Online Learning with Stochastic Recurrent Neural Networks using Intrinsic Motivation Signals

Daniel Tanneberg, Jan Peters,* Elmar Rueckert
Intelligent Autonomous Systems Lab
Technische Universität Darmstadt

Abstract: Continuous online adaptation is an essential ability for the vision of fully autonomous and lifelong-learning robots. Robots need to be able to adapt to changing environments and constraints while this adaption should be performed without interrupting the robot's motion. In this paper, we introduce a framework for probabilistic online motion planning and learning based on a bio-inspired stochastic recurrent neural network. Furthermore, we show that the model can adapt online and sample-efficiently using intrinsic motivation signals and a mental replay strategy. This fast adaptation behavior allows the robot to learn from only a small number of physical interactions and is a promising feature for reusing the model in different environments. We evaluate the online planning with a realistic dynamic simulation of the KUKA LWR robotic arm. The efficient online adaptation is shown in simulation by learning an unknown workspace constraint using mental replay and *cognitive dissonance* as intrinsic motivation signal.

Keywords: Lifelong-learning, Intrinsic Motivation, Recurrent Neural Networks

1 Introduction

The idea of developmental robots [1, 2, 3, 4, 5] that adapt autonomously through lifelong-learning is still one of the big challenges in robotics. Although in recent years a lot of research has been done for learning tasks autonomously, experts with domain knowledge are still required for defining and guiding the learning problem, e.g., for reward shaping or providing demonstrations. In a fully autonomous setup however, these procedures should be carried out by the robot itself. That is, the robot should be able to decide when, what, and how to learn. Furthermore, planning a movement, executing it, and learning from the results should be integrated in a continuous online framework. Thus, closing the loop between training and execution phase is a necessary step towards autonomous lifelong-learning robots. In this work, we use an intrinsic motivation signal for online model adaptation, that tells the agent where its model is incorrect and guides the learning through this mismatch.

From autonomous mental development in humans it is known that intrinsic motivation is a strong factor for learning [6, 7]. While the concept of intrinsically motivated learning has inspired many studies about artificial and robotic systems, e.g., [8, 9, 10, 11, 12, 13], only few approaches so far go beyond a reinforcement learning setup. In these works, intrinsic motivation signals are used for incremental task learning, acquiring of skill libraries, learning of perceptual patterns and object manipulation. For the goal of fully autonomous robots however, the ability to focus and guide learning independently from tasks and specified rewards is crucial.

In this paper we demonstrate how an agent can benefit from using intrinsic motivation for efficient online adaptation. We implement this learning approach into a biologically inspired stochastic recurrent neural network for motion planning and embed our method into a novel continuous framework for probabilistic online motion planning and learning. The online model learning is modulated by an intrinsic motivation signal that is inspired by cognitive dissonance [14, 15], a knowledge-based model of intrinsic motivation [16], that describes the divergence of an expectation to the actual outcome. Additionally, to intensify experienced situations, we use a mental replay mechanism, what has been proposed to be a fundamental concept in human learning [17]. We will show that as a result of these mechanisms, the model can efficiently adapt online to unknown environments.

*also with the Robot Learning Group, Max-Planck Institute for Intelligent Systems

1st Conference on Robot Learning (CoRL 2017), Mountain View, United States.

1.1 Motion Planning with Stochastic Recurrent Neural Networks

The proposed framework builds on the model recently developed by [18], where it was shown that stochastic spiking networks can solve motion planning tasks optimally. Furthermore, the authors showed an approach to scale these models to higher dimensional spaces by introducing a factorized population coding and that the model can be trained from demonstrations [19]. The model was inspired by neuroscientific findings on the mental path planning in rodents [20] and mimics the behavior of the recorded hippocampal place cells. It was shown that the neural activity of these cells are correlated with future paths the rodents planned to take.

This bio-inspired motion planner consists of stochastically spiking neurons forming a multi-layer recurrent neural network, hence, we refer to it as a stochastic recurrent neural network. The basis model consists of two different types of neuron populations: a layer of K state neurons and a layer of N context neurons. The state neurons form a fully connected recurrent layer with synaptic weights $w_{i,k}$, while the context neurons provide feedforward input via synaptic weights $\theta_{j,k}$, with $j \in N$ and $k, i \in K$. This task-related input from the context neurons can be learned [18] or can be used to, for example, include known dynamic constraints in the planning process [19]. Considering the discrete time case, the state neurons probability to spike is proportional to their membrane potential

$$u_{t,k} = \sum_{i=1}^{K} w_{i,k}\tilde{v}_i(t) + \sum_{j=1}^{N} \theta_{j,k}\tilde{y}_j(t) \,, \tag{1}$$

where $\tilde{v}_i(t)$ and $\tilde{y}_j(t)$ denote the presynaptic potential from neurons $i \in K$ and $j \in N$ respectively. The binary activity of the state neurons is denoted by $\mathbf{v}_t = (v_{t,1}, .., v_{t,K})$, where $v_{t,k} \propto u_{t,k}$ and $v_{t,k} = 1$ if neuron k spikes at time t and $v_{t,k} = 0$ otherwise. Analogously, $\mathbf{y}_t$ describes the activity of the context neurons. To find a movement trajectory from position $\mathbf{a}$ to a target position $\mathbf{b}$, the model generates a sequence of states encoding a task fulfilling trajectory. For this planning problem, the network encodes a distribution

$$q(\mathbf{v}_{1:T}|\boldsymbol{\theta}) = p(\mathbf{v}_0)\prod_{t=1}^{T}\mathcal{T}(\mathbf{v}_t|\mathbf{v}_{t-1})\phi_t(\mathbf{v}_t;\boldsymbol{\theta}) \tag{2}$$

over state sequences of T timesteps, where $\mathcal{T}(\mathbf{v}_t|\mathbf{v}_{t-1})$ denotes the transition model and $\phi_t(\mathbf{v}_t;\boldsymbol{\theta})$ the task related input provided by the context neurons. Movement trajectories can be sampled by simulating the dynamics of the stochastic recurrent network using Equations (1) and (2), resulting in a sequence of binary activity vectors. These binary neural activities encode the continuous system state $\mathbf{x}_t$, e.g., end-effector position or joint angle values, using the decoding scheme

$$\mathbf{x}_t = \frac{1}{|\hat{\mathbf{v}}_t|}\sum_{k=1}^{K}\hat{v}_{t,k}\mathbf{p}_k \quad \text{with} \quad |\hat{\mathbf{v}}_t| = \sum_{k=1}^{K}\hat{v}_{t,k}, \tag{3}$$

where $\mathbf{p}_k$ denotes the preferred position of neuron k and $\hat{v}_{t,k}$ is the Gaussian window filtered activity of neuron k at time t. All state neuron have a preferred position $\mathbf{p}_k$ which they encode and can be seen as discretized base functions for encoding the continuous state values. Using a full population coding scheme with equally spacing, the state neuron population forms a grid over the modeled space and can be seen as a simplified version of the biological place cells in the hippocampus, forming a cognitive map of the environment.

2 Online Motion Planning and Learning Framework

The general idea of how we enable the model to plan and adapt online is illustrated in Figure 1. Instead of planning complete movement trajectories over a long time horizon, we consider a short time horizon and call this sub-trajectory *segment*. A trajectory $\boldsymbol{\kappa}$ from position $\mathbf{a}$ to position $\mathbf{b}$ can thus consist of multiple segments. This movement planning segmentation has two major advantages. First, it enables the network to consider feedback of the movement execution in the planning process and, second, the network can react to changing contexts, e.g., a changing target position.

To ensure a continuous execution of segments, the planning phase of the next segment needs to be finished before the execution of the current segment finished. On the other hand, planning of the next segment should be started as late as possible to incorporate the most up-to-date feedback into

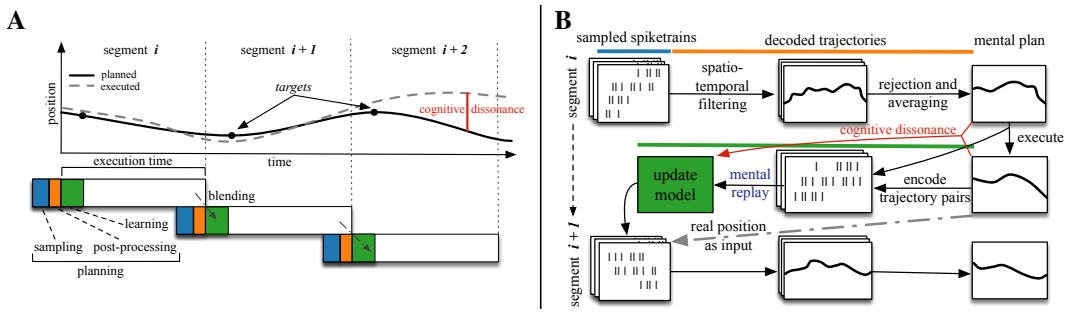

Figure 1: A shows the online planning concept by using short segments. On the upper part the idea of cognitive dissonance is illustrated with a planned and executed trajectory. The three steps learning, sampling and post-processing are organized such that they are performed at the end of the execution of the previously planned segment. **B** shows the blending between successive segments, the multiple samples that are used to generate the final mental trajectories and the model update step with the matching mental and executed trajectory pairs before starting the next segment.

the process. Thus, for estimating the starting point for planning the next segment, we calculate a running average over the planning time and use the three sigma confidence interval compared to the expected execution time. Note that the learning part can be done right after a segment execution is finished. In order to create a smooth movement trajectory, we average over multiple samples drawn from the model when planning each segment.

2.1 Online Model Adaptation with Stochastic Recurrent Networks and Intrinsic Motivation

The online update of the spiking network model is based on the contrastive divergence based learning rules derived recently in [19]. We want to update the state transition function $\mathcal{T}(\mathbf{v}_t|\mathbf{v}_{t-1})$ in Equation (2), which is encoded in the recurrent synaptic weights $\mathbf{w}$ between the state neurons. Thus, learning or adapting the transition model means to change these synaptic connections.

For using the derived model learning rule in our online scenario, we need to make several changes. In the original work, the model was initialized with inhibitory connections. This means that no movement can be sampled from the model for exploration until the learning process has converged. This is not suitable in our case, as we need a *working* model for exploration, i.e., the model needs to be able to generate movements at any time. Therefore, we initialize the synaptic weights between the state neurons with an uniform prior using Gaussian distributions, i.e., a Gaussian distribution is placed at the preferred position of each state neuron and the synaptic weights are drawn from these distributions with an additional additive offset term. The variance of these basis functions are chosen such that only close neighbors get excitatory connections, while distant neighbors get inhibitory connections, ensuring only small state changes within one timestep.

Furthermore, we need to adapt the learning rule as we do not learn with an *empty* model from a given set of demonstrations but rather update a *working* model with online feedback. Therefore, we treat the perceived feedback in form of the executed trajectory as the training data. To encode the trajectories into spiketrains, we use inhomogeneous Poisson processes with the Gaussian responsibilities of each state neuron at each timestep as time-varying input as in [19]. These responsibilities are calculated using Gaussian basis functions centered at the neurons preferred positions with the same parameters as for initializing the synaptic weights.

For online learning, the learning rate typically needs to be small to account for the noisy updates, inducing a long learning horizon, and thus require a large amount of samples. Especially, for learning with robots this is crucial as the number of experiments is limited. Furthermore, the model should only be updated if *necessary*. Therefore, we introduce a time-varying learning rate α_t that controls the update step. This dynamic rate can for example encode uncertainty to update only reliable regions, can be used to emphasize updates in certain workspace or joint space areas, or to encode intrinsic motivation signals. In this work, we employ an intrinsic motivation signal that is motivated by cognitive dissonance [14, 15] for α_t. Concretely, we use the dissonance between the mental movement trajectory generated by the stochastic network and the actual executed movement. Thus, if the executed movement is similar with the generated mental movement, the update is small, while

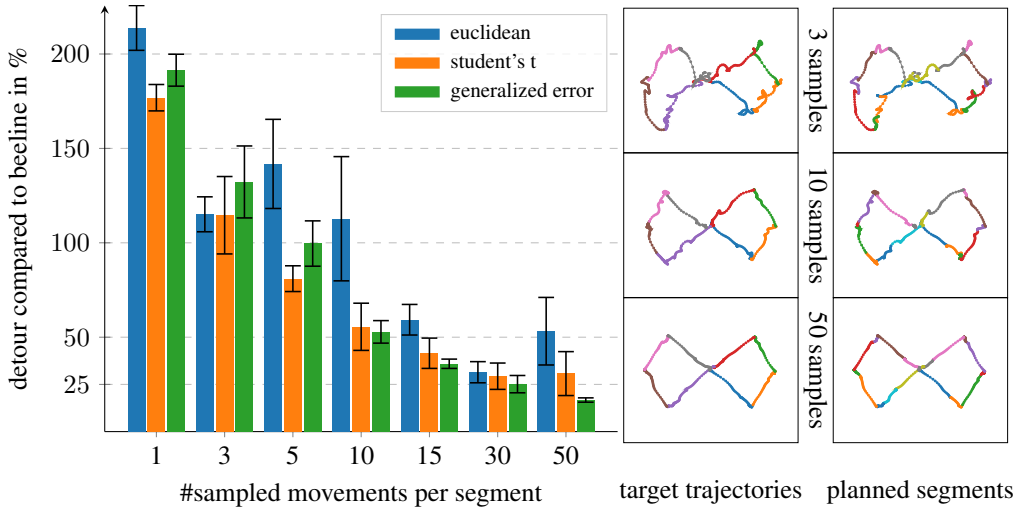

Figure 2: Mean and standard error of the detour compared to the beeline between the target positions for different context neurons weights distributions, where 100% corresponds to the doubled covered distance. Next to the bar plot, three examples are shown of following an eight-shape given by seven via points, highlighting the minimized detour when using more samples. The different colors indicate the considered parts of the total trajectory for calculating the evaluations, i.e. the sections between two target positions. These sections can consist of multiple planned segments, as shown on the right column.

a stronger dissonance leads to a larger update. In other words, learning is triggered and scaled by the mismatch between expectation and observations reflecting reality.

We implement this cognitive dissonance signal by the timestep-wise distance between the mental movement plan $\boldsymbol{\kappa}^{(m)}$ and the executed movement $\boldsymbol{\kappa}^{(e)}$. As distance metric we chose the L^2 norm but other metrics could be used as well depending on, for example, the modeled spaces or the environment specific features. At time t , we update the synaptic connection $w_{k,i}$ with

$$w_{k,i} \leftarrow w_{k,i} + \alpha_t \Delta w_{k,i} \quad \text{with} \quad \alpha_t = \|\boldsymbol{\kappa}_t^{(m)} - \boldsymbol{\kappa}_t^{(e)}\|_2$$
$$\text{and} \quad \Delta w_{k,i} = \tilde{v}_{t-1,k}\tilde{v}_{t,i} - \tilde{v}_{t-1,k}v_{t,i} \quad ,$$

where $\tilde{v}_t$ is generated from the actual executed movement trajectory $\boldsymbol{\kappa}_t^{(e)}$ and v_t from the mental trajectory $\boldsymbol{\kappa}_t^{(m)}$. To stabilize the learning progress, we limit α_t in our experiments to $\alpha_t \in [0, 0.3]$.

Using Mental Replay Strategies to Intensify Experienced Situations. As the encoding of trajectories into spiketrains using inhomogeneous Poisson processes is a stochastic operation, we can obtain a whole population of encodings from a single trajectory. We utilize this feature to implement a mental replay strategy that intensifies experienced situations to speed up adaptation. In particular, we draw 30 trajectory encoding samples per observation in the adaptation experiment, where each sample is a different spike encoding of the trajectory, i.e., a mental replay of the experienced situation. Thus, by using such a mental replay approach, we can apply multiple updates from a single interaction. The two mechanisms, intrinsically motivated learning and mental replay, lower the required number of training data, which is a crucial requirement for learning with robotic systems.

3 Experiments

We conducted two experiments to evaluate the proposed framework. First, for testing the model's ability in online planning, we used a realistic simulation of the KUKA LWR robotic arm. Second, for a first evaluation of the model's online learning ability using intrinsic motivation signals, we altered the motion planning setup. In this new setup, the model has to learn to avoid an unknown workspace constraint in a simple simulation.

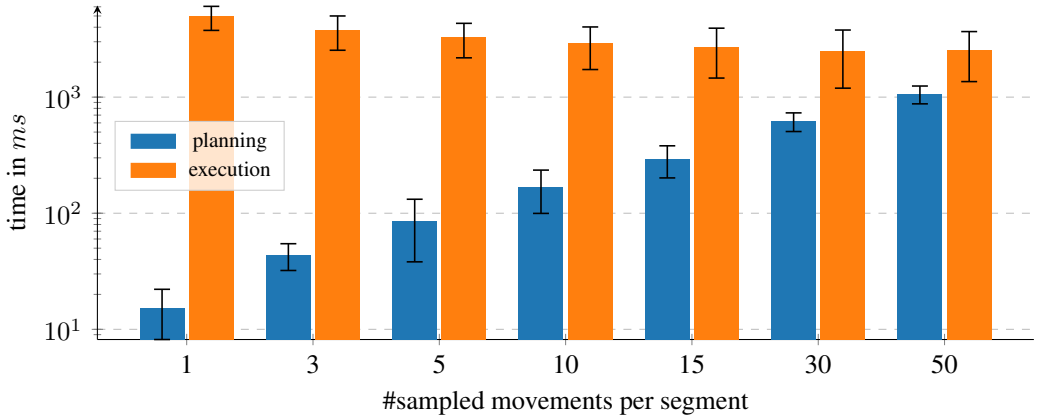

Figure 3: Mean and standard deviation of the computational time required for planning a segment and the execution time on the robot for each segment. All evaluations are calculated over 100 segments of the following task for each number of sampled movements per segment, where each segment consists of 30 simulated timesteps. Note that the time axis is logarithmic.

3.1 Online Motion Planning with a Simulated Robotic Arm

We used a simulated KUKA LWR robotic arm with a Cartesian tracking controller to follow the reference trajectories generated by our model. In this experiment, the task was to follow a given sequence of via points, forming an *eight-shape*, see Figure 2. By activating the via points successively as target positions using appropriate context neurons, the model generates online a trajectory following the given shape. The model has no knowledge about the task, i.e., the target positions or their activation pattern. We considered a two-dimensional workspace that spans $[-1, 1]$ for both dimensions. Each dimension is encoded by 15 state neurons, thus resulting in 225 state neurons using full population coding. The transition model is given by Gaussian distributions centered at the preferred positions of the neurons. For the evaluations, we tested three different implementations of the context neuron input weights: (1) setting the weights corresponding to the euclidean distance as in [19], (2) according to student's t distributions and (3) to generalized error distributions, both centered at the context neuron positions.

We analyzed the *efficiency* in terms of detour compared to the beeline between start and target position shown in Figure 2 and the computational time shown in Figure 3. All evaluations were done using different amounts of samples per segment and generating 100 segments per configuration. We additionally evaluated the influence of the number of samples used for estimating the final mental movement trajectory. Each planned segment consists of 30 discrete timesteps. The execution time is determined based on the covered distance of the planned segment and the desired velocity of the movements on the robot, here set to $0.05m/s$.

For a larger number of samples the mental plan gets smoother and follows the direct paths to subsequent via points. The computational time increases, however, planning can still be done faster than the execution time, enabling the online planning approach and additionally providing *unused* time for online learning. This time overhead allows the framework to plan in advance during execution as well as to add an online learning phase without interrupting the continuous movement. Note that, the computational time can be decreased by parallelizing the sampling process as the samples are independent. Additionally, also the online learning phase may run in parallel.

The effect on the generated movements when using different distributions for generating the context neuron's weights is highlighted in Figure 2. The heavy tailed distributions generate more efficient trajectories as they are more focused on the target positions. In contrast, if the current position is too far away, they sometimes have trouble to guide the activity towards the target without an intermediate target. Common for all settings is the fact that using multiple samples per segment increases the performance.

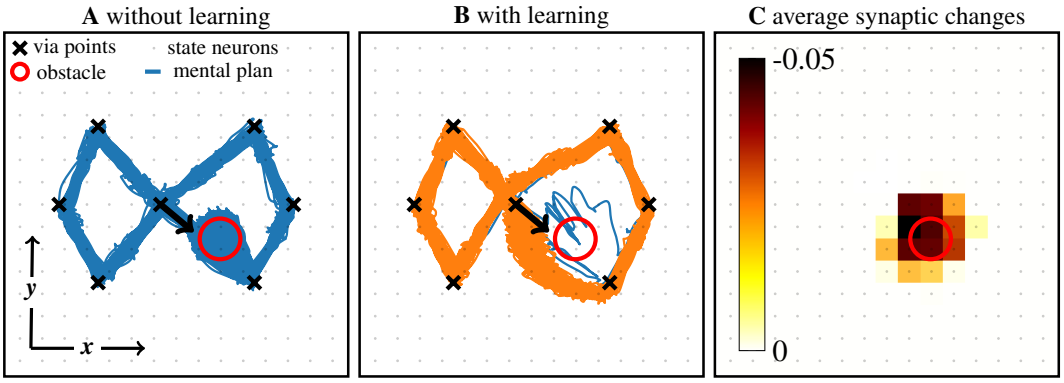

Figure 4: In **A** and **B** the continuously planned mental movement consisting of 2000 segments following the seven via points is shown. The black arrow indicates the direction of the motion. The orange part of the mental movement indicates the last 1970 segments. The red circle depicts the *unknown* workspace constraint. **C** shows the learning effect in the model, highlighted as the *average* change in *synaptic input* each neuron receives.

3.2 Online Model Adaptation to Unknown Workspace Constraints

In this experiment, we want to show the model's ability to adapt continuously during the execution of the planned trajectory. For this purpose, the model has to follow the eight-shape given by seven via points as in the previous experiment. The setup is altered by adding an obstacle that blocks the direct connection between two via points, see Figure 4A. The model does not know anything about this constraint or its existence. We implemented the task in a simple simulation, where the execution of the mental movement trajectory is simulated by adding noise and assume a safety controller that stops the position controller when approaching the obstacle.

The effect of the online learning process is shown in Figure 4, where the mental movement trajectory is shown without and with online adaption. Without online adaption the model struggles to reach the via points. This behavior is shown in Figure 4A. Most of the segments are at the constraint and the model only occasional finds an ineffective solution due to the stochasticity in the mental movement generation. If we activate the proposed intrinsically motivated online learning, the model initially tries to enter the invalid area but recognizes, due to the perceived feedback of the stopped movement, that there is something *unexpected* and adapts accordingly. This fast adaptation behavior is illustrated in Figure 4B. By adapting online to the perceived cognitive dissonances, the model generates new valid solutions much faster. In particular, out of 2000 planned segments, the via point behind the constraint is active in 23% of the segments with online learning enabled and in 70% without adaption. This indicates that the model reaches the blocked via point much faster when online learning is enabled. Moreover, after few samples that collide with the obstacle, the model already learned to avoid this area completely, which is shown in Figure 4B. Note that, the orange part of the mental movement shows the last 1970 out of 2000 planned segments and highlights the efficient learning effect. The fast adaption is also reflected in the number of segments that the model needs to reach the target position behind the obstacle, shown in Figure 5. Already when trying to reach the position for the first time, the model immediately adapted to the recognized change and learned to avoid the blocked area.

When the model adapts, the incoming synaptic weights of neurons with preferred positions at the blocked area are decreased. Thus, transitions to these neurons, and with that movements into these areas, get less likely. Moreover, as the learning is guided using the intrinsic motivation signal motivated by cognitive dissonance, the model only adapts in affected areas. These local changes are shown in Figure 4C. The neurons around the constraint receive strong inhibition during adaption. This inhibition hinders the network to sample mental movements in affected areas, i.e., the model has learned to avoid these areas. Due to the state neurons resolution (here 15 per dimension), the influence of the constraint is larger than it actual is. Using more state neurons to increase the spatial resolution of the modeled workspace lowers the size of the influenced area.

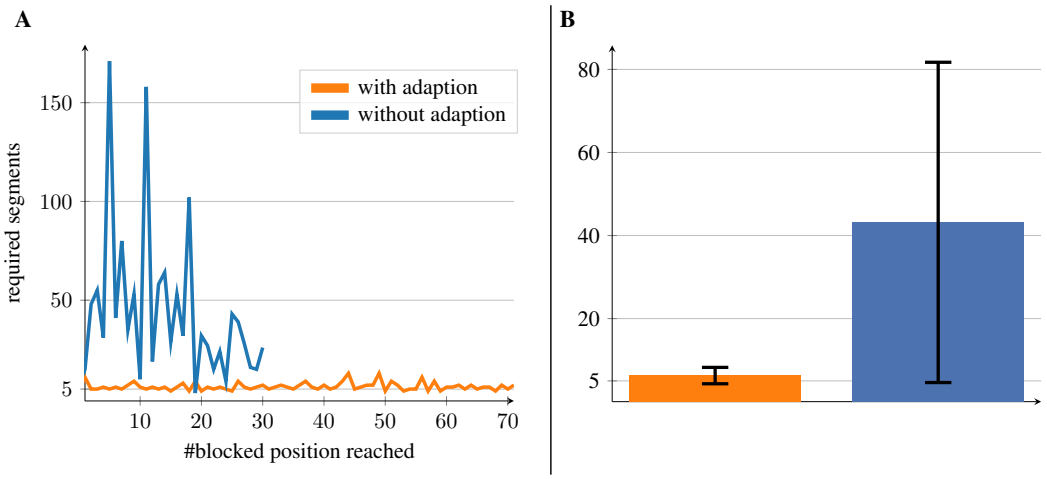

Figure 5: The number of segments required to reach the blocked target position is shown. In **A** plotted over the number of how often this target was reached and in **B** as mean and standard deviation over the whole execution.

4 Conclusion

In this paper, we introduced a novel framework for online motion planning, that can adapt online by using an intrinsic motivation signal which encodes the mismatch between mental expectation and perceived observation. Sample-efficient learning is achieved through a mental replay strategy of experienced situations and highlighted by learning obstacle avoidance strategies from a couple of collisions. This sample-efficient and task-independent adaptation lowers the required expert knowledge and makes the approach promising for learning on robotic systems, for reusability and sets the method apart from classical motion planning methods. The approach can be used for exploration of unknown environments, learning of unknown workspace and body constraints, or for updating robot models, for example, if a joint gets broken [21]. With the presented intrinsic motivation signal, the agent can adapt to novel environments by reacting to the perceived feedback. Active exploration, and thereby *forgetting* of previously learned constraints and finding novel solutions, is a next step that we plan to investigate with intrinsic motivation signals mimicking curiosity [22]. Equipping real robotic systems with such adaptation mechanisms is currently investigated on a KUKA LWR arm and is an important step towards autonomous and lifelong-learning agents.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No #713010 (GOAL-Robots) and #640554 (SKILLS4ROBOTS).

References

[1] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida. Cognitive developmental robotics: A survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34, 2009.

[2] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, 2003.

[3] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.

[4] S. Thrun and T. M. Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15 (1-2):25–46, 1995.

[5] J. Weng. Developmental robotics: Theory and experiments. *International Journal of Humanoid Robotics*, 1(02):199–236, 2004.

[6] R. M. Ryan and E. L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[7] R. M. Ryan and E. L. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1):68, 2000.

[8] G. Baldassarre and M. Mirolli. Intrinsically motivated learning systems: an overview. In *Intrinsically motivated learning in natural and artificial systems*, pages 1–14. Springer, 2013.

[9] A. G. Barto, S. Singh, and N. Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*, pages 112–19. Citeseer, 2004.

[10] S. Hart and R. Grupen. Learning generalizable control programs. *IEEE Transactions on Autonomous Mental Development*, 3(3):216–231, 2011.

[11] U. Nehmzow, Y. Gatsoulis, E. Kerr, J. Condell, N. Siddique, and T. M. McGuinnity. Novelty detection as an intrinsic motivation for cumulative learning robots. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 185–207. Springer, 2013.

[12] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

[13] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

[14] L. Festinger. Cognitive dissonance. *Scientific American*, 1962.

[15] J. Kagan. Motives and development. *Journal of personality and social psychology*, 22(1):51, 1972.

[16] P.-Y. Oudeyer and F. Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in Neurorobotics*, 1:6, 2009.

[17] D. J. Foster and M. A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, 2006.

[18] E. Rueckert, D. Kappel, D. Tanneberg, D. Pecevski, and J. Peters. Recurrent spiking networks solve planning tasks. *Nature PG: Scientific Reports*, 2016.

[19] D. Tanneberg, A. Paraschos, J. Peters, and E. Rueckert. Deep spiking networks for model-based planning in humanoids. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 656–661. IEEE, 2016.

[20] B. E. Pfeiffer and D. J. Foster. Hippocampal place cell sequences depict future paths to remembered goals. *Nature*, 497(7447):74, 2013.

[21] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[22] P.-Y. Oudeyer, J. Gottlieb, and M. Lopes. Intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies. *Progress in brain research*, 229:257–284, 2016.