Robust Exploration/Exploitation Trade-Offs in Safety-Critical Applications

Michel Tokic [*,****] **Philipp Ertle** [**,****] **Günther Palm** [*]
Dirk Söffker [**] **Holger Voos** [***,****]

[*] *University of Ulm, Institute of Neural Information Processing, 89069 Ulm, Germany*
[**] *University of Duisburg-Essen, Chair of Dynamics and Control, 47057 Duisburg, Germany*
[***] *University of Luxembourg, Faculté des Sciences, de la Technologie et de la Communication, L-1359 Luxembourg*
[****] *University of Applied Sciences Ravensburg-Weingarten, 88250 Weingarten, Germany*

Abstract:
With regard to future service robots, unsafe exceptional circumstances can occur in complex systems that are hardly to foresee. In this paper, the assumption of having no knowledge about the environment is investigated using reinforcement learning as an option for learning behavior by trial-and-error. In such a scenario, action-selection decisions are made based on future reward predictions for minimizing costs in reaching a goal. It is shown that the selection of safety-critical actions leading to highly negative costs from the environment is directly related to the exploration/exploitation dilemma in temporal-difference learning. For this, several exploration policies are investigated with regard to worst- and best-case performance in a dynamic environment. Our results show that in contrast to established exploration policies like ε-Greedy and Softmax, the recently proposed VDBE-Softmax policy seems to be more appropriate for such applications due to its robustness of the exploration parameter for unexpected situations.

Keywords: Temporal-difference learning, Safety, Autonomous Systems, Learning.

1. INTRODUCTION

To *[...] overcome the practically impossible problem of preidentifying the full range of kinds of situations robots and other agents will get into during normal interaction with their environments, [...] we should [...] seek to build robots, and artificial agents in general, that are autonomous.* Of course, Smithers (1997) suggested this having the complexity and NOT the safety problem in mind, although this statement probably remains true taking also safety issues into account. In many cases, autonomous systems (AS) are intended to collaborate with humans. Hence, those have to be considered as safety-critical systems. Future service robots are a prominent class of such AS. A typical and desired goal for future service robots is to be able to grip and manipulate environmental objects. As shown in former research (Ertle et al., 2010), various new hazards appear, stemming from dangerous interactions between objects being manipulated by the robot. Thus, the safety complexity becomes worse.

Surely, everything humanly possible must be done to mitigate or eliminate risks - as low as reasonably practicable - and finally, remaining risks and benefits have to be carefully and responsibly balanced in order to decide if such service robots can be allowed for personal use. Nevertheless, it is probable that such robotic systems are also incomplete with regard to their safety specifications.

⋆ This work was conducted within the collaborative center for applied research *ZAFH-Servicerobotik*. The authors gratefully acknowledge the research grants of the state Baden-Württemberg and the European Union.

The problem of incompleteness is tackled by researchers using learning algorithms. But what is about the safety of the system when learning algorithms are applied? Reinforcement Learning (RL) is a well known and established methodology enabling to learn behavior based on trial and error. Therefore, this contribution basically reflects on algorithmic aspects when *temporal-difference* learning algorithms are applied. More specifically, it is focused on the *dilemma of exploration and exploitation* (Sutton and Barto, 1998) which is inherently related to the system's safety. The reason for this is that random exploration actions can provoke dangers (high negative costs), but taking exploitation actions based on uncertain environment knowledge can also lead to suboptimal behavior (Tokic and Palm, 2011) or to dangerous states as well.

In the following, a short review of current research on reinforcement learning and safety is given. Furthermore, established RL algorithms in combination with exploration policies are investigated within a dynamic environment having safety-critical aspects. For environments similar to the proposed dynamic-cliff problem, we give answer to the following question: "How do state-of-the-art learning algorithms perform with regard to safety and robustness in changing operating conditions?".

2. REINFORCEMENT LEARNING

We consider learning in Markov decision processes (Sutton and Barto, 1998). At each discrete time step, $t \in \{0, 1, 2, ...\}$, an agent finds itself in a certain state, $s_t \in \mathcal{S}$ based on sensory observations of the environment. After the selection of an action, $a_t \in \mathcal{A}(s_t)$, the agent receives a

reward signal from the environment, $r_{t+1} \in \mathbb{R}$, and transients into a successor state s_{t+1}. During learning, a policy, $\pi := \mathcal{S} \rightarrow \mathcal{A}$, is maintained, reflecting a mapping from environmental states to actions. Usually such a policy is probabilistic, i.e. selection probabilities for possible actions in state s_t at time t, $a_t \in \mathcal{A}(s_t)$, are reflected by the values: $\pi(s,a) = Pr\{a_t = a | s_t = s\}$. Since the goal is to learn an optimal policy π^* maximizing the cumulative reward, the selection probabilities have to be improved *online* by learning from trial-and-error.

One possibility of learning policies is learning action values, $Q(s,a)$, describing the quality of action a in state s. This quality is quantified as a numerical estimate of the expected discounted reward the agent will receive following the current policy π when starting in state s and selecting action a,

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\} \quad , \quad (1)$$

where γ is a discount factor such that $0 < \gamma \leq 1$ for episodic learning tasks and $0 < \gamma < 1$ for continuous learning tasks.

On- and Off-policy learning In case no model of the environment is present, value functions are sampled incrementally during the agent's interaction with its environment,

$$Q^n(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \cdot \Delta_{\text{learningRule}} \quad , \quad (2)$$

were α is a stepsize parameter (George and Powell, 2006), and $\Delta_{\text{learningRule}}$ is the *temporal-difference error* (TD error) of the current estimate and the improved estimate after applying a learning rule based on the observation tuple (s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}).

Two established learning rules being investigated in the following are Q-learning and Sarsa. Both rules are based on two components: 1) on the immediate reward r_{t+1} from the environment, and 2) on an action value from the successor state s_{t+1}. The latter component makes the technical difference between both algorithms, because Q-learning (*off-policy*) uses the highest action value from the successor state for estimating the TD error,

$$b^* \leftarrow \text{argmax}_{b \in \mathcal{A}(s_{t+1})} Q(s_{t+1}, b)$$
$$\Delta_{\text{Qlearning}} \leftarrow [r_{t+1} + \gamma Q(s_{t+1}, b^*) - Q(s_t, a_t)] \quad , \quad (3)$$

assuming that the agent will finally follow an optimal policy (Watkins, 1989). In contrary, Sarsa (*on-policy*) uses the action value from the actual selected action a_{t+1} in the successor state, $Q(s_{t+1}, a_{t+1})$,

$$\Delta_{\text{Sarsa}} \leftarrow [r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad , \quad (4)$$

which results that costs from taking exploratory actions in s_{t+1} are also considered in $Q(s_t, a_t)$ (Rummery and Niranjan, 1994). This "foresight" provides potential to improve safety as shown in the cliff-walking problem by Sutton and Barto (1998) and also in the experiments later on. Anyway, it's important to know that the convergence of the Q-function, using stochastic policies, is only proven for Q-learning (Watkins and Dayan, 1992) rather than for Sarsa, which might become important for guaranteeing learning success. Finally, if the stochasticity in action selections becomes zero (e.g. greedy), Sarsa technically becomes the same as Q-learning, and thus also convergent under several conditions.

2.1 Exploration/Exploitation Policies

A value function, our knowledge about the environment, must be learned accurately in order to avoid taking of actions leading to undesired or dangerous states. For this, the agent has to decide at each time step whether: (1) so far learned knowledge about the environment should be exploited by selecting a directed action (i.e. an action associated with the highest estimated action value), or (2) if an exploration action should be selected for increasing the knowledge about the environment. On the one hand, taking too much exploration actions prevents from maximizing the short-term reward since exploration actions may yield to negative reward. On the other hand, taking exploitation actions based on uncertain environment knowledge can prevent from maximizing the long-term reward because action values may not be accurate. This raises the question: *How much exploration actions must be taken?* Unfortunately, the answer to this question is still unsolved, but interesting heuristics such as the following exist.

ε-Greedy and Softmax Two basic policies used for balancing exploration and exploitation are ε-Greedy and Softmax (Sutton and Barto, 1998). With ε-Greedy the agent selects at each time step a random action with probability $\varepsilon \in [0,1]$, and with probability $1-\varepsilon$ one of the so far learned optimal actions, $\mathcal{A}^*(s) := \text{argmax}_{a \in \mathcal{A}(s)} Q(s,a)$, with respect to the current action-value estimates,

$$\pi(s,a) = \begin{cases} \frac{1-\varepsilon}{|\mathcal{A}^*(s)|} + \frac{\varepsilon}{|\mathcal{A}(s)|} & \text{if } a \in \mathcal{A}^*(s) \\ \frac{\varepsilon}{|\mathcal{A}(s)|} & \text{if } a \notin \mathcal{A}^*(s) \end{cases} \quad . \quad (5)$$

In contrast, Softmax utilizes action-selection probabilities determined from ranking action values using a Boltzmann distribution,

$$\pi(s,a) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_b e^{\frac{Q(s,b)}{\tau}}} \quad , \quad (6)$$

where τ is a positive parameter called temperature. High temperatures cause all actions to be nearly equiprobable, whereas low temperatures cause greedy action selections.

In practice, both policies have advantages and disadvantages as described by Sutton and Barto (1998). In the literature, both policies have also been reported as models for describing the action-selection process in the human brain, where Softmax seems to be the better approximation (Daw et al., 2006).

Value-Difference Based Exploration The idea of "Value-Difference Based Exploration" (VDBE) is to extend the ε-Greedy policy by introducing a state-dependent exploration probability, $\varepsilon(s)$, instead of hand-tuning a global parameter (Tokic, 2010). The desired behavior is to have the agent being explorative in situations when the knowledge about the environment is uncertain, which is indicated by fluctuating action values during learning, e.g. at the beginning of the learning process. In contrast, the amount of exploration should be reduced as far as the agent's knowledge becomes certain, which is indicated by very small or no value differences. Such an adaptive behavior is obtained by computing a state-dependent exploration probability, $\varepsilon(s)$, after each learning step, according to

$$f(s,a,\sigma) = 1 - e^{\frac{-|Q^n(s,a)-Q(s,a)|}{\sigma}} = 1 - e^{\frac{-|\alpha\cdot\Delta|}{\sigma}}$$

$$\varepsilon_{t+1}(s) = \delta \cdot f(s_t, a_t, \sigma) + (1-\delta)\cdot\varepsilon_t(s) \ , \qquad (7)$$

where σ is a positive constant called *inverse sensitivity* and $\delta \in [0,1)$ a parameter determining the effect of the selected action on the state-dependent exploration probability. A reasonable setting for δ is the inverse of the number of actions in the current state, $\delta(s) = 1/|\mathcal{A}(s)|$. At the beginning of the learning process, all exploration probabilities are initialized arbitrary, e.g. $\varepsilon_{t=0}(s) = 1$ for all states. The parameter σ effects $\varepsilon(s)$ in such way that low values cause full exploration at small action-value changes. On the other hand, high values of σ cause a high level of exploration only at large action-value changes. Finally, the exploration probability approaches zero as far as the Q-function converges, which results in pure greedy action selections.

VDBE-Softmax One drawback of VDBE (in particular of ε-Greedy) is that exploration actions are chosen uniformly distributed among all possible actions in the current state. Such exploration behavior can lead to bad performance when many actions in the current state yield to relatively high negative reward, even if this knowledge is present through already learned action values. Furthermore, action value oscillations cause a non-zero level of $\varepsilon(s)$, e.g. caused by stochastic rewards, function approximators or learning algorithms.

A way of relaxing this drawback is to replace the equally distributed explorative action selection by the ranked Softmax action selection (Tokic and Palm, 2011),

$$\pi(s,a) = \begin{cases} \text{Softmax}(s,a) & \text{if } \xi \leq \varepsilon(s) \\ \dfrac{1}{|\mathcal{A}^*(s)|} & \text{if } \xi > \varepsilon(s), \end{cases} \qquad (8)$$

where ξ is a uniform random number within $[0,1]$.

To further ease the search for reasonable parameters for Softmax, it is proposed to use a normalization of the action values into the interval $[Q_{\text{normMin}}, Q_{\text{normMax}}]$, e.g. $[-1,1]$, and having the temperature parameter of Softmax set constantly to the value of $\tau = 1$. With this, a mean independency of the distribution of action values in state s is achieved that enables the selection of τ more intuitively. In present investigations, such an approach turned out to be sufficient for suppressing the selection of actions yielding to highly negative reward in case of $\xi < \varepsilon(s)$ (Tokic and Palm, 2011).

3. LITERATURE ON RL AND SAFETY

Several investigations on RL being applied in a safety-critical context are available. Some of these introduce safety aspects by giving stability guarantees for controllers (Perkins and Barto, 2003), even using arbitrary learning algorithms (Ng and Kim, 2004). Perkins and Barto (2003) *use domain knowledge [...] to design the action choices available to the agent. An appropriately designed set of actions restricts the agent's behavior so that regardless of precisely which actions it chooses, desirable performance and safety objectives are guaranteed to be satisfied.* The relevant domain knowledge is designed as a *Lyapunov*-function. Pursuing towards minima of the Lyapunov-function means approaching a point of stability. Actions are restricted to those having the probability of a negative gradient in the Lyapunov-function. Therefore, the system converges towards a stable point.

Similar investigations took place, focusing the variance of the reward. Learning algorithms are often extended with *utility functions*, which are intended to represent subjective measures; e.g. risks and benefits in a meaningful relation. For instance, Heger (1994) develops the $\widehat{Q}$-Learning algorithm which considers the worst-case scenario of risks, being expressed as utility functions. $\widehat{Q}$-Learning, as being the *counterpart to Watkin's Q-Learning [...]* (Watkins, 1989) *[...] related to the minimax criterion*, finds policies that minimize the worst-case costs.

Geibel (2001) introduces a separate risk-cost function which allows for limiting and balancing risks. Risks are limited by classifying states as unsafe, when a certain cumulative risk is exceeded. For balancing, Geibel (2001) suggests a parameter weighting benefits and risk-costs between pure greedy and pure risk-optimal policies. Varying this parameter can be used to realize cautiousness, for instance, at the beginning of the learning process.

Mihatsch and Neuneier (2002) propose to be sensitive with regard to estimation errors. Therefore, a parameter $\kappa \in (-1,1)$ is coupled with the weighting of the temporal-difference error with respect to its numeric sign. For $\kappa \to -1$, the algorithm behaves in a risk-avoiding minimax manner, for $\kappa = 0$ the algorithm is risk-neutral and for $\kappa \to 1$ an excessively optimistic behavior results.

Latter contributions are based on either integrating assumptions about the world into the system or assuming that such knowledge is available. The system shall be hindered to enter unsafe states by limiting its decision possibilities or by favoring of risk-adverse strategies, keeping distance from undesired states. In general, the possibility for exploring new and unknown states remains.

3.1 Exploration and Safety

The exploration and exploitation problem of reinforcement learning (Sutton and Barto, 1998) is inherently involved in the safety issue. Basically, the exploration-exploitation problem is rather the decision to either chose an action in a more or less well known state with more or less well known consequences or to try something new. From a safety perspective, entering hazardous states (assuming they are represented by high negative rewards) is to avoid, and the entering of accidental states is intolerable.

Therefore, Hans et al. (2008) focus on safe exploration. Besides introducing a *safety function*, giving pre-modeled safety information with respect to state-action pairs, the idea of a *backup policy* is mentioned. The backup policy shall be able to transfer the system from its current state to a safe state whenever an unsafe state occurs. The exploration itself is suggested to take place structured in a *level-wise* manner: Exploration is locally bounded to a state until respective exploration possibilities are exhausted.

3.2 Short Reflection

Important and surely very valuable investigations were made with the intention to mitigate risks when RL methods are be applied. Introduction of risk, utility or energy functions, the balancing of risk and benefit or risk-adverse characteristics of algorithms in combination with intensive hazard and risk analysis, and intensive efforts to quantify benefits and risks in form of rewards allows for substantial or even sufficient reduction of risks, also possibly for very complex applications. But the safety in mentioned

approaches relies on pre-given definitions of hazards and their risks. Lussier et al. (2004) classifies such situations as being either nominal or adverse. Adverse situations can be foreseen and specified or they might be unexpected as well (see Figure 1). Thus, besides the challenge of principally limiting a system to safe states by specifying adverse situations, the question "*what if*" remains, when the system approaches or enters an unknown unsafe state either with or even without a reasonable backup policy.

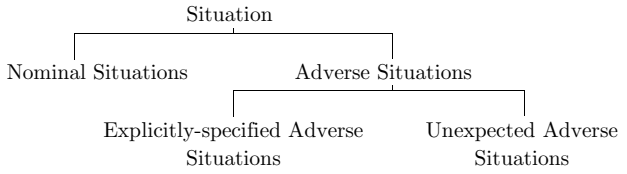


Fig. 1. Classification of situations (Lussier et al., 2004).

Hans et al. (2008) assumed in their safe exploration approach that step-wise exploration from demonstrated data (apprenticeship learning) accompanied by a pre-specified backup policy is safe. This might be valid for their use-case, in which a kind of continuous characteristic curve for controlling a gas turbine is learned. In other applications the transitions to dangerous states might be discontinuous. Imagine a car is driven by an autonomous agent as demonstrated beforehand by an expert. While strictly following the demonstrated state-action paths, changing the steering angle might be one simple exploration action with disastrous consequences (if not being permitted beforehand). Furthermore, initial safe regions and paths or even predefined backup policies might become suddenly dangerous if the operating conditions change (due to unforeseen environmental or failure-related conditions). The level-wise exploration proposed by Hans et al. (2008) is based on the assumption about the world that neighbored states of safe states are safer than others. In a case where nothing else remains than explore a path away from a dangerous state, however, this might result in an intensive exploration of the dangerous area. Thus, the basic assumption of the contribution at hand is that established (pre-given or learned) assumptions about the world can suddenly become inappropriate for specific cases. The inability to fast adapt to changing conditions, for instance based on low exploration rate, pre-defined exploration strategies or predefined exclusion of decision choices, could result in *rigid* [1] behavior.

In that respect, also the learning rate α plays an important role. It seems reasonable to adapt the learning rate in such way, that extreme experiences (e.g. touching a hot kitchen stove) may induce a high learning rate. For instance, George and Powell (2006) proposed such a method to adequately adopt the learning rate (stepsize α). They introduced the optimal stepsize algorithm (OSA), which controls the stepsize to be low in case of low variances of the estimate error (error of value function) and high, in case of high variances.

The contribution at hand is based on a constant stepsize parameter, which surely does not allow best possible results. Indeed, it allows investigating and outlining the relevance of the exploration/exploitation policies to the safety of learning systems. Further investigations should integrate other aspects, too.

[1] rigidity is described by Dörner (2000) as adhering to a strategy although external effects might require for changing the strategy in order to be more efficient or successful at all.

3.3 Research Questions

Finally, exploration is dangerous due to the possibility to enter unsafe states. No exploration might be dangerous as well, as it is inherently difficult for certain very complex systems, to limit them to safe states by a proper and safe system design. In the case of being somehow forced to apply learning systems, for instance in future service robotic applications, and without judging on final risk and benefits, this contribution investigates the "what-if" constellation with regard to safety aspects of different exploration and exploitation strategies.

Basic interests are, how the different exploration and exploitation strategies perform focusing on safety. Hence, these strategies often depend on manually tuning the parameters in accordance to the learning problem, the different parameter settings are from interest. Changed operation conditions in an unforeseen manner might let become these settings inadequate. In that respect, the following topics appear to be significant:

- A certain robustness and hence, increase of safety, might be allowed if the performance variance is low with regard to different parameter settings.
- Furthermore, it seems intuitively conclusive that best learning performance increases safety, as unknown dangerous states (assuming there is respective reward) are learned faster. Under which circumstances that assumption holds and under which not, might be another revealing question.
- When the operating conditions change in such disastrous way, that only few solutions are available as a correct choice, the question arises, how the exploration and exploitation strategies are performing with regard to safety in such case.

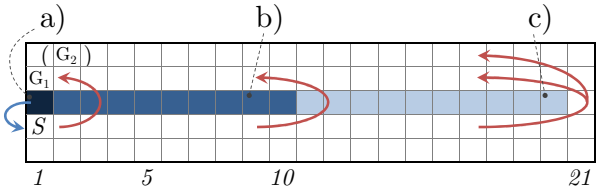
4. INVESTIGATION

Exploration algorithms are typically evaluated in the multi-armed bandit domain, a learning problem having only one state (Tokic, 2010). As the investigated domain is assumed to consist of multiple states, a different learning problem highlighting the different behavior of on- and off-policy learning algorithms is favored. For this, Sutton and Barto (1998) proposed the cliff-walking problem comprising basic aspects of relevance for safety considerations, which in the following is extended to the *dynamic-cliff problem* considering also unexpected situations.

In the dynamic-cliff problem, the goal of the agent is to learn a path from start state S to goal state G_1 , which is rewarded with the absolute costs of the shortest path minus 1 if successful (see Fig. 2). For reaching the goal, the agent has to choose an action at each time step leading to a neighbored state. The costs (reward) for each action is defined as $r_{\text{step}} = -1$ (way costs). The environment also comprises unsafe *cliff states*, which, when entered, lead to a high negative reward of $r_{\text{cliff}} = -100$, and also reset the agent back to the start state S. Thus, cliff states are unsafe assuming they comprise risks above an tolerable level.

At the beginning of the experiment, learning takes place in phase (a) of Fig. 2 having one hazardous state (at left border). After 200 learning episodes, the grid world changes to phase (b), now comprising 10 hazardous situations. This change represents the sudden altering of current operating conditions requiring to adapt the already learned behavior for circumventing the additional hazards. After additional 800 episodes, the problem is tightened

as shown in phase (c): A situation in which most of the trials suddenly lead to a hazardous state is assumed to be typical for a critical-failure mode. Therefore the number of hazardous states is increased to 20. Additionally, an alternative goal state appears, $G2$, which is much higher rewarded with $r_{G2} = 500$ when entered, representing an appearing *fail-safe state*, for instance.



Rewards:

Phase/ Episode	a) *1...200*	b) *201...1000*	c) *1001...3000*
r_{G1}	*3*	*21*	*41*
r_{G2}	n/a	n/a	*500*

Fig. 2. Model of the dynamic-cliff problem and its three subsequent phases: (a) during episodes 1...200, (b) during episodes 201...1000 and (c) during episodes 1001...3000 with additional goal state G_2 and related rewards for reaching $G1$ or $G2$. All rewards for $G1$ reflect the absolute costs of the shortest path leading directly along the cliff.

4.1 Experimental Setup

In each episode the agent begins in state S, as well an episode terminates when a goal state, $G1$ or $G2$, is entered or the time limit of $T_{\max} = 200$ actions is reached. For each parameter setting of all investigated policies, the results are averaged over 5000 runs. Throughout the experiment, the step-size parameter α is constantly set to the value of $\alpha = 0.1$. Since the learning problem is an episodic task, no discounting ($\gamma = 1$) is used. In this experimental setting, Q-learning and Sarsa are investigated in combination with ε-Greedy, Softmax, VDBE and VDBE-Softmax policies using constant parameter settings [2], and using a tabular approximation of the action values. In experiments with VDBE and VDBE-Softmax, all exploration probabilities are initialized with $\varepsilon(s) = 1$, as well all δ's been configured with $\delta(s) = 1/|\mathcal{A}(s)| = 0.25$. For VDBE-Softmax, the normalization interval is set to $[-1, 1]$ using $\tau = 1$ for Softmax exploration actions. Finally, all action values are optimistically initialized with $Q_{t=0}(s, a) = 0$.

4.2 Results

In order to model a safety reflecting measurement, the cliff-falls in relation to performed actions is focused (cliffs/steps). The amount of performed actions represents a time span within whose a certain amount of dangerous/accidental states $n_{\text{cliffFalls}}$ occur. The risk is assumed to be constant ($r_{\text{cliff}} = -100$), therefore $\text{risk}_{\text{total}} = \sum_{i=0}^{T} risk(\text{a}_i) \propto n_{\text{cliffFalls}}$. Hence, the percentage ratio of cliffs and steps is shown in Fig. 3 (upper). Under best-case parameter setting, all experiments approached low ratios (best ratios are green shaded for each phase and its progress), also for difficult circumstances for phase (c). For the worst case, VDBE-Softmax shows lowest ratios for both learning algorithms.

[2] Investigated parameters are: $\varepsilon \in \{0.0, 0.01, 0.1, 0.2, 0.5, 1.0\}$; $\tau, \sigma \in \{0.001, 0.01, 0.04, 0.1, 1.0, 10.0, 25.0, 100.0, 1000.0\}$.

cliffs / actions [%]

		Q-Learning best				Q-Learning worst				Sarsa-Learning best				Sarsa-Learning worst			
phase	episode	e-Greedy	Softmax	VDBE	VDBE-Softm.	e-Greedy	Softmax	VDBE	VDBE-Softm.	e-Greedy	Softmax	VDBE	VDBE-Softm.	e-Greedy	Softmax	VDBE	VDBE-Softm.
a)	10	0,00	0,00	0,01	0,00	0,92	2,19	2,37	0,30	0,00	0,00	0,01	0,00	2,28	2,12	2,38	0,17
	40	0,00	0,00	0,00	0,00	2,82	2,10	2,26	0,95	0,00	0,00	0,00	0,00	2,36	1,90	2,34	0,19
	200	0,00	0,00	0,00	0,00	4,76	2,10	1,48	0,03	0,00	0,00	0,00	0,00	2,38	1,84	2,48	0,65
	param.	0	1	1000	1000	0,5	1000	0,001	0,001	0	1	1000	1000	1	1000	0,001	0,04
b)	210	0,19	0,16	0,19	0,19	14,25	13,16	14,39	1,67	0,18	0,16	0,18	0,18	14,38	12,12	14,39	0,83
	300	0,00	0,06	0,00	0,00	14,33	12,98	12,28	1,45	0,01	0,02	0,01	0,00	14,30	11,30	14,36	0,51
	1000	0,00	0,00	0,00	0,00	14,30	12,97	9,15	0,38	0,00	0,00	0,00	0,00	14,39	8,67	14,30	0,59
	param.	0	0,1	1000	1000	1	1000	0,001	0,001	0	0,1	1000	1000	1	1000	0,001	0,001
c)	1010	0,31	0,30	0,31	0,32	14,46	13,04	9,90	1,96	0,31	0,31	0,32	0,31	14,46	10,60	14,46	1,35
	1100	0,38	0,28	0,33	0,37	14,40	13,03	10,07	2,40	0,38	0,37	0,27	0,34	14,42	10,23	14,39	1,00
	1500	0,00	0,00	0,00	0,00	14,44	13,10	8,62	2,40	0,00	0,00	0,00	0,00	14,57	10,04	14,37	0,92
	3000	0,00	0,00	0,00	0,00	14,52	13,07	7,43	0,18	0,00	0,00	0,00	0,00	14,37	9,16	14,49	0,84
	param.	0	0,04	1000	1000	1	1000	0,001	0,001	0	0,001	1000	1000	1	1000	0,001	0,001
a)-c)		0,05	0,05	0,05	0,05	13,80	12,50	8,62	1,51	0,05	0,05	0,05	0,05	13,80	9,48	13,81	0,81
	param.	0	0,04	1000	1000	1	1000	0,001	0,001	0	0,001	1000	1000	1	1000	0,001	0,001

reward ( )

		Q-Learning best				Q-Learning worst				Sarsa-Learning best				Sarsa-Learning worst			
phase	episode	e-Greedy	Softmax	VDBE	VDBE-Softm.	e-Greedy	Softmax	VDBE	VDBE-Softm.	e-Greedy	Softmax	VDBE	VDBE-Softm.	e-Greedy	Softmax	VDBE	VDBE-Softm.
a)	200	0	0	0	0	-438	-406	-23	0	0	0	0	0	-453	-369	-367	-30
	param.	0	0,01	1000	1000	1	1000	0,001	0,001	0	0,01	1000	1000	1	1000	0,001	0,001
b)	1000	0	0	0	0	-2691	-2485	-1868	-17	0	0	0	0	-2704	-1788	-2691	-129
	param.	0	0,001	1000	1000	1	1000	0,001	0,001	0	0,001	1000	1000	1	1000	0,001	0,001
c)	3000	482	0	492	496	-2731	-2494	-1588	1	484	500	248	466	-2713	-1882	-2703	1
	param.	0,01	0,001	1	0,01	1	1000	0,001	1000	0,01	10	10	1	1	1000	0,001	1000
a)-c)		235	-16	103	165	-2557	-2358	-1721	-159	273	117	112	254	-2557	-1881	-2555	-99
	param.	0,01	0,001	10	1	1	1000	0,001	0,001	0,1	10	10	1	1	1000	0,001	0,001

Fig. 3. Cliff-fall per action ratio (upper) and cumulative rewards (lower) with regard to worst- and best-case parameter settings, and the respective parameter within learning phases.

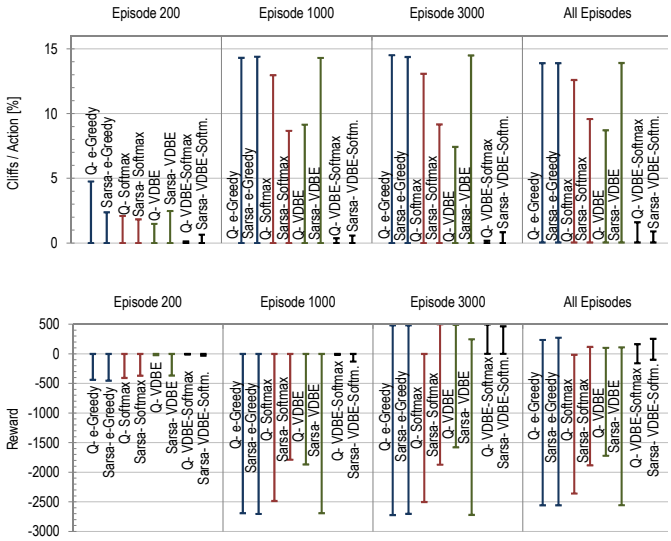


Fig. 4. Range of worst- and best-case results of cliff falls per action (upper) and the cumulative reward (lower).

The safest worst-case performance is achieved by applying Sarsa in combination with VDBE-Softmax. Illustrating this result by assuming the execution time for all actions as $t_i^{\text{action}} = 1s$, the experiment lasts in the worst-case for about $127h$ of operating time with mean probable unsafe states of $29.1 \frac{\text{cliffFalls}}{h}$, in the best case about $38h$ with

$1.8\frac{\text{cliffFalls}}{h}$ (this implicates surely that the mentioned algorithms as presented for this experimental setting cannot be used for applications with severe consequences).

The results regarding to the reward are summarized in Fig. 3 (lower), whereas the bandwidth of worst- and best-case settings are illustrated in Fig. 4. When comparing the results of ε-Greedy and Softmax, we were able to reproduce similar results as Sutton and Barto (1998) within the cliff-walking problem, where Sarsa learns a *safer path* more far away from hazardous states; in contrast to the *shortest path* along the cliff learned by Q-learning. This behavior of Sarsa is due to the consideration of the policy's stochasticity resulting in a bit more way costs, but which in total are not as much as when falling off the cliff. For VDBE, better worst-case results are achieved using Q-learning, which is for the reason that Sarsa produces value oscillations in case of exploration actions. These oscillations cause a non-zero level of $\varepsilon(s)$, thus more equally distributed action selections. In contrast, a significant improvement for worst-case results is achieved by VDBE-Softmax due to the selection of directed exploration actions. As far as the value function converges, the temporal-difference error approaches to $\Delta \to 0$, resulting VDBE-Softmax to greedily exploit so far learned knowledge.

In phases (a) & (b), all investigated policies are able to achieve minimal costs using greedy parameter settings [3]. But as a counter example, such as in phase (c), greedy behavior can also result in sub-optimal behavior when unexpected situations occur, e.g. such as the appearance of the second goal state $G2$ that is much higher rewarded. In such situations, at least a bit of exploration is required for dramatically improving the cumulative reward. On the contrary, worst-case parameter settings exist for ε-Greedy, Softmax and VDBE, which lead to high negative reward due to the constant selection of random actions [4]. For this, the red-shaded values in Fig. 3 highlight those parameters achieving best performance with regard to the reward but not maximal safety. Remarkably, parameter settings with the best reward performance are not always the safest settings, because of utilizing more exploration steps sometimes also provoking cliff falls.

5. CONCLUSION

In this paper, we showed how safety of autonomous systems is related to the ratio of exploration and exploitation when RL algorithms are applied for learning behavior. The results show that improper settings of the exploration parameter can lead to highly negative reward from the environment through the selection of safety-critical actions. Learning algorithms such as Sarsa and Q-learning have been investigated in combination with four policies for measuring worst- and best-case performance of each combination. The results indicate that balancing the ratio of exploration and exploitation on basis of the learning process seems to be valuable for producing safe behavior in unexpected adverse situations having the VDBE-Softmax policy being most robust. The reason for this behavior is that VDBE-Softmax does not select exploration actions equally distributed. Instead, in case of fluctuating values exploration actions are selected value sensitively according to Softmax, and greedily in case the value function has converged. Nonetheless, there remains a trade-off between safety and learning optimal solutions. As a final conclusion, the results indicate that the VDBE-Softmax policy should be combined with other learning algorithms in future research (e.g. such as those discussed in Section 3), which might offer a powerful step towards building *safe autonomous systems*.

[3] Greedy parameter settings: $\varepsilon = 0$ (ε-Greedy), $\tau \to 0$ (Softmax), $\sigma \to \infty$ (VDBE and VDBE-Softmax)

[4] Worst-case parameter settings: $\varepsilon = 1$ (ε-Greedy), $\tau \to \infty$ (Softmax), $\sigma \to 0$ (VDBE and VDBE-Softmax)

REFERENCES

Daw, N.D., O'Doherty, J.P., Dayan, P., Seymour, B., and Dolan, R.J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095), 876–879.

Dörner, D. (2000). *Die Logik des Mißlingens strategisches Denken in komplexen Situationen*. Rowohlt, Reinbek bei Hamburg.

Ertle, P., Voos, H., and Söffker, D. (2010). On risk formalization of on-line risk assessment for safe decision making in robotics. In *7th IARP Workshop on Technical Challenges for Dependable Robots in Human Environments*, 15–22.

Geibel, P. (2001). Reinforcement learning with bounded risk. In *Proceedings of the 18th International Conference on Machine Learning*, ICML'01, 162–169. Morgan Kaufmann Publishers Inc.

George, A.P. and Powell, W.B. (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learning*, 65(1), 167–198.

Hans, A., Schneegaß, D., Schäfer, A.M., and Udluft, S. (2008). Safe exploration for reinforcement learning. In *Proceedings of the 16th European Symposium on Artificial Neural Networks ESANN'08*, 143–148.

Heger, M. (1994). Consideration of risk in reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, 105–111. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA.

Lussier, B., Chatila, R., Ingrand, F., Killijian, M.O., and Powell, D. (2004). On fault tolerance and robustness in autonomous systems. In *3rd IARP - IEEE/RAS - EURON Joint Workshop on Technical Challenges for Dependable Robots in Human Environments*, 7–9.

Mihatsch, O. and Neuneier, R. (2002). Risk-sensitive reinforcement learning. *Machine Learning*, 49(2), 267–290.

Ng, A.Y. and Kim, H.J. (2004). Stable adaptive control with online learning. In *Advances in Neural Information Processing Systems*, 17, 13–18.

Perkins, T.J. and Barto, A.G. (2003). Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3, 803–832.

Rummery, G.A. and Niranjan, M. (1994). On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University.

Smithers, T. (1997). Autonomy in robots and other agents. *Brain and Cognition*, 34, 88–106.

Sutton, R.S. and Barto, A.G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Tokic, M. (2010). Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *KI 2010: Advances in Artificial Intelligence*, 203–210. Springer Berlin / Heidelberg.

Tokic, M. and Palm, G. (2011). Value-difference based exploration: Adaptive exploration between epsilon-greedy and softmax. In *KI 2011: Advances in Artificial Intelligence*, 335–346. Springer Berlin / Heidelberg.

Watkins, C. (1989). *Learning from Delayed Rewards*. Ph.D. thesis, University of Cambridge, England.

Watkins, C. and Dayan, P. (1992). Technical note: Q-learning. *Machine Learning*, 8(3), 279–292.