2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Physics-Model-Regulated Deep Reinforcement Learning Towards Safety & Stability Guarantees

Hongpeng Cao[1], Yanbing Mao[2], Lui Sha[3], Marco Caccamo[1,4]

Abstract— **Deep reinforcement learning (DRL) has demonstrated impressive success in solving complex control tasks by synthesizing control policies from data. However, the safety and stability of applying DRL to safety-critical systems remain a primary concern and challenging problem. To address the problem, we propose the Phy-DRL: a novel physics-model-regulated deep reinforcement learning framework. The Phy-DRL is novel in two architectural designs: a physics-model-regulated reward and residual control, which integrates physics-model-based control and data-driven control. The concurrent designs enable the Phy-DRL the mathematically provable safety and stability guarantees. Finally, the effectiveness of the Phy-DRL is validated by an inverted pendulum system. Additionally, the experimental results demonstrate that the Phy-DRL features remarkably accelerated training and enlarged reward.**

I. Introduction

Deep reinforcement learning (DRL) has achieved tremendous success in many complex decision-making tasks with high-dimensional state and action spaces, such as vision-based control of robots [1]. Recent advances in DRL synthesize control policies to tackle the non-linearity and uncertainties in complex control tasks from interacting with the environment, achieving impressive performance [2]. This is made possible by leveraging deep neural networks (DNN) for effective approximation of value function, action policy, and representation learning of environmental states, to name a few. However, applying DRL to safety-critical autonomous systems remains a challenging problem. A critical reason is that the control policy of DRL is typically parameterized by DNNs, whose behaviors are hard to predict [3] and verify [4], raising concerns about safety and stability.

In practice, in most autonomous systems, it is common to have access to approximations of the nonlinear system dynamics through the process of reasonable linearization. Using these approximations, the model-based controller can be derived for controlling the system with verifiable property. The fundamental question here is whether we can leverage existing model-based knowledge to regulate the behavior of data-driven Deep Reinforcement Learning (DRL) systems without adversely affecting DRL performance. Additionally, the model-based knowledge could offer theoretical support for ensuring the safety and stability of DRL-enabled systems.

Recently, a research focus has been shifted to the integration of data-driven DRL and model-based control, leading to a residual control diagram, which holds the promise for dealing with complex dynamics while retaining the (provable and verifiable) advantages of the model-based approaches [5], [6], [7], [8], [9]. Such a residual control diagram can take advantage of both model-based controllers and data-driven DRL, as the model-based controller can guide the exploration of DRL agents during training and regulate the behavior of the DRL controller. Meanwhile, the DRL controller learns to effectively deal with the uncertainties and compensate for the model mismatch errors faced by the model-based controller. Inspired by the residual control diagram, we propose a novel physics-model-regulated DRL framework to guide and regulate the pure data-driven approach using model-based knowledge. Specifically, we leverage Lyapunov stability theory to design a lyaponov-like reward function that can encourage the DRL to learn to stable the system. Further more, we derive safety and stability conditions using model-knowledge to provide mathematical provable guarantee. At last, we make the model-based controller and DRL to work in conjunction under the residual control diagram to output more robust control commands.

A. Related work

DRL-enabled control systems should satisfy some safety constraints and also features a property that, if it starts from a safe region, it will eventually converge to the goal state, known as *asymptotically stable* [10]. To realize such safety and stability require, model-based approaches focus on constructing a safety set, and the DRL agent is only allowed to act in this constrained space [5], [11], [12], [13], [14], [15]. In this direction, the control Lyapunov function (CLF) is typically used to constraint the state space with the objective that all actions will lead the system to decent on defined CLF, i.e., towards being stable [12], [13]. However, finding such CLF is often a challenging task for nonlinear systems. Given the desired safety specification, one can also leverage control barrier function [5], [14], [16] and reachability analysis [15] to certify the control command to satisfy the safety requirement. The approaches [5], [14], [15] are mainly designed to ensure the safety of the system, where how to guarantee stability remains an open problem. Moreover, the model-based approaches are generally limited to modeling errors and rely on a more accurate dynamics

[1] Hongpeng Cao is with School of Engineering and Design, Technical University of Munich, Munich, 85748, Germany cao.hongpeng@tum.de

[2] Yanbing Mao is with Engineering Technology Division, Wayne State University, Detroit, MI 48201, USA hm9062@wayne.edu

[3] Lui Sha is with Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA lrs@illinois.edu

[1,4] Marco Caccamo is with School of Engineering and Design and Institute of Robotics and Machine Intelligence, Technical University of Munich, Munich, Germany. mcaccamo@tum.de

This work was supported in part by the NSF CPS-2311084, NSF CPS-2311085, and the Alexander von Humboldt Professorship Endowed by the German Federal Ministry of Education and Research.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

model to expand the safe region [5], [13].

Learning-based approaches aim to embed the knowledge of safety and stability during training, such that the agent is guided to learn to stabilize the system [17], [18]. Chang and Gao in [17] proposed to learn a Lyapunov function from sampled data and use it as an additional critic network to regulate the control policy optimization toward the decrease of the Lyapunov critic values. Similarly, Zhao in [16] proposes to incorporate CLF constraint in objective function for training to prevent the system diverging from equilibrium. The challenge moving forward is how to design DRL to exhibit a provable stability guarantee mathematically. Recently, the seminal work [18] discovered that if the reward of DRL is CLF-like, the systems controlled by a well-trained DRL agent can be proved to retain stability. Building on the work, the challenges moving forward are two folds: what is the formal guidance for constructing such CLF-like rewards for DRL and how to regulate DRL to concurrently guarantee safety and stability?

B. Contributions

To address the aforementioned challenges, we propose the Phy-DRL: a physics-model-regulated deep reinforcement learning framework built on the residual control diagram. The novelty of Phy-DRL is summarized as follows:

- *Safety and stability-aware Reward*. We leverage Lyapunov stability theory to design a new lyaponov-like reward function that can encourage the DRL to learn to stable the system and stay safe.
- *Safety and Stability Conditions for Residual Control*. We derive safety and stability conditions using model-knowledge to provide mathematical provable guarantees.

This paper is organized as follows. In Section II, we present preliminaries. In Sections III, we investigate physics-model-regulated reward, residual control and provable safety and stability of Phy-DRL, respectively. We present the experimental results in Section IV and conclude this work in Section V.

II. Preliminaries

For convenience, Table I summarizes the notations used throughout the paper.

TABLE I

Table of Notation

$\mathbb{R}^n$: the set of n-dimensional real vectors
$\mathbb{N}$: the set of natural numbers
$[\mathbf{x}]_i$: the i-th entry of vector $\mathbf{x}$
$[\mathbf{W}]_{i,:}$: the i-th row of matrix $\mathbf{W}$
$[\mathbf{W}]_{i,j}$: the element at row i and column j of matrix $\mathbf{W}$
$\mathbf{P} \succ 0$: the matrix $\mathbf{P}$ is positive definite
$\top$: the matrix or vector transposition
$\mathbf{I}_n$: the the $n \times n$-dimensional identity matrix
$\mathbf{1}_n$: n-dimensional vector of all ones

A. Real Plant and Residual Control

Without loss of generality, the real system is described by

$$\mathbf{s}(k+1) = \mathbf{A}\mathbf{s}(k) + \mathbf{B}\mathbf{a}(k) + \mathbf{f}(\mathbf{s}(k), \mathbf{a}(k)), \qquad k \in \mathbb{N} \quad (1)$$

where $\mathbf{s}(k) \in \mathbb{R}^n$ is the real-time system state, $f(\mathbf{s}(k), \mathbf{a}(k)) \in \mathbb{R}^n$ is the unknown model mismatch, $\mathbf{a}(k) \in \mathbb{R}^m$ is the control command.

In most autonomous systems, the conventional feedback control method is available to design a baseline controller to partially handle the control problem. This enables us only training a DRL agent to deal with the residual part, where the baseline control is incapable of due to the modeling uncertainties and complexity. As shown in Figure 1, the terminal control command $\mathbf{a}(k)$ from Phy-DRL is given in the residual form:

$$\mathbf{a}(k) = \mathbf{a}_{\text{drl}}(k) + \mathbf{a}_{\text{phy}}(k), \quad (2)$$

where $\mathbf{a}_{\text{drl}}(k)$ denotes the date-driven control command from DRL, while $\mathbf{a}_{\text{phy}}(k)$ denotes the physics-model-based control command.

B. Safety Constraints

The considered safety problems stem from practical regulations or constraints on system states, which motivates the following safety set.

$$\textbf{Safety Set: } \mathbb{X} \triangleq \left\{ \mathbf{s} \in \mathbb{R}^n \middle| \, \underline{\mathbf{v}} \leq \mathbf{D} \cdot \mathbf{s} - \mathbf{v} \leq \overline{\mathbf{v}} \right\}, \quad (3)$$

where $\mathbf{D} \in \mathbb{R}^{h \times n}$, $\mathbf{v}$, $\overline{\mathbf{v}}$ and $\underline{\mathbf{v}} \in \mathbb{R}^h$ are given by safety specifications.

The condition in (3) can cover a significant number of safety problems that are associated with operation regulations and/or safety constraints. Considering autonomous vehicles driving in a school zone in Winter as one example [19], according to traffic regulations, the vehicle speed shall be around 15 mph. To prevent slipping and sliding for safe driving in icy roads, the vehicle slip shall not be larger than 4 mph. Given the information on regulation and safety constraints, we can let

$$\mathbf{s} = \begin{bmatrix} v \\ w \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 1 & 0 \\ 1 & -r \end{bmatrix}, \mathbf{v} = \begin{bmatrix} 15 \\ 0 \end{bmatrix}, \overline{\mathbf{v}} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \underline{\mathbf{v}} = \begin{bmatrix} -2 \\ -4 \end{bmatrix},$$

such that condition in (3) can be equivalently transformed to

$$-2 \leq v - 15 \leq 2, \quad (4)$$

$$-4 \leq v - r \cdot w \leq 4, \quad (5)$$

where v, w, and r denote the vehicle's longitudinal velocity, angular velocity and wheel radius. The inequality (4) means the allowable maximum difference with traffic-regulated velocity (i.e., 15 mph) is 2 mph. While the inequality (5) means the vehicle slip (defined as $v - r \cdot w$) is constrained to be not larger than 4 mph.

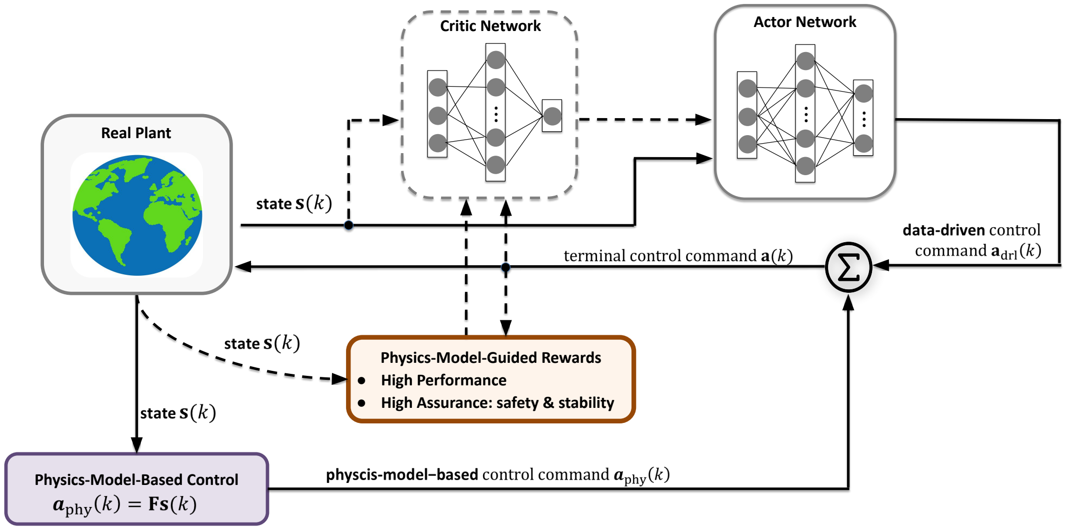

Critic Network
Actor Network
Real Plant
state $\mathbf{s}(k)$
data-driven control command $\mathbf{a}_{\mathrm{drl}}(k)$
terminal control command $\mathbf{a}(k)$
Σ
state $\mathbf{s}(k)$
Physics-Model-Guided Rewards
● High Performance
● High Assurance: safety & stability
state $\mathbf{s}(k)$
Physics-Model-Based Control
$\boldsymbol{a}_{\mathrm{phy}}(k) = \mathbf{Fs}(k)$
physcis-model–based control command $\boldsymbol{a}_{\mathrm{phy}}(k)$

Fig. 1. The plot shows the diagram of the proposed Phy-DRL framework. It consists of a real plant, a physics-model-based controller, a DRL algorithm of *actor-critic* architecture, and a physics-model-guided reward module. The terminal control command is computed by taking the summation of the action generated from the model-based controller and the action output from the actor-network of DRL. The states, control actions and the reward computed from the Physical-Model-Guided Reward module will be saved as training data for optimizing the critic and actor networks. The dashed lines indicate the additional procedures for training.

C. Deep Reinforcement Learning

In this paper, we consider a DRL agent is interacting with the real plant (1) in discrete timesteps, which can be formulated as a Markov Desicion Process (MDP) with $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$. In the MDP, $\mathcal{S}$ represents a set of states, $\mathcal{A}$ a set of actions, and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ the state-transition probability function indicating the probability of a state-action pair leading to a specific next state. The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ maps a state-action pair to a real-valued reward. The discount factor $\gamma \in [0, 1]$ controls the relative importance of immediate and future rewards. The goal in DRL is to find a policy $\pi : \mathcal{S} \mapsto \mathcal{A}$, mapping a state to an action that maximizes the expected return $Q^{\pi}(\mathbf{s}(k), \mathbf{a}_{\mathbf{drl}}(k))$ from step k:

$$\begin{aligned} Q^{\pi}(\mathbf{s}(k), \mathbf{a}_{\mathbf{drl}}(k)) &= \mathbf{E}_{\mathbf{s}(k) \sim \mathbb{S}} \left[\sum_{t=k}^{\infty} \gamma^{t-k} \mathcal{R}\left(\mathbf{s}(t), \mathbf{a}_{\mathbf{drl}}(t)\right) \right] \\ &= \mathbf{E}_{\mathbf{s}(k) \sim \mathbb{S}} \left[\sum_{t=k}^{\infty} \gamma^{t-k} \mathcal{R}\left(\mathbf{s}(t), \pi\left(\mathbf{s}(t)\right)\right) \right]. \end{aligned} \tag{6}$$

We use *actor-critic* architecture for policy searching as it shows low variance and higher sample efficiency during training [20], [21], [22]. As shown in 1, the *actor-critic* architecture comprises critic network to approximate the expected return $Q^{\pi}(\mathbf{s}(k), \mathbf{a}_{\mathbf{drl}}(k))$ and actor network to approximate the policy π and output control action $\mathbf{a}_{\mathrm{drl}}(k)$.

III. Phy-DRL: Physics-Model-Regulated DRL

In this section, we investigate leveraging the available physical knowledge encoded in matrices $\mathbf{A}$ and $\mathbf{B}$ in the dynamics model (1) of the real plant to design DRL towards safety and stability.

A. Safety Envelope

The current safety set formula (3) is not ready for developing the safety- and stability-aware reward $\mathcal{R}(\cdot)$ in (6). To move forward, we introduce an equivalent variant of safety set (3):

$$\widehat{\mathbb{X}} \triangleq \left\{ \mathbf{s} \in \mathbb{R}^n | -\mathbf{1}_h \leq \mathbf{d} \leq \underline{\mathbf{D}} \cdot \mathbf{s}, \quad \overline{\mathbf{D}} \cdot \mathbf{s} \leq \mathbf{1}_h \right\}, \tag{7}$$

where for $i \in \{1, 2, \ldots, h\}$,

$$[\mathbf{d}]_i \triangleq \begin{cases} 1, & \text{if } [\underline{\mathbf{v}} + \mathbf{v}]_i > 0 \\ 1, & \text{if } [\overline{\mathbf{v}} + \mathbf{v}]_i < 0 \\ -1, & \text{if } [\underline{\mathbf{v}} + \mathbf{v}]_i > 0, \ [\underline{\mathbf{v}} + \mathbf{v}]_i < 0 \end{cases} \tag{8}$$

with the $\overline{\mathbf{v}}$, $\underline{\mathbf{v}}$ and $\mathbf{v}$ given in (3), and the subscript $h \in \mathbb{N}$ indicating the number of constraint or regulation conditions. The two sets $\widehat{\mathbb{X}}$ and $\mathbb{X}$ can be equivalent, as stated in the following lemma, whose proof appears in literature [23].

Lemma 1: Consider the set $\mathbb{X}$ defined in (3) and the set $\widehat{\mathbb{X}}$ defined in (7). The $\mathbb{X} = \widehat{\mathbb{X}}$ holds, if and only if $\overline{\mathbf{D}} = \frac{\mathbf{D}}{\overline{\Lambda}}$ and $\underline{\mathbf{D}} = \frac{\mathbf{D}}{\underline{\Lambda}}$, where for $i, j \in \{1, 2, \ldots, h\}$,

$$[\overline{\Lambda}]_{i,j} \triangleq \begin{cases} 0, & \text{if } i \neq j \\ [\overline{\mathbf{v}} + \mathbf{v}]_i\,, & \text{if } [\underline{\mathbf{v}} + \mathbf{v}]_i > 0 \\ [\underline{\mathbf{v}} + \mathbf{v}]_i\,, & \text{if } [\overline{\mathbf{v}} + \mathbf{v}]_i < 0 \\ [\overline{\mathbf{v}} + \mathbf{v}]_i\,, & \text{if } [\overline{\mathbf{v}} + \mathbf{v}]_i > 0, \ [\underline{\mathbf{v}} + \mathbf{v}]_i < 0 \end{cases} \tag{9}$$

$$[\underline{\Lambda}]_{i,j} \triangleq \begin{cases} 0, & \text{if } i \neq j \\ [\underline{\mathbf{v}} + \mathbf{v}]_i\,, & \text{if } [\underline{\mathbf{v}} + \mathbf{v}]_i > 0 \\ [\overline{\mathbf{v}} + \mathbf{v}]_i\,, & \text{if } [\overline{\mathbf{v}} + \mathbf{v}]_i < 0 \\ [-\underline{\mathbf{v}} - \mathbf{v}]_i\,, & \text{if } [\overline{\mathbf{v}} + \mathbf{v}]_i > 0, \ [\underline{\mathbf{v}} + \mathbf{v}]_i < 0. \end{cases} \tag{10}$$

We now introduce the safety envelope, a building block of safety- and stability-aware reward.

$$\Omega \triangleq \left\{ \mathbf{s} \in \mathbb{R}^n \middle| \, \mathbf{s}^\top \mathbf{P} \mathbf{s} \le 1, \; \mathbf{P} \succ 0 \right\}. \tag{11}$$

The following lemma builds a connection between the safety envelope Ω and the safety set $\widehat{\mathbb{X}}$. Specifically, it provides a condition under which the safety envelope Ω is a subset of safety set $\widehat{\mathbb{X}}$. Its formal proof is presented in [23].

Lemma 2: Consider the safety set $\widehat{\mathbb{X}}$ and the safety envelope Ω defined in (7) and (11), respectively. The $\Omega \subseteq \widehat{\mathbb{X}}$ holds, if

$$\overline{\mathbf{D}}\mathbf{P}^{-1}\overline{\mathbf{D}}^\top \le \mathbf{I}_h \text{ and}$$
$$\left[\underline{\mathbf{D}}\mathbf{P}^{-1}\underline{\mathbf{D}}^\top\right]_{i,i} = \begin{cases} \ge 1, \text{ if } [\mathbf{d}]_i = 1 \\ \le 1, \text{ if } [\mathbf{d}]_i = -1 \end{cases}, i \in \{1, \ldots, h\}. \tag{12}$$

The condition (12) in Lemma 2 will be used to compute the model-based control commands (see LMIs (15) and (16)). In addition, Lemma 2 will be used in the proof of safety guarantee.

B. Design of Model-based Controller

We compute the physics-model-based control command $\mathbf{a}_{\text{phy}}(k)$ according to

$$\mathbf{a}_{\text{phy}}(k) = \mathbf{F}\mathbf{s}(k), \text{ with } \mathbf{F} = \mathbf{R}\mathbf{Q}^{-1}, \mathbf{Q}^{-1} = \mathbf{P}. \tag{13}$$

The matrices $\mathbf{Q}^{-1} = \mathbf{P}$ and $\mathbf{R}$ are computed through solving the following LMIs via LMI toolbox [24]:

$$\begin{bmatrix} \alpha \mathbf{Q} & \mathbf{Q}\mathbf{A}^\top + \mathbf{R}^\top \mathbf{B}^\top \\ \mathbf{A}\mathbf{Q} + \mathbf{B}\mathbf{R} & \mathbf{Q} \end{bmatrix} \succ 0, \tag{14}$$

$$\mathbf{I}_h - \overline{\mathbf{D}}\mathbf{Q}\overline{\mathbf{D}}^\top \succ 0, \tag{15}$$

$$\left[\underline{\mathbf{D}}\mathbf{Q}\underline{\mathbf{D}}^\top\right]_{i,i} = \begin{cases} \ge 1, & [\mathbf{d}]_i = 1 \\ \le 1, & [\mathbf{d}]_i = -1 \end{cases}, i \in \{1, \ldots, h\} \tag{16}$$

where $\mathbf{d}$ is given in (8), and $0 < \alpha < 1$ is a predefined scalar when computing the above matrices.

We next present a property of a real plant with residual control, which will be used to prove the safety and stability guarantees of Phy-DRL in the next section. The lemma proof is given in [23].

Lemma 3: For the systems (1) with residual control (2), define the function:

$$V(\mathbf{s}(k)) \triangleq \mathbf{s}^\top(k) \cdot \mathbf{P} \cdot \mathbf{s}(k). \tag{17}$$

If the model-based control (13) in the residual control (2) satisfies the condition (14), the function $V(\mathbf{s}(k))$ along real plant satisfies

$$V(\mathbf{s}(k{+}1)) - V(\mathbf{s}(k)) < r(\mathbf{s}(k), \mathbf{a}(k)) + (\alpha - 1)V(\mathbf{s}(k)) \tag{18}$$

where

$$\begin{aligned} & r(\mathbf{s}(k), \mathbf{a}(k)) \\ & \triangleq (\mathbf{f}(\mathbf{s}(k), \mathbf{a}(k)) {+} \mathbf{B}\mathbf{a}_{\text{drl}}(k))^\top \mathbf{P}(\mathbf{f}(\mathbf{s}(k), \mathbf{a}(k)) {+} \mathbf{B}\mathbf{a}_{\text{drl}}(k)) \\ & \qquad + 2(\overline{\mathbf{A}}\mathbf{s}(k))^\top \mathbf{P} \left(\mathbf{f}(\mathbf{s}(k), \mathbf{a}(k)) {+} \mathbf{B}\mathbf{a}_{\text{drl}}(k)\right). \end{aligned} \tag{19}$$

C. Safety- and Stability-Aware Reward

In light of the condition of the safety envelope (11) and (18), we are ready to propose a safety- and stability-aware reward. For the sake of simplifying the remaining presentations, we define the following:

$$\overline{\mathbf{A}} \triangleq \mathbf{A} + \mathbf{B}\mathbf{F}, \tag{20}$$

The real plant under the control of Phy-DRL can be rewritten as

$$\mathbf{s}(k+1) = \overline{\mathbf{A}}\mathbf{s}(k) + \mathbf{B}\mathbf{a}_{\text{drl}}(k) + \mathbf{f}(\mathbf{s}(k), \mathbf{a}(k)), k \in \mathbb{N} \tag{21}$$

Hereto, the proposed reward is

$$\begin{aligned} \mathcal{R}(\mathbf{s}(k), \mathbf{a}(k)) = & \left[\mathbf{s}^\top(k)\overline{\mathbf{A}}^\top \mathbf{P}\overline{\mathbf{A}}\mathbf{s}(k) - \mathbf{s}^\top(k{+}1)\mathbf{P}\mathbf{s}(k{+}1)\right] \\ & + g(\mathbf{s}(k), \mathbf{a}(k)), \end{aligned} \tag{22}$$

where $\mathbf{P}$ is given in (13). The term $g(\mathbf{s}(k), \mathbf{a}(k))$ is designed for encouraging high operation performance (such as avoiding jerky motions for comfortable driving), while remaining terms are motivated by the aim of safety and stability guarantees.

Remark 1 (Reward Motivation and Explanation): In light of (21), we obtain from (19) that

$$\begin{aligned} & r(\mathbf{s}(k), \mathbf{a}(k)) \\ & = 2\big(\overline{\mathbf{A}}\mathbf{s}(k)\big)^\top \mathbf{P} \left(\mathbf{s}(k+1) - \overline{\mathbf{A}}\mathbf{s}(k)\right) \\ & \qquad + \left(\mathbf{s}(k+1) - \overline{\mathbf{A}}\mathbf{s}(k)\right)^\top) \mathbf{P} \left(\mathbf{s}(k+1) - \overline{\mathbf{A}}\mathbf{s}(k)\right) \\ & = (\mathbf{s}(k+1))^\top \mathbf{P} \left(\mathbf{s}(k+1)\right) - \mathbf{s}^\top(k) \left(\overline{\mathbf{A}}^\top \mathbf{P}\overline{\mathbf{A}}\right) \mathbf{s}(k), \end{aligned}$$

which means the reward (22) includes a sub-reward term $-r(\mathbf{s}_k, \mathbf{u}_k) = \mathbf{s}_k^\top \left(\overline{\mathbf{A}}^\top \mathbf{P}\overline{\mathbf{A}}\right) \mathbf{s}_k - (\mathbf{s}_{k+1})^\top \mathbf{P} \left(\mathbf{s}_{k+1}\right)$ that the data-driven control commands $\mathbf{a}_{\text{drl}}(k)$ from Phy-DRL try to maximize. In another word, the reward (22) has one objective of encouraging choices of control commands for decreasing $r(\mathbf{s}_k, \mathbf{u}_k)$ over time. As proved in Theorem 1, the smaller $r(\mathbf{s}_k, \mathbf{u}_k)$ favors the safety and stability of the system.

D. Phy-DRL: Provable Safety and Stability Guarantees

The conjunctive physics-model-regulated reward (22) and residual control (2) empower the Phy-DRL with the provable safety and stability guarantees. Before presenting the result, we introduce a practical assumption pertaining to the data-driven term (19).

Assumption 1: Along the real plant under the control of a Phy-DRL, the function (19) satisfies

$$r(\mathbf{s}(k), \mathbf{a}(k)) < \beta(\mathbf{s}(k)). \tag{23}$$

Remark 2: The upper bound $\beta(\mathbf{s}(k))$ in (23) is a function of system state $\mathbf{s}(k)$, which is motivated by the fact that both model-based control $\mathbf{a}_{\text{phy}}(k)$ and data-driven control $\mathbf{a}_{\text{drl}}(k)$ depend on system state only. In practice, according to (19), the $\beta(\mathbf{s}(k))$ can be obtained through estimating the residual model mismatch $\mathbf{f}(\mathbf{s}(k), \mathbf{a}(k))$, since the $\overline{\mathbf{A}}$, $\mathbf{B}$, $\mathbf{P}$ and $\mathbf{a}_{\text{drl}}(k)$ in (19) are known. Furthermore, according to (21), the mismatch $\mathbf{f}(\mathbf{s}(k), \mathbf{a}(k))$ can be estimated from the samples $(\mathbf{s}(k), \mathbf{a}(k), \mathbf{s}(k+1))$ generated by the real plant under control of Phy-DRL, for example, using Gaussian Proess (GP) as in [5].

The safety and stability of Phy-DRL are formally presented in the following theorem, whose proof appears in [23].

Theorem 1: Consider the real plant (1) under the control of Phy-DRL, whose reward is given in (22) and the control command is given in (2) with (13), where the involved matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Q}^{-1} = \mathbf{P}$, $\mathbf{F}$ and $\mathbf{R}$ satisfy the conditions (14)–(16). Under Assumption 1, we have:

- If $\frac{\beta(\mathbf{s}(k))}{1-\alpha} < 1$ holds for any $k \in \mathbb{N}$, the control policy of Phy-DRL renders the given safety envelope Ω (11) invariant, i.e., if $\mathbf{s}(1) \in \Omega \subseteq \widehat{\mathbb{X}} = \mathbb{X}$, then $\mathbf{s}(k) \in \Omega \subseteq \widehat{\mathbb{X}} = \mathbb{X}$ for any $k \in \mathbb{N}$.
- If $\beta(\mathbf{s}(k)) + (\alpha - 1) \cdot V(\mathbf{s}(k)) \leq 0$ holds for any $k \in \mathbb{N}$, the control policy of Phy-DRL asymptotically stabilizes the real system (1) and renders the given safety Ω (11) invariant.

IV. Experiments

We demonstrate the proposed Phy-DRL in an inverted pendulum case study. The inverted pendulum system is characterized by the angle of the pendulum from vertical θ, angular velocity of $\omega \triangleq \dot{\theta}$, the position of the cart x and cart velocity $v \triangleq \dot{x}$. The control goal is to stabilize the pendulum at the equilibrium $\mathbf{s}^* = [x^*, v^*, \theta^*, \omega^*]^\top = [0, 0, 0, 0]^\top$.

To obtain system matrix $\mathbf{A}$ and control structure matrix $\mathbf{B}$, we refer to the dynamics model of inverted pendulum described in [25] and consider the approximations $\cos\theta \approx 1$, $\sin\theta \approx \theta$ and $\omega^2 \sin\theta \approx 0$. To better demonstrate the robustness of Phy-DRL, the matrices $\mathbf{A}$ and $\mathbf{B}$ below are obtained without considering friction force, while the real plant is subject to friction force.

$$\mathbf{A} = \begin{bmatrix} 1 & 0.0333 & 0 & 0 \\ 0 & 1 & -0.0565 & 0 \\ 0 & 0 & 1 & 0.0333 \\ 0 & 0 & 0.8980 & 1 \end{bmatrix}, \tag{24}$$

$$\mathbf{B} = [0 \quad 0.0334 \quad 0 \quad -0.0783]^\top. \tag{25}$$

The considered safety conditions are

$$-0.6 \leq x \leq 0.6, \quad -0.4 \leq \theta < 0.4. \tag{26}$$

We let $\alpha = 0.8$. The matrices $\mathbf{P}$ and $\mathbf{F}$ are solved from LMIs (14)–(16) via Matlab LMI toolbox:

$$\mathbf{P} = \begin{bmatrix} 2.0120 & 0.2701 & 1.4192 & 0.2765 \\ 0.2701 & 2.2738 & 5.1795 & 1.0674 \\ 1.4192 & 5.1795 & 31.9812 & 4.9798 \\ 0.2765 & 1.0674 & 4.9798 & 1.0298 \end{bmatrix}, \tag{27}$$

$$\mathbf{F} = \begin{bmatrix} 0.7400 & 3.6033 & 35.3534 & 6.9982 \end{bmatrix}. \tag{28}$$

We let the high-performance reward $g(\mathbf{s}(k), \mathbf{a}(k)) = -a^2(k)$. Given the sub-reward and the knowledge (24)–(25), the residual control (2) and reward (22) of Phy-DRL can be established.

The Phy-DRL's data-driven controller is constructed using a DNN with the structure of *Multi-layer-perception* (MLP) that maps states to continuous actions. As shown in Figure 1, the data-driven controller is in conjunction with the model-based controller (13) to form the terminal residual control command $\mathbf{a}(k)$. For training, we take the cart-pole simulation provided in Open-AI gym and adapt it to a more realistic system with friction and continuous action space. We leverage an off-policy *actor-critic* algorithm DDPG [20] to train the DRL-controller with the reward proposed in (22).

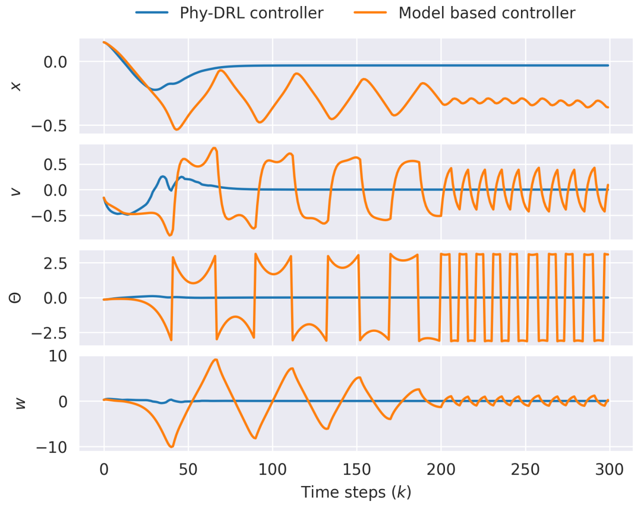


Fig. 2. The plot shows an example of state trajectories of the system controlled by the proposed Phy-DRL. The trajectory of Phy-DRL satisfies the introduced safety and stability conditions.

In the first experiment, we compare system trajectories of cart-poles with a model-based controller and Phy-DRL. We initialize the inverted pendulum in the neighborhood of the equilibrium and let these two controllers control the system, respectively. The system trajectories are shown in Figure 2, observing which we discover that he model-based controller cannot stabilize the car-pole and guarantee its safety, which is due to a large model mismatch between ($\mathbf{A}$, $\mathbf{B}$) and real system model. The control policy of Phy-DRL can stabilize the pendulum system and guarantee system safety, i.e., the safety condition (26) holds for any $k \in \mathbb{N}$.

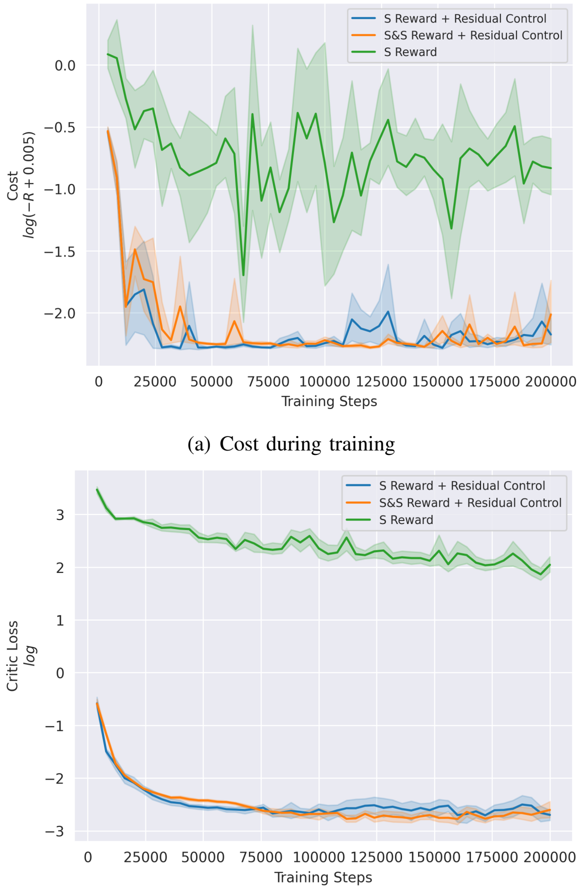
In the second experiment, we showcase the influence of residual control on Phy-DRL training. To perform this, we consider two rewards:

- Stability-aware (S) reward, i.e.,

$$\mathcal{R}(\mathbf{s}(k), \mathbf{a}(k)) = \left[\mathbf{s}^\top(k)\mathbf{P}\mathbf{s}(k) - \mathbf{s}^\top(k+1)\mathbf{P}\mathbf{s}(k+1)\right] + g(\mathbf{s}(k), \mathbf{a}(k)),$$

  which is suggested by the CLF reward studied in [18].
- Safety- and stability-aware (S&S) reward, i.e., the (22).

The plot of log(-reward + 0.005), where a small number is added to make the argument positive, and log(critic loss) are shown in Figure 3. We conclude that in the same Phy-DRL framework, the 'S reward + residual control' and 'S&S + residual control' lead to very similar training behavior. Compared with purely data-driven DRL (see curves of S Reward), the Phy-DRL features remarkably accelerated training and enlarged reward.



(a) Cost during training

(b) Critic loss during training

Fig. 3. The plots illustrate the training progress with and without residual mechanisms. In the residual control diagram, the model-based control guides the exploration of the DRL, significantly improving the converging speed and leading to the enlarged reward (reduced cost).

V. Conclusion

In this paper, we proposed Phy-DRL that leverages model-based knowledge to guide and regulate the data-driven DRL towards safety and stability guarantees for safety-critical autonomous systems. The concurrent physical reward and residual control empower the Phy-DRL with mathematically provable safety and stability guarantees. Through experiments on the inverted pendulum, the Phy-DRL features guaranteed safety, stability, and enhanced robustness while offering remarkably accelerated training and enlarged reward. In the future work, we will experiment with different approaches for estimating the model mismatch and apply the proposed framework to physical systems.

References

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[2] A. Kumar, Z. Fu, D. Pathak, and J. Malik, “RMA: Rapid motor adaptation for legged robots,” in *Robotics: Science and Systems*, 2021.

[3] S. H. Huang, N. Papernot, I. J. Goodfellow, Y. Duan, and P. Abbeel, “Adversarial attacks on neural network policies,” in *5th International Conference on Learning Representations, ICLR 2017, Workshop Track Proceedings*, 2017.

[4] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, “Reluplex: An efficient smt solver for verifying deep neural networks,” in *Computer Aided Verification: 29th International Conference, CAV 2017*, pp. 97–117, Springer, 2017.

[5] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, “End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 3387–3395, 2019.

[6] K. Rana, V. Dasagi, J. Haviland, B. Talbot, M. Milford, and N. Sünderhauf, “Bayesian controller fusion: Leveraging control priors in deep reinforcement learning for robotics.” *arXiv preprint* https://arxiv.org/pdf/2107.09822.pdf.

[7] T. Li, R. Yang, G. Qu, Y. Lin, S. Low, and A. Wierman, “Equipping black-box policies with model-based advice for stable nonlinear control.” *arXiv preprint* https://arxiv.org/pdf/2206.01341.pdf.

[8] R. Cheng, A. Verma, G. Orosz, S. Chaudhuri, Y. Yue, and J. Burdick, “Control regularization for reduced variance reinforcement learning,” in *International Conference on Machine Learning*, pp. 1141–1150, 2019.

[9] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, “Residual reinforcement learning for robot control,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029, IEEE, 2019.

[10] P. G. Drazin and P. D. Drazin, *Nonlinear systems*. No. 10, Cambridge University Press, 1992.

[11] A. Wachi and Y. Sui, “Safe reinforcement learning in constrained markov decision processes,” in *International Conference on Machine Learning*, pp. 9797–9806, 2020.

[12] T. J. Perkins and A. G. Barto, “Lyapunov design for safe reinforcement learning,” *Journal of Machine Learning Research*, vol. 3, no. Dec, pp. 803–832, 2002.

[13] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[14] Z. Qin, T.-W. Weng, and S. Gao, “Quantifying safety of learning-based self-driving control using almost-barrier functions,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 12903–12910, IEEE, 2022.

[15] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, “A general safety framework for learning-based control in uncertain robotic systems,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[16] L. Zhao, K. Gatsis, and A. Papachristodoulou, “A barrier-lyapunov actor-critic reinforcement learning approach for safe and stable control.” *arXiv preprint* https://arxiv.org/abs/2304.04066.

[17] Y.-C. Chang and S. Gao, “Stabilizing neural control using self-learned almost lyapunov critics,” in *2021 IEEE International Conference on Robotics and Automation*, pp. 1803–1809, IEEE, 2021.

[18] T. Westenbroek, F. Castañeda, A. Agrawal, S. Sastry, and K. Sreenath, “Lyapunov design for robust and efficient robotic reinforcement learning,” in *Conference on Robot Learning, CoRL 2022*, vol. 205 of *Proceedings of Machine Learning Research*, pp. 2125–2135, PMLR, 2022.

[19] Y. Mao, Y. Gu, N. Hovakimyan, L. Sha, and P. Voulgaris, “Sl1-simplex: Safe velocity regulation of self-driving vehicles in dynamic and unforeseen environments,” *ACM Transactions on Cyber-Physical Systems*, vol. 7, no. 1, pp. 1–24, 2023.

[20] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” in *4th International Conference on Learning Representations, ICLR*, 2016.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms.” *arXiv preprint* https://arxiv.org/abs/1707.06347.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870, PMLR, 10–15 Jul 2018.

[23] H. Cao, Y. Mao, L. Sha, and M. Caccamo, “Physical deep reinforcement learning: Safety and unknown unknowns.” *arXiv preprint* https://arxiv.org/abs/2305.16614.

[24] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[25] R. V. Florian, “Correct equations for the dynamics of the cart-pole system,” *Center for Cognitive and Neural Studies, Romania*, 2007.