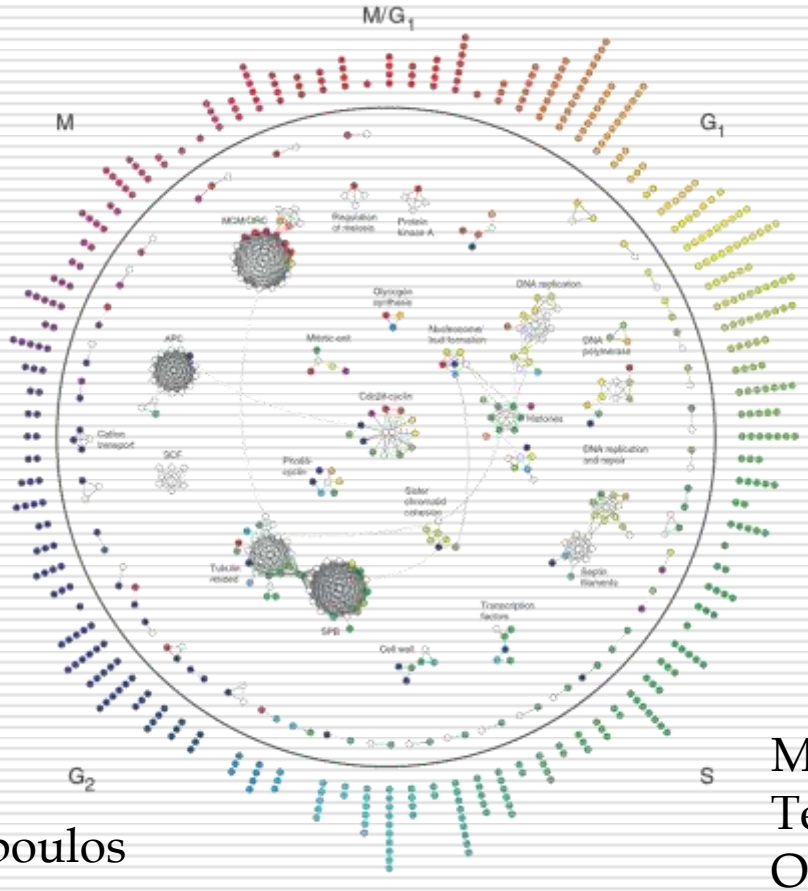
Neural Network Models for prediction of Static and Dynamic behavior of MAPK cascade



Thanasis Iliopoulos

Master of Science thesis,
Technical University of Crete,
October 2005

Outline

- Systems Biology
- Intracellular Signaling Networks
- Conventional Mathematical Modeling Methods and Tools
- The MAPK cascade
- Back-propagation N.N. for steady-state approximation
- Recurrent High Order Neural Network (RHONN) for dynamic behavior approximation in MAPK cascade.
- Conclusions

From Biology to Systems Biology

Ultimate Goal of Biology ⇨ Understand every detail and principle of Biological Systems

- Molecular Biology was born almost 50 years ago.
- Watson and Crick identified DNA structure. They grounded biological phenomena on molecular basis.

- Today, large number of genes have been identified.
- DNA sequences have been fully identified for various organisms (E. coli, C. elegans, homo sapiens, e.t.c.).

⇨ Understanding at the molecular-level, mechanisms of biological systems.

From Biology to Systems Biology

BUT

- Such knowledge does not provide scientists with an understanding of biological systems as *Systems*.
- Scientists understand characteristics and behaviors of the components of the *System*.
- This kind of information is necessary for understanding the system, but not sufficient.

Systems Biology

- Systems Biology is a new field of Biology aiming to develop a System-Level understanding of biological systems.

In order to understand Biological systems as Systems:

- System Structure Identification
- System Behavior Analysis
- System Control (control system's state. Transform malfunctioning cells to healthy cells)

Systems Biology

Functions of a cell do not reside in molecules themselves but in their interactions.

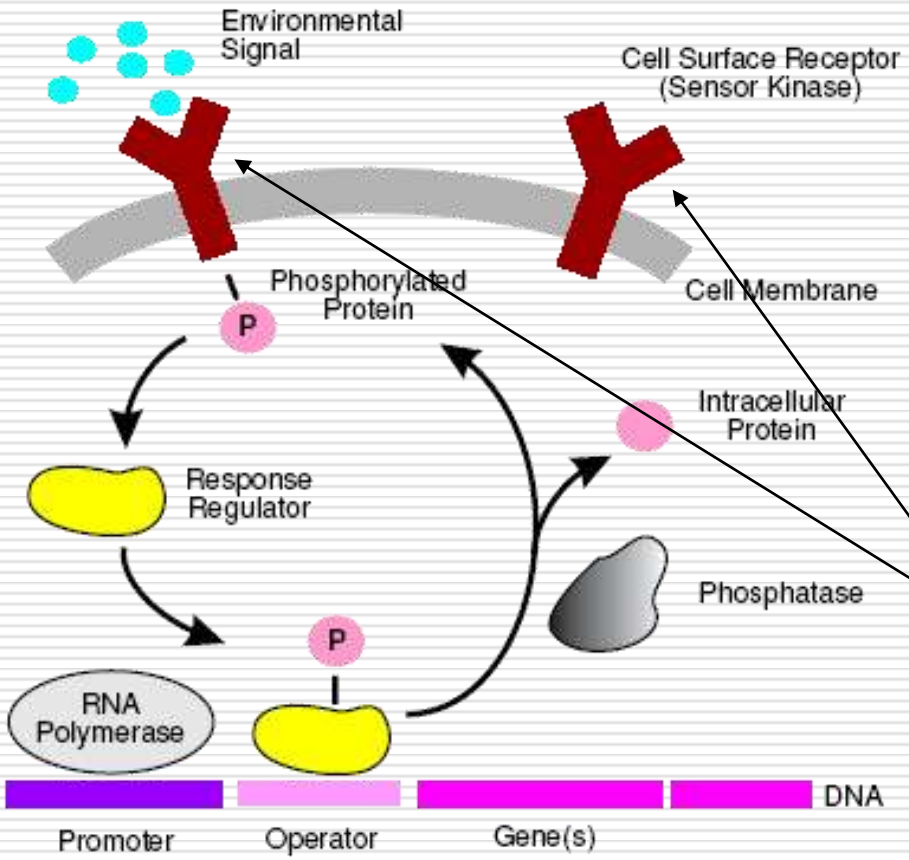
Systems Biology investigates the functioning and function of inter- and intra- cellular dynamic networks, using signal and systems oriented approaches.

- How do components within a cell interact to bring about its structure/function? (intracellular dynamics)
- How do cells interact to bring about coherent cell populations? (intercellular dynamics)

Systems Biology

- Term “Systems” in Systems Biology refers to dynamic system theory.
- System Biology focuses on dynamics and transient changes occurring within cells.
- These changes in most cases are molecule concentrations.
- They carry information and are the root of cellular functions that sustain and develop an organism.

Cell, a system oriented approach



Intercellular Signalling

It is necessary for cells to communicate and exchange information, in order to realize higher levels of organization (tissues, organs, organisms).

Basis of intercellular signalling are the receptors in the cell membrane.

Cell, a system oriented approach

Intracellular Signalling

- Transmission of extracellular information to the genome.

- Transmission of information inside the cell is realized by chemical reaction networks, called *pathways*.

- These networks process environmental signals, induce appropriate cellular responses and sequence internal events, thus allowing cells to perform their basic functions.

Cell death

Cell growth

Specialization

Stress response

Cell cycle control

Cell, a system oriented approach

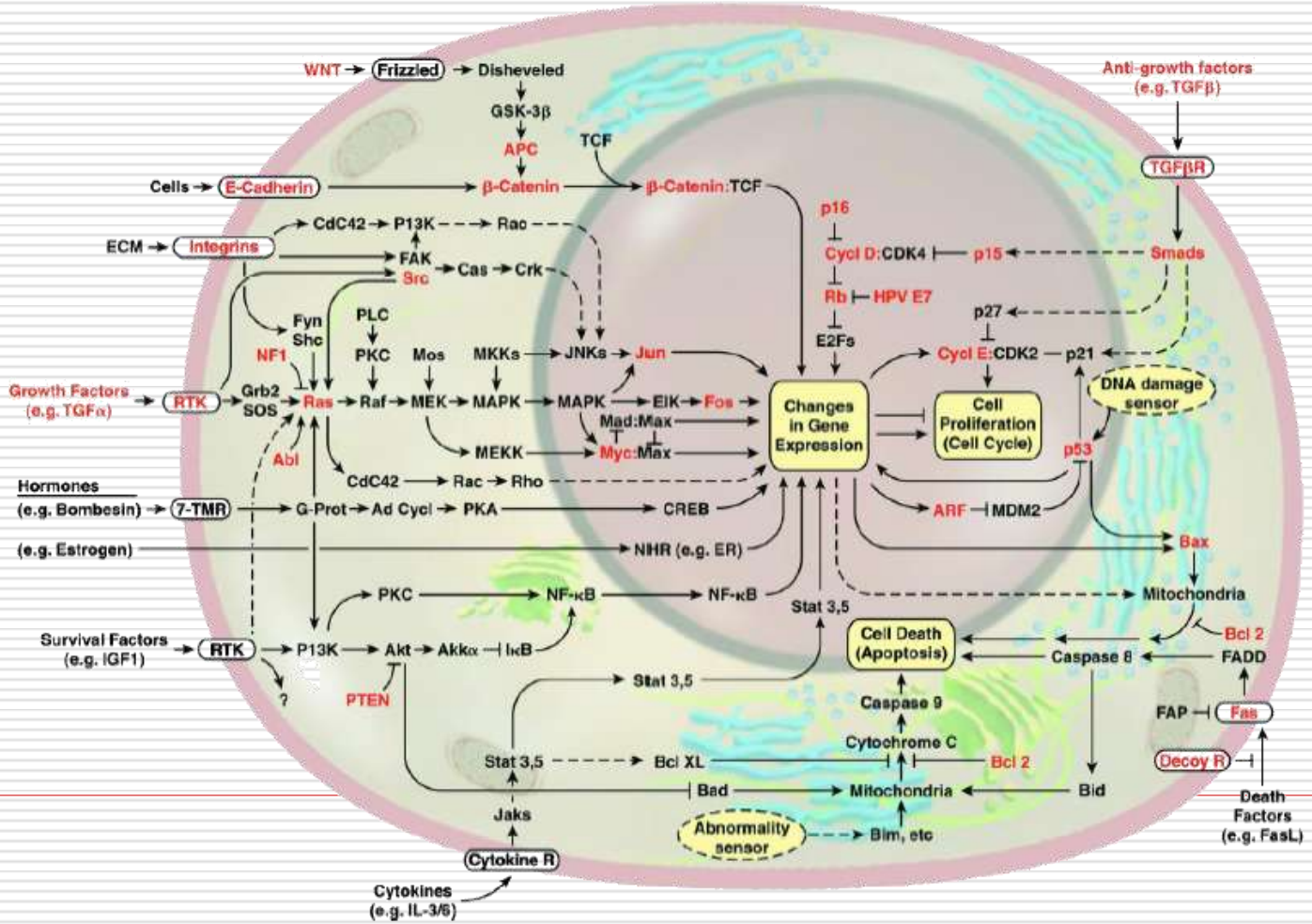
Cells

- Receive external information through inputs that may be:
  - Physical (radiation, temperature, mechanical, e.t.c.)
  - Chemical (drugs, nutrients, hormones, e.t.c.)
- Produce measurable outputs including chemical signals to other cells, movement of pseudopods, e.t.c.

Each cell can be thought as composed of a large number of *subsystems* involving in various processes (growth, death, …)

Intracellular Signaling Networks

Wiring diagram of the growth signaling circuitry of the mammalian cell



Mathematical Modeling of Signal Transduction Pathways

Several methods of modeling intracellular signal transduction pathways have proposed, including:

- Chemical Reaction Schemes using Ordinary Differential Equations
- Stochastic Models
- Petri-Nets
- Rule based Systems
- Boolean Networks

Mathematical Modeling of Signal Transduction Pathways

Dominant concept: Chemical Kinetic Models (transduction pathways i.e. networks of chemical reactions).

- Pathway is an abstraction, a model, of an observed reality.
- In most of the cases these chemical reactions are represented mathematically as *differential equations*.
- Changes in the concentrations of reactants and post reaction products are recorded based on *reaction rates*.

Modeling: Process of abstraction, form of generalization → Simplification of the Physical System .

Mathematical Modeling of Signal Transduction Pathways

<u>Enzymatic reaction scheme</u> ⇨ $E + A \longleftrightarrow EA \longrightarrow E + P$

First step: Enzyme E binds reversibly with substrate A.

Second step: The enzyme releases (irreversibly) the modified substrate P.

Such enzymatic reaction schemes are, in most of the cases, the fundamental building blocks of large intracellular signaling pathways

Mathematical Modeling of Signal Transduction Pathways

Reaction rates

$$
\begin{array}{lcl}
E + A \longrightarrow EA & : & v_1 = a[E][A] \\
E + A \longleftarrow EA & : & v_2 = d[EA] \\
EA \longrightarrow E + P & : & v_3 = V[EA]
\end{array}
$$

System's ODEs

$$
\begin{aligned}
\frac{d[A]}{dt} &= -v_1 + v_2 \\
\frac{d[EA]}{dt} &= v_1 - v_2 - v_3 \\
\frac{d[E]}{dt} &= -v_1 + v_2 + v_3 \\
\frac{d[P]}{dt} &= v_3
\end{aligned}
$$

Initial Conditions

$$
\begin{aligned}
[E](0) &= [E_{tot}] \\
[A](0) &= [A_{tot}] \\
[P](0) &= 0 \\
[EA](0) &= 0
\end{aligned}
$$

Mathematical Modeling of Signal Transduction Pathways

Pressing need for powerful mathematical tools to help understand, quantify and conceptualize signaling networks.

Dominant mathematical tool → chemical kinetic models & ODEs

- It is impossible to experimentally validate the form of nonlinearities, used in reaction terms.
- Accurate estimation of parameters in *vivo* is extremely hard.
- Huge number of simulations required in order to explore state-spaces of large dimension.
- Numerical algorithms cannot guaranteed to produce accurate results.
- Explicit knowledge of pathway's internal structure is essential.
- Errors and omissions due to simplification of the physical system.

Approximation of Dynamic Behavior in Signal Transduction Pathways

However

- Modern non-linear systems theory equips engineers with necessary theoretical tools in order to identify and approximate a large class of complex dynamic systems.

- The *I/O behavior* of a non-linear model could approximate that of an arbitrary dynamic system under certain circumstances.

- The knowledge of biological system's internal structure is no longer prerequisite.

Neural Networks (RHONN) employed to approximate the dynamic behavior of proteins in transduction pathways.

The only requirement is the availability of *experimental data* which describe biological system's I/O behavior.

Systems Biology and Experimental Data

High-throughput technologies has created an enormous amount of biological relevant information, effectively turning biology into an information-rich science.

However (Intracellular signaling pathways)

- Molecular biology experiments are expensive, time consuming and often deliver datasets which fall sort of the expectations of engineers or mathematicians.

- There is only a small amount of quality experimental data available.
- Only a few of the mathematical models proposed are in vivo validated

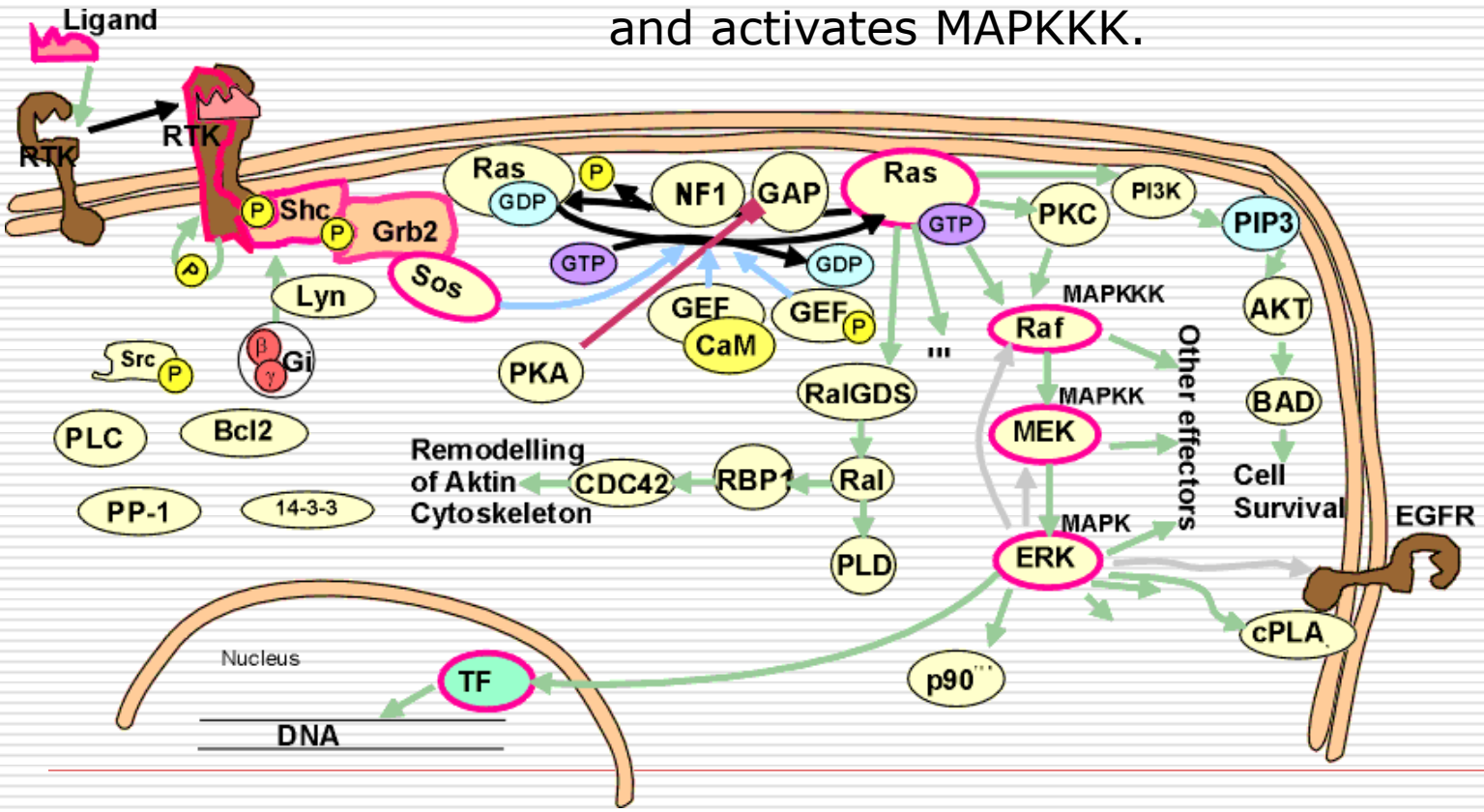
Mitogen Activated Protein Kinase (MAPK) cascade

Mathematical model of MAPK pathway proposed by Huang & Ferrell used for data extraction.

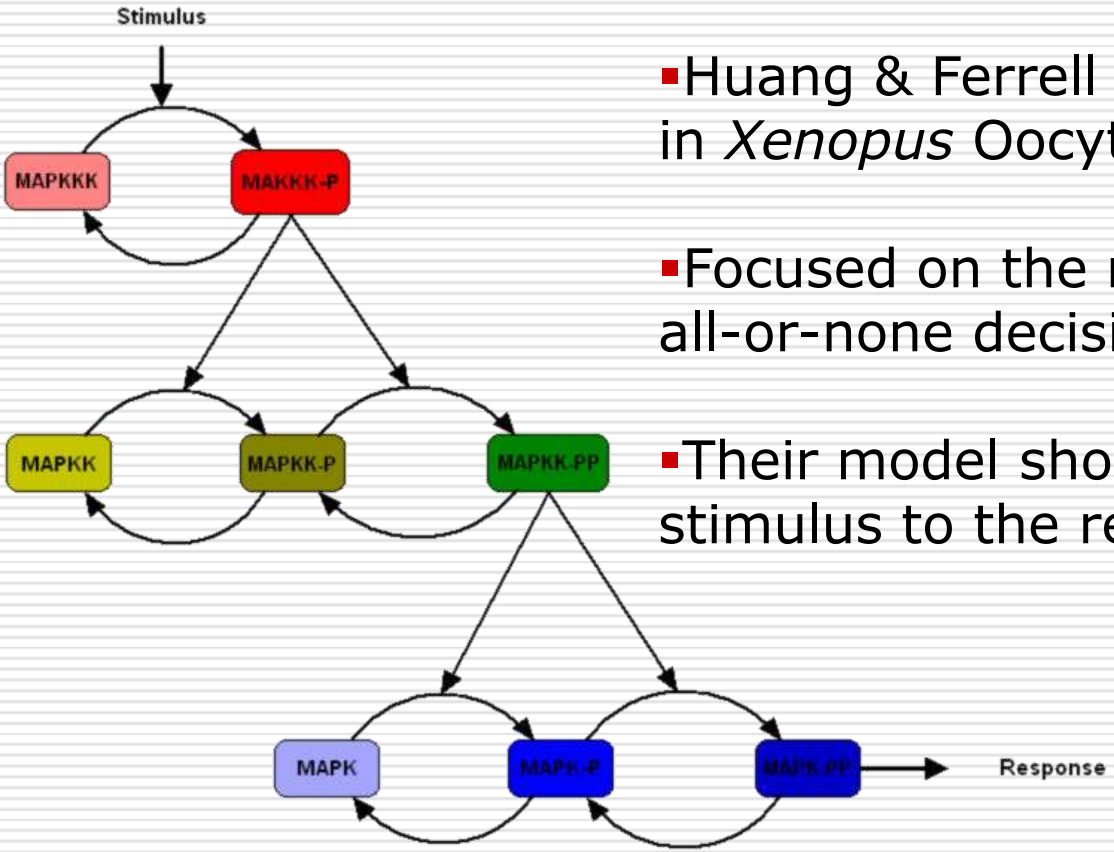
- Three molecule module, present in all eucaryotes.
- Composed of three kinases: MAPK kinase kinase (MAPKKK), MAPK kinase (MAPKK) and MAPK.
- One of the best characterized modules, many reaction parameters estimated.
- Significant role in RAS pathway, which has high influence on cell growth and survival.

MAPK cascade

A small GTP-binding protein or another kinase (PKC) serves as input for the cascade and activates MAPKKK.

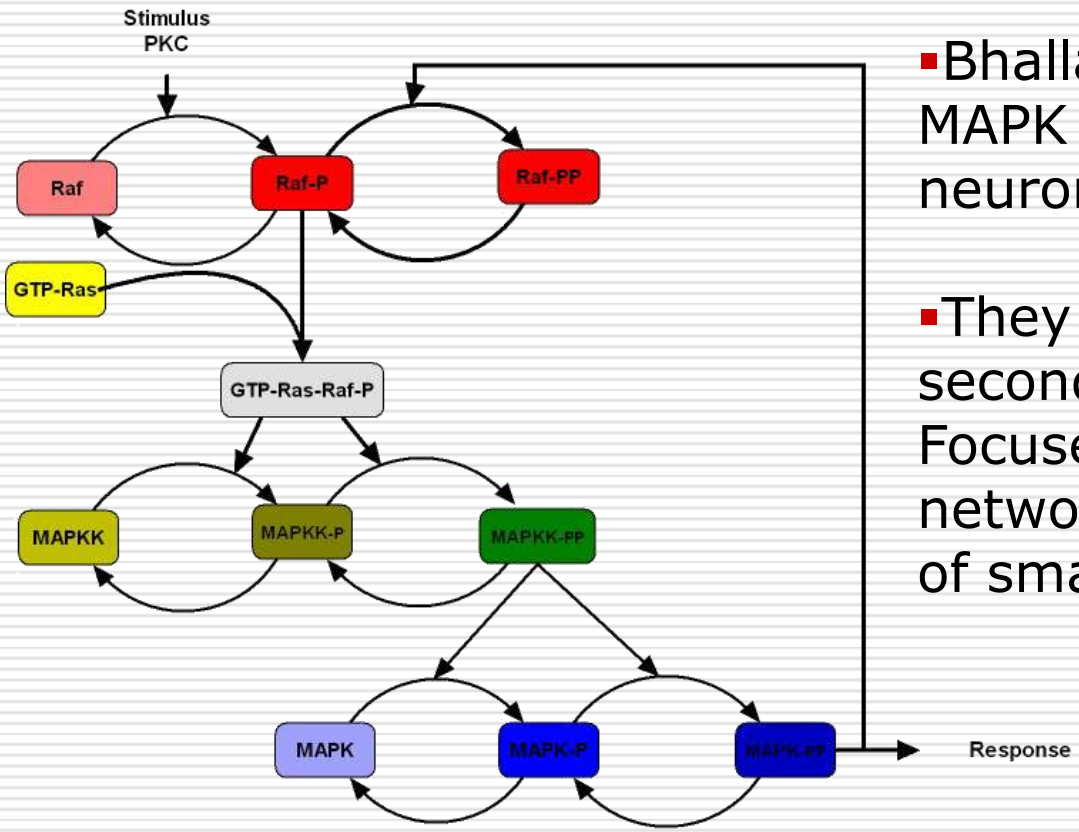


Mathematical Models of MAPK cascade



- Huang & Ferrell model for MAPK cascade in *Xenopus* Oocytes.
- Focused on the role of MAPK cascade in all-or-none decisions.
- Their model shows amplification from the stimulus to the response.

Mathematical Models of MAPK cascade



- Bhalla & Iyengar model for MAPK cascade in mammalian neurons.
- They studied a large model of second messenger cascades. Focused on properties of whole network, rather than on feature of small modules.
- Negative Feedback loop via double phosphorylation of MAPKKK.

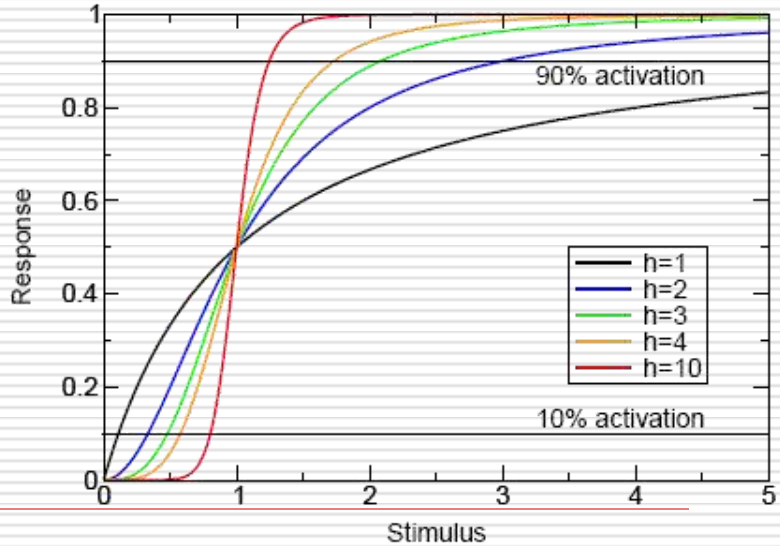
Hill coefficient as ultrasensitivity measure

Steady-state stimulus/response curve is ultrasensitive if it is sigmoidal.

Relative changes in stimulus cause

- Small relative changes in response at low stimuli
- High relative changes in response at higher stimuli

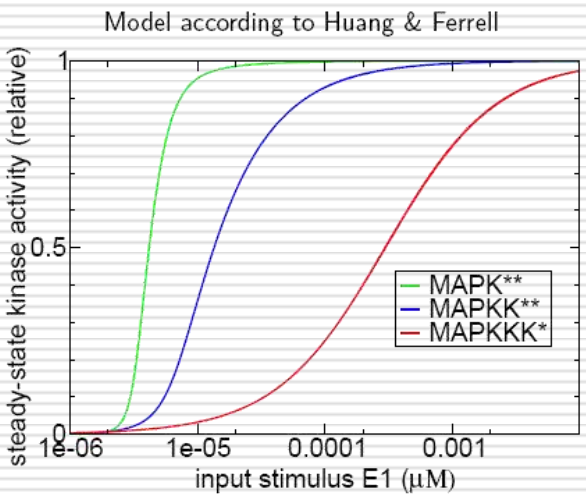
- Hill-coefficient is a measure of ultrasensitivity
- If Hill-coefficient is 1, a 81-fold increase of the stimulus needed to get from 10% activation to 90% activation

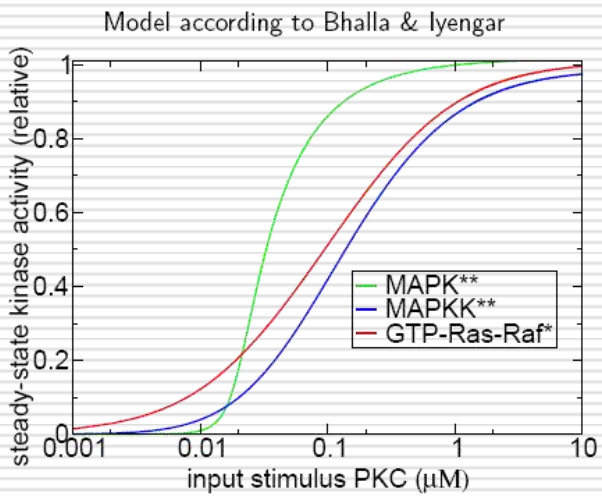


Ultrasensitivity and switch-like behavior of MAPK cascade

Stimulus/response function of the system → steady-state of double phosphorylated MAPK-PP as a function of input enzyme.

Switch-like behavior in cascade
- → 2nd and 3rd level in Huang & Ferrell
- → only in 3rd level in Bhalla & Iyengar





Amplification and Oscillations in MAPK cascade

- Amplification: the absolute change in response divided by absolute change in stimulus.

- Amplification exhibited as the cascade is descended → reason is the existence of three levels in cascade.

- In MAPK model of Bhalla & Iyengar the strength of feedback loop is weak. Increasing the feedback strength → damped oscillations in dynamic behavior of proteins.

MAPK cascade in *Xenopus* Oocytes

- Mathematical Model used → MAPK cascade in *Xenopus* Oocytes proposed by Huang and Ferrell.

- The cascade plays significant role in the maturation of oocytes in response to the steroid hormone progesterone.

- At oocytes' population level → response is graded (the highest the progesterone's concentration, the larger the fraction of oocytes will mature).

- At individual oocyte's level → response is all-or-none. Signal transducers that trigger maturation convert a graded stimulus into an all-or-none cell fate decision.

MAPK model of Huang & Ferrell

Equations of MAPK cascade chemical reactions

$$KKK + E1 \underset{d_1}{\overset{a_1}{\rightleftharpoons}} E1 - KKK \xrightarrow{k_1} KKKP + E1$$

$$KKKP + E2 \underset{d_2}{\overset{a_2}{\rightleftharpoons}} E2 - KKKP \xrightarrow{k_2} KKK + E2$$

$$KK + KKKP \underset{d_3}{\overset{a_3}{\rightleftharpoons}} KKKP - KK \xrightarrow{k_3} KKP + KKKP$$

$$KKP + KKS \underset{d_4}{\overset{a_4}{\rightleftharpoons}} KKS - KKP \xrightarrow{k_4} KK + KKS$$

$$KKP + KKKP \underset{d_5}{\overset{a_5}{\rightleftharpoons}} KKKP - KKP \xrightarrow{k_5} KKPP + KKKP$$

$$KKPP + KKS \underset{d_6}{\overset{a_6}{\rightleftharpoons}} KKS - KKPP \xrightarrow{k_6} KKP + KKS$$

$$K + KKPP \underset{d_7}{\overset{a_7}{\rightleftharpoons}} KKPP - K \xrightarrow{k_7} KP + KKPP$$

$$KP + KS \underset{d_8}{\overset{a_8}{\rightleftharpoons}} KS - KP \xrightarrow{k_8} K + KS$$

$$KP + KKPP \underset{d_9}{\overset{a_9}{\rightleftharpoons}} KKPP - KP \xrightarrow{k_9} KPP + KKPP$$

$$KPP + KS \underset{d_{10}}{\overset{a_{10}}{\rightleftharpoons}} KS - KPP \xrightarrow{k_{10}} KP + KS$$

a_i denotes association, d_i disassociation, k_i product formation.

MAPK model of Huang & Ferrell

Rate equations of MAPK mathematical model

Derived as described before. i.e. rate equation for MAPKKK is:

$$\frac{d(KKK)}{dt} = -v(a_1) + v(d_1) + v(k_2)$$

With reaction rates:

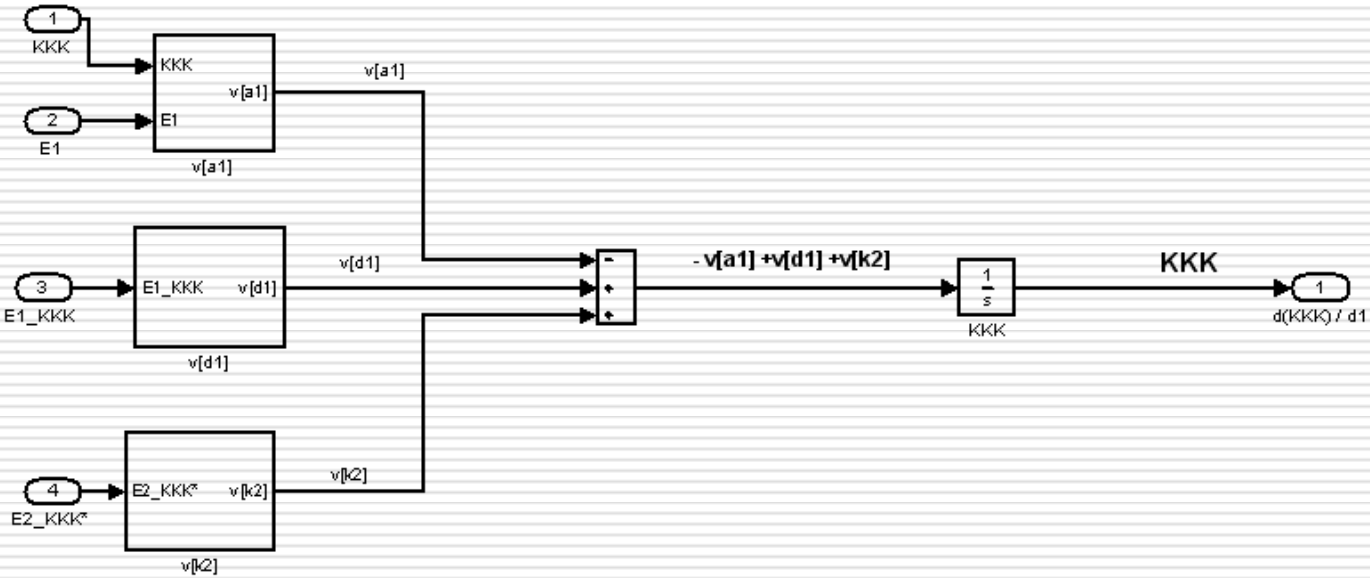
$$v(a_1) = a_1[KKK][E1]$$
$$v(d_1) = d_1[E1 - KKK]$$
$$v(k_2) = k_2[E2 - KKKP]$$

Rate constants calculated according:

$$a_i = (1+f) \cdot \frac{V_i}{K_{m,i}}$$
$$d_i = V_i \cdot f$$
$$V_i = V_{(max,i)}$$

MAPK model of Huang & Ferrell

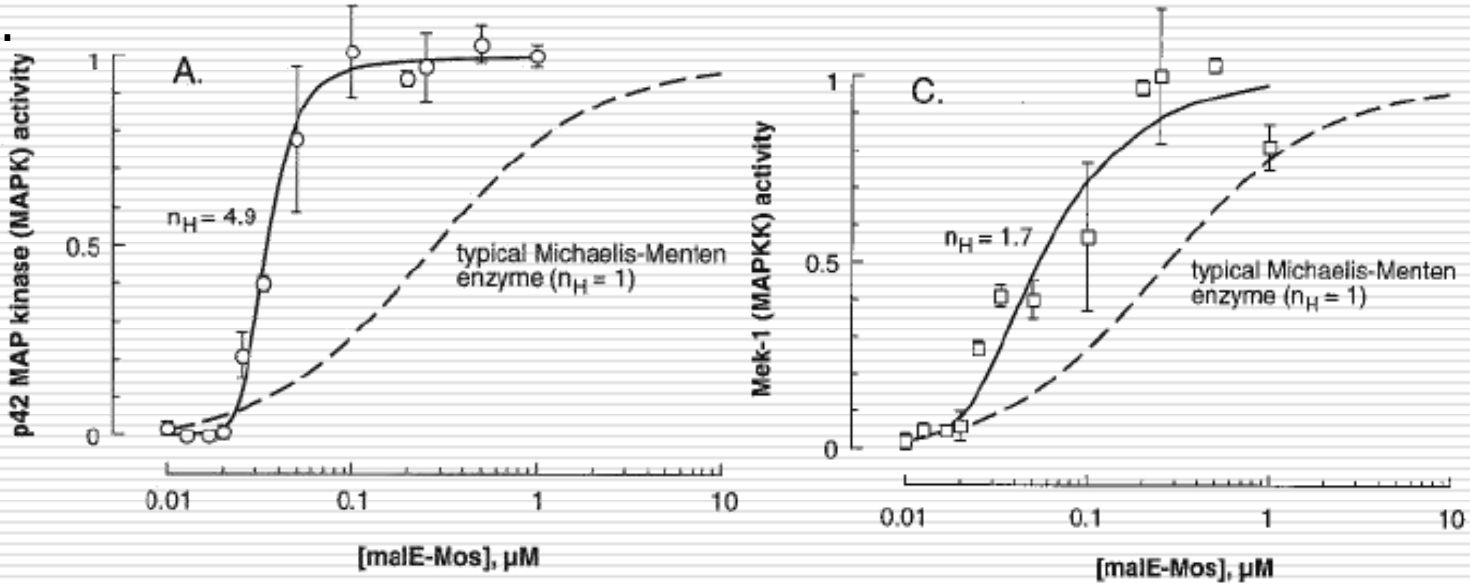
- Rate equations solved numerically using Simulink of Matlab.
- Cascade's model is built by the whole set of the ODEs working in parallel.



MAPK model of Huang & Ferrell

- MAPK mathematical model is experimentally validated.
- Independent experiments for different levels of cascade.

- Cascade filters out noise, switch-like behavior.
- Signaling system that mediates processes like mitogenesis, cell fate induction and oocyte maturation. Cells switch rapidly between discrete states.



Neural Networks

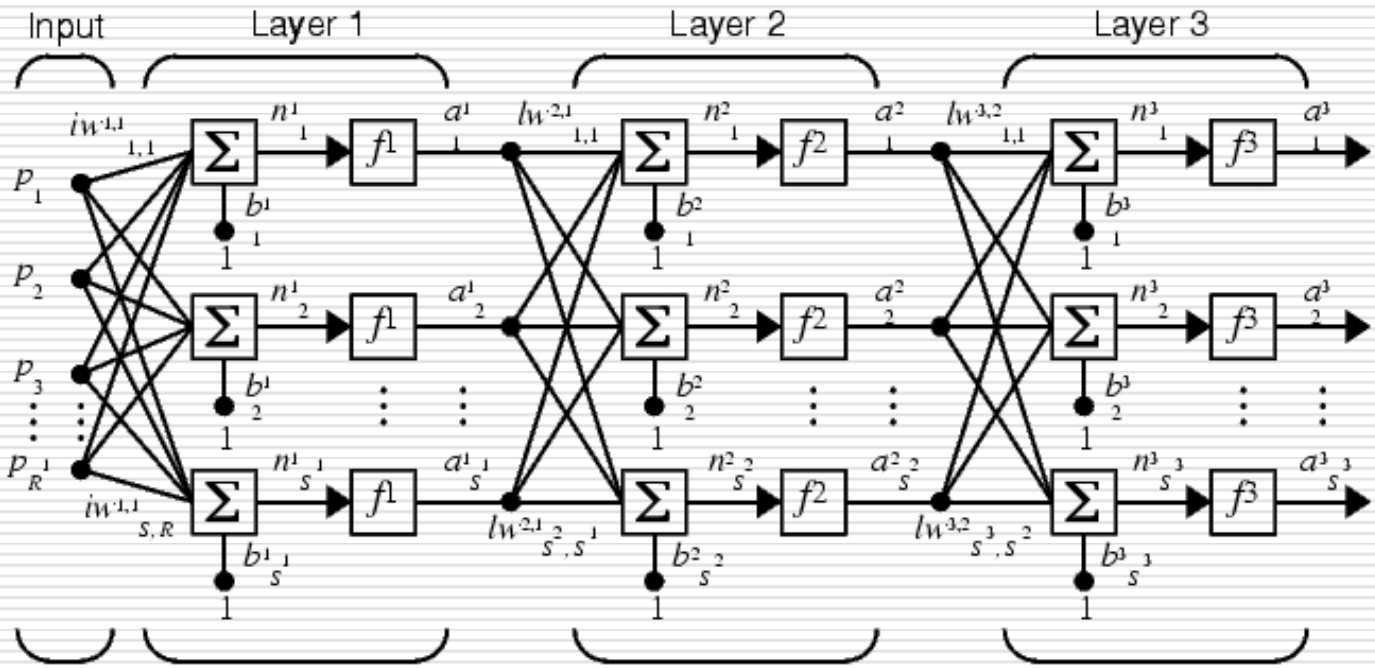
- ANNs are statistical models of real world systems, which are built by tuning a set of parameters (weights).
- Weights describe a model which forms a mapping from a set of given values (inputs) to an associated set of values (outputs).
- Tuning the weights to the correct values (training) → passing a set of examples of I/O pairs through the network models and adjusting weights to minimize the error between model's answer and desired output.
- Once training process is complete the network is able to produce answers for input values not included in the training dataset.

Neural Networks

- ANNs can perform: non-linear function approximation, data classification, clustering, non-parametric regression e.t.c.
- Mathematical model that a NN builds is made up of a set of simple functions linked together by the weights.
- Weights describe the effect that each simple function (neuron) will have on the overall model.
- ANN has a set of input units.
- A set of output units which report network's answer.
- A set of processing hidden units which link input data to output.
- NN arranged in layers. An input layer, an output layer, one or more hidden layers.

Neural Networks Architecture

NN with 2 hidden layers of neurons and one output layer



$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{a}^1 + \mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{a}^2 + \mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3)$$

Neural Networks Training

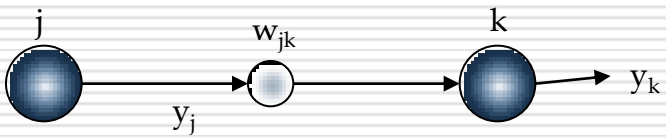
A NN with one hidden layer (sigmoid) and one output layer (linear) can be trained to approximate any static function, with finite number of discontinuities, arbitrarily well.

Neural Network has to be trained to perform a desired function

- Supervised Learning: network is trained by providing it with input and corresponding output patterns.
- Unsupervised Learning (Self Organization): an output unit is trained to respond to clusters of pattern within the input.

Neural Networks Learning Rules

NN learning adjustment of the weights according to some modification rule.



Widrow-Hoff or delta Learning Rule

$$\Delta W_{jk} = \gamma \cdot y_j \cdot (\delta_k - y_k)$$

Where γ is the learning rate, y_j is the output of unit j, δ_k is the desired output and y_k the actual output of the output unit k.

<u>Problem</u>: A 2-layer feed-forward NN can not adjust the weights from input to hidden units.

Back-Propagation Learning Rule

Solution: errors for units of the hidden layers determined by back propagating the errors of the units of the output layer.

- Back-propagation is actually a generalization of delta rule for non-linear activation functions and multilayer networks.

- Back-propagation is a gradient descent algorithm. During training the weights move along the negative of the gradient of the performance function.

Back-Propagation Learning Rule

Problem of local minima: Algorithm designed to always reduce the error will not be able to climb out of a local dip in the error surface.

Momentum term: The weight changes affected by the size of the previous weight changes.
Learning Rate: Tells Network how slowly to progress in the calculation of weight changes.

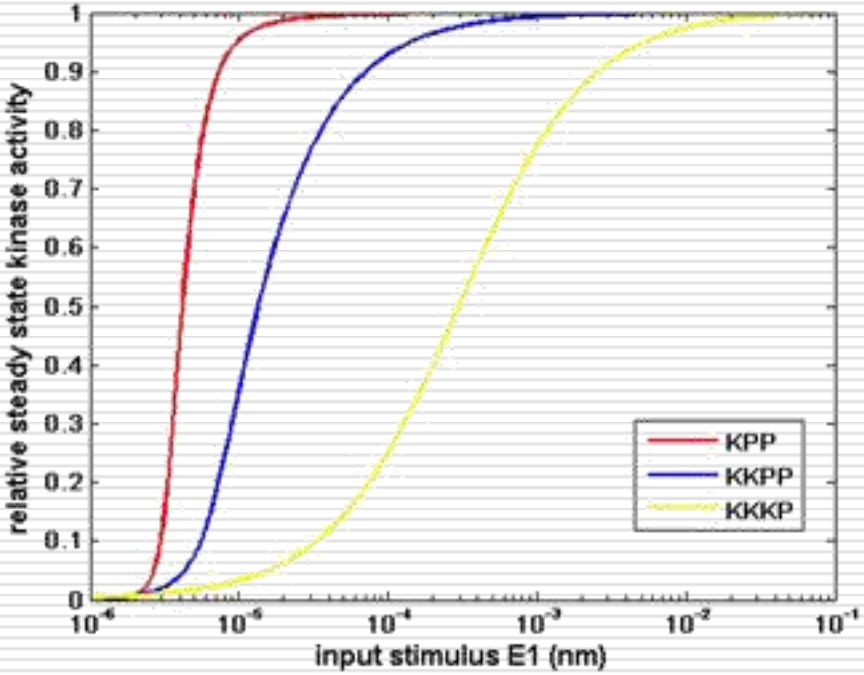
$$\Delta W_{jk}(t+1) = \gamma \cdot y_j \cdot \delta^k + \alpha \cdot \Delta W_{jk}(t)$$

Where α is the momentum term which defines the effect of previous weight changes

Back-Propagation NN for approximation of steady-state behavior in MAPK cascade

Goal: Approximation of the steady-state stimulus/response behavior of MAPKKK-P, MAPKK-PP and MAPK-PP proteins in MAPK cascade.

Classical function approximation problem → static back-propagation neural network.



Back-Propagation NN for approximation of steady-state behavior in MAPK cascade

- MAPK cascade an autonomous system.
- It has an instant external input stimulus in the form of E1 enzyme.
- The variance of E1 initial concentration affects the steady-state behavior of proteins in the cascade.
- E1 initial concentration is varied over the range of 10^{-6} μM to 10^{-1} μM.
- For this range of E1 values we wish to forecast the steady-state activity of MAPKKK-P, MAPKK-PP and MAPK-PP, using a 2-layer back-propagation NN.

Back-Propagation NN for approximation of steady-state behavior in MAPK cascade

The whole process included 4 steps:

1. Assembly of the training data
2. Creation of the network object
3. Train the network
4. Simulate Network to unknown inputs

Assembly of the training data

- Mathematical model of MAPK cascade built in Simulink of Matlab used for training data extraction.

- Our Goal train the network with the least amount of data, since these data correspond to experimental data which is hard to get extracted and limited in number.

- Six training data sets created: First had 455 training pairs, second had 155 t.p., third had 95 t.p., fourth had 50 t.p., fifth had 25 t.p., sixth had 15 t.p.

Creation of the Network Object

NN for the first four (455, 155, 95, 50) training datasets has:

- 1 input unit for E1 initial concentration
- 1 hidden layer with 15 neurons implementing tan-sigmoid function
- 1 output layer with 3 linear neurons

NN for the fifth (25) training dataset has:

- 1 input unit
- 1 hidden layer with 10 neurons (tan-sigmoid)
- 1 output layer with 3 neurons (linear)

NN for the sixth (15) training dataset has:

- 1 input unit
- 1 hidden layer with 8 neurons (tan-sigmoid)
- 1 output layer with 3 neurons (linear)

Preprocessing & Post-processing data

In training and validation datasets → concentration values of proteins were greatly varied making the training process difficult.

NN training → more efficient if preprocessing of training input/targets is performed.

Normalization of the mean and standard deviation of training input/targets → zero mean and unity standard deviation.

Table 4.1: Range of values in the Training Dataset - concentration values in nM

	Net Inputs	Net Targets		
	E1	MAPKKKP	MAPKKPP	MAPKPP
minimum value	10^{-3}	0.0097429	1.5972	0.23109
maximum value	10^{2}	2.7504	1081.2	989.09

NN training algorithm

- Weight adjustment → minimize network's performance function *mse*.

- Standard back-propagation training algorithm converges too slow → we wish to employee a faster algorithm.

- Quasi–Newton algorithms approach second order training speed.

- Newton's algorithm:

$$x_{k+1} = x_k - A_k^{-1} \cdot g_k$$

where A_k is the *Hessian* matrix (second derivatives) of the performance index.

- But it is complex and expensive to compute Hessian matrix

NN training algorithm

- Quasi-Newton algorithms faster, compute an approximation of the Hessian matrix.

- Levenberg-Marquardt training algorithm computes an approximation of Hessian matrix as: $H = J^T \cdot J$, where J is the *Jacobian* matrix (first derivatives) of networks errors with respect to the weights.

- Iteration of Levenberg-Marquardt training algorithm:

$$x_{k+1} = x_k - [\, J^T \cdot J + \mu I \,]^{-1} \cdot J^T \cdot e$$

μ=0: Newton's method.

μ is large: gradient descent with small step size.

Training process and results

Each NN has trained with the corresponding dataset

```
% Levenberg-Marquardt back-propagation training algorithm
clear all; pack; clc;
load data6.mat; %In data6.mat there 455 training pairs

p = x2n; %original network inputs
t = [x4n; x13n; x21n]; %original network targets

%normalization of the input target and vectors with prestd
[pn,meanp,stdp,tn,meant,stdt] = prestd(p,t);

%The network object is created. It has a hidden layer with 15 tansig
net =
newff([minmax(pn)],[15,3],'tansig','purelin','trainlm','learngdm','mse');
```

Training process and results

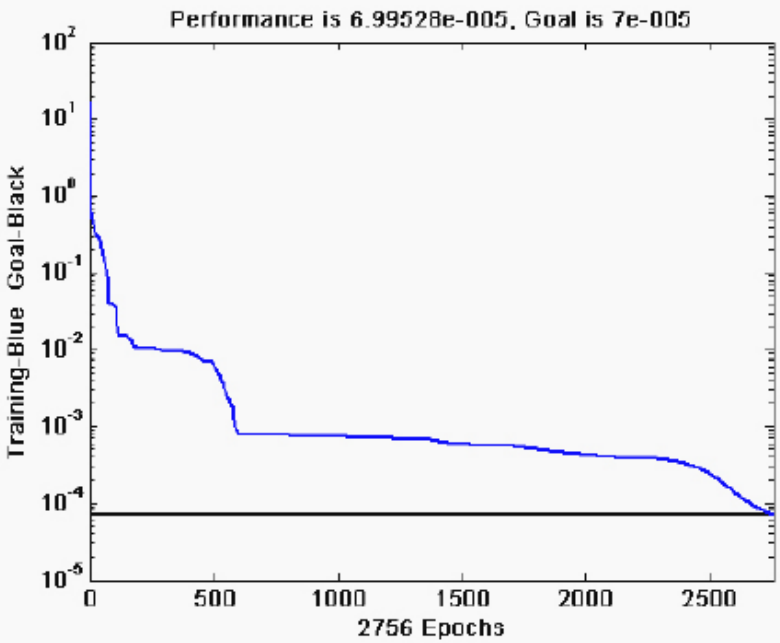
```
net = initnw(net,1) %Initialization of the first layer weights
net = initnw(net,2) %initialization of the second layer weights
net.trainParam.show = 100;
net.trainParam.epochs = 12000; %Maximum number of iterations
net.trainParam.goal = 7e-5; %Performance function desirable value

[net,tr] = train(net,pn,tn); %Train the network
```

Training process and results

Neural Network 1 - data6 (455 training examples of input-output behavior)
Training last 2756 epochs. At the end of the training process the performance function goal was met, as it is shown above

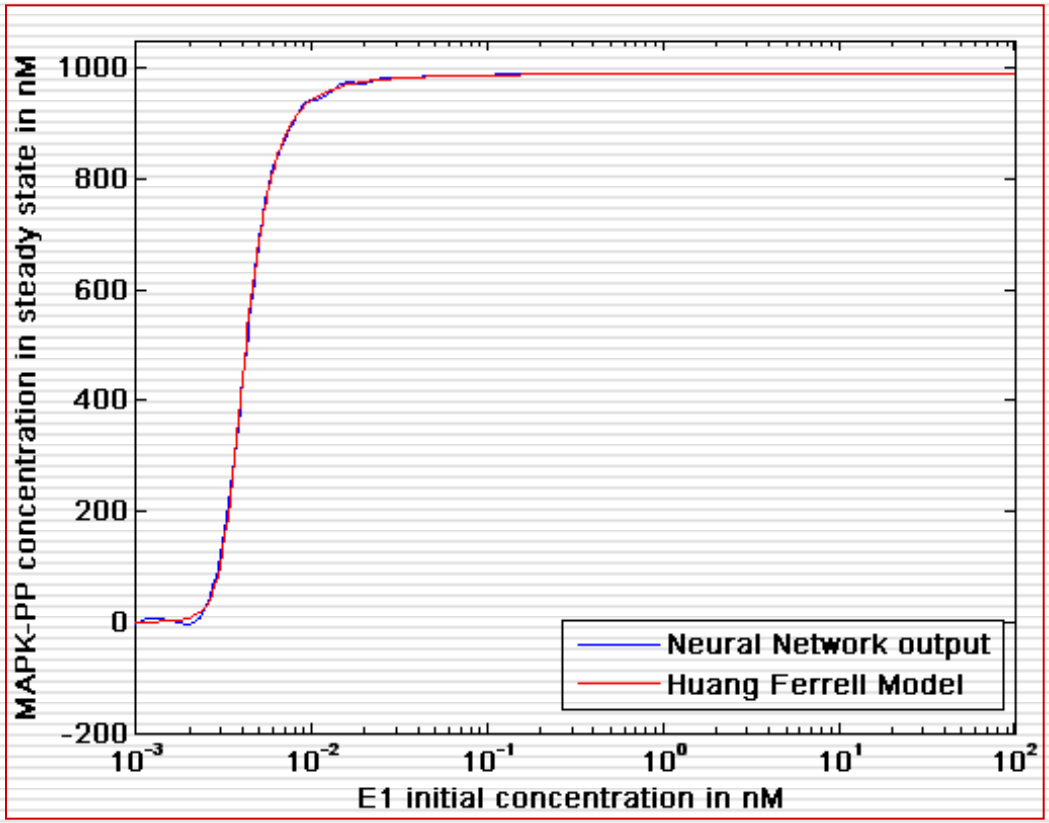
```
TRAINLM, Epoch 0/12000, MSE 17.2683/7e-005, Gradient 11197.4/1e-010
TRAINLM, Epoch 100/12000, MSE 0.0381497/7e-005, Gradient 11.4551/1e-010
TRAINLM, Epoch 200/12000, MSE 0.01051/7e-005, Gradient 12.314/1e-010
:
TRAINLM, Epoch 2500/12000, MSE 0.000241318/7e-005, Gradient 24.9116/1e-010
TRAINLM, Epoch 2600/12000, MSE 0.000140555/7e-005, Gradient 124.69/1e-010
TRAINLM, Epoch 2700/12000, MSE 8.51862e-005/7e-005, Gradient 56.2018/1e-010
TRAINLM, Epoch 2756/12000, MSE 6.99528e-005/7e-005, Gradient 37.0837/1e-010
TRAINLM, Performance goal met.
```

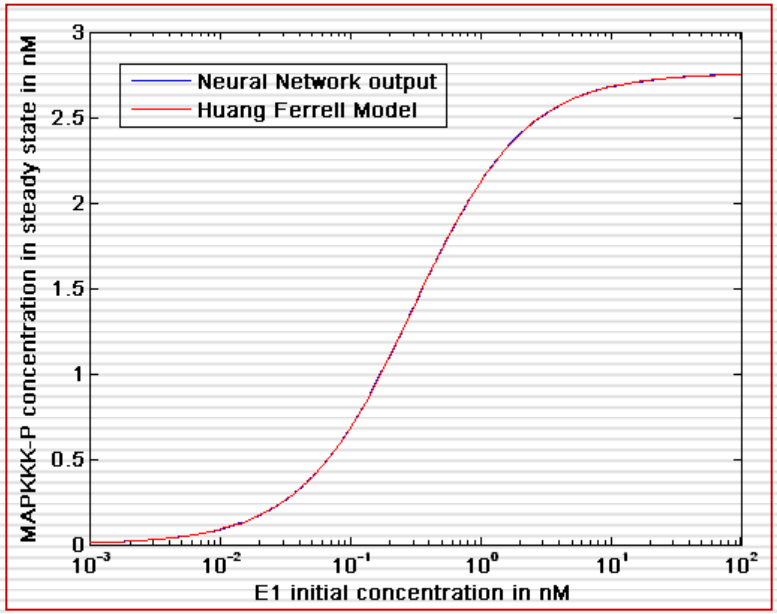


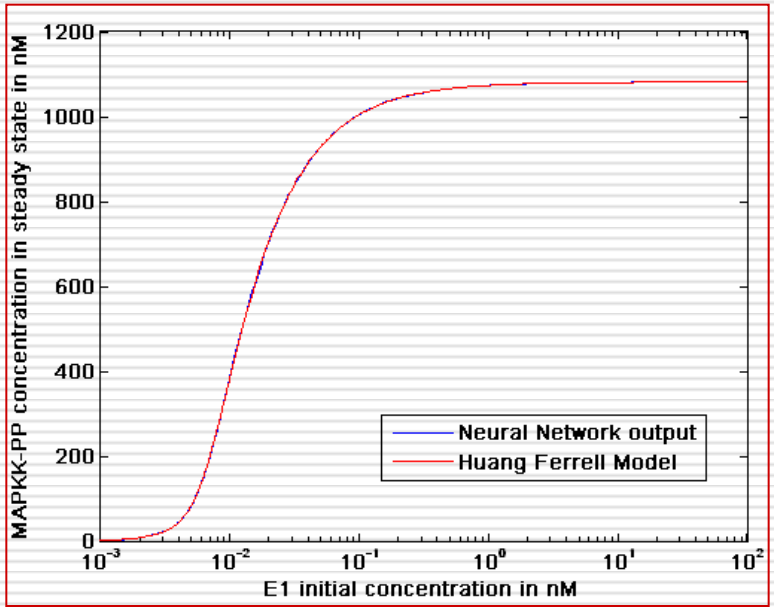
NN1: 2756 epochs – NN2: 2687 epochs – NN3: 3080 epochs – NN4: 4278 epochs – NN5: 1848 epochs – NN6: 1997 epochs.

Validation process

NN1 (455) validation results

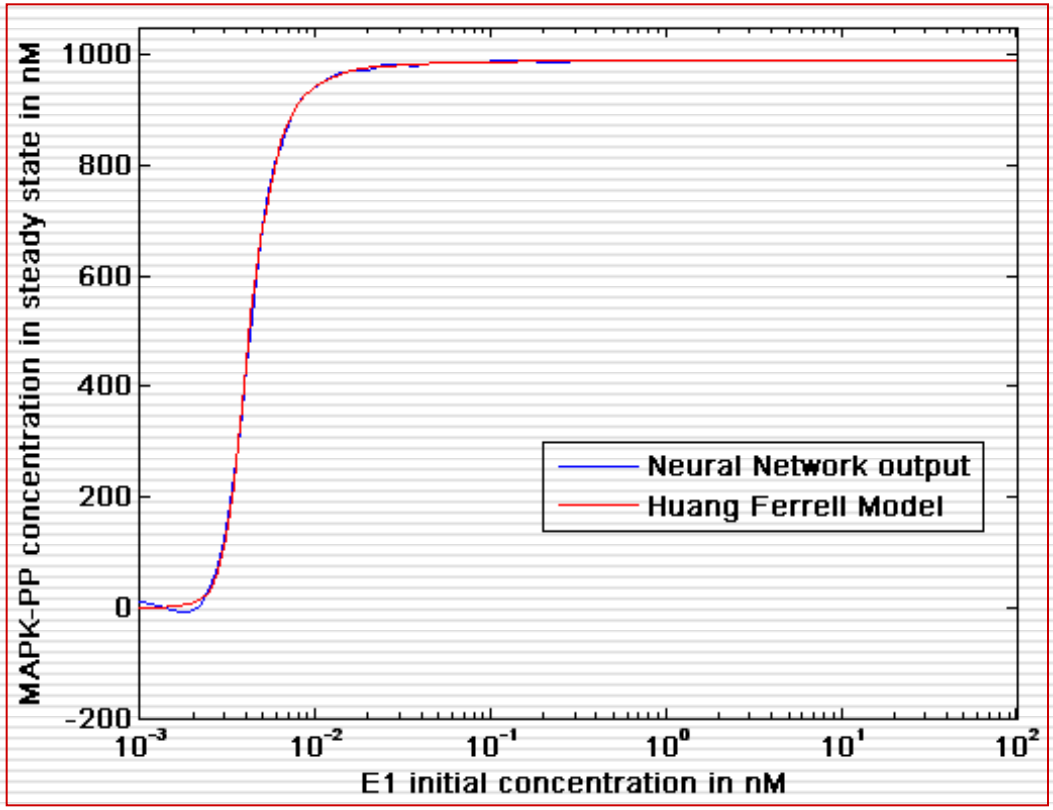


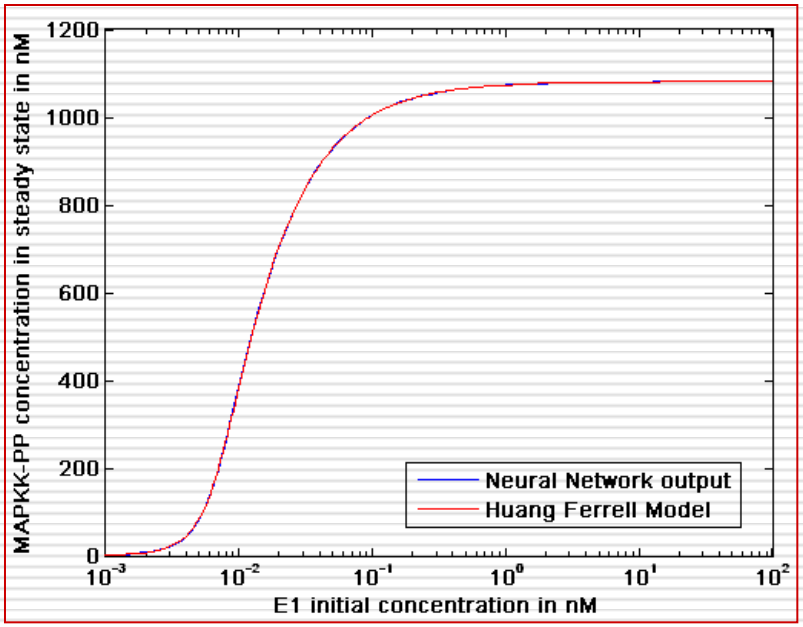


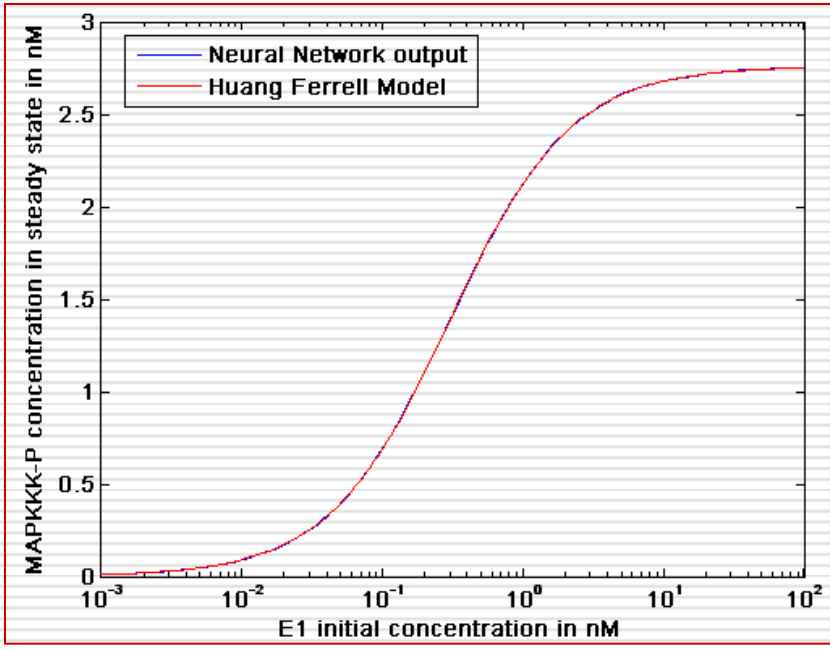


Validation process

NN2 (155) validation results

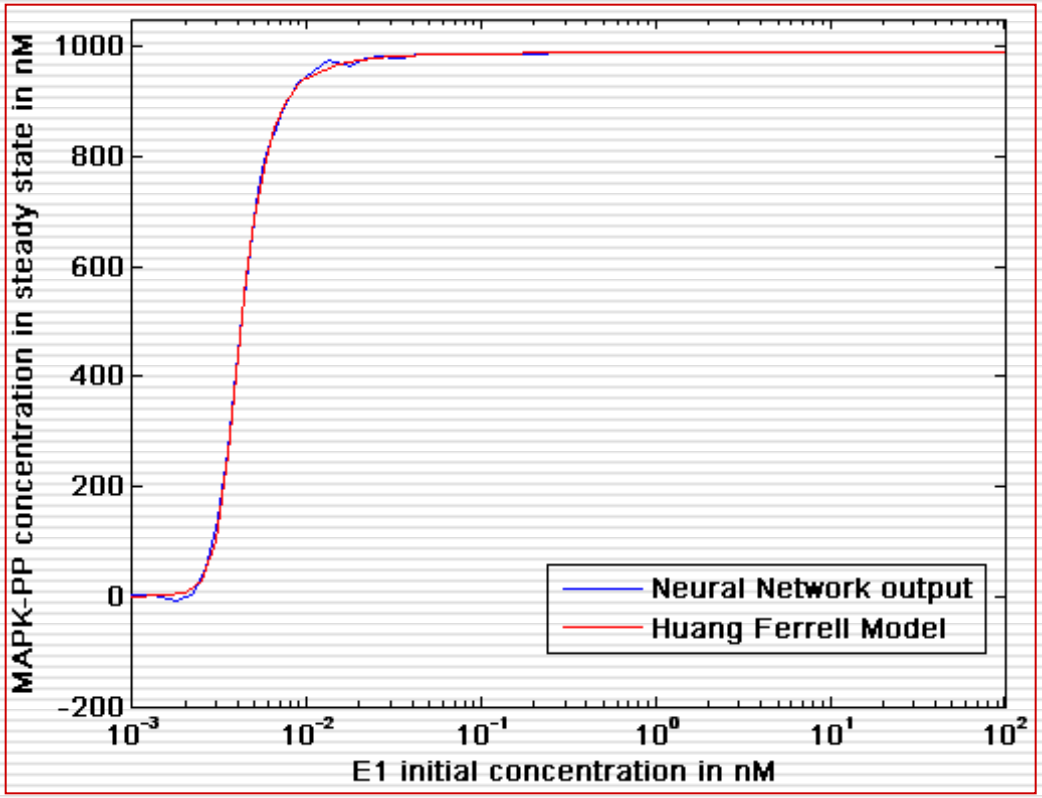


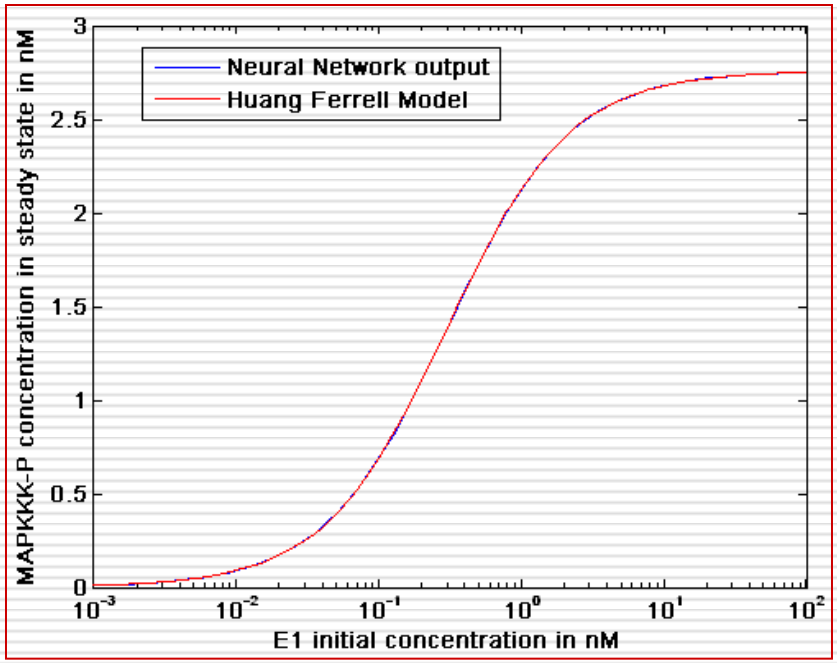


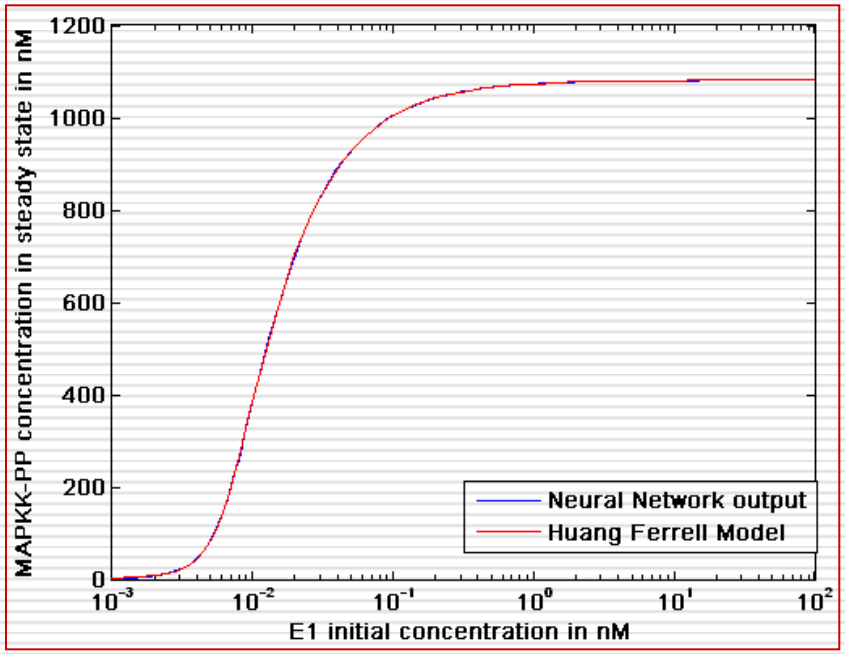


Validation process

NN3 (95) validation results

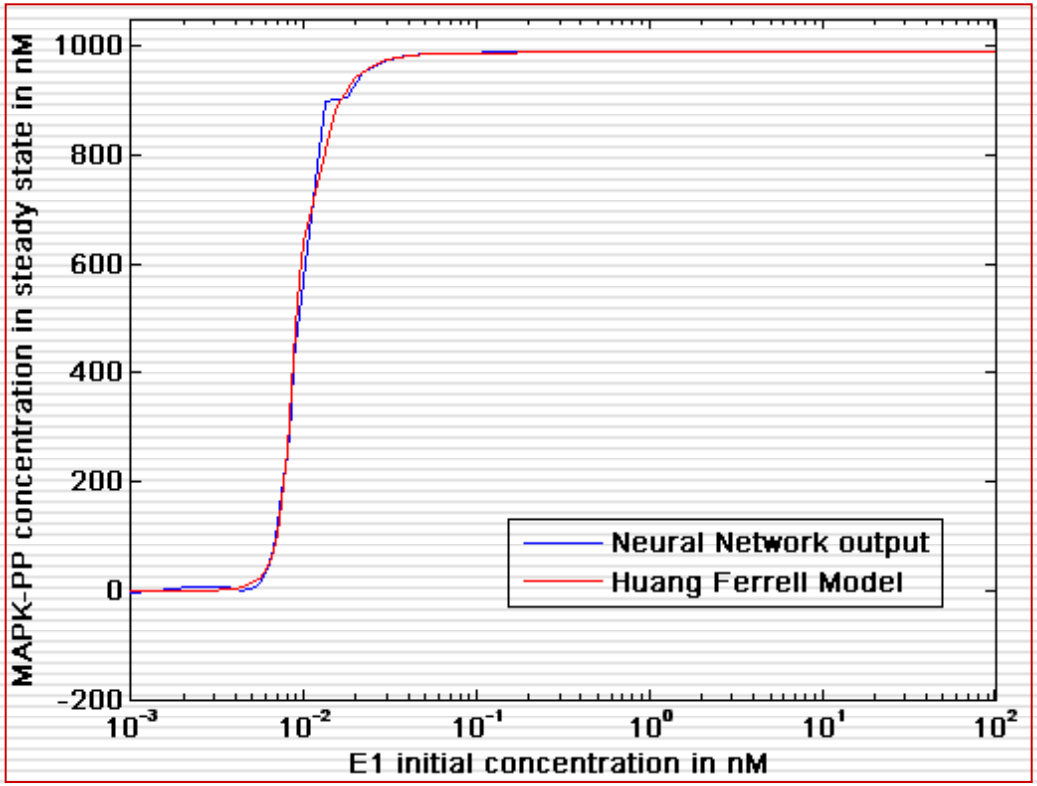


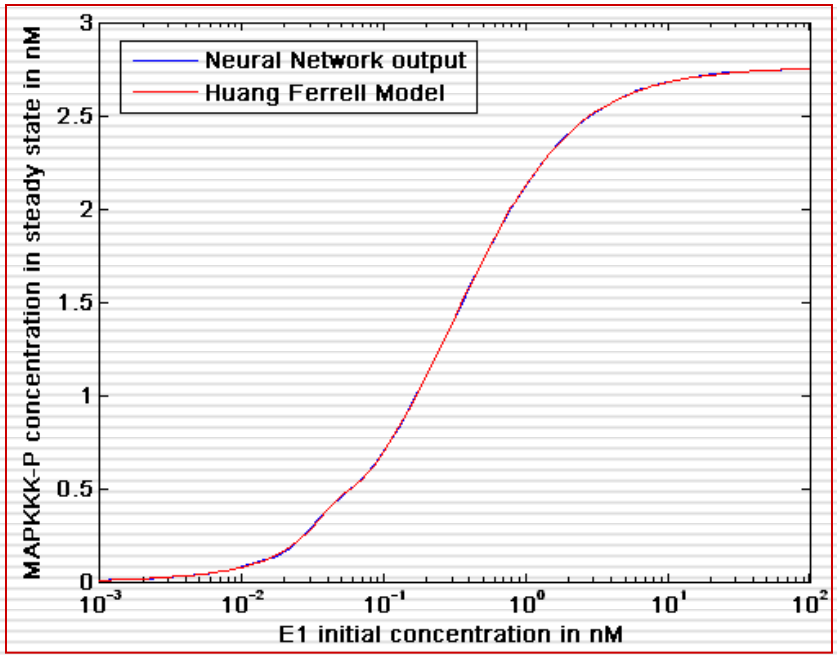


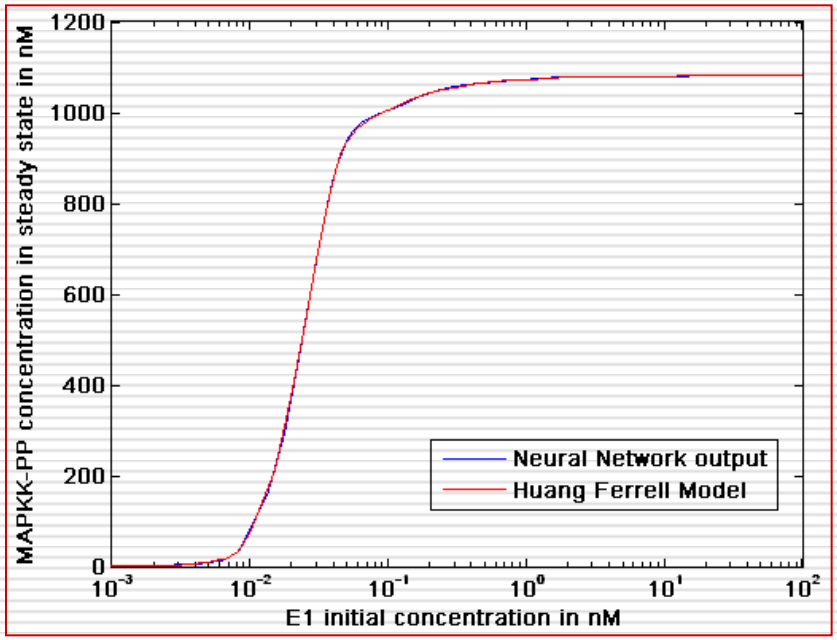


Validation process

NN4 (50) validation results

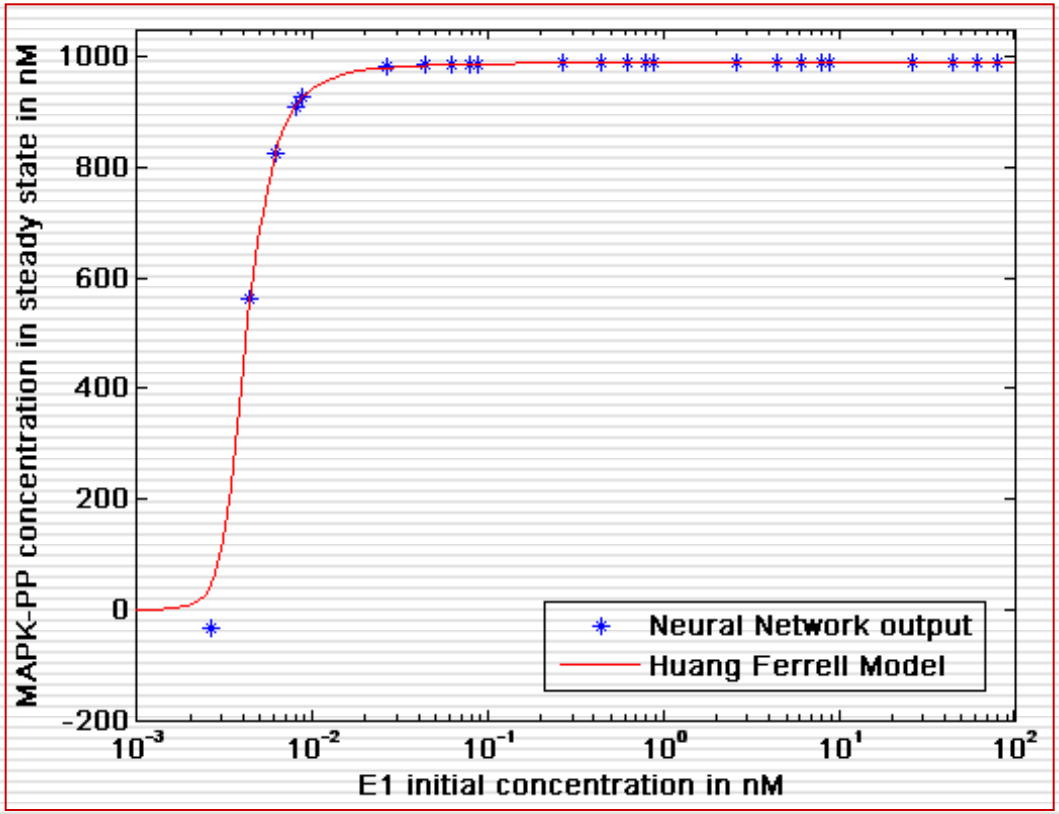


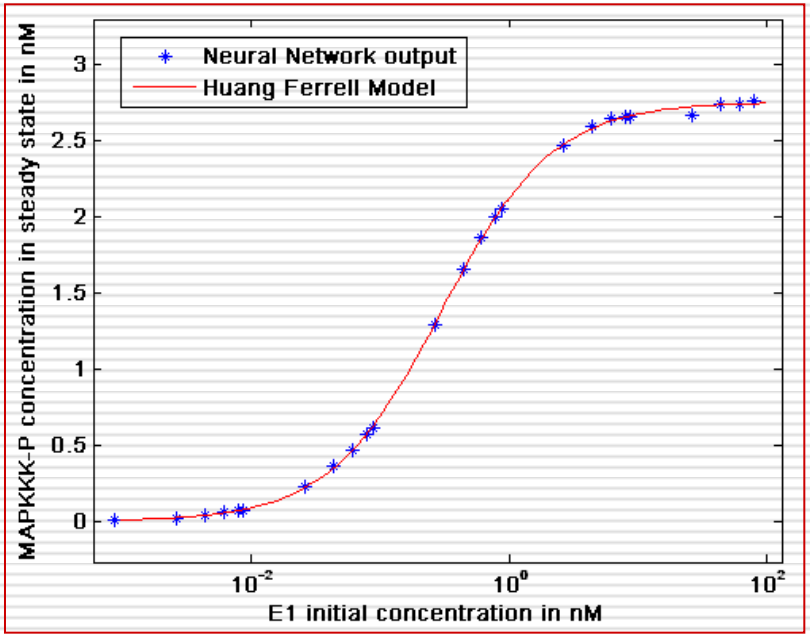


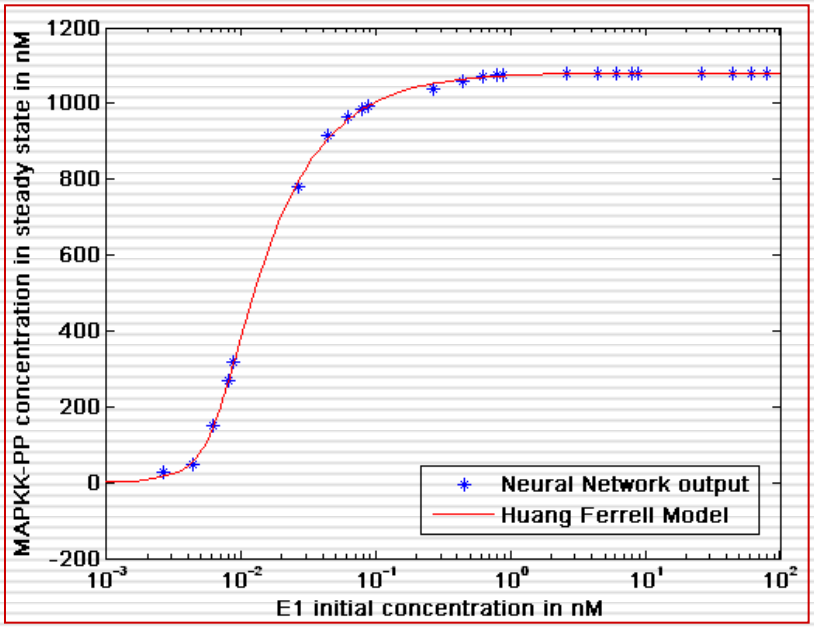


Validation process

NN5 (25) validation results

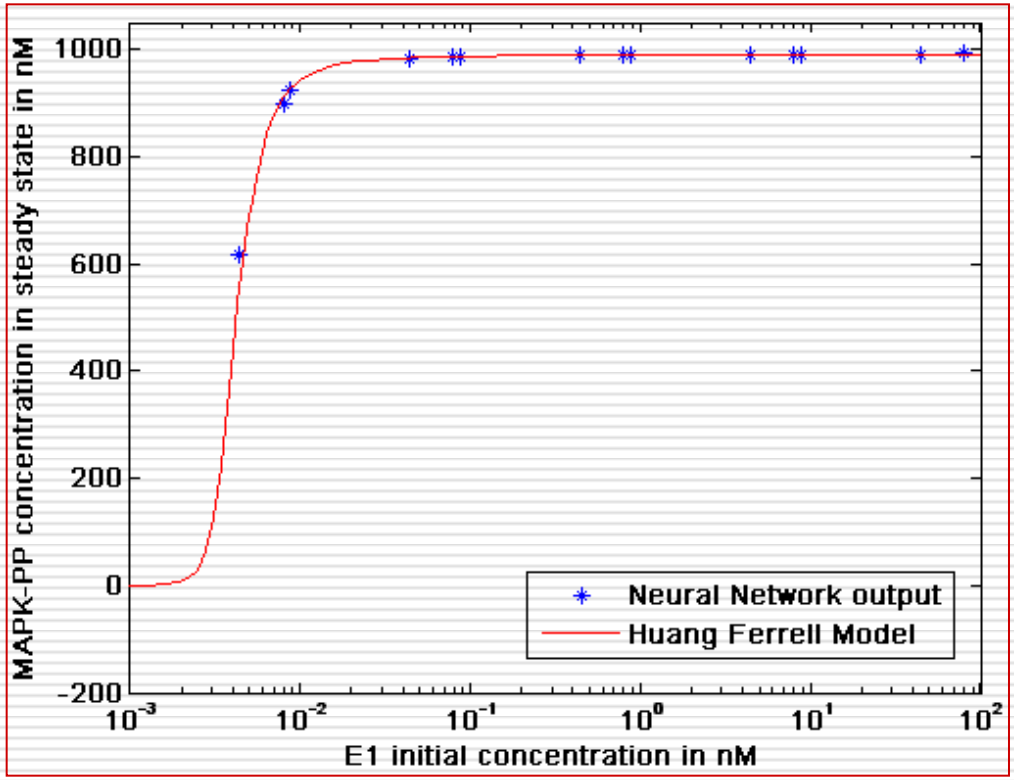


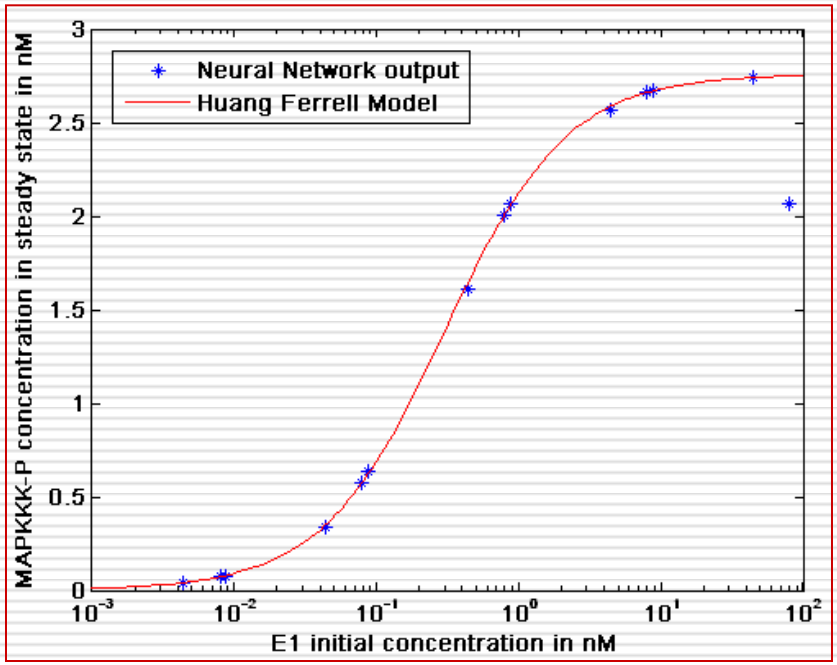


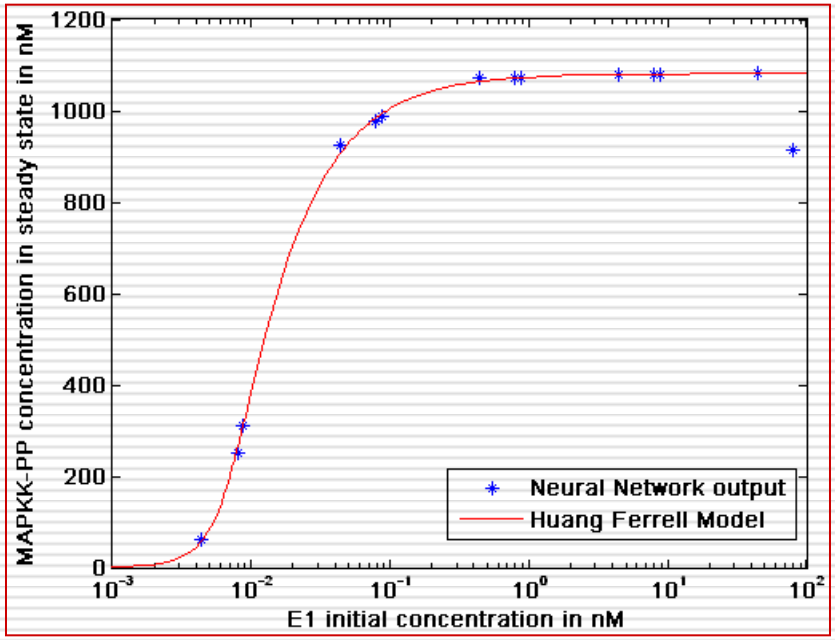


Validation process

NN6 (15) validation results







Conclusions

- Back-propagation Neural Network can predict the steady-state stimulus/response behavior of proteins in MAPK cascade.

- Even in the case where it is trained with a small number (15) of training patterns.

- Steady-state stimulus/response behavior of proteins is of great interest in cases where signaling cascades are responsible for all-or-none decisions in the cell.

Approximation of dynamic behavior in MAPK cascade

A more interesting and challenging problem is the approximation of the dynamic behavior of proteins in MAPK cascade.

⇩

Approximation of a non-linear dynamic system with the use of Neural Networks.

Recurrent High Order Neural Networks

In order that a NN architecture be able to approximate the behavior of a dynamical system it should contain some form of dynamics/feedback connections.

Recurrent Neural Networks (RNN)

- Several training methods have proposed for RNNs, relying on the gradient methodology → extensions of back-propagation.
- Although they share fundamental drawbacks:
  - computationally expensive
  - inability to obtain analytical results concerning convergence and stability of these schemes.

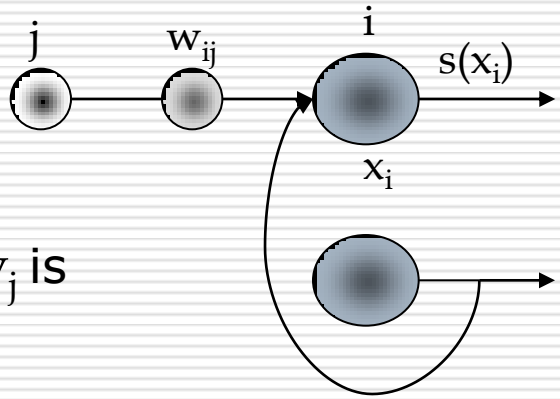
Recurrent High Order Neural Networks

- Design and analysis of learning algorithms based on Lyapunov stability theory → providing stability, convergence and robustness proofs.

- Recurrent High Order Neural Networks employed for prediction of the behavior of unknown non-linear dynamic system.

RHONN Model

RNN model state history: $\dot{x}_i = -a_i x_i + b_i \sum_j w_{ij} y_j$



x_i the state of i-th neuron, a_i and b_i constants, w_{ij} weight connecting j-th input to i-th neuron. Each y_j is either an external input or the state of a neuron passed through the sigmoid (y_j=$s(x_j)$).

- In Recurrent second Order NN, input to neuron is not only a linear combination of y_j, but also of their products $y_j y_k$.

- Interactions of higher order → RHONNs

RHONN Model

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

$\{I_1, I_2, \ldots, I_L\}$ collection of L not-ordered subsets of $\{1,2,\ldots,m+n\}$, $d_j(k)$ real coefficients,

y is the input vector to each neuron: $$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{n+m} \end{bmatrix} = \begin{bmatrix} s(x_1) \\ s(x_2) \\ \vdots \\ s(x_n) \\ u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$$

and $$s(x) = \frac{a}{1 + e^{-\beta x}} - \gamma$$

RHONN Model

Introducing L-dimensional z vector:

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

and the adjustable parameter vector: $w_i = b_i[w_{i1}, w_{i2}, \ldots, w_{iL}]^T$

The RHONN model becomes: $\dot{x}_i = -a_i x_i + w_i^T z$

where the vectors $\{w_i: i=1,2,\ldots,n\}$ represent the adjustable weights of network, while coefficients $\{a_i: i=1,2,\ldots,n\}$ are part of underlying network architecture.

RHONN Model

The dynamic behavior of the overall network is described in vector notation as:

$$\dot{x} = Ax + W^T z,$$

where $x = \{x_1, x_2, \ldots, x_n\}^T \in R^n$, $W = \{w_1, w_2, \ldots, w_n\}^T \in R^{Lxn}$ and $A = \mathrm{diag}\{-a_1, -a_2, \ldots, -a_n\}$ is a nxn stability matrix. Vector z is a function of both network's state x and network's input u.

RHONN Model – Approximation properties

Problem: approximation of a general non-linear

dynamic system whose I/O behavior: $\dot{\chi} = F(\chi, u)$

where $\chi \in R^n$ system state, $u \in R^m$ system input.

Question: By allowing enough high order connections there exist weights W, such that the RHONN model approximates the I/O behavior of an arbitrary dynamical system of the previous form?

Theorem: Suppose that the real system and the RHONN model are initially in the same state $x(0)=\chi(0)$; then for any $\varepsilon>0$ and any finite $T>0$, there exists an integer L and a matrix $W^* \in R^{Lxn}$ such that the state $x(t)$ of the RHONN model, with L high order connections and weight values $W=W^*$ satisfies $\sup_{0 \le t \le T} |x(t) - \chi(t)| \le \varepsilon$

Filtered Error RHONN Learning algorithm

There exist unknown weight vectors w_i^* such that each state χ_i of the unknown dynamic system satisfies: $\dot{\chi}_i = -a_i\chi_i + w_i^\star z(\chi, u)$

The RHONN model for prediction of the I/O behavior of unknown system is: $\dot{x}_i = -a_i x_i + w^T z$

where w_i is the estimate of unknown w_i^*. The state error $e_i = x_i - \chi_i$ satisfies: $\dot{e}_i = -a_i e_i + \phi_i^T z,$

where $\varphi_i = w_i - w_i^*$. The weights w_i are adjusted according to learning law: $\dot{w}_i = -\Gamma_i z e_i.$

RHONN Model Approximation Properties for Autonomous Systems

- MAPK cascade → autonomous dynamic system.
- Only E1's initial concentration affects the dynamic behaviors of proteins in the cascade.
- An autonomous can be modeled by a RHONN architecture!

Lemma: An autonomous system, with arbitrary initial conditions

described by the differential equation: $\dot{\chi}(t) = F(\chi(t))$ (1)

can behave dynamically exactly as the dynamical system, with given

initial conditions: $\dot{x}(t) = F(x(t)) + u$ (2)

if u is of the form: $u(t) = (\chi^0 - x^0) \cdot \delta(t)$ (3)

RHONN Model Approximation Properties for Autonomous Systems

Proof: Consider the two systems (1) and (2). By integrating these equations, we obtain:

$$\begin{aligned} \chi(t) &= \int_{0-}^{t} F(\chi(t))dt + \chi^0 \quad (4) \\ x(t) &= \int_{0-}^{t} F(x(t))dt + x^0 + \int_{0-}^{t} u(t)dt \quad (5) \end{aligned}$$

We choose the external input u to be (3). But the integral of the dirac function is:

$$\int_{0-}^{0+} \delta(t)dt = 1,$$

Then system (5) can be expressed:

$$\begin{aligned} x(t) &= \int_{0-}^{t} F(x(t)) + x^0 + \int_{0-}^{t} (\chi^0 - x^0)\delta(t)dt \\ &= \int_{0-}^{t} F(x(t))dt + \chi^0 \quad (6) \end{aligned}$$

Thus the input u shifts the systems (1) and (2) to the same initial state. So for T>0 their behavior is identical.

Filtered Error RHONN Model

- We proved that an autonomous system of the form (1) with arbitrary initial conditions can behave identical to a system of the form (2) with constant initial conditions, if the external input u is chosen as (3).

- We used a Recurrent Second Order NN model to approximate system (2) with external input (3).

- RHONN model composed by 22 neurons.

- Each neuron correspond to a specific protein of MAPK cascade.
- Neurons transfer function is the log-sigmoid: $s(x) = \frac{1}{1+e^{-x}}$

Filtered Error RHONN Model

- In a Recurrent Second Order NN z vector should include all outputs y_j and all possible combinations $y_j y_k$.
- But we have information about cascade's internal structure.
- A different z vector created for each protein/neuron.
- For a specific protein x_i, z_i vector includes 1st and 2nd order terms of proteins with which this protein interacts.

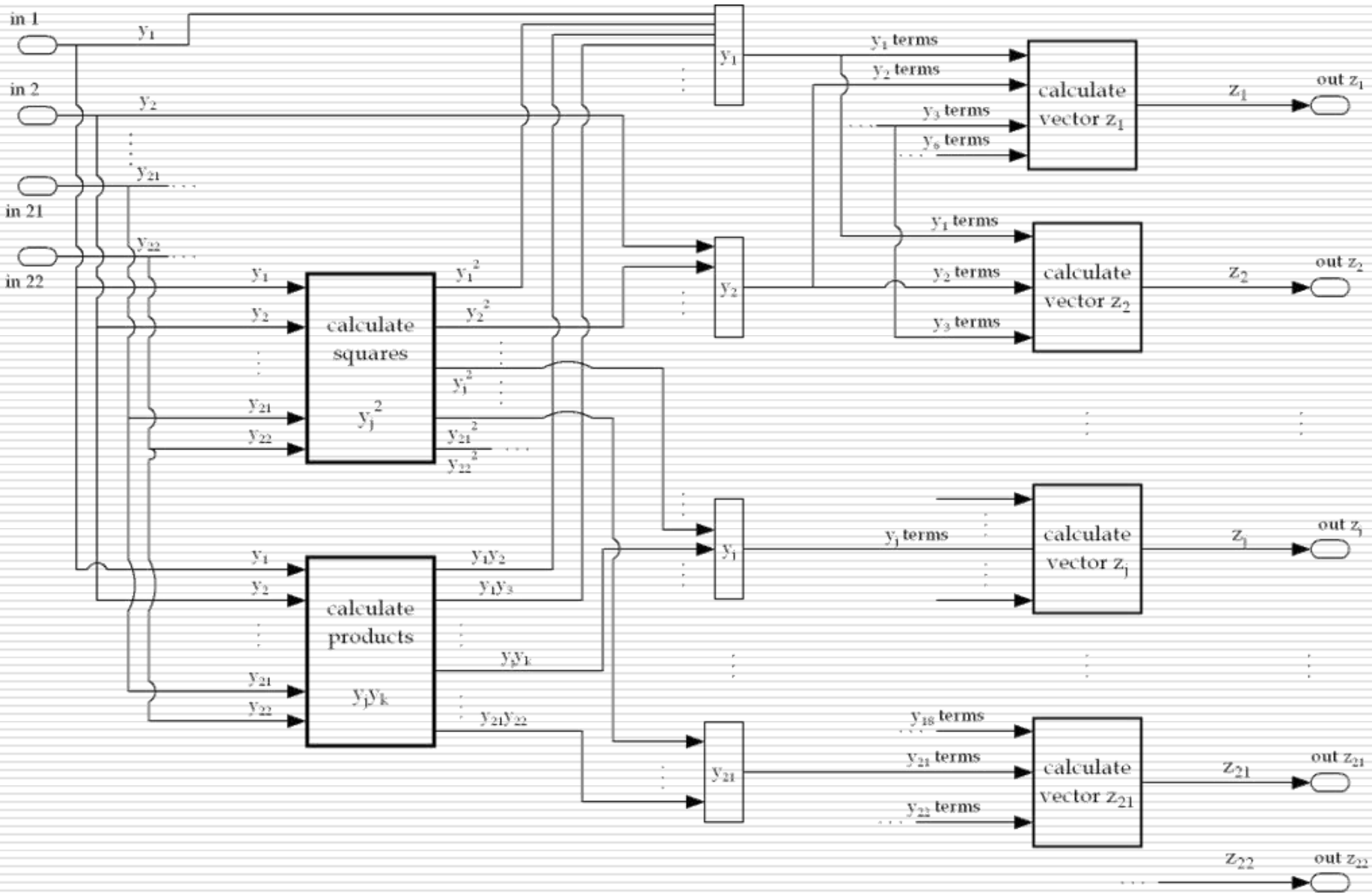
<u>Example</u>: z vector for E1 protein.

Differential Equation: $$\frac{d(E1)}{dt} = -a_1(KKK)(E1) + d_1(E1 - KKK) + k_1(E1 - KKK)$$

Denoting: $x_1 \rightarrow$ KKK, $x_2 \rightarrow$ E1, $x_3 \rightarrow$ E1-KKK, then

$$z_{E1} = [s(x_1) \quad s(x_2) \quad s(x_3) \quad s(x_1)^2 \quad s(x_2)^2 \quad s(x_3)^2 \quad s(x_1)s(x_2) \quad s(x_1)s(x_3) \quad s(x_2)s(x_3)]^T$$

Internal Structure of Simulink Block Calculating z vectors



RHONN training process

Training Algorithm:

1. Initialization of the w_i vectors with zero initial values. This step is performed only in the first iteration of the training algorithm.
2. Initialization of the a_i and γ_i parameters. This step is also performed only in the first iteration of the training algorithm.
3. Initialization of the RHONN and the real system in the same initial condition $x_i(0) = \chi_i(0)$.
4. Extraction of the training data from MAPK cascade model.
5. Training data passed through log-sigmoid function to z_i vector generator.
6. Evaluation of the RHONN state.
7. Calculation of error during training $e_i = x_i - \chi_i$.
8. Calculation of weight vectors w_i values.

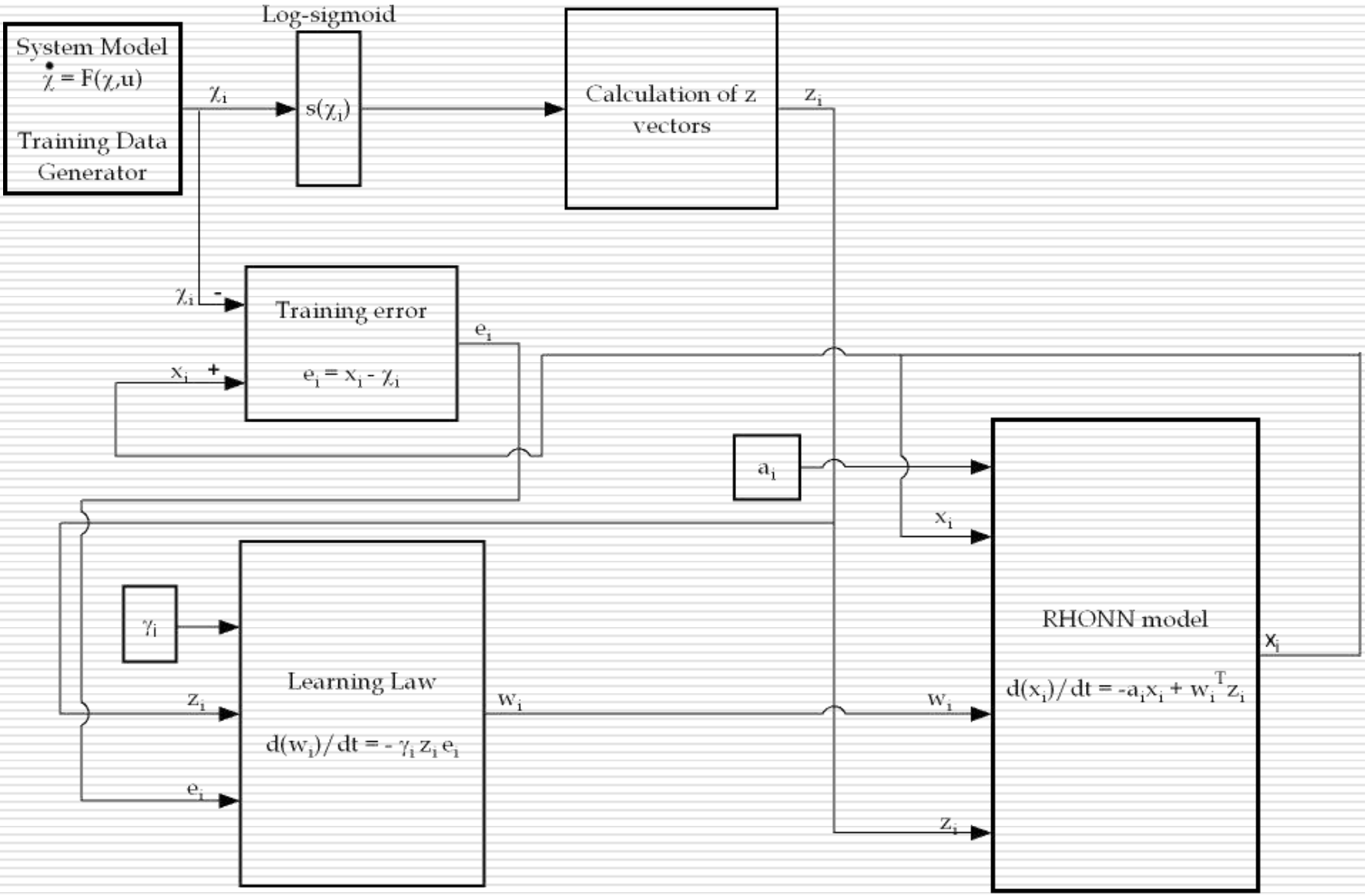
RHONN training process

9. The final w_i values in one iteration of the algorithm are set as initial values for w_i's in the next iteration of the training process.

10. The initial conditions of E1 neuron in RHONN and of E1 differential equation in training data generator are altered, and a new iteration of the algorithm begins.

11. The training algorithm (steps 3-10) continues until the error e_i is driven to an acceptable low value or a number of maximum epochs has reached.

Block Diagram of Simulink Model implementing Filtered Error RHONN training algorithm



RHONN training process

- In order to reliably train RHONN model, four (4) weight sets calculated.
- Each weight set corresponds to a specific range of E1 initial concentration values.
- This is due to the fact that proteins in cascade require more time to reach their steady-state as E1 initial concentration is decreasing.

E1 initial concentration interval	Calculated Weight set	Number of training samples
$8 \cdot 10^{-3}$ to 10^{-1}	w_1	23
$3 \cdot 10^{-4}$ to $8 \cdot 10^{-3}$	w_2	29
$3 \cdot 10^{-4}$ to 10^{-5}	w_3	23
10^{-6} to 10^{-5}	w_4	19

RHONN training process

Code used for training in E1 interval of [10^{-2} μM to 10^{-1} μM].

```
%initialize initial concentrations
InitConc;
%initialize learning rates
InitParam;
%initialize structure
for i = 1 : 66
xFinal.signals(1,i).values = [0]
end

E1mat = [0.01:0.005:0.1];
epoch = 0;
for j = 1 : 1 : 10
      for i = 1 : 1 : 19
```

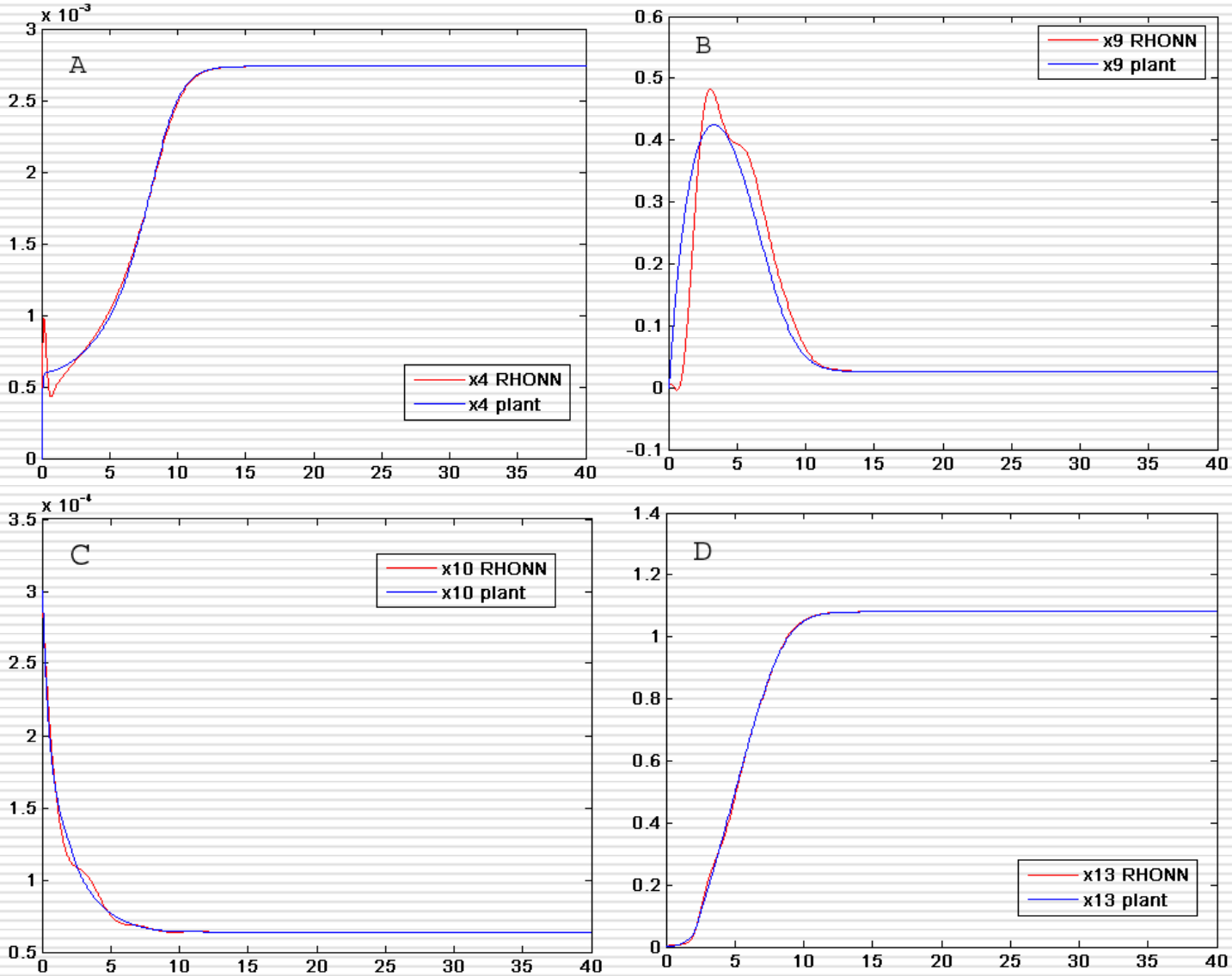
RHONN training process

```
        %random selection of E1 initial concentration
        index = randint(1,1,[1,19])
        E1 = E1mat(index);
        InitConc;

        for i = 1 : 1 : 1
            sim('rhonn train')
       end
      end
      epoch = epoch + 1
end
```

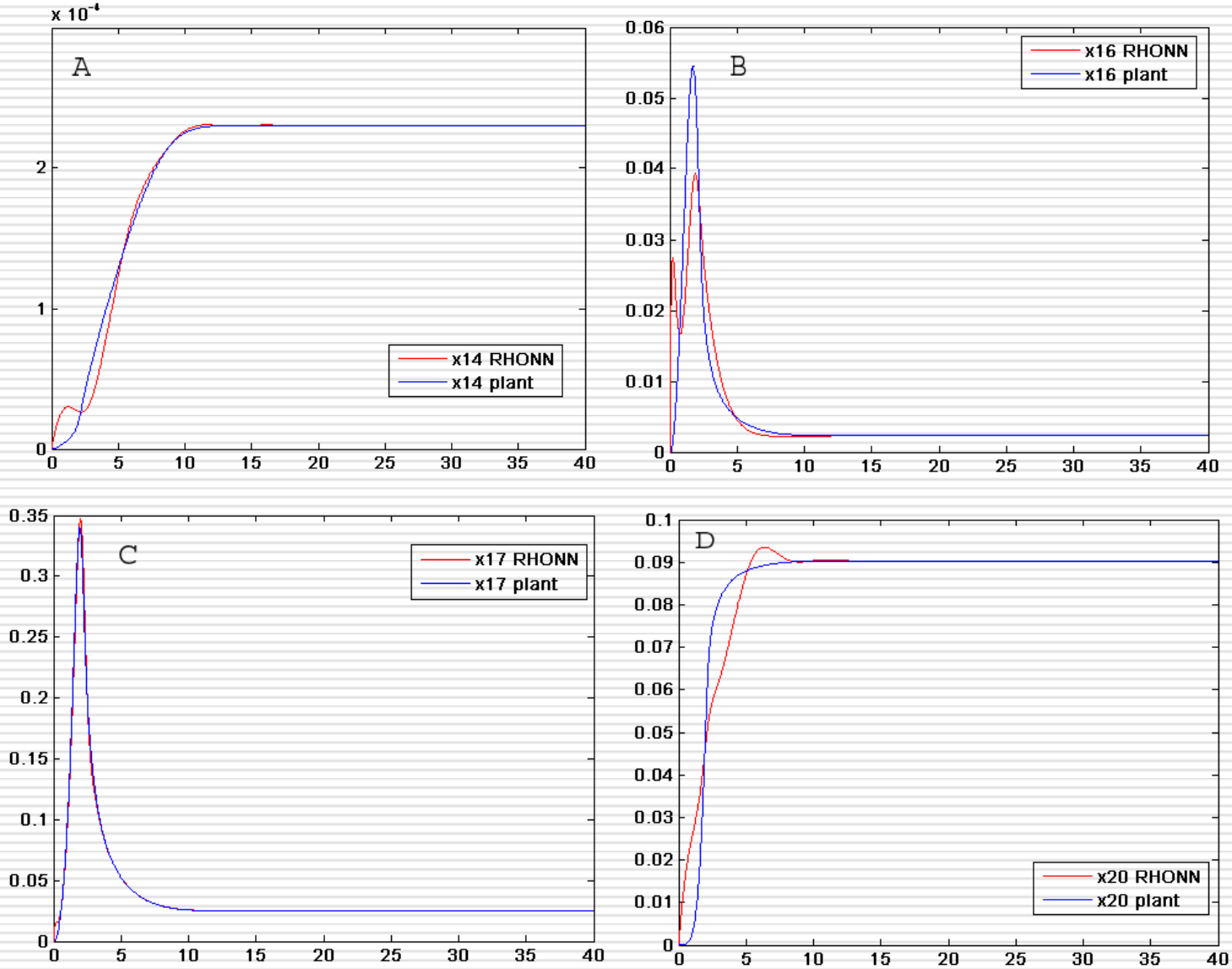
RHONN training process

weight set w_1 – E1 = 0.04 μM – epoch 140



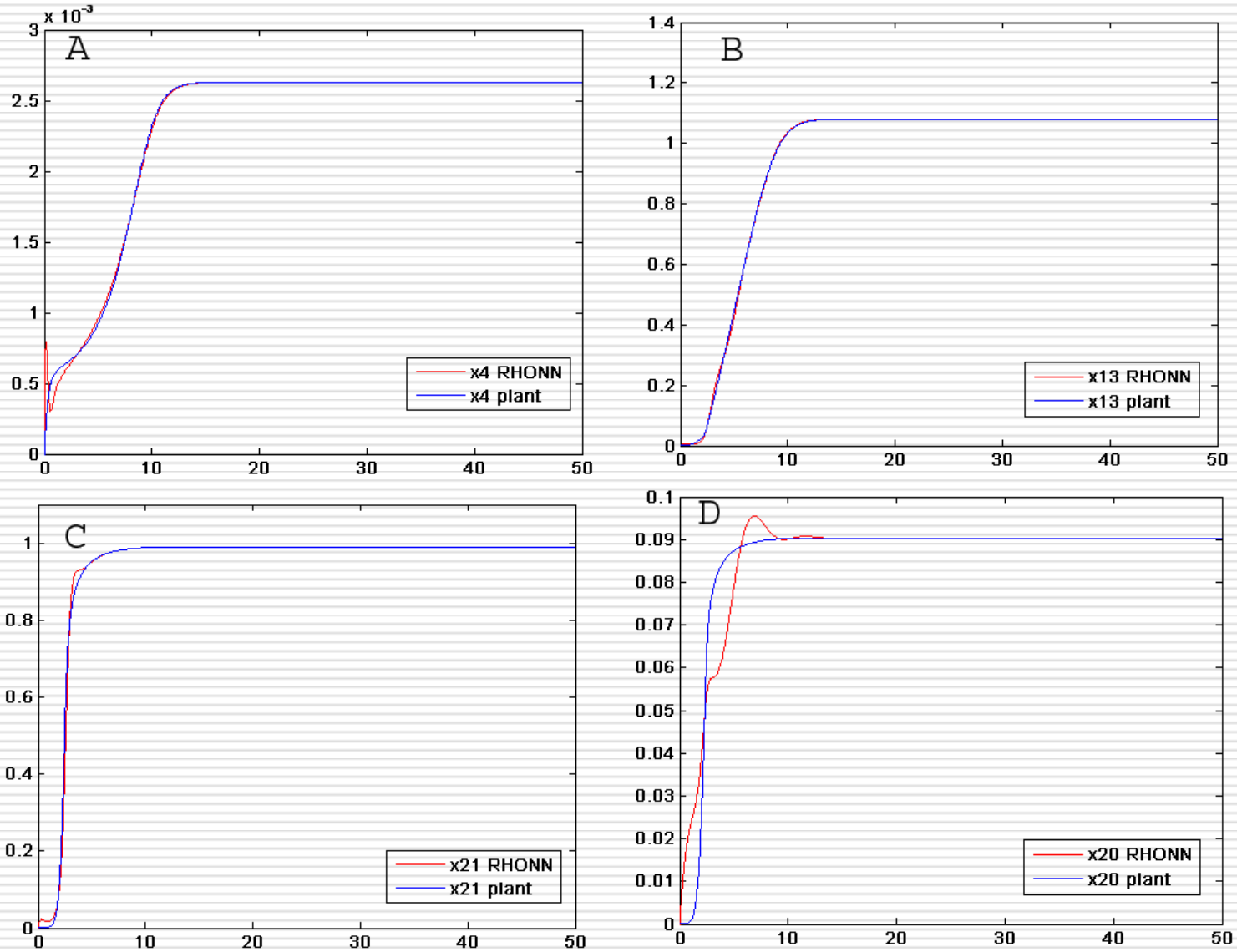
RHONN training process

weight set w_1 – E1 = 0.008 μM - epoch 140



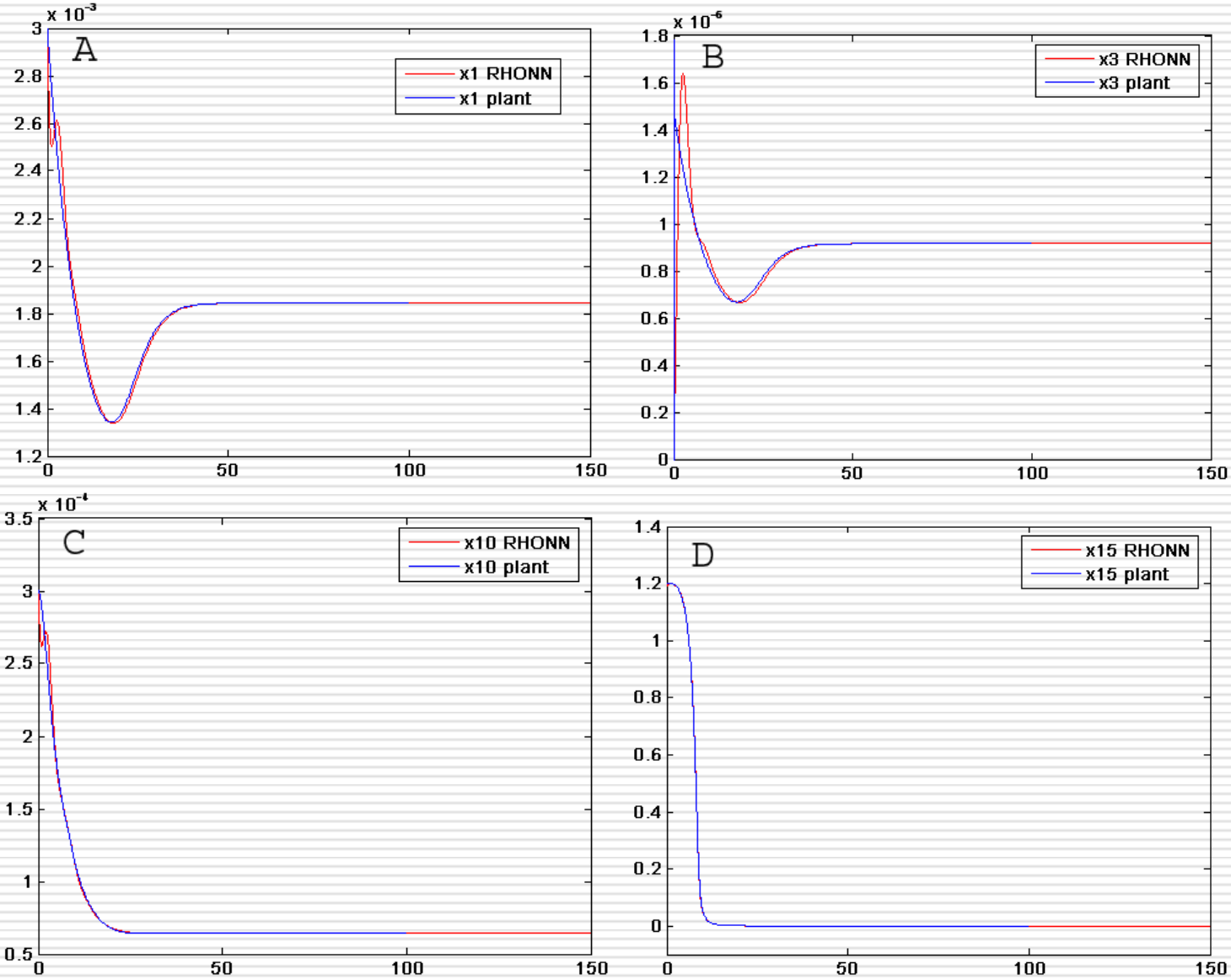
RHONN training process

weight set w_2 – E1 = $6 \cdot 10^{-3}$ μM - epoch 271



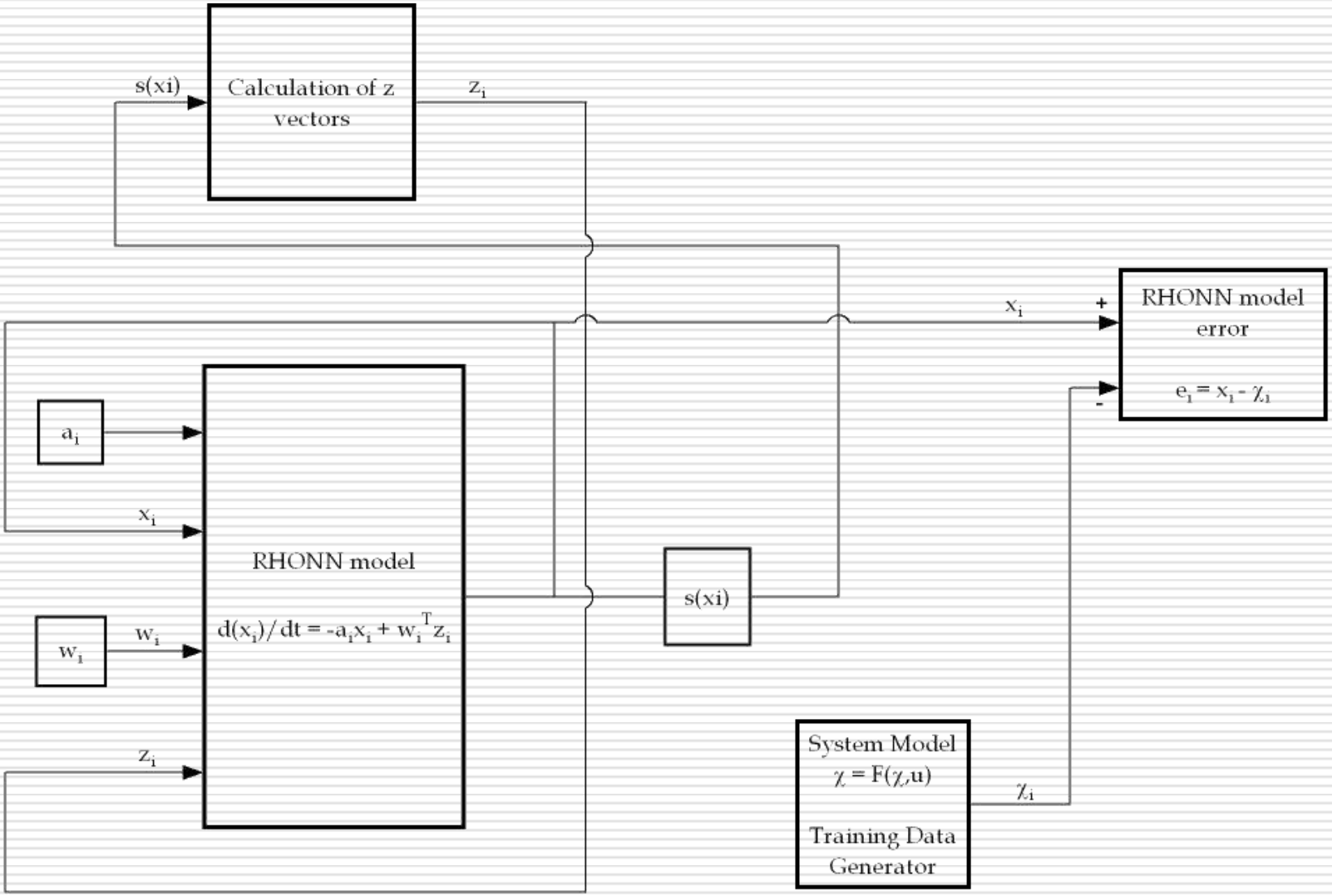
RHONN training process

weight set w_3 – E1 = $1.5 \cdot 10^{-4}$ μM - epoch 205



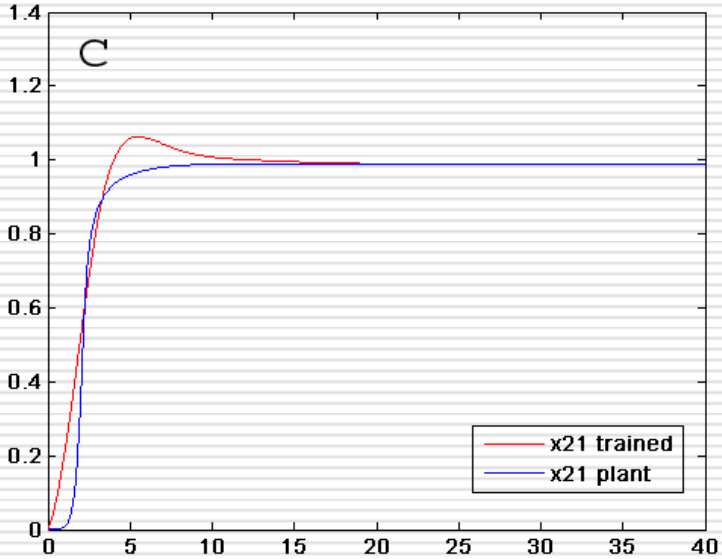
RHONN validation process

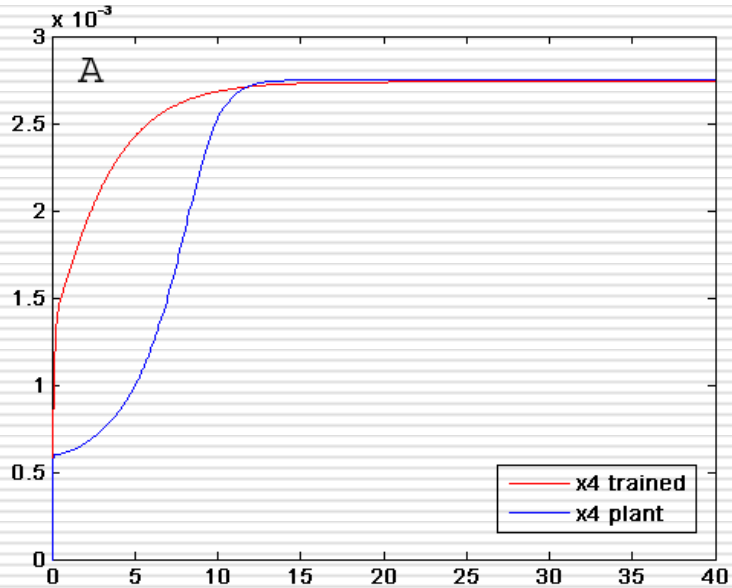
- Network should produce answers for unknown input stimuli.
- Model built in Simulink for validation procedure

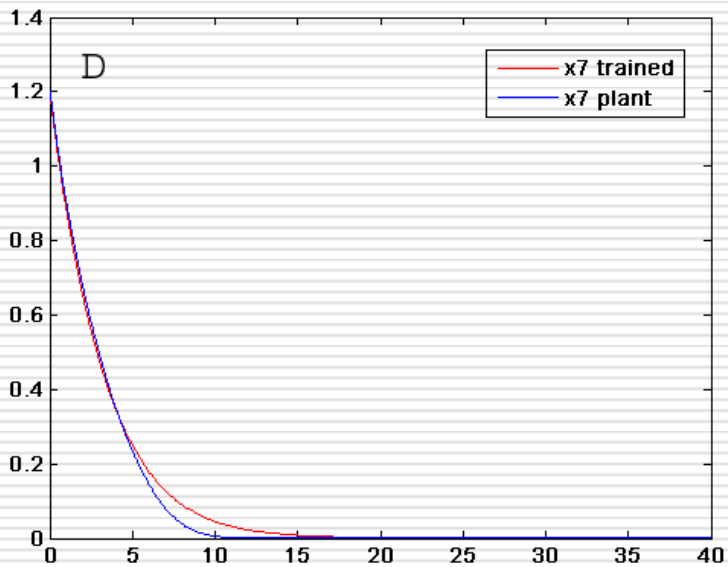


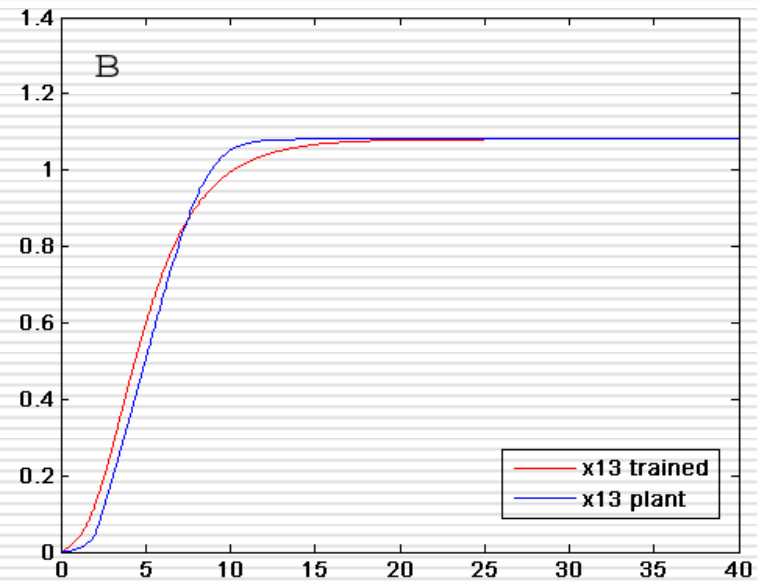
RHONN validation process

E1 1st interval – E1 = 0.088 μM unknown input stimulus



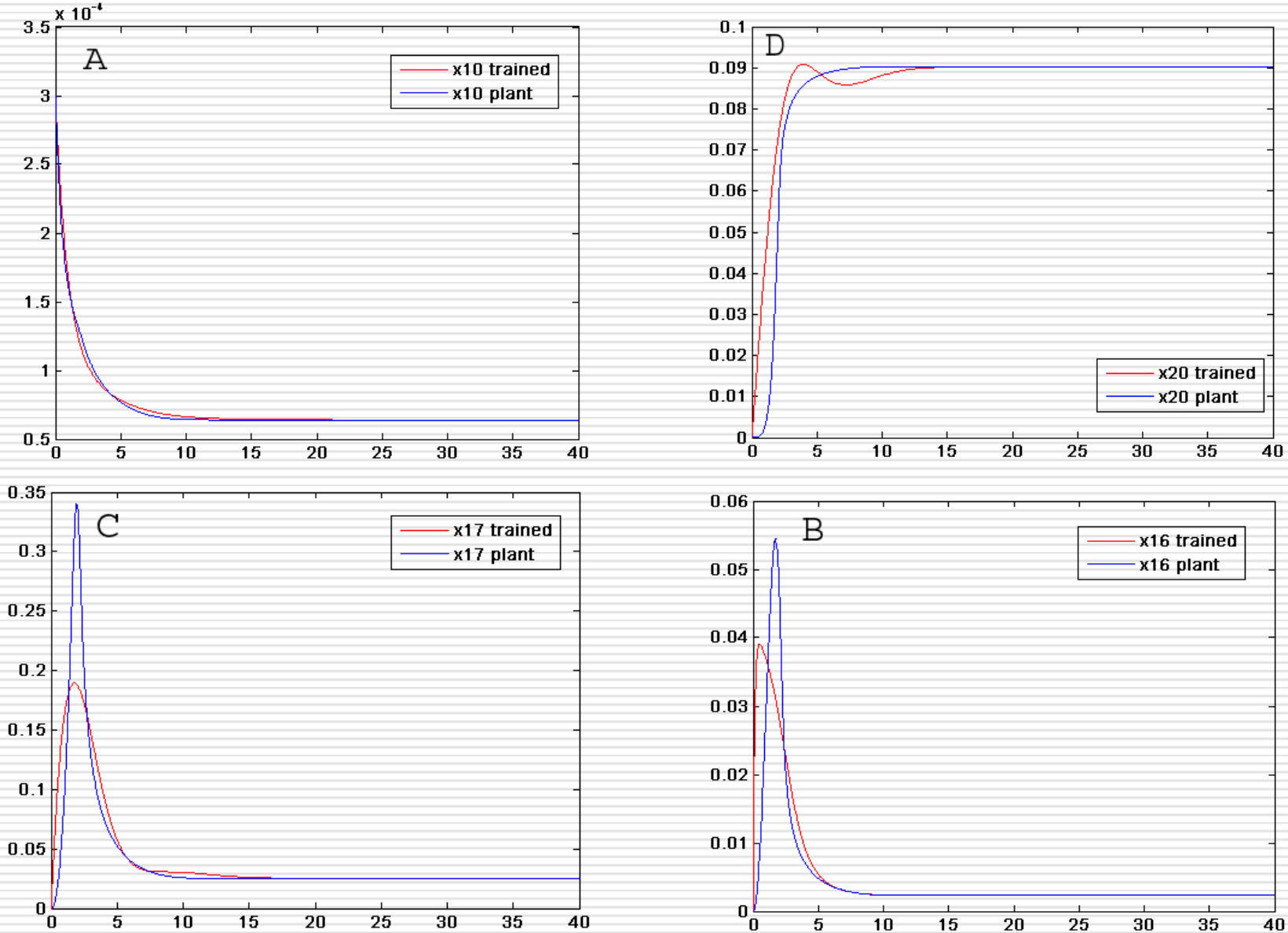






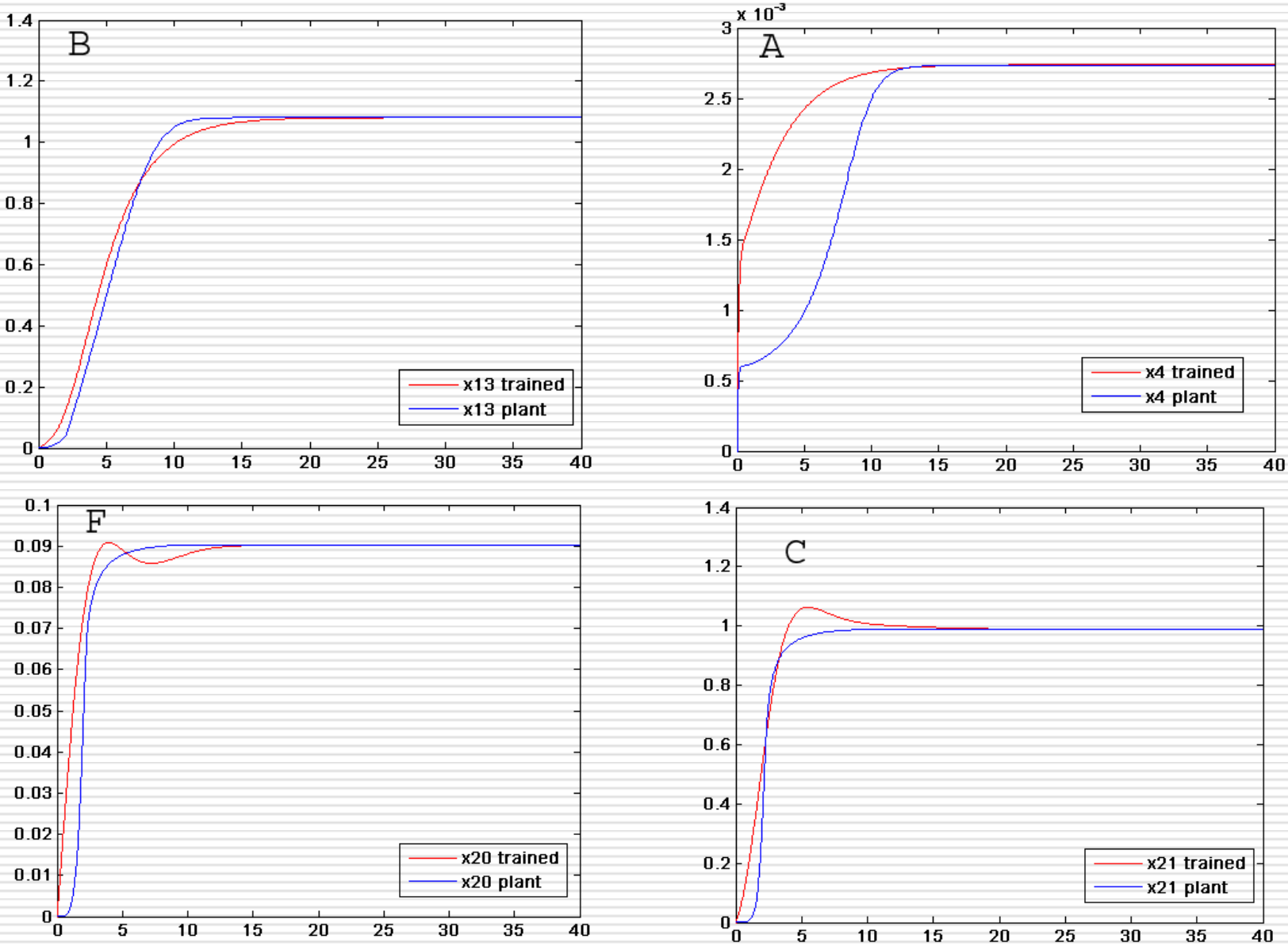
RHONN validation process

E1 1st interval – E1 = 0.088 μM unknown input stimulus



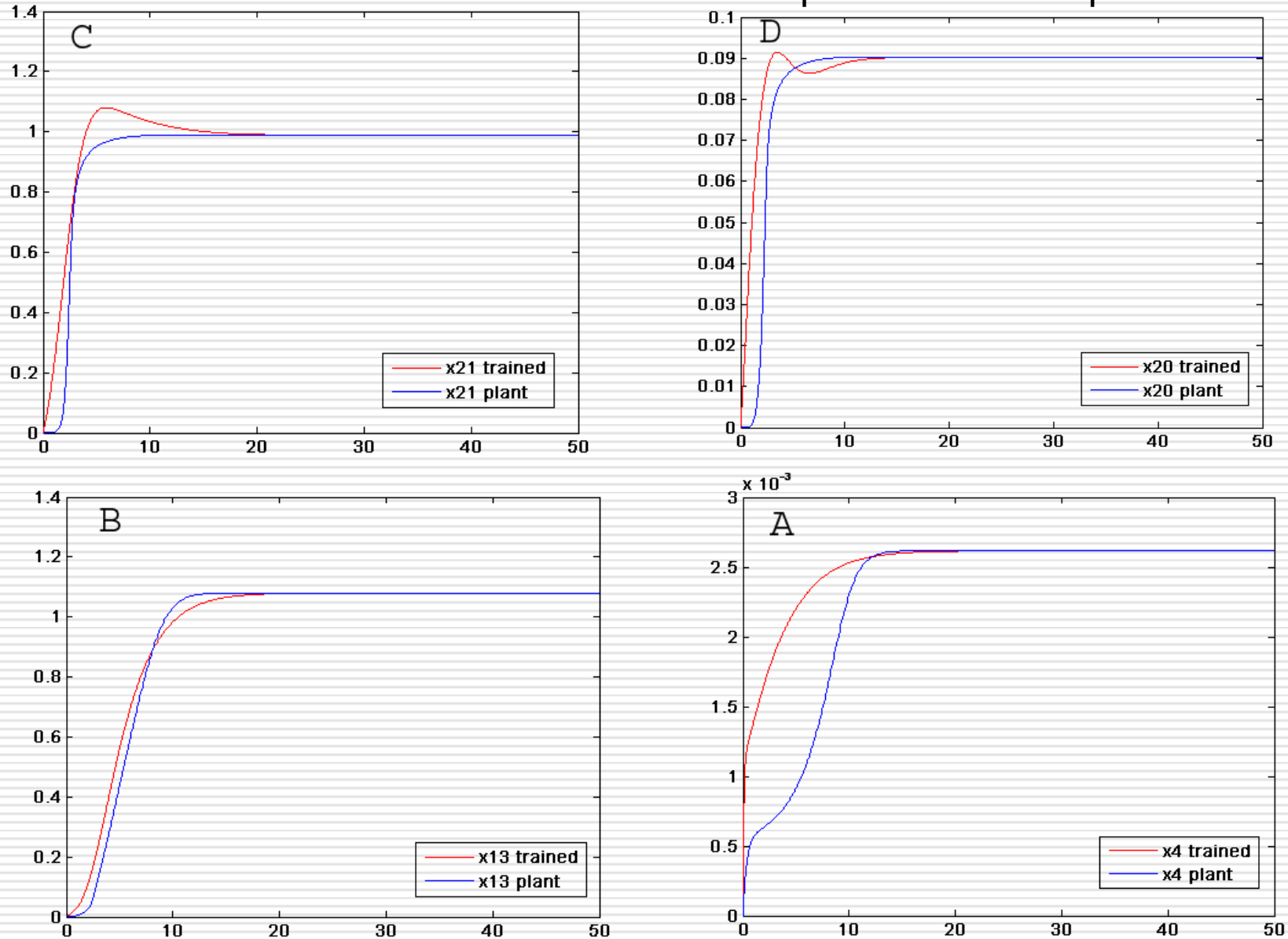
RHONN validation process

E1 1st interval – E1 = 0.0092 μM unknown input stimulus



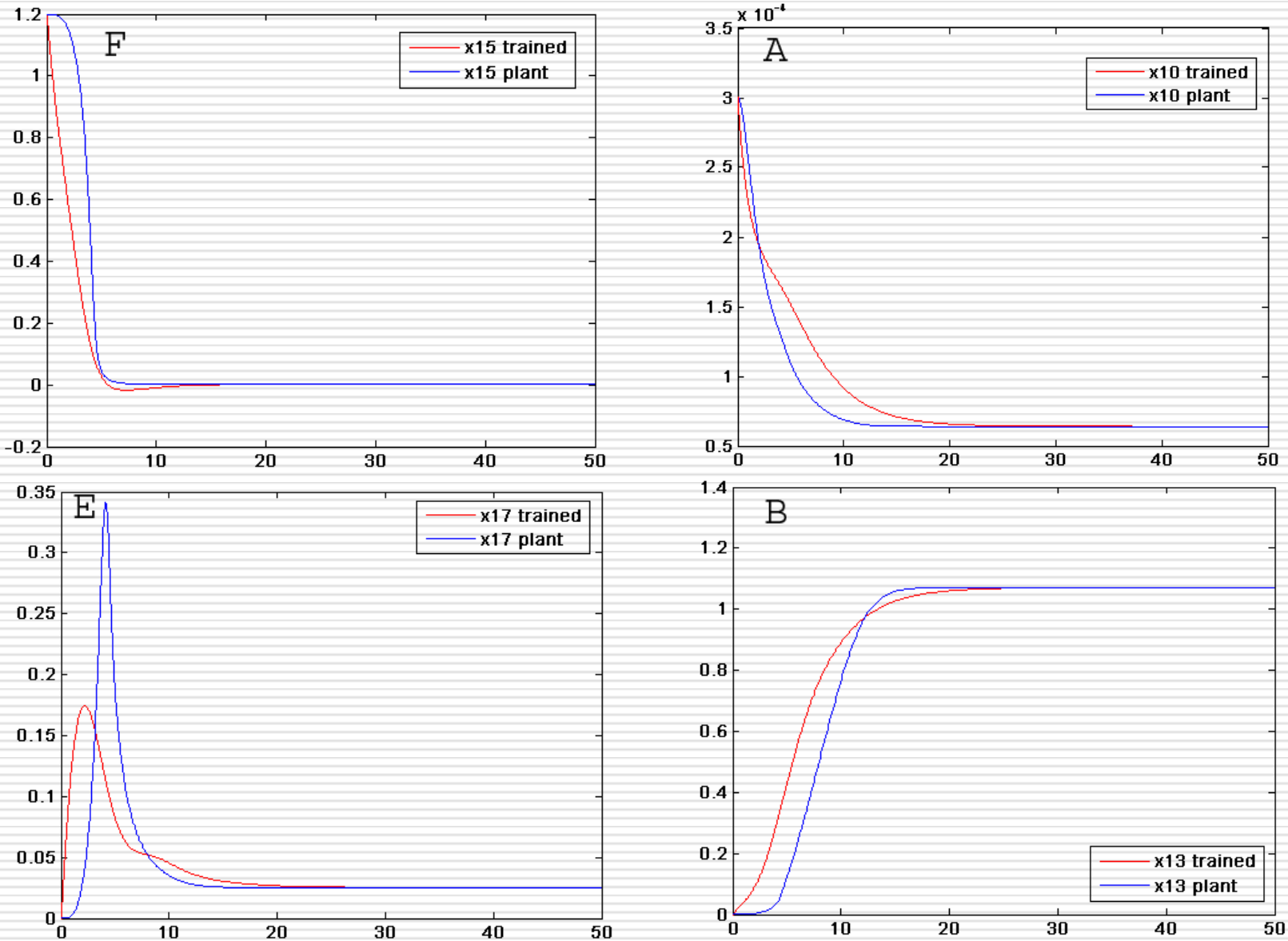
RHONN validation process

E1 2nd interval – E1 = 0.0058 μM unknown input stimulus



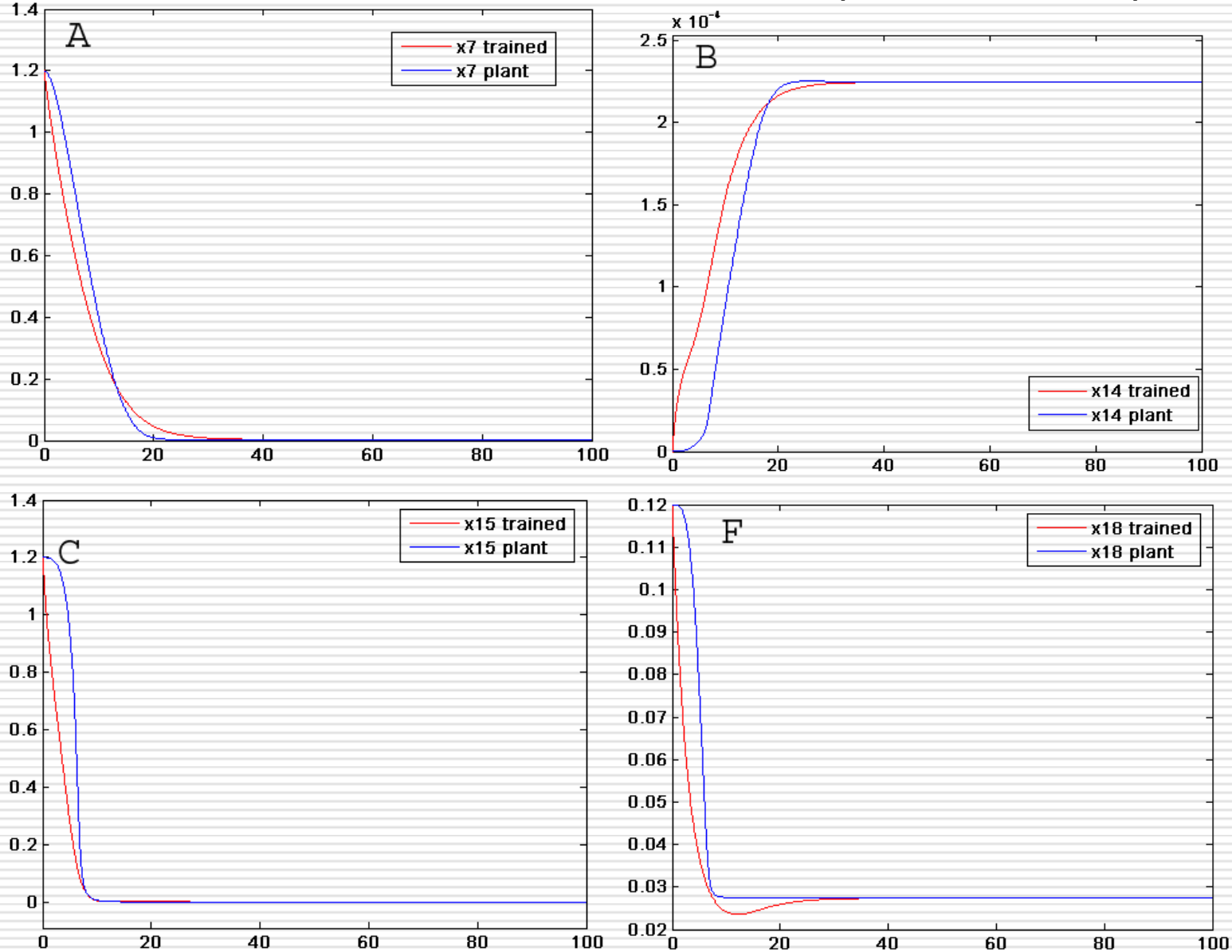
RHONN validation process

E1 2nd interval – E1 = 0.00073 μM unknown input stimulus



RHONN validation process

E1 3rd interval – E1 = $2.53 \cdot 10^{-4}$ μM unknown input stimulus



Conclusions

- RHONN has effectively learn to approximate the dynamic behavior of proteins in MAPK cascade.

- Mathematical tools developed in this work form an alternative for modeling and approximating complex biological systems.

- There is no longer need for extensive knowledge of system's internal structure.
- The only prerequisite is the existence of a relatively small amount of examples of I/O behavior.

- RHONNs are not constrained by system's complexity or large dimension.