An Adaptive Data-Enabled Policy Optimization Approach for Autonomous Bicycle Control

Niklas Persson, *Student member, IEEE*, Feiran Zhao, Mojtaba Kaheni, *Senior Member, IEEE*, Florian Dörfler, *Senior Member, IEEE*, Alessandro V. Papadopoulos, *Senior Member, IEEE*

***Abstract*—This paper presents a unified control framework that integrates a Feedback Linearization (FL) controller in the inner loop with an adaptive Data-Enabled Policy Optimization (DeePO) controller in the outer loop to balance an autonomous bicycle. While the FL controller stabilizes and partially linearizes the inherently unstable and nonlinear system, its performance is compromised by unmodeled dynamics and time-varying characteristics. To overcome these limitations, the DeePO controller is introduced to enhance adaptability and robustness. The initial control policy of DeePO is obtained from a finite set of offline, persistently exciting input and state data. To improve stability and compensate for system nonlinearities and disturbances, a robustness-promoting regularizer refines the initial policy, while the adaptive section of the DeePO framework is enhanced with a forgetting factor to improve adaptation to time-varying dynamics. The proposed FL-DeePO approach is evaluated through simulations and real-world experiments on an instrumented autonomous bicycle. Results demonstrate its superiority over the FL-only approach and a Reinforcement Learning (RL) controller, achieving more precise tracking of the reference lean angle and lean rate.**

***Index Terms*—Adaptive control, policy optimization, direct data-driven control, balance control, autonomous bicycle.**

I. Introduction

An autonomous bicycle is a bicycle equipped with motors, sensors, and algorithms for riderless operation, with applications ranging from autonomous reallocation in bike-sharing systems [1] to acting as realistic targets for vehicle safety testing [2] and steering assistance for riders with limited physical capabilities [3]. While balancing can be achieved by actuating the lean angle [4], [5], this work utilizes steering control [2], [6], [7], which is more energy-efficient and preserves the bicycle's natural appearance.

Balancing control typically relies on explicit models. Linear controllers such as Proportional-Integral-Derivative (PID) controller and Linear Quadratic Regulartor (LQR) perform well near equilibrium [2], while nonlinear approaches like Sliding Mode Control (SMC) [7] and Linear Parameter-Varying (LPV) control [6] address robustness. However, performance often degrades due to parameter uncertainty, unmodeled dynamics, and environmental changes [8]. On the other hand, model-free Reinforcement Learning (RL) methods [9]–[11] handle nonlinearities but suffer from long training times and lack extensive experimental validation.

Recently, direct data-driven control methods inspired by behavioral systems theory [12]–[15] have emerged as an alternative to both model-based and RL approaches. The seminal work [13] shows how the LQR problem can be solved directly from persistently exciting data, without explicit identification of a state-space model. Building on this idea, an covariance-based parameterization of the LQR problem have been introduced, resulting in the Data-enabled Policy Optimization (DeePO) method that updates the feedback gain sample-by-sample from online closed-loop data [16], [17]. Compared to classical PO methods [9]–[11], DeePO is both sample- and computationally efficient, performing a single gradient step per sample interval. The DeePO framework has been extended to LPV control [18], model-reference control [19], and output-feedback control, and validated in simulation on a power converter system [20].

In this paper, we propose a unified framework for nonlinear autonomous bicycle control. We design an inner-loop Feedback Linearization (FL) controller to stabilize the bicycle, and place an adaptive DeePO controller in the outer loop to handle modeling errors and time-varying effects. To our knowledge, this is the first real-world implementation of DeePO.

The contributions of this work compared to the existing literature, including the original DeePO algorithm in [17] are:

- An exponentially weighted forgetting factor to improve adaptation to time-varying dynamics and noise.
- A modified DeePO policy update mechanism that mitigates fast-changing dynamics and noise by using reduced update frequencies.
- A novel robustly stabilizing policy initialization via robustness-promoting regularization.
- DeePO is evaluated experimentally on an autonomous bicycle. In particular, DeePO is embedded within a feedback-linearized controller, enhanced with a forgetting factor and modified policy update mechanism, enabling learning-based control beyond the LTI setting considered previously.

The paper is organized as follows: Section II details the bicycle model and control framework. Section III presents the adaptive DeePO algorithm. Section IV discusses simulation and experimental results, and Section V concludes the paper.

Received February 17, 2025; revised 28 December 2025. This work was supported by the Knowledge Foundation (KKS) with grant "Mälardalen University Automation Research Center (MARC)", n. 20240011.

N. Persson and A.V. Papadopoulos are with the Division of Intelligent Future Technologies, Mälardalen University, 721 23 Västerås, Sweden. (e-mails: niklas.persson@mdu.se, alessandro.papadopoulos@mdu.se).

F. Zhao and F. Dörfler are with the Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland. (e-mail: zhaofe@control.ee.ethz.ch, dorfler@ethz.ch

M. Kaheni is with the Power Consulting group, Hitachi Energy Sweden AB, Evenemangsgatan 17, 169 79 Solna, Stockholm, Sweden. (e-mail: mojtaba.kaheni@hitachienergy.com)

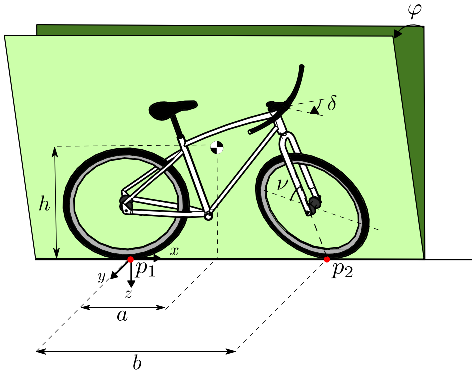

Fig. 1. Illustration of the parameters used in the bicycle model in (1).

A. Notation

We use I_n to denote the n-by-n identity matrix. We use $\rho(\cdot)$ to denote the spectral radius of a square matrix. $A^\top$, Tr(A), and $A^\dagger$ represent the transpose, trace, and pseudoinverse of matrix A, respectively. We use diag$(a, b, \ldots, c)$ to denote a diagonal matrix with diagonal elements being $a, b, \ldots, c$. The 2-norm of matrix A is denoted $\|A\|$. We denote the continuous-time signal x with $x(t)$ and discrete-time with x_t.

II. Autonomous bicycle control

We consider a simple nonlinear model to represent the bicycle dynamics as in [21].

$$\ddot{\varphi}(t) = \frac{g}{h}\sin\big(\varphi(t)\big) + \frac{a}{bh}\cos\big(\varphi(t)\big)v\dot{\delta}(t) - \left(\frac{1}{bh} - \frac{1}{b^2}\tan\big(\delta(t)\big)\tan\big(\varphi(t)\big)\right)\tan\big(\delta(t)\big)v^2, \tag{1}$$

where $\varphi(t)$, $\dot{\varphi}(t)$, $\delta(t)$, and $\dot{\delta}(t)$ denote the lean angle, lean rate, steering angle, and controlled steering rate, respectively. The vertical and horizontal distances from the rear wheel contact point p_1 to the center of gravity are denoted by a and h, respectively. Furthermore, b is the wheelbase, g is gravity, and v is the constant forward velocity. We assume a vertical steering axis, i.e., zero trail and neglect any delay in steering actuation. The parameters in (1) are illustrated in Fig. 1.

A bicycle is self-stabilized between the so-called *weave speed* and *capsize speed*. This region can be identified by analyzing the eigenvalues of the Whipple model and locating the speed interval where all eigenvalues have negative real parts [22]. For the instrumented bicycle used in this work, the required 25 parameters were measured in [23]. In this work, we focus on forward speeds of approximately 8 km/h (2.22 m/s), i.e., below the weave speed as shown in Fig 2. Hence, the system is open-loop unstable, nonlinear, underactuated, and non-holonomic, presenting a challenging control problem.

A. Control overview

A common approach to generate persistently exciting inputs is to apply random signals [24]; however, this is impractical for an unstable bicycle without stabilization. We therefore introduce an FL loop based on (1). Because the system has mismatched relative degree, only partial linearization via output FL is possible [25].

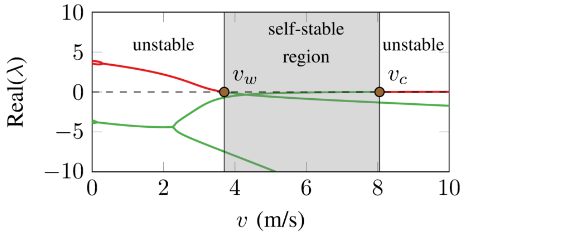

Fig. 2. Eigenvalues of the Whipple model, highlighting the stable and unstable regions of the instrumented bicycle, with the weave speed and capsize speed denoted by v_w and v_c, respectively.

If we choose $x = \begin{bmatrix} x_1, & x_2, & x_3 \end{bmatrix} = \begin{bmatrix} \varphi(t), & \dot{\varphi}(t), & \delta(t) \end{bmatrix}$, $y(t) = \varphi(t)$, $\dot{y}(t) = \dot{\varphi}(t)$, and represent the reference output as $y_r = [y_r(t), \dot{y}_r(t), \ddot{y}_r(t)]$, we can express the considered FL control law as:

$$u(t) = \dot{\delta}(t) = \frac{1}{p(x)}(w - f(x)), \tag{2}$$

where

$$\begin{aligned} f(x) &= \frac{g}{h}\sin(x_1) - \left(\frac{1}{bh} - \frac{1}{b^2}\tan(x_3)\tan(x_1)\right)\tan(x_3)v^2 \\ p(x) &= \frac{a}{bh}\cos(x_1)v, \\ w &= \ddot{y}_r(t) + k_1\left(\dot{y}_r(t) - \dot{y}(t)\right) + k_2\left(y_r(t) - y(t)\right), \end{aligned} \tag{3}$$

with appropriate choices of $k_1 > 0$ and $k_2 > 0$ to partially compensate for the system's nonlinearities and stablize the system. However, because the actual bicycle dynamics are partially unknown and the steering angle $\delta(t)$ remains an internal state, perfect cancellation of nonlinearities is not possible. In particular, the resulting closed-loop system after applying FL is not linear time-invariant (LTI) due to residual nonlinear couplings involving $\tan(\delta(t))$. To establish stability guarantees later, we restrict our analysis to a compact operating region $\Omega := \{(\varphi, \dot{\varphi}, \delta) \mid |\varphi| \le \varphi_{max}, |\delta| \le \delta_{max}\}$, with $\varphi_{max} < \pi/6$ and $\delta_{max} < \pi/4$. Within Ω, all nonlinearities in (1) are continuous, which guarantees that these residual terms remain strictly bounded. Moreover, the FL controller is designed in continuous time using (1) and the estimated parameters $a = 0.550$ m, $h = 0.700$ m, $b = 1.200$ m, $g = 9.82$ m/s^2, $k_1 = 1$, and $k_2 = 6$, but implemented in discrete time with a sample-and-hold mechanism [26], [27]. As a result, discretization, unmodeled dynamics, and parametric uncertainty lead to imperfect cancellation of nonlinearities and potential performance loss. These limitations motivate the use of an additional adaptive layer based on DeePO. In the remainder of the paper, we consider the autonomous bicycle with FL as our target system to control by DeePO, as highlighted by the gray box in Fig. 3.

III. DeePO for autonomous bicycle control

This section first describes a brief overview of LQR. Then, we present a DeePO algorithm enhanced with a forgetting factor for adaptive learning of the LQR based on [16], [17].

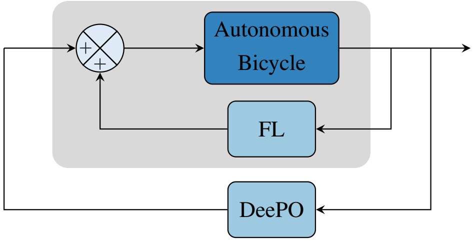

Fig. 3. Control overview

A. The linear quadratic regulator

Consider the discrete-time LTI system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t, \\ h_t &= \begin{bmatrix} Q^{1/2} & 0 \\ 0 & R^{1/2} \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix}, \end{aligned} \tag{4}$$

where $x_t \in \mathbb{R}^n$ is the state, $u_t \in \mathbb{R}^m$ the input, $w_t \in \mathbb{R}^n$ the noise, $h_t \in \mathbb{R}^{n+m}$ the performance output, (A, B) is controllable, and $Q, R \succ 0$. The LQR problem seeks a $K \in \mathbb{R}^{m\times n}$ minimizing the $\mathcal{H}_2$ norm of the closed-loop map $w \mapsto h$. If $A + BK$ is stable, the cost satisfies [28]

$$\|\mathscr{T}(K)\|_2^2 = \mathrm{Tr}\big((Q + K^\top RK)\Sigma_K\big), \tag{5}$$

where Σ_K is the positive definite solution of the Lyapunov equation for the closed-loop dynamics.

B. Data-driven covariance parametrization of the LQR with exponential weighted data

Consider the t-long time series of states, inputs, noises, and successor states

$$\begin{aligned} X_{0,t} &:= \begin{bmatrix} x_0 & x_1 & \dots & x_{t-1} \end{bmatrix} \in \mathbb{R}^{n\times t}, \\ U_{0,t} &:= \begin{bmatrix} u_0 & u_1 & \dots & u_{t-1} \end{bmatrix} \in \mathbb{R}^{m\times t}, \\ W_{0,t} &:= \begin{bmatrix} w_0 & w_1 & \dots & w_{t-1} \end{bmatrix} \in \mathbb{R}^{n\times t}, \\ X_{1,t} &:= \begin{bmatrix} x_1 & x_2 & \dots & x_t \end{bmatrix} \in \mathbb{R}^{n\times t}, \end{aligned} \tag{6}$$

which satisfy the system dynamics

$$X_{1,t} = AX_{0,t} + BU_{0,t} + W_{0,t}. \tag{7}$$

Assume that the data is *persistently exciting (PE)* [12], i.e., the block matrix of input and state data

$$D_{0,t} := \begin{bmatrix} U_{0,t} \\ X_{0,t} \end{bmatrix}, \tag{8}$$

has full row rank, i.e. $\mathrm{rank}(D_{0,t}) = m + n$. Define the covariance of exponentially weighted data as

$$\Phi_t := \frac{1}{t} D_{0,t} S_\lambda D_{0,t}^\top = \begin{bmatrix} U_{0,t} S_\lambda D_{0,t}^\top / t \\ X_{0,t} S_\lambda D_{0,t}^\top / t \end{bmatrix} = \begin{bmatrix} \overline{U}_{0,t} \\ \overline{X}_{0,t} \end{bmatrix}, \tag{9}$$

where $\lambda \in (0, 1)$ is a forgetting factor and $S_\lambda := \mathrm{diag}\{\lambda^{t-1}, \lambda^{t-2}, \dots, 1\}$.

Remark 1: The nonlinear and time-varying nature of an autonomous bicycle, along with parameter variations caused by changes in surface conditions, wind disturbances, and other factors, can render previously measured data less representative of the current operating conditions. To enhance the system's adaptability to new data, we propose adding a forgetting factor. Compared to [17], this approach better captures the evolving system behavior over time by reducing the influence of historical data that may no longer be well aligned with the current dynamics.

Since $D_{0,t}$ has full row rank and $S_\lambda \succ 0$, the covariance matrix is positive definite, i.e., $\Phi_t \succ 0$. Then, for any gain K, there exist a matrix V such that

$$\begin{bmatrix} K \\ I_n \end{bmatrix} = \Phi_t V. \tag{10}$$

where $V \in \mathbb{R}^{(n+m)\times n}$ is the *parameterized policy*. Following the steps discussed in [17], the LQR problem becomes

$$\begin{aligned} &\underset{V}{\text{minimize}} \;\; J_t(V) := \mathrm{Tr}\left((Q + V^\top \overline{U}_{0,t}^\top R \overline{U}_{0,t} V)\Sigma_t(V)\right), \\ &\text{subject to} \;\; \overline{X}_{0,t} V = I_n, \end{aligned} \tag{11}$$

where $\Sigma_t(V) = I_n + \overline{X}_{1,t} V \Sigma_t(V) V^\top \overline{X}_{1,t}^\top$ and the original gain matrix can be recovered as $K = \overline{U}_{0,t} V$.

C. Data-enabled policy optimization for adaptive LQR control with exponentially weighted data

In this subsection, we propose a DeePO algorithm based on our covariance parameterization with exponentially weighted data (10), detailed in Algorithm 1. The DeePO algorithm applies online gradient descent to (11) to update V, taking a single projected gradient step as in (17). Here, the projection

$$\Pi_{\overline{X}_{0,t+1}} := I_{n+m} - \overline{X}_{0,t+1}^\dagger \overline{X}_{0,t+1} \tag{12}$$

onto the nullspace of $\overline{X}_{0,t+1}$ is to ensure the subspace constraint in (11).

Define the feasible set of (11) (i.e., the set of stable closed-loop matrices) as $\mathcal{S}_t := \{V \mid \overline{X}_{0,t} V = I_n, \rho(\overline{X}_{1,t} V) < 1\}$. Then, the gradient can be computed as follows.

Lemma 1 ([17]): For $V \in \mathcal{S}_t$, the gradient of $J_t(V)$ with respect to V is given by

$$\nabla J_t(V) = 2\left(\overline{U}_{0,t}^\top R \overline{U}_{0,t} + \overline{X}_{1,t}^\top P_t \overline{X}_{1,t}\right) V \Sigma_t(V), \tag{13}$$

where P_t satisfies the Lyapunov equation

$$P_t = Q + V^\top \overline{U}_{0,t}^\top R \overline{U}_{0,t} V + V^\top \overline{X}_{1,t}^\top P_t \overline{X}_{1,t} V. \tag{14}$$

For efficient online implementation, Algorithm 1 admits a recursive form. The sample covariance is updated as

$$\Phi_{t+1} = \frac{\lambda t}{t+1}\Phi_t + \frac{1}{t+1}\phi_t \phi_t^\top, \tag{15}$$

where $\phi_t = [u_t^\top, x_t^\top]^\top$. By the Sherman–Morrison formula [29], both Φ_{t+1}^{-1} and V_{t+1} can be updated via rank-one corrections, avoiding matrix inversion [17].

Remark 2: The bounded-input-bounded-output (BIBO) stability of the DeePO controller has been established for LTI systems with bounded process noise [30]. In the present work, the closed-loop system under FL can be represented as a nominal discrete-time LTI system $x_{t+1} = Ax_t + Bu_t + \Delta_t$. The term Δ_t aggregates uncanceled nonlinearities from (1), discretization effects of the sample-and-hold mechanism, and

Algorithm 1 DeePO for direct adaptive LQR control

```
Input: Offline data $(X_{0,t_0}, U_{0,t_0}, X_{1,t_0})$, an initial policy $K_{t_0}$, and a stepsize $\eta$.
for $t = t_0, t_0 + 1, \dots$ do
    Apply $u_t = K_t x_t + e_t$ and measure $x_{t+1}$.
    Update covariance matrices $\Phi_{t+1}$ and $\overline{X}_{1,t+1}$.
    Policy parameterization: given $K_t$, solve $V_{t+1}$ via
        $V_{t+1} = \Phi_{t+1}^{-1} \begin{bmatrix} K_t \\ I_n \end{bmatrix}.$
    Update of the parameterized policy: perform one-step projected gradient descent
        $V'_{t+1} = V_{t+1} - \eta_t \Pi_{\overline{X}_{0,t+1}} \nabla J_{t+1}(V_{t+1}),$    (17)
    where the gradient $\nabla J_{t+1}(V_{t+1})$ is given by Lemma 1.
    Gain update: update the control gain by
        $K_{t+1} = \overline{U}_{0,t+1} V'_{t+1}.$
end for
```

the steering resolution limit. Within the defined operating region Ω, the trigonometric terms in (1) are continuous and locally Lipschitz, ensuring that Δ_t remains bounded. Since the actuator is also subject to saturation, these residual perturbations do not grow unbounded, maintaining the BIBO stability of the overall system.

Remark 3: Using a forgetting factor λ may cause the rank-one update to fail asymptotically. Indeed, the covariance update in (15) follows stable linear dynamics, leading to loss of persistency of excitation, $\Phi_t \to 0$, and hence $\Phi_t^{-1} \to \infty$. A simple remedy is to periodically reset the covariance, i.e., set $\Lambda_t = I_{n+m}$ for $t \in \{T, 2T, \dots\}$. Since the autonomous bicycle operates over a finite horizon, no covariance reset is applied in the experiments of Section IV.

Remark 4: The stepsize η_t is chosen based on the signal-to-noise ratio (SNR) of the online data. A larger SNR allows a more aggressive stepsize, while a smaller SNR requires a conservative choice to maintain stability. Accordingly, we set

$$\eta_t = \frac{\eta_0}{\left\| \overline{U}_{0,t} \Pi_{\overline{X}_{0,t}} \overline{U}_{0,t}^\top \right\|}, \quad t \geq t_0, \tag{16}$$

where η_0 is a constant and the denominator quantifies the SNR.

D. Learning an initial stabilizing policy using robustness-promoting regularization

Algorithm 1 requires a stabilizing initial policy. We promote robustness in the covariance parameterization by adding a regularizer. The feasibility of (11) relies on the Lyapunov equation

$$\Sigma = I_n + \overline{X}_1 V \Sigma V^\top \overline{X}_1^\top, \tag{18}$$

with $\overline{X}_1 V$ as the closed-loop matrix. Under certainty equivalence, the true dynamics satisfy

$$\Sigma = I_n + (\overline{X}_1 - \overline{W}_0) V \Sigma V^\top (\overline{X}_1 - \overline{W}_0)^\top, \tag{19}$$

where $\overline{W}_0$ captures unmodeled disturbances. Limited data can make (18) mismatch (19), yielding non-stabilizing initial policies. To mitigate this, we penalize directions amplifying data uncertainty via

$$\operatorname{Tr}(V \Sigma V^\top \Phi),$$

where Φ is the empirical covariance. The regularized LQR problem becomes

$$\begin{aligned} \min_{V,\, \Sigma \succeq 0} \ & J_t(V) + \gamma \operatorname{Tr}(V \Sigma V^\top \Phi) \\ \text{s.t. } & \Sigma = I_n + \overline{X}_1 V \Sigma V^\top \overline{X}_1^\top, \quad \overline{X}_0 V = I_n, \end{aligned} \tag{20}$$

with $K = \overline{U}_0 V$ and regularization weight $\gamma > 0$. Solving (20) offline using $(X_{0,t_0}, U_{0,t_0}, X_{1,t_0})$ produces a robust stabilizing policy for initializing Algorithm 1.

E. Control gain update rate

To reduce oscillations and noise-induced updates in adaptive control [31], the DeePO policy may be updated at a slower rate than the sampling frequency. A parameter $\xi \in \mathbb{N}$ specifies the update interval, such that the control gain is updated once every ξ iterations of Algorithm 1. In recent work [32], we propose modifications to the DeePO algorithm that further mitigate state perturbations.

IV. Simulations and experiments

In this section, we evaluate the proposed FL-DeePO framework through high-fidelity simulations and real-world experiments on an instrumented autonomous bicycle. We compare performance against a baseline FL controller and a DDPG-based RL controller [9].

A. Instrumented bicycle

The bicycle we consider in experiments is shown in Fig. 4. It is equipped with a rear-wheel hub motor for propulsion and a Dynamixel XH540 servo for steering. The rear wheel is controlled through a Electronic Speed Controller (ESC), and the velocity is estimated using 12 evenly distributed magnets on the rear wheel and a Hall sensor. A constant forward velocity of $v \approx 8$ km/h is maintained via a manually tuned PI controller. The steering servo is controlled through a velocity command, $u_t = \dot{\delta}_t$ rad/s, with a resolution of approximately 0.024 rad/s and saturation at ±4 rad/s. The control algorithms are implemented, and the data is processed using ROS2 Humble running on a Raspberry Pi 4b with Ubuntu 20.04. Finally, a radio controller (RC) lets the operator send wireless commands to the RC receiver mounted on the bicycle.

B. Simulation setup

A CAD model of the instrumented bicycle was designed in SolidWorks, imported into MathWorks Simscape, and controlled through Simulink [21]. The rear and front wheels are connected to the main frame through revolute joints, where the rear joint is actuated with a constant speed corresponding to $v = 8$ km/h. A third revolute joint connects the steering axis to the main frame and is actuated by the control signal $u(t) = \dot{\delta}(t)$. The steering dynamics from $u(t)$ to $\dot{\delta}(t)$ are modeled using the identified transfer function $H(s) = \frac{100+s}{100}$ [2],

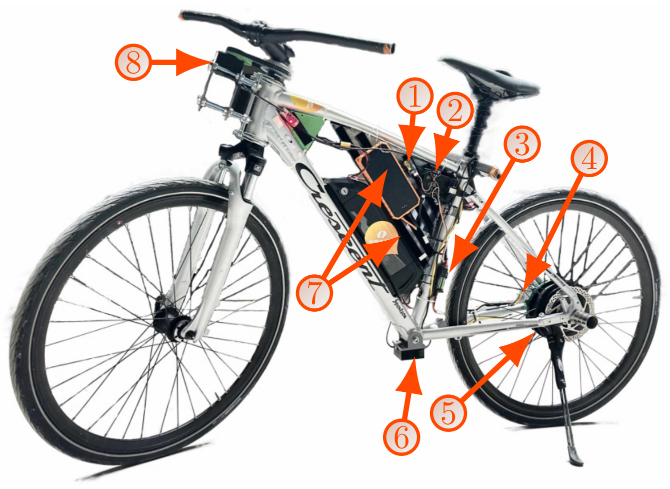

Fig. 4. Instrumented bicycle used in the experiments.

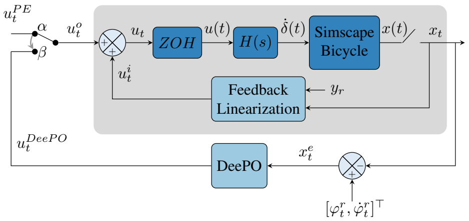

Fig. 5. Control setup where the gray box represents the system controlled using DeePO.

placed in series with the bicycle model, as shown in Fig. 5. The control signal has a resolution of 0.024 rad/s and is saturated at ±4 rad/s.

The control objective is to track a reference lean angle φ_t^r and rate $\dot{\varphi}_t^r$, defining the state as the tracking error $x_t^e = [\varphi_t e, \dot{\varphi}_t^e]^\top = [\varphi_t^r - \varphi_t, \dot{\varphi}_t^r - \dot{\varphi}_t]^\top$. The input u_t combines the inner-loop FL signal u_t^i and an outer-loop signal u_t^o, which is either a persistently exciting input for data collection or the adaptive DeePO control signal, depending on the switch position in Fig. 5. Lean angle, lean rate, and steering angle measurements are corrupted by zero-mean Gaussian noise with standard deviations $\sigma_\varphi = \sigma_\delta = 0.5$ deg and $\sigma_{\dot{\varphi}} = 0.5$ deg/s.

An initial dataset of length $T = 300$ is sampled at 100 Hz with the switch in Fig. 5 in position α. We use the FL controller with added excitation noise $u_t^o = u_t^{PE} \sim \mathcal{N}(0, 0.2)$ tracking the reference $\varphi_t^r = 0$ and its derivative $\dot{\varphi}_t^r = 0$. An initial stabilizing policy is computed by solving the regularized covariance-parameterized LQR problem (20) with $Q = \text{diag}([10, 1])$, $R = 10^{-2}$, and regularization $\gamma = 1$. Higher values of γ promote robustness against uncertainties in the system, while lower values prioritize performance.

Next, the switch in Fig. 5 is set to position β, and Algorithm 1 is used to update the control policy with a learning rate $\eta = 10^{-3}$. To ensure a persistently exciting input, the probing noise e_t is a zero-mean normally distributed random number, which is added to the DeePO output and constructs the input to the system as:

$$u_t^{\text{DeePO}} = u_t^{\text{DeePO}} + e_t, \tag{21}$$

where $e_t = \mathcal{N}(0, 0.2u_t^{\text{DeePO}})$. Moreover, the bicycle tracks a time-varying reference for 60 seconds, as shown in Fig. 7.

To evaluate the update rate for the gain update, line 6, in Algorithm 1, multiple simulations are conducted where ξ varies while the rest of the parameters are kept fixed using an forgetting factor $\lambda = 1 - 10^{-4}$. Four different update rates for the gain update are considered: $\xi = 1$, 10, 50, and 100. The forgetting factor is evaluated in a similar fashion using $\lambda = 1 - 10^{-\zeta}$ with $\zeta = \infty$, 3, 4, 5, 6, 7, where $\zeta = \infty$ corresponds to a controller without a forgetting factor. In these simulations, the control update rate is set to $\xi = 1$, and the rest of the parameters are kept at their initial values. The performance of the controllers is evaluated using the integrated squared error of the lean angle and the lean rate with respect to their respective reference values:

$$\text{ISE}_\varphi = \sum_{i=0}^{t} \varphi_i^{e^2}, \quad \text{ISE}_{\dot{\varphi}} = \sum_{i=0}^{t} \dot{\varphi}_i^{e^2}. \tag{22}$$

We compare the performance of the FL-DeePO controller with only the FL controller and with an RL controller based on the DDPG scheme in [9]. However, with a few important distinctions, their bicycle offers direct control of δ, φ, and v while we assume constant v and only control of $\dot{\delta}$. Moreover, the target system to control is the autonomous bicycle together with FL, as in the case of DeePO. The state vector x_t^e is used as observations, and we track the same reference path as DeePO. The actor has two hidden layers with 300 and 400 Rectified Linear Units (ReLU), and the critic processes state and action in separate 200 ReLU branches, concatenates them, and then applies two further 200 ReLU layers before the output. The discount factor, minibatch size, process noise, Adam optimizer, critic concatenation point, and the use of target networks with soft updates and experience replay are kept the same as in [9]. Training runs for 5000 episodes with 1000 steps at 100Hz, the results from training are presented in Fig.6. The reward function $r(x_t^e, u_t^o)$ is defined as

$$r(x_t^e, u_t^o) = 2c_s - \left(w_\varphi \varphi_i^{e^2} + w_{\dot{\varphi}} \dot{\varphi}_i^{e^2}\right) + r_{term}, \tag{23}$$

where $c_s = 0.01$ is a step survival bonus. The weights $w_1 = \frac{\alpha c_s}{\varphi_{tol}^2}$ and $w_2 = \frac{(1-\alpha)c_s}{\dot{\varphi}_{tol}^2}$ penalize deviations exceeding the tolerances $\varphi_{tol} = 3$ deg and $\dot{\varphi}_{tol} = 10$ deg/s, with $\alpha = 0.85$. The terminal term r_{term} applies a penalty of -50 if the bicycle falls and a bonus of $+20$ upon successful completion of the episode.

C. Simulation results

The tracking performance of the proposed FL-DeePO framework using $\xi = 1$, using only the FL controller and the FL-DPPG controller, is presented in Fig. 7. The results highlight the effectiveness of our unified control framework using FL-DeePO for tracking the time-varying reference lean angle, φ_t^r, and lean rate, $\dot{\varphi}_t^r$. Even though the FL-DeePO only requires one set of offline data, it shows smoother tracking performance, and the lean angle error is reduced compared to using only FL or the FL-DDPG controller, which is trained extensively. Thus, an online, adaptive controller can reduce

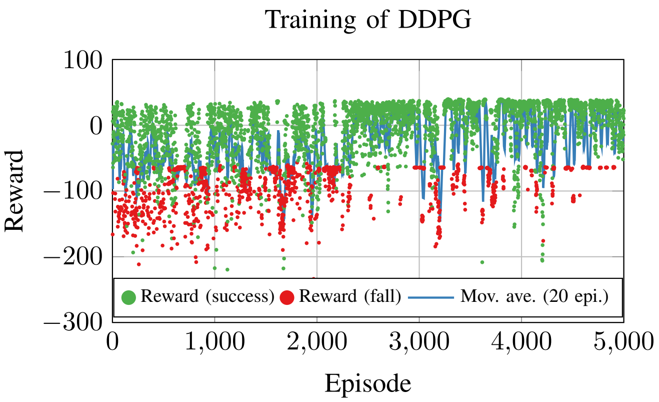

Fig. 6. Training of DDPG for 5000 episodes. The red and green dots denote the episode reward, and the green dots highlight a successful simulation where the bicycle kept its balance for 10 seconds, while a red dot indicates the bicycle fell over during training.

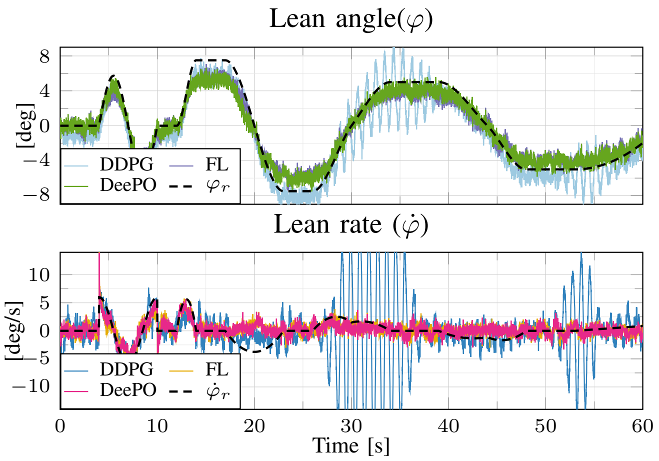

Fig. 7. Tracking performance of the different control approaches in simulation. The top plot illustrates the lean angle tracking results for the FL-DeePO, FL, and the FL-DDPG approach, while the bottom plot highlights the lean rate tracking. The forgetting factor and control policy update rate are set as $\zeta = 4$,$\xi = 1$.

the training time significantly, compared to other direct data-driven approaches, but can still offer satisfactory performance.

When analyzing the impact of the gain update frequency ξ on control performance in Fig. 8 and Fig. 9, the results show that updating the control policy applied to the bicycle at every iteration of the algorithm is unnecessary. Lower update frequencies$\xi = 50$, yield comparable or improved ISE values to updating it at every sample. This suggests that limiting the adaptation rate effectively filters high-frequency measurement noise, preventing the controller from over-fitting to instantaneous disturbances.

Moreover, Fig. 9 presents the ISE values when the forgetting factor varies while the remaining control parameters are kept fixed. The results show that introducing a forgetting factor in DeePO may further enhance control performance. An intermediate forgetting factor of $\lambda = 1 - 10^{-4}$ yields the best performance, while $\lambda = 1 - 10^{-3}$ results in the largest ISE value. Though varying ξ and λ may offer performance enhancements, finding the optimal and likely state-dependent values for these parameters remains an area for future research. Moreover, the consistency of the simulation results highlights the efficiency of our unified control framework, compared to just using FL or using FL together with the DDPG approach, where higher ISE values are present for both the lean angle and lean rate tracking.

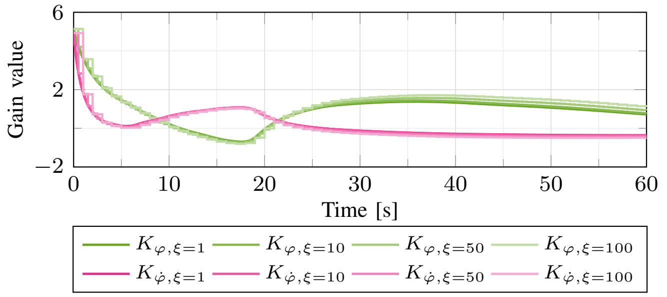

Fig. 8. Evaluation of the control policy in simulation over time with different update rates of the control gain.

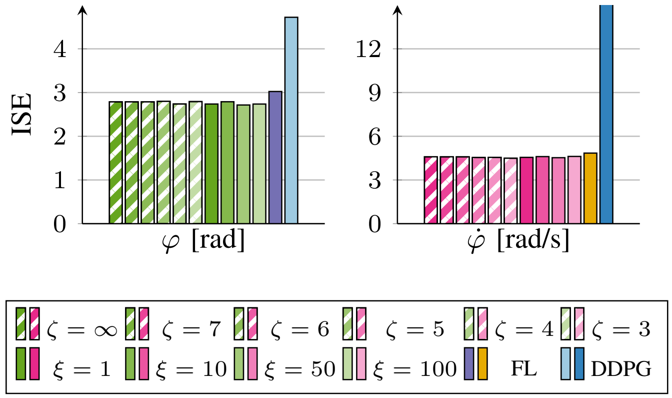

Fig. 9. Integrated squared error of the lean angle and lean rate for different values of control update rate (ξ) and the forgetting factor, $\lambda = 1 - 10^{-\zeta}$ for the FL-DeePO controller in simulation. When ξ is varying, $\zeta = 4$, similarly, when ζ varies, $\xi = 1$. The performance of the FL and the FL-DDPG controller is also included for comparison.

D. Experimental setup

Experiments were conducted in an indoor warehouse with a flat concrete floor (Fig. 10). To generate the initial dataset, the bicycle is accelerated to $v \approx 8$ km/h, after which the FL controller stabilizes the system while tracking a zero reference with added excitation signal $u_{PE} \sim \mathcal{N}(0, 0.2)$. Sensor data and control inputs are processed at 100 Hz on the Raspberry Pi.

The sampled input and state data are post-processed in Matlab, where they are used to construct $U_{0,T}, X_{0,T}, X_{1,T}$, with $T = 300$. Next, the initial policy is obtained by solving (20) in Matlab, with $Q = I_2$, $R = 0.01$, and $\gamma = 1$. In the subsequent experiments, we set the forgetting factor and learning rate as $\lambda = 1 - 10^{-4}$ and $\eta = 10^{-3}$, respectively. While γ, λ, and η are kept the same in the experiments as in the simulation, Q, R, and T are changed in the experiments. The increase in the number of data samples can be explained by the idealized dynamics in simulation compared to experiments, where, for instance, mechanical imperfections, delays, and external

Fig. 10. The indoor environment where the experiments were conducted on a flat concrete floor.

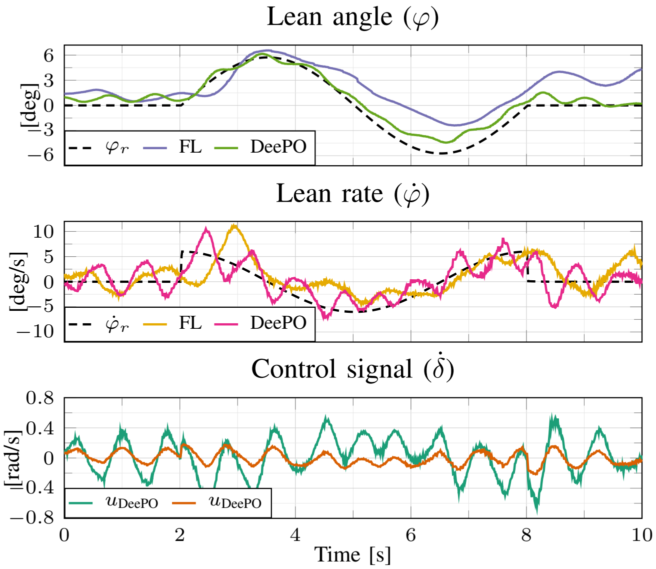

Fig. 11. Experimental results of DeePO where the control policy is updated at every time step, i.e., $\xi = 1$.

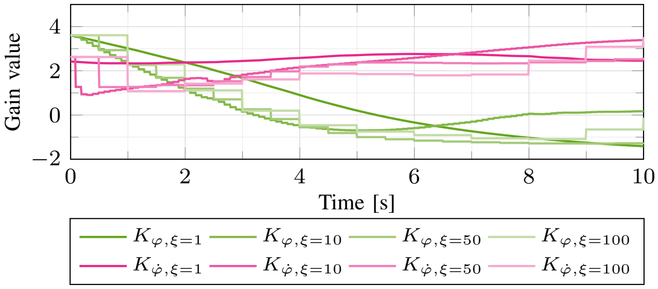

Fig. 12. Evolution of control gains in experiments for different values of ξ.

disturbances are present. Thus, a larger dataset ensures a more reliable estimation of the system's behavior. Moreover, the increase in the weight of $\dot{\varphi}$ in Q is justified by the sensor noise and the need for robustness in experiments. The same time-varying reference lean angle and lean rate used in simulations are also utilized in the experiments. Furthermore, we conduct several experiments where ξ varies as $\xi = 1, 10, 50,$ and 100. Additionally, one experiment is conducted with only the FL controller as a baseline.

E. Experimental results

The lean angle and lean rate tracking performance of the DeePO algorithm is illustrated in the top and middle plots of Fig. 11 using $\xi = 1$. The results of using only FL are also included in the figure. The control signal of DeePO and the total control signal of the system are highlighted in the bottom plot of Fig. 11. The results demonstrate the effectiveness of DeePO and its robust tracking of the reference lean angle and lean rate, with noticeable improvements compared to the FL controller, as evident from the first two plots of Fig. 11. The results also show how DeePO adapts over time, even in the presence of nonlinearities, sensor noise, and external disturbances (e.g., floor imperfections or variations in tire grip). However, the oscillations in lean angle and lean rate have a much higher amplitude in experiments compared to simulations, which can be partly explained by the simulation-to-reality gap. Simulations have unmodelled dynamics and environmental details compared to experiments, such as joint friction, uneven terrain, and approximations made in the model.

The evolution of the control gain values for the lean angle and lean rate are reported in Fig. 12 with varying values for gain update frequency ξ. As observed in simulations, the evolution of the control gains in experiments follows the same trend, even though the update frequencies vary. However, the difference between the evolution of the control gains in experiments and the control gains of the simulation, as presented in Fig. 8, is quite different, which supports the claim of a gap between the simulations and experiments. It also highlights the usefulness of adaptive control methods, which can refine the feedback gain based on online experiment data. The gap between simulations and experiments is also evident when comparing the ISE values in Fig. 9 and Fig. 13. In experiments, updating at every time step or intermediate rates of $\xi = 50$ produces significantly better results than updating at lower rates or using only FL. The considerably higher ISE for $\xi = 100$ and using only FL further highlights the limitations of static or infrequent control gain updates. In simulations where we have control over the initial conditions, noise, and external disturbances, the results are more aligned, and the impact of ξ is not as evident as in experiments. One particular challenge in the experiments was finding a suitable pre-stabilizing control form for initial data and control of the initial conditions. A video demonstration of the simulations and experiments is available online [1].

V. Conclusion

This paper presents a compact FL-DeePO control framework for autonomous bicycle stabilization, where FL provides inital stabilization of the nonlinear bicycle dynamics and DeePO adapts the controller online to compensate for unmodelled dynamics, disturbances, and time-varying effects. In contrast to prior DeePO studies, this work demonstrates

[1] https://youtu.be/5RKnr6tPiuwonline

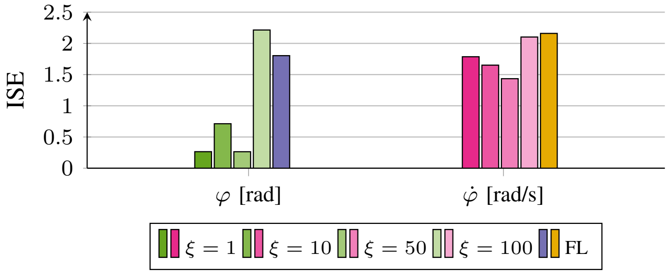

Fig. 13. Integrated squared error for different values of ξ and when using the FL controller alone in experiments.

DeePOs practical integration with FL and its real-world deployment on an autonomous bicycle.

A stabilizing initial policy was obtained from offline data using a covariance-parameterized LQR formulation with a robustness-promoting regularizer. Online adaptation was enabled through an forgetting factor, allowing DeePO to track slowly varying dynamics without requiring explicit system identification. These design choices directly address practical concerns related to noise, unmodeled dynamics, and limited data.

The proposed approach was validated through simulations showing clear performance improvements compared to only using FL or a FL-DDPG control. Moreover, experimental evaluations was conducted on the proposed FL-DeePO controller and the FL, highlighting the effectiveness of the approach in terms of lean angle and lean rate tracking. Additional experiments highlighted the role of the control gain update rate in balancing adaptation. In the future, we plan to enhance the DeePO algorithm by incorporating the robustness regularizer in its online component. Additionally, exploring direct data-driven navigation for the bicycle is another exciting direction for further research.

References

[1] N. C. Sanchez, L. A. Pastor, and K. Larson, "Autonomous bicycles: A new approach to bicycle-sharing systems," in *Int. Conf. Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[2] N. Persson, T. Andersson, A. Fattouh, M. Ekström, and A. V. Papadopoulos, "A comparative analysis and design of controllers for autonomous bicycles," in *Eur. Contr. Conf. (ECC)*, 2021, pp. 1570–1576.

[3] L. Alizadehsaravi and J. K. Moore, "Bicycle balance assist system reduces roll and steering motion for young and older bicyclists during real-life safety challenges," *PeerJ*, vol. 11, p. e16206, 2023.

[4] Y. Zhang, P. Wang, J. Yi, D. Song, and T. Liu, "Stationary balance control of a bikebot," in *IEEE Int. Conf. Rob. & Autom. (ICRA)*, 2014, pp. 6706–6711.

[5] P. Wang, J. Yi, T. Liu, and Y. Zhang, "Trajectory tracking and balance control of an autonomous bikebot," in *IEEE Int. Conf. Rob. & Autom. (ICRA)*, 2017, pp. 2414–2419.

[6] T.-J. Yeh, T.-C. Lin, and A. C.-B. Chen, "Robust balancing and trajectory control of a self-driving bicycle," *IEEE Trans. Contr. Syst. Tech.*, vol. 32, no. 6, pp. 2410–2417, 2024.

[7] M. Defoort and T. Murakami, "Sliding-mode control scheme for an intelligent bicycle," *IEEE Trans. Industrial Electronics*, vol. 56, no. 9, pp. 3357–3368, 2009.

[8] S. Bruni, J. Meijaard, G. Rill, and A. Schwab, "State-of-the-art and challenges of railway and road vehicle dynamics with multibody dynamics approaches," *Multibody System Dynamics*, vol. 49, pp. 1–32, 2020.

[9] S. Choi, T. P. Le, Q. D. Nguyen, M. A. Layek, S. Lee, and T. Chung, "Toward self-driving bicycles using state-of-the-art deep reinforcement learning algorithms," *Symmetry*, vol. 11, no. 2, p. 290, 2019.

[10] L. P. Tuyen and T. Chung, "Controlling bicycle using deep deterministic policy gradient algorithm," in *Int. Conf. Ubiquitous Robots and Ambient Intelligence (URAI)*, 2017, pp. 413–417.

[11] S. Weyrer, P. Manzl, A. Schwab, and J. Gerstmayr, "Path following and stabilisation of a bicycle model using a reinforcement learning approach," *arXiv preprint arXiv:2407.17156*, 2024.

[12] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[13] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Aut. Contr.*, vol. 65, no. 3, pp. 909–924, 2019.

[14] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *Eur. Contr. Conf. (ECC)*, 2019, pp. 307–312.

[15] F. Dörfler, P. Tesi, and C. De Persis, "On the role of regularization in direct data-driven LQR control," in *IEEE Conf. Dec. and Contr. (CDC)*, 2022, pp. 1091–1098.

[16] F. Zhao, F. Dörfler, and K. You, "Data-enabled policy optimization for the linear quadratic regulator," in *IEEE Conf. Dec. and Contr. (CDC)*, 2023, pp. 6160–6165.

[17] F. Zhao, F. Dörfler, A. Chiuso, and K. You, "Data-enabled policy optimization for direct adaptive learning of the LQR," *IEEE Trans. Aut. Contr.*, vol. 70, no. 11, pp. 7217–7232, 2025.

[18] M. Mejari and V. Breschi, "Direct data-driven design of LPV controllers and polytopic invariant sets with cross-covariance noise bounds," *IEEE Control Systems Letters*, 2024.

[19] M. Mejari, V. Breschi, M. B. Dehkordi, S. Formentin, and D. Piga, "Bias correction and instrumental variables for direct data-driven model-reference control," *Eur. Journal of Control*, vol. 86, p. 101327, 2025.

[20] F. Zhao, R. Leng, L. Huang, H. Xin, K. You, and F. Dörfler, "Direct adaptive control of grid-connected power converters via output-feedback data-enabled policy optimization," in *2025 Eur. Contr. Conf. (ECC)*, 2025, pp. 2563–2568.

[21] N. Persson, M. Kaheni, and A. V. Papadopoulos, "A direct data-driven control design for autonomous bicycles," in *IEEE 20th Int. Conf. Automation Science and Engineering (CASE)*, 2024, pp. 114–120.

[22] J. D. Kooijman, A. L. Schwab, and J. P. Meijaard, "Experimental validation of a model of an uncontrolled bicycle," *Multibody System Dynamics*, vol. 19, pp. 115–132, 2008.

[23] N. Persson, *Control and Navigation of an Autonomous Bicycle*. Mälardalen University (Sweden), 2023.

[24] M. Alsalti, V. G. Lopez, and M. A. Müller, "On the design of persistently exciting inputs for data-driven control of linear and nonlinear systems," *IEEE Control Systems Letters*, vol. 7, pp. 2629–2634, 2023.

[25] J. Slotine and W. Li, *Applied Nonlinear Control*, ser. Prentice-Hall International Editions. Prentice-Hall, 1991.

[26] R. Kimber and W. Gray, "On sampled-data implementations of feedback linearizing controllers," in *IEEE Conf. Dec. and Contr. (CDC)*, 1991, pp. 1873–1874.

[27] J. Grizzle and P. Kokotovic, "Feedback linearization of sampled-data systems," *IEEE Trans. Aut. Contr.*, vol. 33, no. 9, pp. 857–859, 1988.

[28] B. D. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

[29] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *The Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 124–127, 1950.

[30] F. Zhao, A. Chiuso, and F. Dörfler, "Policy gradient adaptive control for the LQR: Indirect and direct approaches," *arXiv preprint arXiv:2505.03706*, 2025.

[31] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive control: algorithms, analysis and applications*. Springer Science & Business Media, 2011.

[32] M. Kaheni, N. Persson, V. D. Iuliis, C. Manes, and A. V. Papadopoulos, "A modified adaptive data-enabled policy optimization control to resolve state perturbations," in *IEEE Conf. Dec. and Contr. (CDC)*, 2025, pp. 1999–2005.