DEGREE PROJECT IN TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2022

Template

KTH Thesis Report

Max Schaufelberger

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Author

Max Schaufelberger <maxscha@kth.se>
School of Engineering Sciences
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden
Ottignies-Louvain-la-Neuve, Belgium

Examiner

Prof. Olof Runborg
Department of Numerical Analysis
KTH Royal Institute of Technology
Stockholm, Sweden

Supervisor

Prof. Elias Jarlebring
Department of Numerical Analysis
KTH Royal Institute of Technology
Stockholm, Sweden

Supervisor

Arvind Kumar
Division of Computational Science and Technology
KTH Royal Institute of Technology
Stockholm, Sweden

Supervisor

Frédéric Crevecoeur
Institute of Information and Communication Technologies, Electronics and Applied Mathematics
UCLouvain Catholic University of Louvain
Louvain-la-Neuve, Belgium

Abstract

This is a template for writing thesis reports for the ICT school at KTH. I do not own any of the images provided in the template and this can only be used to submit thesis work for KTH.

The report needs to be compiled using XeLaTeX as different fonts are needed for the project to look like the original report. You might have to change this manually in overleaf.

This template was created by Hannes Rabo <hannes.rabo@gmail.com or hrabo@kth.se> from the template provided by KTH. You can send me an email if you need help in making it work for you.

Write an abstract. Introduce the subject area for the project and describe the problems that are solved and described in the thesis. Present how the problems have been solved, methods used and present results for the project. Use probably one sentence for each chapter in the final report.

The presentation of the results should be the main part of the abstract. Use about ½ A4-page. English abstract

Keywords

Template, Thesis, Keywords ...

Abstract

Svenskt abstract Svensk version av abstract – samma titel på svenska som på engelska.

Skriv samma abstract på svenska. Introducera ämnet för projektet och beskriv problemen som löses i materialet. Presentera

Nyckelord

Kandidat examensarbete, ...

Acknowledgements

Write a short acknowledgements. Don't forget to give some credit to the examiner and supervisor.

Acronyms

DS	Dynamic System
NN	Neural Network
ANN	Artificial Neural Network
SNN	Spiking neural network
GPU	Graphics Processing Unit
SOP	Synaptic Operation
IF	Integrate and Fire
LIF	Leaky-integrate-and-fire
LSM	Liquid State Machine
HH	Hodgkin–Huxley
NLP	Natural Language Processing
LQG	Linear Quadratic Gaussian

Contents

Chapter 1

Introduction

Provide a general introduction to the area for the degree project. Use references!

Link things together with references. This is a reference to a section: 1.1.

The human brain is a brilliant computing unit comprised of around 86 billion[5] neurons. Each of these neurons can have thousands of connections to other neurons.. Between these connections, information travels trough the network as electrical impulses that interact with the neurons own electrical potential. The huge network complex of the human brain is capable of vastly different and intricate tasks. Yet some problems that are still next to impossible to solve by machines and classical algorithms alone. Moreover many machine implementations lack the speed, precision or flexibility of the human counterpart.

Maybe put some exact numbers he and a sourc

Sounds vag

Researchers try to remedy this by mimicking the brain's internal network structure to solve problems deemed unsuitable for classic algorithms.

A variety of different architectures for these Artificial Neural Networks (ANNs) have been proposed, with the most prevalent design being a feed-forward network. In these networks, information travels only in one direction and is not propagated by spikes but gradients usually set in $[0, 1]$ or $[-1, 1]$. These ANNs have made impressive progress in the fields of image recognition, autonomic driving,medical diagnosis[24] or Natural Language Processing (NLP) (using Transformers[26]).

This abstract representation bears advantages e.g in modelling and implementation but also gives away some key features of the human brain. Do to the information travelling only towards the output, feed-forward networks cannot build a memory or easily process temporal data. Recurrent models exist which allow for memory and

cite some L networks o others

sequential data input but loose some of the advantages compared to the Feed-Forward due to its increased complexity.
A third generation[21] of network architectures has risen, which aims to be even more biologically plausible. Inspired from nature, they implement spiking behaviour and recurrence found in the human brain. This newer form of Spiking neural network (SNN) is as powerful as the classic feed-forward but suited for temporal data.
While state of the art feed-forward networks are still outperforming SNNs, in some cases modern SNNs are on par or more performant with older feed-forward implementations. This comes with the added benefit of consuming much less power. Usually deep ANNs are run on Graphics Processing Units (GPUs), especially for training, in which the energy consumption can exceed 300W for modern chips[1]. The brain however is estimated to only consume about 20W [8] for immense computing capacity. Accompanying the SNN with neuromorphic hardware can yield a similar boost in efficiency with processors energy consumption in the pJ per Synaptic Operation (SOP)[16].

hile consuming a fraction of the energy using specialized neuromorphic hardware.

Now list the goal: We want to do it for DS and check how good they are. Then method and then work. Take from below

They lack in some others
Therefore considerable research went into it
Now we have spiking neural networks, that imitate the brain even more
We hope that with that we have even better performance
From very biological to very abstract there have been many proposals
Cost performance trade off.
Spiking networks have gained similar or exceeding performance compared to the artificial one in some areas-> refs
Key advantage is in the temporal dimensional gain.
One field they are suited well is the control of dynamic systems
In this thesis we use a spiking neural network to control a linear dynamic system
The usual way to simulate biological dynamic systems is using LQG control -> ref
I believe because of the energy minimization

[1] e.g. a NVidia RTX 3090

So we can compare them with usual NN and control in terms of performance... i guess
We start by giving an intro into spiking neural networks
Then spiking neural networks for dynamic systems
After control theory with SNN and maybe regular LQG control
Lately the learnign of SNNs for the control of dynamic system

Further work:
Maybe learning methods to control nonlinear dynamic systems
Maybe we can even do the adverserial attack to try to screw with the network.
Implement this on neuromorphic hardware

Problem:
Problem is that it is unnatural for classic NN to use temporal data.
They usually quantize it and make a big input layer -> ref
There are recurrent networks but ... they need to have smth bad as well
The LQG control is also not great for some reason I need to find
There are many spkiking network archetypes like poisson and GLm and balanced
Also problem is that for some spiking networks learning rules could be hard to come by.
There are many prospects though as for example ......->refs
Also usually learning rules smth of an inverse and that the brain does not have or do I believe arvind said

Method:
We use a SNN to to control any arbitrary DS
Balanced networks show a some key motives seen in the brain like poisson distribution and smth else ->ref
The SNN is to be trained with a STDP rule
Then compared to optimal weights
Then investigated about robustness and other things as many before
One part of robustness is trying to get the most essential nodes of the snn to function well.
¸ Then we have the potential to find a classic nn and train it with that ???

With that out of the way we can compare the performance of all the methods.
Then we could study the usability for biological interpretation.
Maybe even train time over performance or smth whatever

Work:
Explain the controller method aka what the math of the controller
In method explain the balanced and the derivation
In work summarize the implementation
Same for the conventional NN
Summarize the training method
Explain and derive the training method in method

Results:
To everthign mentioned in method for performance and so on
Answer the questions of the problem!!!!

1.1 Background

Present the background for the area. Give the context by explaining the parts that are needed to understand the degree project and thesis. (Still, keep in mind that this is an introductory part, which does not require too detailed description).

Use references[2]

Detailed description of the area should be moved to Chapter 2, where detailed information about background is given together with related work.

This background presents background to writing a report in latex.

Example citation [**Jones2017**] or for two authors: [**Jones2017**, **Liu2017**]

Look at sample table 1.1.1 for a table sample.

Boxes can be used to organize content

[2] You can also add footnotes if you want to clarify the content on the same page.

Table 1.1.1: Sample table. Make sure the column with adds up to 0.94 for a nice look.

SAMPLE	TABLE
One	Stuff 1
Two	Stuff 2
Three	Stuff 3

Development environment for prototype

```
Operating systems
computer: Linux - kernel 4.18.5-arch1-1-ARCH
android phone: 8.1.0

Build tools
exp (build tool): version 55.0.4

...
```

1.2 Problem

NN have excelled at many fields
Fields where they are not fit
aka temporal data
They have ways to compromise on that
-> reference
Spiking nn inherently temporal
more natural choice
However they also have problems
like the following:::: reference!!

1.3 Purpose

The purpose of the degree project/thesis is the purpose of the written material, i.e., the thesis. The thesis presents the work / discusses / illustrates and so on.

It is not "The project is about" even though this can be included in the purpose. If so, state the purpose of the project after purpose of the thesis).

Probably delete as a own paragraph but mention smth like that.

1.4 Goal

The goal means the goal of the degree project. Present following: the goal(s), deliverables and results of the project.

exact
fications.
example
ollability

The goal of this project is to create a SNN that can control any given linear Dynamic System (DS). Furthermore should the Neural Network (NN) be robust against failing neurons or connections to some degree. With this we can investigate to find the smallest SNN with acceptable performance. Its performance should be comparable or supersede conventional control systems, like Linear Quadratic Gaussian (LQG) control, or NNs. We expect better results to conventional NNs because of the SNN's natural way to use temporal data. For the SNN itself we desire similarities to the brain, such as high precision or Poisson distributed spiking. To mimic the brain's learning, we want to use local training rules that are biologically plausible. The network should be converge to the optimal parameters.. This requirement would allow us to give a justified measure on how close we are replicating the brain's control power compared to our artificial implementation in specific circumstances.

do we
want
he
.Maybe
pike
t? And
can
o? Also
ences!

1.5 Benefits, Ethics and Sustainability

Describe who will benefit from the degree project, the ethical issues (what ethical problems can arise) and the sustainability aspects of the project.

Use references!

is even
to ask
from a
entional
Say in
od bcs
it has
dy been
en if I am
nistaken

1.6 Methodology

Introduce, theoretically, the methodologies and methods that can be used in a project and, then, select and introduce the methodologies and methods that are used in the

ds vague

degree project. Must be described on the level that is enough to understand the contents of the thesis.

Use references!

Preferably, the philosophical assumptions, research methods, and research approaches are presented here. Write quantitative / qualitative, deductive / inductive / abductive. Start with theory about methods, choose the methods that are used in the thesis and apply.

Detailed description of these methodologies and methods should be presented in Chapter 3. In chapter 3, the focus could be research strategies, data collection, data analysis, and quality assurance.

We build a SNN for a control problem and check it for performance as mentioned above. In addition we design a conventional controller and compare the result. IF we have the time for it we put a conventional NN to it too. We see the performance compared to the others and look at the specs we mentioned above. The SNN is trained by learning using STDP rule. We can compare the learned weights with the optimal weights when we have our own optimal controller/ we simulate our trajectory. For our approach we use a balanced spiking network.

1.7 Stakeholders

Present the stakeholders for the degree project.

1.8 Delimitations

Explain the delimitations. These are all the things that could affect the study if they were examined and included in the degree project. Use references!

1.9 Outline

In text, describe what is presented in Chapters 2 and forward. Exclude the first chapter and references as well as appendix.

Chapter 2

<Theoretical Background>

In this chapter, a detailed description about background of the degree project is presented together with related work. Discuss what is found useful and what is less useful. Use valid arguments.

Explain what and how prior work / prior research will be applied on or used in the degree project /work (described in this thesis). Explain why and what is not used in the degree project and give valid reasons for rejecting the work/research.

Use references!

2.1 Use headings to break the text

Do not use subtitles after each other without text in between the sections.

2.2 Related Work

You should probably keep a heading about the related work here even though the entire chapter basically only contains related work.

Here just what has been done for each of the headlines
Previous efforts were already made to control dynamic systems with SNNs.

List here also efforts with other concepts apart from Balanced Networks

Neural networks in general spiking neural networks and their differences and what they are better for. neuron models, iwazishi neuron and maybe one more mein neuron model und warum ich es ausgewaelt habe: einfach zu implementieren. Bereits fuer dynamische systeme verwendet, Nachteile dieses modells. Vlt vergleich mit einem anderen modell. Ganz kurzer ausflug in die regelung von dynamischen systemen.

What is a neural network? -> not here ref a paper. kurze erkl’rung in der einfuerung in der einfuhurng vlt auch hodgekin huxley erwaehen :)

2.3 Dynamic systems

2.4 Neuron model

2.4.1 Biological Neuron model

The most biologically accurate model of neuron spiking is the Hodgkin–Huxley (HH) model. The HH-model considers the neuron with its ion channels. The membrane acts as a capacitance and the travelling ions in each ion channel contribute a current to the overall membrane potential. These ion gates are voltage dependent and are defined positive in direction out of the cell.
A particular ion channel for ion X can be modelled as

$$I_X = g_X \cdot (V - V_X) \tag{2.1}$$

These currents are summed summed for the different ion channels in question, most commonly for Sodium, Potassium and a leak current. In reality there are a plethora of different channels and channel properties[1]. The V_X are the equilibrium potentials for each of the channels and can be computed using the Nernst equation [19].

Add a reference to monograph

$$C\frac{dV}{dt} = g_{Na} \cdot (V - V_{Na}) + g_K \cdot (V - V_K) + g_l \cdot (V - V_l) \tag{2.2}$$

Do model the voltage dependency of the ion channels, the conductances are described with gating variables, usually called n, h and g for Na-Activation, Na-Inactivation and K-activation respectively. One gating variable is set between $[0, 1]$ and models the permeability of said gate. Multiple gates are used to fit to each ion channel in order to match experimental data and the model behaviour.

[1]See `channelpedia.epfl.ch` for an extensive list

Gates have first order dynamics of the form

$$\frac{dn}{dt} = \alpha_n(1-n) - \beta_n n \tag{2.3}$$

for e.g the n gate. The other gates' dynamics are analogous. The functions α and β are voltage but not time dependent. The discussion of initial values as well as functions for $\alpha_p,\ \beta_p\ \ p = (n, h, m)$ can be found in [12] or [19]. The gates for each ion channel's conductance are found to be

$$\begin{aligned} g_{Na} &= \bar{g}_{Na} n^4 \\ g_K &= \bar{g}_K m^3 h \end{aligned} \tag{2.4}$$

and give form to the final model

$$\begin{aligned} C\frac{dV}{dt} &= I(t) - \bar{g}_{Na} n^4 (V - V_{Na}) - \bar{g}_K m^3 h (V - V_K) - g_L (V - V_L) \\ \frac{dn}{dt} &= (1-n)\alpha_n(V) - \beta_n n(V) \\ \frac{dm}{dt} &= (1-m)\alpha_m(V) - \beta_m m(V) \\ \frac{dh}{dt} &= (1-h)\alpha_h(V) - \beta_h h(V) \end{aligned} \tag{2.5}$$

We did not define a gate for the leak term as it is assumed constant.

2.4.2 "IF and LIF"

In contrast of the HH model in eq. (2.5), the simplest models of neurons are the Integrate and Fire (IF) and Leaky-integrate-and-fire (LIF) models.

IF Neurons IF Neurons, as the name implies, integrate the incoming current over time.

$$\frac{dV(t)}{dt} = \frac{1}{C} I(t) \tag{2.6}$$

The membrane voltage is governed by the incoming current spikes of connected neurons and the membrane capacitance. The neuron potential does not change without a change of input current and thus presents as a perfect integrator of the input.

LIF Neurons In contrast to that the LIF neuron contains a leak term on the RHS which brings the voltage back to its resting potential over time. The model can be expressed as

$$\tau \frac{dV(t)}{dt} = -(V(t) - E_r) + RI(t), \tag{2.7}$$

where $\tau = RC$ is the time constant the composed of the membrane resistance R and the membrane capacitance C and the resting potential E_r. In the absence of input $I(t)$ the voltage settles on the membrane potential E_r.
The input $I(t)$ encapsulates external inputs as well as a sum of Dirac functions indicating a spiking neuron

$$I(t) = \sum_k \delta(t - t^k) \tag{2.8}$$

and t_k being the time of the k-th spike. When the membrane voltage exceeds the threshold potential $\bar{v}$, a spike is sent out by the neuron and the voltage sets back to its reset voltage v_{res}.

This is not correct. Fo weights, bu the same ti only when are more th neuron

2.4.3 Izhikevich Neuron

While the above models deliver a useful and cheap simplification, they lack in accuracy. The Izhikevich model [17] of the neuron tries be the of both worlds in terms of efficiency and accuracy. It is comprised of 2D ODEs with the membrane potential v as

$$\begin{aligned} \frac{dv}{dt} &= 0.04v^2 + 5v + 140 - u + I(t) \\ \frac{du}{dt} &= a(bv - u). \end{aligned} \tag{2.9}$$

With the chosen factors, the neuron experiences a spike when $u \geq 30\text{mV}$, in which case the neuron resets to

$$\begin{aligned} u &\leftarrow u + d \\ v &\leftarrow c \end{aligned} \tag{2.10}$$

The parameters describe a scale of recovery, b sensitivity, c the reset potential of v and d the reset of variable u. Depending on these parameters one can achieve different behaviours of the neuron e.g. regular spiking, fast spiking and low threshold spiking to name a few [17].

Maybe shit explanation which coul extended or

2.5 Neural Networks

2.5.1 Biological Neural Network

2.5.2 Artificial Neural Networks

e clear
nction
een
ard nns
ann. Bcs
rently they
ot the
!

2.5.3 Spiking Neural Networks

A spiking Neural network is one step closer to a biologic representation of a brain. Instead of conveying information using a gradient in conventional NNs, information is propagated using discrete spikes of excitation, similar to biological neurons. Hereby one can distinguish between several ideas of implementation.

main
rence is
notion of

2.5.4 Poisson Networks

Poisson networks are built around the idea that information is encoded in the firing rate of a neuron. The precise timing of a spike is essentially meaningless[7]. This makes a strong contrast to the approach chosen in this paper, where every spike is timed exactly to minimize a cost function. The encoding of a value, e.g. four, is set by endowing the input neurons with a Poisson point process with a suitable encoding rate r_i[9].

ybe add
our
oach does
ule rate
ompletely.

One typically uses probabilistic stimuli because observations in the spiking of the human brain do are different in a trial by trial basis.
Input spikes are travelling through the recurrent network with weighted connections. The decoding is done by counting the spikes of output neurons over a certain time window. The time window plays a crucial rule in the decoding. If it is set smaller, spatio temporal patterns can be captured which can convey information about the input. Equally the sensitivity to noise becomes greater. If the time window is set to large, the firing patterns are lost though the impact of random spikes is reduced.

Theoretically using a lot of spikes is unproblematic and with large spike counts neuromorphic hardware is still much more energy efficient than deep neural networks if one seeks to deploy it.

e that
rror
s pretty
y with the
eroot
herefore
eed a lot
ikes to
ertain

The connection weights are subject to change over the learning/training period[2]. This approach however has some conceptual problems. Firstly, this approach has the problem that responses are limited by the time window the spikes are counted[3]. This means that the rate decoding is to slow to capture the fast travelling information and another needs to be another faster way to transmit information. However it has been shown that the firing rate does convey information about the stimuli's magnitude [1]. The second problem is that due to the Poisson process one needs more spikes to represent an average firing rate. An action potential consumes a lot of the cells energy[4], thus making it unfeasible to use a large number of spikes if one tries to model the brain. Evolution has found a better way to transmit information.

Although the above mentioned problems, rate encoded SNNs have seen interest by research. A big hurdle of deploying SNNs is the lack of performant learning algorithms. For this there efforts have been made to train recurrent or convolutional ANNs using backpropagation and afterwards convert the trained network to a SNN using rate encoding[11][10].

Maybe find a reference that the po consumptio is really lo even though use rate ba encoding

2.5.5 Liquid state machines

write how the offline computing is pretty bad for brain things, but good for chess for example. The online computing is what the brain does and it is not yet as developed. add the TU graz paper to refs and some of it's references too, for example nr. 19

Say that th CV(explain what it is :sigma/mu what it say value is shi you you ne a ton of sp Then say th it has some use and so perfomranc when you transmit fr an ann and move it to a snn with neuromoph

One alternative method has been the use of Liquid State Machine (LSM) or more general Reservoir computing.
The term Reservoir computing was introduced by Benjamin Schrauwen and describes a general group of recurrent network approach[27].
The "reservoir" is a non-linear map from input to outputs that combines the input in various, even random ways. These contain but are not limited to sums, differences, multiplications, division and exponentiation. In general the output $\mathbf{x}(t)$ is higher dimensional that the input $\mathbf{v}(t)$, in order to allow for sufficient variety in the mapping. The output of the reservoir, which is usually treated as a black box, is fed in a linear decoder in order to retrieve the desired output signal.
The liquid can be made of any system that fulfils two properties.

- Non-linear nodes of computation
- Fading memory

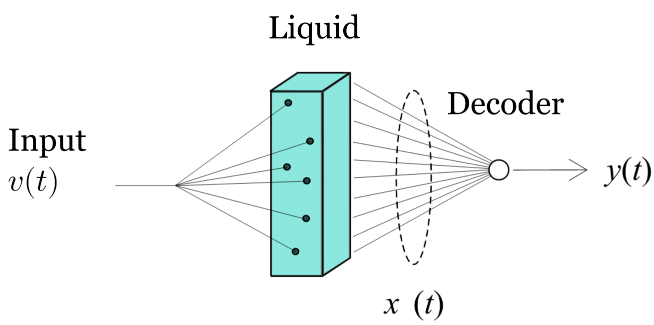

Figure 2.5.1: Abstract idea of Reservoir Computing. Adapted from [20]

To these points it is usually set for the system to be time invariant[20]. A reservoir can be a mathematical abstract formulation or physical object, e.g. a literal bucket of water [25].

After the choice of "liquid" in the reservoir is fixed, its dynamics are not altered. Only the linear decoder is trained to return the desired decoded output. This is a considerable time saver since the training of recurrent networks is expensive. On the contrary the linear decoder can be learned relatively cheaply.

A reservoir computer is called a LSM if one chooses a spiking neural network as the reservoir. The requirements mentioned above are fulfilled by the recurrent structure to retain information of the neurons and its non-linear spiking behaviour.

LSMs are capable of computing any dynamical system of any order of the form of

$$z^{(n)} = G(z, z^{(1)}, z^{(2)}, \dots, z^{(n-1)}) + u \tag{2.11}$$

given a sufficiently large liquid and a suitable feedback and decoder[23]. The systematic structure can be in fig. 2.5.2. The feedback $K(x, u)$ is a function of the dynamical system input $u(t)$ and the output $x(t)$. The result of $K(x, u)$ is fed back replaces the previous input $v(t)$ into the Liquid. The decoder $h(x)$ is not linear but can be simplified to be in a cost-performance trade-off when using a sufficiently large Liquid.

2.5.6 Balanced Networks

The idea of tightly balanced spiking networks was first proposed by Boerlin et al.[6]. It uses predictive coding in combination with spikes to simulate arbitrary linear

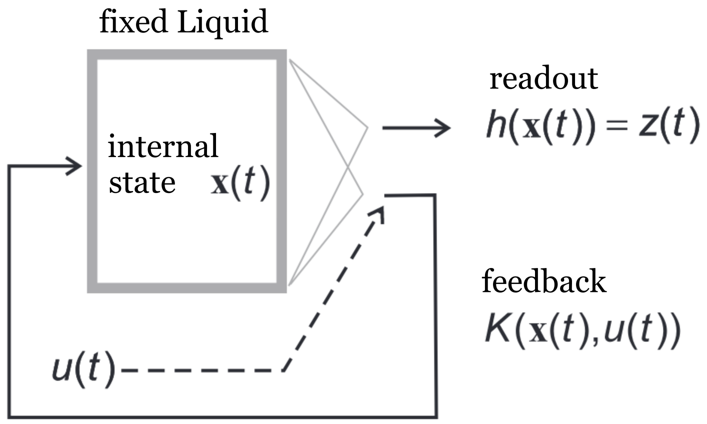

Figure 2.5.2: Adding suitable feedback allows LSMs to be universal approximator. Adapted from [22]

systems. The technical derivation will be described in chapter . The approach defines a cost function measuring the networks' accuracy in addition with regularization terms that moderates the spiking behaviour. Using a greedy algorithm this cost function determines the voltage threshold and therefore the neurons' spiking behaviour.

Add the ref the chapter

For each neuron voltage can be understood as a projection of the global system error to a local error.

Therefore the spiking rule is set to fire if the Voltage of a neuron becomes to large in order to reset the neuron voltage and reduce part of the system error.

Balanced networks differ from the previous rate encoding in that excitation and inhibition is closely tracked. In rate encoded networks both inhibitory and excitatory spikes are received by a single neuron. An change of the variable is then governed by which type dominates. Here a rate coding is also used, however in combination with instantaneous decoding[18].

The derivation of its behaviour is adopted from [6],[14] and [15].

Add some r general stuf here!

The derivation of the method used can be put to method or work/ here we can explain using only words and compare to the other methods and why we chose this one etc

The derivation of the balanced spiking network follows the derivation found in [6] and

[13]. The goal is to describe a dynamical system of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{c}(t) \tag{2.12}$$

with J state variables. The estimating is done by leaky integration of spiking trains $\mathbf{o}(t)$ in

$$\dot{\hat{\mathbf{x}}} = -\lambda_d \hat{\mathbf{x}} + \mathbf{\Gamma}\mathbf{o}(t). \tag{2.13}$$

$\mathbf{\Gamma}$ is a given Matrix of size $\mathbb{R}^{J\times N}$, N being the number of neurons, with the different connection weights between the neurons. This matrix is given as initial and can be optimized by training later on.
In addition to the estimate $\hat{\mathbf{x}}$ we define a spiking rate variable $\mathbf{r}$ following the dynamics of

$$\dot{\mathbf{r}} = -\lambda_d \mathbf{r} + \lambda_d \mathbf{o}(t). \tag{2.14}$$

The rate variable is connected to the state vector in the

The spikes are calculated by minimizing a cost function. A spike is fired if it minimizes the cost function that tracks the error between the true and estimated value over time

$$E(t) = \int_0^t \|\mathbf{x}(u) - \hat{\mathbf{x}}(u)\|_2^2 \, du. \tag{2.15}$$

The cost function integrates the error between the estimate and the real dynamic variable as well as regularization terms.

$$E(t) = \int_0^t \left(\|\mathbf{x}(u) - \hat{\mathbf{x}}(u)\|_2^2 + \nu\|\mathbf{r}(u)\|_1 + \mu\|\mathbf{r}(u)\|_2^2\right) du \tag{2.16}$$

These two regularization terms are added to discourage undesired behaviours.
The first was termed "ping-pong" effect and is described in the supplementary material of [6]. Tp understand the issue, we imagine a minimal network consisting of 2 neurons with equal kernel but opposite sign.

better
ing pong
! Maybe

The second regularization comes into play when there are kernels with different magnitude. Kernels with small kernel magnitude reach their threshold sooner and therefore fire more frequently. In the extreme case, only small number of neurons fire rapidly while the majority remains idle. By penalizing the rate in the 2-norm it forces the network to spread the firing among the whole network.

find the rig place to exp that!

The dynamic variable x is tracked by firing spikes in when the defined "pseudo voltage" of a neuron surpasses its threshold. The voltage for each neuron is defined by

$$V_i(t) = \mathbf{\Gamma}^T(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) - \mu\lambda_d r_i(t) \quad i = 1 \dots N. \tag{2.17}$$

For negligible quadratic cost μ the voltage can be understood as measure of the error projected on $\mathbf{\Gamma}_i$. The explicit derivation of the above equation is found in [6] and will be adapted . The voltage definition and the threshold definition

Where? He in the appe of at all?

$$T_i = \frac{\nu\lambda_d + \mu\lambda_d^2 + \|\mathbf{\Gamma}_i\|^2}{2} \tag{2.18}$$

result from integrating the cost function eq. (2.16) over time step ϵ. Then the condition described earlier fires a spike if the cost gets lowered. If there is no spike fired, the rate and estimated state variable in eq. (2.13) and eq. (2.14) respectively behave as

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= -\lambda_d \hat{\mathbf{x}} \\ \dot{\mathbf{r}} &= -\lambda_d \mathbf{r} \end{aligned} \tag{2.19}$$

and therefore decay exponentially with $e^{-\lambda_d t}$.
If a spike is fired, the inhomogeneous solution is found by variation of constants in eq. (2.14) to

$$\begin{aligned} r_i^h &= c_i(t)e^{-\lambda_d t} \\ c_i'(t)e^{-\lambda_d t} - c_i(t)\lambda_d e^{-\lambda_d t} &= -\lambda_d c_i(t)e^{-\lambda_d t} + \delta(t - t_i^k) \\ c_i'(t) &= \delta(t - t_i^k)e^{\lambda_d t} \\ c_i(t) &= e^{\lambda_d t_i^k}\boldsymbol{H}(t - t_i^k) \end{aligned} \tag{2.20}$$

where $\boldsymbol{H}(t)$ denotes the Heaviside step function. It can been seen that at the time of firing the spike adds a decaying exponential to the rate variable. Similarly it adds a column of the previously defined connection matrix $\boldsymbol{\Gamma}$ to the state vector. Thus we can now compare the effects on cost function eq. (2.16) and compare its impact. The integral is approximated by a greedy optimization method such that for very small time steps ϵ the exponential decays $e^{-\lambda_d t - t_i^k} \approx 1$. The greedy optimization is necessary since the unpredictable firing due to noise makes it impossible to predict future spikes.. After

explain notation of spike time constant wi i and k

Remember that i read somewhere that the no this step the rewriting the terms and using the defintions of the voltage and threshold we arrive at the critera to spike when

$$V_i > T_i \quad i = 1 \dots N \tag{2.21}$$

Neuron Voltage

As mentioned above, a neuron spikes if it meets the condition eq. (2.21). But so far we skipped over the dynamics how neuron voltage evolves over time. We start by defining the left pseudo-inverse of our output matrix $\mathbf{\Gamma}$

find a coherent name for the matrix

$$\mathbf{L} = \left(\mathbf{\Gamma}\mathbf{\Gamma}^T\right)^{-1} \mathbf{\Gamma} \tag{2.22}$$

such that $\mathbf{L}\mathbf{\Gamma}^T = \mathbf{I}$.
Next we take the derivative of eq. (2.17) and arrive at

$$\dot{\mathbf{V}}(t) = \mathbf{\Gamma}^T \left(\dot{\mathbf{x}}(t) - \dot{\hat{\mathbf{x}}}(t)\right) - \mu\lambda_d \dot{\mathbf{r}}(t). \tag{2.23}$$

We now use the pseudo-inverse to rewrite the voltage equation eq. (2.17) as

$$\begin{aligned} \mathbf{V}(t) &= \mathbf{\Gamma}^T (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) - \mu\lambda_d \mathbf{r}(t) \\ \mathbf{L}\mathbf{V}(t) &= (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) - \mu\lambda_d \mathbf{L}\mathbf{r}(t) \end{aligned} \tag{2.24}$$

We now replace the derivative terms in eq. (2.23) with their respective equations eq. (2.12) to eq. (2.14). Lastly we set

2.5.7 Balanced networks as a controller

Here describe the derivation of the controller we set out to design

2.5.8 Learning: SGD and STDP

Key to give any NN the ability to solve a task, it is integral to learn/train the network. The adaption of synapse weights is necessary to accomplish any functionality based on the underlying data[28]. There are various ways to train a network.

his
ence in
say its is
d partly

The most fundamental distinction can be made between supervised, unsupervised and reinforcement learning rules. One needs to remember that ANNs and SNNs require completely different learning algorithms because of their different transport of information.
For a review

Supervised Learning methods

Gradient based methods require differentiability and therefore continuity, thus are only applicable for ANNs.

Explain gradient methods. The derivative of the weights and biases is used for the derivative of the cost function. Efficient methods for building the derivative exists. With reference!

Unsupervised Learning methods

STDP

Reinforcement learning

Here explain the conpects for each of the NNs
Give references for the STDP variances

Chapter 3

<Engineering-related content, Methodologies and Methods>

Describe the engineering-related contents (preferably with models) and the research methodology and methods that are used in the degree project.

Most likely it generally describes the method used in each step to make sure that you can answer the research question.

3.1 Engineering-related and scientific content:

Applying engineering related and scientific skills; modelling, analysing, developing, and evaluating engineering-related and scientific content; correct choice of methods based on problem formulation; consciousness of aspects relating to society and ethics (if applicable).

As mentioned earlier, give a theoretical description of methodologies and methods and how these are applied in the degree project.

was ist meine research question?

zusammensetzung von den beiden systeme: dynamisches system und neuronales netz. mehr oder weniger die herleitung kopieren aus dem paper. Dann mit learning von den gewichten.

Here I describe what how it needs to be done. So this is the place for the derivation The concept and the process whatever that means Later there comes the how I implemented it. Here is what we needs to be implemented.

Here very detailed explanation of the Balanced network for this problem
Very detailed way for the regular NN for this problem Basics of the controller design used in this comparison aka LQG controller
Method of learning the weights for the SNN Method of comparison

Chapter 4

<The work>

Describe the degree project. What did you actually do? This is the practical description of how the method was applied.

4.1 Creating the SNN

How do we make the SNN MAtlab Balanced spiking network (say why to use that) maybe pseudo code Ideally some theorem (convergence???) Simulation? nein kommt in den naechsten part

4.2 Creating the NN

4.3 Creating the regular Controller

Chapter 5

<Result>

Describe the results of the degree project.

Chapter 6

<Conclusions>

Describe the conclusions (reflect on the whole introduction given in Chapter 1).

Discuss the positive effects and the drawbacks.

Describe the evaluation of the results of the degree project.

Describe valid future work.

The sections below are optional but could be added here.

6.1 Discussion

6.1.1 Future Work

6.1.2 Final Words

Todo list

If you are using mendeley to manage references, you might have to export them manually in the end as the automatic ways removes the ”date accessed” field

Bibliography

[1] Adrian, E. D. **and** Zotterman, Yngve. "The impulses produced by sensory nerve-endings: Part II. The response of a Single End-Organ". **in**: *The Journal of Physiology* 61.2 (23 **april** 1926), **pages** 151–171. ISSN: 00223751. DOI: 10.1113/jphysiol.1926.sp002281. URL: https://onlinelibrary.wiley.com/doi/10.1113/jphysiol.1926.sp002281 (**urlseen** 16/11/2022).

[2] Almomani, Ammar, Alauthman, Mohammad, Alweshah, Mohammed, Dorgham, O. **and** Albalas, Firas. "A comparative study on spiking neural network encoding schema: implemented with cloud computing". **in**: *Cluster Computing* 22.2 (**june** 2019), **pages** 419–433. ISSN: 1386-7857, 1573-7543. DOI: 10.1007/s10586-018-02891-0. URL: http://link.springer.com/10.1007/s10586-018-02891-0 (**urlseen** 15/11/2022).

[3] Andrew, Alex M. "Spiking Neuron Models: Single Neurons, Populations, Plasticity". **in**: *Kybernetes* 32.7 (1 **october** 2003). ISSN: 0368-492X. DOI: 10.1108/k.2003.06732gae.003. URL: https://www.emerald.com/insight/content/doi/10.1108/k.2003.06732gae.003/full/html (**urlseen** 15/11/2022).

[4] Attwell, David **and** Laughlin, Simon B. "An Energy Budget for Signaling in the Grey Matter of the Brain". **in**: *Journal of Cerebral Blood Flow & Metabolism* 21.10 (**october** 2001), **pages** 1133–1145. ISSN: 0271-678X, 1559-7016. DOI: 10.1097/00004647-200110000-00001. URL: http://journals.sagepub.com/doi/10.1097/00004647-200110000-00001 (**urlseen** 24/11/2022).

[5] Azevedo, Frederico A. C., Carvalho, Ludmila R. B., Grinberg, Lea T., Farfel, José Marcelo, Ferretti, Renata E. L., Leite, Renata E. P., Jacob Filho, Wilson, Lent, Roberto **and** Herculano-Houzel, Suzana. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate

brain". **in**: *The Journal of Comparative Neurology* 513.5 (10 **april** 2009), **pages** 532–541. ISSN: 1096-9861. DOI: `10.1002/cne.21974`.

[6] Boerlin, Martin, Machens, Christian K. **and** Denève, Sophie. "Predictive Coding of Dynamical Variables in Balanced Spiking Networks". **in**: *PLOS Computational Biology* 9.11 (14 **november** 2013). Publisher: Public Library of Science, e1003258. ISSN: 1553-7358. DOI: `10.1371/journal.pcbi.1003258`. URL: `https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003258` (**urlseen** 20/09/2022).

[7] Brette, Romain. "Philosophy of the Spike: Rate-Based vs. Spike-Based Theories of the Brain". **in**: *Frontiers in Systems Neuroscience* 9 (10 **november** 2015). ISSN: 1662-5137. DOI: `10.3389/fnsys.2015.00151`. URL: `http://journal.frontiersin.org/Article/10.3389/fnsys.2015.00151/abstract` (**urlseen** 16/11/2022).

[8] Clarke, D.D. **and** Sokoloff, L. "Circulation and energy metabolism of the brain". **in**: *Basic Neurochemistry: Molecular, Cellular, and Medical Aspects* (1999), **pages** 637–669.

[9] Denève, Sophie **and** Machens, Christian K. "Efficient codes and balanced networks". **in**: *Nature Neuroscience* 19.3 (**march** 2016), **pages** 375–382. ISSN: 1097-6256, 1546-1726. DOI: `10.1038/nn.4243`. URL: `http://www.nature.com/articles/nn.4243` (**urlseen** 18/10/2022).

[10] Diehl, Peter U., Neil, Daniel, Binas, Jonathan, Cook, Matthew, Liu, Shih-Chii **and** Pfeiffer, Michael. "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing". **in**: *2015 International Joint Conference on Neural Networks (IJCNN)*. 2015 International Joint Conference on Neural Networks (IJCNN). Killarney, Ireland: IEEE, **july** 2015, **pages** 1–8. ISBN: 978-1-4799-1960-4. DOI: `10.1109/IJCNN.2015.7280696`. URL: `http://ieeexplore.ieee.org/document/7280696/` (**urlseen** 17/11/2022).

[11] Diehl, Peter U., Zarrella, Guido, Cassidy, Andrew, Pedroni, Bruno U. **and** Neftci, Emre. "Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware". **in**: *2016 IEEE International Conference on Rebooting Computing (ICRC)*. 2016 IEEE International Conference on Rebooting Computing (ICRC). San Diego, CA, USA: IEEE, **october** 2016, **pages** 1–8. ISBN: 978-1-5090-1370-8. DOI: `10.`

1109/ICRC.2016.7738691. URL: http://ieeexplore.ieee.org/document/7738691/ (**urlseen** 17/11/2022).

[12] Hodgkin, A. L. **and** Huxley, A. F. "A quantitative description of membrane current and its application to conduction and excitation in nerve". **in**: *The Journal of Physiology* 117.4 (1952). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1952.sp004764, **pages** 500–544. ISSN: 1469-7793. DOI: 10.1113/jphysiol.1952.sp004764. URL: https://onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1952.sp004764 (**urlseen** 21/09/2022).

[13] Huang, Fuqiang. "Dynamics and Control in Spiking Neural Networks". **in**: (15 **december** 2019). Publisher: Washington University in St. Louis. DOI: 10.7936/YA3F-RK28. URL: https://openscholarship.wustl.edu/eng_etds/495 (**urlseen** 14/10/2022).

[14] Huang, Fuqiang **and** Ching, ShiNung. "Spiking networks as efficient distributed controllers". **in**: *Biological Cybernetics* 113.1 (**april** 2019), **pages** 179–190. ISSN: 0340-1200, 1432-0770. DOI: 10.1007/s00422-018-0769-7. URL: http://link.springer.com/10.1007/s00422-018-0769-7 (**urlseen** 23/10/2022).

[15] Huang, Fuqiang, Riehl, James **and** Ching, ShiNung. "Optimizing the dynamics of spiking networks for decoding and control". **in**: *2017 American Control Conference (ACC)*. 2017 American Control Conference (ACC). ISSN: 2378-5861. **may** 2017, **pages** 2792–2798. DOI: 10.23919/ACC.2017.7963374.

[16] Indiveri, Giacomo **and** Sandamirskaya, Yulia. "The Importance of Space and Time for Signal Processing in Neuromorphic Agents: The Challenge of Developing Low-Power, Autonomous Agents That Interact With the Environment". **in**: *IEEE Signal Processing Magazine* 36.6 (**november** 2019), **pages** 16–28. ISSN: 1053-5888, 1558-0792. DOI: 10.1109/MSP.2019.2928376. URL: https://ieeexplore.ieee.org/document/8887553/ (**urlseen** 09/12/2022).

[17] Izhikevich, E.M. "Simple model of spiking neurons". **in**: *IEEE Transactions on Neural Networks* 14.6 (**november** 2003). Conference Name: IEEE Transactions on Neural Networks, **pages** 1569–1572. ISSN: 1941-0093. DOI: 10.1109/TNN.2003.820440.

[18] Johnson, Erik C., Jones, Douglas L. **and** Ratnam, Rama. “A minimum-error, energy-constrained neural code is an instantaneous-rate code”. **in**: *Journal of Computational Neuroscience* 40.2 (**april** 2016), **pages** 193–206. ISSN: 0929-5313, 1573-6873. DOI: 10.1007/s10827-016-0592-x. URL: http://link.springer.com/10.1007/s10827-016-0592-x (**urlseen** 24/11/2022).

[19] Johnston, Daniel **and** Wu, Samuel Miao-sin. *Foundations of cellular neurophysiology*. Cambridge, Mass: MIT Press, 1995. 676 **pagetotals**. ISBN: 978-0-262-10053-3.

[20] Maass, Wolfgang. “Liquid State Machines: Motivation, Theory, and Applications”. **in**: Cooper, S Barry **and** Sorbi, Andrea. *Computability in Context*. IMPERIAL COLLEGE PRESS, **february** 2011, **pages** 275–296. ISBN: 978-1-84816-245-7 978-1-84816-277-8. DOI: 10.1142/9781848162778_0008. URL: http://www.worldscientific.com/doi/abs/10.1142/9781848162778_0008 (**urlseen** 31/10/2022).

[21] Maass, Wolfgang. “Networks of spiking neurons: The third generation of neural network models”. **in**: *Neural Networks* 10.9 (**december** 1997), **pages** 1659–1671. ISSN: 08936080. DOI: 10.1016/S0893-6080(97)00011-7. URL: https://linkinghub.elsevier.com/retrieve/pii/S0893608097000117 (**urlseen** 09/12/2022).

[22] Maass, Wolfgang, Joshi, Prashant **and** Sontag, Eduardo D. “Computational Aspects of Feedback in Neural Circuits”. **in**: *PLoS Computational Biology* 3.1 (19 **january** 2007). **byeditor**Rolf Kotter, e165. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.0020165. URL: https://dx.plos.org/10.1371/journal.pcbi.0020165 (**urlseen** 07/11/2022).

[23] Maass, Wolfgang **and** Markram, Henry. “On the computational power of circuits of spiking neurons”. **in**: *Journal of Computer and System Sciences* 69.4 (**december** 2004), **pages** 593–616. ISSN: 00220000. DOI: 10.1016/j.jcss.2004.04.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0022000004000406 (**urlseen** 07/11/2022).

[24] Patel, Jigneshkumar **and** Goyal, Ramesh. “Applications of Artificial Neural Networks in Medical Science”. **in**: *Current Clinical Pharmacology* 2.3 (1 **september** 2007), **pages** 217–226. ISSN: 15748847. DOI: 10.2174/157488407781668811. URL: http://www.eurekaselect.com/openurl/

content.php?genre=article&issn=1574-8847&volume=2&issue=3&spage=217 (**urlseen** 02/12/2022).

[25] Tanaka, Gouhei, Yamane, Toshiyuki, Héroux, Jean Benoit, Nakane, Ryosho, Kanazawa, Naoki, Takeda, Seiji, Numata, Hidetoshi, Nakano, Daiju **and** Hirose, Akira. “Recent advances in physical reservoir computing: A review”. **in**: *Neural Networks* 115 (**july** 2019), **pages** 100–123. ISSN: 08936080. DOI: 10.1016/j.neunet.2019.03.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S0893608019300784 (**urlseen** 29/10/2022).

[26] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz **and** Polosukhin, Illia. *Attention Is All You Need*. 5 **december** 2017. arXiv: 1706.03762[cs]. URL: http://arxiv.org/abs/1706.03762 (**urlseen** 02/12/2022).

[27] Verstraeten, D., Schrauwen, B., D’Haene, M. **and** Stroobandt, D. “An experimental unification of reservoir computing methods”. **in**: *Neural Networks* 20.3 (**april** 2007), **pages** 391–403. ISSN: 08936080. DOI: 10.1016/j.neunet.2007.04.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S089360800700038X (**urlseen** 07/11/2022).

[28] Zheng, Shengjie, Qian, Lang, Li, Pingsheng, He, Chenggang, Qin, Xiaoqin **and** Li, Xiaojian. “An Introductory Review of Spiking Neural Network and Artificial Neural Network: From Biological Intelligence to Artificial Intelligence”. **in**: *arXiv:2204.07519 [cs]* (9 **april** 2022). arXiv: 2204.07519. URL: http://arxiv.org/abs/2204.07519 (**urlseen** 20/09/2022).

Appendix - Contents

Appendix A

First Appendix

This is only slightly related to the rest of the report

Appendix B

Second Appendix

this is the information