Investigation of Stochastic Synchronization in Bursting Oscillators

A thesis submitted towards partial fulfillment of

BS-MS dual degree programme

by

Anand Pathak

Under the guidance of

Dr. Pranay Goel

Assistant Professor

Mathematics & Biology

IISER Pune

Indian Institute of Science Education and Research Pune

Certificate

This is to certify that this dissertation entitled "**Investigation of Stochastic Synchronization in Bursting Oscillators**" towards the partial fulfillment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research Pune, represents original research carried out by Anand Pathak at Indian Institute of Science Education and Research Pune, under the supervision of Dr. Pranay Goel, Assistant Professor, IISER Pune during the academic year 2010-2011.

Name of the student : Anand Pathak

Supervisor	Head (Physical Sciences)
Dr. Pranay Goel	Prof. K.N. Ganesh
Date:	Date:
Place:	Place:

Acknowledgement

I would like to express my gratitude towards my project superviser Dr. Pranay Goel whose conception and initiation led to the start of this project itself. He provided me with useful guidance and new ideas constantly throughout the project and showed confidence in me. He ensured the availability of all the infrastructural and logistic support I needed to carry out my simulations and provided me with all the required reference materials. I would also like to thank the technical staff of IISER Pune, especially the administrator Neeta Deo, who always responded to all the technical grievances promptly. I owe my gratitude towards my labmates and my batchmates, especially Rashmi Kulkarni, Sheetal Kumar Jain and Smrati Katiyar for providing me with their useful suggestions and insights. Without the support of all of them, this project would not have been possible.

Abstract

Stochastic Synchronization in uncoupled oscillators is a well studied phenomenon for many naturally occuring oscillators. The idea of this phenomenon is quite counter-intutive since in this process, introduction of noise in a dynamical system brings about order and synchrony. The basic idea is that when two uncoupled oscillators are undergoing their cycles and are subjected to some noises coming from common sources and thus having a correlation, they can tend to synchronize due to the perturbations of the correlated noises. The theory of stochastic synchronization is based on the concept of Phase Response Curve (PRC) which is the charecteristic of a given oscillator. It has been studied many times through numerical and experimental methods in simple oscillators but has not been much explored for the bursting oscillators. In this study, I have tried to investigate the phenomenon of stochastic synchronization in Hindmarsh Rose (HR) oscillator which is a basic model type of neuronal bursting oscillators. This class of oscillators, called bursters, is complicated one as the system alternates between oscillations and stable fixed points. The aim of this project was to compute an accurate Phase Response Curve of HR using different methods and then to use it to simulate the noise perturbed HR oscillations and finally trying to determine wether the stochastic synchronization is exhibitted in HR or not and if not, what could be the possible reasons.

Contents

Chapter 1

Introduction

Stochastic effects on dynamical systems are known to lead to counter-intuitive behaviors unpredictable by deterministic theory. They seem to increase the orderly behavior in certain dynamical systems in several instances. These effects are counter-intuitive because any addition of noise into a dynamical system is generally seen as having a detrimental effect on the coherence and order of the system. It has been studied that a common noise source can synchronize oscillators that are uncoupled (see [1]). When two non-interacting oscillators are subjected to the same noisy forcing simultaneously, they tend to synchronize. Many studies were undertaken in this direction to study this stochastic synchronization [20-23]. In this study I have tried to delve into the stochastic effects on the uncoupled bursting oscillators.

1.1 Phase formulation of an oscillator

A bursting oscillator is a many variable system whose dynamics can be quite complex to study [17,18]. The study of its oscillations becomes much simpler when we consider the oscillator through phase formulation [19].

Consider a many variable oscillating system in a stable limit cycle Γ [13]. Let the system be governed by system of ordinary differential equations

$$x' = f(x), \qquad x \in R^n,\, n \geq 2$$

then each point $x \in \Gamma$ is associated with a unique phase θ. Suppose the time period of the oscillation is T and at time $t = 0$ the system is at $x_0$on the limit cycle. The phase is actually the mapping of the points on the phase trajectory of the limit cycle of the system to a scalar variable $\theta \in [0, 1)$. This mapping is characterized by:

$$\frac{d\theta}{dt} = \omega$$

where $\omega = 1/T$. Now if x_0 on the limit cycle is taken as a reference point i.e. $\theta(x_0) = 0$ and $x(t) = x$, the phase of the point x on the limit cycle will be

$$\theta(x) = t/T \bmod 1$$

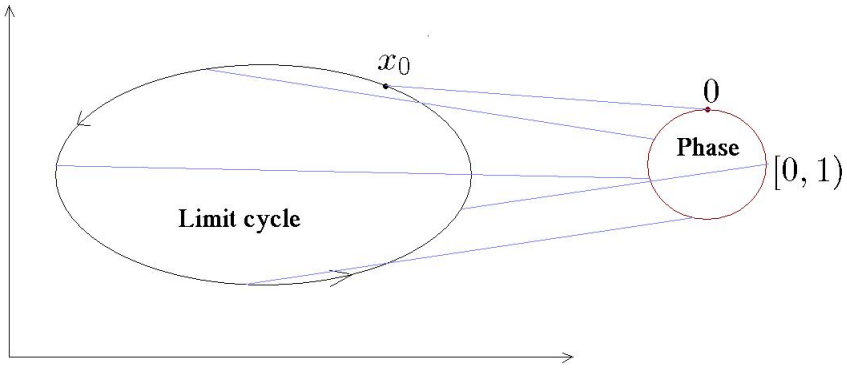


Figure 1.1: Schematic representation of the phase of an oscillator and its limit cycle.

This mapping can be extended to the points on the basin of attraction of the limit cycle. For a point x_0 on the limit cycle with phase $\theta = \theta(x_0)$ and with subsequent trajectory $x(t)$, a point y_0 in the neighborhood of x_0 in the phase space, with its subsequent trajectory $y(t)$, has the asymptotic phase $\vartheta(y_0) = \theta$ if :

$$\lim_{t \to \infty} \| x(t) - y(t) \| = 0$$

for $t \to \infty$. The set of all such points y_0 is called the *isochron* of phase θ or the *isochron* [28] of x_0.

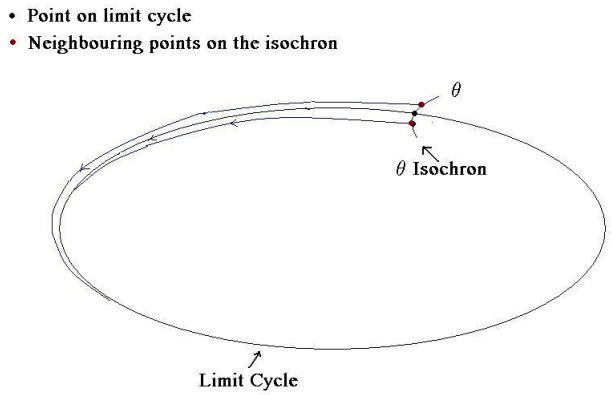


Figure 1.2: Points on the limit cycle and the neighboring points of same phase θ form an isochron.

When we are studying synchronization, it is convenient to study the dynamics of the phase of an oscillator. The oscillator is then reduced to a single variable problem. Here, I studied the dynamics of the oscillator under stochastic fluctuations. Hence to incorporate the stochastic noise in the dynamical system I used the stochastic differential equation (SDE) [8,24] as described in Galan et al. [1] :

$$\frac{d\theta_i}{dt} = \omega_i + Z(\theta_i)\eta_i(t) \tag{1.1}$$

Here, $\theta_i(0) = \theta_i(2\pi)$, θ_i is the instantaneous phase of the oscillator i, ω_i is the average regular bursting frequency, $Z(\theta)$ is the infinitesimal Phase Response Curve (PRC) of the oscillator, and $\eta(t)$ are the zero mean, white noise or Brownian noise stochastic inputs with $\langle \eta_i(t)\eta_i(t-\tau)\rangle = \sigma_i^2\delta(\tau)$, σ_i being the noise amplitude of the oscillator i and δ the Dirac delta function. The Brownian noise not have any temporal correlation but when there are noises from distinct sources, noise can be spatially correlated ie. noise acting on two different oscillators can have a positive cross correlation such that $\langle \eta_i(t)\eta_j(t)\rangle = c\delta(0)$, for some oscillators i and j. Here, the correlation coefficient of the inputs is $r = c/\sigma_i\sigma_j$. This correlation is due to a common sourse of noise among the two oscillators out of many different sources.

Using this phase formulation of the oscillator instead of direct dynamical equations we simplify the study of its dynamics and synchronization. To study the effect of noise inputs on the individual oscillators as well as on the group of oscillators as a whole, we need to solve the stochastic differential equations for every oscillator, whose solutions would be the time evolution of individual phases of the oscillators.

1.2 Phase Response Curve

The effect of a perturbation on the phase an oscillator is measured by its Phase Response Curve or the Phase Resetting Curve [4]. Suppose we perturb a dynamical system in its limit cycle at point $x_0 \in \Gamma$ (limit cycle) having phase $\theta(x_0) = \theta$ and original time period of the oscillator is T. Due to a weak perturbation, the system would typically get displaced to some point in the basin of attraction of the limit cycle. For a weak perturbation, we consider the system to fall back into the limit cycle (or close enough to be considered in the limit cycle) during the completion of that cycle in which it was perturbed and hence cross the isochron of $\theta(x_0)$ after time T'. Without any perturbation, it would return to the same point after the time period T. The difference between T' and T is due to the fact that the perturbation has shifted the system to a point y_0 with phase $\theta(y_0) = \theta'$. The shift of phase due to perturbation is

$$\Delta\theta = \theta' - \theta$$

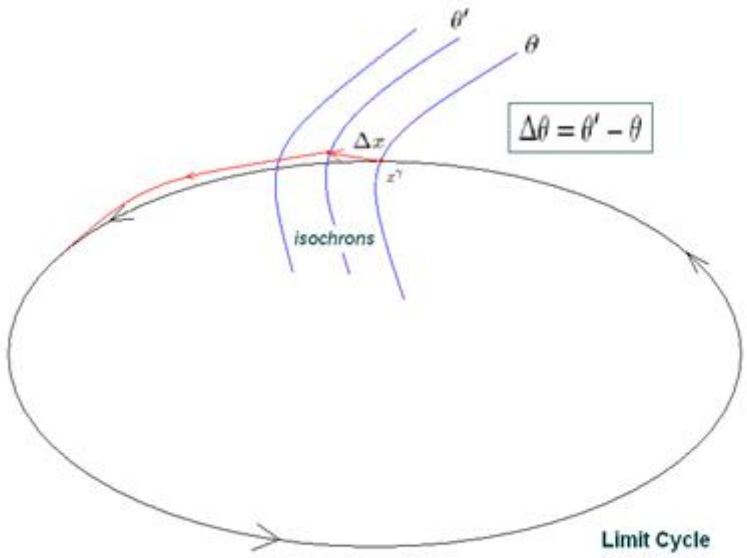


Figure 1.3: When the system is perturbed by Δx , the phase is shifted as the system jumps to another isochron.

Now, since after time T' it returns to the the same phase θ whereas if the system is unperturbed the phase would return to θ after time T. The phase displacement for both cases, perturbed and unperturbed, can be equated as

$$\theta' + \omega T' = \theta + \omega T$$

$$\theta' - \theta = \omega T - \omega T'$$

and thus,

$$\Delta\theta = 1 - \frac{T'}{T} \tag{1.2}$$

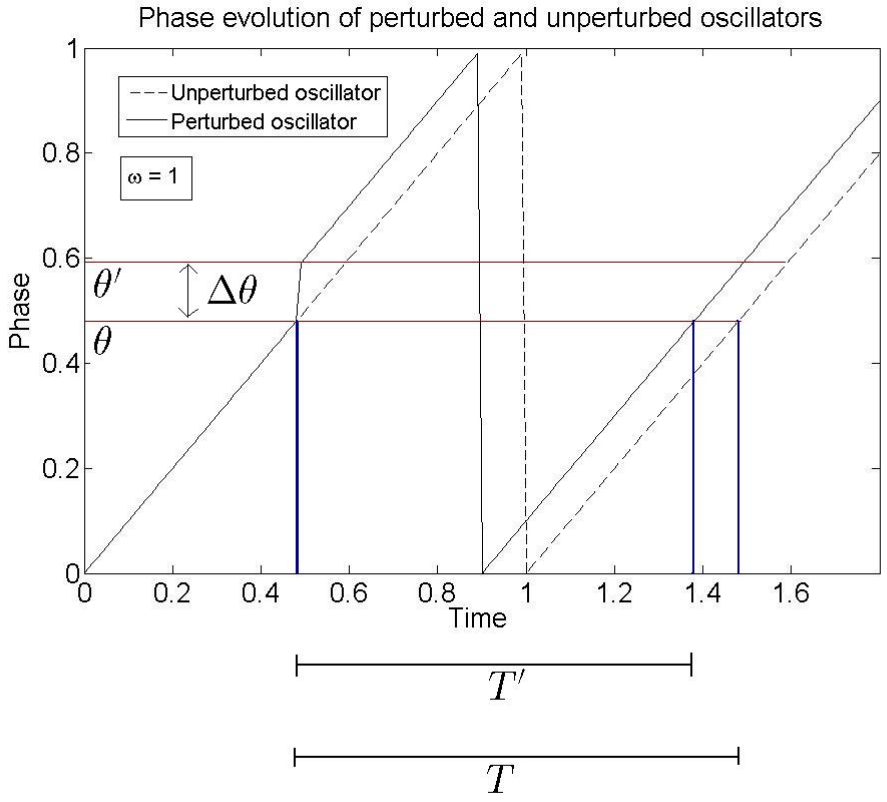


Figure 1.4: When the system is perturbed at phase θ the time taken to return to phase θ is T' , whose relation to time period T is displayed in the diagram. $\theta' + \omega T' = \theta + \omega T$.

This phase shift $\Delta\theta$ is a function of x, the point on the limit cycle where the perturbation is given as well as the perturbation itself. Hence this function $\Delta\theta(x)$ is called the

Phase Response Curve.

Positive phase response (phase shift) indicates phase advancement and negative phase response indicates phase delay. Since $\theta(x)$ is a scalar defined over the limit cycle and the neighboring points we can define for infinitesimally small perturbation i.e. $\parallel \triangle x \parallel \rightarrow 0$, the infinitesimal phase response [2,4] is given by:

$$\Delta\theta(x) = \Delta x \cdot \nabla\theta(x) = \left\langle (\Delta x_1, ..., \Delta x_n), (\frac{\partial\theta}{\partial x_1}(x), ..., \frac{\partial\theta}{\partial x_n}(x)) \right\rangle \tag{1.3}$$

Here we call $\Delta\theta(\theta(x))$ as Phase Response Curve (PRC) and $\nabla\theta(x)$ as infinitesimal PRC or iPRC, then we have

$$PRC = \Delta x \cdot iPRC \tag{1.4}$$

When the perturbation is along only one variable, such as V, the trans membrane potential of the system,

$$\Delta x = (\Delta V, 0, 0....0)$$

then the phase response becomes

$$\Delta\theta(x) = \Delta V \frac{\partial\theta}{\partial V}(x) \tag{1.5}$$

To find the iPRC $\nabla\theta(x)$, there is a direct analytical method called the *adjoint method*. This method is encoded in XPPaut using Graham Bowtell algorithm [7]. There are other two numerical methods developed and explored in this study :

1. The direct perturbation method
2. The pulse method

1.3 Hindmarsh-Rose neuronal bursting model

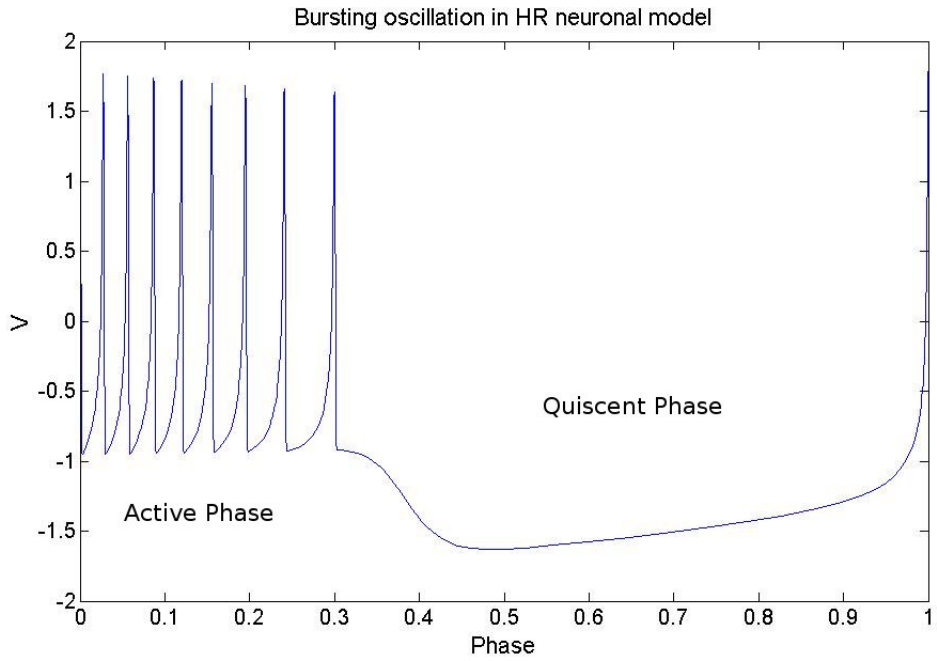
The Hindmarsh Rose model is a prototypical model of neuronal bursting which was originally introduced to model the neurons of the pond snail *Lymnea* and the R15 neuron of the mollusc *Aplysia* [11]. The HR equations are :

$$V' = n - aV^3 + bV^2 - h + I \tag{1.6}$$

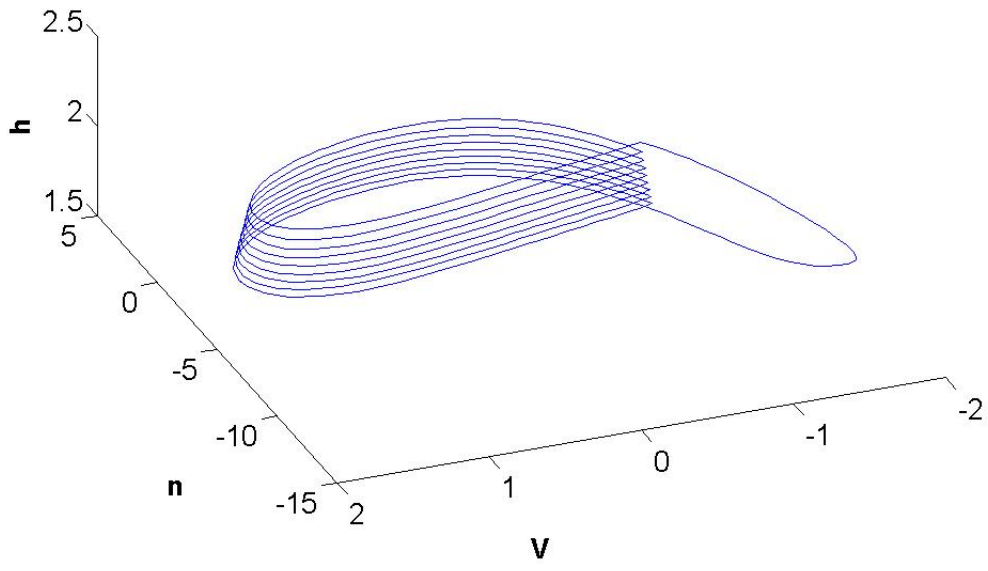
$$n' = c - dV^2 - n \tag{1.7}$$

$$h' = r(\sigma(V - V_0) - h) \tag{1.8}$$

The parameter set used here is a = 1, b = 3, c = 1, d = 5, r = 0.001, $\sigma = 4$, $V_0 = -1.6$, I = 2, as described in Sherwood et al. [2] which produces a bursting periodic orbit with nine spikes per burst (Fig.(1.5)). V is the membrane voltage of the neuron and I is the applied current. Variable h is a slow variable here and the alternate shifts between quiescence and spiking depend upon its dynamics. Parameter r represents the seperation of the time scales between V and h (for further details see [2] or [11]). A typical one cycle of an HR bursting looks like the one shown in Fig.(1.5).



a)



b)

Figure 1.5: Single cycle of neuronal bursting oscillation in HR model in which V oscillates in alternate active and quiescent phases (a) and the limit cycle in the 3-D space (b) are shown. These figures have been plotted using Matlab.

Here the time has been rescaled to make the time period 1. Thus Fig.(1.5, (a)) effectively plots voltage as a function of phase rather than time. This was done since we want

to calculate PRC as a function of phase of the system in which it is perturbed.

Chapter 2

Computation of PRCs

To study the effect on an oscillator by perturbations in general and particularly noise in this case, we require the PRC of the oscillator. In this part I tried to compute PRC and iPRC of HR bursters. There is an analytical method called the adjoint method [4,6,26,27], to compute the iPRC of any oscillator. The adjoint method has its own limitations [29]. Hence I also tried to develop some numerical strategies using the definition of PRC (Eq.(1.2)) and relation between iPRC and PRC (Eq.(1.4)), calling them Direct Perturbation method (or Impulse Method) and Pulse method. I tried to compute an accurate and consistent iPRC of one simple bursting oscillator which is Hindmarsh-Rose oscillator through three different methods. By consistency, I mean that the iPRCs computed through the different methods must agree amongst each other and hence must be accurate and trustable.

2.1 Adjoint method for the calculation of iPRC:

The description of the adjoint method here is taken from in Brown et al. [4]. Consider a non-linear dynamical n-dimensional system

$$\frac{dx}{dt} = F(x)$$

Suppose the system is in its stable limit cycle and is perturbed by a small quantity Δx at a point x^γ in its trajectory (see Fig.(1.3)). Applying Taylor's expansion in the above equation and then the linear approximation for small perturbation, we get an equation for the evolution of the perturbation of the system:

$$\frac{d\Delta x}{dt} = DF(x^\gamma)\Delta x$$

here, DF is the Jacobian matrix of F.

Now, since every point in the trajectory and in the neighborhood of the trajectory, x, is mapped into a scalar quantity called phase i.e. $\theta(x)$, we can denote the shift in the phase due to perturbation

$$\Delta\theta = \theta(x) - \theta(x^\gamma)$$

$$= \nabla\theta\mid_{x^\gamma} \cdot \Delta x$$

Differentiating the above equation with time,

$$\frac{d\Delta\theta}{dt} = \frac{d\nabla\theta\mid_{x^\gamma}}{dt} \cdot \Delta x + \nabla\theta\mid_{x^\gamma} \cdot \frac{d\Delta x}{dt} = 0$$

and the phase shift $\Delta\theta$ is constantly maintained throughout the trajectory. Hence

$$\frac{d\nabla\theta\mid_{x^\gamma}}{dt} \cdot \Delta x = -\nabla\theta\mid_{x^\gamma} \cdot \frac{d\Delta x}{dt}$$

$$= \nabla\theta\mid_{x^\gamma} \cdot (DF\mid_{x^\gamma} \Delta x)$$

therefore,

$$\frac{d\nabla\theta\mid_{x^\gamma}}{dt} \cdot \Delta x = (DF^\dagger\mid_{x^\gamma} \cdot \nabla\theta\mid_{x^\gamma})^\dagger \cdot \Delta x \tag{2.1}$$

$DF^\dagger$ is the adjoint of DF (just transpose in this case since DF is real). Now, since Eq.(2.1) holds for arbitrary Δx we get :

$$\frac{d\nabla\theta\mid_{x^\gamma}}{dt} = (DF^\dagger\mid_{x^\gamma} \cdot \nabla\theta\mid_{x^\gamma}) \tag{2.2}$$

Also, noting that

$$\omega = \frac{d\theta}{dt} = \nabla\theta \cdot \frac{dx}{dt} = \nabla\theta \cdot F(x)$$

where ω is the oscillation frequency, we get one boundary condition for solving Eq.(2.2)

$$\nabla\theta\mid_{x^\gamma} \cdot F(x)\mid_{x^\gamma} = \omega \tag{2.3}$$

at time $t = 0$. It gives 1 of the n required boundary conditions. Remaining conditions arise from the constraint of periodicity of the solution with time period T

$$\nabla\theta(x(t)) \mid_{t=0} = \nabla\theta(x(t)) \mid_{t=T} \tag{2.4}$$

Adjoint method is implemented in XPPaut to solve the Eq.(2.2) using Eq.(2.3) and Eq.(2.4) as boundary conditions and obtain the solution for iPRC $\nabla\theta(x^\gamma)$. In XPPaut, this adjoint method is inbuilt to compute the iPRC [7]. If for a system $\omega \neq 1$, the PRC produced by XPP (say we call it Q_{XPP}), is different from the iPRC $\nabla\theta$ by a scaling factor equal to ω, that is:

$$\nabla\theta = \omega Q_{XPP} \tag{2.5}$$

Results with Adjoint Method

Fig.(2.1) shows the iPRC of HR bursting using the adjoint method. This is actually the V component of the solution ie. $\frac{\partial\theta}{\partial V}$ vs phase.

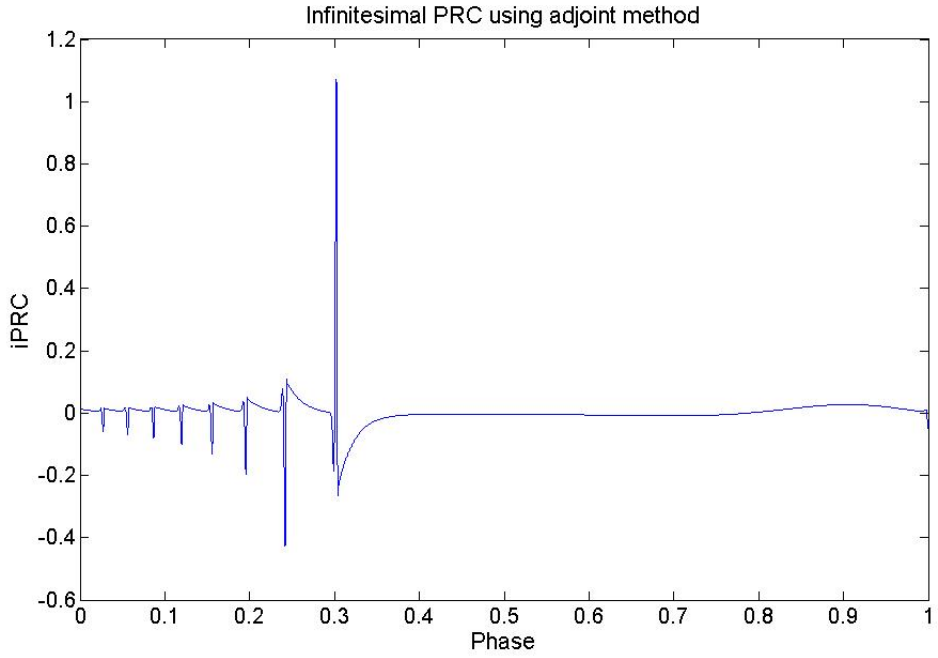


Figure 2.1: Infinitesimal PRC of neuronal bursting in HR model

When this method was used to compute iPRCs of islet bursters [5] (not shown), the result it produced was very low amplitude (upto order 1e-6) in a major portion, even in the active region, and at few points it steeply rose to peaks of values as high as 1000. Moreover, as the iPRC was calculated with a single cycle of the same burster with the initial and the final point changed, that is to say with the reference point $\theta = 0$ shifted to some other

point, the peak values also changed. This should not be the case if the calculated iPRC was correct. This meant that atleast the peaks in the islet iPRCs were not being calculated accurately. One reason could be the numerical limitations of the software and the machine used itself. The time step used in the solving algorithm by XPP might not have been small enough to solve for the peaks accurately. Hence, before accepting the solution for HR iPRC as an accurate iPRC I had to verify its validity with respect to Eq.(1.4) which relates the observed phase response due to a perturbation to the iPRC and the perturbation. To obtain the phase response curve due to a given perturbation, I explored two numerical approaches :

1. The direct perturbation method, or, the impulse method
2. The pulse method

2.2 The direct perturbation method

Perturbation to any of the variables of a dynamic system causes the phase of the system to shift. This phase shift is given by the PRC as we have seen earlier. If this perturbation occurs within an infinitesimal time, the perturbation would be said to caused by an impulse. Suppose the perturbation is caused due to an impulse to one of the system variables, say membrane voltage V, so as to shift the phase of a system instantaneously, then this can be expressed as

$$V' = f(V) + I\delta(t - t^*) \tag{2.6}$$

Here, I is the amplitude of the impulse applied (current in this case) at time t*, V is the membrane voltage and δ is the Dirac-delta function. The perturbation ΔV would be the difference between the V value before the impulse and after the impulse. In this case, the perturbation $\Delta V = I$. From section 1.2, we see that assuming the perturbation is weak, the PRC would be given by Eq.(1.2)

$$\Delta\theta = 1 - \frac{T'}{T}$$

By weak perturbation, here it is meant that the system falls back into the limit cycle or comes close enough to be called in the limit cycle within one time period of perturbation. Thus by simulating this impulse perturbation in an oscillating system having time period T, if we record the altered time period T', we can get the PRC.

2.2.1 Determining the size of a perturbation useful for computing iPRC

One condition for the appropriate perturbation for this method of computing PRC is that it should be weak. One more condition is required to be fulfilled for computing the iPRC from the PRC obtained through this method and that is given by Eq.(1.4). The PRC must be linearly related to ΔV and iPRC. Thus by computing PRC and multiplying it with $\frac{1}{\Delta V}$ must give the iPRC if ΔV is small enough for linear approximation. Hence an appropriate size of perturbation must be weak as well as valid for the linear approximation. The following procedure was followed to determine the appropriate perturbation size.

To simulate the impulse at a phase point θ numerically, I used XPP integrator. First I simulated the dynamical system with appropriate parameters and an initial conditions for a long duration so as to pass the transient period and take the system to the stable limit cycle where the oscillations become uniform in time. Then I set the point of maximum of V as an initial condition with $\theta = 0$ at maximum V. Starting with this as an initial point I simulated the system upto to the phase point θ. At that point, I changed replaced V by $V + \Delta V$ and continued the simulation from that perturbed point upto the point of next V maxima. Thus by picking the phase point of maximum of V after the perturbation I could obtain the altered time period T' and hence $\Delta\theta$. It is because of this direct change of V at a point by adding ΔV for simulation of impulse, this method is called direct perturbation method or the impulse method. Fig.(2.2) shows this schematically.

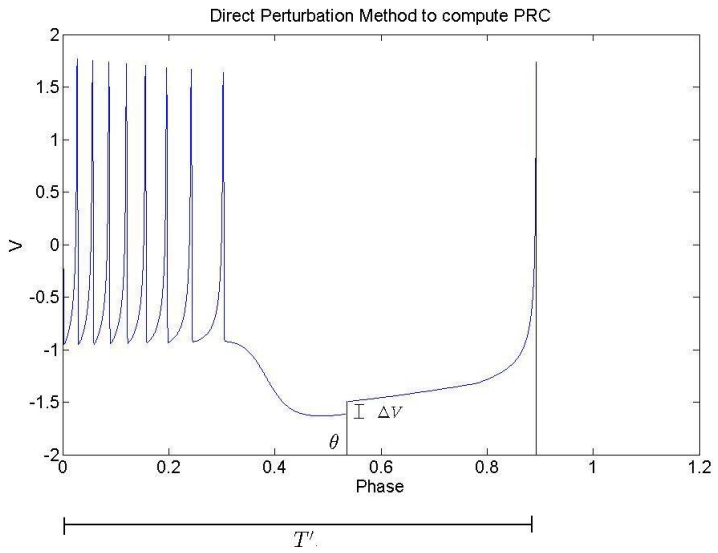


Figure 2.2: Graphical representation of the direct perturbation method to compute PRC

I obtained phase response with varying perturbation size at few phase points. Then I plotted the phase response with the magnitude of perturbation ΔV at few given phase

points. For ΔV of the order 1e-3, I obtained a linear plot with slope approximately equal to the iPRC obtained at that point by the adjoint method, as shown in Fig.(2.3). This was in agreement with Eq.(1.4). This confirmed that the perturbations used here were the appropriate for the method. This also verified the validity of the iPRC from adjoint method in this case as well as the PRC obtained from the direct perturbation method.

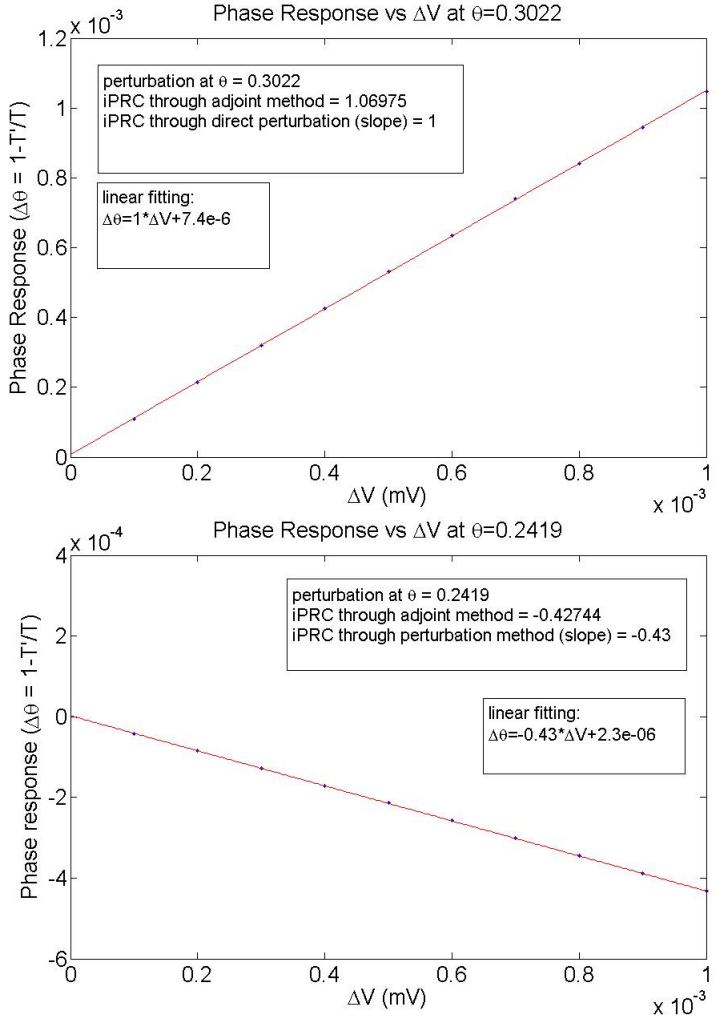


Figure 2.3: The plot between phase response and ΔV is linear as expected from (Eq.1.4). This validates the range of perturbations used as weak perturbations and can be used to calculate the PRCs and iPRCs. The two phase points shown here, $\theta = 0.3022$ and $\theta = 0.2419$, are the points where there are peaks in the iPRC (Fig.(2.1)). The iPRC value given by the slope is close to the one given by adjoint method.

The phase points in Fig.(2.3) are of that of the peaks in the adjoint iPRC. The relative error in impulse method iPRC with respect to the adjoint iPRC must be maximum for these peak points since the absolute values of iPRC are maximum there. Since the impulse

method is showing so less deviation from the adjoint iPRC even at the peaks (as shown in the figure), the relative error is expected to be very low throughout the iPRC. Thus using this perturbation size, I computed the PRC and the iPRC for HR.

2.2.2 Results for HR iPRC by Direct Perturbation method.

Using Matlab, I computed the PRC using direct perturbation at 100 points within one cycle of oscillation with the weak perturbations (Fig.(2.4)). For this I used the Matlab integrator 'ode23s' instead of XPP for simulating the system, with relative tolerance 1e-9 and absolute tolerance 1e-9. PRC through direct perturbation was multiplied by a factor of $\frac{1}{\Delta V}$ to get the iPRC (Eq.(1.4)) and was then compared with the adjoint iPRC. Using this method, iPRC was obtained through the different PRCs corresponding to different ΔV values.

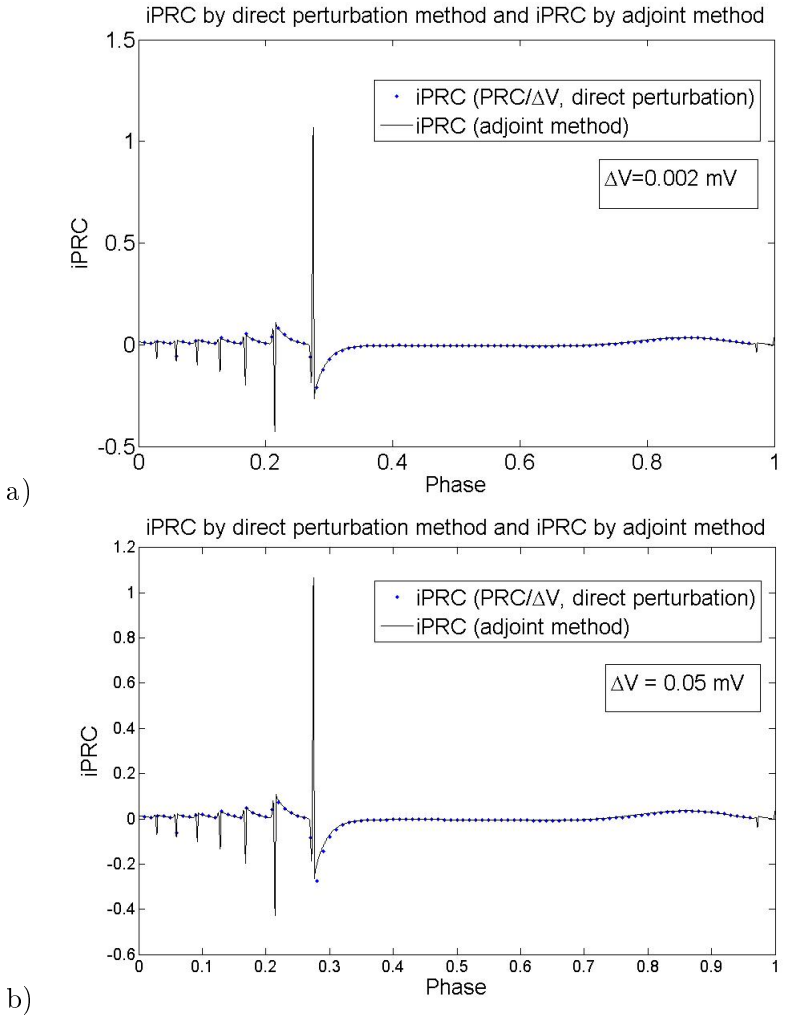


Figure 2.4: iPRCs through the direct perturbation and the adjoint method are compared and there is remarkable agreement between them. (a) $\Delta V = 0.002$, (b) $\Delta V = 0.05$. This shows that the method is valid even for perturbation order of 1e-2 mV

The iPRCs thus obtained showed clear agreement with the iPRC through adjoint method even for high order of perturbation magnitude of 1e-2 mV. Henceforth this method can be used to obtain the PRC of any oscillator but within the appropriate size of perturbation.

2.3 The pulse method

Instead of direct perturbation at desired points, we can also perturb the system by giving a sharp pulse at those points. By sharp pulse here it is meant that the pulse must be very thin and with a high amplitude. In other words a a sharp pulse is a numerical approximation

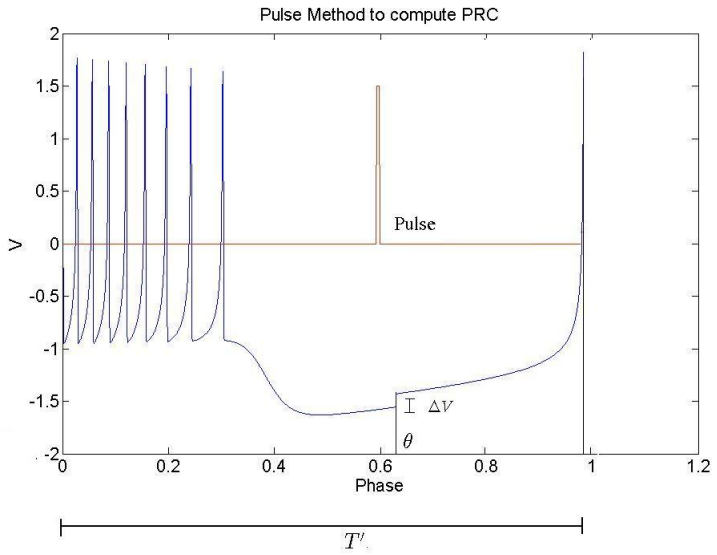


Figure 2.5: Graphical representation of the pulse method to compute PRC

of the delta function mentioned in the earlier section. The process of calculating the PRC is same as that in the direct perturbation method, only the process of perturbing the system is differing. Instead of directly changing V, I gave a rectangular current pulse of very small width and finite amplitudes, at different time points to get the respective PRCs in XPP. Here also, the maximum marks the beginning and the end of the cycle (as shown in Fig.(2.5)) and the altered time period T', could be automatically picked by stopping the simulation at the next maxima. This pulse method is the method closest to the experimental method of finding out the phase response in which a sharp current pulse is given to the neurons [25]. In a way pulse method is more realistic simulation than the adjoint method and the impulse method since direct perturbation or the Dirac-delta impulse is not possible to be applied in the laboratory.

2.3.1 Determining effective ΔV by a pulse

The effective perturbation ΔV can not determined for this method using any direct relation. The impulse is not a delta-function, and hence there is no direct way to calculate the effective value of the perturbation. Since ΔV, as we have seen earlier, is required for obtaining the iPRC from the PRC, it is important to know its value. Following the definition of perturbation used previously, ΔV is the difference between the V values before and after the instance of perturbation, I manually found the ΔV caused by the pulse directly from the data generated by XPP integrator. This was done as follows. I ran the simulation

for one period without any pulse and obtained the data series for V at all the time steps. Time step size for integration was of 1e-6s. Then I simulated the period again with the applied current pulse at a phase point with a small width and a finite amplitude. I then compared the data series obtained now with the previous one and found the difference at the phase point corresponding to the termination of the pulse. For example if the pulse was applied at 0.25 and the pulse width was 0.001, then ΔV was the difference between the V values of both the series at point (0.25+0.001) 0.251. This process I repeated for several pulse amplitudes and applied at different phase points. This exercise showed that the effective ΔV due to a pulse of a given amplitude was uniform all over the time period wherever I gave the pulse. Moreover this ΔV was proportional to the pulse amplitude, as shown in Fig.(2.6).

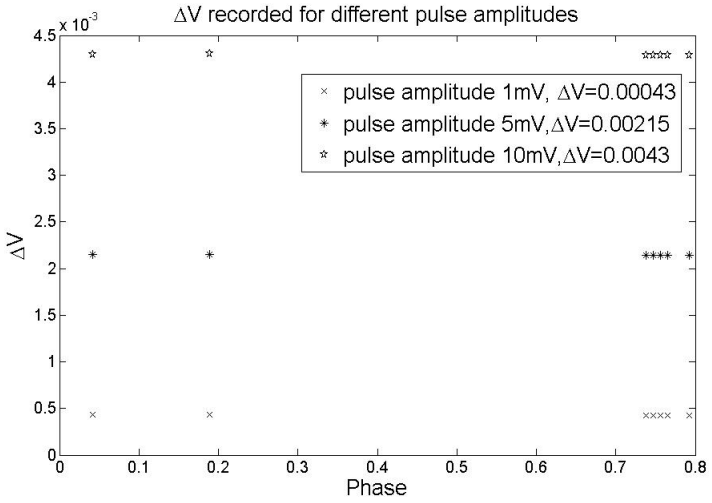
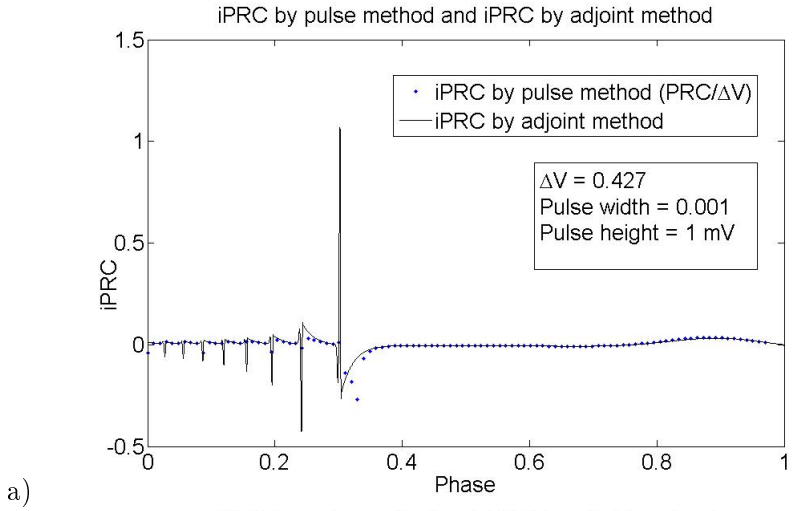


Figure 2.6: Effective ΔV due to different pulse perturbations at different phase points are shown here. ΔV was computed by giving pulses at some phase points with different amplitudes and then finding the difference of the V value at the end of the pulse with the V value at the same phase if there was no pulse. This showed that the effective ΔV due to a pulse of a given amplitude was uniform all over the time period wherever we give the pulse. Moreover this ΔV was proportional to the pulse amplitude.

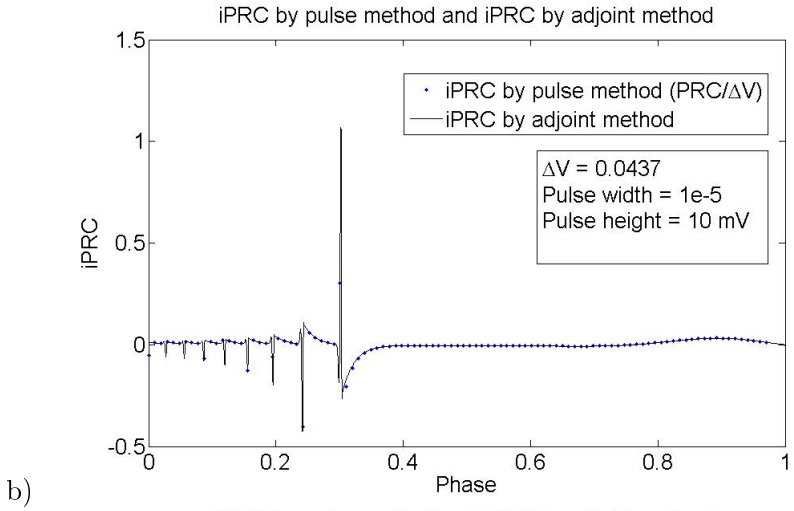
2.3.2 Results for HR iPRC by Pulse Method

Instead of computing PRC for each point singularly, we can automate the application of pulse at different phase points in a loop. This is done in XPP [7] by setting the phase point of pulse application, say τ as a variable (which remains constant valued for a single simulation run) and defining a range of values for initial value of τ. It runs the simulations for every value of τ and compute phase response for every point. To obtain the iPRC, the PRC is multiplied with $\frac{1}{\Delta V}$ as was done for the direct perturbation method. Below in

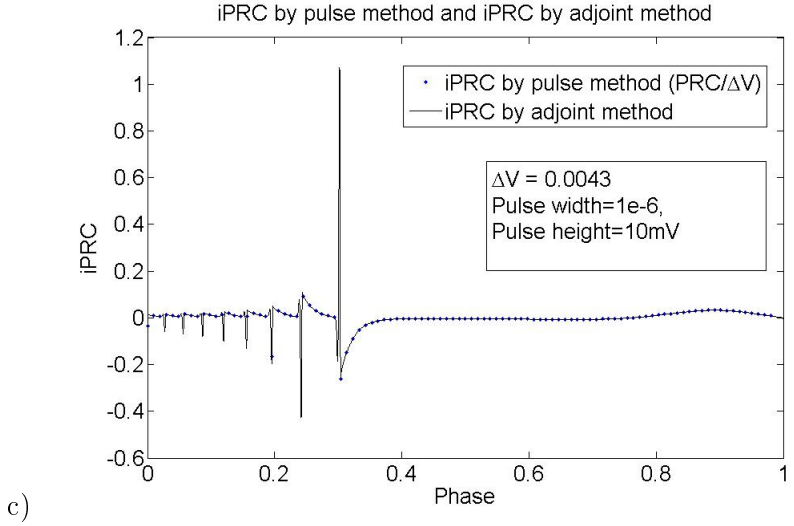
Fig.(2.7) are the iPRCs computed for 3 different pulse widths : 1e-6, 1e-5 and 1e-3. While for thin pulses of 1e-6 and 1e-5 widths, the iPRCs coincide with that from the adjoint method, the one for thicker pulse width of 1e-3 show some deviations at some regions.



a)



b)



c)

Figure 2.7: The iPRCs computed through pulse method is compared with the adjoint iPRC. The iPRCs by pulse method almost agree with adjoint iPRC. Though in (a), there are deviations in some regions. This is due to the fact that the effective ΔV in (a) is quite large for the infinitesimal linear relationship of Eq.2.3. (a) pulse width = 1e-3, pulse height = 1mV, (b) pulse width = 1e-5, pulse height = 10mV, (c) pulse width = 1e-6, pulse height = 10mV.

2.4 Conclusive Remarks

We have seen so far that for the study of perturbations in the phase dynamics of any system we need to have an infinitesimal Phase Response Curve of the system. We saw that the iPRC and the PRC are related for small perturbations through Eq.(1.4). Using a known analytical method to compute the iPRC called adjoint method, I tried to obtain the iPRC by computing the phase response through different numerical approaches which could be used when the analytical method was not viable. These were based on the definition of the PRC (Eq.(1.2)) and the relationship between PRC and iPRC (Eq.(1.4)). Those methods produced results which were confirmed by the known results on PRC of Hindmarsh-Rose bursting oscillators. The perturbations used in the other numerical methods must be within some range of orders of magnitude for getting accurate results and I succeeded in finding that range for the HR PRC. However this could not be achieved for pancreatic islet bursters. It is rather a complex dynamical system with seven variables. The adjoint iPRC could not be obtained using XPP as mentioned earlier. The iPRC it produced was very low amplitude (upto order 1e-6) in a major portion, even in the active region, and at few points it steeply rose to values as high as 1000. Moreover the results with this same method on this same oscillator showed variations in the peak values for different positions of the reference point $\theta = 0$. Even the other methods failed to produce results consistent with each other. Using direct perturbation on islet bursters, the linear relation of Eq.(1.4) was observed at few phase points but the iPRC did not agree with the adjoint one. The pulse method provided a different result altogether, with some iPRC values for every point in the active phase whereas negligible iPRC for the quiescent phase. However, I obtained a reliable HR iPRC which was to be used further.

Chapter 3

Studying the Stochastic Synchronization numerically

In Galan et al. [1], the stochastic dynamics of a system of uncoupled neural oscillators have been studied using the phase formulation which involves solving the SDE in Eq.(1.1)

$$\frac{d\theta_i}{dt} = \omega_i + Z(\theta_i)\eta_i(t)$$

and hence generating the time series of the evolution of respective phases of the neurons and then studying their statistical properties like the coefficient of variation (CV) and the output correlation of the two neurons. In the same study they applied stochastic theory and determined the relevant statistical properties of the system by solving the Fokker-Planck equations [8,9].

I used the first approach in the study of bursting oscillators. I set up a code for solving the SDE using Euler Maruyama method [3,8]. I first tried to reproduce the results published in Galan et al. to verify the validity of my codes and the algorithm used. In that work, they had studied the two oscillator system of neurons, both of which receive noises that are correlated to each other or in other words, spatially correlated noise. Based on the two basic types of PRCs, there were two types of neurons studied :

Type I : $Z(\theta) = M(1 - cos(\theta))$ and Type II : $Z(\theta) = Nsin(\theta)$.

Here M and N are normalization constants used to make the absolute value of area under the PRCs 1.

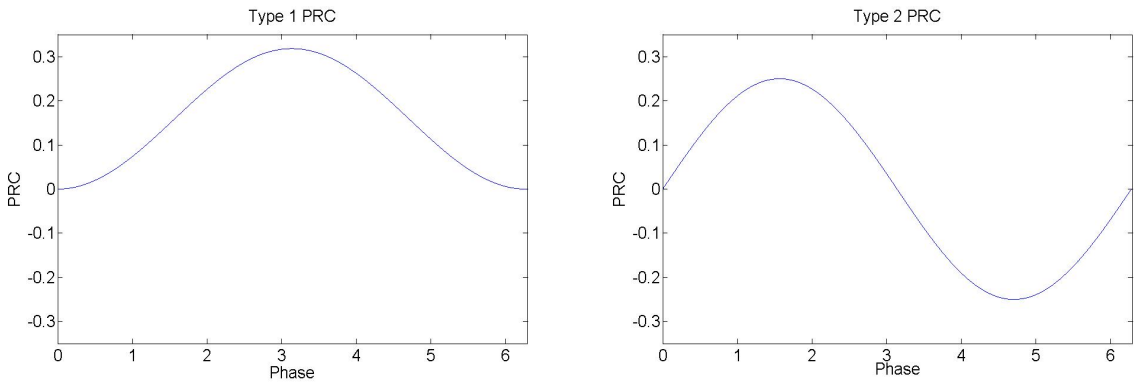


Figure 3.1: Basic types of PRCs of the neural oscillators

I set up the codes in C as well as Matlab with the Euler-Maruyama method described below.

3.1 Generation of Noise

I used the discretized Brownian Noise, also called White Noise [3], for all the simulations. It is a time series in which difference between any two consecutive points is a random variable with normal distribution with mean 0 and variance δt (time step). Suppose there are N discrete points between time 0 to T and $\delta t = T/N$, we generate the Brownian Noise series $\{W_i\}$ such that

$$W_0 = 0$$

$$W_i = W_{i-1} + dW_i$$

where dW_i is an independent random variable of the form $\sqrt{\delta t}N(0,1)$. N(0,1) denotes the normally distributed random variable with mean 0 and variance 1.

Spatially Correlated Noise : The properties of two spatially correlated white noises $\eta_1(t)$ and $\eta_2(t)$ in Eq.(1.1) as stated in section 1.1 are :

$$\langle \eta_i(t)\eta_i(t-\tau)\rangle = \sigma_i^2\delta(\tau)$$

$$\langle \eta_1(t)\eta_2(t)\rangle = c\delta(0)$$

for $i = 1, 2$, σ_i = noise amplitude, $\delta(\tau)$ = Dirac delta function, and coefficient of the noises $r = c/\sigma_1\sigma_2$.

Now, to generate two streams of discretized correlated white noise $\eta_1$and η_2 with the above properties, there is a standard method which involves generating three independent streams of white noise and then mixing them to obtain two streams of correlated noise of desired cross correlation and noise amplitude as discussed in Gardiner [8]. I modified the method by using just two independent noise streams instead of three to get the correlated noise. It helped greatly in reducing the load of running the codes on the memory as there were many more computations to follow after the generation of noise. I first generated two independent series of normally distributed random variables of the form $\sqrt{\delta t}N(0,1)$, say $\{dW_1\}$ and $\{dW_2\}$. In C++, I used the Box-Muller algorithm to convert uniform distribution to normal distribution [12]. Then I obtained the correlated noises η_1 and η_2 with correlation coefficient r and amplitudes σ_1 and σ_2 as follows:

$$\eta_1 = \frac{\sigma_1}{\delta t} dW_1$$

$$\eta_2 = \frac{\sigma_2}{\delta t}(r \centerdot dW_1 + \sqrt{1-r^2} \centerdot dW_2)$$

The cross correlation of the above generated noise :

$$c = \langle \eta_1 \eta_2 \rangle = \frac{\sigma_1 \sigma_2}{\delta t^2} \left\langle dW_1 \centerdot (r dW_1 + \sqrt{1-r^2} dW_2) \right\rangle$$

$$= \frac{\sigma_1 \sigma_2}{\delta t^2} [r \centerdot \langle dW_1^2 \rangle + \sqrt{1-r^2} \centerdot \langle dW_1 dW_2 \rangle]$$

$$= \frac{\sigma_1 \sigma_2}{\delta t^2} [r \centerdot \delta t + \sqrt{1-r^2} \centerdot 0]$$

$$= \sigma_1 \sigma_2 r \frac{1}{\delta t}$$

The auto correlation of the noise streams:

$$\langle \eta_1 \eta_1 \rangle = \left(\frac{\sigma_1}{\delta t}\right)^2 \langle dW_1 \bullet dW_1 \rangle$$

$$= \left(\frac{\sigma_1}{\delta t}\right)^2 (\sqrt{\delta t})^2 \langle \{N(0,1)\}^2 \rangle$$

$$= \left(\frac{\sigma_1}{\delta t}\right)^2 (\sqrt{\delta t})^2(1) = \frac{\sigma_1^2}{\delta t}$$

and,

$$\langle \eta_2 \eta_2 \rangle = \left(\frac{\sigma_2}{\delta t}\right)^2 \left\langle (r \cdot dW_1 + \sqrt{1-r^2} \cdot dW_2)^2 \right\rangle$$

$$= \left(\frac{\sigma_2}{\delta t}\right)^2 \left\langle r^2 (dW_1)^2 + (1-r^2)(dW_2)^2 + r\sqrt{1-r^2}(dW_1 \bullet dW_2) \right\rangle$$

$$= \left(\frac{\sigma_2}{\delta t}\right)^2 \{r^2(\sqrt{\delta t})^2(1) + (1-r^2)(\sqrt{\delta t})^2(1) + r\sqrt{1-r^2}(\sqrt{\delta t})^2(0)\}$$

$$= \left(\frac{\sigma_2}{\delta t}\right)^2 (\sqrt{\delta t})^2(r^2 + 1 - r^2) = \frac{\sigma_2^2}{\delta t}$$

Hence the statistical properties of this discretised are retained in this discretization with r being the input correlation coefficient of the above generated noises and σ_1 and σ_2 are the noise amplitudes.

3.2 Solving the SDE using Euler-Maruyama (EM) method

A scalar, autonomous SDE like the one in Eq.(1.1) can also be written in the differential form as

$$d\theta(t) = f(\theta(t))dt + g(\theta(t))dW(t) \ , \ 0 \leq t \leq T$$

Here, f and g are scalar functions and W is the Brownian function. We first discretize the interval as described by Higham [3]. Let $\Delta t = T/L$ for some positive integer L and $\tau_j = j\Delta t$. Denoting $\theta_j = \theta(\tau_j)$, the Euler-Maruyama (EM) method takes the form

$$\theta_j = \theta_{j-1} + f(\theta_{j-1})\Delta t + g(\theta_{j-1})(W(\tau_j) - W(\tau_{j-1}))$$

The EM algorithm has low order of convergence (strong order of convergence 0.5, weak order of convergence 1) and hence low accuracy [8]. This requires the Δt to be very small. However it serves as a simple and convenient method especially when complicated functions are involved. Other more accurate methods like Milstein algorithm are cumbersome to use since they require the derivatives of the functions as well.

In my simulations, I used EM algorithm for setting up an SDE solving code. The structure of the code is such that it first generates the white noise whose amplitude can be changed and adjusted, and then it uses the EM method to solve the SDE Eq.(1.1) to generate the time evolution series of the phases on every time step whose step size is Δt.

I fed the Type 1 and Type 2 PRCs in my SDE solving code and obtained the solutions for phase evolution which, when plotted with time look like the one given in Fig.(3.2). This figure is an example with Type 2 PRC.

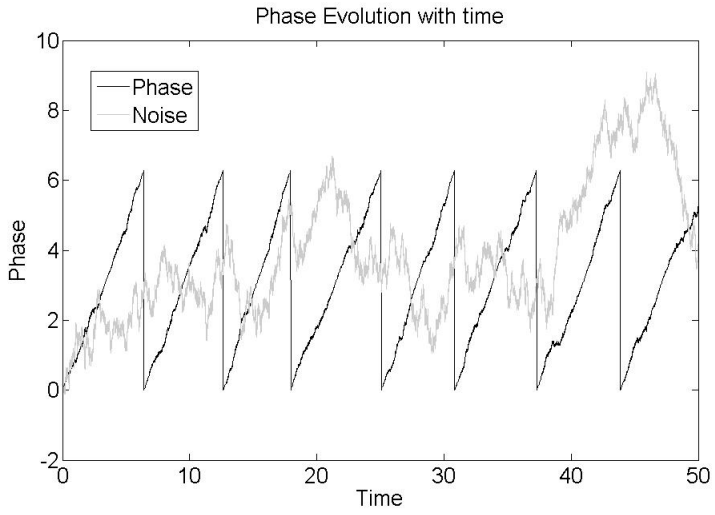


Figure 3.2: Phase evolution of neural oscillator of Type 2 PRC is shown here. The noise given to the oscillator is also shown in the figure. Here the phase is wrapped at 2π rather than 1.

3.3 Computing the Coefficient of variation

Since I was studying the synchronizing effect of noise on a bursting oscillator, it was important to measure the effect which a noise could bring upon the oscillations. Coefficient of variation (CV) is one appropriate index to gauge the magnitude of this effect. CV is the measure of robustness of an oscillator to noise. Large CV values indicate strong deviations from the regular oscillations ie. a low robustness and low CV indicates high robustness. For sufficiently large number of cycles, if T_i is the time duration of ith cycle, then CV is defined as:

$$CV = \frac{\sqrt{\langle T_i^2 \rangle - \langle T_i \rangle^2}}{\langle T_i \rangle} \ ,$$

where numerator is the standard deviation of T_i and denominator is the mean.

The other method to compute the CV in the same study of Galan et al. was through the Fokker-Plank equations (FPE) which are partial differential equations (PDE) rather than SDE.

In terms of Fokker-Plank solution as described by Galan et at. [1]

$$CV = \frac{\sqrt{T_2(0) - T_1^2(0)}}{T_1(0)}$$

Where $T_1(\theta)$ and $T_2(\theta)$ are the temporal moments , calculated from FPE as:

$$-1 = \omega \frac{\partial T_1(\theta)}{\partial \theta} + \frac{\sigma^2}{2} Z^2(\theta) \frac{\partial^2 T_1(\theta)}{\partial \theta^2} \tag{3.1}$$

$$-2T_1(\theta) = \omega \frac{\partial T_2(\theta)}{\partial \theta} + \frac{\sigma^2}{2} Z^2(\theta) \frac{\partial^2 T_2(\theta)}{\partial \theta^2} \tag{3.2}$$

with boundary conditions $T_1'(0) = T_2'(0) = 0$ and $T_1(2\pi) = T_2(2\pi) = 0$, $T_1(\theta)$ is the meantime required to go from θ to the end of the cycle $\theta = 2\pi$ and $T_2(\theta) - T_1^2(\theta)$ is the variance.

I computed the Coefficients of Variation (CV) of different oscillators using the solutions obtained from the SDE. The procedure I used to do this was the following: I extracted out the wrapping points of the solution where the phase reaches the end of a period (2π or 1, according to the scale for a given oscillator). They are represented by the peaks in Fig.(3.3). The time intervals between the consecutive points are the time periods T_i for each cycle.

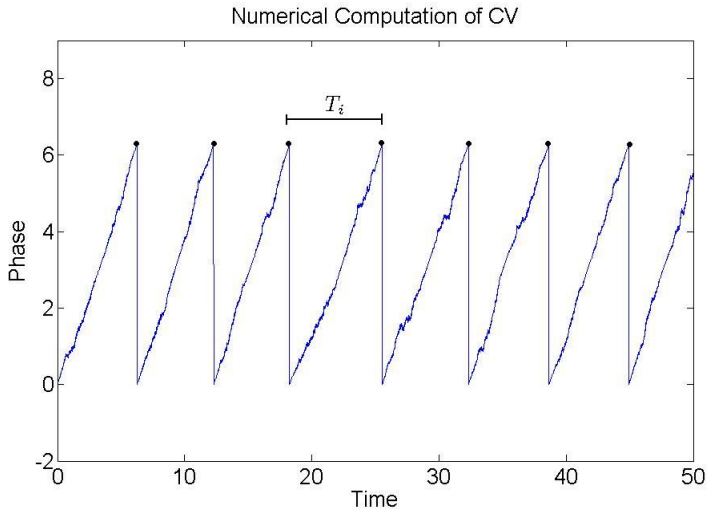


Figure 3.3: The coefficient of variation was computed by picking the wrapping points (peaks) of the phase and the intervals between the consecutive points them are the time periods T_i. CV was then calculated using these T_i.

The accuracy of the SDE solution could be gauged by comparing the CVs obtained through the SDE solutions with that obtained from FPE solutions (Eq.(3.1) and Eq.(3.2)). I obtained the CV values from FPE solution, computed by Pranay Goel for Type 1 and Type 2 PRCs as a function of noise amplitude σ and compared them with the CV values I computed through SDE solutions with different values of σ. The solution to FPEs were computed using the Finite Element Method [10] using COMSOL [30]. To ensure the better and more accurate results, I simulated the code up to sufficient number of cycles (T upto the order of 1e3 cycles) with optimum size of time step for integration ($\Delta t = 1e-4$). By optimum here it is meant that the time step must not be quite large so as to decrease the resolution and the accuracy of the solving algorithm, at the same time it must not be so small so as to increase the data points and flood the memory of the machine. In C and C++, there are memory limitations which show 'Segmentation Error' when the memory limit is surpassed. Hence after trying different values for T and Δt, and comparing the results with the FPE results, I kept improving the results until I determined the optimum value for Δt and sufficient value of duration T.

The results ultimately showed reasonable agreement.

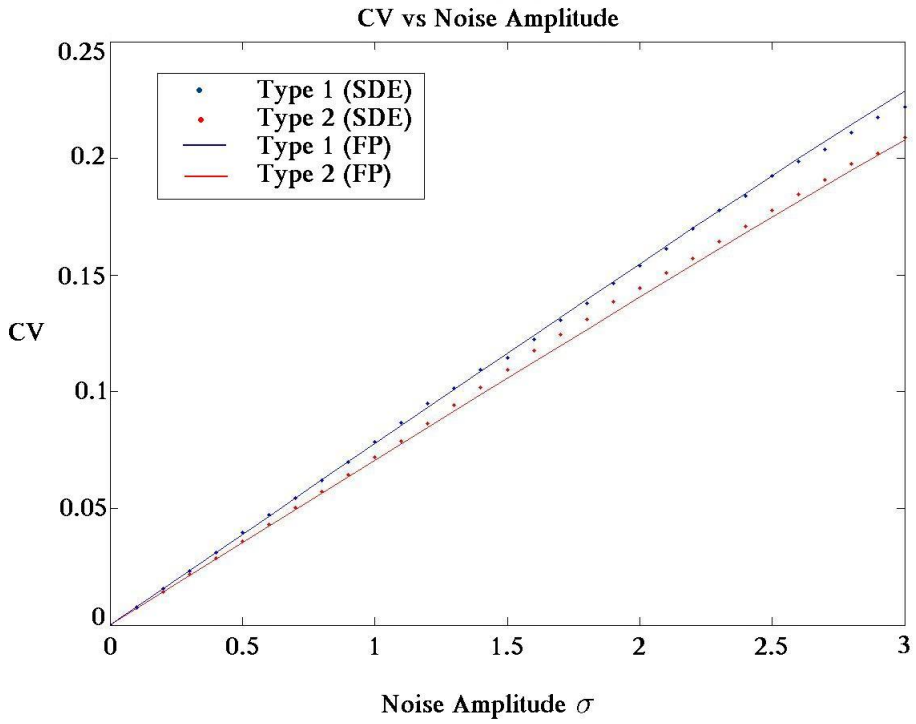


Figure 3.4: CV vs σ as computed through SDE and through FPE show consistency with each other. This confirmed that the amount of data computed was sufficient and the time step size used for the EM method was optimal.

These results were also in accordance with those produced in the Galan et al. [1] study. Using the same procedure of solving SDE, I computed the CV values for the Hindmarsh Rose oscillator as well (Fig.(3.5)).

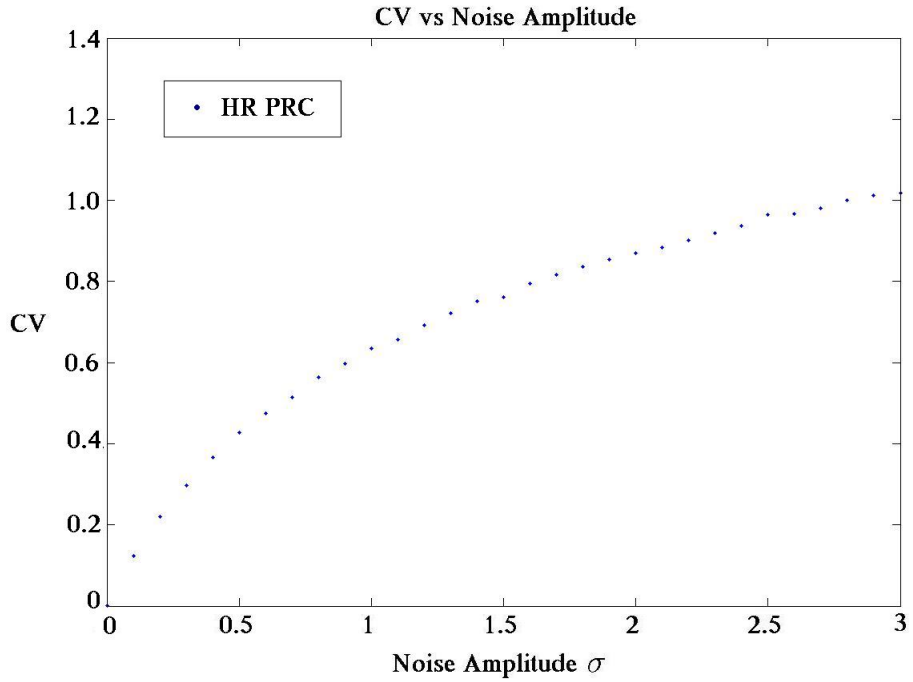


Figure 3.5: The CV vs σ plot for Hindmarsh Rose (HR) PRC shows a smoothly increasing trend and the CV values are relatively higher as compared to Type 1 and Type 2.

Here we see that the CV values are quite high relative to the Type 1 and Type 2 oscillators. This means that the HR oscillators are far less robust to the noise than the Type 1 and 2 PRCs even though the amplitude of the HR PRC (Fig.(2.1)) is quite low except for the peaks in between. The HR oscillators are quite unstable with respect to noise. This was one important aspect to know while studying the effect of noise on the HR bursters. To visualize the effect of noise on the HR oscillator owing to its characteristic PRC, here is the phase evolution plot of the HR oscillator when noise amplitude is 1 (Fig.(3.6)).

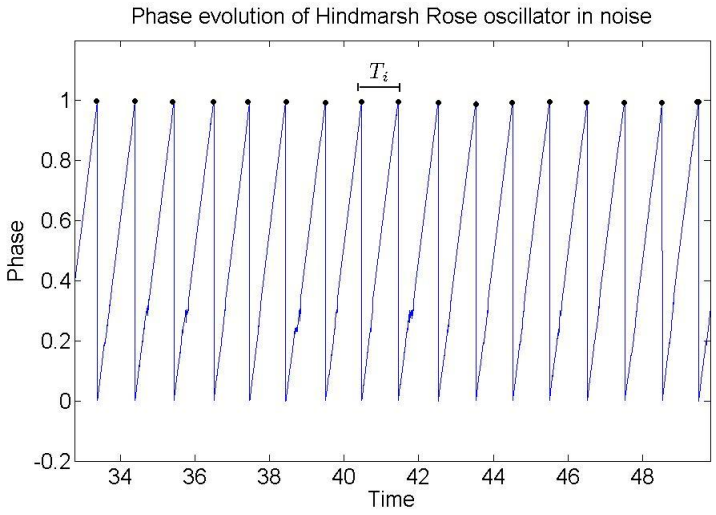


Figure 3.6: The phase evolution plot for an HR oscillator qualitatively shows the disturbance which noise produces in its oscillation cycle. Though the PRC function has few and sharp peaks (Fig.1.3), and major parts of it have very low values especially in the quiescent phase. Yet there is considerable disturbance produced by the phase response of the active part which can be seen in this figure. The jagged region in the lower parts of each cycle represents the active phase comprising of spikes in the PRC.

The finite element method to solve the FPE for HR PRC could not be worked out due the fact that the PRC function had sharp spikes (Fig.(1.3)). Hence owing to the small mesh size required to incorporate the PRC function accurately, the memory limitations were surpassed. Although from the Fig.(3.5) it can be asserted that the noise exercises considerable effect on the oscillations of the HR neurons.

3.4 Input Correlation and Output Correlation

The core of this study was to see what happens when two uncoupled bursting oscillators are subjected to spatially correlated noise. This spatial correlation is the crucial parameter here. Hence we call the coefficient of correlation r as defined in Chapter 1 as the Input Correlation.

When studying any two uncoupled oscillators and their synchronization we need a quantifying parameter. We choose it to be the cross correlation coefficient, of the phases of those two oscillators, φ_1 and φ_2. Analytically, it can be computed as described by Galan et al. [1]

$$R = \frac{\int\int_\Omega \varphi_1\varphi_2 P(\varphi_1,\varphi_2)d\varphi_1 d\varphi_2 - \int\int_\Omega \varphi_1 P(\varphi_1,\varphi_2)d\varphi_1 d\varphi_2 \int\int_\Omega \varphi_2 P(\varphi_1,\varphi_2)d\varphi_1 d\varphi_2}{\int\int_\Omega \varphi_1^2 P(\varphi_1,\varphi_2)d\varphi_1 d\varphi_2 - \left[\int\int_\Omega \varphi_1 P(\varphi_1,\varphi_2)d\varphi_1 d\varphi_2\right]^2} \tag{3.3}$$

Where, R is the Output Correlation (cross correlation coefficient) $P(\varphi_1,\varphi_2)$ is the probability of finding oscillator 1 in phase φ_1 and oscillator 2 in phase φ_2 when the system is in the steady state ie. stationary probability condition in which

$$\frac{\partial P(\varphi_1,\varphi_2)}{\partial t} = 0$$

In my numerical method of solving the SDE, this steady state can be approximated by solving the SDEs of the phases of the two oscillators for long enough time period. Now the time period of simulation for allowing a two oscillator system to achieve a steady state was again a question of trial and error. After obtaining the solutions and omitting the transient phase before the steady state, which are in form of a time series ie. φ_{1i} and φ_{2i}, the output correlation R in this case is the cross correlation of the two time series :

$$R = \frac{\langle\varphi_{1i}\varphi_{2i}\rangle - \langle\varphi_{1i}\rangle\langle\varphi_{2i}\rangle}{\langle\varphi_{1i}^2\rangle - \langle\varphi_{1i}\rangle^2} \tag{3.4}$$

3.5 Stochastic Synchronization in simple PRCs : Type 1 and Type 2

To plot the output correlation R between two uncoupled oscillators as a function of the input correlation r for the Type 1 and type 2 PRCs , I first computed the output correlation using the numerical method described above and verified the result with the analytical results obtained by FPE solutions described in Galan et al. [1] After computing the SDE solutions for two independent oscillators, starting with different initial phases, subjected to correlated noise signals for long enough duration , I omitted the transient and picked the later part of the series where the steady state of probability could be approximated and then output correlation was computed using Eq.(3.4). The duration for stimulation T was fixed upon to 1e4 s and the time step δt was 1e-3. The output correlation thus obtained for a given input correlation varied upto some extent with every simulation. This could be attributed to the limitation in the amount of data that could be generated to tend the results toward the theoretically accurate values. To overcome this component of error I ran 500 iterations of computation for every value of input correlation and took the final result to be mean of all the iterations. The output correlation vs input correlation plot was consistent with the results obtained by Galan et al. [1]

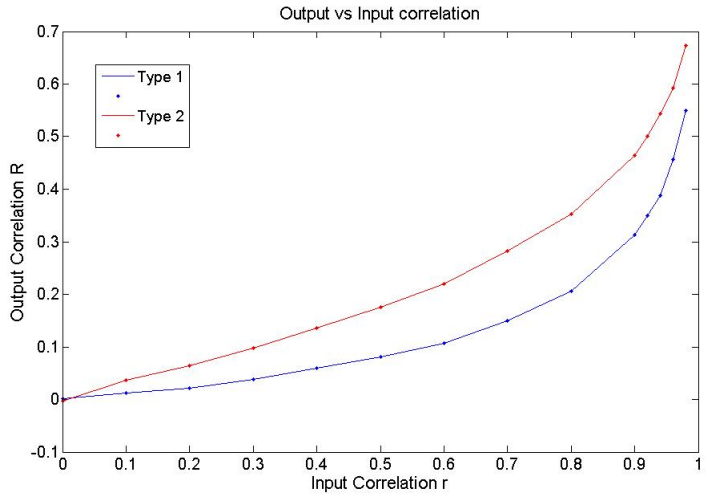


Figure 3.7: The output vs input correlation plot for Type 1 and Type 2 oscillators is showing increasing trend which demonstrates the phenomenon of stochastic synchronization. The results are mean of 500 iterative computations. Time duration of simulation T = 2e4, Time step δt = 1e-3.

Now the normalized PRCs Type 1 and Type 2 are some arbitrary PRCs which represent two basic qualitative types of PRCs. In biological oscillators there is no restriction on the PRCs to be normalized. These two PRC types themselves can have amplitudes of a different order. What happens when a system is so robust to have PRCs with very small amplitudes, say 1/10 of that of Type 1 and Type 2 PRCs, was a naturally arising question. This was also important because major part of the HR PRC and most of the other burster PRCs for that matter, have very low amplitude except for the spikes in between. Thus I computed the output correlation plot for the Type 1 and Type 2 PRCs multiplied by a factor of 0.1. The phenomenon of stochastic synchronisation was observed here as well. Though the steady state in this case is attained after much longer duration (T = 2e5).

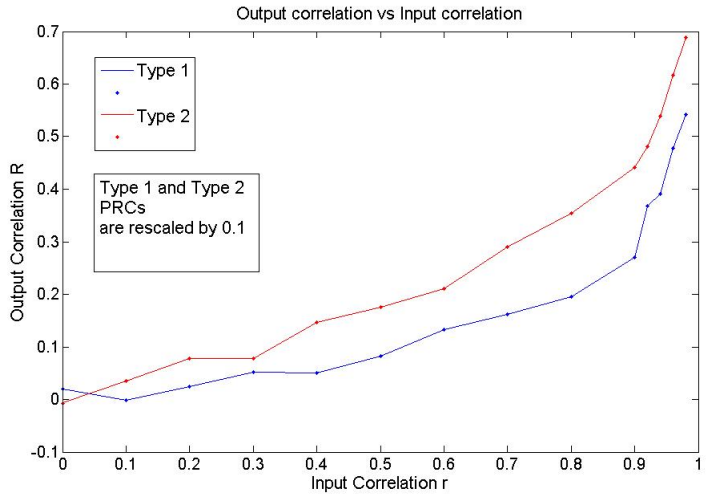


Figure 3.8: The Output correlation plot for small Type1 and Type 2 PRCs (multiplied by 0.1) shows a very similar trend to that of original PRCs. This shows that stochastic synchronization is prevalent even for small PRCs. Time duration of simulation T = 2e5, Time step δt = 1e-3

The considerably high values of output correlations for very small PRCs such as that in the above result show that even very robust oscillators can be synchronized by correlated noises. Having obtained these results it was now appropriate to study this process in the HR bursting oscillators.

3.6 Output Correlation for the Hindmarsh Rose PRC

The main focus of this study was to investigate the phenomenon of stochastic synchronization in bursting oscillators taking HR burster as a model. The numerical tools for this were firmly set up and tested uptill this point. Therefore after verifying the stochastic synchronization for the simple PRCs above I applied the same numerical method to get the output correlation for the Hindmarsh Rose oscillators using the PRC which was already computed and confirmed. Time step δt was set as 1e-3 and time duration for simulation T was kept 2e5 (10 times more than Type 1 and 2). Here again 500 results were obtained and averaged. The output correlation plot obtained is given below.

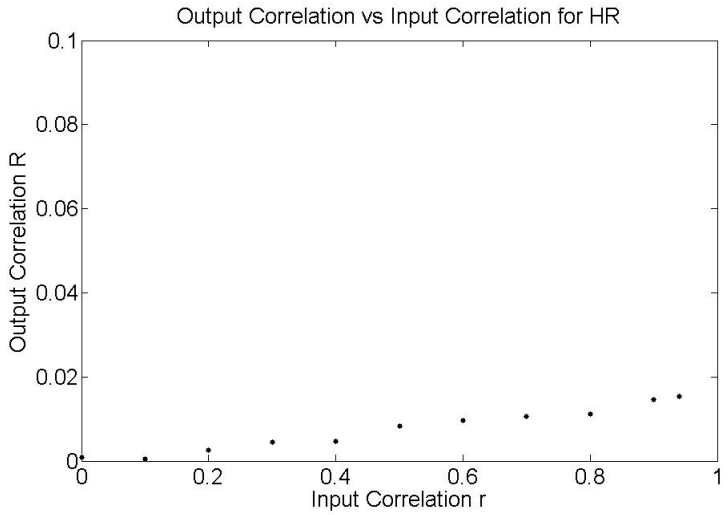


Figure 3.9: The output correlation values for HR are very low even for higher values of input correlation. They show an increasing trend with the input correlation though.

The results obtained for Hindmarsh Rose oscillators were rather dissapointing. The output correlations could not be considered to indicate any synchronization even for input correlation as high as 0.94. It is possible that there exists a steady state of synchronization where a meaningful output correlation is exhibited. If that is the case then few of the possible numerical factors could be as follows:

1. **Steady state was not achieved** : Suppose the HR oscillator is so robust so as to reach to its steady state, where probability $P(\psi_1, \psi_2)$ becomes constant, after very long duration. In this condition there is a possibility that the duration upto which these simulations were run was not sufficient to achieve the steady state of the oscillator system which could possibly contain synchrony. Now this could be due to the fact that a major portion of HR PRC (Fig. (2.4)), in which the system remains for most of the time is of low amplitude. The spikes which are of the order of 1 are very thin and system falls in those region for very less time relatively. Interestingly, the order of amplitude in the low amplitude regions is same as that in the small Type 1 and Type 2 PRCs used in the last section. But those PRCs showed remarkable synchronization inspite of being so robust. Moreover, the robustness of the HR system was very low towards noise as demonstrated clearly by the CV values calculated above (Fig.(3.5)).

2. **The time resolution is low** : The results obtained were based on the solutions of the SDE using the Euler Maruyama method. The order of convergence of this method is quite high and hence requires the time step δt to be as small as possible to

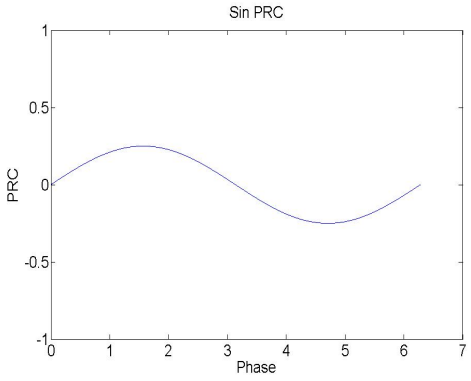
minimize the error. The time step used here was 1e-3 which possibly is not sufficient to generate solutions accurate enough to capture the synchronization even if it is there.

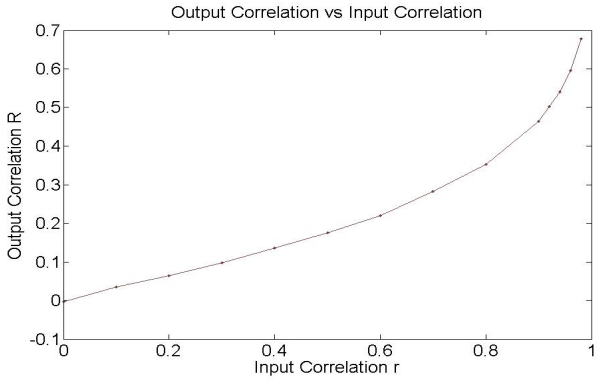
Supposing these two factors do not play such an important role here, and there is no process of stochastic synchronization in the HR bursting oscillators. Then the question that arose here was that what were the factors that accounted for such low output correlation values in the HR oscillator. Qualitatively, the only characteristic which differentiated it with other simple PRCs was the presence of spikes or sharp pulses. This led to a hypothesis that **the spikes in the PRC inhibits the stochastic synchronization**. It is quite possible that the spikes in HR PRC, which are responsible of the high degree of disturbance in the oscillatory cycle as apparent in Fig.(3.6) and also indicated by the CV values, are also the reason for disturbing the synchrony of the oscilltors. The spikes are thus the destructive elements for the stochastic synchronization process even though they are very thin.

Now, this hypothesis needed to be further investigated and analyzed. This led me to follow a new direction of study altogether which aimed at understanding the effect of a spike or a pulse incorporated in an otherwise smooth PRC, on the stochastic synchronization.

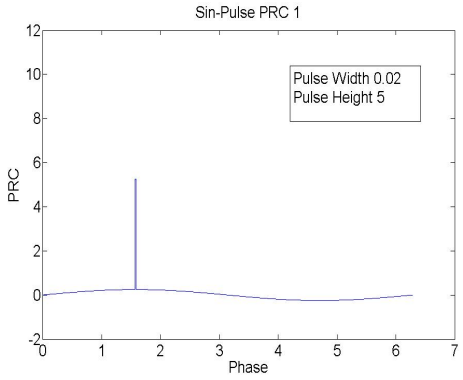
3.7 Output Correlations for some constructed PRCs with pulses

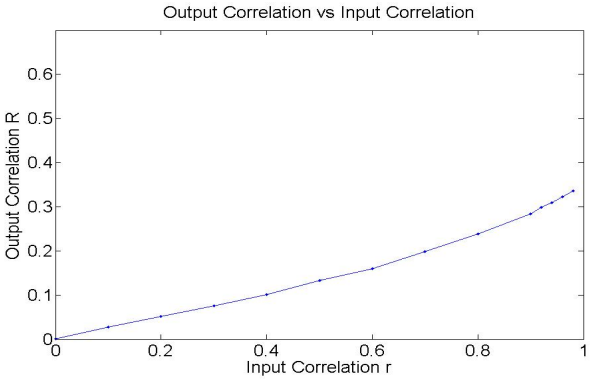
HR PRC can be qualitatively described as comprising of sharp pulses added on an otherwise smooth curve. The aim is to study the effect of such sharp pulses in a PRC with respect to the synchronization or precisely the output correlation. I took the *sine* curve of Type 2 oscillator as a base and constructed different PRCs by incorporating pulses of different widths and heights. The *sine* curve was normalized like the one in Type 2 PRC and the total area under the pulse was same for all the different pulses. The ratio of areas under the pulse to the area under the *sine* curve was 1:10. The output correlation plots of each of the composite PRCs are given below.



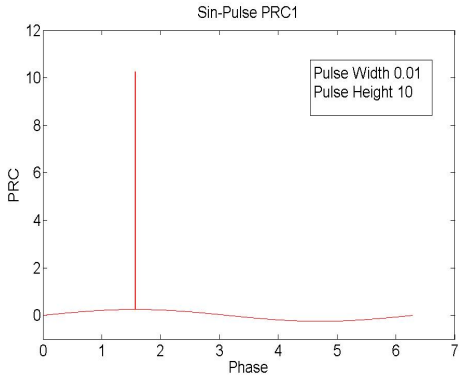


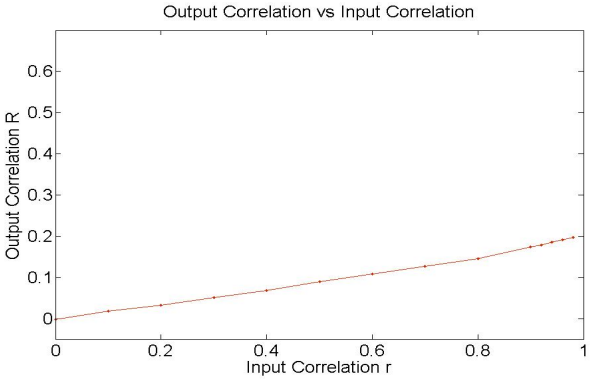
Sin PRC, no pulse



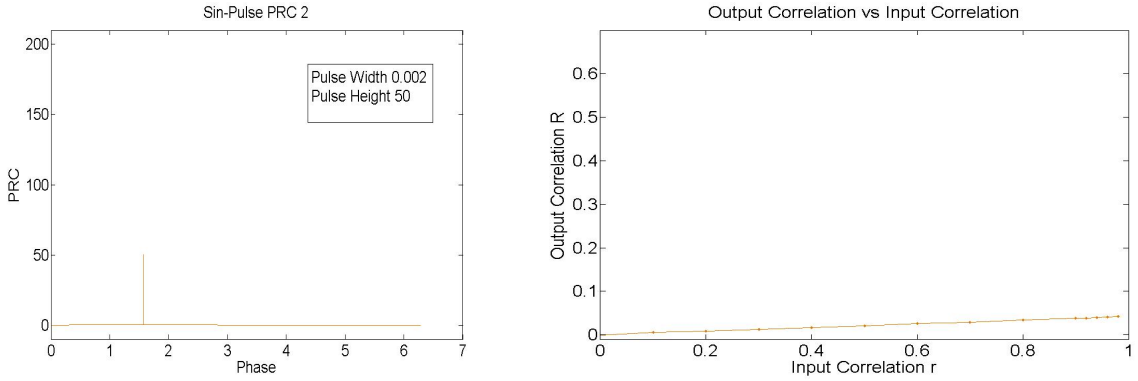


Pulse height = 5
Pulse width=0.02

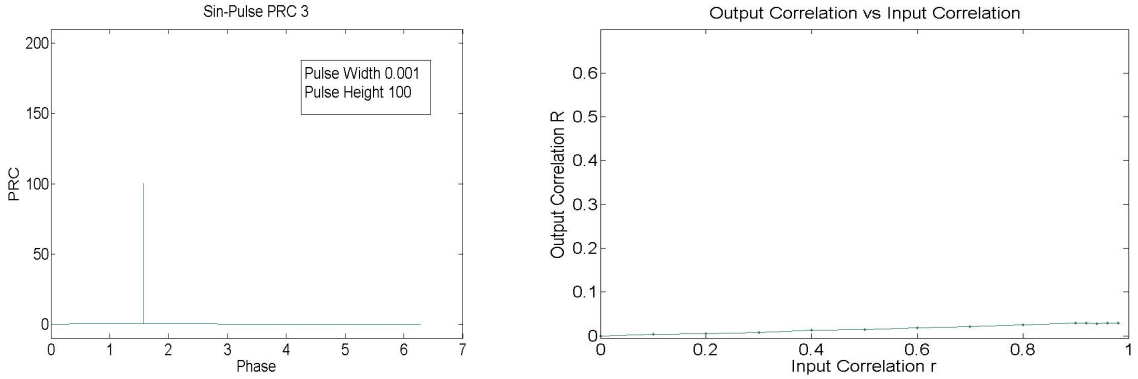




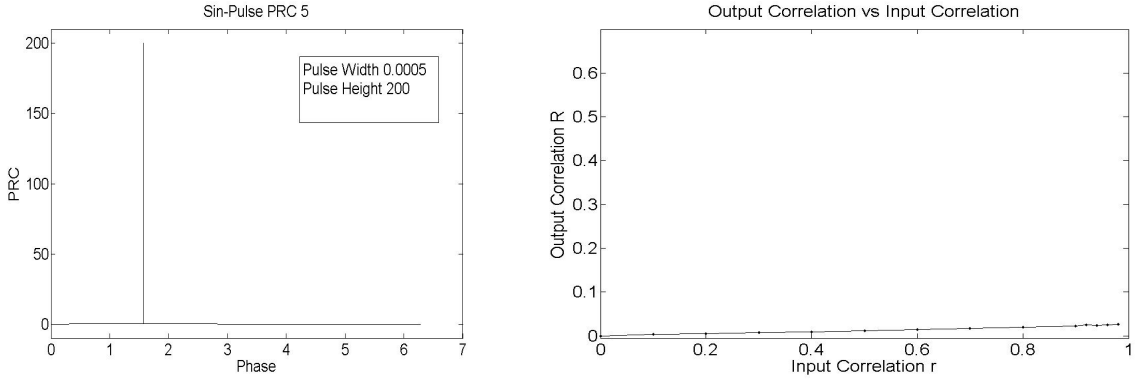
Pulse height = 10
Pulse width=0.01



Pulse height = 50
Pulse width=0.002



Pulse height = 50
Pulse width=0.002



Pulse height = 50
Pulse width=0.002

Figure 3.10: Different PRCs along with their output correlation plots are displayed. Composite PRCs of sine curve (normalized) with the sharp pulses were constructed to study the affect of pulses on the stochastic synchronization. Area under the pulses for all these PRCs are same. The ratios of areas under pulse to the area under sine curve is 1:10 for every PRC. We can see in the output correlation plots that the output correlation is reduced with increase in the pulse height even though the thickness of the pulse is decreasing.

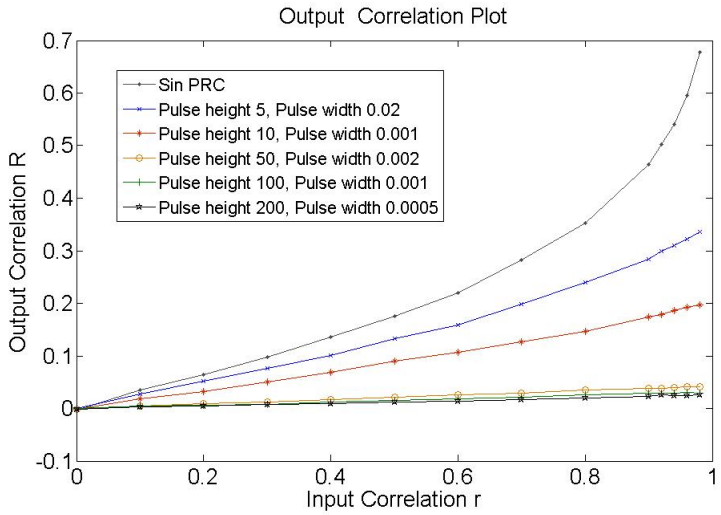


Figure 3.11: All the results of Fig.(3.10) are put together. The trend of output correlation with the increase in height of the pulse can be more clearly seen here. Area under the pulse for all the PRCs were same. The output correlation reduces with increase in pulse amplitude.

In the above results we see that though the total area under the pulse is same, the effect of pulse in the PRCs on the output correlation is governed by the height or amplitude of the pulse. For higher values of pulse amplitude, the output correlation is almost destroyed inspite of their very thin width (0.0005 for pulse of height 200). In case of very thin pulses, the probability of the system to be found within its range of phase is very low and the time spent by the system in that region is almost negligible. Still we can see such a strong effect of these pulses resulting in the destruction of synchronization. Pulse amplitude being a crucial parameter here, the output correlation is not that much sensitively dependent on the width of the pulse. This can be seen the results given below, in which, output correlations are compared for different width pulses of same height added to the sine curve.

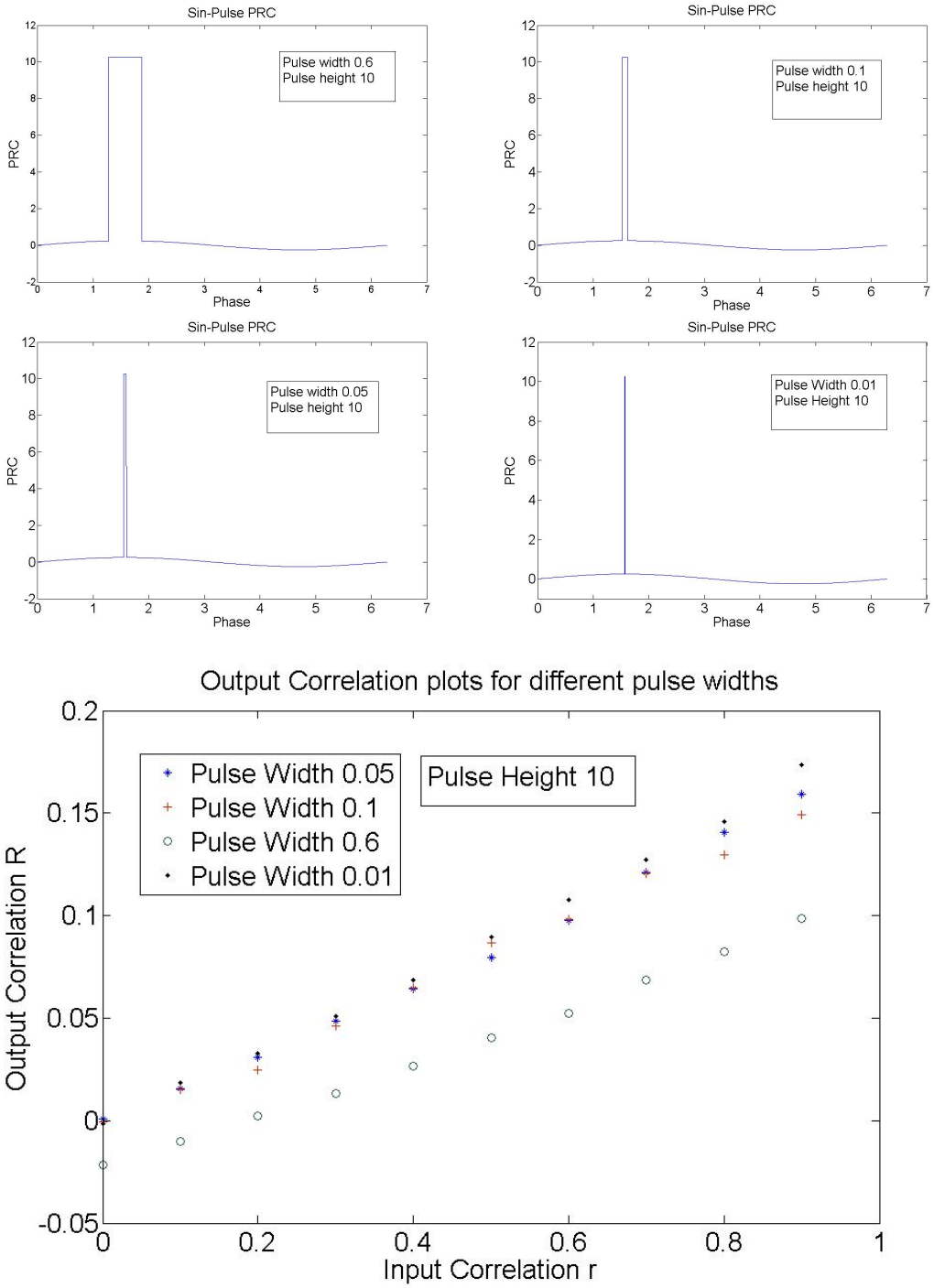


Figure 3.12: Output correlations are compared here for sin-pulse PRCs with different pulse widths and equal pulse amplitudes. Except the PRC of pulse width 0.6, all the other are not much varying. This shows that the synchronization is not that much sensitive to pulse width as it is to pulse height.

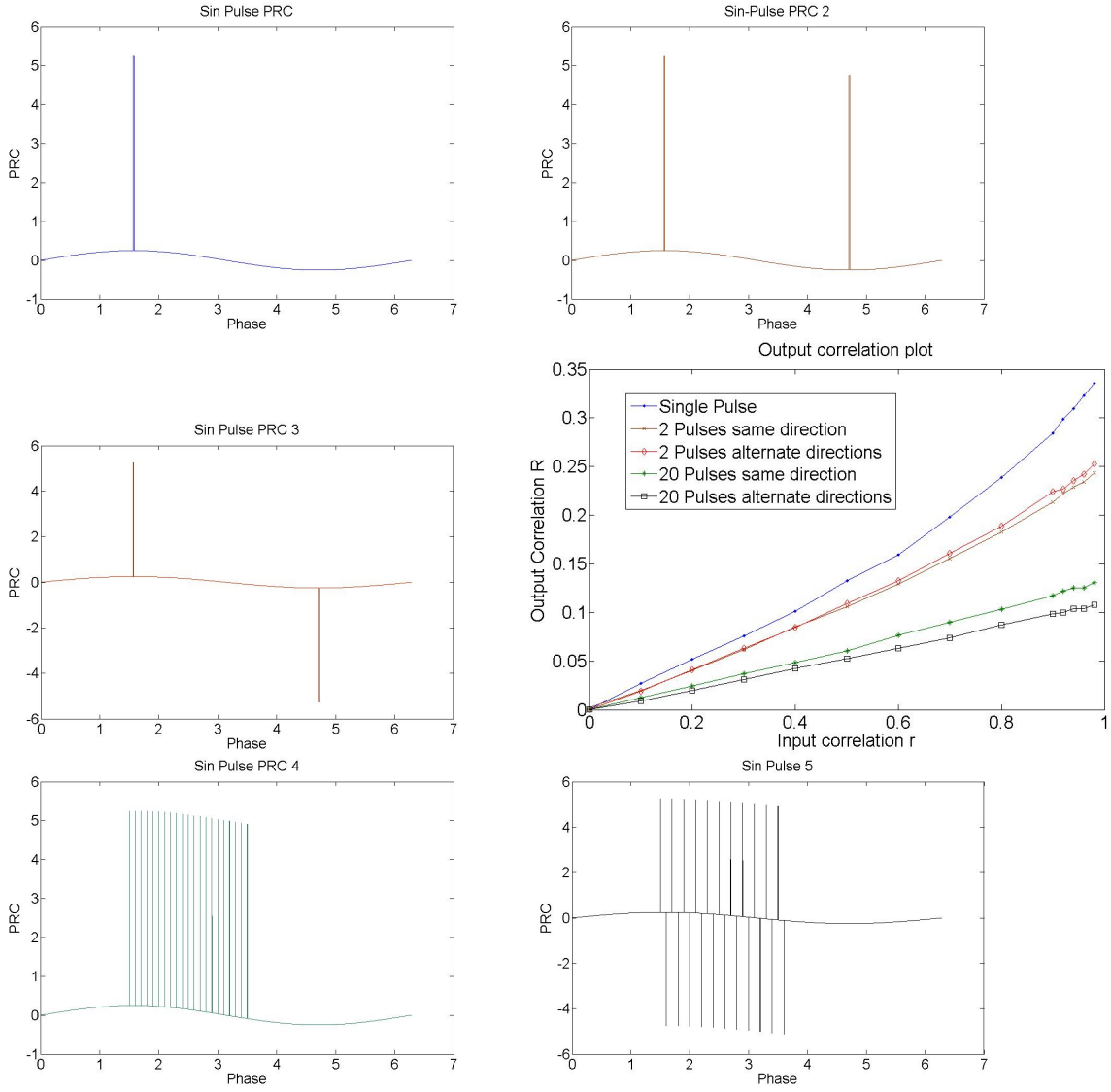


Figure 3.13: The effect of multiple pulses rather than a single pulse in the PRC is displayed in these results. The pulse height of all the pulses is 5 and the widths of each pulses are set so as to make total absolute area under the pulses in all PRCs 0.1. Here we see that the synchronization is further destroyed as the number of pulses increase. Another very important fact arising was that there was not much difference between the PRCs with pulse in the same direction and those with pulses in alternate directions. This means positive phase response is equivalent to negative phase response in destroying the synchronization as far as their magnitudes are large.

The above results support the hypothesis about the role of pulses in PRCs as an opposing force to the stochastic synchronization.

3.8 Conclusive Remarks

We have seen so far that stochastic synchronization occurs in many oscillators having smooth PRCs like Type 1 and Type 2 which I verified using the above described numerical tools and methods. Then I tried to investigate this phenomena for Hindmarsh Rose bursting oscillators. I found that the synchronization is not displayed in the HR oscillators. This led me to investigate about the characteristics of the HR PRC responsible for the destruction of synchronization since major portion of the PRC was smooth, resembling Type 1 and 2 PRCs. I hypothesized that the sharp pulses of very low thickness present in the HR PRC could be one of the factors could be responsible for this lac of synchronization. I tried to examine the effect of sharp pulses added to sine curves which resemble the HR PRC qualitatively. I came up with interesting results asserting that pulses dramatically reduced the output correlation even if the thickness was as low as 0.0005. Pulse amplitude seemed to be more important factor than the thickness. At last I saw that the number of pulses was another important factor reducing the synchronization further. Even if the total area under the pulses is equal, more number of pulses would reduce the output correlation remarkably.

Chapter 4

Discussion and Conclusion

This project was intended to be a numerical investigation of stochastic synchronization in a bursting oscillator. I chose Hindmarsh Rose model as an appropriate model for a simple burster. I explored some numerical methods to compute its PRC and verified their validity. For using those methods, I found out the proper magnitudes of the perturbations required to get accurate results. This was a very fruitful outcome of that exercise since it provided the correct sizes of the perturbations used in the computation of PRCs of HR. If these methods are to be used to compute other PRCs, the direct perturbation method gave a way to find out the correct order of perturbations ie. by plotting the phase response with the perturbations and check for the linearity. Further on, I tried to find the PRC for islet oscillators but did not get reliable results using any of the methods.

Henceforth after getting assured about the HR PRC, I moved on to set up SDE solving codes for simulating the oscillations subjected to noise. To verify their accuracy, I computed the Coefficients of Variation for Type 1 and Type 2 PRCs using the SDE solutions and compared them with the analytically obtained values given by Fokker-Planck equations. The values were in agreement. I reproduced the output correlation results for simple Type 1 and Type 2 PRCs as reported in Galan et. al [1]. The output correlation being the index of stochastic synchronization, was computed for HR oscillators. The resulting values were too low to be meaningful and hence there was not any direct numerical way to prove the existence or non existence of stochastic synchronization in HR model. I hypothesized the reason for this loss of synchronization in HR oscillators to be the thin pulses in the PRC. For investigating it further, I constructed some PRCs with a sine curve and the pulses and computed their output correlations.

The results were quite interesting and meaningful. First thing they suggested was that a pulse was a major opposing factor for stochastic synchronization. Their opposing force was determined by the amplitude of the pulse. The thickness of the pulse is almost insignificant within some range. Pulses with very low thickness mean that the time spent

by the system in the pulse region is negligibly low. However if the amplitude of the pulse is large, the effect would be far from negligible and the output correlation drastically falls down. That means that if an oscillator is unstable even for a very tiny amount of time during its cycle, it would hamper the stochastic synchronization. The other important factor was the number of pulses in the PRC. A single pulse is not as effective in reducing the output correlation as two pulses of the same amplitude even if the total area under the pulses are equal to that one pulse. This negative effect increases with the number of pulses. Thus by far we know that the unstable holes in the cycle of any oscillator (represented by the pulses in the PRC) are responsible for opposing the stochastic synchronization. More the number of holes and higher the degree of instability, greater is the loss of stochastic synchronization amongst the oscillators.

Future Work

The results in this study can be strengthened further by the following ways :

- The numerical results can be fine tuned further by using smaller time steps in integration of SDEs and taking larger data by running the simulations longer. Both of them require high memory capacities. Moreover, the EM algorithm used here has high order of convergence and low accuracy. Using other SDE solving algorithms like the Milstein algorithm will improve the results.
- The numerical results can be further verified by comparing with the solutions from Fokker-Planck equations (FPE). The FPEs give the coefficient of variation and the phase probability distribution for a two oscillator system. Though, the finite element method (FEM) used to solve FPEs also requires large memory space for PRCs with sharp pulses.

Further on, since the pulses and their amplitudes have emerged as critical factors for stochastic synchronization, this study must be extended to the role of another factor : the fraction of the area under the pulses out of the total area of under the PRC. What is the effect when the smooth parts of the PRC increase in area inspite of the high amplitude pulses ?

Bibliography

[1] R. F. Galán, G. B. Ermentrout, N. N. Urban. Stochastic dynamics of uncoupled neural oscillators: Fokker-Planck studies with the finite element method, Phys. Rev E. 76 (2007).

[2] W.E. Sherwood, J. Guckenheimer. Dissecting the phase response of a model bursting neuron, SIAM J. App. Dyn. Sys. 9(3) (2010).

[3] D.J. Higham . An algorithmic introduction to numerical simulation of stochastic differential equations, SIAM Rev. 43(3) (2001).

[4] E. Brown, J. Moehlis, P. Holmes . On the phase reduction and response dynamics of neural oscillator populations, Neural Comput. 16 (2004) .

[5] R. Bertram, L. Satin, M. Zhang, P. Smolen, A. Sherman. Calcium and glycolysis mediate multiple bursting modes in pancreatic islets, Biophy. J. 87 (2004).

[6] G.B. Ermentrout. Type I membranes, phase resetting curves and synchrony, Neural Comput. 8 (5) (1996).

[7] G.B. Ermentrout. Simulating, analyzing, and animating dynamical systems: A guide to XPPAUT for researchers and students, SIAM books, Philadelphia (2002).

[8] C.W. Gardiner. Handbook of stochastic methods for physics, chemistry, and the natural sciences, Springer, Berlin (2004).

[9] H. Risken, The Fokker-Planck equation, Springer, Berlin (1996).

[10] D. Fyfe, A. Weiser, I. Bernstein, S. Eisenstat, M. Schultz. A finite element solution of a reduced Fokker-Planck equation, J. of Comput. Phys. 42(327) (1981).

[11] J. L. Hindmarsh, R.M. Rose. A model of neuronal bursting using three coupled first order differential equations, Proc. R. Soc. B. 221 (1984).

[12] G.E.P. Box, M.E. Muller. A Note on the generation of random normal deviates, Ann. of Math. Stat. 29(2) (1958).

[13] J. Guckenheimer, P. Holmes, 1990. Nonlinear oscillations, dynamical systems and bifurcations of vector fields, Springer, Berlin (1983).

[14] R.F. Galán, N. Fourcaud-Trocmé, G.B. Ermentrout, N.N. Urban. Correlation-induced synchronization of oscillations in olfactory bulb neurons, J. Neurosci., 26(14) (2006).

[15] D. García-Álvarez, A. Bahraminasab, A. Stefanovska P.V.E. McClintock . Competition between noise and coupling in the induction of synchronization, Europhys. Lett., 88 (2009).

[16] L. Cheng, G.B. Ermentrout. Synchronization dynamics of two coupled neural oscillators receiving shared and unshared noisy stimuli, J. Comput. Neurosci. 26 (2009).

[17] E.M. Izhikevich. Neural excitability, spiking and bursting, Int. J. Bifur. Chaos, 10(6) (2000).

[18] E.M. Izhikevich. Dynamical systems in neuroscience: The geometry of excitability and bursting, MIT Press, Cambridge (2007).

[19] Y. Kuramoto. Chemical oscillations, waves, and turbulence, Springer-Verlag, Berlin and New York (1984).

[20] H. Nakao, K. Arai, and Y. Kawamura. Noise-induced synchronization and clustering in ensembles of uncoupled limit-cycle oscillators, Phys. Rev. Lett. 98 (2007).

[21] A. Abouzeid and G.B. Ermentrout. Type-II phase resetting curve is optimal for stochastic synchrony, Phys. Rev. E 80 (2009).

[22] R.F. Galán, G.B. Ermentrout and N.N. Urban. Reliability and stochastic synchronization in type I vs. type II neural oscillators, Neurocomput. 70 (2007).

[23] G.B. Ermentrout, R.F. Galan, N.N. Urban. Reliability, synchrony and noise, Trends Neurosci. 31(8) (2008).

[24] B. Øksendal. Stochastic differential equations, Springer-Verlag, New York (2003).

[25] R.F. Galan, G.B. Ermentrout, N.N. Urban. Ecient estimation of phase-resetting curves in real neurons and its significance for neural-network modeling, Phys. Rev. Lett, 94 (2005).

[26] G.B. Ermentrout, N. Kopell. Multiple pulse interactions and averaging in systems of coupled neural oscillators, J. Math. Biol., 29 (3) (1991).

[27] F. Hoppensteadt, Eugene M. Izhikevich. Weakly connected neural networks, Springer-Verlag, New York (1997).

[28] J. Guckenheimer. Isochrons and phaseless sets, J. Math. Bio., 1 (3) (1975).

[29] W. Govaerts, B. Sautois. Computation of the phase response curve: A direct numerical approach, Neur. Comput., 18 (2006).

[30] W. B. J. Zimmerman. Multiphysics modelling with finite element methods, World Scientific (2006).

Appendix A : Codes

In this appendix I present some of the codes which I used for calculating PRCs, CV and the output correlations. I used many variants of these codes for different PRCs.

Computing HR PRC using Direct Perturbation method in Matlab

Matlab file for differential equations of HR (hr_ode.m)

```
function dp = hr_ode(t,p)
    a=3;
    b=5;
    I=2;
    s=4;
    x=-8/5;
    r=0.001;
    %repar=430.672;
    repar=1;
    dp= zeros(3,1); % a column vector
    dp(1) = repar*(p(2)+ a*p(1)^2 - p(1)^3-p(3)+I);
    dp(2) = repar*(1 - b*p(1)^2 - p(2));
    dp(3) = repar*(r*(s*(p(1)-x)-p(3)));
```

Matlab file for applying direct perturbation method

```
clc
    clear all
    format long e
    deltav=0.05; % setting the delta v
    options = odeset('RelTol',1e-9,'AbsTol',1e-9);
    p_in = [1.769221998732861e+00 -4.036343739735699e+00 1.818313363575842e+00];
```

```
[T1,p1] = ode23s(@hr_ode,[0,1.1],[p_in],options);
v1=p1(:,1);
n1=p1(:,2);
h1=p1(:,3);
maxima=find(v1==max(v1(1000:length(v1))));
period=T1(maxima); %time period calculated
tau=[0.01:0.01:period-0.01];
len=length(tau);
PRC=zeros(1,len);
iprc=zeros(1,len);
for i=1:len
[T1,p1] = ode23s(@hr_ode,[0,tau(i)],p_in,options);
v1=p1(:,1);
n1=p1(:,2);
h1=p1(:,3);
last=length(p1);
%integrator stopped and V changed
[T2,p2] = ode23s(@hr_ode,[tau(i),1.1], [v1(last)+deltav n1(last) h1(last)],options);
v2=p2(:,1);
n2=p2(:,2);
h2=p2(:,3);
maxima=find(v2==max(v2));
tprime=T2(maxima);
PRC(i)=1-tprime/period;
iprc(i)=PRC(i)/deltav
end
plot(tau,iprc,'.')
```

Calculation of CV

/* This is a C code which I used to compute coefficient of variation for Type 1 and Type 2 neurons. The white noise is generated, incorporated in the SDE which is solved using EM algorithm and then the solution is used to calculate CV. Any other PRC can be incorporated in the same code*/

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
```

```
double phi1[50000001],phi2[50000001],wrap_phi[50000001];
double T=50000; double dt=0.001;
unsigned int N = 10000000;
//Generating normal random numbers using Box-Muller algorithm
double normal()
{
double fac,rsq,v1,v2,y;
do
{
v1=2*(double)rand()/RAND_MAX -1;
v2=2*(double)rand()/RAND_MAX -1;
rsq=v1*v1+v2*v2;
}while(rsq>=1 || rsq==0);
fac=sqrt(-2*log(rsq)/rsq);
y=v1*fac;
return(y);
}
//wraps generated time series, finds peaks, calculates CV
double CV()
{
double phi ;
unsigned int i,j,k;
unsigned int pk_cnt=0;
double mean,meansq,tp[10000],peak[10000],cv;
for(i=0;i<N;i=i+1)
{
phi=phi1[i];
while(phi>=2*M_PI) phi=phi-2*M_PI;
wrap_phi[i]=phi; //wrapping
if(i>0)
{
if(wrap_phi[i-1]-wrap_phi[i]>6)
{
peak[pk_cnt]=i*dt;
pk_cnt=pk_cnt+1; //peak finding and counting
}
}
```

```
}
for(j=0;j<=pk_cnt-2;j=j+1){tp[j]=peak[j+1]-peak[j];} //differences btw peaks
for(k=0;k<=pk_cnt-2;k=k+1)
{
mean=mean+ tp[k]/((double)pk_cnt-1);
meansq=meansq + tp[k]*tp[k]/((double)pk_cnt-1); //mean and mean sqare
}
printf("%u peaks \n",pk_cnt);
cv=sqrt(meansq-mean*mean)/mean; //calculating CV
return(cv);
}
// main programme
int main()
{
double dw,dw1,dw2,q,sigma,sigma1,sigma2,omega1,omega2;
double eta1,eta2,n,m;
unsigned int i;
q=0.8; // input correlation
//printf("sigma = ");
//scanf("%f",&sigma);
//printf("%f\n",sigma);
sigma=0;
sigma1=sigma;
sigma2=sigma; // amplitude of noise
omega1=1;
omega2=1; // average angular frequency
phi1[0]=0; //initial conditions
phi2[0]=3;
n=1/(2*M_PI); //normalisation factors
m=0.25;
srand(3413); //seed for random number generation
// EM algorithm
for(i=0;i<N;i=i+1)
{
dw=sqrt(dt)*normal();
dw1=sqrt(dt)*normal(); // generates noise
dw2=sqrt(dt)*normal();
```

```
eta1=(q*dw+sqrt(1-q*q)*dw1)*sigma1;
eta2=(q*dw+sqrt(1-q*q)*dw2)*sigma2; //generates correlated noise
phi1[i+1]=phi1[i]+omega1*dt + n*(1-cos(phi1[i]))*eta1 ;
phi2[i+1]=phi2[i]+omega2*dt + n*(1-cos(phi2[i]))*eta2 ;
//type 1 PRC
//phi1[i+1]=phi1[i]+omega1*dt - m*sin(phi1[i])*eta1 ;
//phi2[i+1]=phi2[i]+omega1*dt - m*sin(phi2[i])*eta1 ;
//type 2 PRC
}
printf("CV = %.5f",CV());
return(0);
}
```

Computaion of Output Correlation

/* This code I used to compute the output correlations for the HR PRC for different input correlations. It imports values for HR PRC from the file "hr_prc.txt", which has 1e5 data points. It computes output correlations for input correlation values 0 to 9.8.

```
*/
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<iostream>
#include<fstream>
using namespace std;
double phi1[20000001],phi2[20000001];
double T=20000; double dt=0.001;
unsigned int N = 20000001;
double ph2[10000][10],ph[100000],v[100000];
//Generate normal random numbers
double normal()
{
double fac,rsq,v1,v2,y;
do
{
v1=2*(double)rand()/RAND_MAX -1;
v2=2*(double)rand()/RAND_MAX -1;
```

```
rsq=v1*v1+v2*v2;
}while(rsq>=1 || rsq==0);
fac=sqrt(-2*log(rsq)/rsq);
y=v1*fac;
return(y);
}
//Phase response function
double prc(double phase)
{
int p;
double prc_v;
p=floor(phase*10000);
int j;
for(j=0;j<10;j++)
{
if ((phase>=ph2[p][j]-0.000005)&&(phase<ph2[p][j]+0.000005))
{
prc_v=v[p*10+j];
break;
}
}
return(prc_v);
}
// main programme
int main()
{
double dw,dw1,dw2,q,sigma,sigma1,sigma2,omega1,omega2;
double eta1,eta2,n,m;
double mean1,mean2,mean_sq_1,cross_mean,iterat[500],r_in[15],R[15],mn_sq[15],std[15];
unsigned int i,j,k,loop,loop1,recur;
//loading the prc file
ifstream myfile;
myfile.open("hr_prc.txt");
unsigned size = 400000;
double a[size];
for (int k = 0;k < size;k++)
{
```

```
myfile >> a[k];
}
double phase;
for(int i=0;i<100000;i++)
{
ph[i]=a[i*4];
v[i]=a[i*4+1];
}
for(int l1=0;l1<10000;l1++)
{
for(int l2=0;l2<10;l2++)
{
ph2[l1][l2]=ph[l1*10+l2];
}
}
//loading complete
sigma=1;
sigma1=sigma;
sigma2=sigma; // amplitude of noise
omega1=1;
omega2=1; // average angular frequency
srand(413); //seed for random number generation
for(loop1=0;loop1<15;loop1++)
{
if(loop1<=9)
{
r_in[loop1]=(double)loop1/10;
}
else
{
r_in[loop1]=0.9 + ((double)loop1-9)/50;
}
R[loop1]=0;
mn_sq[loop1]=0;
q=sqrt(r_in[loop1]);
for(loop=0;loop<500;loop++)
{
```

```
phi1[0]=0; //initial conditions
phi2[0]=0.5;
for (recur=0;recur<20;recur++)
{
if(recur>0)
{
phi1[0]=phi1[N-1];
phi2[0]=phi2[N-1];
}
for(i=0;i<N;i=i+1)
{
dw1=sqrt(dt)*normal(); // generates noise
dw2=sqrt(dt)*normal();
eta1=dw1*sigma1;
eta2=(q*q*dw1+sqrt(1-q*q*q*q)*dw2)*sigma2; //generates correlated noise
phi1[i+1]=phi1[i] + omega1*dt + prc(phi1[i])*eta1 ;
phi2[i+1]=phi2[i] + omega2*dt + prc(phi2[i])*eta2 ;
if(phi1[i+1]>=1)
phi1[i+1]=phi1[i+1]-1;
if(phi2[i+1]>=1)
phi2[i+1]=phi2[i+1]-1;
}
}
mean1=0;
mean2=0;
cross_mean=0;
mean_sq_1=0;
for(j=N-1-8000000;j<N;j++)
{
cross_mean=cross_mean+phi1[j]*phi2[j]/8000000;
mean1=mean1+phi1[j]/8000000;
mean2=mean2+phi2[j]/8000000;
mean_sq_1=mean_sq_1 + phi1[j]*phi1[j]/8000000;
}
iterat[loop]=(cross_mean-mean1*mean2)/(mean_sq_1-mean1*mean1); //ouput cor-
relation
R[loop1]=R[loop1]+iterat[loop]/500;
```

```
mn_sq[loop1]=mn_sq[loop1]+iterat[loop]*iterat[loop]/500;
printf("%5f ",iterat[loop]);
}
std[loop1] = sqrt(mn_sq[loop1]-R[loop1]*R[loop1]);
printf("\n%5f %5f %5f\n ",r_in[loop1],R[loop1],std[loop1]);
}
for(k=0;k<15;k++)
{
printf("\n %5f %5f %5f \n",r_in[k],R[k],std[k]);
}
return(0);
}
```