การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 48
The 48th Electrical Engineering Conference (EECON-48)
วันที่ 19-21 พฤศจิกายน 2568 ณ โรงแรมฟูราม่า จังหวัดเชียงใหม่

The Data-Driven Safeguard: Redefined Intelligent Control as Smart and Safe

Tanet Wonghong*

Department of Electrical Engineering, Faculty of Engineering at Sriracha, Kasetsart University, Thailand,
tanet.w@ku.th

Abstract

The integration of artificial intelligence into control systems remains hindering by a fundamental barrier: the inherent risk of applying learning algorithms to physical, safety-critical systems. This paper introduces a paradigm shift in intelligent control by demonstrating that this high-risk, trial-and-error problem can be transformed into a completely safe, purely data-driven, black-box optimization task. We present a novel framework, the "Data-Driven Safeguard," that constructs a high-fidelity "fictitious fitness landscape" using only a single set of measured input-output data. The surprising core of this method lies not in a complex AI model, but in the elegant solution of a single linear system, which allows us to compute the performance of any candidate controller without it ever interacting with physical plant. By empirically proving that this safe, virtual landscape accurately mirrors the true performance landscape, we unlock the ability for any AI optimizer – demonstrated here with an Evolution Strategy – to find optimal controller parameters with zero physical risk. This work redefines the challenge of intelligent control: the goal is not just to build a smarter AI, but to create a fundamentally safer world for it to learn in.

Keywords: Adaptive Control, Artificial Intelligence, Intelligent Control, Unfalsified Control, Safeguard Control

1. Introduction

The grand vision of Intelligent Control – systems that can self-optimize and adapt – has been a driving force in our field for decades. Early pioneering work established the immense potential of using sophisticated techniques, such as neural networks, to control complex dynamical systems where traditional models fall short [1-3]. This sparked a long-standing quest to create truly autonomous controllers. However, a fundamental and persistent challenge has always stood in the way: "learner's dilemma." How can an intelligent algorithm learn and explore without posing a direct risk to the physical, often safety-critical, system it is connected to?

In response to this critical safety problem, a powerful and uniquely data-driven philosophy emerged: the theory of Unfalsified Control (UC) [4]. Unlike approaches that require detailed system models, UC introduced the revolutionary idea that one could use direct input-output data to invalidate, or "falsify," controllers that are inconsistent with desired performance specifications. This provided a practical, model-free path towards ensuring system safety and was successfully demonstrated for tasks like automatic PID tuning [5]. This line of research established a key principle: that raw data itself, when used correctly, holds the key to safe control. Our work is born from this powerful lineage, seeking to extend its original promise to its ultimate conclusion.

While the UC paradigm offered an elegant, data-centric solution, the broader field has more recently pursued safety through different, more complex avenues. The current state-of-the-art is largely dominated by Safe Reinforcement Learning (Safe RL) and the use of Control Barrier Functions (CBFs) [6-7]. These modern methods attempt to manage risk by adding complex safety layers, constraints, or penalty functions into the learning algorithm, often requiring extensive simulations or an accurate mathematical model. They operate by trying to make the AI learner "smarter" or "more cautious" as it interacts with the world.

This paper argues that this prevailing direction, while valuable, misses a more fundamental opportunity. We return to the elegant spirit of Unfalsified Control and ask a different question: instead of managing the risk of a physical trial, what if we could eliminate the need for a physical trial entirely? Our work introduces the "Data-Driven Safeguard," a framework that uses the core principles of UC to achieve something new: the creation of a complete, high-fidelity "fictitious fitness landscape" from a single data set. This framework is not an incremental improvement; it is a paradigm shift. We demonstrate that the high-risk problem of intelligent control can be transformed into a perfectly safe, offline optimization task by solving a single, simple linear system.

The contribution of this work are as follows:

1. We introduce the Data-Driven Safeguard, a novel framework that transforms the high-risk physical tuning problem into a completely safe, offline computation task.
2. We show this is achieved by leveraging the principles of Unfalsified Control to solve a single, well-behaved system of linear equations, bypassing the need for complex plant models.
3. We provide direct, empirical proof that the resulting "fictitious fitness landscape" is a high-fidelity replica of the true performance landscape, validating it as a safe and accurate virtual testbed.
4. We demonstrate, through a proof-of-concept using an Evolution Strategy, that this framework is optimizer-agnostic and can be readily used by and AI algorithm to find optimal controller parameters with zero physical risk.

*Corresponding Author

This paper presents a simpler, more direct path to achieving unconditionally safe intelligent control, fulfilling the original promise of data-driven methods and bypassing the inherent limitations of more complex, modern approaches.

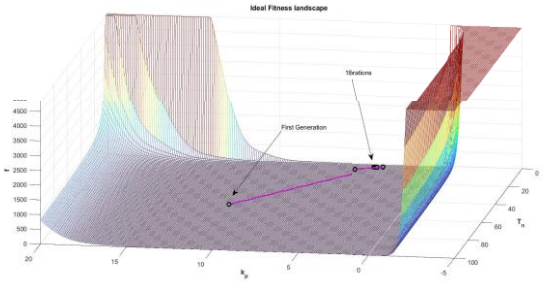

Fig. 1 Ideal Fitness Landscape

Table 1 Routh Table for Stability

s^4	1	11	$6\frac{k_p}{T_n}$
s^3	6	$6k_p$+6	
s^2	$-k_p$+10	$6\frac{k_p}{T_n}$	
s^1	x		
s^0	$6\frac{k_p}{T_n}$		

2. The Ideal Stabilizing Region

We assume that the linear plant model is a third-order sable process with three distinct poles:

$$P(s) = \frac{6}{(s+1)(s+2)(s+3)} \quad (1)$$

For simplicity, we implement the feedback loop with the standard PI controller structure:

$$C(s) = k_p(1 + \frac{1}{T_n s}) \quad (2)$$

This controller is chosen as it represents the most widely used controller in industrial applications, making our safety framework directly relevant to a broad range of real-world problems. From (1) and (2), we generate the Routh Table for our analysis as shown in Table 1.

Using the Routh-Hurwitz criterion, all elements in the first column must be positive:

$$s^2: -k_p + 10 > 0 \rightarrow k_p < 10$$

Since the integral time is always positive,

$$s^0: 6\frac{k_p}{T_n} > 0 \rightarrow k_p > 0.$$

We now have the first condition on the proportional gain $k_p \in (0,10)$ and

$$x = \frac{-36\frac{k_p}{T_n} - 6k_p^2 + 54k_p + 60}{-k_p + 10} \quad (3)$$

For T_n, we consider (3), $x > 0$ and we compute T_n when $k_p \in \{1,2,\dots,9\}$ as shown in Table 2.

To visualize the ideal stabilizing region, MATLAB is used to make it a clear picture and it will be compared to our work under the same situation. Firstly, we compute the true error signal $e(t)$ for 10 seconds for each candidate controller within the candidate controller space $k_{p_i} \in [-5,20]$ and $T_{n_i} \in [-1,100]$.

Table 2 Region of Stabilizing Controllers

k_p	T_n
1	>0.3333
2	>0.5
3	>0.643
4	>0.8
5	>1
6	>1.286
7	>1.75
8	>2.67
9	>5.4

Secondly, to evaluate performances for all candidate controllers, we define the following cost function:

$$J_i(t) = \frac{\|e_i(t)\|^2_{t=[0,10s]}}{\|r(t)\|^2_{t=[0,10s]}} \quad (4)$$

where i is an index of a candidate controller and $r(t)$ is a unit step function. Thirdly, the ideal fitness landscape is plotted as shown in Fig. 1. Next, we implement the standard Evolution Strategy (ES) [8] as our artificial intelligence computing method (any other optimization techniques can be applied) to find ideal optimized controllers using initial sixteen destabilizing controllers as $k_{p_i} \in \{-5,-1,11,15\}$ and $T_{n_i} \in \{1,5,10,50\}$.

Finally, the ES at 18 generations and returns the results as shown in Table 3. The pair $(k_p^* = 3.4204, T_n^* = 3.6401)$ is the ideal optimized controller. This solution will be used to compare with our approach as an intelligent indicator.

Table 3 Sixteen Optimized Controllers for the Ideal Fitness Landscape

Rank	k_p^*	T_n^*
1	3.4202	3.6401
2	3.4433	3.7070
3	3.3581	3.5300
4	3.3630	3.6440
5	3.3909	3.7262
6	3.3891	3.5196
7	3.3545	3.6535
8	3.4507	3.7972
9	3.3652	3.6883
10	3.3165	3.5413
11	3.4761	3.6713
12	3.4861	3.8115
13	3.3121	3.5266
14	3.3125	3.4897
15	3.4921	3.7216
16	3.3555	3.4414

3. Unfalsified Control Theory

3.1 Experimental Plant Data

We assume that $P(s)$ is a completely unknown plant, $C_i(s)$ is a candidate controller where $i \in \boldsymbol{M} = \{1,2,\dots,m\}$. The finite set of m candidate controllers is defined as $\omega_m = \omega_s \cup \omega_d$ where ω_s is a set of stabilizing controllers and ω_d is a set of destabilizing controllers for the unknown plant $P(s)$. Then we assume an active controller $\hat{C}(s) \in \omega_s$ and we can observe the bounded experimental plant input data as

$$\hat{U}(s) = \frac{\hat{C}(s)}{1+\hat{C}(s)P(s)}R(s) \tag{5}$$

and the bounded experimental plant output data as

$$\hat{Y}(s) = \frac{\hat{C}(s)P(s)}{1+\hat{C}(s)P(s)}R(s). \tag{6}$$

3.2 Original Fictitious Signals

The original fictitious reference signal [4] for a candidate controller $C_i(s)$ can be computed as

$$\tilde{R}_i(s) = C_i^{-1}(s)\hat{U}(s) + \hat{Y}(s). \tag{7}$$

Using (6) and (7), the original fictitious error signal for $C_i(s)$ is obtained as

$$\tilde{E}_i(s) = \tilde{R}_i(s) - \hat{Y}(s). \tag{8}$$

3.3 Relationship of Fictitious Reference Signal and True Reference Signal

Taking (5) and (6) into (7), we obtain

$$\tilde{R}_i(s) = \frac{\hat{C}(s)}{C_i(s)}\frac{1+C_i(s)P(s)}{1+C(s)P(s)}R(s) = \frac{\hat{C}(s)\hat{S}(s)}{C_i(s)S_i(s)}R(s)$$

$$\tilde{R}_i(s) = \tilde{\Lambda}_i(s)R(s) \tag{9}$$

where $\hat{S}(s) = \frac{1}{1+\hat{C}(s)P(s)}$ is the unknown sensitivity function for the active closed-loop pair $(\hat{C}(s), P(s))$ and $S_i(s) = \frac{1}{1+C_i(s)P(s)}$ is the unknown sensitivity function for a non-active closed-loop pair $(C_i(s), P(s))$. A black-box mapping $\tilde{\Lambda}_i(s)$ is called the fictitious signal generator for $C_i(s)$. Note that if $C_i(s) = \hat{C}(s)$, then $\tilde{R}_i(s) = R(s)$ and $\tilde{E}_i(s) = E(s)$. Thus, the original fictitious signals for the active controller $\hat{C}(s)$ are the same as the true signals.

3.4 PI Controller Structure

We use the standard PI controller structure, e.g., $C_i(s) = k_{p_i}(1 + \frac{1}{T_{n_i}s})$. Using (9), we obtain

$$\tilde{R}_i(s) = \frac{s}{k_{p_i}s + \frac{k_{p_i}}{T_{n_i}}}\hat{U}(s) + \hat{Y}(s). \tag{10}$$

We realize (10) into a state-space representation,

$$\begin{bmatrix}\dot{x}_i \\ \cdots \\ \tilde{r}_i\end{bmatrix} = \begin{bmatrix}-\frac{1}{T_{n_i}} & \frac{1}{k_{p_i}} & 0 \\ \cdots & \cdots & \cdots \\ -\frac{1}{T_{n_i}} & \frac{1}{k_{p_i}} & 1\end{bmatrix}\begin{bmatrix}x_i \\ \cdots \\ \hat{u} \\ \hat{y}\end{bmatrix}. \tag{11}$$

Note that the state-space system is always stable since the integral time is positive. Thus, the PI controller structure is causally left invertible.

3.5 Deconvolution

We use the relationship of the original fictitious signals [9]:

$$\tilde{E}_i(s) = S_i(s)\tilde{R}_i(s). \tag{12}$$

Considering the transfer function in the time-domain:

$$\{\tilde{e}_i(k)\} = \{s_i(k)\} * \{\tilde{r}_i(k)\}$$

where $\tilde{e}_i(k) = \sum_{j=0}^{k}\tilde{r}_i(k-j)s_i(j), \forall i$ and iterating $k = 0,1,2,..,l$:

$$\begin{aligned}\tilde{e}_i(0) &= \tilde{r}_i(0)s_i(0)\\ \tilde{e}_i(1) &= \tilde{r}_i(1)s_i(0) + \tilde{r}_i(0)s_i(1)\\ &\vdots\\ \tilde{e}_i(l) &= \sum_{j=0}^{l}\tilde{r}_i(l-j)s_i(l).\end{aligned}$$

It is straightforward to compute $\{s_i(k)\}$ iteratively as

$$\begin{aligned}s_i(0) &= \frac{\tilde{e}_i(0)}{\tilde{r}_i(0)}\\ s_i(1) &= \frac{\tilde{e}_i(1) - s_i(0)\tilde{r}_i(1)}{\tilde{r}_i(0)}\\ &\vdots\\ s_i(l) &= \frac{\tilde{e}_i(l) - \sum_{m=0}^{l-1}s_i(m)\tilde{r}_i(l-m)}{\tilde{r}_i(0)}\end{aligned}$$

where

$$\forall i, \tilde{r}_i(0) \neq 0.$$

Thus, the estimated sensitivity vector is

$$\boldsymbol{s_i} = \begin{bmatrix}s_i(0) \\ s_i(1) \\ s_i(2) \\ \vdots \\ s_i(l)\end{bmatrix}. \tag{13}$$

3.6 Artificial Error Signal

The artificial error signal can be defined [6] as

$$E_i^*(s) = S_i(s)R(s) \tag{14}$$

or in the time-domain, we directly compute as

$$\begin{aligned}e_i^*(0) &= r(0)s_i(0)\\ e_i^*(1) &= r(1)s_i(0) + r(0)s_i(1)\\ &\vdots\\ e_i^*(l) &= \sum_{j=0}^{l}r(l-j)s_i(l).\end{aligned}$$

Therefore, the artificial error vector is

$$\boldsymbol{e_i^*} = \begin{bmatrix}e_i^*(0) \\ e_i^*(1) \\ e_i^*(2) \\ \vdots \\ e_i^*(l)\end{bmatrix}. \tag{15}$$

3.7 Artificial Cost Function

In this work, we assume that the stabilizing controller

$$\hat{C}(s) = 1(1 + \frac{1}{s})$$

was in the loop up to 10 seconds after a unit step was applied. We have observed $r_{[0,10]}(t), \hat{u}_{[0,10]}(t), \hat{y}_{[0,10]}(t)$. Note that the unknown plant $P(s)$ is used for generating our plant input and output data. The artificial cost function

is defined similarly with the artificial error signal $e_i^*(k)$ and the true reference signal $r(k)$ as

$$J_i^*(k) = \frac{\|e_i^*(k)\|^2}{\|r(k)\|^2}. \quad (16)$$

In this example, we implement every signal for 10 seconds and the sampling time is 0.1 seconds. Thus, the maximum iteration is 100. Using the artificial cost function, the fictitious landscape is computed as shown in Fig. 2 and the ES terminates at 20 generations as shown in Table 4. Note that the pair $(k_p^* = 2.9430, T_n^* = 3.2396)$ is the fictitious optimized controller. In addition, using MATLAB, we can design a PI controller using "pidtune" function as follows:

sys = zpk([],[-1,-2,-3], 6)
C0 = pidstd(1,1,0)
C = pidtune(sys, C0).

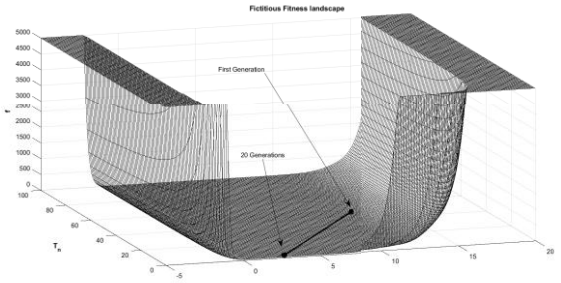

Fig. 2 The Fictitious Fitness Landscape

We obtain the pair $(k_p^* = 1.26, T_n^* = 1.51)$. Finally, three optimized controllers from the three different approaches as shown in Fig. 3.

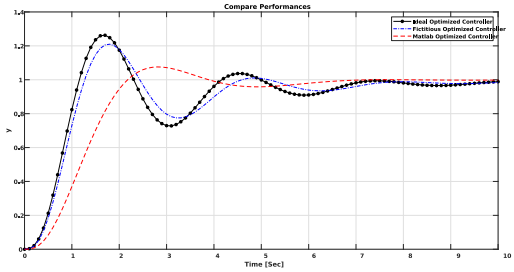

Fig. 3 Comparison

Table 4 Sixteen Optimized Controllers for the Fictitious Fitness Landscape

Rank	k_p^*	T_n^*
1	2.9430	3.2396
2	2.9273	3.1987
3	2.9377	3.3105
4	2.9635	3.2736
5	2.9396	3.3185
6	2.9017	3.2102
7	2.8999	3.1996
8	2.8930	3.1961
9	2.8894	3.1723
10	2.9359	3.3389
11	2.8910	3.1627
12	2.8955	3.1516
13	2.8852	3,1430
14	2.8846	3.1423
15	2.9593	3.1978
16	2.8808	3.1491

4. Discussions

From the experimental results as shown in Fig. 3, unfalsified control can bring the optimized controller out closely to the ideal one. This is the most important thing because we only use data obtained from measurements to evaluate the ability of candidate controllers without connecting them in the real loop. It is not the ability of artificial intelligence alone, but it is due to the method of computing the artificial error signals that makes controller performance prediction possible.

The results presented in this paper validate the core thesis: that the high-risk problem of intelligent control can be transformed into a completely safe, data-driven optimization task. This section discusses the practical implications of this framework, including its computational complexity, hardware deployment considerations, and its generality with respect to the choice of AI optimizer.

A. Computational Complexity and Real-World Deployment

A crucial aspect of any practical control strategy is its computational feasibility. The proposed Data-Driven Safeguard framework consists of two distinct computational stages:

1. Safeguard Construction (Solving Ax=b):

The heart of our method is the construction of the fictitious fitness landscape by solving the linear system Ax=b for the artificial error vector e^*. A key property of the matrix A is that it is a lower triangular Toeplitz matrix. This structure allows the system to be solved with extreme efficiency using forward substitution, a non-iterative and deterministic algorithm. The computational complexity of this step is $O(n^2)$, where n is the length of the data set. For typical control applications, where n might be a few thousand data points, this calculation is computationally trivial for any modern processor and can be completed in well under a second.

2. AI Optimization (Searching the Landscape):

Once the safeguard is in place, the AI optimizer (in our case, an Evolution Strategy) performs its search. This is the most computationally intensive part. However, it is critical to emphasize that this entire process is performed offline. The AI interacts only with the safe, virtual landscape, not the physical plant. Therefore, the runtime of the optimizer is not a real-time constraint. It is a practical engineering consideration, where a search lasting seconds or minutes is perfectly acceptable for finding optimal controller parameters before deployment.

Given this two-stage process, we can now address the question of hardware, While our initial thoughts might have considered low-cost platforms like a Raspberry Pi, the professional and safety-critical nature of this framework demands a more robust architecture. The ideal real-world deployment would utilize a hybrid hardware architecture:

- High-Level Optimizer: An Industrial PC (IPC) or a dedicated System-on-Module (SoM) like the NVIDIA Jetson would be responsible for the offline tasks: collecting the initial data set, solving the Ax=b system, and running the AI optimizer to find the optimal controller gains (Kp, Tn). Its powerful processor is well-suited for these computationally intensive, non-real-time tasks.
- Low-Level Real-Time Controller: The proven, safe controller parameters discovered by the AI are then downloaded to a dedicated Microcontroller (MCU) or a Digital Signal Controller (DSC). This hardware is solely responsible for executing the simple PI control law in hard real-time. This separation of concerns is a classic design pattern in safety-critical systems, ensuring that the complex, intelligent "brain" is isolated from the last, reliable "reflexes" of the real-time control loop. This architecture guarantees both intelligence and safety.

B. Generality and Alternative Optimization Strategies

The second key recommendation was to highlight the generality of our framework by considering other optimization methods. This is an excellent point that underscores a core strength of our approach: the Data-Driven Safeguard is optimizer-agnostic.

The framework's primary contribution is the creation of the safe virtual testbed. Once this testbed exists, any black-box optimization algorithm can be deployed to search it. We chose the Evolution Strategy (ES) for this paper as a robust, proof-of-concept demonstrator. However, other algorithms are equally viable and offer different performance trade-offs:

- Particle Swarm Optimization (PSO): PSO is another population-based algorithm, often lauded for its fast convergence speed. It could potentially find a high-quality solution in fewer iterations than ES. However, it can sometimes be more prone to premature convergence on local optima. Deploying PSO on our fictitious landscape would be a straightforward process and a valid alternative.
- Bayesian Optimization (BO): BO is a powerful technique for optimizing functions that are expensive to evaluate. It builds a probabilistic model of the objective function and uses it to select the most promising points to evaluate next. In our context, each "evaluation" is a quick calculation using the artificial error vector e^* vector, not a physical experiment. BO would be particularly interesting if the landscape were highly complex, as it is extremely sample-efficient.

The crucial point is that our framework makes it safe to use any of these. The choice of ES, PSO, or BO becomes a secondary engineering decision based on desired convergence speed and problem complexity, rather than a primary decision based on physical risk. While a detailed benchmark comparing the performance of these optimizers within our framework is a compelling direction for future work, it falls outside the scope of this paper's primary goal: to introduce and validate the safety concept itself. The success of the ES is sufficient to prove that the fictitious landscape is not only safe but also effectively searchable.

5. Conclusion

This paper has successfully introduced and validated a novel framework that fundamentally redefines the application of AI in control systems by resolving the critical issue of safety. We have demonstrated that it is possible to completely decouple the exploratory, trial-and-error nature of AI optimization from the physical plant. The "Data-Driven Safeguard" presented here constructs a "fictitious fitness landscape" – a safe and accurate virtual testbed for controller tuning – using nothing more than a single set of measured data. Our results empirically confirm that this virtual landscape is a high-fidelity replica of the true performance landscape, allowing an AI optimizer to find optimal solutions without ever posing a risk to the real-world system.

The most striking aspect of this work is perhaps its foundational simplicity. While the world pursues ever-more-complex AI models to solve the control problem, our research reveals that the key to unlocking safe intelligent control was not hidden in complexity, but in a change of perspective. By leveraging the principles of Unfalsified Control to solve for an "artificial error" via a simple linear system (Ax=b), we have turned what was once a high-risk physical experiment into a safe and deterministic computational problem. It is a classic case of a "hidden in plain sight" solution, where the most profound impact comes from the most elegant and unexpected of sources.

The implications of this discovery are far-reaching. This framework is optimizer-agnostic and can serve as a universal safety layer for a wide array of AI-based optimization techniques. It paves the way for the confident application of intelligent controllers in fields where safety is paramount, from industrial manufacturing

and robotics to autonomous vehicles. Ultimately, the "Data-Driven Safeguard" offers more than just a new technique; it offers a new philosophy for building the next generation of control systems – systems that are not only intelligent but are, by their very design, fundamentally safe.

6. Future Work

The success of the Data-Driven Safeguard framework opens up numerous exciting avenues for future research. We believe this work can serve as foundation stepping stone for a new generation of safe intelligent systems. We propose the following directions for researchers interested in building upon this paradigm:

1. Extension to MIMO and Complex Systems: The current work focused on a SISO system. A significant and valuable next step would be to extend the mathematical framework to handle MIMO systems, which are common in industrial processes. This would involve investigating the structure of the resulting linear equations and the scalability of the approach.
2. Online, Adaptive Safeguarding: This paper demonstrated an offline approach where the safeguard is constructed once before deployment. A highly impactful extension would be to develop an online, adaptive version. This would involve techniques for continuously updating the fictitious landscape using real-time data, allowing the AI to safely re-optimize the controller in response to changing system dynamics or environmental conditions, without ever needing to inject risky exploratory signals.
3. Benchmarking and Integration with Advanced AI Optimizers: While we proved the concept with an Evolution Strategy, a comprehensive study is warranted to benchmark the performance of various state-of-the-art optimizers (e.g., Bayesian Optimization, advances Reinforcement Learning agents like PPO or SAC) on the safe landscape. The goal would be to identify which optimizers are most efficient at exploiting the information contained within the data-driven safeguard.
4. Hardware-in-the-Loop (HIL) Validation: To bridge the gap between simulation and real-world application, the next logical step is to deploy and validate this framework on a Hardware-in-the-loop (HIL) testbed or a pilot-scale industrial process. This would provide invaluable data on the real-world performance and robustness of the safeguard under noisy conditions and hardware limitations, confirming its readiness for industrial adoption.
5. Formal Safety Guarantees: While our empirical results show a near-perfect match between the fictitious and real landscapes, a theoretical investigation into providing formal mathematical guarantees on the "closeness" of the two landscapes under specific noise andncertainty conditions would be a powerful contribution, further solidifying the framework's reliability for the most critical of applications.

References

[1] P.J. Antsaklis and K.M. Passino, "An Introduction to Intelligent and Autonomous Control," *Kluwer Academic Publishers*, 1993.
[2] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions Neural Networks*, 1990;1(1):4-27.
[3] P.J. Antsaklis, "Intelligent Control, Encyclopedia of Electrical and Electronic Engineering," *John Wiley & Sons*, 1999;10:493-503.
[4] M.G. Safonov and T.C. Tsao, "The unfalsified control concept and learing," *IEEE Transactions on Automatic Control*, 1997;42(6):843-847.
[5] M. Jun and M.G. Safonov. "Automatic PID tuning: an application of unfalsified control," *1999 IEEE International Symposium on Computer Aided Control System Design*, Kohala Coast, HI, USA, 1999:328-333.
[6] J. Garcia and F. Fernandez. "A Comprehensive Survey on Safe Reinforcement Learning," *Journal of Machine Learning Research*, 2015:1437-1480.
[7] A. Ames, X. Xu, J. Grizzle, and P. Tabuada. "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, 2017:3861-3876.
[8] H. Schwefel. "Evolution and Optimal Seeking," *John Wiley & Sons*, 1995.
[9] S. Engell, T. Tometzki, and T. Wonghong. "A New Approach to Adaptive Unfalsified Control," *2007 European Control Conference*, Kos Island, Greece, 2007:1328-1333.

Tanet Wonghong is an Assistant Professor in the Department of Electrical Engineering at Kasetsart University, Sriracha Campus. He received his Ph.D. in Bio and Chemical Engineering from TU Dortmund, Germany in 2010. His research is focused on creating fundamentally safe and reliable intelligent control systems. His interest include data-driven control, system identification, unfalsified control theory, AI safety, and the application of evolutionary algorithms to complex optimization problems.