Simplex-enabled Safe Continual Learning Machine

Yihao Cai
Wayne State Univerity, USA
yihao.cai@wayne.edu

Hongpeng Cao
Technical University of Munich, Germany
cao.hongpeng@tum.de

Yanbing Mao
Wayne State Univerity, USA
hm9062@wayne.edu

Lui Sha
University of Illinois Urbana-Champaign, USA
lrs@illinois.edu

Marco Caccamo
Technical University of Munich, Germany
mcaccamo@tum.de

Abstract

This paper proposes the **SeC-Learning Machine:** Simplex-enabled safe continual learning for safety-critical autonomous systems. The SeC-learning machine is built on Simplex logic (that is, "using simplicity to control complexity") and physics-regulated deep reinforcement learning (Phy-DRL). The SeC-learning machine thus constitutes HP (high performance)-Student, HA (high assurance)-Teacher, and Coordinator. Specifically, the HP-Student is a pre-trained high-performance but not fully verified Phy-DRL, continuing to learn in a real plant to tune the action policy to be safe. In contrast, the HA-Teacher is a mission-reduced, physics-model-based, and verified design. As a complementary, HA-Teacher has two missions: backing up safety and correcting unsafe learning. The Coordinator triggers the interaction and the switch between HP-Student and HA-Teacher. Powered by the three interactive components, the SeC-learning machine can i) assure lifetime safety (i.e., safety guarantee in any continual-learning stage, regardless of HP-Student's success or convergence), ii) address the Sim2Real gap, and iii) learn to tolerate unknown unknowns in real plants. The experiments on a cart-pole system and a real quadruped robot demonstrate the distinguished features of the SeC-learning machine, compared with continual learning built on state-of-the-art safe DRL frameworks with approaches to addressing the Sim2Real gap.

1 Introduction

Deep reinforcement learning (DRL) has been integrated into many autonomous systems (see exampels in Figure 1) and have demonstrated breakthroughs in sequential and complex decision-making in broad areas, ranging from autonomous driving [1, 2] to chemical processes [3, 4] to robot locomotion [5, 6]. Such learning-integrated systems promise to revolutionize many processes in different industries with tangible economic impact [7, 8]. However, the public-facing AI incident database [9] reveals that machine learning (ML) techniques, including DRL, can deliver remarkably high performance but no safety assurance [10]. Hence, the high-performance DRL with verifiable safety assurance is even more vital today, aligning well with the market's need for safe ML technologies.

Preprint. Under review.

1.1 Related Work on Safe DRL

Significant efforts have been devoted to promoting safe DRL in recent years, including developing safety-embedded rewards and residual action policies and deriving verifiable safety, as detailed below.

The safety-embedded reward is crucial for a DRL agent to learn a high-performance action policy with verifiable safety. The control Lyapunov function (CLF) is the potential safety-embedded reward [11, 12, 13, 14]. Meanwhile, the seminal work [15] revealed that a CLF-like reward can enable DRL with verifiable stability. At the same time, enabling verifiable safety is achievable by extending CLF-like rewards with given safety conditions or regulations. However, systematic guidance for constructing such CLF-like rewards remains open.

The residual action policy is another shift in the focus of safe DRL, which integrates data-driven DRL action policy and physics-model-based action policy. The existing residual diagrams focus on stability guarantee [16, 17, 18, 19], with the exception being [20] on safety guarantee. However, the physics models considered are nonlinear and intractable, which thwarts delivering a verifiable safety guarantee or assurance, if not impossible.

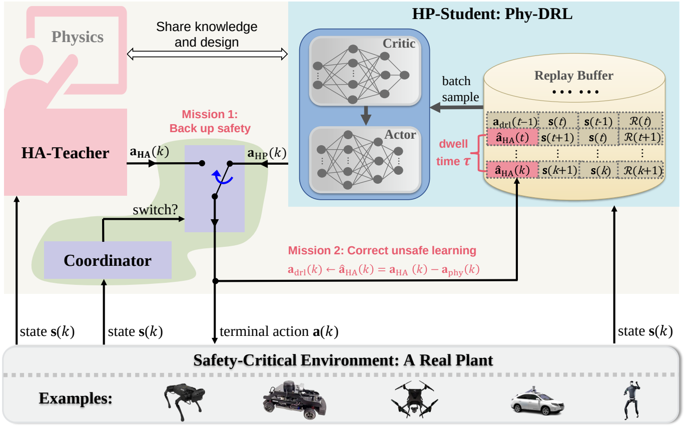

Physics
Share knowledge and design
HP-Student: Phy-DRL
Critic
Actor
batch sample
Replay Buffer
• • • • • •
$\mathbf{a}_{\mathrm{drl}}(t-1)$
$\mathbf{s}(t)$
$\mathbf{s}(t-1)$
$\mathcal{R}(t)$
$\hat{\mathbf{a}}_{\mathrm{HA}}(t)$
$\mathbf{s}(t+1)$
$\mathbf{s}(t)$
$\mathcal{R}(t+1)$
$\hat{\mathbf{a}}_{\mathrm{HA}}(k)$
$\mathbf{s}(k+1)$
$\mathbf{s}(k)$
$\mathcal{R}(k+1)$
dwell time τ
Mission 1: Back up safety
HA-Teacher
$\mathbf{a}_{\mathrm{HA}}(k)$
$\mathbf{a}_{\mathrm{HP}}(k)$
switch?
Coordinator
Mission 2: Correct unsafe learning
$\mathbf{a}_{\mathrm{drl}}(k) \leftarrow \hat{\mathbf{a}}_{\mathrm{HA}}(k) = \mathbf{a}_{\mathrm{HA}}(k) - \mathbf{a}_{\mathrm{phy}}(k)$
state $\mathbf{s}(k)$
state $\mathbf{s}(k)$
terminal action $\mathbf{a}(k)$
state $\mathbf{s}(k)$
Safety-Critical Environment: A Real Plant
Examples:

Figure 1: SeC-Learning Machine.

The verifiable safety (i.e., having verifiable conditions of safety guarantee) is enabled in the recently developed Phy-DRL (physics-regulated DRL) framework [21, 22]. Meanwhile, Phy-DRL can address the aforementioned open problems. Summarily, Phy-DRL simplifies a nonlinear system dynamics model as an analyzable and tractable linear one. This linear model can then be a model-based guide for constructing the safety-embedded (CLF-like) reward and residual action policy.

1.2 Challenges and Open Problems

Although safe DRL has developed significantly, DRL-enabled autonomous systems still face formidable safety challenges, rooting in the Sim2Real gap and unknown unknowns on real plants.

Challenge 1: Sim2Real Gap. Due to the expense of data acquisition and potential safety concerns in real-world settings, the prevalent DRL involves training a policy within a simulator using synthetic data and deploying it onto the physical platforms. The discrepancy between the simulated environment and the real scenario thus leads to the Sim2Real gap that degrades the performance of the pre-trained DRL in a real plant. Numerous approaches have been developed to address Sim2Real gap [23, 24, 25, 26, 27, 28, 29, 30, 31]. The common aim of these approaches is to enhance realism in the simulator through, for example, domain randomization [32] and delay randomization (mimicking asynchronous communication and sampling) [28]. These approaches can address the Sim2Real gap to different degrees. However, the unrevealed gap still persistently impedes safety assurance if DRL (also other ML) is not trained or learning in the real plant in real environments, using real-time data.

Challenge 2: Unknown Unknowns. The unknown unknowns generally refer to outcomes, events, circumstances, or consequences that are not known in advance and cannot be predicted in time and distributions [33]. The dynamics of many learning-enabled systems (e.g., autonomous vehicles [34] and airplanes [35]) are governed by conjunctive known knowns (e.g., Newton's laws of motion), known unknowns (e.g., Gaussian noise without knowing mean and variance), and unknown unknowns (due to, for example, unforeseen operating environments and DNNs' huge parameter space, intractable activation and hard-to-verify). The safety assurance also requires resilience to unknown unknowns, which is very challenging. The reasons root in characteristics of unknown unknowns: almost zero historical data and unpredictable time and distributions, leading to unavailable models for scientific discoveries and understanding.

Intuitively, enabling the continual learning of the DRL agent in the real plant – using its real-time data generated in real environments – is the way to address Challenges 1 and 2. However, two safety problems related to the prospect of continual learning arise.

Problem 1.1. ***How do we teach or assist the DRL agent to correct his unsafe continual learning?***

Problem 1.2. ***Facing an unsafe DRL agent, how can we guarantee the real-time safety of a system?***

1.3 Contribution: Simplex-enabled Safe Continual Learning Machine

If successful, continual learning in a real plant can directly address the Sim2Real gap and learn to tolerate unknown unknowns. However, the DRL's real-time action policy during continual learning cannot be fully verified and can have software faults. Continual learning shall run on a fault-tolerant architecture to address this safety concern. Simplex – using simplicity to control complexity [36, 37] – is a successful software architecture for complex safety-critical autonomous systems. The core of Simplex uses verified and simplified high-assurance controller to control the unverified high-performance and complex controller. Meanwhile, recently developed Phy-DRL theoretically and experimentally features fast training with verifiable safety [21, 22]. These motivate us to develop the **Simplex-enabled safe continual learning (SeC-learning) machine** to address Problem 1.1 and Problem 1.2, which is built on Simplex architecture and Phy-DRL. As shown in Figure 1, the SeC-learning machine constitutes HP (high performance)-Student, HA (high assurance)-Teacher, and coordinator. The HP-Student is a pre-trained Phy-DRL and continues to learn to tune the action policy to be safe in a real plant. The HA-Teacher action is a verified, mission-reduced, and physics-model-based design. As a complementary, HA-Teacher has two missions: backing up safety and teaching to correct unsafe learning of HP-Student. The coordinator triggers the switch and the interaction between the HP-Student and HA-Teacher to ensure lifetime safety (i.e., safety guarantee in any stage of continual learning, regardless of HP-Student's success or convergence).

Table 1: Notations throughout Paper

$\mathbb{R}^n$	set of n-dimensional real vectors
$\mathbb{N}$	set of natural numbers
$[\mathbf{x}]_i$	i-th entry of vector $\mathbf{x}$
$[\mathbf{W}]_{i,:}$	i-th row of matrix $\mathbf{W}$
$[\mathbf{W}]_{i,j}$	matrix $\mathbf{W}$'s element at row i and column j
$\mathbf{P} \succ (\prec)\ 0$	matrix $\mathbf{P}$ is positive (negative) definite
$\top$	matrix or vector transposition
$\lvert\cdot\rvert$	set cardinality, or absolute value
$\mathbf{I}_n$	n-dimensional identity matrix
$\mathbf{O}_{m\times n}$	$m \times n$-dimensional zero matrix
$\lceil x \rceil$	$m \times n$-dimensional zero matrix

2 Preliminaries: Safety Definition

The dynamics of a real plant can be described by

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

where $\mathbf{f}(\mathbf{s}(k), \mathbf{a}(k)) \in \mathbb{R}^n$ is the unknown model mismatch, $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times m}$ denote known system matrix and control structure matrix, respectively, $\mathbf{s}(k) \in \mathbb{R}^n$ is real-time system state of real plant, $\mathbf{a}(k) \in \mathbb{R}^m$ is real-time action from SeC-learning machine.

The actions of the SeC-learning machine aims to constrain the states of a real plant to the safety set:

$$\text{Safety set}: \ \mathbb{X} \triangleq \left\{ \mathbf{s} \in \mathbb{R}^n \middle| \, \underline{\mathbf{v}} \le \mathbf{D}\cdot\mathbf{s} - \mathbf{v} \le \overline{\mathbf{v}}, \mathbf{D} \in \mathbb{R}^{h\times n}, \ \text{ with } \mathbf{v}, \overline{\mathbf{v}}, \underline{\mathbf{v}} \in \mathbb{R}^h \right\}. \tag{2}$$

where $\mathbf{D}$, $\mathbf{v}$, $\overline{\mathbf{v}}$ and $\underline{\mathbf{v}}$ are given in advance for formulating $h \in \mathbb{N}$ safety conditions. In the SeC-learning machine, the safety set is not directly used to embed high-dimensional safety conditions (indicated by $h \in \mathbb{N}$ in Equation (2)) into the DRL reward since the reward is a real one-dimensional value. To address the problem, the concept of safety envelope was introduced in [21, 22, 38], whose condition is one-dimensional and which will be designed to be a subset of the safety set $\mathbb{X}$.

$$\text{Safety envelope: } \Omega \triangleq \left\{ \mathbf{s} \in \mathbb{R}^n | \, \mathbf{s}^\top \cdot \mathbf{P} \cdot \mathbf{s} \leq 1, \ \mathbf{P} \succ 0 \right\}, \tag{3}$$

building on which safety definition is introduced below.

Definition 2.1. *Consider the safety envelope* Ω (3) *and safety set* $\mathbb{X}$ (2)*. The real plant* (1) *is said to be safe, if given any* $\mathbf{s}(1) \in \Omega \subseteq \mathbb{X}$*, the* $\mathbf{s}(k) \in \Omega \subseteq \mathbb{X}$ *holds for any time* $k \in \mathbb{N}$.

3 Design Overview: SeC-Learning Machine

The proposed SeC-learning machine aims to address Problem 1.1 and Problem 1.2 with capabilities of assuring lifetime safety, addressing the Sim2Real gap, and tolerating unknown unknowns. To do so, as shown in Figure 1, the learning machine is designed to have three critical interactive components:

- **HP-Student** is a pre-trained Phy-DRL (physics-regulated DRL [21, 22]) model and continues to learn in a safety-critical real plant to tune his action policy to be safe.
- **HA-Teacher** is a verifiable and analyzable physics-model-based action policy with two missions: backing up the safety of a real plant and correcting unsafe learning of HP-Student.
- **Coordinator** triggers the switch and interaction between HP-Student and HA-Teacher by monitoring the real-time system states. Specifically, when the real-time states of the real plant under the control of HP-Student approach the safety boundary, the coordinator triggers the switch to HA-Teacher and the correction of unsafe actions in learning. In other words, the HA-Teacher takes over the HP-Student to control the real plant to safe (i.e., backing up safety). Meanwhile, the HA-Teacher uses his safe actions to correct the HP-Student's unsafe actions in the replay buffer for learning. Once the real-time states return to a safe region, the coordinator triggers the switch back to HP-Student and terminates the learning correction.

Next, we detail the designs of the three interactive components in Sections 4 to 6, respectively.

4 SeC-Learning Machine: HP-Student Component

The HP-Student builds on Phy-DRL (physics-regulated deep reinforcement learning) proposed in [21, 22]. The critical reason is that Phy-DRL's training mission is pre-defined: searching for an action policy that renders the assigned safety envelope invariant. In this way, HP-Student can share his mission with HP-Student and Coordinator, so they can have a common goal in the learning machine: rendering the safety envelope invariant in a safety-critical real plant in the face of Sim2Real gap and unknown unknowns. We next detail the designs of HP-Student.

4.1 HP-Student: Residual Action Policy and Safety-embedded Reward

Following Phy-DRL in [21, 22], the HP-Student adopts the concurrent residual action policy and safety-embedded reward, as they can offer fast and stable training and successfully encode the safety envelope Ω. The residual action formula is

$$\mathbf{a}_{\text{HP}}(k) = \underbrace{\mathbf{a}_{\text{drl}}(k)}_{\text{data-driven}} + \underbrace{\mathbf{a}_{\text{phy}}(k) \ (:= \mathbf{F} \cdot \mathbf{s}(k))}_{\text{model-based}}, \tag{4}$$

where $\mathbf{a}_{\text{drl}}(k)$ denotes a date-driven action from DRL, while $\mathbf{a}_{\text{phy}}(k)$ is a physics-model-based action ($\mathbf{F}$ is our design). Meanwhile, the safety-embedded reward:

$$\mathcal{R}(\mathbf{s}(k), \mathbf{a}_{\text{drl}}(k)) = \underbrace{\mathbf{s}^\top(k) \cdot \mathbf{H} \cdot \mathbf{s}(k) - \mathbf{s}^\top(k+1) \cdot \mathbf{P} \cdot \mathbf{s}(k+1)}_{\triangleq \, r(\mathbf{s}(k), \, \mathbf{s}(k+1))} + \ w(\mathbf{s}(k), \mathbf{a}_{\text{HP}}(k)), \tag{5}$$

where the sub-reward $w(\mathbf{s}(k), \mathbf{a}(k))$ aims at high-performance operations (e.g., minimizing energy consumption of resource-limited robots [39, 40]). In contrast, the sub-reward $r(\mathbf{s}(k), \mathbf{s}(k+1))$ is safety-critical. Equation (5) also defines:

$$\mathbf{H} \triangleq \overline{\mathbf{A}}^\top \cdot \mathbf{P} \cdot \overline{\mathbf{A}}, \quad \text{with} \quad \overline{\mathbf{A}} \triangleq \mathbf{A} + \mathbf{B} \cdot \mathbf{F} \ \text{and} \ 0 \ \prec \ \mathbf{H} \ \prec \ \alpha \cdot \mathbf{P}, \ \ \alpha \in (0, 1). \tag{6}$$

The matrices $\mathbf{P}$ and $\mathbf{F}$ are design variables, using the available physics-model knowledge ($\mathbf{A}$, $\mathbf{B}$). Their automatic computations will be discussed in Section 4.2.

4.2 HP-Student: Controllable Contribution Ratio

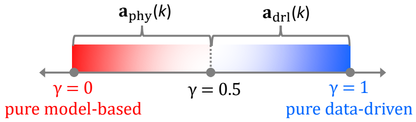


Figure 2: Contribution ratio γ.

In residual diagram (4), the contribution ratio between data-driven and model-based policies controls HP-Student's safety and system performance. To understand this, we define the contribution ratio as $\gamma \triangleq \frac{space\{|\mathbf{a}_{\text{drl}}(k)|, \forall k \in \mathbb{N}\}}{space\{|\mathbf{a}_{\text{drl}}(k)|, \forall k \in \mathbb{N}\} + space\{|\mathbf{a}_{\text{phy}}(k)|, \forall k \in \mathbb{N}\}}$. As depicted in Figure 2, if γ approaches 0, i.e., the physics-model-based action policy dominates the integration, featuring analyzable and verifiable behavior but limited performance. If γ approaches 1, i.e., the data-driven policy dominates the integration, featuring high performance but hard-to-analyze and hard-to-verify. Therefore, a controllable contribution ratio is desired. The controllable space of data-driven actions is innate, as Phy-DRL is built on DDPG [41], which directly maps states to actions within the interval $[-1, 1]$ using the Tanh activation function. The action space can be rescaled with a controllable magnitude factor m to expand the space to $[-m, m]$. So, the remaining job is to enable the controllable space of physics-model-based actions:

$$\textbf{Action Space: } \mathbb{A}_{\text{phy}} \triangleq \left\{ \mathbf{a}_{\text{phy}} \in \mathbb{R}^{m} \mid \underline{\mathbf{z}} \le \mathbf{C} \cdot \mathbf{a}_{\text{phy}} - \mathbf{z} \le \overline{\mathbf{z}}, \text{ with } \mathbf{C} \in \mathbb{R}^{g \times m},\ \mathbf{z}, \overline{\mathbf{z}}, \underline{\mathbf{z}} \in \mathbb{R}^{m} \right\}. \quad (7)$$

where $\mathbf{C}, \mathbf{z}, \overline{\mathbf{z}}$ and $\underline{\mathbf{z}}$ are users' options for controlling the space of model-based actions. Equation (4) shows the model-based action completely depends on $\mathbf{F}$. So,we shall redesign $\mathbf{F}$ to control model-based action to space (7). Due to the page limit, the proposed redesign for delivering $\mathbf{F}$, reward (5) and controllable action space (7) is presented in Appendix B.2.

4.3 HP-Student: Continual Learning

HP-Student is a Phy-DRL model pre-trained in a simulator or another domain, which takes the *actor-critic* architecture-based DRLs such as [41] [42] for training. The pre-trained Phy-DRL model has an action policy and an action-value function. When deployed to a new safety-critical environment, the SeC-learning machine enables the pre-trained policy to continually and safely search for a safe policy that maximizes the expected return.

Sampling efficiency is one of the important considerations for continual learning in the real world. Experience replay (ER) [43] allows off-policy algorithms to reuse the experience collected in the past, greatly improving the sampling efficiency and avoiding forgetting the learned knowledge [44]. ER is also beneficial for breaking the correlation between adjacent transitions to avoid sampling bias for a stable learning process. Those features are very important in continual learning, where online data is limited due to the expensive interaction on the physical system. During the online inference, we continuously store the real transitions realized by safe high-performance action of HP-Student or corrected unsafe data-driven action by HA-Teacher to the replay buffer. Specifically, as illustrated in Figure 1, if the HP-Student's action $\mathbf{a}_{\text{HP}}(k)$ leads to unsafe behavior of a real plant, HA-Teacher takes over his role of controlling real plant to be safe, and corrects his unsafe data-driven action $\mathbf{a}_{\text{drl}}(k)$ to $\widehat{\mathbf{a}}_{\text{HA}}(k)$ according to

$$\mathbf{a}_{\text{drl}}(k) \leftarrow \widehat{\mathbf{a}}_{\text{HA}}(k) \triangleq \mathbf{a}_{\text{HA}}(k) - \mathbf{a}_{\text{phy}}(k), \quad (8)$$

where $\mathbf{a}_{\text{phy}}(k)$ is HP-student's model-based action in residual action policy (4), and $\mathbf{a}_{\text{HA}}(k)$ is the action from HA-Teacher, whose design is presented in Section 6. Meanwhile, the online learning process will uniformly sample a minibatch of transitions for learning or training [45].

Remark 4.1. *Equation* (8) *indicates that for HP-Student's residual action policy* (4)*, the correction in performed only on data-driven action* $\mathbf{a}_{drl}(k)$*, i.e., not including model-based action* $\mathbf{a}_{phy}(k)$*. The reason is that although the model-based design has limited performance and a small safe operation region due to model mismatch, it is analyzable and verifiable, and its policy is invariant because of his invariant physics-model knowledge* $(\mathbf{A},\ \mathbf{B})$.

5 SeC-Learning Machine: Coordinator Component

The Coordinator triggers the switch between HP-Student and HA-Teacher to control the real plant by monitoring the system's state in real-time. The switching logic of terminal action applied to a real plant is described below.

$$\mathbf{a}(k) = \begin{cases} \mathbf{a}_{\text{HA}}(k), & \text{if } \mathbf{s}^\top(t) \cdot \mathbf{P} \cdot \mathbf{s}(t) \geq \varepsilon < 1 \text{ and } t \leq k \leq t + \tau,\ \tau \in \mathbb{N} \\ \mathbf{a}_{\text{HP}}(k), & \text{otherwise} \end{cases} \tag{9}$$

synchronizing with which is the correcting logic of HP-Student's unsafe actions in his replay buffer:

$$\mathbf{a}_{\text{drl}}(k) = \begin{cases} \widehat{\mathbf{a}}_{\text{HA}}(k), & \text{if } \mathbf{s}^\top(t) \cdot \mathbf{P} \cdot \mathbf{s}(t) \geq \varepsilon < 1 \text{ and } t \leq k \leq t + \tau,\ \tau \in \mathbb{N} \\ \mathbf{a}_{\text{drl}}(k), & \text{otherwise} \end{cases} \tag{10}$$

where $\widehat{\mathbf{a}}_{\text{HA}}(k)$ is the corrected unsafe action by HA-Teacher, defined in Equation (8). Noting that real plant initially operates from a safety envelope, we can observe from Equation (9) and Equation (10) that the triggering condition of the switch from HP-Student to HA-Teacher and the unsafe action correction is $\mathbf{s}^\top(t) \cdot \mathbf{P} \cdot \mathbf{s}(t) \geq \varepsilon$. In addition, the requirement '$t \leq k \leq t + \tau$' indicates that once HA-Teacher takes over the role of HP-Student, he has a dwell time $\tau \in \mathbb{N}$; after the dwell time, he returns the control role to HP-Student. The dwell time has two considerations:

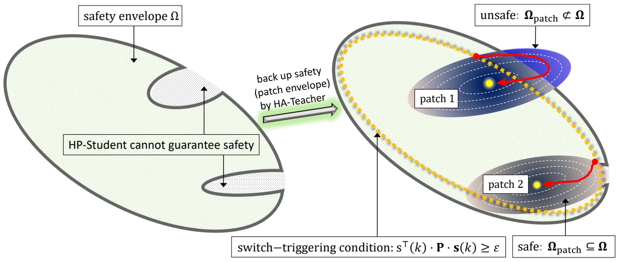

safety envelope Ω
HP-Student cannot guarantee safety
back up safety (patch envelope) by HA-Teacher
unsafe: $\mathbf{\Omega}_{\text{patch}} \not\subset \mathbf{\Omega}$
patch 1
patch 2
switch−triggering condition: $\mathrm{s}^\top(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) \geq \varepsilon$
safe: $\mathbf{\Omega}_{\text{patch}} \subseteq \mathbf{\Omega}$

Figure 3: System phase behavior.

- Constraining the states of a real plant to a very safe space (i.e., approaching the center of a safety envelope patch, illustrated in Figure 3) to preserve sufficient fault-tolerant space for HP-Student's continual learning.
- Collecting sufficient data of safe actions and states to correct the unsafe continual learning of HP-Student (see corrected $\tau \in \mathbb{N}$ unsafe actions in replay buffer in Figure 1).

The dwell time τ is a design of HA-Teacher, which is carried out in Equation (21) in Remark 6.3.

Remark 5.1 (**Parallel Running**)**.** *Finally, we note that the HA-Teacher and HP-Student in the SeC-learning machine run in parallel. This configuration guarantees that when the Coordinator activates the HA Teacher to back up safety and correct unsafe learning, his actions can be available immediately. If operating without parallel running, the system can lose control due to the time delay in sampling, communication, and computation.*

6 SeC-Learning Machine: HA-Teacher Component

HA-Teacher has tasks in the SeC-leaning machine:

- **Back up Safety via Patching Safety Envelope**. As soon as the HP-Student leads to an unsafe real-time system behavior, the HA-Teacher intervenes to safely control the system through patching the safety envelope, depicting in Figure 3.
- **Correct Unsafe Learning**. The HA-Teacher uses safe actions to correct the real-time and potentially (in dwell time τ horizon) unsafe actions of the HP-Student for learning.

HA-Teacher is a physics-model-based design whose function is reduced to be safety-critical only. Compared with the HP-Student, the HA-Teacher has relatively rich dynamics knowledge about real plants. Hereto, the dynamics model leveraged by HA-Teacher updates from (1) as

$$\mathbf{s}(k+1) = \mathbf{A}(\mathbf{s}(k)) \cdot \mathbf{s}(k) + \mathbf{B}(\mathbf{s}(k)) \cdot \mathbf{a}_{\text{HA}}(k) + \mathbf{g}(\mathbf{s}(k)),\ k \in \mathbb{N} \tag{11}$$

where $\mathbf{g}(\mathbf{s}(k)) \in \mathbb{R}^n$ is the <u>unknown</u> model mismatch for HA-Teacher. The physics-model knowledge available to HA-Teacher is thus $(\mathbf{A}(\mathbf{s}(k)), \mathbf{B}(\mathbf{s}(k)))$. Because of the **Parallel Running** configuration, HA-Teacher is always active to compute a safe action policy with control goal as

$$\text{Patch center}: \quad \overline{\mathbf{s}}^* = \chi \cdot \mathbf{s}(k) \quad \text{with} \quad \chi \in (-1, 1). \tag{12}$$

In other words, with real-time sensor data and physics-model knowledge, the HA-Teacher is always in 'active' status to compute a model-based action policy to control the real plant to reach the goal $\overline{\mathbf{s}}^*$. To achieve this, HA-Teacher first obtains tracking-error dynamics from Equation (11) as

$$\mathbf{e}(k+1) = \mathbf{A}(\mathbf{s}(k)) \cdot \mathbf{e}(k) + \mathbf{B}(\mathbf{s}(k)) \cdot \mathbf{a}_{\text{HA}}(k) + \mathbf{h}(\mathbf{e}(k)),\ \text{with}\ \ \mathbf{e}(k) \triangleq \mathbf{s}(k) - \overline{\mathbf{s}}^* \tag{13}$$

where $\mathbf{h}(\mathbf{e}(k)) \in \mathbb{R}^n$ denotes unknown model mismatch, and HA-Teacher's action policy is

$$\mathbf{a}_{\text{HA}}(k) = \widehat{\mathbf{F}} \cdot \mathbf{e}(k), \tag{14}$$

whose aim is to track the goal $\overline{\mathbf{s}}^*$ while constraining system states to the envelope patch:

$$\text{Envelope patch: } \Omega_{\text{patch}} \triangleq \left\{ \mathbf{s} \mid (\mathbf{s}-\overline{\mathbf{s}}^*)^\top \cdot \widehat{\mathbf{P}} \cdot (\mathbf{s}-\overline{\mathbf{s}}^*) \le (1-\chi)^2 \cdot \mathbf{s}^\top(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{s}(k),\ \widehat{\mathbf{P}} \succ 0 \right\}. \tag{15}$$

The matrices $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{P}}$ in Equation (14) and Equation (15) are HA-Teacher's design variables for backing up safety and correcting unsafe learning. To have them, we present a practical and common assumption on unknown model mismatch for computing them.

Assumption 6.1. *The model mismatch in $\mathbf{h}(\cdot)$ in Equation* (13) *is locally Lipschitz in set Ω_{patch}, i.e.,*

$$\left(\mathbf{h}\left(\mathbf{e}_1\right) - \mathbf{h}\left(\mathbf{e}_2\right)\right)^\top \cdot \mathbf{P} \cdot \left(\mathbf{h}\left(\mathbf{e}_1\right) - \mathbf{h}\left(\mathbf{e}_2\right)\right) \le \kappa \cdot \left(\mathbf{e}_1 - \mathbf{e}_2\right)^\top \cdot \mathbf{P} \cdot \left(\mathbf{e}_1 - \mathbf{e}_2\right),\ \forall \mathbf{e}_1, \mathbf{e}_2 \in \Omega_{patch},$$

where $\mathbf{P} \succ 0$ is shared by HP-Student, which defines his safety envelope (3) *and safety-embedded reward* (5)*.*

The designs of $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{P}}$ for delivering HA-Teacher's capabilities of backing up safety and correcting unsafe learning are formally presented in the following theorem, whose proof appears in Appendix C.

Theorem 6.2. *Consider the HA-Teacher's action policy* (14) *and the envelope patch Ω_{patch}* (15)*, whose matrices $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{P}}$ are computed according to*

$$\widehat{\mathbf{F}} = \widehat{\mathbf{R}} \cdot \widehat{\mathbf{Q}}^{-1}, \quad \widehat{\mathbf{P}} = \widehat{\mathbf{Q}}^{-1}, \tag{16}$$

with the matrices $\widehat{\mathbf{R}}$ and $\widehat{\mathbf{Q}}$ satisfying

$$\widehat{\mathbf{Q}} \cdot \mathbf{P} \prec \mathbf{I}_n \prec \eta \cdot \widehat{\mathbf{Q}} \cdot \mathbf{P}, \ \textit{with a given } \eta > 1 \tag{17}$$

$$\begin{bmatrix} (\beta - \kappa \cdot \eta \cdot (1 + \frac{1}{\omega})) \cdot \widehat{\mathbf{Q}} & \widehat{\mathbf{Q}} \cdot \mathbf{A}^\top(\mathbf{s}(k)) + \widehat{\mathbf{R}}^\top \cdot \mathbf{B}^\top(\mathbf{s}(k)) \\ \mathbf{A}(\mathbf{s}(k)) \cdot \widehat{\mathbf{Q}} + \mathbf{B}(\mathbf{s}(k)) \cdot \widehat{\mathbf{R}} & \frac{\widehat{\mathbf{Q}}}{1+\omega} \end{bmatrix} \succ 0, \tag{18}$$

where $\beta \in (0, 1)$ and $\omega > 0$ are given parameters. Under Assumption 6.1, the system (13) *controlled by HA-Teacher has the following properties:*

1. *The $\mathbf{e}^\top(k+1) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}(k+1) \le \beta \cdot \mathbf{e}^\top(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}(k)$ holds for any $k \in \mathbb{N}$.*
2. *The $\Omega_{patch} \subseteq \Omega$ holds if the parameters η in Equation* (17) *and χ in Equation* (12) *satisfy*

$$(1-\chi)^2 \cdot \eta \cdot \varepsilon + \chi^2 \cdot \varepsilon \le 0.5. \tag{19}$$

Remark 6.3 (**Suggestion from Item 1: Dwell time τ of HA-Teacher**)**.** *We obtain from Item 1 that*

$$\mathbf{e}^\top(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}(k) \le \beta^{k-t} \cdot (\mathbf{e}^*)^\top \cdot \widehat{\mathbf{P}} \cdot (\mathbf{e}^*), \tag{20}$$

wherein the t and $\mathbf{e}^*$ denote the activation time of HA-Teacher and the initial distance with goal $\overline{\mathbf{s}}^*$ (12)*, respectively. The real-time tracking error $\mathbf{e}(k)$ can be understood as the distance to the goal $\overline{\mathbf{s}}^*$ (i.e., the center of envelope patch). Meanwhile, we can use $\mathbf{e}^\top(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}(k)$ as distance function or performance metric. Hereto, we consider a safety criteria as $\mathbf{e}^\top(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}(k) \le \delta$. Illustrated in Figure 3, a very small δ means "being very close to patch center" and that HA-Teacher can preserve sufficient fault-tolerance space for HA-Teacher's continual learning. Meanwhile, when the preset safety criteria hold, HP-Student takes back the control role from HA-Teacher. According to Equation* (20)*, the condition of HA-Teacher's dwell time for satisfying the safety criteria is*

$$\tau \ge \left\lceil \frac{\ln \delta - \ln (\mathbf{e}^*)^\top \cdot \widehat{\mathbf{P}} \cdot (\mathbf{e}^*)}{\ln \beta} \right\rceil. \tag{21}$$

In other words, if $k - t \ge \tau$ and τ satisfies Equation (21)*, we have $\mathbf{e}^\top(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}(k) \le \delta$.*

Remark 6.4 (**Suggestion from Item 2: Backing up safety by envelope patches**). *Condition* (18) *is for backing up safety when the HP-Student's actions cannot guarantee real plants' real-time safety. As depicted in Figure 3, if only the property in Item 1 of Theorem B.1 holds, the HA-Teacher cannot back up safety due to the unsafe region, highlighted in blue color. Only after the Item 2 of Theorem B.1 holds can HA-Teacher achieve the concurrent missions of backing up safety and correcting learning, as the unsafe regions (i.e., regions outside the safety envelope) disappear.*

Remark 6.5 (**Fast Computation**). *The* $\widehat{\mathbf{F}}$ *and* $\widehat{\mathbf{P}}$ *are automatically computed from inequalities* (17) *and* (18) *by LMI toolbox [46, 47]. Its computation time is usually significantly small (e.g., 0.01 sec – 0.04 sec), and its influence can be ignored in the configuration of parallel running (see Remark 5.1).*

7 Experiment

We perform the experiments on a cart-pole system (simulator) and a real quadruped robot.

7.1 Cart-Pole System

This experiment aims to demonstrate the effectiveness of the SeC-Learning Machine from perspectives of concurrent safety and training performances in the face of the Sim2Real gap. The pre-training of HP-Student (i.e., Phy-DRL) is performed on the simulator provided in Open-AI Gym [48]. To address the Sim2Real gap, the pre-training adopts domain randomization [24, 32] through introducing random force disturbances and randomizing friction force. We also use the simulator to mimic a real plant whose Sim2Real gap is intentionally created by inducing a friction force that is out of the distribution of the random friction force used in pre-training.

The system's mechanical analog is characterized by the pendulum's angle θ, the cart's position x, and their velocities $\omega = \dot{\theta}$ and $v = \dot{x}$. The mission of HP-Student is to stabilize the pendulum at equilibrium $\mathbf{s}^* = [0, 0, 0, 0]^\top$ while constraining the system state to safety set:

$$\mathbb{X} = \left\{ \mathbf{s} \in \mathbb{R}^4 \middle| -0.9 \leq x \leq 0.9, \ -0.8 < \theta < 0.8 \right\}. \tag{22}$$

Given the safety set, we set the space of physics-model-based action policy as

$$\mathbb{A}_{\text{phy}} \triangleq \{ \mathbf{a}_{\text{phy}} \in \mathbb{R} \mid \ -25 \leq \mathbf{a}_{\text{phy}} \leq 25 \}. \tag{23}$$

With them, the designs of HP-Student and HA-Teacher are presented in Appendix D.3 and Appendix D.4, respectively.

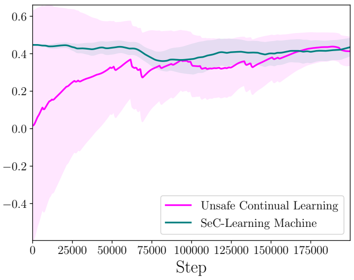
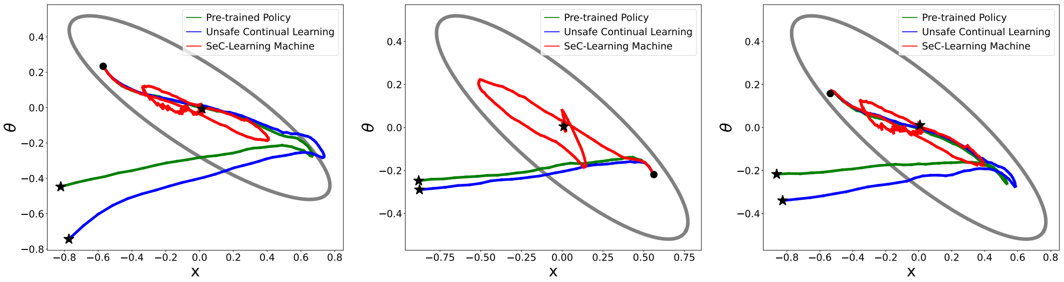


Figure 4: Episode reward.

For continual learning in mimicked real plants, the maximum length of one episode is 1500 steps. For the comparisons, we consider three models: *'Pre-trained Policy'* (i.e., Phy-DRL pre-trained in a simulator, using randomization approaches for addressing Sim2Real gap), *'Unsafe Continual Learning'* (i.e., pre-trained Phy-DRL continues to learn in the real plant but no Simplex support), and our proposed *'SeC-Learning Machine'*. A very distinguished feature of our SeC-Learning Machine is lifetime safety, i.e., a safety guarantee in any stage of continual learning, regardless of the success of HP-Student. To demonstrate this and have fair comparisons, HP-Student in the 'SeC-Learning Machine' and the 'Unsafe Continual Learning' model are picked after training for **only two episodes**. Given the three different initial conditions, phases plots of these three models are shown in Figure 5. To further convincingly demonstrate the feature, additional phase plots are shown in Figure 7, Figure 8, and Figure 9, where the models are picked after training for only **three episodes**, **four episodes**, **five episodes**, respectively. Meanwhile, the reward's training curves (five random seeds) are shown in Figure 4. Observing Figures 4, 5 and 7 to 9, we conclude:

- In the face of a large Sim2Real gap with a pre-training environment, the SeC-Learning Machine can always guarantee safety (system states never leave the safety envelope; see red curves in Figures 5 and 7 to 9) in any stage of continual learning. In contrast, the pre-trained model and continual learning without Simplex cannot guarantee safety (system states left the safety envelope; see blue and green curves in Figures 5 and 7 to 9).



(a) Initial Condition 1 (b) Initial Condition 2 (c) Initial Condition 3

Figure 5: **Two Episodes**. Phase plots, given the same initial condition. The black dot and star denote the initial condition and final location, respectively.

- Unsafe action correction and safety backup from HA-Teacher lead to remarkably stable and fast training, compared with continual learning without Simplex logic (see Figure 4).

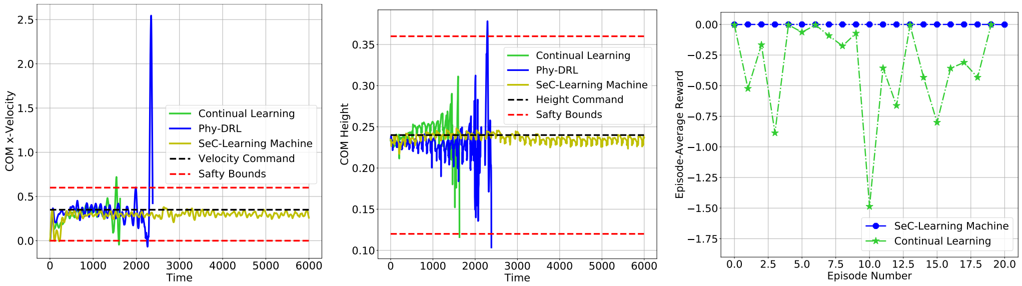
7.2 Real Quadruped Robot

The mission of the action policy is to control the robot's COM height, COM x-velocity, and other states to track the corresponding commands r_{v_x}, r_h, and zeros, under safety constraints: $|\text{yaw}| \le 0.2$ rad, $|\text{CoM x-velocity} - r_{v_x}| \le |r_{v_x}|$, $|\text{CoM z-height} - r_h| \le 0.12$ m, and $|\text{CoM yaw velocity}| \le 0.3$ rad/s. For the Coordinator's switching logic (9) and correcting logic (10), we let $\varepsilon = 0.65$ and $\tau = 10$. The designs of HP-Student and HA-Teacher are presented in Appendix E.3 and Appendix E.4, respectively. For HP-Student's pre-training of addressing the Sim2Real gap, we consider the approaches of delay randomization proposed in [28] and force randomization. During pre-taining in the simulator, the ground friction is set as 0.7, and r_{v_x} = 0.6 m/s and r_h = 0.24 m. In the real quadruped robot, one episode is defined as "running the robot for 15 sec." To better demonstrate the performance of the SeC-Learning Machine, the velocity command for the real robot is r_{v_x} = 0.35 m/s, which very different from the one for HP-Student's training in the simulator.

We compare three models in the real quadruped robot: *'Phy-DRL'* (a pre-trained Phy-DRL model in the simulator, directly deployed on the real robot), *'Continual Learning'* (a pre-trained Phy-DRL model in simulator that continues learning in the real robot for 20 episodes but without Simplex logic), and our proposed *'SeC-Learning Machine'* in the 1st episode in real robot. We consider the 1st episode for the 'SeC-Learning Machine' model, owning to its claimed feature of lifetime safety, i.e., safety guarantee in any stage of continual learning in a real plant regardless of HP-Student's convergence or success. Therefore, showing system trajectories in the 1st episode will be most convincing, as HP-Student cannot converge so fast for a safe action policy within only one episode. We also note that Phy-DRL or HP-Student is pre-trained well in the simulator.

The comparisons are first viewed from the trajectories of the robot's CoM height and x-velocity regarding tracking performance and safety guarantee, which are shown in Figure 6 (a) and (b). Meanwhile, the comparison video of the three models in real robots is available at real-robot-video-1 link. In addition, the real robot's trajectories under the control of the SeC-Learning Machine in the 5th episode, 10th episode, 15th episode, and 20th episode are shown in Figures 10 to 13 in Appendix E.5.1, respectively. These figures straightforwardly depict that the SeC-Learning Machine guarantees the safety of real robots across all selected episodes or stages of continual learning. Observing Figure 6 (a) and (b), Figures 10 to 13, and the comparison video, we discover that Phy-DRL (i.e., HP-Student), a well pre-trained model in simulator, cannot guarantee the safety of the real robot due to the existing Sim2Real gap or unknown unknowns in the physical environment that the delay and force randomization failed to capture. Thus, continuous real-time learning is needed for the real robot. However, without Simplex logic, the safety of real robot during continual learning is not guaranteed. The SeC-Learning Machine successfully addresses these challenges by ensuring lifetime safety for continual real-time learning.

Finally, we emphasize that another critical mission of HA-Teacher is correcting HP-Student's unsafe behavior. In other words, HP-Student learns from HA-Teacher to be safe when his actions are unsafe. To demonstrate this, we compare the episode-average reward curves of HP-Student (with HA-Teacher support) against the one (without Simplex logic or HA-Teacher support) in continual learning, as



(a) CoM x-Velocity

(b) CoM z-height

(c) Episode-average rewards

Figure 6: Comparisons of trajectories and episode-average reward.

depicted in Figure 6 (c). In addition, the reward curves in the iteration step for 20 episodes are shown in Figure 14 in Appendix E.5.2. Meanwhile, We also deactivate HA-Teacher during the 20th episode of HP-Student's continual learning to verify if HP-Student has quickly assimilated safe behaviors from HA-Teacher. The demonstration video for comparison with the initial HP-Student is available at real-robot-video-2 link. The demonstration video, Figure 6 (c) and Figure 14 prove that compared with Continual Learning, HA-Teacher enables stable, fast, and safe learning for HP-Student. Notably, the video shows that after only 20 episodes (i.e., 300 sec), the HP-Student has successfully learned from HA-Teacher to be safe.

8 Conclusion and Discussion

This paper develops the SeC-learning machine for safety-critical autonomous systems. The SeC-learning machine constitutes HP-Student (a pre-trained Phy-DRL, continuing to learn in a real plan), HA-Teacher (verified physics-model-based design), and Coordinator. In the learning machine, the HA-Teacher backs up safety and corrects unsafe learning of HP-Student. The SeC-learning machine aims to assure lifetime safety, address the Sim2Real gap, and learn to tolerate unknowns unknowns in real plants. Experiments also demonstrate that the SeC-learning machine features remarkably fast, stable, and safe training compared with continual learning without Simplex logic.

Intuitively, the SeC-learning machine is also an automatic hierarchy learning machine. Specifically, the HP-student first learns from the HA-teacher to be safe. If safe enough, i.e., the HP-student can be independent of the HA-teacher, the HP-student will learn by himself to achieve the goal of high performance with verifiable safety. This investigation continues our future research direction.

9 Computation Resources

In all case studies, we train and test the deep reinforcement learning (DRL) algorithm on a desktop equipped with Ubuntu 22.04, a 12th Gen Intel(R) Core(TM) i9-12900K 16-core processor, 64 GB of RAM, and an NVIDIA GeForce GTX 3090 GPU. The DRL algorithm was implemented in Python using the TensorFlow framework. We used the open-source Python CVX solver to solve LMIs problems.

In our architecture, the computation of $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{P}}$ for HA-Teacher at each patch need to be done in time when the Safety Coordinator is triggered. To enable real-time computation of CVX and interaction with the environment, we implement a multi-processing pipeline to control the robot and solve LMIs in parallel in real-time. For solving LMIs, we always let the solver take the latest state so that whenever the safety coordinator is triggered, the latest $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{P}}$ will always be ready, where the delay issue was considered and formulated in the LMIs problems.

We also noticed that the MATLAB-based CVX solver could solve the LMIs problem more consistently than the Python-based one, yielding more reliable solutions. However, the data interfacing overhead between Matlab and Python will introduce extra delay when updating $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{P}}$ for HA-Teacher. Besides, the multiprocessing implementation for Matlab and Python is another technical challenge due to software compatibility issues. Therefore, we took the Python-based CVX solver for real-time real-world experiments, and we suggest that the Matlab-based solver is preferable for the less real-time sensitive applications.

References

[1] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 International Conference on Robotics and Automation*, pages 8248–8254. IEEE, 2019.

[2] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

[3] Thomas Savage, Dongda Zhang, Max Mowbray, and Ehecatl Antonio Del Río Chanona. Model-free safe reinforcement learning for chemical processes using gaussian processes. *IFAC-PapersOnLine*, 54(3):504–509, 2021.

[4] Zhenglei He, Kim-Phuc Tran, Sebastien Thomassey, Xianyi Zeng, Jie Xu, and Changhai Yi. A deep reinforcement learning based multi-criteria decision support system for optimizing textile chemical process. *Computers in Industry*, 125:103373, 2021.

[5] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

[6] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[7] Tian Tolentino. Autonomous aircraft market worth usd 23.7bn by 2030. `https://www.traveldailymedia.com/autonomous-aircraft-market-research/`, 2019.

[8] Bernard Marr. How Tesla is using artificial intelligence to create the autonomous cars of the future. *Bernard Marr & Co.* `https://bernardmarr.com/how-tesla-is-using-artificial-intelligence-to-create-the-autonomous-cars-of-the-future/`, 2021.

[9] AI incident database. `https://incidentdatabase.ai/entities/`.

[10] Arnold Zachary and Toner Helen. AI Accidents: An emerging threat. *Center for Security and Emerging Technology*, 2021.

[11] Theodore J Perkins and Andrew G Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(Dec):803–832, 2002.

[12] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems*, 30, 2017.

[13] Ya-Chien Chang and Sicun Gao. Stabilizing neural control using self-learned almost Lyapunov critics. In *2021 IEEE International Conference on Robotics and Automation*, pages 1803–1809. IEEE, 2021.

[14] Liqun Zhao, Konstantinos Gatsis, and Antonis Papachristodoulou. Stable and safe reinforcement learning via a Barrier-Lyapunov actor-critic approach. In *62nd IEEE Conference on Decision and Control*, pages 1320–1325. IEEE, 2023.

[15] Tyler Westenbroek, Fernando Castaneda, Ayush Agrawal, Shankar Sastry, and Koushil Sreenath. Lyapunov design for robust and efficient robotic reinforcement learning. *arXiv:2208.06721*, 2022.

[16] Krishan Rana, Vibhavari Dasagi, Jesse Haviland, Ben Talbot, Michael Milford, and Niko Sünderhauf. Bayesian controller fusion: Leveraging control priors in deep reinforcement learning for robotics. *arXiv preprint* `https://arxiv.org/pdf/2107.09822.pdf`.

[17] Tongxin Li, Ruixiao Yang, Guannan Qu, Yiheng Lin, Steven Low, and Adam Wierman. Equipping black-box policies with model-based advice for stable nonlinear control. *arXiv preprint* `https://arxiv.org/pdf/2206.01341.pdf`.

[18] Richard Cheng, Abhinav Verma, Gabor Orosz, Swarat Chaudhuri, Yisong Yue, and Joel Burdick. Control regularization for reduced variance reinforcement learning. In *International Conference on Machine Learning*, pages 1141–1150, 2019.

[19] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6023–6029. IEEE, 2019.

[20] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel 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*, volume 33, pages 3387–3395, 2019.

[21] Hongpeng Cao, Yanbing Mao, Lui Sha, and Marco Caccamo. Physics-regulated deep reinforcement learning: Invariant embeddings. In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Hongpeng Cao, Yanbing Mao, Lui Sha, and Marco Caccamo. Physics-model-regulated deep reinforcement learning towards safety & stability guarantees. In *62nd IEEE Conference on Decision and Control*, pages 8300–8305, 2023.

[23] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018.

[24] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning, 2019.

[25] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *Robotics: Science and Systems*, 2018.

[26] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *Robotics: Science and Systems*, 2017.

[27] Hongpeng Cao, Mirco Theile, Federico G. Wyrwal, and Marco Caccamo. Cloud-edge training architecture for sim-to-real deep reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9363–9370, 2022.

[28] Chieko Sarah Imai, Minghao Zhang, Yuchen Zhang, Marcin Kierebiński, Ruihan Yang, Yuzhe Qin, and Xiaolong Wang. Vision-guided quadrupedal locomotion in the wild with multi-modal delay randomization. In *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5556–5563. IEEE, 2022.

[29] Yuqing Du, Olivia Watkins, Trevor Darrell, Pieter Abbeel, and Deepak Pathak. Auto-tuned sim-to-real transfer, 2021.

[30] Quan Vuong, Sharad Vikram, Hao Su, Sicun Gao, and Henrik I. Christensen. How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies?, 2019.

[31] Tsung-Yen Yang, Tingnan Zhang, Linda Luu, Sehoon Ha, Jie Tan, and Wenhao Yu. Safe reinforcement learning for legged locomotion. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2454–2461. IEEE, 2022.

[32] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image, 2017.

[33] Thomas Bartz-Beielstein. Why we need an AI-resilient society. *arXiv:1912.08786*, 2019.

[34] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[35] Jan Roskam. *Airplane flight dynamics and automatic flight controls*. DARcorporation, 1995.

[36] Lui Sha et al. Using simplicity to control complexity. *IEEE Software*, 18(4):20–28, 2001.

[37] Stanley Bak, Taylor T Johnson, Marco Caccamo, and Lui Sha. Real-time reachability for verified Simplex design. In *2014 IEEE Real-Time Systems Symposium*, pages 138–148. IEEE, 2014.

[38] Yanbing Mao, Yuliang Gu, Naira Hovakimyan, Lui Sha, and Petros Voulgaris. S$\mathcal{L}_1$-simplex: Safe velocity regulation of self-driving vehicles in dynamic and unforeseen environments. *ACM Transactions on Cyber-Physical Systems*, 7(1):1–24, 2023.

[39] Ruihan Yang, Minghao Zhang, Nicklas Hansen, Huazhe Xu, and Xiaolong Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. *2022 International Conference on Learning Representations*, 2022.

[40] Siddhant Gangapurwala, Alexander Mitchell, and Ioannis Havoutis. Guided constrained policy optimization for dynamic quadrupedal robot locomotion. *IEEE Robotics and Automation Letters*, 5(2):3642–3649, 2020.

[41] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR*, 2016.

[42] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018.

[43] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[44] Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

[45] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[46] Pascal Gahinet, Arkadii Nemirovskii, Alan J Laub, and Mahmoud Chilali. The lmi control toolbox. In *Proceedings of 1994 33rd IEEE conference on decision and control*, volume 3, pages 2038–2041. IEEE, 1994.

[47] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[48] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016.

[49] Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

[50] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint* `https://arxiv.org/abs/1707.06347`.

[51] Răzvan Florian. Correct equations for the dynamics of the cart-pole system. 08 2005.

[52] Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx users' guide. *online: http://www. stanford. edu/boyd/software. html*, 2009.

[53] Lieven Vandenberghe, Stephen Boyd, and Shao-Po Wu. Determinant maximization with linear matrix inequality constraints. *SIAM journal on matrix analysis and applications*, 19(2):499–533, 1998.

[54] Yuxiang Yang. Github: Quadruped robot simulator.

[55] Jared Di Carlo, Patrick M Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1–9. IEEE, 2018.

Appendices

A Auxiliary Lemmas

Lemma A.1 (Schur Complement [49])**.** *For any symmetric matrix* $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{C} \end{bmatrix}$*, then* $\mathbf{M} \succ 0$ *if and only if* $\mathbf{C} \succ 0$ *and* $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^\top \succ 0$*.*

Lemma A.2. *[21] Consider the sets defined in Equation* (2) *and Equation* (3)*. We have* $\Omega \subseteq \mathbb{X}$*, if*

$$[\underline{\mathbf{D}}]_{i,:} \cdot \mathbf{P}^{-1} \cdot \left[\underline{\mathbf{D}}^\top\right]_{:,i} = \begin{cases} \geq 1, & [\mathbf{d}]_i = 1 \\ \leq 1, & [\mathbf{d}]_i = -1 \end{cases}, \ \left[\overline{\mathbf{D}}\right]_{i,:} \cdot \mathbf{P}^{-1} \cdot \left[\overline{\mathbf{D}}^\top\right]_{:,i} \leq 1, \ i \in \{1, \ldots, h\}, \tag{24}$$

where $\overline{\mathbf{D}} = \frac{\mathbf{D}}{\overline{\Lambda}}$, $\underline{\mathbf{D}} = \frac{\mathbf{D}}{\underline{\Lambda}}$*, and for* $i, j \in \{1, \ldots, h\}$*,*

$$[\mathbf{d}]_i \triangleq \begin{cases} 1, & [\underline{\mathbf{v}}+\mathbf{v}]_i > 0 \\ 1, & [\overline{\mathbf{v}}+\mathbf{v}]_i < 0 \\ -1, & \text{otherwise} \end{cases}, \ [\overline{\Lambda}]_{i,j} \triangleq \begin{cases} 0, & i \neq j \\ [\overline{\mathbf{v}}+\mathbf{v}]_i\,, & [\underline{\mathbf{v}}+\mathbf{v}]_i > 0 \\ [\underline{\mathbf{v}}+\mathbf{v}]_i\,, & [\overline{\mathbf{v}}+\mathbf{v}]_i < 0 \\ [\overline{\mathbf{v}}+\mathbf{v}]_i\,, & \text{otherwise} \end{cases}, \ [\underline{\Lambda}]_{i,j} \triangleq \begin{cases} 0, & i \neq j \\ [\underline{\mathbf{v}}+\mathbf{v}]_i\,, & [\underline{\mathbf{v}}+\mathbf{v}]_i > 0 \\ [\overline{\mathbf{v}}+\mathbf{v}]_i\,, & [\overline{\mathbf{v}}+\mathbf{v}]_i < 0 \\ [-\underline{\mathbf{v}} - \mathbf{v}]_i\,, & \text{otherwise} \end{cases}.$$

Lemma A.3. *Consider the action set* $\mathbb{A}_{phy}$ *defined in Equation* (7)*, and*

$$\Phi \triangleq \left\{\mathbf{a} \in \mathbb{R}^m \mid \mathbf{a}^\top \cdot \mathbf{V} \cdot \mathbf{a} \leq 1, \ \mathbf{V} \succ 0\right\}. \tag{25}$$

We have $\Phi \subseteq \mathbb{A}_{phy}$*, if*

$$\left[\overline{\mathbf{C}}\right]_{i,:} \cdot \mathbf{V}^{-1} \cdot \left[\overline{\mathbf{C}}^\top\right]_{:,i} \leq 1, \ \ [\underline{\mathbf{C}}]_{i,:} \cdot \mathbf{V}^{-1} \cdot \left[\underline{\mathbf{C}}^\top\right]_{:,i} = \begin{cases} \geq 1, & \text{if } [\mathbf{c}]_i = 1 \\ \leq 1, & \text{if } [\mathbf{c}]_i = -1 \end{cases}, \ i \in \{1, \ldots, g\} \tag{26}$$

where $\overline{\mathbf{C}} = \frac{\mathbf{C}}{\overline{\Delta}}$, $\underline{\mathbf{C}} = \frac{\mathbf{C}}{\underline{\Delta}}$*, and for* $i, j \in \{1, \ldots, g\}$*,*

$$[\mathbf{c}]_i \triangleq \begin{cases} 1, & [\underline{\mathbf{z}} + \mathbf{z}]_i > 0 \\ 1, & [\overline{\mathbf{z}} + \mathbf{z}]_i < 0 \\ -1, & \text{otherwise} \end{cases}, \ [\overline{\Delta}]_{i,j} \triangleq \begin{cases} 0, & i \neq j \\ [\overline{\mathbf{z}} + \mathbf{z}]_i\,, & [\underline{\mathbf{z}}+\mathbf{z}]_i > 0 \\ [\underline{\mathbf{z}}+\mathbf{z}]_i\,, & [\overline{\mathbf{z}}+\mathbf{z}]_i < 0 \\ [\overline{\mathbf{z}} + \mathbf{z}_\sigma]_i\,, & \text{otherwise} \end{cases}, \ [\underline{\Delta}]_{i,j} \triangleq \begin{cases} 0, & i \neq j \\ [\underline{\mathbf{z}}+\mathbf{z}]_i\,, & [\underline{\mathbf{z}}+\mathbf{z}]_i > 0 \\ [\overline{\mathbf{z}}+\mathbf{z}]_i\,, & [\overline{\mathbf{z}}+\mathbf{z}]_i < 0 \\ [-\underline{\mathbf{z}}-\mathbf{z}]_i\,, & \text{otherwise} \end{cases}.$$

Proof. The proof path of Lemma A.3 is the same as that of Lemma A.2 in [21]. So, it is omitted here. □

B Control Contribution Ratio

B.1 Controllable Data-driven Action Space

In continuous control tasks, actions of the data-driven approaches are typically generated by policies parameterized with deep neural networks, which map states to actions. In DRL, deterministic policies, as described in DDPG [41], directly map states to actions within the interval $[-1, 1]$ using the Tanh activation function at the output layer. On the other hand, stochastic policies, such as those proposed in the Soft Actor-Critic (SAC)[42] algorithm and Proximal Policy Optimization (PPO) [50], sample actions from a learned Gaussian distribution with reparameterization trick, followed by Tanh activation to ensure actions remain within the $[-1, 1]$ interval. To adapt the generated action for actual control commands, the action is rescaled by a magnitude factor M_l, expanding the action space to $[-M_l, M_l]$. This allows for the flexible adaptation of the action space to the specific requirements of different control tasks.

B.2 Controllable Physics-model-based Action Space

The controllable physics-model-based action space is delivered by incorporating two additional conditions (26) and (28) into Theorem 5.3's LMIs in [21]. The updated version is formally stated in the following theorem.

Theorem B.1. *Consider the system* (1) *under control of Phy-DRL, consisting of residual policy* (4) *and safety-embedded reward* (5)*, wherein matrices* $\mathbf{F}$ *and* $\mathbf{P}$ *are computed according to*

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{Q}^{-1}, \quad \mathbf{P} = \mathbf{Q}^{-1}, \tag{27}$$

and $\mathbf{R}$ *and* $\mathbf{Q}^{-1}$ *satisfying the inequalities* (24)*,* (26) *with* $\mathbf{V} = \beta \cdot \mathbf{I}_m \succ 0$*, and*

$$\begin{bmatrix} \mathbf{Q} & \mathbf{R}^\top \\ \mathbf{R} & \frac{1}{\beta} \cdot \mathbf{I}_m \end{bmatrix} \succ 0, \tag{28}$$

$$\begin{bmatrix} \alpha \cdot \mathbf{Q} & \mathbf{Q} \cdot \mathbf{A}^\top + \mathbf{R}^\top \cdot \mathbf{B}^\top \\ \mathbf{A} \cdot \mathbf{Q} + \mathbf{B} \cdot \mathbf{R} & \mathbf{Q} \end{bmatrix} \succ 0, \quad \textit{with a given } \alpha \in (0, 1). \tag{29}$$

With the sets Ω (3)*,* $\mathbb{X}$ (2)*,* $\mathbb{A}$ (7)*, and* Φ (25) *at hand, the system* (1) *has the following properties:*

1. *Given any* $\mathbf{s}(1) \in \Omega$*, the system state* $\mathbf{s}(k) \in \Omega \subseteq \mathbb{X}$ *holds for any* $k \in \mathbb{N}$ *(i.e., the safety of system (1) is guaranteed), if the sub-reward* $r(\mathbf{s}(k), \mathbf{s}(k+1))$ *in Equation* (5) *satisfies* $r(\mathbf{s}(k), \mathbf{s}(k+1)) \geq \alpha - 1$, $\forall k \in \mathbb{N}$.
2. *If the state* $\mathbf{s}(k) \in \Omega \subseteq \mathbb{X}$*, the physics-model-based control command in Equation* (4) *satisfies* $\mathbf{a}_{phy}(k) \in \Phi \subseteq \mathbb{A}_{phy}$.

Proof. Item 1 is a direct result of Theorem 5.3 in [21], which is not influenced by the additional conditions (28) and Equation (26). So, its proof is omitted.

We now focus on the proof of Item 2 of Theorem B.1. Since $\beta > 0$ and $\mathbf{V} = \beta \cdot \mathbf{I}_m$, according to Lemma A.1, the condition (28) implies that

$$\mathbf{Q} - \beta \cdot \mathbf{R}^\top \cdot \mathbf{R} = \mathbf{Q} - \mathbf{R}^\top \cdot \mathbf{V} \cdot \mathbf{R} \succ 0. \tag{30}$$

Substituting $\mathbf{F} \cdot \mathbf{Q} = \mathbf{R}_i$ into (30) leads to $\mathbf{Q} - (\mathbf{F} \cdot \mathbf{Q})^\top \cdot \mathbf{V} \cdot (\mathbf{F} \cdot \mathbf{Q}) \succ 0$, multiplying both left-hand and right-hand sides of which by $\mathbf{Q}^{-1}$ yields $\mathbf{Q}^{-1} - \mathbf{F}^\top \cdot \mathbf{V} \cdot \mathbf{F} \succ 0$. We thus have

$$\mathbf{s}^\top(k) \cdot \mathbf{Q}^{-1} \cdot \mathbf{s}(k) - \mathbf{s}^\top(k) \cdot \mathbf{F}^\top \cdot \mathbf{V} \cdot \mathbf{F} \cdot \mathbf{s}(k) = \mathbf{s}^\top(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) - \mathbf{a}_{\text{phy}}^\top(k) \cdot \mathbf{V} \cdot \mathbf{a}_{\text{phy}}(k) > 0,$$

which is obtained via considering $\mathbf{P} = \mathbf{Q}^{-1}$ and $\mathbf{F} = \mathbf{R} \cdot \mathbf{Q}^{-1} = \mathbf{R} \cdot \mathbf{P}$. The inequality means $\mathbf{s}^\top(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) > \mathbf{a}_{\text{phy}}^\top(k) \cdot \mathbf{V} \cdot \mathbf{a}_{\text{phy}}(k)$. Therefore, in light of the definition of safety envelope (3), we conclude that if $s(k) \in \Omega$, i.e., $\mathbf{s}^\top(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) < 1$, then $\mathbf{a}_{\text{phy}}^\top(k) \cdot \mathbf{V} \cdot \mathbf{a}_{\text{phy}}(k) < 1$. Furthermore, considering (26) in Lemma A.3 in conjunction with defined set (25), we conclude $\mathbf{a}_{\text{phy}}(k) \in \Phi \subseteq \mathbb{A}$, which completes the proof. □

B.3 Automatic Constructions of Residual Action Policy and Safety-embedded Reward

Given the matrices $\mathbf{F}$ and $\mathbf{P}$, the residual policy Equation (4) and safety-embedded reward Equation (5) are delivered immediately. With the available physics-model knowledge $(\mathbf{A}, \mathbf{B})$ at hand, the $\mathbf{F}$ and $\mathbf{P}$ can be automatically computed from LMIs in Equation (24), Equation (26), Equation (28), and Equation (29), by LMI toolbox [46, 47].

We can also find optimal $\mathbf{R}$ and $\mathbf{Q}$ that can maximize the safety envelope. To achieve this, we recall the volume of a safety envelope (3) is proportional to $\sqrt{\det\left(\mathbf{P}^{-1}\right)}$, the interested problem is thus a typical analytic centering problem, formulated as given a $\alpha \in (0, 1)$,

$$\underset{\mathbf{Q},\,\mathbf{R}}{\text{argmin}} \left\{\log\det\left(\mathbf{Q}^{-1}\right)\right\} = \underset{\mathbf{Q},\,\mathbf{R}}{\text{argmin}} \left\{\log\det\left(\mathbf{P}^{-1}\right)\right\}, \text{ subject to LMIs (24), (26), (28), (29).}$$

C Proof Theorem B.1

Proof of Item 1

We define a Lyapunov candidate for a real plant described in Equation (13):

$$V\left(k\right) = \mathbf{e}^{\top}\left(k\right) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}\left(k\right), \tag{31}$$

which along the dynamics (13), in conjunction with the action policy (14), results in

$$\begin{aligned} & V\left(k+1\right) - \beta \cdot V\left(k\right) \\ & = \mathbf{e}^{\top}\left(k+1\right) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}\left(k+1\right) - \beta \cdot \mathbf{e}^{\top}\left(k\right) \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}\left(k\right) \\ & = \mathbf{e}^{\top}\left(k\right) \cdot \left(\overline{\mathbf{A}}^{\top}(\mathbf{s}(k)) \cdot \widehat{\mathbf{P}} \cdot \overline{\mathbf{A}}(\mathbf{s}(k)) - \beta \cdot \widehat{\mathbf{P}}\right) \cdot \mathbf{e}\left(k\right) + \mathbf{h}^{\top}\left(\mathbf{e}\left(k\right)\right) \cdot \widehat{\mathbf{P}} \cdot \mathbf{h}\left(\mathbf{e}\left(k\right)\right) \\ & \qquad + 2\mathbf{e}^{\top}\left(k\right) \cdot \left(\overline{\mathbf{A}}(\mathbf{s}(k)) \cdot \widehat{\mathbf{P}}\right) \cdot \mathbf{h}\left(\mathbf{e}\left(k\right)\right), \end{aligned} \tag{32}$$

where we define:

$$\overline{\mathbf{A}}(\mathbf{s}(k)) \triangleq \mathbf{A}(\mathbf{s}(k)) + \mathbf{B}(\mathbf{s}(k)) \cdot \widehat{\mathbf{F}}. \tag{33}$$

It is straightforward to verify the inequality

$$\begin{aligned} & 2\mathbf{e}^{\top}\left(k\right) \cdot \left(\overline{\mathbf{A}}(\mathbf{s}(k)) \cdot \widehat{\mathbf{P}}\right) \cdot \mathbf{h}\left(\mathbf{e}\left(k\right)\right) \\ & \leq \omega \cdot \mathbf{e}^{\top}\left(k\right) \cdot \overline{\mathbf{A}}^{\top}(\mathbf{s}(k)) \cdot \widehat{\mathbf{P}} \cdot \overline{\mathbf{A}}(\mathbf{s}(k)) \cdot \mathbf{e}\left(k\right) + \frac{1}{\omega} \cdot \mathbf{h}^{\top}\left(\mathbf{e}\left(k\right)\right) \cdot \widehat{\mathbf{P}} \cdot \mathbf{h}\left(\mathbf{e}\left(k\right)\right), \end{aligned} \tag{34}$$

holds for any $\omega > 0$.

Meanwhile, noting $\widehat{\mathbf{Q}} = \widehat{\mathbf{P}}^{-1}$, the inequality in Equation (17) is equivalent to $\eta \cdot \mathbf{P} \succ \widehat{\mathbf{P}} \succ \mathbf{P}$, which in light of Assumption 6.1 then results in

$$\begin{aligned} \mathbf{h}^{\top}\left(\mathbf{e}\left(k\right)\right) \cdot \widehat{\mathbf{P}} \cdot \mathbf{h}\left(\mathbf{e}\left(k\right)\right) \leq \mathbf{h}^{\top}\left(\mathbf{e}\left(k\right)\right) \cdot \eta \cdot \mathbf{P} \cdot \mathbf{h}\left(\mathbf{e}\left(k\right)\right) & \leq \mathbf{e}^{\top}\left(k\right) \cdot \kappa \cdot \eta \cdot \mathbf{P} \cdot \mathbf{e}\left(k\right) \\ & \leq \mathbf{e}^{\top}\left(k\right) \cdot \kappa \cdot \eta \cdot \widehat{\mathbf{P}} \cdot \mathbf{e}\left(k\right). \end{aligned} \tag{35}$$

Substituting inequalities in Equation (34) and Equation (35) into Equation (32) yields

$$\begin{aligned} & V\left(k+1\right) - \beta \cdot V\left(k\right) \\ & \leq \mathbf{e}^{\top}\left(k\right) \cdot \left((1+\omega) \cdot \overline{\mathbf{A}}^{\top}(\mathbf{s}(k)) \cdot \widehat{\mathbf{P}} \cdot \overline{\mathbf{A}}(\mathbf{s}(k)) - (\beta - \kappa \cdot \eta \cdot (1 + \frac{1}{\omega})) \cdot \widehat{\mathbf{P}}\right) \cdot \mathbf{e}\left(k\right). \end{aligned} \tag{36}$$

Recalling Schur Complement in Lemma A.1 in Appendix A and considering $\widehat{\mathbf{P}} \succ 0$, we conclude that the inequality in Equation (18) is equivalent to

$$\begin{aligned} & (\beta - \kappa \cdot \eta \cdot (1 + \frac{1}{\omega})) \cdot \widehat{\mathbf{Q}} \\ & \quad - (1+\omega) \cdot (\widehat{\mathbf{Q}} \cdot \mathbf{A}^{\top}(\mathbf{s}(k)) + \widehat{\mathbf{R}}^{\top} \cdot \mathbf{B}^{\top}(\mathbf{s}(k))) \cdot \widehat{\mathbf{Q}}^{-1} \cdot (\mathbf{A}(\mathbf{s}(k)) \cdot \widehat{\mathbf{Q}} + \mathbf{B}(\mathbf{s}(k)) \cdot \widehat{\mathbf{R}}) \succ 0. \end{aligned} \tag{37}$$

Multiplying both the left-hand side and the right-hand side of inequality (37) by $\widehat{\mathbf{Q}}^{-1}$ yields

$$\begin{aligned} & (\beta - \kappa \cdot \eta \cdot (1 + \frac{1}{\omega})) \cdot \widehat{\mathbf{Q}}^{-1} \\ & \quad - (1+\omega) \cdot (\mathbf{A}^{\top}(\mathbf{s}(k)) + \widehat{\mathbf{Q}}^{-1} \cdot \widehat{\mathbf{R}}^{\top} \cdot \mathbf{B}^{\top}(\mathbf{s}(k))) \cdot \widehat{\mathbf{Q}}^{-1} \cdot (\mathbf{A}(\mathbf{s}(k)) + \mathbf{B}(\mathbf{s}(k)) \cdot \widehat{\mathbf{R}} \cdot \mathbf{Q}^{-1}) \succ 0, \end{aligned}$$

substituting the definitions in Equation (16) into which leads to

$$\begin{aligned} & (\beta - \kappa \cdot \eta \cdot (1 + \frac{1}{\omega})) \cdot \widehat{\mathbf{P}} \\ & \qquad - (1+\omega) \cdot (\mathbf{A}^{\top}(\mathbf{s}(k)) + \widehat{\mathbf{F}}^{\top} \cdot \mathbf{B}^{\top}(\mathbf{s}(k))) \cdot \widehat{\mathbf{P}} \cdot (\mathbf{A}(\mathbf{s}(k)) + \mathbf{B}(\mathbf{s}(k)) \cdot \widehat{\mathbf{F}}) \succ 0. \end{aligned} \tag{38}$$

Recalling (33), the inequality in Equation (38) is equivalent to

$$(1+\omega) \cdot \overline{\mathbf{A}}^{\top}(\mathbf{s}(k)) \cdot \widehat{\mathbf{P}} \cdot \overline{\mathbf{A}}(\mathbf{s}(k)) - (\beta - \kappa \cdot \eta \cdot (1 + \frac{1}{\omega})) \cdot \widehat{\mathbf{P}} \prec 0,$$

which, in conjunction with Equation (36), leads to $V\left(k+1\right) - \beta \cdot V\left(k\right) \leq 0$, i.e., $V\left(k+1\right) \leq \beta \cdot V\left(k\right)$, we thus complete the proof of Item 1.

Proof of Item 2

For patch envelope (15), we introduce its boundary:

$$\partial\Omega_{\text{patch}} \triangleq \left\{ \mathbf{x} \mid (\mathbf{x} - \overline{\mathbf{s}}^{*})^{\top} \cdot \widehat{\mathbf{P}} \cdot (\mathbf{x} - \overline{\mathbf{s}}^{*}) = (1-\chi)^{2} \cdot \mathbf{s}^{\top}(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{s}(k),\ \widehat{\mathbf{P}} \succ 0 \right\},$$

which, in light of $\mathbf{e} \triangleq \mathbf{x} - \overline{\mathbf{s}}^{*}$, is equivalent to

$$\partial\Omega_{\text{patch}} \triangleq \left\{ \mathbf{x} \mid \mathbf{e}^{\top} \cdot \widehat{\mathbf{P}} \cdot \mathbf{e} = (1-\chi)^{2} \cdot \mathbf{s}^{\top}(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{s}(k) \right\}, \tag{39}$$

by which, the proof of $\Omega_{\text{patch}} \subseteq \Omega$ equivalently transforms to the proof of $\partial\Omega_{\text{patch}} \subseteq \Omega$.

To move forward, we first recall $\widehat{\mathbf{P}} = \widehat{\mathbf{Q}}^{-1} \succ 0$, in light of which, the inequality in Equation (17) is equivalent to

$$\mathbf{P} \prec \widehat{\mathbf{P}} \prec \eta \cdot \mathbf{P}, \text{ with } \eta > 1. \tag{40}$$

Meanwhile, recalling $\mathbf{e} = \mathbf{x} - \chi \cdot \mathbf{s}^{*}$, and we define

$$\mathbf{e}^{*} = \mathbf{x}^{*} - \chi \cdot \mathbf{s}(k) = \mathbf{s}(k) - \chi \cdot \mathbf{s}(k) = (1-\chi) \cdot \mathbf{s}(k), \tag{41}$$

considering which, for any $\mathbf{x} \in \partial\Omega_{\text{patch}}$, it is straightforward to verify that

$$\begin{aligned}
\mathbf{x}^{\top} \cdot \mathbf{P} \cdot \mathbf{x} &= (\mathbf{e} + \chi \cdot \mathbf{s}(k))^{\top} \cdot \mathbf{P} \cdot (\mathbf{e} + \chi \cdot \mathbf{s}(k)) \\
&= \mathbf{e}^{\top} \cdot \mathbf{P} \cdot \mathbf{e} + \chi^{2} \cdot \mathbf{s}^{\top}(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) + 2\chi \cdot \mathbf{e}^{\top} \cdot \mathbf{P} \cdot \mathbf{s}(k) \\
&\leq 2 \cdot \mathbf{e}^{\top} \cdot \mathbf{P} \cdot \mathbf{e} + 2\chi^{2} \cdot \mathbf{s}^{\top}(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) && (42)\\
&\leq 2 \cdot \mathbf{e}^{\top} \cdot \mathbf{P} \cdot \mathbf{e} + 2\chi^{2} \cdot \varepsilon && (43)\\
&\leq 2 \cdot \mathbf{e}^{\top} \cdot \widehat{\mathbf{P}} \cdot \mathbf{e} + 2\chi^{2} \cdot \varepsilon && (44)\\
&= 2 \cdot (1-\chi)^{2} \cdot \mathbf{s}^{\top}(k) \cdot \widehat{\mathbf{P}} \cdot \mathbf{s}(k) + 2\chi^{2} \cdot \varepsilon && (45)\\
&= 2 \cdot \eta \cdot (1-\chi)^{2} \cdot \mathbf{s}^{\top}(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) + 2\chi^{2} \cdot \varepsilon && (46)\\
&= 2 \cdot \eta \cdot (1-\chi)^{2} \cdot \varepsilon + 2\chi^{2} \cdot \varepsilon && (47)
\end{aligned}$$

We note Equation (42) is obtained from its previous step via considering the well-known inequality: $2\chi \cdot \mathbf{e}^{\top} \cdot \mathbf{P} \cdot \mathbf{s}(k) \leq \mathbf{s}^{\top}(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) + \chi^{2} \cdot \mathbf{e}^{\top} \cdot \mathbf{P} \cdot \mathbf{e}$. We let $\mathbf{x}^{*} = \mathbf{s}(k)$ be a state sample of triggering condition for guaranteeing $\mathbf{s}^{\top}(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) \leq \varepsilon$, which is the root that derives Equation (43) from Equation (42). Equation (44) from Equation (43) is obtained via considering the left-hand side inequality in Equation (40). Equation (45) from Equation (44) is obtained in light of Equation (39) and the fact $\mathbf{s} \in \partial\Omega_{\text{patch}}$. Equation (46) from Equation (45) is obtained via considering right-hand side inequality in Equation (40). Lastly, Equation (47) is obtained from Equation (46) by considering the fact that the $\mathbf{x}^{*} = \mathbf{s}(k)$ is a state sample inside safety envelope, so satisfying $\mathbf{s}^{\top}(k) \cdot \mathbf{P} \cdot \mathbf{s}(k) \leq 1$.

Recall that the inequality in Equation (19) is equivalent to $2\eta \cdot (1-\chi)^{2} \cdot \varepsilon + 2\chi^{2} \cdot \varepsilon \leq 1$, which, in conjunction with Equation (47), leads to the conclusion that for any $\mathbf{x} \in \partial\Omega_{\text{patch}}$, $\mathbf{x}^{\top} \cdot \widehat{\mathbf{P}} \cdot \mathbf{x} \leq 1$ holds. In other words, in light of Equation (3), for any $\mathbf{x} \in \partial\Omega_{\text{patch}}$, the $\mathbf{x} \in \Omega$ holds. We thus conclude $\partial\Omega_{\text{patch}} \subseteq \Omega$, which completes the proof.

D Experiment: Cart-Pole System

D.1 Pre-training and Contiunal Learning

We leverage the DDPG algorithm [41] to pre-train HP-Student, i.e., Phy-DRL, and support its continual learning. The actor and critic networks are implemented as a Multi-Layer Perceptron (MLP) with four fully connected layers. The output dimensions of critic and actor networks are 256, 128, 64, and 1, respectively. The activation functions of the first three neural layers are ReLU, while the output of the last layer is the Tanh function for the actor-network and Linear for the critic network. The input of the critic network is $[\mathbf{s}; \mathbf{a}]$, while the input of the actor-network is $\mathbf{s}$. In more detail, we let discount factor $\gamma = 0.9$, and the learning rates of critic and actor networks are the same as 0.0003. We set the batch size to 200. The maximum step number of one episode is 500.

D.2 System Dynamics

The physics knowledge about the dynamics of cart-pole systems used by HP-Student and HA-Teacher for their designs is from the following dynamics model in [51]:

$$\ddot{\theta} = \frac{g \sin\theta + \cos\theta \left(\frac{-F - m_p l \dot{\theta}^2 \sin\theta}{m_c + m_p} \right)}{l \left(\frac{4}{3} - \frac{m_p \cos^2\theta}{m_c + m_p} \right)}, \tag{48a}$$

$$\ddot{x} = \frac{F + m_p l \left(\dot{\theta}^2 \sin\theta - \ddot{\theta} \cos\theta \right)}{m_c + m_p}, \tag{48b}$$

whose parameters' physical representations and values are given in Table 2.

Notation		Value	Unit
m_c	mass of cart	0.94	kg
m_p	mass of pole	0.23	kg
g	gravitational acceleration	9.8	$m \cdot s^{-2}$
l	half length of pole	0.32	m
T	sample period	1/30	s
F	actuator input	[-30, 30]	N
μ_c	cart friction coefficient	18	
μ_p	pole friction coefficient	0.0031	
x	position of cart	[-0.8, 0.8]	m
$\dot{x}$	velocity of cart	[-3, 3]	$m \cdot s^{-1}$
θ	angle of pole	[-0.9, 0.9]	rad
$\dot{\theta}$	angular velocity of pole	[-4.5, 4.5]	$rad \cdot s^{-1}$

Table 2: Notation Table for Cart-Pole System

D.3 HP-Student: Physics Knowledge and Design

As Phy-DRL allows us to simply have nonlinear dynamics (48) to an analyzable line model:

$$\dot{\mathbf{s}} = \widehat{\mathbf{A}} \cdot \mathbf{s} + \widehat{\mathbf{B}} \cdot \mathbf{a}, \tag{49}$$

where $\mathbf{s} = [x, v, \theta, \omega]^\top$. To have $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{B}}$ from Equation (48), we let $\cos\theta \approx 1$, $\sin\theta \approx \theta$ and $\omega^2 \sin\theta \approx 0$. Meanwhile, the sampling technique transforms the continuous-time model (49) to the discrete-time model:

$$\mathbf{s}(k+1) = \mathbf{A} \cdot \mathbf{s}(k) + \mathbf{B} \cdot \mathbf{a}(k), \text{ with } \mathbf{A} = \mathbf{I}_4 + T \cdot \widehat{\mathbf{A}}, \ \mathbf{B} = T \cdot \widehat{\mathbf{A}},$$

where we have

$$\mathbf{A} = \begin{bmatrix} 1 & 0.0333 & 0 & 0 \\ 0 & 1 & -0.0565 & 0 \\ 0 & 0 & 1 & 0.0333 \\ 0 & 0 & 0.8980 & 1 \end{bmatrix}, \quad \mathbf{B} = [0 \ \ 0.0334 \ \ 0 \ \ -0.0783]^\top. \tag{50}$$

Considering the safety conditions in Equation (22) and action space condition in Equation (23) and the formulas of safety set in Equation (2) and action set in Equation (7), we have

$$\mathbf{D}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},\ \mathbf{v}=\begin{bmatrix} 0 \\ 0 \end{bmatrix},\ \overline{\mathbf{v}}=\begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix},\ \underline{\mathbf{v}}=\begin{bmatrix} -0.9 \\ -0.8 \end{bmatrix},\ \mathbf{C}=1,\ \mathbf{z}=0,\ \overline{\mathbf{z}}=25,\ \underline{\mathbf{z}}=-25 \quad (51)$$

based on which, then according to the $\overline{\Lambda}$, $\underline{\Lambda}$ and $\mathbf{d}$ defined inLemma A.2, $\overline{\Delta}$, $\underline{\Delta}$ and $\mathbf{c}$ defined in Lemma A.3, we have

$$\mathbf{d} = \begin{bmatrix} -1 \\ -1 \end{bmatrix},\quad \overline{\Lambda} = \underline{\Lambda} = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.8 \end{bmatrix},\quad \overline{\Delta} = \underline{\Delta} = 25,\quad \mathbf{c} = -1, \qquad (52)$$

from which and Equation (51), we then have

$$\overline{\mathbf{D}} = \frac{\mathbf{D}}{\overline{\Lambda}} = \underline{\mathbf{D}} = \frac{\mathbf{D}}{\underline{\Lambda}} = \begin{bmatrix} \frac{10}{9} & 0 & 0 & 0 \\ 0 & 0 & \frac{5}{4} & 0 \end{bmatrix},\quad \overline{\mathbf{C}} = \frac{\mathbf{C}}{\overline{\Delta}} = \underline{\mathbf{C}} = \frac{\mathbf{C}}{\underline{\Delta}} = \frac{1}{25}. \qquad (53)$$

Then, for givn $\alpha = 0.87$, $\beta = 0.002$, and the model knowledge $(\mathbf{A}, \mathbf{B})$ in [21], using the CVX toolbox [52, 53], we obtain:

$$\mathbf{P} = \begin{bmatrix} 13.3812 & 6.9085 & 17.0004 & 3.6284 \\ 6.9085 & 4.1226 & 10.3597 & 2.2293 \\ 17.0004 & 10.3597 & 28.2701 & 5.8142 \\ 3.6284 & 2.2293 & 5.8142 & 1.2723 \end{bmatrix},$$

$$\mathbf{F} = [\ 22.4008 \quad 16.9978 \quad 69.0659 \quad 12.6449\ ],$$

with which, physics model knowledge in Equation (50), and letting $w(\mathbf{s}(k), \mathbf{a}(k)) = -\mathbf{a}^2_{\text{drl}}(k)$, the residual action policy 4 and the safety-embedded reward (5) are then ready for the Phy-DRL.

D.4 HA-Teacher: Physics Knowledge and Design

Compared with HP-Student, HA-Teacher has relatively rich physics knowledge about system dynamics, which is directly and equivalently transformed from Equation (48) as

$$\frac{d}{dt}\underbrace{\begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}}_{\mathbf{s}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-m_p g \sin\theta\cos\theta}{\theta[\frac{4}{3}(m_c+m_p)-m_p\cos^2\theta]} & \frac{\frac{4}{3}m_p l \sin\theta\dot{\theta}}{\frac{4}{3}(m_c+m_p)-m_p\cos^2\theta} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g\sin\theta(m_c+m_p)}{l\theta[\frac{4}{3}(m_c+m_p)-m_p\cos^2\theta]} & \frac{-m_p\sin\theta\cos\theta\dot{\theta}}{\frac{4}{3}(m_c+m_p)-m_p\cos^2\theta} \end{bmatrix}}_{\widehat{\mathbf{A}}(\mathbf{s})} \cdot \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} 0 \\ \frac{\frac{4}{3}}{\frac{4}{3}(m_c+m_p)-m_p\cos^2\theta} \\ 0 \\ \frac{-\cos\theta}{l[\frac{4}{3}(m_c+m_p)-m_p\cos^2\theta]} \end{bmatrix}}_{\widehat{\mathbf{B}}(\mathbf{s})} \cdot \underbrace{F}_{\mathbf{a}}, \qquad (54)$$

where $\widehat{\mathbf{A}}(\mathbf{s})$ and $\widehat{\mathbf{B}}(\mathbf{s})$ are known to HA-Teacher. The sampling technique transforms the continuous-time dynamics model (59) to the discrete-time one:

$$\mathbf{s}(k+1) = (\mathbf{I}_4 + T \cdot \widehat{\mathbf{A}}(\mathbf{s})) \cdot \mathbf{s}(k) + T \cdot \widehat{\mathbf{B}}(\mathbf{s}) \cdot \mathbf{a}(k),$$

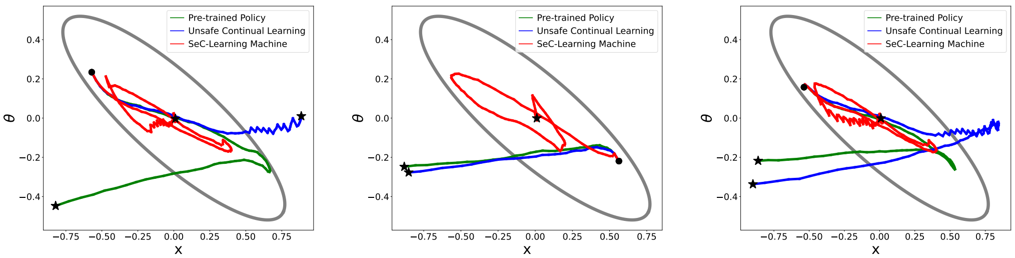
from which we obtain the knowledge of $\mathbf{A}(\overline{\mathbf{s}}^*)$ and $\mathbf{B}(\overline{\mathbf{s}}^*)$ in Equation (11) as

$$\mathbf{A}(\overline{\mathbf{s}}^*) = \mathbf{I}_4 + T \cdot \widehat{\mathbf{A}}(\overline{\mathbf{s}}^*) \ \text{ and }\ \mathbf{B}(\overline{\mathbf{s}}^*) = T \cdot \widehat{\mathbf{B}}(\overline{\mathbf{s}}^*). \qquad (55)$$

Meanwhile, for the center of the envelope patch (12), the model mismatch in Assumption 6.1, and the switch-triggering condition and dwell time in (9), we let $\chi = 0.25$, $\kappa = 0.02$, $\varepsilon = 0.6$, and $\tau = 10$. To always have feasible LMIs (17) and (18), we let $\alpha = 0.95$ and $\eta = 1.1$. These parameters also guarantee the condition in Equation (19) holds.

D.5 Additional Experimental Results

To further demonstrate the distinguished feature – lifetime safety, additional experimental results are presented in Figure 7, Figure 8, and Figure 9, where the ‘Unsafe Continual Learning’ and ‘SeC-Learning Machine’ are picked models trained after only **three episodes**, **four episodes**, **five episodes**, respectively, in continual learning.

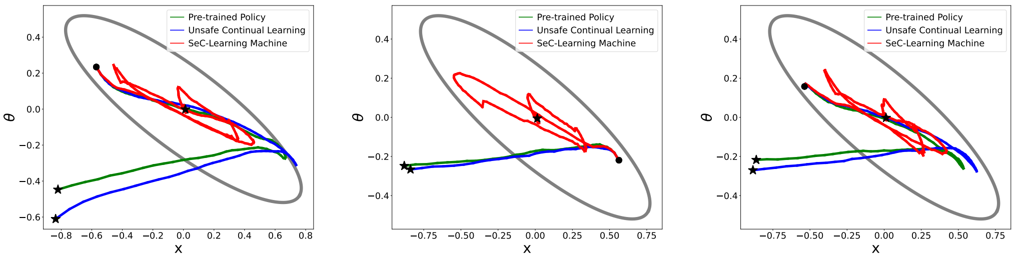


(a) Initial Condition 1

(b) Initial Condition 2

(c) Initial Condition 3

Figure 7: **Three Episodes**. Phase plots, given the same initial condition. The black dot and star denote the initial condition and final location, respectively.

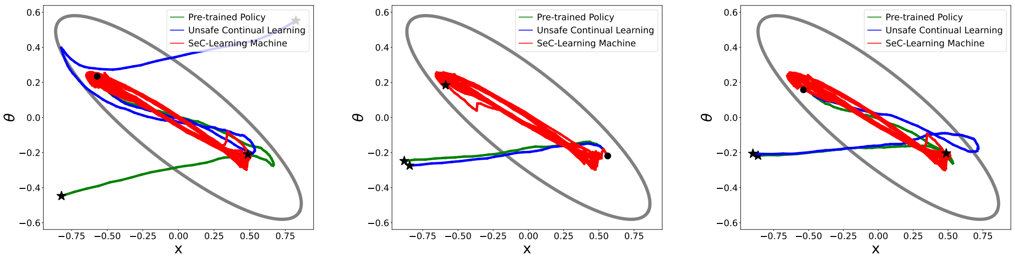


(a) Initial Condition 1

(b) Initial Condition 2

(c) Initial Condition 3

Figure 8: **Four Episodes** Phase plots, given the same initial condition. The black dot and star denote the initial condition and final location, respectively.



(a) Initial Condition 1

(b) Initial Condition 2

(c) Initial Condition 3

Figure 9: **Five Episodes**. Phase plots, given the same initial condition. The black dot and star denote the initial condition and final location, respectively.

E Experiment: Real Quadruped Robot

In the quadruped experiment, we adopt a Python-based framework for a Unitree A1 robot, released in GitHub by [54]. The framework includes a simulation based on Pybullet, an interface for direct sim-to-real transfer, and an implementation of the Convex MPC Controller for basic motion control.

E.1 Policy Learning

We train SeC-learning machine and Phy-DRL to achieve the safe mission introduced in Section 7.2. The observation of the policy is a 12-dimensional tracking error vector between the robot's state vector and the mission vector. The agent's actions offset the desired positional and lateral accelerations generated from the model-based policy. The computed accelerations are then converted to the low-level motors' torque control.

The policy is trained using DDPG algorithm [41]. The actor and critic networks are implemented as a Multi-Layer Perceptron (MLP) with four fully connected layers. The output dimensions of the critic network are 256, 128, 64, and 1. The output dimensions of actor networks are 256, 128, 64, and 6. The input of the critic network is the tracking error vector and the action vector. The input of the actor network is the tracking error vector. The activation functions of the first three neural layers are ReLU, while the output of the last layer is the Tanh function for the actor network and Linear for the critic network. In more detail, we let discount factor $\gamma = 0.9$, and the learning rates of critic and actor networks are the same as 0.0003. We set the batch size to 512. The maximum step number for one episode is 10,000.

E.2 System Dynamics

The physics knowledge about the robot used by HP-Student and HA-Teacher for their designs is from the dynamics model of the robot, which is based on a single rigid body subject to forces at the contact patches [55]. The considered robot dynamics is characterized by the position of the body's center of mass (CoM) height h, the CoM velocity $\mathbf{v} \triangleq \dot{\mathbf{p}} = [\text{CoM x-velocity}; \text{CoM y-velocity}; \text{CoM z-velocity}] \in \mathbb{R}^3$, the Euler angles $\widetilde{\mathbf{e}} = [\phi; \theta; \psi] \in \mathbb{R}^3$ with ϕ, θ and ψ being roll, pitch and yaw angles, respectively, and the angular velocity in world coordinates $\mathbf{w} \in \mathbb{R}^3$.

According to the literature [55], the body dynamics of quadruped robots can be described by

$$\frac{\mathbf{d}}{\mathbf{d}t}\underbrace{\begin{bmatrix} h \\ \widetilde{\mathbf{e}} \\ \mathbf{v} \\ \mathbf{w} \end{bmatrix}}_{\triangleq\, \widehat{\mathbf{s}}} = \underbrace{\begin{bmatrix} \mathbf{O}_{1\times1} & \mathbf{O}_{1\times5} & 1 & \mathbf{O}_{1\times3} \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{R}(\phi,\theta,\psi) \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} \\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} \end{bmatrix}}_{\triangleq\, \widehat{\mathbf{A}}(\phi,\theta,\psi)} \cdot \begin{bmatrix} h \\ \widetilde{\mathbf{e}} \\ \mathbf{v} \\ \mathbf{w} \end{bmatrix} + \widehat{\mathbf{B}} \cdot \widehat{a} + \begin{bmatrix} 0 \\ \mathbf{O}_{3\times1} \\ \mathbf{O}_{3\times1} \\ \widetilde{\mathbf{g}} \end{bmatrix} + \mathbf{f}(\widehat{\mathbf{s}}), \qquad (56)$$

where $\widetilde{\mathbf{g}} = [0; 0; -g] \in \mathbb{R}^3$ with g being the gravitational acceleration, $\mathbf{f}(\widehat{\mathbf{s}})$ denotes model mismatch, $\widetilde{\mathbf{B}} = [\mathbf{O}_{4\times6};\ \mathbf{I}_6]^\top$, and the $\mathbf{R}(\phi,\theta,\psi) = \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi) \in \mathbb{R}^{3\times3}$ is the rotation matrix with

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \mathbf{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

E.3 HP-Student: Physics Knowledge and Design

To have the model knowledge represented by $(\mathbf{A},\ \mathbf{B})$ pertaining to robot dynamics (56), we make the simplification: $\mathbf{R}(\phi,\theta,\psi) = \mathbf{I}_3$, which is obtained through setting the zero angels of roll, pitch and yaw, i.e., $\phi = \theta = \psi = 0$. Referring to (56) and ignoring unknown model mismatch, we can obtain a simplified linear model pertaining to robot dynamics (56):

$$\frac{\mathbf{d}}{\mathbf{d}t}\underbrace{\begin{bmatrix}\widetilde{h}\\ \widetilde{\widetilde{\mathbf{e}}}\\ \widetilde{\mathbf{v}}\\ \widetilde{\mathbf{w}}\end{bmatrix}}_{\triangleq\,\widetilde{\mathbf{s}}}=\underbrace{\begin{bmatrix}\mathbf{O}_{1\times1} & \mathbf{O}_{1\times3} & 1 & \mathbf{O}_{1\times5}\\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{R}(\phi,\theta,\psi)\\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3}\\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3}\end{bmatrix}}_{\triangleq\,\widetilde{\mathbf{A}}}\cdot\begin{bmatrix}\widetilde{h}\\ \widetilde{\widetilde{\mathbf{e}}}\\ \widetilde{\mathbf{v}}\\ \widetilde{\mathbf{w}}\end{bmatrix}+\widehat{\mathbf{B}}\cdot\widetilde{a}. \tag{57}$$

Given the equilibrium point (or control goal) $\mathbf{s}^*$ and $\widetilde{\mathbf{s}}$ given in Equation (57), we define $\mathbf{s} \triangleq \widetilde{\mathbf{s}} - \mathbf{s}^*$. It is then straightforward to obtain a dynamics from Equation (57) as $\dot{\mathbf{s}} = \widetilde{\mathbf{A}} \cdot \mathbf{s} + \widehat{\mathbf{B}} \cdot \widetilde{\mathbf{a}}$, which transforms to a discrete-time model via sampling technique:

$$\mathbf{s}(k+1) = \mathbf{A}\cdot\mathbf{s}(k) + \mathbf{B}\cdot\widetilde{\mathbf{a}}(k), \text{ with } \mathbf{A} = \mathbf{I}_{10} + T\cdot\widetilde{\mathbf{A}} \text{ and } \mathbf{B} = T\cdot\widehat{\mathbf{B}}, \tag{58}$$

where T is the sampling period.

with which and matrices $\mathbf{A}$ and $\mathbf{B}$ in Equation (58), we are able to deliver the residual action policy (4) and safety-embedded reward (5).

E.4 HA-Teacher: Physics Knowledge and Design

Compared with HP-Student, HA-Teacher has relatively rich physics knowledge about system dynamics, which is directly and equivalently transformed from Equation (56) as

$$\frac{d}{dt}\underbrace{\begin{bmatrix}h\\ \widetilde{\mathbf{e}}\\ \mathbf{v}\\ \mathbf{w}\end{bmatrix}}_{\mathbf{s}}=\underbrace{\begin{bmatrix}\mathbf{O}_{1\times1} & \mathbf{O}_{1\times3} & 1 & \mathbf{O}_{1\times5}\\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{R}(\phi,\theta,\psi)\\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3}\\ \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3} & \mathbf{O}_{3\times3}\end{bmatrix}}_{\widehat{\mathbf{A}}(\mathbf{s})}\cdot\begin{bmatrix}h\\ \widetilde{\mathbf{e}}\\ \mathbf{v}\\ \mathbf{w}\end{bmatrix}+\underbrace{\begin{bmatrix}\mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3\\ \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3\\ \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{I}_3 & \mathbf{O}_3\\ \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{I}_3\end{bmatrix}}_{\widehat{\mathbf{B}}(\mathbf{s})}\cdot\mathbf{a} + \mathbf{g}(\mathbf{s}), \tag{59}$$

where $\widehat{\mathbf{A}}(\mathbf{s})$ and $\widehat{\mathbf{B}}(\mathbf{s})$ are known to HA-Teacher. The sampling technique transforms the continuous-time dynamics model (59) to the discrete-time one:

$$\mathbf{s}(k+1) = (\mathbf{I}_4 + T\cdot\widehat{\mathbf{A}}(\mathbf{s}))\cdot\mathbf{s}(k) + T\cdot\widehat{\mathbf{B}}(\mathbf{s})\cdot\mathbf{a}(k) + T\cdot\mathbf{g}(\mathbf{s}),$$

from which we obtain the knowledge of $\mathbf{A}(\overline{\mathbf{s}}^*)$ and $\mathbf{B}(\overline{\mathbf{s}}^*)$ in Equation (11) as

$$\mathbf{A}(\overline{\mathbf{s}}^*) = \mathbf{I}_4 + T\cdot\widehat{\mathbf{A}}(\overline{\mathbf{s}}^*) \text{ and } \mathbf{B}(\overline{\mathbf{s}}^*) = T\cdot\widehat{\mathbf{B}}(\overline{\mathbf{s}}^*). \tag{60}$$

Meanwhile, for the center of the envelope patch (12), the model mismatch in Assumption 6.1, and the switch-triggering condition and dwell time in (9), we let $\chi = 0.25$, $\kappa = 0.02$, $\varepsilon = 0.6$, and $\tau = 10$. To always have feasible LMIs (17) and (18), we let $\alpha = 0.99$ and $\eta = 1.1$. These parameters also guarantee that the condition in Equation (19) is fulfilled.

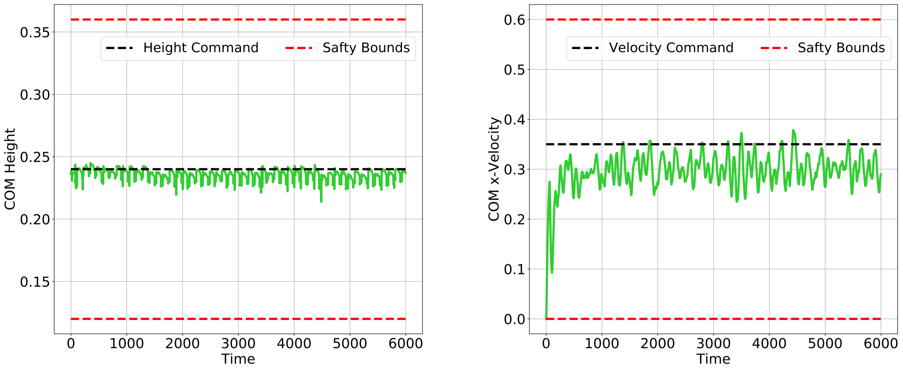
E.5 Additional Experimental Results

E.5.1 Trajectories

The real robot's trajectories of COM height and COM x-velocities under the control of the SeC-learning machine in the 5th episode, 10th episode, 15th episode, and 20th episode are shown in Figures 10 to 13, respectively. The figures straightforwardly depict that the SeC-learning machine guarantees the safety of real robots in all picked episodes of continual learning.

E.5.2 Reward

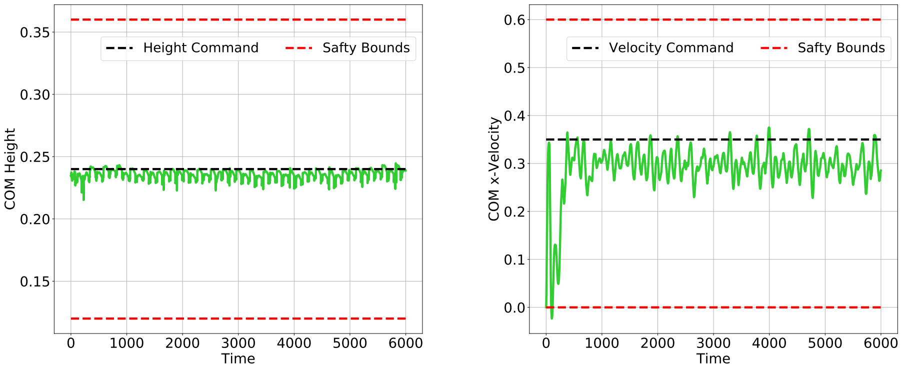
The reward curves in the iteration step for 20 episodes are shown in Figure 14. Observing Figure 14, we conclude that given the same reward for learning, the SeC-Learning Machine exhibits remarkably fast and stable learning, compared with continual learning without Simplex logic.



(a) Height trajectory

(b) Velocity trajectory

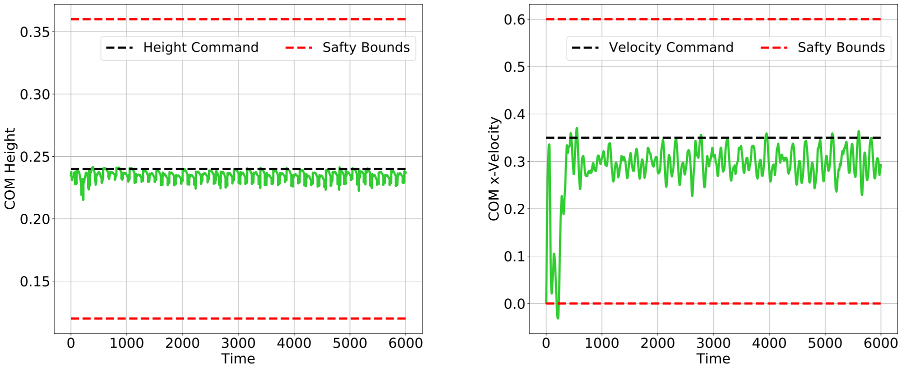
Figure 10: Robot's trajectories under control of SeC learning machine in the **5th Episode**.



(a) Height trajectory

(b) Velocity trajectory

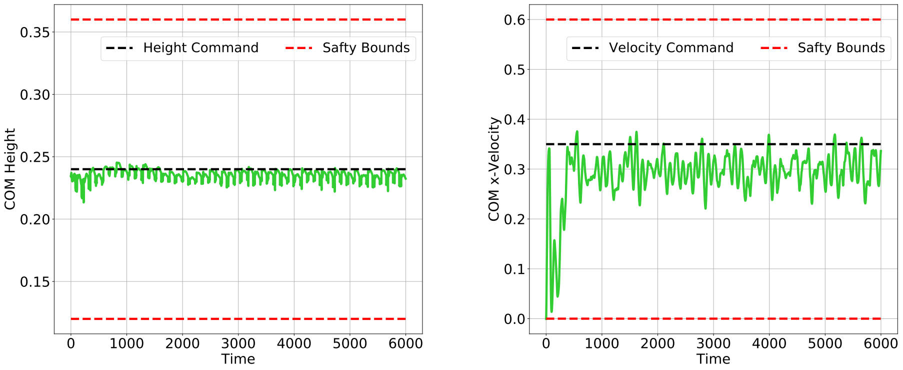
Figure 11: Robot's trajectories under control of SeC learning machine in the **10th Episode**.



(a) Height trajectory

(b) Velocity trajectory

Figure 12: Robot's trajectories under control of SeC learning machine in the **15th Episode**.



(a) Height trajectory

(b) Velocity trajectory

Figure 13: Robot's trajectories under control of SeC learning machine in the **20th Episode**.

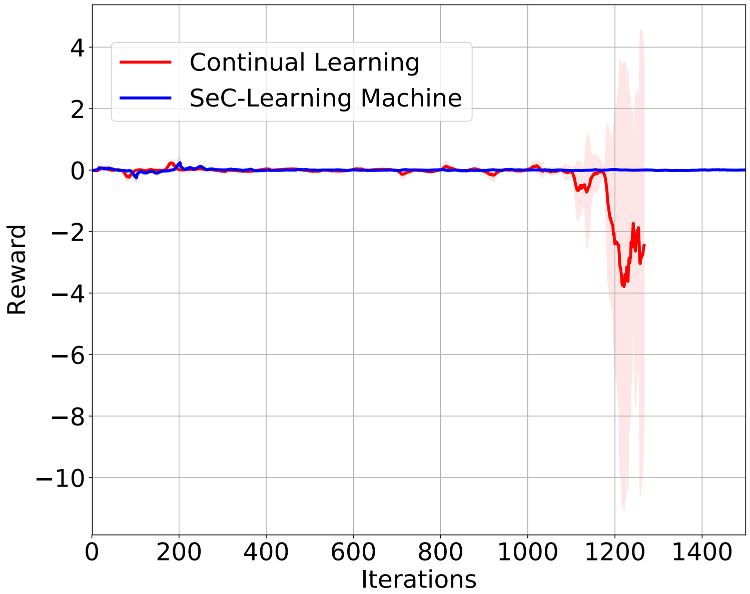


Figure 14: Reward curves in the term of iteration steps.