Global Dynamic Optimization with Hammerstein-Wiener Models Embedded

Chrysoula D. Kappatou[a], Dominik Bongartz[a], Jaromił Najman[a], Susanne Sass[a] and Alexander Mitsos[b,a,c*]

[a]Process Systems Engineering (AVT.SVT), RWTH Aachen University, 52074 Aachen, Germany

[b]JARA-CSD, 52056 Aachen, Germany

[c]Institute of Energy and Climate Research: Energy Systems Engineering (IEK-10), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany

Abstract: Hammerstein-Wiener models constitute a significant class of block-structured dynamic models, as they approximate process nonlinearities on the basis of input-output data without requiring identification of a full nonlinear process model. Optimization problems with Hammerstein-Wiener models embedded are nonconvex, and thus local optimization methods may obtain sub-optimal solutions. In this work, we develop a deterministic global optimization strategy that exploits the specific structure of Hammerstein-Wiener models to extend existing theory on global optimization of systems with linear dynamics. At first, we discuss alternative formulations of the dynamic optimization problem with Hammerstein-Wiener models embedded, demonstrating that careful selection of the optimization variables of the problem can offer significant numerical advantages to the solution approach. Then, we develop the theory for convergence to the global solution focusing on a control parametrization technique. Finally, we apply our optimization strategy to case studies comprising both offline and online dynamic optimization problems. The results confirm the improved computational performance of the proposed solution approach over alternative options not exploiting the linear dynamics discussed in the manuscript, and underline the high potential of our method especially when used for online applications like nonlinear model predictive control.

Keywords: global optimization, surrogate models, optimal control, Hammerstein-Wiener, NMPC

Corresponding author: *A. Mitsos
AVT Process Systems Engineering, RWTH Aachen University, 52074 Aachen, Germany
E-mail: amitsos@alum.mit.edu

1 Introduction

Dynamic optimization problems arise in various domains, examples within the field of chemical engineering being process design, operation and control [3]. Among these problems, only a few – relative simple and small ones – allow for an analytical solution. In most cases, the solution requires numerical methods [6].

Deriving local solutions for dynamic optimization problems has been studied extensively in the literature, and mature and efficient technologies are available, which are able to handle even large-scale and complex systems [32]. The two main solution approaches for dynamic optimization problems are variational (indirect) and discretization (direct) methods. A further classification of discretization methods occurs based on whether or not the discretization refers only to the controls or also to the states; resulting in sequential and simultaneous methods, respectively [3].

In practice, most chemical and biochemical engineering problems are non-convex, and may therefore exhibit multiple local minima [8]. Although the application of local optimization methods to solve these problems is reasonable in terms of computational effort, they do not guarantee global optimality of the final solution. However, in many of these problems global solutions are desired, or even required, e.g., in cases where we are interested in the best fit for model evaluation such as the kinetic mechanism in chemical reactions (cf., e.g., [26, 37]). In general, finding the global solution of a problem can have direct economical, environmental and safety impacts [8].

Deterministic approaches to globally solve problems with ordinary differential equations (ODEs) embedded are an evolving field of study, with significant accomplishments over the past years [11]. Deterministic global optimization guarantees convergence to an ϵ-optimal solution within a finite number of steps. A popular approach to tackle these problems is to combine discretization methods with a spatial branch-and-bound (B&B) algorithm. Such an approach typ-

ically provides solutions to finite dimensional optimization problems. Infinite dimensional problems like optimal control problems, where the optimization variables are continuous functions, can be transformed into finite dimensional NLPs by control vector parametrization [16]. Recently, Houska and Chachuat [12] proposed a global optimization algorithm for optimal control problems that includes an adaptive refinement of the control parametrization to guarantee convergence to the solution of the infinite dimensional problem.

The solution of the parametrized problem relies on extensions of sequential and simultaneous methods for local dynamic optimization. The methods based on extensions of the simultaneous approach, similar to their original simultaneous approach as in full discretization for local dynamic optimization, result in large scale NLPs. As the worst-case computational effort of B&B scales exponentially with the number of variables, the applicability of these methods is limited to small problems. Hence, most research efforts on global dynamic optimization have been focused on extensions of sequential approaches. However, for the latter cases, the construction of the lower bounding problem for a convergent B&B algorithm is a challenging topic [32].

Recent attempts on deterministic global dynamic optimization with main focus on extensions of sequential NLP approaches have been reviewed in [8, 12]. One approach is based on extensions of the αBB method [1] to NLPs containing ODEs. These methods are computationally expensive, as they typically require the calculation of second-order sensitivities to determine a shift parameter that is not known apriori, cf., e.g., [10, 25]. A different approach based on McCormick relaxations [22] is presented by Singer and Barton [35, 36]. These methods are reported to have better performance than αBB-based approaches and can in general handle a wider class of ODEs [8]. Both above mentioned approaches follow a relax-then-discretize fashion, meaning that they first construct relaxations to the infinite dimensional ODE system and then discretize these to get the numerical solution. In contrast, a discretize-then-relax approach that first discretizes the dynamics and then treats the resulting NLP in a reduced space

is proposed by Mitsos et al. [23] based on automatic propagation of McCormick relaxations and their subgradients. Sahlodin and Chachuat [28] provide a rigorous discretize-then-relax approach to account for the truncation error arising during the discretization step. Recently, Scott and Barton [29, 30] presented a novel method for constructing relaxations for semi-explicit index-one differential algebraic equations (DAEs) providing the first algorithm for solving problems with DAEs embedded to global optimality [32]. Yet, progress in this field is still at an early stage, and active research on this topic is necessary to improve computational performance and to make larger problems tractable.

One way of improving computational performance is to exploit special structure of certain important model classes, rather than rely on general-purpose methods. Hammerstein-Wiener (HW) models constitute a significant example of such a class. They are data-driven dynamic models bringing the advantage of capturing nonlinear effects and simultaneously being computationally less complex than fully nonlinear dynamic models. HW models cover a wide range of applications, such as modeling of physical, chemical and biological systems [18]. Extensive research on system identification of those models has been performed in the literature, cf., e.g., [2, 40, 43], and they are often used for model predictive control, cf., e.g., [18, 41]. Upon optimization with HW models embedded, we still get a nonlinear problem. To avoid suboptimal solutions of the resulting optimization problem and high computational effort, tailored deterministic global optimization methods and formulations are required.

In this work, we propose a computational approach for global dynamic optimization with HW models. We utilize the specific structure of HW models by exploiting the special properties of linear dynamics occurring in these models. More precisely, we extend existing theory on global dynamic optimization with linear systems presented by Singer and Barton [34, 35] to account for the input and output nonlinearities of HW models. Finally, we numerically solve some illustrative examples using our open-source optimization software MAiNGO[1] [5]

[1]The open-source version of MAiNGO is available at `https://git.rwth-aachen.de/avt.svt/public/MAiNGO.git`.

following the method presented by Mitsos et al. [23].

The remainder of this manuscript is structured as follows. In Section 2, we present the structure of HW models, we describe the optimization problem and discuss alternative formulations with their impact on the solution approach. In Section 3, we derive the required theory for the solution of the presented problem to global optimality and report on the practical implementation aspects. Computational results for three examined case studies are presented in Section 4. The model implementations for these case studies are being made available as Supplementary Information. Section 5 concludes this work.

2 Problem Description

2.1 General Form of Hammerstein-Wiener Models

In HW models, two static nonlinear blocks precede and follow, respectively, a linear dynamic system (see Figure 1). The input nonlinearity $\mathbf{f}_{\mathrm{H}} : \mathbb{R}^{n_u} \to \mathbb{R}^{n_w}$ is called Hammerstein function and the output nonlinearity $\mathbf{f}_{\mathrm{W}} : \mathbb{R}^{n_z} \to \mathbb{R}^{n_y}$ is the Wiener function:

$$\begin{aligned}
\mathbf{w}(t) &= \mathbf{f}_{\mathrm{H}}(\mathbf{u}(t)), \quad \forall t \in [t_0, t_f] \\
\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{w}(t), \quad \forall t \in [t_0, t_f] \\
\mathbf{z}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{w}(t), \quad \forall t \in [t_0, t_f] \\
\mathbf{y}(t) &= \mathbf{f}_{\mathrm{W}}(\mathbf{z}(t)), \quad \forall t \in [t_0, t_f] \\
\mathbf{x}(t_0) &= \mathbf{x}_0,
\end{aligned} \tag{1}$$

where $\mathbf{u} : [t_0, t_f] \to \mathbb{R}^{n_u}$ are the inputs of the system, $\mathbf{w} : [t_0, t_f] \to \mathbb{R}^{n_w}$ are the inputs to the linear time-invariant (LTI) system, $\mathbf{x} : [t_0, t_f] \to \mathbb{R}^{n_x}$ are the states, $\mathbf{z} : [t_0, t_f] \to \mathbb{R}^{n_z}$ are the outputs of the LTI system, $\mathbf{y} : [t_0, t_f] \to \mathbb{R}^{n_y}$ the outputs of the system, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_w}$, $\mathbf{C} \in \mathbb{R}^{n_z \times n_x}$, $\mathbf{D} \in \mathbb{R}^{n_z \times n_w}$ are system matrices of the LTI , and $\mathbf{x}_0 \in \mathbb{R}^{n_x}$ are the initial states. Note that

due to the physical meaning of real-world applications, the input variables are bounded, i.e., $\mathbf{u}(t) \in U,\ U \subsetneq \mathbb{R}^{n_u}$, U compact.

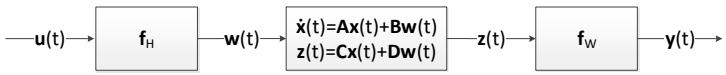


Fig. 1: Structure of a Hammerstein-Wiener model

2.2 Optimization Problem Formulation

The formulation of an optimization problem with embedded HW models can be written as

$$\min_{\mathbf{u}(\cdot)} \quad \Phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{y}(t), \mathbf{u}(t), t)\ dt, \tag{2}$$

where $\mathbf{x}(\cdot)$ and $\mathbf{y}(\cdot)$ derive from the solution of the DAE system (1).

In Problem (2), a few simplifications were made for notational convenience. Nevertheless, more general problems can be handled without requiring changes to the developed theory. The first term of the objective is only dependent on the final time point, yet any additional dependence on the state variables at any finite number of fixed time points can be added. Moreover, since HW models are usually built on an input-output relationship, we consider only a dependence on model outputs $\mathbf{y}$, on the inputs $\mathbf{u}$ and explicitly on time t, for the integrand l. However, l may in general also depend on other variables, e.g., the states $\mathbf{x}$ and their derivatives $\dot{\mathbf{x}}$. In addition, we could, without any significant changes to the theory, generalize the Wiener block to include any relationship of the form $\mathbf{y}(t) = \mathbf{f}_{\mathrm{W}}(\mathbf{x}(t), \mathbf{w}(t))$ for all $t \in [t_0, t_f]$, or even $\mathbf{y}(t) = \mathbf{f}_{\mathrm{W}}(\mathbf{x}(t), \mathbf{u}(t))$ for all $t \in [t_0, t_f]$. Note that the latter case does no longer satisfy the HW model structure, but can be interesting to consider in general.

Problem (2) contains a DAE system with linear dynamic equations. Optimization problems with DAEs embedded are in general very hard to solve

globally, see discussion in Section 1. In the following, we discuss different options for expressing the problem formulation.

2.2.1 Analytical Solution of the LTI

Probably the most intuitive solution approach is to incorporate the analytical solution of the linear dynamic system,

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{w}(\tau)\ d\tau,$$

into Problem (2), and thus eliminate the ODE. By substituting both the input to the LTI system $\mathbf{w}(\cdot)$ and the system output $\mathbf{y}(\cdot)$, with the functions of the Hammerstein and Wiener blocks, respectively, we derive

$$\begin{aligned}\min_{\mathbf{u}(\cdot)} \quad \Phi\Bigg(e^{\mathbf{A}(t_f-t_0)}\mathbf{x}_0 + \int_{t_0}^{t_f} e^{\mathbf{A}(t_f-\tau)}\mathbf{B}\mathbf{w}(\tau)\ d\tau, t_f\Bigg) + \int_{t_0}^{t_f} l\Bigg(\mathbf{f}_{\mathrm{W}}\Bigg(\mathbf{C}e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 \\ +\mathbf{C}\int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{f}_{\mathrm{H}}(\mathbf{u}(\tau))\ d\tau + \mathbf{D}\mathbf{f}_{\mathrm{H}}(\mathbf{u}(t))\Bigg), \mathbf{u}(t), t\Bigg)\ dt.\end{aligned} \tag{3}$$

This problem formulation is complicated to solve, since for the inner integral of Equation (3) there may not exist an analytical solution in dependence of t for arbitrary $\mathbf{f}_{\mathrm{H}}$.

2.2.2 Substitution Approach

Alternatively, we can only exploit the fact that $\mathbf{w}$ and $\mathbf{y}$ are explicit functions of $\mathbf{u}$ and $\mathbf{z}$, respectively, and obtain

$$\min_{\mathbf{u}(\cdot)} \quad \Phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{f}_{\mathrm{W}}(\mathbf{z}(t)), \mathbf{u}(t), t)\ dt, \tag{4}$$

where $\mathbf{x}(\cdot)$, $\mathbf{z}(\cdot)$ now are obtained by the solution of the ODE system

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{f}_{\mathrm{H}}(\mathbf{u}(t)), \quad \forall t \in [t_0, t_f] \\ \mathbf{z}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{f}_{\mathrm{H}}(\mathbf{u}(t)), \quad \forall t \in [t_0, t_f] \\ \mathbf{x}(t_0) &= \mathbf{x}_0.\end{aligned} \tag{5}$$

However, unlike the original Problem (2), Problem (4) has nonlinear dynamics given by the ODE system (5). Therefore, the advantage of the linear dynamics will be lost. This is particularly important since relaxations of nonlinear dynamics are typically weak.

2.2.3 Inversion Approach

To retain a problem with linear dynamics, one alternative is to treat the Wiener model (LTI system plus Wiener function) separately, using existing theory on linear dynamics by Singer and Barton [34, 35] and optimize for $\mathbf{w}(\cdot)$. To treat the dependence on $\mathbf{u}(\cdot)$ in the objective, in case of invertibility of the Hammerstein function $\mathbf{f}_{\mathrm{H}}$ or similar condition, we derive

$$\min_{\mathbf{w}(\cdot)} \quad \Phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l\big(\mathbf{f}_{\mathrm{W}}(\mathbf{z}(t)), \mathbf{f}_H^{-1}(\mathbf{w}(t)), t\big)\, dt, \tag{6}$$

where $\mathbf{x}(\cdot)$, $\mathbf{z}(\cdot)$ are obtained by the solution of the LTI system

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{w}(t), \quad \forall t \in [t_0, t_f] \\ \mathbf{z}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{w}(t), \quad \forall t \in [t_0, t_f] \\ \mathbf{x}(t_0) &= \mathbf{x}_0.\end{aligned} \tag{7}$$

Even in the case where the objective does not depend on $\mathbf{u}(\cdot)$, once the optimal solution $\mathbf{w}^*(\cdot)$ is given, the above approach would still require specific assumptions on existence and uniqueness of the optimal control $\mathbf{u}^*(\cdot)$. With the assumption of an invertible Hammerstein function $\mathbf{f}_{\mathrm{H}}$, once we have the optimal $\mathbf{w}^*(t)$ for all $t \in [t_0, t_f]$, we can solve $\mathbf{u}^*(t) = \mathbf{f}_H^{-1}(\mathbf{w}^*(t))$ to obtain $\mathbf{u}^*(t)$ for all

$t \in [t_0, t_f]$.

Note that the assumption on invertibility of the nonlinear static functions $\mathbf{f}_{\mathrm{H}}$ and $\mathbf{f}_{\mathrm{W}}$ is commonly made for identifiability of HW models [43]. However, this assumption significantly limits the choice of these functions, and thus the applicability of this approach. Furthermore, this approach necessitates exact bounds on $\mathbf{w}(t)$ for all $t \in [t_0, t_f]$, to ensure a feasible $\mathbf{u}(t)$ for all $t \in [t_0, t_f]$. We actually need to find the exact range of $\mathbf{f}_{\mathrm{H}}$ on the domain of $\mathbf{u}$, rather than an overestimated box, which can be as complex as solving the final optimization problem.

2.2.4 Additional Optimization Variables Approach

The idea behind this approach is to introduce additional optimization variables to Problem (4) to re-gain the linearity of the dynamic system. To this end, we optimize with respect to both $\mathbf{u}(\cdot)$ and $\mathbf{w}(\cdot)$. More precisely, by treating $\mathbf{u}(\cdot)$ and $\mathbf{w}(\cdot)$ as independent optimization variables and imposing their dependence in an additional constraint, we can retain the linear dynamic behavior of the system with respect to $\mathbf{w}(\cdot)$ and use existing theory on global optimization of systems with linear dynamics [34, 35]. The optimization problem is then formulated as

$$\begin{aligned} \min_{\mathbf{u}(\cdot),\mathbf{w}(\cdot)} \quad & \Phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{f}_{\mathrm{W}}(\mathbf{z}(t)), \mathbf{u}(t), t)\ dt \\ \text{s.t.} \quad & \mathbf{w}(t) = \mathbf{f}_{\mathrm{H}}(\mathbf{u}(t)), \quad \forall t \in [t_0, t_f], \end{aligned}$$

where $\mathbf{x}(\cdot)$, $\mathbf{z}(\cdot)$ derive from solving the LTI system (7).

The additional optimization variables $\mathbf{w}(\cdot)$ are used in a similar way to the additional module and tear variables presented by Bongartz et al. [4] for decoupling the model equations that would require iterative solution in process flowsheet optimization.

By eliminating the intermediate variables $\mathbf{z}(\cdot)$ and introducing the function $\tilde{l}$ defined as $\tilde{l}(t, \mathbf{x}(t), \mathbf{w}(t), \mathbf{u}(t)) := l(\mathbf{f}_{\mathrm{W}}(\mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{w}(t)), \mathbf{u}(t), t)$ in the integral,

we obtain

$$\begin{aligned} \min_{\mathbf{u}(\cdot),\mathbf{w}(\cdot)} \quad & \Phi(\mathbf{x}(t_f),t_f) + \int_{t_0}^{t_f} \tilde{l}(t,\mathbf{x}(t),\mathbf{w}(t),\mathbf{u}(t))\ dt \\ \text{s.t.} \quad & \mathbf{w}(t) = \mathbf{f}_{\mathrm{H}}(\mathbf{u}(t)), \quad \forall t \in [t_0,t_f], \end{aligned} \tag{8}$$

where $\mathbf{x}(\cdot)$ is the solution of

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{w}(t), \quad \forall t \in [t_0,t_f] \\ \mathbf{x}(t_0) &= \mathbf{x}_0. \end{aligned}$$

In Section 3, we show how Problem (8) can be used to derive a solution strategy for deterministic global dynamic optimization with HW models embedded. In more detail, we are concerned with the derivation of an algorithm that is guaranteed to terminate finitely with an ϵ-optimal $\mathbf{u}^*(\cdot)$, $\mathbf{w}^*(\cdot)$ to Problem (8).

Note that, in contrast to the inversion approach resulting in Problem (6), the additional optimization approach solves the ODE and the equation of the Hammerstein part simultaneously.

3 Solution Strategy

In this section, we present theory and implementation of the additional optimization variables approach solving Problem (8).

As the decision variables associated with this problem refer to continuous control inputs $\mathbf{u}(\cdot)$, we first apply control parametrization to Problem (8) and then derive an algorithm to solve the parametrized problem to global optimality. Therefore, we need to parametrize the control functions $\mathbf{u}$, $\mathbf{w}$. An obvious choice is to use piecewise constant discretization for both and impose their nonlinear relationship at the discretization points. Other choices are conceivable as well, e.g., using a piecewise linear approximation. However, these choices may yield

additional complications and are out of the scope of this article. Note that the solution of the parametrized problem instead of infinite dimensional Problem (8) introduces an additional parametrization error. A method for overcoming this limitation has been recently proposed by Houska and Chachuat [12]. Nevertheless, the implementation and application of a rigorous method for control parametrization is beyond the scope of the present study.

3.1 Theory

Herein, we present the theory for systems with one input ($n_u = 1,\ n_w = 1$) and one output y, for notational simplicity. However, the methodology presented here can be extended for systems with multiple input/output signals with no significant changes.

The discretized input vectors are

$$u(t) = \hat{u}_i,\ \text{if } t \in [t_{i-1}, t_i),\ i = 1, ..., n,$$
$$w(t) = \hat{w}_i,\ \text{if } t \in [t_{i-1}, t_i),\ i = 1, ..., n,$$

with n discretization points and parameter vectors

$$\hat{\mathbf{u}} = \begin{pmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_n \end{pmatrix} \in \mathbb{R}^n, \quad \hat{\mathbf{w}} = \begin{pmatrix} \hat{w}_1 \\ \vdots \\ \hat{w}_n \end{pmatrix} \in \mathbb{R}^n.$$

Hence, we obtain an optimization problem with a finite number of variables and an ODE embedded

$$\begin{aligned} \min_{\hat{\mathbf{u}}, \hat{\mathbf{w}}} \quad & \Phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \tilde{l}(t, \mathbf{x}(t), \hat{\mathbf{w}}, \hat{\mathbf{u}})\ dt \\ \text{s.t.} \quad & \hat{\mathbf{w}} = \mathbf{f}_{\mathrm{H}}(\hat{\mathbf{u}}), \end{aligned} \tag{9}$$

where $\mathbf{x}(\cdot)$ is the solution of

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\sum_i \hat{w}_i \cdot \mathbb{1}_{[t_{i-1},t_i)}(t), \quad \forall t \in [t_0, t_f] \\ \mathbf{x}(t_0) &= \mathbf{x}_0, \end{aligned} \tag{10}$$

and $\mathbb{1}_{[t_{i-1},t_i)}(\cdot)$ is the indicator or characteristic function defined as

$$\mathbb{1}_{[t_{i-1},t_i)}(t) = \begin{cases} 1, & \text{if } t \in [t_{i-1}, t_i) \\ 0, & \text{if } t \notin [t_{i-1}, t_i). \end{cases}$$

Note that the same discretization is applied to both $w(t)$ and $u(t)$, such that the constraint $\hat{\mathbf{w}} = \mathbf{f}_{\mathrm{H}}(\hat{\mathbf{u}})$ can be understood as component-wise equality. In particular, this constraint is only enforced at a finite number of points.

Since Problem (9) contains a finite number of optimization variables, a standard spatial B&B algorithm can be employed. Any feasible point or local solution of Problem (9) constitutes an upper bound. A lower bound can be obtained by solving a convex relaxation of Problem (9). A convex relaxation of Problem (9) is derived in Theorem 1, which is built on the theory presented by Singer and Barton [34, 35]. Note that Theorem 1 follows the notation presented in [34, 35], and thus an explicit dependence of the states $\mathbf{x}$ also on the control parameters $\hat{\mathbf{w}}$ is included.

Theorem 1 *Let $\hat{\boldsymbol{u}} \in U$, where U is a convex subset of $\mathbb{R}^{n_u}$, $\hat{\boldsymbol{w}} \in W$, where W is a convex subset of $\mathbb{R}^{n_w}$, $\mathbf{x}(t, \hat{\boldsymbol{w}}) \in X$, where X is a convex subset of $\mathbb{R}^{n_x}$, such that $\mathbf{x}(t, \hat{\boldsymbol{w}}) \in X \ \forall \ (t, \hat{\boldsymbol{w}}) \in (t_0, t_f] \times W$; $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_w}$ constant matrices; Φ^{cv} a convex relaxation of Φ, both $\Phi, \Phi^{cv} : X \times \mathbb{R} \to \mathbb{R}$ continuous mappings; $\tilde{l}^{cv}$ a convex relaxation of $\tilde{l}$; $\tilde{l}^{cv}, \tilde{l} : (t_0, t_f] \times X \times U \times W \to \mathbb{R}$ Lebesgue integrable, where function $\tilde{l}$ is only permitted a finite number of discontinuities; $\mathbf{f}_H^{cv}$ a convex relaxation of $\mathbf{f}_H$; $\mathbf{f}_H^{cc}$ a concave relaxation of $\mathbf{f}_H$; $\mathbf{f}_H^{cv}, \mathbf{f}_H^{cc}, \mathbf{f}_H : U \to W$ continuous mappings with only finite number of discontinuities; and $\mathbb{1}_{[t_{i-1},t_i)}$ the indicator function. Then a convex relaxation of optimization Problem (9) is*

given by

$$\begin{aligned} \min_{\hat{\boldsymbol{u}},\hat{\boldsymbol{w}}} \quad & \Phi^{cv}(\mathbf{x}(t_f,\hat{\boldsymbol{w}}),t_f) + \int_{t_0}^{t_f} \tilde{l}^{cv}(t,\mathbf{x}(t,\hat{\boldsymbol{w}}),\hat{\boldsymbol{u}},\hat{\boldsymbol{w}})\ dt \\ \text{s.t.} \quad & \mathbf{f}_H^{cv}(\hat{\boldsymbol{u}}) \leq \hat{\boldsymbol{w}} \leq \mathbf{f}_H^{cc}(\hat{\boldsymbol{u}}), \end{aligned} \tag{11}$$

where $\mathbf{x}(\cdot,\hat{\boldsymbol{w}})$ *is the solution of*

$$\dot{\mathbf{x}}(t,\hat{\boldsymbol{w}}) = \mathbf{A}\mathbf{x}(t,\hat{\boldsymbol{w}}) + \mathbf{B}\sum_i \hat{w}_i \cdot \mathbb{1}_{[t_{i-1},t_i)}(t), \quad \forall t \in [t_0,t_f]$$

$$\mathbf{x}(t_0,\hat{\boldsymbol{w}}) = \mathbf{x}_0(\hat{\boldsymbol{w}}).$$

Proof A relaxation of the optimization Problem (9) can be derived by relaxing the objective function and the constraints.

Due to our specific problem formulation, which adds additional optimization variables besides $u(\cdot)$, we can apply the relaxation theory described in [34, 35] for systems with embedded linear dynamics, and therefore obtain a valid relaxation for the objective function. For the point term in the objective, the relaxation can be derived via standard techniques. For the integral term in the objective, integral relaxation (Corollary 3.1 in [35]) follows directly from integral monotonicity (Lemma 3.2 in [35]) and integral convexity (Theorem 3.1 in [35]). More precisely, we relax the objective with respect to $\hat{\mathbf{w}}$, imposing convexity of $\tilde{l}^{cv}$ on both $\hat{\mathbf{w}}$ and $\hat{\mathbf{u}}$.

Up to now, we have a methodology for deriving a relaxation for the objective function including the linear system dynamics. In our problem formulation, there is an additional constraint that relates $\hat{\mathbf{w}}$ and $\hat{\mathbf{u}}$. Relaxations of this constraint can be also obtained via standard techniques. With this, Problem (11) provides a valid relaxation of Problem (9). □

In Theorem 1, standard techniques for relaxations of the point term in the objective as well as the additional constraint refer to any valid relaxation methods for nonconvex functions, e.g., αBB [1] or McCormick [22] relaxations.

Note that integral relaxation following from Corollary 3.1 in [35] requires convexity of the relaxation of the integrand function on the controls. Assuming convexity of the relaxation on both $\hat{\mathbf{u}}$ and $\hat{\mathbf{w}}$, the relaxation of the objective function accounting for the linear dynamics with respect to $\hat{\mathbf{w}}$ follows directly from the theory presented in [34, 35].

3.2 Implementation

In the following, we discuss implementation aspects for the numerical solution of Problem (8). As discussed, Problem (8) is infinite dimensional, and thus the first step to apply the solution strategy presented above, is to parametrize the controls by piecewise constant functions. To numerically solve the resulting Problem (9), we utilize our open-source optimization software MAiNGO [5], based on (multivariate) McCormick relaxations [22, 38] and their subgradient propagation [23] implemented in MC++ [7]. MAiNGO is a deterministic global optimization software for solving mixed-integer nonlinear programs (MINLPs). Hence, to deal with the dynamic nature of our system, we first apply full discretization to the dynamics and then solve the resulting large scale NLP in a reduced space, using a spatial B&B algorithm, as shown in [23]. The reduced-space formulation treats only the values of the controls at all control discretization points as optimization variables. Note that this solution approach could also be understood as a single shooting method with a simple integration scheme, where the states are thus hidden from the optimizer.

The proposed solution approach offers numerical advantages compared to solving a full-space formulation of the NLP resulting from full discretization, i.e., treating also the values of the states at all discretization points as optimization variables and the integration scheme as constraints. This is because the reduced-space formulation dramatically reduces the number of considered optimization variables. However, since we only relax the parametrized problem, we actually optimize an approximate problem, and therefore we introduce an additional inherent error to the solution. This is different from the solution ap-

proach presented in [34, 35], where the authors discretize the relaxed problem. Therefore, by using tight discretization tolerances they can guarantee convergence to the ϵ-optimal solution of the original problem. A rigorous approach to account for truncation error following a discretize-then-relax fashion has been developed by the authors in [28].

For the numerical solution of the ODE, we implement the explicit Runge Kutta schemes up to 4th order. As commonly done in the literature, e.g., [36], we treat the objective as an ODE by rewriting to:

$$\Phi(\mathbf{x}(t_f), t_f) + k(t_f),$$

where

$$\begin{aligned} \dot{k}(t) &= \tilde{l}(t, \mathbf{x}(t), \mathbf{w}(t), \mathbf{u}(t)), \quad \forall t \in [t_0, t_f] \\ k(t_0) &= 0. \end{aligned}$$

Upon numerical integration of the ODE with its initial condition over $t \in [t_0, t_f]$, the original objective function is obtained. Thus, Problem (9) becomes

$$\begin{aligned} \min_{\hat{\mathbf{u}}, \hat{\mathbf{w}}} \quad & \Phi(\mathbf{x}(t_f), t_f) + k(t_f) \\ \text{s.t.} \quad & \hat{\mathbf{w}} = \mathbf{f}_{\mathrm{H}}(\hat{\mathbf{u}}), \end{aligned}$$

where $\mathbf{x}(\cdot)$ is the solution of the ODE system (10).

To achieve an accurate evaluation of the objective function and avoid excessive computational effort due to a large number of control parameters, a denser time discretization for the state grid and a coarser for the control grid might be required in practice. For this, we calculate the (piecewise constant) controls for the ODE and intermediate values for the states within the intervals of the control grid. This enables different time discretization for the states and the controls.

MAiNGO solves the above problem without the introduction of auxiliary

variables, thus only operating in the variable space $\hat{\mathbf{u}}, \hat{\mathbf{w}}$. To solve the optimization problem in MAiNGO, we require bounds for the controls $\hat{\mathbf{u}}, \hat{\mathbf{w}}$ and initial conditions at $t = t_0$ for the states. Yet, the optimizer does not directly see the state variables. Hence, bounds for the states are not required, since they are propagated along with the relaxations. Note that depending on the number of steps used in the integration scheme as well as the nonlinearities of the underlying model, the interval bounds of the intermediate variables computed during the propagation of relaxations may become extremely large similar to the work presented in [34, 35]. We come back to this issue in Section 4.2.1.

4 Case Studies

We demonstrate the feasibility of the presented approach by examining the solution of some numerical case studies. For all case studies presented below, the explicit Runge Kutta scheme of 4th order (ERK4) is applied as integration scheme, and equidistant grids are used. All computations are performed on a desktop computer with an Intel(R) Core(TM) i3-4150 CPU @ 3.50 GHz with 8GB RAM. We use MAiNGO 0.2.0 [5] with default settings unless otherwise stated. CPLEX 12.8.0 is used to solve the lower bounding problems, SLSQP [17] through the NLOPT 2.5.0 toolbox [13] for the upper bounding problems, and IPOPT 3.12.12 [39] is used for preprocessing. Model implementations for these cases studies are provided as Supplementary Information.

The examples presented in this section are similar or somewhat larger than what has been presented in the literature. In general, most studies reporting on global optimization of problems with nonlinear dynamics apply their theory to solve parameter estimation problems, cf., e.g., [10, 26, 37, 19, 23, 9]. The vast majority of these problems are solved for relatively small time horizons (t_f below ten), one to three states and less or equal to five time invariant control parameters. Wilhelm et al. [42] present a global optimization method for initial value problems of stiff parametric ODE systems, and the examples include up to

ten states (and consequently ten initial value parameters). Only a few studies on nonlinear global dynamic optimization for optimal control problems are reported in the literature, cf., e.g., [25, 20, 36]. All optimal control examples presented in these studies include one time variant control with up to five intervals, one to three states and time horizons less or equal than ten. As no open implementation of existing approaches exists, we do not compare them on our computer. Also, we do not attempt any comparison of the CPU times reported for these problems in the original works, since computational power has improved drastically over the past 15 years.

4.1 Case Study 1: A Simple Numerical Example

As a first case study, we consider an extension of Problem 5.4 presented in [34]. The optimization of the original problem is

$$\min_{w(t)\in[-4,4]} \quad \int_0^1 -x^2(t)\ dt, \tag{12}$$

where $\mathbf{x}(\cdot)$ derives from the solution of

$$\begin{aligned} &\dot{x}(t) = -2x(t) + w(t), \quad \forall t \in [0,1] \\ &x(0) = 1. \end{aligned} \tag{13}$$

The objective value as a definite integral in dependence of the parameter w for a completely constant discretization is depicted in Figure 2. From Figure 2 we can see the existence of two local minima, a suboptimal local one at $w = -4$ and a global one at $w = +4$. Depending on the starting point, a local optimizer may converge to the suboptimal local solution.

This problem is a special case of a HW model, as it can be formulated as a Wiener model with $A = -2$, $B = 1$, $C = 1$, $D = 0$ the LTI system's matrices, $x_0 = 1$ and $f_{\mathrm{W}}(z(t)) = -z^2(t)$. Note that these models have a linear correlation between the inputs and the states, since the input nonlinearity deriving from the Hammerstein function f_H is omitted. In the extension, we consider a nonlinear

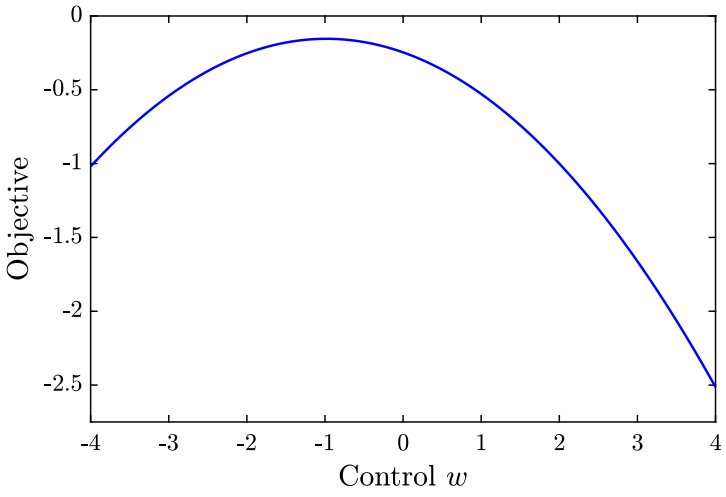


Fig. 2: Objective function for case study 1 (Figure adjusted from [34])

input-states correlation, by adding a nonlinear static function f_{H} that maps $u(\cdot)$ to $w(\cdot)$ to the system (Hammerstein block). For this, we define the function $f_{\mathrm{H}}(u(t)) = -u^2(t) + 5$, $u(t) \in [1, 3]$.

The resulting HW model can be described by the following system of equations

$$
\begin{aligned}
& w(t) = -u^2(t) + 5, \quad \forall t \in [0, 1] \\
& \dot{x}(t) = -2x(t) + w(t), \quad \forall t \in [0, 1] \\
& z(t) = x(t), \quad \forall t \in [0, 1] \\
& y(t) = -z^2(t), \quad \forall t \in [0, 1] \\
& x(0) = 1.
\end{aligned}
$$

Following the additional optimization variables approach presented in Section 2.2.4, we can now formulate our HW optimization problem as

$$
\begin{aligned}
\min_{u(\cdot), w(\cdot)} \quad & \int_0^1 -x^2(t)\, dt \\
\text{s.t.} \quad & w(t) = -u^2(t) + 5, \quad \forall t \in [0, 1],
\end{aligned}
\tag{14}
$$

where $\mathbf{x}(\cdot)$ is the solution of (13) and $u(t) \in [1, 3]$, $w(t) \in [-4, 4]$ for any $t \in [0, 1]$.

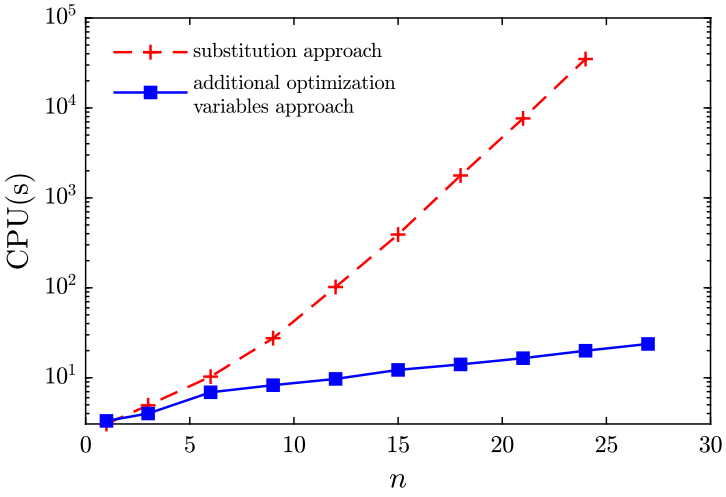


Fig. 3: Optimization results for case study 1; Computational performance as a function of the number of control discretization points n for the additional optimization variables approach and the substitution approach.

Problem (14) is equivalent to Problem (12) with respect to w. Therefore, it is obvious that since the global solution to Problem (12) is $w = +4$, the global solution to our Problem (14) will be $u = +1, w = +4$.

To numerically solve this problem, we apply the solution strategy presented in Section 3. Figure 3 illustrates the solution times when solving Problem (12) with MAiNGO, using different numbers of discretization points n for the controls. For comparison, in Figure 3 we also present the results with the substitution approach (see Section 2.2.2, Problem (4)), which results in a nonlinear dynamic system. Problem (14) following the substitution approach is also solved in MAiNGO, with the difference that in this case only u is a control. Consequently, the mapping from $u(\cdot)$ to $w(\cdot)$ through the Hammerstein function is now directly included in the dynamics, as shown in (5). Relative and absolute optimality tolerances for both solution approaches are set to 10^{-3}. A fine state grid with 1500 time intervals is applied. Thus, the integration tolerance is estimated to be substantially smaller than the optimization tolerance.

All results indeed give an objective value of $\approx$ -2.516 and return as optimal controls $(\hat{\mathbf{u}}, \hat{\mathbf{w}}) = (1, 4)$, where $\hat{\mathbf{u}}$ and $\hat{\mathbf{w}}$ are n-dimensional vectors with all

entries 1 and 4, respectively. In the substitution approach, the solution time scales unfavorably with refining control parametrization (with n=27, CPU time reached our imposed time limit of 12 h). As shown in Figure 3 the scaling in the additional optimization variables approach is more favorable than the substitution approach. Already for this simple example, the benefits of the proposed solution approach are underlined. For the examined case study, the computational time is observed to scale linearly with the state discretization for both approaches (see Appendix A).

Note again that for the numerical solution presented in [34], Problem (12) is first relaxed and then discretized. In contrast, for our customized Problem (14), we first discretize and then relax the dynamics, as in [23]. Although theoretically our implemented method introduces an additional optimization error (see relative discussion in Section 3.2), by imposing a fine state grid, we obtain the same objective value as in [34].

4.2 Case Study 2: Tracking Problem

As a second case study, we consider a tracking problem presented by Ławryńczuk [18]. In particular, our aim is to find the optimal $u(\cdot)$ to minimize the summed squared error between the output $y(\cdot)$ and an arbitrary chosen set-point trajectory $y_{sp}(\cdot)$. The examined system was first presented by Zhu [44] and then used by Ławryńczuk [18] for nonlinear model predictive control (NMPC). Herein, we aim at solving the problem to global optimality for the first time. We consider the following problem formulation

$$\begin{aligned} \min_{u(\cdot),w(\cdot)} \quad & \sum_{j=1}^{n_t} (y(t_j) - y_{sp}(t_j))^2 \\ \text{s.t.} \quad & w(t) = \frac{u(t)}{\sqrt{(0.1 + 0.9u(t))^2}}, \quad \forall t \in [t_0, t_f], \end{aligned} \tag{15}$$

where $u(t) \in [-2.5, 2.5]$, $w(t) \in [-1.045, 1.045]$ for any $t \in [t_0, t_f]$, $t_j = j$ for $j = 1, ..., n_t$, $y(t) = z(t) + 0.2z^3(t)$ for all $t \in [t_0, t_f]$ and $\mathbf{x}(\cdot)$, $z(\cdot)$ deriving from

the solution of the LTI:

$$\dot{x}_1(t) = 0.75x_1(t) + 0.87x_2(t) + 0.58w(t), \quad \forall t \in [t_0, t_f]$$
$$\dot{x}_2(t) = 1.24x_1(t) + 1.11x_2(t) - 0.66w(t), \quad \forall t \in [t_0, t_f]$$
$$z(t) = 0.5x_1(t) + 0.25x_2(t), \quad \forall t \in [t_0, t_f]$$
$$x_1(t_0) = x_{1,t_0}, \qquad x_2(t_0) = x_{2,t_0}.$$

The bounds on $w(\cdot)$ follow naturally from plotting w as a function of u. Note that having tight bounds on w can have a significant effect on numerical performance. The transformation of the discrete transfer function describing the LTI in [44] to continuous state space formulation in Problem (15) was derived in MATLAB [21]. It is worth noticing that the objective function for this example only contains fixed time points. Although herein we preserve the formulation presented in the literature [18], we could easily generalize it to an integral objective.

For all the results presented below, the relative and absolute optimality tolerances are set to 10^{-2}. In order to improve tightness of the relaxations and ultimately the convergence of the B&B algorithm, we implemented the convex and concave envelopes of the univariate Hammerstein function $f_{\mathrm{H}} : \mathbb{R} \to \mathbb{R}, f_{\mathrm{H}}(u) = \frac{u}{\sqrt{(a+bu)^2}}$ for a fixed a, b. For the calculation of the envelopes we use the method presented in Section 4 of [22]. Furthermore, setting higher branching priorities, i.e., branch on specific variables more often than on others during the B&B procedure, can have a significant effect on computational performance. Particularly for this problem, we used higher branching priorities on w ($BP_w = 5$), as we observed that this leads to reduced CPU times.

4.2.1 Offline Optimization

We first solve Problem (15) offline for $t_0 = 0$, $t_f = n_t = 120$ and $x_{1,t_0} = 0$, $x_{2,t_0} = 0$. Note that since the objective in Problem (15) requires function evaluations at 120 points, the state grid should be at least that fine. The

number of intervals in the state grid is set to 480. The grid resolution is decided in such a way that for all examined cases after doubling the discretization of the state grid the obtained relative difference in the objective is less than the optimization tolerance.

We perform the optimizations for different control parameterizations using the additional optimization variables approach. From Figure 4, we observe that the CPU time scales exponentially with the number of control discretization points n. Already for a control grid with ten intervals, the problem requires more than 12 h CPU time to converge to the optimal solution. An alternative approach to deal with this limitation is discussed in the next subsection. We observed a linear scaling of computational time with respect to state discretization, see Appendix A, Figure 9.

Note that the time of set-point changes in the output trajectory (except the first one occurring shortly after $t = 0$), see Figure 6b, coincides with the control steps for the case of an equidistant control grid with four parameters. Therefore, the choice of four control parameters or its multiples, leads to better objective values compared to other numbers of control parameters, as illustrated in Figure 5. Nevertheless, spotting physically superior solutions for Problem (15) is not the primary focus of this study, so this effect is not further discussed.

4.2.2 Nonlinear Model Predictive Control

As a next step, we extend our optimization algorithm to solve the tracking problem with an NMPC strategy. Problem (15) is solved repeatedly for each sampling instant $i \in [1, 120]$, for a prediction horizon N and a control horizon N_u in time, with $t_{0,i} = i - 1$, $t_{f,i} = t_{0,i} + N$, $n_{t,i} = N$ and initial states given by Equation (16). From the N_u elements, indicating the number of control parameters that are determined in each iteration, only the first one is implemented as action to the NMPC scheme. Then, the prediction is shifted one step forward and the process is repeated. At each time instant, from the time between the end of the control horizon until the end of the prediction horizon

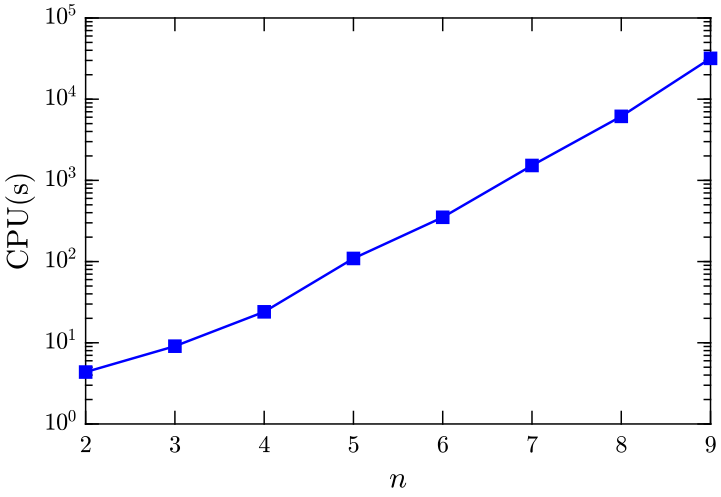


Fig. 4: Case study 2 offline optimization; Comparison of CPU times with the additional optimization variables approach for different numbers of control discretization points n.

zero incremental change in the control signal is considered.

$$\mathbf{x}_{t_0,i} = \begin{cases} \mathbf{0}, & \text{if } i = 1 \\ \mathbf{x}_{t_f,i-1}, & \text{otherwise .} \end{cases} \tag{16}$$

For each iteration i, a state grid with four times the number of intervals used for the prediction horizon N is required, in order to obtain the same final discretization as with the offline approach (i.e., $n = 480$). Unlike what is presented in [18], we do not consider an additional term in the objective function to penalize excessive control incremental changes. This is done to maintain the same objective with the offline approach and be able to compare the results. Therefore, a relative aggressive control scheme is obtained, see Figure 6a.

During the propagation of state values through time, the constructed relaxations may become extremely loose. In this particular case, such a behavior is observed. We anticipate that this might be due to the shorter time intervals that the controls are applied. This enables the control profile to fluctuate more, which leads to a higher flexibility on the potential state values. As the derived

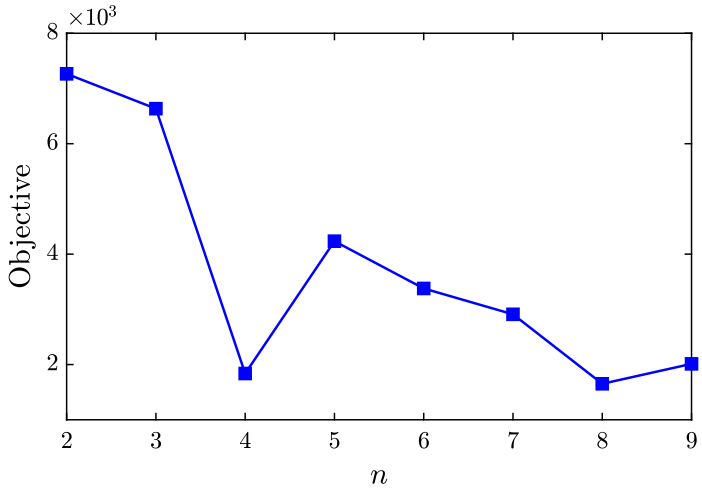


Fig. 5: Case study 2 offline optimization; Comparison of optimization results in terms of objective value with the additional optimization variables approach for different numbers of control discretization points n.

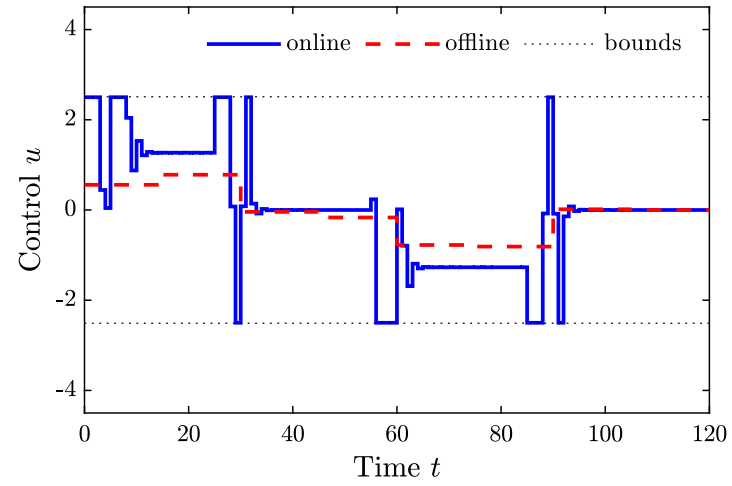
relaxations need to encompass the whole admissible range of the control profile, relaxations get weaker. Different methods to provide tight bounds for the states in parametric ODEs, cf., e.g., [27, 28, 33, 31] have been presented in the literature. However, when the state explosion derives from the enlargement of the admissible set of the state values, the improvements obtained from the tighter relaxations might be secondary. Herein, to avoid the explosion of state values we only consider additional bounds for the states $\mathbf{x}(t) \in [-10, 10] \times [-10, 10]$, the output of the LTI system $z(t) \in [-5, 5]$, as well as for the system output $y(t) \in [-30, 30]$ for all $t \in [t_0, t_f]$. The values on the domain of y are obtained by doubling the range of the desired output trajectory, on z by the functional dependence f_{W} between z and y, and on $\mathbf{x}$ by observing the systems behavior for the given control bounds. Note that these bounds do not need to be exact. The bounds are imposed through inequality constraints and the ranges of the corresponding functions are restricted to these bounds using the min and max functions before passing them as arguments to further computations.

Results for different control and prediction horizons, as well as both apriori known and unknown set-point changes are presented in Table 1. From Table 1,

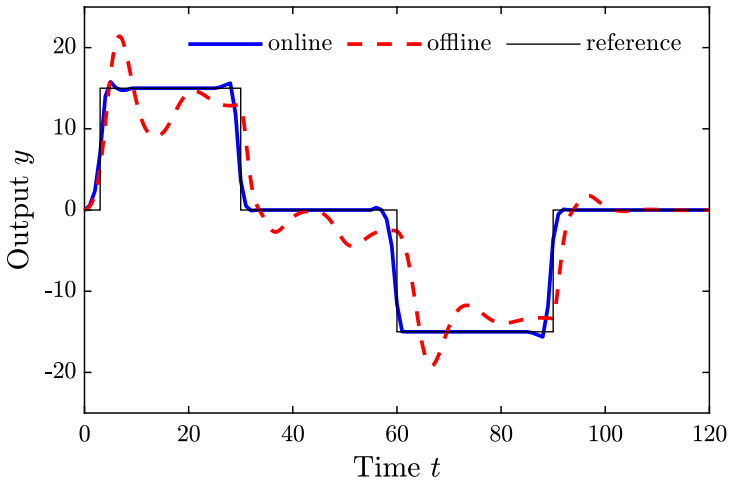
we observe that the consideration of the prediction for the set-point change has a drastic effect on the objective function. However, as the prediction horizon increases, the effect of the first control parameter, which is the one we actually implement after each iteration, decreases. Thus, the derived control policy becomes less effective. As the number of control parameters per iteration increases, the prediction generally improves. Yet, in this approach this does not have such a profound influence on the final objective value, as we only apply the first control element each time. Still, by increasing the number of control parameters for each iteration, the computational time increases significantly. In general, obtaining good values for control and prediction horizons is part of tuning in an NMPC problem and is considered out of the scope of the present study.

Note that the worst CPU times presented in Table 1 occur to iterations close to the set-point changes. For the other iterations the CPU times are considerably lower. Note also that in Table 1 the objective values derive from evaluation of the objective function in Problem (15) for the 120 instances that the controls were implemented. This enables the direct comparison of the objective values with the ones obtained from the offline optimization, as discussed in Section 4.2.3.

For completeness, we have compared our global solution of the online case with $N = 5$ and $N_u = 3$ (depicted in Figure 6) with a local solution, and these coincide. This can be due to a good starting point of the local approach, a large area of attraction for the global solution, or simply an indication that the examined problem is not multimodal. Obtaining the global solution was around three times slower compared to the time of the local solution for this example. This can be considered as a very good performance for global optimization. Although local solutions are in general computationally more tractable, our method has the significant advantage that it guarantees that the obtained solution is globally optimal. Unfortunately, in [44, 18] the time step is nondimensional, so that we cannot compare time in the considered system to CPU time for our solution



(a)



(b)

Fig. 6: Optimization results for case study 2 with the additional optimization variables approach; Comparison of online ($N_i = 5$, $N_{u,i} = 3$, $i = 1 \dots 120$) and offline ($n = 8$) optimization in terms of (a) control profile and (b) output trajectory.

Table 1: Numerical results for case study 2 online optimization with the additional optimization variables approach, with prediction horizon N, control horizon N_u considering either apriori known or unknown set-point changes in the output trajectory. Worst time refers to the iteration of the NMPC problem that required the longest time to converge. Total time refers to the sum of the computational time of all iterations. All CPU times are in s.

N	**N_u**	**set-point changes**	**obj.**	**worst CPU time**	**total CPU time**
5	3	unknown	1874.27	1.7	74.5
5	3	known	151.23	1.4	70.4
5	4	known	150.57	2.8	96.2
5	5	known	150.40	6.8	136.2
10	3	unknown	1881.83	3.6	194.1
10	3	known	208.23	5.4	231.6
10	4	known	176.76	69.1	469.8
10	5	known	155.23	847.0	2760.3
15	3	unknown	1889.43	5.3	332.0
15	3	known	390.79	9.4	385.3
15	4	known	360.90	51.3	706.3
15	5	known	211.63	841.7	3588.1

approach, and thus we are not able to draw any conclusion about whether our approach is real-time capable for this example.

4.2.3 Comparison of Offline and Online Optimization

Overall, we observe that the NMPC scheme can obtain much better results in terms of both CPU time and objective value than the offline optimization. More precisely, the objective values for the NMPC with known set-point change are around one order of magnitude lower than the ones attained with offline optimization. Figure 6 illustrates two exemplary control and output trajectories obtained one from the offline and the other from the online approach. With the online approach, we are able to solve the tracking problem with 120 discretization points for each control u and w in a few minutes, and with each subproblem solved globally. In contrast to this, for the offline approach, we were limited to maximum nine points in the control grid, which took almost ten hours for the global optimization. The computational burden of the presented methodology scales in general unfavorably with increasing number of control

parameters. However, by following an online approach to solve the dynamic optimization problem globally, we avoid this limitation. More precisely, the repeated solution of small problems with few control intervals each in the online approach is much faster than the solution of one large problem in the offline approach. Since this observation is linked to the scaling of B&B algorithms with the number of variables, it likely extends to other global optimization approaches as well. As HW models are used in many applications in control, these results indicate great potential for applications in this field and can contribute substantial benefits in cases where the global solutions are necessitated.

4.3 Case Study 3: Monoclonal Antibodies Production

As a last case study, we consider an example motivated from antibody production [14, 15]. The HW model has two inputs, one output and six states. The LTI system is given by

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{w}(t), \quad t \in [0, 144] \\ z(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{w}(t), \quad t \in [0, 144] \\ \mathbf{x}(0) &= \mathbf{0}, \end{aligned} \tag{17}$$

where information about the system's matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ is given in Appendix B. The input and output nonlinearities are wavelet functions with one or two units and are also provided in Appendix B, see Equations (19)-(21).

The optimization problem with $\mathbf{x}(\cdot)$, $z(\cdot)$ deriving from (17) can be formulated as

$$\begin{aligned} \min_{\mathbf{u}(\cdot),\mathbf{w}(\cdot)} \quad & \int_0^{144} -f_W(z(t))\ dt \\ \text{s.t.} \quad & \mathbf{w}(t) = \mathbf{f}_H(\mathbf{u}(t)), \quad \forall t \in [0, 144] \\ & \sum_i \left((\hat{u}_{1,i} + \hat{u}_{2,i}) \cdot 10^{-4} \cdot \frac{144}{n} \right) - 0.02 \leq 0, \quad i = 1, ..., n, \end{aligned} \tag{18}$$

Table 2: Numerical results for case study 3 with the additional optimization variables approach with n number of control discretization points. All CPU times are in s.

n	**obj.**	**û**	**CPU time**
1	-14401.7	(1.389),(0.0)	32.7
2	-18728.9	(2.684, 0.093),(0.0, 0.0)	52.0
3	-18764.5	(2.681, 0.050, 0.597), (0.0, 0.0, 0.839)	345.0
4	-18852.7	(2.686, 2.683, 0.187, 0.0), (0.0, 0.0, 0.0, 0.0)	3577.3

where $\mathbf{u}(t) \in [0.0, 3.3]^2$, $\mathbf{w}(t) \in [0.6, 2.3] \times [0.35, 1.5]$ for any $t \in [0, 144]$ and n the number of control discretization points. The bounds on $\mathbf{w}$ follow from plotting $\mathbf{w}$ as a function of $\mathbf{u}$. The inequality constraint in Problem (18) provides an upper bound on the permissible control inputs $\mathbf{u}$.

We assume this problem to be multimodal, as different solutions are obtained when a multistart is performed. Thus, global optimization is particularly important. The optimization problem is solved for different numbers of control grid discretization and for a relative and absolute optimality tolerance of 10^{-2} and 20 local searches during preprocessing.

The results for $n = 1, 2, 3$, and 4 with a state grid with 288 intervals are presented in Table 2. Although both $\mathbf{u}(\cdot)$ and $\mathbf{w}(\cdot)$ are optimization variables, in Table 2, we only present (for compactness) the values of $\mathbf{u}(\cdot)$, which are the relevant ones for practical implementations. Interestingly, the objective value does not seem to be significantly affected by the discretization of the controls, taking also into account the imposed optimality tolerance. This can be due to different combinations within the imposed control bounds that can lead to same objective values. Note that due to the high nonlinearity of the static functions and the increased number of states, no convergence to global optimality was attained within 12 h CPU times for a control discretization greater than four. However, we should point out that already a control grid with four elements refers to eight controls for the problem, considering the two control inputs u_1, u_2. In our solution approach, we consider both $\mathbf{u}$ and $\mathbf{w}$ as

control variables, which translates to a total number of 16 optimization variables in Problem (18). Linear scaling with state grid refinement is again observed, see Figure 10 in the Appendix A.

5 Conclusions

Hammerstein-Wiener models are a commonly used class of block-structured models with a wide range of applications in process operations and control. As these models are nonlinear, they can lead to suboptimal local minima when embedding them in process optimization or control problems.

Herein, we propose a novel algorithm for deterministic global optimization with Hammerstein-Wiener models. We extend the theory presented in [34, 35] on global optimization of systems with linear dynamics to HW models. The theory pertains to combining direct methods with a spatial B&B algorithm to tackle dynamic problems based on extensions of sequential methods for local dynamic optimization. We show that different optimization problem formulations can lead to different solution strategies with different levels of difficulty. More precisely, by carefully selecting the optimization variables in the problem formulation, we are able to maintain advantageous properties of linear systems. In a next step, we successfully apply our method to numerical examples from offline and online optimization. For this we follow a discretize-then-relax fashion. The parametrized optimization problems are solved in a reduced space using our open-source global optimization software MAiNGO [5] based on McCormick relaxations [22].

The results demonstrate the potential benefits of the presented approach and enable future utilization to real-world case studies, with special focus on model predictive control. Our method seems to scale favorably with refining the states grid, but is more sensitive to the control grid. This is a typical problem for similar algorithms proposed in the literature as pointed out in [8]. To address this problem, future emphasis should be placed on methods for

obtaining tighter relaxations for the lower bounding problem, cf., e.g., [24]. Furthermore, consideration of sophisticated methods to construct tight state relaxations, cf., e.g., [27, 28, 33, 31] can yield considerable improvements to this work.

In general, due to the exponential worst-case runtime, it makes a profound difference for global optimization whether we solve one problem with a large number of control parameters, or multiple problems with fewer control parameters, although both problems may result in the same total number of control intervals. This work particularly emphasizes the applicability of our approach to NMPC problems, and potentially also of other global dynamic optimization approaches, since they can all benefit from short time intervals and few control parameters at each control iteration.

Acknowledgements

The authors gratefully acknowledge the financial support of the Kopernikus project SynErgie by the Federal Ministry of Education and Research (BMBF) and the project supervision by the project management organization Projektträger Jülich. Susanne Sass is grateful for her association to the International Research Training Group (DFG) IRTG-2379 Modern Inverse Problems. The authors would like to thank Adel Mhamdi, Yannic Vaupel, Jan Christoph Schulze and Adrian Caspari for fruitful discussions and suggestions throughout the development of this work.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Adjiman, C.S., Dallwig, S., Floudas, C.A., Neumaier, A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs—I. Theoretical advances. Comput. Chem. Eng. **22**(9), 1137–1158 (1998)

[2] Bai, E.W.: An optimal two-stage identification algorithm for Hammerstein–Wiener nonlinear systems. Automatica **34**(3), 333–338 (1998)

[3] Biegler, L.T.: Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes, vol. 10. SIAM, Philadelphia (2010)

[4] Bongartz, D., Mitsos, A.: Deterministic global optimization of process flowsheets in a reduced space using McCormick relaxations. J. Glob. Optim. **69**(4), 761–796 (2017)

[5] Bongartz, D., Najman, J., Sass, S., Mitsos, A.: MAiNGO: McCormick based Algorithm for mixed-integer Nonlinear Global Optimization. Tech. rep., Process Systems Engineering (AVT.SVT) (2018). http://permalink.avt.rwth-aachen.de/?id=729717(Accessed April 2020)

[6] Chachuat, B.: Nonlinear and dynamic optimization: From theory to practice. Tech. rep., Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne (2007)

[7] Chachuat, B., Houska, B., Paulen, R., Peri'c, N., Rajyaguru, J., Villanueva, M.E.: Set-theoretic approaches in analysis, estimation and control of nonlinear systems. IFAC-PapersOnLine **48**(8), 981 – 995 (2015). DOI https://doi.org/10.1016/j.ifacol.2015.09.097. URL http://www.sciencedirect.com/science/article/pii/S2405896315011787. https://omega-icl.github.io/mcpp/(Accessed February 2017)

[8] Chachuat, B., Singer, A.B., Barton, P.I.: Global methods for dynamic optimization and mixed-integer dynamic optimization. Ind. Eng. Chem. Res. **45**(25), 8373–8392 (2006)

[9] Čižniar, M., Podmajerskỳ, M., Hirmajer, T., Fikar, M., Latifi, A.M.: Global optimization for parameter estimation of differential-algebraic systems. Chemical Papers **63**(3), 274–283 (2009)

[10] Esposito, W.R., Floudas, C.A.: Deterministic global optimization in nonlinear optimal control problems. J. Glob. Optim. **17**(1-4), 97–126 (2000)

[11] Floudas, C.A., Gounaris, C.E.: A review of recent advances in global optimization. J. Glob. Optim. **45**(1), 3 (2009)

[12] Houska, B., Chachuat, B.: Branch-and-lift algorithm for deterministic global optimization in nonlinear optimal control. J. Optim. Theory Appl. **162**(1), 208–248 (2014)

[13] Johnson, S.G.: The NLopt nonlinear-optimization package (2014). http://ab-initio.mit.edu/nlopt(Accessed October 2016)

[14] Kappatou, C.D., Mhamdi, A., Campano, A.Q., Mantalaris, A., Mitsos, A.: Dynamic optimization of the production of monoclonal antibodies in semi-batch operation. In: Comput. Aided Chem. Eng., vol. 40, pp. 2161–2166. Elsevier (2017)

[15] Kappatou, C.D., Mhamdi, A., Campano, A.Q., Mantalaris, A., Mitsos, A.: Model-based dynamic optimization of monoclonal antibodies production in semibatch operation—Use of reformulation techniques. Ind. Eng. Chem. Res. **57**(30), 9915–9924 (2018)

[16] Kraft, D.: On converting optimal control problems into nonlinear programming problems. In: K. Schittkowski (ed.) Computational Mathematical Programming, pp. 261–280. Springer Berlin Heidelberg, Berlin, Heidelberg (1985)

[17] Kraft, D.: A software package for sequential quadratic programming. Tech. Rep. DFVLR-FB 88-28, Institut für Dynamik der Flugsysteme, Oberpfaffenhofen (1988)

[18] Ławryńczuk, M.: Nonlinear predictive control for Hammerstein–Wiener systems. ISA Trans. **55**, 49–62 (2015)

[19] Lin, Y., Stadtherr, M.A.: Deterministic global optimization for parameter estimation of dynamic systems. Ind. Eng. Chem. Res. **45**(25), 8438–8448 (2006)

[20] Lin, Y., Stadtherr, M.A.: Deterministic global optimization of nonlinear dynamic systems. AIChE J. **53**(4), 866–875 (2007)

[21] MATLAB: 9.6.0.1072779 (R2019a). The MathWorks Inc., Natick, Massachusetts (2019)

[22] McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. Math. Program. **10**(1), 147–175 (1976)

[23] Mitsos, A., Chachuat, B., Barton, P.I.: McCormick-based relaxations of algorithms. SIAM J. Optim. **20**(2), 573–601 (2009)

[24] Najman, J., Mitsos, A.: Tighter McCormick relaxations through subgradient propagation. J. Glob. Optim. **75**(3), 565–593 (2019)

[25] Papamichail, I., Adjiman, C.S.: A rigorous global optimization algorithm for problems with ordinary differential equations. J. Glob. Optim. **24**(1), 1–33 (2002)

[26] Papamichail, I., Adjiman, C.S.: Global optimization of dynamic systems. Comput. Chem. Eng. **28**(3), 403–415 (2004)

[27] Sahlodin, A.M., Chachuat, B.: Convex/concave relaxations of parametric ODEs using Taylor models. Comput. Chem. Eng. **35**(5), 844–857 (2011)

[28] Sahlodin, A.M., Chachuat, B.: Discretize-then-relax approach for convex/concave relaxations of the solutions of parametric odes. Appl. Numer. Math. **61**(7), 803–820 (2011)

[29] Scott, J.K.: Reachability analysis and deterministic global optimization of differential-algebraic systems. Ph.D. thesis, Massachusetts Institute of Technology (2012)

[30] Scott, J.K., Barton, P.I.: Convex and concave relaxations for the parametric solutions of semi-explicit index-one differential-algebraic equations. J. Optim. Theory Appl. **156**(3), 617–649 (2013)

[31] Scott, J.K., Barton, P.I.: Improved relaxations for the parametric solutions of ODEs using differential inequalities. J. Glob. Optim. **57**(1), 143–176 (2013)

[32] Scott, J.K., Barton, P.I.: Reachability analysis and deterministic global optimization of DAE models. In: Surveys in Differential-Algebraic Equations III, pp. 61–116. Springer (2015)

[33] Scott, J.K., Chachuat, B., Barton, P.I.: Nonlinear convex and concave relaxations for the solutions of parametric ODEs. Optim. Control Appl. Methods **34**(2), 145–163 (2013)

[34] Singer, A.B.: Global dynamic optimization. Ph.D. thesis, Massachusetts Institute of Technology (2004)

[35] Singer, A.B., Barton, P.I.: Global solution of optimization problems with parameter-embedded linear dynamic systems. J. Optim. Theory Appl. **121**(3), 613–646 (2004)

[36] Singer, A.B., Barton, P.I.: Global optimization with nonlinear ordinary differential equations. J. Glob. Optim. **34**(2), 159–190 (2006)

[37] Singer, A.B., Taylor, J.W., Barton, P.I., Green, W.H.: Global dynamic optimization for parameter estimation in chemical kinetics. J. Phys. Chem. A **110**(3), 971–976 (2006)

[38] Tsoukalas, A., Mitsos, A.: Convex relaxations of multi-variate composite functions. In: Comput. Aided Chem. Eng., vol. 32, pp. 385–390. Elsevier (2013)

[39] Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)

[40] Wang, D., Ding, F.: Extended stochastic gradient identification algorithms for Hammerstein–Wiener ARMAX systems. Comput. Math. Appl. **56**(12), 3157–3164 (2008)

[41] Wang, Z., Georgakis, C.: Identification of Hammerstein-Weiner models for nonlinear MPC from infrequent measurements in batch processes. J. Process Control **82**, 58–69 (2019)

[42] Wilhelm, M.E., Le, A.V., Stuber, M.D.: Global optimization of stiff dynamical systems. AIChE J. **65**(12), e16836 (2019)

[43] Wills, A., Schön, T.B., Ljung, L., Ninness, B.: Identification of Hammerstein–Wiener models. Automatica **49**(1), 70–81 (2013)

[44] Zhu, Y.: Estimation of an N–L–N Hammerstein–Wiener model. Automatica **38**(9), 1607–1614 (2002)

A Appendix 1- Effect of State Grid Refinement on the Solution Approach

Herein we present the computational performance of the presented solution approach by refining the state discretization for the examined case studies. Linear scaling is observed in all cases.

Figure 7 illustrates the results for case study 1 for both substitution and additional optimization variables approach, for an exemplary control grid with $n = 9$ intervals and piecewise constant control functions. Similar results are obtained also for the other values of n. As already indicated in Section 4, for both cases the results indicate a linear scaling with refining the state grid.

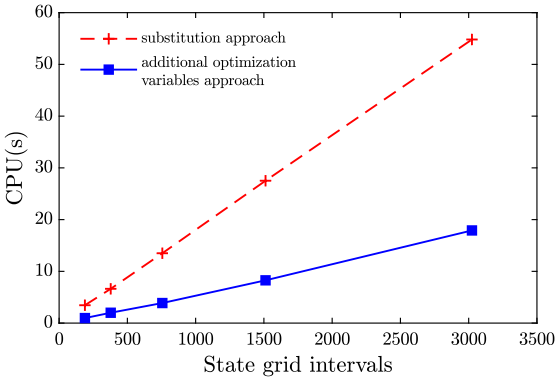


Fig. 7: Case study 1; Scaling of computational performance with refinement of state grid for a control grid with $n = 9$ intervals.

Similar results are obtained for case study 2, as shown in Figure 8 for the offline approach for three exemplary control grids. The results for the other values of n are analogous. In Figure 9, we additionally reproduce the results of Section 4.2.1 referring to the scaling of CPU time with number of control intervals for three different state grids (here the linear scaling corresponds to equidistant spacing between the different lines). Fine discretization refers to 480 intervals, and corresponds to the results presented in Figure 4. Medium discretization corresponds to a state grid with 240 intervals and coarse to 120 intervals, respectively. The results for the objective function are not presented

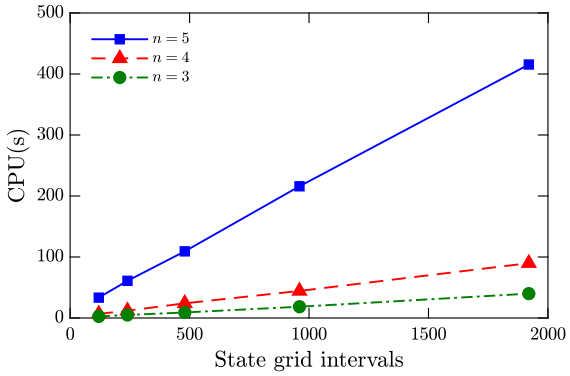


Fig. 8: Case study 2 offline optimization with the additional optimization variables approach; Scaling of computational performance with refinement of state grid for different control grids.

here, since by changing grid refinement the change in the objective value for constant number of control discretization is always less than 10 %.

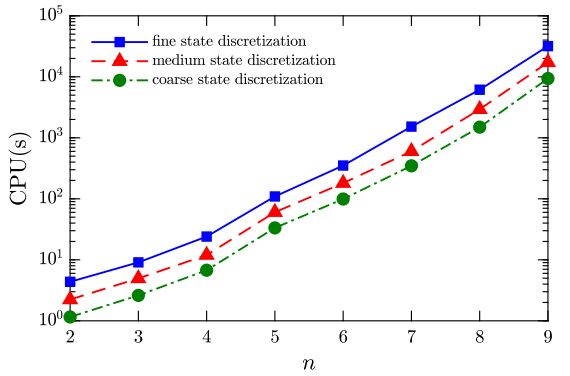


Fig. 9: Case study 2 offline optimization with the additional optimization variables approach; Scaling of computational performance with refinement of state grid for different numbers of control discretization points n.

For case study 3, also linear scaling with the number of state discretization points is observed. However, since this problem is strongly multimodal (multiple different objectives values are obtained from different local searches), solution times may also depend on how good is the initial upper- bound-guess that derives from the local solution of the examined optimization problem. The results for different state discretizations and numbers of control discretization points are illustrated in Figure 10. Fine discretization corresponds to a state grid with 288

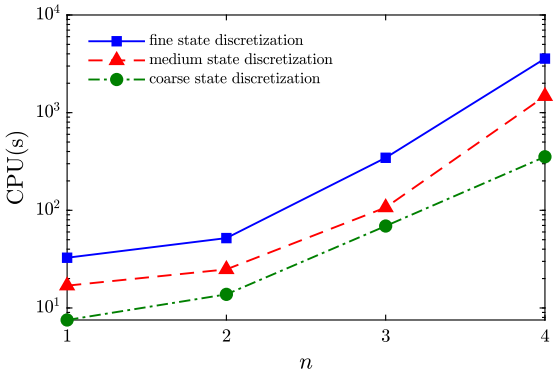


Fig. 10: Case study 3 with the additional optimization variables approach; Scaling of computational performance with refinement of state grid for different numbers of control discretization points, n.

intervals (results presented in Table 2), medium discretization to 144 intervals and coarse to 72 intervals. The state grid refinement for each of the different numbers of control intervals led to differences in the objective always within the optimization tolerance, and thus not presented here. As it can be seen in Figure 10, the CPU time scales unfavorably with increasing the number of control discretization points.

B Appendix 2- Additional Model Information for Case Study 3

The matrices of the LTI system as described in Equation (17) are given by

$$\mathbf{A} = \begin{bmatrix} 0.418 & -0.512 & 0.094 & 0 & 0 & 0 \\ 0.215 & -0.091 & -0.124 & 0 & 0 & 0 \\ -0.284 & 0.886 & -0.603 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.365 & -0.414 & 0.099 \\ 0 & 0 & 0 & 0.261 & -0.176 & -0.171 \\ 0 & 0 & 0 & -0.226 & 0.598 & -0.745 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.134 & 0 \\ -0.082 & 0 \\ 0.202 & 0 \\ 0 & 0.151 \\ 0 & -0.110 \\ 0 & 0.171 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} -0.933 & 1 & 0 & -0.929 & 1 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \end{bmatrix}.$$

The input nonlinearity f_{H1} is given by

$$f_{H1}(u_1(t)) = (u_1(t) - 1.521) \cdot 0.034 \cdot 10^{-4} + 0.852 \cdot 10^{-4} \cdot \big(1 - s_1^2(t)\big) \cdot \exp\big(-0.5 s_1^2(t)\big) \\ -0.629 \cdot 10^{-4} \cdot \big(1 - s_2^2(t)\big) \cdot \exp\big(-0.5 s_2^2(t)\big) + 1.339 \cdot 10^{-4}, \tag{19}$$

with $t \in [0, 144]$, $u_1(t) \in [0.0, 3.3]$ and

$$s_1(t) = 8 \cdot \big((u_1(t) - 1.521) \cdot 1072 \cdot 10^{-4} - 1.250\big)$$
$$s_2(t) = 2 \cdot \big((u_1(t) - 1.521) \cdot 1072 \cdot 10^{-4} + 1.001\big).$$

The input nonlinearity f_{H2} is given by

$$f_{H2}(u_2(t)) = (u_2(t) - 1.624) \cdot 0.039 \cdot 10^{-4} - 0.871 \cdot 10^{-4} \cdot \exp\big(-0.5 \cdot s^2(t)\big) \\ +1.302 \cdot 10^{-4}, \tag{20}$$

with $t \in [0, 144]$, $u_2(t) \in [0.0, 3.3]$ and

$$s(t) = 16 \cdot \big((u_2(t) - 1.624) \cdot 9689 \cdot 10^{-4} + 0.750\big).$$

The output nonlinearity f_W is given by

$$f_W(z(t)) = (z(t) - 0.001) \cdot 38898 + 78.733 \cdot \big(1 - s_3^2(t)\big) \cdot \exp\big(-0.5 s_3^2(t)\big) \\ +15.692 \cdot \big(1 - s_4^2(t)\big) \cdot \exp\big(-0.5 s_4^2(t)\big) + 89.140, \tag{21}$$

with $t \in [0, 144]$ and

$$s_3(t) = 1 \cdot ((z(t) - 0.001) \cdot 516.573 - 2.000)$$
$$s_4(t) = 4 \cdot ((z(t) - 0.001) \cdot 516.573 + 1.250).$$