Attractor Dynamics and Embodiment of Neural Computing

Yulia Sandamirskaya
Institute of Neuroinformatics (INI)
University of Zurich and ETH Zurich
Switzerland

@NICE 2019
What do brains “compute”? 

- biological neural systems evolved to generate **movement**
- **goal-directed movement** requires
  - perception (state estimation)
  - calibration (internal and external alignment)
  - online adaptation (control)

⇒ biological neural networks are intelligent **controllers**
What is required for intelligent control?

- working memory
  - stabilisation of neuronal states
- decision making
  - selection among alternatives
  - “attention”

Not something artificial neural networks solve today!

What is required to enable “purely” neuronal computing / control?

- structure (autonomy)
- interfaces to sensors and motors (embodiment)
Towards neuromorphic SLAM

Simultaneous localisation and mapping

Angular velocity sensing → Estimate Heading direction

Translational velocity sensing → Estimate Position

Landmark sensing → Form memory (map)
Neuromorphic hardware

**ROLLS**
- mixed-signal circuitry
- 256 artificial neurons
- 256 x 256 plastic synapse circuits
- 180nm process
- ultra-low power

(Qiao, Indiveri, 2015)

**DYNAP**
- mixed-signal circuitry
- 4 x 1024 artificial neurons
- 64 x 1024 synapses
- 180nm process
- ultra-low power
- scalable

(Qiao, Indiveri, 2018)

**Loihi**
- fully digital circuitry
- 130,000 artificial neurons
- 130 million synapses
- programmable learning engine
- 14nm process
- low power and scalable

(Davies, 2018)
Reconfigurable OnLine Learning Spiking (ROLLS)

- analog circuits for neurons and synapses
- digital communication of spikes

⇒ “programming” = wiring-up and setting parameters

Qiao et al, 2015
Heading direction network
Heading direction network

Heading Direction

Graph showing a network with a highlighted node.
Heading direction network

- (soft) “winner take all”
- dynamic neural field

- Population activity dynamics:

\[
\tau \dot{u}(x, t) = -u(x, t) + h + \int f(u(x', t))\omega(x - x')dx' + I(x, t)
\]

Amari, S. *Dynamics of pattern formation in lateral-inhibition type neural fields*. Biological Cybernetics, 1977, 27, 77-87


Gerstner, Grossberg, Ermentrout, Coombes, Schöner&Spencer, 2015, Erlhagen…
Heading direction network

Heading Direction

[Diagram of a network with nodes and arrows indicating direction]
Heading direction network

Heading Direction

$AV_{ccw}$

$AV_{cw}$
Heading direction network

Heading Direction

$AV_{ccw}$  

$AV_{cw}$
Heading direction network

- AV
- CW
- AV
- CCW

Analogue

- mismatch
- variability
- low precision
More robust connectivity: desinhibition
Heading direction estimation: hardware results

**Spiking activity on ROLLS**

![Graph showing spiking activity over time.]

**Error accumulation**

![Diagram illustrating error accumulation and correction.]

**Correction using vision**

![Diagram showing correction process using vision.]

**Reset**
Real-time activity on the ROLLS chip

“Loop closure” and calibration

• How fast does the activity bump need to move?

• Error estimation circuit
  • Computing differences
“Loop closure” and calibration

Plastic synapses

Angular velocities: CW, CCW
Internally sensed movement

True orientation

Error detection

Shift faster
Excitatory speed correcting neurons

Shift slower
Inhibitory speed correcting neurons

Plastic synapses

Hebbian learning
Anti-Hebbian learning

Internally sensed movement

Plastic synapses
Matching activity shifting velocity to real velocity

Learning to shift faster

True orientation

Error correction

Learning to shift slower

True orientation

Error correction

Positive error spikes

Negative error spikes

Excitatory pool

Inhibitory pool
Position estimation

Heading direction estimation

Position estimation network
Position estimation

Head direction estimation

“Wall” Feature detection

Directional neurons

N  NW  W  SW  S  SE  E  NE

Position neurons

North  North-West  West  South-West  South  South-East  East  North-East

Integrate Position neurons

Position synapse from IMU

Position
Map formation

Map (LTM) formation

"Wall" Feature detection

Heading direction estimation
Map formation on the ROLLS chip

**Diagram:**
- Arena
- Omnibot
- Collision event
- Position population
- Collision population
- Neurons
- Plastic synapses

**Graphs:**
- ROLLS events in response to robot path integration data
- Robot's trajectory

**Axes:**
- X Position
- Y Position
- Time [ms]
- Neuron number
Learning different maps

Unlearning a map

Kreiser, R.; Pienroj, P.; Renner, A. & Sandamirskaya, Y. Pose Estimation and Map Formation with Spiking Neural Networks: towards Neuromorphic SLAM. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2018
How can we unlearn something?

- LTD and LTP depends on both pre- and postsynaptic frequencies

\[ w_i = w_i + \Delta w^+ \text{ if } V_{mem}(t_{pre}) > \theta_{mem} \text{ and } \theta_1 < C_a(t_{pre}) < \theta_3 \]

\[ w_i = w_i - \Delta w^- \text{ if } V_{mem}(t_{pre}) < \theta_{mem} \text{ and } \theta_1 < C_a(t_{pre}) < \theta_2 \]

\[ \frac{d}{dt} w_i = +C_{drift} \text{ if } w_i > \theta_w \text{ and } w_i < w_{max} \]

\[ \frac{d}{dt} w_i = -C_{drift} \text{ if } w_i < \theta_w \text{ and } w_i > w_{min} \]

\[ J_i = J_{max} f(w_i, q_i) \]

(Brader, Senn, and Fusi, 2007)
Unlearning false collisions on chip

Ground truth recording

With obstacle

Without obstacle

Learned collision maps

Trajectory recorded from firing activity

Robot arena with obstacle
Simultaneous **localisation and mapping**: path integration, learning a map
- state estimation, building representations

**Braitenberg vehicle, sequences**
- attractors in a sensory-motor loop

**Reference frame** transformations
- key for linking modalities

**Adaptive motor control**
- key element for adaptive behavior
Conclusions

➡ lots of structure is needed to control behavior with neurons
  - represent state with neuronal populations ("place code")
  - stabilise states and decision with recurrent connections (WTA)
  - disinhibition for robustness
  - adaptive couplings between sensed quantities and states
  - error estimation and correction

➡ learning can then be very simple
  - one-shot
  - binary weight

➡ object representation as a map-formation problem, not a (just) pattern recognition
Thanks!

• Marie Curie IF
• FET PROACT

• Ambizione
• Project coordination

• Forschungskredit
• GRC Grant

• Junior Group fellowship

MSc, BSc theses
- Gwendolyn English
- Eloy Barrero
- Llewyn Salt
- Mathis Richter
- Tobias Storck
- Christian Bell
- Claudia Rudolph
- Jianlin Lu
- Ammar Bitar
- Jonathan Müller
- Kay Müller
- Sebastian Glatz
- Valery Metry
- Alpha Renner
- David Niederberger
- Raphaela Kreiser

Semester theses
- Alexander Dietmüller
- Mario Blatter
- Frédéric Debraine
- Lukas Blässig
- Lennard de Graf
- Michel Frising
- Zahra Farsijani
- Michael Purcell
- Viviane Yang
- Davide Plozza
- Damiano Steger
- Héctor Vazquez
- Sebastian Glatz
- Herman Blum
- Matteo Cartiglia
- Lin Jin
- David Niederberger
- Nicolas Känzig
- Panin Pienroj
- Paul Joseph
- Nuria Armengol
- Jozef Bucko
- Balduim Dettling

Collaborators
- Gregor Schöner
- Giacomo Indiveri
- Florentin Wörgötter
- J.-C. Quinton
- John Spencer
- Piotr Dudek
- Fatih Yanik
- Jörg Conradt
- Christian Faubel
- Tobi Delbruck
- John Lipinski
- Matt Luciw
- Jürgen Schmidhuber
- Helge Ritter
- Hajar Azgari
- Abhishek Banerjee
Elementary module of neuronal control

“Reaching” task

⇒ Population activity dynamics:

\[
\tau \dot{u}(x, t) = -u(x, t) + h + \int f(u(x', t))\omega(x - x')dx' + I(x, t)
\]

Amari, S. Dynamics of pattern formation in lateral-inhibition type neural fields. Biological Cybernetics, 1977, 27, 77-87


Gerstner, Grossberg, Ermentrout, Coombes, Schöner&Spencer, Erlhagen…
Neural dynamics

Dynamic Neural Field, WTA, bump-attractor networks

\[ \tau \dot{u}(x,t) = -u(x,t) + h + \int f(u(x',t)) \omega(x - x') dx' + I(x,t) \]
“Cognitive” properties of Neural Fields

- “Detection” and “forgetting” instabilities
  - continuous time $\rightarrow$ discrete “events”

- Localised “bumps”
  - continuous space $\rightarrow$ discrete “categories”

- “Selection” instability
  - stabilisation of selection decisions

- Sustained activation
  - modelling working memory

$\Rightarrow$ DNF “Architectures”
Embodied DNF architectures

**Action selection**

**Learning to look**

**Sequence learning**

**Haptic learning**

Sandamirskaya, 2010-2015
Why are these architectures fundamental?

- Braintenberg vehicle, sequences
  - attractors in a sensory-motor loop
  Milde et al 2017a,b; Kreiser et al 2018

- Reference frame transformations
  - key for linking modalities
  Blum et al 2017

- Simultaneous localisation and mapping: path integration, learning a map
  - state estimation, building representations
  Blatter et al, ISCAS, under rev; Kreiser et al 2018a, b

- Adaptive motor control
  - key element for adaptive behavior
  Glatz et al, arxiv, 2018
Reference frames transformation on chip

View-based target representation:

- target in view

- target lost from view

Allocentric target representation:

Neural ref. frame transformation:

Blum, H.; Dietmüller, A.; Milde, M.; Conradt, J.; Indiveri, G. & Sandamirskaya, Y. A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor. Robotics: Science and Systems (RSS), 2017
What will be the bottleneck for purely neuronal control?

- interfaces
  - sensors
    - neuromorphic SLAM as an approach to perception
  - motor system
    - adaptive control
Motor control: results

Obstacle avoidance and target acquisition

Obstacle avoidance

DVS frame

Rolls

Constant excitation

Turn right

Drive right spike count

Drive left spike count

Obstacles

GYRO

Parallella

Speed

Connectivity

Braitenberg vehicle

Target acquisition

DVS frame

Rolls

Constant excitation

Target input

Target DNF

Speed

Drive right

Drive left

Turn right

Turn left

EDVS

Gyro

Target DNF

OR OL DL DR Sp. Exc.

Presynaptic neural groups

Weight W1-

Weight W2-

Weight W3-

Weight W4+

Weight W3+

Weight W2+

Weight W1+

No connection

ROLLS

Host PC

Robot Listener

Robot

OmniRobot

PushBot

USBConnector.h

TCPConnector.h

RobotListener.h

Robot.h

PushBot.h

OmniRobot.h

Robot control

interface

robot listener

robot connectivity

for a list of

DNF

ON

OL

DL

DR

Sp.

Exc.

over

Weight W2+

Weight W3+

Weight W4+

Weight W1-

Weight W2-

Weight W3-

No connection

ROLLS

speed

right turn

left turn

edvs

Gyro

OR OL DL DR Sp. Exc.

Presynaptic neural groups

Grammar: The Robot was controlled by sending the necessary instructions over the serial connection from the Host PC to the Parallella. The two robotic platforms currently supported by NS are the PushBot and the OmniBot. The PushBot has a similar structure and can also be easily accessed with the built-in WiFi module mounted on the robot as well. More robust and the small size, weight, and power consumption of the Parallella allow it to be used in compact robotic systems.

Linux is favored to directly log the mapped keyboard inputs (see also Figure 13). However, logging the robot parameters is currently disabled as it can trigger an emergency stop of the robot until the next drive command is sent. At the moment the robot is polled in an interval of 1 s, future updates of the OmniBot firmware will be required to change this behavior.

These states are logged as formatted text or processed immediately, such as the bumper states, when the bumper has been hit, making use again of the observer pattern as can be seen in Figure 14. Similarly to the DVS camera, different properties of the OmniBot are monitored asynchronously by the RobotListener object, such as the actual servo states, if and which bumper has been hit, etc.

Table 1: Important instructions for the robot control interface as the OmniRob and can also be easily accessed with the built-in WiFi module.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive Right</td>
<td>Moves the robot to the right</td>
</tr>
<tr>
<td>Drive Left</td>
<td>Moves the robot to the left</td>
</tr>
<tr>
<td>Turn Right</td>
<td>Makes the robot turn to the right</td>
</tr>
<tr>
<td>Turn Left</td>
<td>Makes the robot turn to the left</td>
</tr>
<tr>
<td>Stop</td>
<td>Stops the robot</td>
</tr>
</tbody>
</table>

The second supported robot is the PushBot, also provided by NS. The PushBot has a similar structure and can also be easily accessed with the built-in WiFi module mounted on the robot as well. More robust and the small size, weight, and power consumption of the Parallella allow it to be used in compact robotic systems.

Figure 13: The two robotic platforms currently supported by NS are the PushBot and the OmniBot (see also Figure 12) for more information.

Figure 14: Similar to the DVS camera, different properties of the OmniBot are monitored asynchronously by the RobotListener object, such as the actual servo states, if and which bumper has been hit, etc.
Sequence learning “program”

Neuronal mechanisms: Braitenberg Vehicle

Figures I.1 and I.2 illustrate the components of a Braitenberg vehicle. The vehicle consists of a sensory system, a nervous system, a body, and a motor system. The sensory system captures input from the environment, and the nervous system processes this input. The motor system then generates movements based on the nervous system's output.

The figure on the right shows two different behaviors (2a and 2b) of the vehicle in response to stimuli. The behavior of the vehicle is influenced by the interaction between the sensory and motor systems and the structured environment in which it operates.

Different behaviours:
Mathematical formalisation: attractor dynamics

- “behavioral variable”
  - describes the behavior
- its rate of change:
  \[ \tau \dot{\phi}(t) = -\phi(t) + A(t) \]
  - determines its dynamics
- overt behavior corresponds to attractors
  - stability
Chapter 1 begins building this dynamical systems view with an overview of neural dynamics. We will see that to describe real nervous systems, we must move beyond the simple feed-forward picture captured by Braitenberg’s vehicle. Instead, we will use closed loops that take place entirely within the nervous system to create internal attractor states—neural patterns that make decisions, select one input over another, and keep those decisions active even when the input is removed (see right side of Figure I.4).

In Chapter 2, we ask how such neural activation variables come about. The Braitenberg picture suggests that “neurons” must be intricately connected to the sensory surface and the motor surface. In simple vehicles, those surfaces are sampled by a small number of sensor or motor cells, but in real organisms, the sampling is so dense that we can describe these “surfaces” in terms of continuous spaces that are continuously coupled to the nervous system. Dynamic fields are the result—dynamical systems that reflect distributions of activation over appropriate feature spaces, including physical space. This enables the nervous system to know where a stimulus is located in space and to identify its particular features (e.g., color, shape, and so on).

In Chapter 3, we review the neural foundations of dynamic fields. We show that populations of neurons in cortex and many subcortical functions can be thought of using the concept of neural activation fields. In fact, it will turn out that real neurons in the brain operate as if they are smeared out over activation fields.

Finally, in Chapter 4, we come back to behavioral dynamics. We show how behavioral and neural dynamics can be combined within dynamic field theory, linking perception, action, and cognition. We demonstrate how this link enables embodied cognition by implementing a behavioral and neural dynamics on a robotic vehicle that orients toward targets, which it detects, selects, and keeps in working memory.

Turning rate of vehicle

Heading direction

Activation field

Heading direction

Multiple targets

• represent “utility” of options

• stabilise decisions

\[ \dot{\phi}(t) \rightarrow \dot{u}(\phi, t) \]

“activation” and its dynamics

Neural dynamics
Neuronal correlate of behavior: population activity

“Reaching” task

⇒ “Dynamic neural field” model

\[ \tau \dot{u}(x, t) = -u(x, t) + h + \int f(u(x', t)) \omega(x - x') \, dx' + I(x, t) \]

Amari, S. *Dynamics of pattern formation in lateral-inhibition type neural fields*. Biological Cybernetics, 1977, 27, 77-87


Gerstner, Grossberg, Ermentrout, Coombes, Schöner & Spencer, Erlhagen…
Dynamic Neural Fields explain behavior

- **Habit formation**
  - AnotB task

- **Visual working memory**

- **Spatial working memory**

Dineva and Schöner, “How infants’ reaches reveal principles of sensorimotor decision making”, Connection Science, 2018

“Implementation issue”

Neuronal dynamics

\[ \tau \dot{u}(x, t) = -u(x, t) + h + \int f(u(x', t)) \omega(x - x') dx' + I(x, t) \]

“Von Neumann” computer

- analogue values
- parallel processing
- memory and computation interlinked

- digital representations
- sequential processing
- separate memory unit

Match

Input Image

1 to N1 convolutions
N1/M1 to N2 convolutions
N2/M2 to N3 convolutions

output classifier
MLP

output class map

Memory

Control Unit

Arithmetic Logic Unit

Input

Output

Giacomo Indiveri
Neuromorphic processors

Inherently synchronized with the real-world “natural” events.

To process “natural” sensory signals efficiently (low bandwidth/power).

No virtual time (time represents itself).

Input/data driven computation.

Slow silicon

Paradigm shift

Radically different from von Neumann architectures.

Parallel elements with memory and computation co-localized.

Von Neumann computer

Neuronal dynamics

\[ \tau \dot{u}(x, t) = -u(x, t) + h + \int f(u(x', t)) \omega(x - x') dx' + I(x, t) \]
Brain-inspired computing or sensing devices that emulate activity of biological neurons and synapses

**Neuromorphic Hardware**

- "BrainDrop" (Stanford)
- DYNAP (Zurich)
- BrainScaleS (Heidelberg)
- "TrueNorth" (IBM)
- Loihi (Intel)
- SpiNNaker (Manchester)

Create and promote neuromorphic community in Europe: [www.neurotechai.eu](http://www.neurotechai.eu)