KFAE: Kalman Filter-Autoencoder Approach for Handling Faulty and Noisy Environments

Hassanien Zanki[1], Árpád Huszák[1, 2]
[1] Department of Networked Systems and Services, Budapest University of Technology and Economics
Budapest, Hungary
[2] Cloud Applications Research Group, HUN-REN-BME
nhmohammed@hit.bme.hu

***Abstract*—Reinforcement Learning (RL) is widely applied in robotics, autonomous systems, and network optimization but struggles with sensor noise and faulty actions, leading to instability. This paper introduces KFAE, a novel approach that integrates Proximal Policy Optimization (PPO) with Autoencoder and Kalman Filter to de-noise observations and correct action faults, enhancing learning efficiency in noisy environments. KFAE stabilizes learning, improves decision reliability, and significantly outperforms the Noisy Environment (NE), achieving a 131.5% improvement in episode reward mean and an 184.1% increase in steps till collision. Additionally, KFAE closely approximates the Default Environment (DE) with minimal deviations, validating its effectiveness as a robust, fault-tolerant RL framework.**

***Keywords*—Autoencoder; AEs; Kalman Filter; Proximal Policy Optimization; PPO; Reinforcement Learning; RL.**

I. Introduction

RL has achieved remarkable success in decision-making tasks across various domains, including autonomous vehicles, robotics, and games [1]. RL is a computational framework where an agent learns optimal policies by interacting with an environment and maximizing cumulative rewards. Among various RL algorithms, policy-based methods have gained significant attention due to their ability to learn complex behaviors directly from high-dimensional inputs. PPO, in particular, ensures stable learning by balancing exploration and exploitation through its clipping surrogate objective. This property makes PPO sample-efficient, robust, and easy to implement, positioning it as one of the most widely used RL algorithms, alongside Trust Region Policy Optimization (TRPO) and Deep Deterministic Policy Gradient (DDPG) [2].

Despite the impressive performance of RL algorithms in controlled environments, their reliability significantly degrades in noisy and faulty environments [3]. Sensor noise, missing observations, and action faults disrupt learning, leading to suboptimal policies and unstable training. For RL agents to perform effectively in real-world settings, they must adapt to and mitigate the effects of faulty data, ensuring reliable decision-making in uncertain conditions.

The methodological approach taken in this study is a mixed methodology based on the Autoencoder-Kalman Filter technique. The Autoencoder enables us to reduce the dimensions of that data, ensuring effectively extracting meaningful representations [4]. Additionally, using the property of the Kalman Filter to enhance the agent behavior across the environment by estimating the observation states smoothly [5]. The approach to empirical research adopted for our proposal aims to make the decisions made by the agents in dynamic uncertainty environments more robust and efficient.

The remainder of this paper is structured as follows: Section II reviews the related works. The methodology and design are presented in Section III. The experimental setup is described in Section IV. The experimental results are discussed in Section V. Finally, Section VI concludes the paper.

II. Related Works

To date, several studies have investigated training an agent in RL environments with an autoencoder on high-dimensional sensory inputs (e.g., images, LiDAR). The outcome of this combination is shown to enhance the production of state representation and reduce the time of policy recovery [6]. Using this approach, researchers have been able to enhance robustness and generalization, and probabilistic modeling is proposed, named Variational Autoencoders (VAEs) [7].

The Kalman filter technique uses a repetition algorithm to estimate the state of the observation in the environment. It leverages the predictive correction approach for getting the optimal estimation state by minimizing the mean squared error [8]. Combining the Kalman filter in an RL environment will have a critical impact on the agent for the process of decision-making. It will reduce the impact of the noise on observation and the corruption of reading high-dimensional raw data, for instance. The RL agent will get a smooth representation state, certain information, and a mostly denoised environment [9].

Several large cross-sectional studies suggest the incorporation of state estimation techniques in RL to improve policy learning. For instance, Bayesian filtering methods, such as the Kalman filter and particle filters, have been applied in RL for sensor fusion and noise reduction [5]. Researchers have demonstrated that integrating state estimators into RL pipelines improves stability and robustness, particularly in robotics and autonomous systems [10]

Policy-based RL methods, such as PPO have been augmented with learned representations to enhance performance. Several studies have combined PPO with convolutional neural networks (CNNs) to extract features from raw pixels [11]. However, limited research exists on leveraging autoencoders for PPO in conjunction with state estimation techniques like

Kalman filtering. The combination of these approaches has the potential to bridge the gap between perception and decision-making, allowing for improved RL performance in complex environments. Some researchers propose RL4V2X to enhance autonomous driving by integrating CNNs, GRUs, and gate networks to handle intermittent V2X disruptions, improving safety and efficiency in dynamic traffic conditions [12].

Although previous work has demonstrated the benefits of Auto-encoders for representation learning and Kalman filtering for state estimation, their integration within a PPO-based RL framework remains largely unexplored. Existing research often focuses on either representation learning or state estimation individually but not their combined effect on policy optimization. This paper aims to fill this gap by proposing a novel approach that integrates PPO with a Kalman filter for refined state estimation and an autoencoder for compact feature extraction, enabling better policy learning in complex, noisy, and high-dimensional environments.

III. METHODOLOGY

The methodology of our design is illustrated in Fig. 1, which consists of the following phases: DE, NE, and KFAE. Each phase builds upon the previous one, progressively incorporating additional mechanisms and techniques. The final phase KFAE integrates all previous modifications while introducing a comprehensive fault-tolerant mechanism to enhance robustness.

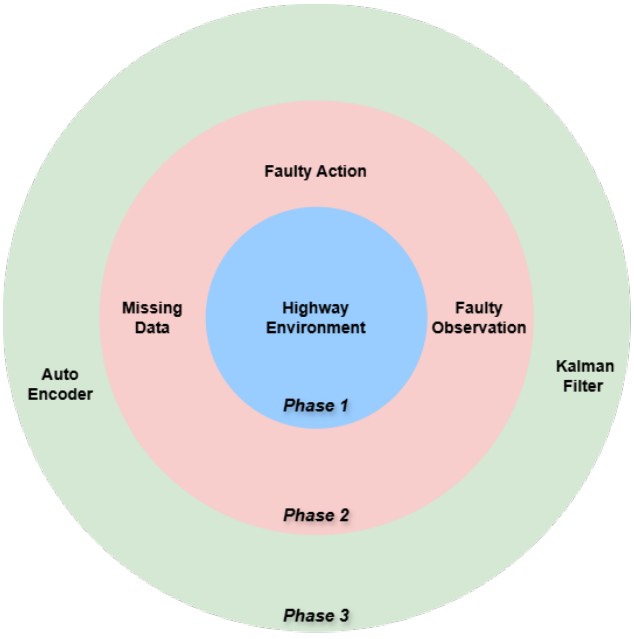

Figure 1. Design Phases

In the first phase, we trained an RL agent using the PPO algorithm within the *highway-fast-v0* environment [13], which is an open-source simulation environment designed for autonomous driving research. It provides a flexible platform for testing and training RL agents in high-speed highway phases.

To simulate realistic faults, we developed a custom Fault Injection Wrapper Function that dynamically introduces faulty observations, missing data, and action disturbances. In the second phase, the environment was modified to simulate sensor and actuator failures by implementing a Fault Injection Wrapper Function (adding faulty observations, missing data, and faulty actions). The faulty observations were computed as shown in Equation 1.

$$\tilde{o}_t = (o_t + \mathcal{N}(0, \sigma)) \cdot M \tag{1}$$

where M represents missing data probability. Faulty actions were perturbed as illustrated in Equation 2.

$$\tilde{a}_t = \text{clip}(a_t + \delta, 0, a_{\max}) \tag{2}$$

for discrete action spaces, where $\delta \in \{-1, 1\}$ represents a random discrete shift, and for continuous action spaces, shown in Equation 3, where $\mathcal{N}(0, \sigma)$ represents Gaussian noise.

$$\tilde{a}_t = \text{clip}(a_t + \mathcal{N}(0, \sigma), a_{\min}, a_{\max}) \tag{3}$$

In addition to the implemented processes in the previous phases, the third phase introduces KFAE. To ensure the agent's robustness, fault injection was applied to both observations and actions, as described earlier. Additionally, an autoencoder-Kalman filter mechanism was introduced to process faulty observations before feeding them into the model. The autoencoder performed dimensionality reduction and denoising using an encoding function as shown in Equation 4:

$$z_t = f_{\text{enc}}(o_t) \tag{4}$$

where z_t is the latent representation of the original (possibly faulty) observation o_t. The Kalman filter then smoothed the latent variables using the update rule as shown in Equation 5:

$$\hat{z}_t = K(z_t) = Az_{t-1} + Bu_t + w_t \tag{5}$$

where A and B are transition matrices, u_t represents control inputs, and w_t is process noise. The smoothed state $\hat{z}_t$ was then used for policy learning.

By integrating these three phases, our methodology systematically evaluates the RL agent's resilience against sensor and actuator faults.

IV. EXPERIMENTAL SETUP

This section details the evaluation methodology, including environment modifications and training parameters. The goal is to assess the robustness and adaptability of the proposed KFAE framework compared to the baseline environment. To systematically assess the validity of RL agents under varying environmental conditions, this study follows a structured multi-phase evaluation approach. The first phase establishes a baseline in DE, where the agent operates under ideal conditions. The second phase introduces NE, incorporating sensor and actuator faults to analyze the impact of noise on learning performance. Finally, the last phase applies adaptive

filtering techniques to mitigate the effects of these faults. This structured approach enables a direct comparison of how each phase influences decision-making stability in RL-based autonomous driving.

The experiments were conducted in the *highway-fast-v0* environment [13], an open-source simulation designed for autonomous driving research. To ensure consistency, the environment was modified as follows:

- Episode Length: Extended to 200 steps per episode.
- Collision Handling: A penalty of -1 reward was applied for collisions.
- Termination Criteria: Off-road termination was enabled.
- The faults were applied dynamically per episode and randomly varied in duration (200–500 steps) but only affected the episode for its maximum length of 200 steps.
- 80% of missing data probability.

The RL agent was trained using PPO with a Multilayer Perceptron (MLP) policy. The used hyperparameters are depicted in Table I. During evaluation, the number of steps till collision was recorded across multiple trials to analyze agent stability under different conditions.

TABLE I. Training Parameters for PPO with MLP Policy

Parameter	Value
Training Steps	20,000
Hidden Layers	2 fully connected layers
Neurons per Layer	256
Episodes for Evaluation	50

To analyze the performance of the proposed KFAE model, the following metrics were used:

- Episode Reward Mean: Measures accumulated rewards over time, reflecting learning efficiency.
- Steps Till Collision: Evaluate system robustness by determining how long an agent can sustain operation before failure.

These metrics were collected to provide insights into KFAE's ability to observe the impact of noisy and faulty environments.

V. PERFORMANCE EVALUATION

The episode reward means and steps till collision reflect the critical insight for the system analysis regarding learning efficiency and stability. To stand out the agent's ability to make decisions under uncertain environments, we conduct the accumulated rewards over time where a higher reward means indicates effective learning and adaptation. On the other hand, to expose the stability and resilience in dynamic environments, the steps till the collision metric measured the duration before failed to reach the end of the episode or collisions. Further analysis of these metrics will provide a clear view of the proposed model compared to the baseline case concerning mitigating noise and improving system performance.

Figure 2 illustrates the episode reward mean across different phases. The x-axis represents the step count, while the y-axis corresponds to the episode reward mean. This comparison evaluates how reward accumulation evolves under varying environmental conditions.

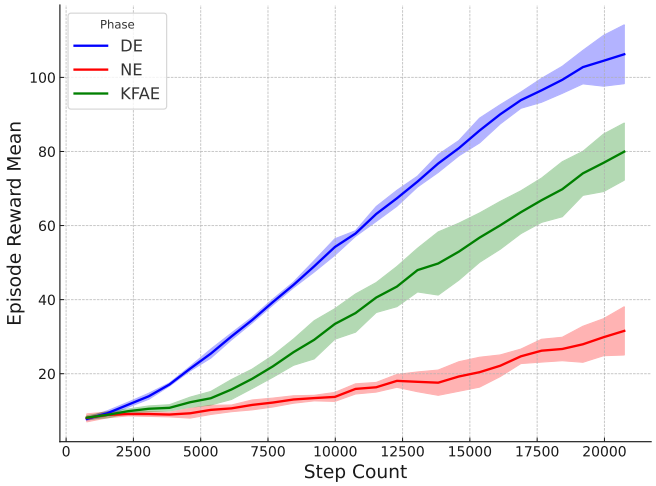

Figure 2. Comparison of episode reward mean across phases

TABLE II. Comparison of improvement for episode reward mean across phases

Metric	DE	NE	KFAE
Avg. Episode Reward	1.234	0.355	0.821
Comparison with DE	–	-71.2%	-33.4%
Comparison with NE	–	–	+131.5%

The property of optimal learning conditions without interference on the DE phase (blue line) has achieved the highest episode reward mean. Compared to the NE phase (red line) which shows a 71.2% decrease in overall reward, as shown in Table II. This confirms that environmental noise significantly disrupts reward optimization and system stability. The substantial gap between NE and DE highlights the negative impact of noisy conditions on learning efficiency and performance consistency.

The proposed KFAE model (green line) consistently outperforms NE, achieving a 131.5% improvement over NE in episode reward means, as quantified in Table II. This significant enhancement confirms that KFAE effectively mitigates the effects of noise, ensuring a more stable and reliable learning process. Despite operating in a dynamic environment, KFAE only falls 33.4% behind DE, demonstrating its ability to approximate near-optimal learning performance while handling noise-induced fluctuations.

A key takeaway from Figure 2 and Table II is that KFAE significantly reduces reward instability compared to NE, suggesting that the Autoencoder-Kalman Filter framework enhances learning adaptability and fault tolerance. The narrower performance gap between KFAE and DE reinforces its potential as a viable alternative for optimizing decision-making in noisy environments. These findings validate KFAE as a robust congestion control mechanism, capable of sustaining high reward accumulation and improving reinforcement learning efficiency in challenging network conditions.

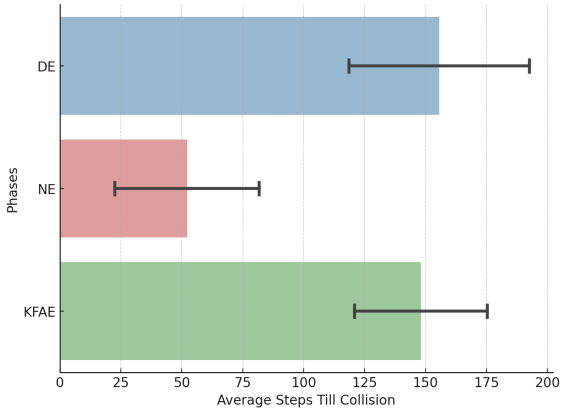

Figure 3. Comparison of average steps till collision across phases

TABLE III. Comparison of improvement for average steps till collision across phases

Metric	DE	NE	KFAE
Avg. Steps	155.6	52.1	148.1
Comparison with DE	–	-66.5%	-4.8%
Comparison with NE	–	–	+184.1%

The results presented in Figure 3 and Table III compare the average steps till collision across different phases: The horizontal bar plot provides a clear visualization of operational stability, with error bars indicating standard deviation across multiple trials, while Table III quantifies the performance differences with exact numerical values.

A key observation is that DE achieves the highest steps till collision, with an average of 155.6 steps indicating that in an ideal, interference-free environment, the system maintains stability for extended periods. This outcome is expected, as DE operates under optimal conditions without the influence of noise or external disruptions. In contrast, NE exhibits significantly fewer steps till collision, averaging just 52.1, representing a 66.5% decrease compared to DE. This confirms that environmental noise severely impacts system performance, leading to increased packet loss, instability, and premature collisions.

The proposed KFAE model significantly outperforms NE, achieving an average of 148.1 steps till collision, which translates to an 184.1% improvement over NE. The green bar in Figure 3 confirms that KFAE sustains operation for a considerably longer duration than NE, reinforcing its adaptability and fault-tolerant nature. Despite operating under non-ideal conditions, KFAE closely approximates DE's performance, with only a minor 4.8% reduction compared to DE. The Autoencoder-Kalman Filter improves robustness, stabilizes learning, and mitigates performance degradation under faults and uncertainties.

A notable trend is the relatively small variance in KFAE compared to NE, suggesting that KFAE provides more consistent and predictable performance across different trials. This is further supported by Table III, which shows that while NE struggles with environmental instability, KFAE maintains nearly the same level of performance as DE. These findings validate KFAE as a highly adaptive RL framework, capable of improving system resilience in dynamic and noisy environments.

Figure 3 and Table III confirm KFAE's effectiveness in mitigating noise and ensuring stable, fault-tolerant RL. The performance gap between NE and KFAE highlights the benefits of adaptive filtering for reliable decision-making in unpredictable conditions.

VI. CONCLUSION

To eliminate sensor noise and faulty actions, we propose KFAE, a hybrid RL framework combining PPO with adaptive filtering to enhance fault tolerance in noisy environments. KFAE de-noises observations, corrects action faults, and improves decision reliability. These experiments confirmed that KFAE outperforms NE, with a 131.5% gain in episode reward mean and an 184.1% increase in steps till collision, while closely approximating DE with minor deviations. These results confirm KFAE's effectiveness in stabilizing RL performance. This study highlights the benefits of adaptive filtering in fault-tolerant RL and suggests future research on real-world applications, alternative filters, and computational efficiency.

REFERENCES

[1] Y. Gu, Y. Cheng, C. P. Chen, and X. Wang, "Proximal policy optimization with policy feedback," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4600–4610, 2021.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[3] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[4] T. Basar, "A new approach to linear filtering and prediction problems," 2001.

[5] H. Durrant-Whyte *et al.*, "Introduction to estimation and the kalman filter," *Australian Centre for Field Robotics*, vol. 28, no. 3, pp. 65–94, 2001.

[6] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," in *Machine Learning for Data Science Handbook* (L. Rokach, O. Maimon, and E. Shmueli, eds.), Springer, Cham, 2023.

[7] S. Liang, Z. Pan, w. Liu, J. Yin, and M. de Rijke, "A survey on variational autoencoders in recommender systems," *ACM Comput. Surv.*, vol. 56, June 2024.

[8] D. Shen, Y. Ma, G. Liu, J. Hu, Q. Weng, and X. Zhu, "Dynamical variational autoencoders and kalmannet: New approaches to robust high-precision navigation," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-1/W2-2023, pp. 1141–1146, 2023.

[9] K. Xiong, C. Wei, and H. Zhang, "Q-learning for noise covariance adaptation in extended kalman filter," *Asian Journal of Control*, vol. 23, no. 4, pp. 1803–1816, 2021.

[10] E. Marchesini and A. Farinelli, "Centralizing state-values in dueling networks for multi-robot reinforcement learning mapless navigation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4583–4588, IEEE, 2021.

[11] J. Marino, A. Piché, A. D. Ialongo, and Y. Yue, "Iterative amortized policy optimization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15667–15681, 2021.

[12] L. Chen, Y. He, F. R. Yu, W. Pan, and Z. Ming, "A novel reinforcement learning method for autonomous driving with intermittent vehicle-to-everything (v2x) communications," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 6, pp. 7722–7732, 2024.

[13] F. Foundation, "Highway-env: A high-speed autonomous driving simulator." https://highway-env.farama.org/quickstart/, 2023. Accessed: March 10, 2025.