การจำลองและการสร้างภาพนามธรรมของการเติบโตของถั่วเหลืองที่เป็นผลจากปริมาณของ ไนโตรเจน ฟอสฟอรัส และโพแทสเซียม ที่แตกต่างกัน

นายอภิชาต ศุรธณี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคณนา ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2546

ISBN 974-17-4591-5

SIMULATION AND VISUALIZATION OF SOYBEAN GROWTH AFFECTED BY DIFFERENT AMOUNT OF NITROGEN, PHOSPHORUS AND POTASSIUM

Mr. Apichat Suratanee

A Thesis Submitted in Partial Fulfillment of the Requirements
For the Degree of Master of Science in Computational Science
Department of Mathematics
Faculty of Science
Chulalongkorn University
Academic Year 2003

ISBN 974-17-4591-5

Thesis Title	SIMULATION AND VISUALIZATION OF SOYBEAN GROWTH AFFECTED BY DIFFERENT AMOUNT OF NITROGEN, PHOSPHORUS AND POTASSIUM
By	Mr. Apichat Suratanee
Field of study	Computational Science
Thesis Advisor	Associate Professor Suchada Siripant
Thesis Co-advisor	Professor Chidchanok Lursinsap, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master 's Degree

…… Dean of the Faculty of Science
(Professor Piamsak Menasveta, Ph.D.)

THESIS COMMITTEE

…… Chairman
(Assistant Professor Peraphon Sophatsathit, Ph.D.)

…….. Thesis Advisor
(Associate Professor Suchada Siripant)

…… Thesis Co-advisor
(Professor Chidchanok Lursinsap, Ph.D.)

…… Member
(Assistant Professor Supachitra Chadchawan, Ph.D.)

…… Member
(Siripun Sanguansintukul, Ph.D.)

อภิชาต ศุรธณี : การจำลองและการสร้างภาพนามธรรมของการเติบโตของถั่วเหลืองที่เป็นผลจากปริมาณของ ไนโตรเจน ฟอสฟอรัส และโพแทสเซียม ที่แตกต่างกัน (SIMULATION AND VISUALIZATION OF SOYBEAN GROWTH AFFECTED BY DIFFERENT AMOUNT OF NITROGEN, PHOSPHORUS AND POTASSIUM) อ.ที่ปรึกษา : รศ. สุชาดา ศิริพันธุ์, อ. ที่ปรึกษาร่วม : ศ.ดร. ชิดชนก เหลือสินทรัพย์ ; 68 หน้า. ISBN 974-17-4591-5.

วิทยานิพนธ์นี้ได้เสนอตัวแบบจำลองการเติบโตของถั่วเหลืองที่ได้รับผลจากธาตุอาหารหลัก ไนโตรเจน ฟอสฟอรัส และโพแทสเซียม โดยใช้โครงข่ายประสาทเทียมแบบเคลื่อนที่ไปข้างหน้าเป็นพื้นฐานของตัวแบบ ข้อมูลนำเข้าของตัวแบบประกอบด้วย เปอร์เซ็นต์ของปริมาณ ไนโตรเจน ฟอสฟอรัส และโพแทสเซียม เวลา และขนาดความสูงของถั่วเหลือง โดยตัวแบบนี้สามารถทำนายความสูงของถั่วเหลืองในระยะเวลาต่างๆ ผลของตัวแบบจะถูกนำมาสร้างภาพนามธรรมโดยระบบลินเดนเมเยอร์

ภาควิชา คณิตศาสตร์ ลายมือชื่อนิสิต……..

สาขาวิชา วิทยาการคณนา ลายมือชื่ออาจารย์ที่ปรึกษา……..

ปีการศึกษา 2546 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม……

4472484423: MAJOR COMPUTATIONAL SCIENCE
KEY WORD: SIMULATION / VISUALIZATION / NUTRIENTS / SOYBEAN / GROWTH

APICHAT SURATANEE : SIMULATION AND VISUALIZATION OF SOYBEAN GROWTH AFFECTED BY DIFFERENT AMOUNT OF NITROGEN, PHOSPHORUS AND POTASSIUM. THESIS ADVISOR: ASSOC. PROF. SUCHADA SIRIPANT, THESIS COADVISOR: PROF. CHIDCHANOK LURSINSAP, Ph.D., 68 pp. ISBN 974-17-4591-5.

This thesis proposed a simulation model of soybean growth which is effected by major nutrient factors, nitrogen, phosphorus and potassium. A feedforward neural network is used as a basis of the modelling. The combination of different percentage of nitrogen, phosphorus, potassium, time steps, and the collected height data of the soybean are used as inputs. The model can predict the height at designated time intervals, whereby the result can be visualized with Lindenmayer systems.

Department Mathematics Student's signature……
Field of study Computational Science Advisor's signature…….…….….
Academic year 2003 Co-advisor's signature…….……..

Acknowledgements

I am greatly indebted to my supervisor, Associate Professor Suchada Siripant and Professor Dr. Chidchanok Lursinsap, for their suggestions, untiring help and guidance of this thesis. Without them my thesis would have never been accomplished.

I also wish to express my special thanks to the thesis committee, Assistant Professor Dr. Peraphon Sophatsathit, Assistant Professor Dr. Supachitra Chadchawan and Dr. Siripun Sanguansintukul, for their valuable advise, reading and criticizing the manuscript.

This work is partially supported by a grant from the Nation Electronics and Computer Technology Center (NECTEC) through the AVIC Research Center.

I would like to thank Miss Thapana Akaraeakpanya and the Department of Botany for furnishing information of plant growth.

I am also greatful to the staff-members of the AVIC Research Center for their promptness and attitude.

Furthermore, I would like to thank all my past and present instructors for their valuable lectures and instructions. Finally, I would like to express my sincere gratitude to my parents and family members for their love, hearty encouragement, and unselfish sacrifice during my study.

I truly believe that all people whom I have not explicitly mentioned here are aware of my deep appreciation.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 The Problem Identification

Plant growth modelling is a real challenge for researchers and scientists, due to the high level of integrated multidisciplinary aspects. In order to understand the nature of plant growth, a plant functional and structural model in the form of mathematical formalism needs to be developed based on the knowledge from botany, agronomy, forestry, eco-physiology, and computer science.

A bulk of algorithms for simulating plant development has been published in the context of computer graphics[1, 2, 3, 4, 5, 6, 7, 8, 9], treating a plant as a closed system without considering an interaction of plant and its environment. The interaction between a plant and its environmental factor has been proposed in a specific assumption for computer graphics. Most of the previous works, regarding virtual plants, relate a the diversity of the plant growth that is caused by the environmental factors (temperature, humidity, photoperiod, amount of light, etc.)[1, 2, 10], whereas the effect of nutrient on the plant growth rate in terms of mathematical equations is discussed in this research.

1.2 The Objective of the Research

The goal of this research was to develop and validate a neural network model to predict the height of soybean grown in the nutrient solution of hydroponics in different amount of the primary nutrient, nitrogen, phosphorus and potassium.

1.3 The Scope of the Research

The following conditions are considered in this study:

We were interested in determining the requirements of primary nutrients for plants, which was recognized that it would be difficult to pursue such studies in a medium as complex as soil. Solution culture or hydroponics, growing plants in a preestablished nutrient solution, was the principal experimental system for the study of the plant nutrient requirements.

Data were collected from soybean experiments. We ignored some factors needed for plant growing, such as light, carbon dioxide, water and soil. We measured individually the structural development of each plant made up of components like internode length.

We had created a mathematical model that captured the development of plants by using a neural network incorporating biological data. L-systems was used to represent the plant topology and visualization development. There are three steps in this research: (1) collection of data from soybean experiment, (2) development of a mathematical model, (3) simulation and visualization of soybean development.

1.4 The Outline of the Thesis

This thesis is organized as follows. Chapter II reviews the related literatures. Chapter III describes the theoretical background. Chapter IV explains the experimental design and plant modelling. The simulation and visualization results are shown in Chapter V. Some final thoughts are summarized in Chapter VI.

CHAPTER 2

LITERATURE REVIEWS

2.1 The Review of Literature Related to L-systems

L-systems is a method of rewriting. The basic idea is to define complex objects by successively replacing parts of a simple object with a set of rewriting rules or productions. The rewriting can be recursively carried out.

The most extensively studied and the best understood rewriting systems operate on character strings was Chomsky's work on formal grammars (1957). Aristid Lindenmayer's work introduced a new type of string rewriting mechanism, subsequently termed L-systems. The essential difference between Chomsky grammars and L-systems is the method of applying productions. Chomsky grammar productions are applied sequentially, whereas L-systems are applied in parallel by simultaneously replacing all letters in a given word. This difference reflects the biological motivation of L-systems. The productions are intended to observe cell divisions in multicellular organisms, while many divisions may occur at the same time.

Recently, L-systems has found several applications in computer graphics (Smith 1984; Prusinkiewicz and Hanan 1989; Prusinkiewicz and Lindenmayer 1991). In 1984, Alvy Ray Smith, showed how L-systems could be used to synthesize realistic

images. He also pointed out the relationship between the concept of Fractals and L-systems [11].

In 1996, Tong Lin implemented the animation of L-systems based on three-dimensional plant growing in Java [7]. His animation used a number of iterations to animate a plant development.

Prusinkiewicz, James, and Mech extended L-systems[12] in manner suitable for simulating an iteraction of a developing plant and its environment. The formalism was illustrated by modeling the response of trees to pruning. It yields synthetic images of plant sculpture found in topiary gardens.

Hammel and Prusinkeiwicz extended the notation of L-systems with turtle interpretation [13] to facilitate the construction of other objects. The extension was based on the interpretation of the entire derivation graph generated by the L-systems, as opposed to the interpretation of individual words. The illustration of the proposed method visualizes the development of compound leaves, a sea shell pigmentation pattern, and a filamentous bacteria.

Prusinkiewicz, Hammel, and Mjolsness introduced a combined discrete/continuous model of plant development that integrates L-system-style productions and differential equations [4]. The model was suitable for simulated developmental processes in a manner resembling time-lapse photography. The proposed techniques were illustrated, using several developmental models including the flowering plants.

Prusinkiewicz and Kari expressed the development of modular branching structures [14] that satisfied three assumptions: (a) subapical branching, meaning that new branches could be created only near the apices of the existing branches, (b)

finite number of module types and states, and (c) absence of the interactions between coexisting components of the growing structure. These assumptions were captured in the notion of subapical bracketed deterministic L-systems without interactions (sBOL-systems). They presented the biological rationale for sBOL-systems and proved that it was decidable whether a given BOL-system was subapical or not.

Hemmel, Prusinkiewicz, Remphrey, and Davidson presented a methodology for creating models that captured the development of plant using the formalism of L-systems and incorporating biological data using *Fraxinus pennsylvanical* shoots based on L-systems [15].

Hemmel, Prusinkiewicz, and Wyvill proposed a method for modelling compound leaves in plants[8]. The layout of leaf lobes was captured by a branching skeleton generated using an L-systems. The leaf margin was then traced around the skeleton. Their work focused on the specification and tracing of the margin, and including references to the techniques described in the literature for performing the other tasks. The margin was defined as an implicit contour.

2.2 The Review of Literatures Related to Plant Model

Lintermann and Deussen presented a modelling method and graphical user interface for the creation of natural branching structures such as plants[9, 16]. Structural and geometric information are encapsulated in objects that are combined to form a description of the model. The model was represented graphically as a structure graph and could be edited interactively. Global and partial constraint techniques were integrated on the basis of tropism, free-form deformations

and pruning operations to allow the modelling of specific shapes.

Deussen, O. developed a system built around a pipeline of tools [17]. The terrain was designed using an interactive graphical editor. Plant distributing was determined by hand, ecosystem, or a combination of both techniques.

AMAP models [18, 19, 20] were originally developed for mainly simulating trees. They use (1) the qualitative knowledge provided by Halle et al. in plant architecture and (2) the quantitative methods perfected within the Plants Modeling Unit of CIRAD. These models were first concerned with tree architecture and landscape visualization and then turned to some agronomic applications but their use was somewhat complex and mainly limited to specialists.[21]

Chiba *et al.*[3] argued that the previous works did not simulate the virtual plant shape formed by unexpected changes in the growth environment, i.e., light and random pruning. If we could make use of the models which would capture these interactions, we would be able to predict the development of huge structures as well as particular entities. We could, for example, predict the shape of a certain tree situated in a shady corner of a yard, or we could estimate the spreading of insects in fields, *etc.*

Benes Bedrich[10] introduced the method principle, and showed how to estimate the incoming amount of light, one of the most important external forces that caused a significant response by the plant. The calculated values were used to determine the growth direction of the buds.

Most of the previous works [10, 22] treated the diversity of plant shapes caused by the external influences. They say that the differences in the shape can be approximated by injecting randomness when the three dimensional structure of

the virtual plant is generated.

2.3 Nutrient Model Literature Review

In 1971, Greenwood et al. presented a simple semi-mechanistic model that described how crop yield would respond to fertilizer on the basis of a "snapshot" of soil condition at planting. That model avoided most of the problems about the interactions between the effect of nutrients.

The effect of a single element, especially N, on crop growth was described by Muchow and Sinclair (1995) [23], and Jamieson and Semenov (2000) [24]. The concept of critical nitrogen concentration on crop growth under optimal and sub-optimal N-supply was proposed and used to predict crop growth (Greenwood e.t. al., 1991).

A conceptually simple model (PARJIB) [25] showed the effects of nutrient supply on crop yield and after the model calibration, it was demonstrated to forcast and analyse the influence of nutrient supply on maize(*Zea mays* L.) yield [26].

CHAPTER 3

BACKGROUND KNOWLEDGE

3.1 An overview of Artificial Neural Networks

The mathematics of neural network is derived from the understanding how biological neural networks in the human brain memorize patterns and learn information. Individual neurons in the human brain (there are over 100 billion) are connected to many other neurons to form a complex network that has the capacity to recognize patterns and learn information.

The basic computational element of a biological neural network is a neuron, which consists of synapses, dendrites, soma, and axons (Figure 3.1). The synapse is an area of electrochemical contact between neurons that transfers electrochemical voltages from axons of nearby neurons to the dendrite, which serves as an input channel to the neuron body (soma). Potassium ions are the primary source for the electrochemical voltage. The dendrite can change the membrane permeability to potassium ions, effectively adding a resistance, or weighting factor, that modifies the signal transmitted to the soma. The soma evaluates the inputs from the dendrites and determines whether to give an output.

An artificial neuron that mimics a biological neuron is the basic computational element in an artificial neural network (Figure 3.1). Each artificial neuron contains

input channels, an activation function, and an output that mimics the synapse, dendrite, soma, and axon of a biological neuron. Input signals representing information are transmitted to the neuron by multiplying the input signal ($\mathbf{X}_i$) and the weight of the input connection ($\mathbf{W}_i$). The sum of the input signals ($\mathbf{X}_i\mathbf{W}_i$) is then transformed to an output signal ($\mathbf{Y}_i$) by a transfer function that is typically a sigmoidal function ranging from 0 to 1. This transformation function mimics the firing mechanism of the biological neuron body.

Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for artificial neural networks as well. Learning typically occurs by example through training, or exposure to a truthed set of input/output data where the training algorithm iteratively adjusts the connection weights (synapses). These connection weights store the knowledge necessary for solving specific problems.

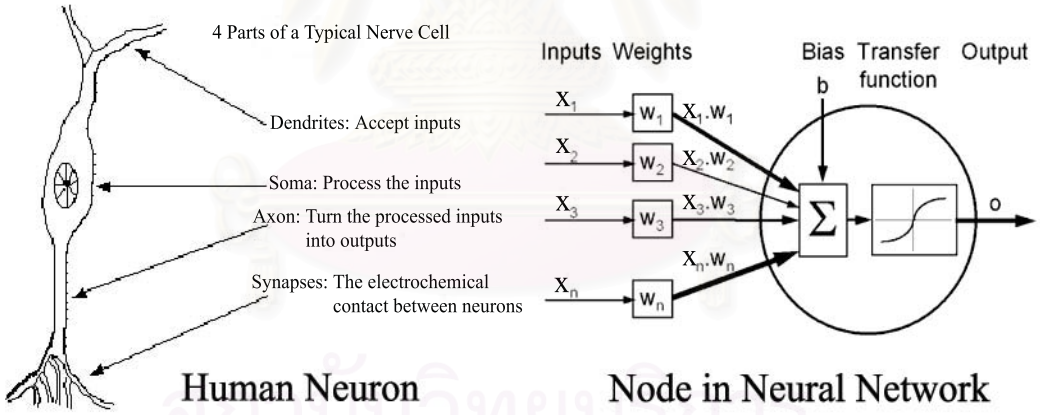


Figure 3.1: A Human Neuron and a Node of Neural Network

3.1.1 Feedforward Multilayer Perceptron

Figure 3.2 shows a standard three-layered feedforward multilayer perceptron (MLP). This type of architecture is part of a large class of feedforward neural networks with the neurons arranged in cascaded layers. The neural network architectures in this class share a common feature that all neurons in a layer are connected to all neurons in adjacent layers through unidirectional branches. This feature is the forward direction, i.e., the branches or links can only broadcast information in one direction. The branches have associated transmittances, that is, synaptic weights, that can be adjusted according to a set of predefined learning rules. Feedforward networks do not allow connections between neurons within any layer of the architecture. At every neuron the output of the linear combiner, that is, the neuron activity level v_q, is input to a nonlinear activation function $f(\bullet)$, whose output is the response of the neuron. The neurons in the network typically have activity levels in the range [-1,1], and in some applications the range [0,1] is used.

In Figure 3.2, there are actually four layers. Therefore, we see that the input layer in Figure 3.2 does not perform any computations, but only serves to feed the input signal to the neurons of the hidden layer. The outputs of the hidden layer are then input to the output layer. The output of the output layer is the network response vector. The network can perform the *nonlinear input/output mapping* $\Omega : \Re^{n\times 1} \rightarrow \Re^{p\times 1}$. In general, there can be any number of hidden layers in the architecture; however, from a practical perspective, only one or two hidden layers are typically used.

In Figure 3.2, each layer has a synaptic weight matrix associated with all

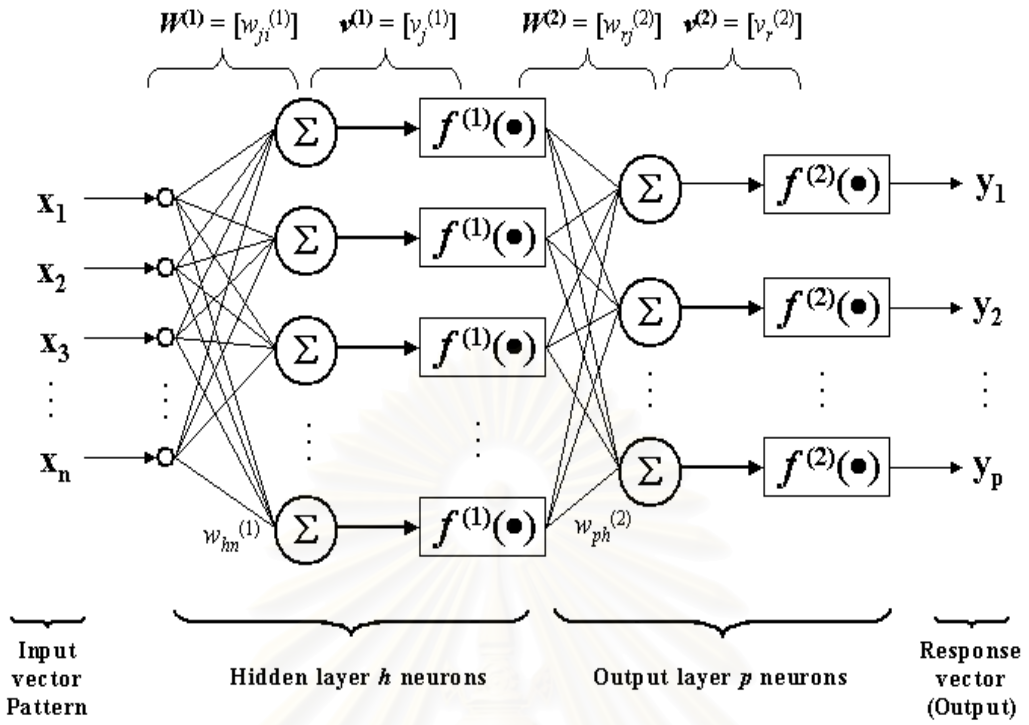


Figure 3.2: Feedforward three-layer perceptron architecture, where $i = 1, 2, \ldots, n; j = 1, 2, \ldots, h; r = 1, 2, \ldots, p$; and $f^{(1)}(\bullet)$ is the nonlinear activation function at each neuron in the second layer, $f^{(2)}(\bullet)$ is the nonlinear activation function at each neuron in the third layer.

the connections made from the previous layer to the next layer, that is, $\mathbf{W}^{(\ell)}$, for $\ell = 1, 2$. The first layer has the weight matrix $\mathbf{W}^{(1)} = \left[w_{ji}^{(1)}\right] \in \Re^{h \times n}$, the second layer's weight matrix is $\mathbf{W}^{(2)} = \left[w_{rj}^{(2)}\right] \in \Re^{p \times h}$, for $i = 1, 2, \ldots, n; j = 1, 2, \ldots, h; and\ r = 1, 2, \ldots, p;$. The nonlinear input-output mapping $\Omega : \Re^{n \times 1} \rightarrow \Re^{p \times 1}$ can be determined directly from Figure 3.2 as follows.

First we define that $f^{(1)}(\bullet)$ is the nonlinear activation function in the hidden layer and that $f^{(2)}(\bullet)$ is the nonlinear activation function in the output layer.

Given the network input vector $\mathbf{x} \in \Re^{n\times 1}$, the output of the hidden layer $\mathbf{x}_{out1} \in \Re^{h\times 1}$ can be written as

$$\mathbf{x}_{out1} = f^{(1)}(\mathbf{v}^{(1)}) = f^{(1)}(\mathbf{W}^{(1)}\mathbf{x}) \tag{3.1}$$

which is the input to the output layer. The output of the output layer, which is the response of the network $\mathbf{y} = \mathbf{x}_{out2} \in \Re^{p\times 1}$ can be written as

$$\mathbf{x}_{out2} = f^{(2)}(\mathbf{v}^{(2)}) = f^{(2)}(\mathbf{W}^{(2)}\mathbf{x}_{out1}) \tag{3.2}$$

Substituting (3.1) into (3.2) for x_{out1} gives the final response of the network as

$$\mathbf{y} = f^{(2)}(\mathbf{W}^{(2)} f^{(1)}(\mathbf{W}^{(1)}\mathbf{x})) = \Omega[\mathbf{x}] \tag{3.3}$$

A training process must be carried out by adjusting the weights to perform a desired mapping, for example, to solve a function approximation problem. Details concerning training of MLPs using backpropagation are discussed in the next section.

3.1.2 The Backpropagation Learning Algorithm

We now consider the supervised learning in a feedforward multilayer perceptron (MLP). Specifically, we want to study the backpropagation algorithm, or the generalized delta rule, for training the MLPs. Backpropagation is the most widely used learning process in neural networks, and it was first developed by Werbos in 1974. However, this work remained unknown for many years. This method has been rediscovered several times, in 1982 by Parker, in 1985 by LeCun, and by Rumelhart et al. in 1986. The presentation of backpropagation by Rumelhart

et al. is probably responsible for the popularization of the algorithm in the areas of science and engineering. Training MLPs with backpropagation algorithms results in a nonlinear mapping or an association task. Thus, given two sets of data, the MLP can have its synaptic weights adjusted by backpropagation algorithm to develop a specific nonlinear mapping. The MLP, with fixed weights after the training process, can provide an association task for classification, pattern recognition, diagnosis, etc. During the training phase of the MLP, the synaptic weights are adjusted to minimize the disparity between the actual and desired outputs of the MLP, averaged over all input patterns (or learning examples).

Standard backpropagation algorithm

Weight Initialization

Each weight is initialized to a small random value.

Calculation of activation function

1. The activation level of an input unit is determined and fed to the network.
2. The activation level O_j of a hidden and output unit is determined by

$$O_j = F(\sum W_{ji} O_i + \theta_j) \tag{3.4}$$

where W_{ji} is the weight of an input O_i, θ_j is the node threshold, and F is the activation function.

Weight Training

1. Start at the output units and work backward to the hidden layers recursively to adjust the weights by

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji} \tag{3.5}$$

where $W_{ji}(t)$ is the weight from unit i to unit j at time t (or t the iteration) and ΔW_{ji} is the weight adjustment.

2. The weight change is computed by

$$\Delta W_{ji} = \eta \delta_j O_i \tag{3.6}$$

where η is a trial-independent learning rate ($0< \eta <1$, e.g., 0.3) and δ_j is the error gradient at unit j. Convergence is sometimes faster by adding a momentum term:

$$W_{ji}(t+1) = W_{ji}(t) + \eta \delta_j O_i + \alpha [W_{ji}(t) - W_{ji}(t-1)] \tag{3.7}$$

where $0< \alpha <1$

3. The error gradient is given by:

- For the output units:

  $\delta_j = (T_j - O_j) F'(net_j)$

  where T_j is the desired (target) output activation, O_j is the actual output activation at output unit j and $net_j = \sum_i W_{ji} O_i$.

- For the hidden unit

  $\delta_j = F'(net_j) \sum_k \delta_k W_{kj}$

  where unit j is a hidden unit, δ_k is the error gradient at unit k to which a connection points from hidden unit j.

4. Repeat iterations until convergence in terms of the selected error criterion or a maximum number of iterations is reached.

The fundamental of backpropagation algorithm employed is a gradient descent optimization procedure which minimizes the mean squared error between network output and the desired output of all input patterns as follows:

$$E = \frac{1}{2P} \sum_{p=1}^{P} \sum_{h=1}^{n_s} (d_{ph} - out_{ph})^2$$

where P is the total number of training pattern, n_s is the number of neurons in the output layer, d_{ph} is the desired output of the hth neuron in the output layer to the pth training input, and out_{ph} is the actual output of the hth neuron in the output layer to the pth training input.

The training set is presented iteratively to operate the network, whereby the weights are updated until their values become stabilized according to the following criterion: (1) a user-defined error tolerance is achieved, or (2) a maximum number of iterations is reached.

3.2 Hydroponic Systems

Solution culture or hydroponics [27], growing plants in a defined nutrient solution, is the principal experimental system for study of plant nutrient requirements. Over the years, a large number of nutrient solutions have been formulated for studying the nutritional requirements of plants. Most modern formulations are based on a solution originally developed by D.R. Hoagland, a pioneer in the study of plant mineral nutrition. Individual investigators may introduce minor modifications to the composition of the nutrient solution in order to accommodate specific needs. Such formulations are commonly referred to as modified Hoagland's solution.

In the simplest form of solution culture, a seedling is supported in the lid of a container, with its roots free to grow in the nutrient solution (Figure 3.3). Noting that the solution must be aerated in order to obtain the optimal growth. It is often necessary to bubble air through the solution surrounding the roots. The water-culture method often fails because of inadequate aeration of the solution. The solution that is not aerated becomes depleted of oxygen, a condition known as anoxia. Anoxia inhibits the respiration of root cells and reduces nutrient uptake. We are aerating the roots by using an aquarium air pump. It is advisable not to stir the solution too vigorously as we may damage the tender roots and cause poor plant growth.

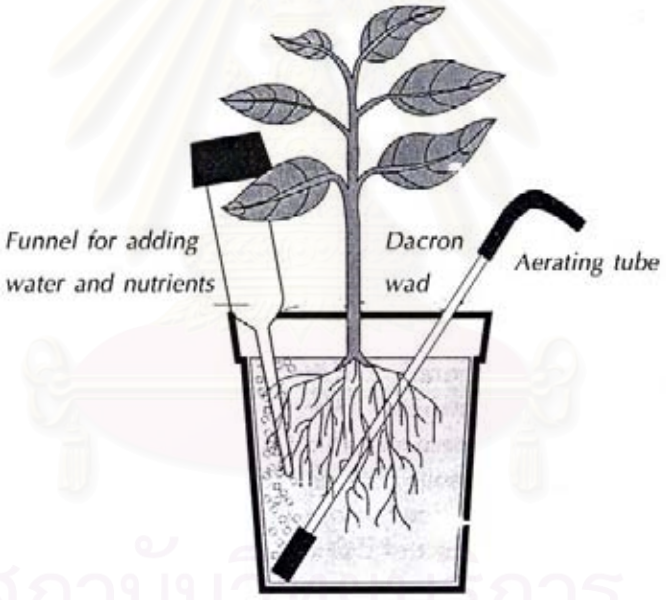


Figure 3.3: Diagram of a typical setup for nutrient solution culture [27].(From Epstein, 1972.)

The container in which the plants are grown is usually painted black or wrapped

with an opaque material in order to keep out light. The purpose of excluding light is to reduce the growth of algae that would compete with the plants for nutrients. The container should be drained completely every two weeks and the nutrient solution renewed from the mixing vats. This operation should be arranged so that it can be accomplished in a short time. If more than a few minutes elapse between the time of draining the containers and refilling them, the roots will dry out. To delay the drying of the roots, we should change the solutions on a cloudy day or after the sunset.

An adequate supply of pure water is essential for the system of hydroponics. Water should be available in adequate amounts in the soilless culture for proper growth. Too little or too much water will not give optimum growth.

The plant must absorb certain minerals through its roots to survive. The minerals required in relatively large amounts are nitrogen, potassium, phosphorus, calcium, magnesium, and sulfer. Those required in small amounts are iron, manganese, boron, zinc, and copper. Molybdenum and chlorine are also useful for plants, but the quantities required are so minute that they are usually supplied in the water or along with the other mineral nutrients as impurities. Nutrient solution may be added by hand, by means of a gravity feed system, or by mechanism. In a small setup, the nutrient solution can be mixed in small containers and added by hand. In a large setup, the gravity-feed system can be effectively used. The nutrient solution is mixed in a vat and tapped from the vat. A large earthen jar or barrel will serve as the vat.

The time for adding nutrient solution depends on the temperature and the growth of the plants. When the plants are young, the space between the seedbed

and the nutrient solution may be quite small. As the plant roots grow, lower the nutrient level slowly, keeping the level of the solution as constant as possible.

When the temperature is high and evaporation rapid, the plants may need additional solution every day. Keep the roots at the correct level in the water to prevent the root from drying and dying. The roots will die if allowed to dry out.

Equipment:

1. plastic storage container.
2. nutrient solution stock.
3. 5-day-old soybean plants.
4. aquarium pump.
5. aquarium valve and tubing.

Procedure:

1. Cover the sides of storage container with a dark material to prevent light from entering. This will prevent algae growth in the nutrient solution.

2. Fill the container with nutrient stock.

3. Carefully remove soybean plant, 5 day old, from tray and place in the container. Support with sponge, cotton, or other available material and place in container. Make plant roots bathe in nutrient solution.

4. Attach tubing coming aquarium pump and place in container.

5. Place containers in sunlight. Replace water loss and replenish nutrient solution as needed.

6. Observe plant growth daily. Record observations such as Internode length, etc.

3.3 The Lindenmayer Systems

Lindenmayer systems (L-systems) were conceived as a mathematical theory of plant development. Originally, they did not include enough detail to allow comprehensive modelling of higher plants. The emphasis was on plant topology, that is, the neighborhood relations between cells or larger plant modules. Their geometric aspects were beyond the scope of the theory. Subsequently, several geometric interpretations of L-systems were proposed with a view to turn them into a versatile tool for plant modelling.

3.3.1 The Rewriting System

The central concept of L-system is that of rewriting. In general, rewriting is a technique for defining complex objects by successively replacing parts of a simple initial object using a set of *rewriting rules* or *productions*.

L-system is a parallel rewriting system that operates on strings used to describe branching structures of modules. When L-system is used to simulate plant growth, modules usually represent organs of the plant such as leaves or internodes. The topology of structure is described in the string : each branch is delimited by a pair of matching brackets.

There are two main components in L-system : a set of rules and an axiom. The rules describe changes that can occur to the modules as well as the creation or destruction of other modules during an iteration.

Example 1: axiom : X

rule1 : F→ FF

rule2 : X→ F[+X]F[+X]-X

At each discrete derivation step, these rewriting or production rules are applied to all the modules of the string, possibly replacing them with their successor modules. According to the type of L-system used, the conditions of application of the rules differ : they can be deterministic (in the most basic forms of L-system) or stochastic, context-sensitive or not, and can depend on values of parameters.

Example 2 : The first three derivations of the L-system described in example 1 are as follows.

step 1 : X

step 2 : F[+X]F[+X]-X

step 3 : FF[+F[+X]F[+X]-X]FF[+F[+X]F[+X]-X]-F[+X]F[+X]-X

3.3.2 The Turtle interpretation of strings

The strings generated by L-systems are lists of symbols. To produce visible L-system plants, an interpretation based on the concept of Turtle Graphics is given below. A state of the turtle is defined as a triplet (x, y, δ), where the Cartesian coordinates (x, y) represent the turtle's position, and the angle δ, called the heading, is interpreted as the direction which the turtle is facing. Given the step size d and the angle increment δ, the turtle can respond to the commands represented by the following symbols:

An example of L-string turtle interpretation is shown in Figure 3.4.

Table 3.1: The two-dimensional turtle interpretation.

Symbols	Meaning
F	Move forward a step of length d and connect the new position with the last position by a line segment.The new position is computed by the formula.
f	Move forward a step of length d.
+	Turn left by angle δ.
-	Turn right by angle δ.

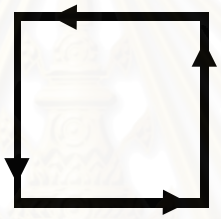


Figure 3.4: Turtle interpretation of a string F+F+F+F at $\delta = 90$ degrees.

3.3.3 The Bracketed OL-systems

The definition of tree L-systems does not specify the data structure for representing axial trees. One possibility is to use a list representation with a tree topology. Alternatively, axial tree can be represented using *strings with brackets* [28]. A similar distinction can be observed in Koch constructions, which can be implemented either by rewriting edges and polygons or their string representations. An extension of turtle interpretation to strings with brackets and the operation of bracketed L-systems [28] are described below.

Two new symbols are introduced to delimit a branch. They are interpreted by the turtle as follows:

Table 3.2: The bracketed symbols.

Symbols	Meaning
[	Push the current state of the turtle onto a pushdown stack. The information saved on the stack contains the turtle's position and orientation, and possibly other attributes such as the color and width of lines being drawn.
]	Pop a state from the stack and make it the current state of the turtle. No line is drawn, although in general the position of the turtle changes.

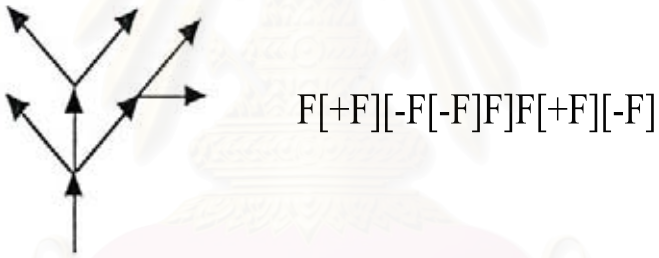


Figure 3.5: Bracketed string representation of an axial tree.

An example of an axial tree and its string representation are shown in Figure 3.5. Derivations in bracketed OL-systems proceed as in OL-systems without brackets. Examples of two-dimensional branching structures generated by bracketed OL-systems are shown in Figure 3.6.

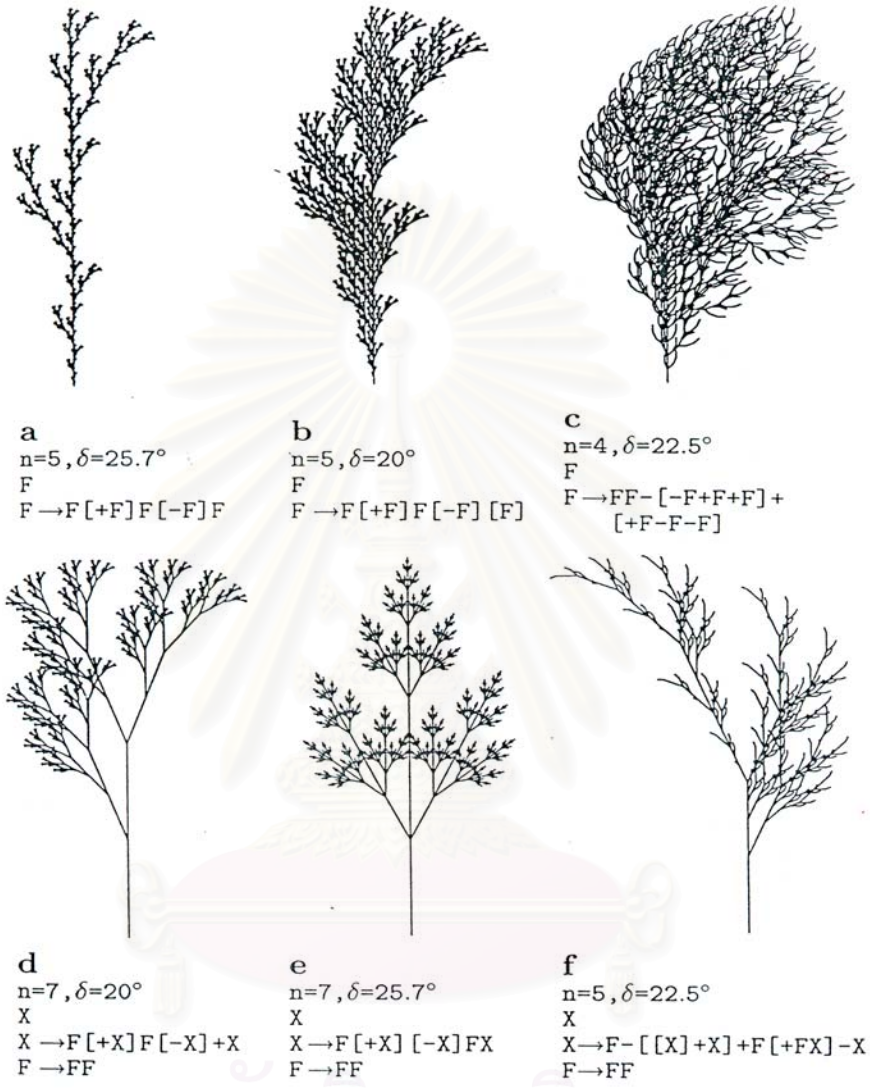


Figure 3.6: Examples of plant-like structures generated by bracketed OL-systems[28].

CHAPTER 4

EXPERIMENTAL DESIGN AND PLANT MODELLING

4.1 Experimental Design

In this section we present the design and implementation of the experimentation in this work.

4.1.1 Hoagland's Solution

There are numerous nutrient solution formulations described in scientific papers on plant nutrition, and in books and articles on hydroponics[27, 29]. Some are designed for general use (like Hoagland's solution), and others for specific plants.

Preparation of the nutrient solution.

The method described below is used to prepare modified Hoagland's solution.

Step 1

Make up 10 stock solutions of each major nutrient. This is achieved by weighing out:

(a) 147.02g $CaCl_2 \cdot 2H_2O$

(b) 236.15g $Ca(NO_3)_2 \cdot 4H_2O$

(c) 203.31g $MgCl_2 \cdot 6H_2O$

(d) 246.48g $MgSO_4 \cdot 7H_2O$

(e) 74.55g KCl

(f) 136.09g KH_2PO_4

(g) 101.10g KNO_3

(h) 137.99g $NaH_2PO_4 \cdot H_2O$

(i) 84.99g $NaNO_3$

(j) 142.04g Na_2SO_4

Place the weighed-out amounts into seperate containers and bring the volume of each up to one liter.

Step 2

Make up a Fe-EDTA(2.5mg/ml Fe) stock solution. This is achieved by

(a) dissolving 22.4g of EDTA disodium salt ($C_{10}H_{14}O_8Na_2 \cdot 2H_2O$) in 372 milliliter of distilled water and

(b) dissolving 13.5g of $FeCl_3 \cdot 6H_2O$ in 728 milliliter of distilled water, then mix all together.

Step 3

Make up a stock solution of the micro-nutrient. All trace elements are added together. Weigh out, and dissolve in 1 liter of distilled water:

(a) 2.86g H_3BO_3

(b) 0.05g $CuCl_2 \cdot 2H_2O$

(c) 1.81g $MnCl_2 \cdot 4H_2O$

(d) 0.11g $ZnCl_2$

(e) 0.025g $Na_2MoO_4 \cdot 2H_2O$

Solutions are made by filling 1500 mL container with distilled water, then add the proper amount (in mL) of each component (for each treatment read down the column). The solution are mixed and brought the volume of each up to two liters.

The complete nutrient solution contains all essential minerals for plant growth, while trace elements are provided by impurities in the chemicals used. All solutions are made up of distilled water. In our experiment, we made a nutrient solutions which are deficient nitrogen, phosphorus or potassium from complete nutrient solution by varying amount of deficient of nitrogen, phosphorus and potassium as 0, 50, 100 for each nutrient to the complete solution N-P-K. The assigned complete nutrient solution is 100-100-100, which means 100 percent of nitrogen, 100 percent of phosphorus and 100 percent of potassium, respectively. Thus 100-100-50 mixture means that complete nutrient solution has less potassium than 50 percent. All of percentage of N-P-K deficient formulas are shown in Table 4.1 to 4.4.

Table 4.1: mL of stock solution per 2 liter of water.

Stock Solution	Complete	100-100-0	100-50-0	100-0-0
1M $Ca(NO_3)_2$	10	10	10	10
1M KNO_3	10	0	0	0
1M $MgSO_4$	4	4	4	4
1M KH_2PO_4	2	0	0	0
Fe-EDTA (2.5 mg/ml)	4	4	4	4
Micronutrients	2	2	2	2
1M $NaNO_3$	0	10	10	10
1M $MgCl_2$	0	0	0	0
1M Na_2SO_4	0	0	0	0
1M NaH_2PO_4	0	2	1	0
1M $CaCl_2$	0	0	0	0
1M KCl	0	0	0	0

Stock Solution	50-100-0	50-50-0	50-0-0	0-100-100
1M $Ca(NO_3)_2$	10	10	10	0
1M KNO_3	0	0	0	0
1M $MgSO_4$	4	4	4	4
1M KH_2PO_4	0	0	0	2
Fe-EDTA (2.5 mg/ml)	4	4	4	4
Micronutrients	2	2	2	2
1M $NaNO_3$	5	5	5	0
1M $MgCl_2$	0	0	0	0
1M Na_2SO_4	0	0	0	0
1M NaH_2PO_4	2	1	0	0
1M $CaCl_2$	0	0	0	10
1M KCl	0	0	0	10

Table 4.2: mL of stock solution per 2 liter of water.

Stock Solution	0-50-100	0-0-100	0-100-0	0-50-0
1M $Ca(NO_3)_2$	0	0	0	0
1M KNO_3	0	0	0	0
1M $MgSO_4$	4	4	4	4
1M KH_2PO_4	1	0	0	0
Fe-EDTA (2.5 mg/ml)	4	4	4	4
Micronutrients	2	2	2	2
1M $NaNO_3$	0	0	0	0
1M $MgCl_2$	0	0	0	0
1M Na_2SO_4	0	0	0	0
1M NaH_2PO_4	0	0	0	1
1M $CaCl_2$	10	10	10	10
1M KCl	11	12	0	0
Stock Solution	0-0-0	100-100-50	100-50-50	100-0-50
1M $Ca(NO_3)_2$	0	10	10	10
1M KNO_3	0	5	5	10
1M $MgSO_4$	4	4	4	4
1M KH_2PO_4	0	1	1	0
Fe-EDTA (2.5 mg/ml)	4	4	4	4
Micronutrients	2	2	2	2
1M $NaNO_3$	0	5	5	0
1M $MgCl_2$	0	0	0	0
1M Na_2SO_4	0	0	0	0
1M NaH_2PO_4	0	1	0	0
1M $CaCl_2$	10	0	0	0
1M KCl	0	0	0	1

Table 4.3: mL of stock solution per 2 liter of water.

Stock Solution	100-0-100	100-50-100	50-0-50	50-0-100
1M $Ca(NO_3)_2$	10	10	5	5
1M KNO_3	10	10	5	5
1M $MgSO_4$	4	4	4	4
1M KH_2PO_4	0	1	0	0
Fe-EDTA (2.5 mg/ml)	4	4	4	4
Micronutrients	2	2	2	2
1M $NaNO_3$	0	0	0	0
1M $MgCl_2$	0	0	0	0
1M Na_2SO_4	0	0	0	0
1M NaH_2PO_4	0	0	0	0
1M $CaCl_2$	0	0	5	5
1M KCl	2	1	1	7
Stock Solution	50-50-50	50-50-100	50-100-50	50-100-100
1M $Ca(NO_3)_2$	5	5	5	5
1M KNO_3	5	5	5	5
1M $MgSO_4$	4	4	4	4
1M KH_2PO_4	1	1	1	2
Fe-EDTA (2.5 mg/ml)	4	4	4	4
Micronutrients	2	2	2	2
1M $NaNO_3$	0	0	0	0
1M $MgCl_2$	0	0	0	0
1M Na_2SO_4	0	0	0	0
1M NaH_2PO_4	0	0	1	0
1M $CaCl_2$	5	5	0	5
1M KCl	0	6	0	5

Table 4.4: mL of stock solution per 2 liter of water.

Stock Solution	0-0-50	0-50-50	0-100-50
1M $Ca(NO_3)_2$	0	0	0
1M KNO_3	0	0	0
1M $MgSO_4$	4	4	4
1M KH_2PO_4	0	1	1
Fe-EDTA (2.5 mg/ml)	4	4	4
Micronutrients	2	2	2
1M $NaNO_3$	0	0	0
1M $MgCl_2$	0	0	0
1M Na_2SO_4	0	0	0
1M NaH_2PO_4	0	0	1
1M $CaCl_2$	10	10	10
1M KCl	6	5	5

4.1.2 Data Collection

The data of each component are collected from an actual soybean. These data concerned the internode length of soybean, which was grown in 27 formula of nutrient solution corresponding to the time of its life cycle. The actual data were obtained daily for 68 days. The data of soybean were collected manually using rulers and a protractor. The physiology of soybean is shown in Figure 4.1. The internode length is designed as in Figure 4.2.

From the experimental results, the height of soybean at 68 day of each formula

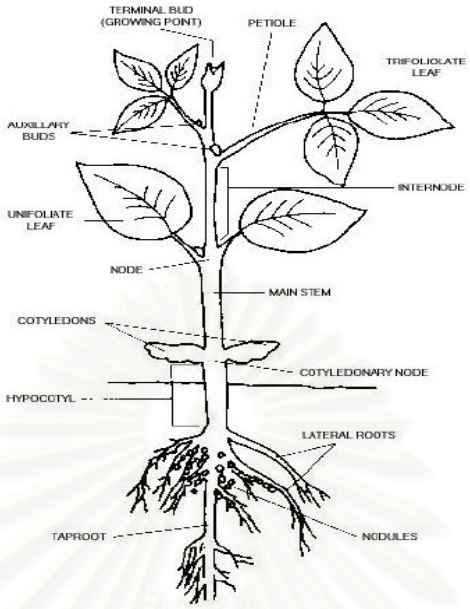


Figure 4.1: The soybean physiology.

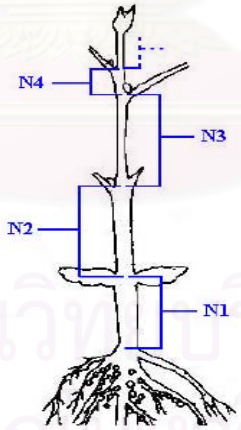


Figure 4.2: Internode data.

are shown in Table 4.5.

Table 4.5: Height of soybean at 68 day of each formula.

Formula	100-100-100	100- 50-100	100- 0-100	50-100-100	50- 50-100
Height(cm.)	88.4	73.5	35.6	64.8	51.2
Formula	50- 0-100	0-100-100	0- 50-100	0- 0-100	100-100- 50
Height(cm.)	30.9	33.2	33.2	30.3	60.1
Formula	100- 50- 50	100- 0- 50	50-100- 50	50- 50- 50	50- 0- 50
Height(cm.)	56.7	32.1	47.8	47.7	23.4
Formula	0-100- 50	0- 50- 50	0- 0- 50	100-100- 0	100- 50- 0
Height(cm.)	24.4	20.6	19.3	58.4	49.1
Formula	100- 0- 0	50-100- 0	50- 50- 0	50- 0- 0	0-100- 0
Height(cm.)	22.3	42.7	42.2	20.0	23.7
Formula	0- 50- 0	0- 0- 0			
Height(cm.)	19.4	15.0			

4.2 Plant Modelling using Neural Network

To simulate the development of soybean growth under the different amount of nitrogen, phosphorus and potassium, it is necessary to develop a mathematical model that describes the process of soybean growth. In this research, a neural networks (NN) technique was employed to model the relationship between soybean and primary nutrients.

Neural networks are similar to nonlinear regression, but they are much more

robust and can expose hidden relationships in large bodies of information by using pattern recognition theory. They have been successfully used in biological applications to predict processes such as optimum temperatures for greenhouses, insect pest treatment thresholds, recognition of patterns from digital images, and predicts pH and electrical conductivity (EC) changes in the root zone of lettuce[30, 31].

A feedforward neural network is used as the basis of modelling. The network has five inputs, namely, (1) percentage of nitrogen-deficient, (2) percentage of phosphorus-deficient, (3) percentage of potassium-deficient, (4) time step (day) and, (5) length of internodes, and one output as the length of internode in next step time. After the training process, its performance and generalization capabilities were evaluated.

In this work, backpropagation algorithm is used to train the networks. The training set consisted of data collected during 68 days of 27 formulas. This model began to predict the height of soybean on the third day by using the height of the first and second day's data as the initial values. Thus, the entire training data set consisted of 1782 columns. Each column was of the following form:

$$[\quad t \quad N \quad P \quad K \quad I(t) \quad I(t+1)]^T$$

and one output which is $I(t+2)$, where $t = 1, 2, 3, ..., 66$. N, P and K are the percentages of Nitrogen, Phosphorus, and Potassium, respectively. $I(t), I(t+1)$ and $I(t+2)$ are the height of soybean at time step $t, t+1$ and $t+2$, respectively.

The activation function at the hidden layer is a *hyperbolic tangent sigmoid* function, while the output layer is a *linear or* (*identity*) function.

The neural network model is of the following form:

$$I(t+2) = W_2 \cdot \left(\frac{2}{1 + e^{-2 \cdot W_1 \cdot X(t)}} - 1 \right) \tag{4.1}$$

where

$t = 1, 2, \ldots, 66$;

$X(t) = [t \quad N \quad P \quad K \quad I(t) \quad I(t+1)]^T$;

W_1 is a matrix given from the neural network.

W_2 is a matrix given from the neural network.

$I(t)$ is the height of soybean at t day.

$I(0)$ and $I(1)$ are the initial value of the height of soybean on the first and second day respectively.

CHAPTER 5

EXPERIMENTAL RESULTS, SIMULATION AND VISUALIZATION

5.1 Experimental Results and Model Simulation

From the experimental results, we concluded that only 100-100-100 formula was the response to complete growth of soybean. If one nutrient is deficient in 50 percent, the height will decrease, which depends on what nutrient it lacks. If more than one nutrient are deficient in 50 percent, the height will not be much different. And if the nutrients are deficient less than 50 percent in either one or all of nutrients, it will grow up incompletely.

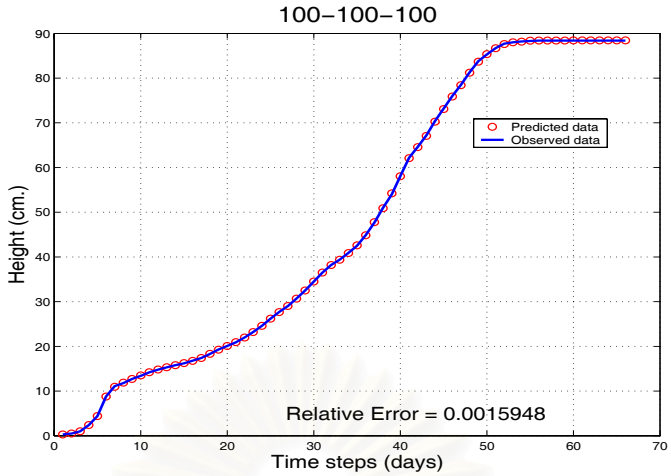
According to the model, it was found that the neural network model gave a good prediction to the actual data. Figures 5.1 to 5.9 are graphs of the height of soybean based on actual data and values predicted by the NN model. The average of relative error of this predictions for the training formulas is less than 0.005. For training set, the relative errors of this prediction in each formula obtained are given in the Table 5.1. The average of relative error of these predictions for the test formulas is less than 0.09. For test set, the relative errors of this prediction in each formula obtained are given in the Table 5.2.

Table 5.1: Relative error of the prediction in each formula (training set).

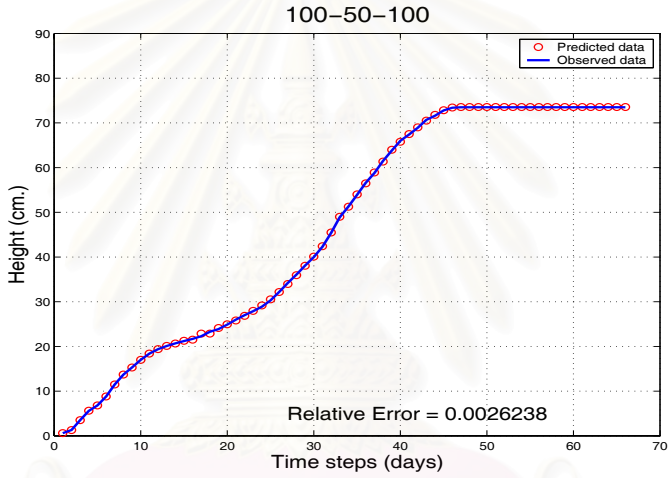
Formula	100-100-100	100- 50-100	100- 0-100	50-100-100	50- 50-100
Relative Error	0.0015948	0.0026238	0.0054022	0.0019723	0.002654
Formula	50- 0-100	0-100-100	0- 50-100	0- 0-100	100-100- 50
Relative Error	0.0019176	0.0018314	0.0012402	0.0017217	0.00098985
Formula	100- 50- 50	100- 0- 50	50-100- 50	50- 50- 50	50- 0- 50
Relative Error	0.0019452	0.0026854	0.0017324	0.0010437	0.0030032
Formula	0-100- 50	0- 50- 50	0- 0- 50	100-100- 0	100- 50- 0
Relative Error	0.0017200	0.0023325	0.0020074	0.0011560	0.0014474
Formula	100- 0- 0	50-100- 0	50- 50- 0	50- 0- 0	0-100- 0
Relative Error	0.0014840	0.0017536	0.0019023	0.0014758	0.0021396
Formula	0- 50- 0	0- 0- 0			
Relative Error	0.0039871	0.0015878			

Table 5.2: Relative error of the prediction in each formula (test set).

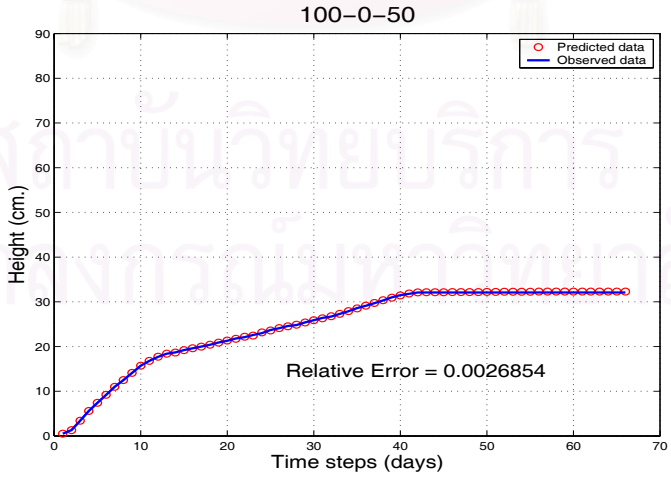
Formula	75-100-50	100-75-25	50-0-75	0-25-50
Relative Error	0.078184	0.030036	0.098627	0.063698



(a)

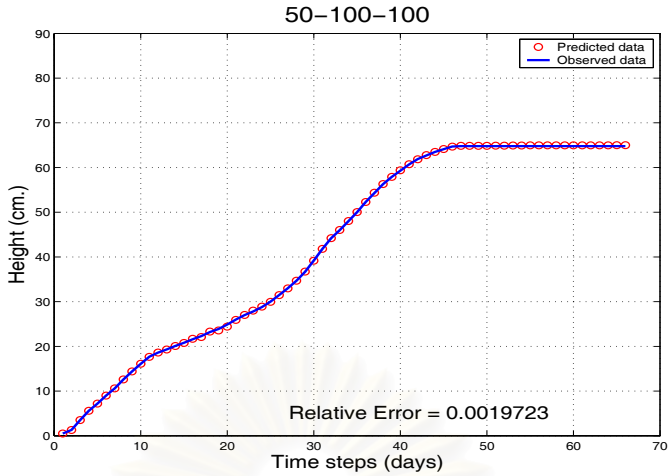


(b)

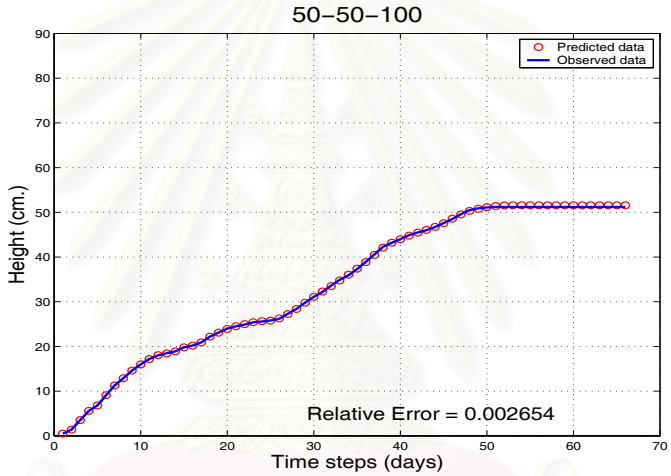


(c)

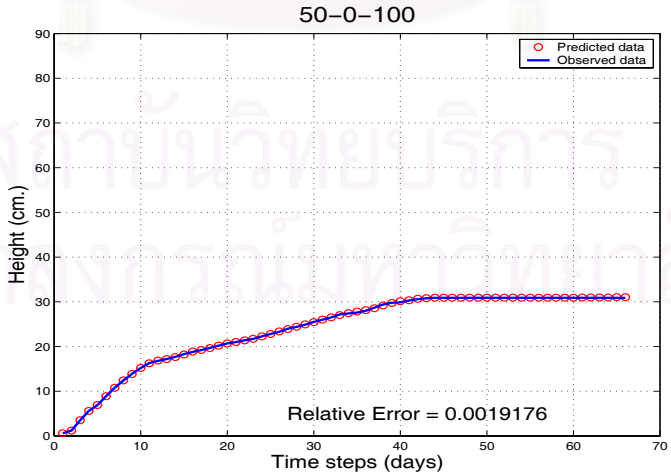
Figure 5.1: Prediction results of formula (a) 100-100-100. (b) 100-50-100. (c) 100-0-50.



(a)

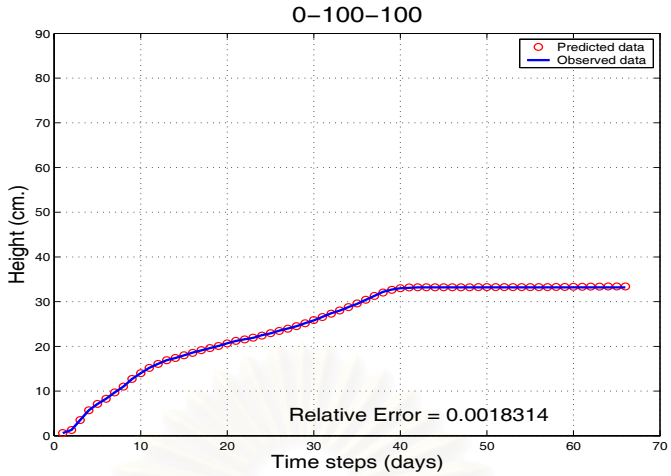


(b)

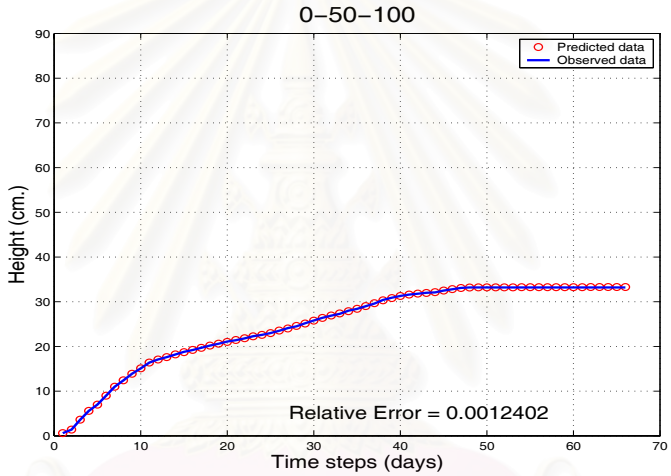


(c)

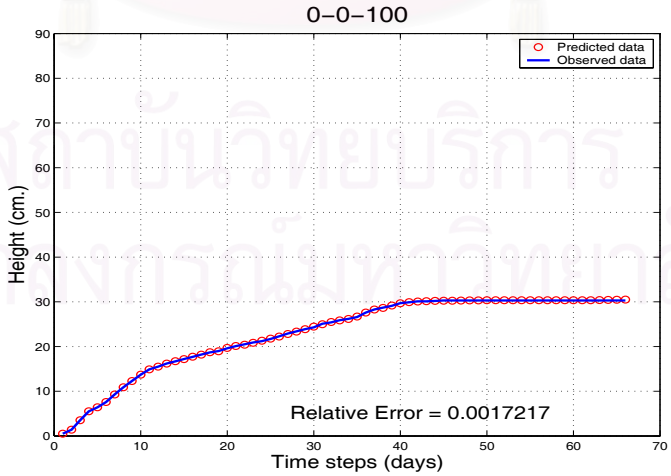
Figure 5.2: Prediction results of formula (a) 50-100-100. (b) 50-50-100. (c) 50-0-100.



(a)

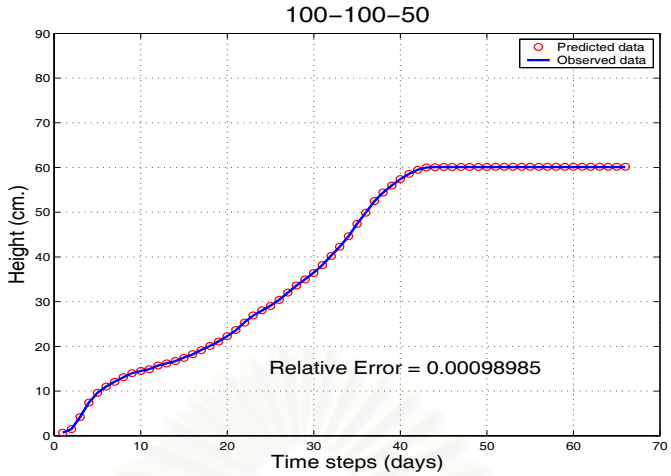


(b)

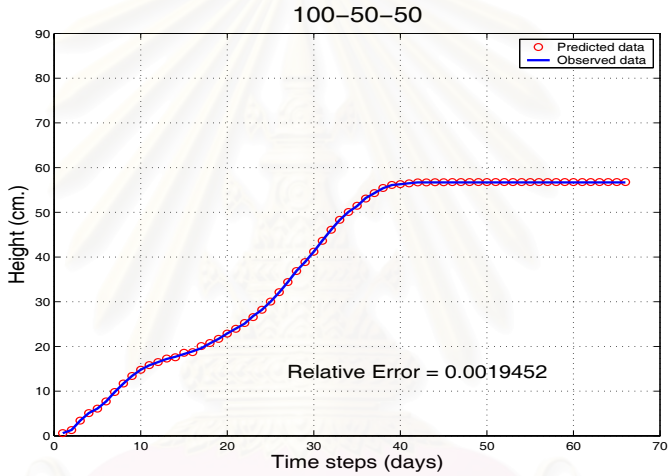


(c)

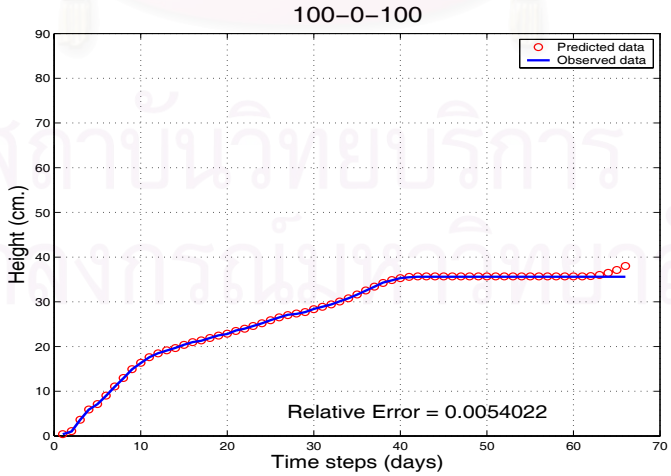
Figure 5.3: Prediction result of formula (a) 0-100-100. (b) 0-50-100. (c) 0-0-100.



(a)

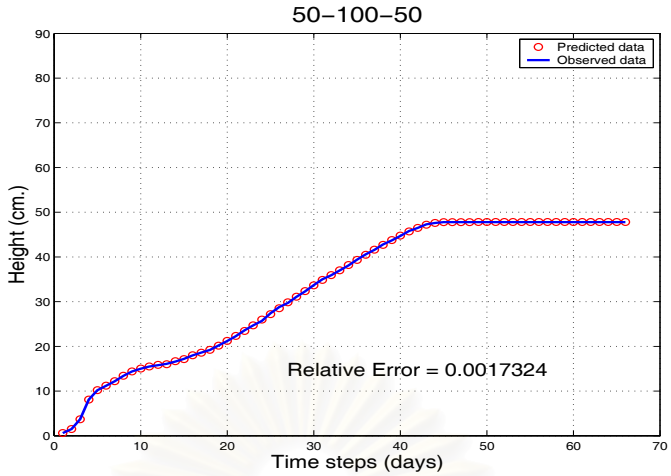


(b)

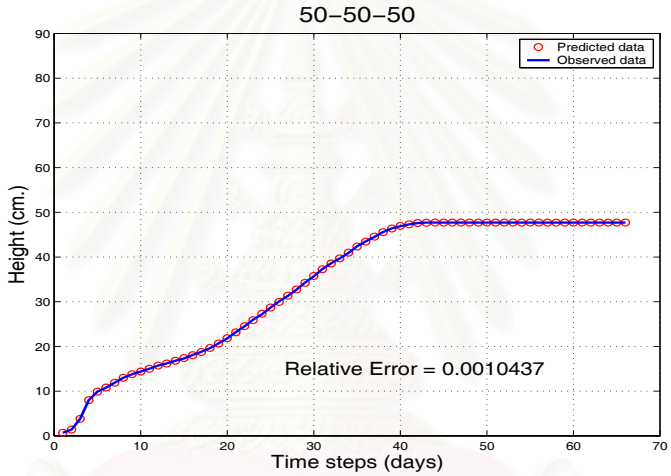


(c)

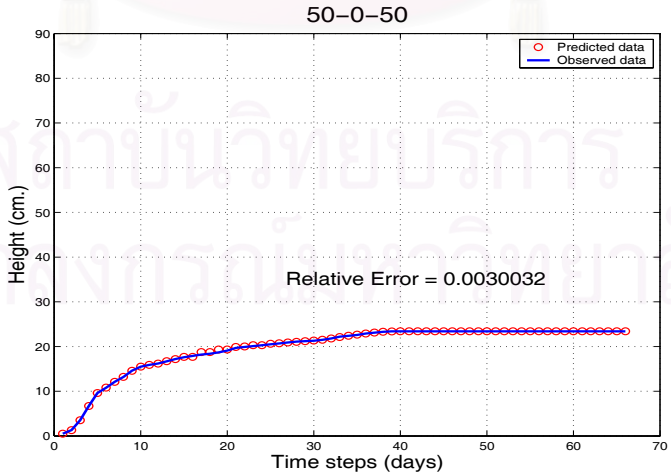
Figure 5.4: Prediction result of formula (a) 100-100-50. (b) 100-50-50. (c) 100-0-100.



(a)

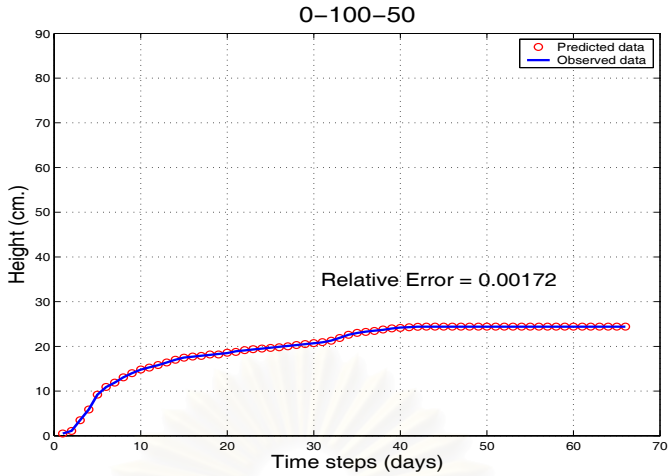


(b)

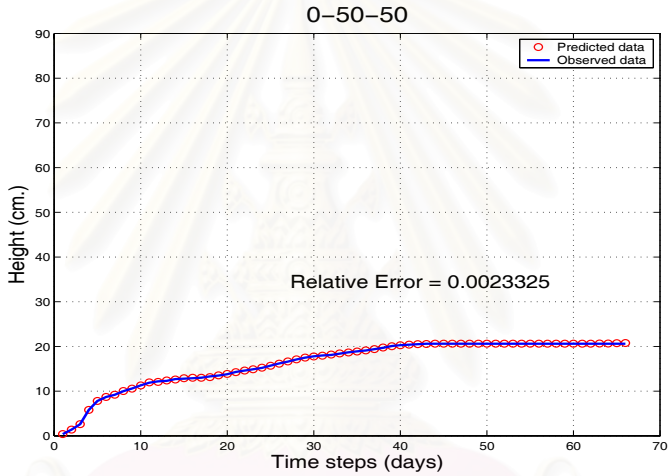


(c)

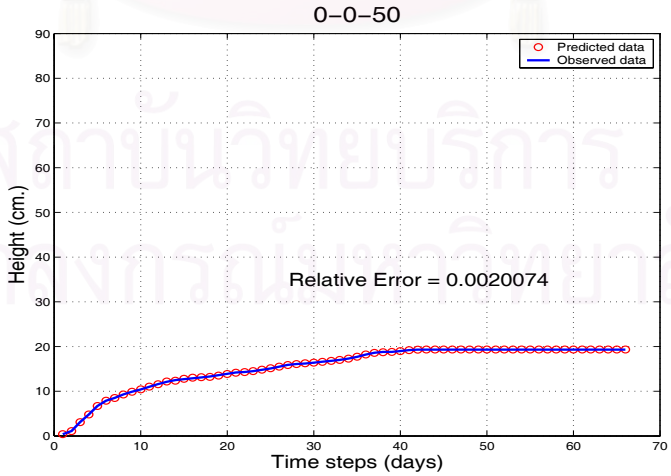
Figure 5.5: Prediction result of formula (a) 50-100-50. (b) 50-50-50. (c) 50-0-50.



(a)

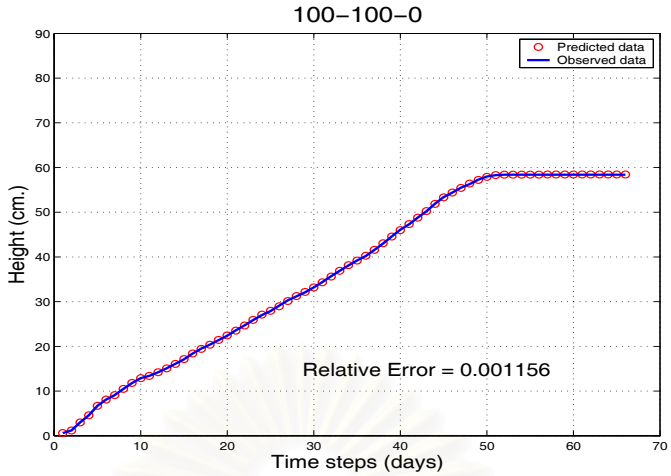


(b)

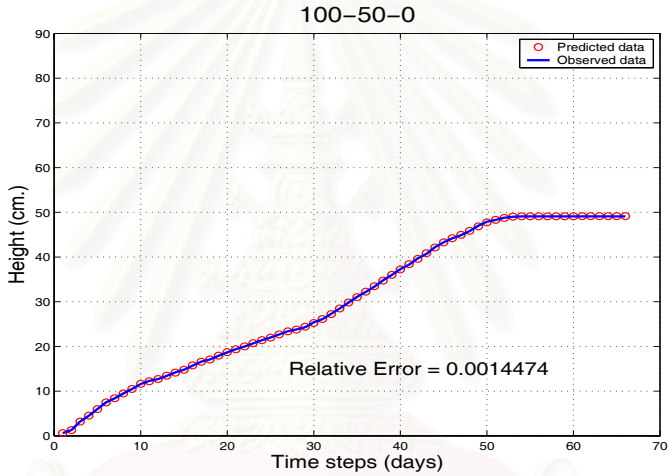


(c)

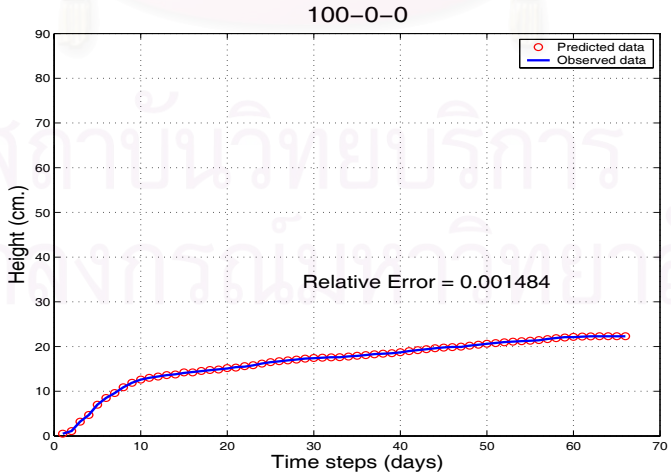
Figure 5.6: Prediction result of formula (a) 0-100-50. (b) 0-50-50. (c) 0-0-50.



(a)

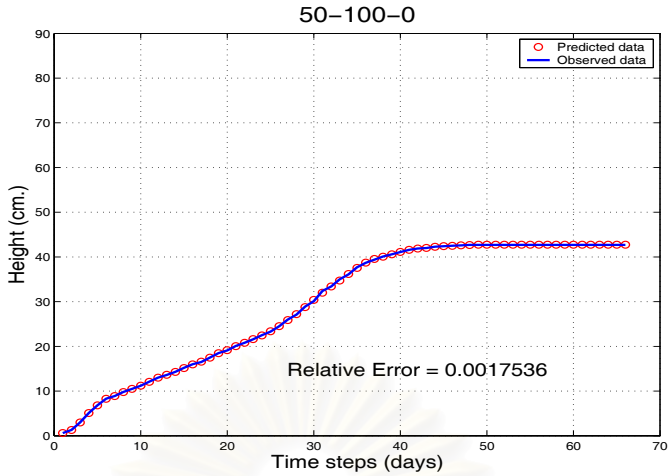


(b)

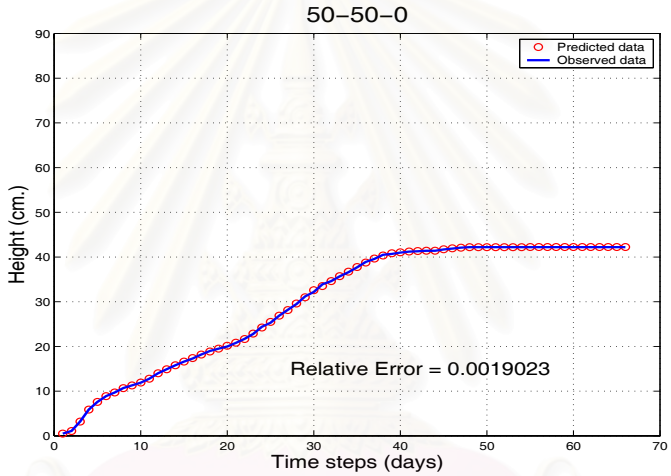


(c)

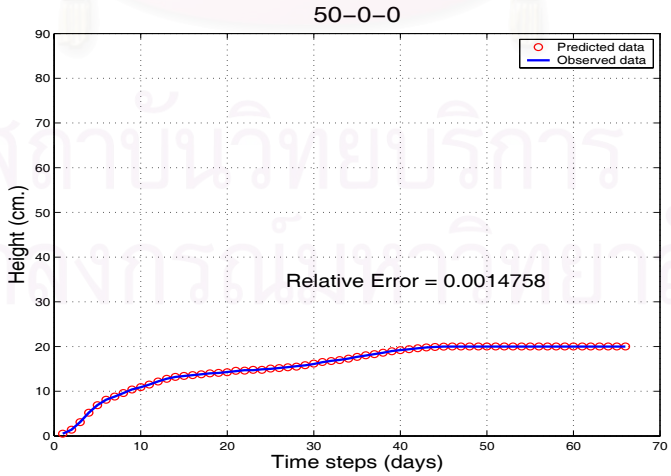
Figure 5.7: Prediction result of formula (a) 100-100-0. (b) 100-50-0. (c) 100-0-0.



(a)

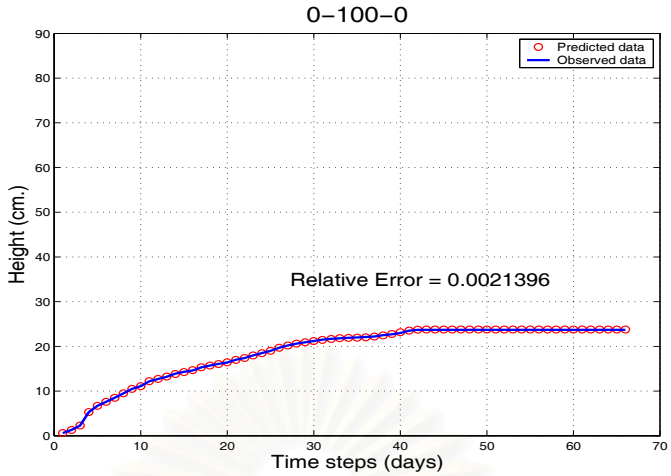


(b)

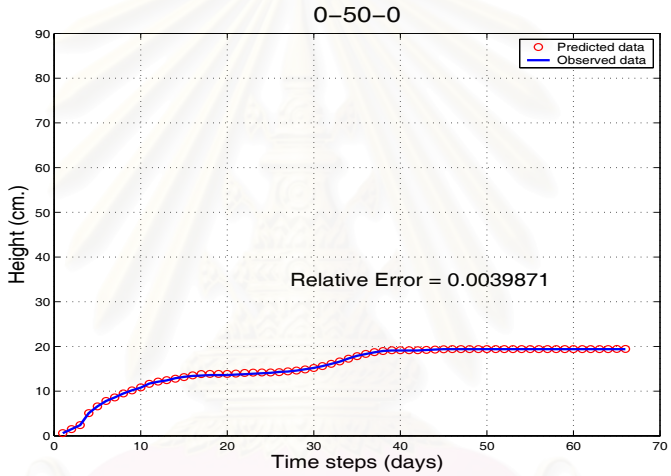


(c)

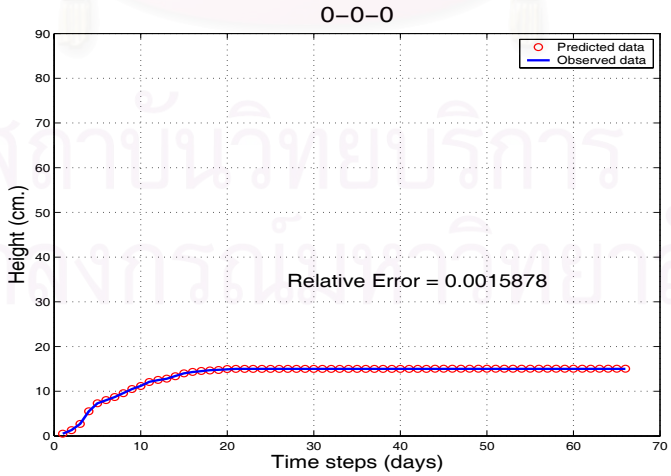
Figure 5.8: Prediction result of formula (a) 50-100-0. (b) 50-50-0. (c) 50-0-0.



(a)



(b)



(c)

Figure 5.9: Prediction result of formula (a) 0-100-0. (b) 0-50-0. (c) 0-0-0.

5.2 Visualization

In this research, the development of soybean in different amount of nitrogen, phosphorus, and potassium is visually presented. The L-system qualitative model represents the plant topology and development. We evaluated the soybean growth and made some parameters adjustment such as the size of each internode, to render a more realistic look of plant growth. An Example of L-system string of soybean is

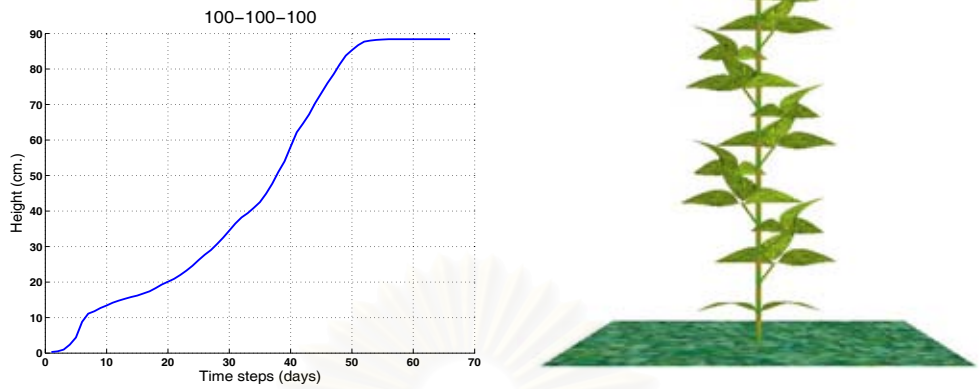
$$I[+p\&L][-pL]$$
$$I[>P[+p[\&L]][-pL][<p\wedge\wedge L]]$$
$$I[+P[<p[\&\wedge\wedge L]][>p\wedge\wedge L][-p\&L]]$$
$$I[<P[+p[\&L]][-pL][>p\&\wedge\wedge L]]$$
$$I[-P[<p[\&\wedge\wedge L]][>p\wedge\wedge L][+pL]]$$
$$I[>P[+p[\&L]][-pL][<p\wedge\wedge L]]$$
$$I[+P[<p[\&\wedge\wedge L]][>p\wedge\wedge L][-p\&L]]$$
$$I[<P[+p[\&L]][-pL][>p\&\wedge\wedge L]]$$
$$I[-P[<p[\&\wedge\wedge L]][>p\wedge\wedge L][+pL]]$$
$$I[>P[+p[\&L]][-pL][<p\wedge\wedge L]]$$
$$I[+P[<p[\&\wedge\wedge L]][>p\wedge\wedge L][-p\&L]]$$
$$I[<P[+p[\&L]][-pL][>p\&\wedge\wedge L]]$$
$$I[-P[<p[\&\wedge\wedge L]][>p\wedge\wedge L][+pL]].$$

The symbols are used to denote the soybean components. All of these symbols are described in Table 5.3. This string has 13 internodes, 12 petioles, and 38 leaves.

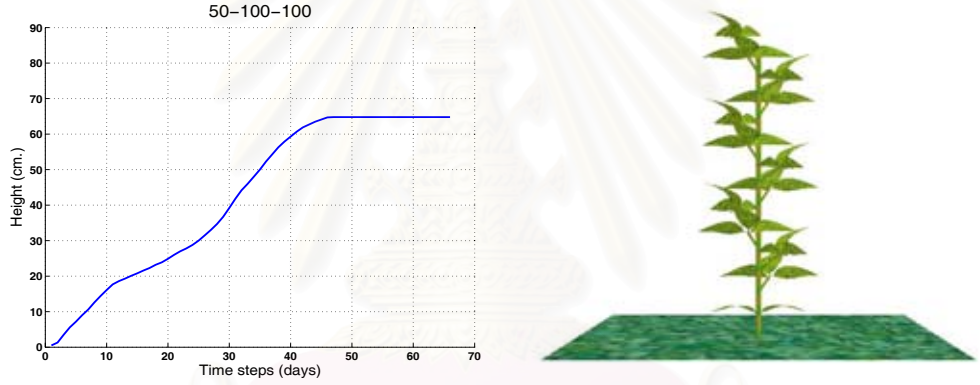
Table 5.3: Symbols used in soybean growth L-system.

Symbols	Meaning
I	To generate the soybean internodes
P	To generate the soybean petioles
p	To generate the soybean sub-petioles
L	To generate the soybean leaves
+	Roll clockwise to positive Y-axis by angle 45^o
-	Roll counter-clockwise to positive Y-axis by angle 45^o
&	Roll clockwise to positive Z-axis by angle 180^o
$\wedge$	Roll counter-clockwise to positive Z-axis by angle 45^o
$<$	Roll clockwise to positive X-axis by angle 45^o
$>$	Roll counter-clockwise to positive X-axis by angle 45^o
[	Push the current state of the turtle onto a pushdown stack to create a new branch
]	Pop a state from the stack and make it the current state of the turtle to close the branch

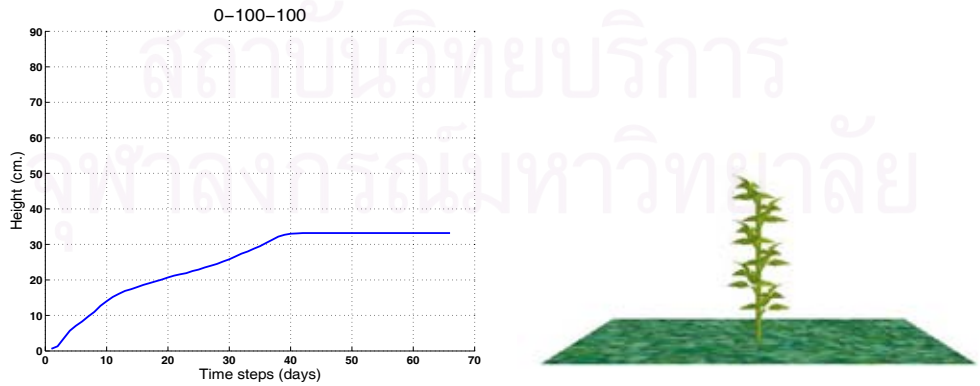
In visualization of soybean, cylinders are used to represent internodes and petioles segments. Meshes are used to represent leaves. Figures 5.10 to 5.12 show some selected stages of the development of a soybean shoot controlled by the neural network model for each formula of nutrients.



(a) (b)



(c) (d)



(e) (f)

Figure 5.10: The development of a soybean growth controlled by neural network model at ((a) and (b))100-100-100. ((c) and (d))50-100-100. ((e) and (f))0-100-100.

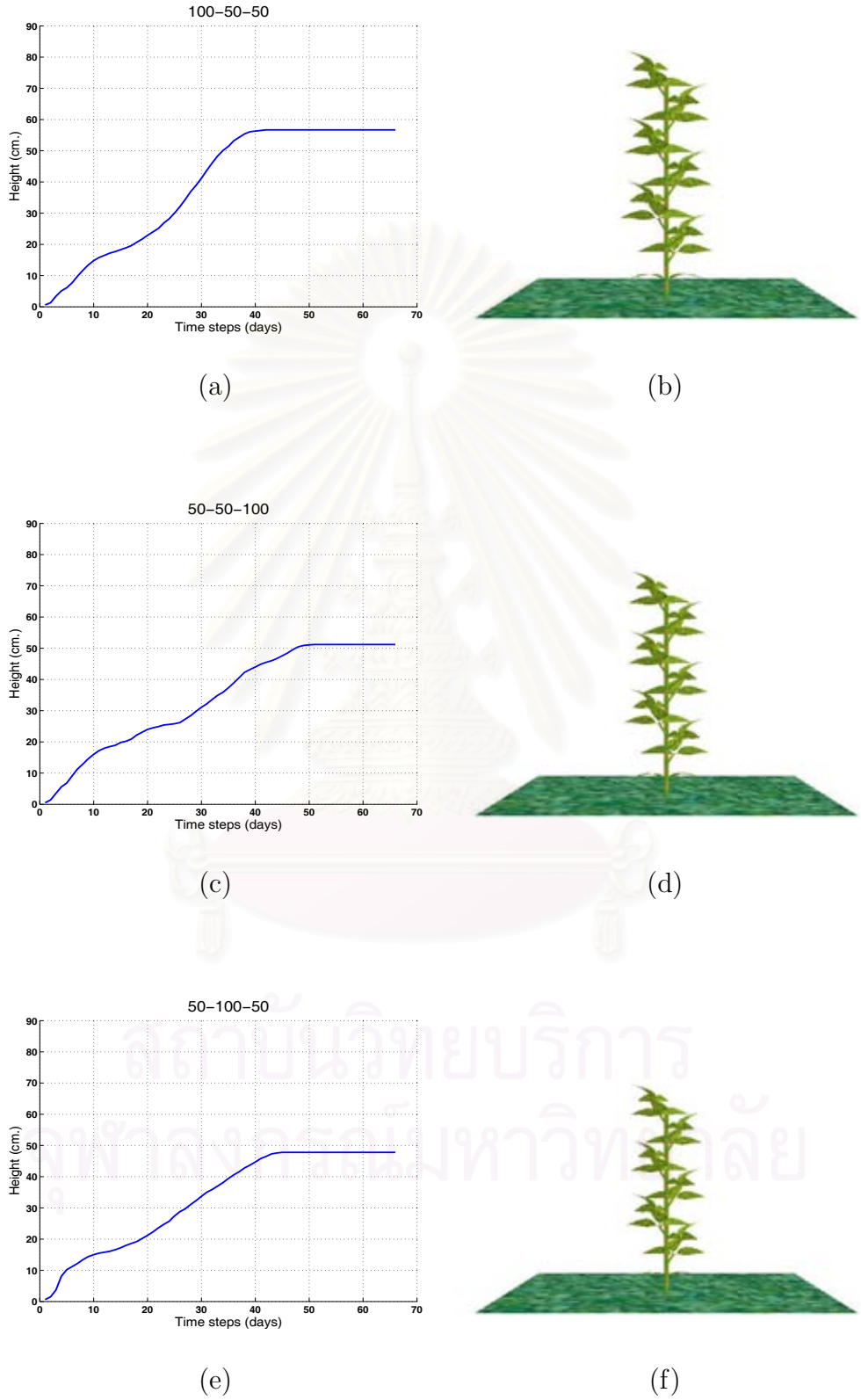


Figure 5.11: The development of a soybean growth controlled by neural network model at ((a) and (b))100-50-50. ((c) and (d))50-50-100. ((e) and (f))50-100-50.

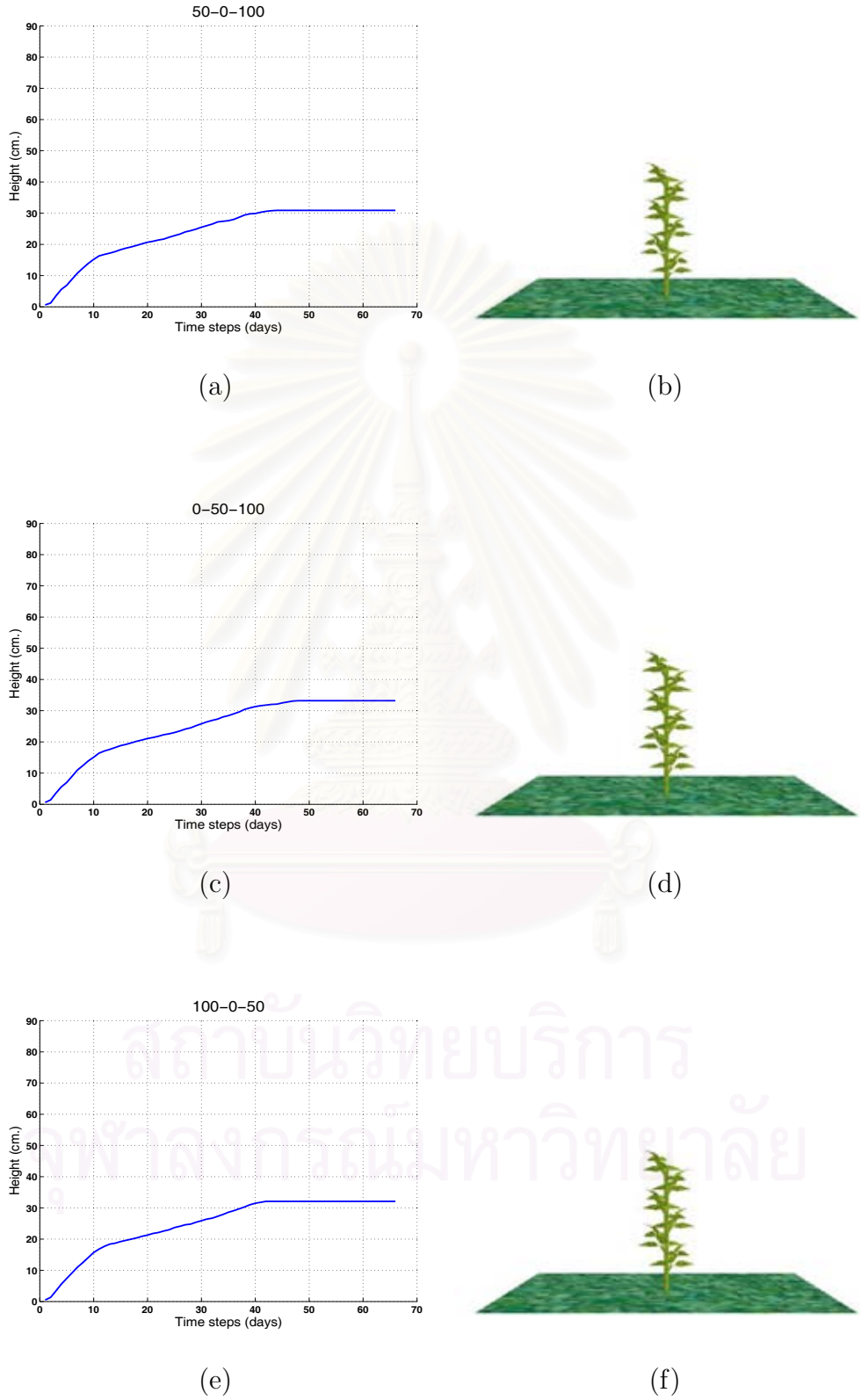


Figure 5.12: The development of a soybean growth controlled by neural network model at ((a) and (b))50-0-100. ((c) and (d))0-50-100. ((e) and (f))100-0-50.

CHAPTER 6

CONCLUSION

A predictive method that uses a supervised feedforward neural network to model the height of soybean growth in solution culture was developed. More specifically, the artificial neural network was applied successfully in a model that predicts the height of the soybean. In addition, this work builds a link between artificial intelligence, hydroponics systems and computer graphics. With its encouraging results, it also opens the way to further development and investigation of "intelligent" systems in the field of agriculture and computer graphics, which will lead to more precise and productive cultivation in agriculture systems.

The main procedures of this research can be concluded as follows:

1. Preparing the solution in each formula by using Hoagland's formula for growing soybean.

2. Measuring the height of soybean growth in each formula collected from actual soybean.

3. Modelling the soybean growth data by using a neural network.

4. Visualizing the result of this model by using L-system.

There are some problems concerning the data collection in this research. The

data collected may have produced some errors due to experimental measurement and some uncontrollable environmental factors such as amount of light, carbon dioxide, temperature, etc. Future work is required to improve the soybean growth model by considering the other factors, such as amount of light, carbon dioxide, and temperature, etc.

REFERENCES

[1] Sikora S., Steinberg D. and Lattaud C. Integration of simulation tools in on-line virtual worlds. Proceedings of the 2nd International Conference on Virtual Worlds (July 2000): 32-43.

[2] Steinberg D., Sikora S., Lattaud C., Fournier C. and Andrieu B. Plant growth simulation in virtual worlds : towards online artificial ecosystems. Proceedings of the 1st Workshop on Artificial Life Integration in Virtual Environments (September 1999): 19-25.

[3] N.Chiba, K. Ohshida, K.Muroaka, and S.Nobuji. A Growth Model Having the Abilities of Growth-Regulations for Simulating Visual Nature of Botanical Trees. Computer and Graphics 18 (1997): 469-479.

[4] Prusinkiewicz, P., Hammel, M.S., and Mjolsness, E. Animaiton of plant development. Proceedings of Computer graphics SIGGRAPH'93 (August 1993): 351-360.

[5] Chuai-Aree, S. An Algorithm for Simulation and Visualization of Plant Shoots Growth. Master's thesis, Department of Mathematics, Faculty of Science, Chulalongkorn University, 2000.

[6] Chuai-Aree, S., Siripant, S. and Lursinsap, C. Animation Plant Growth in L-System by Parametrics functional symbols. Proceeding of International Conference on Intelligent Technology 2000 (December 2000): 135-143.

[7] Lin, T. Animation of L-system based 3-D plant Growing in Java. Available from: http://www.cs.umbc.edu/~ebert/693/TLin/

[8] Hammel, M. S., Prusinkiewicz, P., and Wyvil, B. Modelling Compound Leaves Using Implicit Countours. Proceedings of CG International'92 (June 1992): 119-212.

[9] Lintermann, B., and Deussen, O. A Modelling Method and User Interface for Creating Plants. The Eurographics Association 17, 1 (March 1998): 73-82.

[10] Bedřich Benеš. An Efficient Estimation of Light is Simulation of Plant Development. Dissertation's thesis, Faculty of Electrical Engineering, Czech Technical University, Prague, 1998.

[11] Smith, A.R. Plants, fractals and formal languages. ACM SIGGRAPH 18, 3 (1984): 1-10.

[12] Prusinkiewicz, P., James, M., and Mech, R. Synthetic topiary. Proceedings of SIGGRAPH'94 (July 1994): 351-358.

[13] Hammel, M.S., and Prusinkiewicz, P. Visualization of developmental processes by extrusion in space-time. Proceedings of Graphics Interface '96 (May 1996): 246-258.

[14] Prusinkiewicz, P., and Kari, L. Subapical bracketed L-systems. Lecture Notes in Computer Science 1073 (1996): 550-564.

[15] Prusinkiewicz, P., Remphrey, W., Davidson, C., and Hammel, M. Modeling the architecture of expanding *Fraxinus pennsylvanica* shoots using L-systems. Canadian Journal of Botany 72 (1994): 701-714.

[16] Lintermann, B., and Deussen, O. Interactive Modeling of Plants. IEEE Computer Graphics and Applications 19, 1 (January 1999): 56-65.

[17] Deussen, O., Hanrahan, P., Linterman, B., Mech, R., Pharr, M., and Prusinkiewicz, P. Realistic modeling and rendering of plant ecosystems. SIGGRAPH'98 (July 1998): 275-286.

[18] P. De Reffye, C. Edelin, J. Francon, M. Jaeger and C. Puech. Plant models faithful to botanical structure and development. Computer Graphics 84(3) (1998): 151-158.

[19] P. De Reffye, F. Houllier, F. Blaise, D. Barthelemy, J. Dauzat and D. Auclair. A model simulating above- and below-ground tree architecture with agroforestry applications. Agroforestry Systems 30 (1995): 175-197.

[20] P. De Reffye, T. Fourcaud, F. Blaise, D. Barthelemy and F. Houllier. A functional model of tree growth and tree architecture. Silva Fennica 31 (1997): 297-311.

[21] Zhao X., De Reffye P., Barthelemy D. and Hu B.-G. Interactive Simulation of Plant Architecture Based on a Dual-Scale Automaton Model. International Symposium on Plant Growth modeling, simulation, visualization and their Applications(PMA03) (2003): 144-153.

[22] R.Mech and P.Prusinkiewicz. Visual Models of Plants Interacting With Their Environment. Proceedings in Computer Graphics(SIGGRAPH'96) 30(4) (1996): 397-410.

[23] Muchow, R.C., Sinclair, T.R. Effect of nitrogen supply on maize yield:II. Field and modelling analysis. Agron. J. 87 (1995): 642-648.

[24] Jamieson, P.D., Semenov. M.A. Modelling nitrogen uptake and redistribution in wheat. Field Crops Research 68 (2000): 21-29.

[25] J.B.Reid. Yield response to nutrient supply across a wide range of conditions: 1. Model derivation. Field Crops Research 77, 2-3 (September 2002): 161-171.

[26] J. B. Reid, P. J. Stone, A. J. Pearson and D. R. Wilson. Yield response to nutrient supply across a wide range of conditions: 2. Analysis of maize yields. Field Crops Research 77, 2-3 (September 2002): 173-189.

[27] William G. Hopkins. Introduction to plant physiology New York: Wiley, 1999.

[28] Prusinkiewicz, P., and Lindenmayer, A. The Algorithmic Beauty of Plants New York: Springer-Verlag, 1990.

[29] N. Angkinand and S. Chadchawan. Plant physiology laboratory Department of Botany, Chulalongkorn University, Bangkok, 2000.

[30] Robert M. Peart and R. Bruce Curry. Agricultural Systems Modeling and Simulation New York: Marcel Dekker, 1998.

[31] K. P. Ferentinos and L.D. Albright. Predictive neural network modeling of pH and Electrical Conductivity in deep-trough hydroponics. American Society of Agricultural Engineers 45(6) (August 2002): 2007-2015.

VITAE

Apichat Suratanee was born in October 25, 1979, in Bangkok. He received a bachelor degree in Mathematics from the Department of Mathematics, Faculty of Science, Srinakharinwirot Prasarnmitr University in 2000.

PUBLICATION

A.Suratanee, S.Siripant and C.Lursinsap, "Modeling the Soybean Growth in Different Amount of Nitrogen, Phosphorus and Potassium Using Neural Network", *International Workshop on Functional-Structural Plant Models (FSPM04)*, June 7-11, 2004.