A generic framework for real-time multi-channel neuronal signal analysis and sub-millisecond latency feedback generation

Christoph Zrenner[1,2§], Danny Eytan[1], Hans-Peter Thier[2], Shimon Marom[1]

[1]Technion – Israel Institute of Technology, Haifa 32000, Israel

[2]Hertie-Institute for Clinical Brain Research, University of Tübingen, Otfried-Müller-Strasse 27, 72076 Tübingen, Germany

[§]Corresponding author

Email addresses:

CZ: zrenner@technion.ac.il

DE: dannye@technion.ac.il

H-PT: hans-peter.thier@uni-tuebingen.de

SM: marom@technion.ac.il

Abstract

Background
Different modules of the neural circuitry interact with each other and through the motor-sensory loop with the environment, forming a complex dynamic system. The ability to access and modify the flow of information at the level of neural modules electrically, bi-directionally and in real-time is a fundamental requirement for any application seeking to understand, restore, or modulate function.

Methods
A set of generic tools is presented that allow computationally demanding short-latency bi-directional interactions to be realized in *in-vivo* and *in-vitro* preparations using standard PC data acquisition hardware and software (Mathworks Matlab and Simulink). The set of tools developed here is freely available for download. A commercially available 60-channel extracellular multi-electrode recording and stimulation set-up connected to an *ex-vivo* developing cortical neuronal culture is used as a model system to validate the method.

Results
We demonstrate the efficacy of this system through (1) the application of a precisely timed "interference stimulation" contingent on a pre-defined spontaneous spatio-temporal action potential pattern and (2) by providing continuous feedback as a function of the ongoing neuronal activity. The computational power of such an approach allows complex high-bandwidth (10 megabit per second) data analysis in real-time and the selective generation of a feedback signal with reliable latencies in the sub-millisecond range.

Conclusions

The approach is well suited for the generic implementation of a bi-directional high-bandwidth bio-electric interface in basic science or clinical research where deterministic timing in the processing of information and the generation of a short-latency response is required. Specific applications may range from the detection and prevention of predefined patterns of activity at their onset (e.g. neuronal seizures or cardiac arrhythmia) to the development of brain-computer interfaces in general.

Background

The microscopic visualization of a living neuronal tissue, combined with the ability to interact with nerve cells through localized chemical and electrical means, enables a description of both the circuitry and the cellular and synaptic properties of a nerve cell network. Whereas real-time interaction with single neurons through a dynamic clamp set-up is firmly established as a method for the study of the functional anatomy of a particular circuit [1, 2, 3], the network level poses a number of additional challenges. In this work, we demonstrate a low-cost generic approach of providing complex real-time feedback to nervous tissues, catering to incorporation of dynamic interactions between the organism and its environment.

Until recently, it was necessary to develop custom software and hardware to set up experimental paradigms that incorporated feedback with sufficiently short latencies for realistic motor-sensory loops [4, 5, 6, 7] (For an extensive overview of the problem see Rutten [8]). This constituted both a significant investment in time and funds, but also limited the scope of the methods in being applicable only for the specific set of experiments they were designed for. In recent years, however, the increasing availability of computational power in low-cost desktop computers, has made it possible to perform complex calculations on high-bandwidth signals with sub-millisecond latencies using standard off-the-shelf tools. Here, we demonstrate how the Mathworks Matlab Simulink programming framework can be adapted in a generic way (1) to acquire multi-channel electrophysiological signals from a neuronal experimental preparation, (2) to perform complex data analysis in real-time and, (3) to provide contingent multi-channel feedback stimulation with sub-millisecond latency. Our software programs are freely available and can be adapted to a wide range of closed-loop experimental paradigms. Knowledge of procedural programming is not required as the real-time stage consists of a high-level visually developed Simulink signal processing model.

Methods

Basic Principle

The feedback system consists of two standard desktop computers: A self-contained "target" PC, running a real-time operating system kernel (Mathworks Real-Time Workshop xPC Target) is connected to all the instruments relevant for experimental measurement and control. A visually programmed signal based Mathworks Simulink model is compiled to run on this target PC to acquire data, analyse spike-trains and generate feedback signals at synchronous iteration steps paced by an interrupt at a rate of 50-500 μs (depending on the number of channels, required sampling rate and feedback processing complexity). Another, the "host" PC, runs Mathworks Matlab and an asynchronous Simulink interface providing the experimenter with control of parameters, as well as on-line visualization and data archiving. The two parts of the system communicate through a 100 MBit TCP/IP (UDP for data streaming) network connection (see Fig. 1).

The real-time target kernel supports a large number of hardware interfaces natively including hundreds of multifunction analogue and digital input/output boards from various manufacturers in addition to the standard PC interfaces (serial and parallel ports, audio input and output, etc.). This makes it a versatile platform to access and control all aspects of an experimental set-up automatically. Analogue inputs can be used for data acquisition; digital outputs and RS232 serial lines to program and control peripherals such as an stimulation head-stage and stimulus generator for electrical feedback or high-precision peristaltic pumps to provide neurochemical feedback. The digital output of the multifunction board can be used to drive stepper-motors, light and sound generators, and many other kinds of electronic instruments.

Model Implementation

In this and the following sections, our concrete implementation is described in some detail. A multi-channel electrode array [9, 10] interface with a primary neuronal culture was chosen as a well established experimental model system. Such cultured nerve cell substrate can be patterned structurally [11], exhibit considerable potential for plasticity [12, 13, 14], and information can be transferred to and from the network in a well-defined manner. The model implementation is thus sufficiently complex to demonstrate the capacity of our approach but remains generic enough to serve as a straightforward starting-point for the development of custom *in-vitro* or *in-vivo* applications.

Cortical tissue was taken from newborn rats and plated on commercially available substrate integrated 60-electrode array dishes (Multichannelsystems GmbH, electrode diameter: 30μs, electrode spacing: 200μm or 500μm, material: TiN) using standard procedures described elsewhere [15] and maintained at 37ºC in an atmosphere of 5% CO2 and 95% air. Experiments were performed after a 21 day incubation period, during which time a mature densely inter-connected network forms that spontaneously exhibits a characteristic bursting activity pattern [16, 17], that has been likened to neuronal "avalanches" in neocortical circuits [18, 19]. The dissociated neuronal cultures develop to exhibit a rich repertoire of spontaneous spatio-temporal activity [20, 21] that may be classified by various methods. Electrical stimulation at different electrodes, or using different amplitudes and frequencies have distinguishable effects [22].

Action potentials are recorded from an array of 60 substrate integrated extracellular TiN electrodes using a head-stage and a filter-amplifier (Multichannelsystems MEA1060-BC and FA60, bandwidth: 300 Hz to 3 kHz, amplification: 1100). The signals are then acquired by a standard desktop PC (AMD Athlon 32 bit Processor, Asus Motherboard, Mathworks xPC Target operating system) through a supported PCI data acquisition card (United Electronic Instruments PD2-MF-64-2M, sampling rate: 16 kHz per channel, voltage range: -5 V to +5 V, resolution: 12 bit). Every 500 μs, an interrupt service routine is executed to process the acquired "frame" of data (8×60 12 bit samples) while the card is continuously acquiring the next frame of data in the background.

Action potentials are detected and aligned by a threshold criterion and further processed by the feedback generation system. More complicated detection algorithms for either individual events (e.g. spikes) or spatio-temporal composites may be used without compromising the performance of the system. A tunable refractory period (2.5 ms) prevents multiple detection of single events that cross the threshold several times.

Feedback Stimulation

A two channel stimulus generator (Multichannelsystems STG1002, biphasic symmetric unipolar voltage stimulation, amplitude: 100 mV to 900 mV, duration: 200 μs negative pulse, 200 μs positive pulse) is connected to the two stimulus inputs of the head-stage. The head-stage is programmable, so that two independent subsets of any of the 60 recording electrodes can be dynamically selected for stimulation. During stimulation, a TTL trigger and blanking signal lasting 1ms is applied to the head-stage, electronically switching the designated stimulus electrodes to the stimulus generator lines and all other electrodes to ground. In this way, stimulation artefacts are almost completely suppressed allowing action potentials to be recorded again within 2 ms after stimulation.

The stimulus generator and the head-stage are two independent instruments, each programmed and triggered by the real-time processor. The subset of stimulating electrodes is dynamically selected using a serial command string of 40 to 60 bytes length (depending on the complexity of the desired waveform). Likewise, the applied waveform is dynamically controlled using a serial command string of 54 bytes length. It therefore takes less than 10 ms to transmit the commands for both selecting a new set of stimulating electrodes and changing the stimulus waveform by reprogramming the respective instruments through two independent serial lines. The stimulus control is integrated in the real-time processor which allows a high bandwidth feedback signal to be delivered as a function of the ongoing activity at a rate of up to 100 Hz (the stimulus generator STG1002 is limited to a maximum triggering frequency of 50 Hz). This means that a stimulus of dynamically varying form can be delivered through a dynamically varying set of electrodes every 10 ms (may be improved by using different apparatus and/or firmware).

The above feedback procedure may be adapted to other modalities, e.g. activity-dependent application of neuromodulators, sound, light, motion, and so forth.

Real-Time Data Processing

The Mathworks Simulink platform provides a wide range of commercially available real-time signal processing and analysis block sets, including adaptive filters, statistical operations, look-up tables and Fourier and wavelet analysis. Complex multidimensional feedback loops can be designed very easily and visually by “dragging and dropping” the appropriate blocks inside the model and connecting them with signal lines. For operations that are not supported out-of-the-box, standard procedural code (Matlab M code or C code) can be embedded inside function blocks. In our model application, we analyse spontaneous spatio-temporal activity patterns (monitored by a 60-channel electrode array) as they occur. The specific realisation described here is aimed at detecting the relative timing of the first-spike of each electrode in relation to the onset of a Network Spike (NS) [23]. Network spikes are detected by a threshold criterion, and the recruitment order is extracted in real-time from the delays to first spikes. The resulting recruitment order is compared to pre-defined templates for on-line classification, and a corresponding output is generated using relevant actuators or stimulus generators.

The Simulink model shown in Fig. 2 shows an example for an on-line detection of electrode onset order.

On-Line Visualization and Control

A limited amount of visualization can be done on the target (Fig. 3), but it is the function of the “host” PC running Matlab to receive and archive the data streams from the real-time system. Spike shapes are extracted from the raw data and stored separately for post processing. If a feedback signal is generated dynamically on the target, the parameters of the feedback signal (electrode, time, amplitude) are also transmitted and logged as they occur. In this way, the “host” PC serves as an independent visualization and archiving station allowing the experimenter to observe the behaviour of the neuronal preparation and state of the feedback system without disturbing its operation.

The “host” PC can be used to modify the parameters of the real-time system and to execute different experimental conditions consecutively, making experimental control fully scriptable. The real-time “target” PC can run continuously for days and weeks interacting with a chronic experimental preparation while the “host” PC logs relevant aspects of the information flowing between preparation and experimental set-up.

Results

Overview

In what follows we demonstrate a number of experimental paradigms that can be realized with the method described. Firstly, we show how spontaneously active multi-channel spike-trains can be monitored while detecting events in real-time by a threshold criterion. Raw data in the region of interest before and after the time of threshold crossing can be stored for on-line or off-line analysis and categorization. Secondly, we demonstrate how an electric stimulus can be used to interact and interfere with the ongoing activity, both as a “single-shot” and as a modulated stimulus train. These applications were chosen because they are challenging to implement in a generic fashion, and highly relevant for experimental paradigms aimed at characterization and modulation of complex interactions in neural systems.

Monitoring and Categorization of Multidimensional Signals

The spontaneous activity exhibited by a dissociated neuronal culture after 3 weeks in vitro consists of intermittent bursts of activity (“network spikes”) lasting around 100 ms each that are separated by several seconds of relatively sparse activity. As shown in Fig. 4 (middle and right panel), the spontaneously occurring multi-channel spike-train patterns of individual network spikes are highly variable and short-lasting, properties that are well suited for demonstrating real-time monitoring and classification capacities.

As demonstrated in Fig. 4, the set-up is capable of processing multidimensional data at a rate of 11.5 MBit per second, generate a trigger according to a pre-defined condition and hold the multi-channel data for immediate or subsequent analysis. The real-time processor operates in discrete execution steps paced every 500 μs. During each step, a sliding window of variable size, containing data history, is available for inclusion in the analysis at the current step. It is important to note that the system is capable of interacting with the ongoing neuronal activity at any point during a network spike using the available preceding data (the build-up phase of the activity) to determine the properties of the feedback signal at a resolution and minimum latency of 500 μs.

Spike-train Triggered Interference Stimulation

It is a frequent requirement in biomedicine applications to suppress or otherwise counteract an undesired event, such as synchronized brain activity or re-entrant circular cardiac muscle activity. Here also, a dissociated neuronal culture is used as a model system to demonstrate the efficacy of the method in selectively interfering with neuronal activity using electric stimulation.

The feedback generating model consists of two distinct stages: in the first stage, the 60 channel signal is continuously monitored, spikes are detected and summed across channels over a sliding window of 25 ms. Following an earlier study [23], network spike onset was determined by using a threshold of 15 channels that are active within that 25 ms time window. A second stage then analyses the spatio-temporal pattern within a variable period of time that begins 50 ms before the threshold crossing, and extends into the time envelope of the network spike as required. If a predefined property of the spike-train is detected (in the example of Fig. 5, left panel, the order "1-2-3" was matched) a stimulus is provided at a predefined time in relation to any given temporal reference point (e.g. the network spike threshold crossing or the matching of the spike pattern). This experiment therefore demonstrates the possibility of interfering specifically with a complex activity pattern. The post stimulus time histogram (Fig. 5, right panel, inset) shows one particular channel, in which the feedback stimulus reliably evokes actual spikes. This is thus an example for a short latency closed loop paradigm where a particular activity pattern leads to an interference stimulus which in turn evokes specific activity.

Continuous Variable Feedback

In this example, we consider the case where both network activity as well as feedback signals are continuous rather than discrete. Network activity is expressed in terms of number of individual action potentials recorded throughout the electrode array within a 25 ms sliding time window. The amplitude of the feedback stimulus was matched to the level of network activity following an arbitrarily pre-defined function. In providing a non-linear delayed feedback signal we implement an algorithm based on the principles described by Popovych *et al* [24]: (1) the instantaneous overall activity of the network is estimated by summing events inside a sliding window, (2) an imaginary part is generated for this signal through a Hilbert transform, (3) the result is squared and multiplied with the delayed complex conjugate of the same signal, (4) the real part of the resulting complex signal is taken to determine the feedback amplitude. To approximate a continuous varying feedback signal, a stimulation of variable magnitude between 100 and 900 mV is applied every 20 ms.

Since the Simulink programming environment used in the approach presented here is designed for control systems, it is a trivial matter to implement such mathematical signal processing algorithms efficiently (blocks shown in Fig. 6 left panel, bottom) which would ordinarily require many lines of procedural code.

In this example, feedback is generated by setting the stimulus amplitude of a programmable stimulus generator in real-time through a serial interface. It would of course be equally possible to use an analogue voltage or stepper motor pulse as an interface driving a pump with neuroactive chemicals, changing the velocity of a moving visual scene, controlling the loudness of a tone or almost any other sensory or neurophysiological feedback signal generator. We chose to demonstrate a serial command program because this can frequently be challenging to implement in real-time with other methods.

Discussion

The rapid advances in computer technology over the past years have made it feasible to handle a considerable amount of multi-channel long-term recording data. Advanced and well-supported software tools are available [25] for the specific task of analysing large matrices of spike data *post-hoc* using Mathworks Matlab. Here, we present a framework that complements this approach with with the ability to continuously interact with the experimental preparation while at the same time recording all relevant data for subsequent evaluation. Because we are building on the same underlying technology (Mathworks Matlab), data can be analysed, visualised and archived on-line without any conversion issues.

Beyond the ease of integration with post-processing and control scripts, we consider the main strengths of our approach to be its generic applicability, low cost and relative ease of implementation. In this, it stands out among the many alternative closed-loop approaches, ranging from direct electric circuits through embedded programmable digital signal processor (DSP) architectures all the way to a real-time operating system on standard PC hardware (e.g. real-time Linux among many other free and commercial platforms).

Whereas commercial packages can be easy to get started with, because of the considerable choice and frequently expensive hardware and software requirements it can quickly become a frustrating undertaking when it turns out that the particular approach has inherent limitations regarding its processing capabilities, communication with peripherals, integration with post-processing software, or the availability of interfaces for experimental control scripts. At the other end of the spectrum, custom software for real-time Linux can represent a low-cost and highly flexible alternative that has been successfully employed for complex closed-loop paradigms previously [26, 27]. Nevertheless, developing and maintaining low-level real-time code requires a significant investment of time even for experienced engineers. Beyond the barrier of a steep learning curve and complex development process, custom solutions present with another inherent issue: While, in principle, there is a great deal of flexibility and freedom when using "ones own" framework, the practice is that home-made tools tend to depend on the original developers for maintenance and may be difficult to generalize beyond the very specific task they were designed for.

Here, we are proposing a third, "middle ground", option, whose high-level real-time software models maintain the flexibility of custom-made solutions while leveraging the capabilities of a well-supported commercial rapid prototyping environment. As a conscious design decision, we have chosen an architecture whereby a self-contained real-time processing and interfacing stage is completely separated from an independent visualization, archiving and experimental control flow and automization stage. Data is streamed from the real-time target to the host so that spike trains can be continuously visualized and archived for many weeks without gaps.

In terms of performance, our approach improves on previous methods in several aspects: (1) The feedback latency and jitter is reduced to a deterministic 500 µs while allowing for considerable processing at 10 MBit/s data throughput. If necessary, computationally intense operations can be executed in a concurrent task that is paced at a slower rate. The feedback latency can be further reduced to less than 100 μs if necessary. (2) Use of the high-level Mathworks Simulink visual signal processing and programming language makes this approach broadly accessible and allows rapid development of complex real-time feedback paradigms. (3) Direct integration with Mathworks Matlab enables full scriptability of experimental control, on-line visualization, archiving and immediate analysis of streamed data.

While setting up the Mathworks Simulink rapid prototyping environment for real-time operation can be somewhat involved, our freely available scripts (downloadable or provided by email on request) provide a generic template for interfacing neural systems and can serve as a straightforward starting point. They can be easily adapted for a wide range of applications at the level of whole organisms, large or small scale networks, single neurons, and even the three dimensional volume of the dendritic tree. On the recording side, EEG, MEG or MRI signals can be employed just as easily as local field potentials or multi-unit spike-train activity all the way down to optical monitoring of voltage sensitive dyes. Feedback stimulation is possible electrically, neurochemically or optically and clinical applications include EEG biofeedback as well as the control of bioelectric neuro-interfaces. Due to the openness and accessibility of this approach, complex data analysis and feedback algorithms can be rapidly co-developed and shared among the community for straightforward "plugging" into this framework.

In conclusion, while real-time programming remains a non-trivial matter, we consider it very much a worthwhile investment for applications that benefit from the bi-directional control of the flow of information. Working with the template code provided here, setting up custom feedback loops for *in-vitro* or *in-vivo* applications should be a matter of a few thousand Euro in hardware (see table 1 for a list of parts) and a few weeks in training. We hope that this demonstration of the relative ease by which versatile dynamic feedback interactions can be constructed and integrated using standard hardware and software, the inclusion of a “closed loop” approach in the experimental toolbox may become attractive to a wider range of scientists in basic and clinical neuroscience.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

CZ developed the software, carried out the experiments, and drafted the manuscript. DE supervised the overall design and construction of the model set-up, conceived of the continuous feedback paradigm, participated in validating the hardware and software experimentally, and helped to draft the manuscript. H-PT coordinated the experiments performed in Tübingen. SM conceived of the study and coordinated the experiments performed in Haifa. All authors read and approved the final manuscript..

Acknowledgements

The authors wish to thank Dr. Volker Gauck (Hertie Institute, Tübingen) for experimental guidance and his contribution to the analysis and interpretation of the data, Eleanora Lyakhov (Technion, Haifa) for the preparation and maintenance of neuronal cultures in Haifa, Vladmir Lyakhov (Technion, Haifa) for invaluable input regarding the data acquisition stage of the set-up in Haifa, as well as Dr. Thomas Ott and Thomas Henneck (IZKF, Tübingen) and Ute Großhennig (Hertie Institut, Tübingen) for their assistance with the preparation of neuronal cultures in Tübingen. We are also grateful to Karl-Heinz Boven (Multi Channel Systems MCS GmbH, Reutlingen) for supplying the necessary parts and providing implementation advice, as well as to Gordon Weast (The Mathworks) for technical support regarding the real-time processor hardware interface.

CZ obtained funding from the Böhringer-Ingelheim Foundation, through a Max-Planck Minerva Short Term Research Grant and from the MFG Baden-Württemberg Foundation. DE holds an Aly Kaufmann Postdoctoral Fellowship, SM is funded by grants from the Israel Science Foundation, DE and SM received funding from the Technion Lokey Center. H-PT is funded through the German Research Foundation Collaborative Research Centre SFB550.

References

1. Robinson, HP: **Conductance injection.** *Trends in Neurosciences* 1994, **17**:147-148.
2. Sharp AA, O'Neil MB, Abbott LF, Marder E: **The dynamic clamp: artificial conductances in biological neurons.** *Trends Neurosci* 1993, **16:**389-394
3. Prinz AA, Abbott LF, Marder E: **The dynamic clamp comes of age.** *Trends Neurosci* 2004, **27:**218-224.
4. Ehud Ahissar, David Kleinfeld: **Closed-loop Neuronal Computations: Focus on Vibrissa Somatosensation in Rat.** *Cereb Cort* 2003, **13**(1)**:**53-62
5. Novellino A, D'Angelo P, Cozzi L, Chiappalone M, Sanguineti V, Martinoia S: **Connecting Neurons to a Mobile Robot: An In Vitro Bidirectional Neural Interface.** *Computational Intelligence and Neuroscience* 2007, Article ID 12725
6. Karniel A, Kositsky M, Fleming KM, Chiappalone M, Sanguineti V, Alford ST, Mussa-Ivaldi FA: **Computational analysis in vitro: dynamics and plasticity of a neuro-robotic system.** *J Neural Eng* 2005, **2:**250-265.
7. Wolpawa JR, Birbaumer N, McFarland DJ, Pfurtscheller G, Vaughan TM: **Brain-computer interfaces for communication and control.** *Clinical Neurophysiology* 2002, **113:**767-791.
8. Rutten WLC: **Selective electical interfaces with the nervous system.** *Annu Rev Biomed Eng* 2002, **4:**407-452.
9. Stett A, Egert U, Guenther E, Hofmann F, Meyer T, Nisch W, Haemmerle H: **Biological application of microelectrode arrays in drug discovery and basic research.** *Anal Bioanal Chem* 2003, **377:**486-495.

10. Frey U, Sanchez-Bustamante CD, Ugniwenko T, Heer F, Sedivy J, Hafizovic S, Roscic B, Fussenegger M, Blau A, Egert U, Hierlemann A: **Cell Recordings with a CMOS High-density Microelectrode Array.** *Conf Proc IEEE Eng Med Biol Soc* 2007, **1:**167-170.

11. Chang JC, Brewer GJ, Wheeler BC: **Modulation of Neural Network Activity by Patterning.** *Biosensors and Bioelectronics* 2001, **16:**527-533.

12. Shahaf G, Marom S: **Learning in Networks of Cortical Neurons.** *J Neurosci* 2001, **21**(22)**:**8782-8788.

13. Eytan D, Brenner N, Marom S: **Selective adaptation in networks of cortical neurons.** *J Neurosci* 2003, **23**(28)**:**9349-9356.

14. Van Pelt J, Vajda I, Wolters PS, Corner MA, Ramakers GJ: **Dynamics and plasticity in developing neuronal networks in vitro.** *Prog Brain Res* 2005, **147:**173-188.

15. Eytan D, Minerbi A, Ziv N, Marom S: **Dopamine-induced dispersion of correlations between action potentials in networks of cortical neurons.** *J Neurophysiol* 2004, **92**(3)**:**1817-1824.

16. Maeda E, Robinson HP, Kawana A: The mechanisms of generation and propagation of synchronized bursting in developing networks of cortical neurons. J Neurosci 1995, 15(10):6834-6845.

17. Corner MA, van Pelt J, et al.: **Physiological effects of sustained blockade of excitatory synaptic transmission on spontaneously active developing neuronal networks - an inquiry into the reciprocal linkage between intrinsic biorhythms and neuroplasticity in early ontogeny.** *Neurosci Biobehav Rev* 2002, **26:**127-185.

18. Beggs JM, Plenz D: **Neuronal avalanches in neocortical circuits.** *J Neurosci* 2003, **23**(35)**:**11167-11177.

19. Plenz D, Thiagarajan TC: **The organizing principles of neuronal avalanches: cell assemblies in the cortex?** *Trends Neurosci* 2007, **30:**101-110.

20. Van Pelt J, Corner MA, Wolters PS, Rutten WLC, Ramakers GJA: **Long-term stability and developmental changes in spontaneous network burst firing patterns in dissociated rat cerebral cortex cell cultures on multielectrode arrays.** *Neurosci Lett* 2004, **361**(1-3)**:**86-89.

21. Wagenaar DA, Pine J, Potter SM: **An extremely rich repertoire of bursting patterns during the development of cortical cultures.** *BMC Neurosci* 2006, **7:**11.

22. Marom S, Shahaf G: **Development, learning and memory in large random networks of cortical neurons: lessons beyond anatomy.** *Quart Rev Biophys* 2002, **35:**63-87.

23. Eytan D, Marom S: **Dynamics and Effective Topology Underlying Synchronization in Networks of Cortical Neurons.** *J Neurosci.* 2006, **26**(33)**:**8465-8476.

24. Popovych OV, Hauptmann C, Tass PA: Control of neuronal synchrony by nonlinear delayed feedback. Biol Cybern 2006, 95:69-85.

25. Egert U, Knott T, Schwarz C, Nawrot M, Brandt A, Rotter S, Diesmann M: **MEA-tools: an open source toolbox for the analysis of multielectrode-data with MATLAB.** *J Neurosci Meth* 2002, **177:**33-42.

26. Wagenaar DA, DeMarse TB, Potter SM: **MEABench: A toolset for multi-electrode data acquisition and on-line analysis.** *Proc 2nd Intl IEEE EMBS Conf on Neural Eng* 2005, 518-521.

27. Wagenaar DA, Madhavan R, Pine J, Potter SM: **Controlling bursting in cortical cultures with closed-loop multi-electrode stimulation.** *J Neurosci* 2005, **25**(3)**:**680–688.

Figures

Figure 1 - Overall Set-up

The set-up consists a closed-loop real-time stage (right side) that streams relevant data to a controller for visualization and archiving (left side). In our model implementation, a 60-channel multi-electrode interface head-stage (Multi Channel Systems MEA1060-BC) is used to interface with a neuronal culture (Neural System). Analogue voltage signals are recorded simultaneously from all channels after passing through a filter-amplifier using a multifunction data acquisition card (Signal Recorder) at a sample rate of 16 kHz per channel. A standard PC (Real-Time Processor) is configured to run a Mathworks Simulink Real-Time model paced at 500µs iteration steps to process and analyse the data. If predefined (dynamic) conditions are met, a feedback stimulus is triggered after a sub-millisecond resolution delay (Stimulus Generator). The stimulation parameters can be modified in response to the ongoing activity using a serial instrument control protocol. The analogue "raw voltage" signal, along with other information about the processed data, is streamed via a UDP local area network connection to a host PC running Matlab (Control Computer) for on-line visualization and archiving. The real-time processor PC and parameters of the feedback model are also controlled through the network from the control computer.

Figure 2 - Exemplary Simulink model for the real-time detection of channel onset order during a network spike

Analogue voltage signals from the data acquisition card (10 channels shown) are passed through a threshold detection block which signals a binary "1" on the downward crossing of the negative voltage spike threshold. This 10-element binary vector signal is summed up inside a sliding window of 25 ms and passed again through a threshold detection block that signals the onset of a network spike when a predefined number of spikes have been recorded across all channels in the preceding 25 execution steps (sample model paced at 1 kHz). Within the processing branch shown in the lower section, the spike train is analysed according to the order of onset in a window of 50 ms. The 10-element vector is buffered, yielding a 50x10 element binary matrix (50 one-millisecond time steps into the past, 10 channels), with the earliest events of each channel (column) on the top rows. This binary matrix is multiplied element-wise with a constant monotonically decreasing "ramp"-matrix with rows yielding a result matrix of which the magnitude of each non-zero element indicates how far in the past the spike occurred. By sorting the largest (time of first spike) value of each column (channel), the index of the sort function yields the order in which the channels became active. A selective feedback stimulus can be generated by comparing the order to a desired order and combining the burst detection signal after a time-delay with the order-match block using a logical AND-gate (not shown).

Figure 3 - Screenshot of the real-time feedback system during execution
The Mathworks xPC Target operating system (loader) provides basic information about the real-time model that is currently being executed using standard VGA graphical output. On the top panel, statistical information and an event log is shown, the area below can be used by the real-time Simulink model to display graphical or numerical target scope data. In this example application, we display the ongoing neural activity summed over a sliding 25 ms window (spikes per bin, cyan curve) as well as the stimulus amplitude and the times of feedback stimulation (yellow curve). The numerical scope at the bottom of the screenshot shows a count of serial

commands sent to the head-stage (switching between stimulation electrodes) and the stimulus generator (reprogramming the stimulus waveform) as well as a count of how many stimuli were triggered and how many action potentials and network spikes have been detected up to this point.

Figure 4 - Spontaneous network spikes

The signal at 60 substrate-integrated extracellular electrodes recording the neuronal activity from a dissociated nerve cell network after 3 weeks in culture is continuously monitored and analysed at a sample-rate of 16 kHz / 12 bit. This corresponds to a data processing rate of 11.5 MBit per second. Spikes are detected on-line by a threshold criterion. The multi-channel signal is processed in real-time by summing events across all channels inside a sliding window. When a predefined criterion is reached a trigger is generated and the preceding and immediately following complete raw data is stored for immediate or off-line analysis. The left panel shows the average of the activity summed across channels using sliding windows of 25 ms duration during a 45 minute recording. The variability is indicated by thin lines at ± one standard deviation. The threshold for burst detection (10 spikes in the past 25 ms) is indicated by the star. The inset shows the probability of a spike being recorded at a particular electrode during the time-course of the network spike using colour coding. The centre and right panel show the raw voltage traces recorded on 40 channels during two separate network spikes.

Figure 5 - Demonstration of real-time feedback stimulation contingent upon the spatio-temporal pattern of spikes

The left panel shows raw voltage traces recorded during a network spike. Action potentials are detected by a threshold criterion. When the activity summed across channels in a sliding window of 25ms reaches a predefined threshold (A), the multichannel spike-train is analysed between the time of -50ms to +100ms around the point of threshold crossing to determine whether a feedback stimulus should be applied or not. In this example, the feedback criterion is the order (time to first spike) in which channels become active during the burst of activity. A stimulus is provided through two electrodes (triangle) at +100ms (B) only if the channels become active in the predefined sequence shown (1-2-3). The centre panel shows the events recorded on 20 channels over 35 seconds. The second channel exhibits tonic activity while the other channels fire synchronously during five spontaneous bursts of activity as shown. The third and the fifth burst match the feedback criterion resulting in feedback stimulation as shown (triangle). The right panel shows the average of the activity summed across channels using time-bins of 5 ms during a 2 hour recording. Bursts that match the feedback criterion are averaged separately (dotted line, n=142) from those that did not (solid line, n=474). The variability is indicated by thin lines at ± one standard deviation. The immediate effect of stimulation (triangle) during the matching condition is shown in the inset (events recorded at one channel on 40 separate trials). The probability of a spike occurring is increased at around 12 ms post-stimulus (star).

Figure 6 - Continuous feedback stimulation in response to ongoing activity
The left panel shows the principle of a continuous closed loop set-up. The overall activity of a neuronal system is estimated by summing events across recording channels within a sliding window. A non-linear transformation of the input is used as a delayed feedback signal to the neural system. A stimulation of variable strength going around four predefined electrodes in a circular fashion at a frequency of 40 Hz.

The middle panel shows the raw traces recorded at a subset of channels and the stimulus artefacts at the electrodes. The amplitude of the next stimulus is determined from the activity during the current and previous 25 ms time window and the stimulus generator is then reprogrammed accordingly in real-time through a serial RS232 interface. The top section of right panel (A) shows events detected at various channels. The corresponding summed activity is shown in the lower section (B, blue line). For each point of time at which a feedback stimulus was delivered, the voltage amplitude is plotted (B, red triangle).

Tables

Description	Manufacturer	Estimated Cost
Matlab, Simulink, Real-Time Workshop, xPC Target	The Mathworks Inc.	2.000 € (or campus licence)
High-end PC (Windows)	(any)	1.000 €
Standard PC (Real-Time)	Asus/AMD 32 bit	500 €
Pro/100S Network Card	Intel	25 €
PD2/MF-64-3M	UEI (ueidaq.com)	2.000 €

Table 1 - Generic Parts for a bi-directional high-bandwidth neural interface with real-time feedback

With the parts listed in this table, a closed loop data processing environment with 11.5 Megabit per second data processing rate (across 60 channels) and 500µs feedback generation latency can be constructed.

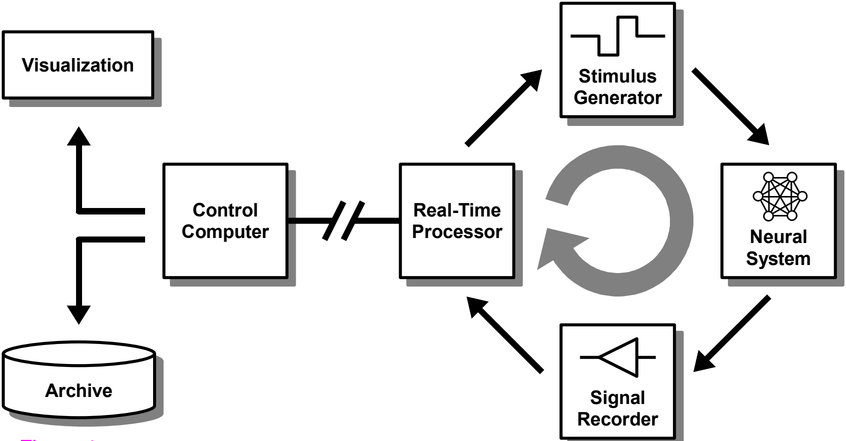

Visualization
Control
Computer
Real-Time
Processor
Stimulus
Generator
Neural
System
Signal
Recorder
Archive

Figure 1

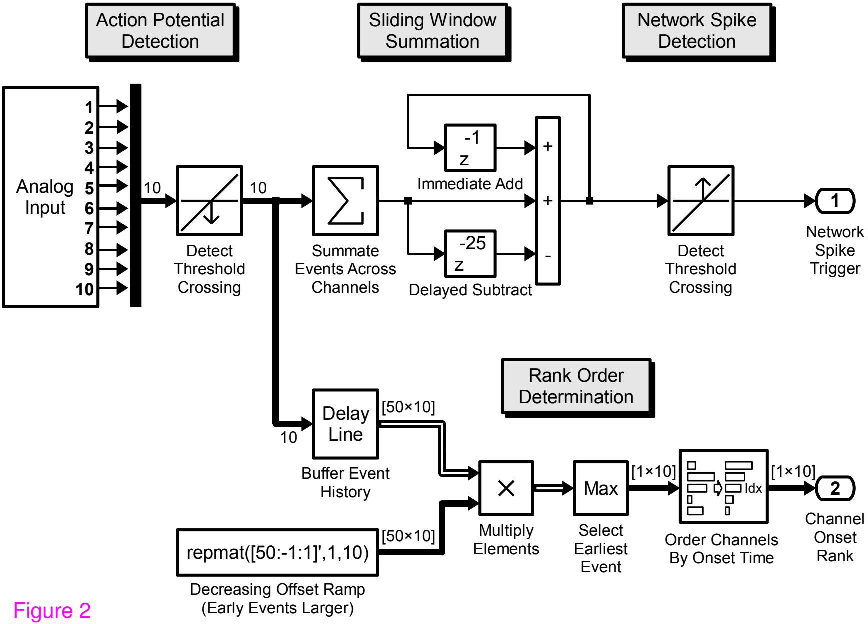

Action Potential Detection
Sliding Window Summation
Network Spike Detection
Analog Input
1
2
3
4
5
6
7
8
9
10
10
Detect Threshold Crossing
10
Summate Events Across Channels
-1
z
Immediate Add
+
+
-
-25
z
Delayed Subtract
Detect Threshold Crossing
1
Network Spike Trigger
Rank Order Determination
Delay Line
[50×10]
10
Buffer Event History
Multiply Elements
Max
Select Earliest Event
[1×10]
Idx
Order Channels By Onset Time
[1×10]
2
Channel Onset Rank
repmat([50:-1:1]',1,10)
[50×10]
Decreasing Offset Ramp (Early Events Larger)

Figure 2

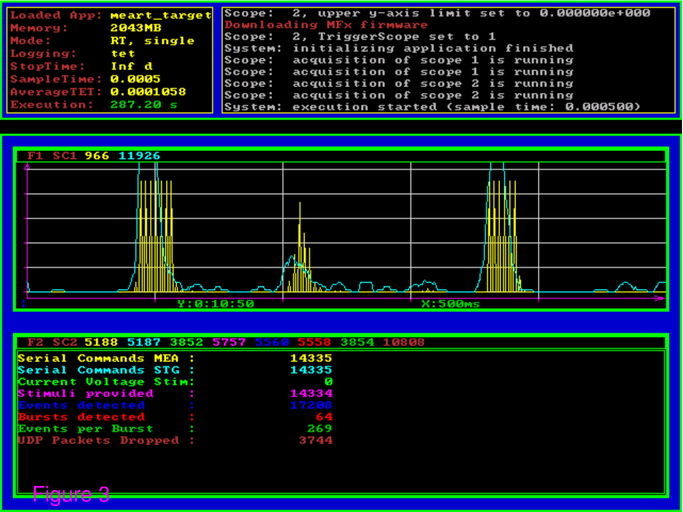
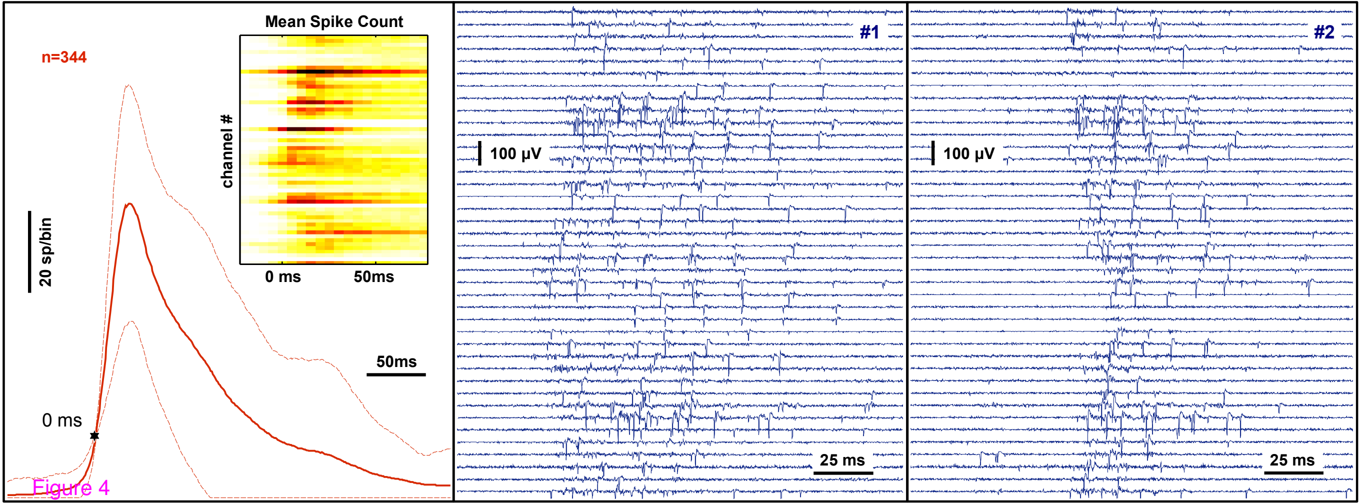

Loaded App: meart_target
Memory: 2043MB
Mode: RT, single
Logging: tet
StopTime: Inf d
SampleTime: 0.0005
AverageTET: 0.0001058
Execution: 287.20 s
Scope: 2, upper y-axis limit set to 0.000000e+000
Downloading MFx firmware
Scope: 2, TriggerScope set to 1
System: initializing application finished
Scope: acquisition of scope 1 is running
Scope: acquisition of scope 1 is running
Scope: acquisition of scope 2 is running
Scope: acquisition of scope 2 is running
System: execution started (sample time: 0.000500)
F1 SC1 966 11926
Y:0:10:50
X:500ms
F2 SC2 5188 5187 3852 5757 5560 5558 3854 10808
Serial Commands MEA : 14335
Serial Commands STG : 14335
Current Voltage Stim: 0
Stimuli provided : 14334
Events detected : 17208
Bursts detected : 64
Events per Burst : 269
UDP Packets Dropped : 3744

Figure 3



Mean Spike Count
n=344
channel #
0 ms
50ms
20 sp/bin
50ms
0 ms
Figure 4
#1
100 μV
25 ms
#2
100 μV
25 ms

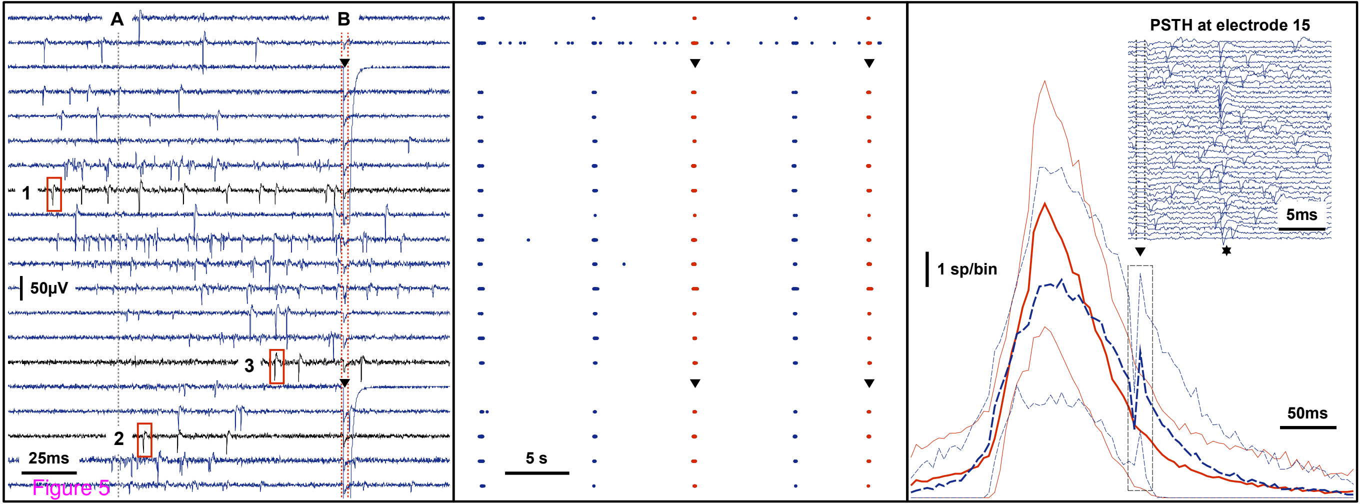
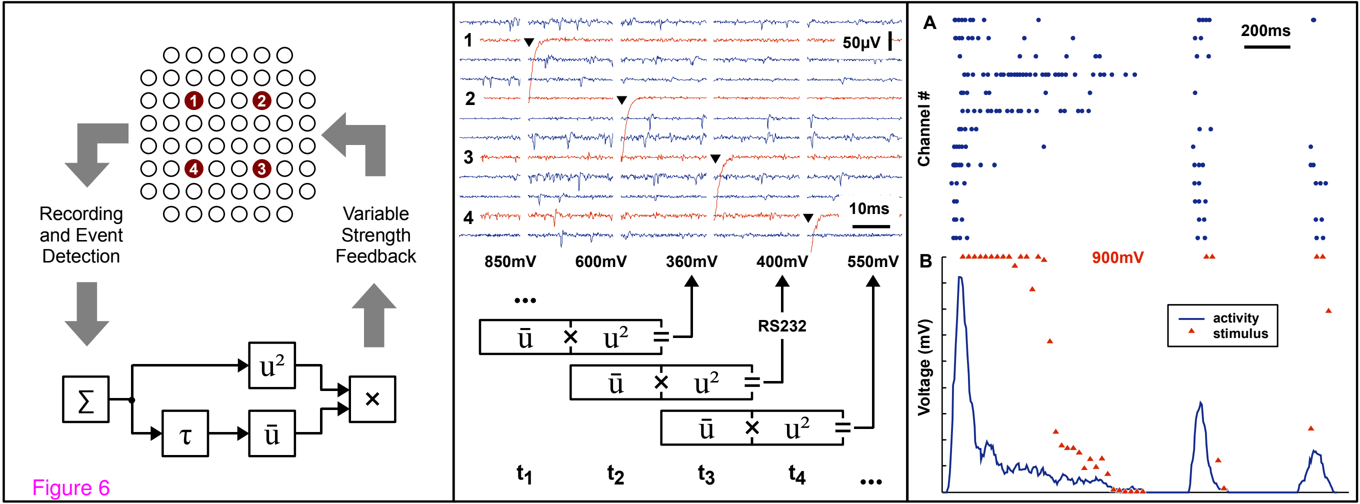


Figure 5


1
2
3
4
Recording and Event Detection
Variable Strength Feedback
Σ
τ
u^2
$\bar{u}$
×
50µV
10ms
850mV
600mV
360mV
400mV
550mV
RS232
$\bar{u}$ × u^2 =
t_1
t_2
t_3
t_4
A
Channel #
200ms
B
900mV
Voltage (mV)
activity
stimulus
Figure 6