Neuroinform 18 (2020): 377–393

DOI: https://doi.org/10.1007s12021-019-09440-z

Noname manuscript No.
(will be inserted by the editor)

Automatic adaptation of model neurons and connections to build hybrid circuits with living networks

Manuel Reyes-Sanchez · Rodrigo Amaducci · Irene Elices · Francisco B. Rodriguez · Pablo Varona

Received: date / Accepted: date

Abstract Hybrid circuits built by creating mono- or bi-directional interactions among living cells and model neurons and synapses are an effective way to study neuron, synaptic and neural network dynamics. However, hybrid circuit technology has been largely underused in the context of neuroscience studies mainly because of the inherent difficulty in implementing and tuning this type of interactions. In this paper, we present a set of algorithms for the automatic adaptation of model neurons and connections in the creation of hybrid circuits with living neural networks. The algorithms perform model time and amplitude scaling, real-time drift adaptation, goal-driven synaptic and model tuning/calibration and also automatic parameter mapping. These algorithms have been implemented in RTHybrid, an open-source library that works with hard real-time constraints. We provide validation examples by building hybrid circuits in a central pattern generator. The results of the validation experiments show that the proposed dynamic adaptation facilitates closed-loop communication among living and artificial model neurons and connections, and contributes to characterize system dynamics, achieve control, automate experimental protocols and extend the lifespan of the preparations.

Keywords Interacting living and model neurons · Closed-loop neuroscience · Experiment automation · Dynamic clamp · Hybrid circuit real-time dynamic adaptation

Grupo de Neurocomputación Biológica, Dpto. de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid, Spain
E-mail: manuel.reyes@uam.es, pablo.varona@uam.es
Manuel Reyes-Sanchez ORCID: 0000-0003-2909-4664
Rodrigo Amaducci ORCID: 0000-0003-2489-5569
Irene Elices ORCID: 0000-0002-3748-8797
Francisco B. Rodriguez ORCID: 0000-0003-4053-099X
Pablo Varona ORCID: 0000-0002-1754-8991

1 Introduction

Hybrid circuits are networks built by connecting model neurons and synapses to living cells. Pioneering works in building such interactions go back almost three decades ago (Yarom, 1991) with many successful implementations since then, e.g. see (Szücs et al, 2000; Pinto et al, 2000; Varona et al, 2001; Le Masson et al, 2002; Nowotny et al, 2003; Olypher et al, 2006; Arsiero et al, 2007; Grashow et al, 2010; Brochini et al, 2011; Wang et al, 2012; Hooper et al, 2015; Norman et al, 2016; Broccard et al, 2017; Mishchenko et al, 2018). Hybrid circuits are typically implemented through a dynamic clamp protocol that injects current computed by a model from an instantaneous voltage recording (Robinson and Kawai, 1993; Sharp et al, 1993; Prinz et al, 2004; Destexhe and Bal, 2009; Nowotny and Varona, 2014). The model can be a simple conductance description or the outcome of a complex biophysical neuron or network simulation. Neuron models used to build hybrid circuits expand from simplified nonlinear equations to Hodgkin-Huxley type paradigms. Synapse models range from standard Ohm's law implementations for gap junctions to nonlinear graded synapse models that require pre- and post-synaptic voltage information. Each model has its own set of parameters and they all require specific adaptations for their use in hybrid circuit preparations.

In spite of the large applicability of hybrid circuits to study neuron and network dynamics, including plasticity and learning mechanisms, their use has been somehow limited by the difficulty of their implementation. Hybrid circuit construction often requires specific hardware and/or soft or hard real-time software technology to accurately implement the associated recording and stimulation cycles (Christini et al, 1999; Pinto et al,

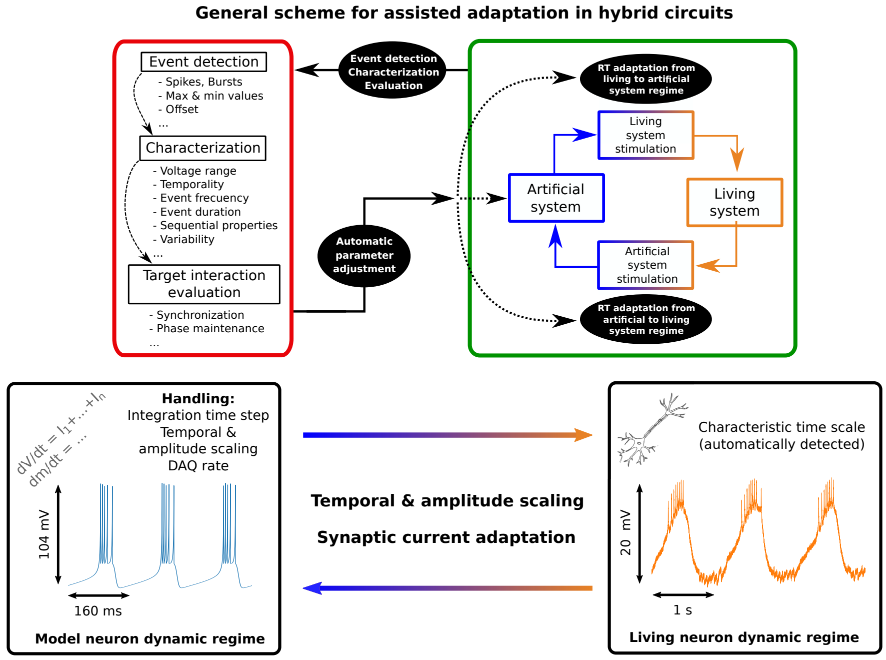
General scheme for assisted adaptation in hybrid circuits
Event detection
- Spikes, Bursts
- Max & min values
- Offset
...
Characterization
- Voltage range
- Temporality
- Event frecuency
- Event duration
- Sequential properties
- Variability
...
Target interaction evaluation
- Synchronization
- Phase maintenance
...
Event detection Characterization Evaluation
Automatic parameter adjustment
RT adaptation from living to artificial system regime
Living system stimulation
Artificial system
Living system
Artificial system stimulation
RT adaptation from artificial to living system regime
Handling:
Integration time step
Temporal & amplitude scaling
DAQ rate
dV/dt = I1+...+In
dm/dt = ...
104 mV
160 ms
Model neuron dynamic regime
Temporal & amplitude scaling
Synaptic current adaptation
Characteristic time scale (automatically detected)
20 mV
1 s
Living neuron dynamic regime

Fig. 1 General closed-loop approach proposed for the assisted automatic adaptation in hybrid circuits. Living and model systems are connected mono- or bi-directionally with the required adaptation in each direction to make interaction signals compatible with their dynamical regime in real-time. During a test observation, the system is evaluated through the detection of reference events that are relevant for the time and amplitude scaling, e.g. spikes, bursts, maximum and minimum voltage amplitudes, offset, etc. Hybrid circuit efficiency measures for the system characterization are calculated and parameters are adjusted accordingly from the reference event evaluation. Several cycles in the interaction are typically used to assess the hybrid circuit target goal. Adaptation is done in both directions in bidirectional interactions.

2001; Muñiz et al, 2005; Arsiero et al, 2007; Muñiz et al, 2009; Kemenes et al, 2011; Nowotny and Varona, 2012; Linaro et al, 2014, 2015; Patel et al, 2017; Amaducci et al, 2019).

Additionally, the construction of hybrid circuits involves handling the different time and amplitude scales of living and model neurons and synapses. In the case of electrophysiological experiments, electrode resistance specifications are also an important issue. All amplitude and time scale adjustments have to be addressed specifically for each preparation. These tasks are time-consuming and often a main source of frustration when building hybrid circuits.

Furthermore, when using hand-tuned parameters in a hybrid configuration, experimentalists frequently have to deal with voltage drift and the natural evolution of membrane potential oscillations during minutes or hours of experimental work. In this paper, we present a set of algorithms to facilitate the building of hybrid circuits using software neurons and synapses. These algorithms perform automatic calibrations and dynamic adaptations of time, voltage amplitude/offset, and current scales to implement open- and closed-loop interactions with living neurons in real-time following the scheme illustrated in Fig. 1. Our validation experiments show that the use of these algorithms contributes to better tuned and more natural interactions between living and model neurons, to a reduction of the risk of preparation damage and time expended on adjustments and, thus, to expand the life expectancy of the experiments. We also show that the proposed dynamic adaptation approach for building hybrid circuits is useful to automate experiments, achieve goal-driven control of neural activity, and explore and map neural dynamics.

To favor their dissemination and use, the algorithms described in this paper have been implemented in RTHybrid (Amaducci et al, 2019). RTHybrid is an open-source library that can be run over different Linux platforms, with or without hard real-time constraints. The program includes the algorithms described in this paper, implemented to work under Linux based systems (with stock Linux kernels or with the real-time patches Xenomai and PREEMPT-RT). The library includes several models of neurons and synapses ready to be used in a large variety of hybrid circuit configurations. The configuration of the experiment can be selected from the user interface or through XML scripts. The program also includes a tool to visualize the recordings of the hybrid interaction. The software is freely available on `www.github.com/GNB-UAM/RTHybrid`. The algorithms described in this paper (summarized in Table 1) can also be easily migrated to other existing platforms.

2 Material and methods

2.1 Experimental setup

To validate the automatic adaptation algorithms described in this paper, we built hybrid circuits using the crustacean pyloric central pattern generator (CPG) (Marder and Calabrese, 1996; Selverston, 2005). The neurons of this circuit can be easily identified by their waveform and are particularly resistant to long recordings while sustaining their characteristic spiking/bursting rhythm. As we illustrate in our validation experiments, these neurons generate sequential activity in hybrid circuit experiments using bidirectional interactions with a wide variety of artificial neuron models and synapses ranging from simplified descriptions to conductance-based paradigms.

Adult shore crabs (*Carcinus maenas*) were used in all preparations. They were purchased locally and maintained in a tank with artificial seawater at 13-15°C. Crabs were anesthetized by ice for 15 min before dissection. The procedures followed the European Commission and Universidad Autónoma de Madrid animal treatment guidelines. The stomatogastric nervous system was dissected following standard procedures and pinned in a Sylgard-coated dish containing *Carcinus maenas* saline (in mM: 433 $NaCl$, 12 KCl, 12 $CaCl_2 \cdot 2H_2O$, 20 $MgCl_2 \cdot 6H_2O$, 10 $HEPES$, adjusted to pH 7.60 with 4 M 287 $NaOH$). After desheathing the stomatognathic ganglion (STG), neurons were identified by their membrane potential waveforms and the spikes times observed in extracellular recordings from the corresponding motor nerves. Membrane potential was recorded using 3 M KCl filled microelectrodes (50 $M\Omega$) and a DC amplifier (ELC-03M, NPI). Current injection to implement the hybrid connection was delivered with a second electrode on the same neuron. Data was acquired using a A/D DAQ board (PCI-6251, National Instruments). The DAQ sampling rate is an input parameter in all algorithms described below, which is chosen by the experimentalist (depending on the neuron being recorded and the setup capability). The validation experiments shown in this paper were run with a 10 kHz acquisition/stimulation cycle. The offset in the amplifier, as well as the DA resolution, are set *a priori* for a given recording.

2.2 Neuron and synapse models

Different neuron model paradigms can be used in hybrid circuit experiments as long as their equations can be integrated in real-time at the desired acquisition rate to implement realistic interactions with living neurons. It is important to note that each model has typically its own time and amplitude scales, which in most cases are quite different from their biological counterparts in a specific experimental setup. The description of some commonly used simplified models considers arbitrary units both for amplitude and time. Other more realistic descriptions, such as conductance-based models, use physiological units whose amplitude ranges may differ from that of a given recording (e.g. see bottom panels inf Fig. 1. Independently of the original amplitude and time units of the model, the acquisition rate determines the time interval available for the model integration at each interaction cycle. Thus, models need to be implemented taking into account the living neuron time scale, the acquisition frequency and the model integration time.

To validate our approach, we used different software neuron models with intrinsic rich dynamics and increased complexity, and thus increased computational cost: (i) the Rulkov map, a two-dimensional iterated map that can display spiking-bursting behavior (Rulkov, 2002); (ii) the Izhikevich model, a two-dimensional system of ordinary differential equations with a quadratic voltage nonlinearity and an auxiliary after-spike resetting mechanism (Izhikevich, 2003); (iii) the Hindmarsh-Rose model, a three-dimensional system of ordinary differential equations with cubic nonlinearities (Hindmarsh and Rose, 1984); and (iv) a conductance based model with the characteristic sigmoid voltage dependencies of Hodgkin-Huxley type descriptions (Ghigliazza and Holmes, 2004). In the experiments depicted below, we employed a chemical graded synapse description frequently used in CPG studies (Golowasch et al, 1999), and also a bidirectional electrical synapse model (Varona et al, 2001). For the integration of the differential equations a (6)5 Runge-Kutta (Hull et al, 1972) numerical method was used. Our results can be generalized to any neuron and synapse models whose equations can be integrated within the interaction cycle.

2.3 Real-time software technology

Living neurons are not fast computation agents, but they can be very precise at the millisecond range. To achieve temporally precise interactions between living and model neurons in a hybrid circuit configuration using a general purpose operative system, real-time patches are needed.

Following the idea of providing easy to install and implement technology, the proposed algorithms have been developed under a real-time operative system in the RTHybrid (`www.github.com/GNB-UAM/RTHybrid`)

Table 1 Automatic adaptation algorithms proposed in this paper and their different requirements, performance measures, advantages and use cases.

Automation algorithm	**Input information parameters (source)**	**Events detected**	**Real time constraints**	**Performance measures**	**Advantages / use cases**
Time Scaling (Alg. 3, 4, 5 & Fig. 3)	Reference event duration valid model step range, sampling rate (DAQ)	Spikes, bursts, hyperpolarization intervals, averages, etc.	Data reading	Event timings, intervals, synchronization levels, target phases	Same timing for living and model neurons, temporal precision
Amplitude and Offset Scaling (Alg. 6 & Fig. 4)	Working range (living/model neurons)	Max. and min. voltage values		Working range assessment	Longer preparation life expectancy, realistic interactions
Real-time Drift Adaptation (Alg. 6 & Fig. 8)			Data reading and re-calculation	Dynamic range assessment	Compensate electrode problems, intrinsic modulations
Synaptic Tuning/ Calibration (Fig. 10)	Synapse parameters and interaction result (living/model signals)	Pre- and post-synaptic voltage, number and temporal structure of spikes, current ranges, hyperpolarization times, burst duration, phases, waveforms, sequences, etc.	Data reading, evaluation and decision making	Synchronization, regularization level, rhythm properties, intervals and phases, robustness of sequential activations, etc	Effective currents
Model Tuning/ Calibration (Fig. 7)	Model parameters and interaction result (living/model signals)				Experiment-specific adaptation, model autocalibration
Automatic Activity Control (Fig. 9)	Synapse and model parameters (stimulus and time constants)				Realistic/natural dynamics, experiment automation
Automatic Mapping (Fig. 10)			Data reading and decision making		Revealing dynamics and bifurcations and characterization as a function of the hybrid circuit parameters

software, which allows the implementation of closed-looop interactions with millisecond hard real-time constraints (Amaducci et al, 2019). For all validation tests, we used a computer with an Intel Core i7-6700 processor running Debian 9 under Linux 4.9 with PREEMPT-RT patch to achieve hard real-time performance. The RTHybrid real-time implementation of the algorithms was also successfully tested in Ubuntu with a Xenomai 3 real-time patch.

2.4 Dynamic clamp hybrid circuit implementation

To implement hybrid circuits in an electrophysiological experiment, it is necessary to record the activity of a presynaptic cell and deliver the corresponding stimulation on a postsynaptic neuron at each step of the acquisition cycle. Typically, this is done using a dynamic clamp protocol to read voltage from the presynaptic neuron and deliver a current into the postsynaptic cell by employing one or multiple electrodes (Destexhe and Bal, 2009). Beyond electrophysiology, a hybrid circuit can use real-time imaging techniques and light (Krook-Magnuson et al, 2013; Prsa et al, 2017) or chemical (Chamorro et al, 2009) stimulation for the implementation of the interaction. The algorithms described in this work aim for automatic adaptation of the signals between living and model networks so that during the interaction they all work in their natural dynamical range. We will illustrate this by implementing the connections with a dynamic clamp protocol in which living neuron voltages are read and adapted to work with synapses that deliver current with the right range both for the living and model neurons at a specified interaction cycle. With the proposed automatic calibration and adaptation protocols, the hybrid circuit is readily implemented in a few seconds regardless of each particular choice for the neuron and synapse models, experimental setup, preparation and/or electrodes.

3 Results

3.1 Closed-loop approach for automatic calibration and adaptation

The automation of hybrid circuit implementations requires online dynamic signal analysis, precise event detection and real-time model integration. Figure 1 shows our overall approach to standardize this process. A typical hybrid circuit system consists of mono and/or bidirectional connections between model and living neurons via synaptic models. Table 1 lists the set of algorithms that we have developed for the automatic construction, calibration and evaluation of hybrid circuits. All these algorithms follow the scheme illustrated in Fig. 1, which is based on real-time reference event detection, and the dynamic characterization and evaluation of the interaction goal (top left box). With this information, online adjustments in the parameters are readily made as a function of predefined performance measurements.

At the beginning of the experiment, the living system dynamics are characterized for a few seconds in order to define a first approximation to the needed adaptations. This is illustrated in the code shown in Algorithm 1, which addresses the initial configuration by setting the DAQ parameters including the interaction cycle rate, establishing the neuron and synapse models to use and the selection of the integration/interpolation algorithm. With this information, the initialization process performs an observation to run the time and amplitude calibration. In this initial phase, the dynamical range of the living neuron is characterized.

After the initial calibration procedures (summarized in Fig. 2), the interaction closed-loop algorithm implements in real-time the hybrid circuit. Algorithm 2 executes the processes that involve the input/output from/to the living neuron and the models, the associated adaptation/calibration protocols and the integration of the models. The different protocols that are used in Algorithms 1 and 2 are explained in detail in the following sections.

```
Initial automatic adaptation procedures {
  Read configuration:
      DAQ setup settings ()
      Neuron and synapse models setup ()
  Open−loop observation ()
  Time calibration () # Alg. 3, 4, 5
  Amplitude/offset scaling cal. () # Alg. 6
}
```

Algorithm 1 Sequence flow of the initial automatic adaptation procedures for the hybrid circuit implementation.

3.2 Time and amplitude scaling

3.2.1 Time scaling

The algorithm that performs the time calibration of the model takes into account reference events that characterize the dynamics of the living neurons under study, such as spike/burst duration, or the neuron activity average period (see Table 1). As illustrated in Fig. 3, different models may describe reference events in their dynamics with distinct time resolution, even when time units are commonly expressed in milliseconds. It is important to note that the model integration time (Bettencourt et al, 2008) and the duration of reference events in the model neuron can be in general very different

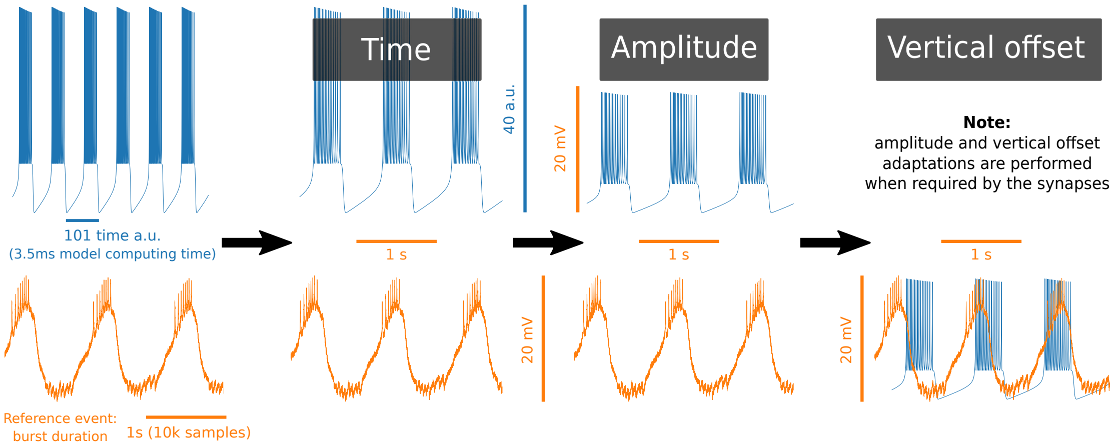

Fig. 2 Process required to adapt model neurons to a living neuron's voltage amplitude and time scales. It is important to distinguish between the original time units of the model, the associated compute time, and the use of the model in a hybrid circuit taking into account the time scale of representative living neuron events (in this case, the burst duration) and the acquisition sampling rate. This is illustrated in the left panel with the Rulkov map model. Once the model has been scaled in time (see section 3.2.1 and Algorithm 5) and amplitude (see section 3.2.2 and Algorithm 6), including the assessment of the vertical offset, it is ready for the hybrid interaction. Right panel corresponds to the resulting adaptation in the working regime of the living neuron.

```
Interaction closed-loop {
  # Interaction with living system
  # m_samples_rate: see Alg. 5
  if ( (i module m_samples_rate) == 0){

    # Manage sampling rate constraints time
    Sleep ()
    # Input/Output
    Read & Write DAQ ()
    # Event references
    Event characterization () # Alg. 3, 4

    # Adaptation re-evaluation
     each N events:
       # Real-time drift adaptation
       # Revaluate Alg. 6
       Amplitude/offset scaling cal. ()
  }
  # Integrate models
  Scale living voltage () # Alg. 6
  Compute synapse to model neuron ()
  Compute model neuron ()
  Scale model voltage () # Alg. 6
  Compute synapse to living neuron ()
  i = i+1
}
```

Algorithm 2 Interaction closed-loop algorithm. The algorithm handles real-time input/output for living and model neurons. Model samples rate (`m_samples_rate`), calculated in Alg. 5, determines how many model integration steps are required per each DAQ sample. The revaluation of amplitude and offset scaling calculations correspond to the real-time drift adaptation and is done every N events as predefined by the experimenter. The sleeep time is calculated by substracting the model integration time, the time taken for the scaling algorithms and the input/output management time from the sampling period (Amaducci et al, 2019).

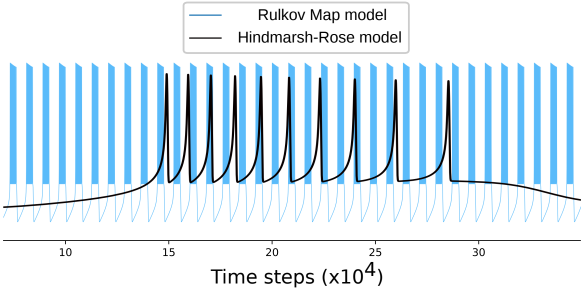

Fig. 3 Illustration of different time scales in two neuron models. In this case, the amplitude range is nearly the same but one of the models (the Rulkov map) produces 35 bursts while the other (a Hindmarsh-Rose model) produces just one with the same amount of time steps for accurate integration and a default parameter choice.

from the characteristic time scale of the living neuron. A good initial choice of the model parameters or a global re-scaling of the model temporal dynamics contributes to the success of the hybrid circuit adaptations explained below. Additionally, biophysical models often require many integration steps per acquisition sample to guarantee a precise simulation as compared to the sampling rate. Thus, the time scaling algorithm performs a time calibration by determining the resolution needed to accurately represent reference events in the dynamics of the living neuron, taking into account the model precise integration and the chosen acquisition sampling rate. Reference events that the user can define are, for example, action potentials or bursts (Arroyo et al, 2013; Varona et al, 2016). The sampling rate determines the required discretization of the signals from the living neuron and the model.

To address this non-trivial task, we adapt the time scale of model neurons and synapses taking into account the characteristic time range of the events defined in the living neuron (see Algorithm 3) together with the chosen DAQ sampling rate, which determines the resolution of the discretization. To define characteristic events, e.g. bursts, subevents building the events need to be detected, e.g. single spikes (see Algorithms 3 and 4). Thus, in our automatic time scaling algorithm, corresponding events (e.g. a burst) are detected both in the signals from the living and model neurons to set the time scale factors. In parallel, information about the DAQ sampling rate and the valid model integration steps are taken into account. With this information two distinct cases can arise (Algorithm 5):

- If the model has less temporal resolution for an event than the discretization of the living neuron signal set by the sampling rate, the algorithm performs interpolation to provide the required model values for the interaction. This is typically the case of a map model description.
- If the integration of the model generates more points per interaction than needed, the integration step is selected to fulfill two conditions: (i) the time step of the integration has to be smaller than the maximum established for an accurate integration of the model; (ii) the number of points for the reference event in the model must be equal or larger than the sampling rate discretization of the event in the living neurons.

The system also subsamples if the number of model voltage values provided by the model integration is larger than those required for the interaction at a particular acquisition rate. It is important to note that an accurate integration may require a small time step. However, the interaction with the living system only requires the value of the model voltage at the chosen interaction sampling rate. This is calculated in Algorithm 5 (`ref_event_t`) and used in the interaction closed-loop (Algorithm 2). Also, Table 2 shows examples of different models and the selected accurate integration step to match bursts lasting one second with a 10 kHz adquisition rate. The time scaling algorithm uses the real-time detection of predefined reference events (Algorithms 3 and 4) from the living neuron signal, such spikes and bursts, to perform the assessments described above. Algorithm 5 optimizes model compute time while keeping integration accuracy, taking also into account the large variety of models, including their parametrization, which can be used to implement hybrid circuits.

3.2.2 Amplitude and offset scaling

Once the time scale has been established, voltage amplitude differences between the living and the model neurons have to be adapted, when the synaptic model requires it. Dynamic-clamp protocols are particularly sensible to this procedure as in many cases both pre- and post-synaptic membrane potential values are involved in the calculation of synaptic currents from/to

```
Event characterization {
  # Anotate new subevent (voltage and time)
  i++
  if (detect_func(signal[time]) == False)
      # No event detected
      return
  else{
      subevent[i].t = current_time()
      ...
  }

  # Subevent characterization
  t_interval = subevent[i].t-subevent[i-1].t
  subevent[i].t_interval = t_interval
  ...

  # Evaluate subevents chain
  # se_index: subevent counter
  if (t_interval == t_chain[se_index]){
    # Current subevent index update
    se_index = se_index+1
    ...
  }else{
    # This was not an event, e.g. timeout
    if (t_interval > time_th[se_index]){
      # Event reset
      se_index = 0
    }
    ...
  }

  # Subevents are grouped in one event
  # total_se: number of subevents
  # contained in the current event
  if (se_index == total_se){
    # n is the number of subevents per event
    event[j] = subevents[from i-n to i]
    # Event characterization
    event[j].start = subevent[i-n]
    event[j].end    = subevent[i]
    event[j].duration  =
                 subevent[i] - subevent[i-n]
    ...
    # Event reset
    se_index = 0
    j++
    ...
  }
}
```

Algorithm 3 Event characterization algorithm. Events are characterized as a sequence of subevents using a detection function (`detect_func`). These subevents are evaluated using a chain approximation. When a subevent chain is completed, the associated sequence of subevents is classified as an event. The algorithm evaluates the duration and other metrics over the event (indicated by ellipsis dots), which is used for the time calibration. A `detect_func` example for neuronal bursting activity is described in Algorithm 4.

the neurons. The parameters for this adaptation are typically calculated manually and offline, but this is a slow and tedious process, often taking precious time from the experiment. It is important to note that, even within the same experiment, phenomena like drift or intrinsic changes in the preparation can also produce amplitude and offset changes, which make dynamic voltage adaptation necessary.

```
# This function is called from:
# Algorithm 3, line 4
Burst detection (v_value,
                 up_th,
                 lw_th)
{
  # Define initial state
  if (search is not defined) {
    search = True
    if (v_value > up_th)
      search = False
  }
  if (search == True && v_value>up_th){
    # Burst beginning detected
    search = False
    return True
  }else if
     (search == False && v_value<low_th){
    # Burst end detected
    search = True
  }
  return False
}
```

Algorithm 4 Example of a simple `detect_func` to detect neuronal bursts which can be used in Algorithm 3. Upper threshold (`up_th`) is used to detect the first spike in the burst. Lower threshold (`lw_th`) is used to detect when the burst ends.

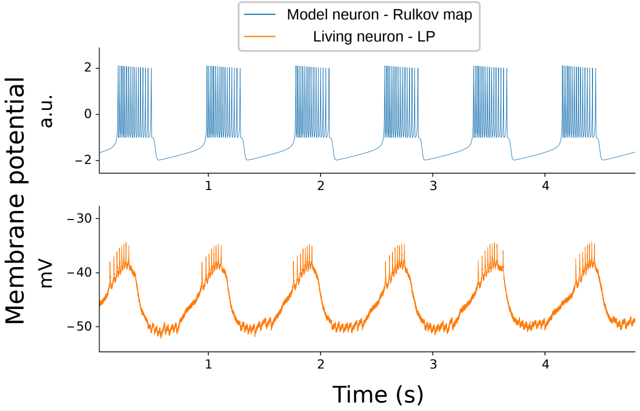

Fig. 4 Illustration of the need for voltage amplitude scaling between a model neuron and a living neuron. In this case, the model neuron is a Rulkov map whose bursting amplitude is in the range between -2 and 2 a.u. The living neuron in this example displays bursting activity in the range between -55 to -35 mV. The bursting frequency has already been adapted in the model to match that of the living neuron using the time scaling algorithm.

At any time, each entity of the circuit (living or artificial) must work on its own dynamical range. The adapted voltage value has to be calculated in both directions (model to living and living to model neuron) and used in the synapses to calculate the current to the target neuron. This dynamic adaptation also protects the preparation from excessive current injection, a common issue in manual adaptations that put at risk the living neuron. The differences consist both on amplitude absolute range and vertical offset (see fig. 4). The scale is done in both directions, as each neuron continues working on its own dynamical range during the hybrid circuit experiment.

```
Time calibration (daq_freq,
                  ref_event_t,
                  dt_max_pts,
                  dt_vect[ ])
{
  # Determine event resolution
  l_event_samples = daq_freq * ref_event_t
  aux = l_event_samples
  m_samples_rate = 1
  while aux < dt_max_pts {
    aux = l_event_samples * m_samples_rate
    m_samples_rate = m_samples_rate + 1
    # Search the maximum valid dt
    for each dt in dt_vect[ ]{
      if dt.samples > aux{
        factor =
               dt.samples / l_event_samples
        # The dt is not accurate
        if factor-m_samples_rate <= tol {
          # End
          return m_samples_rate, dt
        }
      }
    }
  }
}
```

Algorithm 5 Model time calibration algorithm. This algorithm is used to adjust the resolution of the model (determined by its integration step) which, in combination with the DAQ sampling rate (`daq_freq`), determines the duration of a reference event, e.g. a burst. To match the duration of the reference event in the living neuron (`ref_event_t`), the most suitable integration step (`dt`) is chosen from a list (`dt_vect[ ]`) of valid values. This integration step is the one that provides the most similar resolution (`dt.samples`) for the reference event in comparison to the living neuron sampling rate. In some cases, the model needs to produce many more integration steps than the sampling rate of the living neuron activity to accurately generate the corresponding reference event in the model. In that scenario, the interaction closed-loop (Algorithm 2) will discard some of the generated points with the best rate (`m_samples_rate`) as evaluated by a tolerance (`tol`) to match the required resolution.

For this task, before the online bidirectional interaction, a first adaptation is done using data at the observation phase taken from recording and the model. The approach used to calculate both amplitude and offset scaling factors is shown in Algorithm 6. Figure 4 shows an example of the need for amplitude adaptation in two neurons that have different voltage scales.

```
Amplitude/offset scaling calculations () {
  # Signals range
  model_range  = modelv_max  − modelv_min
  living_range = livingv_max − livingv_min

  # Amplitude factor
  factor_to_living =
   living_range / model_range

  factor_to_model  =
   model_range  / living_range

  # Vertical offset
  offset_to_living =
   livingv_min − (modelv_min ∗ v_to_living)

  offset_to_model  =
   modelv_min − (livingv_min ∗ v_to_model)
}

Scale model voltage (v) {
  scaled_v =
   (v ∗ factor_to_living) + offset_to_living
}

Scale living voltage (v) {
  scaled_v =
   (v ∗ factor_to_model) + offset_to_model
}
```

Algorithm 6 Amplitude and offset scaling calculations. This algorithm uses signal ranges (`livingv_max`, `modelv_min`, ...) of the living and model neurons to calculate adaptation factors for both living and model neurons (`v_to_living`, `offset_to_model`, ...). During the closed-loop process (Algorithm 2) this calculations are periodically reevaluated with the updated recordings values to deal with signal drift.

3.2.3 Range-compatible neurons

The joint time and amplitude scaling process is illustrated in Fig. 2 and Fig. 5. The specific values of the scaling factors automatically obtained for different models when building a hybrid circuit with a pyloric neuron are shown in Table 2.

The automatic processes illustrated in Figs. 3 and 4 occur at the same time and are independently but simultaneously addressed by the calibration algorithms. The time series of both the living cell and the model neuron are plotted together in Fig. 5 to illustrate that they are in the same time and amplitude scale as in the living neuron control regime (see also Fig. 2). In this example, the synapses of the hybrid circuit are not connected yet but currents in any direction can already use the scaling factors automatically calculated at this point. With both time and amplitude adaptations working at the same time, model neurons are ready to be connected to specific living neurons in a hybrid circuit.

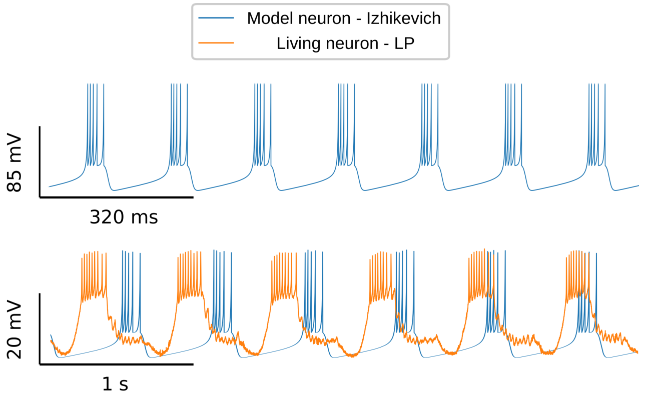

Fig. 5 This example illustrates the time and amplitude adaptation of a Izhikevich neuron model to work in the dynamical regime of a pyloric LP neuron. Top panel shows the initial scale values of the neuron model. Bottom panel shows both neurons with the same scale after the adaptation process. At this point there are no synapses between the model and the living neuron, but time and amplitude scale factors are already set and ready to be used in the synapse models.

3.3 Synapse tuning/calibration

Once amplitude and time scales are compatible between living and model neurons, the connectivity required for the hybrid circuit can be established. Figure 6 shows an example of a hybrid circuit built with a bidirectional connection between the LP neuron of the pyloric CPG and a Izhikevich neuron model (Izhikevich, 2003). The connection models consider one fast and one slow graded synapses to/from the LP, respectively (Golowasch et al, 1999). Despite the previous adaptations, synapses might also require to adjust parameters such as maximum conductances, time scales and kinetic parameters to achieve the connection goal. For example, a desired behavior for the hybrid circuit can be anti-phase rhythmic activity between the living and the model neuron. This process is not trivial, as synapse parameters may play a key role to establish the target goal. For this goal the synaptic conductances can be increased from a low value until anti-phase bursting behavior is reached.

Thus, to determine the right values of synaptic parameters during the hybrid circuit interaction, a set of goal-driven closed-loop protocols have been designed.

Table 2 Example of scaling factors for different neuron models to match the time and amplitude of a pyloric neuron that fires approximately one burst per second. Initial values use an online characterization of the living cell dynamics as reference and are re-calculated dynamically during each experiment. The event was a burst and the DAQ frecuency was set to 10 kHz. The neuron model parameters used in the validation examples correspond to the default parameters of the RTHybrid library.

Model	Time step	Points per cycle	Initial voltage range	Amplitude factor	Offset (Model to cell)	Offset (Cell to model)	Final voltage range
Rulkov map (Rulkov, 2002)	(Interpolation to match sampling rate)	10000	(-1.97; 2.11) a.u.	4.91	-41.04 mV	8.35 a.u.	(-50.68; -30.68) mV
Izhikevich (Izhikevich, 2003)	0.0029	20457	(-74.23; 30.24) mV	0.19	-36.43 mV	191.71 mV	(-50.68; -30.68) mV
Hindmarsh–Rose (Hindmarsh and Rose, 1984)	0.0283	10001	(-1.61; 1.80) a.u.	5.86	-41.23 mV	7.04 a.u.	(-50.68; -30.68) mV
Conductance-based model (Ghigliazza and Holmes, 2004)	0.0062	10153	(-40; 3.5) mV	0.46	-32.29 mV	70.2 mV	(-50.68; -30.68) mV

The general and common structure of these closed-loop protocols consists on three parts (repeated until the connection fulfills the pre-established goal):

(i) The first part establishes a continuous event detection during the closed-loop interaction, as described above (see also fig. 1).
(ii) Secondly, information is used to characterize and evaluate the performance of the established goal for the interaction with a given metric.
(iii) Finally, parameters are changed according to the performance and the goal.

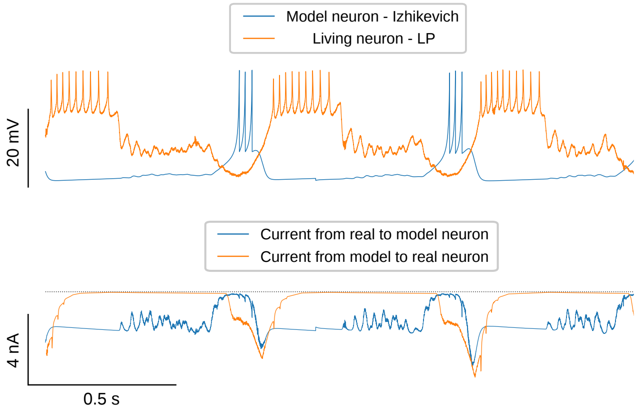

Fig. 6 Hybrid circuit between a pyloric LP neuron and a neuron model (Izhikevich, 2003) built with inhibitory graded synapses (Golowasch et al, 1999) that leads to antiphase bursting behavior. Automatic calibration and adaptation of the model and living neuron time and amplitude scales lead to an effective interaction in the hybrid circuit. Top panel shows the living and model neuron voltage signals in the working space of an LP neuron. The bottom panel shows the injected currents. In this example, the synapse to the model neuron is implemented by a fast graded model and the synapse to the living neuron corresponds to a slow graded synapse model.

The events for the characterization of the activity, the performance measures for the interaction goal, and the model and synapse parameters must be chosen for of each specific experiment. In this context, characterization measures like frequency, phase, level of activity, etc. can be used for multiple goals, just adjusting the target interaction evaluation. As an example of this process, Fig. 7 shows an experiment where the goal set for the hybrid circuit was the in phase synchronization of a living and a model neuron via a bidirectional electrical synapse. Synchronization is a convenient measurement to assess the effectiveness of any synaptic connection in a hybrid-circuit. Thus, as a performance measure, the Mean Square Error (MSE) of the two voltage signals was calculated over a time window of three bursts. In this example, the synchronization goal was met when the MSE reached the minimum defined goal. The MSE was measured during five seconds before connecting the two neurons with the electrical synapse. The target MSE was fixed at 40% of the initial MSE value. The conductance of the connection was increased 0.5 mS every three burst cycles until the MSE target value was reached. As can be seen in the figure, this simple protocol leads to the desired synchronization goal. Model parameters can be adjusted analogously, allowing to match the living neuron behaviour, such as the number of spikes in a burst or their temporal structure (Nowotny et al, 2003).

3.4 Real-time drift and ongoing adaptation

Amplitude and time adaptations performed by the previous algorithms might need to be reevaluated periodically during the experiments. This is the case when dealing with voltage drift in the electrodes or a natural evolution of the membrane potential. Continuous

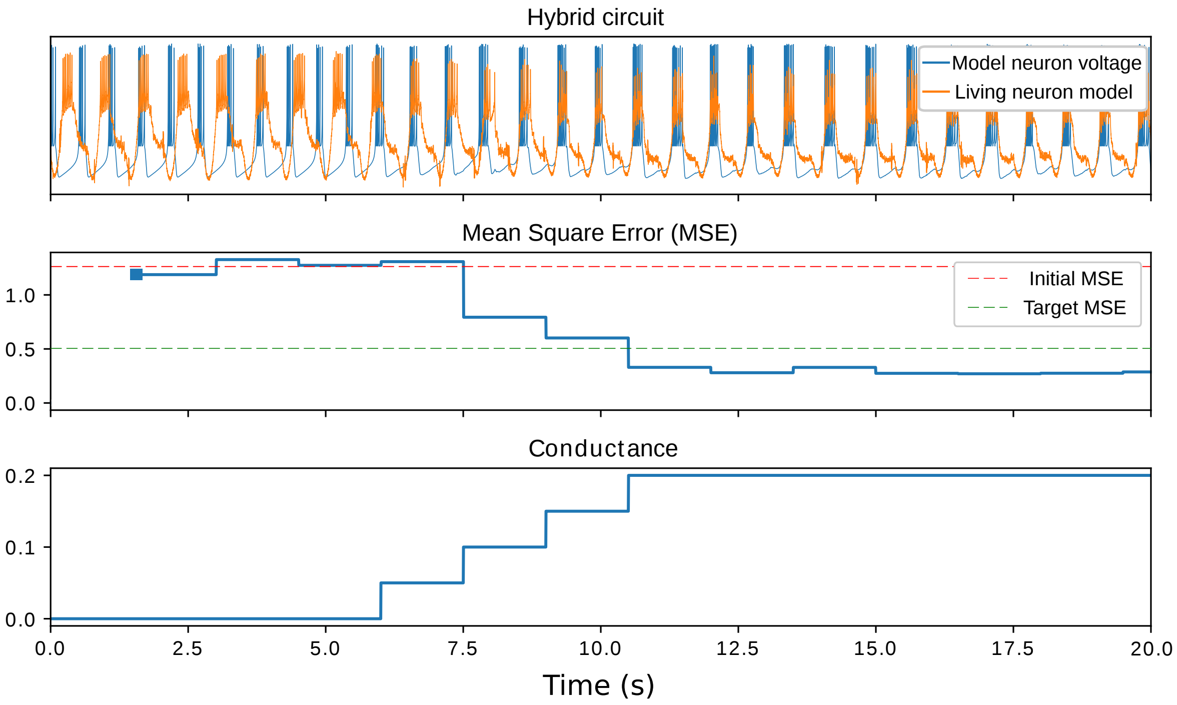

Fig. 7 Illustration of the closed-loop synapse calibration process. In this example, a synchronization goal is set to evaluate the efficiency of a hybrid circuit connection. As a performance measurement, the mean square error of the living and model neuron signals is calculated every three bursts. The parameter chosen for the calibration is the conductance of the bidirectional electrical synapse. The top panel shows the evolution of neurons' membrane potential, middle panel shows the mean square error of the signals and the bottom panel shows the conductance value. This real-time process is also shown in a video included as supplementary material (Online Resource 1).

monitoring of calibrated ranges is highly relevant, as a change in amplitude or offset can lead to a large current injection into the living neuron or to a malfunction of the hybrid-circuit. Thus, real-time evaluation of voltage ranges and a dynamic adaptation of the drift offset is also addressed (Algorithm 6). Figure 8 illustrates this. A time window is defined to set the reevaluation period and the predefined neuron events are again evaluated. Scale factors are thus adjusted as can be seen by the green traces in the figure. In this particular experiment, information was reevaluated every two burst periods and the amplitude factor and vertical offsets were re-calculated accordingly. We chose to adapt the drift offset on the model side.

3.5 Characterization and control of neural dynamics

The event characterization algorithm (Algorithm 3) can also be used to monitor, quantify and control neural dynamics. For example, interactions with the living system can be used to assess the functional role of the system elements, replace damaged elements of the living system while characterizing or sustaining a given dynamics (Szücs et al, 2000; Chamorro et al, 2012; Sakurai and Katz, 2017), or to calculate activity metrics (Couto et al, 2015). A common goal for many experiments in this context is to reach a certain level of activity, for example a specific regular rhythmic regime (Varona et al, 2001; Hooper et al, 2015).

We illustrate this in Fig. 9 with a simple stimulation experiment. In this example current is injected into a living LP neuron from the CPG to achieve rhythm regularization. Real-time burst detection is performed by Algorithm 4. Figure 9 shows how the instantaneous value of the period and its variance is measured within a time window of 5 bursts. This is used as the performance measure for the regularization goal.

In control conditions, the living neuron had irregular spiking-bursting activity. The figure shows that once a certain regularization level was reached, further increasing of the current did not lead to a decrease in the variance (green region), and thus to further regularization. This example illustrates how the event characterization algorithm can be used to track events and deal with measure evaluations to achieve the desired neural behavior with minimal disturbance to the living circuit.

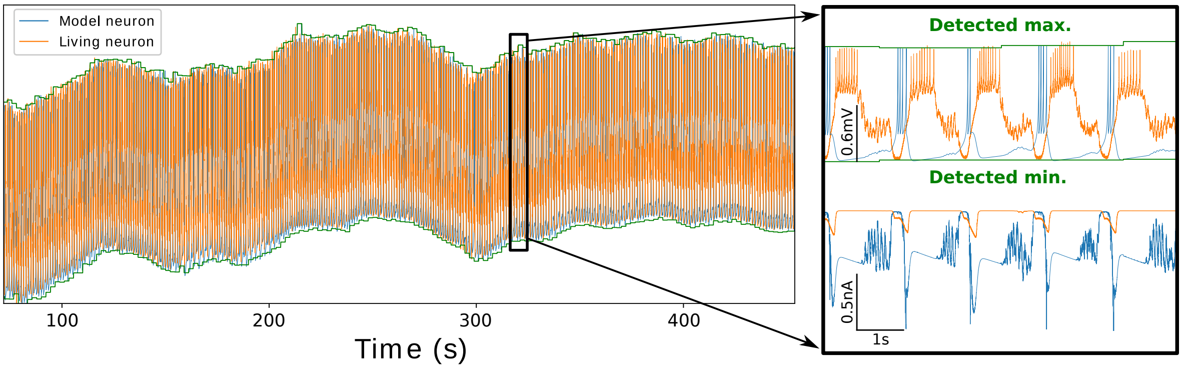

Fig. 8 Real-time drift adaptation during the experiment. Green lines show the maximum and minimum values of voltage amplitude. These values were reevaluated every two cycles of the hybrid circuit interaction based on the activity observed in the living neuron. Adaptation is done in model side. Currents: blue line input to model neuron and orange line input to living neuron.

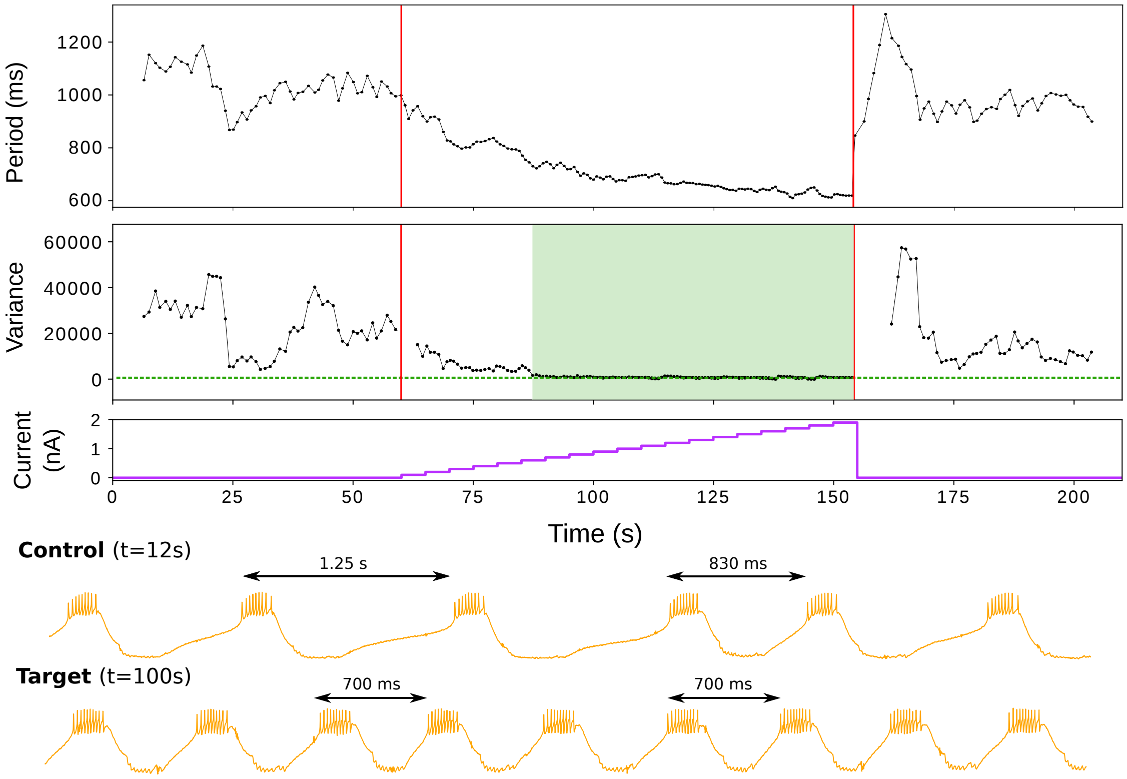

Fig. 9 Example of the event characterization algorithm for activity control: rhythm regularization of a LP neuron. The recording was monitored in real-time, as the system instantaneously measured the period from a predefined event –the burst beginning– and calculated its variance within a time window of 5 bursts. Red lines indicates the start and the end of the current injection. The region where the regularization target goal was reached is indicated in green.

3.6 Automatic mapping

The same approach discussed in the previous sections can be used to perform automatic parametric searches, and thus to achieve automatic mapping of the hybrid circuit dynamics in relation to a predefined goal. To illustrate the concept of automatic mapping, we developed a protocol to look for a dynamical invariant in a hybrid circuit. A dynamical invariant is defined as a preserved relationship between time intervals that define a sequence in a neural rhythm (Elices et al, 2019). Dynamical invariants are preserved cycle-by-cycle, even during transients. In our validation example we mapped the presence of a linear relationship, i.e. the invariant, between the interval defined by the beginning of the bursting activity of the two neurons (first to first spike

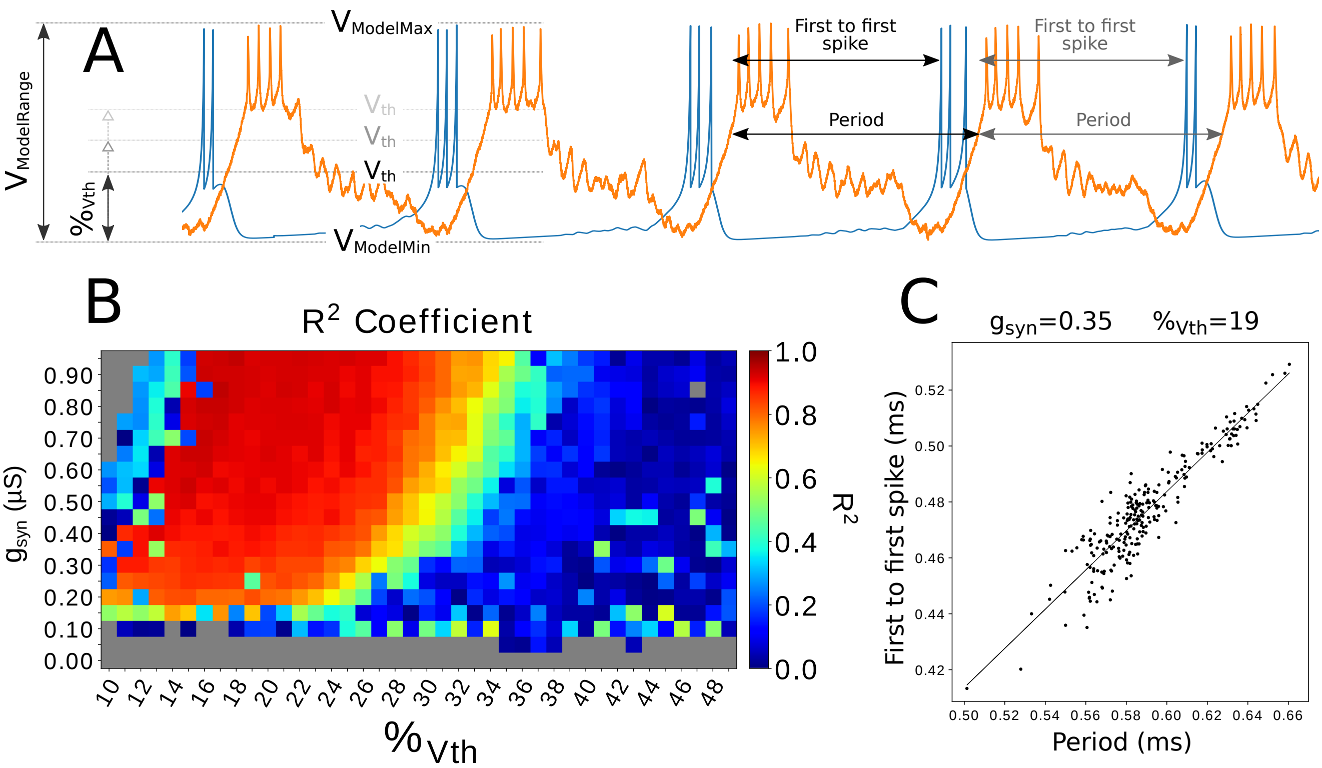

Fig. 10 Automatic mapping of the synaptic parameters that lead to a dynamical invariant between a living LP neuron and an Izhikevich model. Panel A illustrates the time intervals that build the invariant and the meaning of the $\%_{Vth}$ parameter (see Eqns. 1 and 2). The dynamical invariant in this case is a linear relationship between the interval defined by the beginning of the bursting activity of the two neurons (living and artificial) and the instantaneous period of the sequence in a hybrid circuit. The map in Panel B represents the R^2 of the linear regression for each explored combination of the the maximum conductance of the synapse g_{syn} and the presynaptic voltage threshold V_{th}. Red regions correspond to the presence of a dynamical invariant between the living and model neuron. Grey regions represent configurations where antiphase behaviour is not achieved. For the second parameter the values were calculated using a percentage $\%_{Vth}$ as show in section 3.6 and formula 2. Panel C shows a representative example of the dynamical invariant.

interval between the living and model neurons) and the instantaneous period of their sequence in the hybrid circuit (see Panel A in Fig. 10). The invariant can only occur when both neurons are firing in antiphase. The presence of the invariant was evaluated with the R^2 values of the linear regression between the considered time intervals.

To construct the map, we built a hybrid circuit by connecting the LP neuron of the pyloric CPG to a Izhikevich model neuron (Izhikevich, 2003) with a fast graded chemical synapse (Golowasch et al, 1999) by injecting the following current into the model:

$$I_{syn} = \frac{g_{syn}\ (V_{ModelN} - E_{syn})}{1 + exp(s(V_{th} - V_{LivingN}))} \qquad (1)$$

Note that this graded synaptic current depends on both the presynaptic and the postsynaptic potentials. Here, the presynaptic potential $V_{LivingN}$ is scaled to the neuron model range and the presynaptic parameters s and V_{th} are adjusted accordingly.

The two parameters used to build the map and explore the presence of a dynamical invariant in this circuit were the maximum conductance of the synapse g_{syn} and the voltage threshold for the release of the graded synapse V_{th}. The presynaptic voltage threshold V_{th} values were expressed in percentage as illustrated in Fig. 10A and defined as follows:

$$V_{th} = V_{ModelMin} + \%_{Vth} \cdot V_{ModelRange} \qquad (2)$$

The values of the synaptic parameters s and E_{syn} were adjusted as a function of the model voltage range as follows:

$$s = a \cdot V_{ModelRange} \qquad (3)$$

$$E_{syn} = V_{ModelMin} - b \cdot V_{ModelRange} \qquad (4)$$

where a is a parameter to scale the contribution of the model range to set the s value, and b is a parameter to scale the contribution of the model voltage range to the synaptic reversal potential E_{syn}. To built the map shown Fig. 10B we used $a = 0.05$ and $b = 0.15$.

The protocol automatically explored g_{syn} and $\%_{Vth}$ and built the map for the presence of the invariant show in Fig. 10C. Algorithm 6 was used to measure $V_{ModelMin}$ and $V_{ModelRange}$ as illustrated in Fig. 10A.

Panel B in Fig. 10 illustrates the result of this process. For all pairs of parameters shown in the map, the desired antiphase regime was reached when the color is not grey in this figure. The dynamical invariant was only present for high correlation values represented in red ($R^2 > 85\%$). To generate the map, time events were detected with the tools described in the time scaling subsection (Alg. 3 and 4 for event characterization and detection) in order to determine the duration of the intervals defined by the beginning of the burst of the living and the artificial neuron and the instantaneous period of the living neuron. Finally, each R^2 value was calculated with the obtained intervals for each combination of the explored synaptic parameters. Panel C shows the linear invariant for a representative case in the map.

4 Discussion

Hybrid circuits are built by connecting living and model neurons. These circuits have a lot of potential in neuroscience research, but require complex experiment-specific adaptations during their construction to work properly. In particular, parameters related to time and amplitude scaling of the models and of synaptic currents involved in the implementation must be evaluated for each preparation and setup. Typically, these adaptations are performed manually by the researcher and, thus, they are time consuming and often sub-optimal. Difficulties associated with the calibration procedures, together with those related to the real-time neuron and synapse model implementation are a major factor preventing the dissemination of hybrid circuit technology. In this work, we have developed a set of algorithms that address these issues and facilitate the automatic building of hybrid circuits.

The proposed algorithms have a wide range of applications to tune and control the behavior of the living circuit in an automated manner, but also to autonomously map the parameter space to achieve a predefined goal, and in general to explore circuit dynamics. It is important to emphasize that in many experiments hard real-time constraints are needed for an artifact-free implementation of hybrid circuits. The algorithms described in this paper have been implemented and validated in RTHybrid, an open-source platform, and can be easily generalized for other closed-loop interactions with the nervous system. Some of the algorithms can also be employed in model simulations alone to evaluate model candidates to be used in hybrid circuits, see also (Elices and Varona, 2015, 2017).

Although the current version of the proposed algorithms are not meant to modify online the DAQ board parameters or the amplifier DC offset, they can be adapted to take into account any automation by both devices. They could also be implemented together with electrode compensation or artifact removal software solutions (Brette et al, 2008; Samu et al, 2012; Gomez-Gonzalez et al, 2014).

Beyond electrophyology configurations, hybrid circuits can also be implemented using other alternatives such as optical recording and stimulation protocols, e.g. the ones used in optogenetics (Krook-Magnuson et al, 2013) and in neurotransmitter/neuromodulator microinjection protocols (Chamorro et al, 2012). Our algorithms for time and amplitude scaling of the models and, in general, the automation and calibration of goal-driven close-interactions also apply in any context regarding hybrid-circuits. Only the algorithm related to the amplitude scaling in the direction to the living neurons is specific of dynamic-clamp protocols.

Overall, the algorithms described in this paper and the associated standardized strategy to build hybrid-circuits can lead to the dissemination of the use of this technology, contribute to expand the life expectancy of the preparations and favor a new trend in automation of experimental work in neuroscience research.

5 Information Sharing Statement

The presented algorithms are implemented in RTHybrid. The source code is released under the GNU General Public License 3 and is freely available at `www.github.com/GNB-UAM/RTHybrid`. Experimental data is available under request.

Acknowledgements This work was supported by MINECO/FEDER PGC2018-095895-B-I00, DPI2015-65833-P, TIN2017-84452-R and ONRG grant N62909-14-1-N279.

References

Amaducci R, Reyes-Sanchez M, Elices I, Rodriguez FB, Varona P (2019) Rthybrid: A standardized and open-source real-time software model library for experimental neuroscience. Frontiers in Neuroinformatics 13:11, DOI 10.3389/fninf.2019.00011

Arroyo D, Chamorro P, Amigo JM, Rodriguez FB, Varona P (2013) Event detection, multimodality and non-stationarity: Ordinal patterns, a tool to rule them all? The European Physical Journal Special Topics 222(2):457–472, DOI 10.1140/epjst/e2013-01852-9

Arsiero M, Lüscher HR, Giugliano M (2007) Real-time closed-loop electrophysiology: towards new frontiers in in vitro investigations in the neurosciences. Archives italiennes de biologie 145(3):193–209

Bettencourt JC, Lillis KP, Stupin LR, White JA (2008) Effects of imperfect dynamic clamp: Computational and experimental results. Journal of Neuroscience Methods 169(2):282–289, DOI 10.1016/J.JNEUMETH.2007.10.009

Brette R, Piwkowska Z, Monier C, Rudolph-Lilith M, Fournier J, Levy M, Frégnac Y, Bal T, Destexhe A (2008) High-resolution intracellular recordings using a real-time computational model of the electrode. Neuron 59(3):379–391

Broccard FD, Joshi S, Wang J, Cauwenberghs G (2017) Neuromorphic neural interfaces: from neurophysiological inspiration to biohybrid coupling with nervous systems. Journal of neural engineering 14(4):41,002, DOI 10.1088/1741-2552/aa67a9

Brochini L, Carelli PV, Pinto RD (2011) Single synapse information coding in intraburst spike patterns of central pattern generator motor neurons. Journal of Neuroscience 31(34):12,297–12,306

Chamorro P, Levi R, Rodriguez FB, Pinto RD, Varona P (2009) Real-time activity-dependent drug microinjection. BMC Neuroscience 10(1):P296, DOI 10.1186/1471-2202-10-S1-P296

Chamorro P, Muñiz C, Levi R, Arroyo D, Rodriguez FB, Varona P (2012) Generalization of the dynamic clamp concept in neurophysiology and behavior. PLoS ONE 7(7), DOI 10.1371/journal.pone.0040887

Christini DJ, Stein KM, Markowitz SM, Lerman BB (1999) Practical Real-Time Computing System for Biomedical Experiment Interface. Annals of Biomedical Engineering DOI 10.1114/1.185

Couto J, Linaro D, De Schutter E, Giugliano M (2015) On the firing rate dependency of the phase response curve of rat purkinje neurons in vitro. PLoS computational biology 11(3):e1004,112

Destexhe A, Bal T (2009) Dynamic-Clamp: From Principles to Applications. From Principles to Applications 1:443, DOI 10.1007/978-0-387-89279-5

Elices I, Varona P (2015) Closed-loop control of a minimal central pattern generator network. Neurocomputing 170:55–62, DOI 10.1016/j.neucom.2015.04.097

Elices I, Varona P (2017) Asymmetry Factors Shaping Regular and Irregular Bursting Rhythms in Central Pattern Generators. Frontiers in Computational Neuroscience 11, DOI 10.3389/fncom.2017.00009

Elices I, Levi R, Arroyo D, Rodriguez FB, Varona P (2019) Robust dynamical invariants in sequential neural activity. Scientific Reports 9(1):9048, DOI 10.1038/s41598-019-44953-2

Ghigliazza RM, Holmes P (2004) Minimal Models of Bursting Neurons: How Multiple Currents, Conductances, and Timescales Affect Bifurcation Diagrams. SIAM Journal on Applied Dynamical Systems DOI 10.1137/030602307

Golowasch J, Casey M, Abbott LF, Marder E (1999) Network stability from activity-dependent regulation of neuronal conductances. Neural Computation 11(5):1079–1096, DOI 10.1162/089976699300016359

Gomez-Gonzalez J, Destexhe A, Bal T (2014) Application of active electrode compensation to perform continuous voltage-clamp recordings with sharp microelectrodes. Journal of Neural Engineering 11(5), DOI 10.1088/1741-2560/11/5/056028

Grashow R, Brookings T, Marder E (2010) Compensation for variable intrinsic neuronal excitability by circuit-synaptic interactions. Journal of Neuroscience 30(27):9145–9156

Hindmarsh JL, Rose RM (1984) A model of neuronal bursting using three coupled first order differential equations.

Hooper RM, Tikidji-Hamburyan RA, Canavier CC, Prinz AA (2015) Feedback Control of Variability in the Cycle Period of a Central Pattern Generator. Journal of Neurophysiology 114(5):jn.00,365.2015, DOI 10.1152/jn.00365.2015

Hull TE, Enright WH, Fellen BM, Sedgwick AE (1972) Comparing numerical methods for ordinary differential equations. SIAM Journal on Numerical Analysis 9(4):603–637

Izhikevich E (2003) Simple model of spiking neurons. IEEE Transactions on Neural Networks 14(6):1569–1572, DOI 10.1109/TNN.2003.820440, ArXiv

Kemenes I, Marra V, Crossley M, Samu D, Staras K, Kemenes G, Nowotny T (2011) Dynamic clamp with StdpC software. Nature protocols 6(3):405–417

Krook-Magnuson E, Armstrong C, Oijala M, Soltesz I (2013) On-demand optogenetic control of spontaneous seizures in temporal lobe epilepsy. Nature communications 4:1376, DOI 10.1038/ncomms2376

Le Masson G, Renaud-Le Masson S, Debay D, Bal T (2002) Feedback inhibition controls spike transfer in hybrid thalamic circuits. Nature 417(6891):854–858

Linaro D, Couto J, Giugliano M (2014) Command-line cellular electrophysiology for conventional and real-time closed-loop experiments. Journal of neuroscience methods 230:5–19

Linaro D, Couto J, Giugliano M (2015) Real-time Electrophysiology: Using Closed-loop Protocols to Probe

Neuronal Dynamics and Beyond. JoVE (Journal of Visualized Experiments) pp e52,320—-e52,320, DOI 10.3791/52320

Marder E, Calabrese RL (1996) Principles of rhythmic motor pattern generation. Physiol Rev 76:687–717

Mishchenko MA, Gerasimova SA, Lebedeva AV, Lepekhina LS, Pisarchik AN, Kazantsev VB (2018) Optoelectronic system for brain neuronal network stimulation. PLOS ONE 13(6):e0198,396, DOI 10.1371/journal.pone.0198396

Muñiz C, Arganda S, Rodriguez FB, de Polavieja GG, Varona P (2005) Realistic stimulation through advanced dynamic-clamp protocols. Lecture Notes in Computer Science 3561:95–105, DOI 10.1007/11499220_10

Muñiz C, Rodriguez FB, Varona P (2009) RTBiomanager: a software platform to expand the applications of real-time technology in neuroscience. BMC Neurosci 10(Suppl 1):P49, DOI 10.1186/1471-2202-10-S1-P49

Norman SE, Butera RJ, Canavier CC (2016) Stochastic slowly adapting ionic currents may provide a decorrelation mechanism for neural oscillators by causing wander in the intrinsic period. Journal of Neurophysiology DOI 10.1152/jn.00193.2016

Nowotny T, Varona P (2012) Dynamic Clamp, Springer Netherlands, Dordrecht, pp 613–621. DOI 10.1007/978-90-481-9751-4_223

Nowotny T, Varona P (2014) Dynamic Clamp Technique. In: Encyclopedia of Computational Neuroscience, Springer New York, New York, NY, pp 1–4, DOI 10.1007/978-1-4614-7320-6_126-2

Nowotny T, Zhigulin VP, Selverston AI, Abarbanel HDI, Rabinovich MI (2003) Enhancement of synchronization in a hybrid neural circuit by spike-timing dependent plasticity. Journal of Neuroscience 23(30):9776–9785, DOI 23/30/9776[pii]

Olypher A, Cymbalyuk G, Calabrese RL (2006) Hybrid Systems Analysis of the Control of Burst Duration by Low-Voltage-Activated Calcium Current in Leech Heart Interneurons. Journal of Neurophysiology DOI 10.1152/jn.00582.2006

Patel YA, George A, Dorval AD, White JA, Christini DJ, Butera RJ (2017) Hard real-time closed-loop electrophysiology with the Real-Time eXperiment Interface (RTXI). PLoS Computational Biology 13(5), DOI 10.1371/journal.pcbi.1005430

Pinto R, Elson R, Szücs A, Rabinovich M, Selverston A, Abarbanel H (2001) Extended dynamic clamp: controlling up to four neurons using a single desktop computer and interface. Journal of Neuroscience Methods 108(1):39 – 48, DOI https://doi.org/10.1016/S0165-0270(01)00368-5

Pinto RD, Varona P, Volkovskii AR, Szücs A, Abarbanel HD, Rabinovich MI (2000) Synchronous behavior of two coupled electronic neurons. Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 62(2):2644–2656, DOI 10.1103/PhysRevE.62.2644, 0001020

Prinz AA, Abbott L, Marder E (2004) The dynamic clamp comes of age. Trends in Neurosciences 27(4):218–224, DOI 10.1016/j.tins.2004.02.004

Prsa M, Galiñanes GL, Huber D (2017) Rapid Integration of Artificial Sensory Feedback during Operant Conditioning of Motor Cortex Neurons. Neuron 93(4):929–939.e6, DOI 10.1016/j.neuron.2017.01.023

Robinson HPC, Kawai N (1993) Injection of digitally synthesized synaptic conductance transients to measure the integrative properties of neurons. Journal of Neuroscience Methods DOI 10.1016/0165-0270(93)90119-C

Rulkov NF (2002) Modeling of spiking-bursting neural behavior using two-dimensional map. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 65(4)

Sakurai A, Katz PS (2017) Artificial synaptic rewiring demonstrates that distinct neural circuit configurations underlie homologous behaviors. Current Biology 27(12):1721 – 1734.e3, DOI https://doi.org/10.1016/j.cub.2017.05.016

Samu D, Marra V, Kemenes I, Crossley M, Kemenes G, Staras K, Nowotny T (2012) Single electrode dynamic clamp with StdpC. Journal of Neuroscience Methods 211(1):11–21, DOI 10.1016/j.jneumeth.2012.08.003

Selverston AI (2005) A neural infrastructure for rhythmic motor patterns. Cellular and molecular neurobiology 25(2):223–244

Sharp AA, O'Neil MB, Abbott LF, Marder E (1993) The dynamic clamp: artificial conductances in biological neurons. Trends in Neurosciences 16(10):389–394, DOI 10.1016/0166-2236(93)90004-6

Szücs A, Varona P, Volkovskii AR, Abarbanel HDI, Rabinovich MI, Selverston AI (2000) Interacting biological and electronic neurons generate realistic oscillatory rhythms. Neuroreport 11(3):563–569, DOI 10.1097/00001756-200002280-00027

Varona P, Torres JJ, Abarbanel HDI, Rabinovich MI, Elson RC (2001) Dynamics of two electrically coupled chaotic neurons: Experimental observations and model analysis. Biological Cybernetics 84(2):91–101, DOI 10.1007/s004220000198

Varona P, Arroyo D, Rodriguez FB, Nowotny T (2016) Chapter 6 - online event detection requirements in closed-loop neuroscience. In: Hady AE (ed) Closed Loop Neuroscience, Academic Press, San

Diego, pp 81 – 91, DOI https://doi.org/10.1016/B978-0-12-802452-2.00006-8

Wang S, Chandrasekaran L, Fernandez FR, White JA, Canavier CC (2012) Short conduction delays cause inhibition rather than excitation to favor synchrony in hybrid neuronal networks of the entorhinal cortex. PLoS computational biology 8(1):e1002,306

Yarom Y (1991) Rhythmogenesis in a hybrid system-interconnecting an olivary neuron to an analog network of coupled oscillators. Neuroscience 44(2):263–275