Fourth Quarterly Progress Report
N01-DC-9-2106

Effects of Remaining Hair Cells on Cochlear Implant Function

H. Mino, J.T. Rubinstein, C.A. Miller, P.J. Abbas

Department of Otolaryngology - Head and Neck Surgery
Department of Speech Pathology and Audiology
Department of Biomedical Engineering
University of Iowa
Iowa City, IA 52242

July 31, 2000

Contents

1 Introduction **2**

2 Summary of activities in this quarter **3**

3 Focus Topic: Comparison of algorithms in generation of action potentials using stochastic models **4**
3.1 Introduction 4
3.2 Preliminaries 5
3.3 CNT algorithm 6
3.4 Numerical examples 9
3.4.1 Single node model 9
3.4.2 Monophasic stimulus 11
3.4.3 Pre-conditioned stimulus 13
3.4.4 Computation time 16

4 Discussion **16**

5 Plans for the next quarter **17**

6 Appendix: Presentations and publications **17**

1 Introduction

Patients who have significant residual hearing are presently receiving cochlear implants and that it is likely that patients with more hearing will be become implant candidates in the future. Electrical stimulation in ears with functional hair cells brings up the possibility of interactions between hair-cell mediated and direct stimulation of auditory nerve fibers as well as the possibility of acoustic and electrical interactions. To address these issues, we have proposed a sequence of experiments designed to first characterize the response patterns and interactions and then to explore techniques to best exploit them. These experiments employ both physiological measures and computational modeling of the effects of hair cells on the response to electrical stimulation. The initial experiments use measures of the electrically evoked compound action potential (EAP) as well as measures of single fiber activity. In the course of this contract, a direct comparison will be made of the response properties with and without functioning hair cells as well as comparisons of the responses to electrical stimulation with and without acoustic stimulation.

In this fourth quarterly progress report (QPR), we focus on a comparison of algorithms to generate action potentials from the standpoint of computational modeling. We do this because simulation of interactions between synaptic activity and direct electrical stimulation of auditory neurons requires large numbers of Monte Carlo runs. These runs are necessary to explore the relationship between stochastic synaptic currents and stochastic neuronal events due to sodium channel fluctuations. These relationships have never been simulated before due to the computational complexity involved. Several algorithms have been proposed for generating current fluctuations in sodium channels on the basis of stochastic models and eventually for calculating the transmembrane potentials combined with sodium current fluctuations. However, these algorithms have not been explicitly compared with regard to their performance, e.g., response properties, computation time, etc. Since generating action potentials based on stochastic modeling of sodium channels is computationally intensive, the most computationally efficient algorithm is desired. The objective of the work described in this QPR was to compare algorithms so as to identify those that are both computational efficient and physiologically appropriate. We did so by examining single-node models, which form the basis for distributed axon models. These computer simulations can be demonstrated in part using JAVA applets on the Internet at:

http://cessna.oto.uiowa.edu/qpr

Note that the slow execution of JAVA applets currently limits such a web site to simple demonstrations for educational purposes.

2 Summary of activities in this quarter

In our fourth quarter (1 April - 30 June, 2000), the following activities related to this contract were completed:

1. We have conducted preliminary experiments in two guinea pigs exploring interactions of acoustic and electric stimuli in preparations with intact hearing. The data collected to-date indicates that the two modes of stimulation interact, at least at the level of the electrically evoked auditory brainstem response (EABR). Two sets of data support this. First, the presentation of a wideband acoustic noise stimulus reduced the EABR evoked by a pulsatile electric stimulus in a systematic fashion. Second, by using 100 Hz sinusoidal acoustic stimuli, we have noted that the degree of this interaction is dependent upon the phase of the acoustic stimulus at which the electric stimulus is presented. This work is ongoing and will be the subject of a later QPR.

2. We have purchased and tested a new acoustic system for sound delivery.

3. We have completed initial testing of new laboratory software package being developed for more efficient data collection and analysis.

4. We attended the 5th European Symposium on Paediatric Cochlear Implantation in Antwerp.

3 Focus Topic: Comparison of algorithms in generation of action potentials using stochastic models

3.1 Introduction

Several simulation algorithms have been proposed for generating current fluctuations in sodium channels and eventually for calculating the transmembrane potentials (Skaugen and Walloe, 1979; Clay and DeFelice, 1983; Strassberg and DeFelice, 1993; Rubinstein, 1995; Chow and White, 1996; Fox, 1997). The stochasticity of sodium channels plays a key role in accounting for fluctuations, neural excitabilities, and measurable effects such as jitter and RS (the relative spread). However, in practical situations, a serious problem has arisen in computation time in which many sodium channels are incorporated into nodes of Ranvier in a distributed nerve fiber cable model and likewise in a population of such nerve fibers, e.g., the VIII-th cranial nerve has approximately 30,000 fibers. Although recent computer technologies could solve the problem under some restricted conditions (i.e., reducing the number of nodes and/or the number of fibers), it is worthwhile to simulate neuronal responses in a realistic distributed cable model using the most computationally efficient algorithm.

The algorithms for computer simulation can be categorized into two groups: approximation algorithms of the differential equation describing dynamics by Langevin's equation (Fox, 1997), and exact algorithms of sodium current fluctuations by Markov jumping process (Skaugen and Walloe, 1979; Clay and DeFelice, 1983; Strassberg and DeFelice, 1993; Rubinstein, 1995; Chow and White, 1996). The exact algorithms can be subdivided into: Channel-State-Tracking (CST) algorithm, and Channel-Number-Tracking (CNT) algorithm. The CST algorithm tracks the states of each channel and superimposes individual channel currents corresponding to the states in order to generate sodium channel current fluctuations. This algorithm is simple in principle, however it is computationally intensive. This algorithm has been utilized in Clay and DeFelice in 1983, Strassberg and DeFelice in 1993, and Rubinstein in 1995. An alternative algorithm proposed by Gillespie and later implemented by Chow and White in 1996, tracks the number of channels in each state, assuming that multi-channel systems are independent and memoryless. The CNT algorithm promises much greater efficiency in computing sodium current fluctuations in cases where many

channels are considered. This is because the algorithm implements only the single-channel system possessing the "effective transition rate" associated with the transition rates in the multi-channel system and the number of channels in each state.

This fourth QPR deals with evaluating performance of the CNT algorithm in terms of neural excitability and computation time with a comparison to those of the CST algorithm as well as the approximation algorithm. In particular, the CNT algorithm is described in depth in the hope that the algorithm helps efficiently simulate action potentials for investigating not only cochlear prosthesis, but all stochastic ionic channel processes.

3.2 Preliminaries

The CNT algorithm was originally developed by Gillespie, and later implemented by Chow and White. In this subsection, we summarize the background, i.e., eight-state Markov processes, the algorithm, and the method of generating data in practical situations. A review of the CNT algorithm with current fluctuations is found in White, Rubinstein, and Kay (2000).

Stochastic ion channels can be modeled well (Colquhoun and Hawkes, 1977; Neher and Stevens, 1977; Colquhoun and Hawkes, 1981) by continuous-time discrete-state Markov process (Markov Jumping Processes; MJPs) (Papoulis, 1985) . Since the sodium channel has been assumed to possess three activating m gates with four distinct states and one inactivating h gate with two distinct states according to Hodgkin-Huxley model, the kinetic scheme based on Markov process theory has eight states, m_0h_1, m_1h_1, m_2h_1, m_3h_1, m_0h_0, m_1h_0, m_2h_0, and m_3h_0 with twenty transition rates designated by $\alpha_{h,m}$'s and $\beta_{h,m}$'s. The kinetic scheme is expressed as follows:

$$\begin{array}{ccccccc} & \overset{3\alpha_m}{\underset{\beta_m}{\rightleftharpoons}} & & \overset{2\alpha_m}{\underset{2\beta_m}{\rightleftharpoons}} & & \overset{\alpha_m}{\underset{3\beta_m}{\rightleftharpoons}} & \\ m_0h_0 & & m_1h_0 & & m_2h_0 & & m_3h_0 \\ \alpha_h \downarrow\uparrow \beta_h & & \alpha_h \downarrow\uparrow \beta_h & & \alpha_h \downarrow\uparrow \beta_h & & \alpha_h \downarrow\uparrow \beta_h \\ & \overset{3\alpha_m}{\underset{\beta_m}{\rightleftharpoons}} & & \overset{2\alpha_m}{\underset{2\beta_m}{\rightleftharpoons}} & & \overset{\alpha_m}{\underset{3\beta_m}{\rightleftharpoons}} & \\ m_0h_1 & & m_1h_1 & & m_2h_1 & & m_3h_1 \end{array} \tag{1}$$

where the state m_3h_1 denotes all four gates opening and thereby the channel is open. The transition rates in mammalian neuron at $37°C$ are expressed

as:

$$\alpha_m = \frac{1.872(V_m - 25.41)}{1 - e^{(25.41 - V_m)/6.06}} \tag{2}$$

$$\beta_m = \frac{3.973(21.001 - V_m)}{1 - e^{(V_m - 21.001)/9.41}} \tag{3}$$

$$\alpha_h = \frac{-0.549(27.74 + V_m)}{1 - e^{(V_m + 27.74)/9.06}} \tag{4}$$

$$\beta_h = \frac{22.57}{1 + e^{(56.0 - V_m)/12.5}} \tag{5}$$

in which the transition rates have units of ms^{-1} and the transmembrane potential V_m has units of mV. Note that these transition rates are dependent on the transmembrane potential. The transmembrane potential $V_m(t)$ is described as a function of time by:

$$C_m \frac{dV_m(t)}{dt} + \frac{V_m(t)}{R_m} + \gamma_{Na} N_{Na}(t)(V_m(t) - E_{Na}) = I_{app}(t) \tag{6}$$

where $I_{app}(t)$ denotes the stimulus current, C_m and R_m respectively denote capacitance and resistance of membrane, and γ_{Na}, $N_{Na}(t)$ stand for conductance of single sodium ion channels and the number of sodium channels activated at time t in the single node under consideration. In exact algorithms, the number of sodium channels activated $N_{Na}(t)$ plays a key role in introducing its stochasticity into the transmembrane potentials $V_m(t)$.

3.3 CNT algorithm

The CNT algorithm for numerical simulations initializes the number of channels in each state at time t_0, and repeats (i)generating lifetime (length of the channel's open time), (ii)updating the number of channels in the current and new states by determining which one of the state transitions occurs.

The probability density function of the lifetime in a specific state at $t = [t_0, t_{max}]$ in multi-channel systems is represented by

$$p(\tau; t) = \lambda(t) e^{-\lambda(t)\tau} \qquad (\tau \geq 0) \tag{7}$$

where the effective transition rate $\lambda(t)$ as a function of t is given by:

$$\lambda(t) = \sum_{i=0}^{3} \sum_{j=0}^{1} N_{m_i h_j}(t) \gamma_{ij}(t) \tag{8}$$

In (8), $N_{m_i h_j}(t)$ stands for the number of channels in state $m_i h_j (i = 0, 1, 2, 3; j = 0, 1)$, and $\gamma_{ij}(t)$ denotes the sum of transition rates associated with escapes from state $m_i h_j (i = 0, 1, 2, 3; j = 0, 1)$, where

$$\begin{cases} \gamma_{00}(t) = 3\alpha_m + \alpha_h \\ \gamma_{10}(t) = \beta_m + \alpha_h + 2\alpha_m \\ \gamma_{20}(t) = 2\beta_m + \alpha_h + \alpha_m \\ \gamma_{30}(t) = 3\beta_m + \alpha_h \\ \gamma_{01}(t) = 3\alpha_m + \beta_h \\ \gamma_{11}(t) = \beta_m + \beta_h + 2\alpha_m \\ \gamma_{21}(t) = 2\beta_m + \beta_h + \alpha_m \\ \gamma_{31}(t) = 3\beta_m + \beta_h \end{cases} \tag{9}$$

The lifetime t_l from the current state to the next state is a random variable sampled from a sample space specified by the exponential distribution of (7) and is generated by a random number:

$$t_l = -\frac{ln(u_1)}{\lambda(t)} \tag{10}$$

where u_1 denotes a random number uniformly distributed within [0, 1].

The number of channels in each state is tracked on the basis of the probability of state transition from the current state at $t = t_0$ to the next state at $t = t_0 + t_l$. The state transition will occur after the lifetime t_l passes. The specific transition of the twenty possible states that occurs at $t = t_0 + t_l$ is determined by generating another random number uniformly distributed within [0,1] and calculating the cumulative state transition probability at $t = t_0$:

$$P_i(t) = \sum_{j=0}^{i} \eta_j / \lambda(t) \qquad (i = 0, 1, ..., 20) \tag{11}$$

where

$$\begin{cases} \eta_0 = 0 & \\ \eta_1 = 3\alpha_m N_{m_0h_0} & (m_0h_0 \to m_1h_0) \\ \eta_2 = \beta_m N_{m_1h_0} & (m_1h_0 \to m_0h_0) \\ \eta_3 = 2\alpha_m N_{m_1h_0} & (m_1h_0 \to m_2h_0) \\ \eta_4 = 2\beta_m N_{m_2h_0} & (m_2h_0 \to m_1h_0) \\ \eta_5 = \alpha_m N_{m_2h_0} & (m_2h_0 \to m_3h_0) \\ \eta_6 = 3\beta_m N_{m_3h_0} & (m_3h_0 \to m_2h_0) \\ \eta_7 = \alpha_h N_{m_0h_0} & (m_0h_0 \to m_0h_1) \\ \eta_8 = \beta_h N_{m_0h_1} & (m_0h_1 \to m_0h_0) \\ \eta_9 = \alpha_h N_{m_1h_0} & (m_1h_0 \to m_1h_1) \\ \eta_{10} = \beta_h N_{m_1h_1} & (m_1h_1 \to m_1h_0) \\ \eta_{11} = \alpha_h N_{m_2h_0} & (m_2h_0 \to m_2h_1) \\ \eta_{12} = \beta_h N_{m_2h_1} & (m_2h_1 \to m_2h_0) \\ \eta_{13} = \alpha_h N_{m_3h_0} & (m_3h_0 \to m_3h_1) \\ \eta_{14} = \beta_h N_{m_3h_1} & (m_3h_1 \to m_3h_0) \\ \eta_{15} = 3\alpha_m N_{m_0h_1} & (m_0h_1 \to m_1h_1) \\ \eta_{16} = \beta_m N_{m_1h_1} & (m_1h_1 \to m_0h_1) \\ \eta_{17} = 2\alpha_m N_{m_1h_1} & (m_1h_1 \to m_2h_1) \\ \eta_{18} = 2\beta_m N_{m_2h_1} & (m_2h_1 \to m_1h_1) \\ \eta_{19} = \alpha_m N_{m_2h_1} & (m_2h_1 \to m_3h_1) \\ \eta_{20} = 3\beta_m N_{m_3h_1} & (m_3h_1 \to m_2h_1) \end{cases} \tag{12}$$

If the random number generated is within $[P_{i-1}(t_0), P_i(t_0)), (i = 1, 2, ..., 20)$, then i-th state transition is regarded to occur at $t = t_0 + t_l$ in which i-th state transition corresponds to that designated in parentheses in (12). Once the state transition is determined, the number of channels at $t = t_0 + t_l$ is updated such that one is subtracted from the number of channels in the current state, and meantime one is added to that in the next state. For instance, if the random number is within $[P_3(t_0), P_4(t_0))$, then the state transition from the current state m_2h_0 to the next state m_1h_0 (see (12)) occurs at $t = t_0 + t_l$, and the number of channels in states m_2h_0 and m_1h_0 is updated as $N_{m_2h_0} := N_{m_2h_0} - 1$ and $N_{m_1h_0} := N_{m_1h_0} + 1$. This algorithm continues by returning to lifetime generation, until time reaches t_{max}. Note that the number of channels in each state should be updated even if the lifetime generated would be less than the next sampling time for action potential generation.

3.4 Numerical examples

3.4.1 Single node model

A mammalian single node model was represented as the first-order ordinary differential equation described in (6) with the following parameters set at: C_m=1.5 pF, R_m=23.3 MΩ, γ_{Na}=25.69 pS, E_{Na}=66 mV, E_{rest}=-78 mV, and the maximum number of sodium channels N_{Na}^{max}=1000. The differential equation was solved using forward Euler integration at $1\mu s$ steps implemented in C on an IBM compatible PC and in JAVA applets for demonstration on the Internet.

We implemented and compared the approximation algorithm by Fox (F), the exact CST algorithm based on Clay and DeFelice (1983) by Rubinstein (R) and Strassberg and DeFelice (SD), the exact CNT algorithm by Chow and White (CW), and the deterministic model by Hodgkin and Huxley (H). The critical output of each algorithm is the calculation of the number of sodium channels activated, N_{Na}.

H algorithm : The number of sodium channels activated is represented by:

$$N_{Na}(t) = N_{Na}^{max} m^3(t) h(t) \tag{13}$$

where

$$\begin{aligned} \frac{dm(t)}{dt} &= \alpha_m(1 - m(t)) - \beta_m m(t) \\ \frac{dh(t)}{dt} &= \alpha_h(1 - h(t)) - \beta_h h(t) \end{aligned} \tag{14}$$

F algorithm : The number of sodium channels activated is expressed as:

$$N_{Na}(t) = N_{Na}^{max} m^3(t) h(t) \tag{15}$$

Introducing the perturbations $g_m(t)$ and $g_h(t)$ to $m(t)$ and $h(t)$ kinetics yields:

$$\frac{dm(t)}{dt} = \alpha_m(1 - m(t)) - \beta_m m(t) + g_m(t) \tag{16}$$

$$\frac{dh(t)}{dt} = \alpha_h(1 - h(t)) - \beta_h h(t) + g_h(t) \tag{17}$$

where the perturbations are sampled from Gaussian probability density functions, $g_m(t) \approx N(0, \sigma_m^2(t))$, $g_h(t) \approx N(0, \sigma_h^2(t))$ with variance:

$$\sigma_m^2(t) = \frac{2}{N_{Na}^{max}} \frac{\alpha_m(t)\beta_m(t)}{\alpha_m(t) + \beta_m(t)} \tag{18}$$

$$\sigma_h^2(t) = \frac{2}{N_{Na}^{max}} \frac{\alpha_h(t)\beta_h(t)}{\alpha_h(t) + \beta_h(t)} \tag{19}$$

Note that $m(t)$'s and $h(t)$'s are truncated so as to take a value between 0 and 1.

R algorithm : Introducing two states Markov process to each particle, $m_1(t), m_2(t), m_3(t), h(t)$, i.e.,

$$\begin{array}{cccc} C_{m_1} & C_{m_2} & C_{m_3} & C_h \\ \alpha_m \downarrow\uparrow \beta_m & \alpha_m \downarrow\uparrow \beta_m & \alpha_m \downarrow\uparrow \beta_m & \alpha_h \downarrow\uparrow \beta_h \\ O_{m_1} & O_{m_2} & O_{m_3} & O_h \end{array} \tag{20}$$

yields the number of sodium channels activated:

$$N_{Na}(t) = \sum_{k=1}^{N_{Na}^{max}} \tilde{m}_{1,k}(t)\tilde{m}_{2,k}(t)\tilde{m}_{3,k}(t)\tilde{h}_k(t) \tag{21}$$

where $\tilde{m}_{1,k}(t)$, $\tilde{m}_{2,k}(t)$, $\tilde{m}_{3,k}(t)$,$\tilde{h}_k(t)$ designate the Markov jumping processes taking zero(Closed) or one(Open) values.

SD algorithm : The number of channels activated is identical to the number of sodium ion channels activated at the state m_3h_1 shown in (1).

$$N_{Na}(t) = N_{m_3h_1}(t) \tag{22}$$

Note that S and R algorithms are classified into the CST algorithm of the exact algorithm. SD algorithm is based explicitly upon the eight states kinetic scheme described in (1), while R algorithm utilizes four particles equivalent implicitly to the eight state kinetic scheme.

CW algorithm : The number of channels activated is identical to the number of sodium channels activated at the state m_3h_1 shown in (1).

$$N_{Na}(t) = N_{m_3h_1}(t) \tag{23}$$

Note that although CW algorithm looks similar to SD algorithm, calculation is quite different, since CW algorithm is categorized into the CNT algorithm whereas SD algorithm is classified to the CST algorithm.

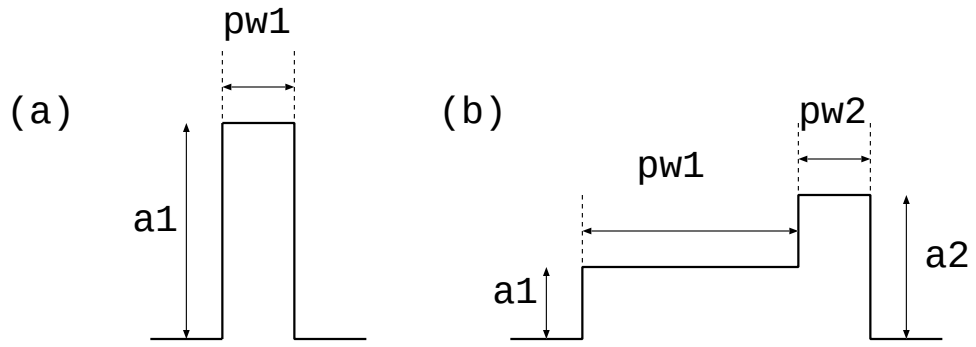

Figure 1: Stimulus waveform: monophasic in (a) where a_1 and pw_1 respectively denote the stimulus current intensity and duration, and pre-conditioned monophasic in (b), where a_1, pw_1, a_2, and pw_2 stand for the pre-conditioned current intensity and duration, and the suprathreshold current intensity and duration.

Two kinds of stimuli, monophasic stimulus and pre-conditioned stimulus, were applied intracellularly to the single node model mentioned above in order to compare the algorithms with regards to the simulation of neural properties. The monophasic stimulus was used for investigating the fundamental statistical parameters, while the pre-conditioned stimulus was utilized for differentiating temporal properties of the algorithms since the small number of sodium channels are expected to generate current fluctuations even at subthreshold levels and thereby to stochastically change temporal properties.

3.4.2 Monophasic stimulus

The monophasic stimulus pulse, as shown in Figure 1(a), was applied intracellularly as $I_{app}(t)$ to the single node in (6). The application of stimuli was repeated one thousand times in order to summarize the neural response properties such as Post Stimulus Time Histogram (PSTH), Firing Efficiency (FE), Latency (LT), and Jitter (JT).

Figure 2 shows ten sample paths of the transmembrane potentials (left column) and the PSTH (right column) generated from one thousand Monte Carlo runs for five different algorithms in which the stimulus parameters in Figure 1(a) were set as follows: the stimulus current intensity a_1=6.2 pA and the duration of monophasic pulse pw_1=100 μs. With the exception of the output of the H algorithm, these transmembrane potentials, PSTHs,

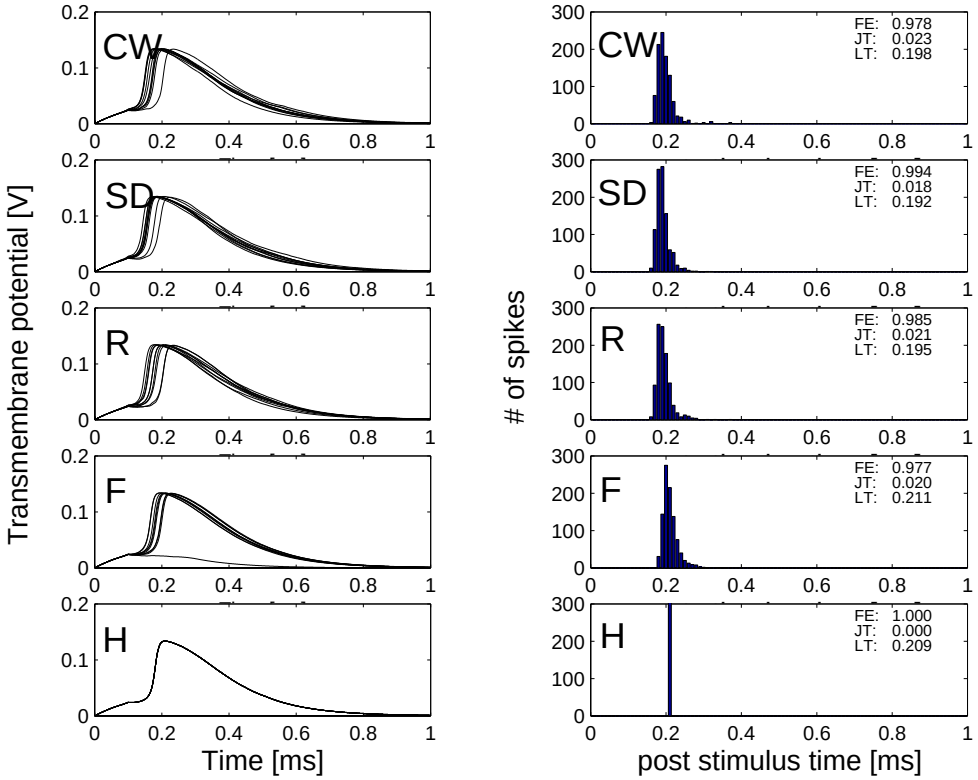

Figure 2: Transmembrane potentials in response to 10 identical monotonic stimulus pulses with an amplitude of 6.2 pA and a duration of 100 μs (left) and post-stimulus time histograms generated from 1000 Monte Carlo runs (right), where FE (Firing Efficiency), JT (Jitter), and LT (Latency) are shown in each inset.

and firing statistics look identical. Note that the H algorithm cannot create jittery occurrences of action potentials because it does not include a stochastic component. As a result, its JT value is 0.0, as shown in the inset of the right-bottom PSTH trace.

Figure 3 shows FE (top), LT(middle), JT(bottom) as a function of stimulus current intensity for four algorithms based on stochastic models in which data of CW, SD, R, F algorithms are respectively plotted by the marks $*$, $\times$, $+$, and $\circ$. The output of the H algorithm is not shown since it is deterministic. Each point was generated from one thousand Monte Carlo runs under the condition of monophasic stimulus pulse. The duration were set at 100 μs as well as that in Figure 2, whereas the stimulus current intensity was varied between 5.2 and 6.4 pA in order to obtain the statistical parameters. For FE curve, these data were fit by the simplex method with appropriate starting values to a Gaussian probability distribution function. These curves are comparable except for that of the F algorithm. The threshold current intensity I_{th} and the relative spread (RS; defined as coefficient of variations) of F algorithm are also quite different from those of CW, SD, R algorithms, as is shown in the inset of FE curves. The latency and jitter curves were drawn by straight lines between data points, and the curves of the F algorithm are not consistent with those of the CW, SD, R algorithms

3.4.3 Pre-conditioned stimulus

A pre-conditioned stimulus pulse, as shown in Figure 1(b), was applied to the single node in (6) in order to investigate temporal dynamics of action potential initiations. The stimulus parameters in Figure 1(b) were set as follows: for the subthreshold stimulus the current intensity a_1=2.5 pA with the duration of pw_1=500 μs, and for the suprathreshold stimuli the current intensity a_2=3.5 pA with the duration of pw_2=100 μs. The application of stimuli was again repeated one thousand times. Figure 4 shows ten sample paths of the transmembrane potentials (left column) and the PSTH (right column) generated from one thousand Monte Carlo runs for five different algorithms. The transmembrane potentials and PSTHs in CW, SD, R algorithm look identical as well as the statistical parameters of FE, JT, and LT shown in the inset of PSTHs. The data produced by the F algorithm are quite different from those of the CW, SD, R algorithms, especially for the LT and JT data shown in the inset of PSTHs. Rather, LT data of F algorithm are comparable to that of the H algorithm.

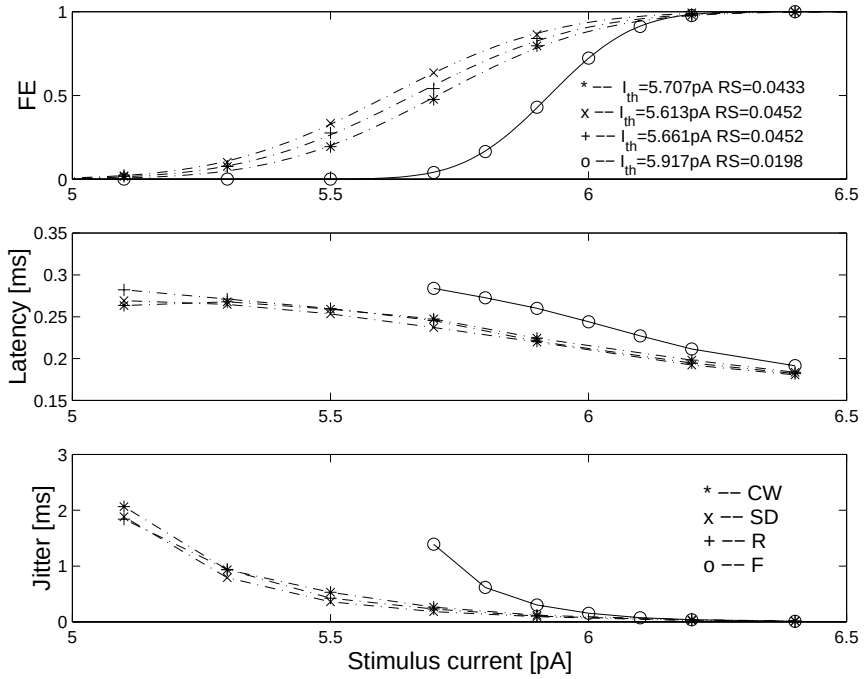

Figure 3: Firing Efficiency (FE)(top), latency (middle), and jitter (bottom) as a function of stimulus current intensity for four algorithms. Stimulus duration was 100 μs.

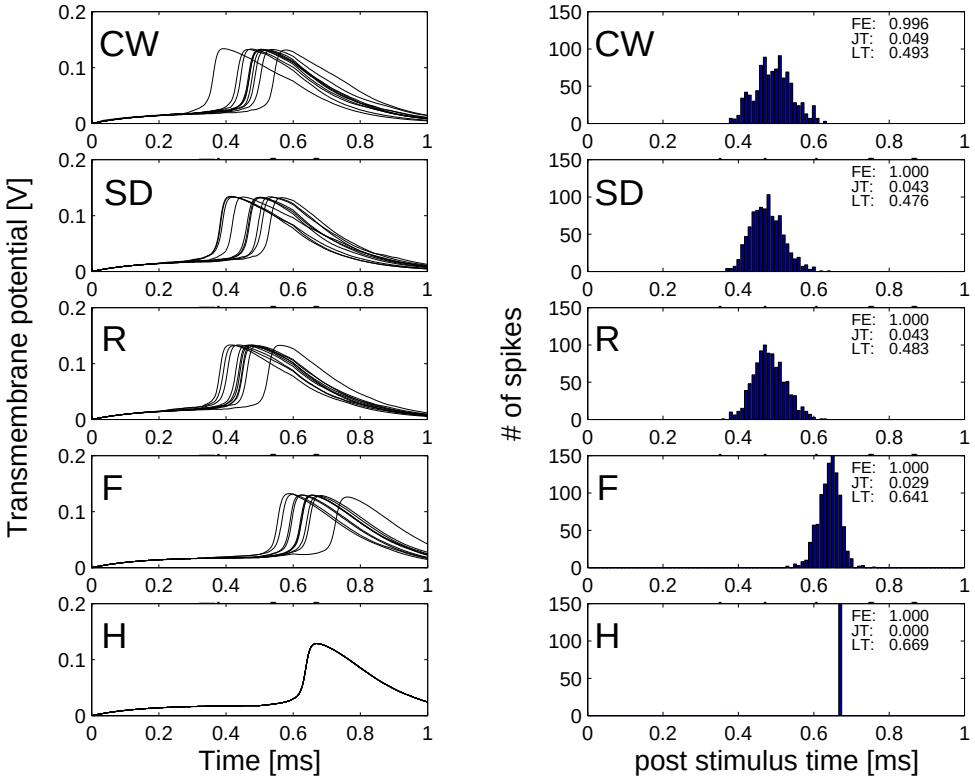

Figure 4: Transmembrane potentials in response to 10 identical stimulus pulses conditioned (left). Post-stimulus time histograms given from 1000 Monte Carlo runs (right). The subthreshold stimulus current of 2.5 pA was applied initially for a duration of 500 μs, followed by a stimulus with an amplitude of 3.5 pA and a duration of 100 μs.

3.4.4 Computation time

The computation time consumed was compared under the case where twenty transmembrane potentials were generated by the CW, SD, R, and F algorithms. Table 1 shows the computation time relative to that of F algorithm.

Table 1: Relative computation time of simulation algorithms

CW	SD	R	F
7	32	39	1

As is intuitively seen, the approximation algorithm was the fastest, while among the exact algorithms the CNT algorithm was faster than the CST algorithm. The most computationally most efficient algorithm among the exact algorithms was shown to be the CW algorithm.

4 Discussion

In this fourth QPR, we have presented performance of the simulation algorithms generating action potentials together with sodium current fluctuations, comparing four stochastic algorithms in terms of the statistical parameters characterizing neural excitabilities and the computation time consumed.

The statistical parameters characterizing neural excitabilities were in good agreement among the exact algorithms (the CW, SD, R algorithms). In particular, the FE, latency, and jitter parameters as a function of stimulus current intensity in the exact algorithm were shown to be very close to each other, as shown in Figure 3. Also, temporal dynamical responses looked identical among the exact algorithms, even if the pre-conditioned stimuli were presented, as is shown in Figure 4. These facts imply that although the ways to implement in each algorithm are different, the essential principle is the same for the CW, SD, R algorithms.

However, the property of neural excitabilities in the approximation algorithm was not in agreement with that of the exact algorithms. Especially, the FE, latency, jitter curves as a function of stimulus current intensity were not similar to those in the exact algorithm. Furthermore, when the pre-conditioned stimuli were presented, the shape of PSTHs, i.e., the latency and jitter parameters, was quite different from those of the exact algorithms, as shown in Figure 4. Rather, the latency parameter was pretty much similar to that of the deterministic H algorithm which does not have any jitter. This phenomenon might be due to that the approximation algorithm was modified from the deterministic H algorithm adding stochastic perturbations, not based on the framework of Markov jumping processes with the eight states kinetic scheme described in (1).

Therefore, we conclude that the most computationally efficient algorithm having appropriate properties of neural excitabilities is the CNT algorithm developed by Gillespie, and implemented by Chow and White, looking at the consumed computation time shown in Table 1. Also, since the computation time of the CW algorithm was approximately 5 to 6 times faster than the R algorithm, the result encouraged us to incorporate the CW algorithm into a distributed axon model. This has been performed in the 3rd QPR of The Neurophysiological Effects of Simulated Auditory Prosthesis Stimulation.

5 Plans for the next quarter

In the fifth quarter, we plan to do the following:

- Initiate data collection with new laboratory system.
- Complete modeling of hair-cell effects on electrical stimulation to compare with physiological data presented in the QPR3.
- Continue data collection with acoustic and electrical stimulation.
- Two invited presentations will be given at the World Congress on Biomedical Engineering in Chicago

6 Appendix: Presentations and publications

The following presentations were made:

- Abbas, PJ, Miller, CA, Brown, CJ, Rubinstein, JT & Hughes, M (2000). Electrophysiological measures with cochlear implants: basic response properties, "Otolaryngology - Head and Neck Surgery for the next Century", University of Iowa, April, 2000.

- Abbas, PJ,, Brown, CJ, Hughes, ML, Wahl, B, & Gehringer, A (2000). Measurements of electrically eoveoked responses to pulse trains using neural response telemetry. 5th European Symposium on Paediatric Cochlear Implantation, Antwerp, June 2000.

- Rubinstein, J.T, R.S. Tyler, K. Gfeller, A. Wolaver, M. Lowder, M. Mehr, C.J. Brown. High-rate conditioning pulses: Effects on speech, music & tinnitus perception. 5th European Symposium on Paediatric Cochlear Implantation, Antwerp, June 2000.

References

[1] Chow, C. C. and White, J. A. (1996) Spontaneous Action Potentials due to Channel Fluctuations. *Biophys. J.*, Vol. 71, pp.3013-3021.

[2] Clay, J. R. and DeFelice, L. J. (1983) Relationship between Membrane Excitability and Single Channel Open-Close Kinetics. *Biophys. J.*, Vol. 42, pp.151-157.

[3] Colquhoun, D., Hawkes, A.G. (1977) Relaxation and fluctuations of membrane currents that flow through drug-operated channels. *Proc. R. Soc. Lond.*, Ser B. Vol.199, pp.231-262.

[4] Colquhoun, D., Hawkes, A.G. (1981) On the stochastic properties of single ion channels. *Proc. R. Soc. Lond.*, Ser B. Vol.211, pp.205-235.

[5] Fox, R.F. (1997) Stochastic Versions of the Hodgkin-Huxley Equations. *Biophys. J.*, Vol.72, pp.2069-2074.

[6] Gillespie, D.T. (1977) Exact Stochastic Simulation of Coupled Chemical Reactions. *J. of Physical Chemistry*, Vol.81, pp.2340-2361.

[7] Hodgkin, A. L., Huxley, A .F. (1952) The Dual Effect of Membrane Potential on Sodium Conductance in the Giant Axon of Loligo. *J. Physiol.(Lond.)*, Vol. 116, pp.497-506.

[8] Neher, E., Stevens, C. F. (1977) Conductance Fluctuations and Ionic Pores in Membranes, *Ann. Rev. Biophys. Bioeng.*, Vol. 6, pp.249-273.

[9] Papoulis, A. (1985) *Probability, Random Variables, and Stochastic Processes*. Singapore: McGraw-Hill, Second edition, p. 392 - 393.

[10] Rubinstein, J.T. (1995) Threshold Fluctuations in an N Sodium Channel Model of the Node of Ranvier. *Biophys. J.* Vol.68, pp.779-785.

[11] Skaugen, E., Walloe, L. (1979) Firing Behaviour in a Stochastic Nerve Membrane Model Based upon the Hodgkin-Huxley Equations. *Acta. Physiol. Scand.* Vol.49, pp.343-363.

[12] Strassberg, A.F. and DeFelice, L.J. (1993) Limitation of the Hodgkin-Huxley Formalism: Effects of Single Channel Kinetics on Transmembrane Voltage Dynamics, *Neural Computation*, Vol.5, pp.843-855.

[13] White J.A., Rubinstein J.T., Kay A.R. (2000). Channel noise in neurons. *Trends. Neurosci.* 23, 131–137.