Millisecond NMPC for Swing-Up and Stabilization of the Furuta Pendulum in Real World

Hannes Homburger, Jonathan Frey, Stefan Wirtensohn, Moritz Diehl, *Member, IEEE*, and Johannes Reuter, *Member, IEEE*

***Abstract*—The Furuta pendulum's swing-up and stabilization control is currently used by many researchers to benchmark nonlinear control algorithms. In this brief, we give a systematic overview of important contributions to the control of the Furuta pendulum presented in the last 15 years. Furthermore, we use nonlinear model predictive control (NMPC) to design a real-time capable holistic controller. An optimal control problem (OCP) including a detailed nonlinear system dynamics model is defined, transcribed into a nonlinear optimization problem via direct multiple shooting, and solved in real time on an embedded system using acados. A breakthrough concerning the control performance was achieved by the usage of efficient discretization via a nonuniform grid, solving the tradeoff between a long prediction horizon and a fast sample time. The control strategy shows excellent performance in simulation and real-world experiments using a custom-made pendulum prototype. Videos of the experiments are available at: https://www.youtube.com/shorts/oJYyD5beMqM/**

***Index Terms*—Control engineering, nonlinear control systems, optimal control, predictive control.**

I. Introduction and Overview

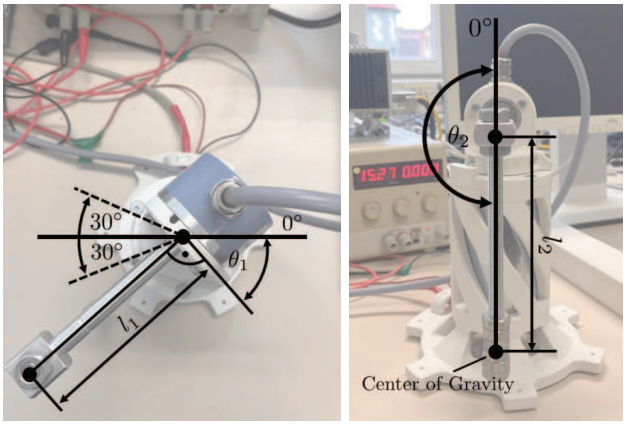
SINCE the first description by Furuta et al. [1], the control of a double rotational pendulum—meanwhile called *Furuta pendulum*—has become a benchmark application for nonlinear control approaches. The Furuta pendulum is characterized by its underactuated, fast, and nonlinear dynamics and its unstable equilibrium point in upright orientation [2]. An example of a Furuta pendulum is shown in Fig. 1. In 2010, an overview article was presented, thus surveying different *classical* nonlinear control approaches for swinging up and stabilizing the Furuta pendulum in simulation and real-world experiments [3]. Most of the considered control architectures switch between a control method for the swing-up task and another method for the stabilization task. While this switching architecture is still used in many publications [4], [5], [6], [7], [8], Acosta [3] states the design of a holistic control approach without switching as an unsolved challenge. Recent advancements in nonlinear model predictive control (NMPC) [9] and reinforcement learning (RL) [10] have since addressed this challenge.

While the initial presentation of the Furuta pendulum already considers optimal control by offline trajectory optimization [1], meanwhile, different researchers have applied various forms of optimization-based feedback control to the Furuta pendulum, such as linear model predictive control (MPC) [6], [22], explicit MPC [23], NMPC using the MATMPC library in simulation [19], and model predictive path integral (MPPI) control [8]. More recently, learning-based predictive controllers have been proposed, including Gaussian process MPC [20], learning MPC [25], and entropy-based or safe RL methods [21], [29]. Regarding the literature listed in Table I, real-world implementations of optimization-based control to address both swing-up and stabilization remain rare. This is likely due to several practical challenges that arise in the real-time deployment of NMPC and RL. For NMPC, a challenge is given by the inherent tradeoff between the long prediction horizons necessary for effective swing-up control and the short sampling times required for stabilization and appropriate disturbance mitigation. Although NMPC is, in general, computationally expensive, an embedded, real-time-capable treatment of optimal control problems (OCPs) can be achieved by recent advancements in the efficient problem formulation, structure-exploiting algorithms [30], and code generation [31]. Contributions of this brief to this research direction are given as follows:

1) review of recent control approaches to the Furuta pendulum, focusing on optimal control approaches;
2) design and application of an OCP formulation, featuring an efficient transcription method using a nonuniform time grid and carefully selected numerical integrators guided by numerical analysis;
3) open-source release of the complete embedded control implementation, including real-time execution on a pre-emptive multitasking system, and custom-designed CAD files for 3-D-printable hardware replication.

These aspects are often either missing or only partially addressed in earlier publications. By demonstrating a successful real-world implementation, this work helps to bridge the gap between theoretical NMPC developments and their technological deployment for demanding nonlinear systems' control tasks.

Structure of this brief is organized as follows. In Section II, the OCP is formulated. In Section III, the OCP is transcribed into a nonlinear program (NLP), and a numerical solution method is



(a) (b)

Fig. 1. Photograph of the Furuta pendulum used for the real-world experiments with selected states and parameters. (a) Top view. (b) Front view.

Received 3 June 2025; revised 9 September 2025; accepted 24 September 2025. This work was supported in part by the HTWG Hochschule Konstanz University of Applied Sciences and DFG under Grant 504452366 (SPP 2364) and Grant 525018088; and in part by BMWK under Grant 03EI4057A and Grant 03EN3054B. Recommended by Associate Editor A. Mesbah. *(Corresponding author: Hannes Homburger.)*

Hannes Homburger, Stefan Wirtensohn, and Johannes Reuter are with the Institute of System Dynamics, HTWG Konstanz—University of Applied Sciences, 78462 Konstanz, Germany (e-mail: hhomburg@htwg-konstanz.de; stwirten@htwg-konstanz.de; jreuter@htwg-konstanz.de).

Jonathan Frey and Moritz Diehl are with the Department of Microsystems Engineering (IMTEK) and the Department of Mathematics, University of Freiburg, 79110 Freiburg, Germany (e-mail: jonathan.frey@imtek.uni-freiburg.de; moritz.diehl@imtek.uni-freiburg.de).

Digital Object Identifier 10.1109/TCST.2025.3616367

TABLE I
CONTROL METHODS OF THE FURUTA PENDULUM

Classical Control Approaches		Simulation		Real-World	
		Swing-up	Stabilization	Swing-up	Stabilization
2005	Acosta et al. [11]	x	x	x	Feedback linearization
2008	Izutsu et al. [12]	x	x	Sliding Mode	Sliding Mode
2013	Aracil et al. [4]	x	x	Speed-Gradient	Forwarding approach
2014	Raminez-Neria et al. [13]	x	x	x	Active Disturbance Reaction
2015	Azar et al. [14]	Adaptive Sliding Mode	Adaptive Sliding Mode	x	x
2016	Aguilar-Avelar et al. [15]	x	Feedback linearization	x	Feedback linearization
2017	Lima et al. [16]	x	Nonlinear PID	x	Nonlinear PID
2017	Fatihu Hamza et al. [5]	Energy-Based	Fuzzy	Energy-Based	Fuzzy

Optimal Control Approaches		Simulation		Real-World	
		Swing-up	Stabilization	Swing-up	Stabilization
1992	Furuta et al. [1]	x	x	Trajectory Opt.	Linear State Feedback
2013	Seman et al. [6]	Energy-Based	Linear MPC	Energy-Based	Linear MPC
2017	Paredes et al. [17]	x	x	x	Variable Structure LQR
2017	Ismail et al. [18]	Trajectory Optimization	x	x	x
2019	Furka et al. [19]	NMPC	NMPC	x	x
2019	Chen et al. [20]	not stated	GP-MPC	not stated	GP-MPC
2019	Lee et al. [21]	Entropy Search	Entropy Search	Entropy Search	Entropy Search
2020	Duarte et al. [22]	x	Linear MPC	x	x
2020	Brugnolli et al. [23]	x	Explicit MPC	x	x
2020	Prado et al. [7]	Energy-Based	Tube-Based MPC	x	x
2021	Acuña-Bravo et al. [24]	x	Embedded Model Control	x	Embedded Model Control
2022	Picotti et al. [25]	Learning-Based MPC	Learning-Based-MPC	x	Learning-Based MPC
2022	Alves et al. [26]	x	H_∞ Integral Control	x	H_∞ Integral Control
2022	Homburger et al. [8]	MPPI	LQR	MPPI	LQR
2023	Rigatos et al. [27]	x	H_∞ Control	x	x
2023	Hong et al. [28]	x	RL as SAC/PPO	x	RL as SAC/PPO
2024	Baumann et al. [29]	x	Safe RL	x	Safe RL
-	This paper	NMPC	NMPC	NMPC	NMPC

presented to solve the NLP. In Section IV, the setup of numerical and real-world experiments is described, and the results are presented in Section V. Finally, Section VI concludes this brief.

II. PROBLEM FORMULATION

The feedback control law appropriate for swinging up and stabilizing the Furuta pendulum can be represented by the solution of the following OCP in standard notation [9]:

$$\arg\min_{x(\cdot),u(\cdot)} \int_0^T L(x(t),u(t))\,\mathrm{d}t + E(x(T)) \tag{1a}$$
$$\text{s.t. } x(0) = \bar{x}_0 \tag{1b}$$
$$\dot{x}(t) = f(x(t),u(t)), \quad t \in [0,T) \tag{1c}$$
$$0 \geq h(u(t)), \quad t \in [0,T) \tag{1d}$$

where $\bar{x}_0 \in \mathbb{R}^{n_x}$ denotes the initial state, $T > 0$ denotes the time horizon, $x : [0,T] \to \mathbb{R}^{n_x}$ denotes the state trajectory, $u : [0,T] \to \mathbb{R}^{n_u}$ denotes the input trajectory, $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ denotes the running cost, $E : \mathbb{R}^{n_x} \to \mathbb{R}$ denotes the terminal cost, $\dot{x} : [0,T] \to \mathbb{R}^{n_x}$ denotes the time derivative of the state trajectory, the system dynamics are denoted by $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, the input constraints are denoted by $h : \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$, and $n_x, n_u, n_h \in \mathbb{N}$ denotes the corresponding dimensions. The different parts of the OCP (1) are defined in the following.

A. Dynamics of the Furuta Pendulum

A comprehensive presentation of the Furuta pendulum's dynamics and the derivation using the Euler–Lagrange approach is given in [2]. The two degrees of freedom (2DOFs) represented by the orientation angles θ_1 and θ_2 of the Furuta pendulum are shown in Fig. 1. The angles are defined in the mathematical positive, counterclockwise direction. This means that $\theta_1 > 0$ and $\theta_2 < 0$ in Fig. 1. The orientation angles and the associated angular velocities ω_1 and ω_2 are combined to the state vector $x = (\theta_1,\ \theta_2,\ \omega_1,\ \omega_2)^\top$ with $n_x = 4$. The motor torque in the direction of θ_1 represents the control input of the pendulum and is denoted by u with dimension $n_u = 1$. The dynamics of the Furuta pendulum are given by

$$\dot{x} = f(x,u) = (\omega_1,\quad \omega_2,\quad \dot{\omega}_1,\quad \dot{\omega}_2)^\top \tag{2}$$

where the angular accelerations are given by

$$\dot{\omega}_1 = \frac{\left[\begin{pmatrix} -\hat{J}_2 b_1 \\ -m_2 l_1 l_2 \cos(\theta_2)\, b_2 \\ -\hat{J}_2{}^2 \sin(2\theta_2) \\ \frac{1}{2}\hat{J}_2 m_2 l_1 l_2 \cos(\theta_2)\sin(2\theta_2) \\ -\hat{J}_2 m_2 l_1 l_2 \sin(\theta_2) \end{pmatrix}^\top \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_1\omega_2 \\ \omega_1^2 \\ \omega_2^2 \end{pmatrix} + \begin{pmatrix} \hat{J}_2 \\ \frac{1}{2} m_2^2 l_2^2 l_1 \sin(2\theta_2) \end{pmatrix}^\top \begin{pmatrix} u \\ g \end{pmatrix}\right]}{\hat{J}_1\hat{J}_2 + \hat{J}_2{}^2 \sin^2(\theta_2) - m_2^2 l_1^2 l_2^2 \cos^2(\theta_2)}$$

and

$$\dot{\omega}_2 = \frac{\left[\begin{pmatrix} -m_2 l_1 l_2 \cos(\theta_2)\, b_1 \\ -b_2\left(\hat{J}_1 + \hat{J}_2 \sin^2(\theta_2)\right) \\ -m_2 l_1 l_2 \hat{J}_2 \cos(\theta_2)\sin(2\theta_2) \\ -\frac{1}{2}\sin(2\theta_2)\left(\hat{J}_1\hat{J}_2 + \hat{J}_2{}^2 \sin^2(\theta_2)\right) \\ -\frac{1}{2} m_2^2 l_1^2 l_2^2 \sin(2\theta_2) \end{pmatrix}^\top \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_1\omega_2 \\ \omega_1^2 \\ \omega_2^2 \end{pmatrix} + \begin{pmatrix} m_2 l_1 l_2 \cos(\theta_2) \\ m_2 l_2 \sin(\theta_2)\left(\hat{J}_1 + \hat{J}_2 \sin^2(\theta_2)\right) \end{pmatrix}^\top \begin{pmatrix} u \\ g \end{pmatrix}\right]}{\hat{J}_1\hat{J}_2 + \hat{J}_2{}^2 \sin^2(\theta_2) - m_2^2 l_1^2 l_2^2 \cos^2(\theta_2)}$$

where m_2 denotes the mass of the front arm, b_1 denotes the coefficient of viscous friction in the motor bearing, b_2 denotes the coefficient of viscous friction in the pin coupling, $\hat{J}_1$ denotes the effective moment of inertia at the motor, and $\hat{J}_2$ denotes the effective moment of inertia at pin coupling [2]. The lengths l_1 and l_2 are shown in Fig. 1 and g denotes the gravitational acceleration. As described later in

Section IV, the desired motor torque u is applied to the system by a dc motor, which is part of an underlying control loop. It is reasonable to assume that the dynamics of this control loop are fast compared to the pendulum dynamics (2) and can be neglected [2]. However, due to the motor specification, the torque is bounded in the range $\underline{u} \leq u \leq \overline{u}$.

B. Problem Specification

The *running cost* is formulated in a least-squares form to ensure that it attains its minimum at the trivial equilibrium point $x_e = (0, 0, 0, 0)^\top$ and $u_e = 0$, and given by

$$L(x, u) = r(x, u)^\top Q\, r(x, u)$$

where $Q \succ 0 \in \mathbb{R}^{5\times5}$ is a weighting matrix and the residuals are

$$r(x, u) = \begin{pmatrix} x - x_e \\ u - u_e \end{pmatrix}.$$

It is important to note that this cost function represents one of the simplest reasonable choices for the swing-up and stabilization tasks. More complex designs can, e.g., account for the periodicity of θ_2 but add formulation effort and can cause numerical issues [8]. Unlike RL methods, which often require extensive *reward shaping*, this brief's approach relies on a straightforward and interpretable cost structure that is exploited in the numerical solution method presented in Section III. The *terminal cost* term is chosen to approximate the costs of the linearized infinite horizon problem $E(x_T) :=$

$$\min_{x(\cdot),u(\cdot)} \int_T^\infty r(x,u)^\top Q r(x,u)\, \mathrm{d}t \tag{3a}$$

$$\text{s.t. } x(T) = x_T \tag{3b}$$

$$\dot{x}(t) = Ax(t) + Bu(t), \quad t \in [T, \infty) \tag{3c}$$

where the linearized system dynamics are represented by $A := \partial f/\partial x(x_e, u_e)$ and $B := \partial f/\partial u(x_e, u_e)$. The explicit solution of (3a) is given by the quadratic function $E(x_T) = x_T^\top P x_T$, where P is the solution of the corresponding matrix Riccati equation [9]. *Input constraints* are required because the torque is bounded in the range $\underline{u} \leq u \leq \overline{u}$ due to the specifications of the motor employed. Therefore, we define the inequality constraints on the input in the form of

$$h(u) = \begin{pmatrix} u - \overline{u} \\ \underline{u} - u \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4}$$

Note that in practical scenarios, additional state constraints may need to be considered, for example, limits on the pendulum's orientation θ_1 (see Fig. 1). In the setup considered in this brief, state constraints are not critical since the system stays far away from the boundaries during standard operation. Nevertheless, if, in other settings, a rigorous treatment of state constraints is necessary, slack variables, penalty methods, and barrier functions are well-established strategies in the literature [9], [32].

C. Discussion on Feasibility and Stability

The feasibility of the presented OCP and the stability of the resulting NMPC scheme are important theoretical aspects briefly discussed in this section. Since the OCP (1) is formulated with input constraints only, feasibility is trivially ensured. Regarding stability, if the prediction horizon is chosen sufficiently long and the input constraints allow the system state at the end of the horizon, x_N, to lie sufficiently close to the equilibrium, then the terminal cost introduced in Section II-B provides a suitable approximation of the infinite-horizon cost [9]. Under these conditions, the resulting closed-loop system can exhibit exponential stability. The design choices of the presented controller, in particular, the terminal cost and the horizon length, are guided by the requirements established in stability theory [33].

III. Efficient OCP Solution

The methods to solve the OCP (1) efficiently and the corresponding NMPC scheme are presented in this section.

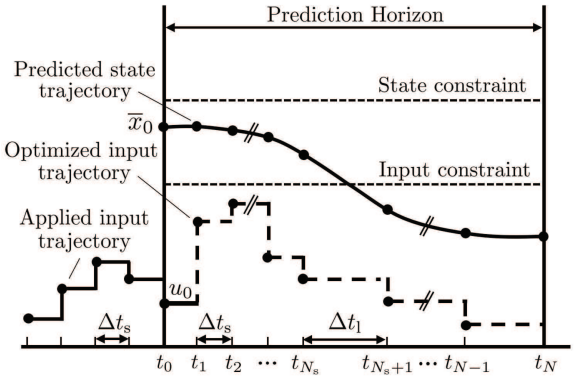


Fig. 2. Schematic drawing of the solution of an exemplary transcripted OCP using a nonuniform time grid and the applied input trajectory.

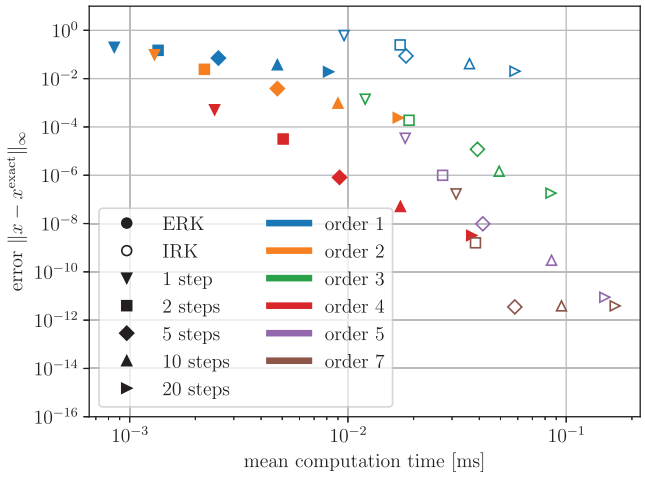


Fig. 3. Numerical comparison of Runge–Kutta integration methods of different orders applied on the Furuta pendulum's dynamics over a time interval of $\Delta t = 60$ ms with a different number of intermediate steps. The resulting error is plotted over the mean computation time.

A. Transcription

By employing direct multiple shooting [34], the original infinite-dimensional OCP (1) is transcribed into a finite-dimensional NLP by discretization. For the discretization, we choose piecewise constant controls in the form of

$$u(t) = u_k, \quad t \in \left[t_k, t_{k+1}\right)$$

on each shooting interval $[t_k, t_{k+1})$ with $k = 0, \ldots, N-1$, where $u_k \in \mathbb{R}^{n_u}$ with $t_0 = 0$ and $t_N = T$. It is widespread to use a uniform grid $t_k = k\Delta t$ with $\Delta t = T/N$ to discretize the problem. However, nonuniform discretization grids have been shown to lead to efficient NMPC schemes [35]. To reach both, a first shooting interval $[0, t_1)$ that corresponds to the minimum sampling time of the actuator and a sufficiently long horizon to ensure the optimizer is aware of future costs and constraints, we choose a nonuniform grid with $t_k = k\Delta t_s$ for $k = 0, \ldots, N_s$ and $t_k = N_s\Delta t_s + k\Delta t_l$ for $k = 1, \ldots, N_l$ with $N = N_s + N_l$ nodes. The NLP resulting from the discretization of the OCP (1) can be stated as

$$\min_{\substack{x_0, u_0, x_1, \ldots, \\ u_{N-1}, x_N}} \sum_{n=0}^{N-1} L_n(x_n, u_n) + E(x_N) \tag{5a}$$

$$\text{s.t. } x_0 = \overline{x}_0 \tag{5b}$$

$$x_{n+1} = F_n(x_n, u_n), \quad n = 0, \ldots, N-1 \tag{5c}$$

$$0 \geq h(u_n), \quad n = 0, \ldots, N-1 \tag{5d}$$

TABLE II
EXPERIMENTAL SETUPS AND RESULTS SWING-UP THE PENDULUM

		Experimental Setup							Stabilization			Swing-up		
		N_s	Δt_s	$N_\mathrm{int,s}$	N_l	Δt_l	$N_\mathrm{int,l}$	Mode	$\bar{t}_\mathrm{c}$	$\bar{J}$	s	$\bar{t}_\mathrm{c}$	$\bar{J}$	s
Nonuniform	Setup 1a	1	1 ms	1	9	38.8 ms	4	Simulation	6.1 %	27.4	✓	7.4 %	353.0	✓
								Real-World	64.9 %	17.3	✓	80.0 %	361.1	✓
	Setup 1b	1	5 ms	1	9	38.3 ms	4	Simulation	1.2 %	28.2	✓	1.5 %	348.0	✓
								Real-World	14.1 %	17.2	✓	17.7 %	367.3	✓
	Setup 1c	1	10 ms	1	9	37.8 ms	4	Simulation	0.6 %	28.5	✓	0.7 %	338.9	✓
								Real-World	7.1 %	29.2	o	8.4 %	372.5	o
Uniform	Setup 2a	70	5 ms	1	-	-	-	Simulation	10.5 %	27.1	✓	12.3 %	346.9	✓
								Real-World	>100 %	-	×	>100 %	-	×
	Setup 2b	35	10 ms	1	-	-	-	Simulation	2.5 %	26.5	✓	2.6 %	339.4	✓
								Real-World	25.9 %	16.5	o	35.4 %	361.0	o
	Setup 2c	10	35 ms	4	-	-	-	Simulation	0.2 %	29.4	✓	0.5 %	-	×
								Real-World	2.1 %	56.6	o	5.0 %	-	×

where $x_n \in \mathbb{R}^{n_x}$ for $n = 0, \ldots, N$ denote the discrete state at t_n. All two adjacent shooting nodes are linked by the functions $F_n : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ that represent the discretized continuous-time dynamics (1c) over the nth shooting interval. Often, it is beneficial to adjust the integration method and the number of immediate integrator steps N_int depending on the length of the corresponding shooting interval within the nonuniform grid. This allows the desired accuracy to be achieved. The inequality constraints $h(\cdot)$ enforce the constraints (1d) at the shooting nodes. Fig. 2 shows a sketch of the nonuniform grid approach.

B. Numerical Solution and NMPC Design

The transcribed OCP (5) is a high-dimensional, nonconvex, and constrained NLP given in a structured form. The OCP's solution can be used to design a state-feedback controller in NMPC fashion. Therefore, the NMPC control law is implicitly given by $u_0^* : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ as the first element of a numerical approximation of the optimal input trajectory of the NLP (5) dependent on the external parameter $\bar{x}_0$ [9]. Therefore, a high-dimensional, nonconvex, and constrained NLP has to be approximately solved in real time. Efficient sequential quadratic programming (SQP)-type algorithms tailored to exploit the structure of the NLPs can treat them numerically [36]. To achieve fast sample times, the real-time iteration (RTI) scheme [37] is applied. This means that per sampling time, one SQP-type iteration with Gauss–Newton Hessian is applied to the NLP with condensing and full derivatives. A theoretical investigation of the contractivity of this scheme can be found in [37, Sec. 5]. The considered OCP is parameterized with $Q = \mathrm{diag}(50, 500, 1, 1, 1000)$ and $\bar{u} = -\underline{u} = 0.45$ N m and transcribed using the grid parameters listed in Table II. The model parameters are taken from [8]. Regarding the numerical experiments shown in Fig. 3, we found that for the Furuta pendulum, implicit integration methods are not competitive with explicit integration methods. The explicit Runge–Kutta integrator of fourth-order (RK4) achieves the smallest computational time for a given error in the region of interest that is defined by an error between 10^{-9} and 10^{-3}. Implicit integrators are only advantageous if a higher level of accuracy is required. In the example under consideration, however, such high accuracy is not required, and an explicit RK4 is selected for each shooting interval. Considering the *computational effort* related to the nonuniform grid formulation, it is important to note that the effort is typically reduced by using a nonuniform grid, as it decreases the number of shooting nodes and thus reduces the dimension of the QP. Model accuracy can be maintained by employing appropriate integration methods for longer time steps of the nonuniform grid, as shown in Fig. 3.

IV. EXPERIMENTAL SETUP

In this section, we present the software and hardware setups as a basis for the following experiments with different time grid parameters. An overview of different modules used in the experimental setup is provided in Fig. 4. The corresponding embedded implementation using the preemptive multitasking architecture and the hardware design files are provided in a corresponding repository [38].

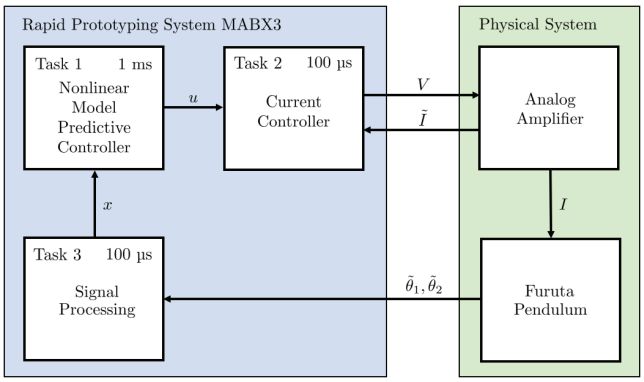


Fig. 4. Signal flow diagram of the closed-loop system used for the real-world experiments.

A. Software Setup

The NMPC algorithm is implemented using the `CasADi` [39] interface of `acados` [31] in `MATLAB`. The NLPs are solved using the RTI scheme [37], solving the quadratic subproblems with `HPIPM` [40], full-condensing, and a tolerance of 10^{-3}. For completeness, we note that the implementation includes a hard constraint on θ_1. However, this constraint remains inactive throughout all predicted trajectories and, as such, does not influence the control behavior. Therefore, it is not considered in the rest of this brief, including the problem statement in (5). The QPs are regularized with a Levenberg–Marquardt term of 20. The presented RTI NMPC scheme is employed to calculate the desired torque. With the motor constant K_M, the torque is assumed to be proportional to the motor current, which is the reference value for the current control loop. The current controller is designed in the form of a linear controller with a sample time of 100 μs. This controller reaches a closed-loop behavior given by a first-order transfer function with a cutoff frequency of $f_\mathrm{c} \approx 915$ Hz. Measurements of the angles θ_1 and θ_2 are provided directly by encoders and are denoted by $\tilde{\theta}_1$ and $\tilde{\theta}_2$. While advanced model-based estimation techniques such as moving horizon estimation or nonlinear Kalman filtering are well-known, the availability of low-noise measurements in the investigated system justifies the use of simple first-order lag differentiators to estimate angular velocities. The different software tasks are implemented in a *pre-emptive multitasking* architecture [38]. This architecture allows the high-frequency current control task to interrupt the NMPC task.

B. Setup of Discretization Grid

The performance of the NMPC controller is significantly dependent on the selection of the time grid for prediction. As a lower bound for the sample time, we consider the cutoff frequency of $f_c \approx 915$ Hz of the subordinated current control loop and determine the shortest reasonable sample time to $t_s = 1$ ms. To look far enough into the future, with the controller, a reasonable prediction horizon of 350 ms is selected because this is the minimum swing-up time determined in simulation experiments. The number of steps in the shooting interval varies for different interval lengths. Considering Fig. 3, an upper bound for the step length of 12 ms is chosen to reach an error less than 10^{-6} with the RK4 integrator. We compare the following discretization setups.

Setup 1: The described nonuniform grid approach with one short interval at the beginning, followed by nine long intervals. Different lengths for the first interval are investigated, while the lengths of the long intervals are determined to reach a prediction horizon of 350 ms.

Setup 2: A standard uniform grid approach. For each case, the number of shooting intervals is selected to reach a prediction horizon of 350 ms.

C. Hardware Setup

The software is executed on the rapid prototyping system `MicroAutoBox III` (MABX3) by `dSPACE`. The MABX3 enables the usage of the described preemptive multitasking scheme necessary for executing the current controller and the NMPC on the same processor. Further, it provides digital-to-analog converters (DACs), analog-to-digital converters (ADCs), and encoder interfaces. The considered Furuta pendulum is custom-made, and some parts are printed using a 3-D printer. The CAD file is provided in the repository [38]. The motor current is provided by a custom-made analog amplifier.

V. Results

The entire system is investigated in the following by performing various experiments in simulation and real-world experiments. To compare the results of different setups, we introduce the metrics.

The *relative CPU loading* is given by $\bar{t}_c = t_c/\Delta t_s$, where t_c denotes the maximum computation time of the solver on a desktop computer with a `12th Gen Intel`[1] `Core`[2] `i5--12400 2.50--GHz` processor for the simulation experiments and on an MABX3 for the real-world experiments. Note that the MABX3 is a high-performance embedded device. To be applicable even with more widespread embedded devices, $\bar{t}_c$ should be lower than 20%.

The *control performance* in the form of average cost of the closed-loop system is given by $\bar{J} = 1/N_{\text{ex}} \sum_{k=0}^{N_{\text{ex}}-1} L(x_k, u_k)\Delta t_s$, where N_{ex} denotes the number of steps in the experiment. This metric represents the weighted mean squared control error.

The *ability to successfully perform the task* is given by the variable $s \in \{\checkmark, o, \times\}$, where $\checkmark$ means successfully performing the given task, o means success with severe oscillation, and $\times$ means that the controller is not capable of swing-up or stabilize the pendulum.

A. Stabilization Task

Considering Table I, controlling the Furuta pendulum in its upright equilibrium position is a widespread task. For this task, a controller with a high sample frequency is required because the pendulum's upright equilibrium point is inherently unstable. The *stabilization* experiment is initialized with $\bar{x}_0^{\text{st}} = (0 \text{ rad}, \pi/8 \text{ rad}, 0 \text{ rad/s}, 0 \text{ rad/s})^\top$ next to the upright position. The results based on different settings are listed in Table II. All of the considered settings with nonuniform grids can stabilize the Furuta pendulum in simulation. Even in real-world experiments, Setups 1a and 1b result in excellent behavior, while Setup 1c causes more costs due to severe oscillations caused by model-plant mismatch. However, considering the real-world experiments, Setups 1a and 1b achieve lower costs than in the simulation. Since no significant differences can be seen between Setups 1a and 1b, it is concluded that a sampling time of 5 ms is a sufficient choice.

[1]Registered trademark.

[2]Trademarked.

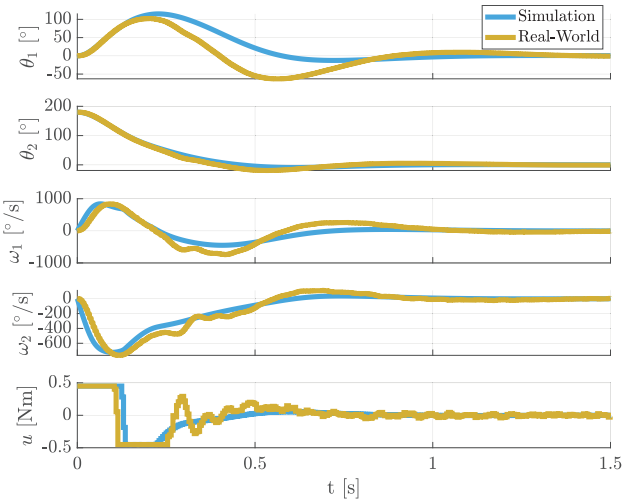


Fig. 5. Measured state trajectories and input trajectories of the closed-loop system while swinging up and afterward stabilizing the Furuta pendulum in simulation and real-world employing the *nonuniform* grid of Setup 1b.

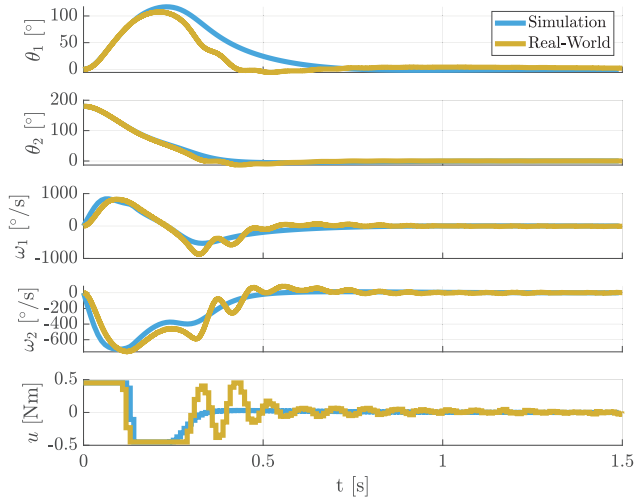


Fig. 6. Measured state trajectories and input trajectories of the closed-loop system while swinging up and afterward stabilizing the Furuta pendulum in simulation and real-world employing the *uniform* grid of Setup 2b.

B. Swing-Up Task

Table I shows that experiments performing swing-up tasks of the Furuta pendulum are rare in the literature, especially in contributions investigating optimal control approaches. However, because a nonlinear model of the Furuta pendulum is used in the OCP, the designed NMPC scheme is not restricted to the stabilization task. Therefore, different settings are investigated in a swing-up task. The state vector is initialized with $\bar{x}_0^{\text{sw}} = (0 \text{ rad}, \pi \text{ rad}, 0 \text{ rad/s}, 0 \text{ rad/s})^\top$. For the swing-up task, the RTI scheme is warm-started with the precomputed solution calculated with 100 SQP iterations with line search globalization implemented in `acados` [31]. The results of the experiments are listed in Table II in the right columns. The trajectories of the swinging up task employing Setups 1b and 2b are shown in Figs. 5 and 6. The simulation and real-world trajectories show roughly similar shapes, especially at the beginning of the experiment. The θ_2 trajectories nearly coincide in both experiments. However, θ_1 and the corresponding angular velocity ω_1 show a different behavior, especially employing Setup 1b in Fig. 5. The control trajectories are similar, but after leaving the maximum torque at ≈285 ms, oscillations appear in the real-world experiments. The investigated

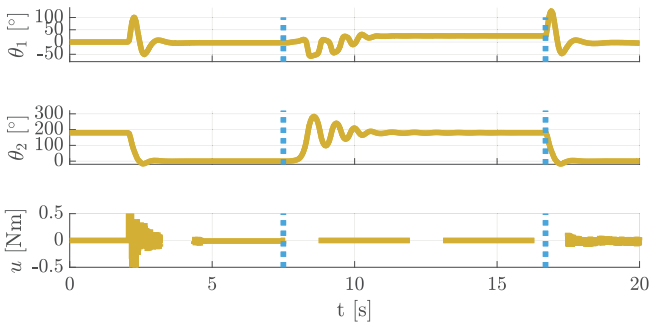


Fig. 7. Measured trajectories of the pendulum's orientations and the input trajectory of the closed-loop system while disturbance *Scenario 1*. Between the blue dotted lines, the pendulum follows its uncontrolled motion.

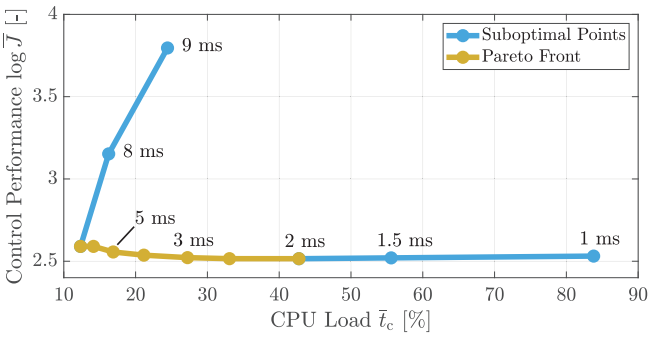


Fig. 8. Closed-loop performance of the Furuta pendulum in disturbance *Scenario 2* over the relative CPU load.

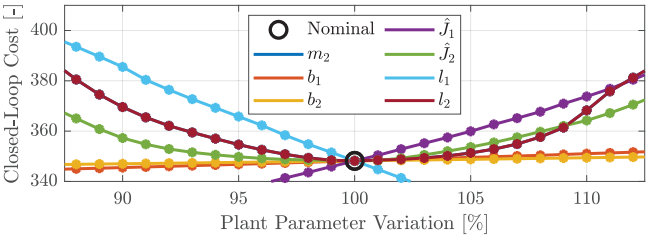


Fig. 9. Numerical sensitivity analysis of the nominal NMPC controlled closed-loop performance over variation of the plant parameters compared to the nominal performance of *Setup 1b*.

approach controls the Furuta pendulum from the lower to upper equilibrium points in ≈ 392 ms and then successfully stabilizes the pendulum.

Discussion: The results stand in line with those from the stabilization experiments. In Setup 2b, stronger oscillations are injected, and about twice the computation time is required than in Setup 1b. Hence, the model-plant mismatch's influence on θ_2 is nearly completely compensated by the controller. As summarized, different real-time capable settings are presented to swing-up and stabilize the Furuta pendulum in simulation and real-world experiments with NMPC. Setup 1b is considered as the best of the investigated settings, exploiting the strengths of the presented nonuniform grid to reach a fast sampling time, a small computational effort, and an excellent performance.

C. Actively Injected Disturbances

The presented real-world experiments show the controller's capability to counteract inevitable disturbances like the torsion of the encoder cable and other unmodeled effects. In the following, further experiments are performed to investigate the behavior of the controlled system in the presence of stronger and reproducible disturbances. In the following, we define two different disturbance scenarios and evaluate the closed-loop system performance with controller Setup 1b. Therefore, we apply the artificially adulterated time-variant control law

$$u_{\text{dis}}(x,t) = \begin{cases} 0, & \text{if } t \in \mathcal{T}_k \\ u_0^*(x), & \text{else} \end{cases}$$

where the set $\mathcal{T}_k$ defines the disturbance times, in that the optimal system input is computed but not applied. In the following, we define two different scenarios by choosing different $\mathcal{T}_k$ with $k = \{1, 2\}$.

1) Falling and Rising Again: This scenario is designed to test the RTI's capability for *self-initialization*. By setting $\mathcal{T}_2 = (\underline{t}_2, \bar{t}_2)$ with $\underline{t}_2 = 7.5$ s and $\bar{t}_2 = 16.7$ s after a successfully finished swing-up, the controller stabilizes the pendulum in its upright position. Then, the torque to stabilize the pendulum is deactivated until the pendulum levels out in a downward position. Then, the disturbance is deactivated.

The measured trajectories of the orientations θ_1 and θ_2 and the corresponding input trajectories are shown in Fig. 7. After deactivating the stabilizing controls, the pendulum falls in damped oscillations and reaches a downright position asymptotically with $\theta_1 = 26°$ and $\theta_2 = 180°$. Upon reapplying the controller, a second successful swing-up is achieved. This experiment demonstrates the effectiveness of the implemented RTI scheme, and the key observation is that the second swing-up is possible without any externally provided warm starting trajectory.

2) Multiple Disturbances: While the disturbance Scenario 1 is designed to evaluate the controller's capability for self initialization, this scenario is designed to find the Pareto front of *CPU Load* $\bar{t}_c$ versus *Control Performance* $\overline{J}$ resulting from different controller setups at the real system under multiple disturbances given by $\mathcal{T}_3 = \{t | \underline{t}_1 < t < \bar{t}_1 \vee \underline{t}_3 < t < \bar{t}_3\}$ with $\underline{t}_3 = 600$ ms and $\bar{t}_3 = 800$ ms. For each setup, $\overline{J}$ is calculated over a time interval of 1.5 s. The multiple disturbances experiments in Scenario 2 were performed with the nonuniform grid setup with different sample times of the controller $\Delta t_s = \{1, 1.5, 2, 2.5, 3, 4, 5, 6, 7, 8, 9\}$ ms. The resulting controller performance is plotted over the CPU load in Fig. 8. With a sampling time of less than 3 ms, the control performance is almost identical, and then, up to 7 ms, the control performance slightly deteriorates. For longer sample times, the controller performance collapses. For sample times higher than 9 ms, the controller did not achieve a swing-up in Scenario 2. The CPU load decreases for longer sample times and increases for sample times longer than $\Delta t_s = 7$ ms. For this sample time, a minimum CPU load is achieved with ≈ 12%. In summary, the actively injected disturbance experiments show the capability of the designed NMPC scheme to react to severe and unforeseen disturbances appropriately. The detailed analysis of the real-world controller performance over the CPU load confirms the conclusion from Section V-B that Setup 1b is a suitable setup for controlling the Furuta pendulum—even under severe disturbances.

D. Model-Plant Mismatch and Sensitivity Analysis

NMPC control schemes can be sensitive to model inaccuracies [9]. In this section, we investigated the sensitivity of the presented NMPC controller through a numerical sensitivity study. To this end, the key physical parameters of the plant, which are presented in Section II, are systematically varied around their nominal values. The resulting variation in closed-loop cost is visualized in Fig. 9, confirming that the nominal NMPC scheme maintains acceptable performance in a reasonable range of parameters. Importantly, the nonuniform grid approach employed in the controller design leads to a fast sampling time, which inherently mitigates some of the effects of moderate model-plant mismatch. The fast reoptimization helps maintain performance in the presence of uncertain model parameters.

VI. Conclusion and Future Work

This brief presents an NMPC approach for swing-up and stabilization control of a Furuta pendulum with a 1-ms sample time in the real world. The key features are the implemented nonuniform discretization within a preemptive multitasking architecture on an embedded device and carefully selected numerical integrators guided by numerical analysis provided in this brief. The examined method unites the contradictory objectives of a short sampling period and a long prediction horizon while ensuring real-time capability. The numerical and real-world experiments illustrate the controller's excellent performance for the swing-up and stabilization task even under disturbances and model-plant mismatch. The controller implementation and the corresponding CAD files are provided in an open-access repository accompanying this brief [38]. The results show a real-time capable, high-performance NMPC approach for a highly dynamic, nonlinear, and constrained system on embedded hardware executed in just 1 ms. The open-source framework provided can drastically reduce the effort required for further investigations in the future.

References

[1] K. Furuta, M. Yamakita, and S. Kobayashi, "Swing-up control of inverted pendulum using pseudo-state feedback," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 206, no. 4, pp. 263–269, Nov. 1992.

[2] B. S. Cazzolato and Z. Prime, "On the dynamics of the Furuta pendulum," *J. Control Sci. Eng.*, vol. 2011, no. 1, pp. 1–8, Jan. 2011.

[3] J. Á. Acosta, "Furuta's pendulum: A conservative nonlinear model for theory and practise," *Math. Problems Eng.*, vol. 2010, no. 1, pp. 1–29, Jan. 2010.

[4] J. Aracil, J. Á. Acosta, and F. Gordillo, "A nonlinear hybrid controller for swinging-up and stabilizing the Furuta pendulum," *Control Eng. Pract.*, vol. 21, no. 8, pp. 989–993, Aug. 2013.

[5] M. F. Hamza, H. J. Yap, and I. A. Choudhury, "Cuckoo search algorithm based design of interval type-2 fuzzy PID controller for Furuta pendulum system," *Eng. Appl. Artif. Intell.*, vol. 62, pp. 134–151, Jun. 2017.

[6] P. Seman, B. Rohal'-Ilkiv, M. Juhas, and M. Salaj, "Swinging up the Furuta pendulum and its stabilization via model predictive control," *J. Electr. Eng.*, vol. 64, no. 3, pp. 152–158, May 2013.

[7] A. Prado, M. Herrera, and O. Menéndez, "Intelligent swing-up and robust stabilization via tube-based nonlinear model predictive control for a rotational inverted-pendulum system," *Revista Politécnica*, vol. 45, no. 1, pp. 49–64, Apr. 2020.

[8] H. Homburger, S. Wirtensohn, and J. Reuter, "Swinging up and stabilization control of the Furuta Pendulum using model predictive path integral control," in *Proc. 30th Medit. Conf. Control Autom. (MED)*, Jun. 2022, pp. 7–12.

[9] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed., San Francisco, CA, USA: Nob Hill, 2020.

[10] R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., Cambridge, MA, USA: MIT Press, 2018.

[11] J. A. Acosta and M. Lopez-Martinez, "Constructive feedback linearization of underactuated mechanical systems with 2-DOF," in *Proc. 44th IEEE Conf. Decis. Control*, Oct. 2005, pp. 4909–4914.

[12] M. Izutsu, Y. Pan, and K. Furuta, "Swing-up of Furuta pendulum by nonlinear sliding mode control," *SICE J. Control, Meas., Syst. Integr.*, vol. 1, no. 1, pp. 12–17, Jan. 2008.

[13] M. Ramírez-Neria, H. Sira-Ramírez, R. Garrido-Moctezuma, and A. Luviano-Juárez, "Linear active disturbance rejection control of underactuated systems: The case of the Furuta pendulum," *ISA Trans.*, vol. 53, no. 4, pp. 920–928, Jul. 2014.

[14] A. T. Azar and F. E. Serrano, "Adaptive sliding mode control of the Furuta pendulum," in *Advances and Applications in Sliding Mode Control Systems*, vol. 576, A. T. Azar and Q. Zhu, Eds., Cham, Switzerland: Springer, 2015, pp. 1–42.

[15] C. Aguilar-Avelar and J. Moreno-Valenzuela, "New feedback linearization-based control for arm trajectory tracking of the Furuta pendulum," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 2, pp. 638–648, Apr. 2016.

[16] B. Lima, R. Cajo, V. Huilcapi, and W. Agila, "Modeling and comparative study of linear and nonlinear controllers for rotary inverted pendulum," *J. Phys., Conf. Ser.*, vol. 783, Jan. 2017, Art. no. 012047.

[17] I. Paredes, M. Sarzosa, M. Herrera, P. Leica, and O. Camacho, "Optimal-robust controller for Furuta pendulum based on linear model," in *Proc. IEEE 2nd Ecuador Tech. Chapters Meeting (ETCM)*, Oct. 2017, pp. 1–6.

[18] J. Ismail and S. Liu, "Efficient planning of optimal trajectory for a Furuta double pendulum using discrete mechanics and optimal control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10456–10461,Jul. 2017.

[19] M. Furka, M. Klaučo, and M. Kvasnica, "Stabilization of Furuta pendulum using nonlinear MPC," *Res. Papers Fac. Mater. Sci. Technol. Slovak Univ. Technol.*, vol. 27, no. 45, pp. 42–48, Sep. 2019.

[20] K. Chen, J. Yi, and D. Song, "Gaussian processes model-based control of underactuated balance robots," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4458–4464.

[21] C. S. Lee and D. E. Chang, "Enhancement of energy-based swing-up controller via entropy search," in *Proc. 12th Asian Control Conf. (ASCC)*, Jun. 2019, pp. 393–398.

[22] J. L. D. Madrid, E. A. G. Querubín, and P. A. O. Henao, "MPC in space state for the control of a Furuta pendulum," in *Proc. RITA*, A. P. P. A. Majeed, J. A. Mat-Jizat, M. H. A. Hassan, Z. Taha, H. L. Choi, and J. Kim, Eds., Singapore: Springer, 2020, pp. 219–235.

[23] M. M. Brugnolli and B. A. Angélico, "Explicit model predictive control for inverted pendulum systems," in *Proc. Anais do Congresso Brasileiro de Automática*, Dec. 2020, pp. 1–6.

[24] W. Acuña-Bravo, A. G. Molano-Jiménez, and E. Canuto, "Embedded model control for underactuated systems: An application to Furuta pendulum," *Control Eng. Pract.*, vol. 113, pp. 1–12, Aug. 2021.

[25] E. Picotti, A. D. Libera, R. Carli, and M. Bruschetta, "LbMATMPC: An open-source toolbox for Gaussian process modeling within learning-based nonlinear model predictive control," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2022, pp. 736–742.

[26] U. Alves et al., "Discrete-time H_∞ integral control via LMIs applied to a Furuta pendulum," *J. Control, Autom. Electr. Syst.*, vol. 33, no. 3, pp. 1–12, 2022.

[27] G. Rigatos, M. Abbaszadeh, P. Siano, G. Cuccurullo, J. Pomares, and B. Sari, "Nonlinear optimal control for the rotary double inverted pendulum," *Adv. Control for Appl.*, vol. 6, no. 2, pp. 1–31, Jun. 2024.

[28] M. R. Hong et al., "Optimizing reinforcement learning control model in Furuta pendulum and transferring it to real-world," *IEEE Access*, vol. 11, pp. 95195–95200, 2023.

[29] D. Baumann and T. B. Schön, "Safe reinforcement learning in uncertain contexts," *IEEE Trans. Robot.*, vol. 40, pp. 1828–1841, 2024.

[30] R. Quirynen, M. Vukov, and M. Diehl, "Multiple shooting in a microsecond, "Multiple shooting in a microsecond," in *Multiple Shooting and Time Domain Decomposition Methods*, vol. 9, T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, Eds., Cham, Switzerland: Springer, 2015, pp. 183–201.

[31] R. Verschueren et al., "Acados—A modular open-source framework for fast embedded optimal control," *Math. Program. Comput.*, vol. 14, no. 1, pp. 147–183, Mar. 2022.

[32] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed., Philadelphia, PA, USA: SIAM,2010.

[33] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*, 2nd ed., Cham, Switzerland: Springer, 2017.

[34] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proc. Volumes*, vol. 17, no. 2, pp. 1603–1608, Jul. 1984.

[35] J. Frey, K. Baumgärtner, and M. Diehl, "Gauss–Newton Runge–Kutta integration for efficient discretization of optimal control problems with long horizons and least-squares costs," *Eur. J. Control*, vol. 2024, pp. 512–518, Apr. 2024.

[36] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., New York, NY, USA: Springer, 2006.

[37] M. Diehl, "Real-time optimization for large scale nonlinear processes," Ph.D. dissertation, Heidelberg Univ. Library, Naturwissenschaftlich-Mathematische Gesamtfakultät Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany, 2001. [Online]. Available: https://archiv.ub.uni-heidelberg.de/volltextserver/1659/1/Diehl_2001_Dissertation.pdf

[38] H. Homburger, "Hhomb/Furutapendulum: First release NMPC Furuta acados," Inst. Syst. Dyn., HTWG Konstanz—Univ, Konstanz, Germany, Tech. Rep. 13329496, 2024, doi: 10.5281/zenodo.13329496.

[39] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, Mar. 2019.

[40] G. Frison and M. Diehl, "HPIPM: A high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.