

a modular software platform for closed-loop experiments



Jan Benda

Department Biologie II Ludwig-Maximilian Universität München



Sensory electrophysiology

Electrosensory systems of weakly eletric fish

Auditory system of grasshopper and crickets



How are sensory stimuli processed by sensory systems?



Closed-loop experiments with RELACS

- 1. Present a stimulus
- 2. Record the response

relacs

- **3.** Immediately analyze and visualize the data
- 4. Generate the next stimulus



Simple closed-loop experiments

- Online visualization of processed data:
 - General infos, e.g. quality of spike detection, sensitivity of the cell, temperature, condition of animal, ...
 - Specific results, e.g. spike raster, firing rates, spike-triggered averages, ...
 - \Rightarrow Speeds up manual ("traditional") closed-loop

Simple closed-loop experiments

- Online visualization of processed data:
 - General infos, e.g. quality of spike detection, sensitivity of the cell, temperature, condition of animal, ...
 - Specific results, e.g. spike raster, firing rates, spike-triggered averages, ...
 - \Rightarrow Speeds up manual ("traditional") closed-loop
- Set stimuli relative to the neuron's dynamic range
- Automatically control motorized electrodes (great for dual unit recordings!)
- Optimize tuning curve measurements

relacs







either:

fast \rightarrow low resolution





either: fast \rightarrow low resolution or: high resolution \rightarrow slow

relacs____



either: fast \rightarrow low resolution or:

high resolution \rightarrow slow





relacs



Closed loop:

relacs



either: fast \rightarrow low resolution or:

high resolution \rightarrow slow

- start with low resolution
- 2. increase resolution where necessary!



Closed loop:



either: fast \rightarrow low resolution or:

high resolution \rightarrow slow

- 1. start with low resolution
- 2. increase resolution where necessary!
- **3.** further increase resolution

relacs____



Closed loop:



either: fast \rightarrow low resolution or:

high resolution \rightarrow slow

- 1. start with low resolution
- 2. increase resolution where necessary!
- **3.** further increase resolution

relacs____



- Optimal search for a neuron's receptive field.
- Search for stimuli that drive a neuron in an "optimal" way.
- Find set's of stimulus parameter that result in the same response (iso-response method).

• ...

Benda et al. (2007): "From response to stimulus: adaptive sampling in sensory physiology." *Curr. Opin. Neurobiol.* **17**: 430–436.



Example: optimal stimulus ensembles



Machens et al. (2005) Neuron 17: 47-56.

relacs___

RELACS ... enjoy your recordings

Relaxed Electrophysiological data Acquisition, Control, and Stimulation RELACS is a framework for closed-loop experiments



⇒ currently 15 scientific publications based on RELACS data in Neuron, J Neurosci, PLoS Biol, Nat Neurosci, J Neurophysiol, etc.

relacs_

RELACS research protocols

In RELACS the closed-loop cycle can be freely programmed as a C++ plugin ("research protocol").

The research-protocol plugins

- take recorded and pre-analyzed data
- perform analysis & display results
- generate next stimulus





Dynamic clamp

Current-clamp, with the current I computed as a function of the measured membrane potential V.



Closed-loop at a per sample time scale (tens of kHz).

relacs____



 $\begin{array}{cccc} & & & & & & & \\ & & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\$

Andrew A. Sharp, Michael B. ONeil, L. F. Abbott, & Eve Marder (1993) J Neurophysiol

- Synaptic conductances
- Voltage-gated conductances

relacs____



relacs

Artificial networks

$$I_1 = g_{syn}(V_2) \cdot (V_1 - E)$$
 $I_2 = g_{syn}(V_1) \cdot (V_2 - E)$



Theoden I. Netoff, Matthew I. Banks, Alan D. Dorval, Corey D. Acker, Julie S. Haas, Nancy Kopell, & John A. White (2005) *J Neurophysiol*

- Artificially couple real neurons
- Couple with simulated neurons

Precision-switch by leak conductance



relacs



Dynamic clamp: leak current $I = g(V - E), E := V_{rest}$



Boucsein, Ammer, Benda (2010) in preparation



Modular design

RELACS core with flexible C++ Plugins for

- hardware abstraction
- data pre-processing (filter, spike detectors)
- research protocols



relacs____

Hardware independent protocols

RELACS integrates all hardware components.

Research protocols for RELACS

- are implemented independently of specific hardware
- can be used on all the different experimental setups in your lab without any modifications
- can be shared with other labs

relacs



Options for research protocols

MembraneResistance version 1.0 (Nov 12, 2009) by Jan Benda Help Stimulus OOD + In Amplitude of output signal (amplitude) OOD + In Compute amplitude from vstep and estimated membrane resistance (useran) Steady-state voltage amplitude induced by output signal (vstep) -1 + In Buration of output (duration) 200 + In In In Duration of pause between outputs (pause) 600 + In In Repetitions of stimulus (repeats) 100 + In In Mindow length for steady-state analysis (sswidth) 50 + In In Fix steady-state potential for fit (nossfit) In In rest only + Bit Set values as default Changes overwrite macro options Froject Experiment	📕 MembraneRe	esistance Settings			
Stimulus Image: Compute signal (amplitude) Image: Compute signal (amplitude) Compute amplitude from vstep and estimated membrane resistance (userms) Steady-state voltage amplitude induced by output signal (vstep) -1 Image: Compute amplitude induced by output signal (vstep) Duration of output (duration) 200 Image: Compute amplitude induced by output signal (vstep) -1 Image: Compute amplitude induced by output signal (vstep) Duration of output (duration) 200 Image: Compute amplitude induced by output signal (vstep) -1 Image: Compute amplitude induced by output signal (vstep) -1 Image: Compute amplitude induced by output signal (vstep) -1 Image: Compute amplitude induced by output signal (vstep) -1 Image: Compute amplitude induced by output signal (vstep) -1 Image: Compute amplitude induced by output signal (vstep) Image: Com	E.	MembraneResistance version 1.0 (Nov 12, 2009) by Jan Benda	Ŀ	feib	
Amplitude of output signal (amplitude) Implitude from vstep and estimated membrane resistance (useran) Steady-state voltage amplitude induced by output signal (vstep) -1 -1 Duration of output (duration) 200 - Duration of pause bewteen outputs (pause) 600 - Repetitions of stimulus (repeate) 600 - Mindow length for steady-state analysis (smaidth) 50 - Fix steady-state potential for fit (nossfit) - - Window length for steady-state analysis (smaidth) 50 - Fix steady-state potential for fit (nossfit) - - Window length or steady-state optimulation of membrane potential (plotetdev) - - Set results to the session variables (setdata) rest only v - Changes overwrite macro options - - Project	Stimulus				
Compute amplitude from vstep and estimated membrane resistance (userm) Steady-state voltage amplitude induced by output signal (vstep) Duration of output (duration) Duration of pause bewteen outputs (pause) Repetitions of stimulus (repeate) Analysis Window length for steady-state analysis (sswidth) Fix steady-state potential for fit (nossfit) Pi Plot standard deviation of membrane potential (plotetdev) Set results to the session variables (setdata) Rest values as default Changes overwrite macro options Project Experiment	Amplitude of ou	utput signal (amplitude)	-0.05	-	nA
Steady-state voltage amplitude induced by output signal (vatep) 1 m m Duration of output (duration) 200 m m Duration of pause bewteen outputs (pause) 600 m m Repetitions of stimulus (repeate) 100 m m Analysis 100 m m Mindow length for steady-state analysis (seswidth) 50 m n Fix steady-state potential for fit (nosefit) 50 m n B< Detailed analysis (setulatea)	Compute a	amplitude from vstep and estimated membrane res	sistance (us	ern	n)
Duration of output (duration) 200 * * n Duration of pause bewteen outputs (pause) 600 * * n Repetitions of stimulus (repeats) 100 * * n Analysis 100 * * n Project 50 * n n Project results to the session variables (setdata) rest only * * Servalues as default Changes overwrite macro options * * Project	Steady-state vo	oltage amplitude induced by output signal ($vstep$) [-1	A	m∨
Duration of pause bewteen outputs (pause) 600 ** Repetitions of stimulus (repeats) 100 ** Inalysis ** Window length for steady-state analysis (sswidth) 50 ** Fix steady-state potential for fit (nossfit) ** Window length for steady-state analysis (sswidth) 50 ** Fix steady-state potential for fit (nossfit) ** Window length for steady-state analysis (sswidth) 50 ** Fix steady-state potential for fit (nossfit) ** Set results to the session variables (setdata) rest only * Changes overwrite macro options ** project ** experiment **	Duration of out	put (duration)	200	-	ms
Repetitions of stimulus (repeats) 100 • Analysis • Window length for steady-state analysis (sewidth) 50 • Fix steady-state potential for fit (nossfit) • KI Plot standard deviation of membrane potential (plotstdev) Set results to the session variables (setdata) rest only • Set set values as default • Changes overwrite macro options • Project • experiment •	Duration of pau	use bewteen outputs (pause)	600	1	ms
Analysis 50 • • Window length for steady-state analysis (###width) 50 • • Fix steady-state potential for fit (no##fit) • Window length for steady-state analysis (###width) 50 • • Fix steady-state potential for fit (no#fit) • Window length for steady-state analysis (###width) • Set results to the session variables (###data) rest only • Set values as default • Changes overwrite macro options • Project • Experiment •	Repetitions of s	100	-		
Window length for steady-state analysis (sswidth) 50 • Fix steady-state potential for fit (nossfit) B Plot standard deviation of membrane potential (plotstdev) Set results to the session variables (setdata) rest only • Set Set values as default Changes overwrite macro options Project Experiment	Analysis		1	-	
Fix steady-state potential for fit (nossfit) Fix steady-state potential (plotstdev) Set results to the session variables (setdata) rest only ▼ Set values as default Changes overwrite macro options Project Experiment	Window length	50	-	ms	
Set results to the session variables (set data) [rest only] • Set values as default Changes overwrite macro options Project Experiment	Fix steady	-state potential for fit (nossfit) ard deviation of membrane potential (plotstdev			
Set Set values as default Changes overwrite macro options Croject Sxperiment	Set results to th	në session variables (setdata)	rest only	•	
Experiment	Set values as o Changes overv Project	default write macro options			
	Experiment				
Ok Apply Run Reset Close	<u>O</u> k	Apply Run Reset		Clo	se





... execute **research protocols** with specific parameter settings:

\$FIField startsession
FIField
SysLatency
FICurve: duration=40ms; pause=260ms;
detector Spikes-1: save



Research-protocol example

```
int Example :: main( void ) {
  double frequency = number( "frequency" );
  double duration = number( "duration", "s" );
  double amplitude = 0.0:
  OutData signal:
  signal.setTrace( "LeftSpeaker" );
  signal.sineWave( frequency, duration, amplitude );
  SampleDataD rate( 0.0, duration, 0.001 ):
  for ( int counter=0: counter<20: counter++ ) {
    write ( signal ):
    sleep( duration + pause );
    EventData spikes (events ("Spikes -1"), signalTime(), signalTime() + duration):
   double meanrate = spikes.rate( 0.3*duration, duration );
    spikes.addBate( rate, counter, GaussKernel( sigma ) ):
   P. lock ():
   P. clear():
   P.setXRange( 0.0. duration ):
   P.plot( rate, 1000.0, Plot::Yellow, 2, Plot::Solid );
   P. draw():
   P. unlock ():
    if ( meanrate < targetrate ) {
      amplitude *= 2.0;
      signal.sineWave( frequency, duration, amplitude ):
    }
  return Completed:
```

relacs_

C++ library for data analysis

Data structures (classes, container):

- Array Basic 1-D vector
- SampleData 1-D data vector with regularly sampled time axis
- *Map* Sequence of x|y data pairs

Algorithms:

- basic statistics (moments, quartiles, histogram)
- power spectra, coherence, transfer function
- linear fits
- non-linear fits (Simplex, Levenberg-Marquardt)

relacs____

C++ library for data analysis

Data structures (classes, container):

- EventData Spikes and other point process data
- EventList Multi-trial spike trains

Algorithms:

relacs

- firing rates (mean, PSTH binned/kernel, 1/ISI)
- CV, Fano factor, ISI correlation
- vector strength, reliability, jitter
- mutual information (lower and upper bound)



Research protocols also run on simulated data:

- test closed-loop algorithms
- directly compare models with experimental data









German neuroinformatics node

www.g-node.de





 All data transfer for analysis, mamagement, and sharing requires talking about data.





- All data transfer for analysis, mamagement, and sharing requires talking about data.
- How to exchange metadata?
- How to record metadata?

relacs____

The meta-data problem

Name-value (+unit) pairs for:

- Stimuli
- Experimental settings
- Cell, preparation, experimental subject
- Hardware properties
- Analysis parameter
- etc.





The meta-data problem

Name-value (+unit) pairs for:

- Stimuli
- Experimental settings
- Cell, preparation, experimental subject
- Hardware properties
- Analysis parameter
- etc.

But:

- What name to choose?
- What does it mean?
- How to share meta-data?

relacs___



- open metadata markup language
 - simple key-value based, hierarchical structure:



- all meta-data can be immediately stored (e.g. no XML namespace extensions required)
- independent of data-base schemas
- standardization through terminologies

relacs____Grewe, Wachtler, Benda (2010) submitted. www.g-node.org/odml



odML terminologies

names & definitions

Hardware:

Amplifier:

name	type	description
Gain	float	The amplifier gain.
HighpassCutoff	float	The cutoff frequency of the amplifier's highpass filter. Given in Hz.
LowpassCutoff	float	The cutoff frequency of the amplifier's lowpass filter. Given in Hz.
Mode	string	The amplifier mode. E.g. Bridge, CC, VC etc.

relacs____Grewe, Wachtler, Benda (2010) submitted. www.g-node.org/odml



- 1. Assemble properties:
 - If you find an appropriate property in the odML-terminologies, use it!
 - Ignore all properties that do not match.
 - Add your own properties that are not yet in the terminology, if possible with a description.





- 1. Assemble properties:
 - If you find an appropriate property in the odML-terminologies, use it!
 - Ignore all properties that do not match.
 - Add your own properties that are not yet in the terminology, if possible with a description.
- 2. Write them into an odML XML file





- 1. Assemble properties:
 - If you find an appropriate property in the odML-terminologies, use it!
 - Ignore all properties that do not match.
 - Add your own properties that are not yet in the terminology, if possible with a description.
- 2. Write them into an odML XML file
- 3. Transfer them to an analysis or database program





- 1. Assemble properties:
 - If you find an appropriate property in the odML-terminologies, use it!
 - Ignore all properties that do not match.
 - Add your own properties that are not yet in the terminology, if possible with a description.
- 2. Write them into an odML XML file
- 3. Transfer them to an analysis or database program
- ⇒ odML flexibility: all available metadata can be immediately stored in a file





- 1. Assemble properties:
 - If you find an appropriate property in the odML-terminologies, use it!
 - Ignore all properties that do not match.
 - Add your own properties that are not yet in the terminology, if possible with a description.
- 2. Write them into an odML XML file
- 3. Transfer them to an analysis or database program
- ⇒ odML flexibility: all available metadata can be immediately stored in a file
- ⇒ odML standard: The G-Node electrophysiology database is based on odML: ww.g-node.org

relacs____



• Every online recording software knows about most of the important meta-data!





- Every online recording software knows about most of the important meta-data!
- ⇒ All available Meta-data should be written to a file directly from the recording software, if possible using odML terminologies.





- Every online recording software knows about most of the important meta-data!
- ⇒ All available Meta-data should be written to a file directly from the recording software, if possible using odML terminologies.
 - Such automated meta-data storage is the basis for making public data bases, such as www.g-node.org, work.



Meta-data acquisition by RELACS



relacs_

Meta-data acquisition by RELACS

X-A RELACS - Relaxed EL	ectrophysiological da	ta Acquisition, Control, an	d Stimulati	on: Version 0.9.6 (11/24/0 🔹 🗆 🗙			
<u>File Plugins Devices De</u>	X - Stop Session Dialo	og		K.			
Spikes-1	Meta Data Setup			MM			
Minimum threshold	Recording						
Threshold	Recording quality	Good	•				
Ratio threshold / size	Comment			111111			
Spike size	Scientist	Jan Benda	•	····			
4,	Temperature	22	¢ °c				
	Humidity	54	÷ %				
mV 0 50 10	Cell						
	Cell type	Low-frequency receptor	*	0 30 40 50 60 70 80			
Info Spectrum	Recording location	Auditory nerve	•	-I Curve F-I Field Search			
- B 100-	Side	Left	-	g rate [Hz]			
00	Depth	0	📮 μm				
	Subject			2			
Frequency [kHz]	Species	Locusta migratoria	•	,			
10 10	Sex	Female	-)			
d rate	Age	Middle-aged	-)			
ili o	Preparation	in vivo dorsal	•				
Intensity [dB SPL]				20 40 60 80 100 Intensity [dB SPL]			
Search (ESC)	<u>S</u> ave <u>D</u> isc	ard <u>R</u> eset	<u>C</u> ancel	Help (F3)			
00:01 2008-12-02-ab/ 12% Frequency 4 kHz, Increment 4, Intensity 55 dB SPL, Loop 2							

relacs_



RELACS records many meta-data:

- General infos about the experiment (from the dialog)
- Main characteristics of the recorded cell
- All RELACS-controlled hardware settings (e.g. sampling rate)
- All settings and version numbers of the research protocols
- Properties of the stimuli





... enjoy your recordings

- \rightarrow Closed-loop experiments
 - \rightarrow Dynamic clamp
 - \rightarrow Simulation mode
 - \rightarrow Hardware independent
 - \rightarrow Data analysis libraries
 - \rightarrow Meta-data storage
- \rightarrow Open source, GPL, Linux
- $\rightarrow \sim$ 160 000 lines of C++ code

www.relacs.net



by Jan Benda