



Technical University of Crete
Electronic & Computer
Engineering Department

Diploma Thesis

*“Adaptive regulation of PKC for
personalized medicine”*

Authors: Parthenidis Euaggelos
Savvas Stavros

Committee : Prof. M. Christodoulou
Prof. P. Stavroulakis
Assoc. Prof. E. Georgiou

Acknowledgements

We would like to express our thanks to our advisor Professor M. Christodoulou for his support and guidance in the preparation of this thesis.

Table of Contents

Protein Kinase C signaling pathway Simulation	7
A General Overview	7
A Closer approach of a Cell.....	7
Introduction.....	9
The model of PKC signal transduction pathway.....	10
Background of the mathematical formulation.....	12
Simulation of PKC pathway	15
Simulation results and conclusion	57
Background of neural networks.....	58
General.....	58
Identification of dynamical systems using Recurrent High- Order Neural Networks.....	61
The RHONN Model.....	62
Identification of Protein Kinase C using Reccurent High Order Neural Networks	64
From theory to action – The scopes of the model	69
Configuration parameters:	69
Scopes:	69
Direct Adaptive Control	73
Introduction.....	73
Direct adaptive regulation for PKC ($n \neq m$)	73
Design of the system for Direct Adaptive Control.....	74
Simulation Results	79
FINAL CONCLUSIONS & FUTURE WORK	85
Final Conclusions	85
Future Work.....	86
Acknowledgement	87
References.....	87

Protein Kinase C signaling pathway Simulation

A General Overview

In the recent years a tremendous progress has been made in the field of biology. With the development of technological assistance biologists have been able to obtain a better image of the microcosm of a living mammalian and especially the cells which are the basic elements of a living matter.

Cells interact with each other, grow, divide and eventually die and all these functions are regulated by reactions caused by proteins. These reactions are very difficult to describe and the use of Systems Theory is obligatory in order to understand the hidden mechanisms beneath them. This field of science is also known as Systems Biology and it is very prominent in the use of medicine.

A Closer approach of a Cell

In the past two decades the structure and function of genes has met a tremendous progress. DNA's sequence led to unfamiliar abilities in terms of genetic engineer. DNA or deoxyribonucleic acid is a polymer (molecule of similar repeating units which are linked together by a common bonding mechanism) made up of a linear arrangement of subunits known as nucleotides. Genes are composed approximately by 6 to 7 billion sequences of DNA and are the basis of heredity. The genetic information which is at the DNA's chromosome is located at the nucleus of the cell. Cells are therefore the fundamental units of living matter.

On the other hand the creation of proteins, which are responsible for the interpretation of genetic information, is made at the cytoplasm. This diversification reflects the fact that human organism is eukaryotic and it means that the nucleus is divided from the cytoplasm by nucleic membrane.

As said above proteins are responsible for the interpretation of genetic information but this is not their only purpose. Proteins can turn the genes on or off but they can also carry out most of the cells functions. Most of them are enzymes which carry out the reactions responsible for the cell's metabolism the reactions that allow it to process nutrients, to build new cellular material, to grow and to divide. Cells can even receive signals from the outer environment through specially designed proteins also known as receptors (Figure 1). Receptors are proteins that span the membrane, with a site for binding the signalling compound on the outer surface. Binding of the extra cellular signalling compound to the outer surface of the receptor results in an activation of an intracellular protein (the "response regulator"), for example, by phosphorylation.

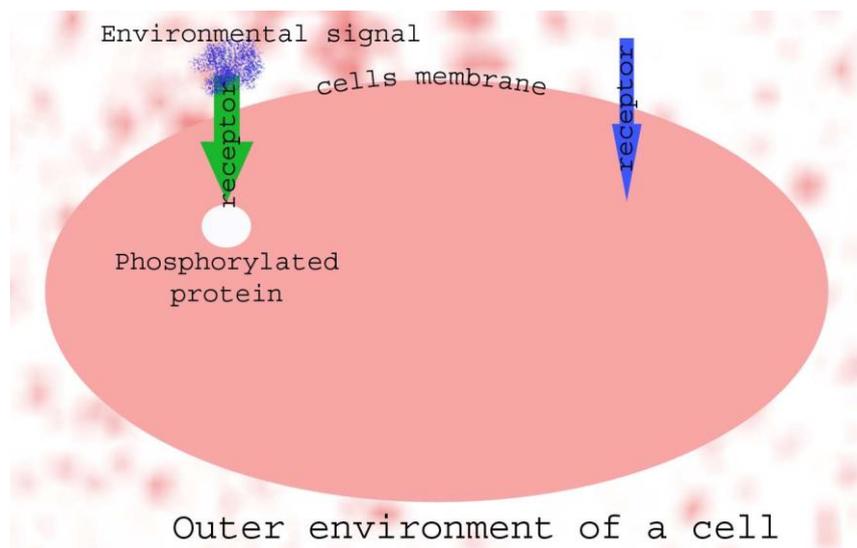


Figure 1

Signal transduction pathways can be represented as sequences of enzyme kinetics reactions which turn a substrate S into a product P via an intermediate complex SE and regulated by an enzyme E . The rate by which the enzyme-substrate complex SE is formed is denoted by k_1 . The complex SE holds two possible outcomes in the next step. It can be dissociated into E and S with a rate constant k_2 or it can further proceed to form a product P with a rate constant k_3 . It is required to express the relations between the rate of catalysis and the change of concentration for the substrate, the enzyme,

the complex, and the product. Figure 2 shows the schematic.

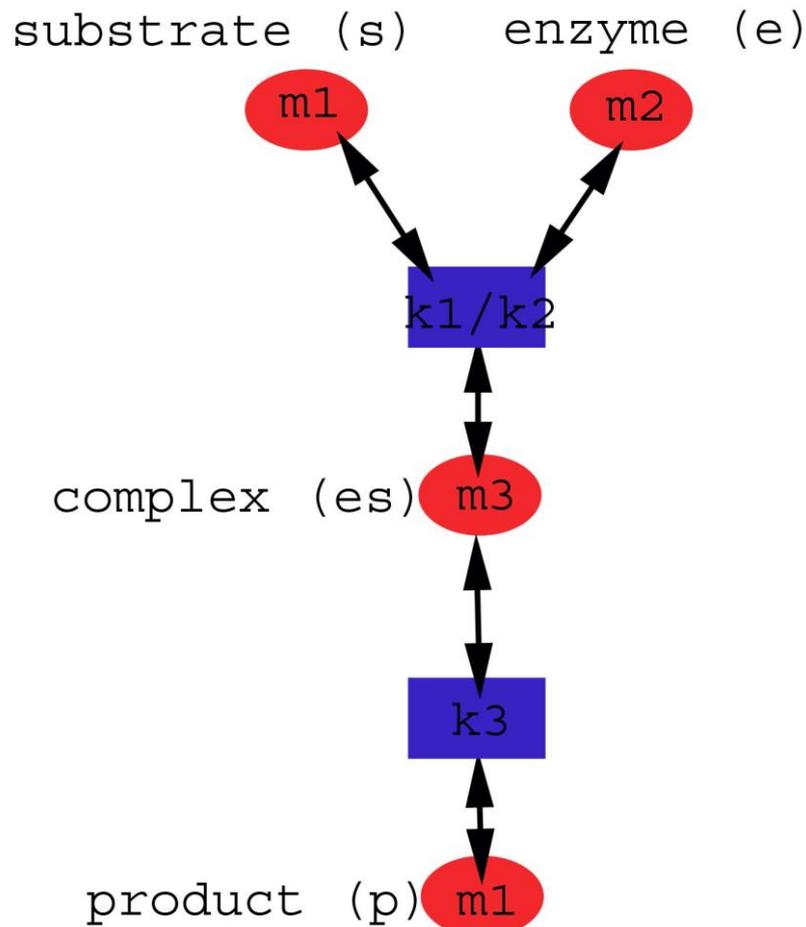


Figure 2

Introduction

From the database of Quantitative Cellular Signaling (<http://doqcs.ncbs.res.in/>) we obtain the model that describes our pathway. The database also contains the model structure (the interactions that play a role in the formulation of the protein), parameter values (as for example rate constant), and the initial concentrations. The activation of the PKC pathway is done by three second messengers, arachidonic acid (AA), diacylglycerol (DAG), and calcium (Ca^{2+}). Using the Matlab/Simulink simulation environment we will study the sensitivity of different types of stimuli, parameter values, and initial concentrations. Simulations have shown that the amplitude of the active PKC concentration is proportional to the time period of Ca^{2+} stimulus. Also

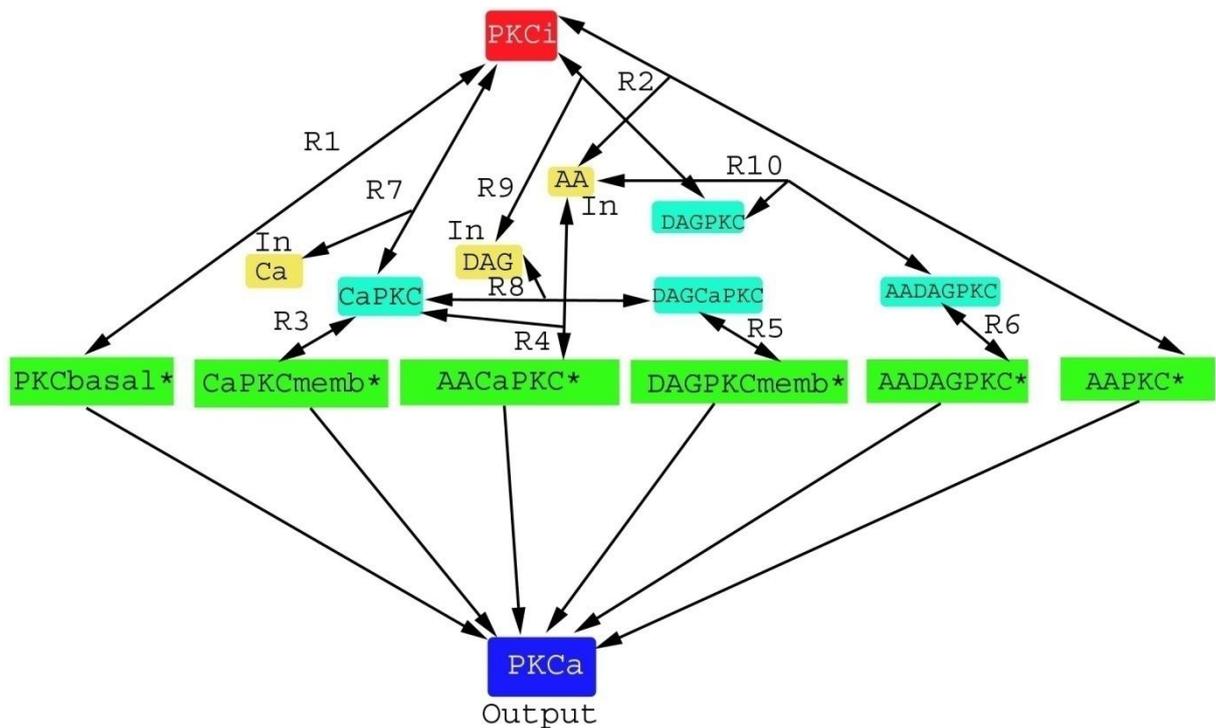
from the simulation we obtain the ascertainties that second messengers AA and DAG also play a role on the active PKC concentration.

The model of PKC signal transduction pathway

As said above in the previous chapter we are going to analyze the pathway developed by Bhalla and Iyengar [3] in order to model the hippocampal neuron. The model describing the PKC pathway is obtained from the database of Quantitative Cellular Signaling (<http://doqcs.ncbs.res.in/>). The database contains explicit chemical reaction kinetic models for signaling pathways, including annotations, information on data sources, and model parameter derivations based on publications. Our model consists of 15 different chemical species. The inputs of the system (shown in figure 1 in yellow) are the three second messengers: Ca^{2+} , AA and DAG. These inputs are used to activate and modulate the system. Also in figure 1 the species in green boxes represent the computational intermediates that are summed in order to study the concentration of active PKC (denoted in blue).

Figure 3: Graphical representation of the PKC model according to DOQCS [3], [8], [19]

Reactants that are green represent the computational intermediates. The inputs of the system Ca^{++} , AA and DAG are in yellow



As we see in figure 1 we have fifteen different interacting chemical species:

- R1 which is the basal activity of PKC transforms Protein Kinase C (inactive) to Protein kinase C (basal level, intermediate) leading to Protein kinase C (active)
- R2 which is Activation of PKC by AA or Arachidonic acid has as an intermediate product AAPKC* or Arachidonic acid-protein kinase C which is membrane-bound and active form of the AAPKC complex
- R3 is the translocation of PKC to membrane. CaPKC or Calcium - protein kinase C has as an intermediate product CaPKCmemb* or Calcium - protein kinase C (on membrane, intermediate) which is finally transformed to Protein kinase C (active)
- R4 is the activation of PKC by AA in presence of Ca and as an intermediate product we have the creation of AACaPKC or Arachidonic acid-Calcium-protein kinase C
- R5 is the activation of DAGCaPKC or diacylglycerol - calcium -protein kinase C with the creation of the intermediate product DAGPKCmemb* or diacylglycerol - protein kinase C

- R6 is the activation of AADAGPKC or arachidonic acid-diacylglycerol-protein kinase C
- R7 is the activation of PKC by Calcium ions leading to the creation of Calcium - protein kinase C (Synergistic activation of PKC by combinations of DAG and Ca as well as AA and Ca and Ca)
- R8 is the activation of PKC by diacylglycerol in presence of Calcium ions leading to the creation of Diacylglycerol - protein kinase C (Synergistic activation of PKC by combinations of DAG and Ca as well as AA and Ca and Ca)
- R9 is the activation of PKC by diacylglycerol
- R10 is the synergistic activation of PKC by diacylglycerol and Arachidonic acid (Membrane translocated form of AADAGPKC complex)

Background of the mathematical formulation

The reactions that are shown in table 2 derive using biochemical reaction kinetics which are based on mass action law .The biochemical reaction kinetics that has A and B as reacting species and C as a product, is described by: $A + B \xrightleftharpoons[k_{-1}]{k_1} C$ (1), where k_1 is the forward and k_{-1} is the backward rate constant. Equation

$$u = \frac{d[C]}{dt} = -\frac{d[A]}{dt} = -\frac{d[B]}{dt} = k_1[A][B] - k_{-1}[C] \quad (2)$$

is an ordinary differential representation of equation 1 where u is the reaction rate and $[A]$, $[B]$ and $[C]$ are concentrations of the chemical species A, B and C (variables in simulations).In the case of a system of differential equations , the differential equation to all species can

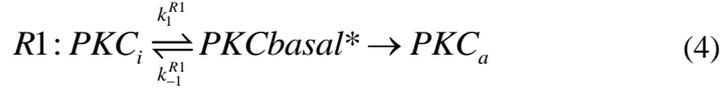
$$\frac{d[C]}{dt} = \sum_{i \in I_{in}} u_{Ri} - \sum_{i \in I_0} u_{Ri} \quad (3)$$

where I_{in} is the set of reaction

indices leading leading to the species C and I_o is the set of reaction indices leading out if the species C. u_{R_i} is the reaction rate in reaction R_i .

Reactions that are represented in Figure 1 are represented in the form of

$A + B \xrightleftharpoons[k_{-1}]{k_1} C$ as follows:



And so by using the ordinary differential equation form we have for the previous equations:

$$U_{R1} = \frac{d[PKC_{basal}^*]R1}{dt} = -\frac{d[PKC_i]R1}{dt} = k_1^{R1}[PKC_i] - k_{-1}^{R1}[PKC_{basal}^*] \quad (14)$$

$$U_{R2} = \frac{d[AAPKC^*]R2}{dt} = -\frac{d[PKC_i]R2}{dt} = -\frac{d[AA]R2}{dt} = k_1^{R2}[PKC_i][AA] - k_{-1}^{R2}[AAPKC^*] \quad (15)$$

$$U_{R3} = \frac{d[CaPKC_{memb}^*]R3}{dt} = -\frac{d[CaPKC]R3}{dt} = k_1^{R3}[CaPKC][AA] - k_{-1}^{R3}[CaPKC_{memb}^*] \quad (16)$$

$$U_{R4} = \frac{d[AACaPKC^*]R4}{dt} = -\frac{d[CaPKC]R4}{dt} = -\frac{d[AA]R4}{dt} = k_1^{R4}[CaPKC][AA] - k_{-1}^{R4}[AACaPKC^*] \quad (17)$$

$$U_{R5} = \frac{d[DAGPKCmemb^*]R5}{dt} = -\frac{d[DAGCaPKC]R5}{dt} = k_1^{R5}[DAGCaPKC] - k_{-1}^{R5}[DAGPKCmemb^*] \quad (18)$$

$$U_{R6} = \frac{d[AADAGPKC^*]R6}{dt} = -\frac{d[AADAGPKC]R6}{dt} = k_1^{R6}[AADAGPKC] - k_{-1}^{R6}[AADAGPKC^*] \quad (19)$$

$$U_{R7} = \frac{d[CaPKC]R7}{dt} = -\frac{d[PKC_i]R7}{dt} = -\frac{d[Ca^{2+}]R7}{dt} = k_1^{R7}[PKC_i] - k_{-1}^{R7}[CaPKC] \quad (20)$$

$$U_{R8} = \frac{d[DAGCaPKC]R8}{dt} = -\frac{d[CaPKC]R8}{dt} = -\frac{d[DAG]R8}{dt} = k_1^{R8}[CaPKC][DAG] - k_{-1}^{R8}[DAGCaPKC] \quad (21)$$

$$U_{R9} = \frac{d[DAGPKC]R9}{dt} = -\frac{d[PKC_i]R9}{dt} = -\frac{d[DAG]R9}{dt} = k_1^{R9}[PKC_i][DAG] - k_{-1}^{R9}[DAGPKC] \quad (22)$$

$$U_{R10} = \frac{d[AADAGPKC]R10}{dt} = -\frac{d[DAGPKC]R10}{dt} = -\frac{d[AA]R10}{dt} = k_1^{R10}[DAGPKC][AA] - k_{-1}^{R10}[AADAGPKC] \quad (23)$$

By adding the concentrations of intermediate species we have the concentration of active PKC: $[PKCa] = [PKC_{\text{basal}}^*] + [AAPKC^*] + [CaPKC_{\text{memb}}^*] + [AACaPKC^*] + [DAGPKC_{\text{memb}}^*] + [AADAGPKC^*]$ (24)

Equations U_{R1} - U_{R10} are used for implementing the PKC pathway to Matlab. Our

system contains differential equations for 11 species: u_{R1} is for $\frac{d[PKC_{\text{basal}}^*]}{dt}$, u_{R2}

is for $\frac{d[AAPKC^*]}{dt}$, u_{R3} is for $\frac{d[CaPKC_{\text{memb}}^*]}{dt}$, u_{R4} is for $\frac{d[AACaPKC^*]}{dt}$, u_{R5} is

for $\frac{d[DAGPKC_{\text{memb}}^*]}{dt}$, u_{R6} is for $\frac{d[AADAGPKC^*]}{dt}$, for $\frac{d[PKC_i]}{dt}$ we have

$-u_{R1} - u_{R2} - u_{R7} - u_{R9}$, for $\frac{d[CaPKC]}{dt}$ we have $u_{R7} - u_{R3} - u_{R4} - u_{R8}$, for

$\frac{d[DAGPKC]}{dt}$ we have $u_{R9} - u_{R10}$, for $\frac{d[DAGCaPKC]}{dt}$ we have $u_{R8} - u_{R5}$ and for

$\frac{d[AADAGPKC]}{dt}$ we have $u_{R10} - u_{R6}$

Using these equations we will simulate the behavior of our pathway.

Simulation of PKC pathway

The use of MATLAB simulation tool has been used in order to study the behavior of the protein kinase C pathway. The simulation of the neuron model was easier by using an extension of matlab the simulink gui. The version that we used is 7.0.0.19920 (R14) which uses different kinds of ordinary differential equation solvers (ode) in order to solves differential equations. We used the ode23s solver for our simulation with absolute tolerance 10^{-15} and relative tolerance 10^{-6} in the simulation parameters drop-down menu at Simulink's toolbar, but other numerical integration methods (ode) can also be used , like ode45 , ode23, ode11 and many others.

The behavior of the PKC pathway is simulated using different kinds of inputs for Ca^{2+} , AA, and DAG. Various numerically generated stimuli are used for Ca^{2+} , such as linear and step functions as well as rectangular and sine waves. The concentrations of other second messengers of the PKC pathway, AA and DAG, are first kept constant but later also triangular and sine waves are used.

We have divided the simulation in 10 different cases:

- 1) The linear case: Ca^{++} , AA, DAG are kept constant to the 1uM, 50uM, 150uM values respectively.
- 2) The step case: Ca^{++} has a step stimulus from 0 to 10uM every 100 sec, while AA and DAG are kept constant to 50uM and 150uM respectively.
- 3) The rectangular case: Ca^{++} has a rectangular stimulus with time period of 100sec followed by a time period of quiescent 100sec, while AA and DAG are kept constant to 50uM and 150uM respectively.
- 4) The rectangular case: Ca^{++} has a rectangular stimulus with time period of 50sec followed by a time period of quiescent 50sec, while AA and DAG are kept constant to 50uM and 150uM respectively.
- 5) The sine wave case: Ca^{++} has a sine wave stimulus from 0uM to 2uM with time period 50sec, while AA and DAG are kept constant to 50uM and 150uM respectively.

- 6) The sine wave case: Ca^{++} has a sine wave stimulus from 0uM to 2uM with time period 10sec, while AA and DAG are kept constant to 50uM and 150uM respectively.
- 7) Varying sine wave Ca^{++} stimulus while AA and Dag are again kept constant at values 50uM and 150uM respectively:
 - Ca^{++} has a sine wave stimulus from 0uM to 2uM with time period 50sec
 - Ca^{++} has a sine wave stimulus from 1uM to 3uM with time period 50sec
 - Ca^{++} has a sine wave stimulus from 0uM to 2uM with time period 10sec
 - Ca^{++} has a sine wave stimulus from 1uM to 3uM with time period 10sec
- 8) Effect of varying AA and DAG stimuli while Ca^{++} has a sine wave stimuli:
 - AA and DAG are kept constant with values 50uM and 150uM respectively
 - AA and DAG are kept constant with values 1uM and 150uM respectively
 - AA and DAG are kept constant with values 150uM and 150uM respectively
 - AA has a triangular stimuli while DAG is kept constant at 150uM
 - Both AA and DAG have triangular stimuli
- 9) Effect of varying sine wave stimuli for Ca^{++} , AA and DAG
 - Ca^{++} has a sine wave stimuli with time period 50sec, AA has a sine wave stimuli with time period 10sec and DAG has a sine wave stimuli with time period 100sec
 - Same as above but the stimulus is between 49-51uM
 - Ca^{++} has a sine wave stimuli with time period 10sec, AA has a sine wave stimuli with time period 50sec and DAG has a sine wave stimuli with time period 100sec
 - Ca^{++} has a sine wave stimuli with time period 10sec, AA has a sine wave stimuli with time period 100sec and DAG has a sine wave stimuli with time period 50sec

- Ca^{++} has a sine wave stimuli with time period 50sec, AA has a sine wave stimuli with time period 10sec and DAG has a sine wave stimuli with time period 100sec
- Ca^{++} has a sine wave stimuli with time period 100sec, AA has a sine wave stimuli with time period 50sec and DAG has a sine wave stimuli with time period 10sec

10) Effect of varying parameter values

- K_1 value is increased by 20% in R6
- K_1 value is increased by 50% in R6
- K_{-1} value is increased by 20% in R6
- K_{-1} value is increased by 50% in R6
- K_1 value is increased by 20% in R7
- K_1 value is increased by 50% in R7
- K_{-1} value is increased by 20% in R7
- K_{-1} value is increased by 50% in R7
- K_1 and K_{-1} values are increased by 20% in R6 and R7
- K_1 and K_{-1} values are increased by 50% in R6 and R7

We must mention that for all these cases the initial concentrations of the PKC pathway are kept constant and are obtained from the database of Quantitative Cellular Signaling (<http://doqcs.ncbs.res.in/>).

Species	Inti. Concentartion
AA	$5 \cdot 10^{-5}$
AACaPKC*	1.75810^{-22}
AADAGPKC	2.5188810^{-25}
AADAGPKC*	$4.9137 \cdot 10^{-24}$
AAPKC*	$1.8133 \cdot 10^{-23}$
Ca	$1 \cdot 10^{-6}$
CaPKC	$3.7208 \cdot 10^{-23}$
CaPKCmemb*	$1.3896 \cdot 10^{-23}$
DAG	$1.5 \cdot 10^{-4}$
DAGCaPKC	$8.4632 \cdot 10^{-29}$
DAGPKC	$1.161 \cdot 10^{-22}$
DAGPKCmemb*	$9.4352 \cdot 10^{-27}$
PKC _a	$2.122 \cdot 10^{-22}$
PKC _i	$1 \cdot 10^{-6}$

PKCbasal*	$2 \cdot 10^{-8}$
-----------	-------------------

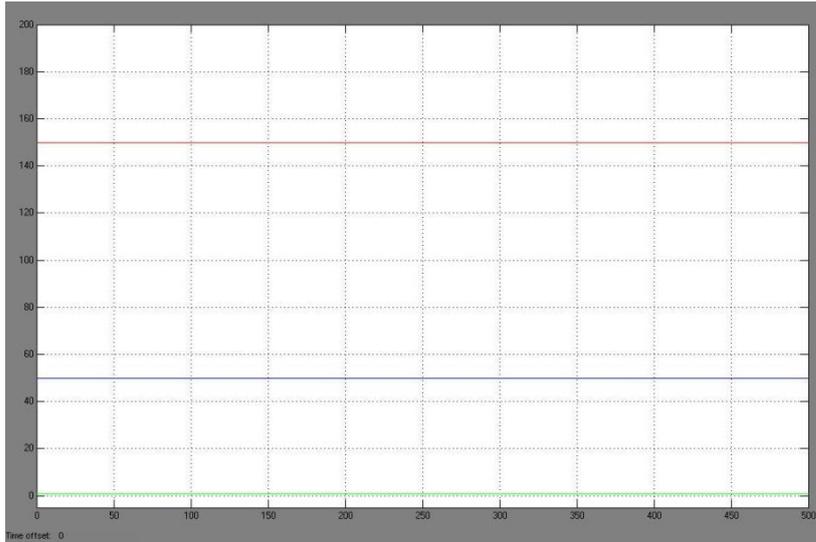
Also the rate constants are kept constant (except from case 10) and are obtained from the database of Quantitative Cellular Signaling.

<i>Reaction</i>	<i>K1</i>	<i>k-1</i>
R1	$1 \frac{1}{s}$	$50 \frac{1}{s}$
R2	$120 \frac{1}{Ms}$	$0.1 \frac{1}{s}$
R3	$1.2705 \frac{1}{s}$	$3.5026 \frac{1}{s}$
R4	$1200 \frac{1}{Ms}$	$0.1 \frac{1}{s}$
R5	$1 \frac{1}{s}$	$0.1 \frac{1}{s}$
R6	$2 \frac{1}{s}$	$0.2 \frac{1}{s}$
R7	$6 \cdot 10^5 \frac{1}{Ms}$	$0.5 \frac{1}{s}$
R8	$7999.8 \frac{1}{Ms}$	$8.6348 \frac{1}{s}$
R9	$600 \frac{1}{Ms}$	$0.1 \frac{1}{s}$
R10	$1.8 \cdot 10^4 \frac{1}{Ms}$	$2 \frac{1}{s}$

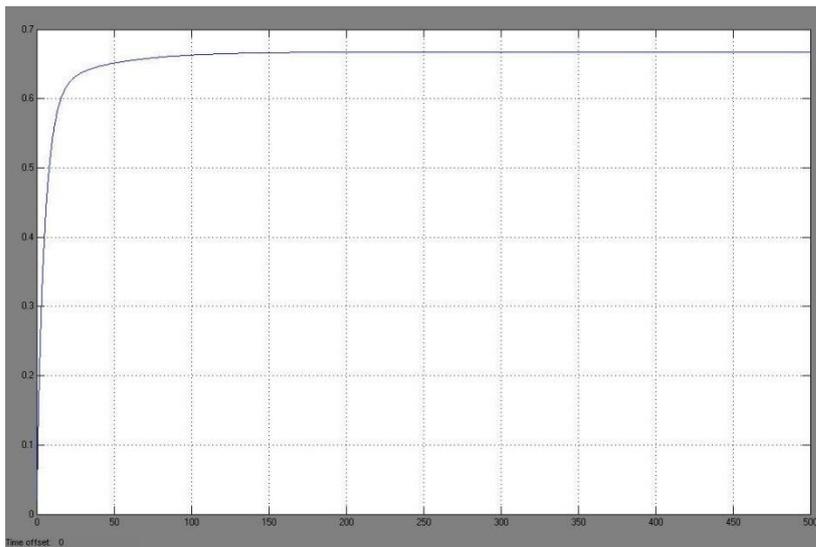
Below we have categorized the 10 cases according to the 10 different stimulus:

Case 1

The linear case: First Ca^{++} (green), AA(blue), DAG(red) are kept constant to the 1uM ,50 uM ,150 uM values respectively.



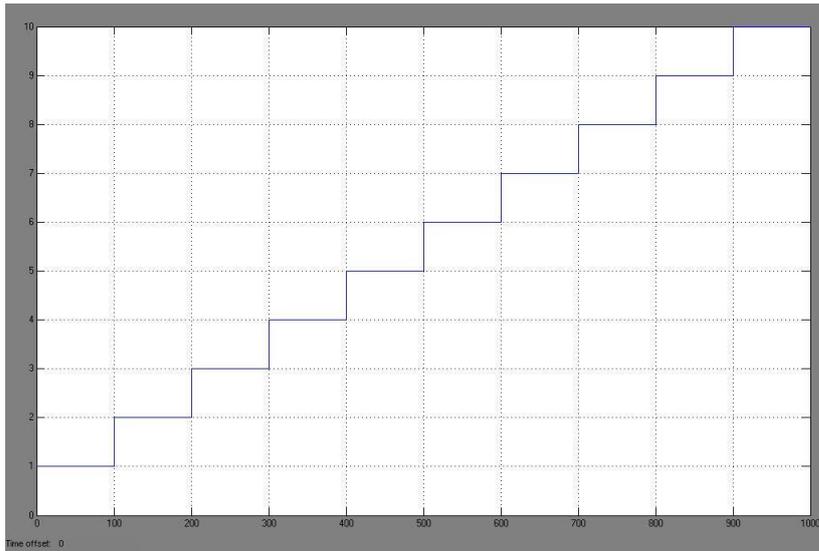
Then, the output value of PKCa has the form as seen below:



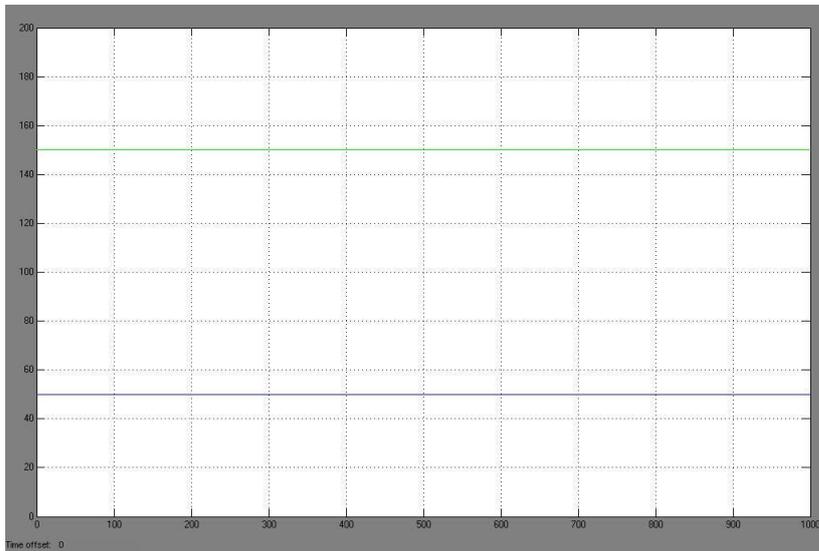
Active PKC converges to 0.66 μM in about 100 seconds.

Case 2

In this case the inputs we use are in the form we see below. At figure A, Ca^{2+} has a step stimulus from 0 to 10 μM every 100secs,

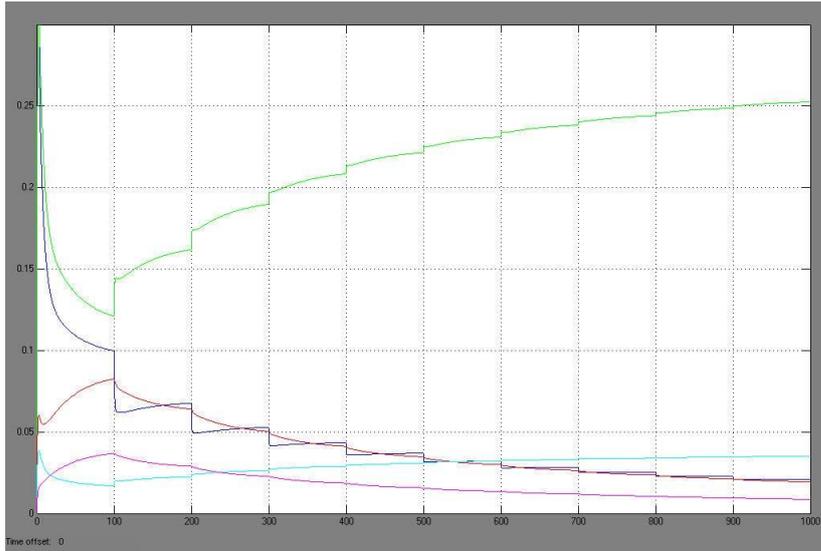


while at figure B [AA] (blue) and [DAG] (green) are kept constants

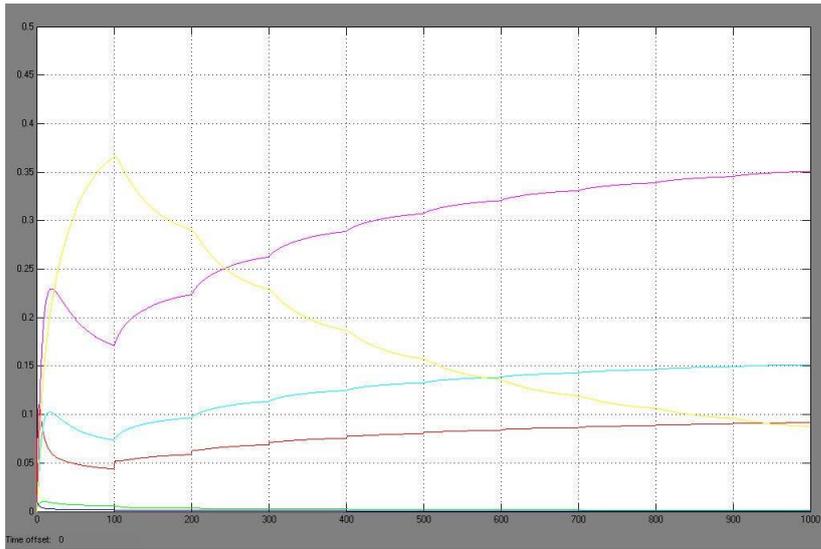


Below the outputs-states of the system are being represented with the following figures:

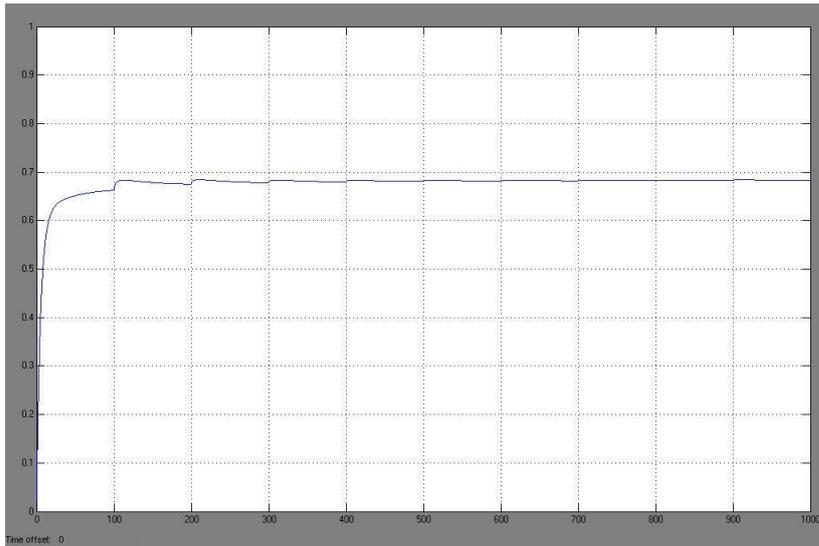
C) [PKCi] (magenta), [CaPKC] (blue), [DAG-PKC] (green), [DAGCaPKC] (cyan), and [AADAGPKC] (black),



D) [PKCbasal*] (magenta), [AAPKC*] (red), [CaPKCmemb*] (blue), [AACaPKC*] (cyan), [DAGPKCmemb*] (green), and [AADAGPKC*] (black)



E) [PKCa].

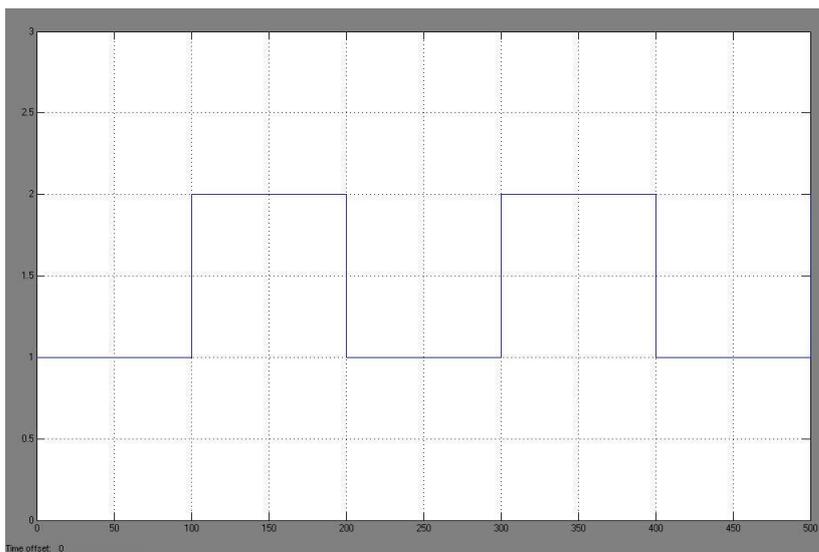


Clear step like behavior is seen in [PKCi] and [CaPKC]. The final product, [PKCa], is also slightly changed at each step.

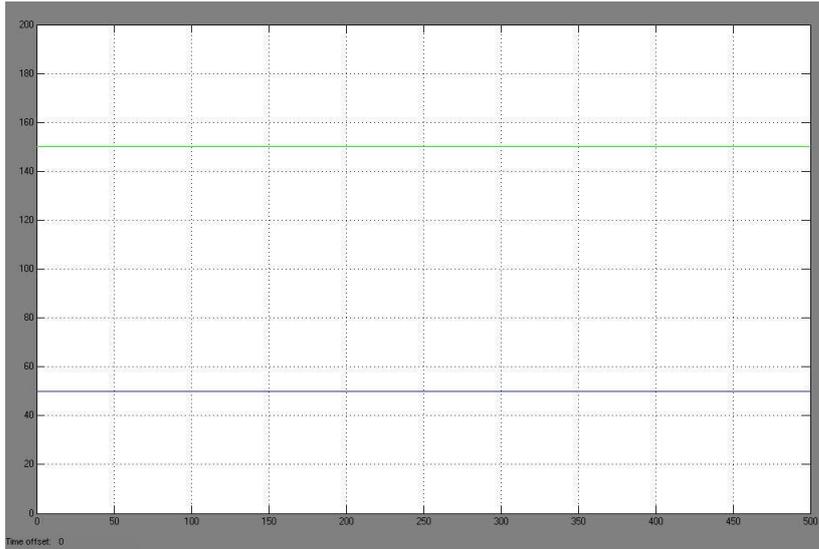
Case 3

In case 3 the inputs we use are of the form we can see in the following graphs.

A) Rectangular Ca^{2+} stimulus with time period of 100 s, followed by a time period of quiescent 100 s

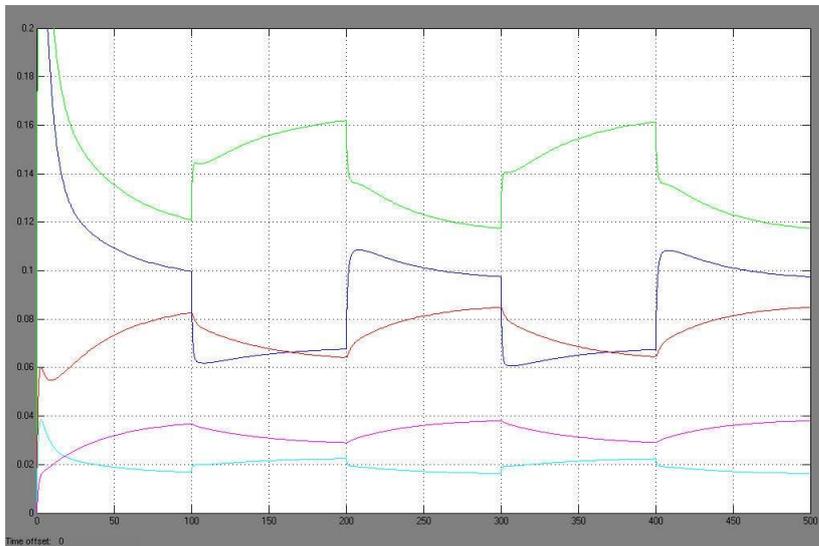


B) [AA] (blue) and [DAG] (green) are kept constant

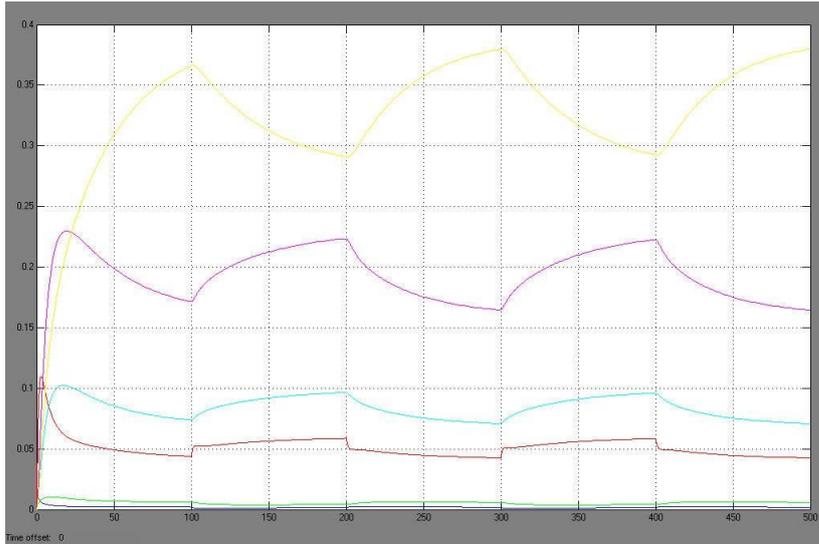


The outputs-states of the system are:

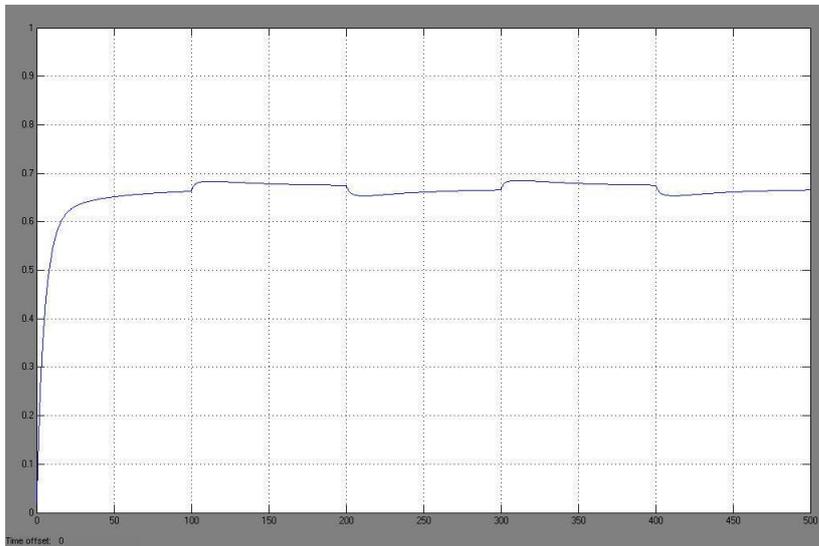
At graph C the following concentrations are being represented: [PKCi] (magenta), [CaPKC] (blue), [DAGPKC] (green), [DAGCaPKC] (cyan), and [AADAGPKC] (black),



while at graph D we have [PKCbasal*] (magenta), [AAPKC*] (red), [CaPKCmemb*] (blue), [AACaPKC*] (cyan), [DAGPKCmemb*] (green), and [AADAGPKC*] (black).



At the last graph E the concentration of PKCa is represented.



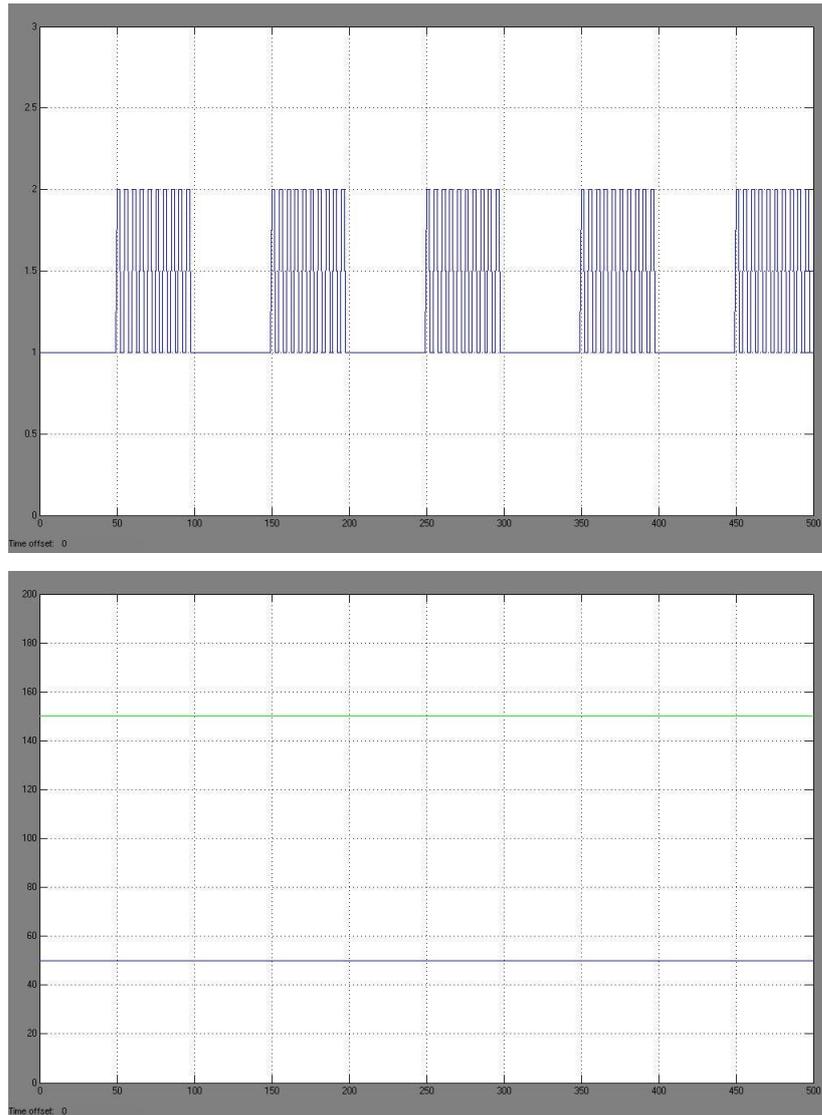
As we can observe clear rectangular behavior is seen in [PKCi], [CaPKC], and [PKCa].

Herein, we believe it is useful to mention that all the graphic images can be seen more clearly if the reader increases the zoom level of the picture. With this way it is more clearly what the behaviors of the studied waveforms are.

Due to the large number of the studied species, it was very difficult for the writer to present all the trajectories one by one, as there are many cases under examination.

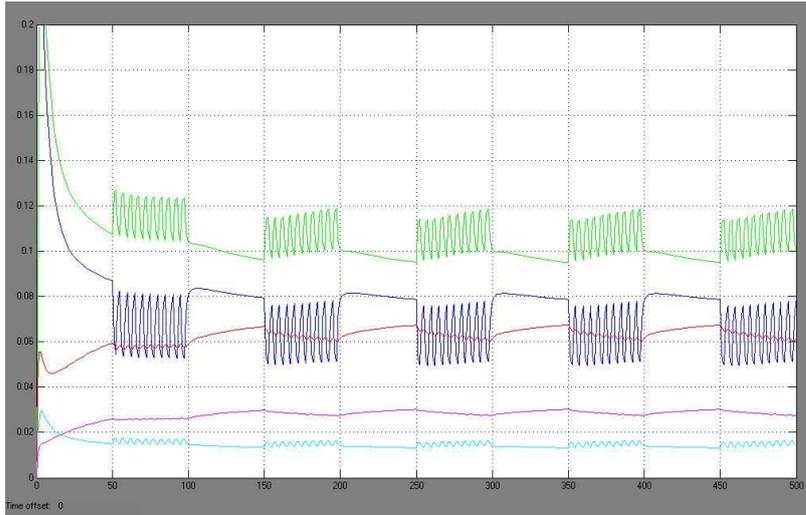
Case 4

Again at the fourth case the concentrations of AA and DAG are kept constant ([AA]=50, [DAG]=150) as we see at figure B while Ca^{2+} has a repetitive rectangular stimulus for 50 secs (10 rectangles), followed by a time period of quiescent 50 secs, (figure A)

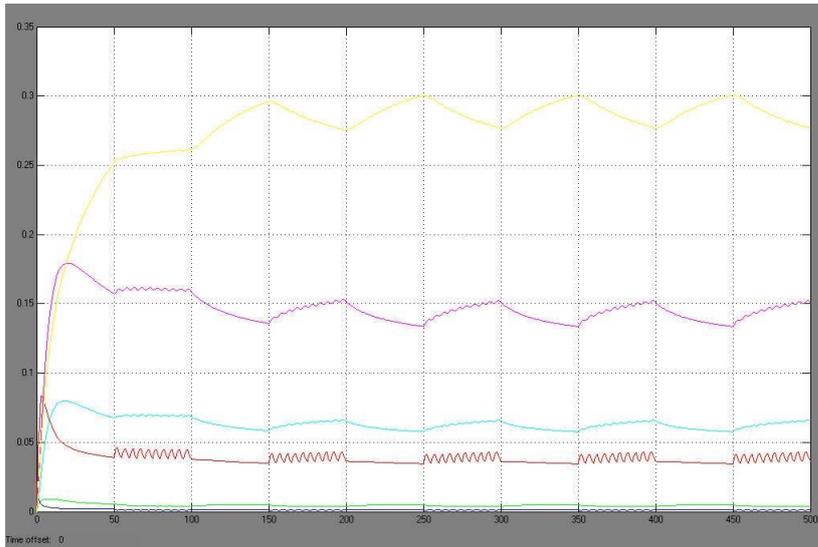


The outputs-states of the system are:

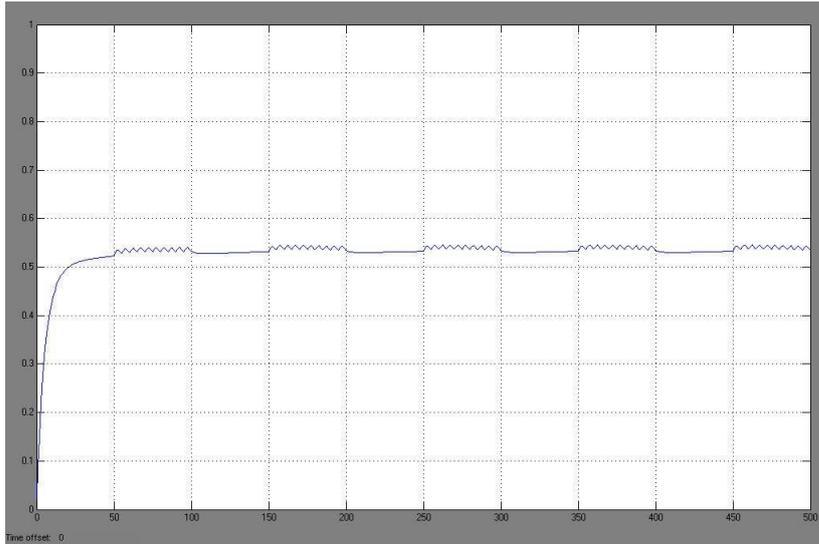
C) [PKCi] (magenta), [CaPKC] (blue), [DAGPKC] (green), [DAGCaPKC] (cyan), and [AADAGPKC] (black)



D) [PKCbasal*] (magenta), [AACPc*] (red), [CaPKCmemb*] (blue), [AACaPKC*] (cyan), [DAGPKCmemb*] (green), and [AADAGPKC*] (black)



E) [PKCa]



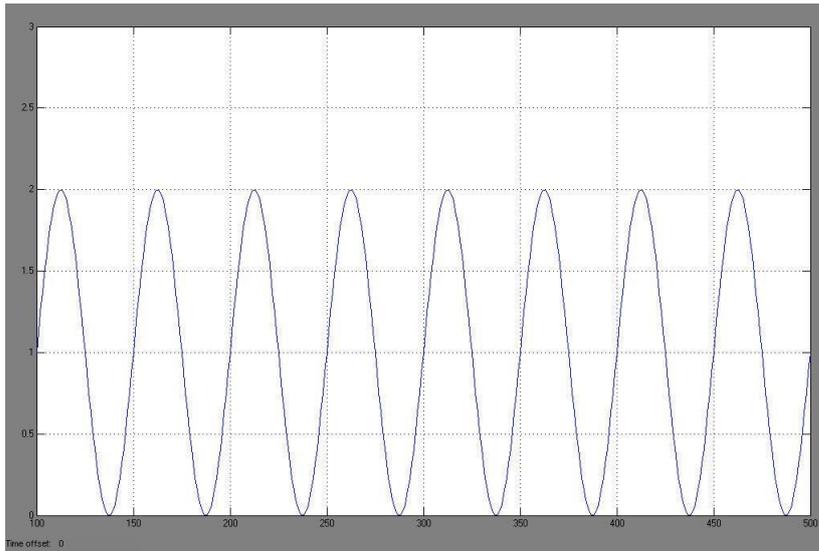
Clear oscillatory behavior is seen in [PKCi], [CaPKC], and [PKCa].

Analytically, if we carefully notice the trajectory of PKCa we will see that there is a slight oscillatory behavior during the time spaces were Ca^{2+} stimulus the whole system. For example notice the PKCa behavior between 50-100 sec 150-200, 250-300 and so on.

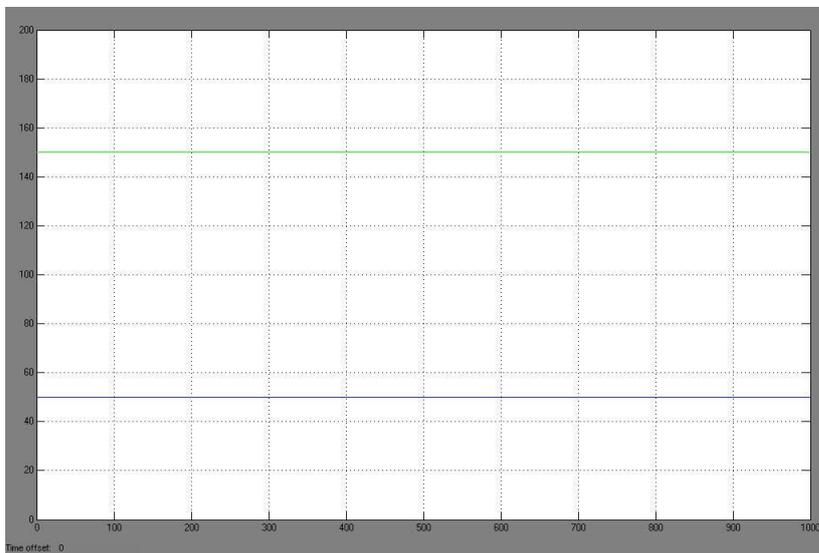
Case 5

At case 5 the inputs we examine are:

Sine wave Ca^{2+} stimulus from 0 uM to 2 uM, time period 50 s, at figure A.



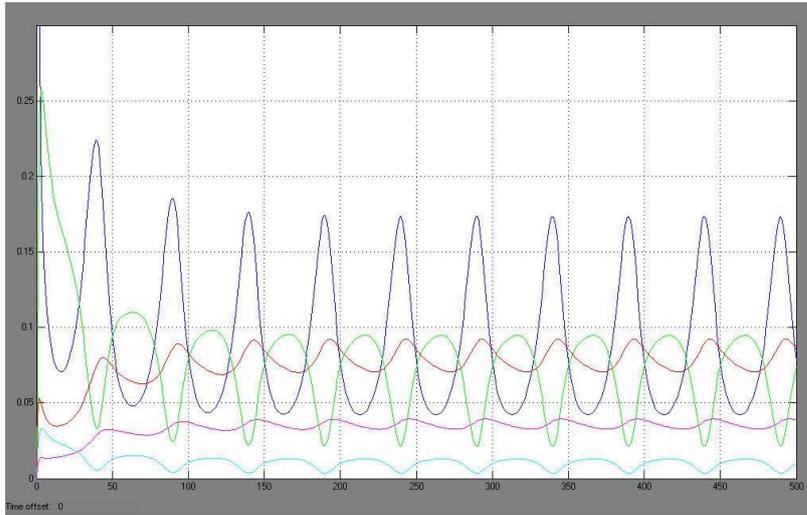
Again at figure B: [AA] (blue) and [DAG] (green) are kept constants



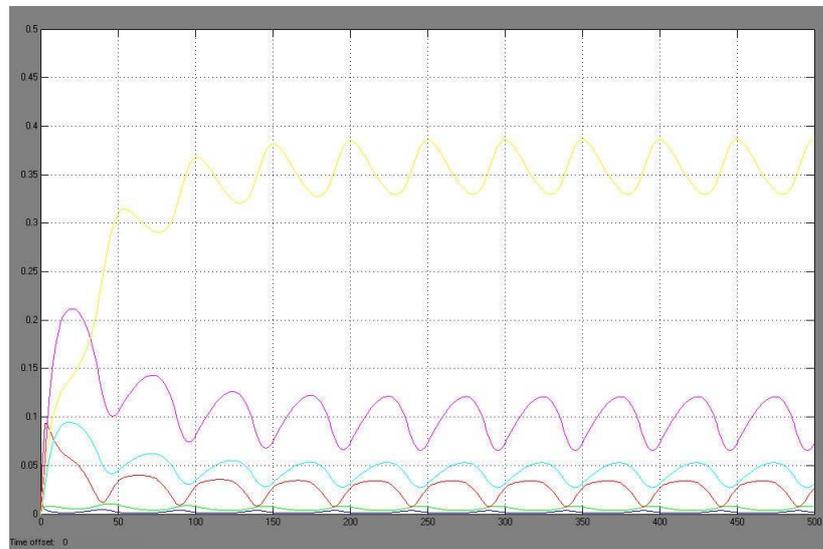
Then for the above stimulations we have the next output waveforms that describe each system state.

Figures:

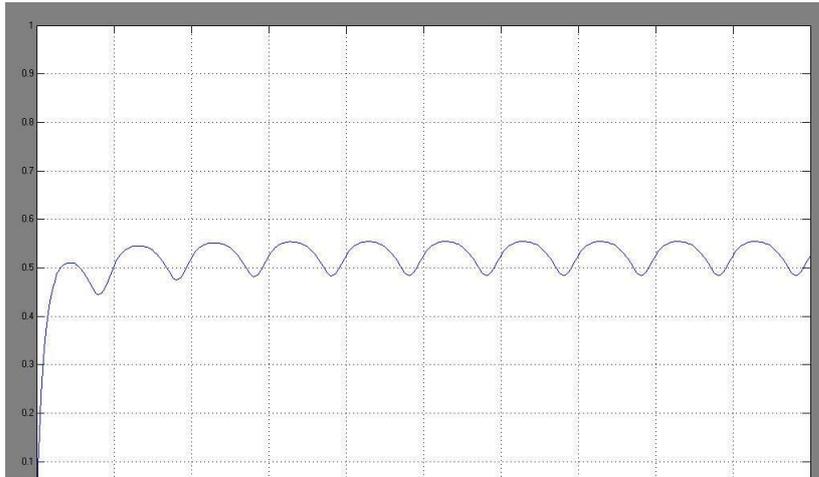
C) [PKCi] (magenta), [CaPKC] (blue), [DAGPKC] (green), [DAGCaPKC] (cyan), and [AADAGPKC] (black)



D) [PKCbasal*] (magenta), [AAPKC*] (red), [CaPKCmemb*] (blue), [AACaPKC*] (cyan), [DAGPKCmemb*] (green), and [AADAGPKC*] (black)



E) [PKCa].

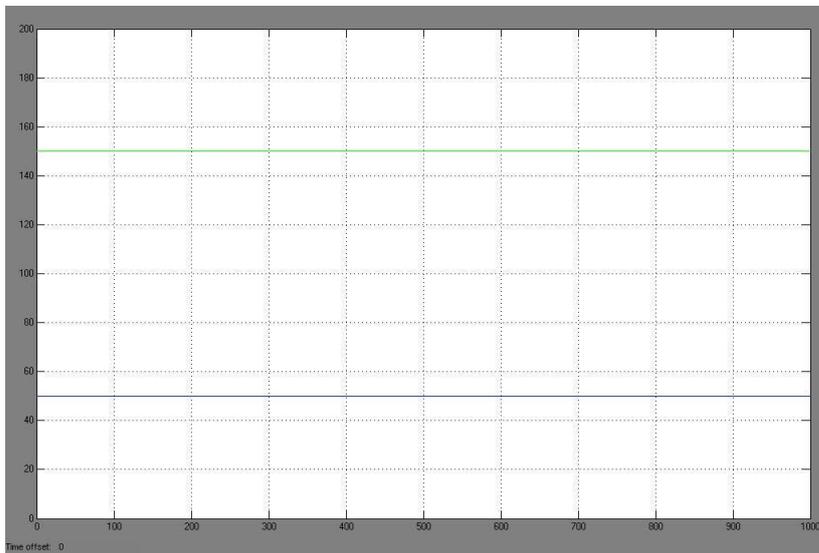
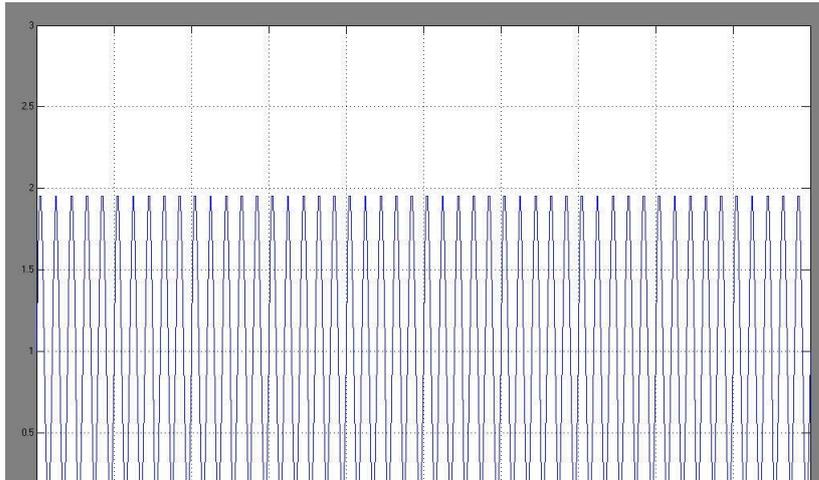


Clear oscillatory behavior is seen in all concentrations.

Specifically, in case 5 we use a simulation in which a sine wave of time period 50 s, as the Ca^{2+} stimulus for the PKC pathway. Immediately can be seen that the concentrations of all species follow the stimulus. In this case, $[\text{PKC}_a]$ has the amplitude of 0.05 μM . We also implement the PKC differential equation model in GENESIS/Kinetikit. We verify that the Matlab simulation gives the same results as the GENESIS/Kinetikit simulation when using the input of figure A.

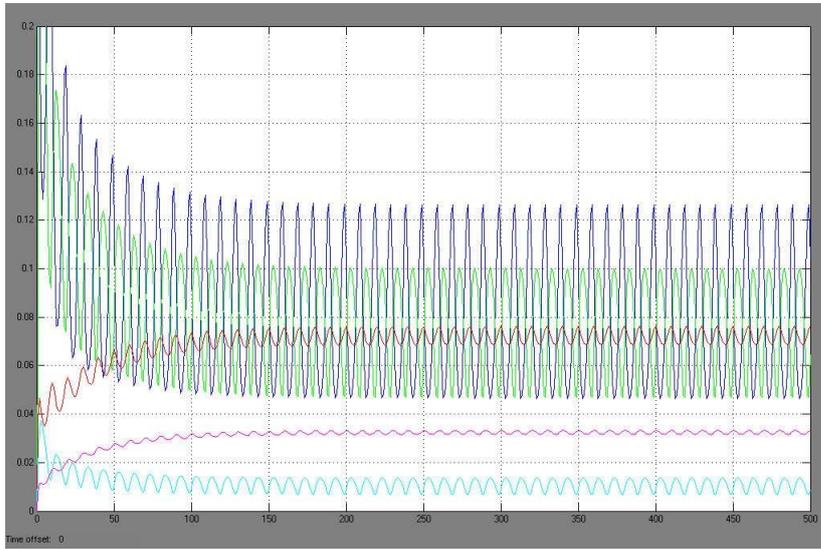
Case 6

At this case we have at the first figure (A) a sine wave Ca^{2+} stimulus from 0 μM to 2 μM , time period 10 s, while at figure B again $[\text{AA}]$ (blue) and $[\text{DAG}]$ (green) are kept constants

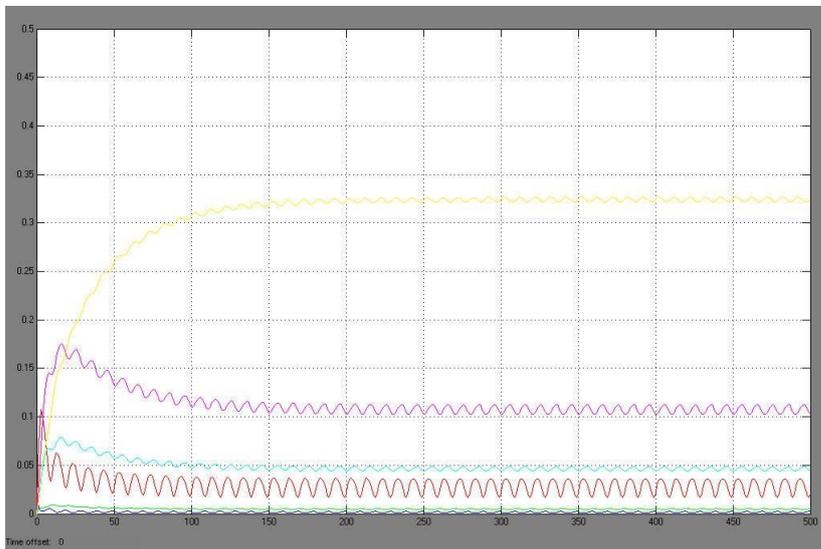


The outputs of the system are:

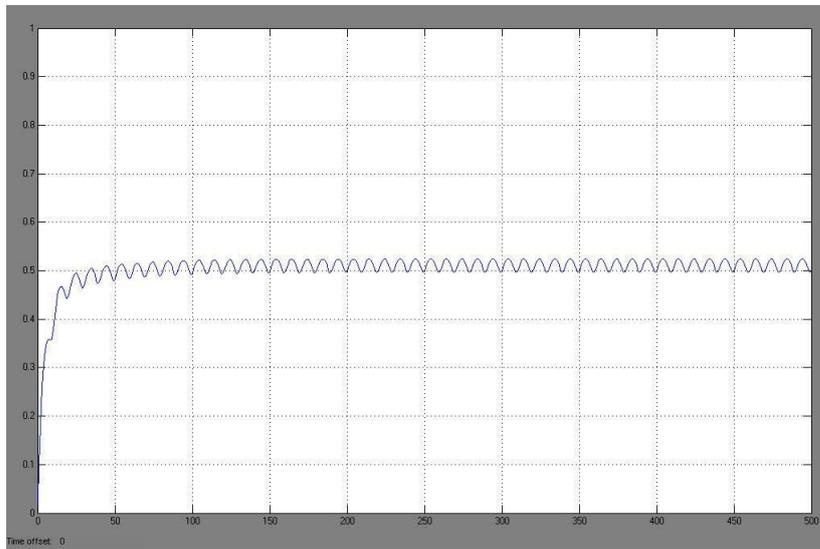
C) [PKCi] (magenta), [CaPKC] (blue), [DAGPKC] (green), [DAGCaPKC] (cyan), and [AADAGPKC] (black).



D) [PKCbasal*] (magenta), [AAPKC*] (red), [CaPKCmemb*] (blue), [AACaPKC*] (cyan), [DAGPKCmemb*] (green), and [AADAGPKC*] (black).



E) [PKCa].



Clear oscillatory behavior is seen in all concentrations.

For a comparison, case 6 (A) shows a sine wave of time period 10 s as the Ca^{2+} stimulus for the PKC pathway. The concentrations of all species in case 6 change according to the Ca^{2+} stimulus. The amplitude of $[\text{PKCa}]$ in Figure 6 (E) is $0.02 \mu\text{M}$. Furthermore, a sine wave of time period 3 s is used as the stimulus for calcium ions (not shown). The concentrations of all species change according to the sine wave but the changes are minimal. In this case, the amplitude of $[\text{PKCa}]$ is $0.002 \mu\text{M}$ which is ten times less compared to the amplitude of $[\text{PKCa}]$ in case 5 and 6.

Case 7

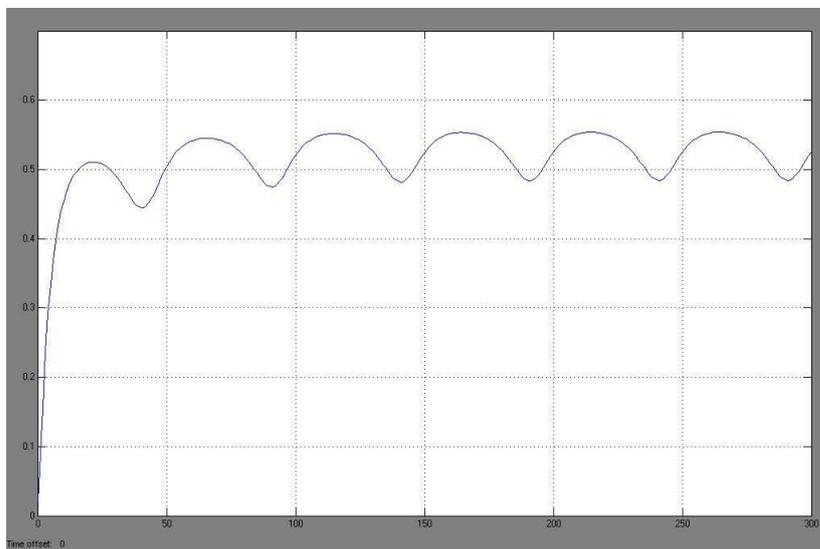
In this case we examine the effect of varying Ca^{2+} stimulus, while $[\text{AA}]$ and $[\text{DAG}]$ are kept constants.

Subcase7.1:

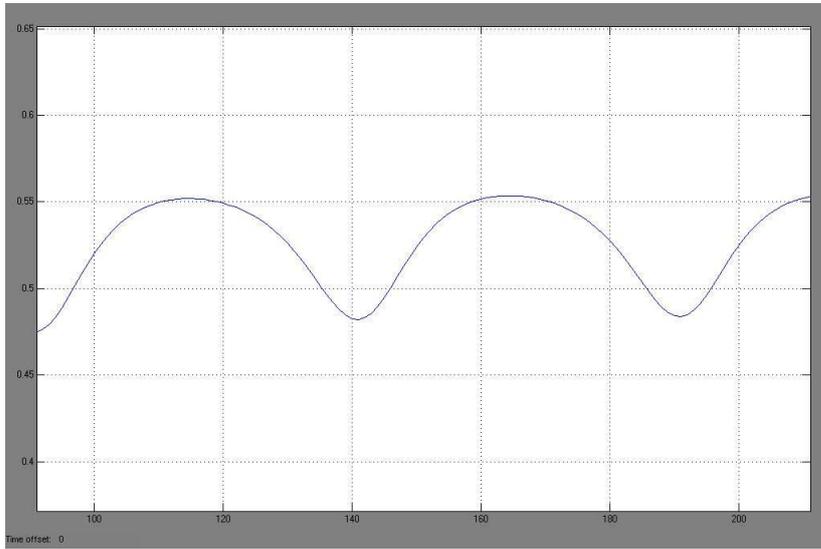
A1) Sine wave Ca^{2+} stimulus, with time period of 50 s.



At figure A2 we see the PKCa response to stimulus from A1 while

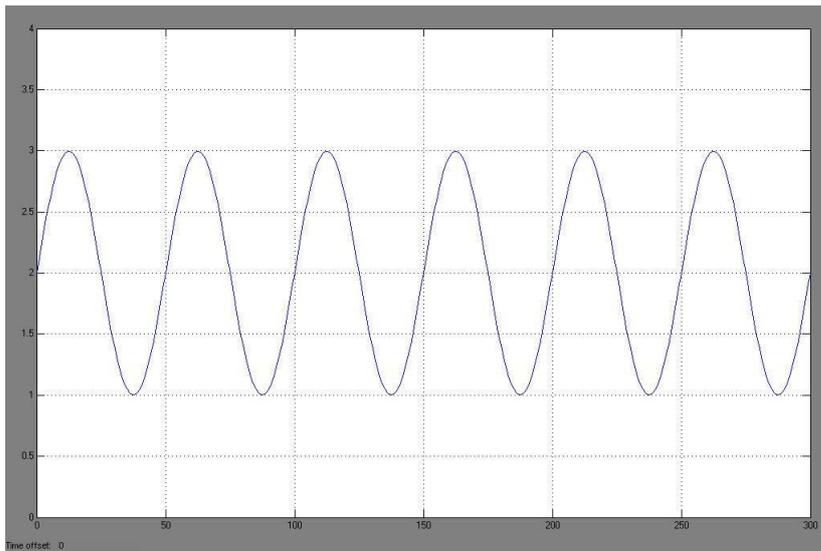


at figure A3 we have a detailed presentation of A2.

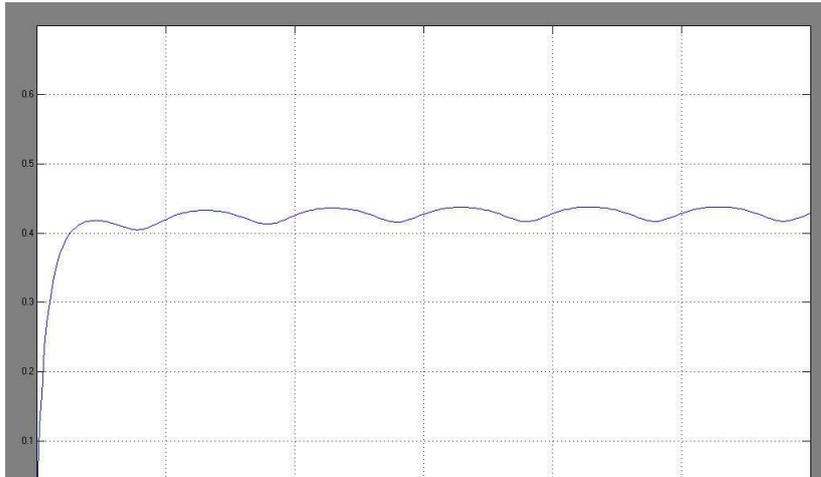


Subcase7.2:

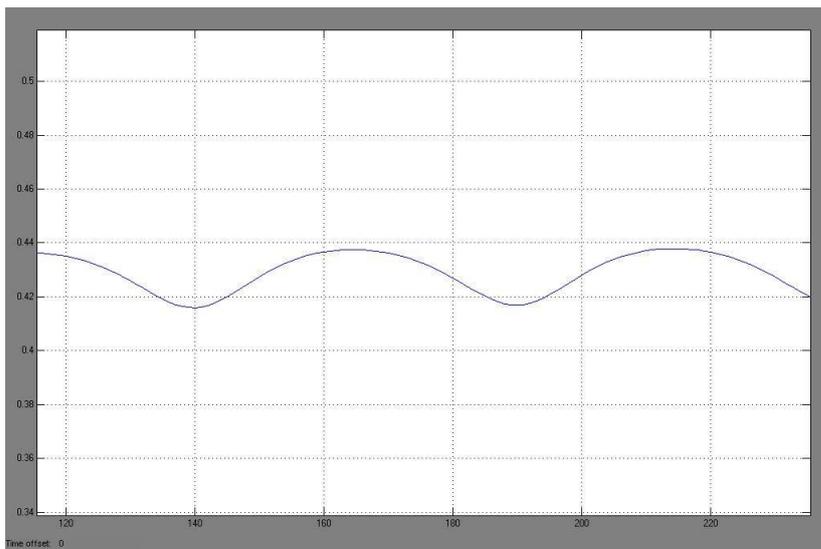
B1) Sine wave Ca^{2+} stimulus, with time period of 50 s.



At figure B2 we see the PKCa response to stimulus from B1



while at figure B3 we have a detailed presentation of B2.

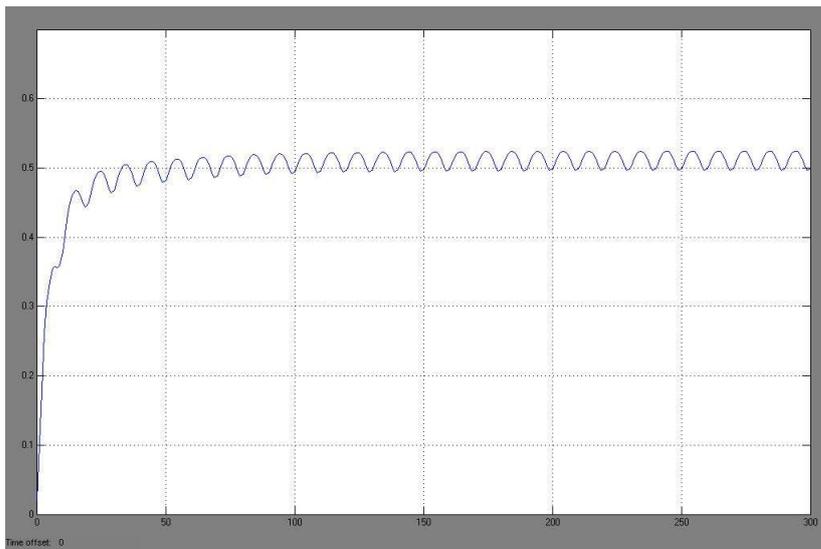


Subcase7.3:

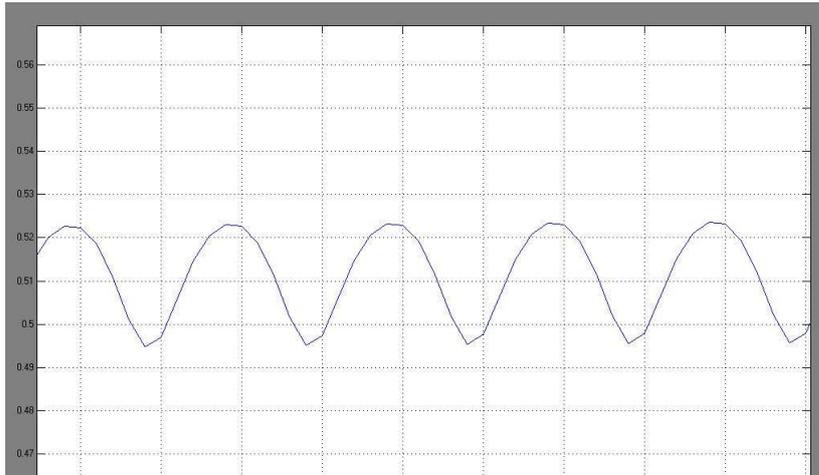
C1) Sine wave Ca^{2+} stimulus, with time period of 10 s.



At figure C2 we see the PKCa response to stimulus from C1

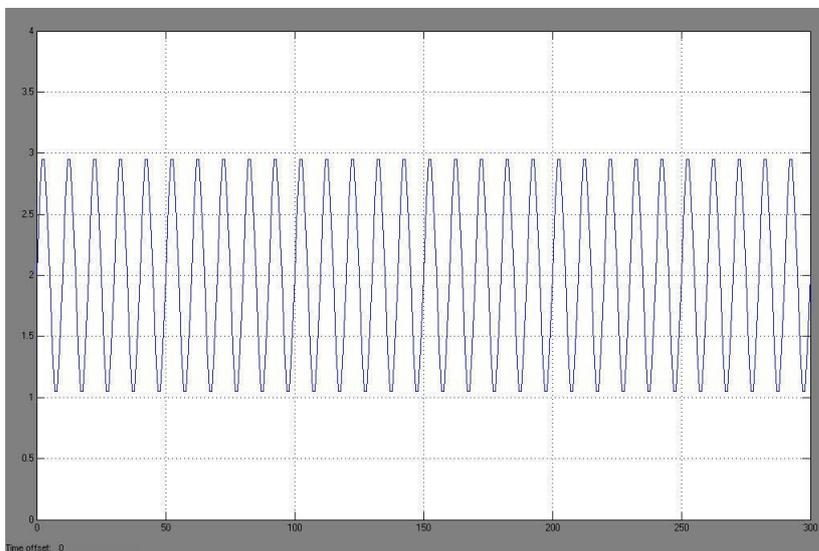


while at figure C3 we have a detailed presentation of C2.

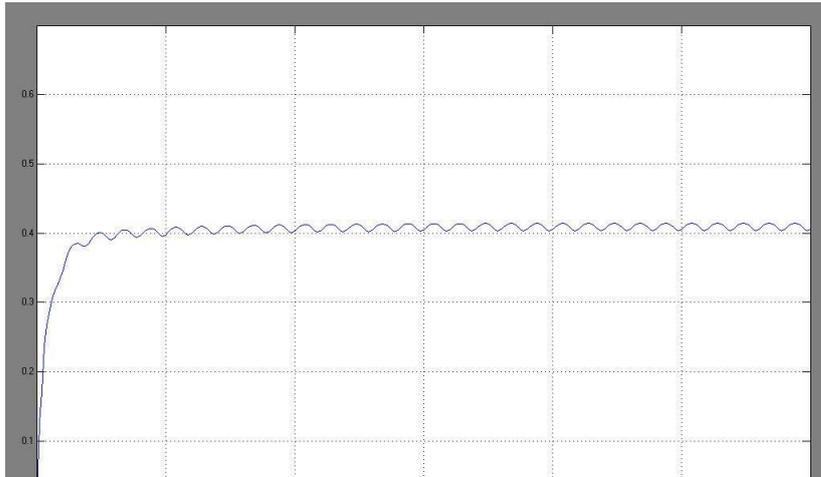


Subcase7.4:

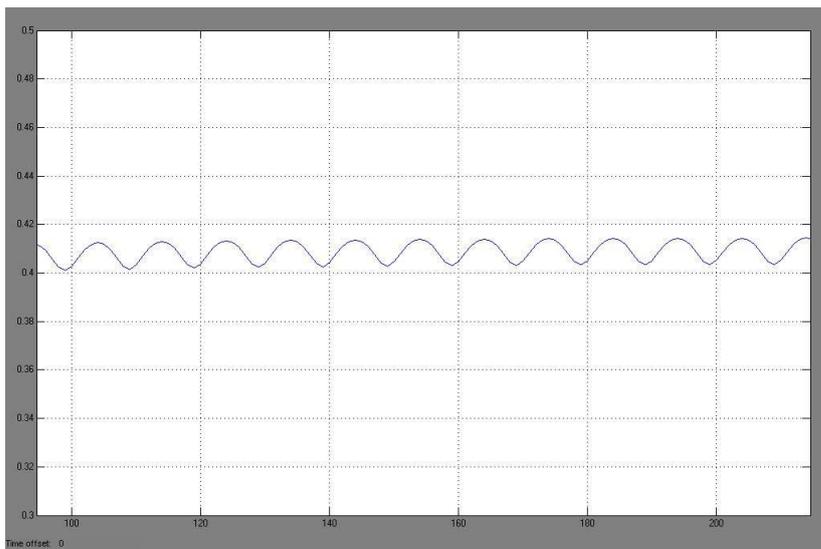
D1) Sine wave Ca^{2+} stimulus, with time period of 10 s.



At figure D2 we see the PKCa response to stimulus from D1 while



at figure D3 we have a detailed presentation of D2.



A detailed sine wave comparison is being represented in case 7. Sine waves of time period 50 secs are used as the stimulus to calcium ions for the first two sub cases while a time period of 10 secs is used for the sub case's 3,4. The difference between the curves of sub case 7.1 and 7.2 is that the concentration of Ca^{2+} changes between $0 \mu\text{M}$ and $2 \mu\text{M}$ on the first sub case and between $1 \mu\text{M}$ and $3 \mu\text{M}$ on the second. Respectively, the same changes in the concentration of Ca^{2+} are made for sub case 7.3

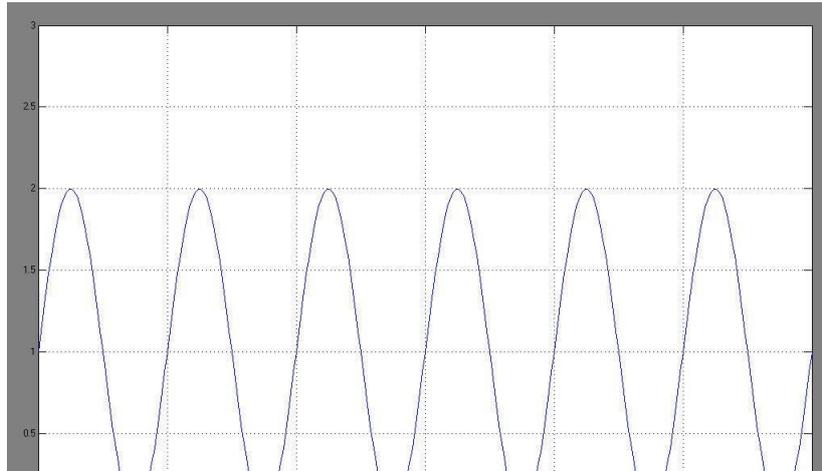
and 7.4. The changes in [PKCa] are not as large on the sub case 7.2 and 7.4 because the stimulus is not reset to 0 μM on these cases. In case where the concentration of Ca^{2+} changes between 1 μM and 3 μM and the time period of the signal is 50 s, the amplitude of [PKCa] is 0.02 μM . Furthermore, when the time period is 10 s, the amplitude of [PKCa] is 0.01 μM .

The table below shows the activation delays and [PKCa] amplitudes in the case of sine wave stimuli. The delay is obtained by comparing the time difference in the concentration of the Ca^{2+} stimulus and PKCa end product at local maximum and minimum concentration points. Delays are calculated at two time points, at 100 s and 450 s. It can be observed that the longer the time period of the Ca^{2+} stimulus, the larger the amplitude of [PKCa] and the longer the activation delay. If the Ca^{2+} stimulus has the concentration of 0 μM at specific time points during the simulation, the amplitude of [PKCa] becomes larger and the delay longer.

Sine wave stimulus	Amplitude of [PKCa]	Delay at 100sec	Delay at 450sec
0-2 μM for period 50s	0.05 μM	3sec	2sec
1-3 μM for period 50s	0.02 μM	2sec	2sec
0-2 μM for period 10s	0.02 μM	2sec	1sec
1-3 μM for period 10s	0.01 μM	1sec	1sec

Case 8

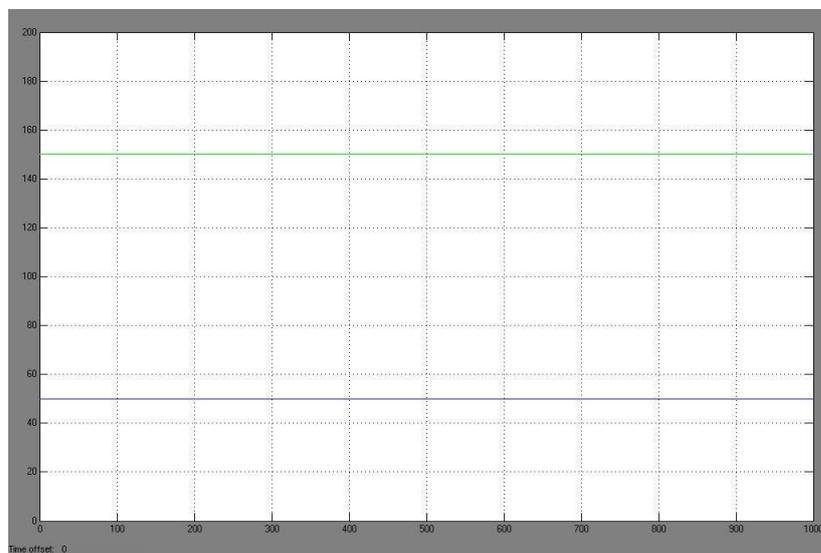
At this case we keep constant the waveform of the Ca^{2+} stimulus. Specifically the sine wave has a time period of 50 secs and concentrations that vary from 0 μM to 2 μM . For the Ca^{2+} we have the following plot:



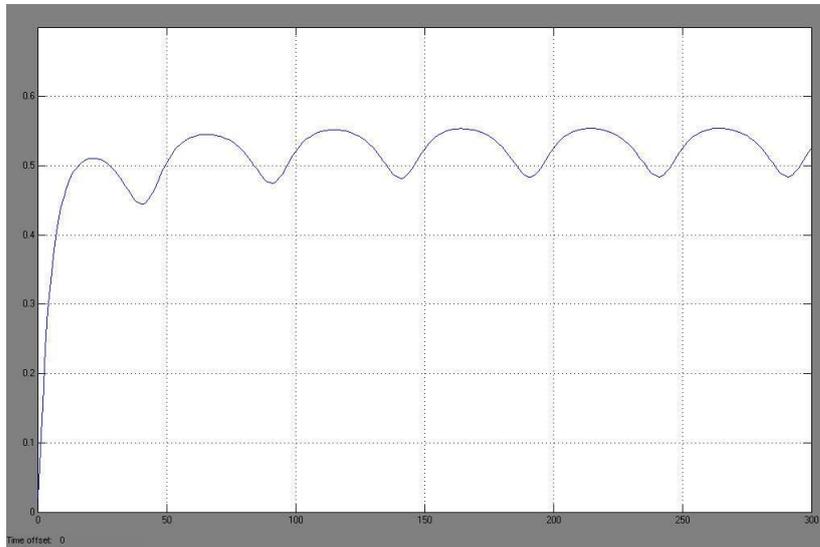
For the next sub cases we study the effect of varying AA and DAG stimuli.

Sub case 8.1

Plot A: Constant AA, 50 μM (blue) and DAG, 150 μM (green) stimuli

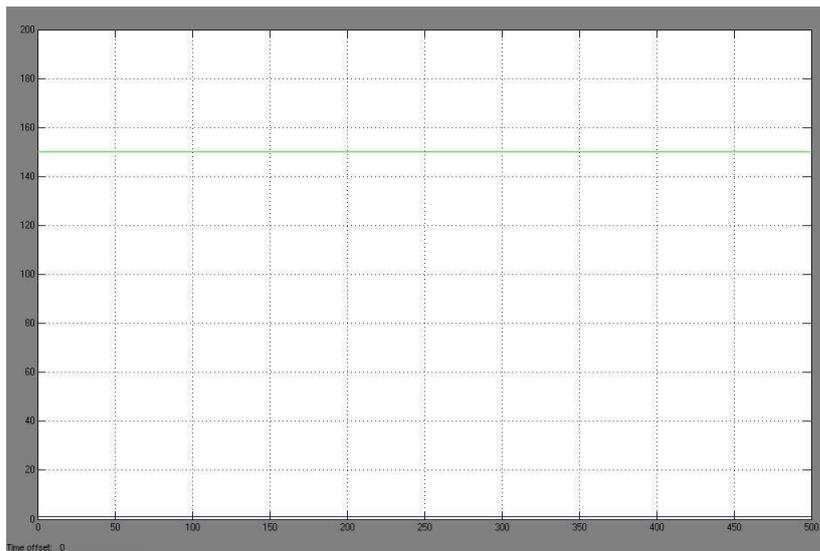


Plot B: PKC_a response to stimuli in A and Ca⁺⁺

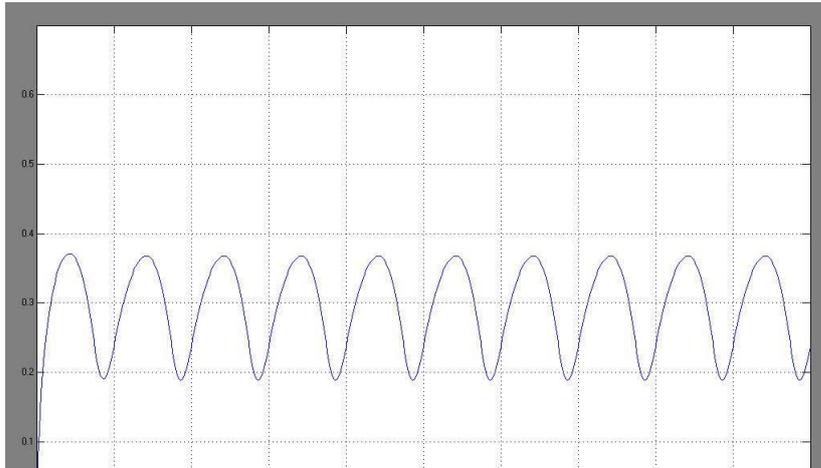


Sub case 8.2

Plot C: constant AA, 1 uM (blue) and DAG, 150 uM (green) stimuli

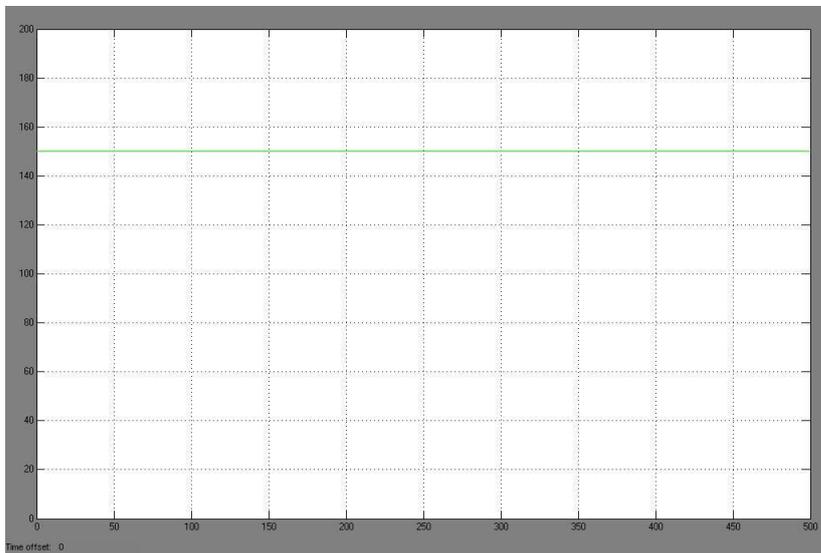


Plot D: PKC_a response to stimuli in C and Ca⁺⁺

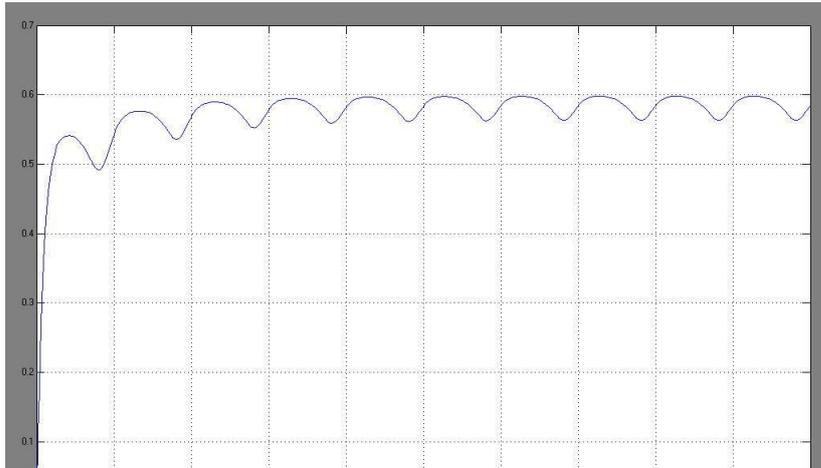


Sub case 8.3

Plot E: constant AA, 150 uM and DAG, 150 uM stimuli

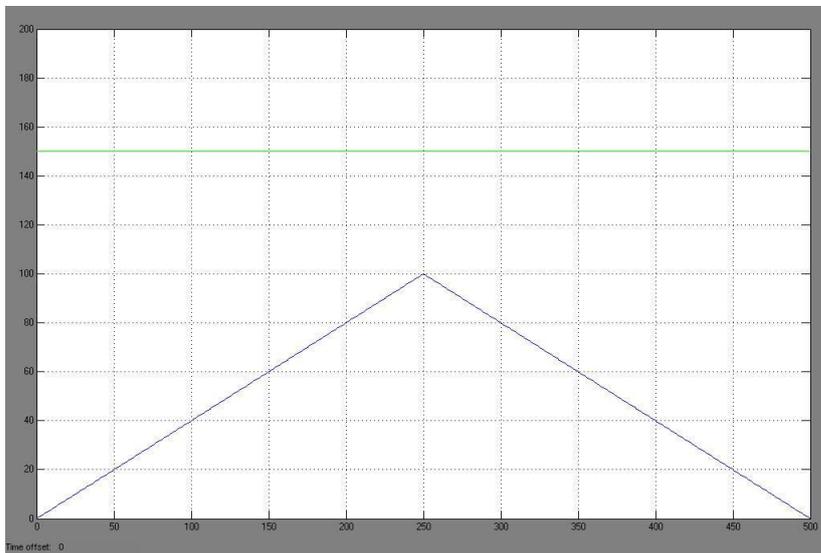


Plot F: PKC_a response to stimuli in E and Ca⁺⁺

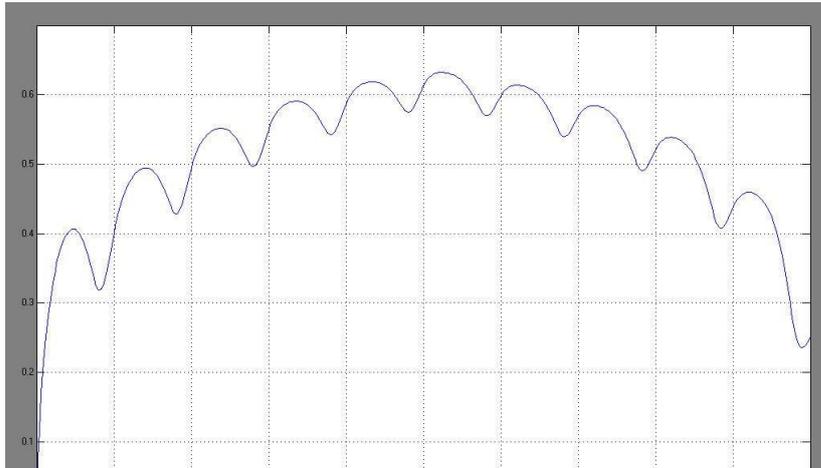


Sub case 8.4

Plot G: triangular AA (blue) and constant DAG (green) stimuli

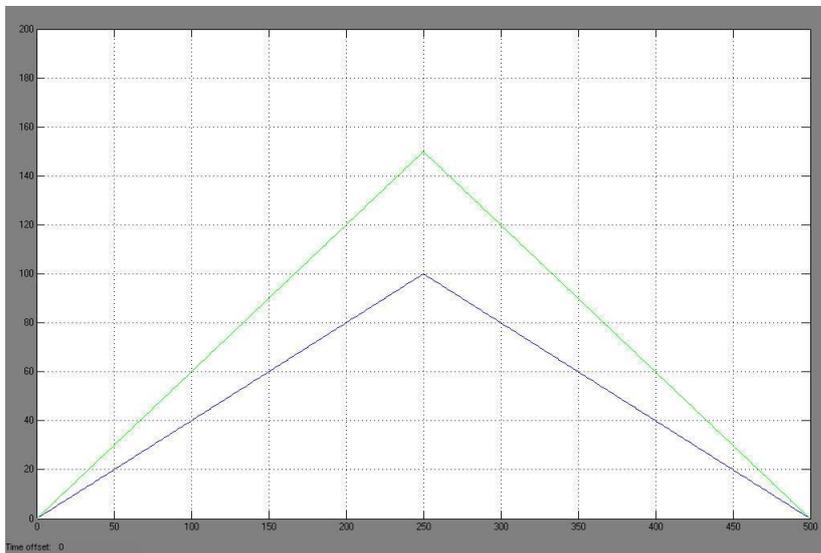


Plot H: PKCa response to stimuli in G and Ca⁺⁺

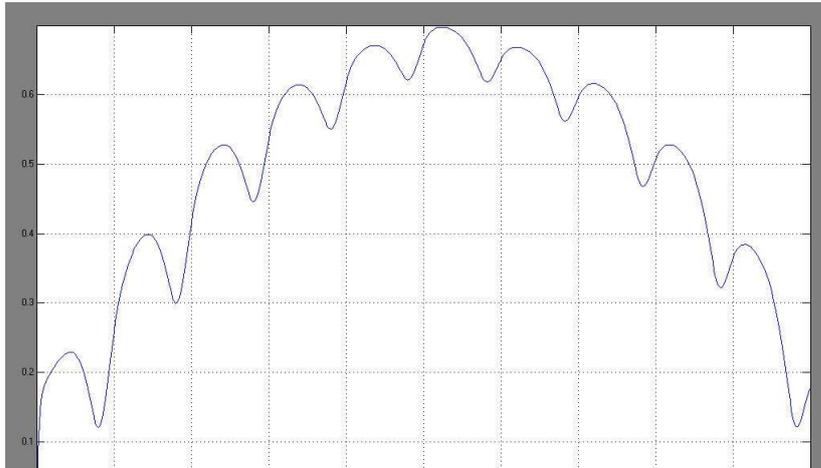


Sub case 8.5

Plot I: triangular AA (blue) and DAG (green) stimuli



Plot J: PKCa response to stimuli in I and Ca^{++}



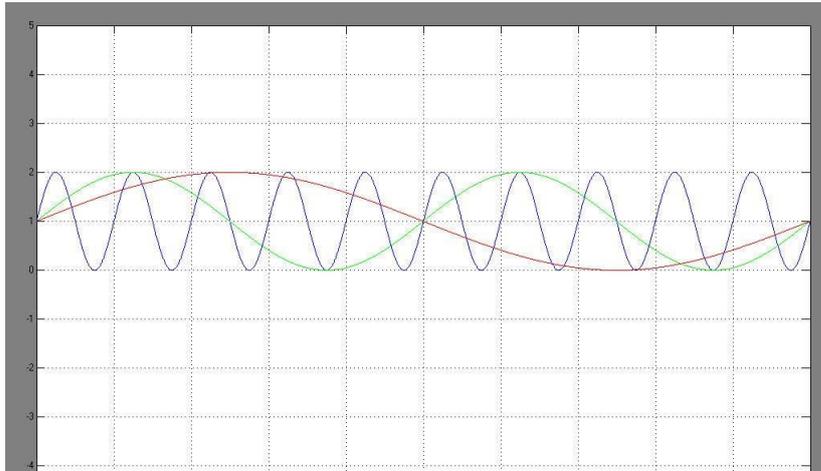
When the Ca^{2+} stimulus is a sine wave and the constant concentration value of AA is increased from 50 μM to 150 μM , also the [PKCa] values are increased, and when the constant concentration value of AA is decreased to 1 μM , the [PKCa] values are decreased. When the Ca^{2+} stimulus is a sine wave, and [DAG] and [AA] are triangular waves, [PKCa] has a sine wave form with the same time period as the Ca^{2+} stimulus but it follows the triangular forms of the AA and DAG stimuli.

Case 9

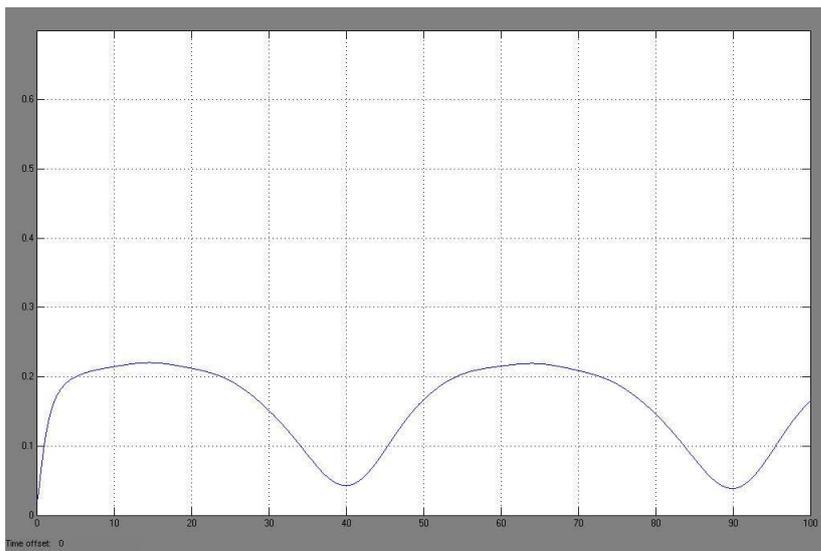
In case 9 we use different kinds of sine waves stimuli for all second messengers Ca^{++} , AA, DAG.

Sub case 9.1

Plot A1: Sine wave Ca^{2+} , time period 50 s (green), AA, time period 10 s (blue), and DAG, time period 100 s (red) stimuli.

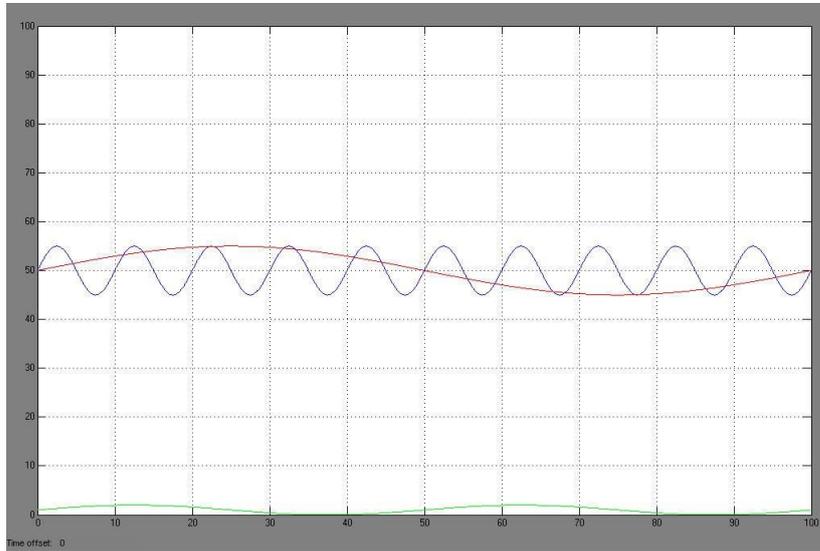


Plot A2: PKCa response to stimuli in A1.

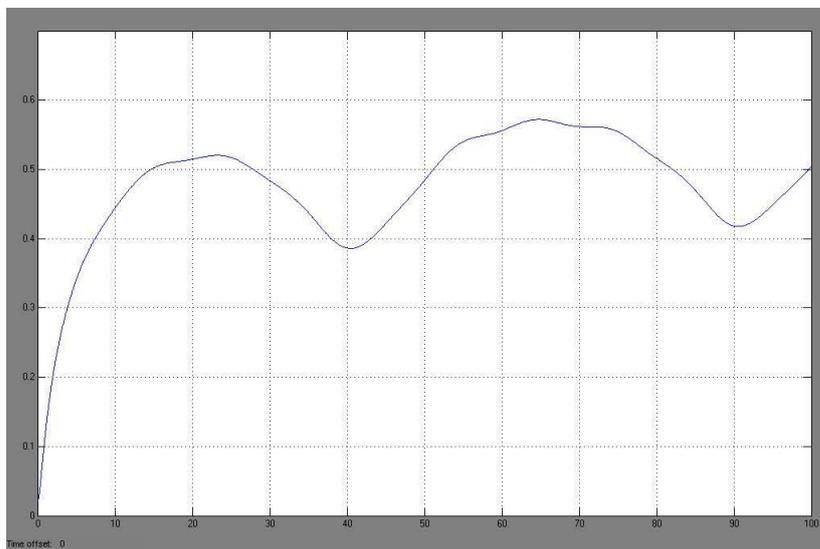


Sub case 9.2

Plot B1: Sine wave Ca^{2+} , time period 50 s (green), AA, time period 10 s (blue), and DAG, time period 100 s (red) stimuli.

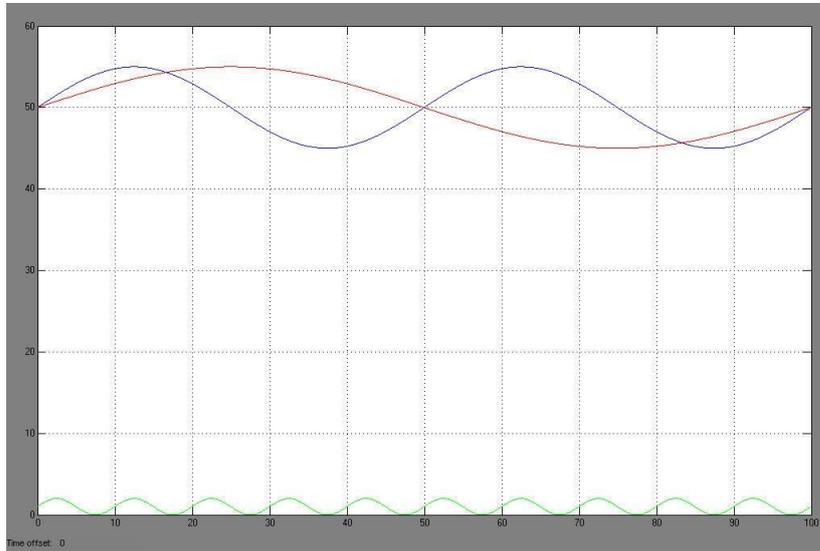


Plot B2: PKCa response to stimuli in B1.

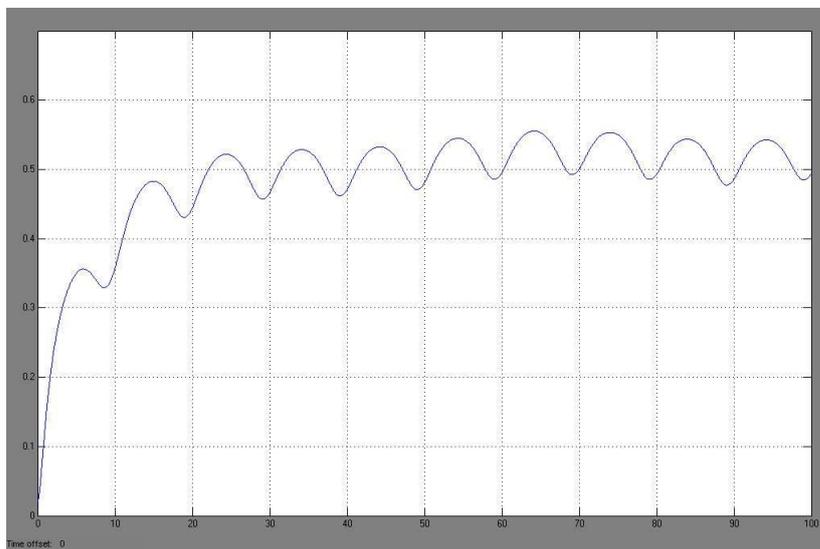


Sub case 9.3

Plot C1: Sine wave Ca^{2+} , time period 10 s (green), AA, time period 50 s (blue), and DAG, time period 100 s (red) stimuli

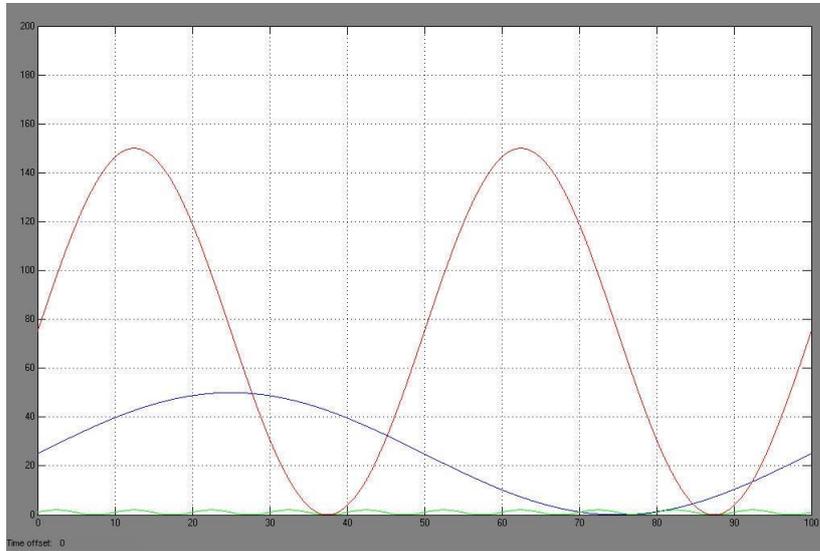


Plot C2: PKCa response to stimuli in C1.

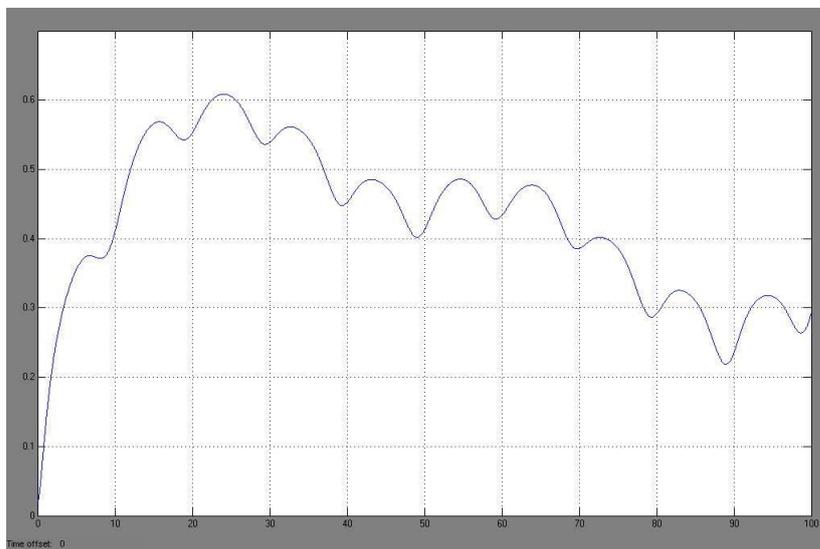


Sub case 9.4

Plot D1: Sine wave Ca²⁺, time period 10 s (green), AA, time period 100 s (blue), and DAG, time period 50 s (red) stimuli.

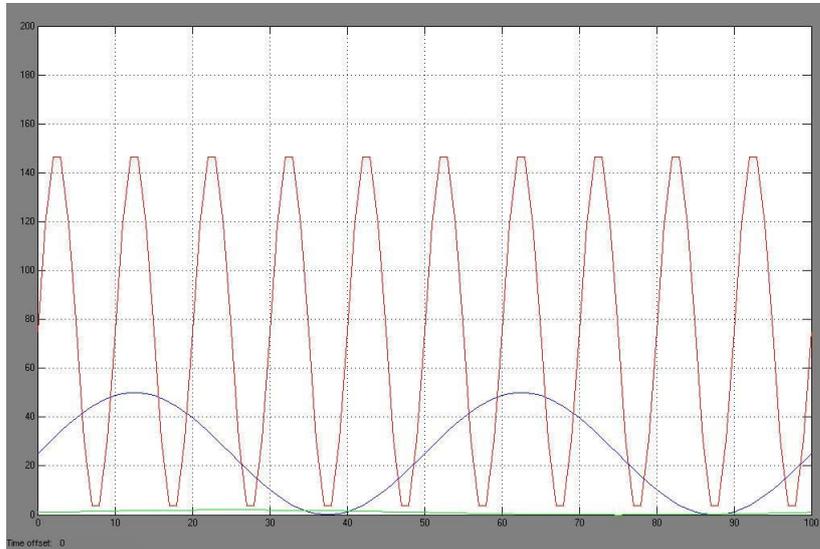


Plot D2: PKCa response to stimuli in D1.

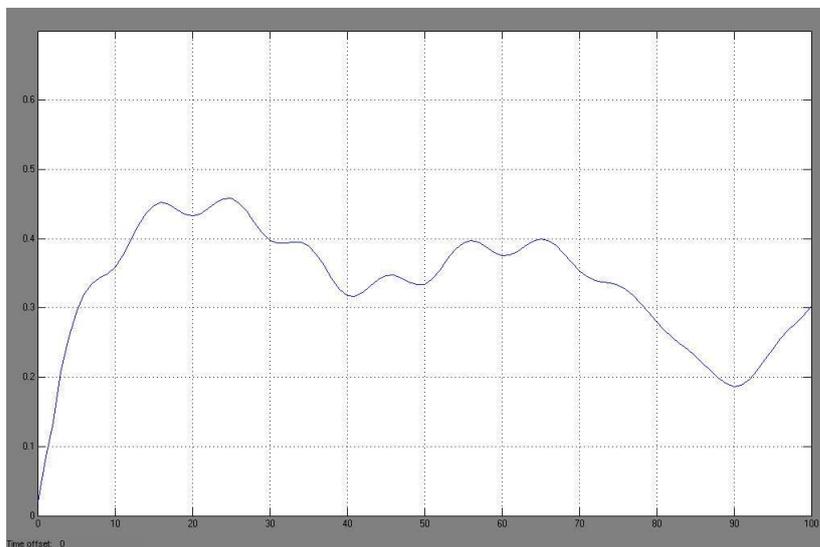


Sub case 9.5

Plot E1: Sine wave Ca^{2+} , time period 100 s (green), AA, time period 50 s (blue), and DAG, time period 10 s (red) stimuli.



Plot E2: PKCa response to stimuli in E1.



We notice that if the values of $[\text{Ca}^{2+}]$, $[\text{AA}]$, and $[\text{DAG}]$ are between 0 and 2 μM , 45 and 55 μM , and 45 and 55 μM , respectively, the form of $[\text{PKC}_a]$ follows the form of

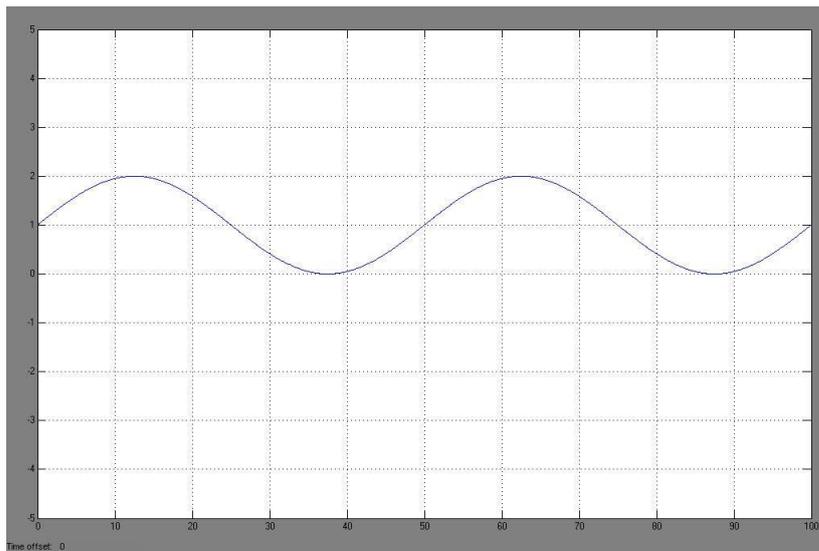
[Ca²⁺]. However, when values of [Ca²⁺], [AA], and [DAG] are between 0 and 2 uM, 0 and 50 uM, and 0 and 150 uM, respectively, the form of [PKC_a] follows all stimuli.

Case 10

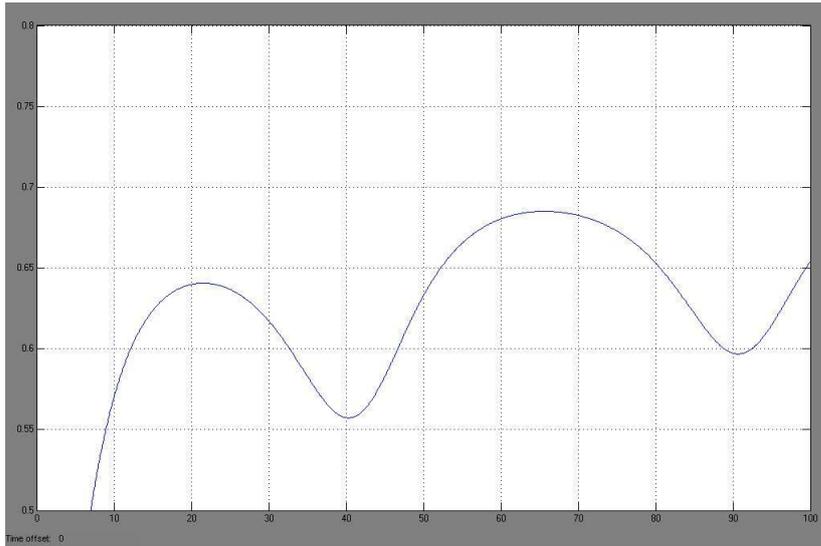
In case 10 we are changing the parameter values of the model and we observe differences in the concentration of the active PKC

Sub case 10.1

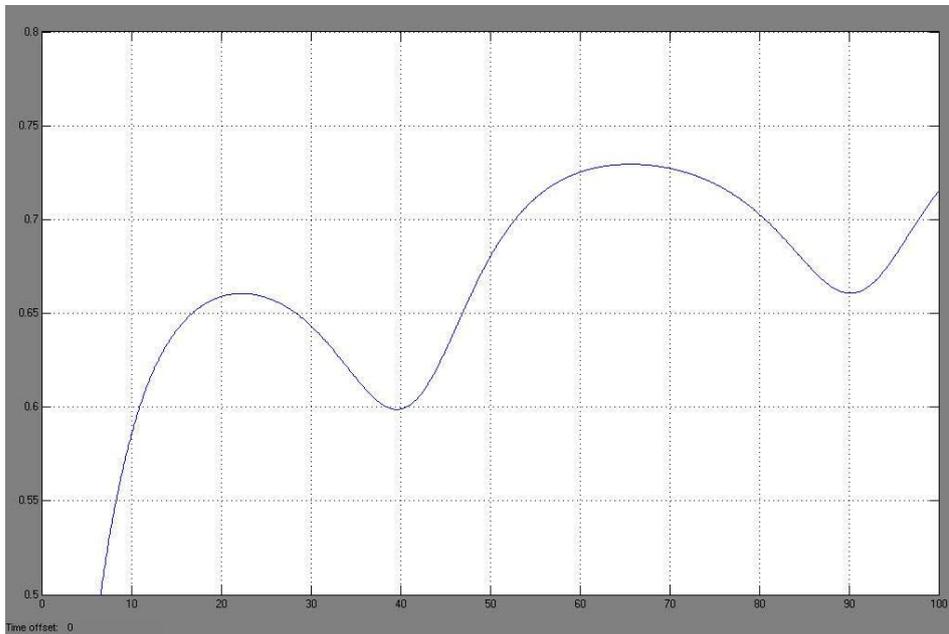
Plot A: Sine wave Ca²⁺ stimulus, time period 50 s,



plot B original [PKCa],



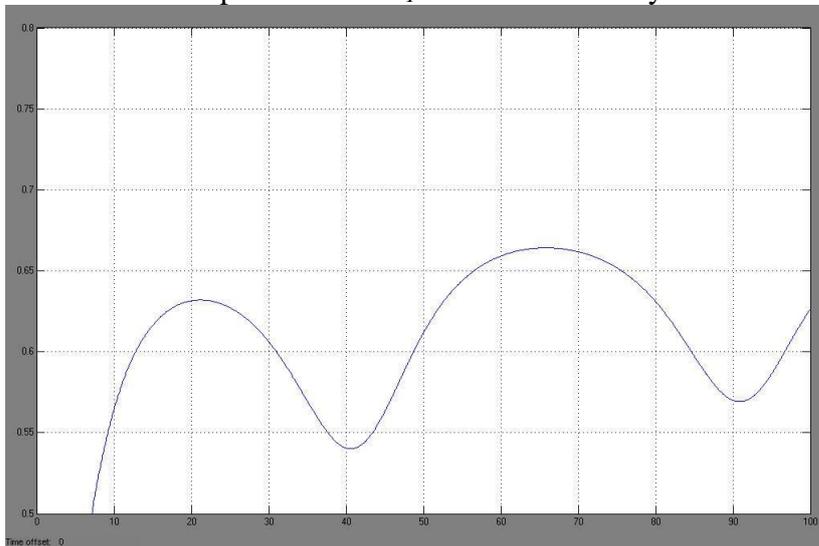
Plot C: PKCa response when kI value increased by 20 % in reaction R6,



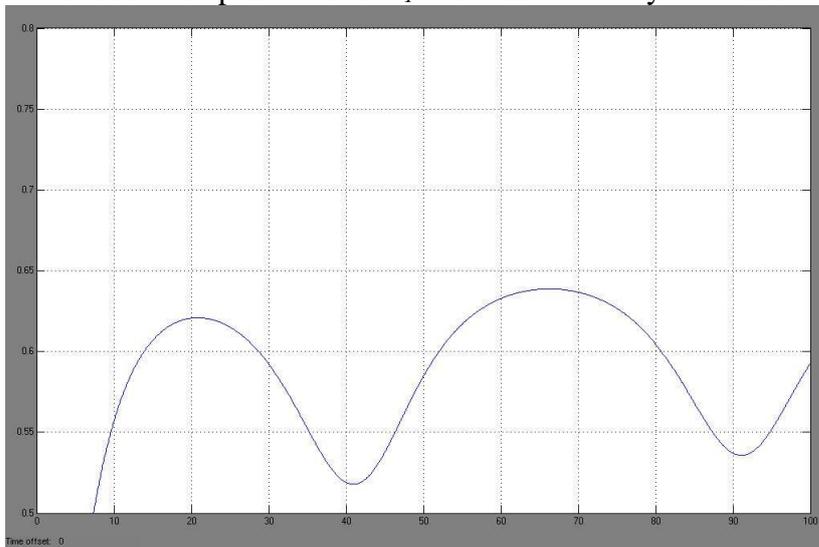
Plot D: PKCa response when kI value increased by 50 % in reaction R6,



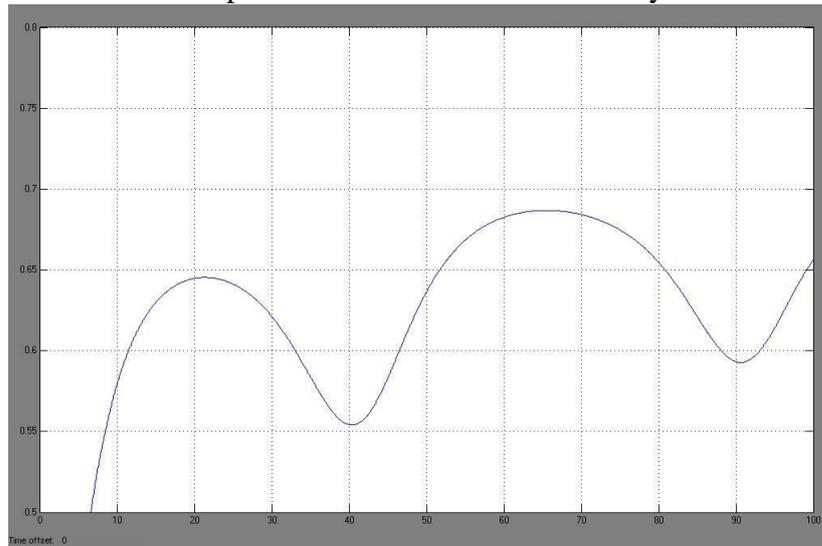
Plot E: PKCa response when k_{-1} value increased by 20 % in reaction R6,



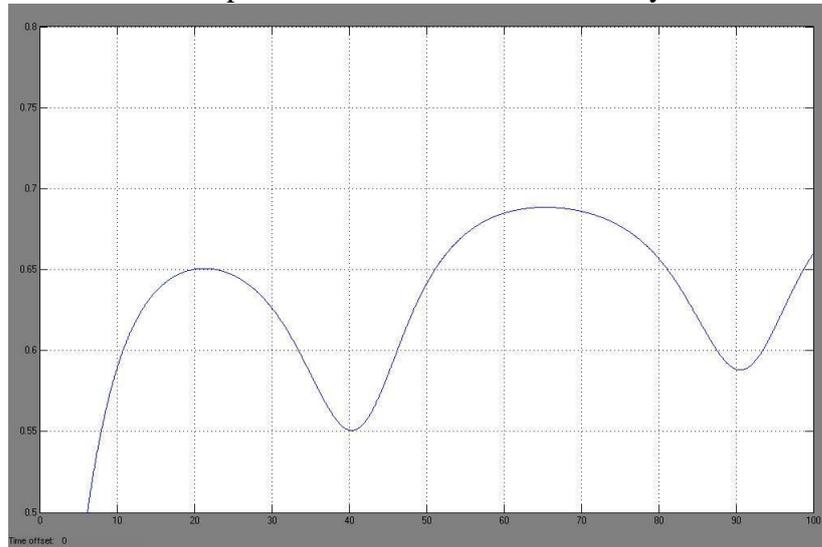
Plot F: PKCa response when k_{-1} value increased by 50 % in reaction R6.



Plot G: PKCa response when k_I value increased by 20 % in reaction R7,



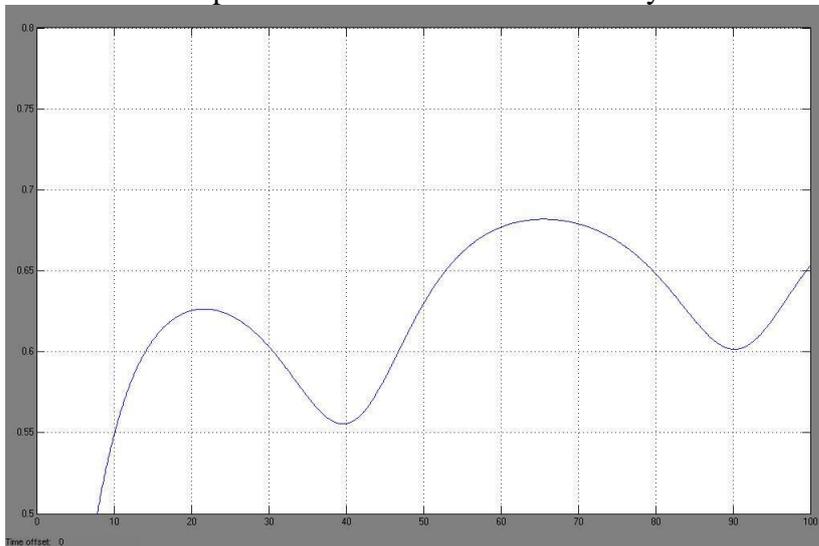
Plot H: PKCa response when k_I value increased by 50 % in reaction R7.



Plot I: PKCa response when k_{-I} value increased by 20 % in reaction R7,



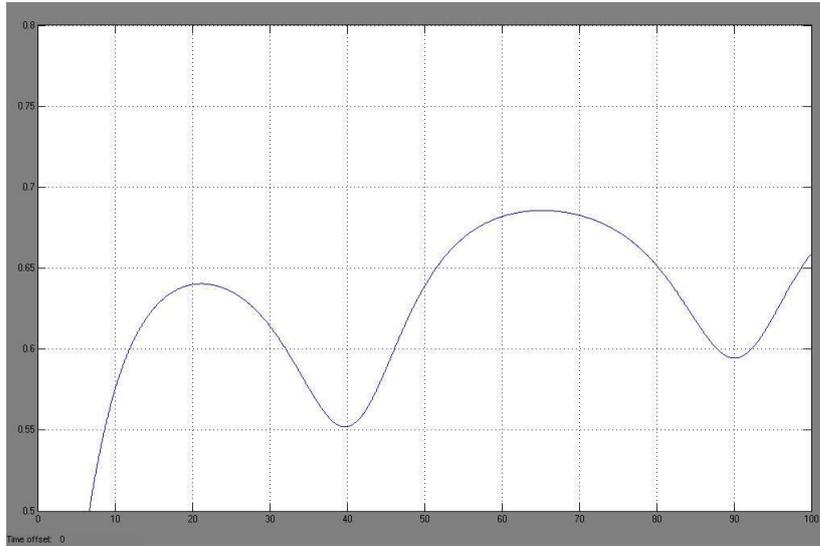
Plot J: PKCa response when $k-1$ value increased by 50 % in reaction R7.



Plot K: PKCa response when $k1$ and $k-1$ values increased by 20 % in reaction R6 and R7,



Plot L: PKCa response when $k1$ and $k-1$ values increased by 50 % in reaction R6 and R7.



Simulation results and conclusion

As said in previous chapter we studied the protein kinase c pathway developed by Bhalla and Iyengar [3]. The model describing the PKC pathway was obtained from the database of Quantitative Cellular Signaling (<http://doqcs.ncbs.res.in/>). In the simulation that we made, we studied the effects of different kind of stimuli in the inputs as well as changes at parameter values. The results show a clear effect of the PKC's active concentration due to changes of Ca^{++} , AA and DAG. In particular, sine wave stimuli for Ca^{++} component, induces oscillating PKC activity. When two sine wave stimuli have the same time period but different concentration levels, also the behavior of [PKCa] is different. If the Ca^{++} stimulus attains 0uM during the simulation, the amplitude of [PKCa] is larger compared to the sine wave stimulus which does not attain 0uM during the simulation. Simulations also show that the longer the time period of Ca^{++} stimulus, the larger the amplitude of [PKCa]. When all inputs are sine waves, the model output follows the Ca^{++} stimulus in most cases. However, we find that the larger the amplitudes of AA and DAG, the clearer their effect on [PKCa].

The model simulation was implemented by using the Matlab simulation tool extended by Simulink, a particular script language used to simulate biochemical pathways.

Background of neural networks

General

There is no precise agreed-upon definition among researchers as to what a neural network is, but most would agree that it involves a network of simple processing elements (neurons), which can exhibit complex global behaviour, determined by the connections between the processing elements and element parameters. The original inspiration for the technique was from examination of the central nervous system and the neurons (and their axons, dendrites and synapses) which constitute one of its most significant information processing elements. In a neural network model, simple nodes (called variously "neurons") are connected together to form a network of nodes — hence the term "neural network." While a neural network does not have to be adaptive per se, its practical use comes with algorithms designed to alter the strength (weights) of the connections in the network to produce a desired signal flow.

There are several types of neural networks used in control systems. The choice of the appropriate network and training method depends on the application. For instance, feedforward multilayer neural network, where no information is feedback during operation. However, there is feedback information during training. Also, supervised learning methods, where the neural network is trained to learn input-output patterns presented to it, are used. This process is slow and time consuming, because the algorithm takes a long time to converge. Moreover other methods as Backpropagation (BP) algorithm, which adjusts the weights during training, or recurrent networks are used.

Theoretical studies have proved that multilayer neural networks with one hidden layer can approximate any continuous function uniformly over a compact domain by adjusting synaptic weights in order to minimize the error between the network output and the output of the unknown map [27], [28], [29], [30].

Forward modelling is the training of a neural network to model the forward dynamics of a plant. The neural network model is placed in parallel with the plant and the error between the plant and the network outputs is the training signal. The training

procedure may needs discrete samples of the plant inputs and outputs. If we consider that the system output at time $k+1$ depends on the past n output values and the past m values of input u , we have:

$$y^p(k+1) = f(y^p(k), \dots, y^p(k-n+1); u(k), \dots, u(k-m+1))$$

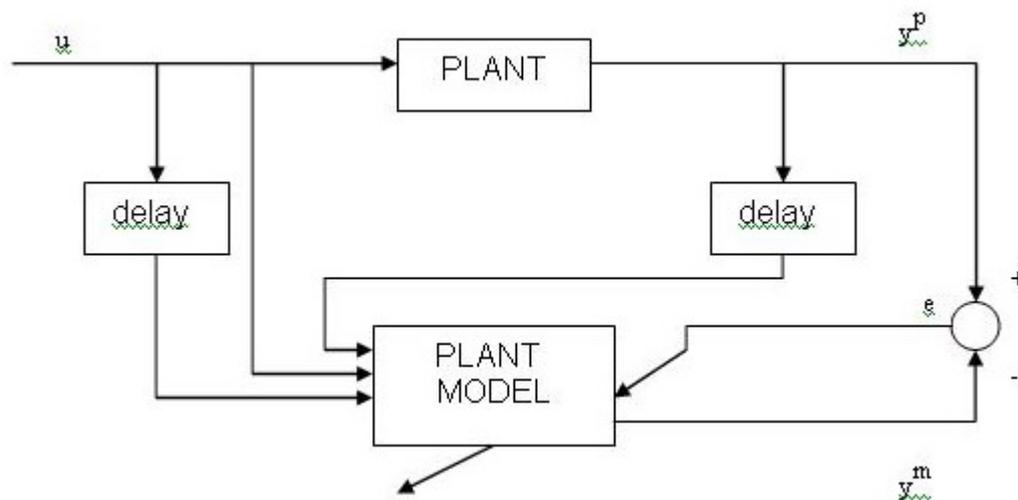
Thus, the output of the neural network is:

$$y^m(k+1) = f_{apr}(y^p(k), \dots, y^p(k-n+1); u(k), \dots, u(k-m+1))$$

f_{apr} represents the nonlinear input-output map of the network, or the approximation of. It is clear that the network's input includes the past values of the real system's output. In other words the system has not feedback. After the training, the network approximates the plant or $y^m \approx y^p$. When this is true, the network's output and the delay values can be fed back and be part of the network's input. In this way, the network can be used independently of the plant and the model function can be written:

$$y^m(k+1) = f_{apr}(y^m(k), \dots, y^m(k-n+1); u(k), \dots, u(k-m+1))$$

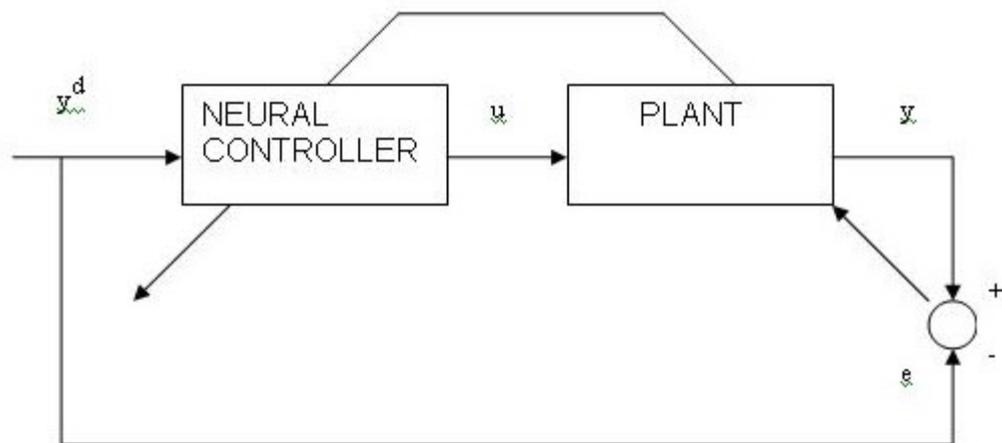
The information about the plant can be in the form of an input-output table. In this case the training of the network necessitates current and previous inputs or outputs of the plant. Alternatively the states of the plant or their derivatives can be used. Consequently, for the case of feedforward multilayer neural network and BP training algorithm we consider discrete or discretized continuous plant, as is described in Figure.



Another approach of training aims to identify the inverse dynamics of the plant. In this case the network's input is the plant's output and the plant's input is the network's output. The training signal is the error between the actual input of the plant

and the network's output. The current input of the plant is the desired output of the network. We have assumed that the inverse of the plant is unique. If the inverse is not unique, we must restrict the ranges of the input to the network.

The architecture of the network is chosen appropriately according to the case. The first step is the identification of the plant. Then, a controller can be designed. In Figure the training of a neural network as open loop controller is described. The error $e = y - y_d$ is used in order to train the network. As we can see the error is backpropagated through the plant.



We must add that the neural model of a controller can include mean squared error between the reference output and model output or other terms as the error between the reference input and real output and the input u .

In most applications we meet dynamical systems which necessitate the use of feedback connections in order to approximate them. Such networks are known as recurrent. A static neural network can also be transformed to a dynamic one, by simply connecting the past neural outputs as inputs to the neural network, thus making the neural network a very complicated and highly nonlinear dynamical system.

The main difficulty with the dynamic neural networks that are based on static multilayer networks is that the synaptic weights appear nonlinearly in their mathematical representation. This creates a number of significant hurdles. Firstly, the learning laws that are used require a high computational time. Secondly, since the synaptic weights are adjusted to minimize a functional of the approximation error and the weights appear nonlinearly, the functional has many local minima so there is no way to ensure the convergence of the weights to the global minimum. Moreover, due to the highly nonlinearity of the neural network architecture, basic properties like

stability, convergence and robustness are very difficult to verify. On the other hand for the recurrent neural networks that possess a linear-in-the weights property, is feasible to prove the stability and convergence properties.

In this chapter Recurrent High-Order Neural Networks' (RHONN) structure is introduced and their approximation capabilities are analyzed. This network scheme approximates nonlinear systems whose vector fields satisfy a local Lipschitz condition.

Identification of dynamical systems using Recurrent High-Order Neural Networks

Recently there is renewed interest in the usage of neural networks for modelling and identification of dynamical systems in the form of feedback connections, most known as recurrent neural networks (RNN). Several training methods are used. For example, recurrent backpropagation [6], backpropagation-through-time algorithms[7], real-time recurrent learning algorithm[8] and the dynamic backpropagation[9] algorithm. The last method is based on the computation of sensitivity models for generalized neural networks. Generalized neural networks combine feedforward neural networks and dynamical components of stable rational transfer functions. All these training methods have been widely used in empirical studies and had many drawbacks. First of all they rely on the approximation of computing a partial derivative. Moreover these methods need much computational time. Also, it is very difficult to produce analytical results for the convergence and stability.

An interesting effort is the design of training methods based on the Lyapunov stability theory [10], [11], [12], [13], [14], [15], [16], [17], [18]. These methods have the advantage of stability, convergence and robustness proofs which promotes control theory.

RHONNs are appropriate for identification models. High-order networks are expansions of the first-order Hopfield [19] and Cohen-Grossberg [36] models which allow higher-order interaction between neurons. Their superior storage capacity has been demonstrated in [20], [21], and their stability properties for fixed-weight values have been studied in [37], [22]. Furthermore, several authors have

demonstrated the feasibility of using these architectures in applications such as grammatical inference [23] and target detection [24]. In [18] was introduced the idea of recurrent neural networks with dynamical components distributed throughout the network in the form of dynamical neurons and their application for identification of dynamical systems. In this chapter, we combine distributed recurrent networks with high-order connections between neurons. The next section shows that recurrent high-order neural networks can model a large class of dynamical systems. Specifically, it is proven that if enough higher-order connections are allowed in the network then there exist weight values such that the input-output behaviour of the RHONN model approximates that of an arbitrary dynamical system whose state trajectory remains in a compact set.

The RHONN Model

Recurrent neural network models have two way connectivity between units (or neurons). On the other hand, feedforward neural networks have the output of one unit connected only to units of the next layer. In the simplest case, the state history of each neuron is represented by the following form:

$$\dot{x}_i = -a_i x_i + b_i \sum_j w_{ij} y_j$$

where x_i , is the state of the i -th neuron, a_i , b_i are constants, w_{ij} are synaptic weights connecting the j -th input to the i -th neuron and y_j is either an external input or the state of a neuron passed through a sigmoid function ($y_j = s(x_j)$), where $s(\cdot)$ is the sigmoidal.

With respect to the order of a RNN (k) the input to the neuron contains the product:

$$\underbrace{y_i \dots y_j}_{k_times}$$

For instance, in a second order RNN the input to the neuron is a linear combination of not only y_j but also of $y_j y_k$

Now we consider a RHONN consisting of n neurons and m inputs. The state of each neuron is represented by:

$$\dot{x}_i = -a_i x_i + b_i \left[\sum_{k=1}^L w_{ik} \prod_{j \in I_k} y_j^{d_j(k)} \right]$$

where I_1, I_2, \dots, I_L is a collection of L not-ordered subsets of $\{1, 2, \dots, m+n\}$, a_i, b_i are real coefficients, w_{ik} are synaptic weights of the neural network and $d_j(k)$ are non-negative integers. The state of the i -th neuron is again represented by x_i and $y = [y_1, y_2, \dots, y_{m+n}]^T$ is the input vector to each neuron defined by:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_n \\ \cdot \\ \cdot \\ y_{n+m} \end{bmatrix} = \begin{bmatrix} s(x_1) \\ s(x_2) \\ \cdot \\ \cdot \\ s(x_n) \\ u_1 \\ \cdot \\ \cdot \\ u_m \end{bmatrix}$$

where $u = [u_1, u_2, \dots, u_m]^T$ is the external input vector to the network. The function $s(\cdot)$ is monotone-increasing, differentiable usually represented by sigmoidals of the form: $s(x) = \frac{\alpha}{1 + e^{-\beta x}} - \gamma$ where α, β are the bound and slope of sigmoid's curvature and γ a bias constant. When $\alpha = \beta = 1, \gamma = 0$, we obtain the logistic function and by setting $\alpha = \beta = 2, \gamma = 1$, we obtain the hyperbolic tangent function; these are the sigmoid activation functions most commonly used in neural network applications.

We now introduce the L -dimensional vector z , which is defined as:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \cdot \\ \cdot \\ z_L \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} y_j^{d_j(1)} \\ \prod_{j \in I_2} y_j^{d_j(2)} \\ \cdot \\ \cdot \\ \prod_{j \in I_L} y_j^{d_j(L)} \end{bmatrix}$$

Therefore, the RHONN model (2.2) becomes

$$\dot{x}_i = Ax + W^T z \quad (2.6)$$

Moreover, if we define the adjustable parameter vector as $w_i = b_i[w_{i1}, w_{i2}, \dots, w_{iL}]^T$ then (2.6) becomes $\dot{x}_i = -a_i x_i + w_i^T z$ where vectors $\{w_i : i=1, 2, \dots, n\}$ represent the adjustable weights of the network and the coefficients $\{a_i : i=1, 2, \dots, n\}$ are fixed during training. In order to guarantee that each neuron x_i is bounded-input bounded-out-put (BIBO) stable, we shall assume that $a_i > 0, \forall i=1, 2, \dots, n$. In the special case of a continuous time Hopfield model [19], we have $a_i = 1/(R_j C_j)$, where $R_j > 0$ and $C_j > 0$ are the resistance and capacitance connected at the i-th node of the network respectively.

In vector form the dynamic behaviour of the overall network is described by:

$$\dot{x} = Ax + W^T z \quad (2.8)$$

Where $x = [w_1, w_2, \dots, w_n]^T \in R^n$, $W = [w_1, w_2, \dots, w_n]^T \in R^{L \times n}$ and $A = \text{diag}(-a_1, -a_2, \dots, -a_n)$ a $n \times n$ diagonal matrix. Since $a_i > 0 \forall i=1, 2, \dots, n$, A is a stability matrix. Vector z is a function of the state x and the external input u .

Identification of Protein Kinase C using Reccurent High Order Neural Networks

In this section we represent the simulation results of our protein kinase C's identification. The efficiency of the identification procedure, depends mainly on the following:

- The error convergence of abrupt input changes
- Stability in cases of abrupt input changes
- Performance of the identification model after the training stops

Below are represented the robust learning algorithms:

System model:

$$\dot{\chi} = F(x, u),$$

Parametric model:

$$\dot{\chi}_i = -a_i \chi_i + w_i^{*T} z - v_i(t),$$

RHONN identifier model:

$$\dot{x}_i = -a_i x_i + w_i^T z,$$

Identifier Error:

$$e_i = x_i - \chi_i,$$

Weight estimation error:

$$\varphi_i = w_i - w_i^*,$$

Modeling error:

$$F_i(x(t), u(t)) + a_i x_i(t) - w_i^{*T} z(x(t), u(t)),$$

Robust learning algorithms:

a) switching σ -modification:

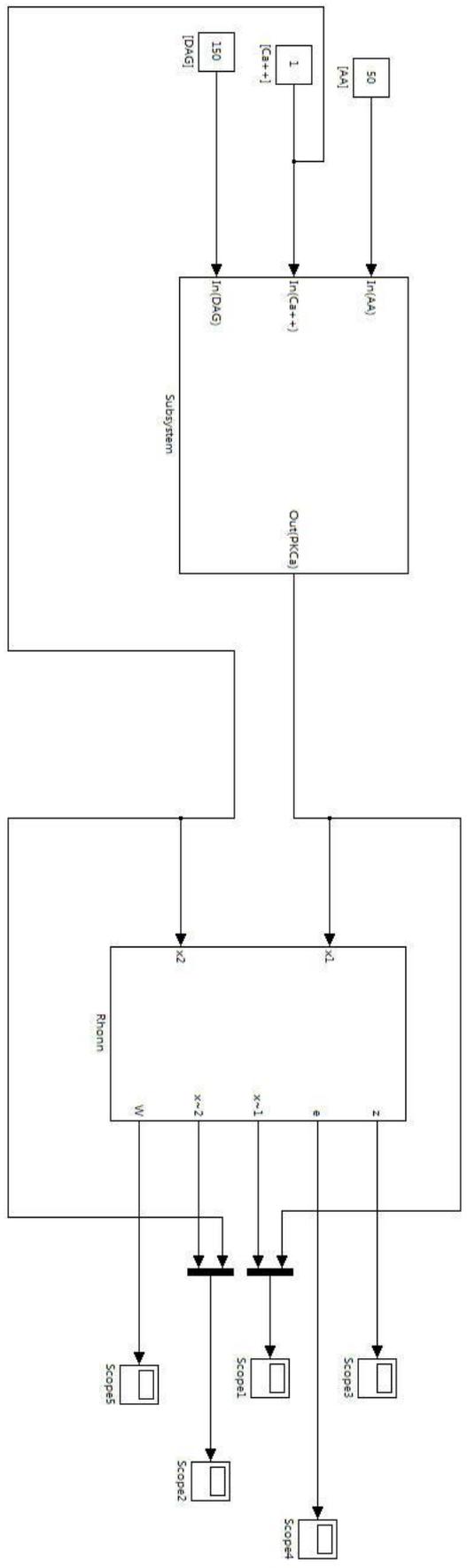
$$\dot{w} = \begin{cases} -\Gamma_i z e_i, & \text{if } |w_i| \leq M_i \\ -\Gamma_i z e_i - \sigma_i \Gamma_i w_i, & \text{if } |w_i| > M_i \end{cases}$$

b)

$$\dot{w} = \begin{cases} -\Gamma_i z e_i, & \text{if } \{|w_i| < M_i\} \text{ or } \{|w_i| = M_i\} \text{ and } \{w_i^T \Gamma_i z e_i > 0\} \\ -\Gamma_i z e_i + \frac{w_i^T \Gamma_i z e_i}{w_i^T \Gamma_i w_i} \Gamma_i w_i, & \text{if } \{|w_i| = M_i\} \text{ and } \{-\sigma_i w_i^T \Gamma_i w_i \leq w_i^T \Gamma_i z e_i \leq 0\} \\ -\Gamma_i z e_i - \sigma_i \Gamma_i w_i, & \text{if } \{|w_i| > M_i\} \text{ or } \{|w_i| = M_i\} \text{ and } \{w_i^T \Gamma_i z e_i < \sigma_i w_i^T \Gamma_i w_i\} \end{cases}$$

$$x \in \mathfrak{R}^n, u \in \mathfrak{R}^m, i=1, 2, \dots, n$$

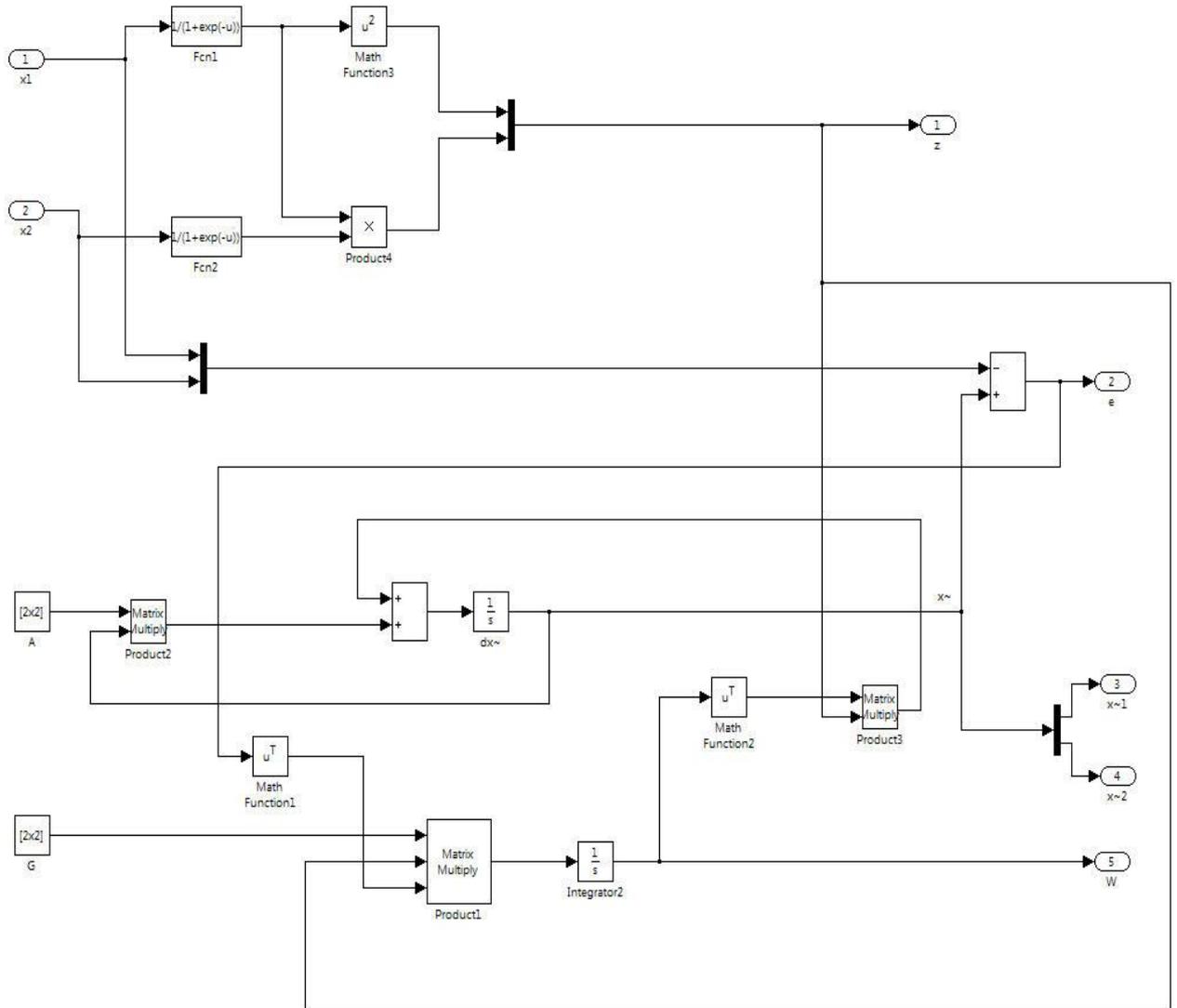
The block diagram of our ‘‘Rhonn-PKC’’ is represented in the following picture:



The output of our actual system is the input to our neural network. In the above diagram we see that the inputs of our actual system (that is [AA], [Ca⁺⁺], [DAG]) are held constants. The values of our concentration are represented in the following table:

Input	Concentration(uM)
[AA]	50
[Ca ⁺⁺]	1
[DAG]	150

The actual system is being fed up to the rhonn which is shown in the following picture:



X1 and X2 are the inputs of the rhonn model, A is a 2x2 diagonal matrix A and has

values: $A = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$

and matrix G: $G = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$

The outputs of the rhonn model are e (the error of the system that has to go to zero), z, x~1 and x~2 are the neural's outputs and w are the weights of the system that have to settle into a specific value.

From theory to action – The scopes of the model

Configuration parameters:

- The systems simulation stop time is set to 10 seconds.
- The solver that we selected is ode23s (stiff/Mod. Rosenbrock)
- Type is variable step
- The step size (max and min) are set to auto
- And the relative tolerance is set to 1e-6

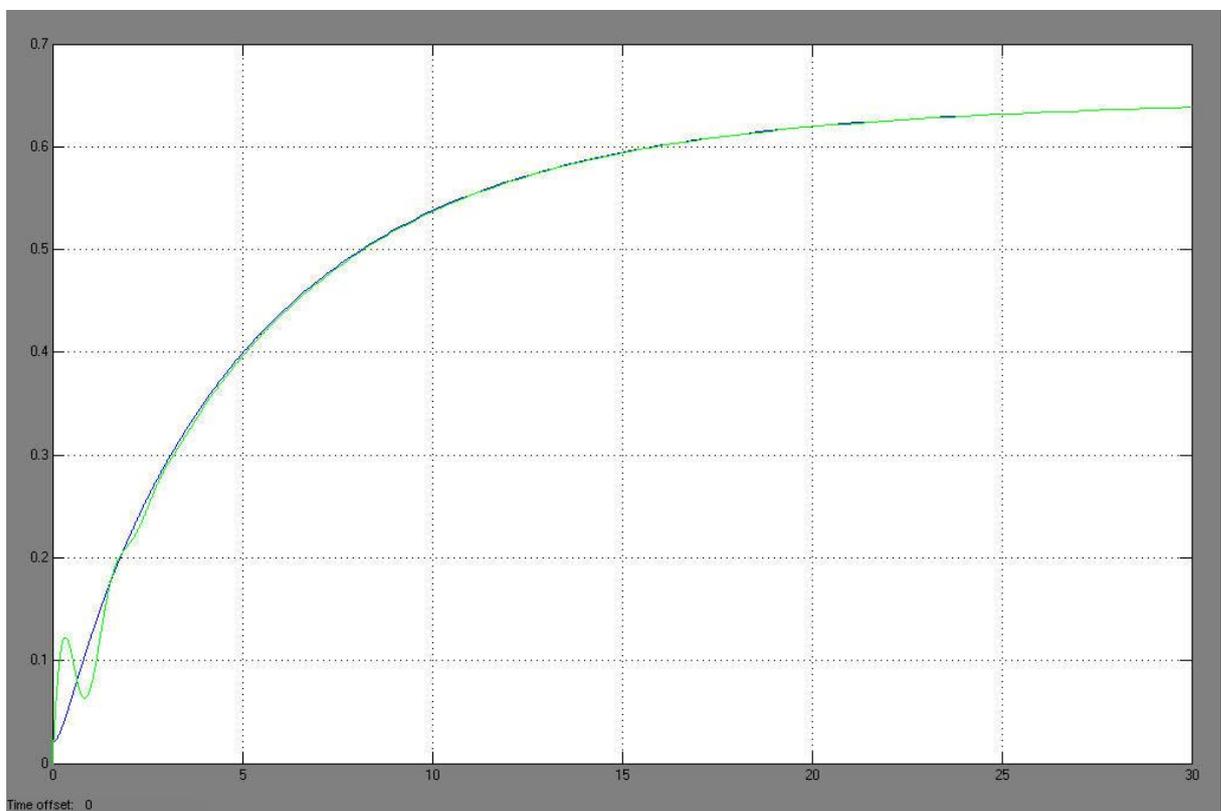
Scopes:

The desired scopes for our system are:

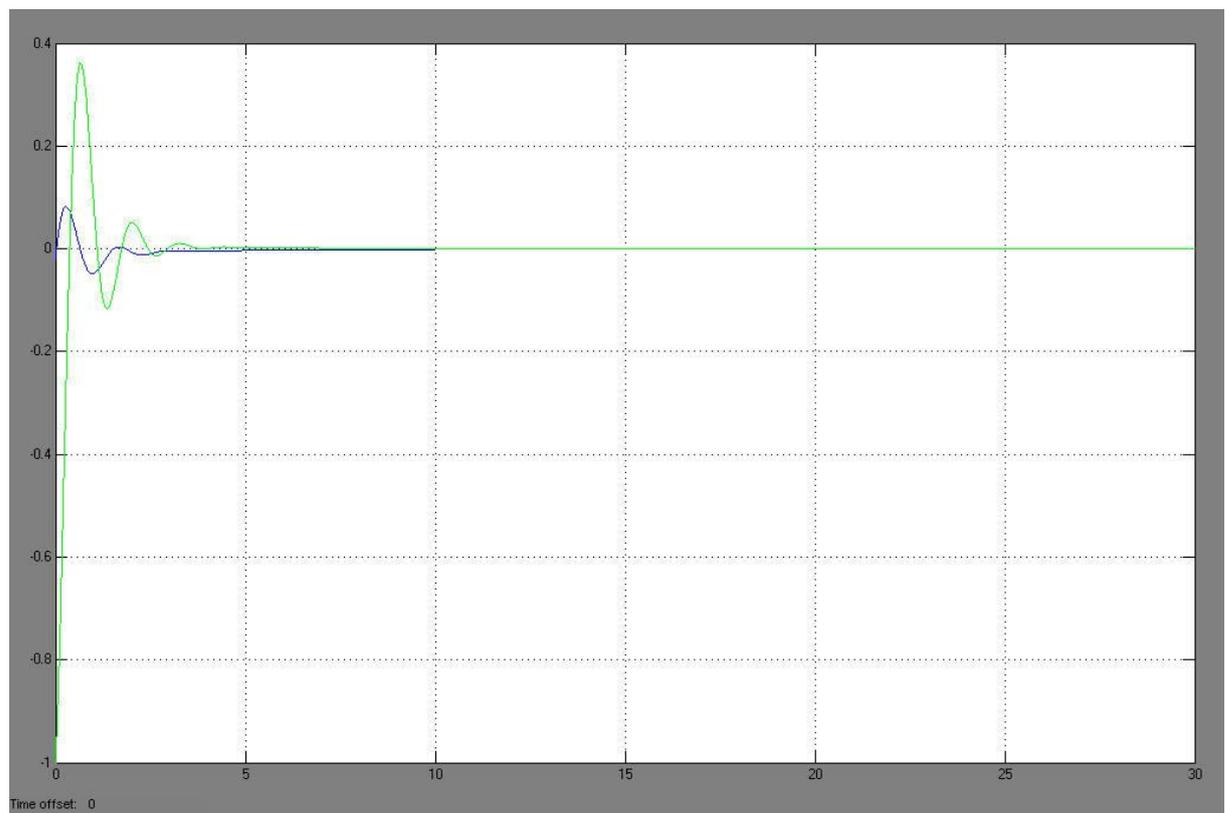
- The concentration of PKC in our real system must coincide after a while with the one of our rhonns
- The error of our system must be driven to zero
- The weights of our rhonn model must be stabilized to a certain value
- The z values also have to be stabilized to a certain value

According to the theory we have the following graphs:

In the first scope we see in blue the concentration of PKC and in green the output of our neural. At about 3 sec time, we see that the two graphs coincide.

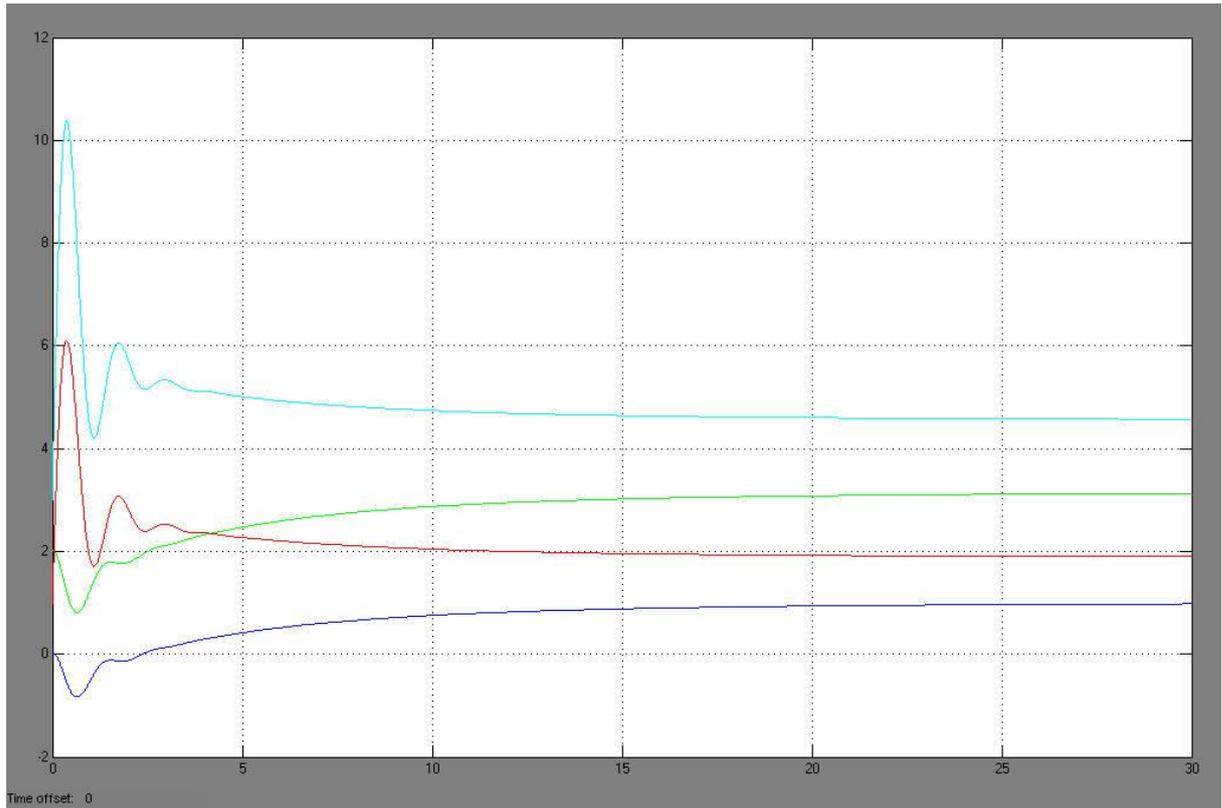


An alternative way of watching our neural network is by plotting the error:



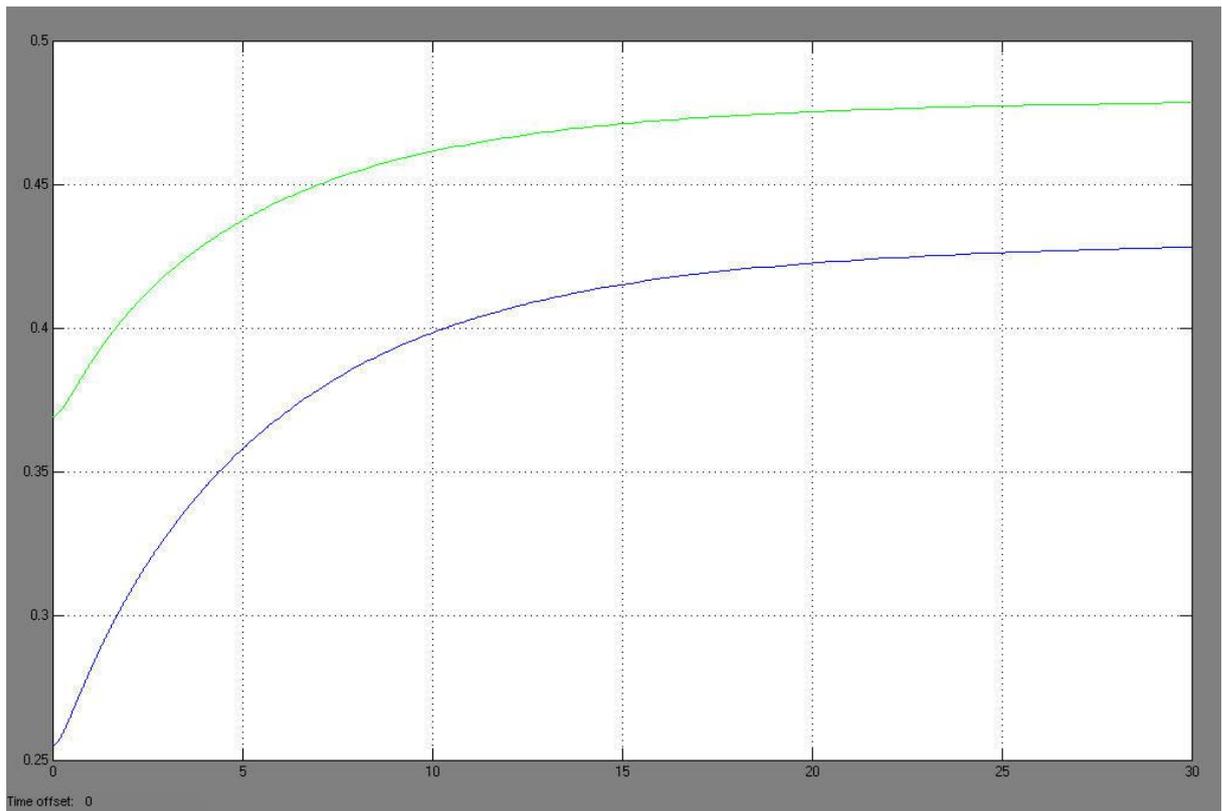
Again we see that at about 3 sec the error becomes zero.

At the next graph the four weights are represented:



As we see the curves of the weights for the first 3 sec's examine big changes and they become smaller as time goes on until they reach a steady state. From this point we are in position to know the neural network has trained itself to simulate the curve of PKC's optimal behavior.

At the forth graph the values of z are represented



It is useful to mention that the sigmoid function that we use is of the following form:

$$S(x) = \frac{1}{1 + \exp(-1 * u)}$$

where $\alpha=1$, $\beta=1$, $\gamma=0$ and $z = \begin{bmatrix} S(x_1)^2 \\ S(x_1)S(x_2) \end{bmatrix}$. We use x_1 to represent PKC active and x_2 for Ca^{++} .

The above neural identification of PKC's system is essential, if we want to use Neural Networks for extended purposes such as regulation. With this term we mean our aim to "drive" the final output trajectory to a specified steady state that suits our plans.

This task is given to Rhonns which already know how to handle the behavior of our studied protein

Direct Adaptive Control

Introduction

This chapter introduces direct adaptive control for affine in the control non-linear dynamical systems. In adaptive control there is estimation of unknown parameters at each instant and a control law is used. The objective is the approximation of the actual system by the model system. There are two basic approaches: direct and indirect.

In indirect adaptive method there is on-line estimation of the actual system parameters and then the controller parameters are calculated. In direct adaptive method the model system's parameters are estimated according to the controller parameters which are estimated directly without estimation of plant parameters.

In this chapter we use RHONNs. Also destabilizing factors as modelling errors are discussed. In this case the appropriate changes to update and control laws guarantee robustness and the uniform ultimate boundedness property. We focus on regulation issues and consider the more general case where the number of states is different from the number of control inputs.

Direct adaptive regulation for PKC ($n \neq m$)

The PKC model as said is implemented in MATLAB/Simulink simulation environment. To describe the system we have the following relationships:

Actual system: $\dot{x} = f(x) + G(x)u, x \in R^n, u \in R^m$

Model: $\dot{x} = -Ax + W^* S(x) + W_1^* S'(x)u$

Control law: $u^T = \frac{1}{2}ku^T - x^T W_1 S'(x)$

Update laws: $\dot{W} \begin{cases} -\frac{1}{2}kW - xS^T(x) \\ -\frac{1}{2}kW - xS^T(x) + \Lambda \end{cases}$

$\dot{W}_1 \begin{cases} -\xi x(S'(x)u)^T \\ -\xi x(S'(x)u)^T + P \end{cases}$

where:

$$\Lambda = tr\left\{\left(\frac{1}{2}kW + xS^T(x)\right)W^T\right\}\left(\frac{1+\|W\|}{w_m}\right)^2 W$$

$$P = tr\left\{\xi x(S'(x)u)^T W_1^T\right\}\left(\frac{1+\|w_1\|}{w}\right)^2 W_1$$

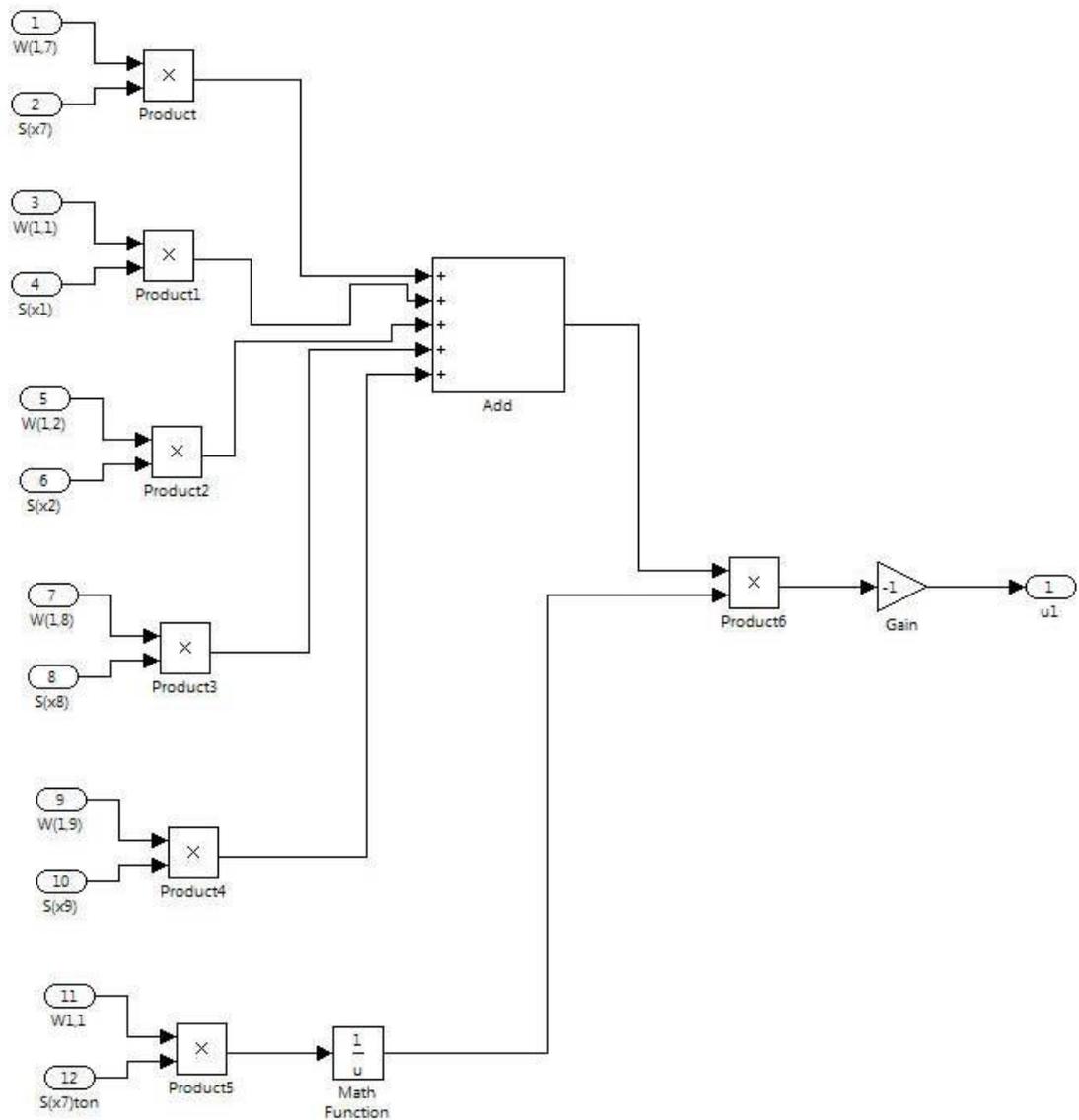
Filter: $\dot{\zeta} = -\kappa\zeta$
 $\xi = \zeta - h$
 $h = \frac{1}{2}(|x|^2 + |u|^2 + tr\{W^T W\})$

Results: $W_1 \in W', u, \xi \in L_\infty$
 $W \in W, |\zeta(t)| \leq |\zeta(0)|,$
 ξ, χ are u.u.b

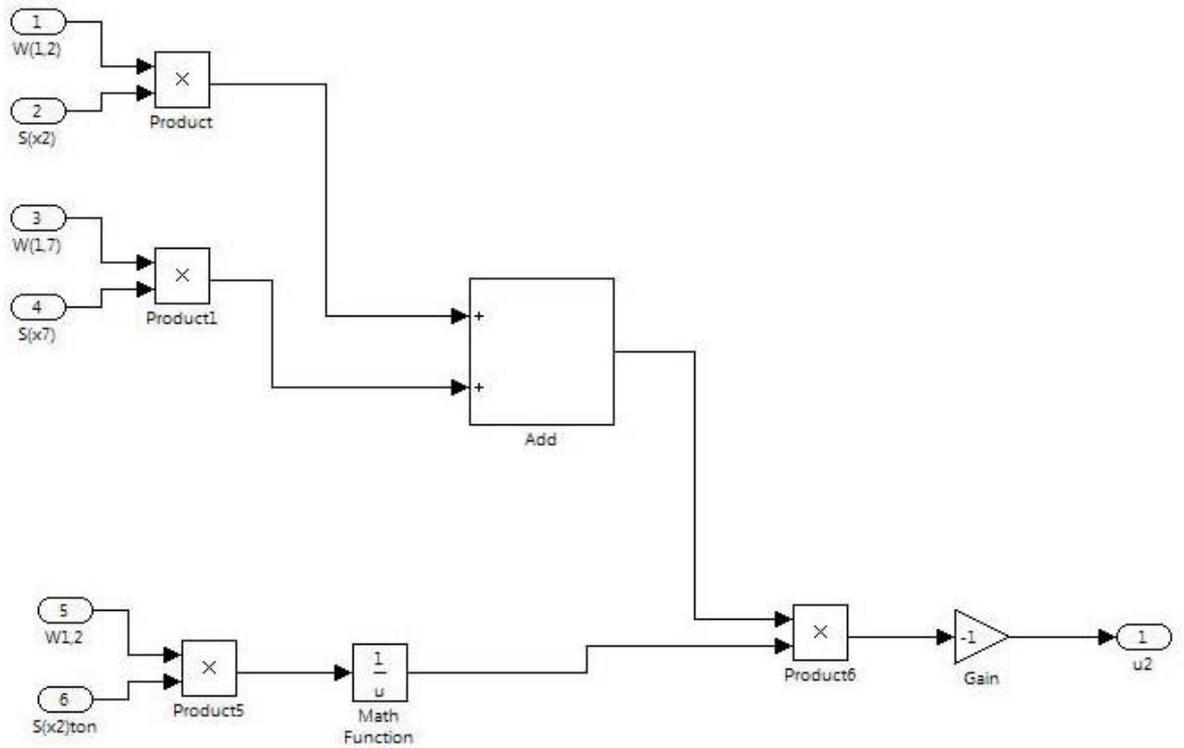
Requirements: $W_1^* \in W', W_1(0) \in W',$
 $W^* \in W, W(0) \in W,$
 $|\omega_0(x, u)| \leq k_1 + k_2 |x|, \zeta(0) \leq 0,$
 $|d_a| \leq \delta_\alpha, |d_m(t)| \leq \delta_m,$
 $\kappa > 4(s_1 \bar{\omega} \delta_m + k_2) > 0$

Design of the system for Direct Adaptive Control

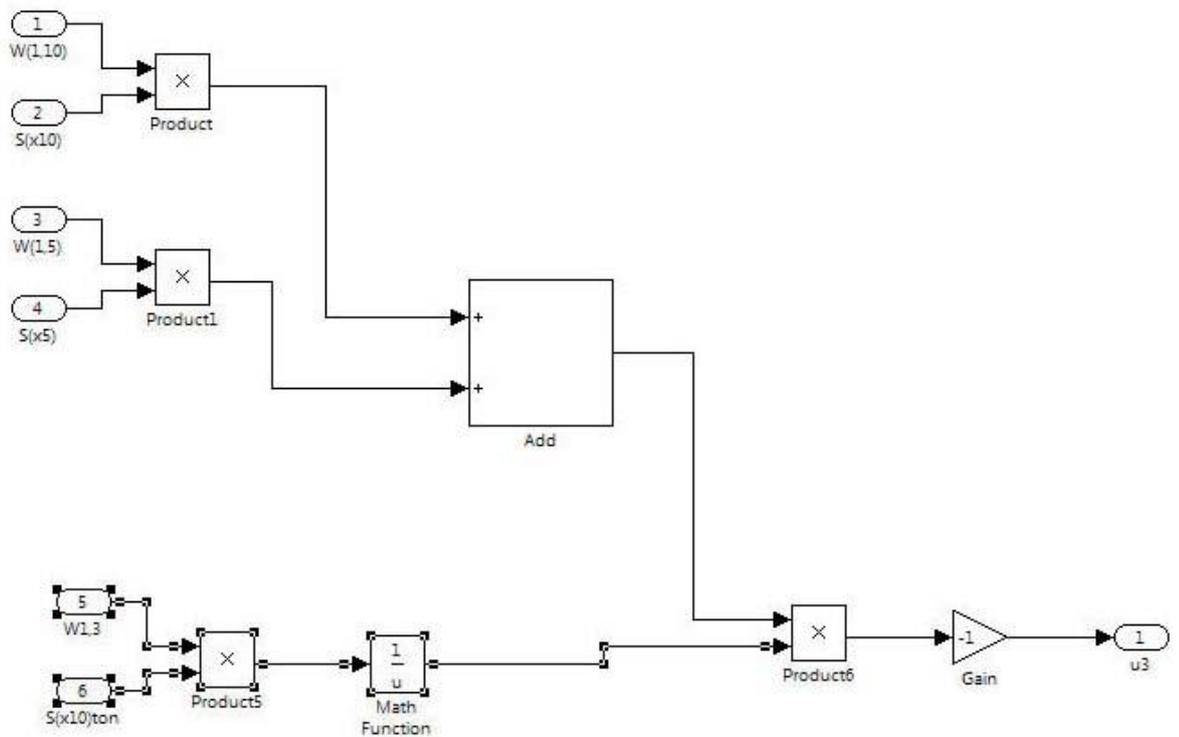
2. The three **control signals** that lead our systems inputs to desired values so that our output is led to zero.
 Control1 leads our PKC's input port Ca++ to a desired level



Control2 leads our PKC's input port AA to a desired level

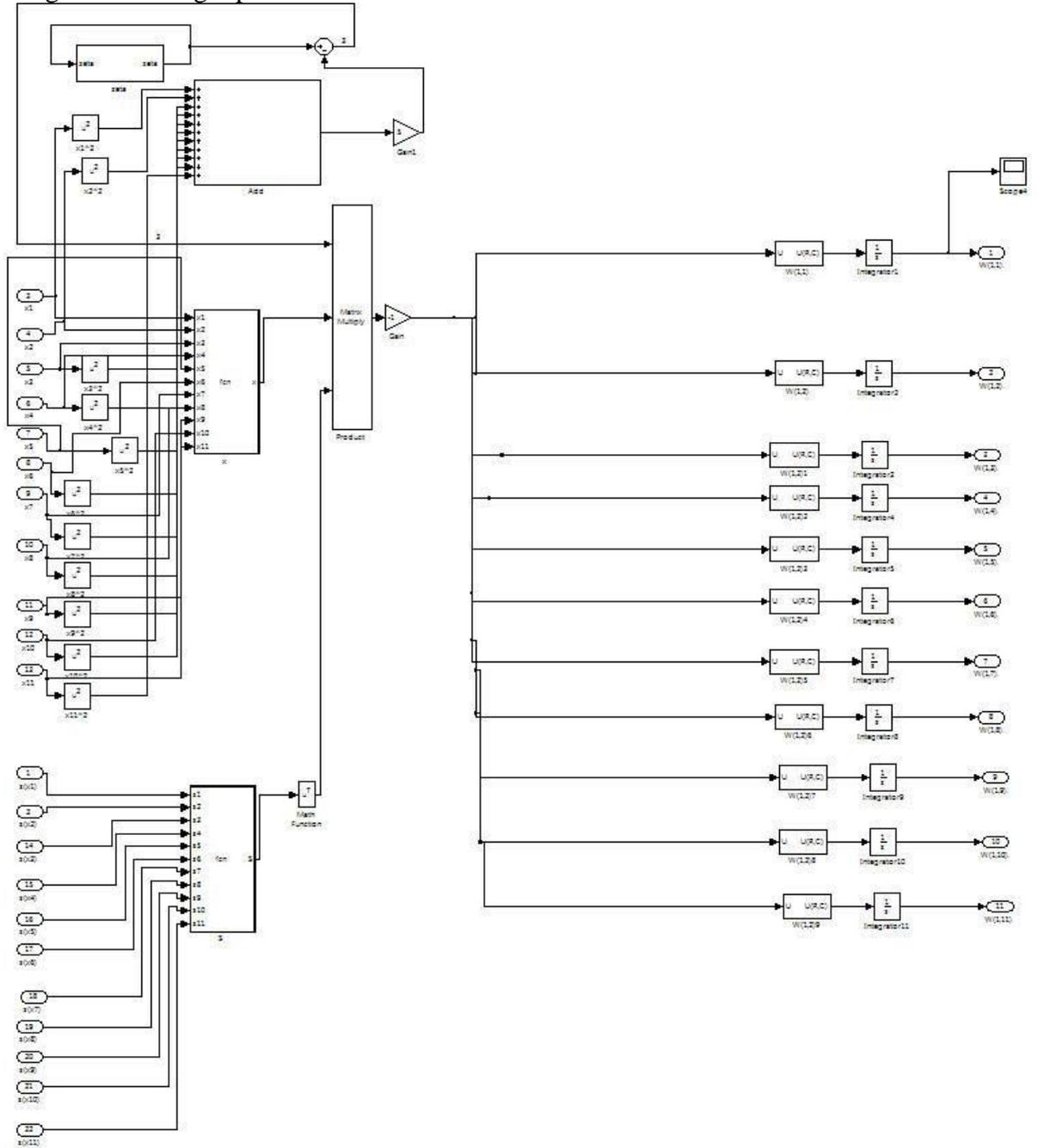


Control3 leads our PKC's input port DAG to a desired level

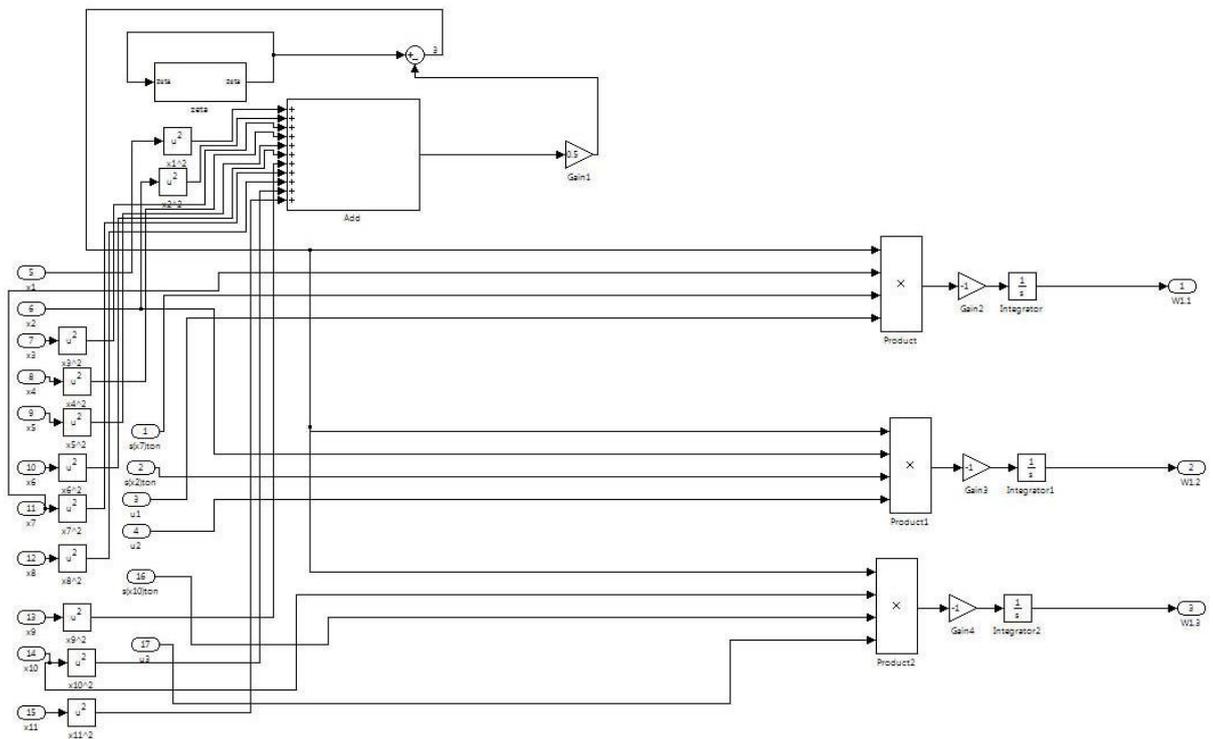


3. The two **weights** **W** and **W1** that by changing their values help the control signals to adjust the inputs.

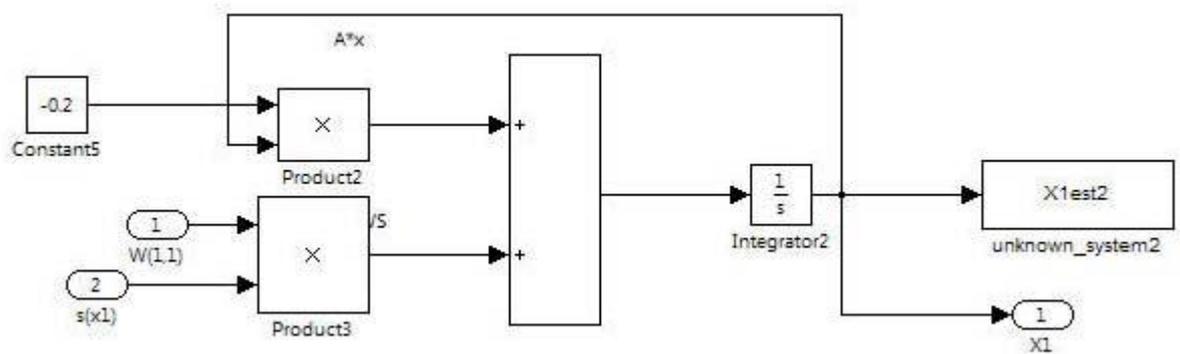
Weight W is being represented



Weight W1



4. The **outputs** of the system (we only represent X1 because they all look alike)



Simulation Results

The PKC model represented in previous chapter is used to test the performance of the neuroadaptive control algorithm developed to cover the case where the number of control inputs is not equal to the number of measurable states ($n \neq m$). The initial state of the weights W are shown below:

variable	Init. value
W(1,1)	0.0015
W(1,2)	-0.000834
W(1,2)1	-0.000834
W(1,2)2	-0.000834
W(1,2)3	-0.000834
W(1,2)4	-0.000834
W(1,2)5	-0.000834
W(1,2)6	-0.000834
W(1,2)7	-0.000834
W(1,2)8	-0.000834
W(1,2)9	-0.000834

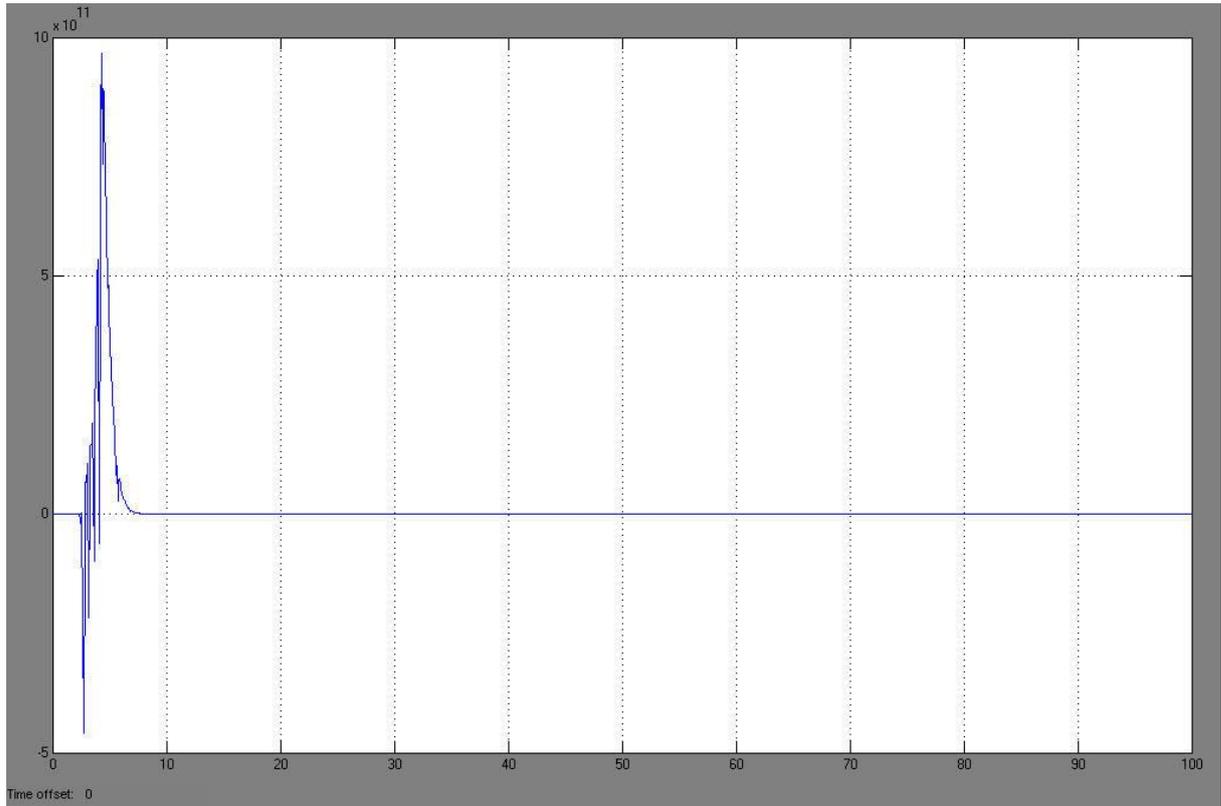
variable	Init. value
W1.1	1
W1.2	1
W1.3	1

constant	value
k	80
A	0.2

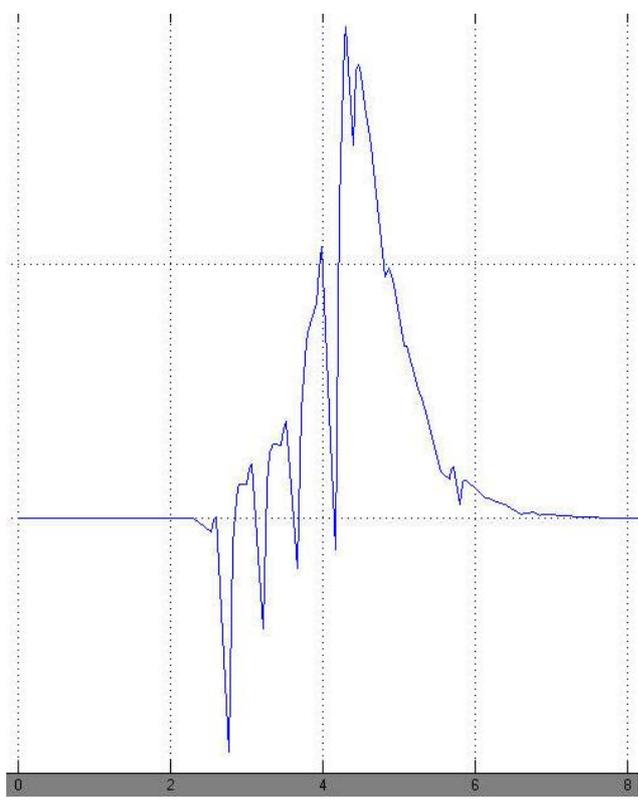
The sigmoid used was: $S(x) = \frac{1}{1 + \exp(-1 * u)}$ where $\alpha=1, \beta=1, \gamma=0$

and $S'(x) = \frac{1}{1 + \exp(-1 * u)} + 1$ where $\alpha=1, \beta=1, \gamma=-1$

The evolution of the PKC's concentration can be seen in the next graph. As we see it converges to zero very fast, at about below 9 msec.

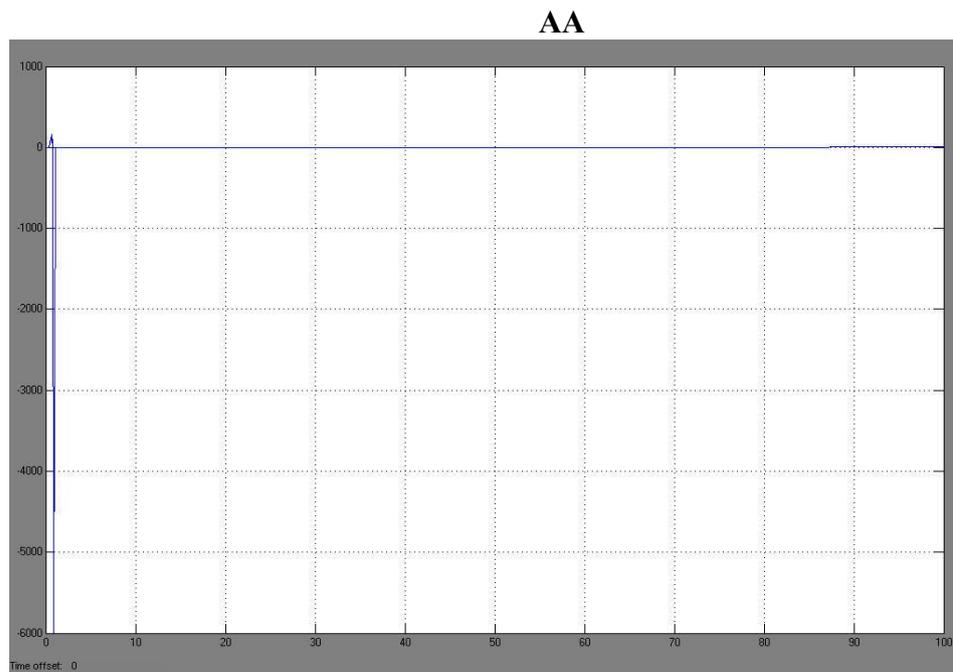


A more detailed representation of the oscillatory behavior for the regulated PKCa can be seen at the next graph.

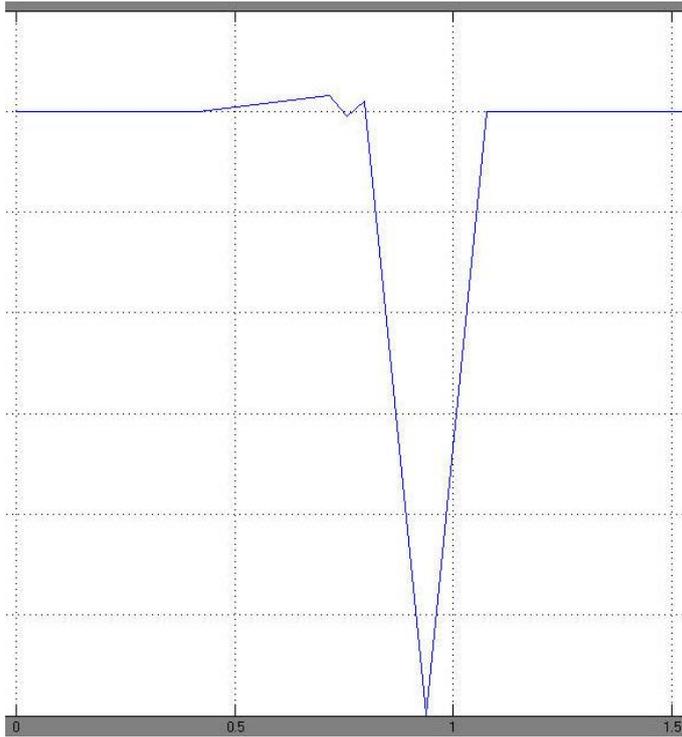


We notice that the oscillating behavior happens for a time period of almost seven seconds which is very small compared with the entire time that is needed for the whole regulation to finish.

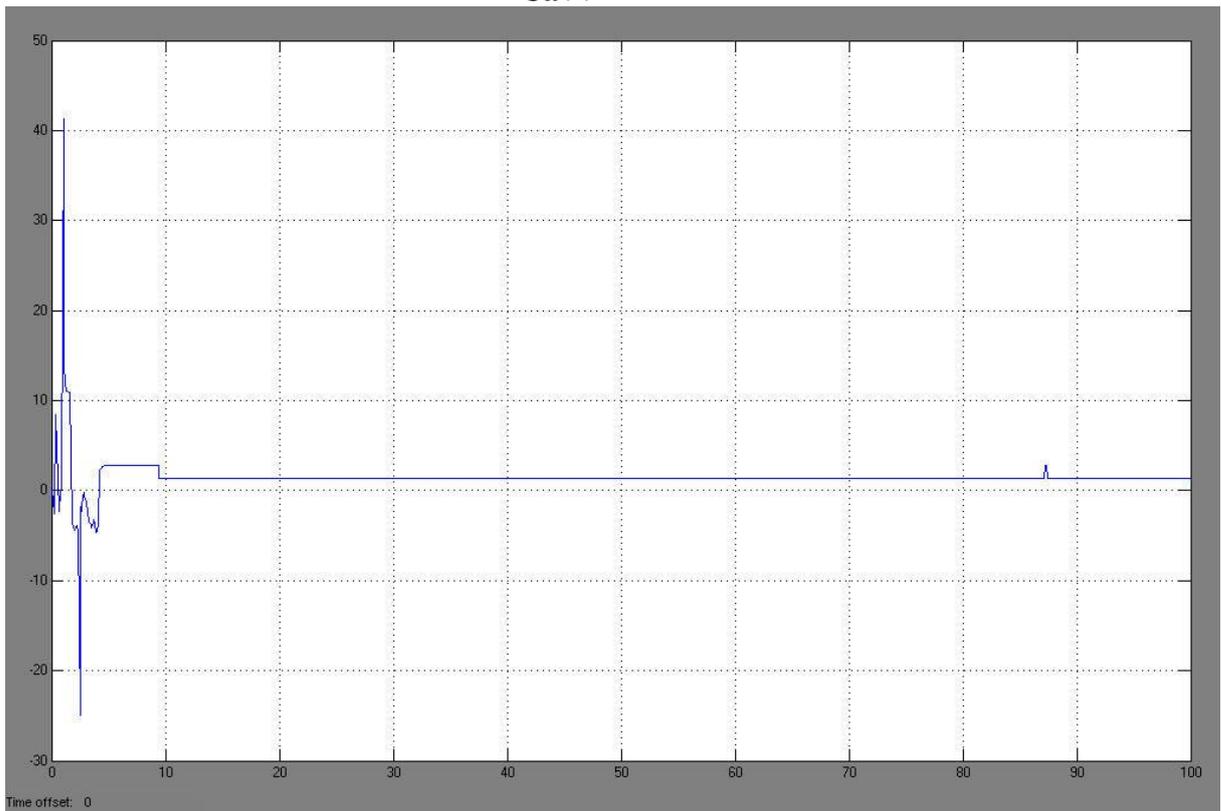
The three graphs below represent the concentration of AA, Ca⁺⁺ and DAG respectively



A more detailed representation of the oscillatory behavior for the regulated AA can be seen at the next graph. As we can see it stabilizes at zero.



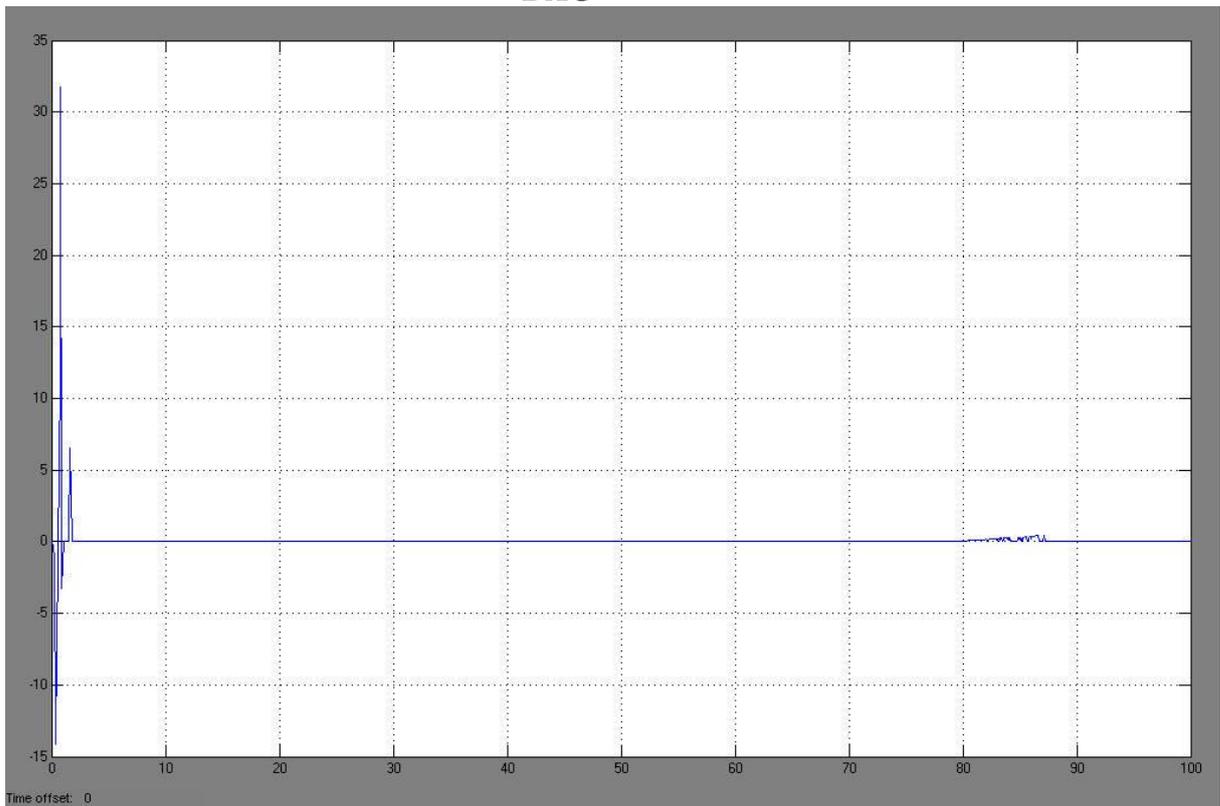
Ca⁺⁺



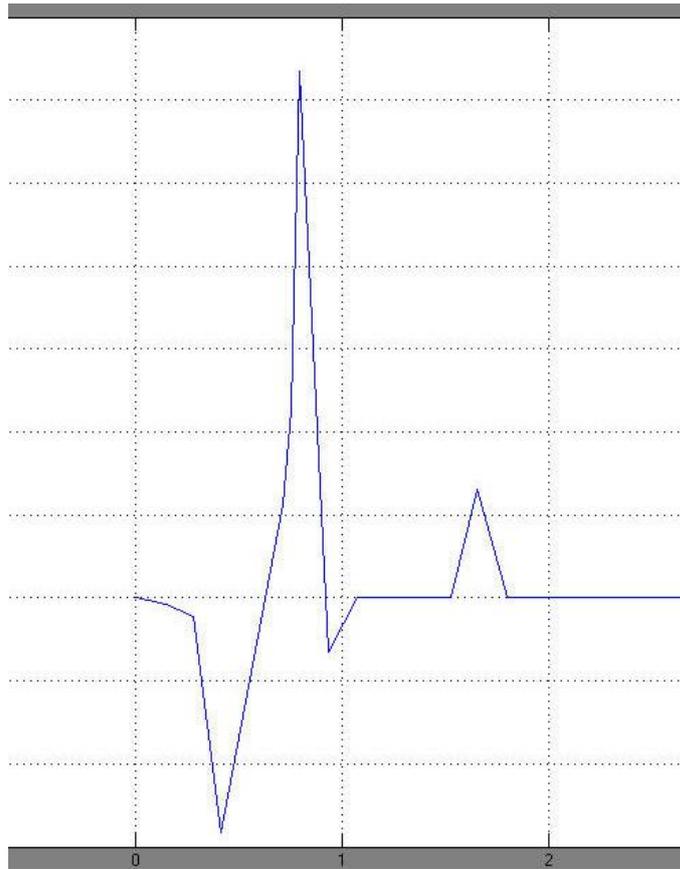
A more detailed representation of the oscillatory behavior for the regulated Ca⁺⁺ can be seen at the next graph. As we can see it stabilizes at 1.4 uM.



DAG



A more detailed representation of the oscillatory behavior for the regulated DAG can be seen at the next graph. As we can see it stabilizes at zero.



We notice from the above that all studied species fulfill the requirements we want to have for our regulation results. That is, the waveforms are bounded and they reach a steady state comfortably fast. Delays that are noticed are due to the complication of our system's ordinary differential equations (odes) which are big enough to justify them. Specifically, we have eleven differential equations (odes) with our three inputs (AA, CA++, DAG) involving to each one of them. So, it's logical to have an oscillating behavior at first, before the system is stabilized to a steady value.

FINAL CONCLUSIONS & FUTURE WORK

Final Conclusions

The regulation of enzyme activity is crucial to the regulation of protein interactions within signal transduction pathways. The aim of this project was the implementation of Direct Adaptive method for regulating enzyme kinetics reactions. For this purpose Direct Adaptive Control method using RHONNs for affine in the control non-linear dynamical systems with $n \neq m$ (n – the number of states, m – the number of control

inputs) was implemented. We have also considered the more general case of modeling error at zero case, which is a usual instability mechanism.

The Direct Adaptive Control method which was implemented produced satisfactory results, as presented in the previous Chapter. From these results we infer that the state error converged to zero and all signals in the closed loop were uniform ultimate bounded, as it was desirable. The appropriate values of parameters, and especially the design constant k , played an important role in the performance of the closed loop system.

It is also remarkable the fact that the method showed stable behaviour for both of the inputs which were used.

Future Work

The design of adaptive controllers with certain robustness properties with respect to modeling errors or external disturbances can be further improved. In this project we have assumed that the modeling error term $\omega_0(x,u)$ satisfies a Lipschitz condition. This condition guarantees the existence and uniqueness of solutions of $\dot{x} = -Ax + W^* S(x) + S'(x)W_1^* u + \omega_0(x,u)$, which is necessary according to Theorem 2.1.1 for the actual system. Furthermore, larger values of k_1, k_2 cause larger modeling error, but we can take small k_1, k_2 because the approximation error ε can be considered arbitrarily small, according to Theorem 2.1.1

Another future development would be the complete model **tracking** for the case where $n \neq m$. In other words when we want the actual system to converge to a value different than zero. In our case we regulated the PKC active to zero in order to prove that we can lead the final value to any other state.

Finally this project's great application could be a pilot for more personalized medicine where for each patient the exact amount of a single protein or other chemical substance could be injected in order this organism not to have any anomalies or side effects.

Acknowledgement

At this point we should thank M.A. Christodoulou, for the book Adaptive Control with Recurrent high-order Neural Networks, Springer Verlag, New York, USA, 2000 by G.A. Rovithakis, and M.A. Christodoulou. This book was a pilot for the study of adaptive control using RHONN's for the mathematical implementation of the algorithms we needed to derive our final results.

References

- [1] D. Beeman, Speeding up GENESIS simulations, in: J.M. Bower and D. Beeman, eds, *The Book of GENESIS: Exploring Realistic Neural Models with the General NEural Simulation Systems*, Springer-Verlag, New York, pp. 3-23, 1995.
- [2] U.S. Bhalla, A network within: Signaling pathways, in: J.M. Bower and D. Beeman, eds, *The Book of GENESIS: Exploring Realistic Neural Models with the General NEural Simulation System*, Springer-Verlag, New York, pp. 169-191, 1998.
- [3] U.S. Bhalla and R. Iyengar, Emergent properties of networks of biological signaling pathways, *Science*, 283: 381-387, 1999.
- [4] U.S. Bhalla, Modeling networks of signaling pathways, in: E. De Schutter, ed., *Computational Neuroscience: Realistic Modeling for Experimentalists*, CRC Press LLC, New York, pp. 25-48, 2001.
- [5] U.S. Bhalla, Use of Kinetikit and GENESIS for modeling signaling pathways, in: J.D. Hildebrandt and R. Iyengar, eds, *Methods in Enzymology*, Academic Press, New York, vol. 345, pp. 3-23, 2002.

- [6] T.V.P. Bliss and G.L. Collingridge, A synaptic model of memory: Long-term potentiation in the hippocampus, *Nature*, 361: 31-39, 1993.
- [7] J.M. Bower and D. Beeman, *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural Simulation System*, Springer-Verlag, New York, 2nd edit., 1998.
- [8] Database of Quantitative Cellular Signaling (DOQCS), <http://doqcs.ncbs.res.in>.
- [9] E. De Schutter and P. Smolen, Calcium dynamics in large neuronal models, in: C. Koch and I. Segev, eds, *Methods in Neuronal Modeling: From Ions to Networks*, The MIT Press, Cambridge, 2nd edit., pp. 211-250, 2001.
- [10] J. Hellgren Kotaleski, D. Lester, and K.T. Blackwell, Subcellular interactions between parallel fibre and climbing fibre signals in Purkinje cells predict sensitivity of classical conditioning to interstimulus interval, *Integ. Phys. and Beh. Science*, 37: 265-292, 2002.
- [11] M.-L. Linne, *Computational model for granule neuron excitability*, Ph.D. thesis, Department of Information Technology, Tampere University of Technology, 2001.
- [12] M.-L. Linne, T. Manninen, and T.O. Jalonen, A model integrating the cerebellar granule neuron excitability and calcium signaling pathways, *Neurocomputing* 58-60: 569 - 574, 2004.
- [13] R. J. MacGregor, *Neural and Brain Modeling*, Academic Press, San Diego, 1987.
- [14] R.C. Malenka and R.A. Nicoll, Long-term potentiation - A decade of progress? *Science*, 285: 1870-1874, 1999.
- [15] T. Manninen, *Mathematical modelling and simulation of intracellular events in a neuron*, M.Sc. thesis (In Finnish), Department of Science and Technology, Tampere University of Technology, 2003.
- [16] Mathworks, *Mailab, release 13*, The Mathworks Inc., 2004.
- [17] A.C. Newton, Regulation of protein kinase C, *Curr. Opin. Cell. BioL*, 9: 161-16, 1997.

- [18] A. Pettinen, T. Aho, O.-P. Smolander, T. Manninen, A. Saarinen, K.-L. Taatola, O. Yli-Harja, and M.-L. Linne, Simulation tools for biochemical networks: Evaluation of performance and usability, *Bioinformatics*, (Epub ahead of print) <http://bioinformatics.oupjournals.org/cgi/reprint/bti018v1.pdf>.
- [19] S. Sivakumaran, S. Hariharaputran, J. Mishra, and U.S. Bhalla, The database of quantitative cellular signaling: Management and analysis of chemical kinetic models of signaling networks, *Bioinformatics*, 19: 408-415, 2003.
- [20] L.D. Snell, K.R. Iorio, B. Tabakoff, and P.L. Hoffman, Protein kinase C activation attenuates N-methyl-D-aspartate-induced increases in intracellular calcium in cerebellar granule cells, *J. Neurochem.*, 62: 1783-1789, 1994.
- [21] K.-H. Yang, J. Hellgren Kotaleski, and K.T. Blackwell, The role of protein kinase C in the biochemical pathways of classical conditioning, *Neurocomputing*, 38-40: 79-85, 2001.
- [22] Kamp Y., Hasler M., *Recursive Neural Networks for Associative Memory*, J. Wiley & Sons (1990)
- [23] Giles C. L., Chen D., Miller C. B., Chen H. H., Sun G. Z., Lee Y. C., *Second-order recurrent neural networks for grammatical inference*, Proc. of Inter. Joint Conf. on Neural Networks, IJCNN91, Volume 2, 273-281 (1991)
- [24] Liou R., Azimi-Sadjadi M. R., Dent R., *Detection of dim targets in high cluttered background using high order correlation neural network*, Proc. of Inter. Joint Conf. on Neural Networks, IJCNN91, Volume 1, 701-706 (1991)
- [25] Hale J. K., *Ordinary differential equations*, New York, NY, Wiley-InterScience (1969)
- [26] Cotter N. E., *The Stone-Weierstrass theorem and its application to neural networks*, IEEE Transactions on Neural Networks, Volume 1, no. 4, 290-295 (1990)
- [27] Cybenko G., *Approximations by superpositions of a sigmoidal function*, *Mathematics of Control, Signals and Systems*, Volume 2, 303-314 (1989)
- [28] Funahashi K., *On the approximate realization of continuous mappings by neural networks*, Neural Networks, Volume 2, 183-192 (1989)
- [29] Hartman E. J., Keeler J. D., Kowalski J. M., *Layered Neural Networks with Gaussian Hidden Units as Universal Approximators*, Neural Computation, Volume 2, 201-215 (1990)

- [30] Hornik K. M., Stinchcombe M., White H., *Multilayer feedforward networks are universal approximators*, *Neural Networks*, Volume 2, 359-366 (1989)
- [31] Marino R., Tomei P., *Global Adaptive Output Feedback Control of Nonlinear Systems, Part II: Nonlinear Parameterization*, *IEEE Transactions Aut. Contr.*, Volume 38, no.1, 33-48 (1993)
- [32] Corless M. J., Leitmann G., *Continuous State Feedback Guaranteeing Uniform Ultimate Boundness for Uncertain Dynamic Systems*, *IEEE Transactions Aut. Contr.*, Volume AC-26, no. 5, 1139-1144 (1981)
- [33] Barmish B. R., Leitmann G., *On ultimate Boundness Control of Uncertain Systems in the Absence of Matching Condition*, *IEEE Transactions Aut. Contr.*, Volume AC-27, no. 1, 153-158 (1982)
- [34] Chen Y. H., *Adaptive Robust Control of Uncertain Systems with Measurement Noise*, *Automatica*, Volume 29, no. 4, 715-728 (1992)
- [35] Freeman R. A., Kokotovic P. V., *Design of Softer Robust Nonlinear Control Laws*, *Automatica*, Volume 29, no. 6, 1425-1437 (1993)
- [36] Cohen M. A., Grossberg S., *Absolute stability of global pattern formation and parallel memory storage by competitive neural networks*, *IEEE, Transactions on Systems, Man, and Cybernetics*, Volume SMC-13, 815-826 (1983)
- [37] Dempo A., Farotimi O., Kailath T., *High-order absolutely stable neural networks*, *IEEE Transactions on Circuits and Systems*, Volume 38, no. 1 (1991)
- [38] Tiina Manninen^{1,2,*}, Antti Saarinen¹ and Marja-Leena Linne^{1,*}, *Simulation study of differential equation model for protein kinase C signaling*

¹Institute of Signal Processing, ²Institute of Mathematics

Tampere University of Technology, Finland

*Corresponding authors: tiina.mannine@tut.fi, marja-leena.linne@tut.fi