Journal of Theoretical Medicine, Vol. 6, No. 4, December 2005, 199–234

Abstract action potential models for toxin recognition

JAMES PETERSON* and TAUFIQUAR KHAN

Department of Mathematical Sciences, Clemson University, Clemson, SC, USA

(Received 14 December 2005)

In this paper, we present a robust methodology using mathematical pattern recognition schemes to detect and classify events in action potentials for recognizing toxins in biological cells. We focus on event detection in action potential via abstraction of information content into a low dimensional feature vector within the constrained computational environment of a biosensor. We use generated families of action potentials from a classic Hodgkin–Huxley model to verify our methodology and build toxin recognition engines. We demonstrate that good recognition rates are achievable with our methodology.

Keywords: Feature classification; Feature recognition; Biological feature vector; Toxin model

1. Introduction

In this paper, we consider the problem of toxin recognition using voltage clamp action potential classification by a hybrid excitable nerve cell/silicon substrate with a MOSFET based signal collection package. To do this, we need a theoretical model of the input/output (I/O) process of a single excitable cell that is as simple as possible yet still computes the salient characteristics of our proposed system. We focus on the general structure of a typical action potential as shown in figure 1.

This wave form is idealized and the actual measured action potentials will be altered by noise and the extraneous transients endemic to the measurement process in the laboratory. We know biotoxins alter the shape of the action potential of an excitable cell in many ways. The toxin guide of Adams and Swanson (1996) shows quite clearly, the variety of ways that a toxin can alter ion gate function in the cell membrane, second messenger cascades and so forth. Older material from Kaul and Daftari (1986), Wu and Narahashi (1988) and Schiavo *et al.* (2000) focus on the details of specific classes of toxins. Kaul and Wu focus on pharmacological active substances from the sea. Schiavo investigates the effects of toxins that interfere with the release of neurotransmitters (NTs) by altering the exocytosis process. The effects of toxins on the presynaptic side are analysed in Harvey (1990) and Strichartz *et al.* (1987) presents a summary of how toxins act on sodium channels. In our research, we only look at a subset of these possible interactions. We deliberately focus on a single action potential. We posit that the toxins introduced into the input side of the biosensor elicit single pulse outputs. We think this is a reasonable idealization as current technology focuses on certain types of pyramidal cells which are usually not in bursting mode in the sensor application. Also, the generic biosensor generally lacks a real dendritic field as it is difficult to encourage a cell to grow its normal dendritic structure on the silicon substrate and the axon is quite simple in comparison to what it would be *in situ*. It is important to note that our modelling and interpretation efforts are constrained to fit into this more limited excitable nerve cell environment and for this reason, it is not necessary to model the full characteristics of such a cell. We will focus on four distinct methods for feature vector extraction and recognition engine construction: the classic covariance technique, a total variation and an associated spline method that uses knots determined by the total variation approach and finally, a method that extracts a low dimensional feature vector from the raw action potentials by looking at biologically relevant information such as the maximum time and amplitude of the voltage peak and so forth.

In order to demonstrate the efficacy of this methodology, we will generate families of action potentials from a classic Hodgkin–Huxley model. We assume a toxin in a given family alters the action potential in a specific way which we will call the *toxin family signature*. We will study two type of signatures: first, families that alter the maximum sodium and potassium conductance parameters by a given percentage and second, families that perturb the standard Hodgkin–Huxley α–β values. We deliberately focus on the design of recognizers that will be feasible to

*Corresponding author. Email: petersj@clemson.edu

Journal of Theoretical Medicine
ISSN 1027-3662 print/ISSN 1607-8578 online © 2005 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/10273660500533898

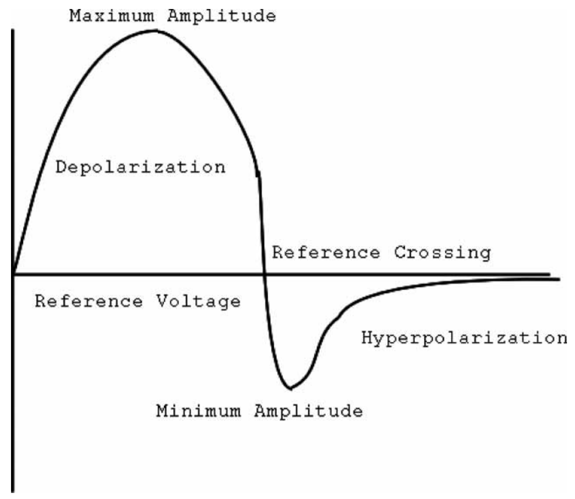

Figure 1. Prototypical action potential.

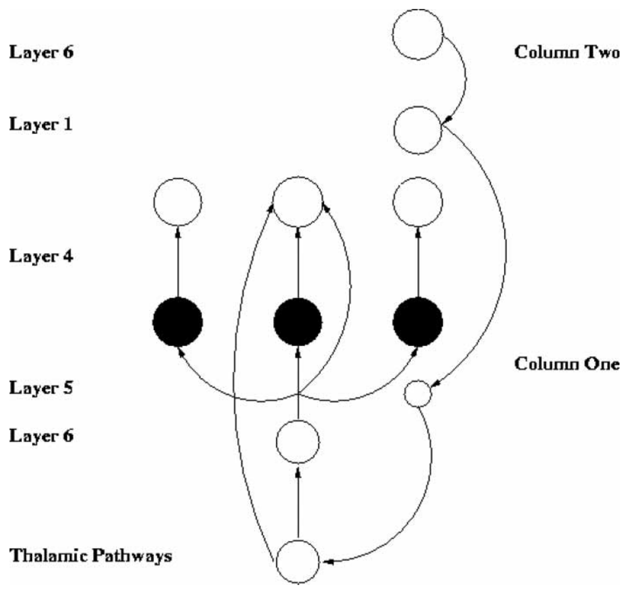

Figure 3. The folded feedback pathway circuit.

implement in a hybrid biological cell/electronic sensor package with limited on-board computational power for feature vector extraction and classification such as the first and second generation biosensors that are being developed.

We also have another motivation for this work. In other work on cognitive modelling, the interactions of cortical modules with the limbic system are modulated by a number of monoamime NTs. A useful model of generic cortex, isocortex, is that given in Grossberg (2003), Grossberg and Seitz (2003) and Raizada and Grossberg (2003). Two fundamental cortical circuits are introduced in these works: the on-centre, off-surround (OCOS) and the folded feedback pathway (FFP) seen in figures 2 and 3. We see that higher level cortical input is fed into the previous column's layer one and then via an OCOS circuit. Outputs from the thalamus (perhaps from the nuclei of the Lateral Geniculate Body) filter upward into the column at the bottom of the picture. At the top of the figure, the three

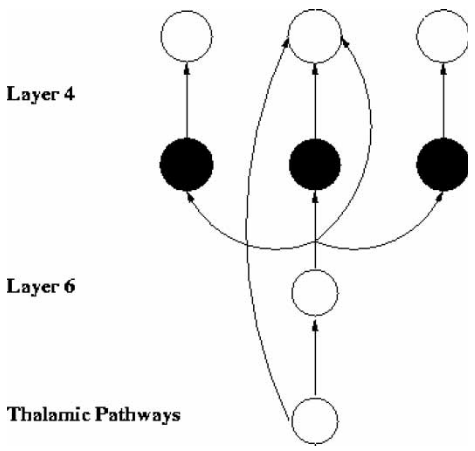

Figure 2. The on-centre, off-surround circuit.

circles that are not filled in represent neurons in layer four whose outputs will be sent to other parts of the column. There are two thalamic output lines: the first is a direct connection to the input layer four, while the second is an indirect connection to layer six itself. This connection then connects to a layer of inhibitory neurons, which are shown as circles filled in with black. The middle layer four output neuron is thus innervated by both inhibitory and excitatory inputs while the left and right layer four output neurons only receive inhibitory impulses. Hence, the centre is excited and the part of the circuit that is off the centre, is inhibited. We could say the surround is off. It is common to call this type of activation the off surround.

Next consider a stacked cortical column consisting of two columns, column one and column two. There are cortico-cortical feedback axons originating in layer six of column two which input into layer one of column one. From layer one, the input connects to the dendrites of layer five pyramidal neurons, which connects to the thalamic neuron in layer six. Hence, the "higher level cortical input" is fed back into the previous column layer six and then can excite column one's fourth layer via the on-centre, off-surround circuit discussed previously. This description is summarized in figure 3. We call this type of feedback a FFP. The OCOS and FFP are the two most basic cortical circuits, but even though there are combined in even more complicated modules, it is clear already that input from areas outside the cortex is important in how the cortical response is shaped.

The reticular formation (RF) is this central core of the brain stem, which contains neurons whose connectivity is characterized by huge fan-in and -out. Reticular neurons therefore have extensive and complex axonal projections. Note, we use the abbreviation *PAG* to denote the cells known as periaqueductal gray and *NG* for the Nucleus

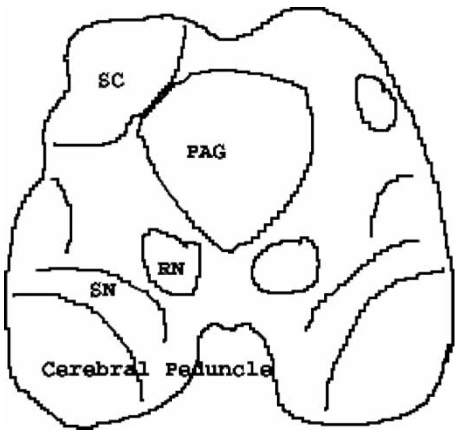

Figure 4. The dopamine neuron sites.

Gracilis neurons. Its neurons have ascending projections that terminate in the thalamus, subthalamus, hypothalamus, cerebral cortex and basal ganglia (caudate nucleus, putamen, globus pallidus, substantia nigra). The midbrain and rostral pons *RF* neurons thus collect sensory modalities, project this information to intralaminar nuclei of thalamus. The intralaminar nuclei project to widespread areas of cortex causes heightened arousal in response to sensory stimuli; e.g. attention. Our cortical column models must therefore be able to accept modulatory inputs from the *RF* formation. For example, dopaminergic neurons in the midbrain (located in figure 4) influence many areas of the brain as shown in figure 5.

These neurons project in overlapping fiber tracts to other parts of the brain. The nigrostriatal sends information to the substantia nigra and then to the caudate, putamen and midbrain. The medial forebrain bundle projects from the substantia nigra to the frontal and limbic lobes. The indicated projections to the motor cortex are consistent with initiation of movement. We know there is disruption to cortex function due to dopamine neurons ablation in Parkinson's disease. The projections to other frontal cortical areas and limbic structures imply there is a motivation and cognition role. Hence, imbalances in these pathways will play a role in mental dsyfunction. Furthermore, certain drugs cause dopamine release in limbic structures, which implies a pleasure connection.

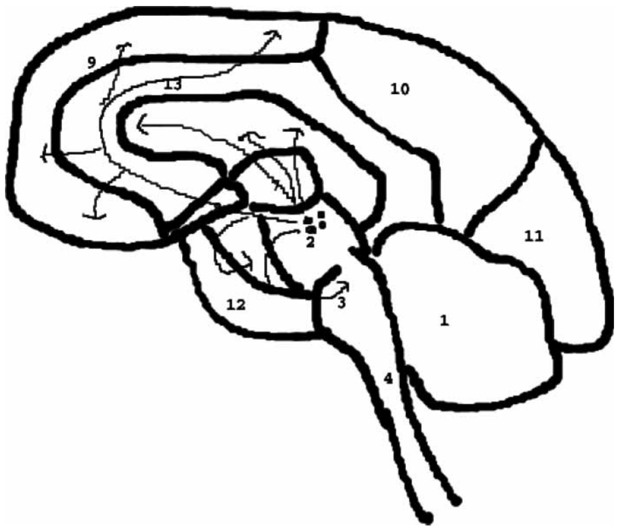

Figure 5. The dopamine innervation pathways.

It is, therefore, clear that cognitive models require abstract neurons whose output can be shaped by many modulatory inputs. Hence, we are interested in how much information can be transferred from one abstract neuron to another using a low dimensional biologically based feature vector (BFV). The toxin study of this paper shows we can successfully use such a BFV to extract information about how a given toxin influences the shape of the output pulse of an excitable neuron. The BFV can then also serve that purpose for other modulatory inputs such as NTs. Thus, we can infer from these studies, that the BFV can be used for information transfer the other way. This allows us to design an artificial neuron whose output is the BFV. Modulatory inputs into the abstract neuron can be modelled as alterations to the components of the BFV. It is then straightforward to develop an algebra of BFV interactions so that we can model cortical interactions modulated by limbic system input. We are pursuing this approach in current cognitive modelling efforts.

The outline of the paper is as follows. First, in section 2, we discuss the basic Hodgkin–Huxley model for the reader who may not be familiar with the cell modelling literature. The Hodgkin–Huxley model depends on a large number of parameters and we will be using perturbations of these as a way to model families of toxins. Of course, more sophisticated action potential models can be used, but the standard two ion gate Hodgkin–Huxley model is sufficient for our needs in this paper. In section 3, we present the toxin recognition methodology for the first toxin family that modify the maximal sodium and potassium conductances. We believe that toxins of this sort include some second messenger effects. Our general classification methodology is then discussed in section 4 and applied to the this collection of toxin families. We show that we can design a reasonable recognizer engine using a low dimensional biologically BFV. In section 5, we introduce the second class of toxin families whose effect on the action potential is more subtle. We show that the biological feature vector performs well in this case also. Finally, in section 6, we make concluding remarks and discuss future directions of our work.

2. The basic Hodgkin–Huxley model

The salient variables needed to describe what is happening inside and outside the cellular membrane for a standard cable model Johnston and Wu (1995) are given in table 1.

Table 1. Hodgkin–Huxley variable units.

Variable	Meaning	Units
V_m	Membrane potential	MV
K_m	Membrane current per length	nA/cm
K_e	Externally applied current	nA/cm
I_i	Inner current	NA
I_o	Outer current	NA
I_e	External current	NA
I_m	Membrane current	NA
V_i	Inner voltage	MV
V_o	Outer voltage	MV
r_i	Resistance inner fluid per length	μohms/cm
r_o	Resistance outer fluid per length	μohms/cm
g_m	Membrane conductance per length	μSiemens/cm
c_m	Membrane capacitance per length	nano Fahrads/cm
G_M	Membrane conductance	μSiemens/cm
C_M	Membrane capacitance	nano Fahrads/cm

The cable equations are generally expressed in a per length form as the variables in table 1 show. The current density variables are traditionally denoted as K's and the corresponding currents as I's. It is also traditional to use lower case letters for the membrane capacitance and conductance densities, with upper case letters for the total values. The membrane voltage can be shown to satisfy the partial equation (1) whose dimensional analysis shows has units of volts/(cm^2).

$$\frac{\partial^2 V_m}{\partial z^2} = (r_i + r_o)K_m - r_o K_e. \qquad (1)$$

A realistic description of how the membrane activity contributes to the membrane voltage uses models of ion flow controlled by gates in the membrane. A simple model of this sort is based on Hodgkin and Huxley (1952), Hodgkin (1952), (1954). We start by expanding the membrane model to handle potassium, sodium and an all purpose current, called leakage current, using a modification of our original simple electrical circuit model of the membrane. We will think of a gate in the membrane as having an intrinsic resistance and the cell membrane itself as having an intrinsic capacitance. Thus, we expand the single branch of our old circuit model to multiple branches—one for each ion flow we wish to model. The ionic current consists of the portions due to potassium, K_k, sodium, K_{Na} and leakage K_L. The leakage current is due to all other sources of ion flow across the membrane which are not being explicitly modelled. This would include ion pumps; gates for other ions such as Calcium, Chlorine; NT activated gates and so forth. We will assume that the leakage current is chosen so that there is no excitable neural activity at equilibrium. The standard Hodgkin–Huxley model of an excitatory neuron then consists of the equation for the total membrane current density, K_m, obtained from Ohm's law:

$$K_m = c_m \frac{\partial V_m}{\partial t} + K_K + K_{Na} + K_L. \qquad (2)$$

The new equation for the membrane voltage is thus

$$\frac{\partial^2 V_m}{\partial z^2} = (r_i + r_o)c_m \frac{\partial V_m}{\partial t} + (r_i + r_o)(K_K + K_{Na} + K_L) + (r_i + r_o) + (r_i + r_o) - r_o K_e.$$

which can be simplified to what is seen in equation (3):

$$\frac{1}{r_i + r_o}\frac{\partial^2 V_m}{\partial z^2} = c_m \frac{\partial V_m}{\partial t} + K_K + K_{Na} + K_L - \frac{r_o}{r_i + r_o} K_e.$$

Under certain experimental conditions (the voltage clamped protocol), we can force the membrane voltage to be independent of the spacial variable z. In this case, we find $\partial^2 V_m/\partial z^2 = 0$ which allows us to rewrite equation (3) as follows

$$c_m \frac{dV_m}{dt} + K_K + K_{Na} + K_L - \frac{r_o}{r_i + r_o} K_e = 0. \qquad (3)$$

The replacement of the partial derivatives with a normal derivative reflects the fact that in the voltage clamped protocol, the membrane voltage depends only on the one variable, time t. Since, c_m is capacitance per unit length, the above equation can also be interpreted in terms of capacitance, C_M, and currents, I_K, I_{Na}, I_L and an external current I_e. This leads to equation (4) which has units of amps.

$$C_m \frac{dV_m}{dt} + I_K + I_{Na} + I_L - \frac{r_o}{r_i + r_o} I_e = 0. \qquad (4)$$

which we note has units of amps. Finally, if we label as external current, I_E, the term $(r_o/r_i + r_o)I_e$, the equation we need to solve under the voltage clamped protocol becomes equation (5).

$$\frac{dV_m}{dt} = \frac{1}{C_m}(I_E - I_K - I_{Na} - I_L). \qquad (5)$$

We can think of a gate as having some intrinsic resistance and capacitance. Now for our simple Hodgkin–Huxley model here, we want to model a sodium and potassium gate as well as the cell capacitance. So, we will have a resistance for both the sodium and potassium. In addition, we know that other ions move across the membrane due to pumps, other gates and so forth. We will temporarily model this additional ion current as a leakage current with its own resistance. We also know that each ion has its own equilibrium potential which is determined by applying the Nernst equation. The driving electromotive force is the difference between the ion equilibrium potential and the voltage across the membrane itself. Hence, if E_c is the equilibrium potential due to ion c and V_m is the membrane potential, the driving force is $V_c - V_m$. In figure 6, we see an electric schematic that summarizes what we have just said. We model the membrane as a parallel circuit with a branch for the

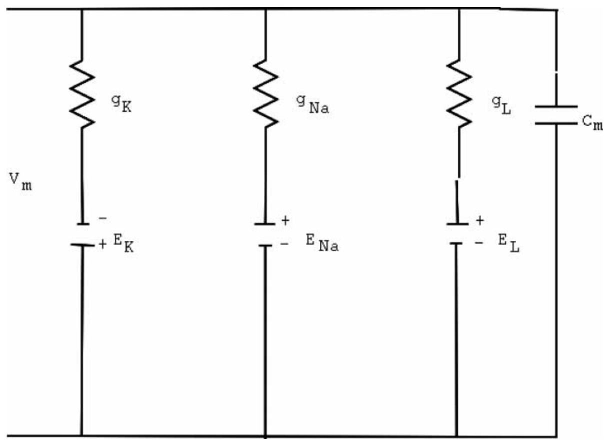

Figure 6. The simple Hodgkin–Huxley membrane circuit model.

sodium and potassium ion, a branch for the leakage current and a branch for the membrane capacitance.

From circuit theory, we know that the charge q across a capacitor is $q = CE$, where C is the capacitance and E is the voltage across the capacitor. Hence, if the capacitance C is a constant, we see that the current through the capacitor is given by the time rate of change of the charge. If the voltage E was also space dependent, we could write $E(z,t)$ to indicate its dependence on both a space variable z and the time t. From Ohm's law, we know that voltage is current times resistance; hence for each ion c, we can say $V_c = I_c R_c$ where we label the voltage, current and resistance due to this ion with the subscript c. This implies $I_c = G_c V_c$, where G_c is the reciprocal resistance or conductance of ion c. Hence, we can model all of our ionic currents using a conductance equation of the form above. Of course, the potassium and sodium conductances are nonlinear functions of the membrane voltage V and time t. This reflects the fact that the amount of current that flows through the membrane for these ions is dependent on the voltage differential across the membrane which in turn is also time dependent. The general functional form for an ion c is thus $I_c = G_c(V,t)(V_m(t) - E_c(t))$ where as we mentioned previously, the driving force, $V_m - E_c$, is the difference between the voltage across the membrane and the equilibrium value for the ion in question, E_c. Note, the ion battery voltage E_c itself might also change in time (for example, extracellular potassium concentration changes over time). Hence, the driving force is time dependent. The conductance is modelled as the product of an activation, m and an inactivation, h, term that are essentially sigmoid nonlinearities. The activation and inactivation are functions of V_m and t also. The conductance is assumed to have the form

$$G_c(V_m, t) = G_0 m^p(V_m, t) h^q(V_m, t),$$

where appropriate powers of p and q are found to match known data for a given ion conductance. We model the leakage current, I_L, as $I_L = G_L(V_m(t) - E_L)$ where the leakage battery voltage, E_L, and the conductance g_L are constants that are data driven. Hence, in terms of current densities, letting g_K, g_{Na} and g_L, respectively denote the ion conductances per length, our full model would be

$$K_K = g_K(V_m - E_K), \quad K_{NA} = g_{Na}(V_m - E_{Na}),$$

$$K_L = g_L(V_m - E_L)$$

We know the membrane voltage satisfies:

$$\frac{1}{r_i + r_o}\frac{\partial^2 V_m}{\partial z^2} = c_m \frac{\partial V_m}{\partial t} + K_K + K_{Na} + K_L - \frac{r_o}{r_i + r_o} K_e.$$

We can, therefore, rewrite this in current density form as

$$\begin{aligned}\frac{1}{r_i + r_o}\frac{\partial^2 V_m}{\partial z^2} = c_m \frac{\partial V_m}{\partial t} &+ g_K(V_m - E_K) \\ &+ g_{Na}(V_m - E_{Na}) + g_L(V_m - E_L) \\ &- \frac{r_o}{r_i + r_o} K_e.\end{aligned}$$

which under the voltage clamped protocol has the current form

$$\begin{aligned}C_m \frac{dV_m}{dt} = I_E - G_K(V_m - E_K) + G_{Na}(V_m - E_{Na}) \\ + G_L(V_m - E_L)\end{aligned}$$

We assume that the voltage dependence of our activation and inactivation has been fitted from data. Hodgkin–Huxley modelled the sodium and potassium gates as

$$G_{Na}(V_m, t) = G_0^{Na} m^3(V_m, t)\, h(V_m, t),$$

$$G_K(V_m, t) = G_0^K n^4(V_m, t).$$

where the activation variables, m and n, and the inactivation variable h satisfy first order kinetics of the form $\tau_x\, dx/dt = x_\infty - x, \quad x_0 = x_0$, where x can be any of the choices m, h or n. The model is further complicated by the voltage dependence of the critical constant pairs (τ_x, x_∞) and so forth. These parameters are computed at any given voltage by $\tau_x = 1.0/(\alpha_x + \beta_x)$ and $x_\infty = \alpha_x/(\alpha_x + \beta_x)$. The needed (α_x, β_x) pairs for each activation/inactivation variable are modelled using curve fits to data as sigmoid type nonlinearities. The classic Hodgkin–Huxley model computed the following curve fits:

$$\begin{aligned}\alpha_m &= -0.10 \frac{V_m + 35.0}{e^{-.1(V_m+35.0)} - 1.0}, \\ \beta_m &= 4.0\, e^{-(V_m+60.0)/18.0}, \\ \alpha_h &= 0.07\, e^{-.05(V_m+60.0)}, \\ \beta_h &= \frac{1.0}{1.0 + e^{-.1(V_m+30.0)}}, \\ \alpha_n &= -\frac{0.01*(V_m + 50.0)}{e^{-.1(V_m+50.0)} - 1.0}, \\ \beta_n &= 0.125 e^{-0.0125(V_m+60.0)}.\end{aligned} \tag{6}$$

Our model of the membrane dynamics here thus consists of the following differential equations:

$$\frac{dV_m}{dt} = \frac{I_M - I_K - I_{Na} - I_L}{C_M}, \quad \tau_m \frac{dm}{dt} = m_\infty - m, \tag{7}$$
$$\tau_h \frac{dh}{dt} = h_\infty - h, \quad \tau_n \frac{dn}{dt} = n_\infty - n,$$

$$V(0) = V_m^0, \quad m(0) = m_\infty(V_m^0, 0), \tag{8}$$
$$h(0) = h_\infty(V_m^0, 0), \quad n(0) = m_\infty(V_m^0, 0),$$

We note that at equilibrium there is no current across the membrane. Thus, the leakage conductance and leakage battery voltage must be adjusted so that current flow is zero at equilibrium. This then implies the sodium and potassium currents are zero and the activation and inactivation variables should achieve their steady state values which would be m_∞,h_∞ and n_∞ computed at the equilibrium membrane potential which is here denoted by V_m^0.

3. Toxin recognition

When two cells interact via a synaptic interface, the electrical signal in the pre-synaptic cell in some circumstances triggers a release of a NT from the pre-synapse which crosses the synaptic cleft and then by docking to a port on the post cell, initiates a post-synaptic cellular response. The general pre-synaptic mechanism consists of several key elements: one, NT synthesis machinery so the NT can be made locally; two, receptors for NT uptake and regulation; three, enzymes that package the NT into vesicles in the pre-synapse membrane for delivery to the cleft. We will focus on two pre-synaptic types: monoamine and peptide. In the monoamine case, all three elements for the pre-cell response are first manufactured in the pre-cell using instructions contained in the pre-cell's genome and shipped to the pre-synapse. Hence, the monoamine pre-synapse does not require further instructions from the pre-cell genome and response is therefore fast. The peptide pre-synapse can only manufacture a peptide NT in the pre-cell genome; if a peptide NT is needed, there is a lag in response time. Also, in the peptide case, there is no re-uptake pump so peptide NT can't be reused. On the post-synaptic side, we will focus on two general responses. The fast response is triggered when the bound NT/receptor complex initiates an immediate change in ion flux through the gate thereby altering the electrical response of the post cell membrane and hence, ultimately its action potential and spike train pattern. Examples are glutumate (excitatory) and GABA (inhibitory) NTs. The slow response occurs when the initiating NT triggers a second response in the interior of the cell. In this case, the first NT is called the first messenger and the intracellular response (quite complex in general) is the second messenger system. Further expository details of first and second messenger systems can be found in Stahl (2000).

From the above discussion, we can infer that a toxin introduced to the input system of an excitable cell influences the action potential produced by such a cell in a variety of ways. For a classic Hodgkin–Huxley model, there are a number of critical parameters which influence the action potential. The nominal values of the maximum sodium and potassium conductance G_{Na}^0 and G_K^0 can be altered. We view these values as a measure of the density of a classic Hodgkin–Huxley voltage activated gates per square cm of biological membrane. Hence, changes in this parameter require the production of new gates or the creation of enzymes that control the creation/destruction balance of the gates. This parameter is thus related to second messenger activity as access to the genome is required to implement this change. We can also perturb the parameters that shape the α and β functions used in equation (7). These functions are all special cases of the general mapping $\zeta(V_m,p,q)$ with $p \in \Re^4$ and $q \in \Re^2$ defined by

$$\zeta(V,p,q) = \frac{p_0(V_m + q_0) + p_1}{e^{p_2(V_m+q_1)} + p_3},$$

with

$$\alpha_m = \zeta(V_m, p_m^\alpha = \{-0.10, 0.0, -0.1, -1.0\},$$
$$q_m^\alpha = \{35.0, 35.0\}),$$

$$\beta_m = \zeta(V_m, p_m^\beta = \{0.0, 4.0, 0.0556, 0.0\},$$
$$q_m^\beta = \{60.0, 60.0\}),$$

$$\alpha_h = \zeta(V_m, p_h^\alpha = \{0.0, 0.07, 0.05, 0.0\},$$
$$q_h^\alpha = \{60.0, 60.0\}),$$

$$\beta_h = \zeta(V_m, p_h^\beta = \{0.0, 1.0, -0.1, 1.0),$$
$$q_h^\beta = \{30.0, 30.0\}),$$

$$\alpha_n = \zeta(V_m, p_n^\alpha = \{0.01, 0.0, -0.1, -1.0\},$$
$$q_n^\alpha = \{50.0, 50.0\}),$$

$$\beta_n = \zeta(V_m, p_n^\beta = \{0.0, 0.125, 0.0125, 0.0\},$$
$$q_n^\beta = \{60.0, 60.0\}).$$

The p and q parameters control the shape of the action potential in a complex way. From our discussions about the structure of ion gates, we could think of alterations in the (p,q) pair associated with a given α and/or β as a way of modelling how passage of ions through the gate are altered by the addition of various ligands. These effects are immediate and changes in ion flow follow implying changes in membrane potential.

Now assume that the standard Hodgkin–Huxley model for $g^0_{\mathrm{Na}} = 120$, $g^0_{\mathrm{K}} = 63.0$ and the classical α, β functions are labeled as the nominal values. Hence, there is a nominal vector Λ_0 given by

$$\Lambda_0 = \begin{bmatrix} G^0_{\mathrm{Na}} & = & 120.0 \\ G^0_{\mathrm{K}} & = & 36.0 \\ (p^{\alpha}_{\mathrm{m}})^0 & & \{-0.10, 0.0, -0.1, -1.0\} \\ (p^{\beta}_{\mathrm{m}})^0 & & \{0.0, 4.0, 0.0556, 0.0\} \\ (p^{\alpha}_{\mathrm{h}})^0 & & \{0.0, 0.07, 0.05, 0.0\} \\ (p^{\beta}_{\mathrm{h}})^0 & & \{0.0, 1.0, -0.1, 1.0\} \\ (p^{\alpha}_{\mathrm{n}})^0 & & \{0.01, 0.0, -0.1, -1.0\} \\ (p^{\beta}_{\mathrm{n}})^0 & & \{0.0, 0.125, 0.0125, 0.0\} \\ (q^{\alpha}_{\mathrm{m}})^0 & & \{35.0, 35.0\} \\ (q^{\beta}_{\mathrm{m}})^0 & & \{60.0, 60.0\} \\ (q^{\alpha}_{\mathrm{h}})^0 & & \{60.0, 60.0\} \\ (q^{\beta}_{\mathrm{h}})^0 & & \{30.0, 30.0\} \\ (q^{\alpha}_{\mathrm{n}})^0 & & \{50.0, 50.0\} \\ (q^{\beta}_{\mathrm{n}})^0 & & \{60.0, 60.0\} \end{bmatrix}$$

A toxin G thus has an associated toxin signature, $\mathcal{E}(G)$ which consists of deviations from the nominal classical Hodgkin–Huxley parameter suite: $\mathcal{E}(G) = \Lambda_0 + \delta$, where δ is a vector, or percentage changes from nominal, that we assume the introduction of the toxin initiates. As you can see, if we model the toxin signature in this way, we have a rich set of possibilities we can use for parametric studies.

3.1 Simple second messenger toxins

We begin with toxins whose signatures are quite simple as they only cause a change in the nominal sodium and potassium maximum conductance. These are, therefore, second messenger toxins. This simulation will generate five toxins whose signatures are distinct. Using C++ as our code base, we have designed a TOXIN class whose constructor generates a family of distinct signatures using a signature as a base. Here, we will generate twenty sample toxin signatures clustered around the given signature using a neighbourhood size of 0.02. First, we generate five toxins using the toxin signatures: of table 2.

Table 2. Toxin conductance signature.

Toxin	Signature $[\delta g^0_{\mathrm{Na}}, \delta g^0_{\mathrm{K}}]$
A	[0:45, −0:25]
B	[0:05, −0:35]
C	[0:10, 0:45]
D	[0:55, 0:70]
E	[0:75, −0:75]

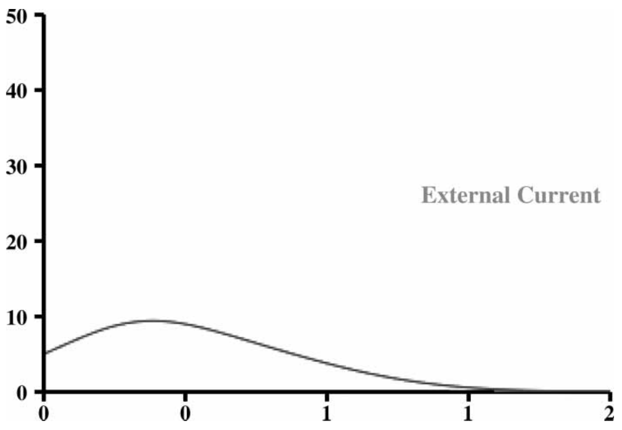

Figure 7. The applied synaptic pulse.

These percentage changes are applied to the base conductance values to generate the sodium, potassium and leakage conductances for each simulation. We use these toxins to generate 100 simulation runs using the 20 members of each toxin family. We believe that the data from this parametric study can be used to divide up action potentials into disjoint classes each of which is generated by a given toxin. Each of these simulation runs use the synaptic current injected as in figure 7 which injects a modest amount of current over approximately 2 s. The current is injected at four separate times to give the gradual rise seen in the picture. For additional details, see Peterson (2002b). In figure 8, we see all 100 generated action potentials. You can clearly see that the different toxin families create distinct action potentials.

4. Building the recognition engines

We can now use the toxin data discussed above to build a toxin recognition engines. There are many approaches to recognition as are discussed in Theodoridis and Koutroumbas (1999) including the use of artifical neural technologies. In this paper, we choose four distinct approaches to build a recognizer. We can

(1) Use the eigenvectors corresponding to the largest eigenvalues of the covariance matrix. We will discuss this below, but essentially we use the average trace

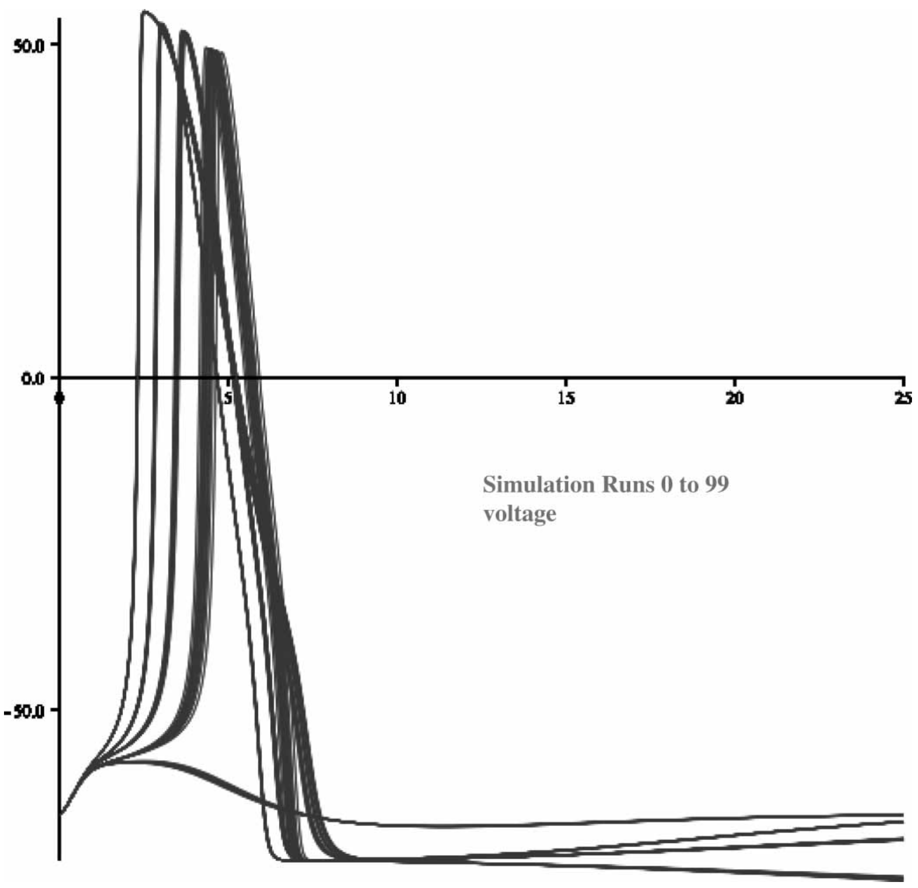

Figure 8. The toxin families for the applied pulse.

for each variable to construct the difference vectors used in the computation of the covariance matrix. This approach tends to remove noise in the data, so we expect it to be robust in the face of errors in measuring a signal. Our simulations add 40% noise to the data to test this. As expected, the recognition of toxins is unaffected by noise in this case.

(2) Use a total variation approach which focuses on the places in the data where the variable in question changes the most. We would expect that the components of the resulting feature vector have some physical meaning. This recognition engine should be adversely affected by noise.

(3) Use a spline approach with the knots suggested by the total variation method. This approach will lack physical meaning and will also be affected by the noise in the data.

(4) Use a biological relevant feature vector obtained by extracting from each variable of interest information that has relevant physical meaning. This is, of course, more subjective, but has the advantage that the components of the feature vector have some physical meaning. Of course, it is important that the extraction algorithm that we use is not sensitive to noise in the raw data.

4.1 The covariance recognizer

A standard covariance recognition algorithm is now described. We assume the data size N is generally much larger than the number of samples M. Here, the simulation generates time traces of the form (t_i,z_i) for any variable z of interest. We compute the variables over 25 mS and generate a waypoint every 0.025 mS; hence, each such vector is size 1001. The number of runs in each simulation is 100. Thus, the restriction that N is much larger than M is clearly followed. For the voltage variable, the data is therefore $\{V_\mathrm{i}, 1 \leq i \leq M\}$, each $V_\mathrm{i} \in \Re^\mathrm{N}$. We store this in the matrix B where the ith row of B is the data V_i and compute the average of each column in B to create an average vector μ. Then, we compute the difference vectors $d_i = v_i - \mu$ for $1 \leq i \leq M$ and store them in the $M \times N$ matrix A. Letting $C^* = (1/M)AA^T$, an $M \times M$ matrix, we compute the M eigenvalues and associated eigenvectors of C^*, $\lambda_1,\ldots,\lambda_M$ and $\{\phi_1,\ldots,\phi_M\}$. The eigenvalues of C^* are known to be the same as the eigenvalues of the much larger $N \times N$ covariance matrix $C = (1/M)A^TA$ with eigenvectors $\Theta_i = A^T\phi_i$, normalized into unit vectors, ζ_i. To construct a simple recognizer, simply choose an integer Q less than or equal to M. The eigenvectors $\{\zeta_1,\ldots,\zeta_Q\}$ corresponding to the top Q eigenvalues then provide a low dimensional subspace of

$\Re^N$, E_Q. Project all difference vectors to E_Q to create a feature vector $f_i = \{\langle d_i,\zeta_1\rangle,\ldots,\langle d_i,\zeta_Q\rangle\}^T$.

For each toxin class there are samples. Assume there are P toxins, P divides M evenly and each class has the same number of samples, $L = M/P$. The feature vectors can thus be organized into classes using the index sets $\{I_1,\ldots,I_p\}$ where each index set I_k consists of the L indices that correspond to toxin class k. Hence, for a given toxin class k, the feature vector set F^k defined as $\{f_i^k, i \in I_k\}$ collects all the samples corresponding to toxin k. If we then compute the average feature vector for each toxin class, $T_k = (1/L)\sum_{i\in I_k} F_i^k$, we have a way to determine if a new sample belongs to one of the toxin classes we have identified. Given a new sample ξ to test, we compute its difference $\delta = \xi - \mu$ and compute the feature vector F_ξ by projecting δ to E_Q. Then, we calculate the euclidean distance from this feature vector to each toxin class vector, $\rho_k = \|F_\xi - T_k\|$, for $1 \le k \le P$. The index i corresponding to the minimum, $\rho_i = \min_k \rho_k$, is considered the correct toxin class for the sample.

Using the data discussed in section 3.1, we can use the algorithm and code above to implement a recognition engine. From the simulation data, we then compute the average variable time traces for the variables of interest. It is instructive to look at a plot of the logarithm of the eigenvalues of the covariance matrix. When the covariance matrix is built without noise, a plot of the logarithm of the resulting eigenvalues is shown in figure 10. There is a clean break in eigenvalue size. When noise is added, the eigenvalues grow in size and separation is diminished, but there is still a significant drop in the eigenvalue size at eigenvalue 5 as shown in figure 10.

In figure 11, we look at two recognizer engines built according to the discussion in section 4; one using a one dimensional ($Q = 1$) and the other, a five dimensional projection ($Q = 5$). In the left panel, we show the results for the recognizer built using the one-dimensional projection; i.e. only the eigenvector corresponding to the maximal eigenvalue of the covariance matrix is used (figure 12). The distance of each toxin to the toxin class feature vectors are plotted in five separate curves. The samples are organized as 20 from each toxin class starting with Toxin A. Hence, for the Toxin A samples, if our recognizer is working well, we expect to see the first 20 distances are very low and the other 80 distances are very high. In the right panel, the results are shown for a recognizer built using a 5 dimensional projection; i.e. the 5 eigenvectors corresponding to the 5 top ranked eigenvalues of the covariance matrix are used. Clearly, these toxin classes are very nicely separated from each other as you can see from the distance plots and we get stable

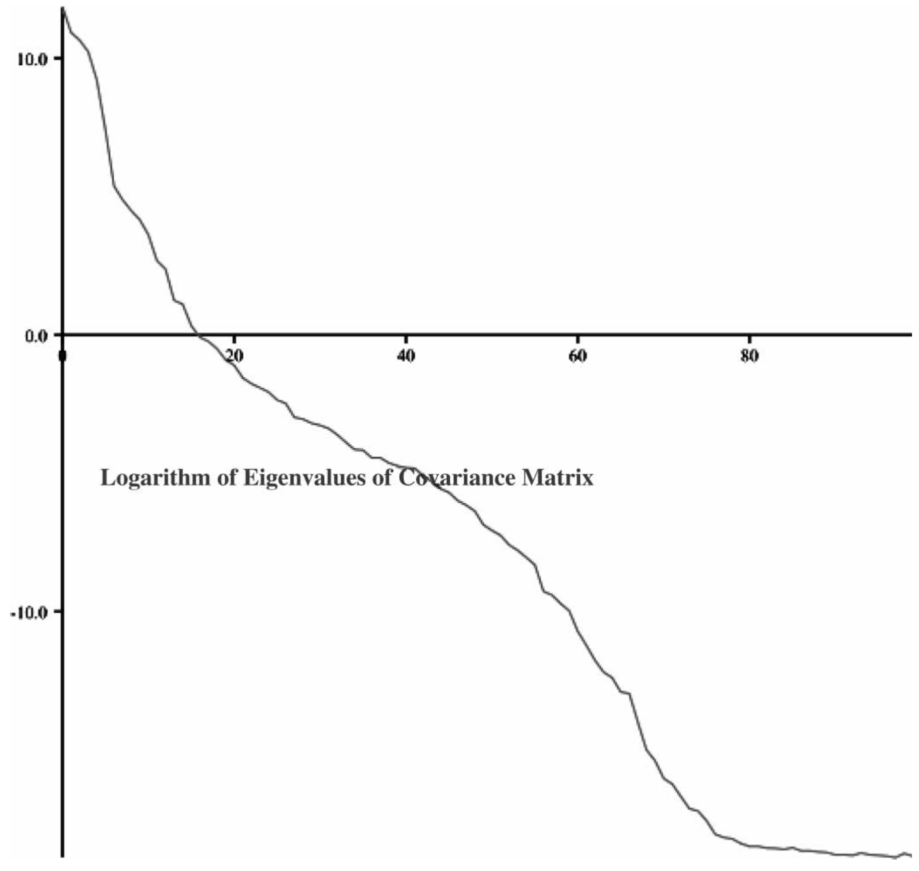

Figure 9. Logarithm of eigenvalues without noise.

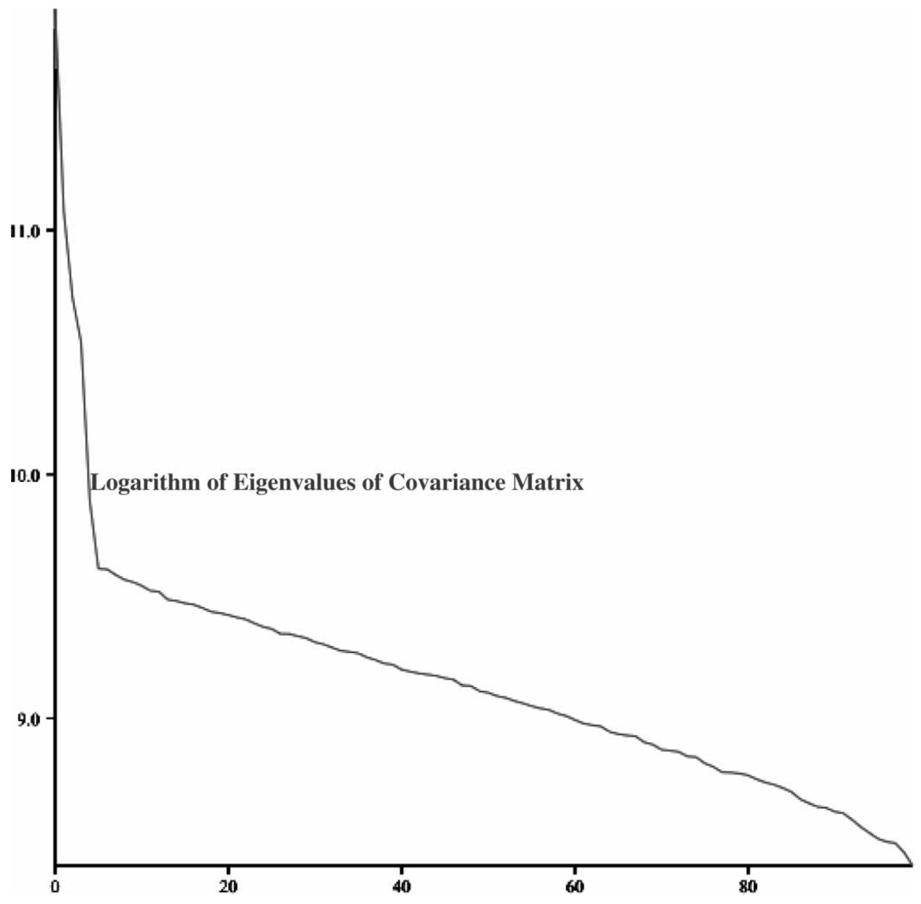

Figure 10. Logarithm of eigenvalues with noise.

recognition even if we simply use the dominant eigenvector direction for our projections. In figures 13 and 14, we see the results of a ten dimensional projection on the data used to build the recognizer—the training data. We also show in the right panel, how the ten dimensional recognizer performs on toxin samples that are generated with a larger neighbourhood window.

In figures 13 and 14, we show results for a recognizer built using a 10 dimensional projection ($Q = 10$); i.e. the 10 eigenvectors corresponding to the 10 top ranked eigenvalues of the covariance matrix are used. The distances within each class are lower as the approximation is better; i.e. if we look at the distance from a Toxin A class sample to the Toxin A feature vector, this distance is lower here in the 10 dimensional approximation. However, the separation between toxins is very good even when we use the 1 dimensional recognizer. In section 3.1, we detailed the construction of the original training samples using a neighbourhood size of 0.02 around each toxin class signature. For the results shown in the right panel, we also generated samples using the neighbourhood size of 0.2, which is ten times larger. This should generate samples in each toxin neighbourhood, which are not in that Toxin class because the toxin signature used exceeds the toxin class specifications. The plot shown in figure 13 shows the distance results for all the samples using the neighbourhood size of 0.02. The plot for the larger neighbourhood size of 0.2 is shown in figure 14. As you can see, some of the samples from each toxin class now have a fairly high distance from their toxin class feature vector.

We will now test the viability of this approach by attempting to build a recognizer from very noisy data. We created the noisy version of the data by adding noise to the data we had generated in section 3.1. The randomly altered data is then used to build a covariance based recognition engine. The covariance matrix is quite different for this data. The first 100 eigenvalues are all quite large, although there is a drop off from the initial high value of eigenvalue 0. We show this in a log eigenvalue plot in figure 9. If you look at eigenvalues 0 to 5, {148378, 64738.3, 45474.3, 37868.7, 19793.3, 14954.3}, you note there is a sharp drop off at eigenvalue 5.

To see how this recognizer performs, in figures 15 and 16, we see the plots of a 5 and 10 dimensional recognizer. In figure 15, we show how the noisy recognizer built using a 5 dimensional projection performs on itself;

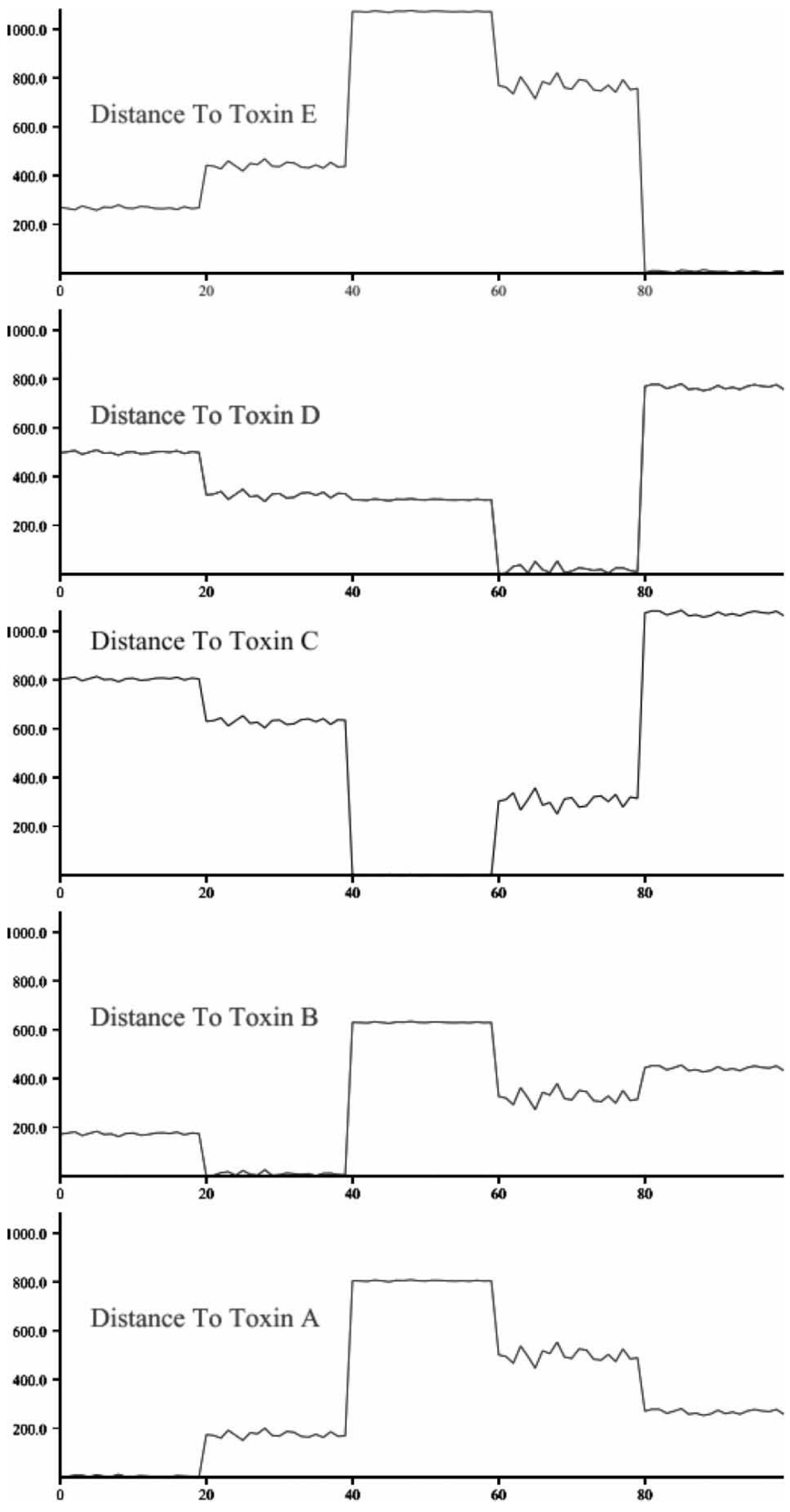

Figure 11. 1D covariance recognizer.

i.e. on the data that was used to build the recognizer. The distance of each toxin to the toxin class feature vector, vectors are plotted in five separate curves as we did in earlier graphs. In figure 16, the results are shown for a recognizer built using a 10 dimensional projection. Despite the fact that the distances to the class feature vectors are larger than in the case, where the recognizer was built for data that was not noisy, there is still good

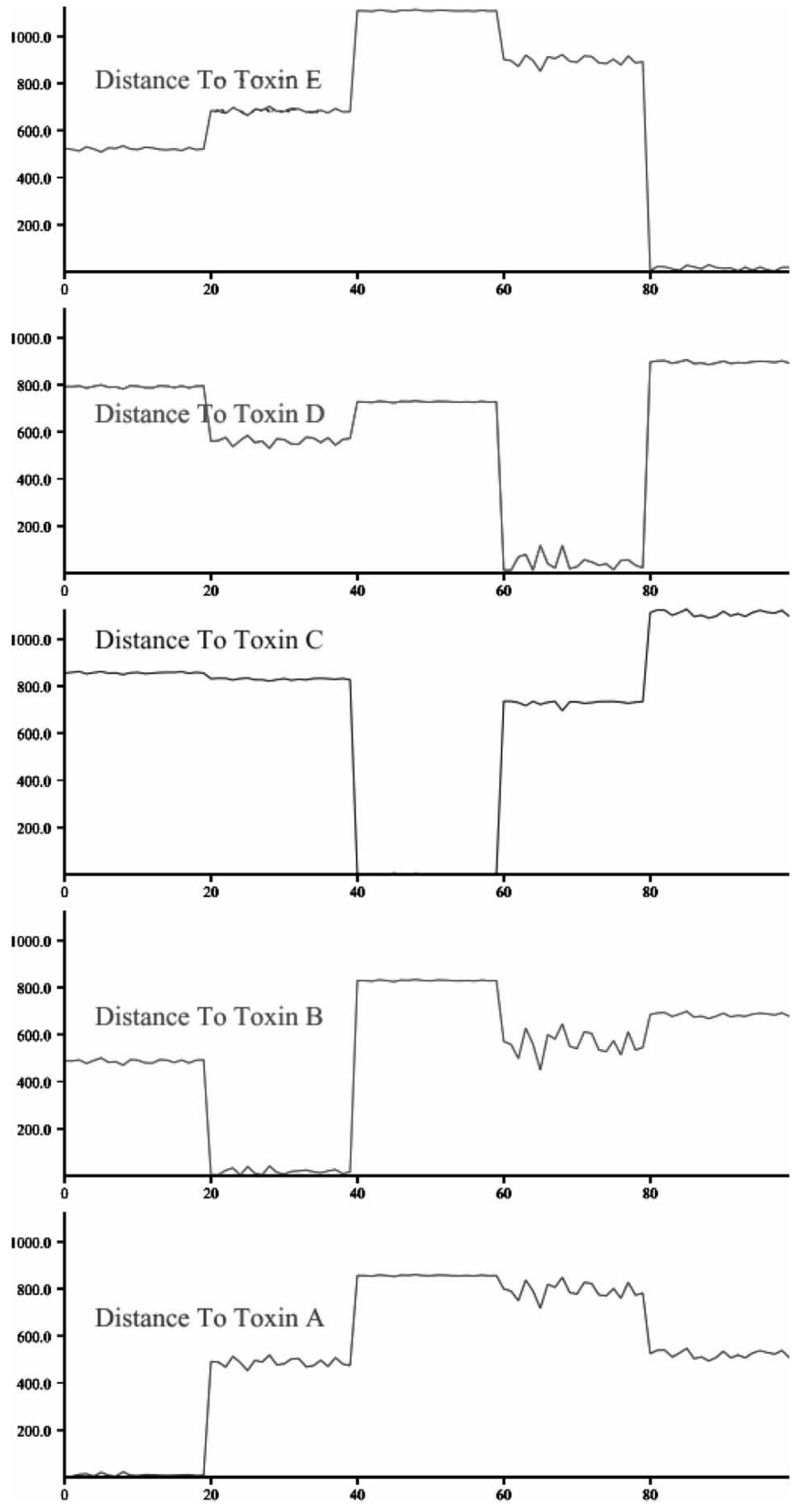

Figure 12. 5D covariance recognizer.

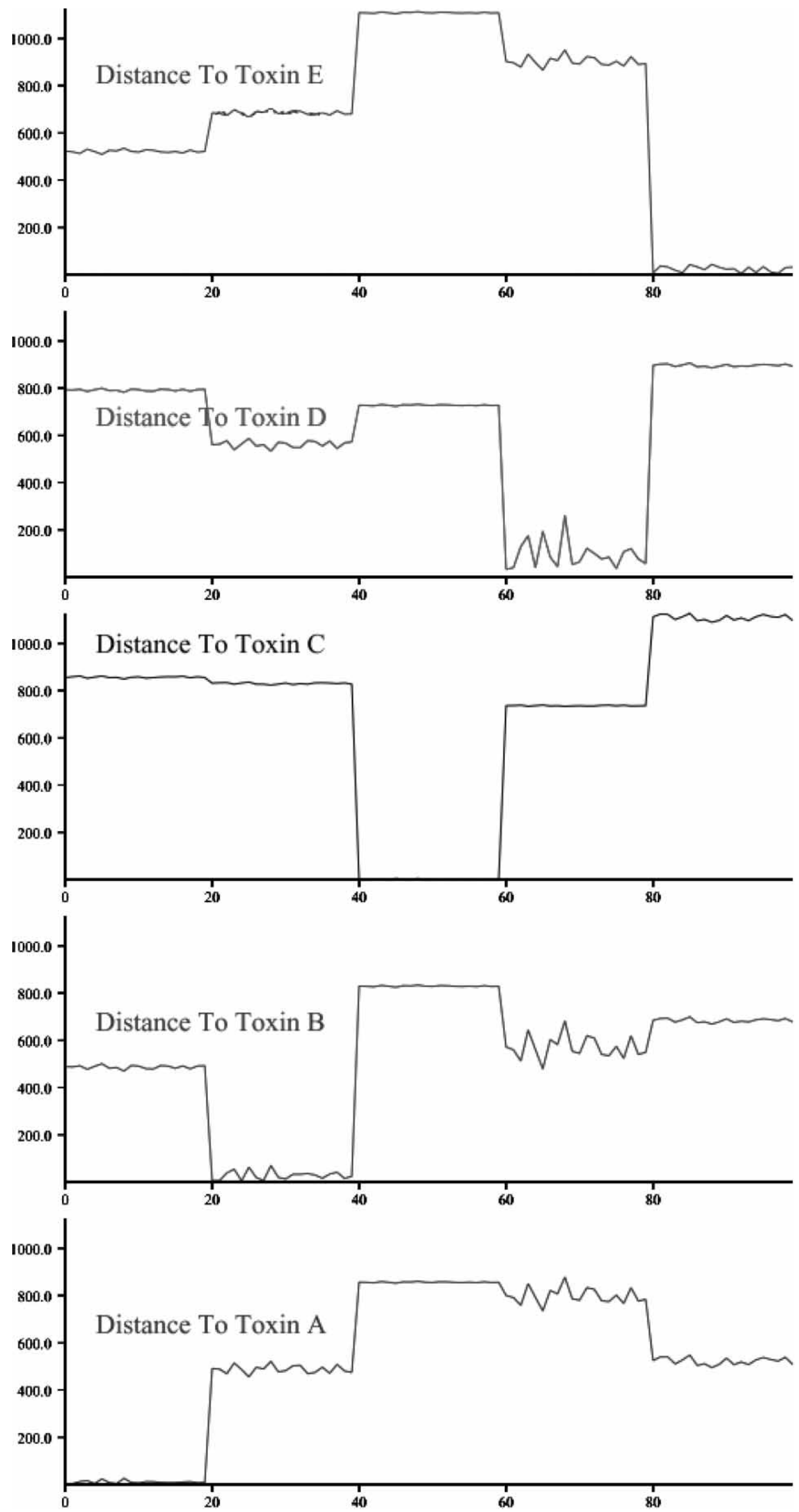

Figure 13. 10D covariance recognizer size 0.02.

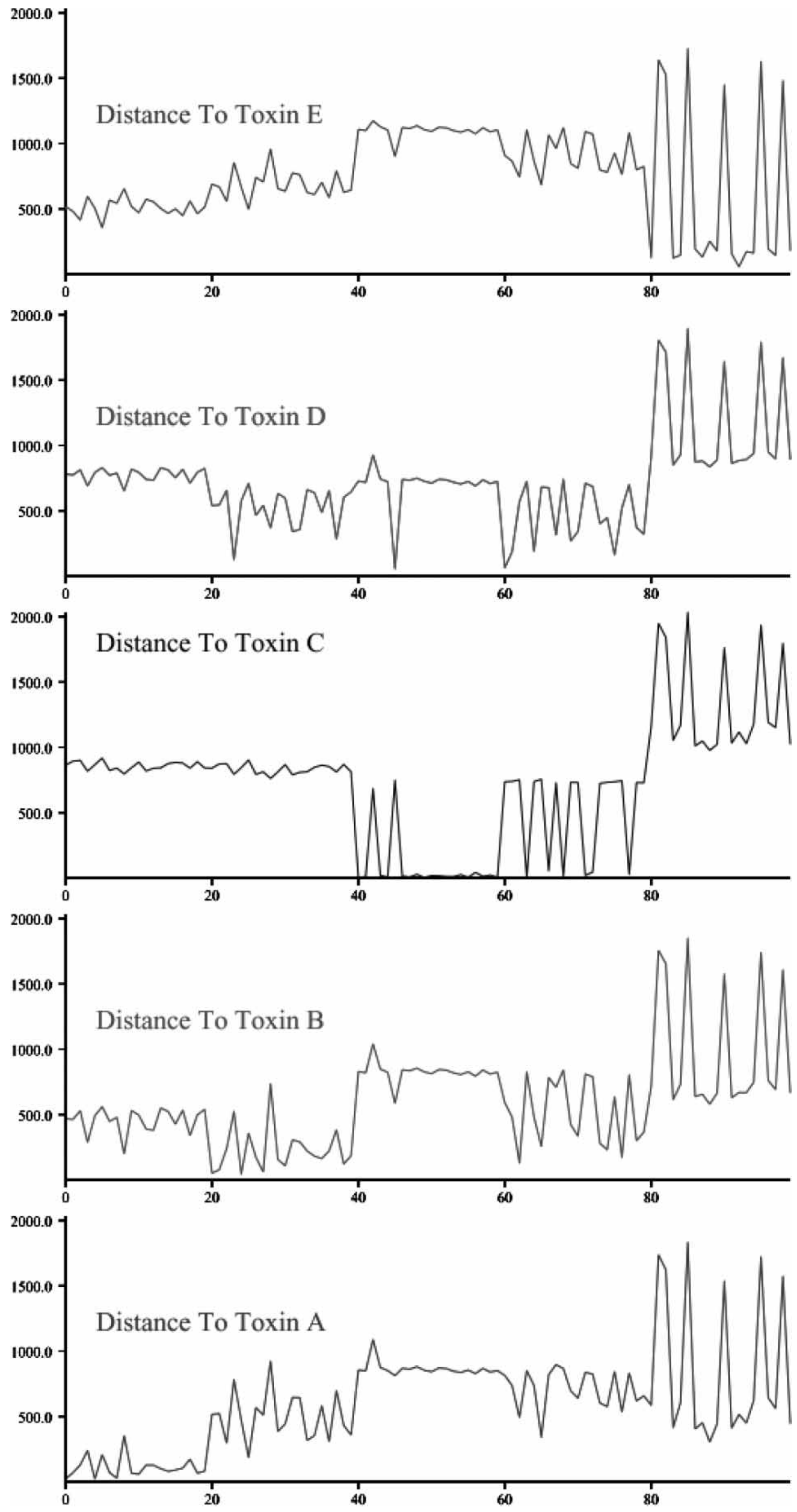

Figure 14. 10D covariance recognizer size 0.2.

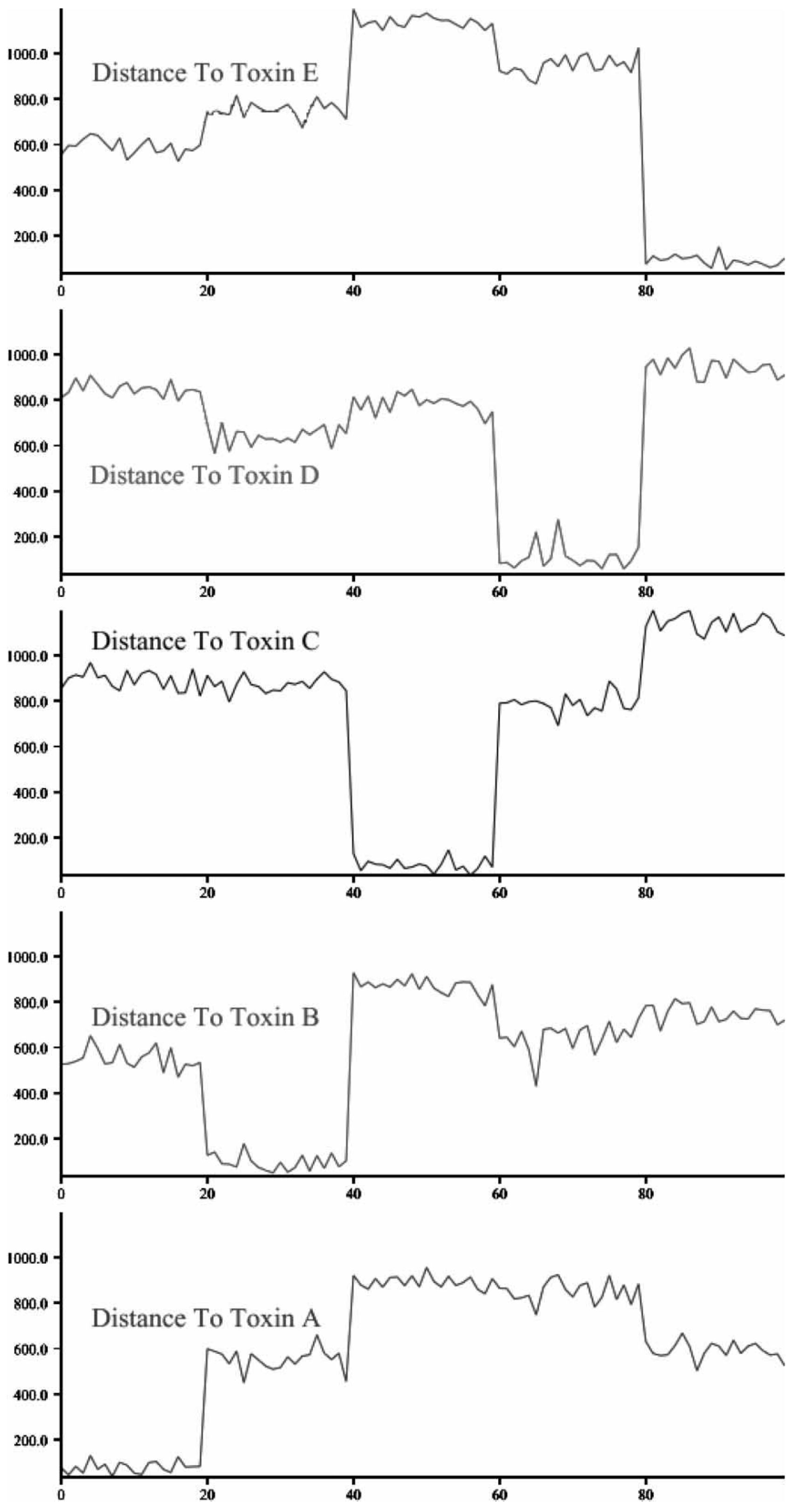

Figure 15. 5D covariance recognizer on noisy data.

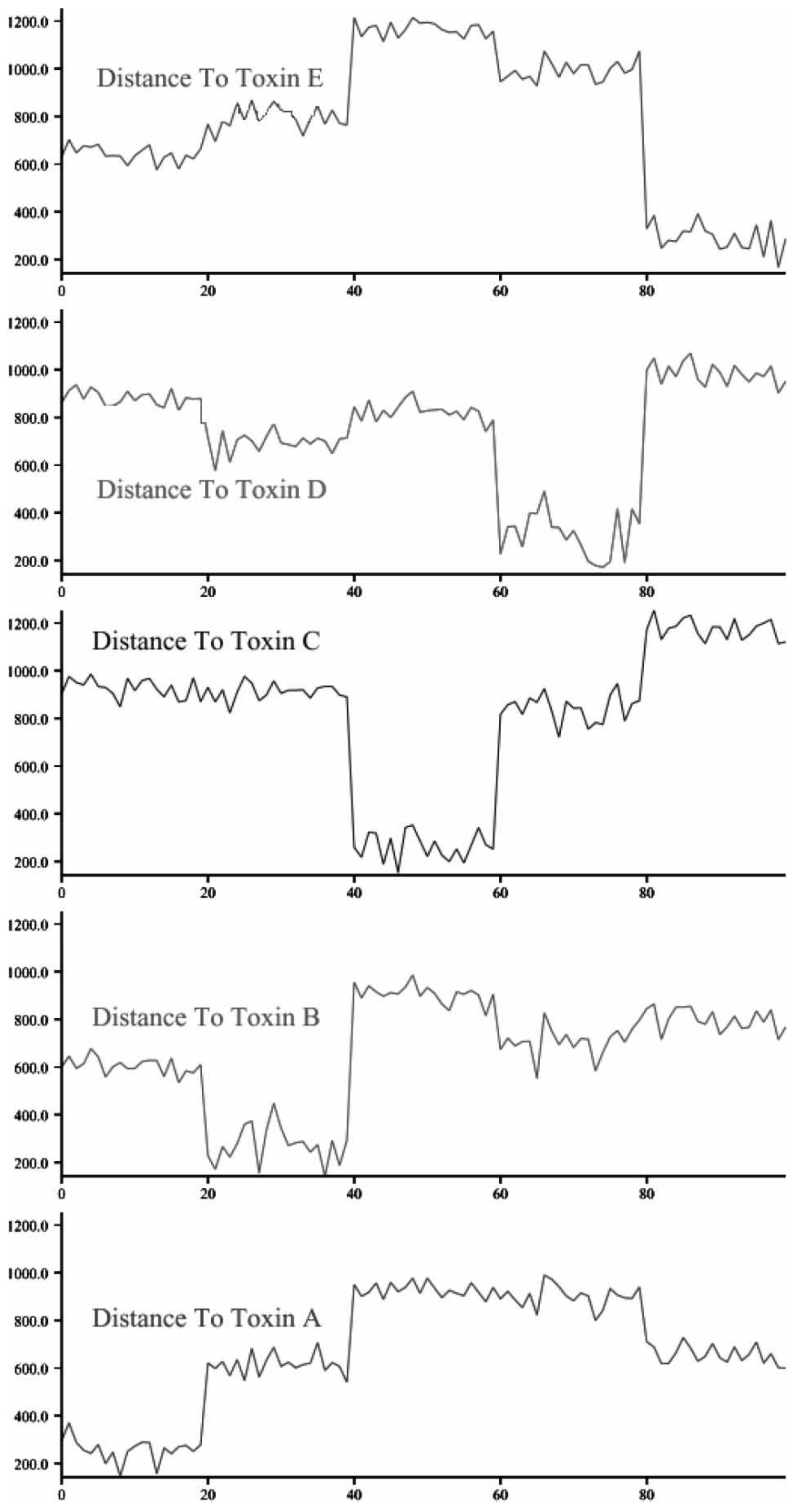

Figure 16. 10D covariance recognizer on noisy data.

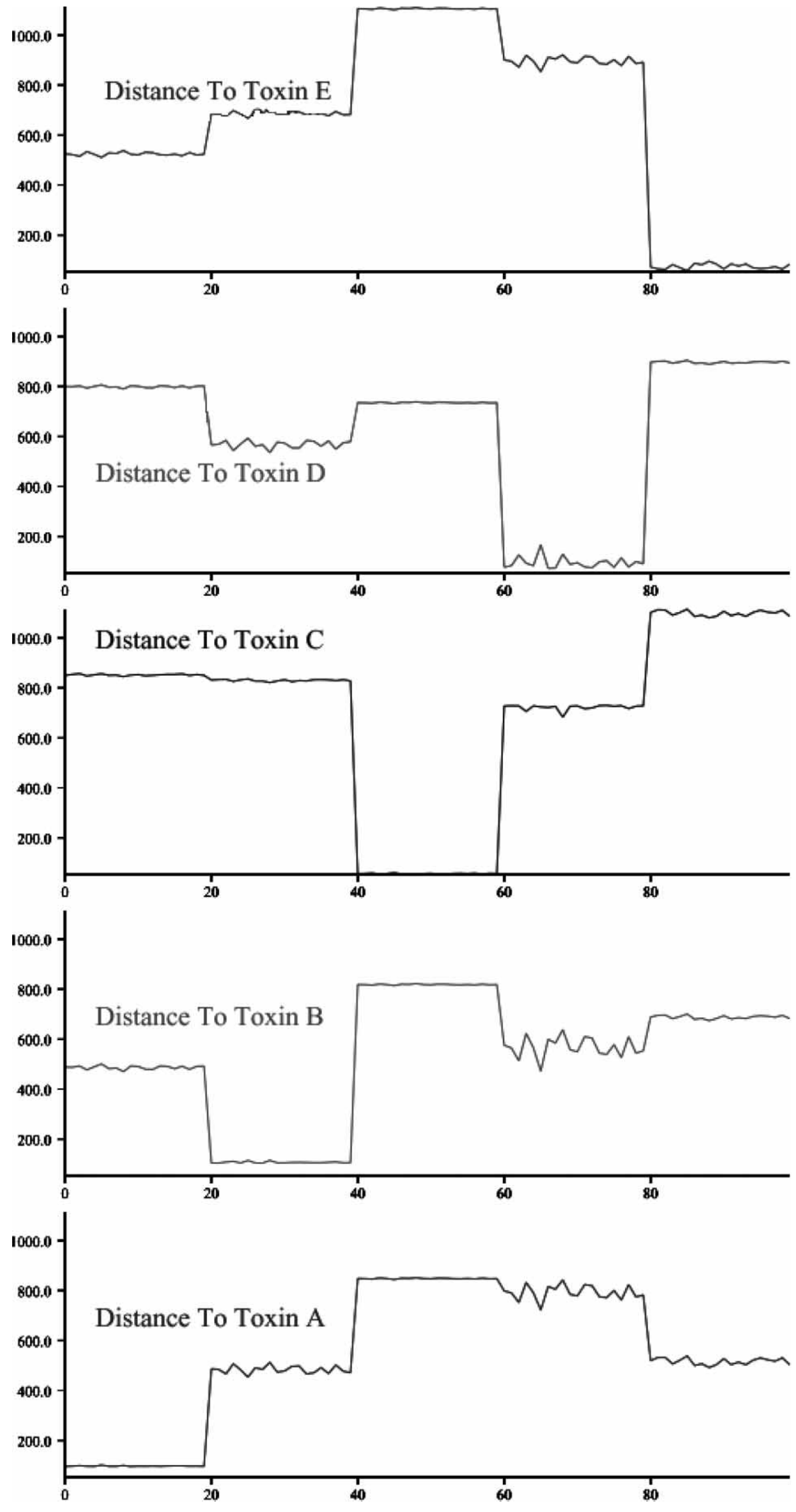

Figure 17. 5D covariance noisy recognizer on clean data.

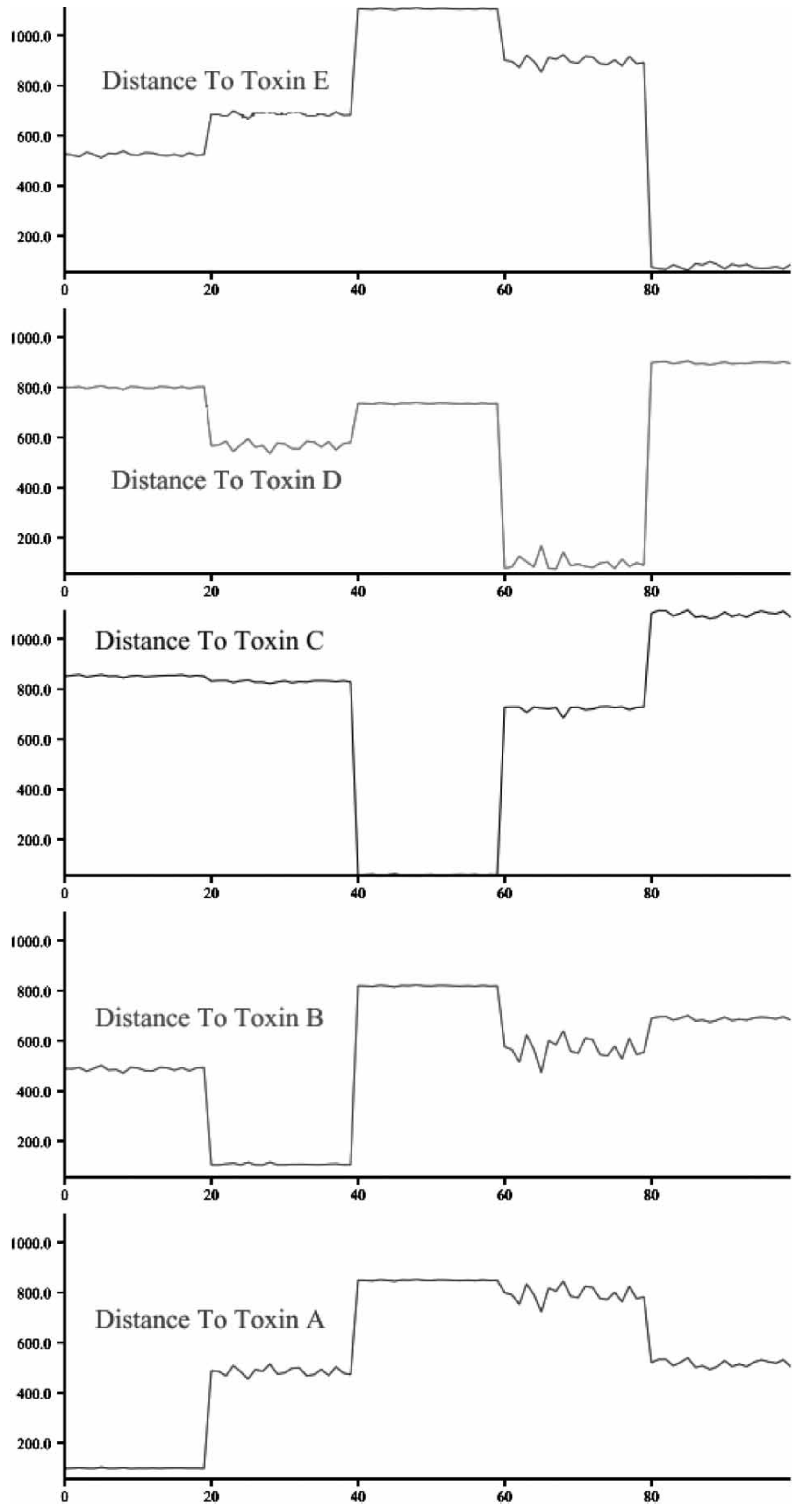

Figure 18. 10D covariance noisy recognizer on clean data.

class distance separation. From the log eigenvalue plot, we see that the five dimensional engine is probably adequate; that is indeed the case as we can see from the left hand plot. The noisy recognizer here is being tested on the data that was used to build it. Next, we test it on the noiseless data from section 3.1. This is shown in figures 17 and 18. In figure 17, we show how the noisy recognizer built using a 5 dimensional projection performs on the clean toxin data. The results for a recognizer built using a 10 dimensional projection are shown in figure 18. Again, there is still good class distance separation.

4.2 The total variation recognizer

The general feature of the action potential as shown in figure 7 exhibits an upside down cone like shape with a narrow top and two steep sides. Therefore the purpose of developing an adaptive non-uniform grid or feature vector generation method is to best accommodate these specific features of the action potential. In fact, for each action potential curve, the grid or feature vector generation program distributes N grid point t_i in such a way that grid points are spaced with higher density in the region where the variation of the action potential are larger. More specifically, the adaptive grid generation program evaluates the change of a combined variation function $B(t)$ and ensuring that the changes between any two neighbouring grid points are kept to be the same while positioning the grid points Wang *et al.* (1999) for similar grid generation algorithm for data surface reconstruction). For convenience, define $\delta_j^V = V_{j+1} - V_j$ and $\Delta_j^V = (V_{j+1} - V_j/t_{j+1} - t_j)$. The combined variation function $B(t)$ used in our algorithm then takes the form

$$C(V,j) = (\lambda|_0 + \lambda_1\delta_{j-1}^V| + \lambda_2|\Delta_j^V - \Delta_{j-1}^V|)$$

$$B(t) = B(t_{j-1}) + \frac{t - t_{j-1}}{t_J - t_{j-1}}C(V,j)$$

for t in the interval $[t_{j-1},t_j]$ and $B(t_1) = 0$. It can be seen that, using this combined variation function, the effects of both the action potential variation (second term in parenthesis) and their rate of variation (third term in parenthesis) are considered together with the time variation in the distribution of the grid points. The coefficients λ_k are the adjustable weights for different components. For a typical action potential curve, the algorithm calculates the combined variation $B(t)$ which is shown in figure 19.

It is clear from figure 19 that the feature vector generation program calculates the combined variation function $B(t)$ given in equation above for a given action potential. We note that $B(t)$ is an increasing function of time with $0 \leq B(t) \leq 1$ and hence one to one. Therefore, if we equally subdivide the interval from [0,1] into $N = 12$ points (shown in circles in figure 19) then we can project these points into the t axis which are given by crosses in figure 19. So these feature vectors (in crosses) are distributed such that the density is higher where combined variation function is steeper.

In order to compare how this total variation feature vector generation compares with our previous recognition engines, we computed the total variation BFVs for all of the 100 action potential curves. In figure 20, we show how the total variation based recognizer built using a 5 dimensional feature vector performs on the clean data while figure 21 shows the results for the toxin data with 40% noise added. Similarly, we show results for clean and noisy data using a 10 dimensional recognizer in figures 22 and 23.

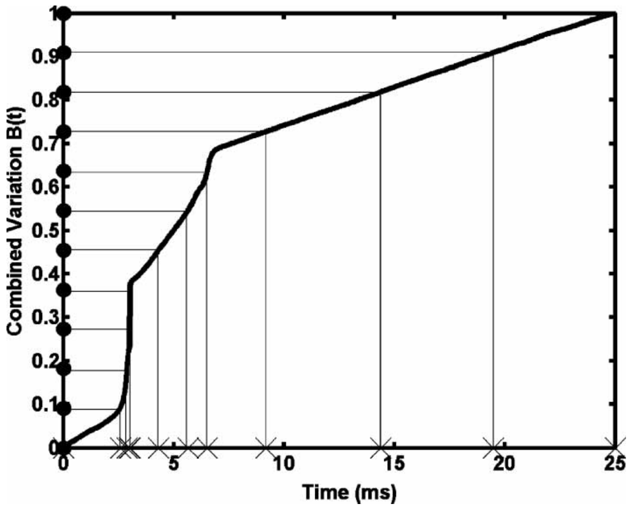

Figure 19. Variation function for one potential.

4.3 The spline based recognizer

The spline approximation to the action potential generated by our algorithm has the form Schumaker (1980)

$$V_{\mathrm{m}}(t) = \sum_{k=1}^{N+L-1} q_k\phi_k(t) \tag{9}$$

where L is the order of the spline function used and $\{\phi_k\}$ are the basis elements for polynomial spline functions of order L defined on the generated adaptive non-uniform grid t_j. The coefficients q_k or the feature vector $q = (q_1,\ldots,q_{N+L-1})$ in equation (9) are determined by minimizing the following functional:

$$J(q) = \gamma_0\sum_{j=1}^{N_1}|V_{\mathrm{m}}(t_j) - W(t_j)|^2 + \gamma_1\int_{t_0}^{t_f}|V_{\mathrm{m}}(t) - W(t)|^2\mathrm{d}t$$

$$+ \gamma_2\int_{t_0}^{t_f}|\dot{V}_{\mathrm{m}}(t) - \dot{W}(t)|^2\mathrm{d}t,$$

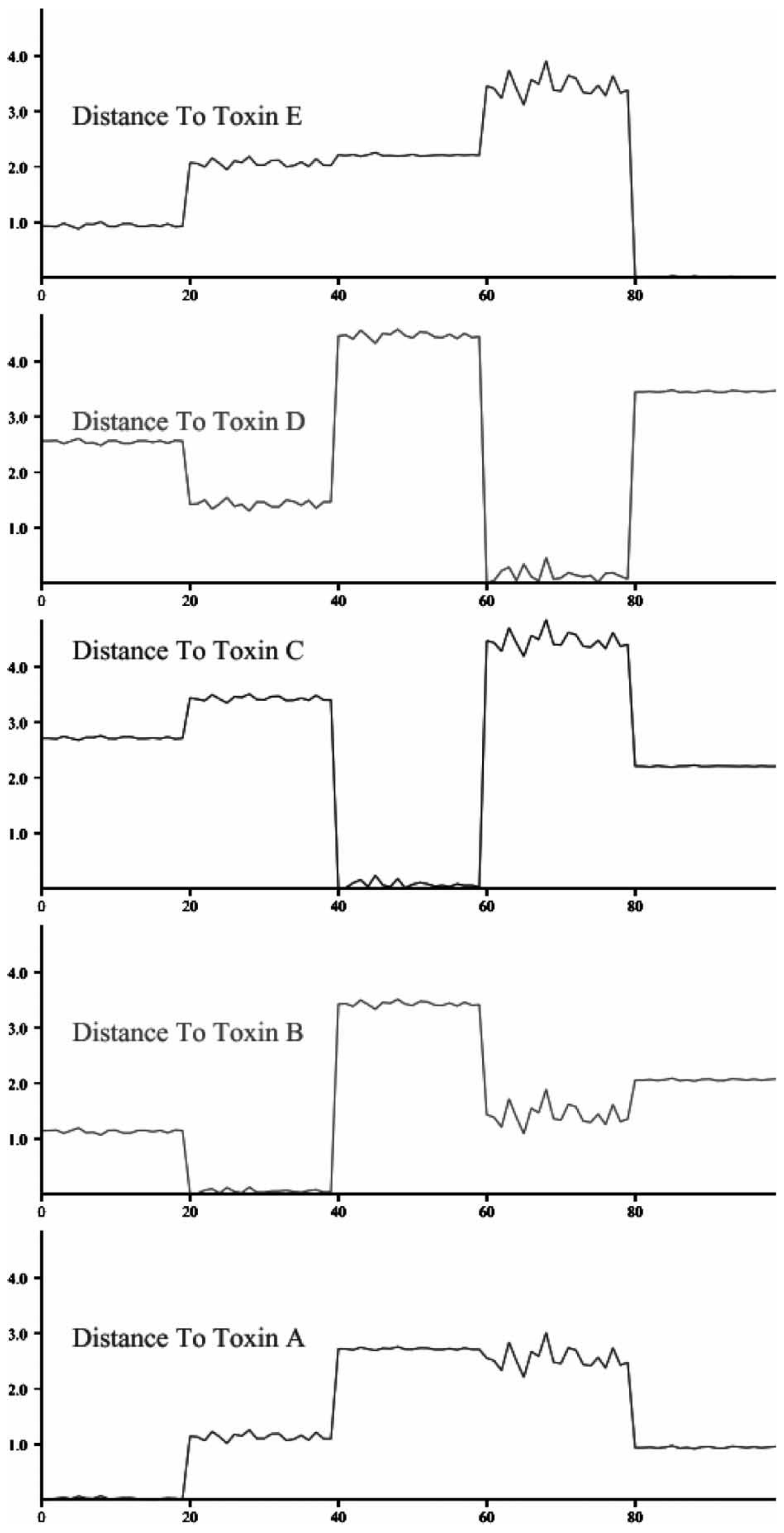

Figure 20. 5D total variation recognizer: on clean data.

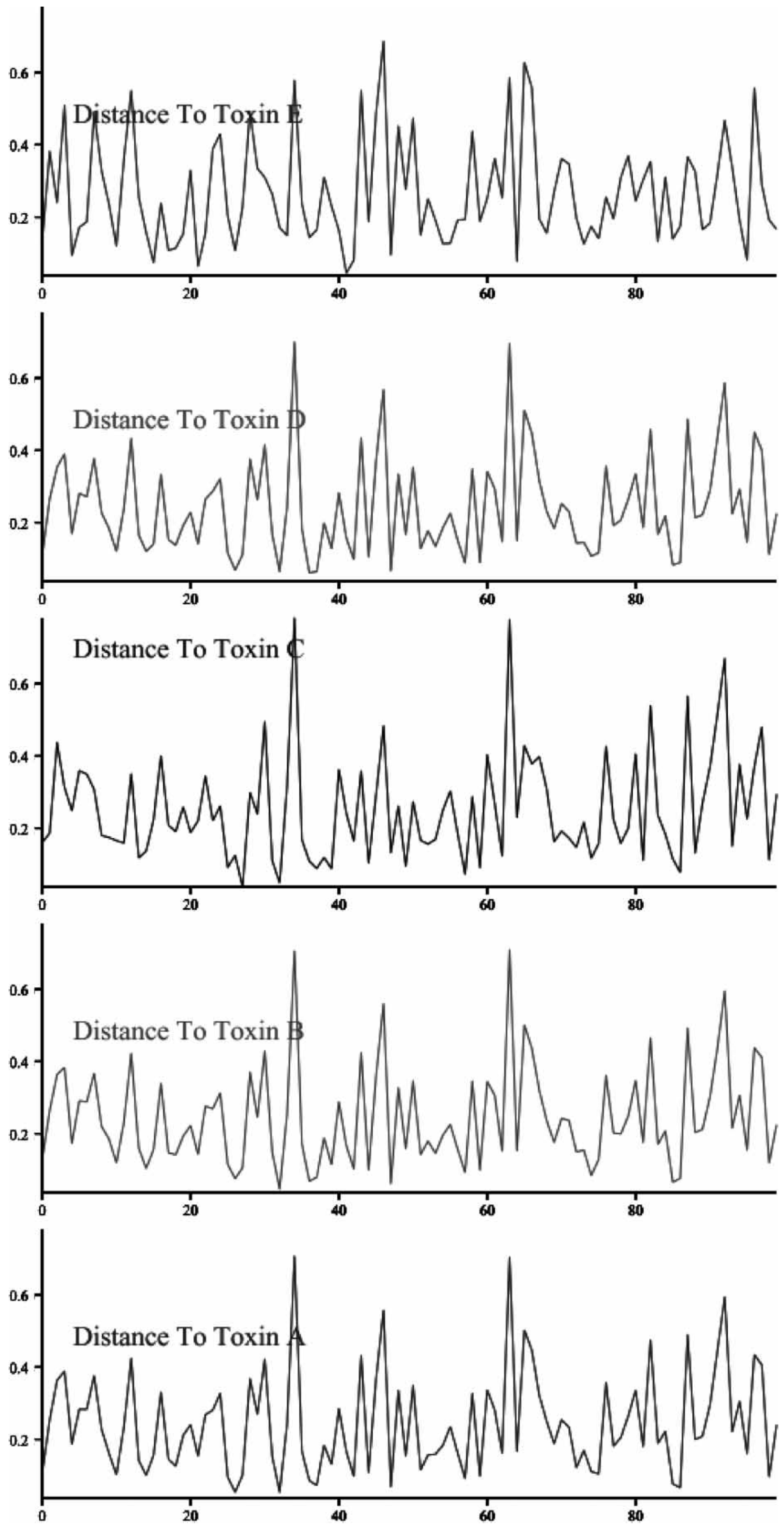

Figure 21. 5D total variation recognizer: on noisy data.

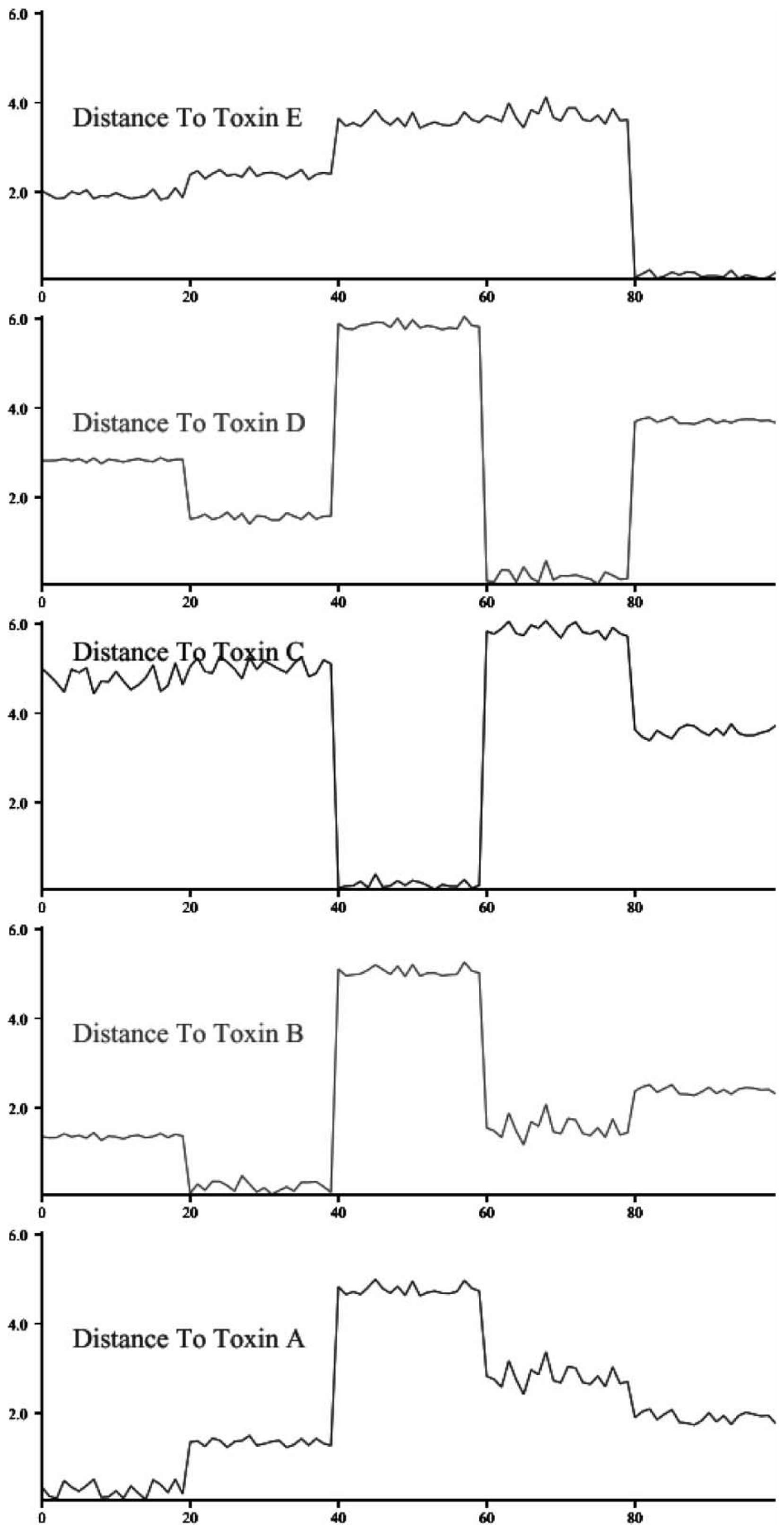

Figure 22. 10D total variation recognizer: on clean data.

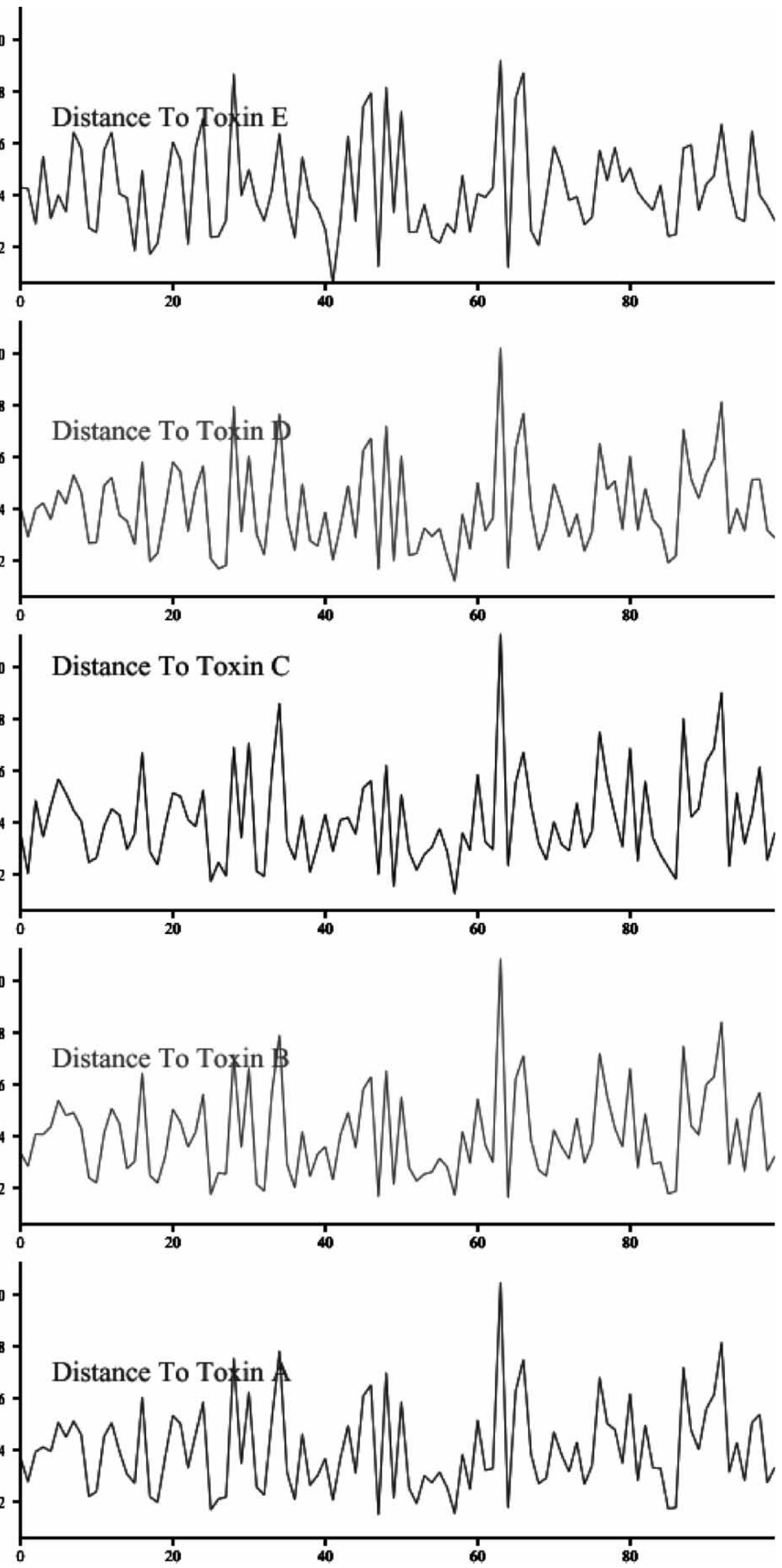

Figure 23. 10D total variation recognizer: on noisy data.

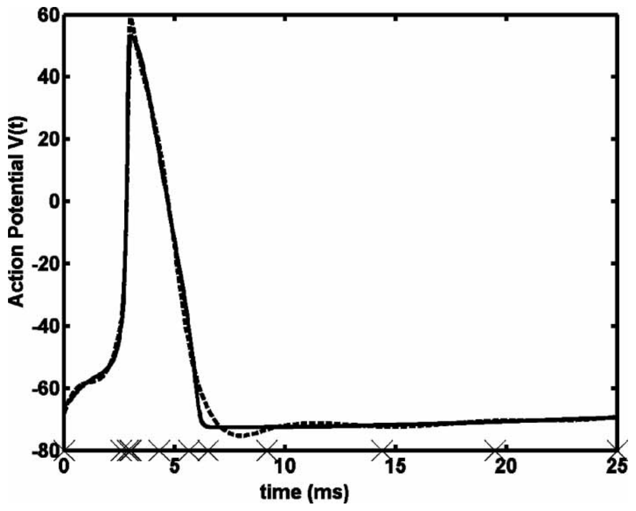

Figure 24. Feature vector construction.

where N_1 is the number of experimental data points for action potential. $W(t)$ is the linear interpolation of the experimental data, $V_m(t_j)$. Note that, in our feature vector calculation algorithm, not only the difference in the values of the action potential but also the difference in the rates of change between approximation and the experimental data (third integral including the differential term) are considered. The inclusion of this term allows us to control the regularity of the approximated action potential and reduce the amount of oscillations often observed in the approximation of rapidly varying data. In fact, by choosing coefficients γ_0,γ_1 and γ_2, we can significantly increase the robustness of the calculated featured vector and yet produce satisfactory fit to the experimental action potential. To demonstrate our ideas, we construct a feature vector for the simple Hodgkin–Huxley model. The simulated action potential and the associated spline approximation are shown in figure 24.

Here, the number of non-uniform grid points generated is $N = 12$ and the order of the spline used is cubic ($L = 3$). Therefore, we need only 14 basis elements (ϕ_1,ϕ_2,...,ϕ_{14}). Therefore, the feature vector is a vector in R^{14} which approximates the feature of the simulated action potential accurately as shown in figure 24. In figure 24, the dotted line represents the approximated $V(t)$ using cubic splines and the solid line represents the actual action potential. The crosses on the t axis represents the non-uniform grid points used for the cubic spline approximation from the combined variation approach explained in the previous section. For this choice of the splines used, we get the following coefficients for the approximation depicted in figure 24,

$$\begin{aligned} q = (&-2.3388, -0.1799, -0.4169, \\ &-0.0874, 0.1066, 0.1122, 0.1008, -0.2977, \\ &-0.8060, -0.9457, -1.3772, -1.4718, \\ &-1.5603, -1.2548). \end{aligned}$$

In order to compare how this total variation feature vector generation compares with our previous recognition engines, we computed the cubic spline BFVs (mainly ($q = q_1, \ldots, q_{N+L-1}$)) for all of the 100 action potential curves.

The performance of the spline based recognizer using a 10 dimensional recognizer is shown in figure 25 for clean data and figure 26 for the noisy data.

4.4 The biological feature vector based recognizer

One can easily observe that for a typical response, as in figure 1, the potential exhibits a combination of cap-like shapes. We can use the following points on this generic action potential to construct a low dimensional feature vector:

$$\xi = \begin{bmatrix} (t_0, V_0) & \text{start point} \\ (t_1, V_1) & \text{maximum poing} \\ (t_2, V_2) & \text{return to reference voltage} \\ (t_3, V_3) & \text{minimum point} \\ (g, t_4, V_4) & \text{sigmoid model of tail} \end{bmatrix},$$

where the model of the tail of the action potential is of the form

$$V_m(t) = V_3 + (V_4 - V_3) \tanh (g(t - t_3)),$$

Note that

$$V'_m(t_3) = (V_4 - V_3)g.$$

We approximate $V'_m(t_3)$ by a standard finite difference. We pick a data point (t_5,V_5) that occurs after the minimum—typically, we use the voltage value at the time t_5 that is 5 time steps downstream from the minimum and approximate the derivative at t_3 by

$$V'_m(t_3) \approx \frac{V_5 - V_3}{t_5 - t_3}$$

The value of g is then determined to be

$$g = \frac{V_5 - V_3}{(V_4 - V_3)(t_5 - t_3)}$$

which reflects the asymptotic nature of the hyperpolarization phase of the potential. Note that ξ is in $\Re^{11}$. We have deliberately chosen a very simple feature vector extraction to focus on how well such a simplistic scheme will perform in comparison to the more sophisticated covariance, total variation and spline methods. We shall see that even this rudimentary feature vector extraction is quite capable of generating a functional recognizer.

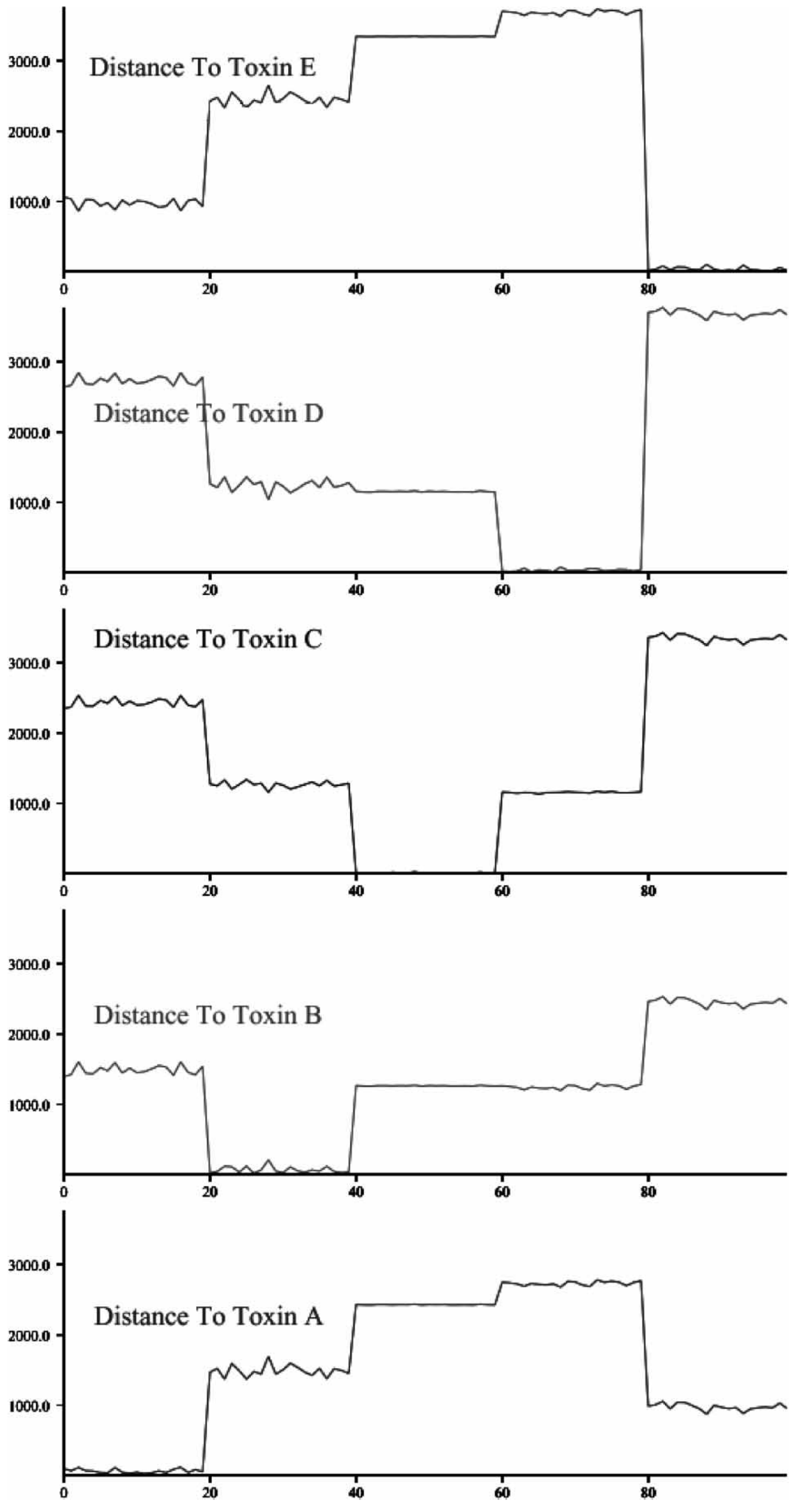

Figure 25. 10D spline recognizer: on clean data.

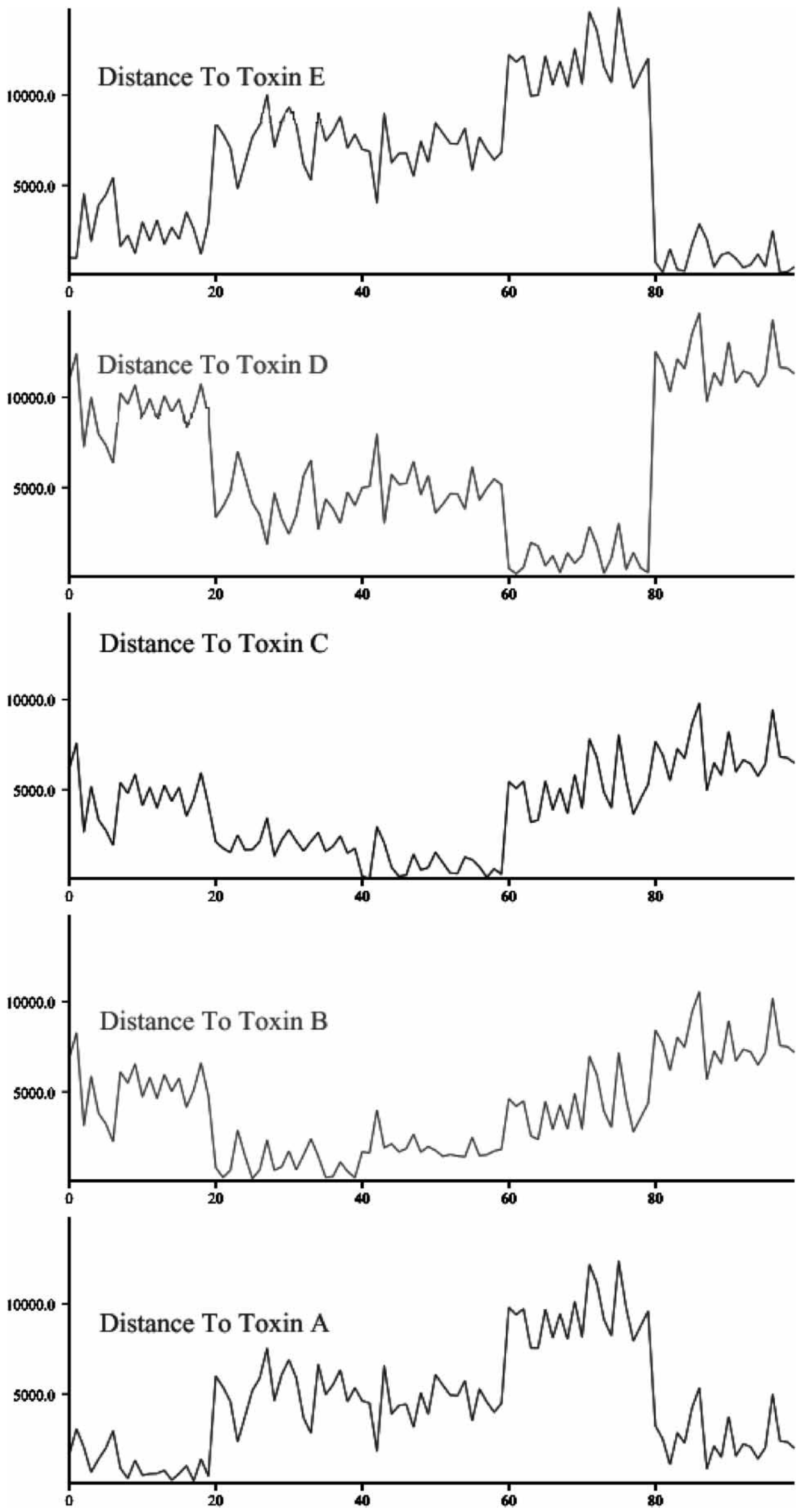

Figure 26. 10D spline recognizer: on noisy data.

Table 3. BFV: 10D recognizer data.

Distance				
A	B	C	D	E
0.32988	3.02567	5	5	5
0.09334	2.6891	5	5	5
0.10856	2.6964	5	5	5
2.34489	0.50764	5	4.50301	5
2.25101	0.62203	5	4.61276	5
2.88308	0.27830	5	4.05954	5
5	5	0.15268	5	5
5	5	0.29684	5	5
5	5	0.20283	5	5
5	4.03615	5	0.14421	5
5	3.94194	5	0.21688	5
5	4.12045	5	0.37244	5
5	5	5	5	0.08755
5	5	5	5	0.71222
5	5	5	5	0.56732

In all of the recognizers we have constructed, the classification of the toxin is determined by finding the toxin class, which is minimum distance from the sample. The distances generated by the biological feature vector are still very capable of discerning toxin class; however, the raw distances are very different. For example, in table 3, we show the computed distances for some of the samples. The first three rows are distances from Toxin A samples, the next three are Toxin B sample distances and so forth. The code that generated these distances set the distance to 5 if the distance was more than that for ease of display in the graphs shown in figures 27 and 28. Note that the classification of each toxin is still quite easy as the distances to each toxin class are cleanly separated.

Figures 27 and 28 show how the biologically based recognizer built using a 5 dimensional feature vector performs on both the clean and noisy data. Similar results are shown for the performance of the 10 dimensional recognizer in figures 29 and 30. The manner in which the biological feature is constructed shows that it is not a robust technique in the presence of noise. Searching the data for minimum, crossing and maximum points is problematic when there are many local extrema. However, the raw signal could easily be filtered to remove such ripple.

5. Toxins that reshape α–β parameters

From our earlier discussions, we know toxins of this type that only cause a change in the nominal sodium and potassium maximum conductance are effectively second messenger toxins. However, the 36 additional α–β parameters that we have listed all can be altered by toxins to profoundly affect the shape of the action potential curve. In this section, we will focus on parameter changes that effect a small subset of the full range of α–β possibilities.

This simulation will generate five toxins whose signatures are distinct. Using C++ as our code base, we have designed a TOXIN class whose constructor generates a family of distinct signatures using a given signature as a base. Here, we will generate 20 sample toxin signatures clustered around the given signature using a neighbourhood size for each of the five toxin families. First, we generate five toxin families. The types of perturbations each toxin family uses are listed in table 4. In this table, we only list the parameters that are altered. Note that Toxin A perturbs the q parameters of the $\alpha - \beta$ for the sodium activation m only; Toxin B does the same for the q parameters of the sodium h inactivation; Toxin C alters the q parameters of the potassium activation n; Toxin D changes the p (Grossberg 2003) value of the α–β functions for the sodium inactivation h; and Toxin E, does the same for the potassium activation n. This is just a sample of what could be studied. These particular toxin signatures were chosen because the differences in the generated action potentials for various toxins from family A, B, C, D or E will be subtle. For example, in Toxin A, a perturbation of the given type generates the α–β curves for m_{NA} as shown in figure 31. Note all we are doing is changing the voltage values of 35.0 slightly. This small change introduces a significant ripple in the α curve.

Note, we are only perturbing two parameters at a time in a given toxin family. In all of these toxins, we will be leaving the maximum ion conductances for we use these toxins to generate 100 simulation runs using the 20 members of each toxin family. We believe that the data from this parametric study can be used to divide up action potentials into disjoint classes each of which is generated by a given toxin.

Each of these simulation runs use the synaptic current injected as in figure 7 which injects a modest amount of current over approximately 2 s using the protocol described earlier. The current is injected at four separate times to give the gradual rise seen in the picture. For additional details, see Peterson (2002b). In figure 32, we see all 100 generated action potentials. You can clearly see that the different toxin families create distinct action potentials.

5.1 *The recognition engines*

We will only build the covariance and the biological feature vector recognition engine for this data. The previous sections have shown that the simple biological feature vector approach is quite good at capturing information content from the action potential pulse. In figures 33 and 34, we look at the recognizer engines built according to the discussion in section 4 using a twenty dimensional projection ($Q = 20$). The distance of each toxin to the toxin class feature vector, vectors are

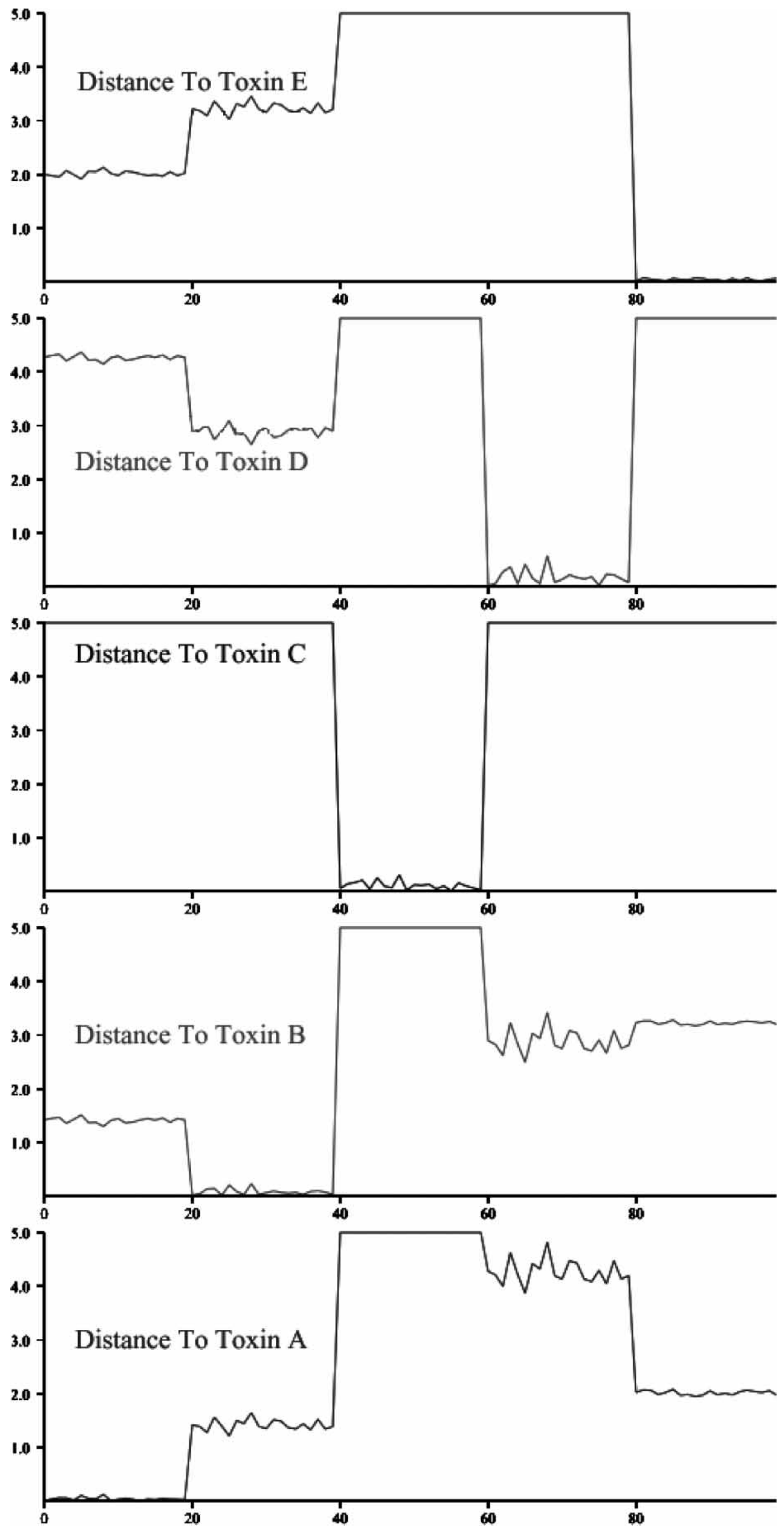

Figure 27. 5D biological feature vector recognizer: on clean data.

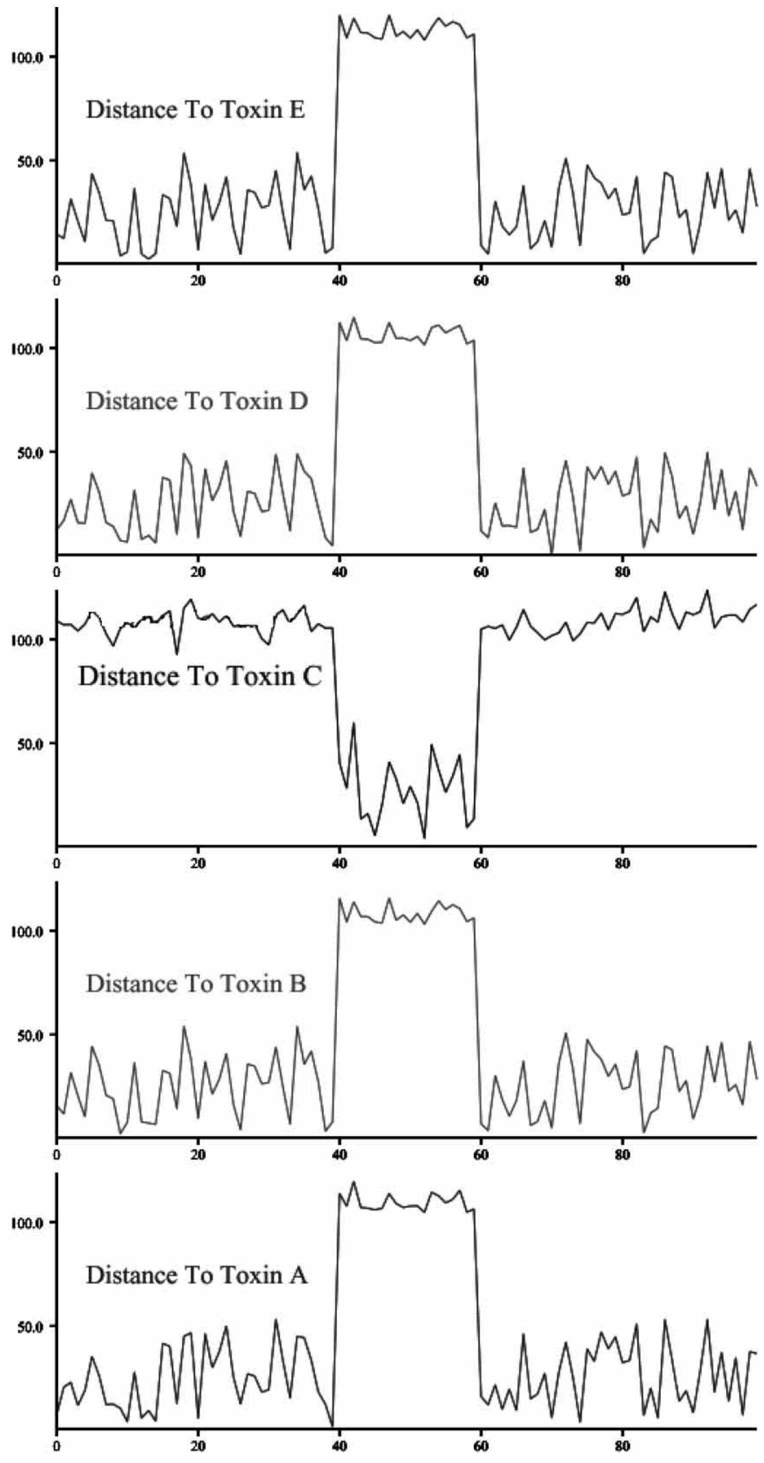

Figure 28. 5D biological feature vector recognizer: on noisy data.

plotted in five separate curves presented in two separate figures. The samples are organized as 20 from each toxin class starting with Toxin A. Hence, for the Toxin A samples, if our recognizer is working well, we expect to see the first 20 distances are very low and the other 80 distances are very high. The toxins corresponding to class A and C are actually fairly close, although their distances are cleanly separated. In order to have clean graphical output, the distances we plot in these graphs are set to 200 is the actual distance exceeds 200. The Toxin A and C distances are typically in the 20–80 range, while the other distances are much higher; on the order to 800 when we calculate the out of class distance. Hence, to show the distance separation more clearly, we

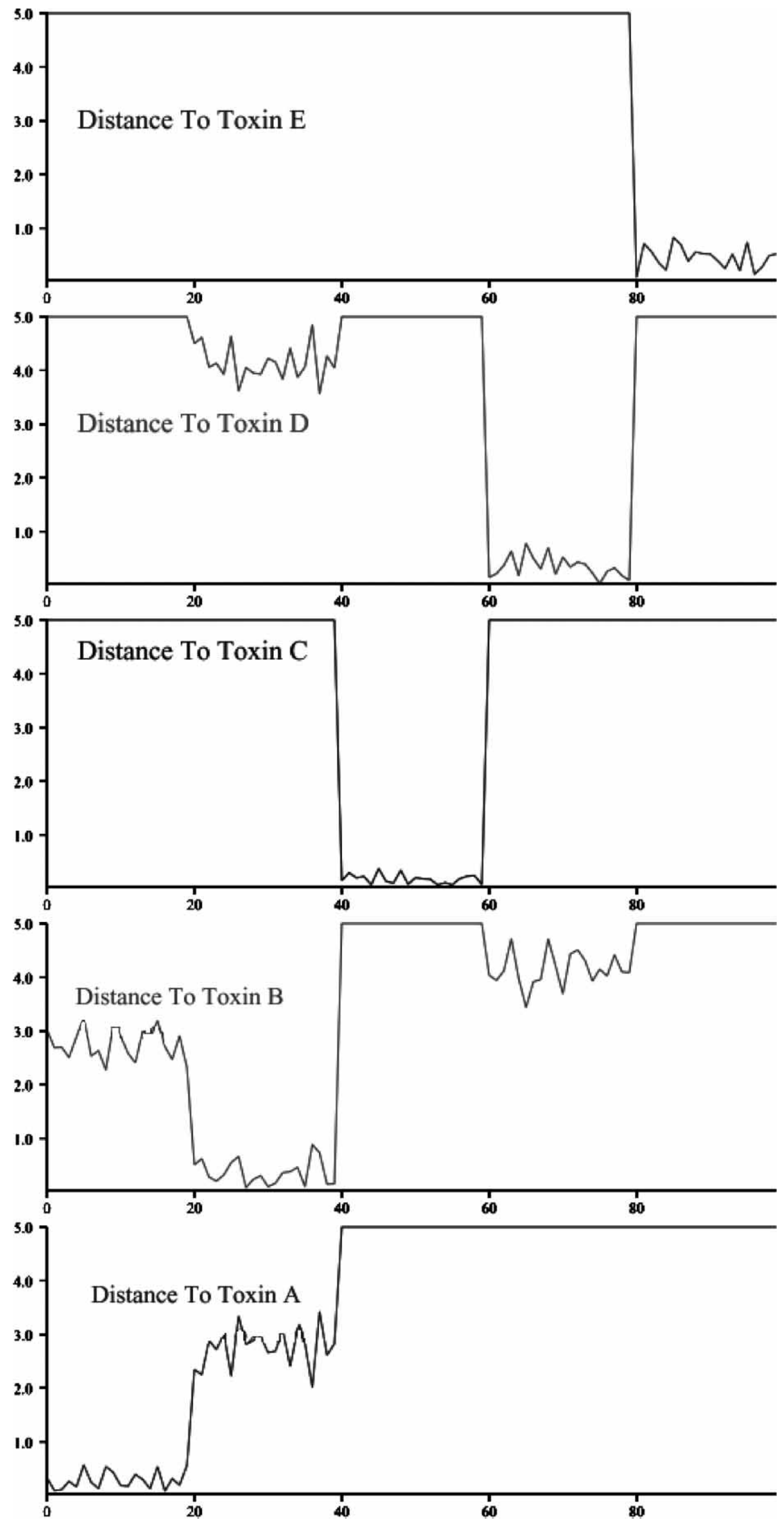

Figure 29. 10D biological feature vector recognizer: on clean data.

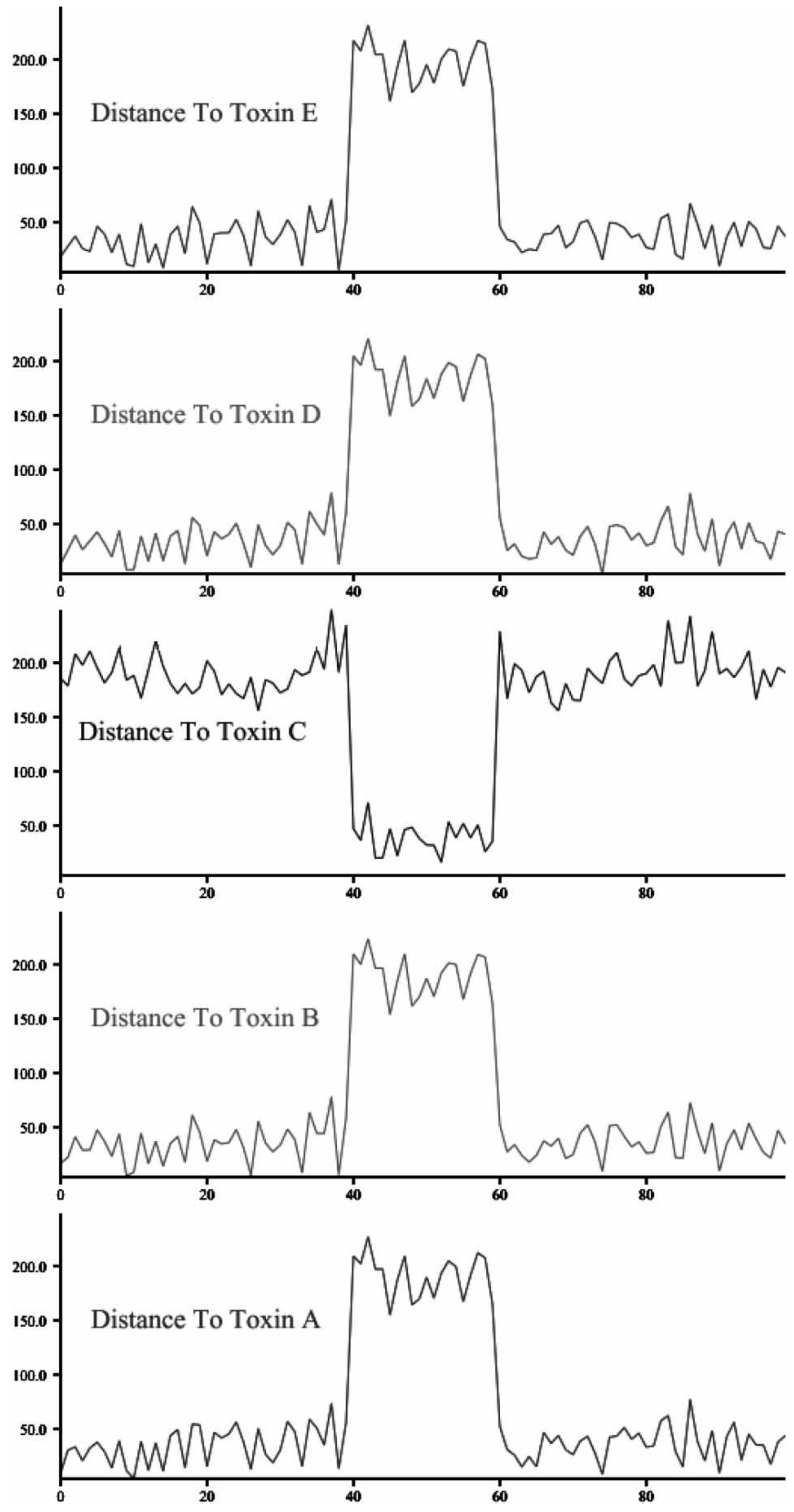

Figure 30. 10D biological feature vector recognizer: on noisy data.

Table 4. Toxin $\alpha-\beta$ signatures.

Toxin A	$\delta q_{m}^{\alpha} = \{0.2, -0.1\}$
	$\delta q_{m}^{\beta} = \{-0.1, 0.1\}$
Toxin B	$\delta q_{h}^{\alpha} = \{-0.1, 0.1\}$
	$\delta q_{h}^{\beta} = \{0.1, -0.2\}$
Toxin C	$\delta q_{n}^{\alpha} = \{-0.2, 0.1\}$
	$\delta q_{n}^{\beta} = \{-0.1, 0.1\}$
Toxin D	$\delta p_{h}^{\alpha} = \{0.0, 0.0, 3.0, 0.0\}$
	$\delta p_{h}^{\beta} = \{0.0, 0.3, 0.0, 0.0\}$
Toxin E	$\delta p_{n}^{\alpha} = \{0.0, 0.3, 0.0, 0.0\}$
	$\delta p_{n}^{\beta} = \{0.0, 0.3, 0.0, 0.0\}$

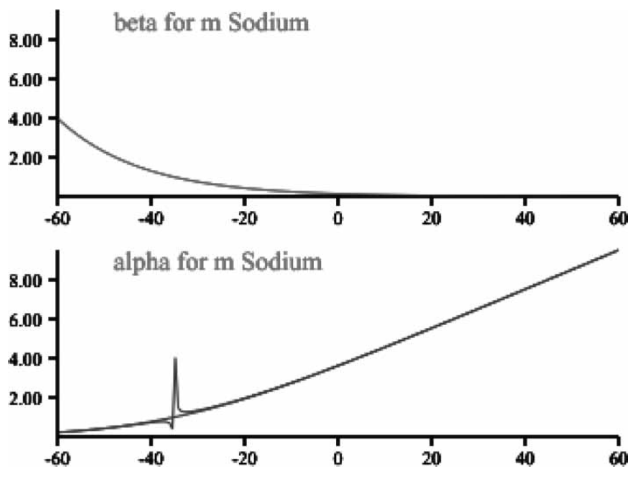

Figure 31. m perturbed Alpha–Beta curves.

graph the Toxin A and C distances figure 33 and the Toxin B, D and E distances figure 34. In general, these toxin classes are very nicely separated from each other as you can see from the distance plots and we get stable recognition.

Using the same biological feature as before, we see this rudimentary feature vector extraction is quite capable of generating a functional recognizer. The classification of the toxin is still determined by finding the toxin class, which is minimum distance from the sample. The distances generated by the biological feature vector in this case are still very capable of discerning toxin class; however, the raw distances are now much smaller. For purposes of exposition, we set all distances to 10 if the actual distance to a toxin class exceeds 10; this will show the class recognition more clearly. Figures 35 and 36 show how the biologically based recognizer built using a 5 and 11 dimensional feature vector performs on the data.

6. Conclusions

In this work, we have shown that a low dimensional feature vector based on biologically relevant information extracted from the action potential of an excitable nerve cell is capable of subserving biological information processing. We can successfully use such a BFV to extract

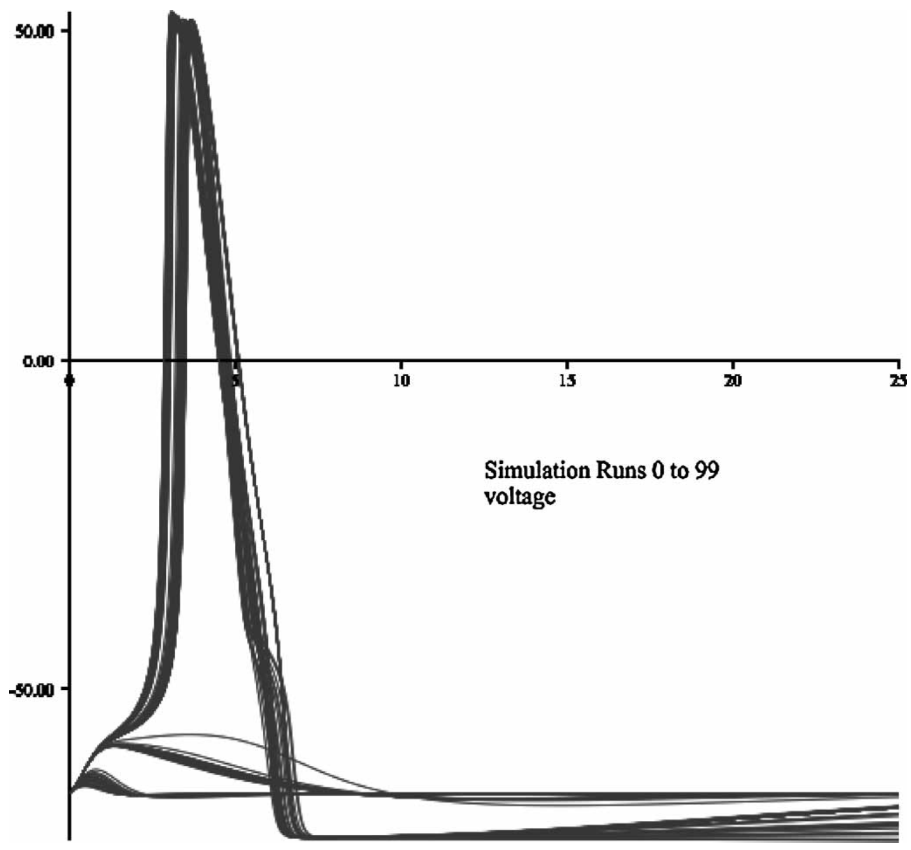

Figure 32. Generated voltage traces for the $\alpha-\beta$ Toxin families.

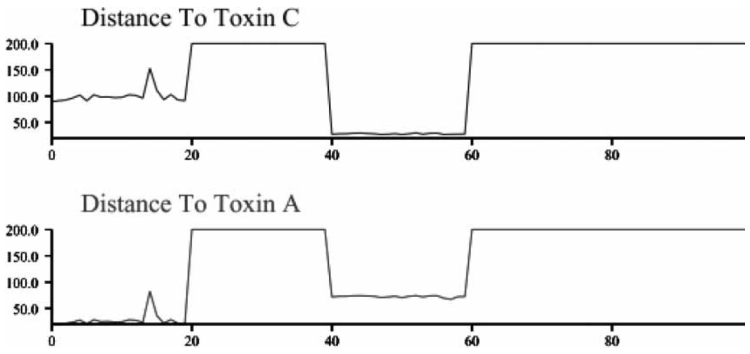

Figure 33. 20D covariance recognizer: Toxin A and C distances.

information about how a given toxin influences the shape of the output pulse of an excitable neuron. Hence, with suitable filtering of the raw measured voltage traces via an embedded sensor, the BFV can serve as a very low computational cost implementation of a toxin recognition strategy.

This work also implies the BFV can extract such information for other modulatory inputs such as NTs. Our studies therefore suggest we can design an artificial neuron whose output is the BFV. Modulatory inputs into the abstract neuron can be modelled as alterations to the components of the BFV. It is then straightforward to develop an algebra of BFV interactions so that we can model cortical interactions modulated by limbic system input. We believe artificial neurons whose outputs are such low dimensional feature vectors are suitable for the modelling of these complicated neural modules. Future directions of this research therefore include cognitive modelling efforts using abstract neuron models in a distributed processing environment.

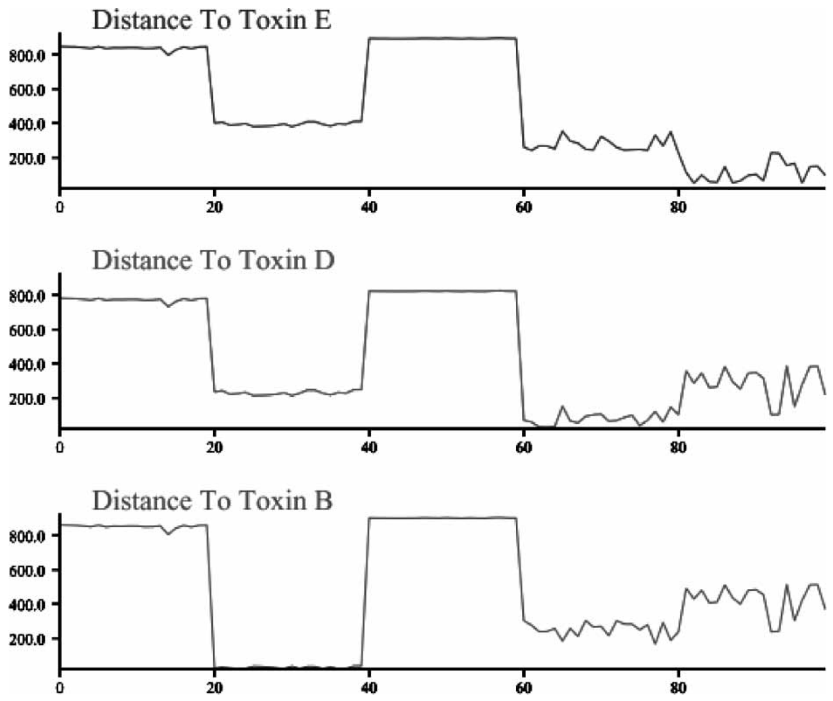

Figure 34. 20D covariance recognizer: Toxin B, D and E distances.

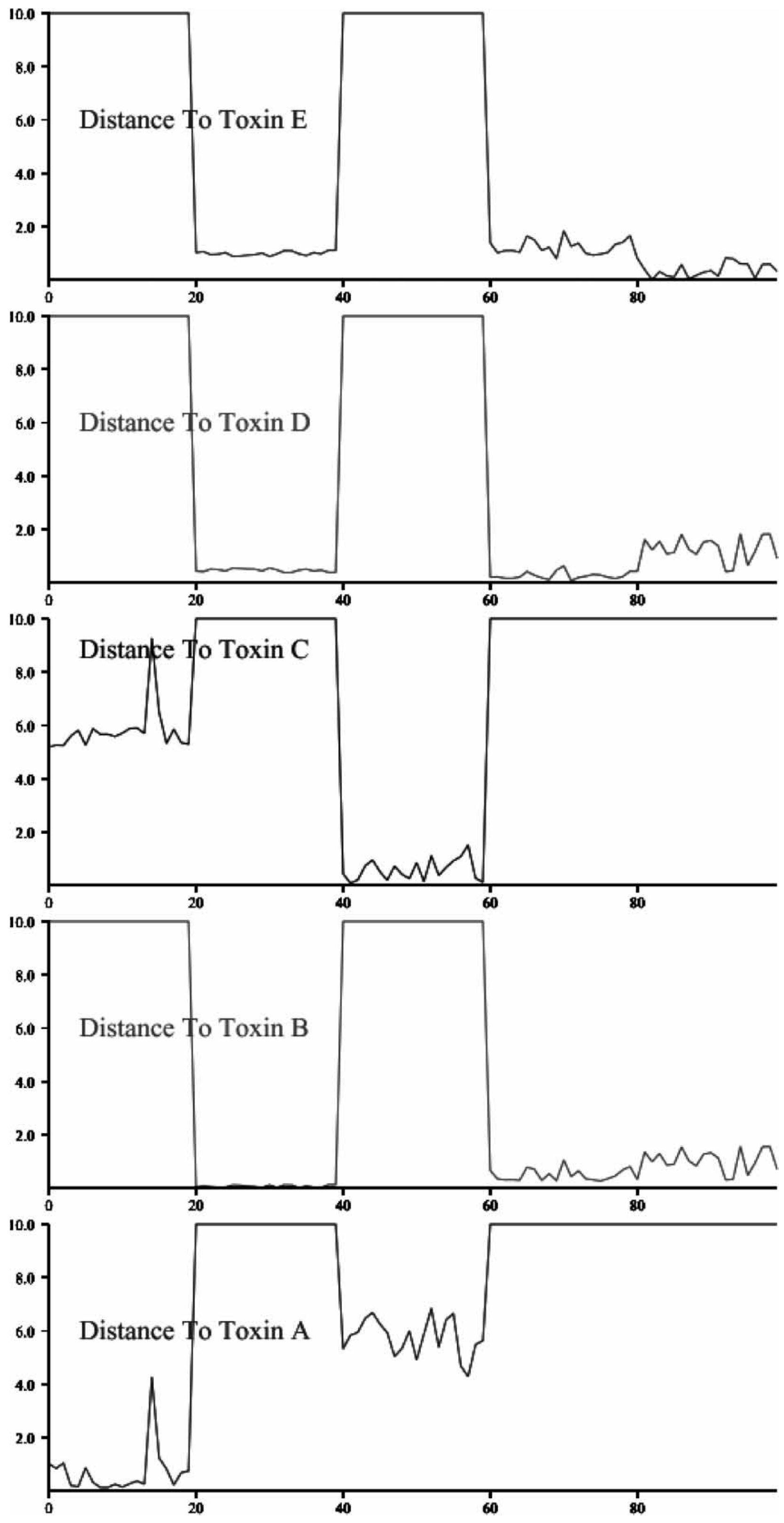

Figure 35. BFV recognizer: 5D.

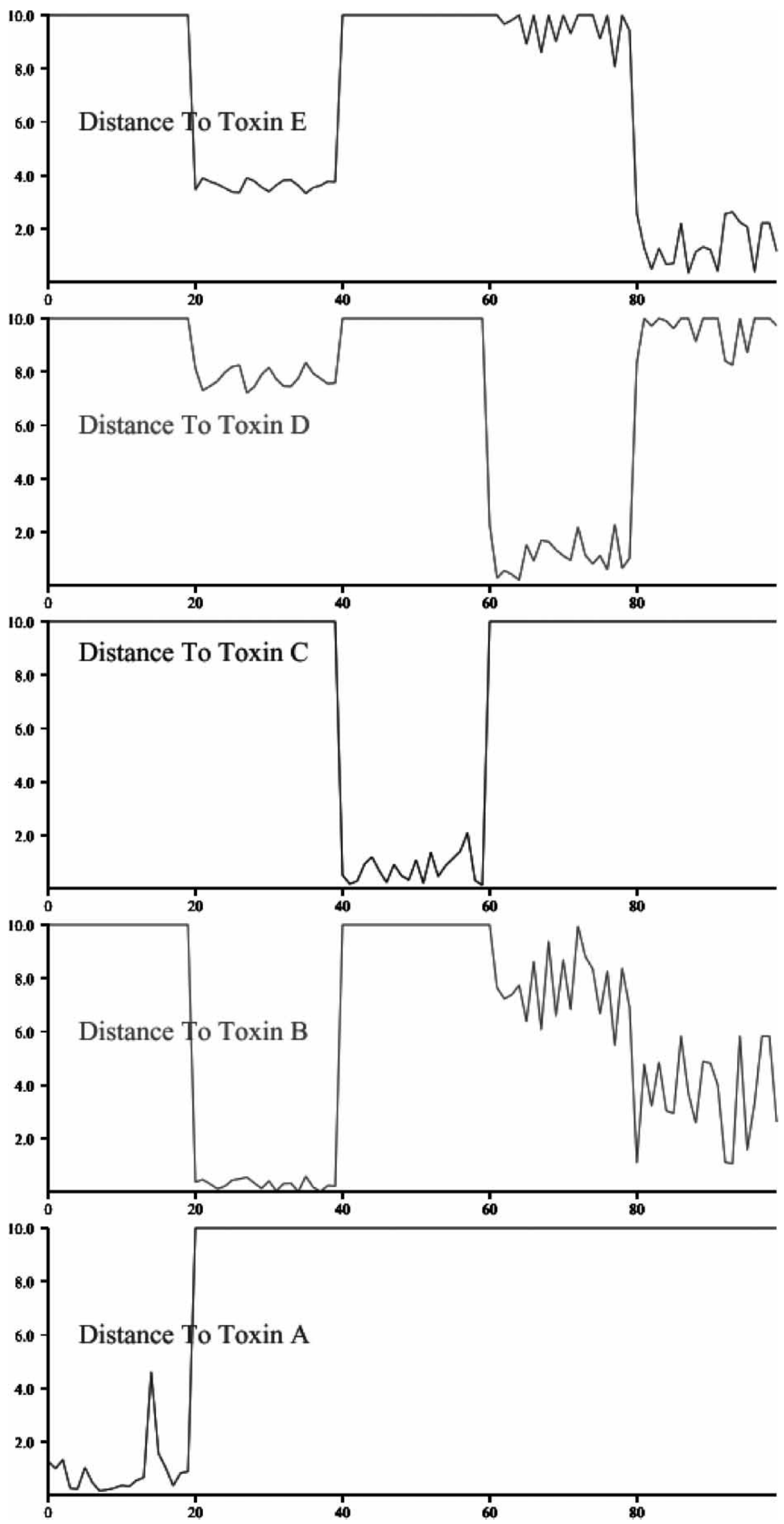

Figure 36. BFV recognizer: 11D.

References

Adams, M. and Swanson, G., 1996, Tins neurotoxins supplement, *Trends In Neurosciences*, Suppl.: S1–S36.

Grossberg, S., 2003, How does the cerebral cortex work? Development, learning, attention and 3d vision by laminar circuits of visual cortex, Technical Report TR-2003-005, Boston University, CAS/CS.

Grossberg, S. and Seitz, A., 2003, Laminar development of receptive fields, maps, and columns in visual cortex: the coordinating role of the subplate, Technical Report 02-006, Boston University, CAS/CS.

Harvey, A., 1990, Presynaptic effects of toxins, *International Review of Neurobiology*, **32**, 201–239.

Hodgkin, A., 1952, The components of membrane conductance in the giant Axon of Loligo, *Journal of Physilogy (London)*, **116**, 473–496.

Hodgkin, A., 1954, The ionic basis of electrical activity in nerve and muscle, *Biological Review*, **26**, 339–409.

Hodgkin, A. and Huxley, A., 1952, Currents carried by sodium and potassium ions through the membrane of the giant Axon of Loligo, *Journal of Physiology (London)*, **116**, 449–472.

Johnston, D. and Wu, S.M-S., 1995, *Foundations of Cellular Neurophysiology* (Cambridge, MC: MIT Press).

Kaul, P. and Daftari, P., 1986, Marine pharmacology: bioactive molecules from the sea, *Annual Review Pharmacological Toxicology*, **26**, 117–142.

Peterson, J., 2002b, Excitable Cell Modeling, Department of Mathematical Sciences, www.ces.clemson.edu/~petersj/Books/HodgkinHuxley.pdf

Raizada, R. and Grossberg, S., 2003, Towards a theory of the laminar architecture of cerebral cortex: computational clues from the visual system, *Cerebral Cortex*, 100–113.

Schiavo, G., Matteoli, M. and Montecucco, C., 2000, Neurotoxins affecting neuroexocytosis, *Physiological Reviews*, **80**(2), 717–766, April.

Schumaker, 1980, *Spline Functions: Basic Theory* (New York: Wiley-Interscience).

Stahl, S., 2000, *Essential Psychopharmacology: Neuroscientific Basis and Practical Applications*, 2nd edition (New York: Cambridge University Press).

Strichartz, G., Rando, T. and Wang, G., 1987, An integrated view of the molecular toxicology of sodium channel gating in excitable cells, *Annual Review of Neuroscience*, **10**, 237–267.

Theodoridis, S. and Koutroumbas, K., 1999, *Pattern Recognition* (New York: Academic Press).

Wang, C., Chen, P., Madhukar, A. and Khan, T., 1999, A machine condition transfer function approach to run-to-run and machine-to-machine reproducibility of iii–v compound semiconductor molecular beam epitaxy, *IEEE Transactions on Semiconductor Manufacturing*, **12**(1), 66–75.

Wu, C. and Narahashi, T., 1988, Mechanism of action of novel marine neurotoxins on ion channels, *Annual Review Pharmacological Toxicology*, **28**, 141–161.