

DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

**Autonomous Control for Wheel-less snake
Robot under the use of an Neuromorphic
Vision Sensor and Spiking Neural
Networks**

Richard Otto

DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

**Autonomous Control for Wheel-less snake
Robot under the use of an Neuromorphic
Vision Sensor and Spiking Neural
Networks**

**Autonome Steuerung für "Wheel-less
Snake" Roboter, aufbauend auf einem
Neuromorphic Vision Sensor und Spiking
Neural Networks**

Author: Richard Otto

Supervisor: Prof. Dr.-Ing. habil. Alois Christian Knoll

Advisor: Jiang Zhuangyi

Submission Date: 16.08.2019

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 16.08.2019

Richard Otto

Acknowledgments

Abstract

Contents

Acknowledgments	iii
Abstract	iv
1 Theoretical Background	1
1.1 Biological Background	1
1.2 Artificial Neural Networks	3
1.3 Second generation of ANN	4
1.4 Spiking neural Networks	7
1.4.1 Leaky integrate-and-fire neuron	7
1.4.2 Stdp learning	7
1.4.3 R-STDP	8
1.4.4 Dynamic Vision Sensor	9
1.4.5 Snake like Robot	11
1.4.6 Neuro Robotics Platform	13
List of Figures	15
List of Tables	16
Bibliography	17

1 Theoretical Background

Spiking Neural Networks (SNNs) try to mimic the functional mechanisms of brains. This can give a better understanding on how brains work and learn, as well as the option to build artificial brains in form of SNNs and use them to solve tasks. The complexity of these Artificial Neural Networks (ANN) is capped by the available computation time or specialised hardware as well as the understanding of how to structure the network and set it's parameters. To get a complete picture of the ideas and mechanisms that are enabling a SNN to learn how to solve tasks, this chapter gives an introduction to basic mechanisms in biological brains and an overview about the evolution of Artificial Neural Networks.

1.1 Biological Background

The knowledge about the structure and understanding of what enables brains to perform complex tasks is still incomplete. As result of lacking methods and instruments to examine the structure of brains only in the late 19th century the Neuron as primary functional unit was discovered. Spanish anatomist Santiago Ramón y Cajal proposed that neurons are, other to former believes, discrete functional units and not connected in a meshwork. This is called the Neuron doctrine and got updated and refined in the following centuries[Bul59]. Up to today it is the foundation of todays understanding of the brain. The key elements of the doctrine specify that the brain is made up of individual units which are cells and can be specialised. These units are called Neurons and are generated by cell division. They are connected by Synapses and are mainly one directional and can either be excitatory or inhibitory[LBA06].

Todays model of a neuron consists of many different parts necessary to function. These are shown in the schemata in Figure 1.1. The main part of a cell is called soma. It is the body of a cell and holds the nucleus. Together they are responsible for keeping the cell alive as well as reproduction, as the nucleus also holds the DNA. The treelike structure reaching out from the Soma is called Dendrite. The different branches end near the axons of other cells and accept stimuli from them. It also tends to branch out from few main stems, channeling the connection from many other cells. Stimuli from other neurons change the electrical potential of the cell membrane. While the membrane potential degrades slowly over time, receiving enough excitation overcoming the decay,

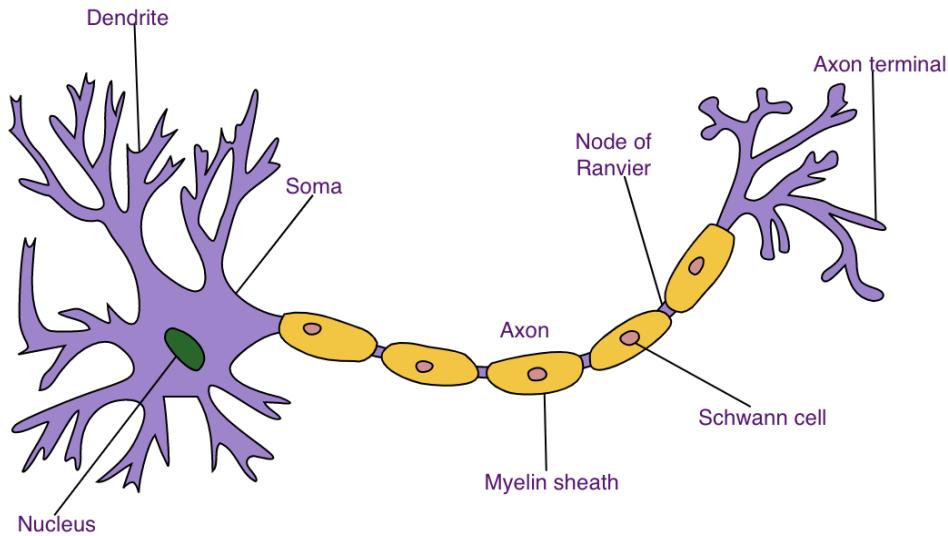


Figure 1.1: Overview about the main parts of a Neuron [19c]

the potential rises to a threshold level and an electrical impulse discharges along the axon to the axon terminals resetting the membrane potential. The axon leads the signal to other neurons and largely varies in length as it may connect to other brain regions. In the area around the end of the axon it again branches out in small axon terminals connecting to the dendrit of different neurons. Human brains are estimated to have 86 billion neurons forming a complex network with their connection.

As the neurons are separate cells there is a gap between axon terminals and dendrit which is called Synaptic Gap. There are two main types of synapses differentiated by the way the information is passed. One type is based on electric transmission of the signal the other uses chemicals to travel the gap and stimulate the neuron. The chemical transmission is believed to make up most of the connection and is also the one simulated by the spiking neural networks used for this thesis. In Figure 1.2 the main parts building a chemical synapse are shown. To transmit an incoming signal the axon terminal releases neurotransmitters to the gap which can be received by the dendrit through receptors. Neurotransmitters are build in the terminal or reabsorbed from the gap. In the terminal they are hold in Synaptic Vesicles which can fuse with voltage gated calcium channels releasing the transmitters to the gap[Cat+05]. How good the connection transmits the signal depends mainly on the amount of neurotransmitters released as well as the amount of receptors at the dendrite. Additionally to the

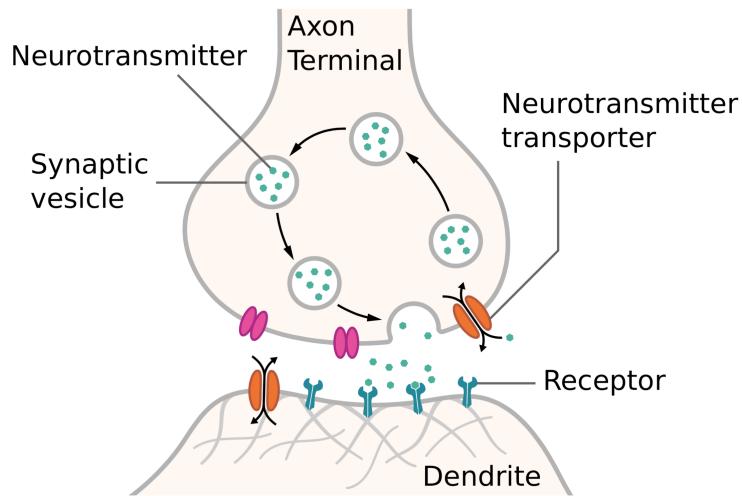


Figure 1.2: Schemata of a Synapse [19d]

neurotransmitters from the axon neuromodulators eg. Dopamine can be present in the synaptic gap which are believed to change how the synapse evolves over time and thus amplify or diminish changes in the strength of the connection.

1.2 Artificial Neural Networks

To explore how these networks act neurophysiologist Warren McCulloch and mathematician Walter Pitts developed 1943 the first mathematical model [MP43]. This model, known as McCulloch-Pitts-Model is a strongly simplified description of a Neuron and shown in Figure 1.3. It has binary inputs and a binary output. If the sum over the inputs exceeds a threshold the output is 1 otherwise it is 0. Inventing the Perceptron Frank Rosenblatt proposed 1962 a model overcoming the limitations to boolean inputs and using real-valued weights on the inputs.

Equation 1.1 describes the behaviour of the perceptron where x is a real-valued input vector, w the vector of the weights of the same size and b a bias.

$$f(x) = \begin{cases} 1 & \text{if } w * x + b > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (1.1)$$

By choosing the right weights the perceptron can be used as classifier for linear separable data. Further it can be shown that weights can be learned and converge

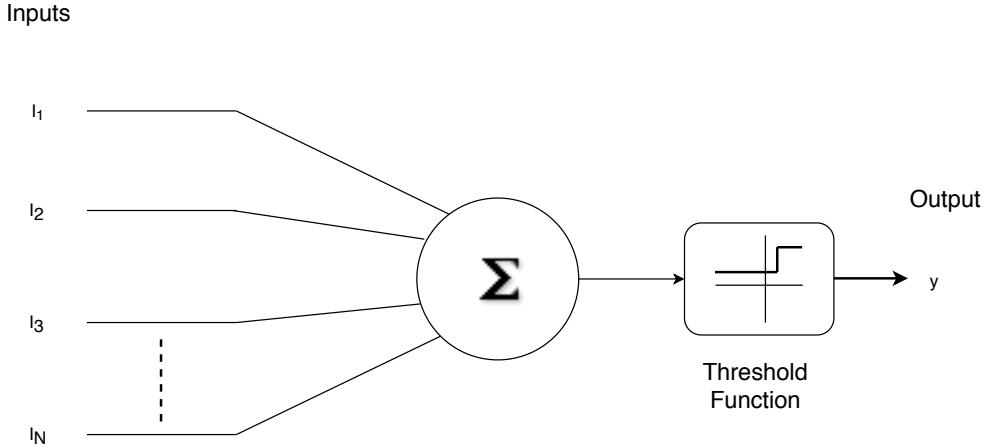


Figure 1.3: Schemata of a Synapse [19d]

in linear separable training sets [Bis06]. The algorithm calculates the weight change from comparing the result y_j calculated by the neuron from the feature set x_j , with the desired result d_j for each training sample j . The weight change for every connection is the result of multiplying the error $d - y$ by the learning rate r and the input x of that connection as shown in Equation 1.2 .

$$w_i(t+1) = w_i(t) + r * (d_j - y_j(t)) * x_{ji} \text{ , for all features } 0 \leq i \leq n \quad (1.2)$$

By iterating over the set of training samples and updating the weights after each step these converge to the correct values to classify the data.

For this learning rule the bias is described by setting x_0 always to 1 and have the feature set of size n represented by x_1 to x_n . Doing so w_0 function as bias for the threshold function and gets adjusted in the learning process.

1.3 Second generation of ANN

To be able to solve non-linear classifiable data sets multiple perceptrons can be chained to form a network. This construct is referred as Multilayer Perceptron (MP) and consists of at least three layers, the input, hidden and output layer. Different to a single perceptron the activation function of neurons in MPs are sigmoid shaped. Historically most common is the logistic function:

$$y_i(v_i) = (1 + e^{-v_i})^{-1} \quad (1.3)$$

where y is the output of the i -th neuron for the weighted sum v_i of its input connections. Figure 1.4 shows the shape of this function.

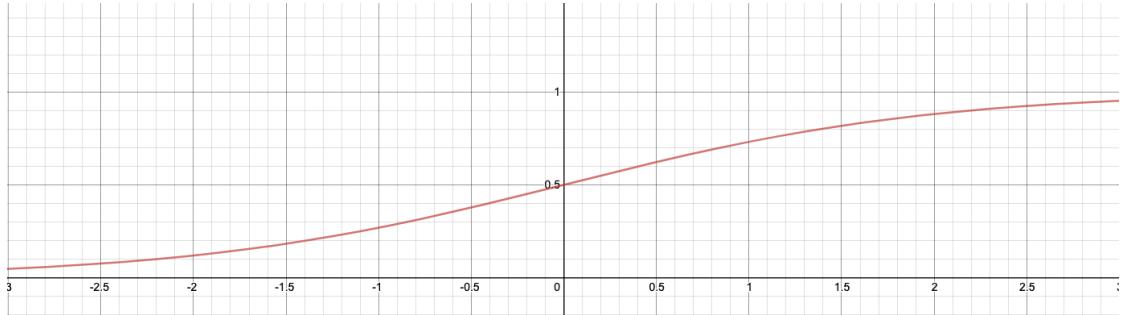


Figure 1.4: Plot of a logistic function

For learning the weights an algorithm called back-propagation was proposed 1986 by David Everett Rumelhart and others [RHW+88]. It generalizes the learning function of the single perceptron. It consists of two main steps for each training sample. First the output for the feature set x is calculated what is also referenced as information forward propagation. The overall error is the sum of the squared difference in output o to the target t in each output node and is described by following functions:

$$E = \frac{1}{2} \sum_{i=1}^n (t_i - o_i)^2 \quad (1.4)$$

The aim of the back-propagation algorithm is to minimize this error. This is realized by propagating the error in each output back through the network and changing the weights of each connection which is the second step and referred as error back propagation. In this step for each connection the partial derivative of the error with respect to its weight is calculated which describes the impact a change of this weight has on the error and in which direction it changes the error. For each connection this depends on the impact it has on the result o_j of the node it connects to and its impact on the error.

This can formally be described by the first step in this equation:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}} \quad (1.5)$$

The second step is to add in that the impact on the output in o_j is dependent on the weights influence on the activation sum net_j of node. In this last factor only one term

in the sum depends on w_{ij} thus its derivat is o_i

$$\frac{\partial \text{net}_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left(\sum_{k=1}^n w_{kj} o_{kj} \right) = \frac{\partial}{\partial w_{ij}} w_{ij} o_i = o_i \quad (1.6)$$

The second factor in Equation 1.5 is the derivative of the output of a node with respect to its input which is the partial derivative of the used activation function and also the reason backpropagation needs a differentiable activation function.

For connections to output neurons o_j equals to y_i and the first factor can be rewritten as

$$\frac{\partial E}{\partial o_j} = \frac{\partial E}{\partial y_j} \quad (1.7)$$

The derivative for the square error from Equation 1.4 again only depending in one sum on y resolves to

$$\frac{\partial E}{\partial o_j} = \frac{\partial E}{\partial y_j} = \frac{\partial}{\partial y_j} \frac{1}{2} (t - y)^2 = y - t \quad (1.8)$$

Plugging everything together through substitution, the derivative can be rewritten as

$$\frac{\partial E}{\partial w_{ij}} = o_i \delta_j \quad (1.9)$$

with

$$\delta_j = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} = \begin{cases} \frac{\partial L(o_j, t)}{\partial o_j} \frac{\partial \varphi(\text{net}_j)}{\partial \text{net}_j} & \text{if } j \text{ is an output neuron,} \\ (\sum_{l \in L} w_{jl} \delta_l) \frac{\partial \varphi(\text{net}_j)}{\partial \text{net}_j} & \text{if } j \text{ is an inner neuron.} \end{cases} \quad (1.10)$$

Where L represents the loss function which in our case is $(t - y)^2$ and φ the activation function in Equation 1.3. For these the derivatives simplify to

$$\delta_j = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} = \begin{cases} (o_j - t_j) o_j (1 - o_j) & \text{if } j \text{ is an output neuron,} \\ (\sum_{l \in L} w_{jl} \delta_l) o_j (1 - o_j) & \text{if } j \text{ is an inner neuron.} \end{cases} \quad (1.11)$$

The weight change of the connection from node i to node j is calculated by following equation

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta o_i \delta_j \quad (1.12)$$

where the gradient is multiplied by $\eta > 0$ which is the learning rate and can be thought as small step along the gradient. Is the gradient positive an increase in w_{ij} would increase E and vice versa. Therefore the negative gradient is used as the minimum for the function should be found. As for inner notes its δ is recursively defined by the δ of all nodes it connects to, the weight change depends on the error

information traveling from the output layer backwards to the input layer[SG13]. While the model of a neuron in the MP is not simulating electrical pulses as they occur in biological neurons the activation level of the artificial ones can be viewed as an abstraction of the mean spike rate of that neuron.

1.4 Spiking neural Networks

The neuron model used in Spiking Neural Networks closes the gap to real neurons even further by communicating through impulses the spikes. In difference to the previous generations neurons can activate and send out an impulse at any time rather than have an activation level at discrete timesteps. This adds spatial-temporal information to the dynamics of the network and in theory it has been shown that these networks are more powerful than their predecessors and considered the third generation of artificial neural networks [Maa97].

The change to spike events as encoding information brings several changes to the mathematical description of the neurons and synapses.

1.4.1 Leaky integrate-and-fire neuron

Probably the best known model used for neurons in this regard is the Leaky Integrate and Fire model. In this model the membrane is thought of as electric capacitor which losses potential over time. The derivative of the membrane Voltage V is calculated by the function

$$\frac{\partial V}{\partial t} = -\frac{1}{\tau_m}V + \frac{1}{C_m}I(t) \quad (1.13)$$

τ_m is the membrane time constant inversely scaling the loss of voltage, $I(t)$ is the sum of all incoming current at time t and C_m is the capacity of the membrane. This holds in case the of the potential being smaller than the threshold value. Is the threshold reached the neuron fires and for the refractory period τ_{ref} its potential is set to the a reset potential V_{reset} . The current I_{syn} of the spike released to another neuron is determined by the function

$$I_{syn}(t) = we^{\frac{-t}{\tau_{syn}}} \quad (1.14)$$

with w being the weight of the synapse connecting the neurons and t_{syn} being the synaptic time constant.

1.4.2 Stdp learning

The Hebbian rule was proposed by psychologist Donald Hebb in 1949 claiming that the connection from neuron A to neuron B should be strengthened if A consistently takes

part in firing **B**. Which is often simplified to “Cells that fire together, wire together”. As the learning happens without influences it can be categorized as unsupervised learning. One shortcoming of the Hebbian rule is that it doesn’t cover depression of a connection as well as that to take part in the firing of a neuron the presynaptic neuron has to fire not at the same time but slightly before the postsynaptic neuron. Later experiments by Henry Markram suggested that the plasticity of a synapse is changed by exact timings of spikes in the pre- and postsynaptic membranes. If the presynaptic spike is followed by a postsynaptic one the connection is strengthened in the other case weakened. The time difference Δt is

$$\Delta t = t_{post} - t_{pre} \quad (1.15)$$

and is used for calculating the weight change defined by

$$STDP(\Delta t) = \begin{cases} A_+ e^{-\frac{|\Delta t|}{\tau_+}} & \text{if } \Delta t > 0 \\ A_- e^{-\frac{|\Delta t|}{\tau_-}} & \text{if } \Delta t \leq 0 \end{cases} \quad (1.16)$$

with A_+ and A_- as scaling factors for potentiation and depression and τ_- and τ_+ defining the height of the learning window. Figure 1.5

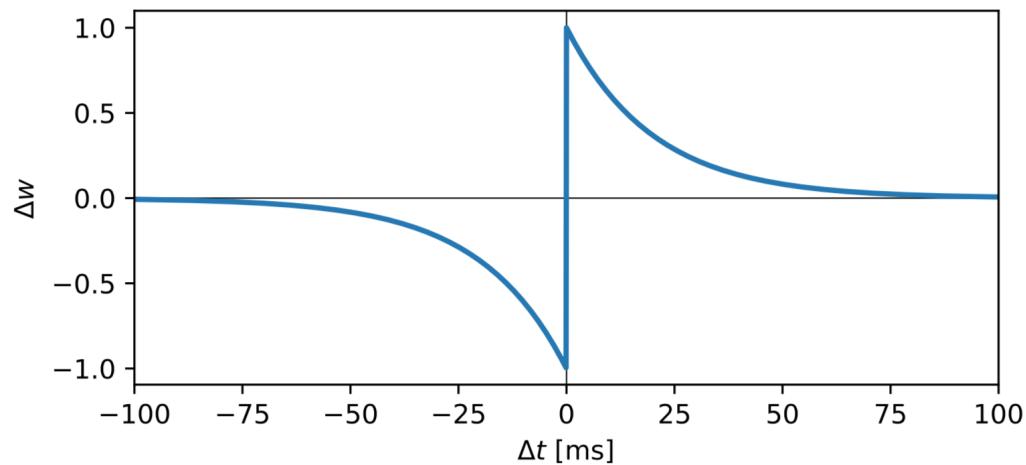


Figure 1.5: Plot of the STDP learning window

shows the plot of the STDP learning function for A_+ and A_- of the same magnitude of 1, setting the maximum values for the function.

1.4.3 R-STDP

In 2007 Izhikevich proposed a learning rule for Reward-modulated Spike-Timing-Dependent-Pasticity. In his model the weight changes calculated by the STDP mecha-

nism are collected in an eligibility trace rather then be applied directly. This is necessary as the reward from the environment is not available at the time of the spike event and can thus not be factored in for modifying the weight. The change of the eligibility trace is defined by

$$\dot{c} = -\frac{c}{\tau_c} + STDP(\Delta t)\delta(t - s_{pre/post})C_1 , \quad (1.17)$$

where τ_c is the time constant, δ the dirac delta function, $s_{pre/post}$ the timing of the second spike of a spike pair s_{pre} and s_{post} and C_1 a constant coefficient. If no spikes occur the eligibility will decay. Only at the time of the second spike the result of the STDP learning rule multiplied by the coefficient is added, which will increase the eligibility if the presynaptic spike occurs before the postsynaptic. The dynamics of the reward are described by

$$\dot{n} = -\frac{n}{\tau_n} + \frac{\delta(t - s_n)}{\tau_n}C_2 , \quad (1.18)$$

here n is the neuromodulator concentration, τ_n its time constant, s_n the spike time of the neuromodulator and C_2 a constant coefficient. The weight change according to R-STDP is then given by

$$\dot{w} = c(n - b) \quad (1.19)$$

With b being the baseline concentration of the neuromodulator. A visualisation of this dynamic is shown in Figure 1.6 taken from and further described in [LPM08]. **A** shows the eligibility function for a positive STDP outcome with the second spike time at $t = 0$. The second part **B** shows in the first row the spike-trains of the connected neurons. In the second row the red line is the effect of a positive STDP result on the eligibility trace, green the effect of a negative one and black the resulting eligibility. The third row displays the neuromodulator concentration with a spike occurring and the fourth row displays how the weight changes in this scenario.

In the experiments described in this thesis the Leaky integrate-and-fire neuron will be used in combination of the R-STDP learning rule for synaptic weights to build spiking neural networks.

1.4.4 Dynamic Vision Sensor

While conventional cameras are getting better and better this is not particularly great for robotic and autonomous mobile applications. The reason is that newer cameras tend to have a higher resolution or refresh rate of their frames causing a higher data volume from the cameras. The data probably includes more information than cameras delivered a year ago but more data also means more work to do not only analysing it but also just by handling it. This leads to the consumption of more energy and waiting for processing to happen can be a hindrance for real time applications. Thus for robotic vision an

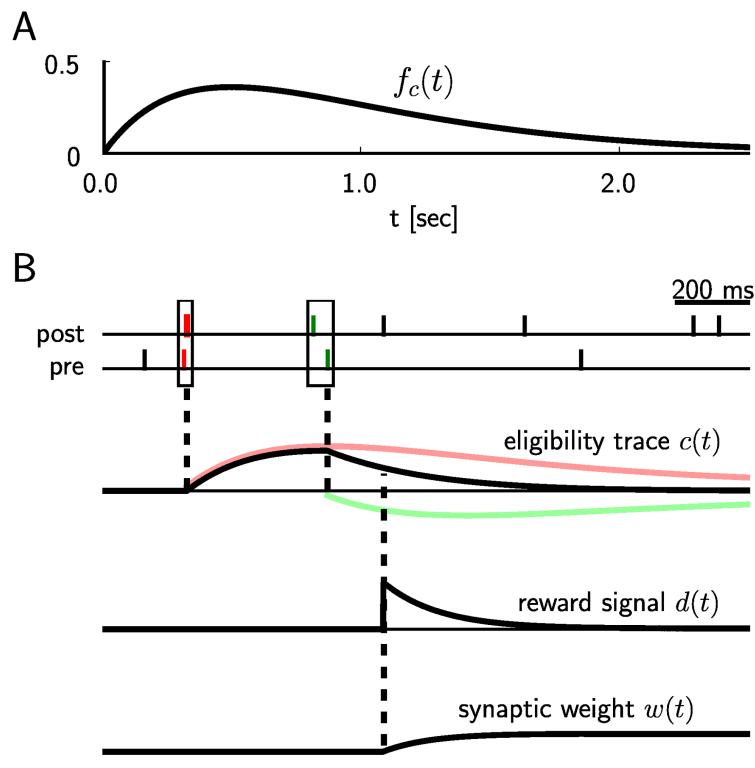


Figure 1.6: In his paper Legenstein described the R-StdP Mechanism [LPM08]:

(A) Eligibility function $f_c(t)$, which scales the contribution of a pre/post spike pair (with the second spike at time 0) to the eligibility trace $c(t)$ at time t . (B) Contribution of a pre-before-post spike pair (in red) and a post-before-pre spike pair (in green) to the eligibility trace $c(t)$ (in black), which is the sum of the red and green curves.

optical system reducing data load while keeping up the information flow from the real world to the system is needed. The huge amount of data of cameras is structured in frames and the frames contain the information of their pixels. This way of capturing video data is that for called framebased. Naturally but especially for tasks with fast moving a very high frame rate is needed this leads to a huge amount of overhead as most pixel in each frame tend to have almost the same values in the next frame and thus the data for these pixels in the next frame has no additional information. New information is only introduced if the value of a pixel changes meaningfully. To only transmit data if there is a meaningful change is the concept behind Dynamic Vision Sensors DVS entering the market only in recent years. Their functional principles are often compared to the one of the human retina as their receptor circuits keep track of the change in luminance and are able to send events at any time the change overcomes a threshold. The events are sent from the camera over an eventbus and can be described as the tuple $\langle x, y, t, p \rangle$ containing the x and y coordinates of the pixel, the time the event occurred and the polarity of the change. This event based principle enables the camera to achieve a very high time resolution, at the moment in the area of microseconds. Furthermore, the similarity between the events of a DVS camera to spikes makes these sensors ideal for the use with SNNs. For a better understanding it is helpful to keep in mind that in no point in time a complete picture of the environment is present in the system and thus the output for a completely static environment would be entirely black. For a rotating dot the graphic in Figure 1.7 shows the output of a dvs sensor over time as well as a slice, accumulating the events over 300 microseconds. The graphics is the result of experiments done by Lichtensteiner and are more closely described in his paper [LPD08].

The other examples are different scenes where either the camera was moved, resulting in the darker images, or the camera was static resulting in the bright images where white means that no event was detected for that pixel over the time period of accumulation.

1.4.5 Snake like Robot

The Robot used for the experiments tries to reassemble the shape and degrees of freedom a snake has. In difference to a real snake, which can continuously curve its body, the robot has a modular design with rotational joints between the modules. The axes of the joints are rotated by 90° along the body axe enabling three dimensional movement with abundant degrees of freedom. This allows the robot to perform snake like movement patterns like slithering, climbing and sidewinding as well as rolling sideways in an arch shaped body configuration, but various other movement patterns can also be thought of depending on environment. Examples for these different gaits are shown in Figure 1.8 made by Jiang for a presentation[Jia18] Snake like robots can

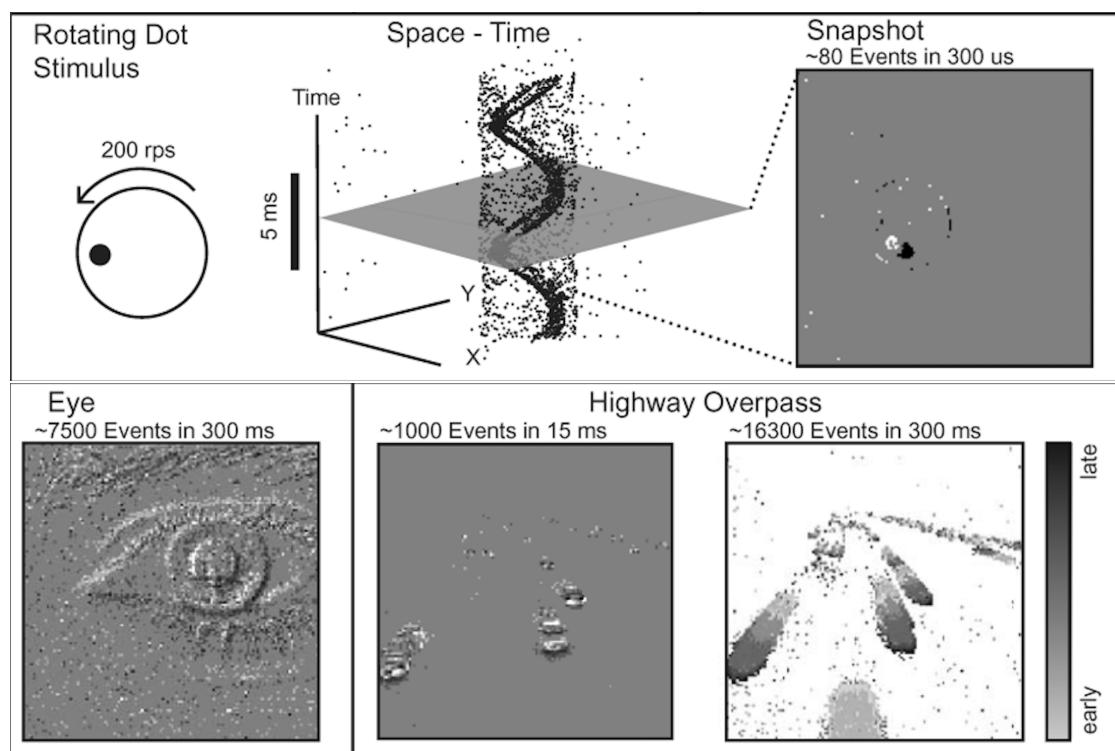


Figure 1.7: Rotating Dot stimulus visualised over time and additional Dvs pictures

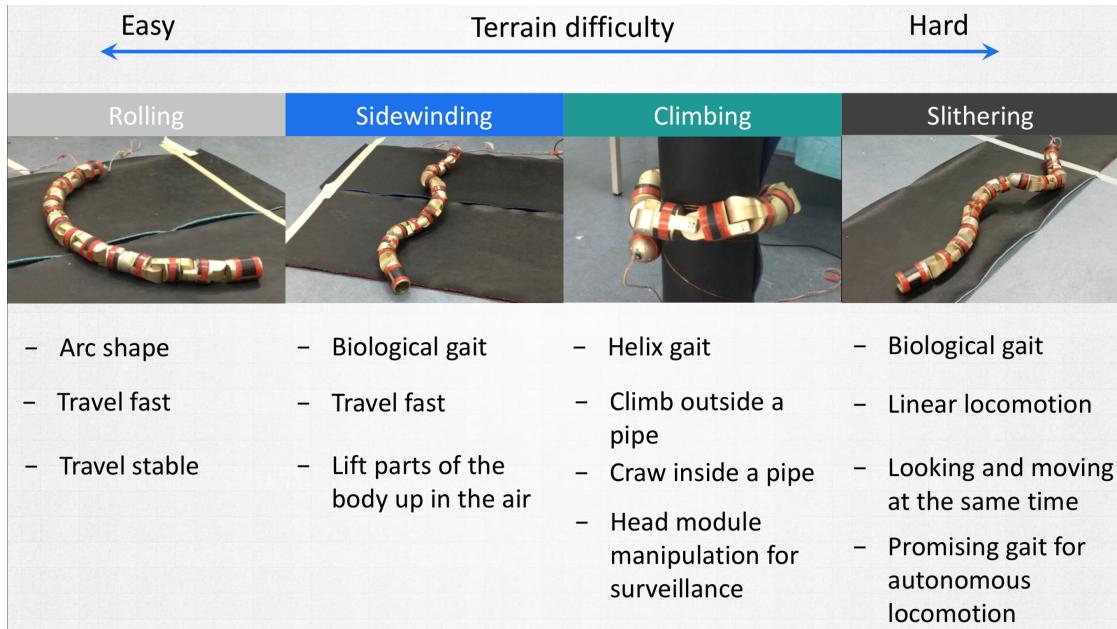


Figure 1.8: Slide of Presentation by Jiang showing the different gaits of the robot[Jia18]

navigate through tiny space like pipes and holes as well are able to climb which makes them especially interesting for the use in difficult terrain such as in rescue missions after earthquakes. For the experiments the slithering gait is used and the network only tells the robot in which direction it should go. In the slithering gait the robots head will move from one side to the other while keeping its orientation parallel to the movement direction of the robot. In this way the robot is able to sense the environment in front of it while moving. As a result of this movement pattern objects in front of the robot will change their position in sensing area throughout the motion.

1.4.6 Neuro Robotics Platform

The experiments done in this paper where simulated in the Neuro Robotics Platform (NRP) [19a] developed in the Human Brain Project [19b]. NRP enables the physical simulation of robots in environments as well as the integration of SNN to control the robots and thus gives the boilerplate to conduct experiments. Its core is a Closed Loop Engine (CLE) synchronizing the simulation of the SNN in NEST and the simulation of the world in ROS/Gazebo. To do so it pauses both simulations after a defined timespan and exchanges data between the simulations and updating their state. The interaction between the environment and the SNN is defined in Transfer Functions (TF) which for

example set the target angles for each joint of the snake.

List of Figures

1.1	Overview about the main parts of a Neuron [19c]	2
1.2	Schemata of a Synapse [19d]	3
1.3	Schemata of McCulloch-Pitts-Model	4
1.4	Plot of a logistic function	5
1.5	Plot of the STDP learning window	8
1.6	In his paper Legenstein described the R-StdP Mechanism [LPM08]: (A) Eligibility function $f_c(t)$, which scales the contribution of a pre/post spike pair (with the second spike at time 0) to the eligibility trace $c(t)$ at time t . (B) Contribution of a pre-before-post spike pair (in red) and a post-before-pre spike pair (in green) to the eligibility trace $c(t)$ (in black), which is the sum of the red and green curves.	10
1.7	Rotating Dot stimulus visualised over time and additional Dvs pictures	12
1.8	Slide of Presentation by Jiang showing the different gaits of the robot[Jia18]	13

List of Tables

Bibliography

- [19a] *HBP Neurorobotics Platform*. Human Brain Project. July 4, 2019. URL: <https://neurorobotics.net>.
- [19b] *Human Brain Project*. Human Brain Project. July 4, 2019. URL: <https://www.humanbrainproject.eu/en/about/overview/>.
- [19c] *Neuron*. Wikipedia. June 12, 2019. URL: <https://en.wikipedia.org/wiki/Neuron>.
- [19d] *Synapse*. Wikipedia. June 12, 2019. URL: <https://en.wikipedia.org/wiki/Synapse>.
- [Bis06] C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Softcover published in 2016. New York, NY: Springer, 2006.
- [Bul59] T. H. Bullock. “Neuron Doctrine and Electrophysiology.” In: *Science* 129.3355 (1959), pp. 997–1002. ISSN: 00368075, 10959203.
- [Cat+05] W. A. Catterall, E. Perez-Reyes, T. P. Snutch, and J. Striessnig. “International Union of Pharmacology. XLVIII. Nomenclature and Structure-Function Relationships of Voltage-Gated Calcium Channels.” In: *Pharmacological Reviews* 57.4 (2005), pp. 411–425. ISSN: 0031-6997. DOI: 10.1124/pr.57.4.5. eprint: <http://pharmrev.aspetjournals.org/content/57/4/411.full.pdf>.
- [Jia18] Z. Jiang. *Development and Control of a Bio-snake Robot*. Oct. 2018.
- [LBA06] F. López-Muñoz, J. Boya, and C. Alamo. “Neuron theory, the cornerstone of neuroscience, on the centenary of the Nobel Prize award to Santiago Ramón y Cajal.” In: *Brain research bulletin* 70.4-6 (2006), pp. 391–405.
- [LPD08] P. Lichtsteiner, C. Posch, and T. Delbrück. “A 128times 128 120 dB 15μs Latency Asynchronous Temporal Contrast Visions Sensor.” In: *IEEE journal of solid-state circuits* 43.2 (2008), pp. 566–576.
- [LPM08] R. Legenstein, D. Pecevski, and W. Maass. “A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback.” In: *PLoS computational biology* 4.10 (2008), e1000180.

Bibliography

- [Maa97] W. Maass. “Networks of spiking neurons: the third generation of neural network models.” In: *Neural networks* 10.9 (1997), pp. 1659–1671.
- [MP43] W. S. McCulloch and W. Pitts. “A logical calculus of the ideas immanent in nervous activity.” In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.
- [RHW+88] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. “Learning representations by back-propagating errors.” In: *Cognitive modeling* 5.3 (1988), p. 1.
- [SG13] I. Sporea and A. Grüning. “Supervised learning in multilayer spiking neural networks.” In: *Neural computation* 25.2 (2013), pp. 473–509.