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 $T_\mathrm{F}$ 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 $i$th 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 $T_{\text{samp}}$-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 $T_{\text{samp}}$-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 $\alpha_l$ 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 $\lambda$ 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, $T_{\mathrm{F}}$]. 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 $\lambda$, 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 $T_{\text{samp}}$, 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 $T_{\text{samp}}$-sampled optimal input function. Rather than decrease $T_{\text{samp}}$, 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 $p_\mathrm{A}$ 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:

$$
\begin{aligned}
&\min\ N_\mathrm{P} \\
\text{subject to}:\ &\mathcal{J}(u(\mathbf{P}, \mathfrak{F}, t)) \leq \mathcal{J}(\overline{\mathbf{u}}) + \Delta\mathcal{J} \\
&\text{constraints (2)} \\
&\Delta\mathcal{J} \geq 0 \\
&\overline{\mathbf{u}} = \{\overline{u}(k) \,|\, k = 1, 2, \ldots, P\} \\
&\mathbf{P} = \mathbf{P}_\mathrm{T} \bigcup \mathbf{P}_\mathrm{S} \bigcup \mathbf{P}_\mathrm{A};\ |\mathbf{P}| = N_\mathrm{P} \\
&\mathbf{P}_\mathrm{T} = \{p_{\mathrm{T},i} \,|\, i = 1, 2, \ldots, N_\mathrm{T};\ 0 < p_{\mathrm{T},1} < p_{\mathrm{T},2} < \cdots < p_{\mathrm{T},N_\mathrm{T}} = T_\mathrm{F}\} \\
&\mathbf{P}_\mathrm{S} = \left\{ p_{\mathrm{S},i,q} \,|\, u_i = \sum_{q=1}^{s_i} p_{\mathrm{S},i,q} \mathfrak{f}_{i,q};\ \mathfrak{f}_{i,q} \in \mathfrak{F} \right\} \\
&\mathbf{P}_\mathrm{A} = \{p_{\mathrm{A},i,q,m} \,|\, \mathfrak{f}_{i,q} = \mathfrak{f}_{\mathrm{A},i,q}(p_{\mathrm{A},i,q,m}, t);\ \mathfrak{f}_{\mathrm{A},i,q} \in \mathfrak{F}_\mathrm{A}\} \\
&\mathfrak{F} = \mathfrak{F}_\mathrm{S} \bigcup \mathfrak{F}_\mathrm{A}
\end{aligned}
$$

$$\mathfrak{F}_S = \{f_S^r(t) = t^r \mid r \in \mathbb{Z}\}$$
$$\mathfrak{F}_A = \{f_A(\mathbf{P}_A, t)\}. \tag{11}$$

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:

$$\mathbf{P}_T = \{k\, T_{samp} \mid k = 1, 2, \ldots, P\}$$
$$\mathbf{P}_S = \{\overline{u}(k) \mid k = 1, 2, \ldots, P\}$$
$$\mathbf{P}_A = \emptyset$$
$$\mathfrak{F}_S = \{t^0\}$$
$$\mathfrak{F}_A = \emptyset.$$

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

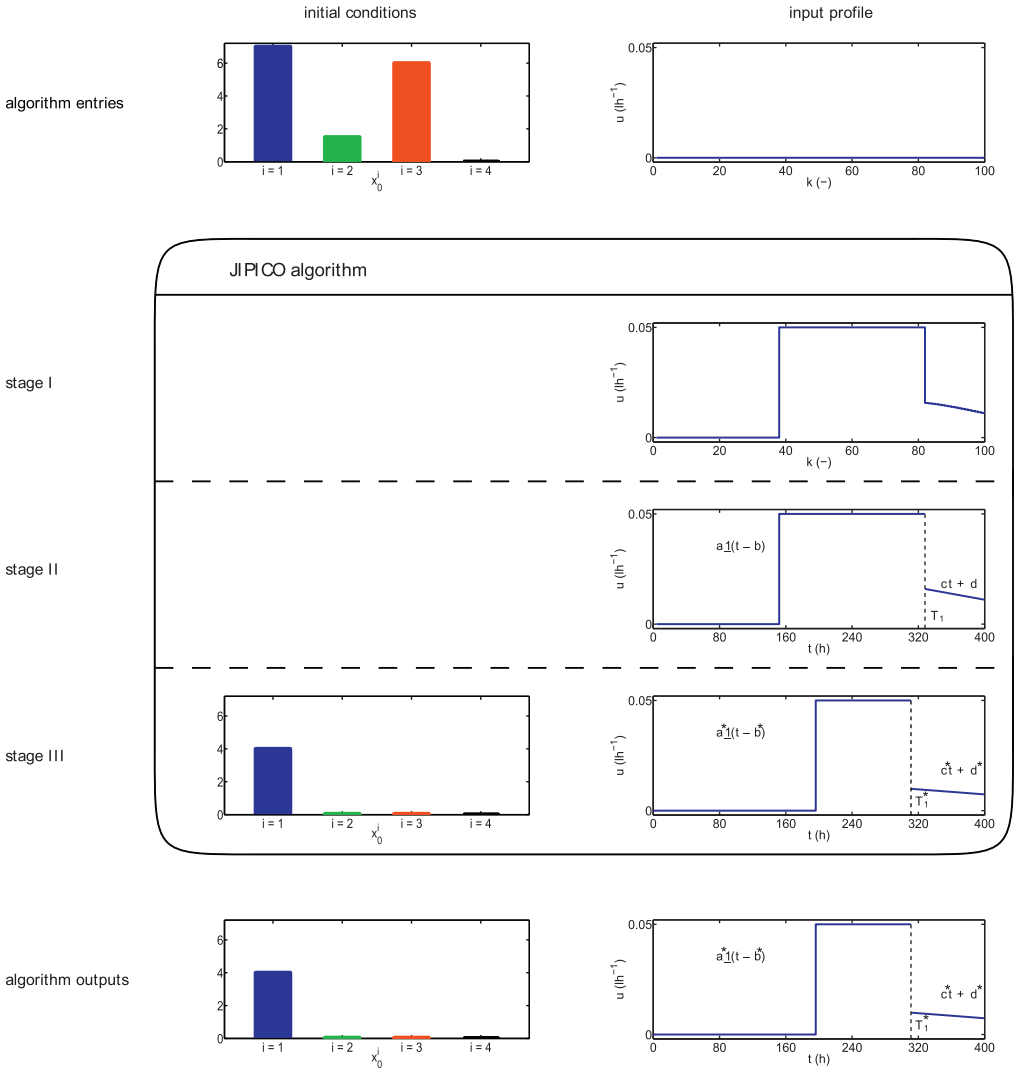


Fig. 1. JIPICO algorithm.

significantly reduced compared to the $T_{\text{samp}}$-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 ($\text{g l}^{-1}$), essential nutrient (glucose) concentration ($\text{g l}^{-1}$) and penicillin concentration ($\text{g l}^{-1}$) while input $u$ ($\text{l h}^{-1}$) 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 $\mu$ and creates the penicillin (whose concentration is represented by $x_4$) at specific production rate $\pi$. Penicillin cultivation is a typical secondary metabolism example with the Contois formula describing $\mu$ and Haldane kinetics modeling $\pi$. In this case, $\mu$ considers a saturation of growth with respect to the substrate and a growth decrease related to the cells accumulation while $\pi$ is activated by law substrate level, reaches its maximum at a precise level and is inhibited by the substrate itself beyond this level. Mathematically, $\mu$

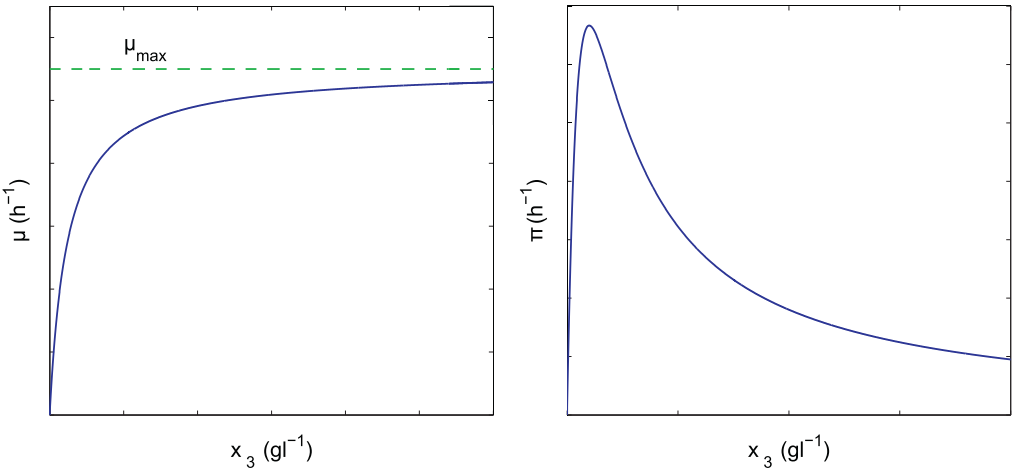


Fig. 2. Growth and production rate profiles.

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 | $K_{vap}$ $(h^{-1})$ | $6.23 \times 10^{-4}$ |
| $Y_{X/S}$ $(-)$ | 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 $\pi$ 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 $T_{\mathrm{F}}$ 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 $T_{\mathrm{samp}}$-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$1^{-1}$, g$1^{-1}$, g$1^{-1}$]$^{\mathrm{T}}$) | $[7, 1.5, 6, 0]^{\mathrm{T}}$ |
| | $T_{\mathrm{samp}}$ (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 $\lvert\tilde{\mathcal{J}}(p_i)\rvert/\lvert\mathcal{J}(\overline{\mathbf{u}})\rvert$.

| 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 $T_{\mathrm{samp}}$-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 $\lvert\Delta\mathcal{J}\rvert/\lvert\mathcal{J}(\overline{\mathbf{u}})\rvert$. 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 $N_{\mathrm{P}} = \lvert\mathbf{P}\rvert = \lvert\mathbf{P}_{\mathrm{T}} \bigcup \mathbf{P}_{\mathrm{S}} \bigcup \mathbf{P}_{\mathrm{A}}\rvert = 6 \le 2P$. 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 $\lvert\tilde{\mathcal{J}}(p_i)\rvert/\lvert\mathcal{J}(\overline{\mathbf{u}})\rvert$ 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-

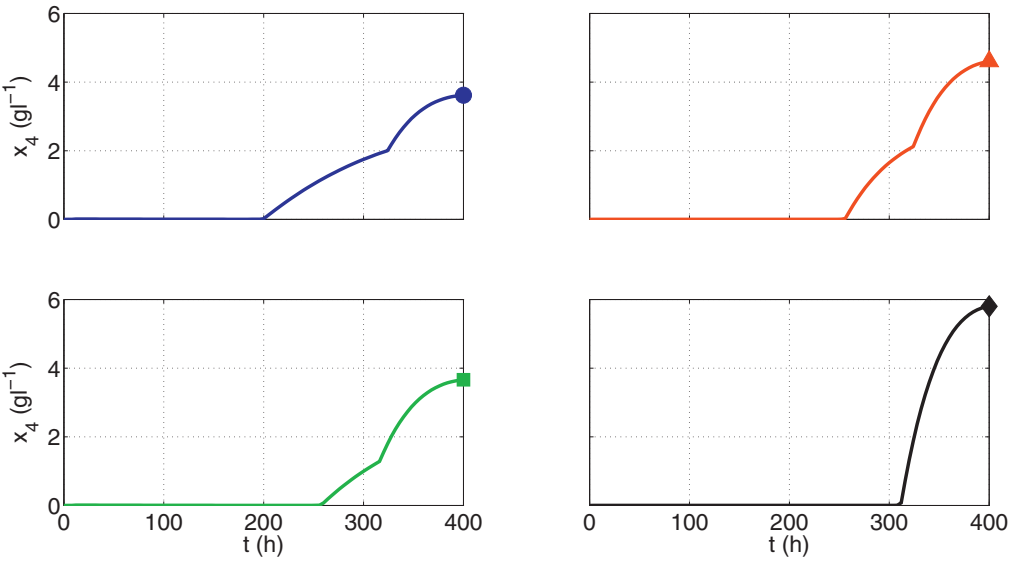


Fig. 3. Penicillin concentration profiles (—●—HGO, —▲—ICO, —■—IPRO, —◆—JIPICO).

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 $\pi$ and $\pi x_2$ 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

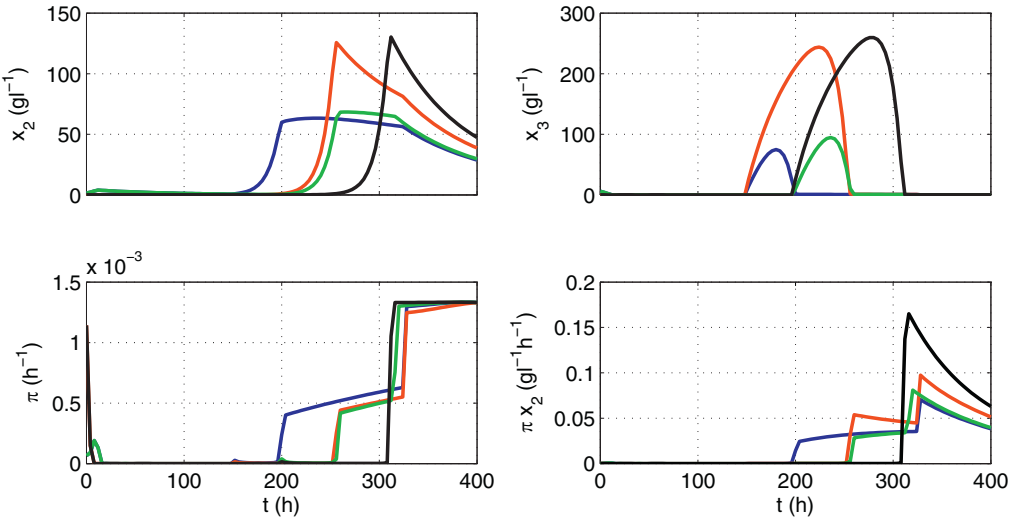


Fig. 4. Suplementary profiles (—HGO, —ICO, —IPRO, —JIPICO).

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 $\pi$ 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 $\pi$ timing leads to sovereignly highest $\pi x_2$ 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

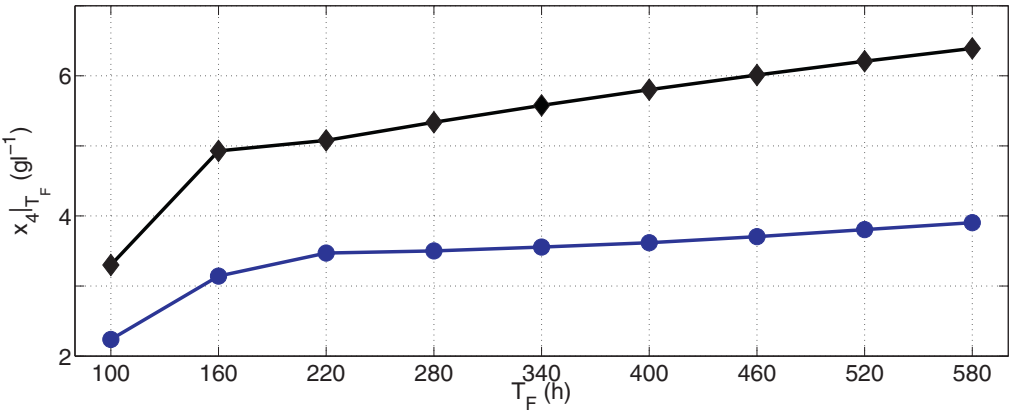


Fig. 5. Terminal penicillin concentration (—●—HGO, —♦—JIPICO).

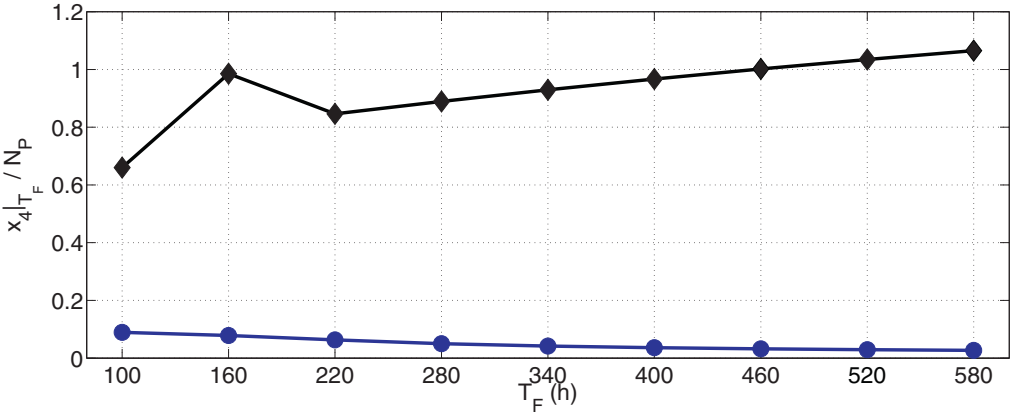


Fig. 6. Relative terminal penicillin concentration with respect to the number of optimized parameters (—●—HGO, —♦—JIPICO).

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 $T_{\text{samp}}$-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.