Fear Field: Adaptive constraints for safe environment transitions in Shielded Reinforcement Learning

Haritz Odriozola-Olalde[1,2,*], Nestor Arana-Arexolaleiba[2], Maider Zamalloa[1], Jon Perez-Cerrolaza[1] and Jokin Arozamena-Rodríguez[1]

[1]*Ikerlan Technology Research Centre, José María Arizmendiarrieta 2, Arrasate-Mondragón, Gipuzkoa, 20500, Spain*

[2]*Mondragon Unibertsitatea, Loramendi 4, Arrasate-Mondragón, Gipuzkoa, 20500, Spain*

Abstract

Shielding methods for Reinforcement Learning agents show potential for safety-critical industrial applications. However, they still lack robustness on nominal safety, a key property for safety control systems. In the case of a significant change in the environment dynamic, shielding methods cannot guarantee safety until their inherent dynamics model is updated to the new scenario. The agent could reach risky states because the model cannot predict well. These situations could lead to catastrophic outcomes, such as damage to the cyber-physical system or loss of human lives, which are not allowed on safety-critical applications. The novel method presented in this paper, Fear Field, replicates human behaviour in those scenarios, adapting safety constraints whenever a drastic environmental change is introduced. Fear Field reduces safety violations by one order of magnitude compared to an RL agent implementing only a shield.

Keywords

Reinforcement Learning, Shielding, Adaptive constraints, Robustness, Safe AI

1. Introduction

The design of controllers for autonomous systems has emerged in a new era with the remarkable evolution of Machine Learning (ML). Techniques such as Supervised Learning, Unsupervised Learning and Reinforcement Learning have shown extraordinary value both for their great adaptability to highly complex problems and reduced computational cost in inference.

One of the emerging techniques within ML is Reinforcement Learning (RL), linked to optimal control theory [4]. In RL, an agent interacts with its environment through the paradigm of trial and error, and exploration and exploitation [20]. Depending on the performance shown by the agent in a given task, it receives a reward. During a series of trial and error, the agent learns how to maximise the accumulated discounted reward. The resulting agent is expected to be able to select the optimal action for a given state, which will be called the policy.

Safe Reinforcement Learning (SRL) research methods to minimise unsafe situations that can be risky for the cyber-physical system, during the exploration process carried out during learning and subsequent execution process [1, 5, 7]. Among the different methods proposed for this purpose is Shielded Reinforcement Learning. In this method, each action proposed by the agent is checked, so the Shield only allows the action to be executed if the environment transit to a safe state. For this, most of the proposed methods use a model of the environment.

The shielded RL algorithms proposed in the literature focus mainly on nominal safety, while functional safety is relegated to a second level. The nominal safety focuses on making safe logical decisions, and it is part of the wider functional safety considerations that must also consider underlying hardware and software failures. This work focuses on nominal safety, and the complete functional safety considerations must be studied in future research work.

A drawback of the Shielded RL methodology is that an eventual change in the dynamics of the environment can make its transition model obsolete; therefore, the Shield would not act correctly until the model is updated to the new environment. Another scenario where the agent may take risky actions due to an outdated model is transferring a policy trained in the simulator to the target or real scenario, known as the Sim2real gap [9, 13].

In order to improve the safety of Shielded RL methods, this paper presents the Fear Field (FF) framework, which aims to reduce the number of unsafe states reached when a significant modification is produced in the environment dynamic. As humans do, when faced with

Macao'23: AISafety-SafeRL 2023 Workshop (IJCAI), August 19–21, 2023, Macao, SAR, China

[*]Corresponding author.
hodriozola@ikerlan.es (H. Odriozola-Olalde); narana@mondragon.edu (N. Arana-Arexolaleiba); mzamalloa@ikerlan.es (M. Zamalloa); jmperez@ikerlan.es (J. Perez-Cerrolaza); jarozamena@ikerlan.es (J. Arozamena-Rodríguez)
0000-0001-6597-0225 (H. Odriozola-Olalde); 0000-0002-3305-8108 (N. Arana-Arexolaleiba); 0000-0003-3872-9908 (M. Zamalloa); 0000-0001-6389-648X (J. Perez-Cerrolaza); 0000-0002-5687-7325 (J. Arozamena-Rodríguez)

previously unknown scenarios, Fear Field acts cautiously by dynamically adapting the constraints to the situation.

OpenAI Gym's [3] slightly modified Frozen Lake benchmark and the open-source Skrl library [19] are used for testing and validating the Fear Field framework. Frozen Lake benchmark consists of a tile-based discrete environment, where a robot has to learn the path to reach the goal while avoiding holes (unsafe states) in its way. Initially, the RL agent is trained with normal environment conditions, where each action makes the robot move only one tile. When the environment's dynamic changes, an action taken leads the robot to move an additional tile following the same direction. The agent has been trained using tabular Q-Learning, a temporal-difference-based algorithm. It has been observed that an agent trained with only Shielded RL avoids the holes while there are no changes in the environment dynamic, but it fails just when the environment dynamic changes. After some episodes, it adapts to the new situation. Thus, an agent trained with shielded RL does not guarantee to fulfil safety constraints after the environment dynamic changes.

The contributions of this work can be summarised as follows:

- The Fear Field framework is proposed, which identifies the change in the environment, adapts the imposed safety constraints to the new scenario and acts accordingly.
- The Fear Field framework is defined and validated in the OpenAI Gym's Frozen Lake environment.
- The robustness of the Fear Field to significant environmental changes and the improvement over a Shielded RL-based control system safety is evaluated in an experimental setup.

The rest of the paper proceeds as follows. Section 2 briefly defines the Markov Decision Process, Q-Learning and Shielded Reinforcement Learning. In section 3, related works to the problem studied in this work are resumed. In section 4, constraints and Shield implementation are defined. The Fear Field framework is defined in section 5. In section 6, the experimentation methodology is given. Following, results obtained in testing are discussed in section 7. Finally, in section 8, the main conclusions and future work are summarised.

2. Preliminaries

Reinforcement Learning is one of the most popular techniques of Machine Learning, where an agent that interacts with its environment learns through the paradigm of trial-error and exploration-exploitation [23]. The mathematical idealisation of Reinforcement Learning algorithms is the Markov Decision Process (MDP).

2.1. Markov Decision Process

A Markov Decision Process is called a sequential decision-making system, in which the action a taken in the state s determines the immediate reward r, the next states to transit, and the future rewards to receive. Let it be a tuple $\langle S, A, P, R, \gamma \rangle$, where S is the space of states, A is the space of actions, $P : S \times A \times S \rightarrow [0, 1]$ is the probabilistic transition function associated to the environment, $R : S \times A \times S \rightarrow \mathbb{R}$ determines the reward function and the discount factor γ determines the present value of future rewards [1].

2.2. Reinforcement Learning

Several learning techniques are found within Machine Learning, such as Supervised Learning, Unsupervised Learning, Imitation Learning and Dimensionality Reduction [4]. One technique that has recently been showing potential is Reinforcement Learning (RL). RL differs from other subfields of ML in that it does not require a supervisor or complete models of the environment [20]. It is, therefore, of interest in complex problems that span different engineering domains and where the only way to learn the properties of the environment is to interact with it. RL algorithms are primarily linked to optimal control theory [4], as they are based on the interaction of an agent with its environment through the paradigm of trial-error and exploitation-exploration [23]. Reinforcement Learning use ranges from object manipulation control problems to the compression of search-based schedulers or optimal controllers, where neural networks reduce their computational cost [2].

The learning process of the agent over the environment, i.e. mapping the states to the actions [16], is carried out because the agent receives a *reward* for each action performed and ultimately tries to maximise the sum of the rewards received [8, 23]. Such interaction allows the agent to learn and ideally generalise the knowledge gained to deal with inexperienced situations, obtaining a policy determining the agent's behaviour. The policy can range from a simple relationship, such as a lookup table, to complex relationships that require high computation, being in general stochastic relationships. The policy can be identified as the core of an RL agent because it determines the behaviour of the agent [20].

While the reward is a short-term indication of how good an action performed by the agent is, the Q-value function $Q_\pi(s, a)$ defines the expected discounted accumulated reward following policy π after taking action a in state s. Because this long-term estimation is computationally expensive, the correct method choice is considered a key component of most RL algorithms [20].

2.3. Q-Learning

Q-Learning is a temporal-difference-based control algorithm in which the Q-value function $Q(s, a)$ directly approximates its optimal value q_*, maximising it regardless of the policy (off-policy) being applied. It should be noted that, even so, the policy determines which action-state is visited and updated at each instant [23]. For a time instant t the Q-Learning algorithm is defined as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \Big[r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \Big] \tag{1}$$

2.4. Shielded RL

The reactive Shielded RL methods are among the various methods proposed to ensure the safety of the controlled cyber-physical system. Shielded RL was proposed by Alshiek et al. [1]; it defines the use of a shield, which acts as a filter to block those actions that transit the environment to an unsafe state (See Figure 1).

Through safety specifications a safety automaton $\varphi^s = (S, s_0, P, S_{safe})$ is defined where s_0 correspond to the initial state and S_{safe} is the set of safe states [1, 12].

The Shield monitors the action a_t proposed by the agent at each time step. It checks the expected state s_{t+1} after applying an action a_t in state s_t. If the expected state s_{t+1} is unsafe with respect to φ^s, the Shield will block the action and offer another safe action using the safe policy π^s. This safe policy π^s is defined in advance using particularly severe constraints.

Another aspect to consider is whether the Shield should penalise the agent's proposal of an unsafe action in the reward. As summarised by Odriozola-Olalde [17], several authors [6, 11] see favourable to introducing a penalty so that a bias is generated in the agent's policy that reduces the need for shield intervention in the future. Still, other authors [1, 2, 14, 21] find the introduction of the penalty mentioned above detrimental.

3. Related Work

Although Shielded RL is a relatively recently developed framework, many studies [1, 2, 21] propose different methodologies for implementing shielding in decision-making control systems. These methodologies show promising results for ensuring nominal safety but lack experimentation in environments that may suffer drastic dynamic changes [17]. This situation emphasises the analysis of the robustness of proposed controllers in environments that may suffer dynamic changes.

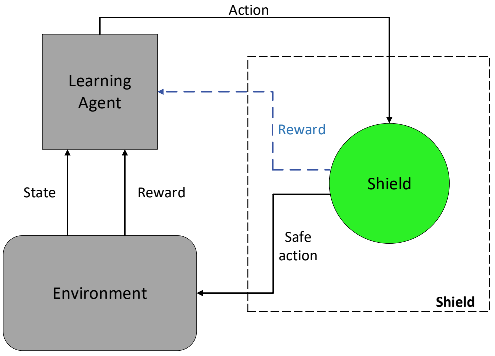

Figure 1: Shielded Reinforcement Learning schematic.

Zhu et al. [22] consider the effectiveness of their proposed method in different environments. Starting from the shield defined for an initial environment, they propose the synthesis of a new shield for a modified environment. But they do not analyse what happens to the agent during the time needed to synthesise the new Shield, as the agent is partially safe or not guaranteed to be safe.

Bastani [2] modifies the cart-pole environment increasing the time horizon of the controller to demonstrate that Model Predictive Shielding (MPS) still fulfils the safety guarantees obtained on the initial environment. Although, the cumulative reward obtained decreases significantly, reducing the performance of the controller at the cost of assuring safety.

Thumm and Althoff [21] proposed method, *Failsafe Planner*, is the only one that considers Functional Safety standards in the Shielded RL field. They propose a framework to guarantee the *speed and separation monitoring* for human-robot interaction environments defined in DIN EN ISO 10218-1 2021, 5.10.3. A combination of a low-frequency RL agent and a formal high-frequency safety verification algorithm is used to synthesise a safety shield. Even though Failsafe Planner is able to avoid all human-robot collisions, it has a goal-reaching success rate of 65%. Modelled human movements on experimentation are quite limited, so safety is not guaranteed in more complex environments. Also, they do not study how Failsafe Planner could behave in a dynamic changing environment, thus the robustness of Failsafe Planner is not assured.

Lazarus et al. [15] propose a similar approach to Fear Field for Runtime Safety Assurance (RTSA) in order to ensure the nominal safety of an Unmanned Aerial Vehicle (UAV). Using Safety Envelopes, a subspace of the state space defined through safety constraints, they shrink it specifying a distance δ. While the UAV is inside the shrunk subspace, a π_n black-box nominal policy is used. But once the UAV exits the shrunk subspace, a π_r simple and safe recovery policy is deployed. A Reinforcement Learning agent is trained in order to choose when to swipe from one policy to another. The drawbacks of this proposition are that δ is a hyperparameter that may be difficult to tune, it has no adaptability at all and that the recovery policy π_r consists of turning off the UAV rotors and deploying a parachute.

Therefore, most works do not study how they perform in significantly changing environments, so it is necessary to study the lack of safety guarantees that could be shown when the environment dynamic changes and, thus, during the time the new shield becomes available. Also, the verifiability of the proposed algorithms must be studied in order to obtain formal safety guarantees.

4. Problem Setup

This section defines the techniques used to define the constraints and the methodologies used to synthesise the Shield.

4.1. Constraints

In this work, it has been decided to establish the safety constraints using a tabular table or constraint table (4). The table has the exact dimensions as the GridWorld used in the experimentation.

$$\mathcal{T}(s) = \begin{cases} 0 & s \notin S_{safe} \\ 1 & otherwise \end{cases} \tag{2}$$

where $\mathcal{T}(s)$ is the constraint table, s is the state and S_{safe} is the aforementioned set of safe states.

This implies that it is necessary to know in advance the constraints of the environment, which sometimes cannot be known due to the complexity of the cyber-physical system to be controlled and changes in the environment's dynamics. In this work, it is assumed that S_{safe} and S_{unsafe} are known, as it is a reach-avoid problem.

4.2. Shield

The shield implementation contains a model of the dynamics of the environment $P(s_{t+1}|s_t, a_t)$ [2, 6, 11, 14, 21]. The model input is the proposed action, and the output is the predicted state that the robot will reach on the next state.

Each time a deviation further than a predefined threshold λ is detected between the next state prediction and the real one $\left|s_t - s_t^{pred}\right| > \lambda$, the environment dynamic model is updated. This threshold λ is defined to avoid that insignificant deviation that the model can have relative to the real environment dynamic will affect in runtime. New data $P(s'|s, a)$ is first collected in n_{buffer}, and then it is used to update the model. This process is repeated until the predicted state deviation is lower than the threshold $\left|s_t - s_t^{pred}\right| \leq \lambda$ in all steps of T_{train} period, i.e. until no meaningful difference between the movement predicted by the model and the actual movement of the agent is found.

At each step, the shield studies the feasibility of each possible action, predicting the next state s_{t+1} using the dynamics model and observing if it belongs to the ensemble of safe states S_{safe}. Following, it orders the actions by its expected reward. The selected action to be applied a_t is selected if and only if it is safe and its expected reward is the highest of all safe actions. The environment transition is predicted over a finite time horizon h:

$$s_{t+h}^{pred} \leftarrow P(s_{t+h}|s_t, s_{t+1}, ..., s_{t+h-1}, a_t) \tag{3}$$

5. Fear Field

As humans adapt the caution measures taken in our activities according to our confidence and knowledge of the environment at a specific moment, Fear Field proposes adapting the safety constraints depending on the Shield's confidence in the environment's model accuracy. The difference between the model's predicted state and the real one quantifies the model's accuracy.

In an initial environment, for an agent with a shield and where there is an accurate model of the environment, an action a_t taken in the state s_t transits the environment to state s_{t+1}. In this case, the Shield can predict whether this transition is safe since the associated model matches the initial environment. Suppose now that the dynamics of the environment have changed so that the model associated with the Shield does not match reality.

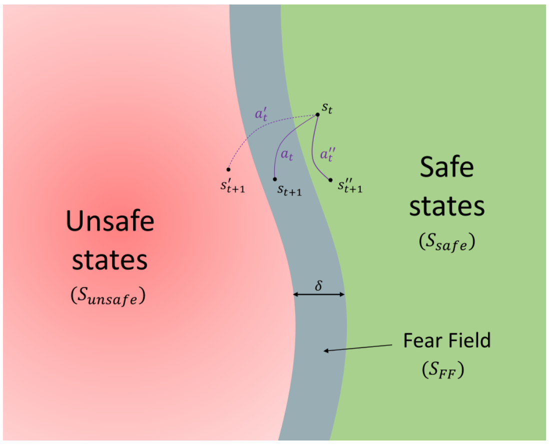

Figure 2: Fear Field: Adaptive constraints for variable environments.

In the second case, the same action a_t, let's call a'_t, taken in state s_t causes the environment to transit to a state s'_{t+1} which can be an unsafe or a hidden unsafe state [17].

Therefore, the use of the Fear Field (See Figure 2) is proposed, which is reflected as an extension of the safety constraints defined in the problem. Fear Field has been designed in order to support the Shield when the model predictions are not accurate. For this purpose, a new constraint table is defined such that:

$$\mathcal{T}'(s) = \begin{cases} 0 & s \notin S_{safe} \\ 0.5 & s \in FearField \\ 1 & otherwise \end{cases} \tag{4}$$

The width of the Fear Field $\delta_t(s) \propto \left| s_{t-1} - s^{pred}_{t-1} \right|$ is directly proportional to the distance between the predicted state s^{pred}_t and the real state s_t. Thus, the width $\delta(s)$ is a dynamic variable linked to the difference between the predicted state and the real one. In the case of $\delta(s)$ having different values through timesteps, the highest value is taken as the worst-case scenario. Fear Field states S_{FF} is the set of states where each state s_{FF} is within a maximum Euclidean distance $\delta(s)$ from each unsafe state such that:

$$s \in S_{FF} \Leftrightarrow \exists s_{unsafe} : |s - s_{unsafe}| \leq \delta(s) \tag{5}$$

Once the new constraints have been defined, the shield analyses if the action $a`_t$ would transit the environment to either an unsafe state or a state that is part of the Fear Field $\mathcal{T}'(s_{t+1}) = 0.5$. Even though the model is outdated due to that the environment dynamic has changed, the Shield will predict the transition to the state s_{t+1} which is now within Fear Field so that it will block that action a'_t and will look for an action a''_t that does not lead to an unsafe state or a state within the Fear Field, i.e. it will look for $\mathcal{T}'(s''_{t+1}) = 1$, and has the highest Q-Value Q(s,a).

Only when the model is retrained, and the difference between the model and the reality is non-existent after n_{Steps} steps, the width δ of the Fear Field will be zero again. It is necessary to be noted that during the Fear Field intervention, the agent keeps following the same previous policy and gathers $n_{Dataset}$ steps data as a retraining dataset. The algorithm used for Fear Field implementation is shown on Algorithm 1, which is executed continuously, and the schematic representation is shown in Figure 3.

The Fear Field algorithm works as follows: In every iteration, the previous timestep state reached and the predicted state are compared (Line 2). If they are not equal, Fear Field's width $\delta_{t+1}(s_t)$ is calculated using the difference existing between the state reached and the predicted state (Line 3). Using $\delta_{t+1}(s_t)$, a new constraint table $\mathcal{T}'(s)$ is generated (Line 4). Since the predicted state does not match the state reached, the environment dynamic model is outdated. If $n_{Dataset}$ number of steps has been passed from the last timestep $t_{LastTrain}$ that the model was updated (Line 5), then, the model is updated with the new dataset (Line 6) and $t_{LastTrain}$ is restored to the current timestep (Line 7).

Algorithm 1 Fear Field algorithm

```
1:  Given (Environment Dynamic Model, $\mathcal{T}(s)$, $n_{Dataset}, n_{Steps}$)
2:  if $s_t \neq s^{pred}_t$ then
3:      Calculate $\delta_t(s_t)$ value
4:      Generate $\mathcal{T}'(s)$ s.t. $\delta_t(s_t)$
5:      if $t = t_{LastTrain} + n_{Dataset}$ then
6:          Update environment dynamic model
7:          $t_{LastTrain} = t$
8:      end if
9:  else if $(s_{t-n_{Steps}-1}, ..., s_{t-1}) = (s^{pred}_{t-n_{Steps}-1}, ..., s^{pred}_{t-1})$ then
10:     Load $\mathcal{T}(s)$
11: end if
12: Check the safety of all actions
13: Take highest Q(s,a) safe action
14: Apply the action $a_t$ to the environment
15: Save last $n_{Steps}$ steps data on buffer
```

Only when the model is updated such that it is capable of matching all last n_{Steps} steps of state reached and predicted states (Line 9), the initial constraint table $\mathcal{T}(s)$ is loaded again. This allows the shield to check the safety of the actions proposed by the agent (Line 12) and choose the action that has the highest Q-value and is safe (Line 13). Finally, once the action is applied to the environment and it transits to the new state s_{t+1} (Line 14), the last n_Steps of states reached and predicted states are saved in a buffer (Line 15).

Note that the Fear Field algorithm is being executed when the RL agent is trained, and also when it is in execution; thus, the model updating process is conducted independently of the agent's learning process.

6. Experiments

As mentioned above, the benchmark environment used for validating the Fear Field method was an OpenAI GridWorld [3]. Specifically, a modified version of Frozen Lake was used. This benchmark environment consists of a robot that starts in a box in one corner of the two-dimensional state space and must reach the goal in the opposite corner. If the robot falls into a hole (S_{unsafe}), it has to start the episode again. So, the Frozen Lake environment is a reach-avoid problem. The environment grid size is 10x10 blocks, with 16 of them being holes. The

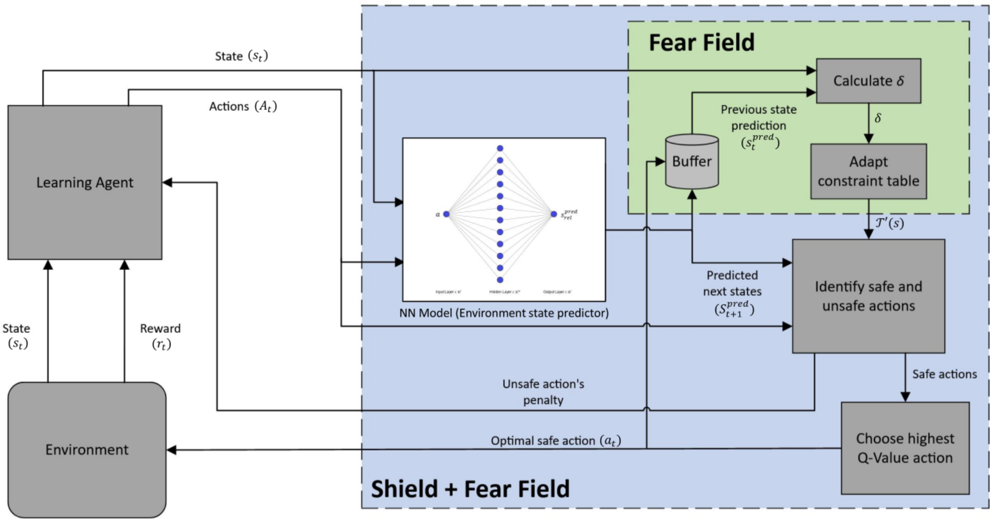

Figure 3: Schematic of Fear Field implementation on Shielded RL.

unique feature of the modified version of Frozen Lake is that periodically, the world is slippery, meaning that the robots will move one additional square for the same action. It is necessary to be noted that the environment used for experimentation is deterministic.

The open-source modular library Skrl, which integrates several RL algorithms and supports OpenAI Gym, has been used to implement the benchmark [19]. The software code provided by Skrl has been modified to incorporate both the Shield and the Fear Field. Q-Learning algorithm has been used as an RL learning algorithm, specifically a tabular Q-learning algorithm, as it matches the discrete nature of the benchmark environment.

In order to the environment dynamic model, a Neural Network (NN) based model has been chosen due to its capacity to adapt to the changes in the dynamics of the environment, its accuracy and the reduced computational cost in inference. Also, training over the past model reduces the computational cost associated with the relearning process [22].

The output of the model has been defined as the relative movement of the robot. This way, the required minimum NN topology has been reduced to one hidden layer with 12 neurons on it. Also, working with relative movement helps to reduce the problem's complexity and, consequently, the needed data and training time. The finite time horizon used is $h = 1$. This way, the shield is capable of predicting the next step state adding to the actual state the predicted relative movement:

$$s_{t+1}^{pred} = s_t + NN(a_t) \tag{6}$$

It is necessary to be noted that grid world border states movement is not taken into account for retraining the NN model, and also while applying the relative movement calculated by the NN to the current state, it is taken into account if the predicted state will be out of the border.

The testing set, composed of 50 trials of 700 episodes each, has been performed to obtain a significant number of results in order to validate the algorithm presented in this paper. Each episode is terminated if the agent falls into a hole, reaches the goal or if it takes more than 100 steps. In the first 350 episodes (Non-slippery), the robot moves only one square for each action. The following 150 episodes correspond to the slippery world (Slippery). Finally, in the last 150 episodes, the robot moves only one square again for each action (Non-slippery). The testing procedure consists of analysing the performance and nominal safety level of the non-shield RL algorithm, a shielded RL baseline, and the Fear Field integrated shielded RL algorithm. The first one (Q-Learning) is the tabular Q-Learning [20] without any safety measures. The second (Shield) corresponds to the previous Q-Learning algorithm with a shield integrated. Finally, the third one (FF) is based on the second one but incorporates the Fear Field framework.

The values of the hyperparameters are shown in Table 1. If a mismatch is detected between the state predicted by the NN model and the actual state, 1000 samples are collected, and the network is retrained using batches of 6 samples for 100 epochs. The equation used to calculate the width $\delta(s)$ of Fear Field is the following one:

$$\delta_{t+1}(s) = \left| s_t - s_t^{pred} \right| \tag{7}$$

Table 1
Hyperparameters values used in experimentation.

	Hyperparameter	Value
Agent	ϵ	0.4
	ϵ decay	-5×10^{-5} per timestep
	γ	0.999
	α	0.4
NN training	Topology	1-12-1
	Buffer size	1000
	Batch size	6
	Epochs	100
	Learning rate	0.001
	Optimiser	Adam
	Activation function	ReLU
Fear Field	$n_{Dataset}$	1000
	n_{Steps}	800
	λ	0
Rewards	Step taken	-0.01
	Hit a wall	-1
	Fall in hole	-100
	Reach goal	100
	Action blocked by the shield	-10
	Action blocked by Fear Field	-2

7. Results

The testing results are shown in Figure 4. Three phases are shown in the x-axis, corresponding to the environment state: Slippery or non-slippery. Also, a pink interval is shown on the y-axis, indicating an unsafe state has been reached. It is observed that both the Shield and Fear Field methods accelerate the convergence of the agent by almost five times in the first training process. This is because the Shield allows the agent to explore safely, significantly improving the number of steps performed in each episode, thus gaining more knowledge of the environment per episode.

Another significant advantage of shield-assisted learning is that no safety breaches (red zone) are committed during the initial (first 350 episodes) learning process. In the Q-Learning test, the robot persists in an insecure state for the first 150 episodes.

In the slippery period (episodes 350-550), it can be observed that both Q-Learning and Shielded Q-Learning suffer from reaching unsafe states in the episodes immediately after the change made in the environment's dynamic (episode 350). As hypothesised in the shielded reinforcement learning review [17], the prediction made by the model does not match the actual movement made by the agent, so it cannot predict the state to which the agent will transit correctly, and the Shield will not block the unsafe action; obtaining a 0.0156% probability of reaching an unsafe state. As can be seen, the agent adapts to the new environment over time because the model associated with the shield is updated. Despite this, the average number of unsafe states reached compared to Q-Learning without the Shield is approximately 50 times lower (see Table 2).

For the Fear Field method (FF), it can be observed that the transition from the normal environment to the slippery one is performed with a significant reduction of the unsafe state visited. However, it should be noted that in some trials, the Fear Field approach still encountered unsafe states (see Table 2). Specifically, in 60% of the trials performed, FF obtained a null number of unsafe states. This is because sometimes the retrained NN model is not accurate enough and therefore fails to predict the state transition, obtaining a 0.00179% probability of reaching an unsafe state in total. Thus, the reason behind visiting unsafe states is the inaccuracies associated with the NN model. Despite this, one order of magnitude is reduced compared to the Shielded Q-Learning case.

One of the main shortcomings of the Fear Field is that after transitioning to a previously unknown environment, the convergence time of the policy is increased. This behaviour is related to the agent being more constrained in taking action and taking fewer risks. Due to that, the previously learned path cannot be taken, so the agent must learn a new path to reach the goal. The learning process to obtain a safe path to the goal can take some episodes to be learned.

Also, no policy convergence has been observed in 2 of the 50 trials performed when the environment changes to non-slippery after being slippery. This phenomenon is due to the model not being updated properly, keeping the robot transiting to an unsafe state constantly. Thus, no useful dataset needed to update the model properly is obtained, and the agent enters a non-ending cycle.

Thus, the dataset obtained does not

8. Conclusions and future work

Shielded Reinforcement Learning is a method of great interest in control and decision-making fields because it drastically reduces the number of unsafe states reached. Since Shield uses a dynamic model of the environment to predict future states that the environment will transit, the Shield's effectiveness is low when faced with changes in the environment's dynamics. This problem persists until the model associated with the Shield adapts to the new environment.

In order to reduce this impact, the Fear Field method has been proposed and validated in this paper. Adapting the safety constraints in proportion to the difference between the model prediction and the real transitions, Fear Field is able to reduce the unsafe states reached by order of magnitude when compared to the Shielded RL

Table 2
Mean unsafe state reached per test.

	Q-Learning	Shield	FF
Mean unsafe states	0,77192%	0.0156%	0.00179%

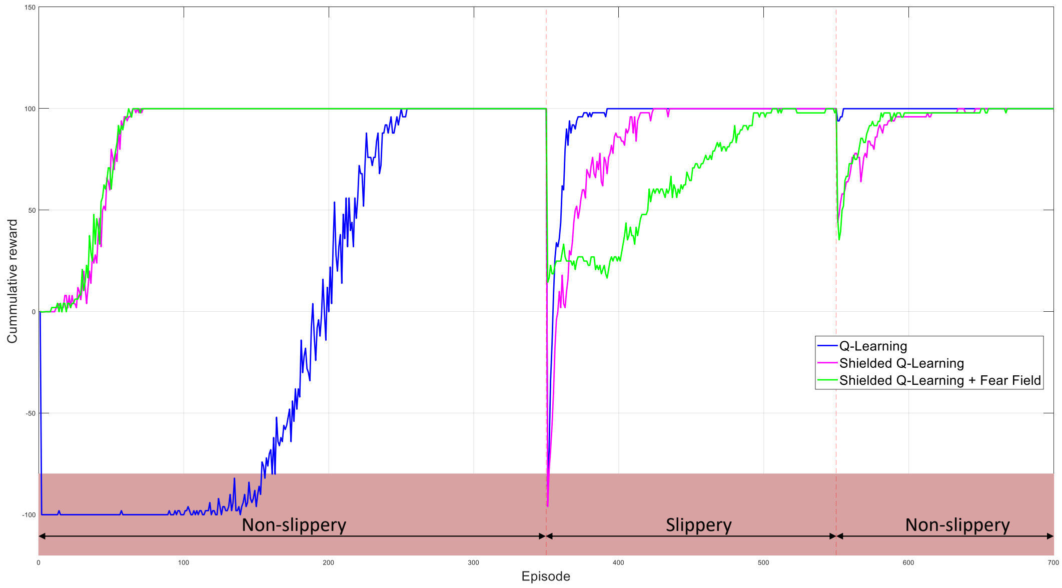

Figure 4: Cumulative reward mean for each episode over 50 trials.

method.

Looking ahead, the retraining of the model needs to be improved to reduce the number of unsafe states further reached. Also, procedures must be developed to address the observed increase in convergence time, such as introducing a small value of exploration rate ϵ when the robot cannot find a new path to the goal.

Regarding the applicability of the Fear Field framework, a use case of an autonomous guided vehicle (AGV) in a warehouse scenario is proposed as future work. The behaviour of the AV will be observed in a scenario with significant changes in the wheels grip, and therefore Fear Field will be tested.

For the experimentation conducted in this paper, a deterministic environment is used. In future work, the effects of the introduction of stochasticity must be studied. the Fear Field framework shows potential for more complex scenarios, where model capacity might be limited in order to capture the environment dynamics perfectly. In this case, the Fear Field framework could be helpful in reducing safety constraint violations.

Shielded RL is an AI-based safety-focused algorithm that must still be developed in compliance with applicable safety standards (e.g., IEC 61508, ISO 5469). Therefore, the integration of methodologies such as Safety Envelopes, certified according to the required safety standard, may be interesting in order to provide the decision-making controller with formal safety guarantees. For many research areas, there are methodologies developed to define the Safety Envelope through safety standards, e.g. Responsibility-Sensitive Safety (RSS) for autonomous driving vehicles [10, 18].

Finally, further research must be conducted to find how the values of the hyperparameters regarding the Shielded RL and Fear Field frameworks affect the safety constraint violation rate.

9. Acknowledgments

We want to thank Alexander Lenail for allowing the use of NN SVG to generate NN topology diagrams.

References

[1] Mohammed Alshiekh et al. “Safe Reinforcement Learning via Shielding”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[2] Osbert Bastani. “Safe Reinforcement Learning with Nonlinear Dynamics via Model Predictive Shielding”. In: *American Control Conference (ACC). IEEE*. 2021, pp. 3488–3494.

[3] Greg Brockman et al. “OpenAI Gym”. In: *arXiv preprint arXiv:1606.01540* (2016).

[4] Giuseppe Carleo et al. “Machine learning and the physical sciences”. In: *Reviews of Modern Physics* 91(4) (2019), p. 045002.

[5] Steven Carr et al. "Safe Reinforcement Learning via Shielding for POMDPs". In: *arXiv preprint* (2022).

[6] Ingy ElSayed-Aly et al. "Safe Multi-Agent Reinforcement Learning via Shielding". In: *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS* 1 (2021), pp. 483–491.

[7] Javier García and Fernando Fernández. "A Comprehensive Survey on Safe Reinforcement Learning". In: *Journal of Machine Learning Research* 16 (2015), pp. 1437–1480.

[8] Mirco Giacobbe et al. "Shielding Atari Games with Bounded Prescience". In: *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS* 3 (2021), pp. 1495–1497.

[9] Andrew Harris and Hanspeter Schaub. "Spacecraft command and control with safety guarantees using shielded deep reinforcement learning". In: *AIAA Scitech 2020 Forum.* Vol. 1 PartF. American Institute of Aeronautics and Astronautics Inc, AIAA, 2020, pp. 386–403.

[10] Ichiro Hasuo. "Responsibility-Sensitive Safety: an Introduction with an Eye to Logical Foundations and Formalization". In: *arXiv:2206.03418* (2022).

[11] Peter He, Borja G León, and Francesco Belardinelli. "Do Androids Dream of Electric Fences? Safety-Aware Reinforcement Learning with Latent Shielding". In: *SafeAI@ AAAI* (2022).

[12] Floris den Hengst et al. "Planning for potential: efficient safe reinforcement learning". In: *Machine Learning* 111.6 (2022), pp. 2255–2274.

[13] Kai-Chieh Hsu et al. "Sim-to-Lab-to-Real: Safe Reinforcement Learning with Shielding and Generalization Guarantees". In: *Artificial Intelligence* 314 (2023).

[14] Nils Jansen et al. "Safe Reinforcement Learning Using Probabilistic Shields". In: *31st International Conference on Concurrency Theory (CONCUR).* 2020.

[15] Christopher Lazarus, James G Lopez, and Mykel J Kochenderfer. "Runtime Safety Assurance Using Reinforcement Learning". In: *AIAA/IEEE 39th Digital Avionics Systems Conference (DASC).* 2020, pp. 1–9.

[16] Islam Nazmy et al. "Shielded Deep Reinforcement Learning for Multi-Sensor Spacecraft Imaging". In: *American Control Conference (ACC)* (2022), pp. 1808–1813.

[17] Haritz Odriozola-Olalde, Maider Zamalloa, and Nestor Arana-Arexolaleiba. "Shielded Reinforcement Learning: A review of reactive methods for safe learning". In: *IEEE/SICE International Symposium on System Integrations.* 2023, pp. 1–8.

[18] Mateusz Orłowski et al. "Safe and Goal-Based Highway Maneuver Planning with Reinforcement Learning". In: *Advanced, Contemporary Control: Proceedings of KKA - The 20th Polish Control Conference.* Springer, 2020, pp. 1261–1274.

[19] Antonio Serrano-Muñoz et al. "skrl: Modular and Flexible Library for Reinforcement Learning". In: *arXiv:2202.03825* (2022).

[20] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction.* Tech. rep. 2018.

[21] Jakob Thumm and Matthias Althoff. "Provably Safe Deep Reinforcement Learning for Robotic Manipulation in Human Environments". In: *International Conference on Robotics and Automation (ICRA).* 2022, pp. 6344–6350.

[22] He Zhu et al. "An inductive synthesis framework for verifiable reinforcement learning". In: *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI).* New York, NY, USA: Association for Computing Machinery, 2019, pp. 686–701.

[23] Zeyu Zhu and Huijing Zhao. "A Survey of Deep RL and IL for Autonomous Driving Policy Learning". In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 14043–14065.