

# APPROXIMATING SMALL OPEN ECONOMY MODELS WITH NEURAL NETWORK TRAINED BY GENETIC ALGORITHM

A Master's Thesis

by  
YEŞİM COŞKUN

Department of  
Economics  
Bilkent University  
Ankara  
September 2008

**APPROXIMATING SMALL OPEN  
ECONOMY MODELS WITH NEURAL  
NETWORK TRAINED BY GENETIC  
ALGORITHM**

**The Institute of Economics and Social Sciences  
of  
Bilkent University**

**by**

**YEŞİM COŞKUN**

**In Partial Fulfillment of the Requirements For the Degree of  
MASTER OF ARTS**

**in**

**THE DEPARTMENT OF  
ECONOMICS  
BİLKENT UNIVERSITY  
ANKARA**

**September 2008**

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Economics.

---

Assist. Prof. Nedim Alemdar  
Supervisor

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Economics.

---

Assist. Prof. H. Çağrı Sağlam  
Examining Committee Member

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Economics.

---

Prof. Dr. Uğurhan Muğan  
Examining Committee Member

Approval of the Institute of Economics and Social Sciences

---

Prof. Dr. Erdal Erel  
Director

# ABSTRACT

## APPROXIMATING SMALL OPEN ECONOMY MODELS WITH NEURAL NETWORK TRAINED BY GENETIC ALGORITHM

COŞKUN, Yeşim

M.A., Department of Economics

Supervisor: Asst. Prof. Nedim Alemdar

September 2008

This thesis work presents a direct numerical solution methodology to approximate the small open economy models with debt elastic interest rate premium and with convex portfolio adjustment cost, both studied by Stephanie Schmitt-Grohé and Martin Uribe(2003). This recent method is compared with the first-order approximation to the policy function from the aspect of second moments of endogenous variables and their impulse responses. The proposed methodology, namely genetic algorithm-neural network (GA-NN), parameterizes the policy function with a feed-forward neural network that is trained by a genetic algorithm. Thus, unlike the first-order approximation, GA-NN does not require the continuity and the existence of derivatives of objective and policy functions. Importantly, since genetic algorithm is an evolutionary algorithm that enables global search over the feasible set, it provides a robust result in any solution space. Also GA-NN method gives not only the moments of the model but also the optimal path.

*Keywords:* Genetic Algorithm, Neural Network, Debt-elastic Interest Rate, Portfolio Adjustment Cost.

## ÖZET

# KÜÇÜK AÇIK EKONOMİ MODELLERİNİN GENETİK ALGORİTMA İLE EĞİTİLMİŞ SİNİRSEL AĞ İLE YAKINSANMASI

COŞKUN, Yeşim

Yüksek Lisans, Ekonomi Bölümü

Tez Yöneticisi: Asst. Prof. Nedim Alemdar

Eylül 2008

Bu tez çalışması, Stephanie Schmitt-Grohé ve Martin Uribe(2003)'ın çalışmış olduğu borç esnekliği olan faiz primini ve dışbükey portföy maliyetlerini içeren küçük açık ekonomik modellerini yakınsamak için doğrudan numerik bir çözüm metodu sunmaktadır. Yeni geliştirilmiş bu metod politika fonksiyonunu birinci dereceden yakınsayan metod ile içsel değişkenlerin ikinci momentleri ve etki tepkileri göz önüne alınarak karşılaştırılmıştır. Burada sunulan, genetik algoritma-sinirsel ağ olarak adlandırılan metod, politika fonksiyonunu genetik algoritma ile eğitilmiş yapay sinirsel ağlar ile parametrize etmektedir. Böylece bu numerik metod, birinci dereceden yakınsama tekniğinden farklı olarak amaç ve politika fonksiyonlarının türevlerinin sürekliliğini ve varlığını gerektirmez. En önemlisi de, Genetik algoritmalar, olası küme üzerinde global arama yapma imkanı veren evrimsel algoritmalar olduğundan, herhangi bir çözüm uzayında sağlıklı sonuçlar sunabilmektedir. Ayrıca, metodumuz, sadece modelin momentlerini değil, en optimal yolu da sunmaktadır.

*Anahtar Kelimeler:* Genetik Algoritma, Sinirsel Ağlar, Borç esnekliği olan faiz primi, Dışbükey portföy maliyetleri.

## ACKNOWLEDGMENTS

I would like to thank Asst. Prof. Nedim Alemdar for his valuable guidance, supervision and kindness throughout this thesis study.

I would like to thank Asst. Prof. Çağrı Sağlam for his effort in graduate courses and providing me considerable knowledge in Macroeconomic Theory.

Thanks to TÜBİTAK, for their financial support during my graduate study.

Special thanks to my friends for their continuous support towards the realization of of this thesis work.

Finally, I would like to express my special thanks to my family and my beloved for their great support during this thesis work.

# TABLE OF CONTENTS

|                                                                                               |             |
|-----------------------------------------------------------------------------------------------|-------------|
| <b>ABSTRACT</b> . . . . .                                                                     | <b>iii</b>  |
| <b>ÖZET</b> . . . . .                                                                         | <b>iv</b>   |
| <b>ACKNOWLEDGMENTS</b> . . . . .                                                              | <b>v</b>    |
| <b>TABLE OF CONTENTS</b> . . . . .                                                            | <b>vi</b>   |
| <b>LIST OF TABLES</b> . . . . .                                                               | <b>viii</b> |
| <b>LIST OF FIGURES</b> . . . . .                                                              | <b>ix</b>   |
| <b>CHAPTER I: INTRODUCTION</b> . . . . .                                                      | <b>1</b>    |
| <b>CHAPTER II: MODELS</b> . . . . .                                                           | <b>5</b>    |
| 2.1 A Small Open Economy Model with Debt-Elastic Interest Rate<br>Premium . . . . .           | 5           |
| 2.2 A Small Open Economy Model with Portfolio Adjustment Costs                                | 7           |
| <b>CHAPTER III: ARTIFICIAL NEURAL NETWORK</b>                                                 |             |
| <b>PARAMETERIZATION AND GENETIC ALGORITHM</b>                                                 | <b>9</b>    |
| 3.1 Neural Networks and Neural Network<br>Parameterizations of the Policy Functions . . . . . | 9           |
| 3.2 Genetic Algorithm . . . . .                                                               | 12          |
| 3.2.1 Process of a Standard Genetic Algorithm . . . . .                                       | 13          |
| <b>CHAPTER IV: RESULTS</b> . . . . .                                                          | <b>18</b>   |
| 4.1 Simulation Results of Models . . . . .                                                    | 19          |

|                              |           |
|------------------------------|-----------|
| <b>CHAPTER V: CONCLUSION</b> | <b>26</b> |
| <b>BIBLIOGRAPHY</b>          | <b>28</b> |
| <b>APPENDICES</b>            | <b>28</b> |
| A. SOURCE CODES              | 28        |
| B. APPROXIMATED PATHS        | 28        |

## LIST OF TABLES

|    |                                                                |    |
|----|----------------------------------------------------------------|----|
| 1. | Parameters . . . . .                                           | 6  |
| 2. | Parameters calibrated by Grohe and Uribe(2003) for Model 1 . . | 7  |
| 3. | Parameters calibrated by Grohe and Uribe(2003) for Model 2 . . | 8  |
| 4. | Statistical moments . . . . .                                  | 20 |

# LIST OF FIGURES

|     |                                                                                                                        |    |
|-----|------------------------------------------------------------------------------------------------------------------------|----|
| 1.  | Neural Network Structures . . . . .                                                                                    | 10 |
| 2.  | Terminology of Natural Organisms versus GA . . . . .                                                                   | 13 |
| 3.  | Flow Chart of a Standard GA . . . . .                                                                                  | 14 |
| 4.  | Generation of new population . . . . .                                                                                 | 16 |
| 5.  | Pro-cyclic behaviors, for Model 1 . . . . .                                                                            | 21 |
| 6.  | Pro-cyclic behaviors, for Model 2 . . . . .                                                                            | 22 |
| 7.  | Impulse-responses to unit technology shock for Model 1 ob-<br>tained by GA-NN . . . . .                                | 23 |
| 8.  | Impulse-responses to unit technology shock for Model 1 ob-<br>tained by LA, taken from Grohe and Uribe(2003) . . . . . | 24 |
| 9.  | Impulse responses to unit technology shock for Model 2 obtained<br>by GA-NN . . . . .                                  | 25 |
| 10. | Impulse responses to unit technology shock for Model 2 obtained<br>by LA, taken from Grohe and Uribe(2003) . . . . .   | 25 |
| 11. | Technology shocks, in logarithms for open economy with debt<br>elastic interest rate premium . . . . .                 | 38 |
| 12. | Technology shocks, in logarithms for open economy with port-<br>folio adjustment cost . . . . .                        | 39 |
| 13. | Investment path, in logarithms for open economy with debt elas-<br>tic interest rate premium . . . . .                 | 40 |
| 14. | Investment path, in logarithms for open economy with portfolio<br>adjustment cost . . . . .                            | 41 |

|     |                                                                                                                 |    |
|-----|-----------------------------------------------------------------------------------------------------------------|----|
| 15. | Output path, in logarithms for open economy with debt elastic interest rate premium . . . . .                   | 41 |
| 16. | Output path, in logarithms for open economy with portfolio adjustment cost . . . . .                            | 42 |
| 17. | Consumption path, in logarithms for open economy with debt elastic interest rate premium . . . . .              | 42 |
| 18. | Consumption path, in logarithms for open economy with portfolio adjustment cost . . . . .                       | 43 |
| 19. | Hours path, in logarithms for open economy with debt elastic interest rate premium . . . . .                    | 43 |
| 20. | Hours path, in logarithms for open economy with portfolio adjustment cost . . . . .                             | 44 |
| 21. | Capital stock path, in logarithms for open economy with debt elastic interest rate premium . . . . .            | 44 |
| 22. | Capital stock path, in logarithms for open economy with portfolio adjustment cost . . . . .                     | 45 |
| 23. | Trade balance/GDP path, in logarithms for open economy with debt elastic interest rate premium . . . . .        | 45 |
| 24. | Trade balance/GDP path, in logarithms for open economy with portfolio adjustment cost . . . . .                 | 46 |
| 25. | Foreign interest payment path, in logarithms for open economy with debt elastic interest rate premium . . . . . | 46 |
| 26. | Foreign interest payment path, in logarithms for open economy with portfolio adjustment cost . . . . .          | 47 |

# CHAPTER 1

## INTRODUCTION

For many decades, to better understand how the real life works in macroeconomics, nonlinear stochastic dynamic models have been adapted. However, it is not possible to obtain an analytical solution for most of the nonlinear, stochastic and dynamic models. Thus, to solve nonlinear and stochastic models, new methodologies based on numerical techniques and approximations have been developed.

Linear approximation technique was first proposed by Kydland and Prescott (1982) and King et al. (1988) and has become one of the most preferred tool in solving these kind of models. Because this linear approximation method made it possible to analyze the properties of the endogenous variables of the model. Stephanie Schmitt-Grohé and Martin Uribe (2004) stated that if the support of the shocks driving aggregate fluctuations is small and an interior solution exists, linear approximations provide adequate answers to questions such as the size of the second moments of endogenous variables, local existence and determinacy. On the contrary, Kim and Kim (2004) showed that this technique fails to provide correct welfare comparisons between different stochastic and policy environments. Also, Alemdar, Sirakaya and Turnovsky (2006) stated that since this technique requires a Taylor series expansion around the steady state, it is possible to fail away from the equilibrium.

It has been well established to solve standard RBC models applying linear approximations to the policy functions. Stephanie Schmitt-Grohé and Martin Uribe (2003) applied log-linearization to the expanded versions of standard RBC model of Canada and succeeded to mimic the Canadian data over the period 1946-1985, that are expressed in per capita terms and transformed into logarithms.

In the first model, Stephanie Schmitt-Grohé and Martin Uribe (2003) expanded the borders of standard RBC model by assuming that interest rate is sensitive to the level of debt that households are possessing at any time  $t$ . That is the interest rate is increasing in the level of foreign debt. After solving the model with linear approximation by using a MATLAB code, they concluded that the model captures a number of features of business cycles of Canada. First, investment has the highest volatility while consumption has the lowest. Second, hours, investment and consumption are pro-cyclical while trade balance is counter-cyclical. However, the study does not mention the correlation between saving and investment that is being one of the most important stylized facts of modern open economies. Besides, it fails to catch the serial correlations of endogenous variables, except output and consumption.

In the second model, infinitely lived households are assumed to face convex costs of holding amount of debts different than a long-run level. The results of this model resemble to the previous model's. Similarly, model captures a number of features of business cycles. Again hours, investment and consumption is pro-cyclical while trade balance is counter-cyclical. However, as in the previous one, the approximated serial correlations do not match with the observed ones.

This thesis work aims to approximate the above models developed by Stephanie Schmitt-Grohé and Martin Uribe (2003) using GA-NN method and analyze whether this recent method can beat the first order approximation technique in reflecting the real life. Indeed, there are some advantages of the

GA-NN when compared to linear approximation. Unlike the linear approximation, GA-NN requires neither Euler nor first order equations. Besides, it is free of the continuity and the existence of derivatives of objective and policy functions. GA-NN method directly parameterizes the policy function with a neural network that is to be trained by genetic algorithm.

The GA-NN works as follows for the above RBC models; first, the policy functions are modeled in a neural network structure where weights of the network constitute the parameters of these policy functions. Then, to find these weights of the network structure, GA performs a global search over the feasible set of solutions starting from a random set of initial solutions. GA tries to maximize the sum of payoffs of the households defined over the given period  $T$ . If the problem is set for multiple initials, then the objective function becomes sum of overall payoffs across all initial states. So, before running the program, it requires to decide on the time period  $T$  and the number of initials that are going to be given to the program. This step of implementing the GA-NN method becomes the most challenging and most time consuming part of the study. Because, only way is to try all possible time periods and kinds of initials.

There are reasons for using genetic algorithm in training neural networks. Some special features of genetic algorithm make it preferable in solving computational problems over the gradient-descent and local search algorithms. First, since genetic algorithms are a class of adaptive and population based search methods that are inspired by biological evolution, it provides an additional advantage of being able to combine good solutions to get hopefully better solutions. Second, they perform global search over the solution space, while gradient-descent and local search algorithms explore the solution space by moving each step from the current solution to another solution in its neighborhood. Third, unlike local search methods, it prevents premature convergence and preserve the diversity in the population. Fourth, unlike gradient-descent

and local search algorithms, it is less likely to trap in a local optimum. In addition to these advantages of GA, it may encounter a number of problems when solving combinatorial problems. They may fail to find satisfactory results for some reasons. Here, the important point is to operate the most appropriate selection, crossover and mutation procedures. Additionally, it may take too much time to reach the optimum. But it is worth to try GA to solve complex computational problems.

In this work, Genesis 5.0 package developed by John J. Grefenstette (1990) is used. Genesis is a task independent optimization toolbox that just requires user to define his or her own problem by coding an evolution function. It is written in language C and run in Linux environment. The major procedures that are defined in the Genesis system are representation, initialization and generation including the selection, mutation, crossover, evaluation and some data collection procedures. When the user defined evolution is combined with these pre-defined operations, the program is ready to run.

The paper is organized as follows: In chapter 2, we describe the settings of the Stephanie Schmitt-Grohé and Martin Uribe's real business cycle models. In the first part of chapter 3, how the models are structured in a neural network is explained with a brief summary of neural network systems. Then, in the second part, procedures of genetic algorithm are presented in details. In chapter 4, statistical results and the impulse responses of the model are presented. Lastly, chapter 5 concludes.

## CHAPTER 2

### MODELS

#### 2.1 A Small Open Economy Model with Debt-Elastic Interest Rate Premium

First model considered in this study is a small open economy model with debt elastic interest rate premium in which the infinitely lived agents are supposed to solve the following problem;

$$\max E_0 \sum_{t=0}^{\infty} \beta^t \frac{[c_t - \omega^{-1} h_t^{\omega}]^{1-\gamma} - 1}{1-\gamma}, \quad 0 < \beta < 1 \quad (2.1)$$

subject to

$$y_t = A_t k_t^{\alpha} h_t^{1-\alpha}, \quad (2.2)$$

$$d_{t+1} = (1 + r_t)d_t - y_t + c_t + i_t + \Phi(k_{t+1} - k_t), \quad (2.3)$$

$$\Phi(k_{t+1} - k_t) = \frac{\phi}{2}(k_{t+1} - k_t)^2, \quad \phi > 0 \quad (2.4)$$

$$k_{t+1} = i_t + (1 - \delta)k_t, \quad \forall t \text{ and } k_0 \geq 0 \text{ given} \quad (2.5)$$

$$\lim_{j \rightarrow \infty} E_t \frac{d_{t+j}}{j} \leq 0 \quad (2.6)$$
$$\prod_{s=1}^{\infty} (1 + r_s)$$

Table 1: Parameters

| $\gamma$ | $\omega$ | $\phi$ | $\alpha$ | $r$  | $\delta$ | $\rho$ | $\sigma$ | $\beta$ |
|----------|----------|--------|----------|------|----------|--------|----------|---------|
| 2        | 1.455    | 0.028  | 0.32     | 0.04 | 0.1      | 0.42   | 0.0129   | 0.96    |

The representative agent maximizes the utility function (2.1) subject to Eqs. (2.2)-(2.5) and no-Ponzi constraint equation (2.6) where  $i_t, c_t, y_t, d_t, h_t$  and  $k_t$  are respectively gross investment, consumption, domestic output, foreign debt, labor devoted to production and physical capital stock at time  $t$ .  $\beta$  is the discount factor which is assumed to be constant over time.  $\delta$  is the rate of depreciation of physical capital and is between  $(0, 1)$ .  $\Phi$  denotes the capital adjustment cost and it is assumed that  $\Phi(0) = \Phi'(0) = 0$ . These assumptions on  $\Phi$  function ensures that steady state value of adjustment cost is zero in a non-stochastic model and domestic interest rate equals to the marginal product of capital minus its depreciation.

In the model,  $A_t$  is an exogenous productivity shock and evolves according to following equation;

$$\ln A_{t+1} = \rho \ln A_t + \epsilon_{t+1}; \epsilon_{t+1} \text{ NIID}(0, \sigma_\epsilon^2) \quad (2.7)$$

The structural parameters of the model are standard in real business cycle literature and match with the study of Mendoza (1991). The parameters are shown in Table 1.  $r_t$  is the interest rate at which the domestic agents can borrow in international markets at time  $t$ . In this model,  $r_t$  is supposed to increase in level of foreign debt and given by

$$r_t = r + p(d_t) \quad (2.8)$$

where  $r$  denotes the world interest rate and  $p(d_t)$  is the interest rate premium in the form of

$$p(d_t) = \psi(e^{d_t - \bar{d}} - 1) \quad (2.9)$$

Table 2: Parameters calibrated by Grohe and Uribe(2003) for Model 1

| $\bar{d}$ | $\psi$   |
|-----------|----------|
| 0.7442    | 0.000742 |

Here  $\bar{d}$  and  $\psi$  are calibrated by Grohe and Uribe(2003) so that the steady state level of foreign debt and volatility of current account to GDP ratio in this model are equal to the values imposed by their previous study "endogenous discount rate model" in which the parametrization is constructed based on the Canadian annual data over the period of 1946-1985. The calibrated parameters are shown in Table 2.

First order conditions of the model are

$$\lambda_t = \beta(1 + r_t)E_t\lambda_{t+1} \quad (2.10)$$

$$U_c(c_t, h_t) = \lambda_t \quad (2.11)$$

$$-U_h(c_t, h_t) = \lambda_t A_t F_h(k_t, h_t) \quad (2.12)$$

$$\lambda_t[1 + \Phi'(k_{t+1} - k_t)] = \beta E_t \lambda_{t+1} [A_{t+1} F_k(k_{t+1}, h_{t+1}) + 1 - \delta + \Phi'(k_{t+1} - k_t)] \quad (2.13)$$

Eqs. 2.11 and 2.12 together will yield

$$h_t^{\omega-1} = A_t F_h(k_t, h_t) \quad (2.14)$$

## 2.2 A Small Open Economy Model with Portfolio Adjustment Costs

Next model is a small open economy model with portfolio adjustment costs in which infinitely lived agents are supposed to solve the similar problem above. The only difference here is that, rather than a debt-elastic interest rate premium, agents are assumed to face convex costs of holding amount of debts

Table 3: Parameters calibrated by Grohe and Uribe(2003) for Model 2

| $\bar{d}$ | $\psi_2$ |
|-----------|----------|
| 0.7442    | 0.00074  |

different than a long-run level.

Interest rates constant and equal to world interest rate. Implying that,

$$r_t = r, \forall t \in (0, \infty) \quad (2.15)$$

And budget constraint of the agent is;

$$d_{t+1} = (1 + r_t)d_t - y_t + c_t + i_t + \Phi(k_{t+1} - k_t) + \frac{\psi_2}{2}(d_{t+1} - \bar{d})^2, \quad (2.16)$$

where  $\psi_2$  is parameter of the portfolio adjustment cost. Similarly,  $\bar{d}$  and  $\psi_2$  are calibrated by Grohe and Uribe(2003) to ensure that the steady state level of foreign debt and the volatility of the current account to GDP ratio in this model equal to the values imposed by the previous model. The calibrated parameters are shown in Table 3.

Other parameters of the model are the same with the values shown in Table 2.1.

First order conditions of the model are Eqs. (2.2), (2.6), (2.11)-(2.13), (2.15), (2.16) with the corresponding Euler equation in the form of;

$$\lambda_t[1 - \psi_2(d_t - \bar{d})] = \beta(1 + r_t)E_t\lambda_{t+1} \quad (2.17)$$

## CHAPTER 3

# ARTIFICIAL NEURAL NETWORK PARAMETERIZATIONS AND GENETIC ALGORITHM

### 3.1 Neural Networks and Neural Network

#### Parameterizations of the Policy Functions

Artificial neural network is a computational model that is inspired by the biological neural networks. It is composed of highly interconnected neurons. As in biological neural networks, learning is the major feature of the artificial neural networks and it is performed by adjusting the synaptic weights of the network. A trained neural network can be used to perform prediction and classification or recognize patterns.

A neural network is a system of any number of inputs and outputs. The inputs that reach to an neuron evolve with a given function at each layer and then continue to the last layer of the network. Outputs come out of the network after the last or hidden layer of the network. Policy functions of the above models are parameterized in a neural network as follows:

According to the model, at the end of time  $t-1$ ,  $k_t$ ,  $A_t$  and  $d_t$  are known. Therefore, at the beginning of period  $t$ , the household should choose  $k_{t+1}$ ,  $d_{t+1}$  and  $h_t$  to obtain  $y_t$  and  $c_t$  that is going to maximize his payoff. Thus, the

policy functions of  $k_{t+1}$ ,  $d_{t+1}$  and  $h_t$  are required to be some functions of  $k_t$ ,  $A_t$  and  $d_t$ . This is why, we accept  $k_t$ ,  $A_t$  and  $d_t$  as inputs to the system and  $k_{t+1}$ ,  $d_{t+1}$  and  $h_t$  as the outputs. However, rather than getting  $k_{t+1}$  from the network, we prefer to get  $i_t$  for computational issues. The network structures are shown in Figure 1.

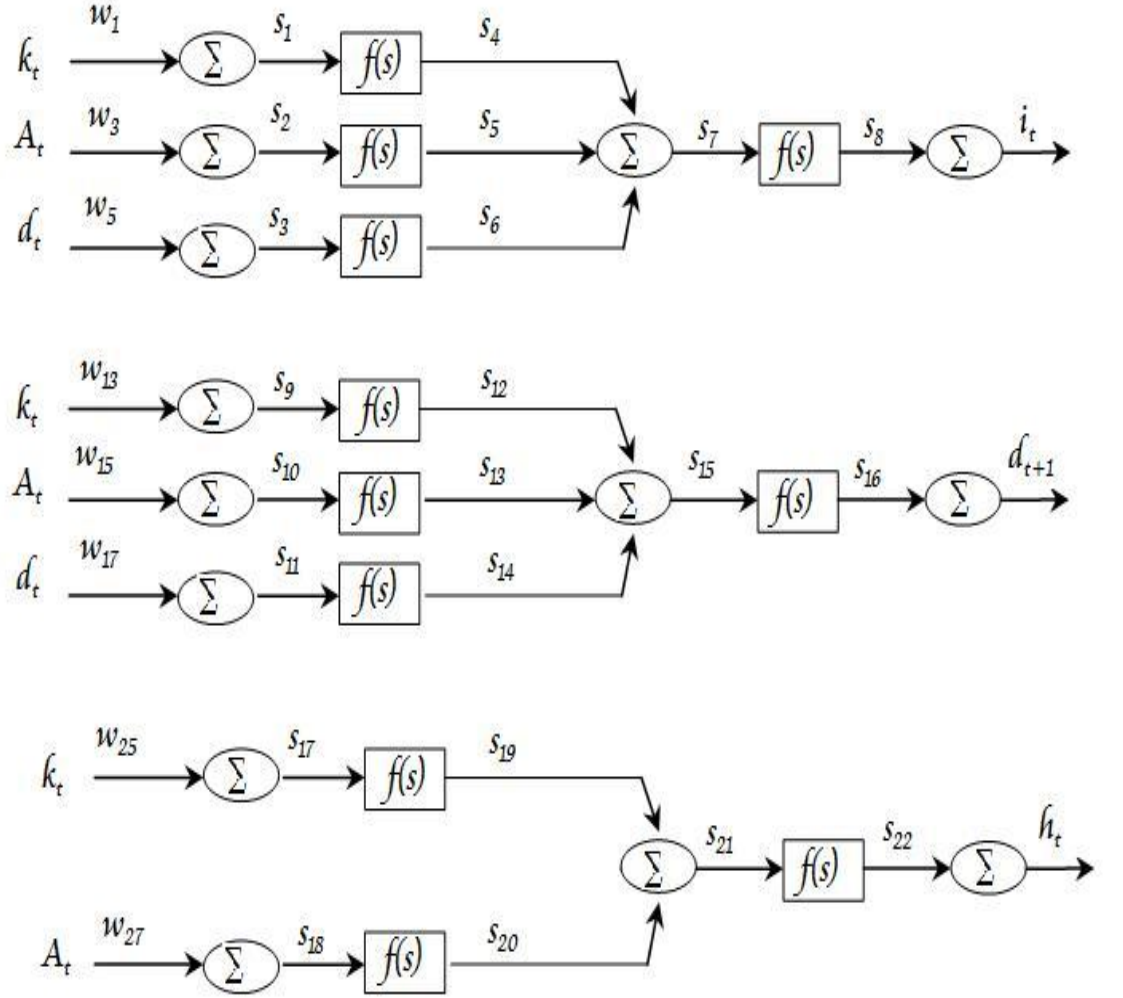


Figure 1: Neural Network Structures

In the first network,  $i_t$  is determined by a system of equations. In the first layer, inputs enter to a neuron and alter according to the following equations:

$$s_1 = w_0 + w_1 k_t \quad (3.1)$$

$$s_2 = w_2 + w_3 A_t \quad (3.2)$$

$$s_3 = w_4 + w_5 d_t \quad (3.3)$$

Then, network applies a sigmoidal activation or transfer function on the resulting values.

$$s_4 = \frac{1}{1+e^{-s_1}} \quad (3.4)$$

$$s_5 = \frac{1}{1+e^{-s_2}} \quad (3.5)$$

$$s_6 = \frac{1}{1+e^{-s_3}} \quad (3.6)$$

In the rest of the network, system continues as follows:

$$s_7 = w_6 + w_7 s_4 + w_8 s_5 + w_9 s_6 \quad (3.7)$$

$$s_8 = \frac{1}{1+e^{-s_7}} \quad (3.8)$$

$$i_t = (w_{10} - w_{11}) s_8 + w_{11} \quad (3.9)$$

Policy functions of  $d_t$  and  $h_t$  are represented by similar neural networks, yet their own corresponding vectors, as shown in Figure 1. It should be noted that when policy function of  $h_t$  is modeled in an network structure, to avoid computational complexities, first order conditions are considered and it is found to be appropriate to take  $h_t$  as a function of  $k_t$  and  $A_t$ .(see Equation (2.14)).

After the policy functions are represented in a network, next step is to train the network in order to obtain an optimal weight set that is going to maximize the payoff function of the household given in Equation (2.1). So at this point, to find the optimal weights, genetic algorithm is used. Once the optimal weights are obtained, the optimal path can be driven. In this study, the algorithm is run over the period  $T=500$  and GA finds the optimal path over

this period. In order to prevent the dependence of optimal path on the given initials, the program is run with 3 initials and the overall payoff is maximized.

## 3.2 Genetic Algorithm

Genetic algorithms are a class of adaptive and population based search methods inspired by biological evolution. These techniques were first developed by John Holland and his associates at the University of Michigan in the 1960s and 1970s for solving combinatorial optimization problems. The first systematic usage of genetic algorithm was in Holland's book *Adaptation in Natural and Artificial Systems* published in 1975. Besides these pioneering works of genetic algorithm, they were also used for game-playing programs, biological processes and solving pattern-recognition problems. As the name represents, genetic algorithm mimics genetic evolution of biological systems. The basic idea is to construct a population of candidate solutions at each period that evolves under a selective process that favors better solutions in the name of Darwin's survival of the fitness.

In all organisms, the fundamental unit of information is gene. A gene is defined as a portion of chromosome that affects a single character or phenotype. Two peer genes located in the mutual chromosomes determine specific information about a feature of organism. This location of the features on the chromosomes is called locus. The individuals are just a reflection of the chromosomes formed by the order of the genes. Similarly, in GA, solutions to the problem are represented by a linear string that is comprised of either binary or real numbers. In Figure 2, the terminology of natural organisms versus GA are listed.

In this thesis study, genes of the information string are weights of the neural network that is given in Section 3.1. So, each solution of this problem is composed of the set of these weights. Genetic algorithm tries to obtain the optimal set of weights that are going to maximize the utility of the household.

| Natural vs GA Terminology |                                  |
|---------------------------|----------------------------------|
| Natural                   | GA                               |
| chromosome                | → string                         |
| gene                      | → feature, character or detector |
| locus                     | → string position                |
| genotype                  | → structure                      |
| phenotype                 | → decoded structure              |
| population                | → set of solutions               |

Figure 2: Terminology of Natural Organisms versus GA

Which steps genetic algorithm follows to find the optimal set of weights are explained in details in the coming section.

### 3.2.1 Process of a Standard Genetic Algorithm

In this section, main components and the steps of GA will be explained. Flow chart of a standard GA is shown in Figure 3. The basic steps are as follows:

#### Step1: Generate an initial population of strings (genotypes)

In order to start searching the feasible solution space, firstly the algorithm should start from an initial feasible population or generation. This initial population in GA is usually generated at random. Indeed, it is possible to start with a population of good quality by using some heuristic techniques. However, some reports have found that this kind of approaches may increase the chance of premature convergence meaning that losing diversity in the population.

Another important point about the initialization of GA is to determine the population size of the generations. It is the case that, small populations have the risk of under-covering the solution space and early convergence, while large populations result in severe computational penalties. Some experimental work that have been conducted by J.T. Alander(1992) suggested that a value between the length of the string and twice of the length of the string is optimal

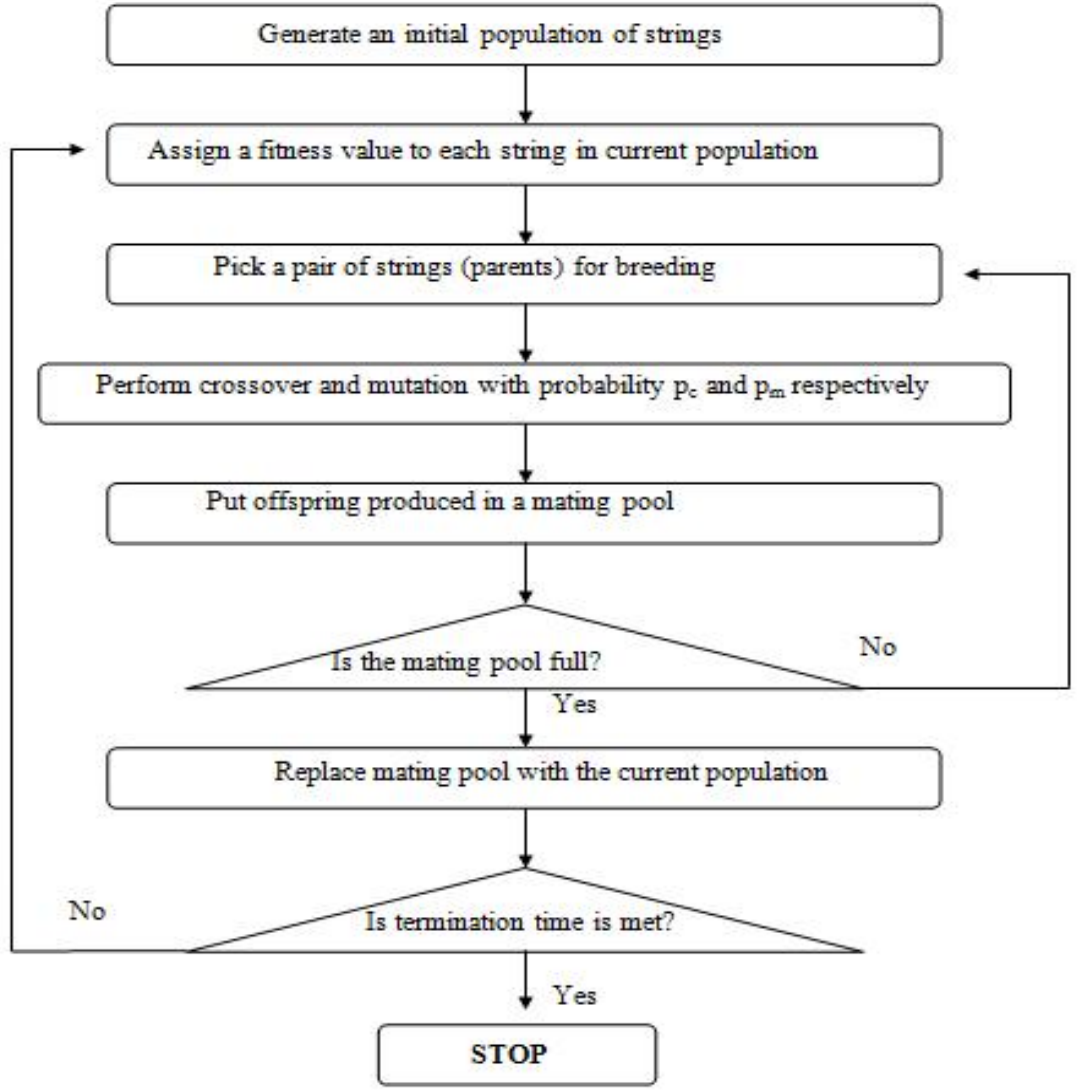


Figure 3: Flow Chart of a Standard GA

for the problems. We chose 50 as the population size since length of the string is 33 in this problem. In this study, real valued genes are considered rather than binary ones.

### Step2: Assign a fitness value to each string in the population

Each string in the population is evaluated according to a fitness function. So, the strings are decoded to the real problem and their relative fitness is obtained. In this study, each string that is composed of a set of weights is used to calculate the sum of utility over all initial values and over the period  $T$ .

### **Step3: Pick a Pair of Strings for Breeding**

This step is the process of selecting parents that is going to form the next generation. Highly fit strings are selected according to some procedures. Here ranking algorithm is preferred, which means higher the fitness score higher the probability of selecting that string. Moreover, elitist selection is also used to guarantee that the best member of the current population will survive into the next generation. So for our problems, the weight sets that give better utilities to household will have more chance to survive in the next generation and the best weight set, meaning that providing the highest utility will be absolutely taken to the next generation.

### **Step 4: Generate a New Population by Applying Genetic Operators**

Selected parents are next exposed to genetic operators and the resulting offsprings are put into the mating pooling. New strings become a member of the next generation and this selection and reproduction processes are repeated until the mating pool reaches the population size. The Genetic Operators are as follows:

Crossover:

In crossover, one cut-point is selected at random and the two chromosomes swap their information on these points. So in GA, two genes on the string, one from the first parent and the other from the second parent are exchanged.

Mutation:

One of the major problems encountered while using population search algorithms is the premature convergence, all solutions in the population having a tendency to be equal to the best solution in that population. If this is the case, then algorithm behaves like a local search algorithm and loses its advantage of being an population based search. So, to overcome such kind of problems, it is desirable to implement operators that is going to preserve the diversity in the population. The best known tool in doing this is running a mutation

operator. In mutation, one point on chromosome is selected and its value is changed. Similarly, in GA, one gene over the string is replaced one another value. This procedure will prevent to stuck in a local optimum.

These two operators, crossover and mutation are applied with a probability of  $p_c$  and  $p_m$  respectively. In Figure 4, a general procedure of generating new population is visualized.

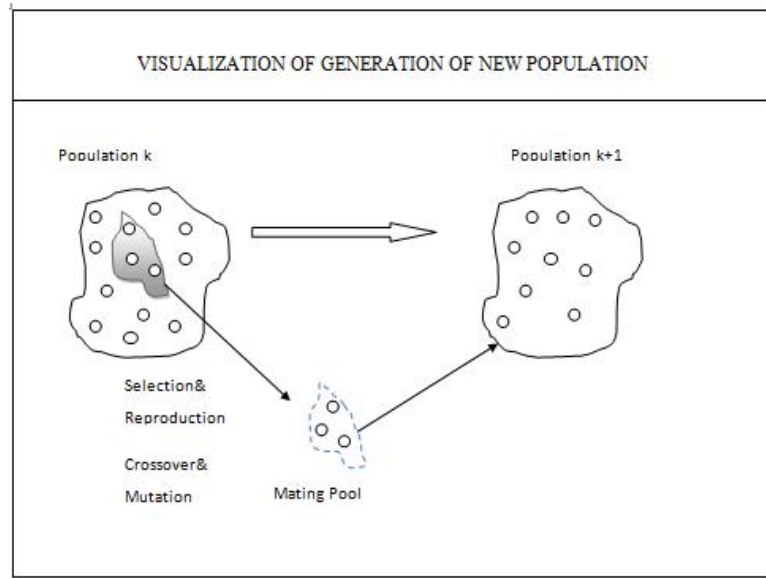


Figure 4: Generation of new population

In this study, probability of crossover that is the probability of having a crossover between two parents is accepted as 0.6 while the probability of mutation that is having a mutation is accepted as 0.001. Some studies suggest that these values are optimal for large population sizes.

## CHAPTER 4

### RESULTS

In this part, the results obtained by solving the underlying small open economy models using GA-NN method are presented. At the first step, by using the network weights that are maximizing the payoff of the household, the optimal paths are computed. Then the standard deviations, serial correlations and correlations with output of the endogenous variables are calculated to get the dynamics of business-cycles. In analyzing the statistics of the resulting path of endogenous variables, first and last 50 periods are excluded to avoid biases arising due to the initial set selections. Then the impulse responses of the variables to unit technology shock are presented. Lastly, the second moments and impulse responses found by GA-NN method are compared with the Grohe and Uribe(2003)'s results obtained by applying linear approximations to the policy functions. Second moments of the Canadian data over the period 1946-1985 are also compared with our results.

To solve the models with GA-NN, Genesis Version 5.0 developed by Grefenstte (1990) as a free genetic algorithm software package is used. After writing an evaluation program for these specific problems, Genesis is run on a Sun Microsystems Enterprise 4000 over Unix operating system using a Telnet connection to the server. For each model, we run the simulation for 1000000 generations with a population size of 50, crossover probability of 0.6 and mu-

tation probability of 0.001. The models are run over the period  $T=500$  with 3 initials including the steady state level.

The initial set of capital stock is taken as 3.22, 3.3977, 3.567 where 3.3977 is the steady state level and initial set of debt is taken as 0.706, 0.7442, 0.78 where 0.7442 is the steady state level.

## 4.1 Simulation Results of Models

Our results with the observed second moments of Canadian data are summarized in Table 4.

In the comparison of the underlying methods, namely GA-NN and linear approximation, the standard deviations, serial correlations and correlations with output will be considered separately. In addition, the computed impulse responses to a unit technology shock will be compared for each model.

The GA-NN method approximates the standard deviations of variables fairly same with the data, except the standard deviation of investment and trade balance to output ratio. Both models solved by GA-NN underestimate the standard deviation of investment and overestimate the standard deviation of trade balance to output ratio. On the other hand, linear approximation technique gives better approximations for the standard deviations of these two variables. The other approximated standard deviations of these two techniques are very close to each other. So considering these findings, linear approximation can be said to be more powerful in the approximation of the standard deviations. However, it should be stated that GA-NN approximations capture one of the stylized facts of real business cycles. It meets the ranking requirement of standard deviation of consumption, output and investment, in the increasing order of consumption, output and investment. That is, the GA-NN can be said to be good in predicting the consumption smoothing behavior of households. Lastly, GA-NN generates a pro-cyclical behavior in consumption, technological shocks, investment and hours as expected in real business cyc-

Table 4: Statistical moments

|                                                           | Data  | GA-NN M1 | GA-NN M2 | LA M1  | LA M2  |
|-----------------------------------------------------------|-------|----------|----------|--------|--------|
| $\text{std}(y_t)$                                         | 2.81  | 3.1      | 2.91     | 3.1    | 3.1    |
| $\text{std}(c_t)$                                         | 2.5   | 2.5      | 2.5      | 2.7    | 2.7    |
| $\text{std}(i_t)$                                         | 9.8   | 5.4      | 5        | 9      | 9      |
| $\text{std}(h_t)$                                         | 2     | 2.2      | 2        | 2.1    | 2.1    |
| $\text{std}(\frac{tb_t}{y_t})$                            | 1.9   | 3        | 2.9      | 1.8    | 1.8    |
| $\text{corr}(y_t, y_{t-1})$                               | 0.61  | 0.6      | 0.62     | 0.62   | 0.62   |
| $\text{corr}(c_t, c_{t-1})$                               | 0.70  | 0.65     | 0.73     | 0.78   | 0.78   |
| $\text{corr}(i_t, i_{t-1})$                               | 0.31  | 0.51     | 0.43     | 0.069  | 0.069  |
| $\text{corr}(h_t, h_{t-1})$                               | 0.54  | 0.53     | 0.61     | 0.62   | 0.62   |
| $\text{corr}(\frac{tb_t}{y_t}, \frac{tb_{t-1}}{y_{t-1}})$ | 0.66  | 0.65     | 0.62     | 0.51   | 0.5    |
| $\text{corr}(c_t, y_t)$                                   | 0.59  | 0.99     | 0.98     | 0.84   | 0.85   |
| $\text{corr}(i_t, y_t)$                                   | 0.64  | 0.98     | 0.94     | 0.67   | 0.67   |
| $\text{corr}(h_t, y_t)$                                   | 0.8   | 0.99     | 0.99     | 1      | 1      |
| $\text{corr}(\frac{tb_t}{y_t}, y_t)$                      | -0.13 | -1       | -0.99    | -0.044 | -0.043 |

Second moments obtained by GA-NN and Linear Approximation Methods for the open economies with debt elastic interest rate premium (M1) and with portfolio adjustment costs (M2), the LA results are taken from Grohe and Uribe(2003) and standard deviations are measured in percent per year

les.(see Figure 5 and Figure 6).

In the second part of Table 4.1, approximated serial correlations are reported. The serial correlations obtained by GA-NN are very close to the Canadian data. It approximates the serial correlations even better than the linear approximation. Especially, while linear approximation gives a serial correlation of investment far from the data, GA-NN succeeds to mimic this serial correlation.

In the last part of Table 4.1, correlations of variables with output is presented. Here, GA-NN fails to get the correlations of data, while linear approximation provides relatively closer results to the data and same signs with the data. However, linear approximation also overestimates the correlations of consumption, trade balance to output and hours with output. So, it can

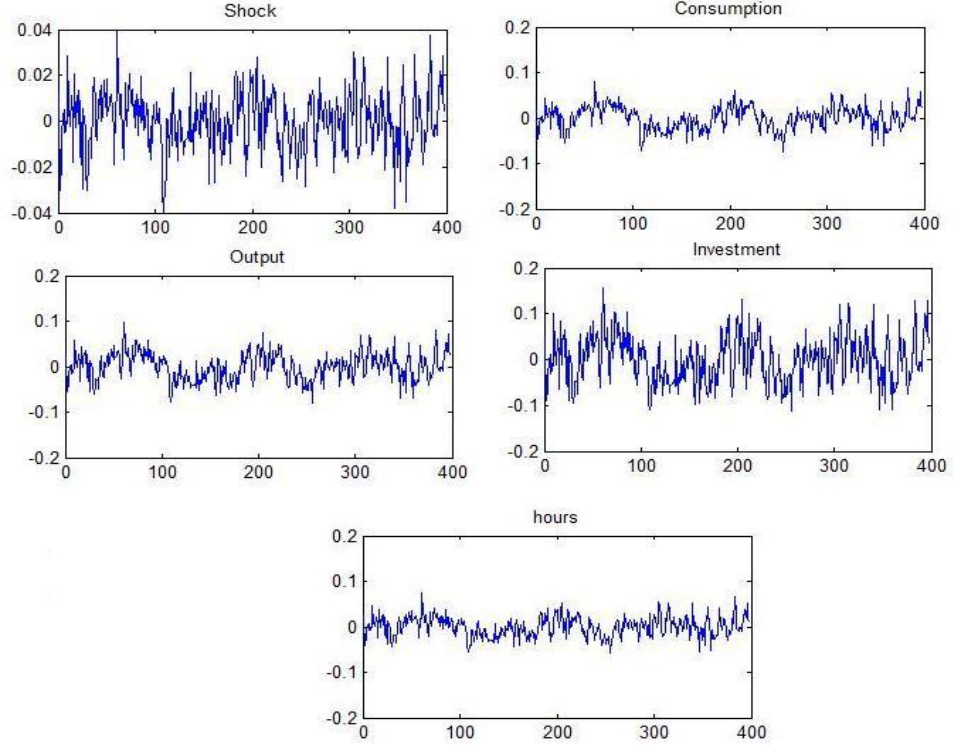


Figure 5: Pro-cyclic behaviors, for Model 1

be stated that linear approximation is not very good in mimicking these correlations. Since both methods have a weak performance, the models used to mimic the real business cycle of Canada may not be appropriate for the approximation of correlations with output.

The unpredicted results we observe in our findings can be summarized as follows. Although it is expected to have an acyclical behavior between trade balance to output ratio and output, we get a value that is close to -1, meaning that they are counter-cyclical. Next, GA-NN generates a correlation of 0.99 between investment and saving, implying that there is a high co-movement between saving and investment. This correlation in the observed data is 0.45. So, our method overestimates the correlation between saving and investment.

In the rest of this chapter, we will consider the impulses responses to unit technology shocks that are obtained for each model and each method. Here, the impulse responses are obtained by calculating the optimal path using the same optimal set of weights. But when generating the technology shocks, we

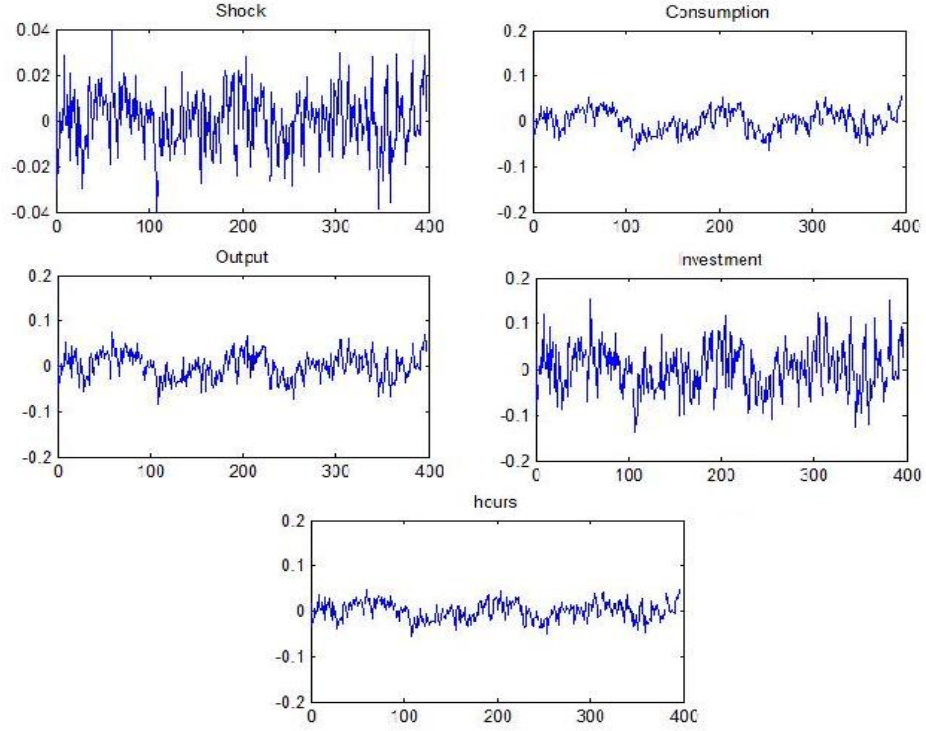


Figure 6: Pro-cyclic behaviors, for Model 2

just give a positive shock to the technology at time 0. That is for the Eqn. (2.7), we set  $\epsilon_0$  to 1 and  $\sigma_\epsilon$  to zero. Since the logarithm of technology follows an AR(1) process, the generated logarithm of technology at time  $t$  becomes  $\rho^t$  for all  $t$ .

For Model 1, the impulse responses found by GA-NN and linear approximation are shown in Figure 7 and in Figure 8. The direction of the impulse responses obtained by these two different methods are same for the variables and the magnitudes of impulse responses of the variables are close to each other except investment. This result is an unavoidable cause of GA-NN which has not managed to mimic the standard deviation of investment. In addition, there is a small difference between the impulse responses of the consumption. GA-NN provides a higher increase in consumption in the period in which the positive technology shock occurs.

For Model 2, namely model with portfolio adjustment cost, impulses responses obtained by GA-NN and linear approximation are shown in Figure 9

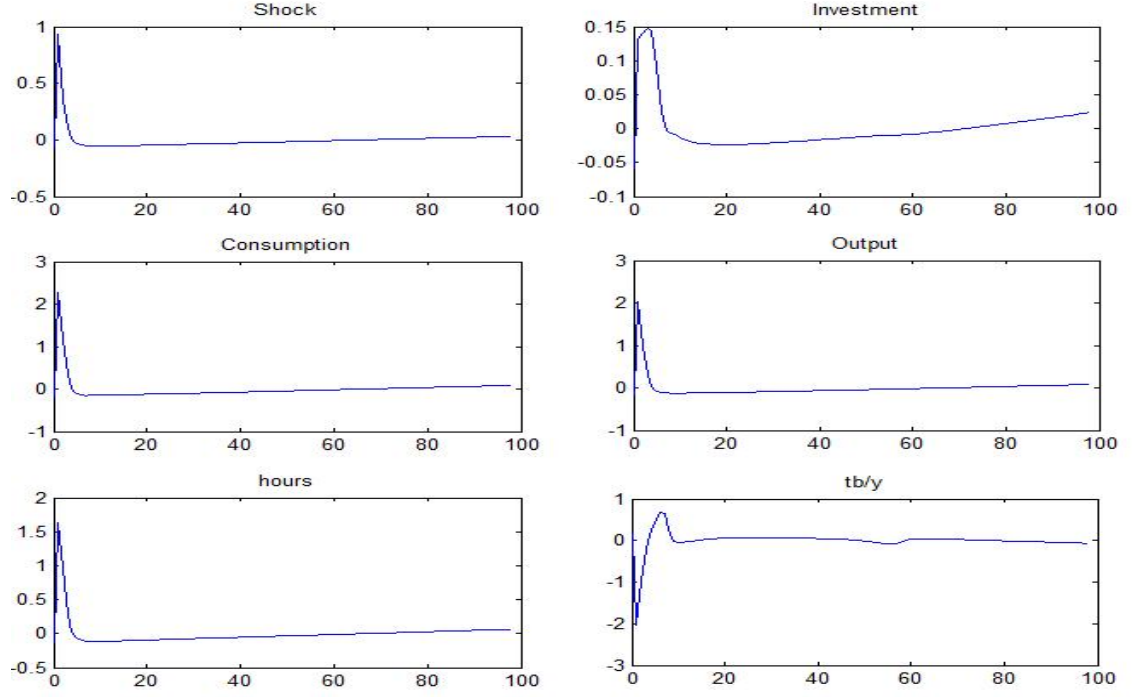


Figure 7: Impulse-responses to unit technology shock for Model 1 obtained by GA-NN

and Figure 10. Here, GA-NN provides different impulse responses than linear approximation.

Grohe and Uribe(2003) states that when these two models are solved by linear approximation technique, their log-linearized first order conditions will imply similar dynamics. So in the Eqns. (2.3) and (2.16), existence of interest premium and portfolio adjustment cost will imply same dynamics after the log-linearization of first order conditions. However, these two models are not same. From the Eqn (2.3), we have

$$y_t - c_t - i_t = (1 + r + \psi(e^{d_t - \bar{d}} - 1))d_t - d_{t+1} + \Phi(k_{t+1} - k_t) \quad (4.1)$$

and from Eqn (2.6) we have

$$y_t - c_t - i_t = (1 + r)d_t - d_{t+1} + \Phi(k_{t+1} - k_t) + \frac{\psi_2}{2}(d_{t+1} - \bar{d})^2 \quad (4.2)$$

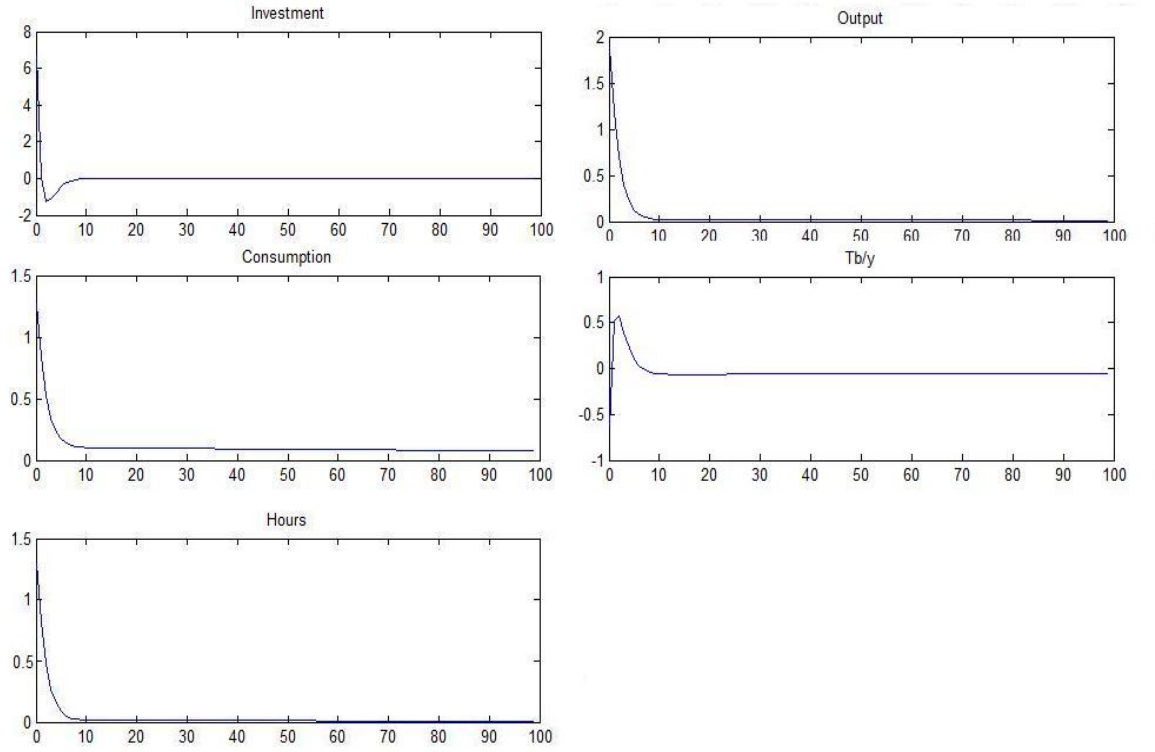


Figure 8: Impulse-responses to unit technology shock for Model 1 obtained by LA, taken from Grohe and Uribe(2003)

In the first equation above, households are punished through an interest rate premium, if their debt levels are above their long run level. On the other hand, in the next equation, households are punished if their debt level is far from the long-run level. Thus, their dynamics should not be same as linear approximation imply. At this point, unlike log-linear approximation, GANN which directly considers the functional forms of functions, can be said to provide better results for observing the different dynamics of different models.

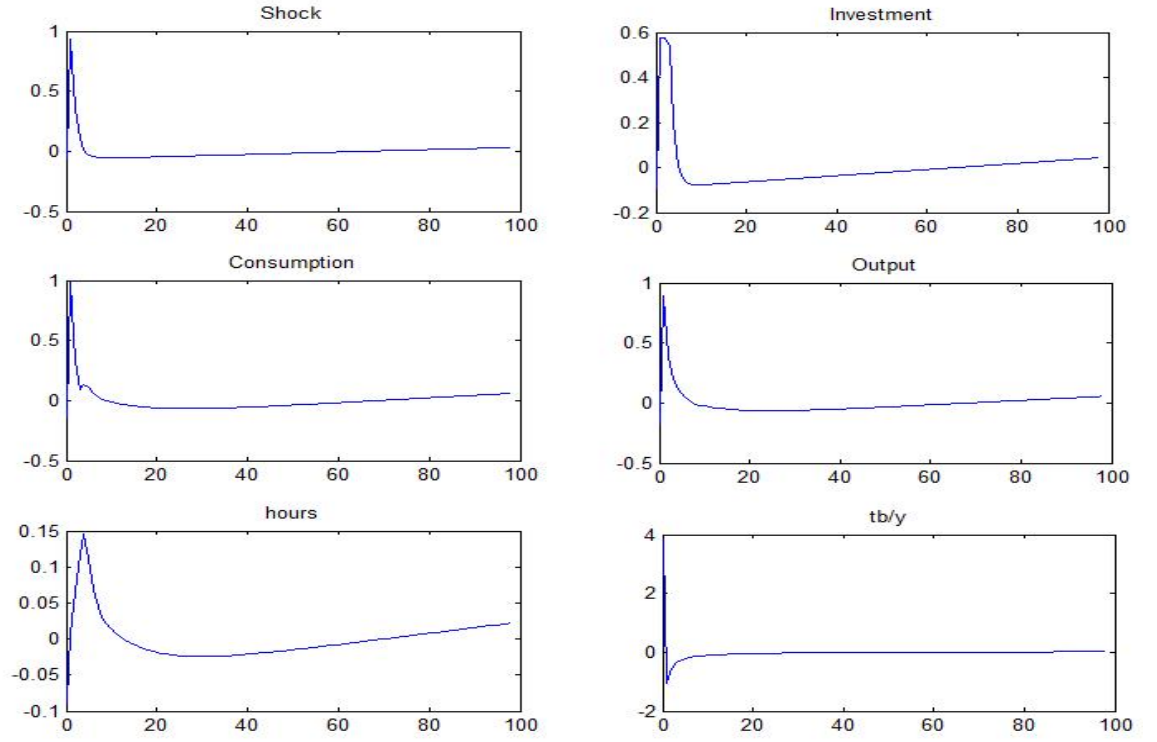


Figure 9: Impulse response to unit technology shock obtained by GA-NN

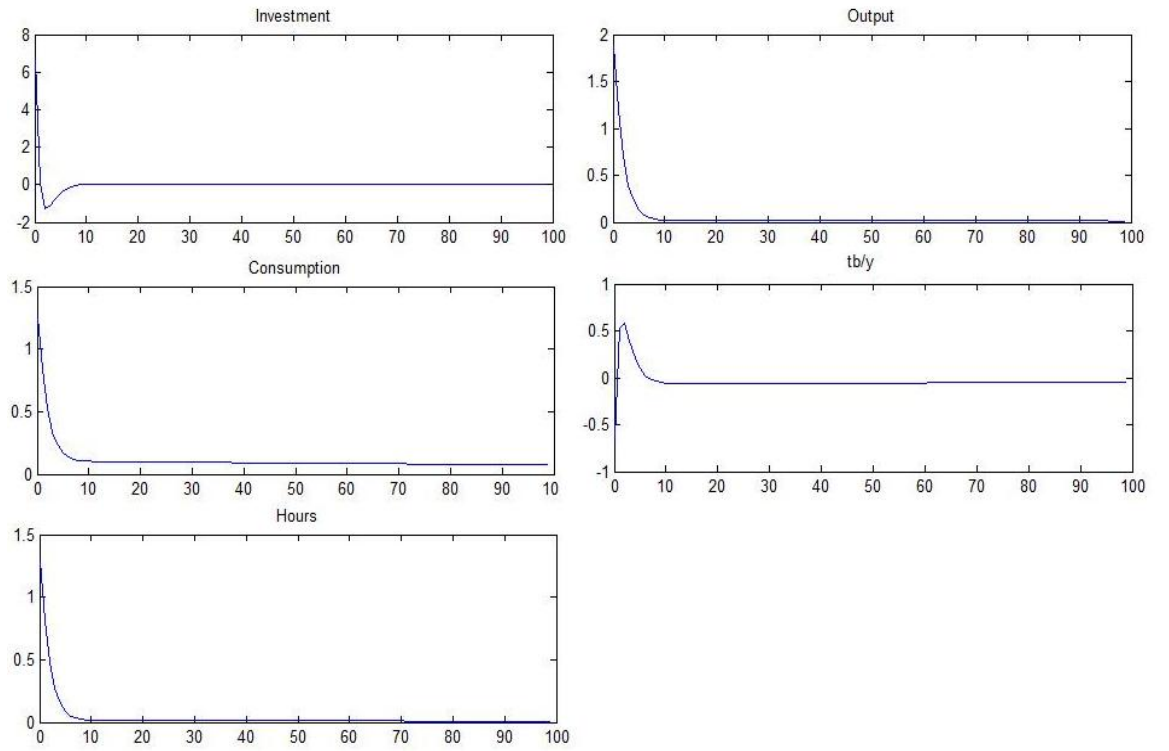


Figure 10: Impulse responses to unit technology shock for Model 2 obtained by LA, taken from Grohe and Uribe(2003)

## CHAPTER 5

## CONCLUSION

In this thesis work, we proposed a direct numerical solution methodology, namely neural network trained by genetic algorithm, in order to solve the open economies with debt elastic interest rate and with portfolio adjustment cost that are studied and calibrated by Grohe and Uribe(2003). In the model with debt elastic interest rate premium, households are assumed to face with an interest rate premium that is increasing with level of foreign debt. In the model with portfolio adjustment cost, households are assumed to face a convex cost of holding debt far from its long run level. We applied GA-NN to get the optimal paths of the models and then we calculated the second moments and impulse responses. These second moments and impulse responses were then compared with the corresponding ones that are obtained by linear approximation. Linear approximation results were taken from Grohe and Uribe (2003).

We expect GA-NN to provide better results than linear approximation, since GA-NN searches the whole solution space and directly approximates the optimal network weights that are going to maximize the utility function of household. Unlike linear approximation, GA-NN does not require the existence of objective and policy functions. These features of GA-NN make its results more robust than the results of linear approximation. GA-NN also enables to solve large problems in which calculating Euler and first order conditions

are quite difficult. So, for many problems GA-NN can be preferred to linear approximation technique.

In these two models, GA-NN performed quite well in some aspects, but it also failed to capture some data features. While linear approximation could not approximate the serial correlations of variables well, GA-NN showed better performance in the approximation of these correlations. On the other hand, GA-NN did not perform as well as linear approximation in mimicking the standard deviation of investment and correlations of variables with output. Lastly, while linear approximation ends with same dynamics for these two different models, GA-NN succeeded to distinguish the differences in the dynamics, since it directly deals with the functional forms of functions rather than the log-linearized versions. This result is obviously observed in the impulse response functions of these two different approximation techniques.

For the extension of this study, performance of GA-NN can be improved. First, a study can be conducted to find the optimal time period, since deciding on time period manually makes GA-NN time consuming. The results we had all depend on the assumptions on transfer functions of the neural network and operations of GA-NN. So, next for these kinds of problems, the assumptions can be studied. Transfer functions of neural network can be developed in order to have more accurate representation of the policy functions. Another extensions can be to solve other real business cycle models by GA-NN in order to conclude whether GA-NN performs well in analyzing the real business cycles.

# BIBLIOGRAPHY

- Alander, J. T. 1992. *On Optimal Population Size of Genetic Algorithms*. European Computer Conference, IEEE Computer Society Press, **92**, 65-70.
- Alemdar, N. M., Sirakaya, S. and Turnovsky S.J. 2006. *Feedback Approximation of the Stochastic Growth Model by Genetic Neural Networks*. Computational Economics, **27**, 185-206.
- Grefenstette, J. J. 1990. *A User's Guide to GENESIS Version 5.0*
- Hertz, A. and Kobler, D. 2000. *A Framework for the Description of Evolutionary Algorithms*. European Journal of Operational Research, **126**, 1-12.
- Holland, J., *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: The University of Michigan Press, (1975).
- Kim, J. and Kim S.H. 2003 *Spurious Welfare Reversals in International Business Cycle Models*. Journal of International Economics, **60**, 471-500
- Kydland, E. F. and Prescott, E. C. 1982. *Time to Build and Aggregate Fluctuations*. Econometrica, **50**, 1345-1370.
- Lucas, Robert E. 1977. *Understanding Business Cycles*. Carnegie-Rochester Conference Series on Public Policy, **5**, 7-29.
- Mendoza, E. G. 1991 *Real Business Cycles in a Small Open Economy*. The American Economic Review, **81**, 797-818.
- Plosser, Charles I. 1989 *Understanding real business cycles*. Journal of Economic Perspectives, **3**, 51-77.
- Reeves, Colin R. and Rowe, Jonathan E. 2002 *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory*. Kluwer Academic Publishers.
- Sargent, T. J. 1987 *Dynamic Macroeconomic Theory*. Cambridge MA: Harvard University Press.
- Schmitt-Grohé, S. and Uribe, M. 2003 *Closing Small Open Economy Models*. Journal of International Economics, **61**, 163-185.
- Schmitt-Grohé, S. and Uribe, M. 2004 *Solving Dynamic General Equilibrium Models Using a Second Order Approximation to the Policy Function*. Journal of Economic Dynamics and Control, **28**, 755-775.

# APPENDICES

## APPENDIX A. SOURCE CODES

### 1. Open Economy with Debt Elastic Interest Rate Premium

```
#include "extern.h"
#include <time.h>
#define T 5
#define TMAX 500

extern double gaussrand();

double eval(str, length, vect, genes)
char str[]; /*string representation */
int length; /*length of bit string */
double vect[]; /*floating point representation */
int genes; /*number of elements in vect*/

{

    /*----- Defining Variables -----*/

    register int t, i;
    double ans=0.0;
    double k[TMAX]; /* capital */
    double h[TMAX]; /* labor */
    double j[TMAX]; /* investment */
    double d[TMAX]; /* debt stock at the begining of time t */
    double c[TMAX]; /* consumption */
    double r[TMAX]; /* interestrate */
    double tb[TMAX]; /* trade balance */
    double y[TMAX]; /* output */
    double coef[TMAX];
    double epsilon[TMAX-1];
    double ltheta[TMAX];
    double s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16,s17,
        s18,s19,s20,s21,s22,s23,s24;

    /*----- Initilization of variables----- */

    double ltheta0[T]={-0.18633,-0.058,0.000,0.05558,0.1555};
    double k0[T]={3.051,3.22,3.3977,3.567,3.737};
    double d0[T]={0.66,0.706,0.7442,0.78,0.82};
    double delta=0.1;
    double beta=0.96;
    double rho=0.42;
```

```

double sigma=0.0129;
double mu=0.0;
double alpha=0.32;
double fi2=0.000742;
double fi=0.028;
double dbar=0.7442;
double tau=1.455;
double gama=2;
double r_world=0.04;
static unsigned char first=1;
FILE *fp, *fopen();

/*----- Generation of technology(t) -----*/

if (first==1)
{
    fp=fopen("deirdata4", "w"); // open deirdata.txt
    srand(151982);

    for (i = 0; i < T; i++)
    {
        ltheta[0] = ltheta0[i];
        theta[i][0] = exp(ltheta[0]);
        fprintf(fp, "%7.4f ", theta[i][0]);
        for (t = 1; t < TMAX; t++)
        {
            epsilon[t-1]=gaussrand()*sigma;
            ltheta[t] = rho*ltheta[t-1] + epsilon[t-1];
            theta[i][t] = exp(ltheta[t]);
            fprintf(fp, "%7.4f ", theta[i][t]);
        }
        fprintf(fp, "\n");
    }
    fclose(fp);
    first=0;
}
/*-----*/

for (i = 1; i < 4; i++)
{
    for (t = 0; t < TMAX-1 ; t++)
    {
        if (t == 0)
        {
            k[t]=k0[i];
            d[t]=d0[i];
            coef[t]=1.0;

            // getting investment through network weights

            s1=vect[0]+vect[1]*k[t];
            s2=vect[2]+vect[3]*theta[i][t];
            s3=vect[4]+vect[5]*d[t];
            s4=1/(1+exp(-s1));
            s5=1/(1+exp(-s2));
            s6=1/(1+exp(-s3));
            s7=vect[6]+vect[7]*s4+vect[8]*s5+vect[9]*s6;
            s8=1/(1+exp(-s7));

```

```

j[t]=(vect[10]-vect[11])*s8+vect[11];
      k[t+1]=j[t]+(1-delta)*k[t];
      if( k[t+1]< 0)
      {
          ans+=-10000000000;
          k[t+1]=0;
      }
r[t]=r_world+fi2*(exp(d[t]-dbar)-1);

// getting debt through network weights

s9=vect[12]+vect[13]*k[t];
s10=vect[14]+vect[15]*theta[i][t];
s11=vect[16]+vect[17]*d[t];
s12=1/(1+exp(-s9));
s13=1/(1+exp(-s10));
s14=1/(1+exp(-s11));
s15=vect[18]+vect[19]*s12+vect[20]*s13+vect[21]*s14;
s16=1/(1+exp(-s15));
d[t+1]=(vect[22]-vect[23])*s16+vect[23];
tb[t]=d[t]*(1+r[t])-d[t+1];

// getting labor through network weights

s17=vect[24]+vect[25]*k[t];
s18=vect[26]+vect[27]*theta[i][t];
s19=1/(1+exp(-s17));
s20=1/(1+exp(-s18));
s21=vect[28]+vect[29]*s19+vect[30]*s20;
s22=1/(1+exp(-s21));
h[t]=(vect[31]-vect[32])*s22+vect[32];
      if( h[t]< 0)
      {
          ans+=-10000000000;
          h[t]=0;
      }
y[t]=theta[i][t]*pow(k[t],alpha)*pow(h[t],(1-alpha));
c[t]=y[t]-j[t]-tb[t]-(fi/2)*(k[t+1]-k[t])*(k[t+1]-k[t]);
      if (c[t] <= (1/tau)*pow(h[t],tau))
          ans+=-10000000000;
      else
          ans+=coef[t]*(pow((c[t]-(1/tau)*pow(h[t],tau)),(1-gama))-1)
          /(1-gama);
}

else // for t from [1,TMAX-2]
{
    coef[t]=coef[t-1]*beta;

// getting investment through network weights

s1=vect[0]+vect[1]*k[t];
s2=vect[2]+vect[3]*theta[i][t];
s3=vect[4]+vect[5]*d[t];
s4=1/(1+exp(-s1));
s5=1/(1+exp(-s2));
s6=1/(1+exp(-s3));
s7=vect[6]+vect[7]*s4+vect[8]*s5+vect[9]*s6;

```

```

s8=1/(1+exp(-s7));
j[t]=(vect[10]-vect[11])*s8+vect[11];
      k[t+1]=j[t]+(1-delta)*k[t];
      if( k[t+1]< 0)
      {
          ans+=-10000000000;
          k[t+1]=0;
      }
r[t]=r_world+fi2*(exp(d[t]-dbar)-1);

// getting debt through network weights

s9=vect[12]+vect[13]*k[t];
s10=vect[14]+vect[15]*theta[i][t];
s11=vect[16]+vect[17]*d[t];
s12=1/(1+exp(-s9));
s13=1/(1+exp(-s10));
s14=1/(1+exp(-s11));
s15=vect[18]+vect[19]*s12+vect[20]*s13+vect[21]*s14;
s16=1/(1+exp(-s15));
d[t+1]=(vect[22]-vect[23])*s16+vect[23];
tb[t]=(1+r[t])*d[t]-d[t+1];

// getting labor through network weights

s17=vect[24]+vect[25]*k[t];
s18=vect[26]+vect[27]*theta[i][t];
s19=1/(1+exp(-s17));
s20=1/(1+exp(-s18));
s21=vect[28]+vect[29]*s19+vect[30]*s20;
s22=1/(1+exp(-s21));
h[t]=(vect[31]-vect[32])*s22+vect[32];
      if( h[t]< 0)
      {
          ans+=-10000000000;
          h[t]=0;
      }
y[t]=theta[i][t]*pow(k[t],alpha)*pow(h[t],(1-alpha));
c[t]=y[t]-j[t]-tb[t]-(fi/2)*(k[t+1]-k[t])*(k[t+1]-k[t]);
      if (c[t] <= (1/tau)*pow(h[t],tau))
          ans+=-10000000000;
      else
          ans+=coef[t]*(pow((c[t]-(1/tau)*pow(h[t],tau)),(1-gama))-1)
          /(1-gama);
    }
  }
}
return(ans);
}

```

## 2. Open Economy with Portfolio Adjustment Cost

```
#include "extern.h"
#include <time.h>

#define T 5
#define TMAX 500

extern double gaussrand();

double eval(str, length, vect, genes)
char str[]; /*string representation */
int length; /*length of bit string */
double vect[]; /*floating point representation */
int genes; /*number of elements in vect*/

{

/*----- Defining Variables -----*/

register int t, i;
double ans=0.0;
double k[TMAX]; /* capital */
double h[TMAX]; /* labor */
double j[TMAX]; /* investment */
double d[TMAX]; /* debt stock at the begining of time t */
double c[TMAX]; /* consumption */
double tb[TMAX]; /* trade balance */
double y[TMAX]; /* output */
double coef[TMAX];
double epsilon[TMAX-1];
double ltheta[TMAX];
double s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16,s17,s18,
s19,s20,s21,s22,s23,s24;

/*----- Initilization of variables----- */

double ltheta0[T]={-0.18633,-0.058,0.000,0.05558,0.1555};
double k0[T]={3.051,3.22,3.3977,3.567,3.737};
double d0[T]={0.66,0.706,0.7442,0.78,0.82};
double delta=0.1;
double beta=0.9615;
double rho=0.42;
double sigma=0.0118;
double mu=0.0;
double alpha=0.32;
double fi3=0.00074;
double fi=0.028;
double dbar=0.74421765717098;
double tau=1.455;
double gama=2;
double r=0.04;
static unsigned char first=1;
FILE *fp, *fopen();

/*----- Generation of technology(t) -----*/

if (first==1)
{
```

```

fp=fopen("pac", "w"); // open pac.txt
srand(151982);

for (i = 0; i < T; i++)
{
    ltheta[0] = ltheta0[i];
    theta[i][0] = exp(ltheta[0]);
    fprintf(fp, "%7.4f ", theta[i][0]);
    for (t = 1; t < TMAX; t++)
    {
        epsilon[t-1]=gaussrand()*sigma;
        ltheta[t] = rho*ltheta[t-1] + epsilon[t-1];
        theta[i][t] = exp(ltheta[t]);
        fprintf(fp, "%7.4f ", theta[i][t]);
    }
    fprintf(fp, "\n");
}
    fclose(fp);
    first=0;
}
/*-----*/

for (i = 1; i < 4; i++)
{
    for (t = 0; t < TMAX-1 ; t++)
    {
        if (t == 0)
        {
            k[t]=k0[i];
            d[t]=d0[i];
            coef[t]=1.0;

            // getting investment through network weights

            s1=vect[0]+vect[1]*k[t];
            s2=vect[2]+vect[3]*theta[i][t];
            s3=vect[4]+vect[5]*d[t];
            s4=1/(1+exp(-s1));
            s5=1/(1+exp(-s2));
            s6=1/(1+exp(-s3));
            s7=vect[6]+vect[7]*s4+vect[8]*s5+vect[9]*s6;
            s8=1/(1+exp(-s7));
            j[t]=(vect[10]-vect[11])*s8+vect[11];
            k[t+1]=j[t]+(1-delta)*k[t];
            if( k[t+1]< 0)
            {
                ans+/-10000000000;
                k[t+1]=0;
            }

            // getting debt through network weights

            s9=vect[12]+vect[13]*k[t];
            s10=vect[14]+vect[15]*theta[i][t];
            s11=vect[16]+vect[17]*d[t];
            s12=1/(1+exp(-s9));
            s13=1/(1+exp(-s10));
            s14=1/(1+exp(-s11));

```

```

s15=vect[18]+vect[19]*s12+vect[20]*s13+vect[21]*s14;
s16=1/(1+exp(-s15));
d[t+1]=(vect[22]-vect[23])*s16+vect[23];
tb[t]=d[t]*(1+r)-d[t+1];

// getting labor through network weights

s17=vect[24]+vect[25]*k[t];
s18=vect[26]+vect[27]*theta[i][t];
s19=1/(1+exp(-s17));
s20=1/(1+exp(-s18));
s21=vect[28]+vect[29]*s19+vect[30]*s20;
s22=1/(1+exp(-s21));
h[t]=(vect[31]-vect[32])*s22+vect[32];
if( h[t]< 0)
{
    ans+=-10000000000;
    h[t]=0;
}
y[t]=theta[i][t]*pow(k[t],alpha)*pow(h[t],(1-alpha));
c[t]=y[t]-j[t]-tb[t]-fi/2*(k[t+1]-k[t])*(k[t+1]-k[t])-
fi3/2*(d[t+1]-dbar)*(d[t+1]-dbar);

    if (c[t] <= (1/tau)*pow(h[t],tau))
        ans+=-10000000000;
    else
ans+=coef[t]*(1-(1/(c[t]-(1/tau)*pow(h[t],tau))));
}

else // for t from [1,TMAX-2]
{

    coef[t]=coef[t-1]*beta;

// getting investment through network weights

s1=vect[0]+vect[1]*k[t];
s2=vect[2]+vect[3]*theta[i][t];
s3=vect[4]+vect[5]*d[t];
s4=1/(1+exp(-s1));
s5=1/(1+exp(-s2));
s6=1/(1+exp(-s3));
s7=vect[6]+vect[7]*s4+vect[8]*s5+vect[9]*s6;
s8=1/(1+exp(-s7));
j[t]=(vect[10]-vect[11])*s8+vect[11];
k[t+1]=j[t]+(1-delta)*k[t];
if( k[t+1]< 0)
{
    ans+=-10000000000;
    k[t+1]=0;
}

// getting debt through network weights

s9=vect[12]+vect[13]*k[t];
s10=vect[14]+vect[15]*theta[i][t];
s11=vect[16]+vect[17]*d[t];

```

```

s12=1/(1+exp(-s9));
s13=1/(1+exp(-s10));
s14=1/(1+exp(-s11));
s15=vect[18]+vect[19]*s12+vect[20]*s13+vect[21]*s14;
s16=1/(1+exp(-s15));
d[t+1]=(vect[22]-vect[23])*s16+vect[23];
tb[t]=d[t]*(1+r)-d[t+1];

// getting labor through network weights

s17=vect[24]+vect[25]*k[t];
s18=vect[26]+vect[27]*theta[i][t];
s19=1/(1+exp(-s17));
s20=1/(1+exp(-s18));
s21=vect[28]+vect[29]*s19+vect[30]*s20;
s22=1/(1+exp(-s21));
h[t]=(vect[31]-vect[32])*s22+vect[32];
if( h[t]< 0)
{
    ans+=-10000000000;
    h[t]=0;
}
y[t]=theta[i][t]*pow(k[t],alpha)*pow(h[t],(1-alpha));
c[t]=y[t]-j[t]-tb[t]-fi/2*(k[t+1]-k[t])*(k[t+1]-k[t])-
fi3/2*(d[t+1]-dbar)*(d[t+1]-dbar);

if (c[t] <= (1/tau)*pow(h[t],tau))
ans+=-10000000000;
else
ans+=coef[t]*(1-(1/(c[t]-(1/tau)*pow(h[t],tau)))));
}
}

}
return(ans);
}

```

### 3. Technology Generation for Unit Technology Shock, $\sigma=0$

```
if (first==1)
{
    fp=fopen("unitshock", "w"); // open deirdata.txt
    srand(151982);

    for (i = 0; i < T; i++)
    {
        ltheta[0] = ltheta0[i];
        theta[i][0] = exp(ltheta[0]);
        fprintf(fp, "%7.4f ", theta[i][0]);
        epsilon[0]=1;
        ltheta[1] = rho*ltheta[0] + epsilon[0];
        theta[i][1] = exp(ltheta[1]);
        fprintf(fp, "%7.4f ", theta[i][1]);
        for (t = 2; t < TMAX; t++)
        {
            epsilon[t-1]=gaussrand()*sigma;
            ltheta[t] = rho*ltheta[t-1] + epsilon[t-1];
            theta[i][t] = exp(ltheta[t]);
            fprintf(fp, "%7.4f ", theta[i][t]);
        }
        fprintf(fp, "\n");
    }
    fclose(fp);
    first=0;
}
```

## APPENDIX B. APPROXIMATED PATHS

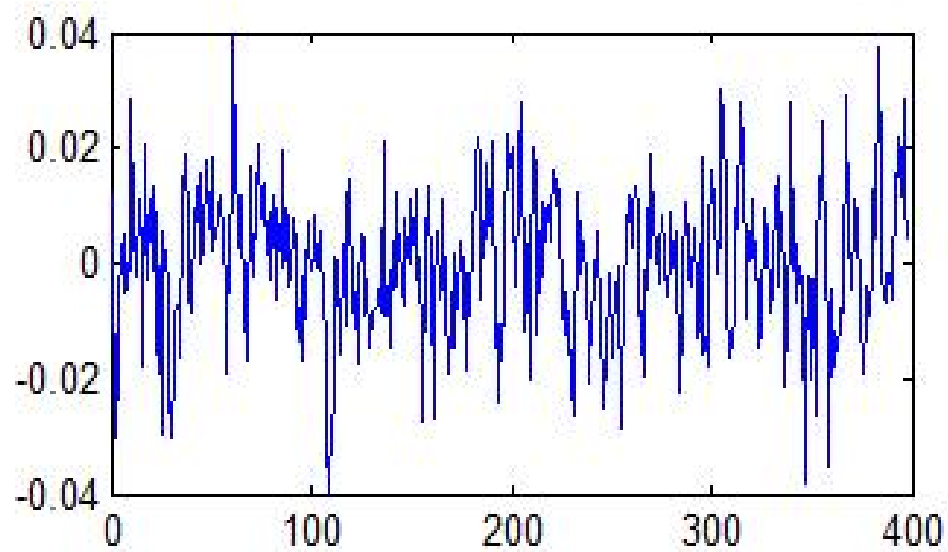


Figure 11: Technology shocks, in logarithms for open economy with debt elastic interest rate premium

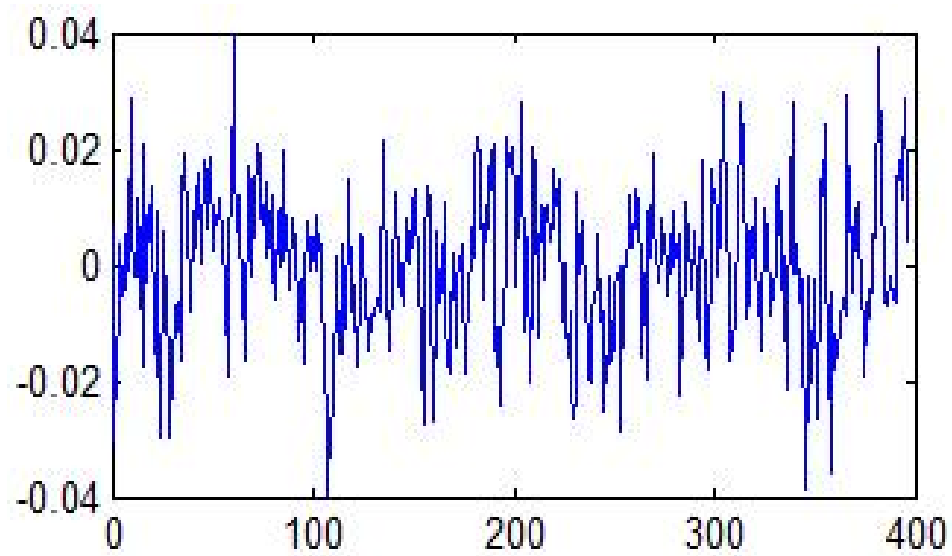


Figure 12: Technology shocks, in logarithms for open economy with portfolio adjustment cost

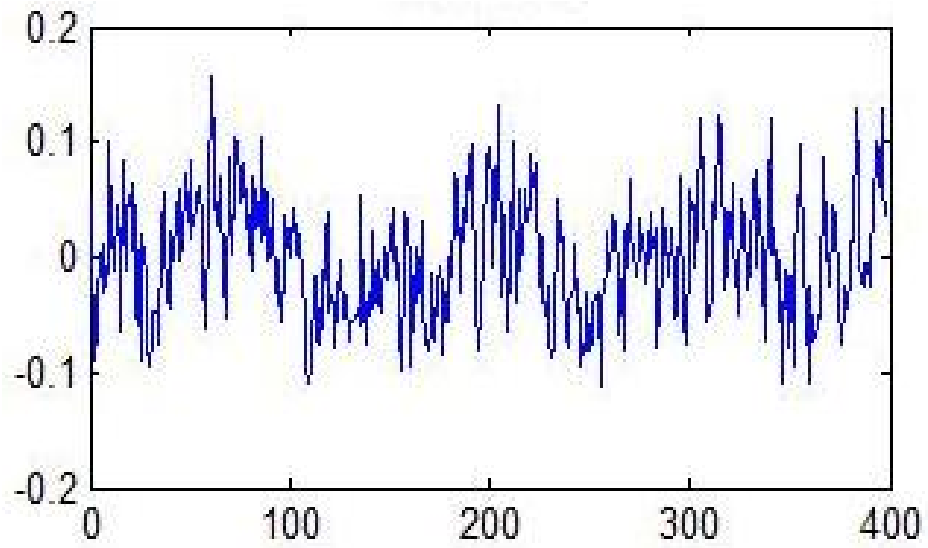


Figure 13: Investment path, in logarithms for open economy with debt elastic interest rate premium

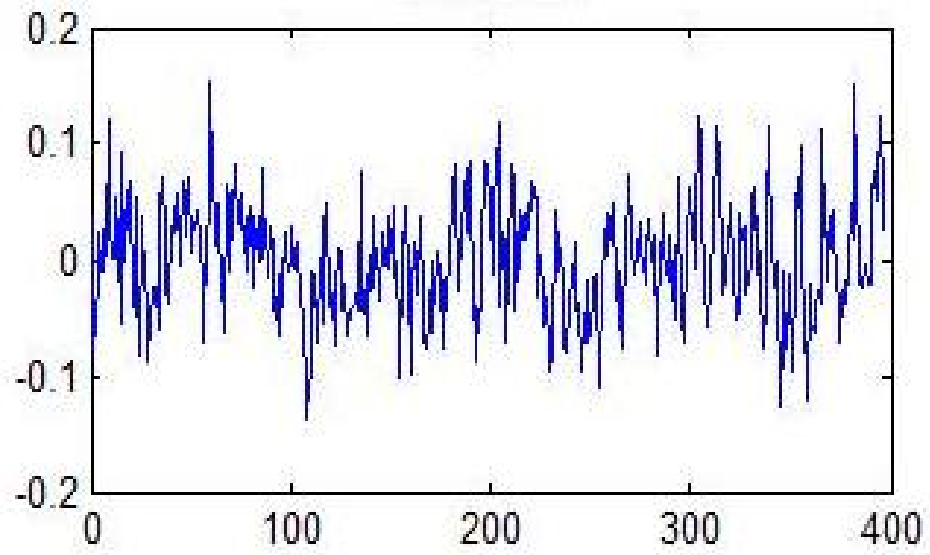


Figure 14: Investment path, in logarithms for open economy with portfolio adjustment cost

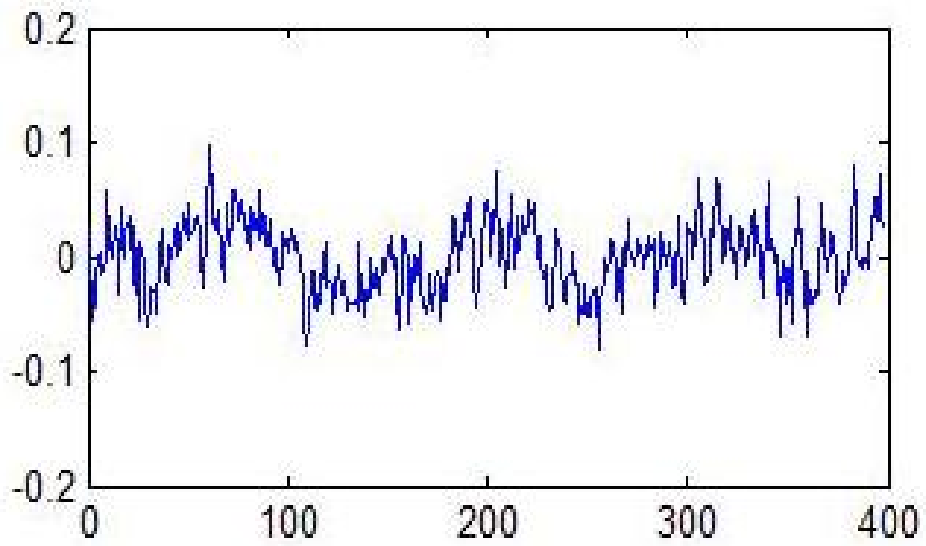


Figure 15: Output path, in logarithms for open economy with debt elastic interest rate premium

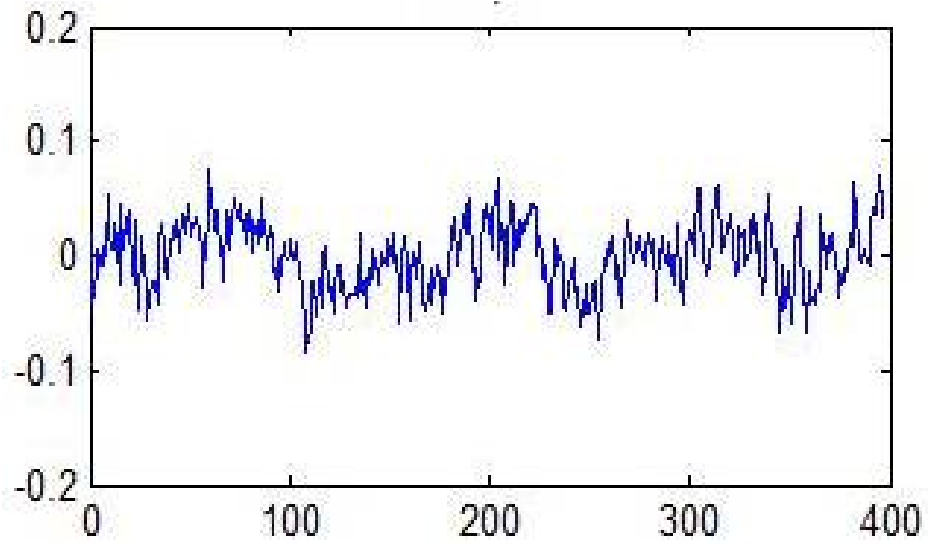


Figure 16: Output path, in logarithms for open economy with portfolio adjustment cost

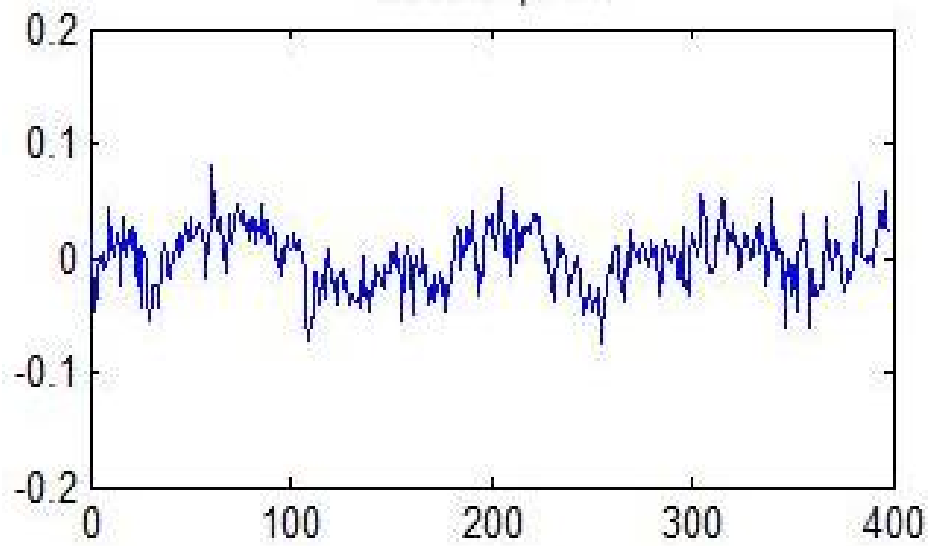


Figure 17: Consumption path, in logarithms for open economy with debt elastic interest rate premium

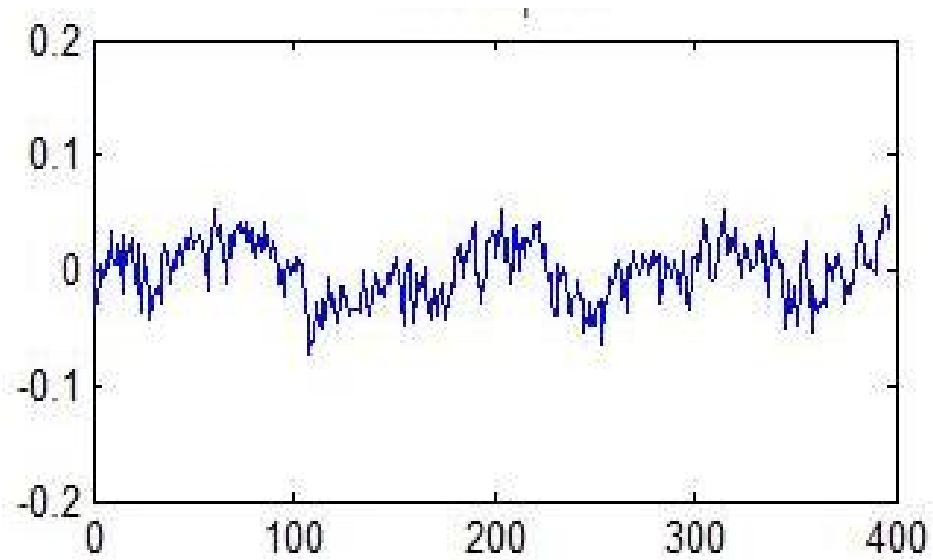


Figure 18: Consumption path, in logarithms for open economy with portfolio adjustment cost

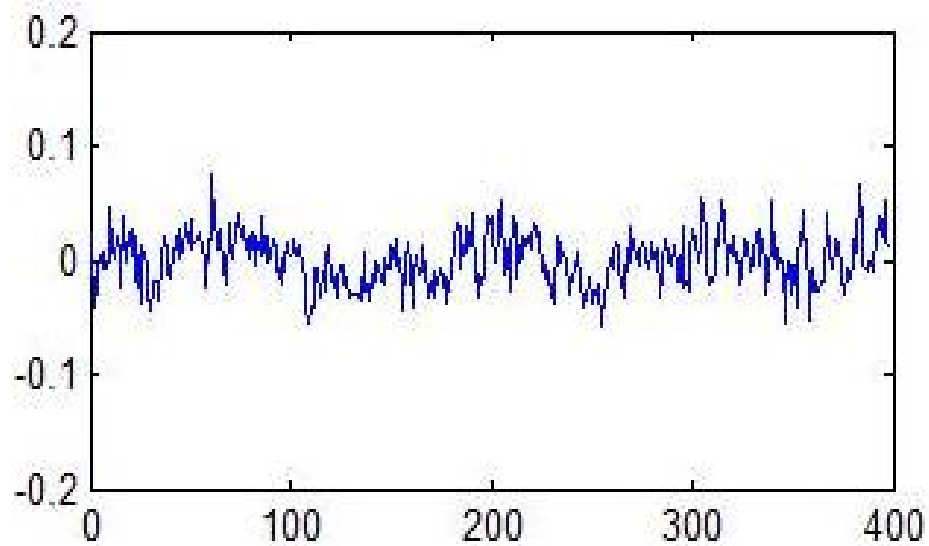


Figure 19: Hours path, in logarithms for open economy with debt elastic interest rate premium

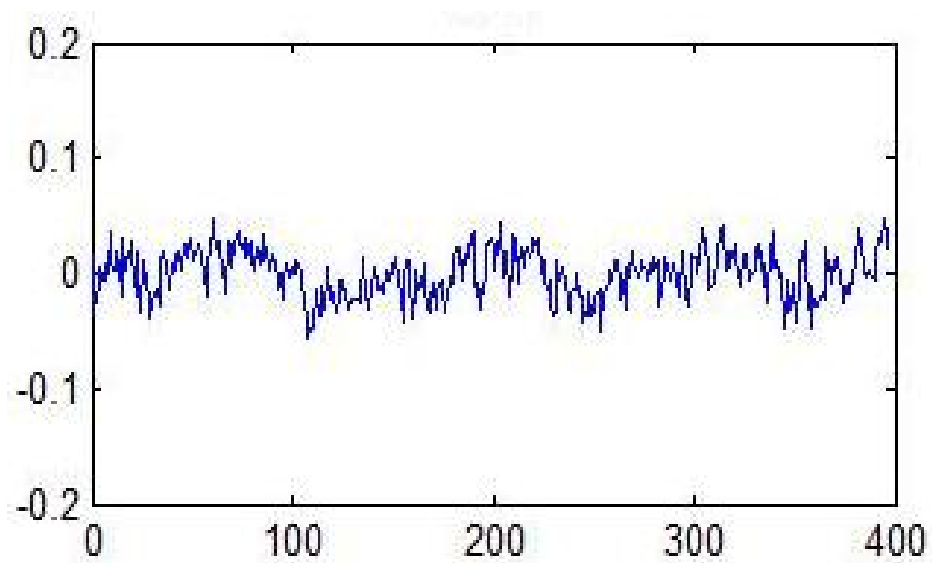


Figure 20: Hours path, in logarithms for open economy with portfolio adjustment cost

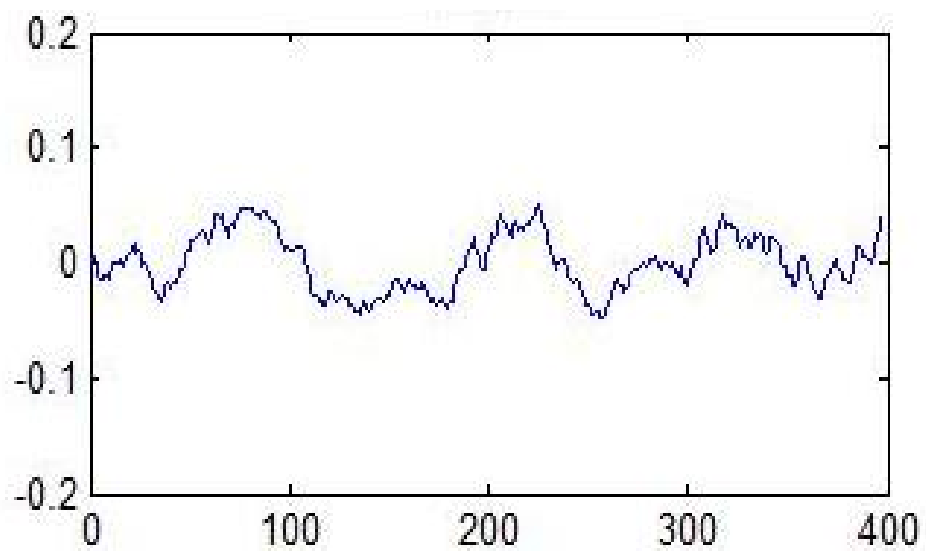


Figure 21: Capital stock path, in logarithms for open economy with debt elastic interest rate premium

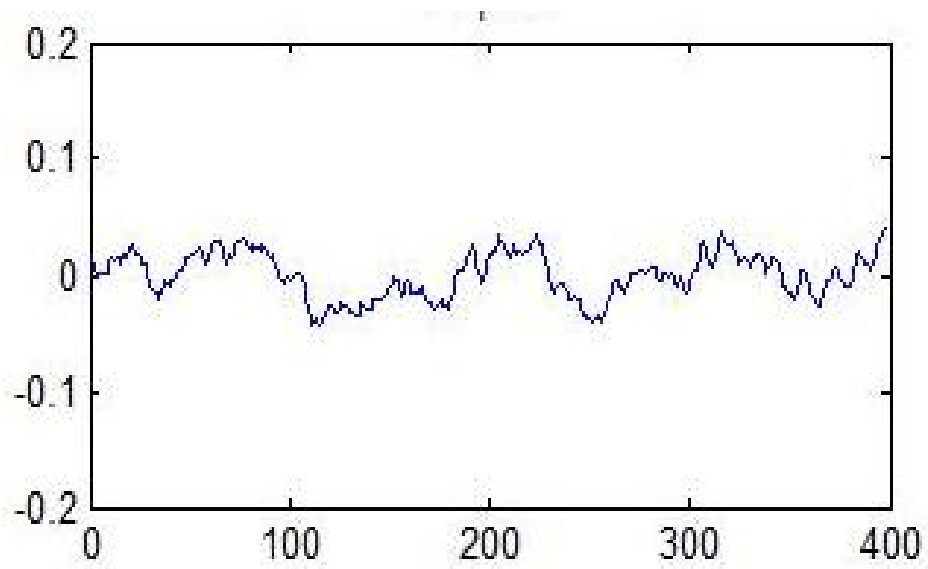


Figure 22: Capital stock path, in logarithms for open economy with portfolio adjustment cost

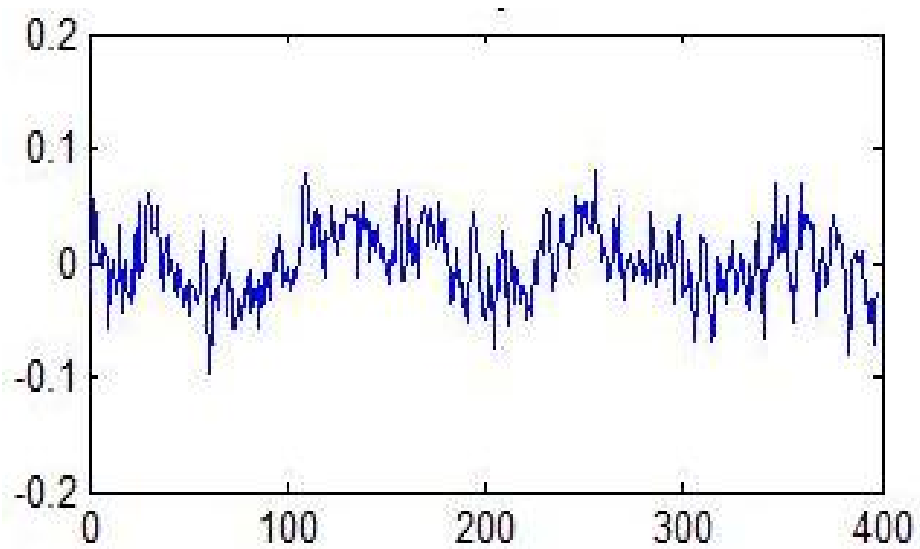


Figure 23: Trade balance/GDP path, in logarithms for open economy with debt elastic interest rate premium

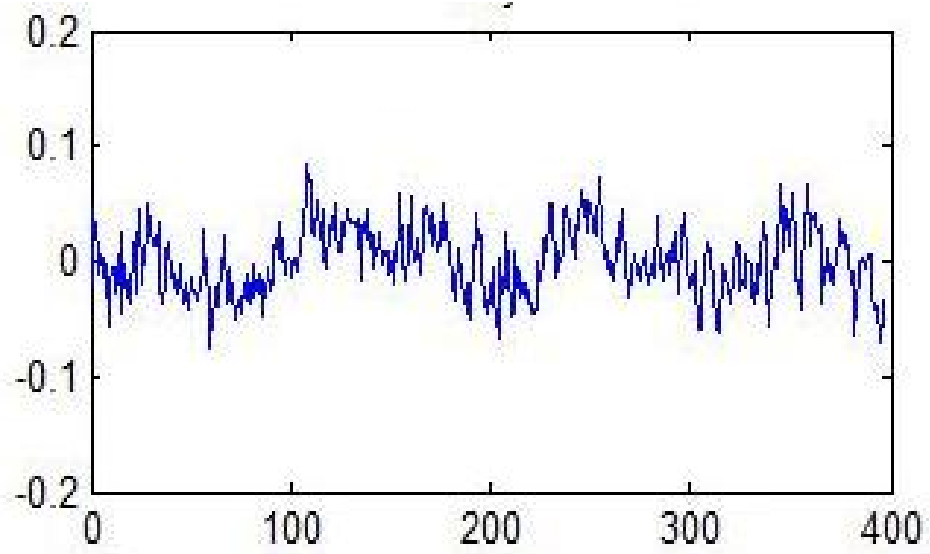


Figure 24: Trade balance/GDP path, in logarithms for open economy with portfolio adjustment cost

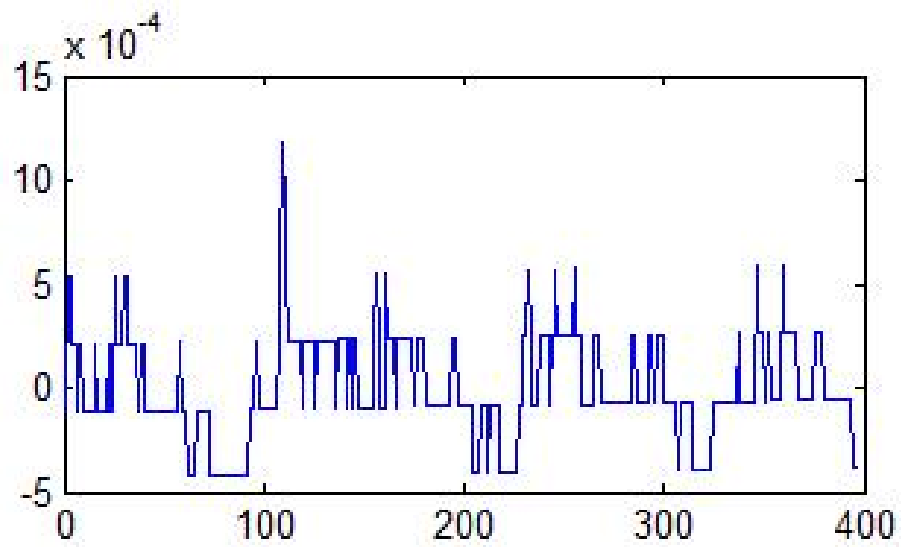


Figure 25: Foreign interest payment path, in logarithms for open economy with debt elastic interest rate premium

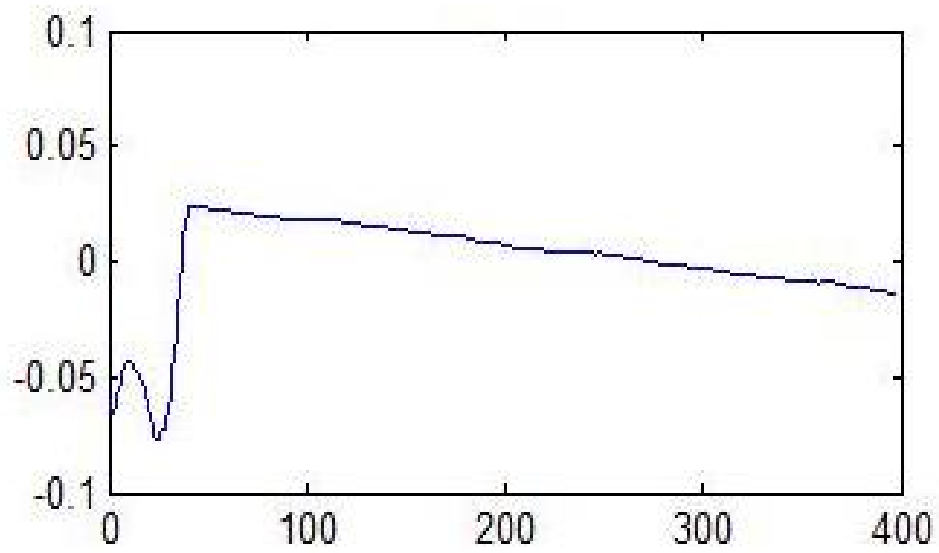


Figure 26: Foreign interest payment path, in logarithms for open economy with portfolio adjustment cost