, set of input constraints, dynamics of the controlled process $\dot{x} = f(x, u)$, length of the optimization horizon T_F, initial approximation of state initial conditions $x_{0,0}$, initial approximation of input profile u_0, suboptimality threshold $\Delta\mathcal{J}$.

Outputs: vector x_0^* of optimal state initial conditions and vector $\mathfrak{P}^*$ of optimal input profile parameters.

I. Consider the state initial conditions to be fixed, $x(0) = x_{0,0}$, find the optimal input profile $\overline{u}^*(t)$ minimizing the given optimization criterion on $\mathcal{U}_{T_{\text{samp}}}$.

II. To reduce the complexity of the input profile, find a set of parametric basis functions $\mathfrak{F}(t, \mathbf{P}_A)$, $t \in [0, T_F]$, vector of parameters $\mathbf{P} = [\mathbf{P}_T^{\mathrm{T}}, \mathbf{P}_S^{\mathrm{T}}, \mathbf{P}_A^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{P} \in \mathbb{R}^{N_T+N_S+N_A}$, N_T, N_S, $N_A \in \mathbb{Z}_0^+$, and the mapping $\Pi : \mathbb{R}^{N_T+N_S+N_A} \to \mathcal{U}$,

$$\Pi(\mathbf{P}) := u(\mathbf{P})(t) = \begin{cases} u_1 = \sum_q p_{\mathrm{S},1,q}\mathfrak{F}(t, \mathbf{P}_A) & 0 \le t < p_{\mathrm{T},1}, \\ u_2 = \sum_q p_{\mathrm{S},2,q}\mathfrak{F}(t, \mathbf{P}_A) & p_{\mathrm{T},1} \le t < p_{\mathrm{T},2}, \\ \vdots & \vdots \\ u_{N_\mathrm{T}} = \sum_q p_{\mathrm{S},N_\mathrm{T},q}\mathfrak{F}(t, \mathbf{P}_A) & p_{\mathrm{T},N_\mathrm{T}-1} \le t \le p_{\mathrm{T},N_\mathrm{T}} = \mathrm{T_F}, \end{cases}$$

$p_{T,1}, p_{T,2}, \ldots, p_{T,N_T} \in \mathbf{P}_T$, $p_{S,1,q}, p_{S,2,q}, \ldots, p_{S,N_T,q} \in \mathbf{P}_S$, such that for the given suboptimality threshold $\Delta\mathcal{J} \in \mathbb{R}_0^+$, the performance deterioration condition

$$\mathcal{J}(u(\overline{\mathbf{P}}^*)(t)) - \mathcal{J}(\overline{u}^*(t)) \leq \Delta\mathcal{J}$$

is satisfied.

III. Minimize function $\overline{\mathcal{J}}(\mathbf{P}, x_0)$ of $N_T + N_S + N_A + n$ real variables defined as follows:

$$\overline{\mathcal{J}}(\mathbf{P}, x_0) = \mathcal{J} \circ \Pi : \mathbb{R}^{N_T+N_S+N_A+n} \to \mathbb{R}.$$

Note that the first two stages of the JIPICO algorithm are to be performed off-line before the production process starts. The third stage is performed on-line every time a measurement of the current system variables arrives in order to introduce necessary measurement feedback and enable disturbance rejection. The parameters that are irrelevant with respect to the time moment of the arrival of the new measurements are omitted. Let us also remark that no assumptions on frequency or regularity of the measurements are considered which makes the procedure more robust against irregular or non-coherent measurements.

The particular stages of the JIPICO algorithm are described in more detail in the following text.

3.1. JIPICO stage I

During the JIPICO stage I, the optimization problem is treated as a fixed initial state, fixed time interval and free terminal state optimal control problem. The state initial conditions x_0 are chosen such that they satisfy the initial condition constraints $x_{0,\min} \leq x_0 \leq x_{0,\max}$. As the candidate for the optimization routine, sampled Hamiltonian-based projected gradient method described in detail in Section 2.2 belonging to the optimal control methods family [23] has been chosen. More details on this method can be found in [22,24,27] and references therein. For possible alternatives, see [28–30].

Summarizing, the JIPICO stage I takes the entry data $x_{0,0}$, u_0, $T_{\text{samp}} > 0$ and produces $\overline{u}^*$ being T_{samp}-sampled optimal input function. Rather than decrease T_{samp}, the next JIPICO stage is more convenient.

3.2. JIPICO stage II

The systematic and rigorous input profile re-parametrization—the purpose of the JIPICO stage II—provided in the sequel is one of the main contributions and was motivated by the successful first attempt in this direction in [25].

The coherently sampled input sequence $\overline{\mathbf{u}} = \{\overline{u}(k) \mid k = 1, 2, \ldots, P\}$, $P = T_{\text{F}}/T_{\text{samp}}$ obtained by JIPICO stage I will be re-parametrized and represented as follows:

$$u(\mathbf{P}, \mathfrak{F}, t) = \begin{cases} u_1 = \sum_{q=1}^{s_1} p_{\text{S},1,q} \mathfrak{f}_{1,q} & 0 \leq t < p_{\text{T},1}, \\ u_2 = \sum_{q=1}^{s_2} p_{\text{S},2,q} \mathfrak{f}_{2,q} & p_{\text{T},1} \leq t < p_{\text{T},2}, \\ \vdots & \vdots \\ u_{N_\text{T}} = \sum_{q=1}^{s_{N_\text{T}}} p_{\text{S},N_\text{T},q} \mathfrak{f}_{N_\text{T},q} & p_{\text{T},N_\text{T}-1} \leq t \leq p_{\text{T},N_\text{T}} = \text{T}_\text{F}. \end{cases} \tag{10}$$

To specify the ingredients of Eq. (10), the following definitions are given.

Definition 2. *Time stamps* $p_{\mathrm{T},1},\ p_{\mathrm{T},2},\ \ldots,\ p_{\mathrm{T},N_\mathrm{T}} \in \mathbb{R}^+$ define particular time subintervals on which piece-wise continuous parts of $u(\mathbf{P}, \mathfrak{F}, t)$ are defined. These subintervals are denoted by $i \in \{1, 2, \ldots, N_\mathrm{T}\}$. Collection of all time stamps $\mathbf{P_T}$ is then defined as $\mathbf{P}_\mathrm{T} = \{p_{\mathrm{T},i} \,|\, i = 1, 2, \ldots, N_\mathrm{T}\ ;\ 0 < p_{\mathrm{T},1} < p_{\mathrm{T},2} < \cdots < p_{\mathrm{T},N_\mathrm{T}} = T_\mathrm{F}\}$.

Definition 3. *Simple* parameters $p_{\mathrm{S},1},\ p_{\mathrm{S},2},\ \ldots \in \mathbb{R}$ are coefficients of linear combinations of basis functions that define the piece-wise continuous parts $u_1,\ u_2,\ \ldots,\ u_{N_T}$ of $u(\mathbf{P}, \mathfrak{F}, t)$. Collection of all simple parameters $\mathbf{P_S}$ is then defined as $\mathbf{P}_\mathrm{S} = \{p_{\mathrm{S},i,q} \,|\, u_i = \sum_{q=1}^{s_i} p_{\mathrm{S},i,q}\mathfrak{f}_{i,q}\ ,\ i = 1, 2, \ldots, N_\mathrm{T}\}$.

Definition 4. *Simple* basis function $\mathfrak{f}_\mathrm{S}^r$ is r-th power of continuous time t, $\mathfrak{f}_\mathrm{S}^r(t) = t^r$. Set of all simple functions $\mathfrak{F}_\mathrm{S}$ is then defined as $\mathfrak{F}_\mathrm{S} = \{t^r \,|\, r \in \mathbb{Z}\}$.

Definition 5. *Advanced* basis function $\mathfrak{f}_\mathrm{A}$ is a function of both continuous time t and parameters, $\mathfrak{f}_\mathrm{A} = \mathfrak{f}_\mathrm{A}(\cdot\,, t)$. Set of advanced functions $\mathfrak{F}_\mathrm{A}$ is then defined as $\mathfrak{F}_\mathrm{A} = \{\mathfrak{f}_\mathrm{A}(\cdot\,, t)\}$.

Typical examples of advanced basis functions are goniometric functions, hyperbolic functions or sigmoid function.

Definition 6. *Advanced* parameters $p_A \in \mathbb{R}$ are parameters included in advanced basis functions. For each advanced parameter p_A there exists advanced function $\mathfrak{f}_\mathrm{A}$ such that $\mathfrak{f}_\mathrm{A} = \mathfrak{f}_\mathrm{A}(p_\mathrm{A}, \cdot\,, t)$. Collection of advanced parameters $\mathbf{P_A}$ is then defined as $\mathbf{P}_\mathrm{A} = \{p_{\mathrm{A},i,q,m} \,|\, \mathfrak{f}_{i,q} = \mathfrak{f}_{\mathrm{A},i,q}(p_{\mathrm{A},i,q,m}, \cdot, t)\ ;\ \mathfrak{f}_{\mathrm{A},i,q} \in \mathfrak{F}_\mathrm{A};\ m \in \mathbb{N}^+\}$.

Remark 1. Sets of simple and advanced basis functions are mutually disjoint, $\mathfrak{F}_\mathrm{S} \bigcap \mathfrak{F}_\mathrm{A} = \emptyset$. Family of sets $\mathbf{P_T}$, $\mathbf{P_S}$ and $\mathbf{P_A}$ is mutually disjoint.

Definition 7. Set of all basis functions $\mathfrak{F}$ is defined as $\mathfrak{F} = \mathfrak{F}_\mathrm{S} \bigcup \mathfrak{F}_\mathrm{A}$.

Definition 8. Collection of all used parameters $\mathbf{P}$ is defined as $\mathbf{P} = \mathbf{P}_\mathrm{T} \bigcup \mathbf{P}_\mathrm{S} \bigcup \mathbf{P}_\mathrm{A}$.

Now the re-parametrization problem can be formulated as an optimization problem.
Re-parametrization optimization problem. Fix an admissible performance degradation threshold $\Delta\mathcal{J} \geq 0$. Then, the choice of the proper parametrization can be expressed as the following optimization task:

$$
\begin{aligned}
&\min\ N_\mathrm{P}\\
\text{subject to}:\ &\mathcal{J}(u(\mathbf{P}, \mathfrak{F}, t)) \leq \mathcal{J}(\overline{\mathbf{u}}) + \Delta\mathcal{J}\\
&\text{constraints (2)}\\
&\Delta\mathcal{J} \geq 0\\
&\overline{\mathbf{u}} = \{\overline{u}(k) \,|\, k = 1, 2, \ldots, P\}\\
&\mathbf{P} = \mathbf{P}_\mathrm{T} \bigcup \mathbf{P}_\mathrm{S} \bigcup \mathbf{P}_\mathrm{A}\ ;\ |\mathbf{P}| = N_\mathrm{P}\\
&\mathbf{P}_\mathrm{T} = \{p_{\mathrm{T},i} \,|\, i = 1, 2, \ldots, N_\mathrm{T}\ ;\ 0 < p_{\mathrm{T},1} < p_{\mathrm{T},2} < \cdots < p_{\mathrm{T},N_\mathrm{T}} = T_\mathrm{F}\}\\
&\mathbf{P}_\mathrm{S} = \left\{ p_{\mathrm{S},i,q} \,|\, u_i = \sum_{q=1}^{s_i} p_{\mathrm{S},i,q}\mathfrak{f}_{i,q}\ ;\ \mathfrak{f}_{i,q} \in \mathfrak{F} \right\}\\
&\mathbf{P}_\mathrm{A} = \{p_{\mathrm{A},i,q,m} \,|\, \mathfrak{f}_{i,q} = \mathfrak{f}_{\mathrm{A},i,q}(p_{\mathrm{A},i,q,m}, t)\ ;\ \mathfrak{f}_{\mathrm{A},i,q} \in \mathfrak{F}_\mathrm{A}\}\\
&\mathfrak{F} = \mathfrak{F}_\mathrm{S} \bigcup \mathfrak{F}_\mathrm{A}
\end{aligned}
$$

$$\mathfrak{F}_S = \{\mathfrak{f}_S^r(t) = t^r \mid r \in \mathbb{Z}\}$$
$$\mathfrak{F}_A = \{\mathfrak{f}_A(\mathbf{P}_A, t)\}. \tag{11}$$

In other words, given the set of basis functions $\mathfrak{F}$ (user-defined entry of the algorithm) and the vector of optimized input samples $\overline{\mathbf{u}}$ (obtained at first stage of the algorithm, see Section 3.1), we are looking for the smallest set of parameters $\mathbf{P}$ of cardinality N_P such that the degradation of the performance index $\mathcal{J}$ is not higher than $\Delta\mathcal{J}$ while the original constraints given by Eq. (2) are satisfied. Here, $\Delta\mathcal{J}$ is a user-defined tuning parameter and can be chosen either absolutely or relatively with respect to $\mathcal{J}(\overline{\mathbf{u}})$.

Theorem 1. *Given the initial estimate* $\overline{\mathbf{u}}$ *and the corresponding cost criterion value* $\mathcal{J}(\overline{\mathbf{u}})$, *the optimization problem (11) has always a feasible solution* $N_P \leq 2P$ *for any* $\Delta\mathcal{J} \geq 0$.

Proof. Let us start with the most strict condition, $\Delta\mathcal{J} = 0$. In such case, the initial estimate $\overline{\mathbf{u}}$ can be directly used to derive the parametrization with the following sets:

$$\mathbf{P}_T = \{k\,T_{\text{samp}} \mid k = 1, 2, \ldots, P\}$$
$$\mathbf{P}_S = \{\overline{u}(k) \mid k = 1, 2, \ldots, P\}$$
$$\mathbf{P}_A = \emptyset$$
$$\mathfrak{F}_S = \{t^0\}$$
$$\mathfrak{F}_A = \emptyset.$$

With this parametrization, $\mathcal{J}(u(\mathbf{P}, \mathfrak{F}, t)) - \mathcal{J}(\overline{\mathbf{u}}) = 0$ which satisfies the condition for maximal allowed perturbation of the cost criterion. Moreover, $N_P = |\mathbf{P}| = |\mathbf{P}_T \bigcup \mathbf{P}_S \bigcup \mathbf{P}_A| = |\mathbf{P}_T| + |\mathbf{P}_S| + |\mathbf{P}_A| = 2P$. This solution is valid and feasible also for any $\Delta\mathcal{J} > 0$. This completes the proof. □

Remark 2. Theorem 1 and its proof provide a solution that ensures the feasibility of task (11). Moreover, they provide also the upper estimate for the cardinality of the set of parameters $\mathbf{P}$ and a criterion according to which the "meaningfulness" of any parametrization can be evaluated. Assumption of the most strict condition $\Delta\mathcal{J} = 0$ results in $N_P = 2P$ and therefore, parametrizations with more than $2P$ parameters are *ineffective* with respect to the optimization task (11).

To find a parametrization satisfying the constraints given by Eq. (11) and being more effective in the sense of cardinality of the set of used parameters, the following procedure is proposed.

Algorithm 2 (Simple parameter exclusion)**.** Exclusion of simple parameter $p_{S,i,q}$ from the set $\mathbf{P}$ is equivalent to setting $p_{S,i,q}$ equal to zero. The rest of parameters is updated to $\overline{\mathbf{P}}$ such that $\|u(\overline{\mathbf{P}}, \mathfrak{F}, t) - u(\mathbf{P}, \mathfrak{F}, t)\|_2$ is minimized and input constraints (2) are satisfied. Then, the updated profile $u(\overline{\mathbf{P}}, \mathfrak{F}, t)$ is equivalent to $u(\{\ldots, \overline{p}_{S,i,q-1}, 0, \overline{p}_{S,i,q+1}, \ldots\}, \mathfrak{F}, t)$.

Algorithm 3 (Advanced parameter exclusion)**.** Exclusion of advanced parameter $p_{A,i,q,m}$ from the set $\mathbf{P}$ is equivalent to setting $p_{A,i,q,m}$ equal to zero. The rest of parameters is updated to $\overline{\mathbf{P}}$ such that $\|u(\overline{\mathbf{P}}, \mathfrak{F}, t) - u(\mathbf{P}, \mathfrak{F}, t)\|_2$ is minimized and input constraints (2) are satisfied. Then, the updated profile $u(\overline{\mathbf{P}}, \mathfrak{F}, t)$ is equivalent to $u(\{\ldots, \overline{p}_{A,i,q,m-1}, 0, \overline{p}_{A,i,q,m+1}, \ldots\}, \mathfrak{F}, t)$.

Table 1
Influence $\mathcal{I}$ of the parameters.

Parameter	T_1	T_2	a	b	c	d	e	f	g	h	i
Influence $\mathcal{I}$ (−)	8	–	1	1	1	3	3	1	3	3	1

Algorithm 4 (Time stamp exclusion)**.** If time stamp $p_{\mathrm{T},i}$ is excluded from the set $\mathbf{P}$, the input profile $u(\mathbf{P}\backslash\{p_{\mathrm{T},i}\}, \mathfrak{F}, t)$ changes into:

$$u(\mathbf{P}\backslash\{p_{\mathrm{T},i}\}, \mathfrak{F}, t) = \begin{cases} u_1 & 0 \le t < p_{\mathrm{T},1}, \\ \vdots & \vdots \\ u_{i-1} & p_{\mathrm{T},i-2} \le t < p_{\mathrm{T},i-1}, \\ \hat{u}_i & p_{\mathrm{T},i-1} \le t < p_{\mathrm{T},i+1}, \\ u_{i+2} & p_{\mathrm{T},i+1} \le t < p_{\mathrm{T},i+2}, \\ \vdots & \vdots \\ u_{N_\mathrm{T}} & p_{\mathrm{T},N_\mathrm{T}-1} \le t \le p_{\mathrm{T},N_\mathrm{T}} = \mathrm{T_F}. \end{cases} \tag{12}$$

Parameters of $\hat{u}_i$ are updated such that $\|\hat{u}_i - u_{i:i+1}\|_2$ is minimized and input constraints (2) are satisfied. Here $u_{i:i+1}$ corresponds to unification of i-th and $(i+1)$-st subfunctions of the original input profile $u(\mathbf{P}, \mathfrak{F}, t)$,

$$u_{i:i+1} = \begin{cases} u_i & p_{\mathrm{T},i-1} \le t < p_{\mathrm{T},i}, \\ u_{i+1} & p_{\mathrm{T},i} \le t < p_{\mathrm{T},i+1}. \end{cases}$$

Remark 3. In order to have input profile $u(\mathbf{P}, \mathfrak{F}, t)$ well-defined, time stamp $p_{\mathrm{T},N_\mathrm{T}}$ can not be eliminated.

In order to quantify the effect of excluding certain parameter p_i from the original set $\mathbf{P}$, its *influence* is defined as follows.

Definition 9. *Influence* $\mathcal{I}(p_i) \in \mathbb{N}$ of the parameter $p_i \in \mathbf{P}$ is the effective decrease of the cardinality of the set $\mathbf{P}$ caused by excluding the i-th parameter p_i from the set $\mathbf{P}$,

$$\mathcal{I}(p_i) = |\mathbf{P}| - |\mathbf{P}\backslash\{p_i\}|.$$

Example 1. Let us consider a function parametrized as follows:

$$u = \begin{cases} u_1 = a + bt & 0 \le t < T_1, \\ u_2 = c + d\cos(et + f) + g\tanh(ht + i) & T_1 \le t \le T_2 = T_F. \end{cases} \tag{13}$$

Based on the parametrization, the used basis functions are $\mathfrak{F}_\mathrm{S} = \{t^0, t^1\}$, $\mathfrak{F}_\mathrm{A} = \{\cos(\cdot), \tanh(\cdot)\}$ and the sets of parameters are $\mathbf{P}_\mathrm{T} = \{T_1, T_2\}$, $\mathbf{P}_\mathrm{S} = \{a, b, c, d, g\}$ and $\mathbf{P}_\mathrm{A} = \{e, f, h, i\}$. Table 1 provides the influence $\mathcal{I}$ of the used parameters.

The fact that the influence $\mathcal{I}(T_2)$ for the time stamp T_2 is missing results directly from Remark 3. Moreover, from Algorithm 4 it straightforwardly follows that the time stamp T_1 is the most influential parameter. Furthermore, $\mathcal{I}(e) = 3$ and $\mathcal{I}(h) = 3$ result from the fact that if one of the parameters $\{e, h\}$ is eliminated, the corresponding advanced functions ($\cos(\cdot)$ or $\tanh(\cdot)$) turns into a constant. However, the constant function is already included (parameter c) and therefore, the particular advanced function does not effectively add any new degree of freedom and thus it can be absolutely eliminated.

Remark 4. Example 1 shows that the least influential parameters are usually the advanced parameters and simple parameters multiplying simple functions—if one of them is eliminated, the number of parameters is reduced usually only by one. Simple parameters corresponding to advanced functions are moderately influential—their influence $\mathcal{I}$ is equal to 1 + number of the advanced parameters included in the particular advanced function. Last of all, time stamps are usually the most influential group of parameters. Their elimination is equivalent to elimination of the whole following time subinterval and as a result, the effective number of the remaining parameters is decreased by 1 + number of the simple parameters + number of the advanced parameters related to the following time subinterval.

Exploiting the influence $\mathcal{I}$, the procedure solving the task (11) can be proposed. First of all, initial guesses of the sets $\mathbf{P}_{\text{init}}$ and $\mathfrak{F}_{\text{init}}$ shall be obtained.

Assumption 1. For the initial parametrization $\{\mathbf{P}_{\text{init}}, \mathfrak{F}_{\text{init}}\}$, the following holds:

$$\begin{aligned}\mathcal{J}(u(\mathbf{P}_{\text{init}}, \mathfrak{F}_{\text{init}}, t)) &= \mathcal{J}(\overline{\mathbf{u}}),\\ |\mathbf{P}_{\text{init}}| &\geq C_{\mathbf{P}},\\ |\mathfrak{F}_{\text{init}}| &\geq C_{\mathfrak{F}},\end{aligned}$$

where $C_{\mathbf{P}} \gg 2P$ and $C_{\mathfrak{F}}$ are sufficiently large positive scalars.

Assumption 1 introduces an initial estimate that is obviously ineffective, on the other hand, it offers a broad space for reduction of the number of parameters. Suitable initial parametrization can be obtained using nonparametric identification techniques [31] and extending the result with suitable basis functions. Then, the performance deterioration caused by its exclusion from the original parameter set is calculated for each of the parameters p_i.

Definition 10. *Performance deterioration* $\tilde{\mathcal{J}}(p_i)$ is the increase of criterion $\mathcal{J}$ that is caused by exclusion of p_i from $\mathbf{P}$, $\tilde{\mathcal{J}}(p_i) = \mathcal{J}(u(\mathbf{P}\backslash\{p_i\}, \mathfrak{F}, t)) - \mathcal{J}(\overline{\mathbf{u}})$.

Based on the introduced performance deterioration, the parameters are divided into two sets—the *expendable* parameters and *indispensable* parameters.

Definition 11. *Expendable parameter* is such parameter p_i that $\tilde{\mathcal{J}}(p_i) \leq \Delta\mathcal{J}$. Set of all expendable parameters is then $\mathbf{P}_{\text{E}} = \{p_i | \tilde{\mathcal{J}}(p_i) \leq \Delta\mathcal{J}\}$.

Definition 12. *Indispensable parameter* is such parameter p_i that $\tilde{\mathcal{J}}(p_i) > \Delta\mathcal{J}$. Set of all indispensable parameters is then $\mathbf{P}_{\text{I}} = \{p_i | \tilde{\mathcal{J}}(p_i) > \Delta\mathcal{J}\}$.

Corollary 1. *Given* $\Delta\mathcal{J} \geq 0$, *indispensable parameters* $\mathbf{P}_{\text{I}}$ *can be directly omitted from the parameters reduction (11). Therefore, only the expendable parameters* $\mathbf{P}_{\text{E}}$ *can be reduced by task (11).*

Based on this, it can be seen that the parameters $\mathbf{P}_{\text{O}}$ that can be excluded from the original parameters set $\mathbf{P}_{\text{init}}$ such that $\tilde{\mathcal{J}}(\mathbf{P}_{\text{O}}) \leq \Delta\mathcal{J}$ form a subset of expendable parameters, $\mathbf{P}_{\text{O}} \subset \mathbf{P}_{\text{E}}$. Now, let us formulate an optimization task complementary to the task (11):

$$\text{find } \mathbf{P}_{\text{O}} = \left\{ p_i \in \mathbf{P}_{\text{E}} | \arg\max \left(\sum_{\text{i}=1}^{|\mathbf{P}_{\text{O}}|} \mathcal{I}(\text{p}_\text{i}) \right); |\mathbf{P}_{\text{O}}| \geq |\mathbf{P}_{\text{init}}| - 2\text{P}; \tilde{\mathcal{J}}(\mathbf{P}_{\text{O}}) \leq \Delta\mathcal{J} \right\}. \tag{14}$$

The set $\mathbf{P}_{\text{O}}$ being the solution of the optimization task (14) can be found using combinatorial optimization techniques [32–34]. Furthermore, it can be seen that tasks (11) and (14) are

complementary to each other. Therefore, having obtained the solution of the task (14) $\mathbf{P}_{\mathrm{O}}$, the solution $\overline{\mathbf{P}}$ of task (11) is $\overline{\mathbf{P}} = \mathbf{P}_{\mathrm{init}} \backslash \mathbf{P}_{\mathrm{O}}$ with cardinality $N_{\mathrm{P}} = |\overline{\mathbf{P}}| = |\mathbf{P}_{\mathrm{init}}| - |\mathbf{P}_{\mathrm{O}}| \leq 2P$.

Regarding the method used to solve the re-parametrization task in this paper, a genetic algorithm implemented by the authors was used to provide the solution, however, any of a fair variety of the available MIP solvers [35–38] can be exploited as an alternative.

3.3. JIPICO stage III

Having succesfully performed the parametrization, set of parameters $\overline{\mathbf{P}}$ is obtained. This set is extended with the vector of n optimizable state initial conditions to form the initial set of optimizable parameters, $\mathcal{P}_0 = \overline{\mathbf{P}} \bigcup x_{0,0}$. The optimization of these parameters is then performed iteratively as follows:

First, at every iteration l of the third-stage optimization, each of the optimizable parameters $\mathfrak{p}_m \in \mathcal{P}$, $m \in \{1, 2, \ldots, N_{\mathcal{P}} = |\mathcal{P}|\}$, is perturbed while the other parameters are kept fixed at their values from the previous iteration,

$$\tilde{\mathcal{P}}_{m,l} = [\mathfrak{p}_{1,l-1}, \mathfrak{p}_{2,l-1}, \ldots, \mathfrak{p}_{m-1,l-1}, \tilde{\mathfrak{p}}_m, \mathfrak{p}_{m+1,l-1}, \ldots, \mathfrak{p}_{N_{\mathcal{P}},l-1}]^{\mathrm{T}}. \tag{15}$$

The range of the perturbation might be chosen with respect to different criteria such as physical and/or technical limits of the particular parameters, etc. Additive perturbations in the following form are considered:

$$\tilde{\mathfrak{p}}_m = \mathfrak{p}_{m,l-1} + \Delta_{\mathfrak{p}_m}.$$

Here, $\Delta_{\mathfrak{p}_m}$ refers to a set of perturbations of the m-th optimizable parameter. It should be remarked that $\Delta_{\mathfrak{p}_m}$ might differ for each parameter $\mathfrak{p}_m$, however, the most straightforward way is to choose

$$\Delta_{\mathfrak{p}_m} \in PS^m \times LPS^m$$

where PS^m is a set of (both negative and positive) multiples and LPS^m stands for the elementary (least) perturbation step of the m-th parameter. PS^m might vary from iteration to iteration since only those perturbations that do not violate constraints (2) are admitted and the sets PS^m are accordingly updated. LPS^m can be considered as tuning parameter. It should be realized that due to perturbation by sets of perturbations $\{\Delta_{\mathfrak{p}_m}\}$, $\{\tilde{\mathcal{P}}_{m,l}\}$ is not just a single set but a *set of perturbed parameter sets*.

Exploiting $\{\tilde{\mathcal{P}}_{m,l}\}$, the sets of cost criterion values $\{\mathcal{J}_{\tilde{m}}\}$ corresponding to the perturbed set of each optimizable parameter can be obtained, namely

$$\{\mathcal{J}_{\tilde{m}}\} = \mathcal{J}(\tilde{\mathcal{P}}_{m,l}). \tag{16}$$

Having gathered the sets of the perturbed cost criterion values for all optimized parameters, spline interpolation [39] of each of these sets $\mathcal{J}_{\tilde{m}}$ is performed. After the interpolation, $N_{\mathcal{P}} = |\mathcal{P}|$ splines $S_1, S_2, \ldots, S_{N_{\mathcal{P}}}$ are at disposal and the spline approximation of the optimization criterion is expressed as a function of the particular optimization parameter perturbation,

$$S_m \approx \mathcal{J}(\Delta_{\mathfrak{p}_m}). \tag{17}$$

Here, the piece-wise polynomial character of the splines can be exploited—this feature enables to find the minimum of each spline analytically. Finally, parameter perturbation values

$$\mathfrak{d}_m = \arg\min S_m \tag{18}$$

corresponding to the minima of the interpolated splines are lined up to form the optimization direction

$$\mathcal{D} = [\mathfrak{d}_1, \mathfrak{d}_2, \ldots, \mathfrak{d}_{N_\mathcal{P}}] \tag{19}$$

which is then used to move along in the $N_\mathcal{P}$-dimensional optimization parameter space,

$$\mathcal{P}_l = \mathcal{P}_{l-1} + \mathcal{D}. \tag{20}$$

To prevent confusion, let us note that $\mathcal{P}_l$ stands for the whole set of $N_\mathcal{P}$ optimized parameters at l-th iteration of the procedure.

In order to satisfy the constraints imposed on the input profile and initial states, the following postprocessing is performed. Each parameter $\mathfrak{p}_m$ is at every iteration l projected on the corresponding admissible interval $[\underline{\mathfrak{p}}_{m,l}, \overline{\mathfrak{p}}_{m,l}]$,

$$\mathfrak{p}_{m,l} = \begin{cases} \underline{\mathfrak{p}}_{m,l} & \mathfrak{p}_{m,l} \leq \underline{\mathfrak{p}}_{m,l}, \\ \mathfrak{p}_{m,l} & \mathfrak{p}_{m,l} \in [\underline{\mathfrak{p}}_{m,l}, \overline{\mathfrak{p}}_{m,l}], \\ \overline{\mathfrak{p}}_{m,l} & \mathfrak{p}_{m,l} \geq \overline{\mathfrak{p}}_{m,l}. \end{cases}$$

Let us note that while handling of the constraints for the state initial conditions is straightforward, the constraints for the input profile parameters need to be extracted from the original input profile constraints such that $u_{\min} \leq u(\mathbf{P}, \mathfrak{F}, t) \leq u_{\max}$ is satisfied. Furthermore, each optimizable state initial condition $\mathfrak{p}^{\text{init}} \in \{\mathcal{P} \backslash \mathbf{P}\}$ that is expected to be set with finite resolution is then projected on the nearest integer multiple of its admissible resolution R,

$$\mathfrak{p}^{\text{init}} = \arg\min(\|\mathfrak{p}^{\text{init}} - q\,R\|),\ q \in \mathbb{Z}.$$

The above-described procedure is performed until the chosen convergence criterion is satisfied. After convergence, the set of optimal input profile parameters corresponding to parametrization (10) and the set of optimal state initial conditions are obtained.

3.4. Summary

The whole procedure is illustrated by its flow chart diagram in Fig. 1. At the very beginning, the JIPICO algorithm is provided with the estimates of the state initial conditions and input profile (the algorithm entries). During the off-line stage I, the gradient search for the optimal input profile is performed starting from the provided estimate while the given estimates of the state initial conditions are considered. Let us note this is the only time the input profile optimization is carried out in a sampled-data fashion. Stage II (also performed off-line) leaves the state initial conditions intact and re-parametrizes the input profile such that the parameters set is reduced and the user-defined suboptimality threshold $\Delta\mathcal{J}$ is satisfied. Stage III starts with the original state initial conditions and the parameters of the re-parametrized input profile and performs the gradient optimization as described in Section 3.3. After the convergence, the optimal state initial conditions and input profile parameters being the algorithm outputs are obtained.

3.5. Practical implementation remarks

Let us note that although the state initial conditions optimization might not be realizable in some applications, the algorithm can still be used. The input profile parametrization offers more flexibility for the optimization and as the number of the optimized parameters is usually

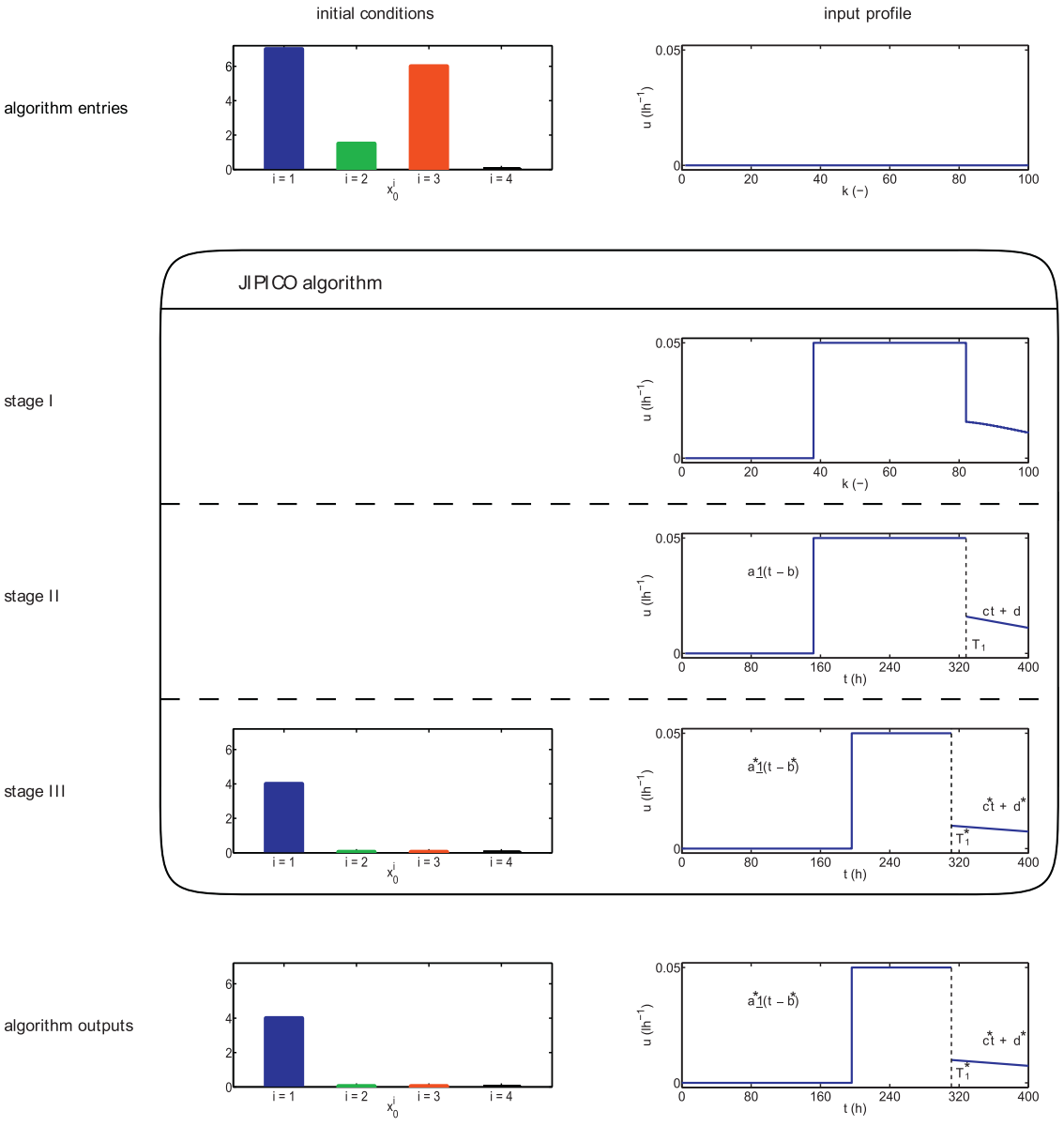

Fig. 1. JIPICO algorithm.

significantly reduced compared to the T_{samp}-sampled input profile, the proposed algorithm can offer an attractive alternative to the currently used optimization approaches.

In real-life applications, robustness of the optimal control is usually ensured by employing the receding horizon principle where the control moves are re-calculated based on the currently available measurements. Following this paradigm, the on-line JIPICO stage III can be repeated each time the new measurements arrive. Nevertheless, in the process industry and in the bioengineering area in particular, the states are very rarely measured at exact specific times with fixed sampling period. Being partly motivated by this practical phenomenon, the JIPICO algorithm makes no assumptions on frequency or regularity of the measurements. This is achieved thanks to the fact that the re-parametrization (stage II) brings the input profile from

"discrete time/sampled world" into "continuous time world". As a result, only the preliminary off-line stage I optimizes in the sampled-data fashion while the optimized input profile is considered to be a function of continuous time at the on-line stage III. Here, the improved robustness compared to the sampled-data optimal control can be gained—as long as the optimal profile is considered to be a function of continuous time, it is not necessary to extrapolate the measurement values for $t = k \times T_{\text{samp}}$ to obtain the state values at the multiples of the sampling period and the re-calculation of the optimal input profile might be performed at any time the new measurement is available. Although not crucial for tiny sampling periods and frequently performed measurements, the severity of this issue increases for larger sampling periods and non-coherent measurements where the extrapolation errors grows. This holds especially in case of unknown/neglected system dynamics, uncertain system parameters, etc.

4. Case study: penicillin optimization

In this section, a specific case of penicillin optimization is introduced and the performance of the proposed algorithm is demonstrated.

4.1. Model of the controlled system

For the purposes of this work, the process of penicillin cultivation is considered. This cultivation is carried out in the fed-batch mode, which means that no significant cultivation broth withdrawal is allowed except of the measurement samples and the feed supply of nutritive elements is provided according to the needs of the micro-organisms and therefore, the process might be operated at specific rates close to their optimal values [40]. The basic dynamic behavior of this system can be described in terms of differential equations as follows:

$$\begin{aligned}
\dot{x}_1 &= u - K_{\text{vap}} x_1, \\
\dot{x}_2 &= (\mu - K_{\text{D}}) x_2 - \left(\frac{u}{x_1} - K_{\text{vap}} \right) x_2, \\
\dot{x}_3 &= -\left(\frac{\mu}{Y_{\text{X/S}}} + \frac{\pi}{Y_{\text{P/S}}} \right) x_2 + \frac{C_{\text{S,in}} u}{x_1} - \left(\frac{u}{x_1} - K_{\text{vap}} \right) x_3, \\
\dot{x}_4 &= \pi x_2 - K_{\text{H}} x_4 - \left(\frac{u}{x_1} - K_{\text{vap}} \right) x_4.
\end{aligned} \tag{21}$$

Here, states $x = [x_1, x_2, x_3, x_4]^{\text{T}}$ correspond to volume (l), biomass concentration (g l^{-1}), essential nutrient (glucose) concentration (g l^{-1}) and penicillin concentration (g l^{-1}) while input u (l h^{-1}) represents the feed-flow rate.

Biomass (its concentration is represented by the second state variable x_2) can be regarded as the "driving engine" of the cultivation—it consumes essential nutrient (whose concentration corresponds to the third state variable x_3) and thanks to this "fuel", it ensures its own reproduction at growth rate μ and creates the penicillin (whose concentration is represented by x_4) at specific production rate π. Penicillin cultivation is a typical secondary metabolism example with the Contois formula describing μ and Haldane kinetics modeling π. In this case, μ considers a saturation of growth with respect to the substrate and a growth decrease related to the cells accumulation while π is activated by law substrate level, reaches its maximum at a precise level and is inhibited by the substrate itself beyond this level. Mathematically, μ

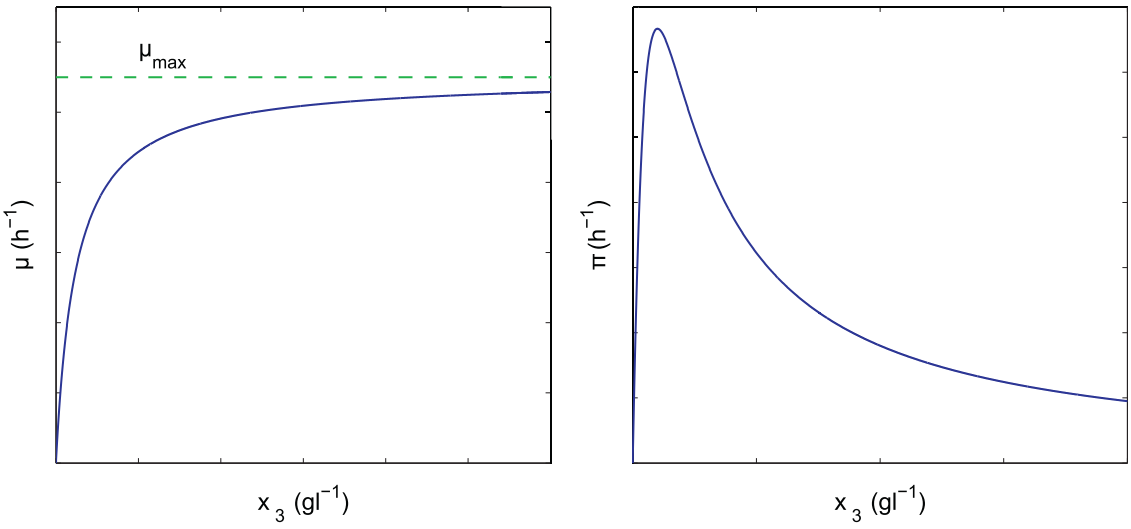

Fig. 2. Growth and production rate profiles.

Table 2
Model parameters.

Parameter	Value	Parameter	Value
$\mu_{\max}$ (h^{-1})	0.11	$Y_{P/S}$ $(-)$	1.2
$\pi_{\max}$ (h^{-1})	0.004	$C_{S,in}$ $(g\,l^{-1})$	500
K_P $(g\,l^{-1})$	0.1	K_{vap} (h^{-1})	6.23×10^{-4}
$Y_{X/S}$ $(-)$	0.47	K_I $(g\,l^{-1})$	0.1
K_D (h^{-1})	0.0136	K_X $(-)$	0.06
K_H (h^{-1})	0.01		

and π are expressed as follows:

$$\mu = \mu_{\max}\frac{x_3}{K_X x_2 + x_3},$$
$$\pi = \pi_{\max}\frac{x_3}{K_P + x_3 + x_3^2/K_I}. \tag{22}$$

Graphical representation of both formulas is given in Fig. 2.

The biomass mortality is expressed by the constant death rate K_D at which the amount of biomass decreases. Usual way of complementing the consumed nutrient is pouring the feed with nutrient concentration $C_{S,in}$ into the tank while the feed flow rate u is then the manipulated variable of this control task. Every control action increases the volume level x_1 which naturally decreases due to the vaporization described by the specific vaporization constant K_{vap}. Moreover, penicillin hydrolysis caused by the fact that the penicillin is not stable in the liquid environment is modeled by its hydrolysis rate K_H. Model (21) is further exploited as both the optimization model and the simulation test-bed for the results evaluation.

Values of the system parameters can be found in Table 2. Interested readers looking for a more detailed description are referred to [12,22].

Table 3
Optimization constraints.

Parameter	Value
$x_{0,\min}$ ([l, g l^{-1}, g l^{-1}]$^{\mathrm{T}}$)	$[4, 0.05, 0.05]^{\mathrm{T}}$
$x_{0,\max}$ ([l, g l^{-1}, g l^{-1}]$^{\mathrm{T}}$)	$[14, 10, 20]^{\mathrm{T}}$
LPS_1 (l)	0.01
LPS_2 (g l^{-1})	0.05
LPS_3 (g l^{-1})	0.05
$u_{\min}$ (l h^{-1})	0
$u_{\max}$ (l h^{-1})	0.05

4.2. Optimization task

Usual control goal for penicillin production is to obtain a highly concentrated product at the end of the cultivation [12,41]. This corresponds to maximization of the terminal penicillin concentration, which is mathematically expressed by the following minimization criterion:

$$\mathcal{J} = -x_4(T_{\mathrm{F}}), \tag{23}$$

where T_{F} is the cultivation period. Moreover, the solution must satisfy the following constraints:

$$\begin{aligned} \dot{x} &= f(x,u), \quad x(0) = x_0, \\ x_{0,\min} &\le x_0^{\{1,2,3\}} \le x_{0,\max}, \\ x_0 &\in \{[p \times LPS_1, r \times LPS_2, s \times LPS_3, 0]^{\mathrm{T}} | p, r, s \in \mathbb{N}^0\}, \\ u_{\min} &\le u \le u_{\max}. \end{aligned} \tag{24}$$

All constraint values are provided in Table 3. Moreover, it can be shown that the following holds for every state variable x_i:

$$x_i(t) = 0 \ \wedge \ u(t) \ge 0 \implies \frac{\mathrm{d}x_i}{\mathrm{d}t} \ge 0.$$

Thus, the nonnegativity of the state variables requested by the physical meaningfulness of the resulting system is satisfied by the input/initial state constraints and the mathematical model itself and there is no further need to pay attention to it.

4.3. Results

To demonstrate the effectiveness of the results obtained by the JIPICO algorithm, nominal cultivation period $T_F = 400$ h was chosen. The settings used for JIPICO stage I are listed in Table 4. Sampled Hamiltonian-based projected gradient method was used to obtain T_{samp}-sampled optimized input profile $\overline{\mathbf{u}}$ which served as the starting point for JIPICO stage II. Besides the input profile from the previous stage, a set of basis functions $\mathfrak{F} = \mathfrak{F}_S \bigcup \mathfrak{F}_A$ needs to be provided to solve the re-parametrization optimization problem (11). The chosen functions $\mathfrak{F}_S$ and $\mathfrak{F}_A$ are presented in Table 4. Here, $\underline{1}(t - \cdot)$ denotes the Heaviside step function. Regarding the performance degradation threshold, relative performance degradation $|\Delta\mathcal{J}|/|\mathcal{J}(\overline{\mathbf{u}})| = 1\%$ was chosen. Although such choice might seem too conservative and restrictive, it should be realized that the higher the allowed deterioration is, the more parameters are eliminated and the less parameters are available for optimization in the JIPICO stage III.

Table 4
JIPICO algorithm: settings.

JIPICO stage	Parameter	Value
I	$x_{0,0}$ ([l, g l^{-1}, g l^{-1}, g l^{-1}]$^{\mathrm{T}}$)	$[7, 1.5, 6, 0]^{\mathrm{T}}$
	T_{samp} (h)	4
	T_F (h)	400
	P (−)	$T_F/T_{\mathrm{samp}} = 100$
	u_0	$\mathbf{0}_{P\times 1}$
II	$\mathfrak{F}_S$	$\{t^0, t^1, t^2, t^3\}$
	$\mathfrak{F}_A$	$\{\underline{1}(t-\cdot)\}$
	$\lvert\Delta\mathcal{J}\rvert/\lvert\mathcal{J}(\overline{\mathbf{u}})\rvert$	1%

Table 5
Influence $\mathcal{I}(p_i)$ and relative performance deterioration $|\tilde{\mathcal{J}}(p_i)|/|\mathcal{J}(\overline{\mathbf{u}})|$.

Parameter	$\mathbf{P}_T$		$\mathbf{P}_S$			$\mathbf{P}_A$
	T_1	T_2	a	c	d	b
$\lvert\tilde{\mathcal{J}}(p_i)\rvert/\lvert\mathcal{J}(\overline{\mathbf{u}})\rvert$ (%)	30.9	–	25.7	1.7	3.6	25.7
$\mathcal{I}(p_i)$ (−)	3	–	2	1	1	1

To illustrate the performance of the JIPICO algorithm and its particular stages, let us refer to Fig. 1. During the stage I, the zero-valued optimal input profile estimate is processed and a T_{samp}-sampled input profile with a length of $P = 100$ samples is obtained. The total cardinality of the set of optimizable parameters obtained after the JIPICO stage I is $2P$; P time stamps and P constant input values. During the stage II, the re-parametrization optimization problem (11) is solved with respect to the chosen set of basis functions $\mathfrak{F}$, input profile constraints $u_{\min}$, $u_{\max}$, and the relative performance degradation $|\Delta\mathcal{J}|/|\mathcal{J}(\overline{\mathbf{u}})|$. After convergence of the JIPICO stage II, the re-parametrized input profile is expressed as:

$$u = \begin{cases} u_1 = a\,\underline{1}(t-b) & 0 \le t < T_1, \\ u_2 = c\,t + d & T_1 \le t \le T_2 = T_F, \end{cases} \tag{25}$$

with the following sets of optimizable parameters: $\mathbf{P}_{\mathrm{T}} = \{T_1, T_2\}$, $\mathbf{P}_{\mathrm{S}} = \{a, c, d\}$, $\mathbf{P}_{\mathrm{A}} = b$. The cardinality of the set of all input profile parameters is then $N_{\mathrm{P}} = |\mathbf{P}| = |\mathbf{P}_{\mathrm{T}} \bigcup \mathbf{P}_{\mathrm{S}} \bigcup \mathbf{P}_{\mathrm{A}}| = 6 \le 2P$. To prove that none of the parameters $p_i \in \mathbf{P}$ obtained at the end of JIPICO stage II belongs to the set of *expendable parameters* $\mathbf{P}_{\mathrm{E}}$, relative performance deterioration $|\tilde{\mathcal{J}}(p_i)|/|\mathcal{J}(\overline{\mathbf{u}})|$ caused by exclusion of p_i from $\mathbf{P}$ is presented in Table 5. To complete the overview, influence $\mathcal{I}(p_i)$ of all parameters is also provided. It can be seen that elimination of any of the considered input profile parameters would lead to the relative deterioration of the performance criterion higher than the allowed threshold.

After the re-parametrization, the set of input profile parameters $\mathbf{P}$ is joined together with the set of optimizable state initial conditions and the JIPICO stage III is performed. After convergence, the optimal input profile parameter values (marked with asterisks in Fig. 1) and the optimal state initial conditions are provided as the JIPICO algorithm outputs.

In order to finalize the evaluation of the JIPICO algorithm, Fig. 3 shows the penicillin concentration profiles obtained for various optimization scenarios. Here, HGO stands for Hamiltonian-based gradient optimization (equivalent to stage I optimization), ICO means initial condition optimization and IPRO represents input profile re-parametrization and op-

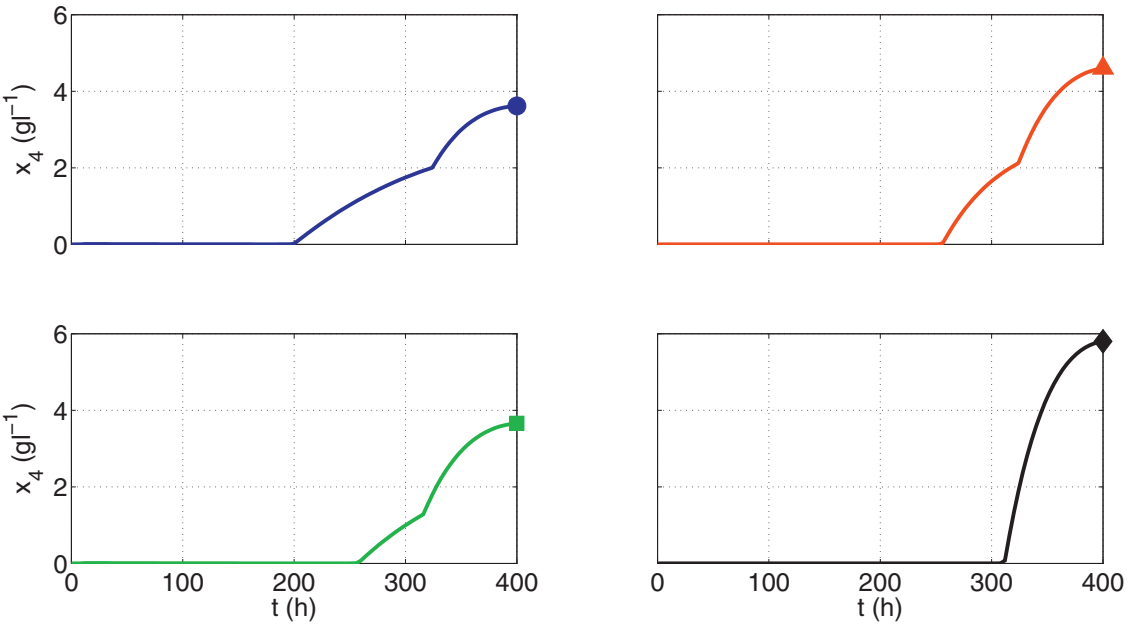

Fig. 3. Penicillin concentration profiles (—●—HGO, —▲—ICO, —■—IPRO, —◆—JIPICO).

Table 6
Optimization strategies (details and comparison).

Strategy	Optimization of x_0	Re-parametrization and optimization of u	$x_4\|_{T_F}$ (g l^{-1})
HGO	✗	✗	3.6
IPRO	✗	✓	3.7
ICO	✓	✗	4.6
JIPICO	✓	✓	5.8

timization. The last two mentioned exploit the input profile provided by the HGO method and, moreover, they perform one of two additional optimizations. More details about the particular optimization approach can be found in Table 6. In Fig. 3, the markers show the value of penicillin concentration at the end of cultivation These values are also listed in Table 6.

From both Fig. 3 and Table 6, it can be observed that while incorporation of any of the "extra" optimizations (either optimization of the state initial conditions or re-parametrization and subsequent optimization of the input profile) improves the value of the penicillin concentration at the end of the cultivation, it is none of them but the ultimate joint optimization performed by JIPICO that makes the most significant difference. The gain of more than 60% is much higher than simple addition of the partial improvements of about 3% (IPRO) and 28% (ICO), respectively, and this clearly demonstrates the meaningfulness and importance of the joint optimization.

For better illustration, x_2 and x_3 profiles are presented in Fig. 4 together with the time profiles of the production rate π and πx_2 which determines the effective rate of increase of the penicillin concentration in the broth.

Although only insignificant amount of penicillin is produced during the first 200–300 h (see Fig. 3), this earlier period is crucial for the terminal penicillin concentration due to the

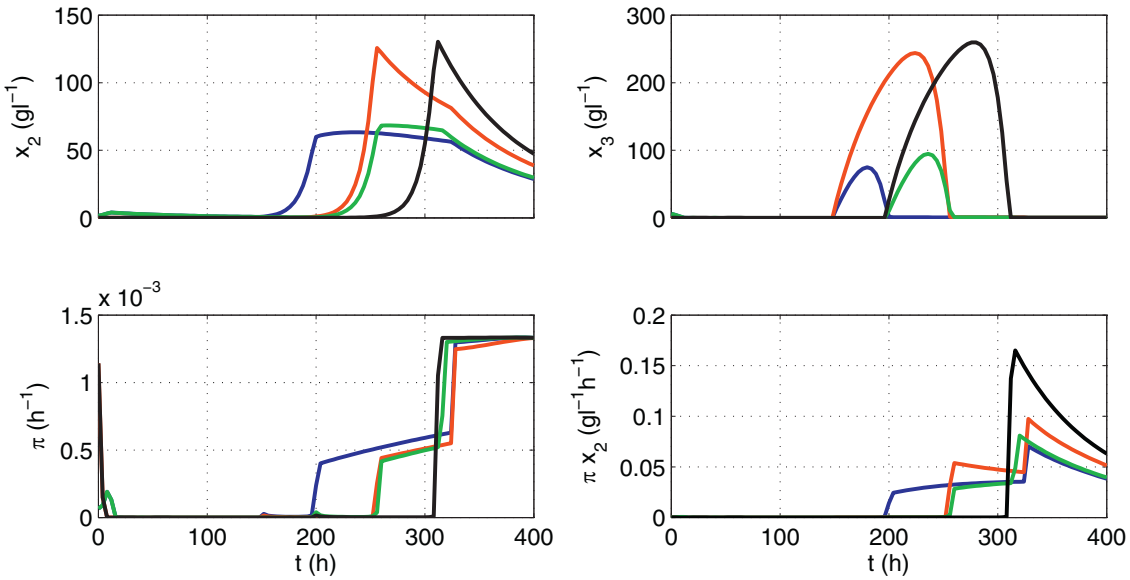

Fig. 4. Suplementary profiles (—HGO, —ICO, —IPRO, —♦—JIPICO).

Table 7
HGO vs. JIPICO.

Strategy	Evaluator	T_F (h)								
		100	160	220	280	340	400	460	520	580
HGO	$x_4\|_{T_F}$ (g l^{-1})	2.24	3.14	3.47	3.50	3.56	3.62	3.70	3.80	3.90
	N_P (–)	25	40	55	70	85	100	115	130	145
JIPICO	$x_4\|_{T_F}$ (g l^{-1})	3.30	4.93	5.08	5.34	5.58	5.80	6.01	6.21	6.39
	N_P (–)	5	5	6	6	6	6	6	6	6

ongoing biomass growth (see the first subfigure of Fig. 4). This period corresponds to the growth phase while the latter one represents the production phase (let us remind that the penicillin cultivation is a secondary metabolism example with distinct growth and production phase). Recalling the model (21), the first and the most significant term of the right-hand side of the $\dot{x}_4$-equation is a product of the production rate π and the biomass concentration x_2; this product is plotted in the last subfigure of Fig. 4. Inspecting Figs. 3 and 4, it can be seen that the newly proposed JIPICO algorithm ensures the highest biomass concentration which together with optimal π timing leads to sovereignly highest πx_2 value and yields the highest terminal penicillin concentration.

To support the credibility of the previously presented evaluation, a wider range of the cultivation periods $\mathbf{T_F} = \{100, 160, 220, 280, 340, 400, 460, 520, 580\}$ was examined. For each $T_F \in \mathbf{T_F}$, both the HGO and the JIPICO optimization were performed. As evaluators, the value of penicillin concentration at the end of cultivation $x_4|_{T_F}$ and N_P were inspected. The achieved results are listed in Table 7 and graphically presented in Figs. 5 and 6.

As can be expected, the penicillin concentration obtained at the end of the cultivation period increases with prolonging of the cultivation period—this holds for both the HGO and JIPICO algorithm. It can also be seen that for all of the inspected cultivation periods $T_F \in \mathbf{T_F}$, JIPICO algorithm provides *significantly* better terminal penicillin concentrations with improvement

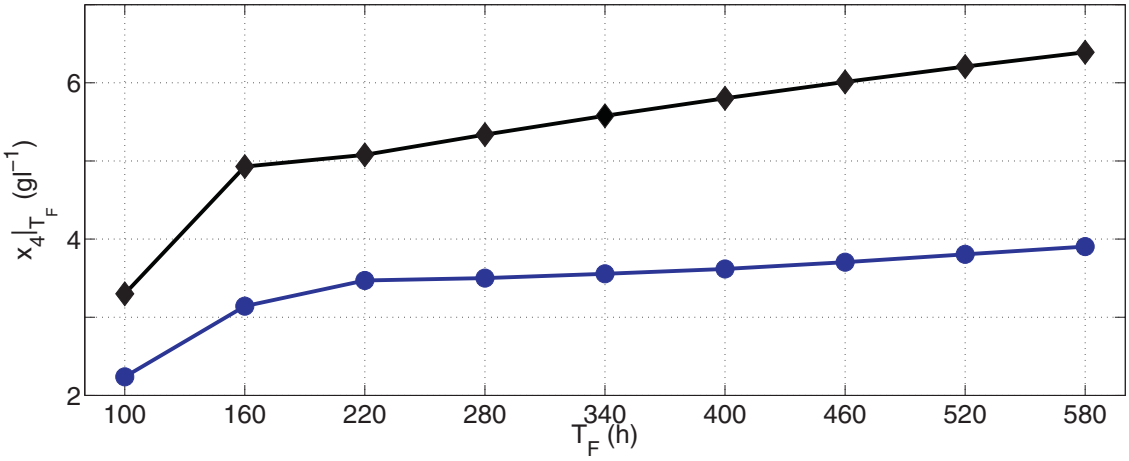

Fig. 5. Terminal penicillin concentration (—●—HGO, —♦—JIPICO).

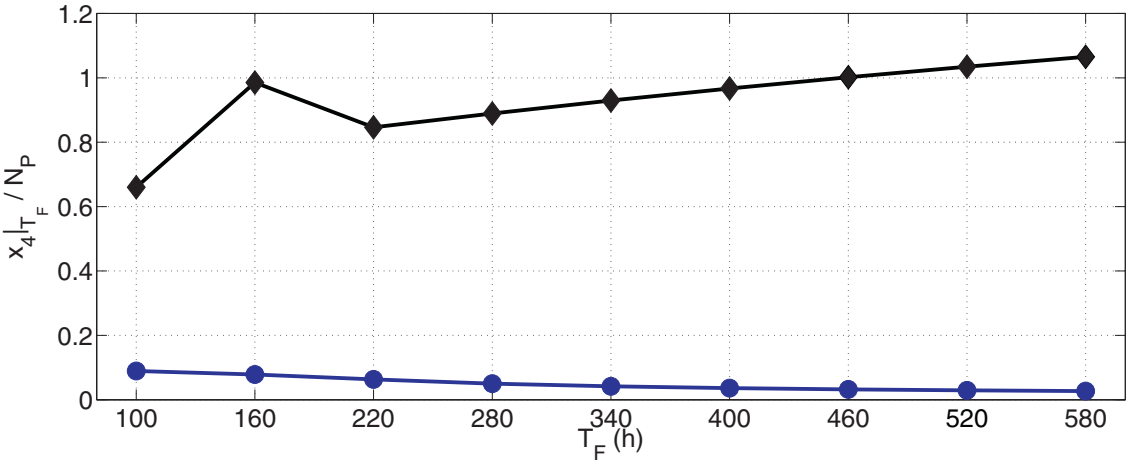

Fig. 6. Relative terminal penicillin concentration with respect to the number of optimized parameters (—●—HGO, —♦—JIPICO).

ranging from more than 46 up to almost 64%, which clearly validates the results obtained for $T_F = 400$ h presented earlier. Furthermore, it can be deduced that JIPICO algorithm is more effective in optimizing the penicillin concentration since even for the second shortest cultivation period, $T_F = 160$ h, it ensures much higher penicillin concentration than is reached with HGO algorithm for the longest inspected cultivation period, $T_F = 580$ h.

Fig. 6 provides a comparison of the "computational resource efficiency" of either of the examined approaches. Here, it should be realized that the number of input profile parameters N_P corresponds to the memory demands required to store the solution of the optimization problem (1) and is also related to the computational complexity of the problem (1). For the optimization of T_{samp}-sampled input profile performed by HGO, this ratio is very low and keeps decreasing with increase of the cultivation period—this is caused by the fact that although the terminal penicillin concentration grows for longer cultivation period, its growth is relatively small and is outweighed by the need for more optimized parameters. On the other hand, the JIPICO algorithm requires much less optimized input profile parameters and

moreover, N_P is kept constant for cultivation periods from 220 to 580 h, which in combination with solid growth of terminal penicillin concentration results in "computational resource efficiency" that is much higher and fairly increasing toward longer cultivation periods. This comparison also shows that not only the HGO algorithm is outperformed in the terms of "computational effectiveness" by JIPICO being its counterpart, but the difference between the two algorithms gets even more considerable toward the higher cultivation periods.

Based on the above presented evaluations, it can be concluded that the JIPICO algorithm provides substantial improvement in both the optimization criterion value and complexity of the optimized input profile.

5. Conclusion

In this paper, a novel three stage JIPICO algorithm reducing the complexity of the optimal input profile and performing joint optimization of the input profile parameters and state initial conditions was proposed and verified.

The comparison with the optimization period $T_F = 400$ h shows that the JIPICO algorithm outperforms all the co-evaluated control strategies. Moreover, the percentage increase of the terminal penicillin concentration introduced by the JIPICO algorithm (normalized with respect to the ordinary used HGO optimization algorithm) is much higher than increase brought by the two other strategies performing some kind of additional optimization (more than 60% increase vs. 3 and 28%, respectively). For a broader range of the inspected optimization periods, the improvement in the cost criterion of the JIPICO algorithm ranges from 46 up to nearly 64% compared with the baseline HGO algorithm. This improvement is achieved with much less input-profile parameters whose number is kept basically constant even with increase that cultivation/optimization period (5 or 6 parameters of the JIPICO profile vs. 25 up to 145 parameters of the HGO profile). Based on the achieved results, the newly proposed JIPICO algorithm can be considered as a promising candidate for the optimization algorithm with use in the process control area.

Acknowledgment

This research has been supported by the Czech Science Foundation through the grant no. 17-04682S.

References

[1] R. Mahadevan, S.K. Agrawal, et al., Differential flatness based nonlinear predictive control of fed-batch bioreactors, Control Eng. Pract. 9 (8) (2001) 889–899.
[2] L.O. Santos, L. Dewasme, D. Coutinho, A.V. Wouwer, Nonlinear model predictive control of fed-batch cultures of micro-organisms exhibiting overflow metabolism: assessment and robustness, Comput. Chem. Eng. 39 (2012) 143–151.
[3] W. Wu, S.-Y. Lai, M.-F. Jang, Y.-S. Chou, Optimal adaptive control schemes for PHB production in fed-batch fermentation of Ralstonia eutropha, J. Process Control 23 (8) (2013) 1159–1168.
[4] M. Abouzlam, R. Ouvrard, D. Mehdi, F. Pontlevoy, B. Gombert, N. Karpel Vel Leitner, S. Boukari, An optimal control of a wastewater treatment reactor by catalytic ozonation, Control Eng. Pract. 21 (1) (2013) 105–112.
[5] S. Craven, J. Whelan, B. Glennon, Glucose concentration control of a fed-batch mammalian cell bioprocess using a nonlinear model predictive controller, J. Process Control 24 (4) (2014) 344–357.
[6] B. Bao, H. Yin, E. Feng, Computation of impulsive optimal control for 1,3-pd fed-batch culture, J. Process Control 34 (2015) 49–55.

[7] L. Wang, Q. Lin, R. Loxton, K.L. Teo, G. Cheng, Optimal 1,3-propanediol production: exploring the trade-off between process yield and feeding rate variation, J. Process Control 32 (2015) 1–9.
[8] G. de Andrade, M. Berenguel, J. Guzmán, D. Pagano, F. Acién, Optimization of biomass production in outdoor tubular photobioreactors, J. Process Control 37 (2016) 58–69.
[9] Y. Fan, C. Li, J.-J. Lay, H. Hou, G. Zhang, Optimization of initial substrate and pH levels for germination of sporing hydrogen-producing anaerobes in cow dung compost, Bioresour. Technol. 91 (2) (2004) 189–193.
[10] G.-l. Cao, N.-q. Ren, A.-j. Wang, W.-q. Guo, J. Yao, Y.-j. Feng, Q.-l. Zhao, Statistical optimization of culture condition for enhanced hydrogen production by Thermoanaerobacterium thermosaccharolyticum W16, Bioresour. Technol. 101 (6) (2010) 2053–2058.
[11] S.S. Veeravalli, S.R. Chaganti, J.A. Lalman, D.D. Heath, Optimizing hydrogen production from a switchgrass steam exploded liquor using a mixed anaerobic culture in an upflow anaerobic sludge blanket reactor, Int. J. Hydrogen Energy 39 (7) (2014) 3160–3175.
[12] A. Ashoori, B. Moshiri, A. Khaki-Sedigh, M. Bakhtiari, Optimal control of a nonlinear fed-batch fermentation process using model predictive approach, J. Process Control 19 (7) (2009) 1162–1173.
[13] J.D. le Roux, R. Padhi, I.K. Craig, Optimal control of grinding mill circuit using model predictive static programming: a new nonlinear MPC paradigm, J. Process Control 24 (12) (2014) 29–40.
[14] J. Peralez, P. Tona, M. Nadri, P. Dufour, A. Sciarretta, Optimal control for an organic rankine cycle on board a diesel-electric railcar, J. Process Control 33 (2015) 1–13.
[15] C. Liu, Z. Gong, E. Feng, H. Yin, Modelling and optimal control for nonlinear multistage dynamical system of microbial fed-batch culture, J. Ind. Manage. Optim. 5 (4) (2009) 835–850.
[16] C. Liu, Optimal control for nonlinear dynamical system of microbial fed-batch culture, J. Comput. Appl. Math. 232 (2) (2009) 252–261.
[17] Z. Gong, C. Liu, E. Feng, L. Wang, Y. Yu, Modelling and optimization for a switched system in microbial fed-batch culture, Appl. Math. Modell. 35 (7) (2011) 3276–3284.
[18] C. Liu, Z. Gong, Modelling and optimal control of a time-delayed switched system in fed-batch process, J. Franklin Inst. 351 (2) (2014) 840–856.
[19] C. Liu, Z. Gong, E. Feng, H. Yin, Optimal switching control of a fed-batch fermentation process, J. Global Optim. 52 (2) (2012) 265–280.
[20] C. Liu, Z. Gong, B. Shen, E. Feng, Modelling and optimal control for a fed-batch fermentation process, Appl. Math. Modell. 37 (3) (2013) 695–706.
[21] J. Richalet, Industrial applications of model based predictive control, Automatica 29 (5) (1993) 1251–1274.
[22] M. Pčolka, S. Čelikovský, Gradient method optimization of penicillin production, in: 24th Chinese Control and Decision Conference (CCDC), IEEE, 2012, pp. 74–79.
[23] A. Bryson, Y. Ho, Applied Optimal Control, Blaisdell, New York, 1969.
[24] K.L. Teo, C. Goh, K. Wong, A Unified Computational Approach to Optimal Control Problems, Longman Scientific & Technical, Essex, 1991.
[25] M. Pčolka, S. Čelikovský, On nonlinear continuous-time optimal control of penicillin cultivation, in: European Control Conference (ECC), EUCA, 2013, pp. 4442–4447.
[26] H.B. Keller, Numerical Methods for Two-Point Boundary-Value Problems, Dover Publications, New York, NY, 1992.
[27] M. Pčolka, S. Čelikovský, Enhancement of practical applicability of optimal control of a nonlinear process, in: American Control Conference (ACC), 2013, pp. 5000–5005.
[28] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2009.
[29] A. Ben-Tal, A. Nemirovski, Robust convex optimization, Math. Oper. Res. 23 (4) (1998) 769–805.
[30] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, vol. 87, Springer, 2004.
[31] D. Tjøstheim, B.H. Auestad, Nonparametric identification of nonlinear time series: projections, J. Am. Stat. Assoc. 89 (428) (1994) 1398–1409.
[32] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, vol. 18, Wiley, New York, 1988.
[33] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Courier Dover Publications, 1998.
[34] M. Grotschel, L. Lovász, Combinatorial optimization, Handbook of combinatorics, vol. 2, 1995, pp. 1541–1597.
[35] J. Lofberg, Yalmip: a toolbox for modeling and optimization in matlab, in: 2004 IEEE International Symposium on Computer Aided Control Systems Design, IEEE, 2004, pp. 284–289.
[36] R.A. Waltz, J. Nocedal, Knitro User's Manual, Technical Report OTC-2003/5, Northwestern University, Evanston, Illinois, 2003.
[37] T. Achterberg, SCIP: solving constraint integer programs, Math. Prog. Comput. 1 (1) (2009) 1–41.

[38] Z. Gu, E. Rothberg, R. Bixby, Gurobi Optimizer Reference Manual, Version 5.0, Gurobi Optimization Inc., Houston, USA, 2012.
[39] C.D. Boor, A Practical Guide to Splines, Springer Verlag, 1978.
[40] D. Dochain, Automatic Control of Bioprocesses, John Wiley & Sons, 2013.
[41] J. Mandur, H. Budman, Robust optimization of chemical processes using Bayesian description of parametric uncertainty, J. Process Control 24 (2) (2014) 422–430.

Chapter 7

Conclusions

In this thesis, numerical aspects of optimal control design for nonlinear systems have been tackled. The conducted research was inspired by real-life issues and questions accompanying deployment of nonlinear model predictive controllers and could be divided into the following parts (each of them addressed in a separate chapter): choice of optimization step for underlying local optimization routine, choice of optimization horizon, effects of various discontinuities and corresponding robustification of the controller against them and comprehensive optimization of production processes.

7.1 Summary

The main contributions of the thesis can be summarized as follows:

- Size of the optimization step used by the local optimization routine has been studied. Different variants of the search step size choice have been inspected with a special emphasize on their influence on the convergence speed and the achieved value of the optimization criterion. Several adaptive search step size expressions have been provided—a formula exploiting approximation of the second difference of an optimal control task Hamiltonian, a formula based on logarithm of the iteration-difference of the cost criterion and a formula exploiting sigmoid function of the distance from the imposed constraints. All of these expressions have been successfully validated on various nonlinear programming task examples.
- Feedback introduction and the corresponding optimization horizon choice has been investigated. To overcome disadvantages of the receding horizon principle, a control

scheme employing *shrinking horizon* has been introduced and the performance of these two approaches has been evaluated for an example task of penicillin production maximization. Unlike the receding horizon principle which has been demonstrated to lack robustness against decrease of number of optimizable input samples and, in some cases, has paradoxically yielded better results for more restricted number of allowed optimization routine iterations, the shrinking horizon version has been almost immune against such limitations and also in nominal case has significantly outperformed its commonly-used counterpart. Bridging these two, a hybrid approach has been provided as well.

- Two algorithms considering the optimization horizon as a member of the optimizable variables set have been presented. The first of them manipulates the horizon length directly according to the numerical gradient and then adjusts the sampling period trying to preserve equal duration of all input samples. The second one makes use of an auxiliary input variable which enables to optimize time scaling of particular input samples separately and, in turn, also the length of the whole optimization horizon. As a test bed, maximization of time efficiency of penicillin production has been chosen. It has been revealed that prolonging the batch duration by a factor of 4.5 results in only 2.5-times higher yields. Efficiency evaluation performed with a longer operational period has shown that the horizon-optimizing algorithms are capable of providing 80 % higher efficiency compared with a traditional fixed-horizon algorithm.

- Ways and means of reducing the computational and memory burden of a predictive controller for a racing car exploiting an idea of adaptive prediction horizon have been explored. Three mathematical rules calculating the length of adaptive prediction horizon from the currently measured velocity have been proposed: one linear formulation and two logarithmic formulations making use of the information about the system dynamics and constraints. For the logarithmic expressions, certain guarantees of maneuverability preservation and satisfaction of safety requirements—which can be regarded as a measure of closed-loop system stability—have been derived. All three alternatives have been evaluated on a set of numerical experiments and it has been shown that all of them have the potential to outperform the standard constant-horizon approach. For the latter two, moreover, Pareto-optimality has been demonstrated as well.

- Motivated by the frequent use of actuators with finite input signal resolution, optimization of quantized input has been investigated. An innovative optimization algorithm relying on a regularly performed *mid-processing* iteration including projection of the input sequence on the admissible finite set of input levels and filtering the sequence extended with the previously applied input samples with a low-pass filter has been developed. Thanks to incorporating the input quantization information into the optimization routine and performing mid-processing filtering, an attractive control performance has been attained and the need for computationally expensive mixed-integer nonlinear programming techniques has been eliminated. An evaluation considering quantized input control of both fast- and slow-dynamics system has been performed and as an alternative to the proposed algorithm, a posteriori quantization has been considered. The following results have been observed: median of the control performance degradation has decreased by a factor of almost 2 for the slow dynamics system and more than 13 for the fast dynamics system and the input signal oscillations have been massively suppressed.

- A part of this thesis has addressed optimal control tasks considering discontinuous system dynamics and/or discontinuous optimization criterion and a *hamiltonian-switcher*-based algorithm (HaSH-NPC) tackling them has been presented. Its functionality has been validated on an example of lap-time minimization with race car with hybrid dynamics and non-smooth criterion and compared with the approximation-based a priori switching nonlinear predictive controller. A comparison of fixed-horizon variants of these two has demonstrated that even with shorter horizon, HaSH-NPC algorithm outperforms the approximation-based solution. Confronting HaSH-NPC with mixed-integer nonlinear programming solver, it has been manifested that to achieve comparable performance, the latter one has consumed enormously more computational time. The robustness of the HaSH-NPC algorithm has been proven by the means of a detailed sensitivity analysis.

- Last but not least, the thesis has dealt with optimization of production processes—especially bioprocesses—as a whole and a novel three-stage algorithm has been developed with *i)* stage I serving as a starting point of the overall scheme consisting in sampled-input optimization with fixed initial conditions, *ii)* stage II performing a reparameterization of the obtained input profile reducing the complexity of the subsequent optimization; and *iii)* joint parametric optimization of both the process

initial conditions *and* the input profile parameters. A comparison with three other algorithms has unveiled that although partial improvement could be achieved performing either optimization of the reparameterized input profile *or* initial conditions optimization, their combination representing the proposed JIPICO algorithm might yield as much as 60 % improvement compared with the original optimization with fixed initial conditions and sampled input. Moreover, the number of optimizable parameters compared with the sampled-input optimization has been substantially reduced suggesting also considerable savings from the computational point of view.

7.2 Future work

While research itself might never be completely finished, a thesis *must* be and thus, much has been left to be done yet. For those willing to continue this journey and grow the planted sprouts, the following suggestions for future work can be given:

- **Adaptive search step formulas.** Performance of the provided adaptive search step expressions—namely the logarithm-based formula and the formula exploiting sigmoid function of constraints—could be studied in more depth. In particular, the proposed ideas should be validated and examined with a larger and richer ensemble of optimization tasks taken into account and special attention should be paid to their theoretical properties.

- **Enhanced input profile reparameterization.** Recalling stage II of JIPICO introduced in Chapter 6 of this thesis, the algorithm reparameterizing the input profile and expressing it as a parametric function of time could be enhanced. This thesis has provided a rigorous optimization-based reparameterization procedure, however, the set of the basis functions has been considered to be given in advance. Although this is not a restrictive assumption from practical point of view, certain improvement could be achieved by exploiting nonparametric regression method—in particular, the length of stage II of the production-process optimization algorithm could be decreased. This step could also contribute to automation and—most of all—considerable *autonomization* of the whole procedure making it less dependent upon the user entries.

- **Adaptive input profile reparameterization.** The reparameterization of the input profile (stage II of the JIPICO algorithm) could be made adaptive and recursive.

The algorithm provided in this thesis assumes that the input profile is reparameterized *prior* to bringing the controller into action and over its whole remaining lifetime, the same parameterization is used. This could change; the initial reparameterization should be used only for a limited amount of time and once enough samples of the closed-loop input profile have been stored, machine learning algorithms could be employed to recognize the emerging patterns and modify the controller itself by updating the input profile parameterization. A new stage II-a could be introduced; this stage would be performed online and its mission would be extracting the information from the past input samples. In this way, the input profile could be reparameterized in an *adaptive* manner and both low dimensionality and the appropriate parameterization of the input profile could be maintained.

- **Online parametric optimization of a virtual feed-forward filter response.** Majority of the current approaches consider the input profile samples to be more or less independent optimization variables which leads to increase of complexity when either the prediction horizon is increased or the sampling period is decreased. This can be remedied by the so-called move blocking strategy which, however, causes a loss of control authority and optimization freedom. As an alternative, the input to the controlled system could be generated as a step/impulse response of a simple system and only parameters of this system would be optimized. This would considerably decrease the number of optimized variables and make the optimal control almost independent of prediction horizon or sampling period changes while eliminating too severe restriction of the control authority.

- **Enhancement of quantized input optimization.** In the thesis, the mid-processing iteration representing "the heart" of the proposed quantized input optimization algorithm has been said to be performed every M iterations with M being defined by the user. More rigorous rules for the frequency of performing the mid-processing iteration based on the distances between the quantization levels should be derived. Additionally, the extended input sequence has been considered to be filtered with a suitable filter and as a result, the high-frequency portion of the signal has been eliminated, indeed. However, the already applied input samples cannot be changed anymore and therefore, certain mismatch between the frequency content of the sequence internally calculated by the optimizer and of that finally applied to the

system might arise. This should be studied in more detail and a suitable procedure eliminating this mismatch should be developed.

- **Use of global optimization techniques.** For a significant set of the optimal control tasks, replacing the local optimization techniques with those focusing on finding a global solution could bring significant control performance improvement. In general, looser formulations of the optimization task unfold wider room for (global) optimization and therefore, control problems including zone control or maximization of certain quantity at a given point of time might greatly benefit from use of the corresponding optimization tools. Here, the potential computational and memory demands associated with large-scale tasks might be terrifying and therefore, certain reduction of the parameter space is desirable. Linking these ideas to this thesis, a promising candidate for enhancement using global optimization is the optimization of the reparameterized input profile and process initial conditions aiming at maximization of the terminal-time yield (stage III of the JIPICO algorithm).

Bibliography

[Abdollahi and Dubljevic, 2012] Abdollahi, J. and Dubljevic, S. (2012). Lipid production optimization and optimal control of heterotrophic microalgae fed-batch bioreactor. *Chemical engineering science*, 84:619–627.

[Aguilera and Quevedo, 2013] Aguilera, R. P. and Quevedo, D. E. (2013). Stability analysis of quadratic mpc with a discrete input alphabet. *IEEE Transactions on Automatic Control*, 58(12):3190–3196.

[Alexis et al., 2011] Alexis, K., Nikolakopoulos, G., and Tzes, A. (2011). Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice*, 19(10):1195–1207.

[Alexis et al., 2012] Alexis, K., Nikolakopoulos, G., and Tzes, A. (2012). Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory & Applications*, 6(12):1812–1827.

[Allgöwer and Zheng, 2012] Allgöwer, F. and Zheng, A. (2012). *Nonlinear model predictive control*, volume 26. Birkhäuser.

[Anilkumar et al., 2018] Anilkumar, M., Padhiyar, N., and Moudgalya, K. (2018). Multicriterion control of a bioprocess in fed-batch reactor using ekf based economic model predictive control. *Chemical Engineering Research and Design*.

[Ashoori et al., 2009] Ashoori, A., Moshiri, B., Khaki-Sedigh, A., and Bakhtiari, M. R. (2009). Optimal control of a nonlinear fed-batch fermentation process using model predictive approach. *Journal of Process Control*, 19(7):1162–1173.

[Bazaraa et al., 2013] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (2013). *Nonlinear programming: theory and algorithms*. John Wiley & Sons.

[Bemporad, 2003] Bemporad, A. (2003). Multiparametric nonlinear integer programming and explicit quantized optimal control. In *Proc. of the 42nd IEEE Conference on Decision and Control*, pages 3167–3172.

[Bemporad et al., 2002] Bemporad, A., Borrelli, F., and Morari, M. (2002). Model predictive control based on linear programming˜ the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985.

[Bengea and DeCarlo, 2005] Bengea, S. C. and DeCarlo, R. A. (2005). Optimal control of switching systems. *automatica*, 41(1):11–27.

[Bertsekas, 1999] Bertsekas, D. P. (1999). *Nonlinear programming*. Athena scientific Belmont.

[Camacho and Bordons, 2004] Camacho, E. F. and Bordons, C. (2004). *Model predictive control*, volume 2. Springer London.

[Camacho et al., 2010] Camacho, E. F., Ramírez, D. R., Limón, D., De La Peña, D. M., and Alamo, T. (2010). Model predictive control techniques for hybrid systems. *Annual reviews in control*, 34(1):21–31.

[Chakrabarty et al., 2017] Chakrabarty, A., Buzzard, G. T., and Żak, S. H. (2017). Output-tracking quantized explicit nonlinear model predictive control using multiclass support vector machines. *IEEE Transactions on Industrial Electronics*, 64(5):4130–4138.

[Chang et al., 2016] Chang, L., Liu, X., and Henson, M. A. (2016). Nonlinear model predictive control of fed-batch fermentations using dynamic flux balance models. *Journal of Process Control*, 42:137–149.

[Clarke et al., 1987a] Clarke, D. W., Mohtadi, C., and Tuffs, P. S. (1987a). Generalized predictive control—part i. the basic algorithm. *Automatica*, 23(2):137–148.

[Clarke et al., 1987b] Clarke, D. W., Mohtadi, C., and Tuffs, P. S. (1987b). Generalized predictive control—part ii extensions and interpretations. *Automatica*, 23(2):149–160.

[Cortes et al., 2012] Cortes, P., Rodriguez, J., Silva, C., and Flores, A. (2012). Delay compensation in model predictive current control of a three-phase inverter. *IEEE Transactions on Industrial Electronics*, 59(2):1323–1325.

[Cougnon et al., 2011] Cougnon, P., Dochain, D., Guay, M., and Perrier, M. (2011). Online optimization of fedbatch bioreactors by adaptive extremum seeking control. *Journal of Process Control*, 21(10):1526–1532.

[Diehl et al., 2009] Diehl, M., Ferreau, H. J., and Haverbeke, N. (2009). Efficient numerical methods for nonlinear mpc and moving horizon estimation. In *Nonlinear Model Predictive Control*, pages 391–417. Springer.

[Ghouali et al., 2015] Ghouali, A., Sari, T., and Harmand, J. (2015). Maximizing biogas production from the anaerobic digestion. *Journal of Process Control*, 36:79–88.

[Giua et al., 2001] Giua, A., Seatzu, C., and Van Der Mee, C. (2001). Optimal control of switched autonomous linear systems. In *IEEE conference on decision and control*, volume 3, pages 2472–2477. IEEE; 1998.

[Gong et al., 2011] Gong, Z., Liu, C., Feng, E., Wang, L., and Yu, Y. (2011). Modelling and optimization for a switched system in microbial fed-batch culture. *Applied Mathematical Modelling*, 35(7):3276–3284.

[Grancharova and Johansen, 2009] Grancharova, A. and Johansen, T. A. (2009). Explicit approximate model predictive control of constrained nonlinear systems with quantized input. In *Nonlinear Model Predictive Control*, pages 371–380. Springer.

[Grancharova and Johansen, 2012] Grancharova, A. and Johansen, T. A. (2012). Explicit mpc of constrained nonlinear systems with quantized inputs. In *Explicit Nonlinear Model Predictive Control*, pages 111–125. Springer.

[Himmelblau, 1972] Himmelblau, D. M. (1972). *Applied nonlinear programming*. McGraw-Hill Companies.

[Houska et al., 2011] Houska, B., Ferreau, H. J., and Diehl, M. (2011). An auto-generated real-time iteration algorithm for nonlinear mpc in the microsecond range. *Automatica*, 47(10):2279–2285.

[Jeong et al., 2010] Jeong, J.-H., Kim, J.-N., Wee, Y.-J., and Ryu, H.-W. (2010). The statistically optimized production of poly (γ-glutamic acid) by batch fermentation of a newly isolated bacillus subtilis rky3. *Bioresource technology*, 101(12):4533–4539.

[Karthikeyan et al., 1996] Karthikeyan, R., Rakshit, S., and Baradarajan, A. (1996). Optimization of batch fermentation conditions for dextran production. *Bioprocess Engineering*, 15(5):247–251.

[Kim et al., 2012] Kim, W., Park, J. M., Gim, G. H., Jeong, S.-H., Kang, C. M., Kim, D.-J., and Kim, S. W. (2012). Optimization of culture conditions and comparison of biomass productivity of three green algae. *Bioprocess and biosystems engineering*, 35(1-2):19–27.

[Kuhn and Tucker, 2014] Kuhn, H. W. and Tucker, A. W. (2014). Nonlinear programming. In *Traces and emergence of nonlinear programming*, pages 247–258. Springer.

[Kwon and Han, 2006] Kwon, W. H. and Han, S. H. (2006). *Receding horizon control: model predictive control for state models*. Springer Science & Business Media.

[Liu, 2009] Liu, C. (2009). Optimal control for nonlinear dynamical system of microbial fed-batch culture. *Journal of computational and applied mathematics*, 232(2):252–261.

[Liu and Gong, 2014] Liu, C. and Gong, Z. (2014). Modelling and optimal control of a time-delayed switched system in fed-batch process. *Journal of the Franklin Institute*, 351(2):840–856.

[Liu et al., 2012] Liu, C., Gong, Z., Feng, E., and Yin, H. (2012). Optimal switching control of a fed-batch fermentation process. *Journal of Global Optimization*, 52(2):265–280.

[Liu et al., 2013] Liu, C., Gong, Z., Shen, B., and Feng, E. (2013). Modelling and optimal control for a fed-batch fermentation process. *Applied Mathematical Modelling*, 37(3):695–706.

[Ma et al., 2012] Ma, Y., Borrelli, F., Hencey, B., Coffey, B., Bengea, S., and Haves, P. (2012). Model predictive control for the operation of building cooling systems. *IEEE Transactions on control systems technology*, 20(3):796–803.

[Maciejowski and Huzmezan, 1997] Maciejowski, J. M. and Huzmezan, M. (1997). *Predictive control*. Springer.

[Mattingley and Boyd, 2009] Mattingley, J. and Boyd, S. (2009). Automatic code generation for real-time convex optimization. *Convex optimization in signal processing and communications*, pages 1–41.

[Mattingley et al., 2010] Mattingley, J., Wang, Y., and Boyd, S. (2010). Code generation for receding horizon control. In *Computer-Aided Control System Design (CACSD), 2010 IEEE International Symposium on*, pages 985–992. IEEE.

[Mattingley et al., 2011] Mattingley, J., Wang, Y., and Boyd, S. (2011). Receding horizon control. *IEEE Control Systems*, 31(3):52–65.

[Mayne, 2014] Mayne, D. Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986.

[Mears et al., 2017] Mears, L., Stocks, S. M., Sin, G., and Gernaey, K. V. (2017). A review of control strategies for manipulating the feed rate in fed-batch fermentation processes. *Journal of biotechnology*, 245:34–46.

[Mhaskar et al., 2005] Mhaskar, P., El-Farra, N. H., and Christofides, P. D. (2005). Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Transactions on Automatic Control*, 50(11):1670–1680.

[Morari and H Lee, 1999] Morari, M. and H Lee, J. (1999). Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682.

[Muller et al., 2011] Muller, C., Quevedo, D. E., and Goodwin, G. C. (2011). How good is quantized model predictive control with horizon one? *IEEE Transactions on Automatic Control*, 56(11):2623–2638.

[Muske and Rawlings, 1993] Muske, K. R. and Rawlings, J. B. (1993). Model predictive control with linear models. *AIChE Journal*, 39(2):262–287.

[Picasso et al., 2003] Picasso, B., Pancanti, S., Bemporad, A., and Bicchi, A. (2003). Receding-horizon control of lti systems with quantized inputs. In *Analysis and Design of Hybrid Systems 2003 (ADHS 03): A Proceedings Volume from the IFAC Conference, St. Malo, Brittany, France, 16-18 June 2003*, volume 259. Elsevier Science.

[Qin and Badgwell, 2003] Qin, S. J. and Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764.

[Rawlings and Risbeck, 2017] Rawlings, J. B. and Risbeck, M. J. (2017). Model predictive control with discrete actuators: Theory and application. *Automatica*, 78:258–265.

[Rodriguez et al., 2013] Rodriguez, J., Kazmierkowski, M. P., Espinoza, J. R., Zanchetta, P., Abu-Rub, H., Young, H. A., and Rojas, C. A. (2013). State of the art of finite control set model predictive control in power electronics. *IEEE Transactions on Industrial Informatics*, 9(2):1003–1016.

[Sanfelice, 2013] Sanfelice, R. G. (2013). Control of hybrid dynamical systems: An overview of recent advances. *Proc. Hybrid Syst. Constraints*.

[Santos et al., 2012] Santos, L. O., Dewasme, L., Coutinho, D., and Wouwer, A. V. (2012). Nonlinear model predictive control of fed-batch cultures of micro-organisms exhibiting overflow metabolism: assessment and robustness. *Computers & Chemical Engineering*, 39:143–151.

[Wu et al., 2006] Wu, C., Teo, K. L., Li, R., and Zhao, Y. (2006). Optimal control of switched systems with time delay. *Applied Mathematics Letters*, 19(10):1062–1067.

[Xu et al., 2015] Xu, S., Li, S. E., Deng, K., Li, S., and Cheng, B. (2015). A unified pseudospectral computational framework for optimal control of road vehicles. *IEEE/ASME Transactions on Mechatronics*, 20(4):1499–1510.

[Xu and Antsaklis, 2000] Xu, X. and Antsaklis, P. J. (2000). A dynamic programming approach for optimal control of switched systems. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 2, pages 1822–1827. IEEE.

[Xu and Antsaklis, 2004] Xu, X. and Antsaklis, P. J. (2004). Optimal control of switched systems based on parameterization of the switching instants. *IEEE transactions on automatic control*, 49(1):2–16.

[Yuan, 2008] Yuan, Y.-x. (2008). Step-sizes for the gradient method. *AMS IP Studies in Advanced Mathematics*, 42(2):785.

[Zhou et al., 2006] Zhou, B., Gao, L., and Dai, Y.-H. (2006). Gradient methods with adaptive step-sizes. *Computational Optimization and Applications*, 35(1):69–86.

Publications of the Author

Publications related to the thesis

Articles in journals with impact factor

[A.1] Pčolka, M.; Čelikovský, S. (2014; authorship: 50 %). *Numerical Aspects of Optimal Control of Penicillin Production*. Bioprocess and Biosystems Engineering (2017 impact factor: 2.139). DOI: 10.1007/s00449-013-0929-z.

[A.2] Pčolka, M.; Žáčeková, E.; Robinett, R.; Čelikovský, S.; Šebek, M. (2016; authorship: 50 %). *Bridging the Gap between the Linear and Nonlinear Predictive Control: Adaptations for Efficient Building Climate Control*. Control Engineering Practice (2017 impact factor: 2.616). DOI: 10.1016/j.conengprac.2016.01.007.

[A.3] Pčolka, M.; Žáčeková, E.; Čelikovský, S.; Šebek, M. (2017; authorship: 55 %). *Toward a Smart Car: Hybrid Nonlinear Predictive Controller With Adaptive Horizon*. IEEE Transactions on Control Systems Technology (2017 impact factor: 4.883). DOI: 10.1109/TCST.2017.2747504.

[A.4] Pčolka, M.; Čelikovský, S., (2017; authorship: 60 %). *Production-process Optimization Algorithm: Application to Fed-batch Bioprocess*. Journal of the Franklin Institute (2017 impact factor: 3.576). DOI: 10.1016/j.jfranklin.2017.10.012.

Invited conference talks

[A.5] Pčolka, M.; Žáčeková, E.; Robinett, R.; Čelikovský, S.; Šebek, M. (2014; authorship: 50 %). *Economical Nonlinear Model Predictive Control for Building Climate Control*. 2014 American Control Conference (2014 ACC), Portland, U.S. (indexed in *Web of Science*). DOI: 10.1109/ACC.2014.6858928.

[A.6] Pčolka, M.; Žáčeková, E.; Robinett, R.; Čelikovský, S.; Šebek, M. (2014; authorship: 50 %). *From Linear to Nonlinear Model Predictive Control of a Building*. 19th World Congress of the International Federation of Automatic Control (IFAC 2014), Cape Town, South Africa. DOI: doi.org/10.3182/20140824-6-ZA-1003.02783.

[A.7] Žáčeková, E.; Pčolka, M.; Tabaček, J.; Těžký, J.; Robinett, R.; Čelikovský, S.; Šebek, M. (2015; authorship: 18 %). *Identification and Energy Efficient Control for a Building: Getting Inspired by MPC*. 2015 American Control Conference (2015 ACC), Chicago, U.S. (indexed in *Web of Science*). DOI: 10.1109/ACC.2015.7170973.

Conference proceedings

[A.8] Pčolka, M.; Čelikovský, S. (2012; authorship: 50 %). *Gradient Method Optimization of Penicillin Production*. 24th Chinese Control and Decision Conference (CCDC), Taiyuan, China (indexed in *Web of Science*). DOI: 10.1109/CCDC.2012.6244012.

[A.9] Pčolka, M.; Čelikovský, S. (2012; authorship: 50 %). *Gradient Method Optimization of Penicillin Production: New Strategies*. 20th Mediterranean Conference on Control & Automation (MED 2012), Barcelona, Spain. DOI: 10.1109/MED.2012.6265808.

[A.10] Pčolka, M.; Čelikovský, S. (2012; authorship: 50 %). *Multiple-input Cultivation Model Based Optimization of Penicillin Production*. 51st IEEE Conference on Decision and Control (CDC), Maui, U.S. (indexed in *Web of Science*). DOI: 10.1109/CDC.2012.6426276.

[A.11] Pčolka, M.; Čelikovský, S. (2013; authorship: 50 %). *Enhancement of Practical Applicability of Optimal Control of a Nonlinear Process*. 2013 American Control Conference (2013 ACC), Washington, DC, U.S. (indexed in *Web of Science*). DOI: 10.1109/ACC.2013.6580613.

[A.12] Pčolka, M.; Čelikovský, S. (2013; authorship: 50 %). *On Nonlinear Continuous-time Optimal Control of Penicillin Cultivation*. European Control Conference 2013 (ECC 2013), Zurich, Switzerland (indexed in *Web of Science*). DOI: 10.23919/ECC.2013.6669524.

[A.13] Žáčeková, E.; Pčolka, M.; Čelikovský, S.; Šebek, M. (2014; authorship: 30 %). *Semi-receding Horizon Algorithm for "Sufficiently Exciting" MPC with Adaptive Search Step*. 53rd IEEE Conference on Decision and Control (CDC), Los Angeles, U.S. (indexed in *Web of Science*). DOI: 10.1109/CDC.2014.7040034.

[A.14] Pčolka, M.; Žáčeková, E.; Robinett, R.; Čelikovský, S.; Šebek, M. (2015; authorship: 50 %). *Quantized Nonlinear Model Predictive Control for a Building.* 2015 IEEE Conference on Control and Applications (CCA 2015), Sydney, Australia (indexed in *Web of Science*). DOI: 10.1109/CCA.2015.7320653.

[A.15] Pčolka, M.; Čelikovský, S. (2016; authorship: 50 %). *Algorithms for Nonlinear Predictive Control Maximizing Penicillin Production Efficiency.* 2016 American Control Conference (2016 ACC), Boston, U.S. (indexed in *Web of Science*). DOI: 10.1109/ACC.2016.7525460.

Publications not related to the thesis

Articles in journals with impact factor

[A.16] Žáčeková, E.; Prívara, S.; Pčolka, M. (2013; authorship: 10 %). *Persistent Excitation Condition within the Dual Control Framework.* Journal of Process Control (2017 impact factor: 2.787). DOI: 10.1016/j.jprocont.2013.08.004.

[A.17] Žáčeková, E.; Pčolka, M.; Šebek, M. (2018; under review). *Zone MPC with Guaranteed Identifiability in Presence of Predictable Disturbances.* Journal of the Franklin Institute.

Conference proceedings

[A.18] Žáčeková, E.; Pčolka, M.; Šebek, M. (2014; authorship: 35 %). *On Satisfaction of the Persistent Excitation Condition for the Zone MPC: Numerical Approach.* 19th World Congress of the International Federation of Automatic Control (IFAC 2014), Cape Town, South Africa. DOI: doi.org/10.3182/20140824-6-ZA-1003.02705.

[A.19] Vey, D.; Lunze, J.; Žáčeková, E.; Pčolka, M.; Šebek, M. (2015; authorship: 25 %). *Control Reconfiguration of Full-state Linearizable Systems by a Virtual Actuator.* 9th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (Safeprocess 2015), Paris, France. DOI: 10.1016/j.ifacol.2015.09.550.

[A.20] Žáčeková, E.; Pčolka, M.; Šebek, M.; Čelikovský, S. (2015; authorship: 30 %). *MPC for a Class of Nonlinear Systems with Guaranteed Identifiability.* 2015 IEEE

Conference on Control and Applications (CCA 2015), Sydney, Australia (indexed in *Web of Science*). DOI: 10.1109/CCA.2015.7320627.

Responses to author's publications

Pčolka, M.; Čelikovský, S. (2014). *Numerical Aspects of Optimal Control of Penicillin Production*. Bioprocess and Biosystems Engineering.

- Wang, B.; Ji, X. F.; Sun, Y. K. (2015). *Soft-sensing Modeling of Crucial Parameters for Penicillin Fed-batch Fermentation Process.* Bulgarian Chemical Communications.

Pčolka, M.; Žáčeková, E.; Robinett, R.; Čelikovský, S.; Šebek, M. (2016). *Bridging the Gap between the Linear and Nonlinear Predictive Control: Adaptations for Efficient Building Climate Control*. Control Engineering Practice.

- Killian, M.; Kozek, M. (2016). *Ten Questions Concerning Model Predictive Control for Energy Efficient Buildings*. Building and Environment.
- O'Dwyer, E.; De Tommasi, L.; Kouramas, K.; Cychowski, M.; Lightbody, G. (2016). *Modelling and Disturbance Estimation for Model Predictive Control in Building Heating Systems*. Energy and Buildings.
- Alamin, Y.I.; del Mar Castilla, M.; Domingo Álvarez, J.; Ruano, A. (2017). *An Economic Model-based Predictive Control to Manage the Users' Thermal Comfort in a Building*. Energies.
- Cao, P.; Luo, X.; Song, X. (2017). *Feasibility Analysis and Online Adjustment of Constraints in Model Predictive Control Integrated with Soft Sensor*. Chinese Journal of Chemical Engineering.
- Killian, M.; Kozek, M. (2018). *Implementation of Cooperative Fuzzy Model Predictive Control for an Energy-efficient Office Building*. Energy and Buildings.
- Domingo Álvarez, J.; Costa-Castello, R.; del Mar Castilla, M. (2018). *Repetitive Control to Improve Users' Thermal Comfort and Energy Efficiency in Buildings*. Energies.

- Hang, L.; Kim, D.H. (2018). *Enhanced Model-based Predictive Control System Based on Fuzzy Logic for Maintaining Thermal Comfort in IoT Smart Space.* Applied Sciences.

Pčolka, M.; Žáčeková, E.; Robinett, R.; Čelikovský, S.; Šebek, M. (2014). *Economical Nonlinear Model Predictive Control for Building Climate Control.* American Control Conference.

- Al-amin, Y.I.M. (2015). *Predictive Models of Buildings Energy Consumption.* Master thesis, Universidade do Algarve.
- Dobbs, J.S.; Razmara, M.; Shahbakhti, M.; Paudyal, S. (2016). *Selecting Control Input Type for a Building Predictive Controller Integrated in a Power Grid.* American Control Conference.

Pčolka, M.; Žáčeková, E.; Robinett, R.; Čelikovský, S.; Šebek, M. (2014). *From Linear to Nonlinear Model Predictive Control of a Building.* World Congress of the International Federation of Automatic Control.

- Pangalos, G. (2016). *Model-based Controller Design Methods for Heating Systems.* PhD thesis, Technischen Universität Hamburg-Harburg.

Žáčeková, E.; Pčolka, M.; Tabaček, J.; Těžký, J.; Robinett, R.; Čelikovský, S.; Šebek, M. (2015). *Identification and Energy Efficient Control for a Building: Getting Inspired by MPC.* American Control Conference.

- Dobbs, J.S.; Razmara, M.; Shahbakhti, M.; Paudyal, S. (2016). *Selecting Control Input Type for a Building Predictive Controller Integrated in a Power Grid.* American Control Conference.
- Ji, R.; Du, D.; Chang, B.; Wang, Li; Zeng, J.; Hong, Y. (2017). *Research on the Coordination of Multiple Air Circulating Tempering Furnaces Using System Identification and Predictive Control in Manufacturing of Non-combustible Aluminum Composite Panels.* ASME International Manufacturing Science and Engineering Conference.

- Quachio, R.; Garcia, C. (2017). *MPC Relevant Identification Method for Hammerstein Models*. Control Conference Africa.
- Drgoňa, J.; Picard, D.; Kvasnica, M.; Helsen, L. (2018). *Approximate Model Predictive Building Control via Machine Learning*. Applied Energy.

Pčolka, M.; Čelikovský, S. (2012). *Gradient Method Optimization of Penicillin Production: New Strategies*. Mediterranean Conference on Control & Automation.

- Zhai, C., Wang, R., Ren, Z., Sun, W. (2017). *Dynamic Optimization of Fedbatch Fermentation with Constraint on Wastewater Discharge*. Chemical Engineering Transactions.

Žáčeková, E.; Prívara, S.; Pčolka, M. (2013). *Persistent Excitation Condition within the Dual Control Framework*. Journal of Process Control.

- Benosman, M.; Di Cairano, S.; Weiss, A. (2014). *Extremum Seeking-based Iterative Learning Linear MPC*. IEEE Conference on Control Applications.
- Kumar, K.; Heirung, T.A.N.; Patwardhan, S.C.; Foss, B. (2015). *Experimental Evaluation of a MIMO Adaptive Dual MPC*. IFAC Symposium on Advanced Control of Chemical Processes.
- Heirung, T.A.N.; Foss, B.; Ydstie, B. (2015). *MPC-based Dual Control with Online Experiment Design*. Journal of Process Control.
- Hernandez, B.; Trodden, P. (2016). *Persistently Exciting Tube MPC*. American Control Conference.
- Benosman, M. (2016). *Learning-Based Adaptive Control: An Extremum Seeking Approach - Theory and Applications*.
- Anderson, A.; Ferramosca, A.; Gonzalez, A. H.; Kofman, E. (2016). *Probabilistic Invariant Sets for Closed-Loop Re-Identification*. IEEE Latin America Transactions.
- Bustos, G.A.; Ferramosca, A.; Godoy, J. L.; Gonzalez, A.H. (2016). *Application of Model Predictive Control Suitable for Closed-loop Re-identification to a Polymerization Reactor*. Journal of Process Control.

- Houska, B.; Telen, D.; Logist, F.; Van Impe, J. (2017). *Self-reflective Model Predictive Control*. SIAM Journal on Control and Optimization.
- Mesbah, A. (2018). *Stochastic Model Predictive Control with Active Uncertainty Learning: A Survey on Dual Control*. Annual Reviews in Control.
- Sedghizadeh, S.; Beheshti, S. (2018). *Particle Swarm Optimization Based Fuzzy Gain Scheduled Subspace Predictive Control*. Engineering Applications of Artificial Intelligence.
- Feng, X.; Houska, B. (2018). *Real-time Algorithm for Self-reflective Model Predictive Control*. Journal of Process Control.
- Yin, X.; Pan, Li (2018). *Enhancing Trajectory Tracking Accuracy for Industrial Robot with Robust Adaptive Control*. Robotics and Computer-integrated Manufacturing.
- Feng, X.; Jiang, Y.; Villanueva, M.E.; Houska, B. *Parallelizable Real-Time Algorithm for Integrated Experiment Design MPC*.
- Anderson, A.; González, A.H.; Ferramosca, A.; D'Jorge, A.; Kofman, E. (2018). *Robust MPC Suitable for Closed-loop Re-identification, Based on Probabilistic Invariant Sets*. Systems and Control Letters.