A Reconfigurable Architecture for Real-Time Prediction of Neural Activity

Will X.Y. Li, Ray C.C. Cheung
and Rosa H.M. Chan
Department of Electrical Engineering
City University of Hong Kong, Hong Kong
xyli@ee.cityu.edu.hk
{rosachan,rcheung}@cityu.edu.hk

Dong Song and Theodore W. Berger
Department of Biomedical Engineering
University of Southern California
Los Angeles, CA 90089 USA
dsong@usc.edu
berger@bmsr.usc.edu

***Abstract*— In this paper, we propose an FPGA-based hardware architecture for conducting real-time prediction of neural activity using a second-order generalized Laguerre-Volterra model (GLVM). This architecture serves as a rapid prototype of the prediction module of the future cognitive neural prosthetic device. We validate the functionality of the hardware model by utilizing the neuronal firing data of behaving rats trained to perform the delayed nonmatch-to-sample (DNMS) memory task.**

I. Introduction

Cognitive neural prosthesis design has been a field of interest in recent years. This type of prosthesis, if successfully developed, would provide more fundamental treatment to diseases related to cognitive impairment, such as the Alzheimer's disease (AD) [1].

A mathematical model describing how information carried by the biosignals flows through the brain regions is important to the development of the neural prosthetic devices. One approach is parametric modeling [2], which has been intensively implemented for simulating the detailed biological mechanisms/processes underlying the information processing. However, this approach typically requires a large number parameters, which are difficult to estimate, and intensive computation, which are not feasible in real time applications.

In view of the above, we refer to the non-parametric models (data-driven models), which use engineering modeling techniques such as network analyses, information theory and statistical methods to investigate the behavior of biological neurons or neural networks. The generalized Laguerre-Volterra model (GLVM), which is proposed by Song et al. [3], is one of these well-functioning non-parametric models.

Previous studies carried by Volterra [4], Wiener [5] and Marmarelis [6] have demonstrated that for any nonlinear and time-invariant system with finite memory length, the system output can be represented as a functional power series of the system input, as described by eq. 1.

In (1), the system dynamics is revealed through the temporal convolution between system input and the kernel functions *k*; while the system nonlinearity is suggested by multiple convolutions between the input and the higher order kernel functions. In the GLVM, we use the real-time Laguerre expansion of Volterra kernels and the point process filters to track the nonlinear time-variant neural system. The detailed description of the generalized Laguerre-Volterra (GLV) algorithm can be found in [7].

$$y(t) = k_0 + \sum_{\tau=0}^{M} k_1(\tau)x(t-\tau) + \sum_{\tau_1=0}^{M}\sum_{\tau_2=0}^{M} k_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2) + ... \quad (1)$$

Real-time prediction of neural activity is critical to the neural prosthetic applications because the spiking activity of the neural ensemble is very time-sensitive to the stimuli imposed. Comparing to the software-based prototyping of the GLVM (which is unable to guarantee real time), cycle-accurate hardware can well achieve this purpose. Among various hardware platforms, the Field-Programmable Gate Array (FPGA), due to its reconfigurability and excellent parallel processing capability, becomes an ideal choice of implementation and early-stage prototyping tool for the prosthetic device.

The complete flow of the GLV algorithm consists of two stages. The first is parameter estimation, which is to use the recorded neuronal firing input/output data to estimate the model coefficients. The second is output prediction, which uses the estimated coefficients and novel input to predict the model output. Accordingly, for a working prosthetic device, there are two main modules, i.e. estimation module and prediction module. For power and area considerations, it is desirable that the estimation module be designed as the *extracorporeal system* and the prediction module the *brain implant*.

In [8], we have demonstrated an efficient hardware architecture which adopts the GLV algorithm to estimate the model coefficients. That serves as an early prototyping of the estimation module for the prosthesis. In this paper, we describe the FPGA-based hardware framework of the prediction module, utilizing second-order Volterra kernels.

The main contribution of our work consists of three parts: 1) we propose the first FPGA-based hardware architecture optimized for predicting neural activity using high-order Volterra kernels; 2) we demonstrate that this architecture is easily scalable and is area-efficient; 3) we validate the functionality

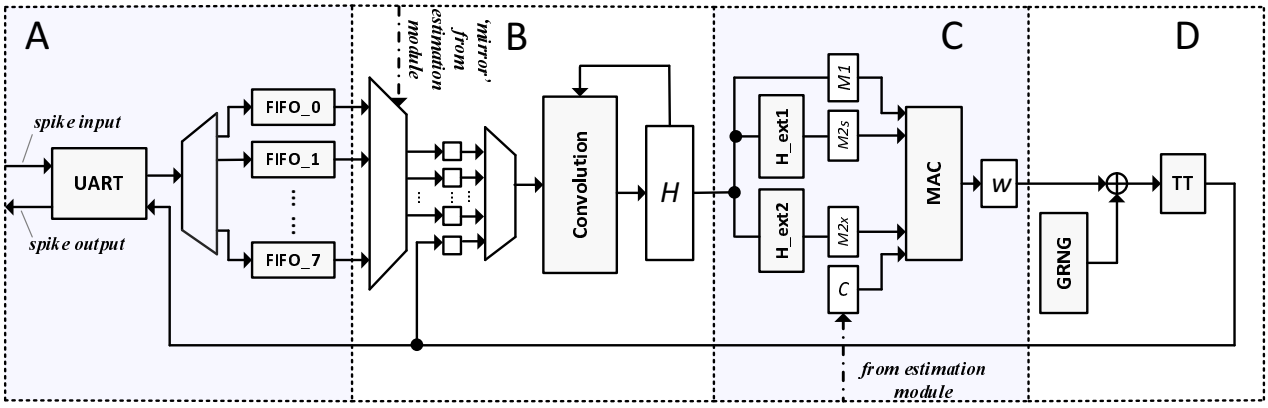

Fig. 1. Overview of the hardware framework. The light-blue boxes indicate functional units. The white rectangular boxes indicate key registers or register arrays. The prediction module can be divided into four major components (A to D) according to their functionalities. H_ext1 and H_ext2 are augmented horizontal vector (*H*) extension units. The three MUXes (from left to right) are FIFO writing selection MUX, channel selection (64 to *N*) MUX and input selection MUX for the convolution operation.

of this architecture using neuronal firing data from behaving rats performing the DNMS task.

II. Hardware Framework

The hardware framework of the prediction module is shown in Fig. 1. This architecture consists of four parts (as indicated by the four regions in the figure): 1) a Universal Asynchronous Receiver/Transmitter (UART) component which serves as the data interface between the FPGA and the host device; 2) a convolution component which conducts the temporal convolution between system input and the kernel functions; 3) an augmented horizontal vector *H* (convolution product) expansion component which operates among the elements of *H*, expanding the vector space, and a multiplication-and-accumulation (MAC) component which performs the MAC operation between *H* and the Laguerre coefficients *C*; 4) an threshold trigger (TT) which generates the neural spikes using a threshold function.

A. Data Interface (Region A)

The RS-232 serial interface is adopted in this prediction module. In current experimental settings, the sampling rate by the multi-electrode array is 2kHz; the 115,200 baud rate is fast enough to meet this sampling frequency. The serial interface employing the UART protocol has a lower power-area product than its counterpart in the estimation module, which is suitable for implantable applications. The prototyping board comprises an embedded component which transforms the RS-232 voltage to the FPGA operation voltage. The 16x oversampling technique is adopted to reduce data transmission error. A customizable control unit is designed to transform serial data into 8-bit data frames and store the data frames into input FIFOs. The number of FIFOs to be used depends on the number of inputs. In current experimental settings, 64 electrodes are used and there are 8 input FIFOs in the interface. The input FIFO caches data between different clock domains (UART and processing core). The output data are sent back to be host device via identical datapath.

B. The Vector Convolution Component (Region B)

The vector convolution algorithm can be found in Sec. III-D of [8]. This component is similar to the one shown in [8], but with two distinct features. First, the datapath here can be fully pipelined due to the elimination of the feedback path of the Laguerre coefficients in the estimation module. In the prediction module, the coefficient appears a constant vector. Second, before conducting vector convolution, the component allows to choose *N* most significant inputs from the 64 sampling channels. Previous experiments suggest that not all inputs to the system contribute to the output spikes. Introducing too many inputs would, on the contrary, deteriorates the quality of prediction. The effective inputs can be identified by conducting GLV model selection [9]. And the selection result can be sent to the prediction module via the data interface proposed in Sec. IV-A.

C. Processing of Convolution Products (Region C)

The number of variables *M* of a second order model after the convolution stage can be counted as $M = M_1 + M_{2s} + M_{2x}$. M_1 is the number of variables in the augmented horizontal vector *H* produced by the vector convolution unit. M_{2s} accounts for the interactions among different basis functions of each individual input by multiplication between the element pairs (in all permutation). Given *L* the number of basis functions, $L(L+1)/2$ multiplication operations are needed. M_{2x} accounts for the interactions among different basis functions from different inputs. $C_N^2 * L^2$ pair-wise multiplications are needed in this case. The extended horizontal vector and the Laguerre coefficients are then sent to the MAC component. The value of the membrane potential *w* can be acquired at the root stage of the adder array in the MAC component.

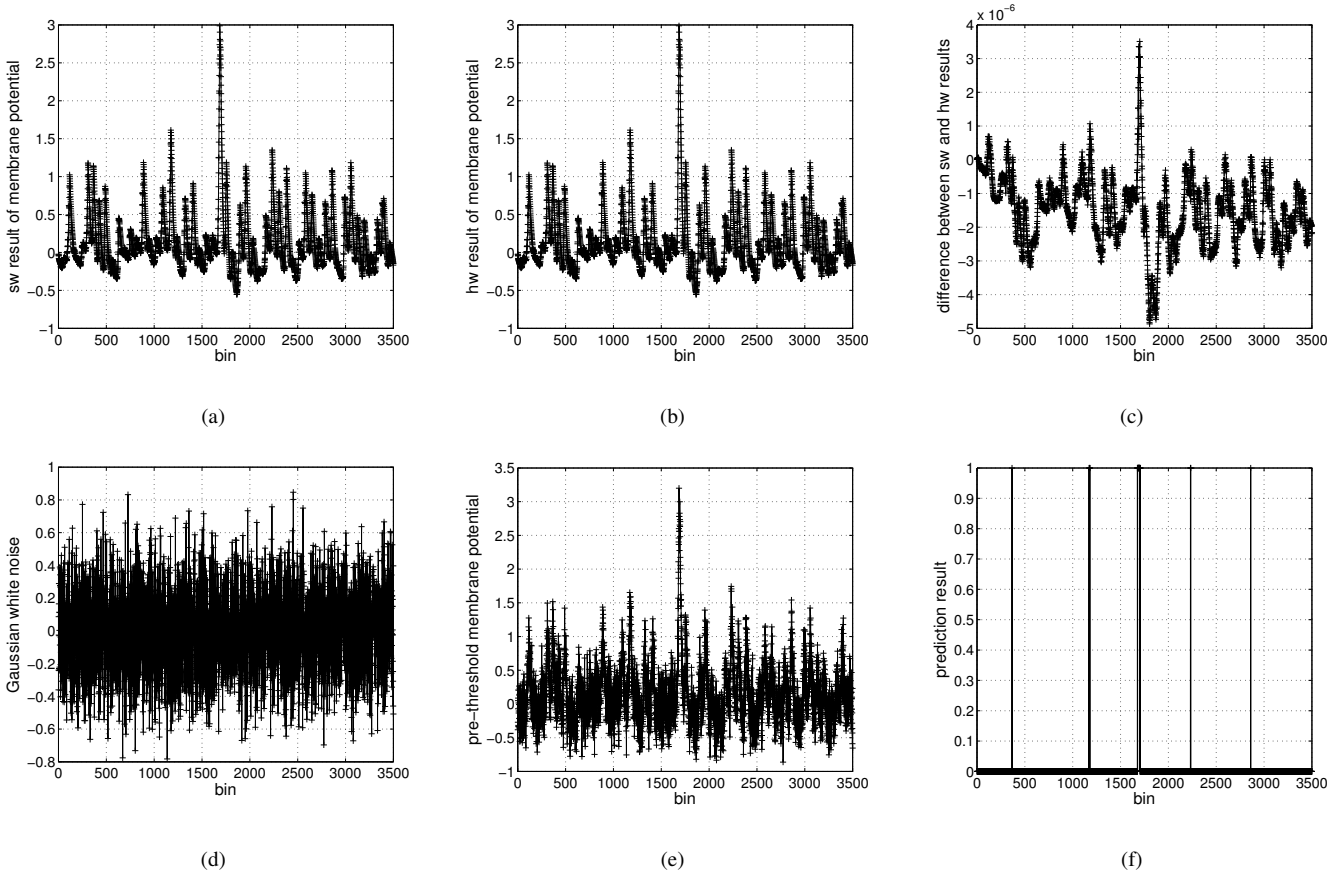

Fig. 2. The calculation results by the FPGA-based hardware platform and the software platform (bin size: 3,500). (a)-(c): calculation results of the membrane potentials by software (sw), hardware (hw) and difference of the two data sets; (d): Gaussian white noise (μ=0, σ^2=0.25); (e): calculation result of pre-threshold membrane potential by hardware; (f): predicted model output (θ=1.50).

D. The Threshold Trigger (Region D)

Upon being calculated, w is first added to a Gaussian noise quantity which is invented to simulate the intrinsic neuronal noise and the noise contributed by unobserved model inputs. Therefore, a Gaussian random number generator (GRNG) is constructed to simulate this noise term. Our GRNG is designed based on a uniform random number generator (URNG) whose structure is first proposed by Tkacik et al. [10]. It is implemented by the bitwise XOR operations between the lower 32 bits of a 43-bit Linear Feedback Shift Register (LFSR) and the lower 32 bits of a 37-bit Cellular Automata Shift Register (CASR). Each 100 numbers generated by the URNG are added to form the Gaussian distribution. The summation of the membrane potential and the noise is compared to a threshold value θ which can be defined by the user. If the threshold is crossed, an output spike will be generated. The prediction result is fed back to the vector convolution unit as the 'after potential' for iterative calculation.

III. Results

Our test data are acquired from the male Long-Evans rats aged from four to six months which are trained to perform the delayed nonmatch-to-sample (DNMS) task [11]. The hippocampal neuronal firing activities are recorded by the multi-electrode array when the rats are performing the task. The recorded signals are processed by the spike sorting algorithm.

We use the FPGA-based hardware module and software (whose precision is already validated by experiments [3]) to process one session of the neuronal firing data (Animal # 1150). The calculation results of the membrane potential are shown in Fig. 2(a) (software) and Fig. 2(b) (hardware) respectively. The difference between the two data sets are shown in Fig. 2(c). We calculate the normalized mean square error between the two data sets to be at 10^{-11} scale. The functionality of the hardware platform can thereby be validated.

Fig. 2(d) plots the noise wave. Fig. 2(e) plots the summation product of the membrane potential and the Gaussian noise. The predicted neuronal spikes by the TT component are shown in Fig. 2(f). The spiking threshold θ can be defined by the user. The higher θ is set, the higher the false negative rate (FNR) will be. On the other hand, lower θ will incur higher false positive rate (FPR).

The prediction module runs on a Xilinx Virtex-5 XC5VLX110T FPGA which is part of the XUPV5-LX110T board. This prediction core consumes 35,045 FPGA Slice LUTs, and 4,542 Slice Registers while operating at a frequency of 16MHz (optimal speed to coordinate with the Baud). There is a remarkable reduction in the area compared with our

previous estimation module architecture. This is mainly due to the elimination of the Laguerre coefficient estimation circuitry in the prediction module and also reduced model inputs by the introduction of the '64 to *N*' input selection component.

Another feature of our design is its multi-fold scalability, which can be derived by module reuse and MISO model extension. In some deign components, such as vector convolution and MAC, different number of processing elements can be implemented, according to the number of effective model inputs that would affect the system data throughput. In case that the device to be used is resource-limited, the design can be implemented with multiple FPGAs, each representing a MISO model. This is due to the data-irrelevance of MIMO model outputs in the proposed GLV algorithm [7].

IV. Discussion

A. Filtering Techniques for Coefficients Calculation

The coefficients used in this work are estimated using the Steepest Descent Point Process Filter (SDPPF). While it works effectively for estimation of the Laguerre coefficients, other filtering techniques such as the Stochastic State Point Process Filter (SSPPF) may perform better in terms of accuracy. With the SSPPF, we also estimate the variance of the Laguerre coefficients as shown in eq. 10 of [12].

We have implemented the SSPPF in our software model. For the FPGA implementation of the SSPPF, there are particular challenges, mainly lying in the inversion operation of the large-size covariance matrices. The size of *W* in eq. 10 of [12] can be identified when the number of effective inputs *N* and the number of basis functions *L* are determined, accordingly to the GLV algorithm, as shown in Fig. 3. Efficient hardware architecture has yet to be invented to accommodate this intensive computational requirement ($N \geq 3$), taking into consideration of the datapath and memory access pattern optimization for efficient chip resource allocation and power reduction.

B. Adoption of Advanced Design Paradigms

As an early-stage prosthetic prototype, the current hardware system has its limitations and is subject to further optimizations. For the future neuroprosthetic device (an ASIC chip), power and area are two important concerns. In order to reduce power consumption and extend battery life, both circuit-level design optimization and process technology improvement are needed. In the fabrication stage, transistors with reduced characteristics length and the high-K metal gate process optimized for low power applications can be well utilized. Advanced design paradigms, such as fault-tolerance can also be introduced to enhance the flexibility and robustness. Bio-compatibility should to be considered in future design effort.

V. Conclusions

We propose the first FPGA-based circuit architecture for research of neural activity utilizing the second order Volterra kernels. This architecture is very efficient for online prediction of neuronal firing signals in experimental settings. Potentially, it has a wide usage for the development of future neuroprosthetic devices. The proposed architecture can be integrated with our previous architecture [8] in order to form a complete full-scale neural signal processing system.

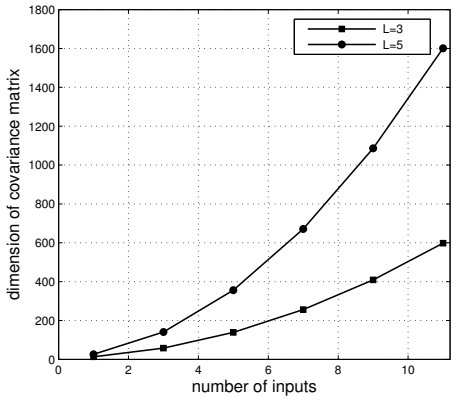

Fig. 3. Dimension of the covariance matrix with different *N* and *L*.

References

[1] T. W. Berger, D. Song, R. H. M. Chan, and V. Z. Marmarelis, "The neurobiological basis of cognition: identification by multi-input, multi-output nonlinear dynamic modeling," *Proceedings of the IEEE*, vol. 98, pp. 356–374, 2010.

[2] D. Song, V. Z. Marmarelis, and T. W. Berger, "Parametric and non-parametric modeling of short-term synaptic plasticity. Part I: computational study," *Journal of Computational Neuroscience*, vol. 26, pp. 1–19, 2009.

[3] D. Song, R. H. M. Chan, V. Z. Marmarelis, R. E. Hampson, S. A. Deadwyler, and T. W. Berger, "Nonlinear dynamic modeling of spike train transformations for hippocampal-cortical prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 54, pp. 1053–1066, Jun 2007.

[4] V. Volterra, "Theory of functionals and of integral and integro-differential equations," *New York: Dover*, 1959.

[5] N. Wiener, "Nonlinear problems in random theory," *New York: Technology Press MIT/Wiley*, 1958.

[6] V. Z. Marmarelis, "Nonlinear dynamic modeling of physiological systems," *Hoboken: Wiley-IEEE Press*, 2004.

[7] D. Song, R. H. M. Chan, V. Z. Marmarelis, R. E. Hampson, S. A. Deadwyler, and T. W. Berger, "Nonlinear modeling of neural population dynamics for hippocampal prostheses," *Neural Networks*, vol. 22, pp. 1340–1351, 2009.

[8] W. X. Y. Li, R. H. M. Chan, W. Zhang, R. C. C. Cheung, D. Song, and T. W. Berger, "High-performance and scalable system architecture for the real-time estimation of generalized Laguerre-Volterra MIMO model from neural population spiking activity," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, pp. 489–501, 2011.

[9] D. Song, R. H. M. Chan, V. Z. Marmarelis, R. E. Hampson, S. A. Deadwyler, and T. W. Berger, "Sparse generalized Laguerre-Volterra model of neural population dynamics," *Proceedings of the 31st Annual International Conference of the IEEE EMBS*, pp. 4555–4558, 2009.

[10] T. E. Tkacik, "A hardware random number generator," *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 450–453, 2002.

[11] R. E. Hampson, D. Song, R. H. M. Chan, A. J. Sweatt, M. R. Riley, G. A. Gerhardt, D. C. Shin, V. Z. Marmarelis, T. W. Berger, and S. A. Deadwyler, "A nonlinear model for hippocampal cognitive prosthesis: Memory facilitation by hippocampal ensemble stimulation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 2, pp. 184–197, 2012.

[12] R. H. M. Chan, D. Song, and T. W. Berger, "Tracking temporal evolution of nonlinear dynamics in hippocampus using time-varying Volterra kernels," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 54, pp. 4996–4999, 2008.