

Efficient Online Adaptation with Stochastic Recurrent Neural Networks

Daniel Tanneberg¹, Jan Peters^{1,2} and Elmar Rueckert¹

Abstract—Autonomous robots need to interact with unknown and unstructured environments. For continuous online adaptation in lifelong learning scenarios, they need sample-efficient mechanisms to adapt to changing environments, constraints, tasks and capabilities. In this paper, we introduce a framework for online motion planning and adaptation based on a bio-inspired stochastic recurrent neural network. By using the intrinsic motivation signal *cognitive dissonance* with a mental replay strategy, the robot can learn from few physical interactions and can therefore adapt to novel environments in seconds. We evaluate our online planning and adaptation framework on a KUKA LWR arm. The efficient online adaptation is shown by learning unknown workspace constraints sample-efficient within few seconds while following given via points.

I. INTRODUCTION

One of the major challenges in robotics is the concept of developmental robots [1], [2], [3], [4], i.e., robots that adapt autonomously through lifelong learning. Although a lot of research has been done for learning tasks autonomously in recent years, experts with domain knowledge are still required to define and guide the learning problem, e.g., for reward shaping, for providing demonstrations or for defining the tasks that should be learned. In a fully autonomous self-adaptive robot however, these procedures should be carried out by the robot itself. Thus, the robot should be equipped with mechanisms to decide when, what, and how to learn [5].

Furthermore, in the case of robot movements, planning a movement, executing it, and learning from the results should be integrated in a continuous online framework. This idea is investigated in iterative learning control approaches [6], [7], which can be seen as a primitive adaptation mechanism that tracks given repetitive reference trajectories. More complex adaptation is investigated in model-predictive control approaches [8], [9] that simultaneously plan, execute and re-plan motor commands. However, the used models are fixed and cannot adapt to new challenges.

In this work, we combine the scheduling of model-predictive control with learning from intrinsic motivation signals for online adaptation of transition models for motion planning. This intrinsic motivation signal tells the agent where its model is incorrect and updates the model with this mismatch.

From autonomous mental development in humans it is known that intrinsic motivation is a strong factor for learn-

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

¹Intelligent Autonomous Systems, Technische Universität Darmstadt, Darmstadt, Germany, {daniel, elmar}@robot-learning.de

²Robot Learning Group, Max-Planck Institute for Intelligent Systems, Tübingen, Germany, mail@jan-peters.net

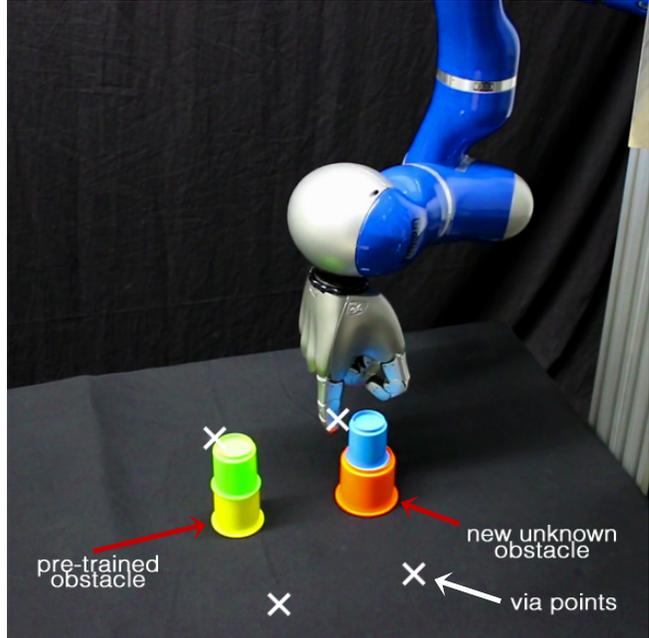


Fig. 1: Experimental setup. Picture of the used setup for online planning and learning, showing the KUKA LWR arm and the environment with two obstacles. The crosses depict the via points that need to be reached successively. One obstacle was pre-trained with a realistic dynamic simulation of the robot and the second is learned additionally online on the real system.

ing [10], [11]. The concept of intrinsic motivated learning has inspired many studies about artificial and robotic systems, e.g. [12], [13], which investigate intrinsic motivated learning in a reinforcement learning setup. Typically, such systems learn the consequences of actions and choose the action that maximizes a novelty or prediction related reward signal [14], [15], [16]. Furthermore, the majority of the related work on intrinsic motivated learning focuses on concepts and simulations, and only few applications to real robotic systems exist [17], [18].

The contribution of this work is the application of a novel framework for probabilistic online motion planning and learning that combines the concepts of model-predictive and iterative learning control on a real robotic system. Based on a recent bio-inspired stochastic recurrent neural network, online adaptation is done by updating the recurrent synaptic weights encoding the state transition model. The model can adapt efficiently to novel environments without specifying a learning task or other human input within seconds by using an intrinsic motivation signal and a mental replay strategy. The framework is evaluated on a KUKA LWR arm in simulation and on the real system shown in Figure 1.

II. RELATED WORK

In early work on intrinsic motivated reinforcement learning, the intrinsic motivation signal was used to learn a hierarchical collection of skills autonomously [12]. An incrementally learned skill collection based on implementations of the intrinsic motivations signals novelty and prediction error is shown in [19] and evaluated in simulation.

A competence based learning approach is presented in [20], where an agent is equipped with a set of skills trying to gain competence about its environment. The approach is evaluated in a gridworld domain.

In [17], the *intelligent adaptive curiosity* system is used to lead a robot to maximize its learning progress, i.e., guiding the robot to situations, that are neither too predictable nor too unpredictable. The mechanism is used on a robot that learns to manipulate objects. The idea is to equip agents with mechanisms *computing* the degree of novelty, surprise, complexity or challenge from the learning robots point of view and use these signals for guiding the learning process.

Using intrinsic motivation for learning a repertoire of movement primitives and subsequently how to sequence and generalize them, is shown in [18] for object detection and manipulation tasks.

Multiple prediction based intrinsic motivation signals were investigated on a simulated robot arm learning reaching movements in [16].

A coherent theory and fundamental investigation of using intrinsic motivation in machine learning is given in [21], where it is stated that the improvement of prediction can be used as an intrinsic reinforcement for efficient learning. Another comprehensive overview of intrinsically motivated learning systems and how to investigate these is given in [13]. In [22] a psychological view on intrinsic motivation is discussed and a formal typology of computational approaches for studying such learning systems is presented.

Intrinsic motivation signals have been used for incremental task learning, acquiring skill libraries, learning perceptual patterns and for object manipulation. For the goal of fully autonomous robots however, the ability to focus and guide learning independently from tasks, specified rewards and human input is crucial.

The main contribution of this paper is the demonstration of how an agent can use intrinsic motivation for efficient online adaptation without an explicit task to learn. We implement this learning approach into a biologically inspired stochastic recurrent neural network for motion planning [23], [24]. The method is embedded into a novel proposed framework for continuous online motion planning and learning that combines the scheduling concept of model-predictive control with the adaptation of iterative learning control.

The online model adaptation is modulated by an intrinsic motivation signal that is inspired by cognitive dissonance [25], [26]. We use a knowledge-based model of intrinsic motivation [22] that describes the divergence of the expectation to the observation. Additionally, to intensify the effect of the

experience, we use a mental replay mechanism, what has been proposed to be a fundamental concept in human learning [27]. This mental replay is implemented by exploiting the stochastic nature of spike encodings of trajectories to generate multiple sample encodings for every experienced situation. We will show that as a result of these two learning mechanisms, the model can adapt online to unknown environments efficiently on a real robotic system.

III. MOTION PLANNING WITH STOCHASTIC RECURRENT NEURAL NETWORKS

The proposed framework builds on the model recently developed by [23], where it was shown that stochastic spiking networks can solve motion planning tasks optimally. Furthermore, in [24] 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 was shown.

Inspired by neuroscientific findings on the mental path planning of rodents, the model mimics the behavior of hippocampal place cells. It was shown that the neural activity of these cells is correlated not only with actual movements, but also with future mental plans. This bio-inspired motion planner consists of stochastic spiking neurons forming a multi-layer 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 $k, i \in K$. The state neurons are uniformly spaced within the modeled state space, forming a grid where every neuron has a preferred position. The context neurons are Gaussian distributed locally around the location they encode. Considering the discrete time case, the neurons probability to spike is proportional to their membrane potential

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

where $\tilde{v}_i(t)$ and $\tilde{y}_j(t)$ denote the presynaptic potential from neurons $i \in K$ and $j \in N$ respectively. This definition implements a simple stochastic spike response model [28]. The binary activity of the state neurons is denoted by $\mathbf{v}_t = (v_{t,1}, \dots, v_{t,K})$, where $v_{t,k} = 1$ if neuron k spikes at time t and $v_{t,k} = 0$ otherwise. Analogously, \mathbf{y}_t describes the activity of the context neurons. The synaptic weights θ which connect context neurons to state neurons provide task related information. They modulate the random walk behavior of the state neurons towards goal directed movements. In contrast to [24], where θ was set according to the euclidean distance directly between context and state neurons, we set θ according to generalized error distributions. At each context neuron position such a distribution is located and the weights to the state neurons are drawn from this distribution using the distance between the connected neurons. By this mechanism, the context neurons – if they encode a desired position –

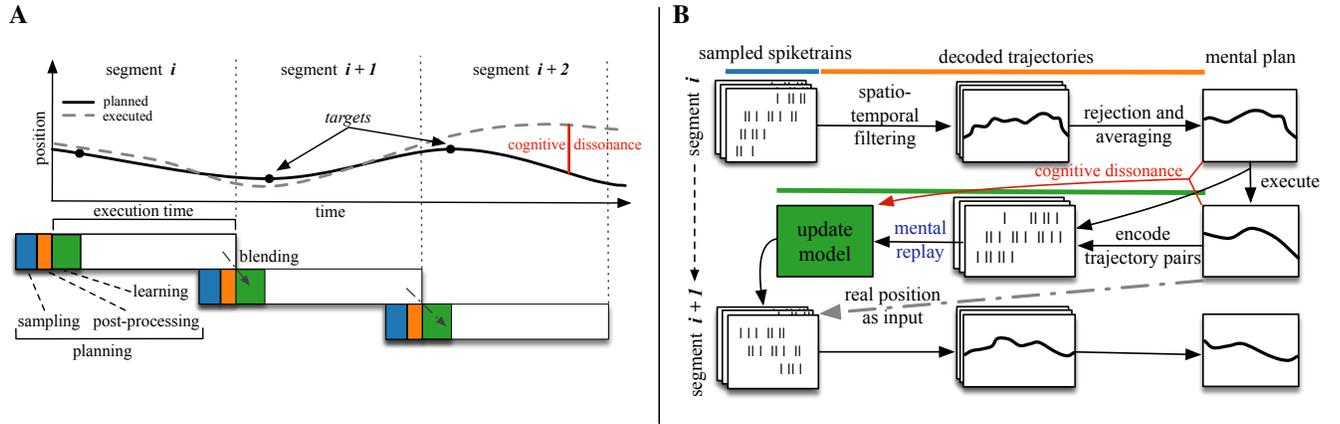


Fig. 2: Conceptual sketch of the framework. **A** shows the online planning concept of using short segments. On the upper part the idea of cognitive dissonance is illustrated with a planned and executed trajectory. The three steps learning, sampling and post-processing are organized such that they are performed at the end of the execution of the previously planned segment. **B** shows the process with two segments in detail, including sampling of movements, rejection and averaging for creating the mental plan. The executed segment provides feedback for planning the next segment and the matching mental and executed trajectory pairs are considered for updating the model based on their cognitive dissonance.

install a *gradient* towards the position they encode. Context neurons spike with a fixed probability and timing based on the kind of constraint they encode, e.g., are active until the associated via-point is reached. This task-related input can also be learned using reinforcement learning techniques as shown in [23], but needs to be relearned for every new task.

For planning, the stochastic network encodes a distribution

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

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 [29]. The binary neural activity of the grid-aligned state neurons encodes the continuous system state \mathbf{x}_t , e.g., end-effector position, using the decoding scheme

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

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 . 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.

IV. ONLINE MOTION PLANNING AND LEARNING

For efficient online adaptation, the model should be able to react during the execution of a planned trajectory. Therefore, we consider a short time horizon instead of planning complete movement trajectories over a long time horizon. This short time horizon sub-trajectory is called a *segment*. A trajectory κ 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. Furthermore, it allows the network to update itself during planning, providing a mechanism for online model learning and adaptation to changing environments or constraints. The general idea of how we enable the model to plan and adapt online is illustrated in Figure 2.

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 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. The learning part can be done right after a segment execution is finished.

As the recurrent network consists of stochastic spiking neurons, the network models a distribution over movement trajectories rather than a single solution. In order to create a smooth movement trajectory, we average over multiple samples drawn from the model when planning each segment. Before the final mental movement trajectory is created by averaging over the drawn samples, we added a sample rejection mechanism. As spiking networks can encode arbitrary complex functions, the model can encode multi-modal movement distributions. Imagine that the model faces a known obstacle that can be avoided by going around either left or right. Drawn movement samples can contain both solutions and when averaging over the samples, the robot would crash into the obstacle. Thus, only samples that encode the same solution should be considered for averaging.

Clustering of samples could solve this problem, but as our framework has to run online, this approach is too expensive. Therefore, we implemented a heuristic based approach that

uses the angle between approximated movement directions as distance. First a reference movement sample is chosen such that its average distance to 1/3 of the population is minimal. Subsequently only movement samples with a movement direction that differs maximally by 90 degrees to the reference sample are considered for averaging.

The feedback provided by the executed movement is incorporated before planning the next segment in two steps. First, the actual position is used to initialize the sampling of the next segment such that planning starts from where the robot actually is, not where the previous mental plan indicates and, second, the executed movement is used for updating the model.

A. Efficient Online Model Adaptation using Intrinsic Motivation Signals and Mental Replanning Strategies

The online update of the spiking network model is based on the contrastive divergence (CD) [30] based learning rules derived recently in [24]. CD draws multiple samples from the current model and uses them to approximate the likelihood gradient. In practice often only one single sample is used for this process. The general CD update rule for learning parameters Θ of some function $f(x; \Theta)$ is given by

$$\Delta\Theta = \left\langle \frac{\partial \log f(x; \Theta)}{\partial \Theta} \right\rangle_{\mathbf{X}^0} - \left\langle \frac{\partial \log f(x; \Theta)}{\partial \Theta} \right\rangle_{\mathbf{X}^1}, \quad (1)$$

where \mathbf{X}^0 and \mathbf{X}^1 denote the state of the Markov chain after 0 and 1 cycles respectively, i.e., the data and the model distribution. We want to update the state transition function $\mathcal{T}(\mathbf{v}_t | \mathbf{v}_{t-1})$, which is encoded in the synaptic weights 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 the online scenario, we need to make several changes. In the original work, the model was initialized with inhibitory connections. Thus, no movement can be sampled from the model for exploration until the learning process has converged. This is not suitable in the online learning scenario, as a *working* model for exploration is required, i.e., the model needs to be able to generate movements at any time. Therefore, we initialize the synaptic weights between the state neurons using Gaussian distributions, i.e., a Gaussian 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.

This process initializes the transition model with an uniform prior, where for each position, transitions in all directions are equally likely. 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, the learning rule has to be adapted 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 a sample from the training data

distribution and the mental trajectory as a sample from the model distribution in Equation (1).

For encoding the mental and executed trajectories into spiketrains, inhomogeneous Poisson processes with the Gaussian responsibilities of each state neuron at each timestep as time-varying input are used as in [24]. 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 use an intrinsic motivation signal that is motivated by cognitive dissonance [25], [26]. Concretely, the dissonance between the mental movement trajectory generated by the stochastic network and the actual executed movement is used. Thus, if the executed movement is similar with the generated mental movement, the update is small, while a stronger dissonance leads to a larger update. In other words, learning is guided by the mismatch between expectation and observation.

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

$$w_{k,i} \leftarrow w_{k,i} + \alpha_t \Delta w_{k,i} \quad \text{with} \quad \alpha_t = \|\kappa_t^{(m)} - \kappa_t^{(e)}\|_2^2 \quad (2)$$

$$\text{and} \quad \Delta w_{k,i} = \tilde{v}_{t-1,k} \tilde{v}_{t,i} - \tilde{v}_{t-1,k} v_{t,i},$$

where \tilde{v}_t is generated from the actual executed movement trajectory $\kappa_t^{(e)}$ and v_t from the mental trajectory $\kappa_t^{(m)}$. For a more detailed derivation of this spiking CD learning rule, we refer to [24].

To stabilize the learning progress w.r.t. to the noisy observations, we limit α_t in our experiments to $\alpha_t \in [0, 0.3]$ and use a learning threshold of 0.03, such that the model is only updated when the cognitive dissonance is larger than this threshold. Note that, in the experiments α_t did not reach the limit, i.e., this safety limit did not influence the results. With this intrinsic motivated learning factor, the update is regulated according to the model error, i.e., only *invalid* parts of the model are updated.

As the encoding of the trajectories into spiketrains is a stochastic operation, we can obtain a whole population of encodings from a single trajectory. Therefore, populations of training and model data pairs can be generated and used for

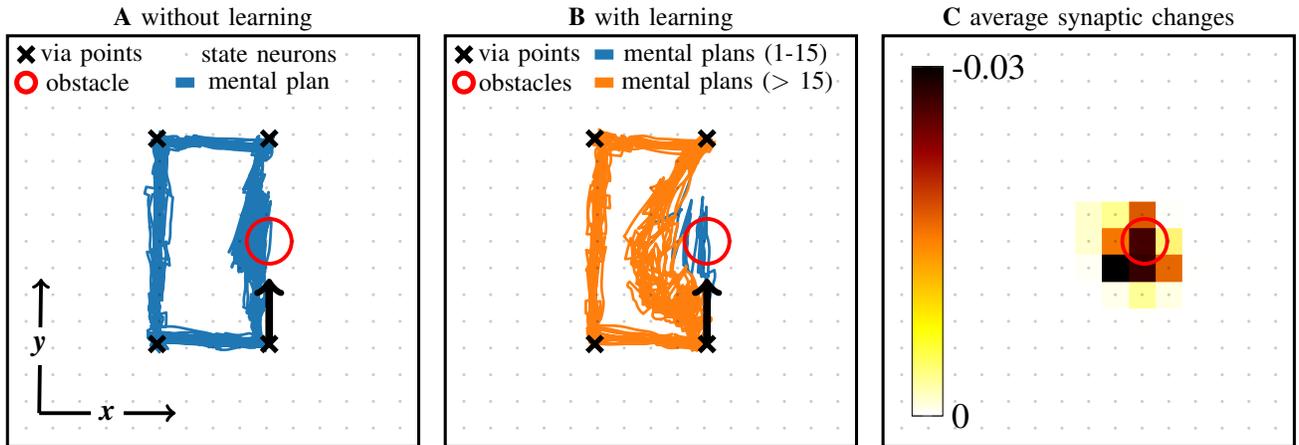


Fig. 3: Simulation results. Online adaptation with the KUKA LWR arm in simulation. **A** and **B** show the continuously planned mental movement consisting of 500 segments following the via points. The black arrow indicates the direction of the motion. The orange part of the mental movement indicates the last 485 segments. The red circle depicts the *unknown* workspace constraint. **C** shows the learning effect in the model as the *average* change of *synaptic input* of each neuron. Total execution time was 400s, where the adaption was done within 4 – 5s after colliding with the obstacle for the first time.

learning. By using this mental replay approach, we apply multiple updates from a single interaction. The two mechanisms, using intrinsic motivation signals for scaling the updates and mental replay strategies, lower the required number of experienced situations, which is a crucial requirement for learning with robotic systems.

V. EXPERIMENTS

We conducted two experiments to evaluate the proposed framework for online planning and learning based on intrinsic motivation and mental replay. In both experiments the model had to follow a path given by via points that are activated successively one after each other. Each via point remains active until it is reached by the robot. In the first experiment a realistic dynamic simulation of the KUKA LWR arm was used. The model had to adapt to an unknown obstacle that blocks the direct path between two via points. In a second experiment, we used the pre-trained model from the simulation in the real robotic arm experiment. The robot had to adapt online to a second unknown obstacle.

A. Experimental setup

In the first experiment, we used a realistic dynamic simulation of the KUKA LWR arm with a Cartesian tracking controller – using inverse kinematics to obtain reference joint trajectories and inverse dynamics to execute them – to follow the reference trajectories generated by our model. The tracking controller is equipped with a safety controller that stops the tracking when an obstacle was hit. The task was to follow a given sequence of four via points, where an obstacle blocks the direct path between two via points. In the real robot experiment, the same tracking and safety controllers were used. The tasks are shown Figure 3 and Figure 4, and the real robot setup can be seen in Figure 1.

By activating the via points successively one after each other as target positions using appropriate context neurons, the model generates online a trajectory tracking the given

shape. For creating a mental trajectory, we used 40 samples, resulting in an average planning time under one second without parallelization. The model has no knowledge about the task or the constraint, i.e., the target via points, their activation pattern and the obstacle. We considered a two-dimensional workspace that spans $[-1, 1]$ for both dimensions encoding the 60×60 cm operational space of the robot. Each dimension is encoded by 15 state neurons, which results in 225 state neurons using full population coding. The transition model is given by Gaussian distributions centered at the preferred positions of the neurons as explained in Section IV-A. For the mental replay we used 20 iterations, i.e., 20 pairs of training data were generated for each executed movement.

B. Efficient Online Model Adaptation in Simulation

In this experiment, we want to show the model’s ability to adapt continuously during the execution of the planned trajectory. The effect of the online learning process is shown in Figure 3, where the mental movement trajectory is shown without and with online adaption. Without online adaption (Fig. 3A) the model struggles to reach the via points. 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 (Fig. 3B), the model initially tries to enter the invalid area but recognizes, due to the perceived feedback of the interrupted movement, the *unexpected* obstacle. As a result the model adapts and no planned movement after the 15th segment hits the obstacle anymore. This adaptation happens within 4 to 5 seconds from only 7 to 9 experiences.

By adapting online to the perceived cognitive dissonances, the model generates new solutions avoiding the obstacle. Moreover, after the first successful avoidance of the obstacle, the model already learned to avoid this area completely. This efficient adaptation is depicted in Figure 3B, where the blue trajectory indicates the first 15 planned segments and the orange trajectory the subsequently planned segments.

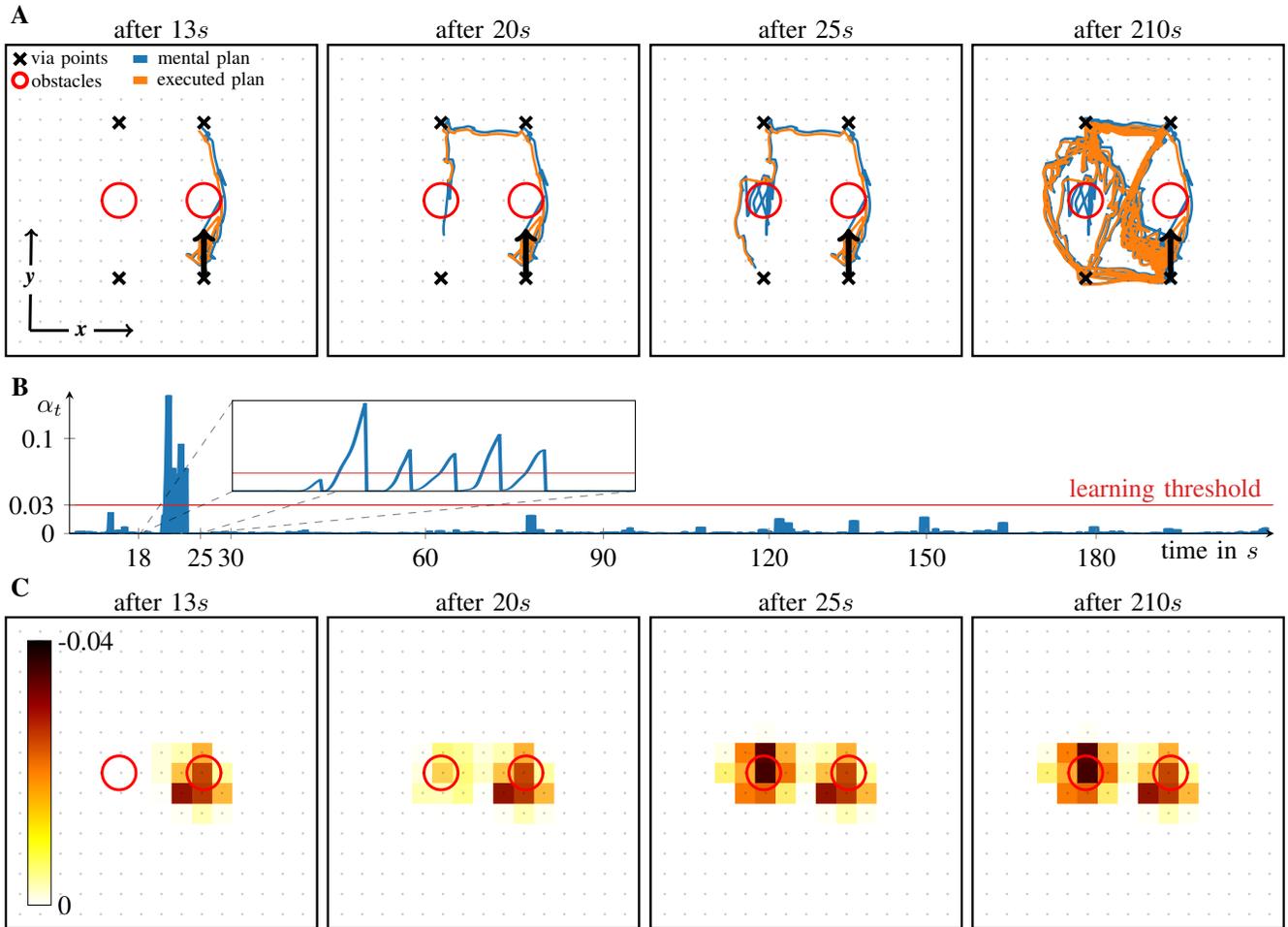


Fig. 4: Real robot results. Online adaptation with the real KUKA LWR arm, initialized with the simulation results, i.e., the right obstacle is already learned and known. The left obstacle is added to the real environment and is unknown at the beginning. **A** and **C** show snapshots at the indicated execution times with red circles illustrating the obstacles. **A** shows the mental (blue) and executed (orange) movement plans, where via points are depicted as black crosses and the black arrow indicates the movement direction. **C** shows the learning progress in the model depicted as the *average* change of *synaptic input* of each neuron. **B** shows the continuous cognitive dissonance signal α_t over execution time and the learning threshold, where the inlay highlights the time horizon between 18s and 25s. As the plots show, after colliding with the new obstacle the first time, the model adapts within 4 to 5 seconds.

Local spatial adaptation: When the model adapts, the incoming synaptic weights of neurons with preferred positions at the blocked area are decreased. Thus, state transitions to these neurons get less likely. Moreover, as the weight update is modulated by the intrinsic motivation signal, denoted by α_t in Equation (2), the model only adapts in affected areas. These local changes are shown in Figure 3C. The neurons around the constraint are inhibited after adaptation. 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 actually is. Using more state neurons to increase the spatial resolution of the modeled workspace lowers the size of the influenced area.

C. Model Transfer and Online Adaptation on the Robot

In the second experiment we show that the model learned in simulation can be transferred directly onto the real system

and, furthermore, that the efficient online adaptation can be done on a real and complex robotic system. Therefore, we adapted the simulated task of following the four given via points. Additionally to the obstacle that was already present in simulation, we added a second unknown obstacle to the real environment. The setup is shown in Figure 1. The model parameters were the same as in the simulation experiment. On average, an experimental trial took about 3.5 minutes and Figure 4 shows the execution and adaptation over time.

As we started with the model trained in simulation, the robot successfully avoids the first obstacle right away and no adaptation is needed (Fig. 4 first column). After about 20 seconds, the robot collides with the second obstacle and adapts to it in about 4 – 5 seconds (Fig. 4 second and third column). The mismatch between the mental plan and the executed trajectory is above the learning threshold and the online adaptation is triggered and scaled with α_t (Fig. 4B).

The adaption of the synaptic weights is only applied within

few seconds and illustrated in Figure 4C. To highlight that, we depicted the mental plans and the executed plans after 20 and 25 seconds in Figure 4A. For the corresponding execution time, the cognitive dissonance signal shows a significant mismatch that leads to the fast adaptation, illustrated in the inset of Figure 4B. After the successful avoidance of the new obstacle, the robot performs the following task while avoiding both obstacles now and no further weight changes are triggered.

Learning multiple solutions: Even though during the adaptation phase the model only experienced one successful strategy to avoid the obstacle, it is able to generate different solutions, i.e., bypassing the obstacle left or right. That is also true for the obstacle learned in simulation, where only one solution was experienced in simulation, and the model generates the second solution on the real system without further adaptation. This is best shown in Figure 3B and Figure 4A. The feature of generating different solutions is enabled by the model's intrinsic stochasticity and the ability of spiking neural networks to encode arbitrary complex functions.

VI. CONCLUSION

In this paper, we applied a novel framework for probabilistic online motion planning with an efficient online adaptation mechanism on a real robotic system. This framework is based on a recent bio-inspired stochastic recurrent neural network. The online learning is modulated by an intrinsic motivation signal inspired by *cognitive dissonance* that encodes the mismatch between mental expectation and observation and relates to a tracking error. By combining this learning mechanism with a mental replay strategy of experienced situations, sample-efficient online adaptation within seconds is achieved. This fast adaptation is highlighted in a simulated and real robotic experiment, where the model adapts to an unknown environment within 4 – 5 seconds from few collisions with the unknown obstacles without a specified learning task or other human input. Learning to avoid unknown obstacles by updating the state transition model encoded in the recurrent synaptic weights is a proof of concept for the aim of recovering from failures. In future therefore, we want to combine this approach with the factorized population coding from [24] to scale to higher dimensional problems and apply the framework to recover from failure tasks with broken joints [31], [32]. Equipping robotic systems with such adaptation mechanisms is an important step towards autonomously developing and lifelong-learning systems.

REFERENCES

- [1] S. Thrun and T. M. Mitchell, "Lifelong robot learning," *Robotics and autonomous systems*, 1995.
- [2] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, 2003.
- [3] J. Schmidhuber, "Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts," *Connection Science*, 2006.
- [4] 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*, 2009.
- [5] J. Weng, "Developmental robotics: Theory and experiments," *Int. Journal of Humanoid Robotics*, 2004.
- [6] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, 2004.
- [7] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, 2006.
- [8] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.
- [9] A. Ibanez, P. Bidaud, and V. Padois, "Emergence of humanoid walking behaviors from mixed-integer model predictive control," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.
- [10] R. M. Ryan and E. L. Deci, "Intrinsic and extrinsic motivations: Classic definitions and new directions," *Contemporary educational psychology*, 2000.
- [11] R. M. Ryan and E. L. Deci, "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being," *American psychologist*, 2000.
- [12] A. G. Barto, S. Singh, and N. Chentanez, "Intrinsically motivated learning of hierarchical collections of skills," in *Int. Conf. on Development and Learning*, 2004.
- [13] G. Baldassarre and M. Mirolli, "Intrinsically motivated learning systems: an overview," in *Intrinsically motivated learning in natural and artificial systems*, Springer, 2013.
- [14] U. Nehmzow, Y. Gatsoulis, E. Kerr, J. Condell, N. Siddique, and T. M. McGuinness, "Novelty detection as an intrinsic motivation for cumulative learning robots," in *Intrinsically Motivated Learning in Natural and Artificial Systems*, Springer, 2013.
- [15] A. Stout, G. D. Konidaris, and A. G. Barto, "Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning," tech. rep., Massachusetts Univ., Dept. of Computer Science, 2005.
- [16] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Intrinsic motivation signals for driving the acquisition of multiple tasks: a simulated robotic study," in *Int. Conf. on Cognitive Modelling (ICCM)*, 2013.
- [17] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE transactions on evolutionary computation*, 2007.
- [18] S. Hart and R. Grupen, "Learning generalizable control programs," *IEEE Transactions on Autonomous Mental Development*, 2011.
- [19] J. H. Metzen and F. Kirchner, "Incremental learning of skill collections based on intrinsic motivation," *Frontiers in neurorobotics*, 2013.
- [20] A. Stout and A. G. Barto, "Competence progress intrinsic motivation," in *Int. Conf. on Development and Learning (ICDL)*, 2010.
- [21] J. Schmidhuber, "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)," *IEEE Transactions on Autonomous Mental Development*, 2010.
- [22] P.-Y. Oudeyer and F. Kaplan, "What is intrinsic motivation? a typology of computational approaches," *Frontiers in Neurorobotics*, 2009.
- [23] E. Rueckert, D. Kappel, D. Tanneberg, D. Pecevski, and J. Peters, "Recurrent spiking networks solve planning tasks," *Nature PG: Scientific Reports*, 2016.
- [24] D. Tanneberg, A. Paraschos, J. Peters, and E. Rueckert, "Deep spiking networks for model-based planning in humanoids," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2016.
- [25] L. Festinger, "Cognitive dissonance," *Scientific American*, 1962.
- [26] J. Kagan, "Motives and development," *Journal of personality and social psychology*, 1972.
- [27] D. J. Foster and M. A. Wilson, "Reverse replay of behavioural sequences in hippocampal place cells during the awake state," *Nature*, 2006.
- [28] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
- [29] L. Buesing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons," *PLoS computational biology*, 2011.
- [30] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, 2002.
- [31] D. J. Christensen, U. P. Schultz, and K. Stoy, "A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots," *Robotics and Autonomous Systems*, 2013.
- [32] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, 2015.