

# *cedar*

a framework for cognition, embodiment,  
dynamics, and autonomy in robotics

Oliver Lomp, Stephan K. U. Zibner

# What is *cedar*?

- Software for neural dynamics in robotics
- Open source
- Implemented in C++
- For Linux, Mac OS, Windows (in that order)
- Usage: code, GUI, configuration files



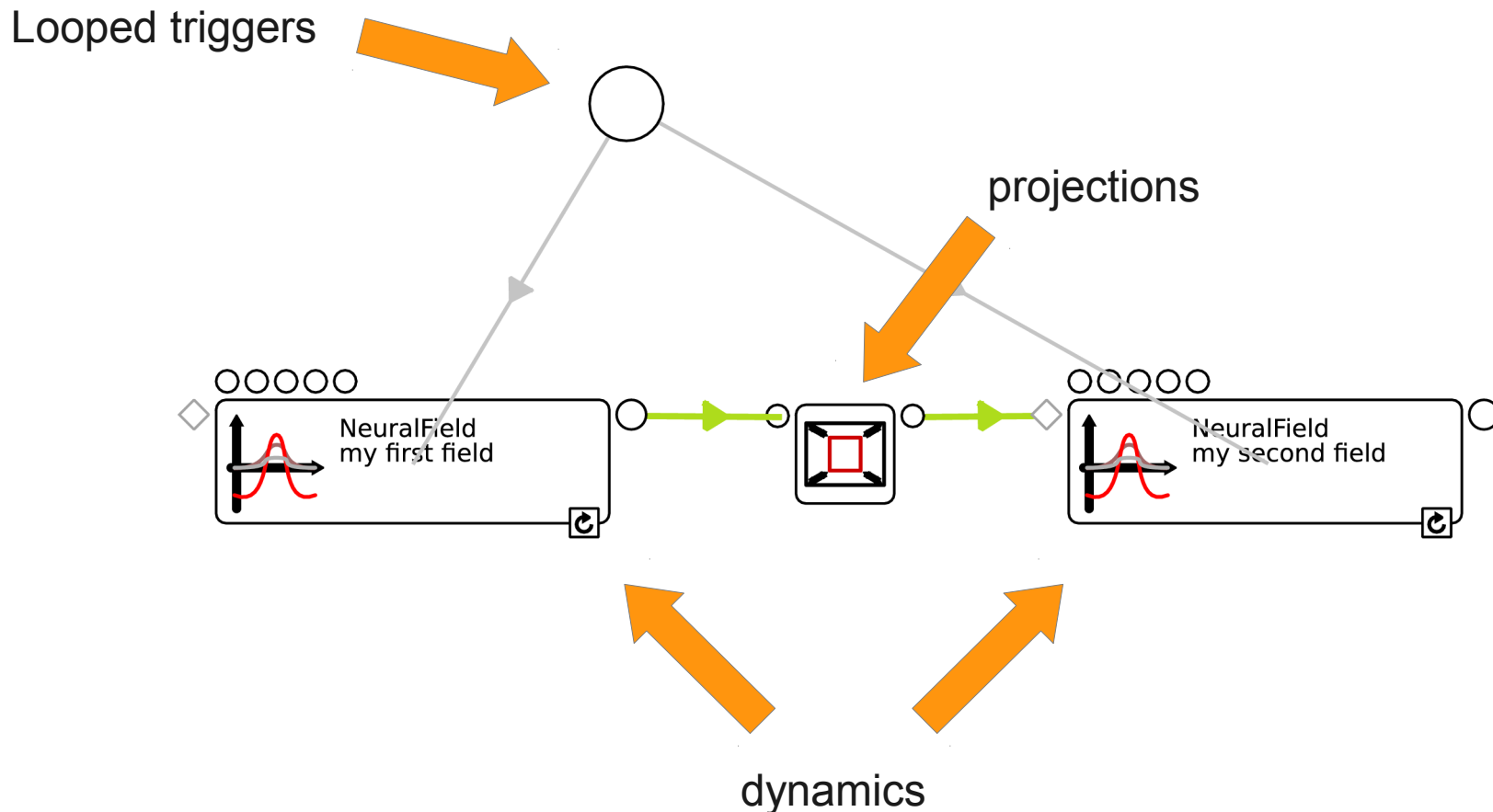
# *cedar* and DFT

- provides a set of essential elements for building architectures
- elements are dynamical systems or projections between them
- elements can be connected, parametrized, simulated, plotted, recorded etc.



# Concepts of architectures

- Architectures consist of the following elements

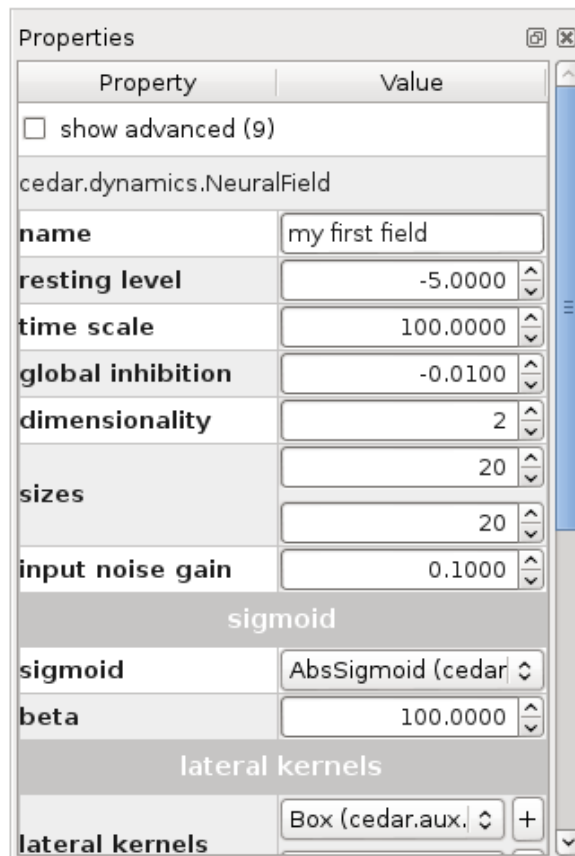


# Concepts of architectures

- Dynamics in architectures are iteratively solved using the forward Euler method
- Connecting a looped trigger to dynamics sends a periodical update signal and a measured time step
- Once the computation of an Euler step is done, subsequent projections are updated as well

# Concepts of architectures

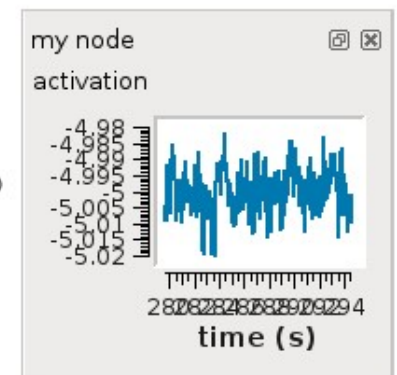
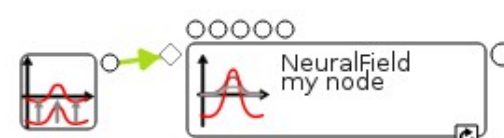
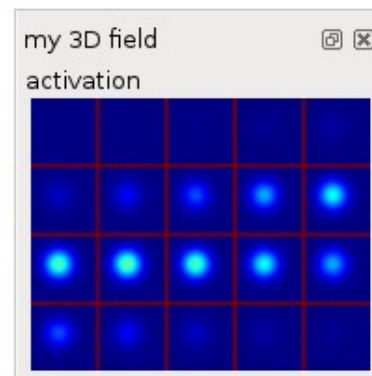
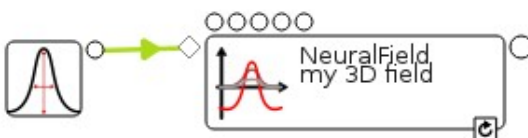
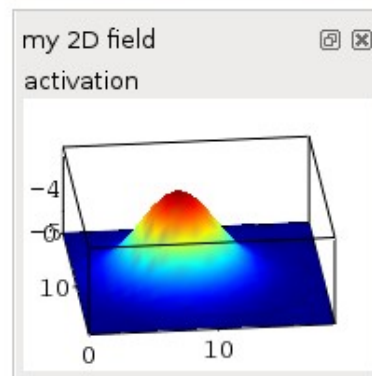
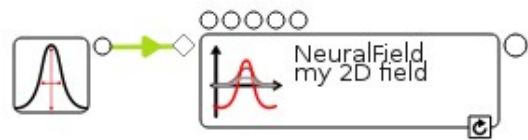
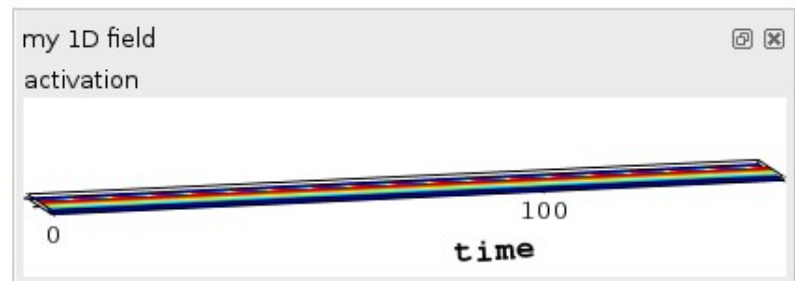
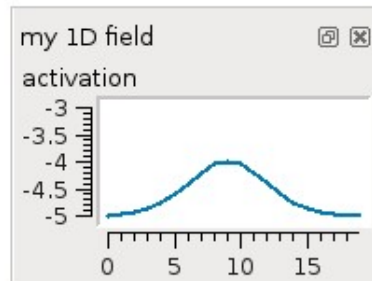
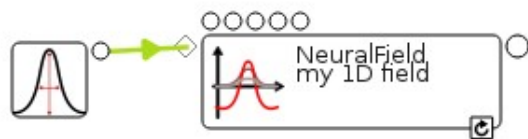
- Parameters of all elements can be altered, saved, and loaded in both the GUI and plain C++ code.



```
architecture.json
"steps":
{
  "cedar.dynamics.NeuralField":
  {
    "name": "my first field",
    "threaded": "false",
    "resting level": "-5",
    "time scale": "100",
    "global inhibition": "-0.01",
    "activation as output": "false",
    "discrete metric (workaround)": "false",
    "dimensionality": "2",
    "sizes":
    [
      "10",
      "10"
    ],
    "input noise gain": "0.1",
    "sigmoid":
    {
      "type": "cedar.aux.math.AbsSigmoid",
      "threshold": "0",
      "beta": "100"
    },
    "lateral kernels":
    {
      "cedar.aux.kernel.Gauss":
      {
        "dimensionality": "2",
```

# Concepts of architectures

- Populations can be plotted in multiple ways.



# More useful tools

- *cedar* supports a variety of inputs, spanning artificial inputs (boosts, noise), sensor input (images, cameras, videos), proprioception (joint angles, encoders counting wheel convolutions)
- *cedar* can be connected to motor surfaces (robot arms, hands, e-pucks ...)
- *cedar* can be used on multiple connected machines using matrix writers and readers
- *cedar* can be easily extended through plugins (more elements, data, plots ...) without recompiling the core software



**Live demonstrations!**

# cedar resources

- [cedar.ini.rub.de](http://cedar.ini.rub.de)
- [bitbucket.org/cedar](http://bitbucket.org/cedar)
- Paper at ICANN 2013  
this week (Lomp et al.)
- Developed for EU  
project NeuralDynamics  
[www.neuraldynamics.eu](http://www.neuraldynamics.eu)

## A Software Framework for Cognition, Embodiment, Dynamics, and Autonomy in Robotics: *cedar*

Oliver Lomp,<sup>1</sup> Stephan K. U. Zibner,<sup>1</sup> Mathis Richter,<sup>1</sup>  
Iñaki Rañó,<sup>2</sup> and Gregor Schöner<sup>1\*</sup>

<sup>1</sup> Institut für Neuroinformatik, Ruhr-Universität Bochum  
Universitätsstr. 150, 44780 Bochum, Germany  
{oliver.lomp, stephan.zibner, mathis.richter,  
gregor.schoener}@ini.rub.de  
<http://www.ini.rub.de>

<sup>2</sup> Intelligent Systems Research Centre, University of Ulster  
Northland Road, Derry, Northern Ireland BT48 J7L, UK  
[i.rano@ulster.ac.uk](mailto:i.rano@ulster.ac.uk)  
<http://isrc.ulster.ac.uk>

**Abstract.** We present *cedar*, a software framework for the implementation and simulation of embodied cognitive models based on Dynamic Field Theory (DFT). DFT is a neurally inspired theoretical framework that integrates perception, action, and cognition. *cedar* captures the power of DFT in software by facilitating the process of software development for embodied cognitive systems, both artificial and as models of human cognition. In *cedar*, models can be designed through a graphical interface and interactively tuned. We demonstrate this by implementing an exemplary robotic architecture.

**Keywords:** software framework, embodied cognition, neural dynamics, Dynamic Field Theory, cognitive robotic models

### 1 Introduction

As scientists from diverse fields recognize the critical importance of grounding cognitive function in sensory-motor processes, the embodiment stance is becoming a shared perspective in the study of both artificial and natural cognition [4]. Embracing embodiment has consequences for cognitive modeling. Models of human cognition that account for psychophysical or neural data must include motor control and the associated sensory processes. Artificial cognitive systems have to be implementable on robotic hardware so that intelligent behavior may be generated while the system is situated in the real world. This requires that cognitive

\* The authors acknowledge the financial support of the European Union Seventh Framework Programme FP7-ICT-2009-6 under Grant Agreement no. 270247—NeuralDynamics. This work reflects only the authors' views; the EC is not liable for any use that may be made of the information contained herein.

**Tutorial!**