

Koopman Based Trajectory Optimization with Mixed Boundaries

Mohamed Abou-Taleb *

MOHAMED.ABOU-TALEB@IAMS.UNI-STUTTGAERT.DE

Maximilian Raff *

MAXIMILIAN.RAFF@IAMS.UNI-STUTTGAERT.DE

Kathrin Flaßkamp †

KATHRIN.FLASSKAMP@UNI-SAARLAND.DE

C. David Remy *

DAVID.REMY@IAMS.UNI-STUTTGAERT.DE

* *Institute for Adaptive Mechanical Systems, University of Stuttgart, Germany*

† *Systems Modeling and Simulation, Saarland University, Germany*

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

Trajectory optimization is a widely used tool in the design and control of dynamical systems. Typically, not only nonlinear dynamics, but also couplings of the initial and final condition through implicit boundary constraints render the optimization problem non-convex. This paper investigates how the Koopman operator framework can be utilized to solve trajectory optimization problems in a (partially) convex fashion. While the Koopman operator has already been successfully employed in model predictive control, the challenge of addressing mixed boundary constraints within the Koopman framework has remained an open question. We first address this issue by explaining why a complete convexification of the problem is not possible. Secondly, we propose a method where we transform the trajectory optimization problem into a bilevel problem in which we are then able to convexify the high-dimensional lower-level problem. This separation yields a low-dimensional upper-level problem, which could be exploited in global optimization algorithms. Lastly, we demonstrate the effectiveness of the method on two example systems: the mathematical pendulum and the compass-gait walker.

Keywords: Bilevel Optimization, Periodic Optimization, Convexification, Koopman Generator, EDMD, Optimal Gaits, Compass-Gait Walker

1. Introduction

Trajectory optimization problems with mixed boundary conditions (MBCs) that couple initial and terminal conditions through implicit constraints, are of significant interest across various applications. For example, in the field of legged locomotion, trajectory optimization is used as a tool for the design and control of legged robots (Wensing et al., 2024). Here, MBCs emerge because we want to find optimally actuated periodic solutions for which an initial condition can only be implicitly defined by a periodicity constraint. Furthermore, the non-smooth nature of the contact dynamics models render the boundary constraints nonlinear, which adds to the computational challenge.

Due to the nonlinear dynamics, the nonlinear MBCs, and the unknown period time, trajectory optimization problems are typically non-convex. Finding a (globally) optimal solution is thus difficult if not impossible, as numerical solvers are of local nature. Therefore, the quality of the obtained (locally) optimal solution heavily depends on the initial guess provided by the user.

Figure 1 illustrates a simple trajectory optimization problem with MBCs. Although the cost function is convex in the control input $u(\cdot)$, the dynamics are linear, and the MBCs are affine, incorporating the period time T as a decision variable makes the optimization problem non-convex. However, for any fixed T , the optimization problem remains convex, allowing us to uniquely determine the optimal solution for that fixed period. Exploiting this structure, we can formulate a

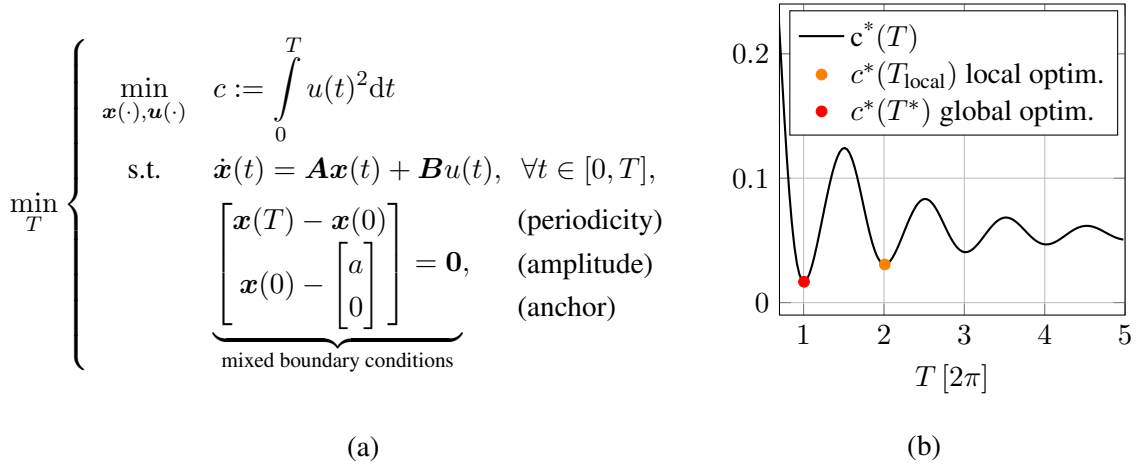


Figure 1: Trajectory optimization problem of the harmonic oscillator shown in (a) and its solution family for fixed periods T in (b). The oscillator ($\mathbf{A} = \begin{bmatrix} 1 & 0 \\ -1 & -0.2 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$) operates at a desired amplitude of $a = 30$ while maintaining periodicity, satisfying mixed boundary conditions. These conditions, being affine in boundary states with linear dynamics, ensure a unique solution for fixed T . However, allowing T to vary introduces non-convexity, leading to multiple local minima as illustrated in (b).

secondary problem that optimizes over T in a one-dimensional space. In this work, we focus on approximating a broad class of trajectory optimization problems with MBCs, particularly in the context of nonlinear dynamics, by leveraging Koopman operator theory. This approach enables us to achieve a structure similar to the introductory example presented in Figure 1, where the subproblem involves linear dynamics, making it convex.

In optimal control, convexity is a fundamental property for the efficient synthesis of control policies and system dynamics (Boyd and Vandenberghe, 2004). Consequently, the convex formulation of optimization problems has attracted significant attention (Horst and Tuy, 1996; Scherer, 2006). A prominent method to achieving this is (exact) convexification, which transforms the original optimization problem into a convex one. A well-known technique in this context is semidefinite relaxation, which expands non-convex constraint spaces into convex ones, often by introducing slack variables to “lift” the problem into a higher-dimensional space (Açikmeşe and Blackmore, 2011; Lasserre, 2001). If the relaxation is proven to be tight, equivalence with the original problem is guaranteed. Another approach to (exact) convexification, explored in this paper, leverages Koopman operator theory (Koopman, 1931). This method lifts the nonlinear dynamics into a higher-dimensional space, where they are represented by linear dynamics, thereby enabling a convex problem formulation. However, since the Koopman operator is typically infinite-dimensional and challenging to compute, approximate methods, such as the extended dynamic mode decomposition (EDMD), are employed to lift the dynamics into finite-dimensional spaces (Williams et al., 2015; Alexandre Mauroy, 2020; Williams et al., 2016; Iacob et al., 2024; Proctor et al., 2018). This approach has been successfully implemented in model predictive control (MPC), where it has enabled the optimization problems to be solved efficiently and in real time (Bruder et al., 2019; Korda and Mezić, 2018; Kanai and Yamakita, 2022; Schaller et al., 2023). While those works apply the Koopman framework to MPC, they do not consider mixed boundary constraints, since in MPC the prediction horizon and

the initial condition are fixed. In their review of the Koopman operator in robot learning, [Shi et al. \(2024\)](#) discuss the challenge of incorporating constraints within the Koopman framework, which remains an open problem.

In this work, we investigate how the Koopman framework may be leveraged to simplify trajectory optimization problems with mixed boundary constraints. We explain why a complete convexification is not possible. Furthermore, we propose a method where the original problem is approximated by formulating a bilevel optimization problem, encompassing both an upper- and a lower-level problem. This structure allows us to solve the lower-level with fixed boundaries and fixed terminal time as a convex optimization problem which is similar to the MPC subproblem. We present three different ways to formulate the boundary constraints in the lower-level. The upper-level problem then optimizes the boundary values and the terminal time. That is, while we still need to solve a nonlinear program, it has a significantly reduced dimensionality. The efficacy of the presented method is investigated along two examples from periodic trajectory optimization.

2. Theory

2.1. Problem Definition

We aim to solve a trajectory optimization problem with mixed boundary constraints (MBCs) of the following form:

$$\mathcal{P} : \begin{cases} \underset{\mathbf{x}(\cdot), \mathbf{u}(\cdot), T}{\text{minimize}} & c(\mathbf{x}(\cdot), \mathbf{u}(\cdot), T) \\ \text{subject to} & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{G}(\mathbf{x}(t))\mathbf{u}(t), \quad \forall t \in [0, T], \\ & \mathbf{b}(\mathbf{x}(0), \mathbf{x}(T), T) = \mathbf{0}. \end{cases} \quad (1a)$$

$$(1b)$$

The trajectories $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ and $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ provide the state and input of a control-affine dynamical system with the dynamics $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ and $\mathbf{G} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x \times n_u}$, which is evaluated from time $t = 0$ to the terminal time T . The cost is given with the functional c which maps state- and input trajectories and the terminal time to cost. We assume c to be jointly convex in the state and input. The MBCs are defined in an implicit form with the function $\mathbf{b} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_g}$. We assume that all functions introduced above are smooth.

Solving \mathcal{P} is inherently challenging due to its pronounced non-convexity, stemming from the variable terminal time T , the nonlinear dynamics (1a) and the nonlinear MBCs (1b).

2.2. Koopman Generator Surrogate Modeling

To deal with the nonlinearity in the dynamics (1a), we can lift the dynamical system into an infinite-dimensional space, in which the dynamics are linear ([Koopman, 1931](#)). In the following, we briefly summarize, how we obtain a finite-dimensional approximation of these lifted dynamics.

Let us initially limit ourselves to the time-autonomous case, in which the input is equal to zero ($\mathbf{u} \equiv \mathbf{0}$), such that the dynamics are governed only by the differential equation $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$. Let $\psi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$, with $n_z > n_x$, be a nonlinear function, also known as *observable* or *lifting function*, living itself in a Banach-space, which we denote by \mathcal{F} . The family of Koopman operators $\mathcal{K}_t : \mathcal{F} \rightarrow \mathcal{F}$ parameterized by the time t is defined as $(\mathcal{K}_t \psi)(\mathbf{x}_0) = (\psi \circ \varphi_t)(\mathbf{x}_0) = \psi(\mathbf{x}(t))$,

where $\varphi_t(x_0)$ denotes the solution of the system (1a) with the initial condition x_0 . The Koopman operator is linear but infinite dimensional, as it acts on functions. Moreover, in the context of continuous-time flows, it is possible to define the infinitesimal Koopman generator as $\mathcal{L}\psi = \lim_{t \rightarrow 0} 1/t(\mathcal{K}_t\psi - \psi)$; i.e., the time derivative of the lifting function along solutions, also known as the Lie derivative:

$$\frac{d}{dt}\psi(x(t)) = (\mathcal{L}\psi)(x(t)) = \nabla_x \psi(x(t)) \cdot f(x(t)). \quad (2)$$

As in the case of the Koopman operator, the Koopman generator is linear and infinite-dimensional.

When we now re-introduce control inputs $u(t)$, Peitz et al. (2020) state in Theorem 3.2 that the Koopman generators inherit the property of control-affinity. Consequently, the Koopman generator $\mathcal{L}_{u(t)}$ for a known input function $u(t)$ may be expressed as $\mathcal{L}_{u(t)} = \mathcal{L}_0 + \sum_{i=0}^{n_u} u_i(t)(\mathcal{L}_i - \mathcal{L}_0)$, where $u(t) = \sum_{i=1}^{n_u} u_i(t)e_i$ with the i th canonical basis vector e_i , \mathcal{L}_0 is the Koopman generator for $u \equiv 0$ and each \mathcal{L}_i denotes the Koopman generator for each basis element of $u(t)$. Thus, for control affine systems, the Lie derivative of the observables yields

$$\frac{d}{dt}\psi(x(t)) = (\mathcal{L}_{u(t)}\psi)(x(t)) = (\mathcal{L}_0\psi)(x(t)) + \sum_{i=0}^{n_u} u_i(t)(\mathcal{L}_i - \mathcal{L}_0)\psi(x(t)), \quad (3)$$

which is bilinear in the lifted state $\psi(x)$ and the input $u(t)$.

Here, we focus on approximating the Koopman generator rather than the Koopman operator, because we aim to ultimately optimize over the terminal time T . When discretizing the optimization problem \mathcal{P} with a fixed number of discrete time steps, changes in T would directly affect the step size and would thus require a repeated identification of the Koopman operator. In contrast, the Koopman generator allows us to use a continuous-time dynamics description, which is identified once. This can then be applied to discretizations over arbitrary grids.

To obtain a finite-dimensional approximation of the Koopman generator, we use the *generator Extended Dynamic Mode Decomposition* (gEDMD), proposed by Klus et al. (2020), which approximates the generator as:

$$\frac{d}{dt}\psi(x(t)) \approx L_0\psi(x(t)) + \sum_{i=1}^{n_u} u_i(t)(L_i - L_0)\psi(x(t)), \quad (4)$$

where $L_i \in \mathbb{R}^{n_z \times n_z}$, $i \in \{0, \dots, n_u\}$, are finite dimensional matrices identified from data. While in this work, we assume that the dynamical system is known and measurements of \dot{x} are available, our approach may be extended to the case of unknown dynamical systems by applying techniques proposed by Mauroy and Goncalves (2020); Meng et al. (2024).

2.3. Koopman-Based Trajectory Optimization

Exact convexification of \mathcal{P} using Koopman operator theory would be possible if the terminal time T and the initial state $x(0)$ are explicitly defined by the MBCs (1b). In this case, we could simply lift the initial state with the observables to yield a lifted initial state $z(0) = \psi(x(0))$. This does not only ensure that the resulting solution corresponds to a trajectory starting at $x(0)$, but also that the lifted trajectory is bound to the manifold \mathcal{M} , defined by the observable functions:

$$\mathcal{M} := \{z \in \mathbb{R}^{n_z} \mid z = \psi(x), x \in \mathbb{R}^{n_x}\}. \quad (5)$$

Analogously, this process could be done with the final state $\mathbf{x}(T)$, or for any other time instant $\bar{t} \in [0, T]$ along the trajectory.

For general MBCs (1b), however, the boundary states and terminal time are not explicitly defined. While the implicit function \mathbf{b} could be convexified through a suitable choice of observables ψ , yielding a corresponding lifted constraint $\mathbf{b}_{\text{lift}}(\mathbf{z}(0), \mathbf{z}(T), T) = \mathbf{0}$, this alone is insufficient to guarantee a valid solution: the lifted constraint can be satisfied by choices of $\mathbf{z}(0)$ and $\mathbf{z}(T)$ that do not lie within the manifold \mathcal{M} . Therefore, an additional constraint is required to ensure that, for some \bar{t} , it holds: $\mathbf{z}(\bar{t}) \in \mathcal{M}$. Since ψ is nonlinear, this additional constraint may not be convex, meaning the resulting lifted optimization problem could be non-convex, even if \mathbf{b} or \mathbf{b}_{lift} are convex.

A similar issue arises with the terminal time T (and other parameters). One approach to address the issue of an unknown terminal time T (as illustrated in the introductory example in Figure 1) is to scale all derivatives by $1/T$ and evaluate the dynamics over a normalized time interval $\tau \in [0, 1]$. This allows us to treat T as a parameter, which could be included into the dynamics as an additional state with $T'(\tau) = 0$, resulting in an augmented lifted state $\mathbf{z}(\tau) = \psi(\mathbf{x}(\tau), T(\tau))$. However, since neither $T(0)$ nor $T(1)$ are explicitly defined, the same problem arises as above: $\mathbf{z}(\bar{\tau})$ must be constrained to lie on \mathcal{M} for some time $\bar{\tau} \in [0, 1]$, requiring an additional constraint that may be non-convex.

To address these issues, we approximate \mathcal{P} within a bilevel optimization problem $\hat{\mathcal{P}}$. This encompasses a non-convex upper-level optimization problem $\hat{\mathcal{P}}_{\text{upper}}$ and a convex approximation of the lower-level optimization problem $\hat{\mathcal{P}}_{\text{lower}}$. Note that the approximative nature of this problem stems solely from the approximation of the lower-level problem and not in the bilevel structure itself. As a consequence, the global minimum of $\hat{\mathcal{P}}_{\text{lower}}$ can be found without providing an initial guess for the state- and input trajectories, while the upper-level optimizes over a low-dimensional space, where it is easier to provide a good initial guess for the decision variables. We define the upper-level problem as

$$\hat{\mathcal{P}}_{\text{upper}} : \begin{cases} \underset{\mathbf{x}_0, \mathbf{x}_T, T}{\text{minimize}} & c(\mathbf{x}^*(\cdot), \mathbf{u}^*(\cdot), T) \\ \text{subject to} & (\mathbf{z}^*(\cdot), \mathbf{u}^*(\cdot)) = \hat{\mathcal{P}}_{\text{lower}}(\mathbf{x}_0, \mathbf{x}_T, T), \\ & \mathbf{x}^*(\cdot) = \mathbf{C}\mathbf{z}^*(\cdot), \\ & \mathbf{b}(\mathbf{x}_0, \mathbf{x}_T, T) = \mathbf{0}, \end{cases}$$

and the lower-level as

$$\hat{\mathcal{P}}_{\text{lower}}(\mathbf{x}_0, \mathbf{x}_T, T) : \begin{cases} \underset{\mathbf{z}(\cdot), \mathbf{u}(\cdot)}{\text{argmin}} & (1 - w_i) \cdot c(\mathbf{C}\mathbf{z}(\cdot), \mathbf{u}(\cdot); T) + w_i \cdot \hat{c}_{\text{lower}}(\mathbf{z}(0), \mathbf{z}(T); \mathbf{x}_0, \mathbf{x}_T) \\ \text{subject to} & \dot{\mathbf{z}}(t) = \mathbf{L}_0\mathbf{z}(t) + \mathbf{L}_u(\bar{\mathbf{z}})\mathbf{u}(t), \quad \forall t \in [0, T], \\ & \hat{\mathbf{b}}_i(\mathbf{z}(0), \mathbf{z}(T); \mathbf{x}_0, \mathbf{x}_T) = \mathbf{0}, \end{cases}$$

where the lift $\mathbf{z} = \psi(\mathbf{x})$ includes a copy of \mathbf{x} such that we can recover the original state by applying a linear operation $\mathbf{x} = \mathbf{C}\mathbf{z}$, with $\mathbf{C} = [\mathbf{I}_{n_x \times n_x} \mathbf{0}]$.

In the upper-level $\hat{\mathcal{P}}_{\text{upper}}$, we introduce the initial state \mathbf{x}_0 and final state \mathbf{x}_T as decision variables, together with the terminal time T . We enforce the MBCs on \mathbf{x}_0 , \mathbf{x}_T and T in a nonlinear fashion, while the cost is evaluated by obtaining optimal state- and input trajectories for fixed boundary values and terminal time in the lower-level $\hat{\mathcal{P}}_{\text{lower}}$. Note that the upper-level $\hat{\mathcal{P}}_{\text{upper}}$ is a low-dimensional non-convex problem.

For each iteration of $\hat{\mathcal{P}}_{\text{upper}}$, the bilinear dynamics (4) are linearized in the lower-level $\hat{\mathcal{P}}_{\text{lower}}$ around a chosen point $\bar{z} = \psi(\bar{x})$ with $u = 0$. This results in the following lifted linear dynamics:

$$\dot{z}(t) = L_0 z(t) + L_u(\bar{z}) u(t), \quad (6)$$

where $L_u \in \mathbb{R}^{n_x \times n_u}$ is the input matrix of the resulting LTI dynamics. Furthermore, we incorporate an additional cost \hat{c}_{lower} with weights w_i and boundary constraints \hat{b}_i into $\hat{\mathcal{P}}_{\text{lower}}$.

The implementations of \hat{b}_i must include $Cz(0) = x_0$ and $Cz(T) = x_T$, ensuring that the lifted state adheres to the original state constraints. As discussed above, this alone is insufficient, as z must additionally be constrained to \mathcal{M} . A naive approach to ensure this, would be to fully lift both boundary conditions, setting $z(0) = \psi(x_0)$ and $z(T) = \psi(x_T)$. However, this leads to practical issues if the lifted state z drifts away from \mathcal{M} due to the approximative nature of L . Since the lifted linear dynamics (6) are not necessarily controllable in directions orthogonal to \mathcal{M} , no input u would be able to steer the lifted state back onto \mathcal{M} .

Alternatively, this work explores and compares three different approaches ($i \in \{0, T, \text{soft}\}$) for lifting the boundary conditions. For the first two, we simply apply the lift to only one of the boundaries, yielding the two implicit formulations:

$$\hat{b}_0(z(0), z(T); x_0, x_T) = \begin{bmatrix} z(0) - \psi(x_0) \\ Cz(T) - x_T \end{bmatrix}, \quad (7a)$$

$$\hat{b}_T(z(0), z(T); x_0, x_T) = \begin{bmatrix} Cz(0) - x_0 \\ z(T) - \psi(x_T) \end{bmatrix}, \quad (7b)$$

where the weights $w_0 = w_T = 0$ are chosen to exclusively account for the original cost c . As a third approach, we implement the requirements $z(0) = \psi(x_0)$ and $z(T) = \psi(x_T)$ as soft constraints, by invoking the additive penalty cost \hat{c}_{lower} with weights $w_{\text{soft}} \in (0, 1)$:

$$\hat{c}_{\text{lower}}(z(0), z(T); x_0, x_T) = \|z(0) - \psi(x_0)\|^2 + \|z(T) - \psi(x_T)\|^2, \quad (7c)$$

$$\hat{b}_{\text{soft}}(z(0), z(T); x_0, x_T) = \begin{bmatrix} Cz(0) - x_0 \\ Cz(T) - x_T \end{bmatrix}. \quad (7d)$$

Note: Equations (7a) and (7b) will enforce the lifted state to lie on \mathcal{M} only at the beginning or end of the trajectory, respectively. Since \mathcal{M} is only approximately Koopman-invariant under L , the lifted state will drift away from \mathcal{M} throughout the remainder of the trajectory. Equations (7c) and (7d) seek to balance this drift across both boundaries, ideally yielding a solution that is closer to \mathcal{M} overall. Yet, it may be prone to the ill-conditioning associated with soft constraints (Betts, 2010).

Finally, we have to select an appropriate point of linearization for each case. Given that the only known points on the solution trajectory are the initial and terminal state, it is reasonable to use these for the linearization. In the case where we lift the initial constraints \hat{b}_0 , we should choose $\bar{z} = \psi(x_0)$. If we lift the terminal constraint \hat{b}_T , we should choose $\bar{z} = \psi(x_T)$. In the case of soft constraints, both options are equally valid. This linearization yields L_u from L_i , $i \in \{1, \dots, n_u\}$ and is conducted repeatedly for each call to the lower-level problem.

The cost functions c and \hat{c}_{lower} , as well as convex combinations of these functions, are jointly convex in the decision variables $z(\cdot)$ and $u(\cdot)$. Furthermore, the constraints $\hat{b}_i(z(0), z(T); x_0, x_T)$, with $i \in \{0, T, \text{soft}\}$, are affine and define a convex feasible set. Consequently, $\hat{\mathcal{P}}_{\text{lower}}$ is a convex optimization problem.

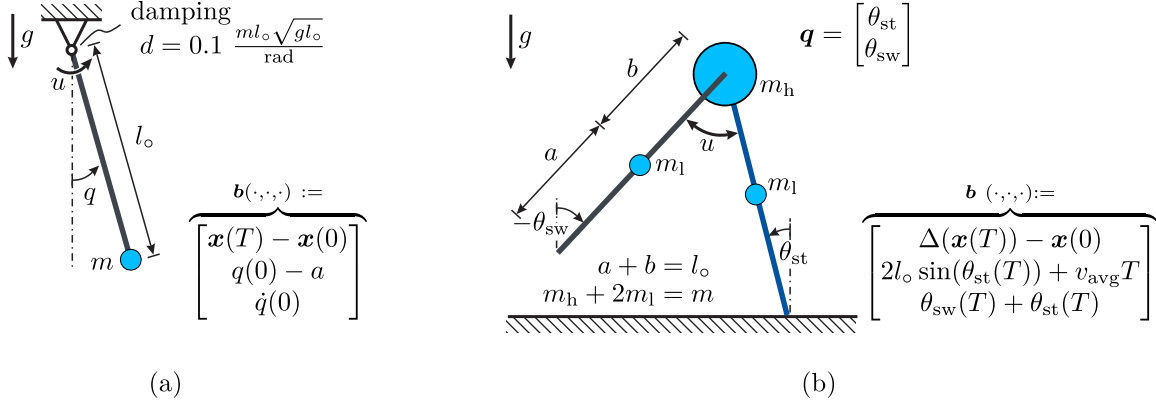


Figure 2: Illustrated are two mechanical systems: (a) a mathematical pendulum and (b) a compass-gait walker. The MBCs \mathbf{b} include periodicity, operating, and anchor constraints. For (a), the operating constraint is the desired amplitude a , and for (b), it is the desired average forward velocity v_{avg} . The system states are $\mathbf{x}^\top = [\mathbf{q}^\top \dot{\mathbf{q}}^\top]$, with all parameters normalized by gravity g , length l_o and mass m .

3. Examples from Periodic Trajectory Optimization

To investigate the efficacy of the presented method, we consider two (periodic) example systems, namely a mathematical pendulum and a compass-gait walker (Figure 2). For the numerical evaluation of these problems, all system parameters are normalized with respect to gravity g , length l_o and mass m . To numerically solve the optimization problems \mathcal{P} and $\hat{\mathcal{P}}$, the input space is approximated with piecewise constant functions. Nonlinear dynamics within \mathcal{P} are approximated with an explicit fourth-order Runge-Kutta integration scheme whereas linear dynamics within $\hat{\mathcal{P}}_{\text{lower}}$ are discretized exactly using matrix exponentials. In both examples, we utilize the cost function $c(\mathbf{x}(\cdot), \mathbf{u}(\cdot), T) = \int_0^T u(t)^2 dt$. An optimal solution for $\hat{\mathcal{P}}$ is obtained utilizing MATLAB's `fmincon` for $\hat{\mathcal{P}}_{\text{upper}}$ and `quadprog` for $\hat{\mathcal{P}}_{\text{lower}}$. To evaluate the solution of our proposed bilevel optimization problem, we also solve the resulting nonlinear program of the original problem \mathcal{P} , again using `fmincon`. Here, we use the solution of $\hat{\mathcal{P}}$ as the initial guess for \mathcal{P} . For further details on the implementation, including the specific choice of lifting functions, we refer to the actual code that is available on [GitHub](https://github.com/MohamedAbou-Taleb/KoopmanBasedTrajectoryOptimizationWithMixedBoundaries)¹.

3.1. Mathematical Pendulum

As a nonlinear extension of the harmonic oscillator trajectory optimization problem presented in the introduction (Figure 1), we consider the simple pendulum (Figure 2a). Apart from the nonlinear dynamics, the optimization problem is identical, seeking periodic trajectories with a desired amplitude a that are anchored at a velocity of $\dot{q}(0) = 0$. This problem was rewritten in the bilevel form $\hat{\mathcal{P}}$, introduced in Section 2.3. To convexify the dynamics, we lifted them by identifying a bilinear Koopman generator surrogate model. To this end, we utilized a 12-dimensional Koopman basis consisting of trigonometric functions and polynomials of the state. We collected data by sampling

1. The code can be found at the following GitHub repository: <https://github.com/MohamedAbou-Taleb/KoopmanBasedTrajectoryOptimizationWithMixedBoundaries>

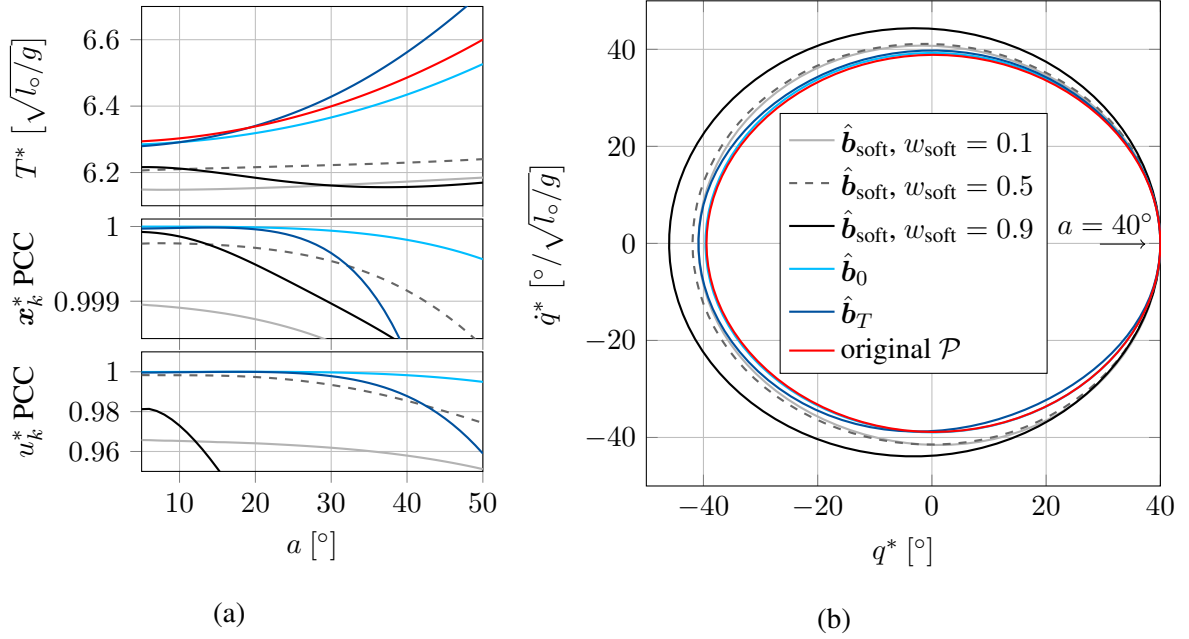


Figure 3: Results of the mathematical pendulum, compared across different choices for the boundary constraints \hat{b}_i and weights w_{soft} . The discretization utilizes 101 points. (a) evaluates the similarity of optimal period T^* , optimal states $x^*(\cdot)$ and optimal inputs $u^*(\cdot)$, as a function of amplitude a between the approximated problem and the original NLP. Similarity of trajectories is expressed via the Pearson correlation coefficient (PCC). (b) shows a phase portrait for an amplitude of $a = 40^\circ$.

45000 uniformly distributed points in the state space, subsequently lifted them, and computed their Lie-derivatives. With this data we solved two least-squares problems to obtain the matrices L_0 , L_1 for the bilinear approximation of the Koopman generator. We solved this problem with all three versions of the lifted boundary conditions \hat{b}_i , including $w_{\text{soft}} \in \{0.1, 0.5, 0.9\}$. Figure 3 compares these results with the numerically obtained solution of the original problem \mathcal{P} . In all cases, we linearized the dynamics about $\bar{z} = \psi(x_0)$.

Figure 3a shows the similarity of optimal period time T^* , optimal states $x^*(\cdot)$ and optimal inputs $u^*(\cdot)$, as a function of amplitude a . The similarities of the latter are expressed via the Pearson correlation coefficient (PCC), for which a full agreement between original and approximated trajectories would yield a value of 1. We observe that \hat{b}_0 and \hat{b}_T lead to the best agreement, while the soft constraint are introducing a larger approximation error, in particular if the relative weight of the two cost components is unbalanced.

For an amplitude of $a = 40^\circ$, the cost of \mathcal{P} yields $c = 1.48 \cdot 10^{-2} (mgl_o)^2 \sqrt{l_o/g}$, while the hard constraints with the lift on the initial- and the terminal condition yield a cost of $c = 3.34 \cdot 10^{-2} (mgl_o)^2 \sqrt{l_o/g}$ and $c = 1.53 \cdot 10^{-2} (mgl_o)^2 \sqrt{l_o/g}$ as well as $\hat{c}_{\text{lower}} = 23.88 \cdot 10^{-2}$ and $\hat{c}_{\text{lower}} = 102.21 \cdot 10^{-2}$, respectively. The costs in the case of the soft constraints evaluate to $c = 0.22 \cdot 10^{-2} (mgl_o)^2 \sqrt{l_o/g}$ and $\hat{c}_{\text{lower}} = 3.013 \cdot 10^{-2}$ for $w_{\text{soft}} = 0.1$; $c = 1.02 \cdot 10^{-2} (mgl_o)^2 \sqrt{l_o/g}$ and $\hat{c}_{\text{lower}} = 1.34 \cdot 10^{-2}$ for $w_{\text{soft}} = 0.5$; as well as $c = 4.00 \cdot 10^{-2} (mgl_o)^2 \sqrt{l_o/g}$ and $\hat{c}_{\text{lower}} = 0.84 \cdot 10^{-2}$ for $w_{\text{soft}} = 0.9$. As expected, for the soft constraints, we see a clear trade-off between cost c and accuracy of the solution, as measured by \hat{c}_{lower} . As per our initial hypothesis,

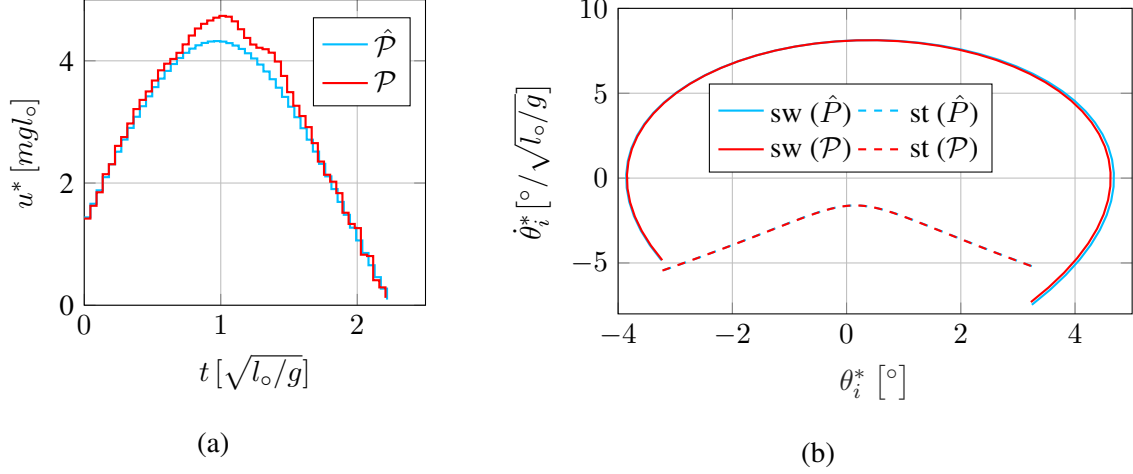


Figure 4: Comparison between the solutions of $\hat{\mathcal{P}}$ and \mathcal{P} for an average speed of $v_{\text{avg}} = 0.05\sqrt{gl_o}$, using a discretization with 51 points. The period time and the state trajectories are captured accurately. The input provides a qualitatively good approximation. As boundary constraints for the lower-level, we impose $\hat{\mathbf{b}}_0$, meaning that the lifted constraints are applied only at the initial time.

enforcing the constraints only at one boundary via the hard constraints, leads to much larger constraint violations at the other end of the lifted trajectory (expressed here via $\hat{\mathbf{c}}_{\text{lower}}$). However, when comparing this to the results presented in Figure 3, the better performance of the soft constraints in the lifted space, does not translate to a better agreement in terms of un-lifted trajectories.

3.2. Compass-Gait Walker

The compass-gait walker, shown in Figure 2b, is a bipedal robot consisting of a stance and a swing leg for which we describe the orientation relative to the vertical with θ_{st} and θ_{sw} respectively. Furthermore, a motor torque is provided at the hip. The task is to perform a periodic forward motion at the desired average speed v_{avg} , while minimizing the control effort. Due to the left-right symmetry of the system, we solely optimize over a single step and obtain a complete stride by flipping the generalized coordinates of stance and swing afterwards. In addition, velocities jump upon the collision of the swing leg with the ground. For details on continuous dynamics (1a) and the discrete jump map $\Delta : \mathbb{R}^4 \rightarrow \mathbb{R}^4$, the reader is referred to Manchester et al. (2010). The trajectory optimization problem \mathcal{P} is subject to the MBCs described in Figure 2b. These MBCs follow a structure similar to the previous examples, incorporating symmetric periodicity and the operating condition at v_{avg} . The anchor is chosen at the instance of touch down, when both legs are in contact with the ground.

Again, the problem was rewritten in the bilevel form $\hat{\mathcal{P}}$, introduced in Section 2.3. The boundary conditions were lifted at the initial time with $\hat{\mathbf{b}}_0$. To convexify the dynamics, we lifted them by identifying a bilinear Koopman generator surrogate model to apply the presented method. Here, we used a 29-dimensional lift consisting of a mix of trigonometric functions and polynomials of the state. As in the previous example, data was obtained by collecting 45000 uniformly distributed samples in the state space which were subsequently lifted and used to compute the Lie-derivatives. With this data, we solved two least-squares problems to obtain the matrices \mathbf{L}_0 , \mathbf{L}_1 for the bilinear approximation of the Koopman generator.

The solutions of $\hat{\mathcal{P}}$ are compared to those of \mathcal{P} in Figure 4 for an average speed of $v_{\text{avg}} = 0.05\sqrt{gl_o}$. We observe that the lifted dynamics are able to fairly accurately compute the optimal state- and input trajectories (with a PCC of 0.9999 and 0.9987, respectively) as well as the optimal period T^* ($2.2635\sqrt{l_o/g}$ compared to $2.2518\sqrt{l_o/g}$).

4. Discussion

In this work, we have investigated how the Koopman operator framework can be leveraged to solve non-convex trajectory optimization problems \mathcal{P} including nonlinear- dynamics and MBCs. The problem \mathcal{P} is reformulated as a bilevel optimization problem $\hat{\mathcal{P}}$, encompassing both an upper-level ($\hat{\mathcal{P}}_{\text{upper}}$) and a lower-level problem ($\hat{\mathcal{P}}_{\text{lower}}$). By exploiting this structure, we are able to achieve a convexification of $\hat{\mathcal{P}}_{\text{lower}}$. This yields a low-dimensional $\hat{\mathcal{P}}_{\text{upper}}$ which can be solved efficiently due to the convexity of $\hat{\mathcal{P}}_{\text{lower}}$. In addition, an initial guess must only be provided for the boundary values and the terminal time while the initialization of the trajectories is eliminated. The bilevel structure handles the nonlinear boundary constraints well, by including them in the upper-level and not in the lower-level. Furthermore, we have demonstrated the effectiveness of this method using two examples from the domain of periodic trajectory optimization while exploring different formulations of how the boundary constraints are represented in the lower-level. The results indicate that enforcing one boundary as a hard constraint outperforms soft constraints. For the original problem \mathcal{P} , hard constraints enable more accurate computation of the optimal states, inputs, and terminal time.

The presented method is still subject to a number of limiting factors. As demonstrated in [Otto et al. \(2024\)](#), linear time-invariant approximations may not be able to fully capture dynamics with products of state and input. While this could potentially pose a problem in cases where control inputs tend to become large, we did not find it to be an issue in our examples where the inputs remained small. Furthermore, the drift from the lifting manifold presents a challenge, particularly for long-term predictions, as the observables do not span a Koopman-invariant subspace. While re-projecting the lifted state onto the manifold during optimization could address this, it involves the observable functions and thus compromises the problem’s convexity. Future work should explore combining the presented approach with deep learning to jointly learn the Koopman generator and the observables, as proposed by [Lusch et al. \(2018\)](#) and [Han et al. \(2020\)](#). This approach could potentially reduce prediction error while requiring fewer lifting functions.

In our problem definition of \mathcal{P} , we assume that the cost must be jointly convex in the state and the input. This assumption can be potentially relaxed to the requirement that the cost function is jointly convex in the *lifted* state defined by the observables and the input. In addition, we may include nonlinear path constraints in the lower-level as long as it is possible to design the observables such that those constraints define a convex set in the lifted space. This straightforward extension would allow us to solve a larger class of problems. Finally, we could extend the optimization to include system parameters in the upper-level, similar to the terminal time, forming a co-design problem. To achieve this, it would be essential to treat these parameters as additional states with zero derivatives, ensuring that the Koopman generator does not need to be re-identified in the lower level.

In this work, we have addressed the challenge of incorporating constraints within the Koopman framework by formulating the bilevel optimization problem and investigating three different methods to translate the constraints to the lifted trajectory optimization problem.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 501862165. It was further supported through the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for Maximilian Raff. Additionally, for Mohamed Abou-Taleb, the work is supported through the Cybernetics School for Talents. The School for Talents program is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy (UP 31/1 – 422052263).

References

- Yoshihiko Susuki Alexandre Mauroy, Igor Mezić. *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Springer International Publishing, 2020. ISBN 9783030357139. doi: 10.1007/978-3-030-35713-9.
- Behçet Açıkmeşe and Lars Blackmore. Lossless convexification of a class of optimal control problems with non-convex control constraints. *Automatica*, 47(2):341–347, February 2011. ISSN 0005-1098. doi: 10.1016/j.automatica.2010.10.037.
- John T. Betts. *Practical methods for optimal control and estimation using nonlinear programming*. Society for Industrial and Applied Mathematics, January 2010. ISBN 9780898718577. doi: 10.1137/1.9780898718577.
- Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, March 2004. ISBN 9780511804441. doi: 10.1017/cbo9780511804441.
- Daniel Bruder, Brent Gillespie, C. David Remy, and Ram Vasudevan. Modeling and control of soft robots using the koopman operator and model predictive control. In *Robotics: Science and Systems XV*, RSS2019. Robotics: Science and Systems Foundation, June 2019. doi: 10.15607/rss.2019.xv.060.
- Yiqiang Han, Wenjian Hao, and Umesh Vaidya. Deep learning of koopman representation for control. In *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, December 2020. doi: 10.1109/cdc42340.2020.9304238.
- Reiner Horst and Hoang Tuy. *Global Optimization*. Springer Berlin Heidelberg, 1996. ISBN 9783662031995. doi: 10.1007/978-3-662-03199-5.
- Lucian Cristian Iacob, Roland Tóth, and Maarten Schoukens. Koopman form of nonlinear systems with inputs. *Automatica*, 162:111525, April 2024. ISSN 0005-1098. doi: 10.1016/j.automatica.2024.111525.
- Masaki Kanai and Masaki Yamakita. Linear model predictive control with lifted bilinear models by koopman-based approach. *SICE Journal of Control, Measurement, and System Integration*, 15(2):162–171, June 2022. ISSN 1884-9970. doi: 10.1080/18824889.2022.2104006.
- Stefan Klus, Feliks Nüske, Sebastian Peitz, Jan-Hendrik Niemann, Cecilia Clementi, and Christof Schütte. Data-driven approximation of the koopman generator: Model reduction, system identification, and control. *Physica D: Nonlinear Phenomena*, 406:132416, May 2020. ISSN 0167-2789. doi: 10.1016/j.physd.2020.132416.

- B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, May 1931. ISSN 1091-6490. doi: 10.1073/pnas.17.5.315.
- Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, July 2018. ISSN 0005-1098. doi: 10.1016/j.automatica.2018.03.046.
- Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, January 2001. ISSN 1095-7189. doi: 10.1137/s1052623400366802.
- Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1), November 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-07210-0.
- Ian R. Manchester, Mark M. Tobenkin, Michael Levashov, and Russ Tedrake. Regions of attraction for hybrid limit cycles of walking robots, 2010.
- Alexandre Mauroy and Jorge Goncalves. Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*, 65(6):2550–2565, June 2020. ISSN 2334-3303. doi: 10.1109/tac.2019.2941433.
- Yiming Meng, Ruikun Zhou, Melkior Ornik, and Jun Liu. Resolvent-type data-driven learning of generators for unknown continuous-time dynamical systems, 2024.
- Samuel Otto, Sebastian Peitz, and Clarence Rowley. Learning bilinear models of actuated koopman generators from partially observed trajectories. *SIAM Journal on Applied Dynamical Systems*, 23(1):885–923, March 2024. ISSN 1536-0040. doi: 10.1137/22m1523601.
- Sebastian Peitz, Samuel E. Otto, and Clarence W. Rowley. Data-driven model predictive control using interpolated koopman generators. *SIAM Journal on Applied Dynamical Systems*, 19(3): 2162–2193, January 2020. ISSN 1536-0040. doi: 10.1137/20m1325678.
- Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, January 2018. ISSN 1536-0040. doi: 10.1137/16m1062296.
- Manuel Schaller, Karl Worthmann, Friedrich Philipp, Sebastian Peitz, and Feliks Nüske. Towards reliable data-based optimal and predictive control using extended dmd. *IFAC-PapersOnLine*, 56(1):169–174, 2023. ISSN 2405-8963. doi: 10.1016/j.ifacol.2023.02.029.
- C.W. Scherer. Lmi relaxations in robust control. *European Journal of Control*, 12(1):3–29, January 2006. ISSN 0947-3580. doi: 10.3166/ejc.12.3-29.
- Lu Shi, Masih Haseli, Giorgos Mamakoukas, Daniel Bruder, Ian Abraham, Todd Murphey, Jorge Cortes, and Konstantinos Karydis. Koopman operators in robot learning, 2024.
- Patrick M. Wensing, Michael Posa, Yue Hu, Adrien Escande, Nicolas Mansard, and Andrea Del Prete. Optimization-based control for dynamic legged robots. *IEEE Transactions on Robotics*, 40:43–63, 2024. ISSN 1941-0468. doi: 10.1109/tro.2023.3324580.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, June 2015. ISSN 1432-1467. doi: 10.1007/s00332-015-9258-5.

Matthew O. Williams, Maziar S. Hemati, Scott T.M. Dawson, Ioannis G. Kevrekidis, and Clarence W. Rowley. Extending data-driven koopman analysis to actuated systems. *IFAC-PapersOnLine*, 49(18):704–709, 2016. ISSN 2405-8963. doi: 10.1016/j.ifacol.2016.10.248.