Journal of Process Control 154 (2025) 103513

Contents lists available at ScienceDirect

Journal of Process Control

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

Review

Obscured by terminology: Hidden parallels in direct methods for open-loop optimal control

Susanne Sass[a], Alexander Mitsos[b,a,c,*]

[a] Process Systems Engineering (AVT.SVT), RWTH Aachen University, Aachen, 52074, Germany
[b] JARA-CSD, Aachen, 52056, Germany
[c] Institute of Climate and Energy Systems: Energy Systems Engineering (ICE-1), Forschungszentrum Jülich GmbH, Jülich, 52425, Germany

ARTICLE INFO

Keywords:
Dynamic optimization
Control vector parametrization
Collocation
Single shooting
Multiple shooting
Steady-state assumption

ABSTRACT

Active research on optimal control methods comprises the developments of research groups from various fields, including control, mathematics, and process systems engineering. Although there is a consensus on the classification of the main solution methods, different terms are often used for the same method. For example, solving optimal control problems with control discretization and embedded state integration may be called sequential method or direct single shooting. Equally severely, the same term may be used ambiguously: Is control vector parameterization a synonym for control discretization or for direct single shooting? Both misleading distinctions and ambiguity complicate the scientific discourse. Thus, we delineate standard terms from open-loop optimal control in this tutorial. More precisely, we formulate and challenge hypotheses on the terminology of direct methods, i.e., solution methods using control discretization combined with state integration and/or state discretization. In particular, we point out the parallel of the embedded state integration with a numerical integration scheme and the reduced-space formulation of approaches using state discretization. Taking a step further towards integrated scheduling and control problems, we additionally investigate the similarities and differences between the discrete-time solution of optimal control problems and optimal quasi-steady operation. In this context, we also hint on the discrete-time representation in scheduling which refers to the handling of controls rather than the handling of process dynamics. Rather than concluding with the "correct" term to use, this tutorial concludes with recommendations on how to avoid misunderstandings in the versatile research community.

Contents

[*] Correspondence to: RWTH Aachen University, Aachener Verfahrenstechnik (AVT), Process Systems Engineering (SVT), Forckenbeckstraße 51, D - 52074 Aachen, Germany.
E-mail address: amitsos@alum.mit.edu (A. Mitsos).

https://doi.org/10.1016/j.jprocont.2025.103513
Received 19 March 2025; Received in revised form 21 June 2025; Accepted 23 July 2025
Available online 24 August 2025

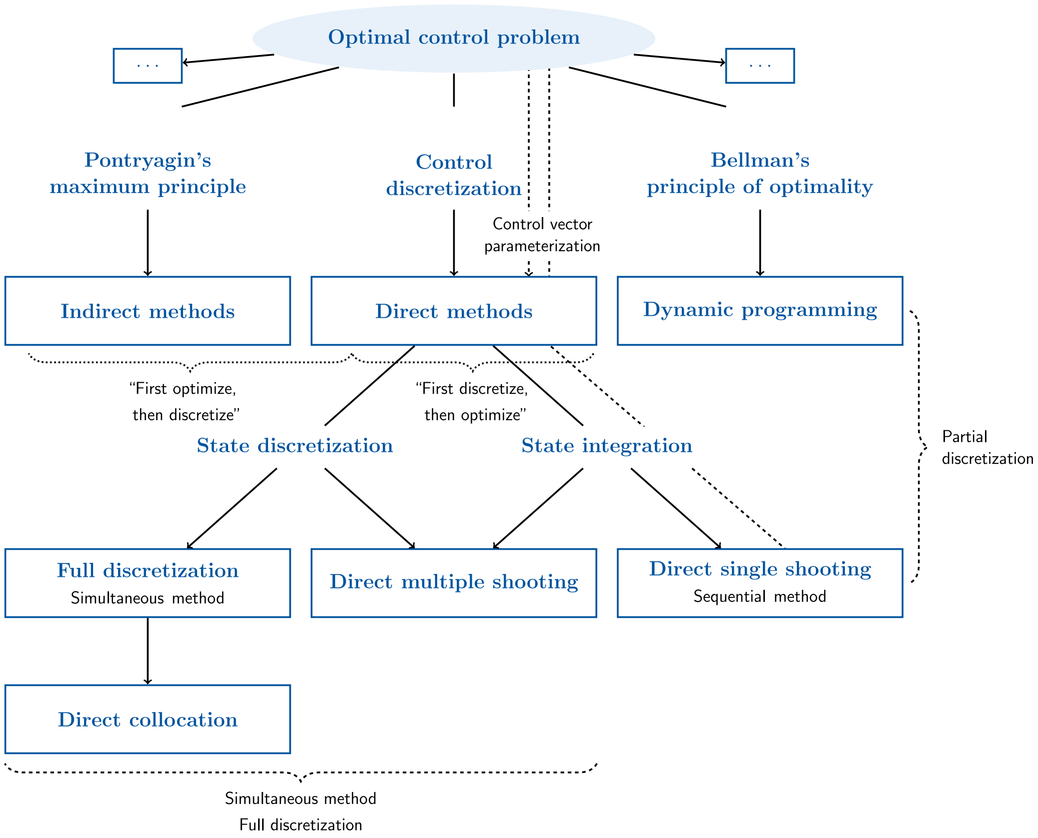


Fig. 1. Overview on standard methods for solving optimal control problems and their common terminology. The bold, colored terms are used throughout this tutorial.

1. Introduction

For dynamic processes, the future state of the system depends on the current state and control values. Such processes are typically modeled with differential equations. Consequently, dynamic optimization addresses optimization problems with differential or differential–algebraic equations (DAEs) embedded. We, in particular, focus on dynamic optimization problems determining control *functions*. Such infinite-dimensional dynamic optimization problems are also known as optimal control problems.

In our context, both terms "dynamic optimization problem" [e.g.,1–3] and "optimal control problem" [e.g.,4–9] are ambiguous: often, "dynamic optimization" is also used as an umbrella term including both finite dimensional problems like dynamic parameter estimation problems and infinite dimensional optimal control problems [e.g.,10,11]. Similarly, optimization with differential equations or DAEs embedded may comprise both finite and infinite dimensional problems. However, "optimal control" may refer to closed-loop control or model-predictive control, i.e., optimization problems with a strong focus on online applications and, typically, least-squares objectives [e.g.,12]. In this tutorial, we focus on the offline optimization of open-loop optimal control problems. In other words, our hidden conjecture is that the control functions, i.e., the system inputs which can be manipulated from a technical or human controller, depend on time but not explicitly on feedback from the system states.

Throughout this tutorial, we use an example rather than a general optimal control problem for our discussions. Without loss of generality, we refer to the optimal control problem

$$
\begin{aligned}
\min_{u(\cdot)} \quad & \int_0^{t_f} \mathcal{L}(x(t))\,\mathrm{d}t \\
\text{s.t.} \quad & \frac{\mathrm{d}x}{\mathrm{d}t}(t) = \frac{u(t)}{x(t)} \quad \forall t \in [0, t_f] \qquad \text{(OCP)} \\
& x(0) = x_0 \\
& x(t_f) = x_f
\end{aligned}
$$

with infinitely many optimization variables $u(t),\ t \in [0, t_f]$, a Lagrange-type objective function, a nonlinear but control-affine ordinary differential equation (ODE), state $x : [0, t_f] \to \mathbb{R}$, an explicitly known initial value x_0 and an endpoint constraint. The feasible control domain $u(t) \in [u^{\min}, u^{\max}] \subsetneq \mathbb{R},\ t \in [0, t_f]$ and the initial value x_0 are chosen such that we can ensure $x(t) > 0$ for any $t \in [0, t_f]$. Together with essentially bounded controls $u \in L^{\infty}([0, t_f])$, we obtain a well-posed optimization problem. Note that (OCP) is designed to be as simple as possible for showcasing hidden parallels between direct methods.

The standard approaches for solving optimal control problems like (OCP) are essentially, but not exclusively, based on three principles [e.g.,2,4,13,14], see Fig. 1: Firstly, Pontryagin's maximum principle gives indirect methods [15]. Secondly, control discretization yields direct methods. Thirdly, Bellman's principle of optimality results in dynamic programming [16,17]. There is a variety of further approaches out of our scope, e.g., solution approaches based on the calculus of variations [e.g.,18–20].

Literally, "indirect" and "direct" refers to the indirect optimization of the controls via derived optimization variables, namely the adjoints, or the direct determination of the (discretized) controls. Referring to the terminology used for optimal control subject to partial differential equations (PDEs), indirect and direct methods are therefore also called "First optimize, then discretize" and "First discretize, then optimize" methods, respectively [e.g.,21]. However, we can imagine a direct "First optimize, then discretize" method, namely when deriving exact sensitivity equations and, subsequently, applying control discretization. Moreover, "First optimize, then discretize" suggests that the boundary value problem obtained by Pontryagin's maximum principle (*optimize*) is solved with discretization methods. In principle, we may determine the control *function* via control vector iteration [e.g.,22] or, for comparatively simple problems, the analytical solution. In other words, it is a hidden conjecture that discretization is required for the solution of optimal control problems.

In the following, we formulate and challenge hidden conjectures for standard direct methods as hypotheses. We validate or disprove (parts of) each hypothesis based on the literature as well as reformulations showcased with (OCP). Our aim is to increase understanding, not to pass judgement. In line with the objectives of a tutorial, we recall the standard approaches of direct methods depicted in the bottom of Fig. 1 based on (OCP) right before the related discussions. Note again that there are further (hybrid) solution approaches which are not depicted in Fig. 1, compare also Section 2.3.

We may conjecture that an external integration software, i.e., a dedicated ODE or DAE solver, provides solution trajectories for the continuous-time differential equations, while numerical integration schemes with a fixed step size yield discrete-time solutions. Consequently, we also study the usage of "discrete-time" and "continuous-time". This excourse brings us to the field of scheduling and, more specifically, integrated scheduling and control problems. In this field of research, continuous-time dynamics may be solved by discretization methods or approximated with the help of steady-state assumptions. Thus, we also investigate the parallels and differences between discrete-time solutions and quasi-steady operation. Either way, we aim at the optimal piecewise constant control of a continuous process which relates the scope of this tutorial with the field of sampled-data control [e.g.,23,24]. The focus of this tutorial is therefore distinct from purely discrete-time models where the determination of state and control values at intermediate time points may not be possible or even infeasible.

A detailed examination of control methods apart from direct methods for offline open-loop optimal control as well as of numerical integration schemes and solution methods for DAEs, PDEs, boundary value problems, or mixed-integer problems lies beyond the scope of this tutorial. In particular, we do not focus on robust control or stochastic optimal control problems [e.g.,25,26].

The remainder of this article is structured as follows. In Section 2, we investigate hidden conjectures on direct methods for the solution of optimal control problems like (OCP). In particular, we focus on methods using state discretization in Section 2.1, methods using state integration in Section 2.2, and methods combining both of these approaches in Section 2.3. In Section 3, we link and compare these methods with approaches postulating quasi-steady behavior. We challenge our conclusions with the common knowledge available via ChatGPT [27] in Section 4, before we summarize and formalize them in the form of recommendations for (mostly) unambiguous terminology in Section 5.

2. Hidden conjectures on direct methods

In all direct methods, we start with *discretizing* the infinite-dimensional function space $L^\infty([0,t_f])$ which hosts the control function $u(\cdot)$ into a finite-dimensional vector space $\mathbb{R}^{n_u}$ which hosts the control coefficients $\hat{\boldsymbol{u}} = (\hat{u}_i)_{i=1,\ldots,n_u}$. This is also known as *transcription* in the context of trajectory optimization [4]. In detail, we choose finitely many basis functions $\varphi_i : [0,t_f] \to \mathbb{R}$, $i = 0,\ldots,n_u-1$ and *parameterize* the control function via $u(t) \approx \sum_{i=0}^{n_u-1} \hat{u}_i \cdot \varphi_i(t)$, $t \in [0,t_f]$. As an example, we exploit the indicator function $\mathbf{1}_T : [0,t_f] \to \{0,1\}$ with $\mathbf{1}_T(t) = 1$ if $t \in T$ and $\mathbf{1}_T(t) = 0$ otherwise for defining piecewise constant controls.

Approach 1 (*Control Discretization*)**.** *For solving (OCP) with direct methods, we use piecewise constant controls given by specific basis functions $\varphi_i(\cdot) = \mathbf{1}_{[t_i,t_{i+1})}(\cdot)$ for any $i = 1,\ldots,n_u$ defined on subintervals given by time discretization $0 = t_0 < t_1 < \cdots < t_{n_u} = t_f$ which yields the discretized optimal control problem*

$$
\begin{aligned}
\min_{\hat{\boldsymbol{u}}} \quad & \int_0^{t_f} \mathcal{L}(x(t))\, dt \\
\text{s.t.} \quad & \frac{\mathrm{d}x}{\mathrm{d}t}(t) = \frac{\hat{u}_i}{x(t)} \qquad \forall t \in [t_i,t_{i+1})\ \forall i = 0,\ldots,n_u-1 \qquad \text{(discOCP)} \\
& x(0) = x_0 \\
& x(t_f) = x_f \qquad .
\end{aligned}
$$

Note that piecewise constant discretization is often used without an explicit function rule for the control values $u(t)$ in the interior of the subintervals $t \in (t_i,t_{i+1})$. Another common discretization is the choice of piecewise polynomials [e.g.,13, Sec. 10]. Note further that the solution obtained by direct methods is, in general, only optimal for the discretized problem (discOCP). Some optimization algorithms guarantee convergence to the solution of (OCP) by an adaptive control and/or state discretization [e.g.,28,29].

Direct methods mainly use two approaches for solving the differential equations embedded: a-priori state discretization and state integration, cf. Fig. 1. In the following subsections, we therefore successively focus on approaches using state discretization, state integration, or both. In particular, we argue that state integration may be performed with an external integration software or an embedded numerical integration scheme hidden from the optimizer, see Hypothesis 6 and Section 2.3.

2.1. Distinct terminology for approaches using state discretization

When using state discretization, we determine finitely many state values which are the solution of the differential equations or the related boundary value problem. Collocation [30,31] is a standard approach for the solution of boundary value problems where the states are interpolation polynomials on subintervals $[t_i,t_{i+1}]$, $i = 0,\ldots,n$, the so-called collocation elements. The differential equations and the boundary values are enforced at the n_C collocation points $c_j \in [0,1]$, $j = 1,\ldots,n_C$, i.e., at time points $t_i + c_j \cdot h_i$. Alternative choices for the interpolation polynomials and collocation points result in specific collocation approaches like trapezoidal collocation and Hermite–Simpson collocation [e.g.,4,32,33]. In *orthogonal* collocation, the collocation points are the roots of orthogonal polynomials, e.g., Legendre or Chebyshev polynomials [34]. Since direct methods using state discretization ("full discretization") typically apply collocation, we formulate the hidden conjecture:

Hypothesis 1. *"Full discretization" and "direct collocation" refer to the same method.*

Already in the late 1960's, Tabak and Kuo proposed to discretize both controls and states where the discretized state values satisfy the differential equations according to a numerical integration scheme [9]. For example, we may apply the explicit Euler scheme.

Approach 2 (*Full Discretization with the Explicit Euler*)**.** *We discretize the state of (discOCP) according to the explicit Euler scheme with step width $h_i := t_{i+1} - t_i$, $i = 0,\ldots,n_u-1$ and obtain*

$$
\begin{aligned}
\min_{\hat{\boldsymbol{u}},\hat{\boldsymbol{x}}} \quad & \sum_{i=0}^{n_u} \hat{\mathcal{L}}(\hat{x}_i) \\
\text{s.t.} \quad & \hat{x}_{i+1} = \hat{x}_i + h_i \cdot \frac{\hat{u}_i}{\hat{x}_i} \qquad \forall i = 0,\ldots,n_u-1 \qquad \text{(FullDisc:Euler)} \\
& \hat{x}_0 = x_0 \\
& \hat{x}_{n_u} = x_f \qquad .
\end{aligned}
$$

Subsequently, Reddien and Biegler et al. proposed to use orthogonal collocation for an accurate and efficient state discretization [11,35–37].

As an example, we interpolate the first derivative of the state based on one collocation point c_1 and, thus, with constant interpolation polynomials. With this, the differential equation of (OCP) reads as

$$
x(t_i + \tau \cdot h_i) = x(t_i) + h_i \cdot \frac{u(t_i + c_1 \cdot h_i)}{x(t_i + c_1 \cdot h_i)} \cdot \int_0^\tau 1\ \mathrm{d}\sigma \quad \forall i = 0,\ldots,n_u-1
$$

resulting in the following fully discretized optimal control problem.

Approach 3 (*Direct Collocation*). *We discretize the state of* (discOCP) *according to collocation with the single collocation point* $c_1 = 0$ *and obtain*

$$\begin{aligned}
\min_{\hat{\mathbf{u}},\hat{\mathbf{x}}} \quad & \int_0^{t_f} \mathcal{L}(x(t))dt \\
\text{s.t.} \quad & x(t_i + \tau h_i) = \hat{x}_i + \tau \cdot h_i \cdot \frac{\hat{u}_i}{\hat{x}_i} && \forall \tau \in [0,1]\ \forall i = 0,\dots,n_u - 1 \\
& \hat{x}_{i+1} = x(t_i + h_i) && \forall i = 0,\dots,n_u - 1 \\
& \hat{x}_0 = x_0 \\
& \hat{x}_{n_u} = x_f && .
\end{aligned} \tag{Coll}$$

In comparison, (FullDisc : Euler) results in the same state values $\hat{x}_i = x(t_i)$ at the grid points but does not explicitly define state values $x(t_i + \tau \cdot h_i)$ at the interior of the collocation elements $\tau \in (0,1)$. Note that *direct* indicates that we are solving the problem with discretized controls, i.e., using a direct method, compared to solving the boundary value problem obtained from Pontryagin's maximum principle, i.e., using an indirect method, cf. Fig. 1.

Conclusion 1. Direct collocation is a specific variant of the full discretization approach. Since (orthogonal) collocation is the state-of-the-art approach for handling state discretization, "direct collocation" is sometimes equated with the full discretization approach [e.g.,6,10, 38]. □

In the literature, we find further synonyms for full discretization:

Hypothesis 2. *"Full discretization" and "simultaneous method" refer to the same method.*

Literally, both states and controls are discretized by "full discretization" and simultaneously optimized by the "simultaneous method". We can imagine to decouple these two approaches: we discretize both states and controls (*full discretization*) but evaluate the states sequentially while optimizing only the controls (*sequential method*), see Hypotheses 3 and 7. However, the standard approach is to use a combination, namely to optimize an algebraic optimization problem where the differential equations are reformulated as equality constraints (*simultaneous method*) based on discretized states (*full discretization*), see Approaches 2 and 3. "Full discretization", "simultaneous method", and "all-at-once method" are therefore often used synonymously [e.g.,10,13,14]. Note that there are further related terms like "one-shot approach" [e.g.,39] or "simultaneous analysis and design" (SAND) [40] in the field of optimal control subject to PDEs.

In direct multiple shooting [41], the states are partially discretized and optimized simultaneously to the discretization and optimization of controls, see Approach 5. Thus, direct multiple shooting may also be classified as "simultaneous method" or "full discretization method" [e.g.,6,14], although it is more often regarded as hybrid method, see Section 2.3.

Conclusion 2. The use of "full discretization" and "simultaneous method" is ambiguous: On the one hand, both terms may synonymously refer to a simultaneous full discretization approach. On the other hand, either of the terms may be used as umbrella term for the other as well as hybrid approaches like direct multiple shooting. □

In this tutorial, we use the term "full discretization" for the specific direct method to stress the usage of state discretization as opposed to state integration, see Fig. 1.

2.2. Distinct terminology for approaches based on state integration

Apart from collocation, single shooting is a common approach for the solution of boundary value problems [e.g.,42]. When combining single shooting with control discretization, we obtain a direct method for optimal control where the integration is hidden in the evaluation of a black-box function.

Approach 4 (*Direct Single Shooting*). *We assume that function* $F^{int}(\cdot;\hat{\mathbf{u}},x_0) : [0,t_f] \to \mathbb{R}$ *evaluates the solution* $x(\cdot)$ *of the differential equation of* (discOCP) *with control values* $\hat{\mathbf{u}}$ *initialized at* $x(0) = x_0$*. Thus, we can replace the differential equation with a function evaluation and obtain*

$$\begin{aligned}
\min_{\hat{\mathbf{u}}} \quad & \int_0^{t_f} \mathcal{L}\left(F^{int}(t;\hat{\mathbf{u}},x_0)\right) dt \\
\text{s.t.} \quad & F^{int}(t_f;\hat{\mathbf{u}},x_0) = x_f \quad .
\end{aligned} \tag{SiSh}$$

Since the state integration is embedded in the optimization when using direct single shooting, we may conjecture:

Hypothesis 3. *"Direct single shooting" and "sequential method" refer to the same method.*

Literally, "sequential method" comprises all methods where state integration and optimization of controls is performed sequentially such that the states are dependent variables. Apart from direct single shooting, see Approach 4, this is also true for hybrid methods like direct multiple shooting where all state values except those at the grid points are dependent variables, see Approach 5. However, in the literature, direct multiple shooting is classified as a hybrid or simultaneous method, compare Hypothesis 2, while direct single shooting is called a (pure) sequential method [e.g.,2,43]. In parts of the literature, direct methods using embedded state integration, which we call "direct single shooting", are explicitly introduced as "sequential methods" [e.g.,14].

Conclusion 3. Even though direct single shooting is a specific variant of the sequential method in theory, "direct single shooting" and "sequential method" are used synonymously in the literature. □

Since the controls but not the states are discretized in direct single shooting, we may also conjecture:

Hypothesis 4. *"Direct single shooting" and "partial discretization" refer to the same method.*

While "full discretization" refers to the discretization of both controls and states, "partial discretization" is an umbrella term indicating any approach solely discretizing the controls. This includes, e.g., direct single shooting and a dynamic programming approach [14]. However, "full discretization" and "partial discretization" may also be used for the opposed direct methods applying state discretization and state integration, respectively.

Conclusion 4. The use of the term "partial discretization" is ambiguous: it may comprise or equate direct single shooting. □

In large parts of the literature, the counterpart of full discretization is called "control vector parameterization", leading to the following hidden conjecture:

Hypothesis 5. *"Direct single shooting" and "control vector parameterization" refer to the same method.*

In fact, the usage of "control vector parameterization" is ambiguous. Literally, "control vector parameterization" means to parametrize the control vector, i.e., to discretize the controls. Control discretization was originally proposed for the solution of boundary value problems [44]. Since the late 1960's and early 1970's, it has also been increasingly used for optimal control, first by applying it to the solution of the boundary value problems obtained from indirect methods [45–48]. Subsequently, control discretization was combined with embedded state integration giving rise to a novel direct method [8,49,50]. With these and further advancements, control discretization with embedded state integration became a capable state-of-the-art method and was shortly referred to as "control vector parameterization".

Conclusion 5. Some groups [e.g.,51–53] use "control vector parameterization" as a synonym for direct single shooting, while others [e.g.,41] use the term literally, i.e., as a synonym for control discretization. □

Either way, the embedded state integration is often related to an external integrator:

Hypothesis 6. *In direct shooting methods, the differential equations are solved with an external integration software.*

Nowadays, direct shooting methods apply external ODE or DAE solvers which provide accurate solutions of the differential equations by limiting the integration error via adaptive grid refinement and the implementation of state-of-the-art methods, e.g., for handling stiff differential equations [e.g.,54–56]. However, when control discretization with embedded state integration was proposed for optimal control, there had hardly been any capable ODE or DAE solvers, yet. Still, the differential equations were not treated as equality constraints to be handled by the optimizer but hidden within the optimizer's function evaluations.

Originally, integration and optimization were decoupled by directly including the sensitivity equations of the states within the optimization problem [e.g.,8,49]. Subsequently, state integration was included within the evaluation of the objective function, e.g., in combination with derivative-free optimization methods or penalty formulations [e.g.,7,50,57]. A further advancement was the proposal of forward and backward differentiation based on a numerical integration scheme for the state integration embedded in the optimization of the controls [51]. Only then, capable solvers were available for integrating the states externally [e.g.,5,58].

Conclusion 6. In direct shooting methods, using an external integration software for the embedded state integration is the standard. This standard allows for exploiting all recent advancements for the efficient and accurate solution of differential equations. However, judging from the origins of direct single shooting, the embedded call of an integration software may alternatively be replaced by the evaluation of problem-specific sensitivity equations or numerical integration schemes. □

The use of numerical integration schemes points towards a hidden parallel between direct collocation and direct single shooting.

Hypothesis 7. *Direct single shooting handles the differential equations substantially differently compared to the full discretization method.*

In the standard approaches, the optimizer implicitly handles the integration by calling an external integration software like a black-box function F^{int} within direct single shooting, see (SiSh), while it explicitly handles an algebraic equation system given by orthogonal collocation within full discretization, see (FullDisc : Euler). Following our conclusions for Hypothesis 6, we may also include the evaluation of the numerical integration scheme hidden in F^{int} in the optimization problem.

As an example, we use an explicit Euler scheme on the time grid used for discretizing the controls. With this, (SiSh) transforms into

$$\begin{aligned} \min_{\hat{\boldsymbol{u}}} \quad & \sum_{i=0}^{n_u} \hat{\mathcal{L}}(\hat{x}_i) \\ \text{s.t.} \quad & \hat{x}_{i+1} := \hat{x}_i + h_i \cdot \frac{\hat{u}_i}{\hat{x}_i} \qquad \forall i = 0, \dots, n_u - 1 \\ & \hat{x}_0 := x_0 \\ & \hat{x}_{n_u} = x_f \end{aligned} \tag{SiSh : Euler}$$

Note that we distinguish between equality constraints "=" handled by the optimizer, see the endpoint constraint, and assignments ":=", see the numerical integration scheme. The assignments refer to simple function evaluations triggered by the optimizer. In particular, the assignments can be substituted into the objective and constraints, cf. (SiSh), when using an explicit numerical integration scheme. Thus, (SiSh : Euler) is a reduced-space formulation of (FullDisc : Euler): we reduce the number of optimization variables by identifying and determining the states as dependent variables.

If the objective function or constraints require knowledge of intermediate state values, we can add an interpolation scheme for the states. A piecewise linear interpolation connecting the state values determined based on the explicit Euler scheme yields

$$\begin{aligned} \min_{\hat{\boldsymbol{u}}} \quad & \int_0^{t_f} \mathcal{L}(x(t))\mathrm{d}t \\ \text{s.t.} \quad & x(t_i + \tau h_i) := \hat{x}_i + \tau \cdot h_i \cdot \frac{\hat{u}_i}{\hat{x}_i} \qquad \forall \tau \in [0,1] \ \forall i = 0, \dots, n_u - 1 \\ & \hat{x}_{i+1} := x(t_i + h_i) \qquad \forall i = 0, \dots, n_u - 1 \\ & \hat{x}_0 := x_0 \\ & \hat{x}_{n_u} = x_f \end{aligned} \tag{SiSh : interpol}$$

which is the reduced-space formulation of (Coll).

Reduced-space formulations [59,60] are similar to decomposition approaches: the structure of the model equations is exploited for reducing the dimensionality of the optimization problem. However, the dimensionality reduction is performed during modeling with reduced-space formulations and within the optimization algorithm with decomposition algorithms [61, Sec. 3.4.3]. The parallel between direct collocation combined with such reduction approaches and direct single shooting has been already hinted in the 1990's:

> *"In fact, the use of decomposition techniques for the solution of the large NLP arising from the complete discretization approach leads to methods that are, in general, very similar to the control parameterization approach. The main remaining difference between the two approaches is the precise method used for approximating the infinite dimensional DAEs by a finite number of constraints. It can readily be shown that the collocation methods on finite elements (that are primarily used in the context of the complete discretization approach) are equivalent to fully implicit Runge–Kutta single-step integration methods. On the other hand, most implementations of control vector parametrization rely on multistep backward-difference formula (BDF) integration methods (Gear, 1971). However, in principle, there is no reason why either integration method could not be used in either approach although one would expect multistep methods to lead to more efficient (i.e., smaller) approximations especially at high accuracies."* (Vassiliadis, Sargent, and Pantelides 1994 [58], p. 2114)

The equivalency of specific collocation and Runge–Kutta methods has been shown around 1970 [62,63]. For the proof, the first derivative of the states are interpolated with Lagrange polynomials. Consequently, the right-hand side of the explicit ODE equals the interpolation polynomial which gives the well-known formulas for the Runge–Kutta coefficients based on Lagrange polynomials defined with the collocation points [e.g.,64]. Still, collocation is slightly distinct from Runge–Kutta methods:

> *"The only difference is that in collocation one works in terms of polynomial coefficients and in implicit Runge–Kutta in terms of values of the function f."* (Wright 1970 [63], p. 219)

We may close this gap with a natural continuous extension [65] which basically defines an interpolation scheme for obtaining a continuous-time trajectory based on the determined state values. As an example, compare the linear interpolation used in (Coll) and (SiSh : interpol).

Overall, the state integration may be modeled with exactly the same equations within both direct single shooting and full discretization. Even though, the structural differences between the resulting optimization problems remain [13,14,28,41,43,53,66]: In the reduced-space/direct single shooting formulation, we have few optimization variables, namely n_u in (SiSh : Euler), but a dense optimization problem. In the full-space/full discretization formulation, we have a significantly larger number of optimization variables, namely n_u^2 in the small example (FullDisc : Euler), but obtain a highly structured, sparse optimization problem. Moreover, using equality constraints rather than assignments in the full-space/full discretization formulation gives the optimizer more flexibility for choosing the related control values and allows for easily imposing initial values and/or constraints on intermediate state values. Together, this results in less sensitivity to integration errors or unstable differential equations. Note that specialized integrators, e.g., for handling stiff differential equations, can be used for stabilizing direct shooting methods.

Conclusion 7. Briefly speaking, the state-of-the-art approaches of direct single shooting and full discretization are substantially different since these methods follow different paradigms: the former exploits the capabilities of state-of-the-art ODE or DAE solvers for an efficient integration, while the latter formulates an algebraic optimization problem and exploits its structure for an efficient optimization. However, the state integration may be performed equally, in particular, when using Runge–Kutta methods within direct single shooting and collocation within the full discretization approach. Still, the structural differences in the respective optimization problems remain and come down to the comparison of reduced-space and full-space formulations. □

In the following section, we investigate reduced-space formulations and other hybrid methods combining state discretization and state integration in more detail.

2.3. Multiple shooting and other hybrid methods

Direct multiple shooting is commonly referred to as hybrid method [e.g.,2,3,43,67] since it combines both (partial) state discretization and state integration, see the corners of Fig. 2. More precisely, state integration is performed independently on subintervals, the so-called *shooting intervals*. For this, *shooting variables* are introduced which serve as initial values for the integration. As for single shooting, see Approach 4, the integration is typically performed by an external solver, see Hypothesis 6. Closing constraints enforce the equality of the shooting variables with the last state value obtained in the previous shooting intervals such that we obtain continuous state trajectories. The “closing constraints” are also called “matching constraints” [e.g.,41] or “continuity conditions” [e.g.,3] in the literature.

As an example, we use the time grid used for the control discretization to define the shooting intervals.

Approach 5 (*Direct Multiple Shooting*)**.** *We assume that function $F^{int}(\cdot;\hat{u}_i,s_i) : [t_i,t_{i+1}] \to \mathbb{R}$ evaluates the solution $x_i(\cdot)$ of the differential equation of (discOCP) with fixed control value $\hat{u}_i$ initialized at $x_i(t_i) = s_i$. Thus, we can replace the differential equation within each subinterval with a function evaluation and obtain*

$$\begin{aligned}
\min_{\hat{\boldsymbol{u}},\boldsymbol{s}} \quad & \int_0^{t_f} \mathcal{L}\big(x(t)\big)\, dt \\
\text{s.t.} \quad x(t) &:= \sum_{i=0}^{n_u-1} \mathbf{1}_{[t_i,t_{i+1})}(t)\cdot x_i(t) && \forall i = 0,\dots,n_u-1 \\
x_i(t) &:= F^{int}(t;\hat{u}_i,s_i) && \forall t \in [t_i,t_{i+1}]\ \forall i = 0,\dots,n_u-1 \\
x_i(t_i) &:= s_i && \forall i = 0,\dots,n_u-1 \\
s_0 &:= x_0 \\
x_i(t_{i+1}) &= s_{i+1} && \forall i = 0,\dots,n_u-1 \\
s_{n_u} &:= x_f && .
\end{aligned} \tag{MuSh}$$

Analogously to (SiSh : Euler), we can replace black-box function $F^{\text{int}}(\cdot)$ with an integration scheme which is hidden from the optimizer as depicted in the middle row in Fig. 2. When using the explicit Euler scheme with step width $h_{i,j} := t_{i,j+1} - t_{i,j}$ defined for intermediate time points $t_i = t_{i,0} < t_{i,1} < \cdots < t_{i,n_i} = t_{i+1}$, $i = 0,\dots,n_u{-}1$, (MuSh) transforms into

$$\begin{aligned}
\min_{\hat{\boldsymbol{u}},\boldsymbol{s}} \quad & \sum_{i=0}^{n_u}\sum_{j=0}^{n_i} \hat{\mathcal{L}}(\hat{x}_{i,j}) \\
\text{s.t.} \quad \hat{x}_{i,j+1} &:= \hat{x}_{i,j} + h_{i,j}\cdot \frac{\hat{u}_i}{\hat{x}_{i,j}} && \forall j = 0,\dots,n_i-1\ \forall i = 0,\dots,n_u-1 \\
\hat{x}_{i,0} &:= s_i && \forall i = 0,\dots,n_u-1 \\
s_0 &:= x_0 \\
\hat{x}_{i,n_i} &= s_{i+1} && \forall i = 0,\dots,n_u-1 \\
s_{n_u} &:= x_f && .
\end{aligned} \tag{MuSh : Euler}$$

In principle, full discretization and direct single shooting can be viewed as the limiting cases of direct multiple shooting: We obtain the former when restricting ourselves to a single integration step per shooting interval and the latter when restricting ourselves to a single shooting interval. In line with that, stability and sparsity properties of direct multiple shooting are settled between those of full discretization and direct single shooting, compare the concluding paragraphs of Hypothesis 7. The decision whether to use state integration and/or a reduced-space formulation affects also the accessibility of intermediate state values for imposing constraints or for exploiting knowledge on the solution, e.g., state values at specific time points. When using reduced-space formulations, i.e., direct shooting methods where the state values are assigned via integration schemes, the optimizer can easily impose constraints on the intermediate state values although it can adapt at most the shooting variables.

Compared to (SiSh : Euler), we may interpret (MuSh : Euler) as a partial reduced-space formulation: a subset of the state values is assigned while the rest, the shooting variables, are optimization variables. Similarly, we can design different reduced-space formulations for handling the objective function. In fact, we can replace the Lagrange-type objective $\int_0^{t_f} \mathcal{L}(x(t))\, \mathrm{d}t$ with a Mayer-type objective $y(t_f)$, where $\mathrm{d}y/\mathrm{d}t = \mathcal{L}(x)$ and $y(0) = \mathcal{L}(x(0))$. This additional differential equations may be handled equivalently or differently to the other differential equations. In particular, we obtain a partial reduced-space formulation when reformulating a quadrature formula for the numerical integration of the Lagrange-type objective within full discretization, see (FullDisc : Euler), as assignments for evaluating a Mayer-type objective

$$\begin{aligned}
\min_{\hat{\boldsymbol{u}},\hat{\boldsymbol{x}}} \quad & y_{n_u} \\
\text{s.t.} \quad y_{i+1} &:= y_i + h_i\cdot \mathcal{L}(\hat{x}_i) && \forall i = 0,\dots,n_u-1 \\
y_0 &:= \mathcal{L}(\hat{x}_0) \\
\hat{x}_{i+1} &= \hat{x}_i + h_i\cdot \frac{\hat{u}_i}{\hat{x}_i} && \forall i = 0,\dots,n_u-1 \\
\hat{x}_0 &= x_0 \\
\hat{x}_{n_u} &= x_f && ,
\end{aligned}$$

where we observe $\hat{\mathcal{L}}(\hat{x}_0) = \mathcal{L}(\hat{x}_0)$ and $\hat{\mathcal{L}}(\hat{x}_i) = h_i\cdot\mathcal{L}(\hat{x}_i)$ for any $i = 1,\dots,n_u$.

Direct single or multiple shooting have been combined with full discretization approaches to obtain more efficient solution methods. Similar to the assignments used in the reduced-space formulation [e.g.,59,68,69], state integration may be nested within the optimization by applying collocation [70–72] or the implicit Euler scheme [3] in each iteration of the optimizer, see the middle right scheme in Fig. 2. In the same way, the state integration within direct multiple shooting may be performed with collocation which, in turn, is hidden from the optimizer with the help of the lifted Newton algorithm [12,73], see the middle left scheme in Fig. 2.

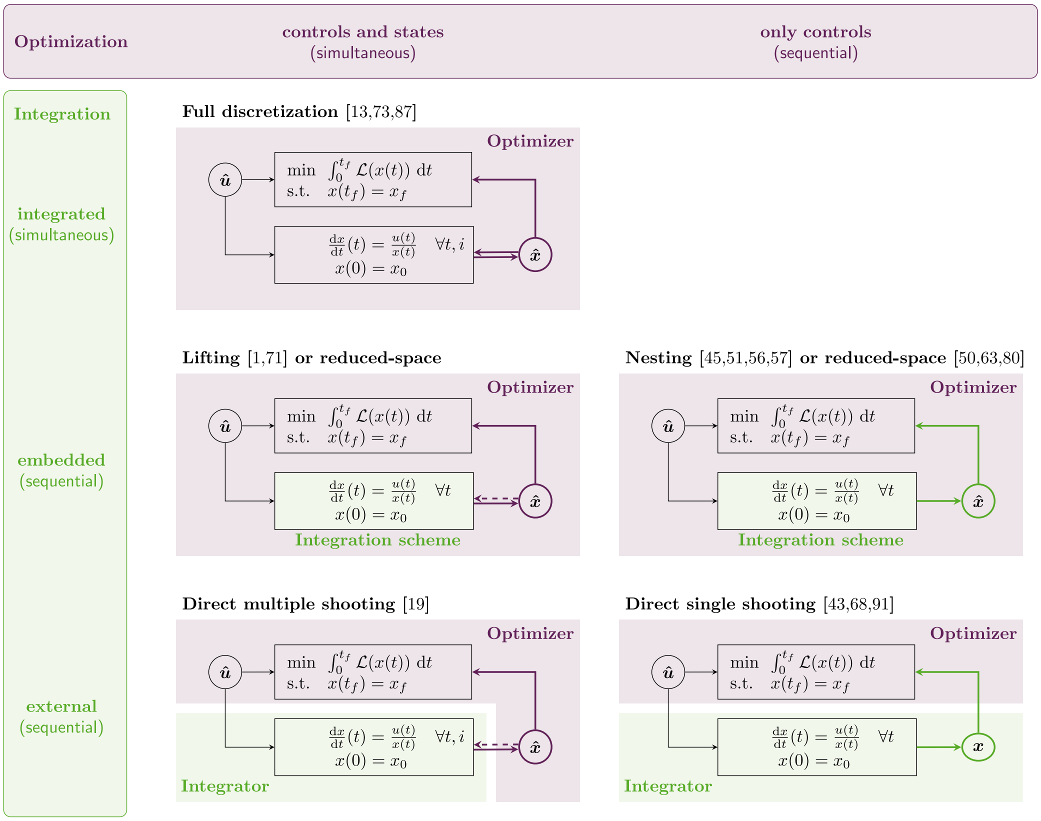

Optimization
controls and states
(simultaneous)
only controls
(sequential)
Integration
integrated
(simultaneous)
embedded
(sequential)
external
(sequential)
Full discretization [13,73,87]
Optimizer
$\hat{u}$
$\min \int_0^{t_f} \mathcal{L}(x(t))\,dt$
s.t. $x(t_f) = x_f$
$\frac{dx}{dt}(t) = \frac{u(t)}{x(t)} \quad \forall t, i$
$x(0) = x_0$
$\hat{x}$
Lifting [1,71] or reduced-space
Optimizer
$\frac{dx}{dt}(t) = \frac{u(t)}{x(t)} \quad \forall t$
Integration scheme
Nesting [45,51,56,57] or reduced-space [50,63,80]
Optimizer
Integration scheme
Direct multiple shooting [19]
Optimizer
Integrator
Direct single shooting [43,68,91]
Optimizer
x
Integrator

Fig. 2. Schematic overview of standard and hybrid methods for solving optimal control problems like (OCP). Such optimal control problems extend an algebraic optimization problem (upper box) by differential equations (lower box). From top to bottom, the handling of the differential equations changes, namely from transforming it into constraints handled by the optimizer (🟪) via using an integration scheme hidden from the optimizer (□) to using an external integration software (🟩). Arrows from controls and/or states to the model boxes indicate optimization variables. In the schemes on the left, both controls and states are optimized after applying state discretization (⟺). The dashed arrow indicates that only a part of the state values are optimized (⇠⇢). In the schemes on the right, the state values are dependent variables since state integration is used (⟶).

Overall, full discretization, direct single shooting, and hybrid methods like direct multiple shooting differ in the handling of optimization variables and constraints. However, all of these methods may use the same approximate solution of the embedded differential equations no matter whether the integration is integrated or embedded within the optimization or whether an external integration software is called, see also Hypotheses 6 and 7.

We may conjecture that an external integrator gives an accurate solution for the continuous-time differential equations, whereas a numerical integration scheme embedded in the optimization problem gives only an approximation, namely the solution of the related discrete-time difference equations. However, external integration software are just implementations of numerical integration schemes. Either way, we obtain a discrete-time trajectory. We therefore investigate further implications for discrete-time solutions in the following section.

3. Hidden parallels between discrete-time dynamics and quasi-steady operation

We clarify the terminology before we compare discrete-time and quasi-steady solutions. With direct methods in mind, we formulate a first hidden conjecture:

Hypothesis 8. *A "discrete-time system" is obtained by discretizing differential equations.*

Commonly, differential equations are solved by transforming them into algebraic equations with the help of time discretization and a numerical integration scheme. In control theory, the resulting equation system is called "discrete-time model" [e.g.,74, p. 107]. In particular for linear ODEs, we can exploit the knowledge of the analytical solution to obtain an equivalent reformulation by the discretization.

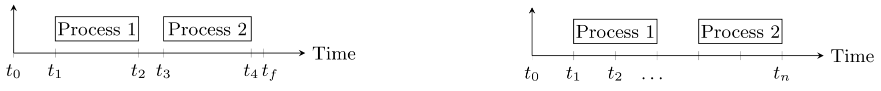
In the context of scheduling problems, we also distinguish between discrete-time and continuous-time systems [e.g.,75–77]. Scheduling problems aim at the optimal timing of processes including the determination of starting and end time points. When using continuous-time systems, these time points can take any real value in the time horizon $[t_0, t_f]$, see Fig. 3(a). When using discrete-time systems, these time points are chosen from a set of a-priori given time points t_i, $i = 0, \ldots, n$ with $t_n = t_f$, see Fig. 3(b). In fact, the time representation, namely the choice between continuous-time or discrete-time systems, refers to the handling of the scheduling controls rather than the handling of differential equations potentially embedded.

We interpret scheduling problems as optimal control problems: a control variable decides whether to run the process. When solving this optimal control problem with direct methods, in particular, a control discretized on a fixed time grid, we obtain a discrete-time system in terms of scheduling. This is still true if the underlying process is characterized by differential equations which we solve with the analytical continuous-time solution.

Conclusion 8. "Discrete-time" is used differently in different fields of research. While it refers to the handling of differential equations in control theory, it refers to the handling of (scheduling) controls in scheduling. □

For the term "quasi-steady", we first check whether there is a distinction to a similar term:

Hypothesis 9. *"Quasi-steady" and "pseudo-steady" are synonyms.*



(a) Continuous-time system

(b) Discrete-time system

Fig. 3. Schematic overview of time representation in scheduling problems.

Even though the different names "pseudo-steady" and "quasi-steady" may be used for slight distinctions, see Section 2.10 in Supplementary Material 1, there is no universal distinction between these two terms in the literature. As an example, the steady-state approximations for chemical reaction kinetics [78–80] is called quasi-steady state assumption (QSSA) [81–83], pseudo-steady state hypothesis (pssh) [83,84] or, by the International Union of Pure and Applied Chemistry (IUPAC), steady-state approximation [85]. Note also that in these established terms, the words "assumption", "hypothesis" and "approximation" are used with the same meaning, namely for an approximation whose validity is rarely checked but which typically results in a small error. Still, these terms as well as "quasi-steady" and "pseudo-steady" may refer to distinct terms.

Conclusion 9. In the literature, "quasi-steady" and "pseudo-steady" are mainly treated as synonyms. □

The term "quasi-steady" itself is ambiguous. We conjecture:

Hypothesis 10. *"Quasi-steady" operation refers to a time series of steady-state operating points.*

In the literature, we primarily find two different usages of the term "quasi-steady". On the one hand, "quasi-steady" refers to almost steady-state processes, namely if

(i) a state subject to a dynamic process is (approximately) constant in time [e.g.,86,87];
Example: The fluid level in a tank is constant if the inflow equals the outflow.

(ii) the rate of change is small compared to other changes in time [e.g.,78,88];
Example: The concentration of a highly reactive intermediate substance has a negligible rate of change compared to less reactive products of chemical reactions.

(iii) states describe (almost) time invariant properties [e.g.,89];
Example: The amplitude of a time dependent harmonic oscillation is constant.

(iv) the steady state is temporary [e.g.,90].
Example: Between two transient phases, the system remains in a steady state for a specific time period.

On the other hand, "quasi-steady" refers to piece-wise constant state trajectories. This includes both trajectories switching between a fixed number of modes [e.g.,91–93] and trajectories with continuous-valued operating points [e.g.,94–96]. In the former, the optimizer decides which of the discrete operation modes is chosen at any point in time, while it can choose any real value within given bounds for the current operating point in the latter.

Conclusion 10. States are called "quasi-steady" if they are almost constant for a specific time period and, in particular, if they are governed by piece-wise constant operating points. □

In the following, quasi-steady operation refers to a time series of (almost) steady-state operating points. This description is also valid for discrete-time solutions of differential equations obtained with the help of numerical integration schemes. Thus, we challenge the following hidden conjecture:

Hypothesis 11. *Quasi-steady operation and discrete-time dynamics are distinct approaches for approximating the continuous-time dynamics embedded in optimal control problems.*

The continuous-time solution of differential equations embedded in an optimal control problem can be approximated with different levels of accuracy, see Fig. 4. We achieve the smallest but least accurate model when replacing the transient state trajectory with a steady state. The most accurate but computationally costly approximation is given by a discrete-time trajectory obtained from an accurate integration scheme. As a compromise, we may use steady-state assumptions resulting in quasi-steady behavior. The discussion of whether the system dynamics can be adequately represented by decoupled operating points is not in scope of this tutorial but discussed elsewhere [e.g.,95,97–99]. Instead, we focus on quasi-steady operation where the dynamic system behavior is explicitly considered by coupling subsequent state values in time.

In quasi-steady operation, we may introduce a coupling in time via transition modes [e.g.,100], ramping constraints limiting the difference of subsequent state values [e.g.,99,101,102] or a combination of these approaches [e.g.,103]. Pointing to a parallel with discrete-time solutions of differential equations, we focus on first-order static ramping where the difference between subsequent state values is constrained by a constant ramp

$$\begin{aligned} \hat{x}_{i+1} &\leq \hat{x}_i + h_i \cdot \frac{\Delta x^{\text{ramp}}}{\Delta t^{\text{ramp}}} \qquad \forall i = 0, \dots, n_u - 1 \\ \hat{x}_{i+1} &\geq \hat{x}_i - h_i \cdot \frac{\Delta x^{\text{ramp}}}{\Delta t^{\text{ramp}}} \qquad \forall i = 0, \dots, n_u - 1 \quad . \end{aligned} \tag{staticRamp}$$

Moving the ramping into bounds of an auxiliary variable yields

$$\begin{aligned} \hat{x}_{i+1} &= \hat{x}_i + h_i \cdot \hat{u}_i \qquad \forall i = 0, \dots, n_u - 1 \\ \hat{u}_i &\in \left[-\frac{\Delta x^{\text{ramp}}}{\Delta t^{\text{ramp}}}, \frac{\Delta x^{\text{ramp}}}{\Delta t^{\text{ramp}}} \right] \qquad \forall i = 0, \dots, n_u - 1 \quad . \end{aligned} \tag{explEuler}$$

For operational optimization, the main difference between (staticRamp) and (explEuler) is that we exclusively optimize for the state values in the former while we optimize for the control and, when using full discretization, also the state values in the latter.

The difference equation of (explEuler) resembles the explicit Euler scheme for differential equation $\mathrm{d}x/\mathrm{d}t = u$. We extend (explEuler) to general explicit ODEs $\mathrm{d}x/\mathrm{d}t = f(x, u)$ based on a dynamic ramping approach proposed for integrated scheduling and control [104,105]. Therein, Baader et al. replace the process dynamics with time-dependent bounds on the scheduling control which corresponds to the backward direction of our reformulation, namely from (explEuler) to (staticRamp). For this, the existence of at least two distinct controls is exploited: at least one for scheduling, i.e., for determining the optimal operating point, and at least one for control, i.e., for maintaining the operating point.

We adapt the dynamic ramping approach from Baader et al. [104,105] to our example which contains only one control. In fact, we replace the process dynamics with time-dependent bounds on subsequent state values by applying the explicit Euler scheme and introducing an auxiliary variable

$$\hat{x}_{i+1} = \hat{x}_i + h_i \cdot \underbrace{f(\hat{x}_i, \hat{u}_i)}_{=: v(t_i)} \qquad \forall i = 0, \dots, n_u - 1 \quad .$$

The process dynamics and the control bounds determine the bounds on the ramping variable v. We extend our example (OCP) with control

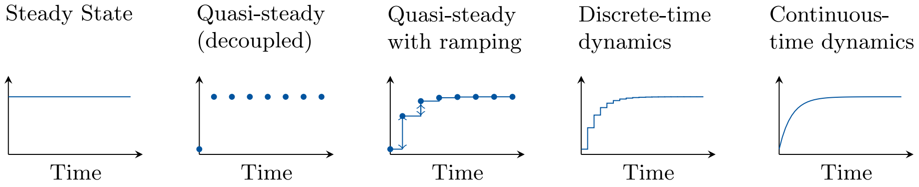


Fig. 4. Schematic overview of different paradigms for modeling process dynamics.

bounds $u(t) \in [-u^{\max}, u^{\max}]$ and state bounds $x(t) \in [x^{\min}, x^{\max}]$ for any $t \in [0, t_f]$. We obtain state-dependent ramping limits $v(x(t)) \in [-u^{\max}/x(t), u^{\max}/x(t)]$, $t \in [t_i, t_{i+1}]$. However, we have only limited information on the transient state values at the interior of the subintervals $t \in (t_i, t_{i+1})$ in quasi-steady operation: We only know the neighboring state values x_i and x_{i+1} as well as the general state bounds. Thus, we propose three different ramping strategies:

(R1) *Maximum flexible ramping* allowing for the largest deviations possible with the given differential equation, i.e.,

$$v(t) \in [\min_{x\in[x^{\min},x^{\max}]} \frac{-u^{\max}}{x}, \max_{x\in[x^{\min},x^{\max}]} \frac{u^{\max}}{x}] = [\frac{-u^{\max}}{x^{\min}}, \frac{u^{\max}}{x^{\min}}] \quad \forall t \in [t_i, t_{i+1})$$

(R2) *Minimum flexible ramping* guaranteeing that we cannot exceed the inertia given by the differential equation, i.e.,

$$v(t) \in [\max_{x\in[x^{\min},x^{\max}]} \frac{-u^{\max}}{x}, \min_{x\in[x^{\min},x^{\max}]} \frac{u^{\max}}{x}] = [\frac{-u^{\max}}{x^{\max}}, \frac{u^{\max}}{x^{\max}}] \quad \forall t \in [t_i, t_{i+1})$$

(R3) *Dynamic ramping* with time-dependent bounds assuming that we have a steady state in the interval, i.e.,

$$v(t) \in [\frac{-u^{\max}}{x(t_i)}, \frac{u^{\max}}{x(t_i)}] \quad \forall t \in [t_i, t_{i+1})$$

Maximum and minimum flexible ramping are static, i.e., independent from the time discretization. When using dynamic ramping with a refined time grid, we can, however, obtain a more accurate approximation of the reachable state differences.

Quasi-steady operational optimization with dynamic ramping

$$\begin{aligned}
\min_{\hat{x}} \quad & \sum_{i=0}^{n_u} \hat{\mathcal{L}}(\hat{x}_i) \\
\text{s.t.} \quad & \hat{x}_{i+1} \leq \hat{x}_i + h_i \cdot \frac{u^{\max}}{\hat{x}_i} && \forall i = 0, \dots, n_u - 1 \\
& \hat{x}_{i+1} \geq \hat{x}_i - h_i \cdot \frac{u^{\max}}{\hat{x}_i} && \forall i = 0, \dots, n_u - 1 \\
& \hat{x}_0 = x_0 \\
& \hat{x}_{n_u} = x_f
\end{aligned} \qquad \text{(dynRamp)}$$

shows strong parallels to direct methods using the explicit Euler scheme for solving the differential equations. In fact, dynamic ramping (R3) is an equivalent formulation for the system of (in)equalities consisting of the related difference equation and the control bounds. Thus, the feasible set for the discretized states $\hat{x}$ is the same in optimization problems (dynRamp), (FullDisc : Euler), and (SiSh : Euler). Despite this similarity, there are three structural differences between the quasi-steady optimization problem (dynRamp) and the discrete-time optimal control problems (FullDisc : Euler) and (SiSh : Euler). Firstly, the subsequent state values are coupled via inequality constraints in (dynRamp), while they are coupled via equality constraints and assignments in (FullDisc : Euler) and (SiSh : Euler), respectively. Which of these modeling alternatives results in more favorable convergence properties depends on the optimization algorithm used and its specific implementation. Secondly, the piecewise constant states and controls share the same time discretization in (dynRamp), while (FullDisc : Euler) and (SiSh : Euler) allow for two different time grids similar to (MuSh : Euler). Thirdly, the control values are chosen implicitly in (dynRamp), while (FullDisc : Euler) and (SiSh : Euler) explicitly provide the optimal control values. In fact, the quasi-steady problem formulation determines the optimal state values such that the difference of subsequent values stays in the limits given by any control value within $[u^{\min}, u^{\max}]$. However, we may need explicit information on the control values for implementing the optimal control strategy in reality. In the best case, we can reformulate the ramping constraints such that we can calculate the control values from the optimal state values. In the worst case, we need to solve another optimal control problem, e.g., where the objective value is the deviation of the current state prediction to the optimal state trajectory. For (OCP), we obtain

$$\begin{aligned}
\min_{u(\cdot)} \quad & \int_0^{t_f} \left(x(t) - x^{\text{opt}}(t)\right)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & \frac{\mathrm{d}x}{\mathrm{d}t}(t) = \frac{u(t)}{x(t)} \quad \forall t \in [0, t_f] \\
& x(0) = x_0
\end{aligned}$$

when using the integral least squares deviation as the error measure.

Conclusion 11. Both discrete-time dynamics and quasi-steady operation provide the same approximate solution to the differential equations if the numerical integration scheme used in the former is captured by an adequate ramping in the latter. The resulting optimization problems show nevertheless structural differences in the handling of constraints and optimization variables as well as the specific information provided for the optimal control strategy. □

4. Retrospective with chatgpt

Since ChatGPT is trained with publicly available data [106], we assume that its responses represent the common knowledge. We can draw three main conclusions when comparing our conclusions with the responses of ChatGPT, see Supplementary Material 1:

(i) ChatGPT focuses on the literal definitions of the terms, whereas we additionally included the common usage of the terms in the literature, see the conclusions on Hypotheses 1–6, 8, 9.

(ii) The terminology used by ChatGPT is not fully consistent. For example, solvers and numerical methods are mixed in the response to Hypothesis 6; and distinct terminology in different communities may be neglected, see response to Hypothesis 8, or emphasized, see response to Hypothesis 9.

(iii) ChatGPT does not point out the hidden parallels developed for Hypotheses 7 and 11.

For the remaining Hypothesis 10, ChatGPT gives basically the same overview as we do in this tutorial.

In summary, the comparison with ChatGPT's responses emphasizes the motivation of this tutorial: There is a consensus on the general methods and the corresponding terminology in the research communities, but ambiguities remain. Moreover, parallels between specific variants of standard methods may easily be overlooked. We therefore develop recommendations for a well-defined terminology in the following section.

5. Conclusions

In this tutorial, we investigated ambiguous terminology for standard optimal control methods to clear potential misunderstandings between different research groups. We will now use this analysis to dare to extend some warnings ("!") and recommendations ("⇒").

In the introduction, we started with briefly delineating "optimal control" from which we conclude:

!1 "Optimal control" refers to the class of infinite-dimensional optimization problems with differential(–algebraic) equations embedded *or* closed-loop control problems.
⇒ *When using "optimal control", specify whether it is open-loop or closed-loop optimal control.*

In the main part, we focused on direct methods for offline open-loop optimal control. In detail, we argued that direct collocation is a specific variant of (simultaneous) full discretization which, in turn, may be considered as a specific simultaneous method. Note, however, that pairs of these three terms are used synonymously in parts of the literature. Analogously, "direct single shooting" is handled as synonym or specification of "partial discretization", while "direct single shooting" and "sequential method" mainly refer to the same method. Literally, "control vector parameterization" means discretization of controls and is therefore a paradigm shared by all direct methods. Historically, however, it became a synonym for direct single shooting in parts of the literature and, thus, may just refer to a specific direct method. In brief:

!2 Although collocation is the standard approach used within the full discretization approach, there are alternative state discretization approaches.
⇒ *Distinguish between full discretization and direct collocation.*
!3 The terms "full discretization", "simultaneous method", and "control vector parameterization" are ambiguous, cf. Fig. 1.
⇒ *Specify the intended meaning when using any of these terms.*

Commonly, the embedded state integration within direct single shooting is performed by an external ODE or DAE solver. Alternatively, a numerical integration scheme may be used which is evaluated within the optimizer's iterations. This alternative has been used in the origins of direct single shooting and is used in hybrid methods combining full discretization and direct shooting. The numerical integration scheme can be hidden from the optimizer when using assignments rather than equality constraints. This yields a reduced-space formulation of the full discretization approach. Based on the usage of numerical integration schemes, we continued to elaborate a strong parallel between the discrete-time solution obtained by direct methods and the optimal quasi-steady operation considering adequate ramping constraints. Note, however, that there are distinct definitions of "quasi-steady" and "discrete-time". We conclude:

!4 Although the use of an external integration software is the standard for the state integration embedded in direct single shooting, there are alternative approaches.
⇒ *Recall that most external integrators implement a numerical integration scheme when studying and extending direct shooting approaches.*
!5 "Quasi-steady operation" may refer to a(n) (almost) constant operating point *or* a time-series of (almost) constant operating points.
⇒ *Specify the intended meaning when using "quasi-steady".*
!6 "Discrete-time system" refers to a reformulation of dynamic systems by discretizing the differential equations *or* a specific time representation used in scheduling problems.
⇒ *Specify the intended meaning when using "discrete-time system".*

Following the motivation of this tutorial, investigating hidden parallels between solution methods for mixed-integer nonlinear programming, mixed-integer optimal control, and scheduling with finitely many alternative control values merits careful attention. For example, continuous-time systems in scheduling [e.g.,75] show parallels to switching time optimization in mixed-integer optimal control [e.g.,1,107,108]: in both approaches the optimal time point of control switches, e.g., for changing modes or starting a new process, is chosen freely from a continuous-valued time interval. Besides, both scheduling and optimal control problems may contain discrete-continuous dynamic systems, i.e., differential equations which switch their behavior at specific time points. For both, methods for solving "hybrid systems" may be of interest [e.g.,21,109–111].

Taking one step further, methods for solving integrated scheduling and control problems are related to those for integrated design and control problems. However, what is the difference between "design" and "control"? "Optimal control" problems may contain "design parameters" [e.g.,7,51], while "design" problems in the context of model-based optimal experimental design optimize "controllable variables" of processes which may or may not be dynamic [e.g.,112,113]. Even more general, is there a difference between "mathematical optimization" and "mathematical programming"? While an answer of the latter question has been found by the Mathematical Optimization Society (MOS) [114], the answer of the former question as well as investigations for further related fields of research are left for future work.

CRediT authorship contribution statement

Susanne Sass: Methodology, Investigation, Writing – original draft, Writing – review & editing, Funding acquisition, Formal analysis, Conceptualization. **Alexander Mitsos:** Writing – review & editing, Resources, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Funding: This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Germany under grant MI 1851/10-1 "Parameter estimation with (almost) deterministic global optimization".

Susanne Sass is grateful for her association to the International Research Training Group (DFG) IRTG-2379 "Hierarchical and Hybrid Approaches in Modern Inverse Problems" under grant 333849990/GRK2379.

We thank the anonymous reviewers whose comments helped us with improving and clarifying this article.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jprocont.2025.103513.

Data availability

No data was used for the research described in the article.

References

[1] M.P. Avraam, N. Shah, C.C. Pantelides, Modelling and optimisation of general hybrid systems in the continuous time domain, Comput. Chem. Eng. 22 (1998) S221 – S228, http://dx.doi.org/10.1016/S0098-1354(98)00058-1.

[2] E. Aydin, D. Bonvin, K. Sundmacher, Toward fast dynamic optimization: An indirect algorithm that uses parsimonious input parameterization, Ind. Eng. Chem. Res. 57 (30) (2018) 10038–10048, http://dx.doi.org/10.1021/acs.iecr.8b02109.

[3] M.T. Kelley, R. Baldick, M. Baldea, A direct transcription-based multiple shooting formulation for dynamic optimization, Comput. Chem. Eng. 140 (2020) 106846, http://dx.doi.org/10.1016/j.compchemeng.2020.106846.

[4] J.T. Betts, Practical Methods for Optimal Control using Nonlinear Programming, SIAM, Philadelphia, 2010.

[5] H.G. Bock, Numerical treatment of inverse problems in chemical reaction kinetics, in: V.I. Goldanskii, F.P. Schäfer, R. Gomer, J.P. Toennies, K.H. Ebert, P. Deuflhard, W. Jäger (Eds.), Modelling of Chemical Reaction Systems, 18 of Springer Series in Chemical Physics, Springer Berlin Heidelberg, Berlin, Heidelberg, 1981, pp. 102–125, http://dx.doi.org/10.1007/978-3-642-68220-9_8.

[6] M. Diehl, H.G. Bock, H. Diedam, P.-B. Wieber, Fast direct multiple shooting algorithms for optimal robot control, in: M. Diehl, K. Mombaur (Eds.), Fast Motions in Biomechanics and Robotics, in: Lecture Notes in Control and Information Sciences, Vol. 340, Springer, Berlin, Heidelberg, 2006, pp. 65–93, http://dx.doi.org/10.1007/978-3-540-36119-0_4.

[7] R.G. Brusch, R.H. Schappelle, Solution of highly constrained optimal control problems using nonlinear programing, AIAA J. 11 (2) (1973) 135–136, http://dx.doi.org/10.2514/3.50443.

[8] G.P. Pollard, R. Sargent, Off line computation of optimum controls for a plate distillation column, Automatica 6 (1) (1970) 59–76, http://dx.doi.org/10.1016/0005-1098(70)90075-0.

[9] D. Tabak, B.C. Kuo, Application of Math. Program. in the design of optimal control systems, Internat. J. Control 10 (5) (1969) 545–552, http://dx.doi.org/10.1080/00207176908905855.

[10] S. Sager, Numerical methods for mixed-integer optimal control problems, in: Der Andere Verlag Tönning, Lübeck, Marburg, 2005.

[11] L.T. Biegler, Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation, Comput. Chem. Eng. 8 (3–4) (1984) 243–247, http://dx.doi.org/10.1016/0098-1354(84)87012-X.

[12] R. Quirynen, S. Gros, M. Diehl, Lifted implicit integrators for direct optimal control, in: IEEE CDC 2015, Osaka, Japan, 2015, pp. 3212–3217, http://dx.doi.org/10.1109/CDC.2015.7402701.

[13] L.T. Biegler, Nonlinear Programming: Concepts, Algorithms, and Applications To Chemical Processes, MOS-SIAM, Philadelphia, 2010, http://dx.doi.org/10.1137/1.9780898719383.

[14] L.T. Biegler, I.E. Grossmann, Retrospective on optimization, Comput. Chem. Eng. 28 (8) (2004) 1169–1192, http://dx.doi.org/10.1016/j.compchemeng.2003.11.003.

[15] L.S. Pontryagin, V.G. Boltyanski, R.V. Gamkrelidze, E.F. Miscenko, The Mathematical Theory of Optimal Processes, John Wiley & Sons, New York, 1962.

[16] R. Bellman, The theory of dynamic programming, Bull. Am. Math. Soc. 60 (6) (1954) 503–515, http://dx.doi.org/10.1090/s0002-9904-1954-09848-8.

[17] D. Mayne, A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems, Internat. J. Control 3 (1) (1966) 85–95, http://dx.doi.org/10.1080/00207176608921369.

[18] R. Vinter, Convex duality and nonlinear optimal control, SIAM J. Control Optim. 31 (2) (1993) 518–538, http://dx.doi.org/10.1137/0331024.

[19] D. Hernández-Hernández, O. Hernández-Lerma, M. Taksar, The linear programming approach to deterministic optimal control problems, Appl. Math. 24 (1) (1996) 17–33.

[20] J.B. Lasserre, D. Henrion, C. Prieur, E. Trélat, Nonlinear optimal control via occupation measures and LMI-relaxations, SIAM J. Control Optim. 47 (4) (2008) 1643–1666, http://dx.doi.org/10.1137/070685051.

[21] S. Sager, Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control, J. Process Control 19 (8) (2009) 1238–1247, http://dx.doi.org/10.1016/j.jprocont.2009.03.008.

[22] R. Luus, Control vector iteration CVI, in: C.A. Floudas, P.M. Pardalos (Eds.), Encyclopedia of Optimization, Springer, New York, 2009, pp. 509–513, http://dx.doi.org/10.1007/978-0-387-74759-0_93.

[23] K.J. Åström, B. Wittenmark, Computer-Controlled Systems: Theory and Design, Dover Publications, Mineola, New York, 2013.

[24] G.F. Franklin, J.D. Powell, M.L. Workman, Digital Control of Dynamic Systems, Vol. 3, Addison-Wesley Publishing Company, Menlo Park, California, 1998.

[25] S. Thangavel, S. Lucia, R. Paulen, S. Engell, Dual robust nonlinear model predictive control: A multi-stage approach, J. Process Control 72 (2018) 39–51, http://dx.doi.org/10.1016/j.jprocont.2018.10.003.

[26] J. Puschke, A. Mitsos, Robust feasible control based on multi-stage eNMPC considering worst-case scenarios, J. Process Control 69 (2018) 8–15, http://dx.doi.org/10.1016/j.jprocont.2018.07.004.

[27] OpenAI, ChatGPT, 2025, Release June 7 2025, URL https://chat.openai.com/chat. (Accessed on June 09 2025).

[28] J.T. Betts, W.P. Huffman, Mesh refinement in direct transcription methods for optimal control, Optimal Control Appl. Methods 19 (1) (1998) 1–21, http://dx.doi.org/10.1002/(SICI)1099-1514(199801/02)19:1<1::AID-OCA616>3.0.CO;2-Q.

[29] B. Houska, B. Chachuat, Branch-and-lift algorithm for deterministic global optimization in nonlinear optimal control, J. Optim. Theory Appl. 162 (1) (2014) 208–248, http://dx.doi.org/10.1007/s10957-013-0426-1.

[30] R.A. Frazer, W.N.P. Jones, S.W. Skan, Approximations to functions and to the solutions of differential equations, 1937, p. 1799, Reports and Memoranda of the Aeronautical Research Committee.

[31] C. Lanczos, Trigonometric interpolation of empirical and analytical functions, J. Math. Phys. 17 (1–4) (1938) 123–199, http://dx.doi.org/10.1002/sapm1938171123.

[32] M. Kelly, An introduction to trajectory optimization: How to do your own direct collocation, SIAM Rev. 59 (4) (2017) 849–904, http://dx.doi.org/10.1137/16M1062569.

[33] C.R. Hargraves, S.W. Paris, Direct trajectory optimization using nonlinear programming and collocation, J. Guid. Control Dyn. 10 (4) (1987) 338–342, http://dx.doi.org/10.2514/3.20223.

[34] J.V. Villadsen, W.E. Stewart, Solution of boundary-value problems by orthogonal collocation, Chem. Eng. Sci. 22 (11) (1967) 1483–1501, http://dx.doi.org/10.1016/0009-2509(67)80074-5.

[35] G.W. Reddien, Collocation at Gauss points as a discretization in optimal control, SIAM J. Control Optim. 17 (2) (1979) 298–306, http://dx.doi.org/10.1137/0317023.

[36] J.E. Cuthrell, L.T. Biegler, On the optimization of differential–algebraic process systems, AIChE J. 33 (8) (1987) 1257–1270, http://dx.doi.org/10.1002/aic.690330804.

[37] S. Kameswaran, L.T. Biegler, Convergence rates for direct transcription of optimal control problems using collocation at Radau points, Comput. Optim. Appl. 41 (1) (2008) 81–126, http://dx.doi.org/10.1007/s10589-007-9098-9.

[38] A.B. Singer, P.I. Barton, Global solution of optimization problems with parameter-embedded linear dynamic systems, J. Optim. Theory Appl. 121 (3) (2004) 613–646, http://dx.doi.org/10.1023/B:JOTA.0000037606.79050.a7.

[39] H.G. Bock, A. Potschka, S. Sager, J.P. Schlöder, On the connection between forward and optimization problem in one-shot one-step methods, in: G. Leugering, S. Engell, A. Griewank, M. Hinze, R. Rannacher, V. Schulz, M. Ulbrich, S. Ulbrich (Eds.), Constrained Optimization and Optimal Control for Partial Differential Equations, Springer, Basel, 2012, pp. 37–49.

[40] R.T. Haftka, Simultaneous analysis and design, AIAA J. 23 (7) (1985) 1099–1103, http://dx.doi.org/10.2514/3.9043.

[41] H.G. Bock, K.-J. Plitt, A multiple shooting algorithm for direct solution of optimal control problems, IFAC Proc. 17 (2) (1984) 1603–1608, http://dx.doi.org/10.1016/S1474-6670(17)61205-9.

[42] J. Stoer, R. Bulirsch, third ed., Introduction To Numerical Analysis, vol. 12, Springer, New York, NY, 2002.

[43] T. Binder, L. Blank, H.G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J.P. Schlöder, O. von Stryk, Introduction to model based optimization of chemical processes on moving horizons, in: M. GrÖtschel and S.O. Krumke and J. Rambau, Springer, Berlin, Heidelberg, 2001, pp. 295–339, http://dx.doi.org/10.1007/978-3-662-04331-8_18.

[44] W. Ritz, Über eine neue methode zur Lösung gewisser variationsprobleme der mathematischen physik, J. Reine Angew. Math. 135 (1909) 1–61, http://dx.doi.org/10.1515/crll.1909.135.1.

[45] H.H. Rosenbrock, C. Storey, Computational Techniques for Chemical Engineers, Pergamon Press, London, 1966, http://dx.doi.org/10.1016/C2013-0-01841-5.

[46] R. Jackson, An approach to numerical solution of time-dependent optimisation problems in 2-phase contacting devices, Trans. Inst. Chem. Eng. 45 (4) (1967) T160.

[47] H. Sirisena, Computation of optimal controls using a piecewise polynomial parameterization, IEEE Trans. Autom. Control 18 (4) (1973) 409–411, http://dx.doi.org/10.1109/TAC.1973.1100329.

[48] G.J. Lastman, Sub-optimal open-loop control of non-linear systems using approximations for the controls, Internat. J. Control 20 (2) (1974) 289–303, http://dx.doi.org/10.1080/00207177408932738.

[49] W.E. Williamson, Use of polynomial approximations to calculate suboptimal controls, AIAA J. 9 (11) (1971) 2271–2273, http://dx.doi.org/10.2514/3.6499.

[50] G.A. Hicks, W.H. Ray, Approximation methods for optimal control synthesis, Can. J. Chem. Eng. 49 (4) (1971) 522–528, http://dx.doi.org/10.1002/cjce.5450490416.

[51] R.W.H. Sargent, G.R. Sullivan, The development of an efficient optimal control package, in: J. Stoer (Ed.), Optimization Techniques, in: Lecture Notes in Control and Information Sciences, Vol. 7, Springer, Berlin, Heidelberg, 1978, pp. 158–168, http://dx.doi.org/10.1007/BFb0006520.

[52] W.F. Feehery, P.I. Barton, Dynamic simulation and optimization with inequality path constraints, Comput. Chem. Eng. 20 (1996) S707–S712, http://dx.doi.org/10.1016/0098-1354(96)00127-5.

[53] L.T. Biegler, Nonlinear programming strategies for dynamic chemical process optimization, Theor. Found. Chem. Eng. 48 (5) (2014) 541–554, http://dx.doi.org/10.1134/S0040579514050157.

[54] L.R. Petzold, A Description of DASSL: A Differential/Algebraic System Solver, Sandia Report, (SAND82-8637) Sandia National Laboratories, Livermore, CA, 1982.

[55] L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, SIAM J. Sci. Comput. 18 (1) (1997) 1–22, http://dx.doi.org/10.1137/S1064827594276424.

[56] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, ACM Trans. Math. Software 31 (3) (2005) 363–396, http://dx.doi.org/10.1145/1089014.1089020.

[57] J.W. Evans, J. Szekely, W.H. Ray, Y.K. Chuang, On the optimum temperature progression for irreversible non-catalytic gas—solid reactions, Chem. Eng. Sci. 28 (3) (1973) 683–690, http://dx.doi.org/10.1016/0009-2509(77)80002-X.

[58] V.S. Vassiliadis, R.W. Sargent, C.C. Pantelides, Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints, Ind. Eng. Chem. Res. 33 (9) (1994) 2111–2122, http://dx.doi.org/10.1021/ie00033a014.

[59] A. Mitsos, B. Chachuat, P.I. Barton, McCormick-based relaxations of algorithms, SIAM J. Optim. 20 (2) (2009) 573–601, http://dx.doi.org/10.1137/080717341.

[60] D. Bongartz, A. Mitsos, Deterministic global optimization of process flowsheets in a reduced space using McCormick relaxations, J. Global Optim. 69 (4) (2017) 761–796, http://dx.doi.org/10.1007/s10898-017-0547-4.

[61] D. Bongartz, Deterministic Global Flowsheet Optimization for the Design of Energy Conversion Processes, Dissertation, RWTH Aachen University, Aachen, Germany, 2020.

[62] A. Guillou, J.L. Soulé, La résolution numérique des problèmes différentiels aux conditions initiales par des méthodes de collocation, Rev. Fr. Inf. Rech. Oper. Ser. Rouge 3 (R3) (1969) 17–44, http://dx.doi.org/10.1051/m2an/196903R300171.

[63] K. Wright, Some relationships between implicit Runge–Kutta, collocation and Lanczos τ methods, and their stability properties, BIT 10 (2) (1970) 217–227, http://dx.doi.org/10.1007/BF01936868.

[64] H. Brunner, Collocation methods for Volterra integral and related functional differential equations, in: Cambridge Monographs on Applied and Computational Mathematics, Vol. 15, Cambridge University Press, Cambridge, UK, New York, 2004.

[65] M. Zennaro, Natural continuous extensions of Runge–Kutta methods, Math. Comp. 46 (173) (1986) 119, http://dx.doi.org/10.2307/2008218.

[66] A.M. Cervantes, L.T. Biegler, A stable elemental decomposition for dynamic process optimization, J. Comput. Appl. Math. 120 (1) (2000) 41–57, http://dx.doi.org/10.1016/S0377-0427(00)00302-2.

[67] R. Amrit, J.B. Rawlings, L.T. Biegler, Optimizing process economics online using model predictive control, Comput. Chem. Eng. 58 (2013) 334–343, http://dx.doi.org/10.1016/j.compchemeng.2013.07.015.

[68] C.D. Kappatou, D. Bongartz, J. Najman, S. Sass, A. Mitsos, Global dynamic optimization with Hammerstein–Wiener models embedded, J. Global Optim. 84 (2) (2022) 321–347, http://dx.doi.org/10.1007/s10898-022-01145-z.

[69] S. Sass, A. Mitsos, D. Bongartz, I.H. Bell, N.I. Nikolov, A. Tsoukalas, A branch-and-bound algorithm with growing datasets for large-scale parameter estimation, European J. Oper. Res. 316 (1) (2024) 36–45, http://dx.doi.org/10.1016/j.ejor.2024.02.020.

[70] J.S. Logsdon, L.T. Biegler, Decomposition strategies for large-scale dynamic optimization problems, Chem. Eng. Sci. 47 (4) (1992) 851–864, http://dx.doi.org/10.1016/0009-2509(92)80272-E.

[71] P. Li, H.A. Garcia, G. Wozny, E. Reuter, Optimization of a semibatch distillation process with model validation on the industrial site, Ind. Eng. Chem. Res. 37 (4) (1998) 1341–1350, http://dx.doi.org/10.1021/ie970695l.

[72] W. Hong, S. Wang, P. Li, G. Wozny, L.T. Biegler, A quasi-sequential approach to large-scale dynamic optimization problems, AIChE J. 52 (1) (2006) 255–268, http://dx.doi.org/10.1002/aic.10625.

[73] J. Albersmeyer, M. Diehl, The lifted Newton method and its application in optimization, SIAM J. Optim. 20 (3) (2010) 1655–1684, http://dx.doi.org/10.1137/080724885.

[74] B.A. Ogunnaike, W.H. Ray, Process Dynamics, Modeling, and Control, Vol. 1, Oxford University Press, New York, 1994.

[75] C.A. Floudas, X. Lin, Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review, Comput. Chem. Eng. 28 (11) (2004) 2109–2129, http://dx.doi.org/10.1016/j.compchemeng.2004.05.002.

[76] C.A. Méndez, J. Cerdá, I.E. Grossmann, Iiro Harjunkoski, Marco Fahl, State-of-the-art review of optimization methods for short-term scheduling of batch processes, Comput. Chem. Eng. 30 (6–7) (2006) 913–946, http://dx.doi.org/10.1016/j.compchemeng.2006.02.008.

[77] Q. Zhang, I.E. Grossmann, Enterprise-wide optimization for industrial demand side management: Fundamentals, advances, and perspectives, Chem. Eng. Res. Des. 116 (2016) 114–131, http://dx.doi.org/10.1016/j.cherd.2016.10.006.

[78] M. Bodenstein, Eine theorie der photochemischen reaktionsgeschwindigkeiten, Z. Phys. Chem. 85U (1) (1913) 329–397, http://dx.doi.org/10.1515/zpch-1913-8512.

[79] L. Michaelis, M.L. Menten, Die kinetik der invertinwirkung, Biochem. Z. 49 (1913) 333–369.

[80] G.E. Briggs, J.B. Haldane, A note on the kinetics of enzyme action, Biochem. J. 19 (2) (1925) 338–339, http://dx.doi.org/10.1042/bj0190338.

[81] J.R. Bowen, A. Acrivos, A.K. Oppenheim, Singular perturbation refinement to quasi-steady state approximation in chemical kinetics, Chem. Eng. Sci. 18 (3) (1963) 177–188, http://dx.doi.org/10.1016/0009-2509(63)85003-4.

[82] W.H. Ray, The quasi–steady–state approximation in continuous stirred tank reactors, Can. J. Chem. Eng. 47 (6) (1969) 503–508, http://dx.doi.org/10.1002/cjce.5450470601.

[83] B. Srinivasan, A guide to the Michaelis–Menten equation: steady state and beyond, FEBS J. 289 (20) (2022) 6086–6098, http://dx.doi.org/10.1111/febs.16124.

[84] F.G. Heineken, H.M. Tsuchiya, R. Aris, On the mathematical status of the pseudo-steady state hypothesis of biochemical kinetics, Math. Biosci. 1 (1) (1967) 95–113, http://dx.doi.org/10.1016/0025-5564(67)90029-6.

[85] P. Muller, Glossary of terms used in physical organic chemistry (IUPAC recommendations 1994), Pure Appl. Chem. 66 (5) (1994) 1077–1184, http://dx.doi.org/10.1351/pac199466051077.

[86] M.A. LeMone, Boundary layer (atmospheric) and air pollution: Convective boundary layer, in: G.R. North, J. Pyle, F. Zhang (Eds.), Encyclopedia of Atmospheric Sciences, second ed., Academic Press, Oxford, 2015, pp. 250–257.

[87] R.X. Huang, Oceanographic topics: Thermohaline circulation, in: G.R. North, J. Pyle, F. Zhang (Eds.), Encyclopedia of Atmospheric Sciences, second ed., Academic Press, Oxford, 2015, pp. 315–328.

[88] K.B. Bischoff, Accuracy of the pseudo steady state approximation for moving boundary diffusion problems, Chem. Eng. Sci. 18 (11) (1963) 711–713, http://dx.doi.org/10.1016/0009-2509(63)85050-2.

[89] L.N.G. Filon, The forces on a cylinder in a stream of viscous fluid, Proc. R. Soc. Lond. Ser. A 113 (763) (1926) 7–27, http://dx.doi.org/10.1098/rspa.1926.0136.

[90] D.C. Alsop, The sensitivity of low flip angle RARE imaging, Magn. Reson. Med. 37 (2) (1997) 176–184, http://dx.doi.org/10.1002/mrm.1910370206.

[91] Q. Zhang, A. Sundaramoorthy, I.E. Grossmann, J.M. Pinto, A discrete-time scheduling model for continuous power-intensive process networks with various power contracts, Comput. Chem. Eng. 84 (2016) 382–393, http://dx.doi.org/10.1016/j.compchemeng.2015.09.019.

[92] X. Zhang, G. Hug, I. Harjunkoski, Cost-effective scheduling of steel plants with flexible EAFs, IEEE Trans. Smart Grid 8 (1) (2017) 239–249, http://dx.doi.org/10.1109/TSG.2016.2575000.

[93] G. Cimini, Y. Kim, B. McCain, J. Siegel, A. Stefanopoulou, Model predictive control for real-time position tracking of a catenary-free tram, IFAC- Pap. 50 (1) (2017) 1000–1005, http://dx.doi.org/10.1016/j.ifacol.2017.08.206.

[94] A. Ghobeity, A. Mitsos, Optimal time-dependent operation of seawater reverse osmosis, Desalination 263 (1) (2010) 76–88, http://dx.doi.org/10.1016/j.desal.2010.06.041.

[95] P. Voll, S. Kirschbaum, A. Bardow, Evaluation of quasi-stationary simulation for the analysis of industrial energy systems, in: ECOS 2010, Lausanne, Switzerland, 2010, pp. 34–41.

[96] P. Schäfer, A.M. Schweidtmann, P.H. Lenz, H.M. Markgraf, A. Mitsos, Wavelet-based grid-adaptation for nonlinear scheduling subject to time-variable electricity prices, Comput. Chem. Eng. 132 (2020) 106598, http://dx.doi.org/10.1016/j.compchemeng.2019.106598.

[97] A. D'Amato, Y. Wang, D. Filev, E. Remes, On the tradeoffs between static and dynamic adaptive optimization for an automotive application, SAE Int. J. Commer. Veh. 10 (1) (2017) 346–352, http://dx.doi.org/10.4271/2017-01-0605.

[98] S. Sass, A. Mitsos, Optimal operation of dynamic (energy) systems: When are quasi-steady models adequate? Comput. Chem. Eng. 124 (2019) 133–139, http://dx.doi.org/10.1016/j.compchemeng.2019.02.011.

[99] Y. Qin, P. Liu, Z. Li, Multi-timescale hierarchical scheduling of an integrated energy system considering system inertia, Renew. Sustain. Energy Rev. 169 (2022) 112911, http://dx.doi.org/10.1016/j.rser.2022.112911.

[100] D. Zhou, K. Zhou, L. Zhu, J. Zhao, Z. Xu, Z. Shao, X. Chen, Optimal scheduling of multiple sets of air separation units with frequent load-change operation, Sep. Purif. Technol. 172 (2017) 178–191, http://dx.doi.org/10.1016/j.seppur.2016.08.009.

[101] M. Carrion, J.M. Arroyo, A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem, IEEE Trans. Power Syst. 21 (3) (2006) 1371–1378, http://dx.doi.org/10.1109/TPWRS.2006.876672.

[102] L.C. Brée, K. Perrey, A. Bulan, A. Mitsos, Demand side management and operational mode switching in chlorine production, AIChE J. 65 (7) (2019) http://dx.doi.org/10.1002/aic.16352.

[103] S. Mitra, I.E. Grossmann, J.M. Pinto, N. Arora, Optimal production planning under time-sensitive electricity prices for continuous power-intensive processes, Comput. Chem. Eng. 38 (2012) 171–184, http://dx.doi.org/10.1016/j.compchemeng.2011.09.019.

[104] F.J. Baader, P. Althaus, A. Bardow, M. Dahmen, Dynamic ramping for demand response of processes and energy systems based on exact linearization, J. Process Control 118 (2022) 218–230, http://dx.doi.org/10.1016/j.jprocont.2022.08.017.

[105] F.J. Baader, P. Althaus, A. Bardow, M. Dahmen, Demand response for flat nonlinear MIMO processes using dynamic ramping constraints, Comput. Chem. Eng. 172 (2023) 108171, http://dx.doi.org/10.1016/j.compchemeng.2023.108171.

[106] OpenAI, How ChatGPT and our foundation models are developed, 2025, URL https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-foundation-models-are-developed?utm_source=chatgpt.com#h_dc0f614b54 .(Accessed on June 21 2025).

[107] M. Gerdts, A variable time transformation method for mixed-integer optimal control problems, Optimal Control Appl. Methods 27 (3) (2006) 169–182, http://dx.doi.org/10.1002/oca.778.

[108] M. Ringkamp, S. Ober-Blöbaum, S. Leyendecker, On the time transformation of mixed integer optimal control problems using a consistent fixed integer control function, Math. Program. 161 (1) (2017) 551–581, http://dx.doi.org/10.1007/s10107-016-1023-5.

[109] P.I. Barton, C.K. Lee, Modeling, simulation, sensitivity analysis, and optimization of hybrid systems, ACM Trans. Model. Comput. Simul. 12 (4) (2002) 256–289, http://dx.doi.org/10.1145/643120.643122.

[110] J. Oldenburg, W. Marquardt, Disjunctive modeling for optimal control of hybrid systems, Comput. Chem. Eng. 32 (10) (2008) 2346–2364, http://dx.doi.org/10.1016/j.compchemeng.2007.12.002.

[111] H.G. Bock, C. Kirches, A. Meyer, A. Potschka, Numerical solution of optimal control problems with explicit and implicit switches, Optim. Methods Softw. 33 (3) (2018) 450–474, http://dx.doi.org/10.1080/10556788.2018.1449843.

[112] D.C. Montgomery, Design and Analysis of Experiments, Vol. 10, John Wiley & Sons, Arizona State University, 2019.

[113] F. Cenci, A. Pankajakshan, P. Facco, F. Galvanin, An exploratory model-based design of experiments approach to aid parameters identification and reduce model prediction uncertainty, Comput. Chem. Eng. 177 (2023) 108353, http://dx.doi.org/10.1016/j.compchemeng.2023.108353.

[114] Mathematical Optimization Society (MOS), About MOS: What is mathematical optimization?, 2025, URL https://www.mathopt.org/?nav=about. (Accessed on June 21 2025)..