Soft-Grasping With an Anthropomorphic Robotic Hand Using Spiking Neurons

J. Camilo Vasquez Tieck, Katharina Secker, Jacques Kaiser, Arne Roennau, and Rüdiger Dillmann

***Abstract*—Evolution gave humans advanced grasping capabilities combining an adaptive hand with efficient control. Grasping motions can quickly be adapted if the object moves or deforms. Soft-grasping with an anthropomorphic hand is a great capability for robots interacting with objects shaped for humans. Nevertheless, most robotic applications use vacuum, 2-finger or custom made grippers. We present a biologically inspired spiking neural network (SNN) for soft-grasping to control a robotic hand. Two control loops are combined, one from motor primitives and one from a compliant controller activated by a reflex. The finger primitives represent synergies between joints and hand primitives represent different affordances. Contact is detected with a mechanism based on inter-neuron circuits in the spinal cord to trigger reflexes. A Schunk SVH 5-finger hand was used to grasp objects with different shapes, stiffness and sizes. The SNN adapted the grasping motions without knowing the exact properties of the objects. The compliant controller with online learning proved to be sensitive, allowing even the grasping of balloons. In contrast to deep learning approaches, our SNN requires one example of each grasping motion to train the primitives. Computation of the inverse kinematics or complex contact point planning is not required. This approach simplifies the control and can be used on different robots providing similar adaptive features as a human hand. A physical imitation of a biological system implemented completely with SNN and a robotic hand can provide new insights into grasping mechanisms.**

***Index Terms*—Learning systems, robot learning, neural networks, biological neural networks, learning (artificial intelligence), force feedback, grasping, robot programming, robot control, robot motion, cognitive robotics, humanoid robots, manipulators, robot learning, robot sensing systems, service robots.**

I. Introduction

WITH evolution humans developed advanced and flexible grasping capabilities thanks to a combination of an adaptive hand and efficient control. Humans do not plan a gasping motion and execute it, there is actually a combination of control loops working together to grasp an object. Based on sensor feedback, the hand can adapt its motion if the object moves or deforms. This is called soft-grasping [1], [2]. Soft-grasping with a anthropomorphic hand (as in Fig. 1) is an important capability for robots interacting in an environment with objects shaped for humans. Nevertheless, most robotic applications use vacuum, 2-finger or custom grippers [3] which are fine for production applications, but lack adaptability grasping objects without knowing their exact properties.

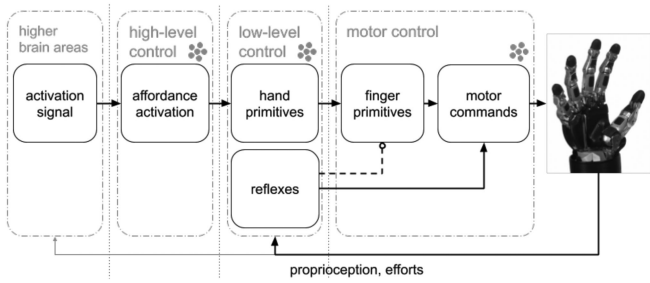

Fig. 1. General view of the closed-loop architecture with SNN for soft-grasping using motor primitives and reflexes.

There are studies on human motor control providing insights about the grasping mechanisms. In this work focuses on the biological principles for motion representation using motor primitives, adaptive and compliant control, and event-based computation [4] with spiking neural networks (SNN) [5]. SNN model closer the characteristics of real neurons, which enables research on learning mechanisms and information representation in the brain. An accepted hypothesis is that the central nervous system uses a small number of muscle synergies that are combined to produce motions [6], [7]. The activation of the synergies can change based on sensor feedback. A motion can be decomposed with neurons sensitive to different parts of it [8]. These insights have been successfully applied in robotics, for example with the dynamic movement primitives [9] and the eigengrasps [10]. Studies show evidence of muscle synergies for grasping [11], the relation between human responses and the stiffness regulation in the hand [12], the classification of grasping motions [13], and the generalization of muscle patterns as building blocks for grasping [14]. An anthropomorphic robotic hand enables further investigation of the neural response of grasping motions [15], or evaluation of the different affordances [16], or the use of synergies from human demonstration for grasping

Manuscript received June 5, 2020; accepted October 2, 2020. Date of publication October 26, 2020; date of current version March 18, 2021. This letter was recommended for publication by Associate Editor S. Farokh Atashzar and Editor Allison M. Okamura upon evaluation of the Reviewers' comments. This research was supported by the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement 785907 (Human Brain Project SGA2) and 945539 (Human Brain Project SGA3). *(J. Camilo Vasquez Tieck and Katharina Secker contributed equally to this work.) (Corresponding author: J. Camilo Vasquez Tieck.)*

J. Camilo Vasquez Tieck, Katharina Secker, Jacques Kaiser, and Arne Roennau are with the FZI Research Center for Information Technology, 76131 Karlsruhe, Germany (e-mail: tieck@fzi.de; secker@fzi.de; jkaiser@fzi.de; roennau@fzi.de).

Rüdiger Dillmann is with the FZI Research Center for Information Technology, 76131 Karlsruhe, Germany, and also with the Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany (e-mail: ruediger.dillmann@kit.edu).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.3034067

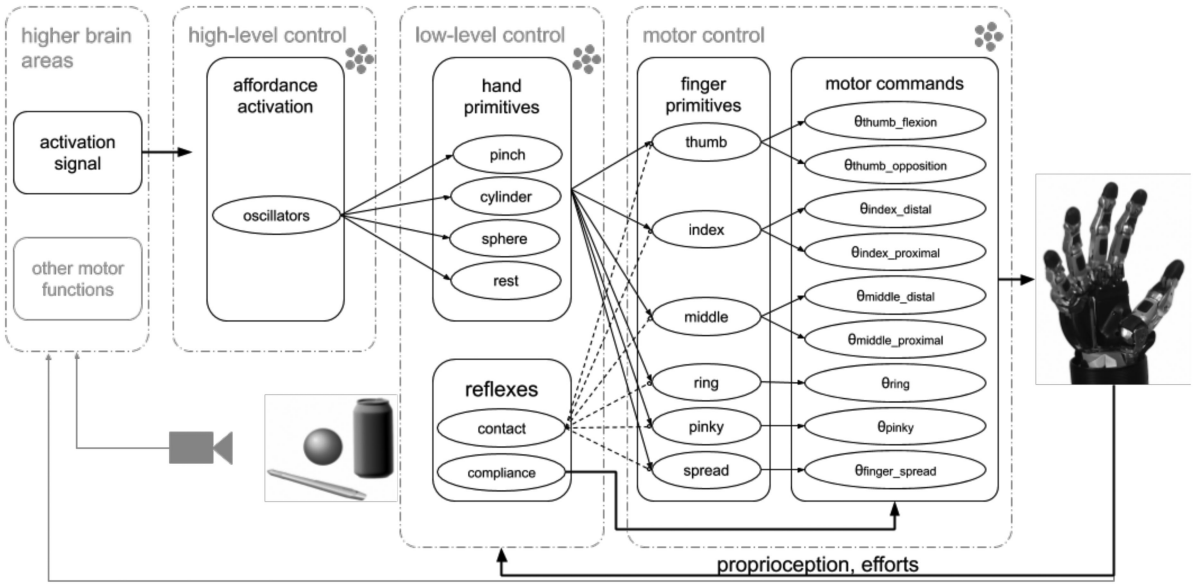

Fig. 2. Detailed view of the closed-loop architecture for soft-grasping with SNN. In *motor commands*, a population of neurons generate joint positions for each joint. The *finger primitives* represent joint synergies. The *hand primitives* represent different grasping motions. In *reflexes*, a mechanism for contact detection triggers reflexes to inhibit the motion of the fingers and activate the compliant controller. The *affordance activation* provides continuous activation signals for the hand primitives.

control [17]. Other approaches explore human-like grasping using deep learning (DL) to create representations using autoencoders [18], for extensive training in simulation for in-hand manipulation [19], or with a combination of an object classifier with reactive and anticipatory motor primitives [20]. But despite recent successes, DL also has some drawbacks. To train a system based on DL, a lot of data and simulation time is required. In [21], 800,000 samples were collected to train a robotic arm to perform reaching and grasping. Compliant control with a robotic hand can be performed with soft [22] or flexible hardware [2] or with software using sensor feedback [1]. Different approaches use software compliant control using a torque sensor with a robust modelling [23], or using online learning for adaptive control [24], or using cerebellar principles [25], or combining reflexes with predictive control [26]. Although, there are other approaches using SNN for motion control using force feedback, to the best of our knowledge, there is no implementation of an SNN for soft grasping with a 5-finger anthropomorphic hand performing compliant control without force sensors using the standart joint interface.

We can model a system using SNN for soft-grasping using an anthropomorphic robotic hand taking inspiration from biology and using the principles presented in previous work for a hierarchy of motor primitives with SNN to model the hand [27], to model finger reflexes [28], to coordinate multiple primitives [29], and to combine activation modalities [30].

The main components are presented in Fig. 1, from right to left the layers have an increasing level of abstraction. In *motor control* there are the finger primitives, in *low-level control* the hand primitives and the reflexes, in *high-level control* the affordance activation mechanisms, and in *higher brain areas* the activation signals. Two control loops are combined, one is based on the motor primitives and the other one is a compliant controller activated by a reflex. The motions are modelled with motor primitives in a hierarchy with finger primitives representing synergies between joints and with hand primitives representing different affordances coordinating the fingers. The reflex is triggered by a contact detection mechanism modelled as the circuits of inter-neurons in the spinal cord. This modelling simplifies the control of the hand and generalizes each grasp for different objects. Objects with different shapes, stiffness and sizes are graspable without knowing their exact properties. The hand fits the grasping motions to the different objects using the compliant controller. It is not necessary to compute the inverse kinematics or to calculate complex contact point planning.

II. Approach

In this work, we present an approach for soft-grasping with objects of different shapes, stiffness and sizes using a SNN to control an anthropomorphic robotic hand. Motions are represented with a hierarchy of motor primitives that allow a reduction of the control parameters. The network combines two control loops to generate a complex grasping behaviour. The first one is reactive, generated by the trajectory control from the primitives, to close the hand using human affordances [13]. The second one is adaptive, generated by the compliant control triggered with a reflex using the motor currents.

The detailed architecture for the SNN is presented in Fig. 2. There are four main components of the network: finger primitives, hand primitives, affordance activation and reflexes. Each oval is a sub-network. The connection between compliance and the joints is simplified, it is also all to all. The finger primitives represent the joint synergies in a finger for a closing motion. The hand primitives represent different affordances coordinating the fingers. The affordance activation mechanism creates the

activation patterns for the hand primitives. There are two types of reflexes activated by contact, one inhibits the movement of the fingers and the other activates the compliant controller. Contact detection is modelled as the circuits of inter-neurons in the spinal cord. The compliant controller uses the efforts from the motors to control the force the finger can apply.

The SNN is developed using the Neural Engineering Framework (NEF) [31] which allows the generation of large-scale SNN.First, the model is divided into vectors, functions, and differential equations. The connections between sub-networks compute the functions. The connection weights for each sub-network are optimized separately, and then combined together into one large neural network. By changing the connection weights, we change the function being computed. Finding connection weights locally means we can generate large systems without using the traditional neural network approach of optimizing over huge amounts of training data. The trade-off is that we need expert knowledge to define what each sub-part of the model is doing. This principles are used to implement a SNN for soft-grasping.

A. Finger Primitives and Robot Kinematics

For each finger there is a motor primitive that represents the joint synergies between the joints of the finger during a closing motion. The principles to model the finger primitives are based on [28], [30]. In addition to the five fingers, we extended the modelling with two more degrees of freedom – thumb opposition and finger spread. There are seven finger primitives – thumb, thumb opposition, index, middle, ring, pinky and finger spread – as illustrated in Fig. 2.

A finger primitive is defined with the min initial and max final values of each active joint for one trajectory. It is modelled as a mapping of an activation parameter $u \in [0, 1]$ to a sequence of joint activations during the execution of the motion. The activation function $f : [0, 1] \rightarrow [0, 1]$ is defined as

$$f(u) = \frac{\sin(u \cdot \pi - \frac{\pi}{2})}{2} + \frac{1}{2}. \tag{1}$$

It is important for $f(u)$ to have smooth initial and final phases to prevent wear in the motors and transmissions of the real robot. This type of functions are commonly used in robotics for interpolation. Then the activation function has to be mapped to the robot kinematics. A mapping $g : [0, 1] \rightarrow \mathbb{R}^n$ is defined as

$$g(f(u)) = f(u) \cdot (\theta_{\max} - \theta_{\min}) + \theta_{\min}, \tag{2}$$

to generate appropriate motor commands. Which means, scaling $f(u)$ to the motion interval $(\theta_{max} - \theta_{min})$ with offset θ_{min} of the joint θ.

B. Hand Primitives and Hierarchy

The motion of the hand is also modelled with a motor primitive, but instead of controlling joints, it controls the activation parameters of the finger primitives. The hand primitives are organized in a hierarchy coordinating the finger primitives as in [27], [29]. The hand primitives represent different grasping affordances — sphere, cylinder and pinch according to [13] and rest position. This reflects the muscle synergies of human grasping [14]. The network does not learn individual objects, it learns different affordances for different object types. By using motor primitives for grasping motions, the complexity of the hand control is reduced to one activation parameter for each affordance.

A hand primitive is modelled as a mapping of u to a sequence of activations of the finger primitives during the execution of the motion. The initial grasping pose (pre-shaping) and the final pose with the hand closed, define the primitive with the min open and max closed activation parameters for the finger primitives. Each hand primitive is connected to all finger primitives. There are four hand primitives as illustrated in Fig. 2. To learn a new affordance, a sub-network in *hand primitives* is required, and the initial and final positions of the joints to define the primitive (see table in Fig. 8). The rest of the SNN can be reused and the compliant controller adapts the motions online to the shape of the objects.

C. Affordance Activation Mechanisms

The affordance activation generate the activation patterns for the hand primitives. An external activation signal is used to activate the hand primitives. A population of neurons generates neural activity for the duration of the grasping motion. This is an oscillator that oscillates only once ([28], [30]), and the activity for the activation parameter u is decoded as

$$u = -\frac{1}{2}\cos\left(t \cdot \frac{2\pi}{T}\right) + \frac{1}{2}. \tag{3}$$

D. Reflexes and Contact Detection

The reflexes and the circuit for contact detection are the parts that provide the adaptation and flexibility to the grasping motions, required for soft-grasping. The modelling is taken from [30], with additions to activate the compliant controller and to change the activation parameters from the SNN. The flexion is not being measured, only the currents of the motors are used to detect contact. The contact detection circuit is modelled as an alternative selection mechanism as the networks with inter-neurons in the spinal cord [32]. It combines inhibitory and excitatory connections, as illustrated in Fig. 3(a). The proprioception is used to calculate $\Delta\Theta$ as the change in time of the joint, using the current joint position Θ_t and the previous joint position Θ_{t-1} provided by a delayed recurrent connection. The interneuron, is excited by the effort feedback from the motor and inhibited by $\Delta\Theta$. This ensures that the interneuron only detects contact if the effort increases and the corresponding joint is not moving, and thus ignores changes in the effort caused by the non linearity of the robot dynamics. There are two types of reflexes that are triggered with contact. The first type provides inhibition and stops the motion. When contact is detected in one finger, the reflex inhibits the respective primitive and the contact joint position is mapped as the new target position. The second type of reflex mechanism activates a compliant controller for that finger.

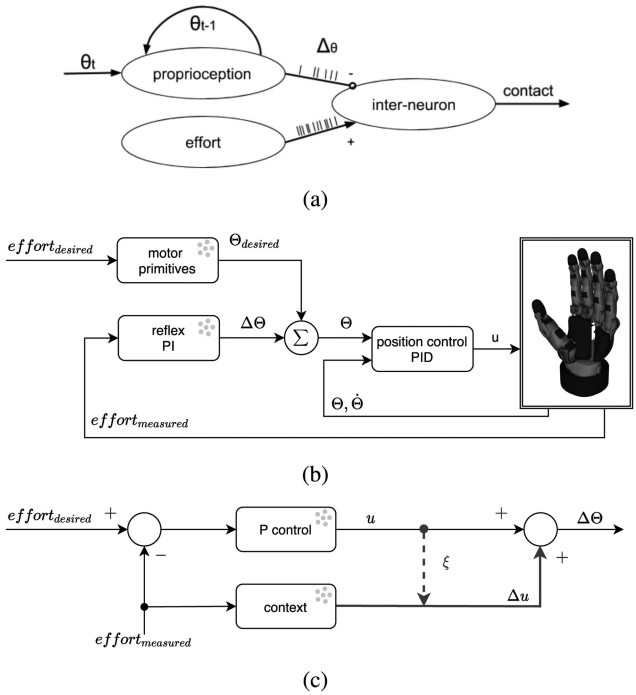

Fig. 3. (a) Contact detection circuit (adapted from [30]). (b) Cascaded compliant controller for each finger. (c) Adaptive part with online learning, reflex PI in (b).

E. Compliant Controller and Adaptation

The control schema is a cascaded impedance setup of two controllers, the motor primitives and the compliant controller (see Fig. 3(b)). The two controllers are combined based on the feedback from the motors. With the measurement of the motor current, an effort of each joint can be estimated. The compliant controller uses the effort feedback as control parameter to control the force a finger can apply. It is activated with contact, and inhibited if no contact is detected or if the hand is opening. Target efforts can be set to each joint individually to determine the force and sensibility of the grasp. The target values for the effort controller can be changed on-the-fly by the network for each finger, which provides flexibility.

The controller was initially modelled a classical PI controller implemented with SNN. The I part was divided in two parts with different parameters to reduce oscillation. A fast reacting I part with an offset to compensate the delayed system answer. This prevents the system for overshooting while the system recovers from a delayed response. A slow reacting I part is adjusted by the system delay and produces less oscillation. It is worth mentioning, that no D part was used because the current measurements from the motors are very noisy. A digital I part can be described as

$$I_{(k)} = k_I \cdot \sum e_{\mathrm{diff},(k\cdot\Delta t)} \cdot \Delta t \tag{4}$$

where k_I is the factor of the I part and e_{diff} is the control error between the target value and the actual measured value. This can be converted into the difference equation with the additional *P* component for the controller output y_k as

$$y_k = y_{k-1} + k_I \cdot u_k \cdot \Delta t. \tag{5}$$

Due to the interaction between the fingers and the object, the initial contact points have to change if the object moves or deforms. Ideally, there is a part in the controller that can learn online to compensate for these changes without calculating the exact contact points or the inverse kinematics. For this, we propose an adaptive control schema (see Fig. 3(c)). Therefore, the adaptive part works as an additional I part of the controller with dynamic parameters. For the online adaptive part (green connections), the prescribed error sensitivity (PES) rule is used as implemented in NEF [33]. The number of neurons and the learning rate determine the factor of the adaptive control, and define how fast the adjustments are made. PES adjusts the decoders Δd_i of a connection to minimize an error signal. The change in weights w_{ij}, is given by

$$\Delta w_{ij} = \Delta d_i \cdot e_j \alpha_j \tag{6}$$

$$\Delta d_i = -\frac{\kappa}{n} \cdot \xi a_i, \tag{7}$$

where α_j is the gain, κ the global learning rate, n the number of neurons, ξ the error signal, e_j the encoder of the postsynaptic neuron and a_i the presynaptic activity. The pre-synaptic population is indexed by *i* and *j* indexes the post-synaptic population. The resulting connection Δu is added to control signal and it is defined as

$$\Delta u(t) = \sum_{i=0}^{n} d_i \cdot a_i(x(t)), \tag{8}$$

where $a_i(x(t))$ is the activity of neuron i given the input $x(t)$. The error signal ξ is given by the PI part and the correction by Δu is affected by the learning rule adapting d_i.

III. Results

An anthropomorphic Schunk SVH 5-finger hand was used to evaluate the SNN for soft-grasping. For the experiments, the hand was mounted in a test base as well as in a robotic arm. We modelled three types of grasping motions — sphere, pinch and cylinder. The affordances were activated with an external signal to trigger the motion. The activation of the different motor primitives was evaluated to test how the affordances adapted to different objects. Then, the sensibility of the compliant controller and its activation was evaluated. Finally, we evaluated how the adaptive controller can learn online and how it compares to the PI controller.

A. Motor Primitives Activation and Affordance Evaluation

A grasping motion showing the activation of the motor primitives, is presented in Fig. 4(a). The plots show the activation signal and the individual activations of the hand and finger layers. Observe in *output finger layer* that the network generates a smooth trajectory with stable final states for each parameter. The hierarchy of the primitives is shown on the right as a tree color coded with the plot lines and the resulting grasping motion as a frame sequence on the bottom. To evaluate how the affordances

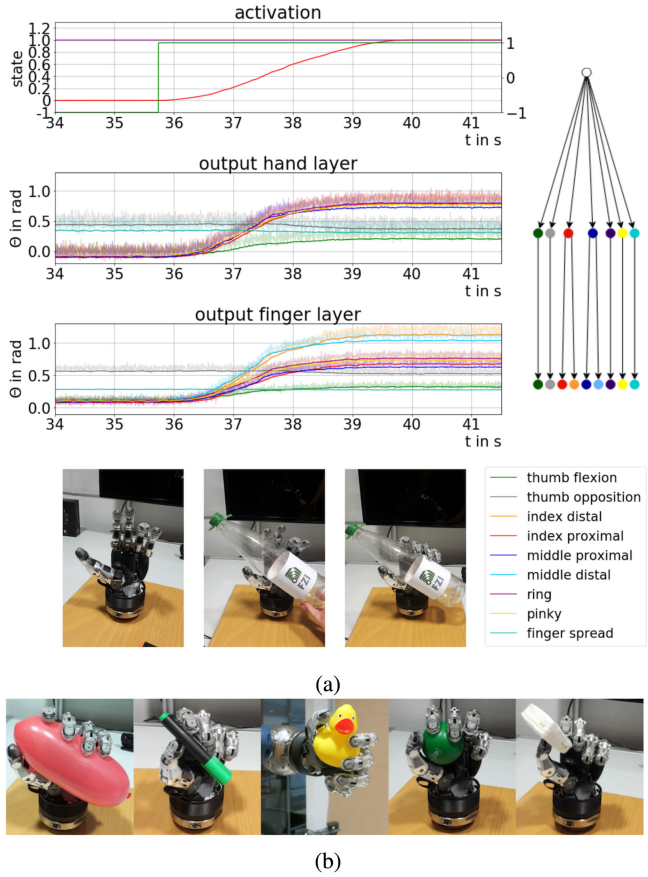

Fig. 4. (a) Affordance activation, network output of the hand and finger primitives, and a frame sequence of a grasp. (b) Experiments with different objects.

adapted, we used objects with different shapes, stiffness and sizes. Among them: a plastic bottle, a soft ball, a tennis ball, a sponge, a rubber duck, different balloons, a pen, and a tissue pack (see Fig. 4(b)).

B. Compliant Controller Evaluation

The effort threshold controls the force that a finger can apply to the object being grasped. The robot drivers have the option to set a safety maximum value for the allowed effort. The driver clips the control to protect the robot, which causes crucial deviations of the control error and results in delays at the start of the control. But changing this on-the-fly is problematic as the configuration of driver has to be reloaded. With our SNN, it is possible to change the effort threshold on-the-fly, which provides flexibility and another degree of freedom for the control. The compliant controller proved to be sensitive and the threshold could be set to be very low allowing the manipulation of balloons (see Fig. 5). In Fig. 5(a) the hand is pressing hard using the maximum effort threshold, whereas in Fig. 5(b) it is pressing soft using the effort threshold. In Fig. 5(c) the effort plot for the thumb finger for the maximum and minimum effort. Notice that the maximum is clipped by the driver. This maximum value is actually as low as the driver can go, because the effort caused by the non-linearities

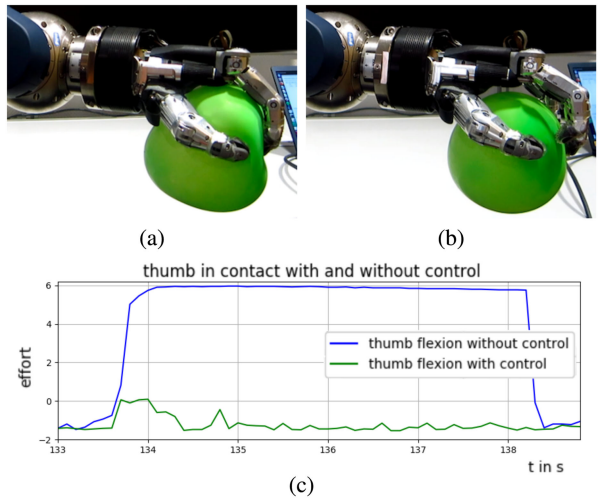

Fig. 5. Compliant control with (a) high and (b) low effort thresholds. (c) Effort plot for both cases.

of the robot dynamics will be higher and the driver will not move the robot. The internal driver defines maximum efforts for each joint as: thumb flexion 6; index proximal 5.5; index distal 4.1; middle distal 4.1; middle proximal 5.5; ring 2; and pinky 1.6. Our controller is able to go below these values and maintain the effort between -0.5 and $+0.5$ around the target value. The minimum effort we can achieve depends on the joint and it is between -1.5 and 1.0 thanks to the contact detection mechanism (see Sec. II-D).

A detailed grasp with the adjusted parameters is shown in Fig. 6. The output of the *effort control* is converted to a joint position and is added to the finger layer output. When contact is detected, the switch activates the controller and the *measured position* is affected by the effort controller.

C. Adaptive Control With Online Learning Evaluation

The network for the classical PI control is larger than for the adaptive controller (see Fig. 8). The parameters for the PI controller are presented in Fig. 8 (*bottom*). The initial controller parameters are calculated with the Ziegler Nichols method [34]. Then they are manually tuned to the system to avoid noise and oscillations. To compensate the delay between contact detection and reflex activation, an offset was aded to the controller. To use the adaptive control loop, the learning rate and the number of neurons had to be adjusted doing a manual parameter search. With higher learning rates, the system reached the desired efforts a faster but also oscillated. This effect was caused by the latency of the system, caused through the filters to reduce the noise of the spiking neurons. With the addition of an adaptive I part, the controller is reduced to a P controller, which is easier to parameterize.

With online learning, the controller can adapt and learn over multiple grasps. In Fig. 7, four consecutive grasps of the same object are shown. The first diagram shows the activation of the control triggered by the contact detection. If a contact is detected,

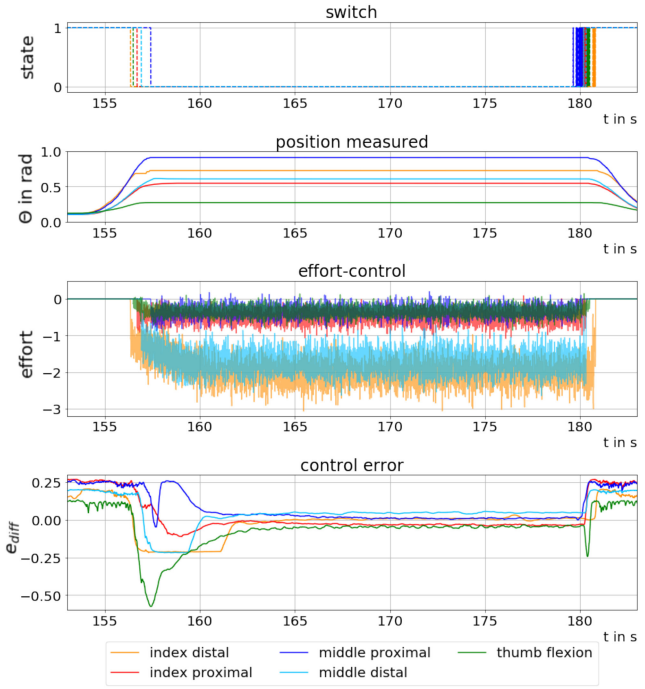

Fig. 6. Activation of the effort controller, output of the reflexes network and measured effort.

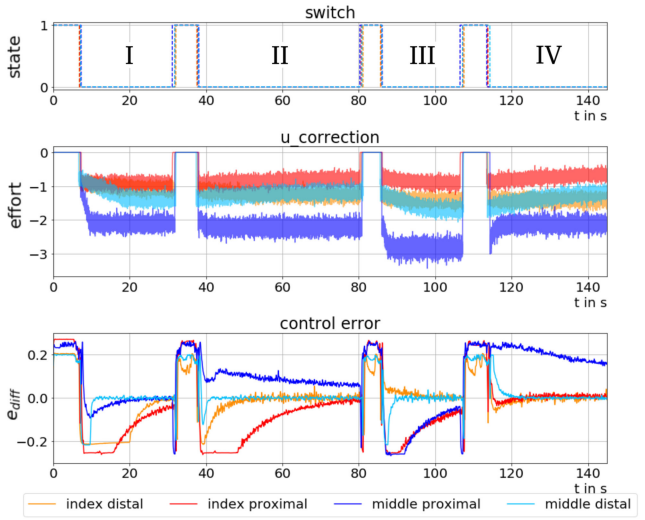

Fig. 7. The adaptive controller learns after multiple grasps.

the inhibitory neurons are inhibited (selective desinhibition [35]) and the control is activated. The plot *u_correction* is the signal added by the adaptive control to the PI control. The plot *control_error* shows the control error of four joints that are related by synergies – distal and proximal joints of the index and middle finger. In the fourth attempt, the control error is controlled faster than at the beginning. The delay is compensated and the gripping force is maintained to a constant value. The PI controller needs 5 s on average to minimize the control error into a range of 0.05. With the adaptive controller, this effect can be minimized to a delay of 3 s, after 5 grasps in average. As the repetition of a

Parameters finger primitives:

Finger	proximal		distal	
	Θmin	Θ_{max}	Θmin	Θ_{max}
thumb	0,1	0,9		
thumb opposition	0,25	0,9		
index finger	0,1	0,8	0,1	1,33
middle finger	0,1	0,8	0,1	1,33
ring finger	0,1	0,9		
pinky	0,1	0,9		
finger spread	0,2	0,5		

Parameters hand primitives:

Finger	cylinder		pinch		sphere	
	min	max	min	max	min	max
thumb	0	0,4	0	0,6	0	0,4
thumb opposition	0,5	0,5	1	1	0,2	0,8
index finger	0	0,9	0	0,9	0	0,9
middle finger	0	0,9	1	1	0	0,9
ring finger	0	0,9	1	1	0	0,4
pinky	0	0,9	1	1	0	0,9
finger spread	0,4	0,5	0,3	0,3	0,5	0,5

Parameters PI control:

	thumb	index finger		middle finger		ring finger	pinky
	flexion	distal	prox	prox	distal		
P part:	0.3	0.6	0.2	0.2	0.6	0.6	0.6
I part:	0.1	5	1.5	1.5	5	4.5	4.5
I part with offset:							
factor:	1	1	1	1	1	1	1
offset:	-0.1	-1.1	-0.5	-0.5	-1.1	-0.8	-1.1
adaptive P part:	0.2	1.8	1	1	2.5	2.0	1.3

Fig. 8. Parameter tables for the finger primitives, the hand primitives and the compliant controller.

grasp with the robot can not be the same every time, some joints need more grasps to be adapted. The signal of the middle finger proximal (dark blue) is overshooting in the second and fourth grasps. This is caused by the adjustment of the middle finger distal (light blue), nevertheless, after a period of time it stabilizes.

D. SNN Implementation and Parameters

The SNN is implemented with the neurosimulator Nengo [36] with leaky integrate and fire spiking neurons. The optimizer for the network connections is a least squares method defined in NEF [31]. The motor primitives are represented with 22 ensembles and 4400 spiking neurons, based on the modelling of the fingers in [28]. The motor primitives have a mathematical description that can be reused and parameterized ([30]), which allows further extension of the network. The reflexes with the classical PI compliant controller are represented with 36 ensembles and 20700 neurons, and with the adaptive compliant controller with 21 ensembles and 7350 neurons. The number of ensembles depends on the functions being modelled, and it is not a hyper parameter that was optimized. It also depends on the amount of inputs and outputs of each sub-network. The number of neurons is manually adjusted depending on the noise and the precision. The basic process is to start with 200 neurons for each ensemble, then depending on the noise and the required precision we increase the number of neurons and do a manual binary search. Additionally, to use the SNN for robot control, it is necessary to filter the network output to reduce noise from the spikes. This leads to a delay in the control. To keep this delay small, the number of neurons has to be increased (better fit of the function and less noise). Heuristic strategies were used to adjust the number of neurons to fit the chosen filter.

The Schunk SVH 5-finger hand has 9 active degrees of freedom – thumb distal, thumb opposition, index distal and proximal, middle distal and proximal, ring distal, pinky distal, and finger spread. We use the Robot Operating System (ROS) [37] as a communication layer with the official ROS driver for the hand [38]. The selected affordances are based on [13] with the parameters based on [14]. The primitives are defined with the presented functions, and the mapping to the robot is defined by moving the robot to the desired positions once, and then reading out the joint configuration. We only need one example motion for each affordance, from which we take the start and end configuration in joint space. The parameters for the finger and hand primitives, and for the compliant controller are presented in Fig. 8. In the (*top*) the corresponding Θ_{min} and Θ_{max} of each joint to define the finger primitives. In the (*middle*) the corresponding min and max values for the *cylinder, pinch and sphere* primitives – for the *rest* primitive the min and max values are set to zero. In the (*bottom*) the parameters for the PI, offset and adaptive P parts for each finger.

IV. Discussion

We presented a biologically inspired SNN to perform soft-grasping with an anthropomorphic robotic hand. Soft-grasping is mostly made with mechanical features and compliant hardware. Indeed, most of the robots are not hardware compliant and do not have force sensors, as the hand that was used. Nevertheless, with the combined control loops and using the current of the motors, it was possible to perform soft-grasping. The experiments show that it is possible to grasp objects with different shapes, stiffness and sizes without calculating the inverse kinematics or complex contact point planning. The system is based on motor primitives implemented with SNN organized in a hierarchy of joints, fingers, reflexes and grasping affordances representing the hand. Using separate primitives, it is possible to control different combinations of joints and activate other primitives with only one activation signal. The approach is flexible and can be used on different robot hands and can also be extended to incorporate signals from other networks. The compliant control is implemented in the same SNN using a cascaded PI effort controller that was extended with online learning for adaptive control. The controller was able to adapt the grasping motions to the different objects without knowing their exact properties. We modelled three grasping types – sphere, cylinder and pinch – but there is no limitation with this number and the SNN can be extended with more motions. An important characteristic of our approach is that the SNN requires one example or mathematical description of each grasping motion to train the primitives, and after learning, the SNN adapts the motion to the different objects. This is a real advantage in comparison to other bio-inspired grasping and manipulation approaches based on deep learning [18], [19], because training data and extensive datasets are expensive and not easily generated with real robots.

The same affordances were tested on objects with different sizes and stiffness. One of the experiments showed how the same spherical grasp was adapted for a tennis ball and for a balloon. The threshold for the effort controller can be changed on-the-fly by the network for each finger, which provides even more flexibility and another degree of freedom for the control. The compliant controller proved to be sensitive, and the threshold could be set as low as to allow even the manipulation of balloons (see Fig. 5). Even with the intrinsic inaccuracy of the current measurements and necessary filters, the efforts could be maintained below the maximum limits, which means that the compliant controller was actively controlling the joints. There are still bottlenecks in the control pipeline and the control signal has a delay. This is due to the different filters to get the information without noise in and out of the SNN. As a consequence both controllers present small oscillations. Parts of this problem could be improved by the online learning capabilities of the adaptive effort controller. The plot in Fig. 7 shows the online adaption of the controller with every repetition of the same grasping motion. Additionally, there are limitations with the internal controller of the robot driver. The positions are converted into currents that are cut off at a certain threshold for safety. In case of contact, this has the consequence that the output of the SNN does not control directly the joints, and the measured values do not match the controller output. The controller experiences a kind of wind up effect and the system oscillates when the threshold value in the driver-side controller is reached. Due to the damping properties of soft objects, the controller reacts without exceeding the threshold of the driver. Nevertheless, it also works with with stiff objects, as the force is controlled with the same threshold for soft or hard objects. For example the marker or the bottle were hard, and it had no problems with them.

The continuation of this work has three main directions: improve the controller, incorporate visual information and integrate arm motion. To improve the controller, the network parameters can be pre-trained with domain randomization in simulation as in [19], [39] to fine tune the adaptive controller. To increase the performance of the controller the SNN can be executed with neuromorphic hardware such as SpiNNaker [40] or Loihi [41] to take advantage of the efficient real time execution of SNN [42].Neuromorphic hardware can also be used to directly use the spike activity of the network to control the motors [43] or by using event-based touch sensors [44] to further exploit the characteristics of SNN in terms of energy consumption and information processing [4]. To incorporate visual information, we consider event-based cameras a natural match for SNN. An stereo even-based camera setup [45] can be used to get the target point is 3D space for pre-grasping [46] and use micro-saccades [47] to detect the type of object and identify which grasping affordance to use. To integrate arm motion, an arm controller as in [48] can be incorporated to do visual servoing, or as in [35] to have primitives reach different points on a surface.The combination of visual information and arm motion, together with soft-grasping can achieve a more natural grasping process from recognition of the object to positioning the arm to grasping. The whole system implemented completely with SNN as physical imitation of a biological system and an anthropomorphic robotic hand can be compared to brain neural responses for the grasping process as shown by [15] and provide new insights into its sub-processes.

REFERENCES

[1] D. Caldwell and N. Tsagarakis, ""Soft" grasping using a dextrous hand," *Ind. Robot*, vol. 27, no. 3, pp. 194–199, 2000.

[2] M. Bonilla *et al.*, "Grasping with Soft Hands," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 581–587.

[3] A. Wolf and H. A. Schunk, *Grippers in Motion: The Fascination of Automated Handling Tasks*. Munich, Germany: Carl Hanser Verlag GmbH Co KG, 2018.

[4] D. Zambrano, R. Nusselder, H. S. Scholte, and S. Bohte, "Efficient computation in adaptive artificial spiking neural networks," 2017, *arXiv:1710.04838*.

[5] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.

[6] E. Bizzi, V. Cheung, A. d'Avella, P. Saltiel, and M. Tresch, "Combining modules for movement," *Brain Res. Rev.*, vol. 57, no. 1, pp. 125–133, 2008.

[7] A. d'Avella, P. Saltiel, and E. Bizzi, "Combinations of muscle synergies in the construction of a natural motor behavior," *Nat. Neurosci.*, vol. 6, no. 3, pp. 300–308, 2003.

[8] P. Bogdan *et al.*, "Event-based computation: Unsupervised elementary motion decomposition," in *Proc. Emerg. Technol. Conf.*, 2019, pp. 20–23.

[9] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no. 2, pp. 328–373, 2013.

[10] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem," in *Proc. Robot.: Sci. Syst. Manipulation Workshop-Sens. Adapting to Real World*, 2007.

[11] A. I. Sburlea and G. R. Müller-Putz, "Exploring representations of human grasping in neural, muscle and kinematic signals," *Sci. Rep.*, vol. 8, no. 1, pp. 1–14, 2018.

[12] P. E. Crago, R. J. Nakai, and H. J. Chizeck, "Feedback regulation of hand grasp opening and contact force during stimulation of paralyzed muscle," *IEEE Trans. Biomed. Eng.*, vol. 38, no. 1, pp. 17–28, Jan. 1991.

[13] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 269–279, Jun. 1989.

[14] A. Scano, A. Chiavenna, L. Molinari Tosatti, H. Müller, and M. Atzori, "Muscle synergy analysis of a hand-grasp dataset: A limited subset of motor modules may underlie a large variety of grasps," *Front. Neurorobotics*, vol. 12, p. 57, 2018.

[15] S.-M. Kim, S.-Y. Hyun, J.-w. Sohn, S. Chae, and S.-P. Kim, "Neural response to grasp of robot hand from M1 area of Rhesus monkey," in *Proc. 7th Int. Winter Conf. Brain-Comput. Interface (BCI)*, 2019, pp. 1–4.

[16] S. W. Ruehl, C. Parlitz, G. Heppner, A. Hermann, A. Roennau, and R. Dillmann, "Experimental evaluation of the schunk 5-finger gripping hand for grasping tasks," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2014, pp. 2465–2470.

[17] F. Ficuciello, A. Federico, V. Lippiello, and B. Siciliano, "Synergies evaluation of the SCHUNK S5FH for grasping control," in *Adv. Robot Kinematics*, 2018, pp. 225–233.

[18] J. Starke, C. Eichmann, S. Ottenhaus, and T. Asfour, "Human-inspired representation of object-specific grasps for anthropomorphic hands," *Int. J. Humanoid Robot.*, vol. 17, no. 2, 2020, Art. no. 2050008.

[19] O. M. Andrychowicz *et al.*, "Learning dexterous in-hand manipulation," *Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[20] C. D. Santina *et al.*, "Learning from humans how to grasp: A data-driven architecture for autonomous grasping with anthropomorphic soft hands," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1533–1540, Apr. 2019.

[21] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Robot. Res.*, vol. 37, pp. 421–436, 2018.

[22] T. G. Thuruthel, S. H. Abidi, M. Cianchetti, C. Laschi, and E. Falotico, "A bistable soft gripper with mechanically embedded sensing and actuation for fast closed-loop grasping," *29th IEEE Int. Conf. Robot and Human Interactive Commun. (RO-MAN)}*, 2019, pp. 1049–1054.

[23] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward dynamics compliance control (FDCC): A new approach to cartesian compliance for robotic manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 4568–4575.

[24] T. DeWolf, T. C. Stewart, J.-J. Slotine, and C. Eliasmith, "A spiking neural model of adaptive arm control," *Proc. Roy. Soc. B: Biol. Sci.*, vol. 283, no. 1843, 2016, Art. no. 20162134.

[25] M. C. Capolei, E. Angelidis, E. Falotico, H. Hautop Lund, and S. Tolu, "A biomimetic control method increases the adaptability of a humanoid robot acting in a dynamic environment," *Front. Neurorobotics*, vol. 13, p. 70, 2019.

[26] G. Urbain, V. Barasuol, C. Semini, and J. Dambre *et al.*, "Stance control inspired by cerebellum stabilizes reflex-based locomotion on HyQ robot," 2020, *arXiv:2003.09327*.

[27] J. C. V. Tieck *et al.*, "Towards grasping with spiking neural networks for anthropomorphic robot hands," in *Proc. Int. Conf. Artif. Neural Netw.*, 2017, pp. 43–51.

[28] J. C. V. Tieck, S. Weber, T. C. Stewart, A. Roennau, and R. Dillmann, "Triggering robot hand reflexes with human EMG data using spiking neurons," in *Proc. Int. Conf. Intell. Auton. Syst. IAS-15*, vol. 867, pp. 902–916, 2018.

[29] J. C. V. Tieck *et al.*, "Combining spiking motor primitives with a behavior-based architecture to model locomotion for six-legged robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4161–4168.

[30] J. C. V. Tieck, L. Steffen, J. Kaiser, A. Roennau, and R. Dillmann, "Multi-modal motion activation for robot control using spiking neurons," in *Proc. IEEE Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, 2018, pp. 291–298.

[31] C. Eliasmith, *How to Build a Brain: A Neural Architecture for Biological Cognition*. London, U.K.: Oxford Univ. Press, 2013.

[32] E. Jankowska, "Spinal interneuronal systems: Identification, multifunctional character and reconfigurations in mammals," *J. Physiol.*, vol. 533, no. 1, pp. 31–40, 2001.

[33] D. MacNeil and C. Eliasmith, "Fine-Tuning and the Stability of Recurrent Neural Networks," *PLoS ONE*, vol. 6, no. 9, 2011, Art. no. e22885.

[34] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *ASME. J. Dyn. Sys. Meas. Control.*, vol. 115, no. 2B, pp. 220–222, Jun. 1993.

[35] J. C. V. Tieck, T. Schnell, J. Kaiser, F. Mauch, A. Roennau, and R. Dillmann, "Generating pointing motions for a humanoid robot by combining motor primitives," *Front. Neurorobotics*, vol. 13, p. 77, 2019.

[36] T. Bekolay *et al.*, "Nengo: A Python tool for building large-scale functional brain models," *Front. Neuroinformatics*, vol. 7, no. 48, pp. 1–13, 2014.

[37] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, vol. 3, p. 5, 2009.

[38] G. Heppner, "Schunk_svh_driver - ROS wiki," 2017. [Online]. Available: http://wiki.ros.org/schunk_svh_driver

[39] A. Vandesompele *et al.*, "Body randomization reduces the sim-to-real gap for compliant quadruped locomotion," *Front. Neurorobotics*, vol. 13, p. 9, 2019.

[40] S. B. Furber, S. Temple, and A. Brown, "High-performance computing for systems of spiking neurons," in *Proc. AISB'06 Workshop. GC5: Archit. Brain Mind*, vol. 2, pp. 29–36, 2006.

[41] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan./Feb. 2018.

[42] O. Rhodes *et al.*, "Real-time cortical simulation on neuromorphic hardware," *Philosoph. Trans. Roy. Soc. A*, vol. 378, no. 2164, 2020, Art. no. 20190160.

[43] E. Donati, F. Perez-Pena, C. Bartolozzi, G. Indiveri, and E. Chicca, "Open-loop neuromorphic controller implemented on VLSI devices," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics (Biorob)*, 2018, pp. 827–832.

[44] G. Haessig *et al.*, "Event-based computation for touch localization based on precise spike timing," *Front. Neuroscience*, vol. 14, p. 420, 2020.

[45] J. Kaiser *et al.*, "Microsaccades for neuromorphic stereo vision," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics (BIOROB)*, 2018, pp. 244–252.

[46] L. Steffen, S. Ulbrich, A. Roennau, and R. Dillmann, "Multi-view 3D reconstruction with self-organizing maps on event-based data," in *Proc. IEEE 19th Int. Conf. Adv. Robot.*, 2019, pp. 501–508.

[47] J. Kaiser *et al.*, "Embodied neuromorphic vision with event-driven random backpropagation," 2019, *arXiv:1904.04805*.

[48] J. C. V. Tieck, L. Steffen, J. Kaiser, D. Reichard, A. Roennau, and R. Dillmann, "Combining motor primitives for perception driven target reaching with spiking neurons," *Cogn. Informat. Natural Intell.*, vol. 13, no. 1, p. 12, 2019.