Final demonstrator of the deep learning and SNN model for robot control in a conditional task using an industrial manipulator UR5 (SPIKEFERENCE) *(D3.16 – SGA3)*

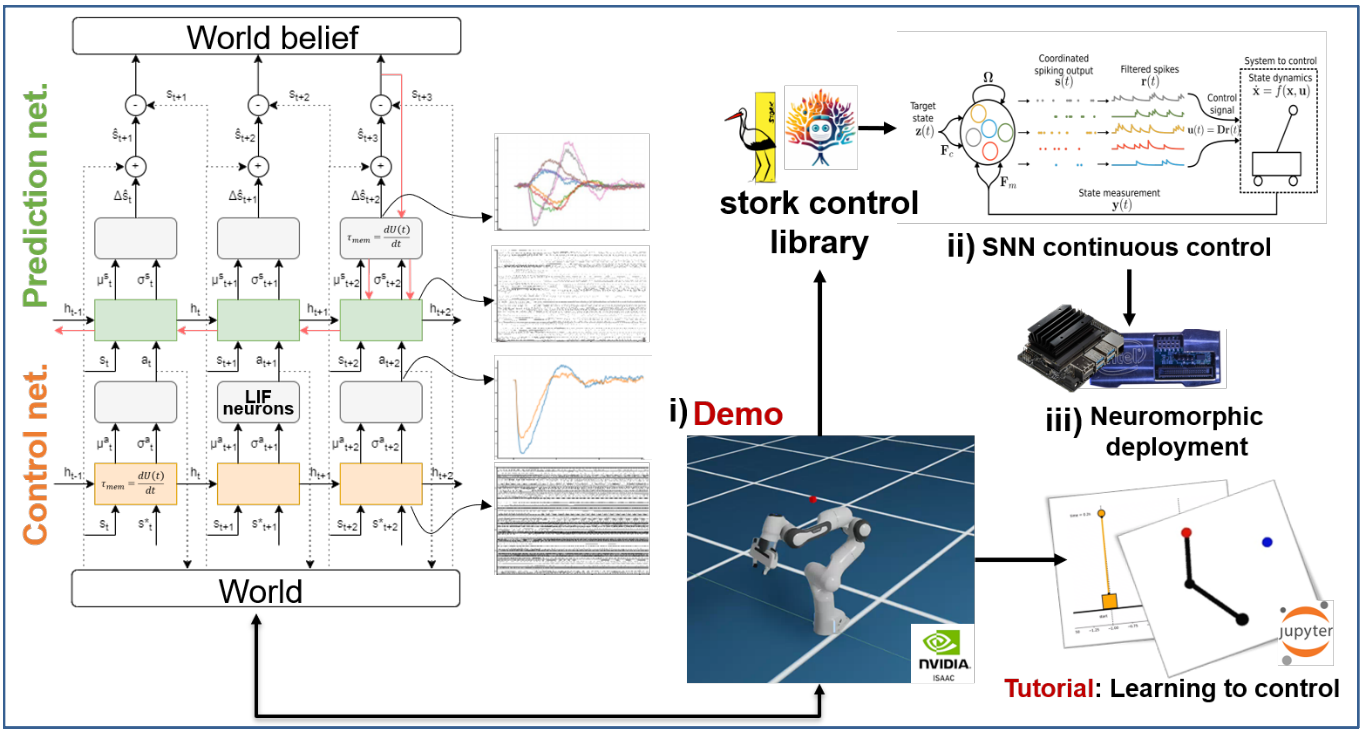

Figure 1: Demonstrator of Spiking Neural Network Control of Robots and Dynamical Systems

This Deliverable is composed of i) a main demo of a robotic arm controlled by a deep spiking neural network architecture and ii) the description of an SNN model for estimation and control, inspired by the predictive coding account of perception. Along with the models we also describe iii) its deployment in neuromorphic hardware with the collaboration of the High-Level Support Team. The main demo uses our developed deep SNN architecture to control an industrial manipulator in simulation (using the NVIDIA Isaac sim). This demo targets researchers who want to train their own SNN controllers and for education purposes, as we provide a tutorial-like Jupiter notebook. As an aside contribution we developed a control software library at the top of the stork SNNs library.

Project Number:	945539	Project Title:	HBP SGA3

Document Title:	Final demonstrator of the deep learning and SNN model for robot control in a conditional task using the industrial manipulator UR5 (SPIKEFERENCE)
Document Filename:	D3.16 (D94) SGA3 M42 SUBMITTED 230927.docx
Deliverable Number:	SGA3 D3.16 (D97)
Deliverable Type:	Demonstrator
Dissemination Level:	PU = Public
Planned Delivery Date:	SGA3 M42 / 30 SEP 2023
Actual Delivery Date:	SGA3 M42 / 27 SEP 2023
Author(s):	Pablo LANILLOS, SKU (P51)
Compiled by:	Pablo LANILLOS, SKU (P51)
Contributor(s):	Justus HÜBOTER, SKU (P51), contributed to Section 1 Serge THILL, SKU (P51), contributed to Section 1 and 2
WP QC Review:	Jeannette BOSCHMA, UM (117); Anita VAN OERS, UM (117)
WP Leader / Deputy Leader Sign Off:	Rainer GOEBEL, UM (P117)
T7.4 QC Review:	N/A
Description in GA:	Final demonstrator illustrating both the deep learning and the SNN models for solving a conditional task using an industrial robotic arm UR5e. The precise sequence of actions depends on external conditions and the robot will improvise using the abstracted prior information already learnt.
Abstract:	Spiking Neural Networks (SNN) have been shown to have comparable or better performance on perceptual tasks, e.g., classification, with the advantage of being directly deployable in neuromorphic chips. However, research in SNN control is still insufficiently investigated. In fact, robust estimation and control with spikes is still an open problem. The closed-loop generation of control signals that allow an agent (natural or artificial) to produce behaviour has particular characteristics. For instance, sensory input is dynamic and the temporal response is critical. This Deliverable shows how biologically plausible architectures using spiking deep neural networks can control robots and dynamical systems in general. We provide i) a main demonstrator of a robotic arm controlled by a neuroscience-inspired deep spiking neural network architecture (using Nvidia Isaac simulator) and ii) the description of an SNN model for estimation and control, inspired by the predictive coding account of perception. Furthermore, we also describe iii) the SNN models' ongoing deployment in neuromorphic hardware with the collaboration of HBP service category four (High-Level Support Team) to connect to physical systems. The demo and SNN model descriptions are accompanied by open-source software and libraries that can be downloaded and executed.
Keywords:	Spiking Neural Networks, Predictive coding, Perception and control, Robotics

Target Users/Readers:	Researchers, students, computational neuroscience, computer scientists, artificial intelligence, experts in spiking neural networks, experts in robotics, neuroscientific community.

Table of Contents

1. Context 4
   - 1.1 Progress vs State of the Art & Scientific Problems Addressed 4
   - 1.2 Relation to and use of EBRAINS and service categories 5
   - 1.3 How to access the demo 6
2. Neuroscience-inspired SNN robot control 6
   - 2.1 Learning to control: SNN World Models 6
   - 2.2 Predictive coding and active inference with Spikes 8
   - 2.3 Neuromorphic Control (hardware) 9
3. Looking Forward 10

Table of Figures

Figure 1: Demonstrator of Spiking Neural Network Control of Robots and Dynamical Systems 1
Figure 2: Learning to control with SNNs. 7
Figure 3: Predictive coding and active inference with spikes 9
Figure 4: Neuromorphic deployment of Spiking Neural Network Control developed in Task 3.11 10

1. Context

This Deliverable belongs to Task 3.11, a dedicated task inside WP3 funded by the SGA3 call: High-level neuro-symbolic processing for guidance of goal-directed behaviour. The aim of Task T3.11 is twofold: i) to investigate and develop brain-inspired architectures to address the challenge of high-level reasoning and behaviour (See Output 3.31 for a detailed outcome of the task in this direction); and ii) to demonstrate how biologically plausible architectures with spiking neural networks, can control dynamical systems (e.g., industrial robots). The latter is described in this Deliverable in the form of a demo accompanied with open source software, and the description of the models developed. **This Deliverable D3.16 shows how biologically plausible architectures using spiking deep neural networks can control robots**. Hence, targeting WP3 overall goal WPO3.1 (Enhanced real-world task performance through biologically plausible adaptive cognitive architectures running on neuromorphic hardware and closed-loop neuro-robotics platform) within the HBP scientific area of Cognitive Functions.

The Deliverable is composed of i) a main demo of a robotic arm controlled by a neuroscience-inspired deep spiking neural network architecture (in simulation) and ii) the description of a SNN model for estimation and control, inspired by the predictive coding account of perception. Along with the models we also describe iii) its deployment in neuromorphic hardware with the collaboration of HBP service category four (High-Level Support Team) to connect to physical systems.

1.1 Progress vs State of the Art & Scientific Problems Addressed

On the one hand, recent advances in SNNs and neuromorphic hardware have shown their strong potential to reduce energy consumption while achieving high performance in some tasks[1]. On the other hand, within the Human Brain Project, there have been important scientific breakthroughs to allow the scalation of SNNs to non-toy examples thanks to, for instance, the development of improved surrogate gradient methods[2]. However, studies on SNN control are scarce as their evaluation usually targets perceptual tasks, such as image classification. While in the mindset of the neuromorphic community control may be a solvable problem thanks to current technology, efficient and robust control using SNNs is still an open and challenging problem.

The developed models presented in this Deliverable open new avenues for SNN control in neuromorphic hardware in artificial intelligence, robotics and cognitive neuroscience, and have direct application in industrial automation and robotic wearables (e.g., exoskeletons and closed-loop prosthetics). Action generation with spikes is a very challenging endeavour in HBP, needed to understand behaviour and develop new AI solutions that can interact with the physical world.

The closed-loop generation of control signals that allow an agent (natural or artificial) to produce behaviour has its own characteristics that differ from perceptual classification. For instance, sensory input is dynamic and temporal response is critical. There are outstanding exceptions in continuous control, such as Eliasmith and colleagues' works[3], where their SNNs architectures target cognition but also control robotic arms[4]; or recent works on quadratic optimisation[2]. These methods are, whilst bioinspired, closer to an engineering solution than a functional mimicry of brain processing. Therefore, D3.16 describes the key ingredients of biologically-inspired closed-loop behaviour generation with SNNs, a major goal in Task 3.11 and a subgoal of WP3. Particularly Task 3.11

[1] Davies, M. et al. (2021). Advancing neuromorphic computing with loihi: A survey of results and outlook. Proceedings of the IEEE, 109(5), 911-934.

[2] Bellec G, Scherr F, Subramoney A, Hajek E, Salaj D, Legenstein R, Maass W (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. Nat. Commun. 11(1):3625. (P1998)

[3] Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A large-scale model of the functioning brain. science, 338(6111), 1202-1205.

[4] DeWolf, T., Patel, K., Jaworski, P., Leontie, R., Hays, J., & Eliasmith, C. (2023). Neuromorphic control of a simulated 7-DOF arm using Loihi. Neuromorphic Computing and Engineering, 3(1), 014007.

investigated i) how to learn to control robots with SNNs, ii) how to develop efficient and fast neuroscience-inspired SNNs architectures for estimation and control of dynamical systems (e.g., a spiking active inference model) and iii) the deployment of these approaches into neuromorphic hardware.

1) **Learning to control with SNNs**. We developed a new SNN architecture that can learn complex continuous control policies from self-generated data of the robot (See Demo of this Deliverable). While current solutions are restricted to position control, discrete action generation or in the case of NENGO there is a need to approximate engineered dynamics mappings with large SNNs, our solution can learn from scratch the dynamics (world model[5]) and, at the same time, the generation of the control signals in a continuous action domain. This architecture has a strong brain inspiration, as *i)* it is based on predictive coding account to perception and *ii)* the neurons of the SNNs possess LIF internal dynamics and their recurrency—conversely to other machine learning SNNs solutions that transform SNN neurons into standard artificial neurons with binary output and payload messages.
2) **Predictive coding and active inference with Spikes.** Although predictive coding[6] and active inference[7] have been presented as a biologically plausible functional description of brain processing (for perception and action), there is still a big gap between its mathematical formulation and the biological neural substrate. In particular, a spiking neural network formulation is missing. We investigated in Task 3.11 algorithms that integrate the characteristics of these theoretical accounts through spiking dynamics. This is essential to achieve the following objectives: *i)* creating low-power highly-efficient inference chips for perception and control, *ii)* aiding in parsing and decoding the actual signals generated within the brain's circuitry[8] and *iii)* deploying cybernetic/biological computers[9]. This Deliverable describes a model that is able to robustly estimate and control known dynamical systems (e.g., robots) with biologically plausible spiking patterns and that can mimic active inference dynamics. These novel methods named **Spike Coding Networks Control** and **Spike Active Inference** do not need learning or optimization thus, offering important opportunities for deploying fast and efficient task-specific on-chip spiking controllers with biologically realistic activity.
3) **Neuroscience-inspired neuromorphic control.** One of the main advantages of our proposed SSN architectures is that they can be directly deployed into neuromorphic chips. However, there are always hardware restrictions and specifications that need translation from the theoretical model into chip-specific computations. While this is more a technical problem than a scientific one, this Deliverable also describes the collaboration of SKU (Task 3.11) with the service category 4 (T4.13) High-Level Support Team (Thomas Nowotny, Sussex University and Jörg Conradt, KTH) to deploy the developed models into hardware.

1.2 Relation to and use of EBRAINS and service categories

We are collaborating with the service category 4 High-Level Support Team through a Voucher programme. In particular, they supported Task 3.11 to deploy the developed SNN models—described in this Deliverable—into neuromorphic hardware and allow the connection to a physical robot. First, the team of Thomas Nowotny (Partner UoS) has ported our closed-form Spike Coding Networks

[5] Taniguchi, T. et al. (2023). World models and predictive coding for cognitive and developmental robotics: frontiers and challenges. Advanced Robotics, 1-27. (P4156)
[6] Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature neuroscience, 2(1), 79-87.
[7] Parr, T., Pezzulo, G., Friston, K. J. (2022). Active inference: the free energy principle in mind, brain, and behavior. MIT Press. (P2988)
[8] Denève, S., Machens, C.K. (2016). Efficient codes and balanced networks. Nature neuroscience, 19, 375-382.
[9] Kagan, B.J. et al. (2022). In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. Neuron 110(23), 3952–3969. (P4008)

Control to a GPU-based accelerated hardware—using their developed framework GeNN[10]. The model has been deployed into a Nvidia Jetson Nano to have a fast interface connection between the physical robot and the SNN computation. This reduces the challenges of SNN simulated solutions, which normally cannot provide perception-action fast loops needed to control a robot. Second, Jörg Conradt's group (Partner KTH) is working on the deployment of the SNN models into the Spinnaker board using their developed Peripheral Interface (SPIF) to obtain a high-frequency sensor/motor robot connection.

1.3 How to access the demo

The demonstrator shows the deep SNN architecture (described in Sec. 2.1) controlling a robotic arm with spikes in the Nvidia Isaac Simulator (See Figure 1). The demo has two levels of use: i) open-source software for researchers investigating regression or control problems with SNNs and ii) a tutorial that can be followed step by step through a Jupiter notebook, or executed directly in Python. Hence, this demo is thought to aid researchers in training their spiking controller for any robot and also for teaching purposes—to train students in spike learning and control. It can be accessed in the following GitHub repository: https://github.com/jhuebotter/spiking_control

Requirements, instructions and a link to a video of the demo can be found in the README.md file of the repository. The backbone software of the demo is a library developed for SNNs learning for control (on the top of the stork library): https://github.com/jhuebotter/control_stork

2. Neuroscience-inspired SNN robot control

2.1 Learning to control: SNN World Models

SKU, in this sub-task, developed a new deep SNN computational model that is suitable for learning to control robots. In particular, our primary goal was to construct an SNN for the low-level continuous control of industrial robotic arms—See the main demo of this Deliverable. Instead of creating an anatomically detailed model of how human brains perform control (e.g., cerebellum model), we sought to harness basic principles of efficient information processing. Our proposed architecture takes inspiration from the predictive brain approach to control—See[11] for preliminary work on motor control with the proposed architecture)—and state-of-the-art model-based deep learning approaches for continuous control[12]. The core ability of SNNs to process temporal dynamics makes them an ideal candidate for robotic arm control. The provided repository in this Deliverable also includes tutorial notebooks, offering an introduction to SNNs control in the machine learning context for research and education purposes. Besides, we foresee that this contribution will lay the foundation for future implementations of SNNs for control in neuromorphic hardware.

We evaluated the developed model in two exemplar simulated robotic manipulanda (Figure 2): a 2DoF planar arm (for tutorial/teaching purposes) and a 7-DoF robot arm by Franka Emika using the Isaac Sim framework[13]. In the latter, we examined several DoF cases and control modes (velocity, acceleration, and torque) under more realistic conditions, including joint constraints and self-collision. In both robots, the primary learned task was reach & follow—in a full observability setting—where the robot end-effector was programmed to reach a target pose and track moving target positions.

[10] https://genn-team.github.io/

[11] Huebotter, J., Thill, S., Gerven, M. V., & Lanillos, P. (2022, September). Learning policies for continuous control via transition models. In International Workshop on Active Inference (pp. 162-178). Cham: Springer Nature Switzerland. (P3997)

[12] Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603.

[13] https://developer.nvidia.com/isaac-sim

Our model architecture is composed of two connected pathways. The prediction/world model allows the agent to learn the world dynamics (predicting observations and state transitions) and a policy/inverse model for control signal generation[14]. This modular architecture is adaptable and can be extended for higher-level cognitive functions, such as visual perception, value assessment, and high-level policy planning and learning (e.g., including rewards-based learning). Supervised learning in deep SNNs via surrogate gradients has been shown to be a powerful tool for feed-forward networks on the classification using temporal data. Here, we adapted this method in a self-supervised setting in the domain of continuous control problems. For this purpose, we extended the stork framework[15,16] for surrogate learning in SNNs in GPUs via PyTorch for control problems. The software library has been released as open source[17].

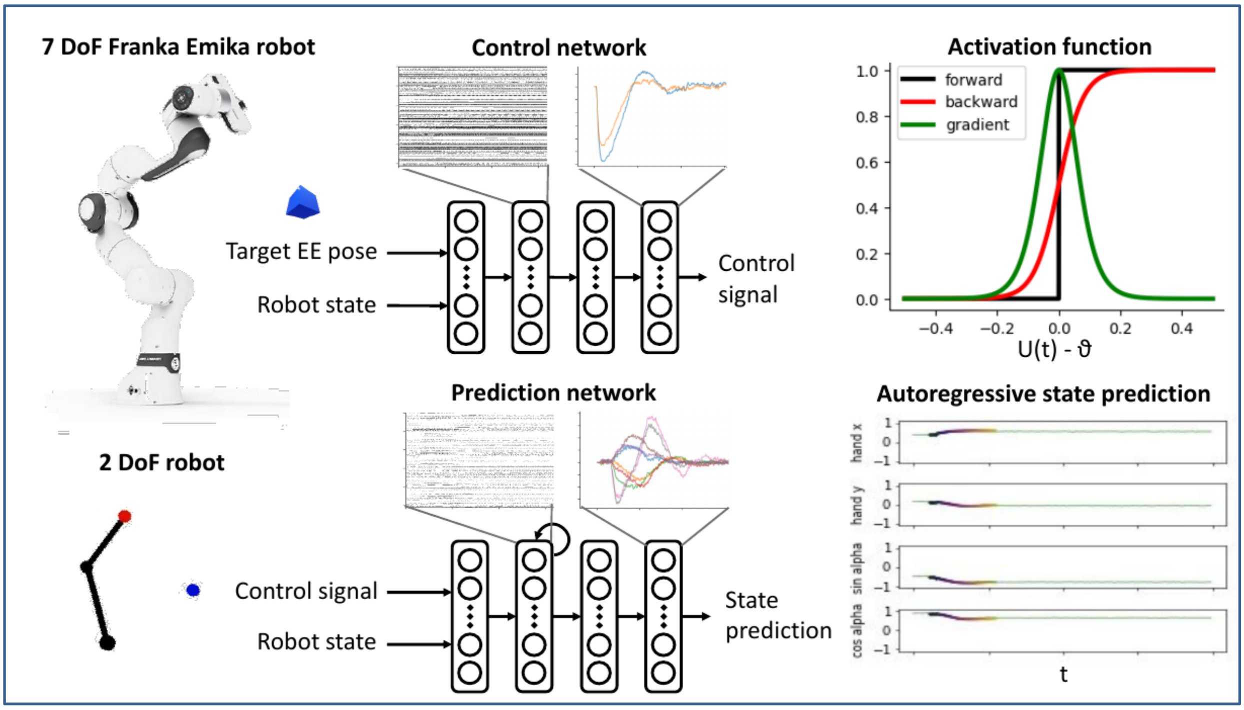

Figure 2: Learning to control with SNNs.

Left: Robots used for simulated reach & follow tasks. Centre: Schematic and example activity of the spiking neural networks for control signal generation and robot state prediction. Right top: The sigmoid surrogate gradient function approximates the step activation function during the backward pass for leaky integrate-and-fire neurons. The shape of the gradient has been normalized and tuned to tackle vanishing gradients during sequence learning. Right bottom: Example of unrolled autoregressive state prediction under a given control sequence used for updating the prediction network (predicted state vs. observed state).

Training SSNs for control

The (recurrent) prediction model was trained to comprehend the robot arm dynamics from experience. The robot primarily rolls out imagined robot state trajectories based on control signals from the policy/inverse model. This enables us to determine the gradient of the end-effector pose error concerning the control signal and its network parameters. We then employed Backpropagation Through Time (BPTT) to optimise both networks simultaneously using an end-to-end learning approach. A schematic of this process is illustrated in the figure below.

Our model implementation was able to reduce the well-known problem in SNNs of vanishing gradients during BPTT, a challenging problem in non-spiking RNNs before the advent of LSTMs. We compared the performance of our SNN model against baselines from optimal control (linear quadratic regulator

[14] Iacob, S., Kwisthout, J., & Thill, S. (2020, July). From models of cognition to robot control and back using spiking neural networks. In *Conference on Biomimetic and Biohybrid Systems* (pp. 176-191). Cham: Springer International Publishing.
[15] Rossbroich, J., Gygax, J., and Zenke, F. (2022). Fluctuation-driven initialization for spiking neural network training. Neuromorph. Comput. Eng. https://doi.org/10.1088/2634-4386/ac97bb
[16] https://github.com/fmi-basel/stork
[17] https://github.com/jhuebotter/control_stork

for 2D arm and Riemannian Motion Policies for 3D) and non-spiking probabilistic neural network models (that incorporate LSTMs). Our findings indicate that the SNNs' performance was competitive with both baseline methods for the reach & follow task. All code developed for this project was written in Python 3 and is openly available in this repository:

https://github.com/jhuebotter/spiking_control

2.2 Predictive coding and active inference with Spikes

SKU, within Task 3.11, developed a novel computational model for control using spike-coding networks (SCNs). This is a special type of SNN that can track any dynamic system and implement its state estimator and controller without the need for training when the system is known. The details of this model are described in the accepted publication[18] in the journal IEEE Trans. Cog. Dev. Sys, Special Issue on Advancing Machine Intelligence with Neuromorphic Computing. The model has been presented at the HBP 2023 summit and the Neuromorphic Computing in the Netherlands Conference (NCN2022). The code to execute this model can be found here: https://github.com/FSSlijkhuis/SCN_estimation_and_control

The basic idea behind this coordinated SNN is that neurons describe a population code that minimises prediction error (i.e., neurons spike when prediction error rises)[19]. To encode the dynamics of a system, the connectivity of the network is analytically computed (exploiting the leaky-integrate and fire neuron mathematical model). Thus, we can encode any robot (with its state-space equations) into the SCN and obtain a spiking estimator and controller. Figure 3(A-D) depicts the SCN high-level description along with its evaluation with numerical examples (Spring-Mass-Damper and Cartpole systems) of optimal estimation and control and its robustness/efficiency analysis in terms of neuron silencing and the tuning of the spike patterns sparsity.

Transforming SCNs into Spike Active Inference Control

Active inference is grounded on the free energy principle[20], a very influential theory on how the brain may process information and generate behaviour that may solve some of the challenges in robotics[21]. While multiple computational models account for designing artificial agents and robots using this framework[22], there is no spiking model in the literature that implements active inference. This is very relevant as the biological plausibility of the theory needs to be proven in spiking dynamics. Hence, we investigated how spiking neural dynamics can control systems in the same way active inference works. We designed and implemented an SCN that follows the continuous-time active inference differential equations, thus obtaining an SNN that outputs control commands that minimise the free energy functional. Hence, obtaining the first implementation of Active Inference with spikes. Figure 3F shows the comparison between continuous-time active inference (non-spike) and the proposed spike active inference control for the spring-mass-damper system. State dynamics, generated actions and the minimisation of the free energy (F) shows the equivalence between the two models.

[18] Slijkhuis, F. S., Keemink, S. W., & Lanillos, P. (2022). Closed-form control with spike coding networks. arXiv preprint arXiv:2212.12887. accepted at IEEE Tans. Cog. Dev. Sys. (P4175)
[19] Denève, S., Machens, C.K. (2016). Efficient codes and balanced networks. Nature neuroscience 19(3), 375
[20] Friston, K. (2010). The free-energy principle: a unified brain theory? Nature reviews neuroscience, 11(2).
[21] Da Costa, L., Lanillos, P., Sajid, N., Friston, K., & Khan, S. (2022). How active inference could help revolutionise robotics. Entropy, 24(3), 361. (P3294)
[22] Lanillos, P., Meo, C., Pezzato, C., Meera, A. A., Baioumy, M., Ohata, W., ... & Tani, J. (2021). Active inference in robotics and artificial agents: Survey and challenges. arXiv preprint arXiv:2112.01871.

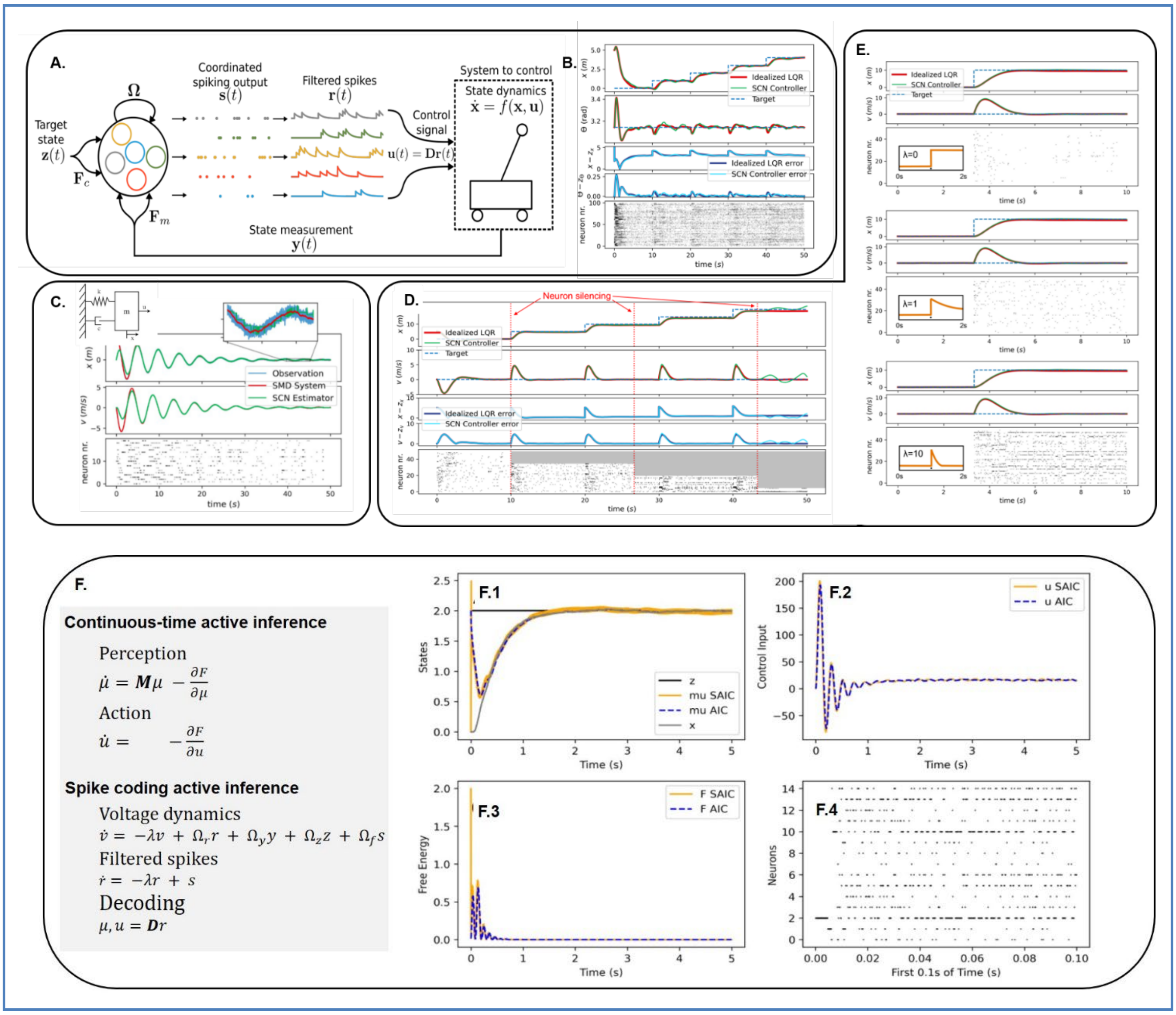

Figure 3: Predictive coding and active inference with spikes

Neuroscience-inspired robots estimation and control of known dynamical systems (e.g., robots). (A) Schematic of the Spike Coding Network (SCN), whose neurons coordinate the firing to output the correct control signal, while estimating the system state. (B) Cartpole SCN control. The top panel describes the desired state (position of the cart pole, dashed blue line) and the position of the system when controlling it with SCNs in comparison with classical optimal control Linear-Quadratic Gaussian. The second and third panels show the cart pole angle velocity estimation and the estimation error. The bottom panel shows the spike raster of the SCN while controlling the system. (C) Robust SCN estimator (Spiking version of a Kalman filter) under the presence of sensory and plant noise evaluated in the Spring-Mass-Damper system. (D) SCN has interesting characteristics. The left panel shows its robustness to neuron silencing thus, describing how the coordinated spikes can recover the control when more than 50% of the spikes are not working anymore. The right panel describes how the sparsity of the SCN can be tuned by with a trade of performance. (F) Comparison between Continuous-time Active Inference Control (AIC) and Spike Active Inference Control (SAIC). F.1) Controller acting on a Spring-Mass-Damper system; F.2) Control signal generated by the network; F.3) the free energy of the system being minimised over time by the network; F.4) the spike raster for the network activity.

2.3 Neuromorphic Control (hardware)

At SKU, in Task 3.11. we investigated the deployment of the SNN control models in hardware. Figure 4 shows a detailed description of the different neuromorphic devices and models. At SKU we developed our own software for the Intel Loihi 2 neuromorphic chip (using Lava interface) to deploy Spike Coding Networks and Spike Active Inference models. Figure 4b shows the SCN model running on the Loihi simulator. Besides, we have evaluated multiple Spiking Neural Network Communication and Synchronization Methods on Event-driven Neuromorphic Systems[23]. Furthermore, through the SGA3 voucher program Task 3.11 has been supported by the High-Level Support Team (HLST) service category to port our approach into neuromorphic hardware. UoS partner ported our SCN models into GeNN (GPU SNNs implementation) to develop a superfast SNN deployment able to control a robotic arm in high frequency. The interface with the robot is currently being developed with an NVIDIA Jetson nano. KTH partner is porting our SNN model into Spinnaker 1.0 making use of their self-developed input-output hardware that allows fast speed communication between the neuromorphic board and the robot.

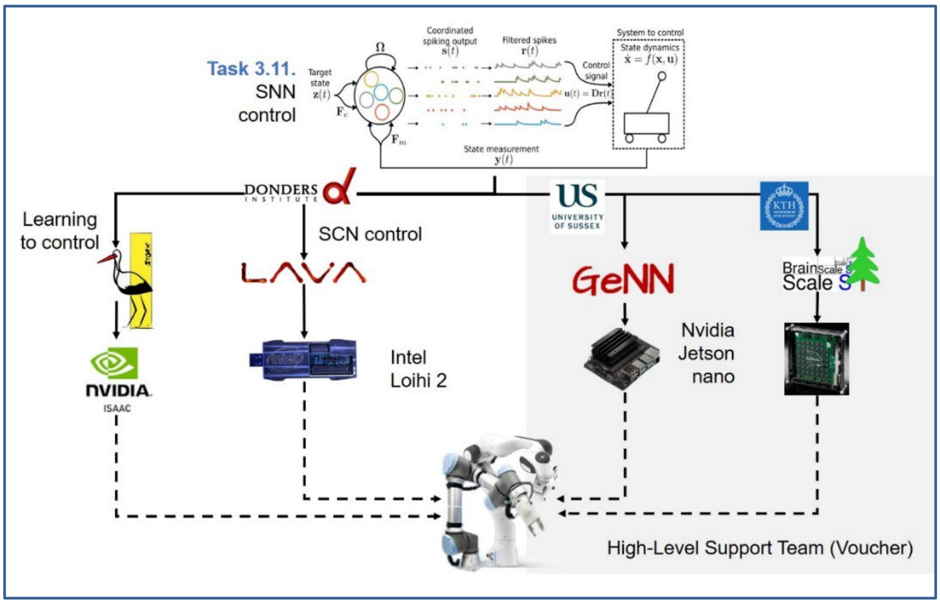

Figure 4: Neuromorphic deployment of Spiking Neural Network Control developed in Task 3.11

To deploy our models into hardware we followed different routes. SNN learning architectures have been developed for Isaac Simulator (Nvidia), and SCN models have been deployed in Intel Loihi 2 chip and ported to GeNN to allow GPU-accelerated hardware (Nvidia Jetson Nano) execution. Finally, HLST is also supporting its deployment into Spinnaker 1.0 boards.

3. Looking Forward

To **summarise**, the work documented in this Deliverable has advanced the state of the art by i) a deep spiking neural network architecture that can learn to control robots end-to-end in a self-supervised fashion (a demonstrator is provided), ii) a closed-form SNN model that can robustly estimate and control known dynamical systems without the need of learning (the code and the publications are described), and iii) the ongoing work in the deployment of SNN control into neuromorphic hardware.

While the hardware implementation of the SNN models was out of the scope of Task 3.11, it is planned that within one year we will achieve a full deployment of the proposed SNN models into hardware and control two different physical robotic platforms. This is an ongoing collaboration between SKU, UoS and KTH, which will be extended after the end of the HBP. Furthermore, Task 3.11, with Deliverable 3.16 and Output 3.31 provides the elements needed to design artificial systems that combine high-level reasoning and closed-loop SNN control. Hence, our mid-term goal besides neuromorphic implementation is to transfer these general neuroscience-inspired estimation and control solutions to applications ranging from general-purpose robotics[24] to space robotics or wearables.

[23] Shahsavari, M et al. Advancements in Spiking Neural Network Communication and Synchronization Techniques for Event-Driven Neuromorphic Systems. http://dx.doi.org/10.2139/ssrn.4481982 (P4176)
[24] Taniguchi, T., Murata, S., Suzuki, M., Ognibene, D., Lanillos, P., Ugur, E., ... & Pezzulo, G. (2023). World models and predictive coding for cognitive and developmental robotics: frontiers and challenges. Advanced Robotics, 1-27. (P4156)