Modeling Genetic Regulatory Networks by Sigmoidal Functions: A Joint Genetic Algorithm and Kalman Filtering Approach

Haixin Wang and Lijun Qian
Department of Electrical and Computer Engineering
Prairie View A&M University
Prairie View, Texas 77446
Email: HWang, LiQian@pvamu.edu

Edward Dougherty
Computational Biology Division
Translational Genomics Research Institute (TGen)
Phoenix, AZ 85004
Email: edward@ee.tamu.edu

Abstract— **In this paper, the problem of genetic regulatory network inference from time series microarray experiment data is considered. A noisy sigmoidal model is proposed to include both system noise and measurement noise. In order to solve this nonlinear identification problem (with noise), a joint genetic algorithm and Kalman filtering approach is proposed. Genetic algorithm is applied to minimize the fitness function and Kalman filter is employed to estimate the weight parameters in each iteration. The effectiveness of the proposed method is demonstrated by using both synthetic data and microarray measurements.**

I. Introduction

A genetic regulatory network (GRN) is a collection of DNA segments in a cell which interact with each other and with other substances in the cell, thereby governing the gene transcriptions. In light of the recent development of high-throughput DNA microarray technology, it becomes possible to discover GRNs, which are complex and nonlinear in nature. Specifically, the increasing existence of microarray time-series data makes possible the characterization of *dynamic* nonlinear regulatory interactions among genes. The modeling, analysis and control of GRNs are critical for finding medicine for gene-related diseases. Figure 1 shows the genomic signal processing process and its relation to drug development such as gene therapy.

Because GRN models are difficult to deduce solely by means of experimental techniques, computational and mathematical methods are indispensable. Much research has been done on GRN modeling by *linear* differential/difference equations using time-series data, for example, [20], [16], [19], [18], [17], [1], [21], [22], just to name a few. The basic idea is to approximate the combined effects of different genes by means of a weighted sum of their expression levels. In [20], a connectionist model is used to model small gene networks operating in the blastoderm of Drosophila. In [16], the concentrations of mRNA and protein are modeled by linear differential equations. A simple form of linear additive functions is suggested by [17], where $dx_i/dt = \sum_{j=1}^{n} w_{ij}x_j$. The degradation rate of gene i's mRNA and environmental effects are assumed to be incorporated in the parameters w_{ij} and their influence on gene i's expression level x_i is assumed to be linear. A method to obtain a continuous linear differential equation model from sampled time-series data is proposed in [21]. For added biological realism (all concentrations get saturated at some point in time), a sigmoid (squashing) function may be included into the equation. It has been show that this sort of quasi-linear model can be solved by first applying the inverse of the squashing function [18], [9].

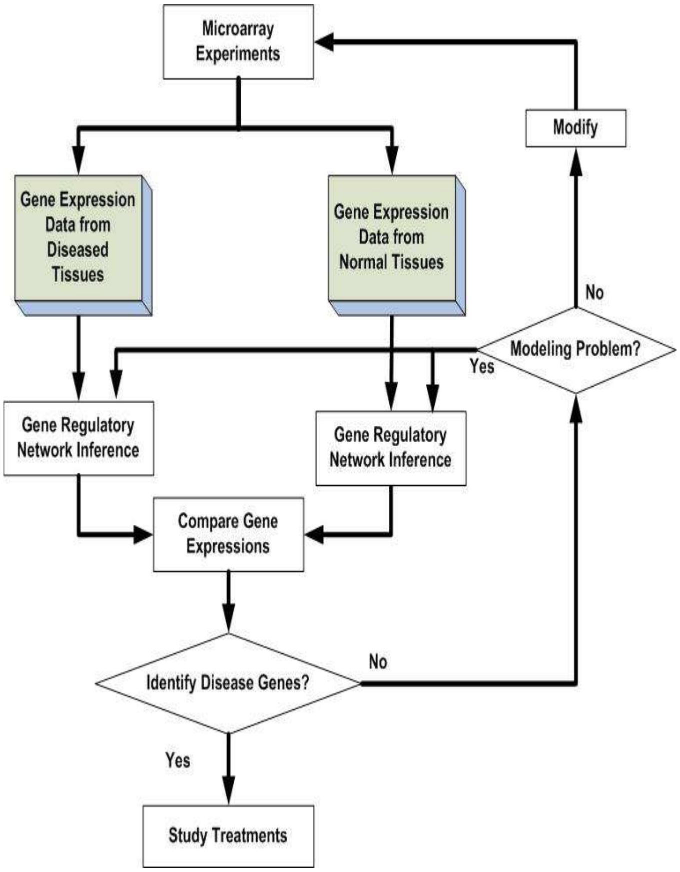
Microarray Experiments
Modify
Gene Expression Data from Diseased Tissues
Gene Expression Data from Normal Tissues
No
Modeling Problem?
Yes
Gene Regulatory Network Inference
Gene Regulatory Network Inference
Compare Gene Expressions
Identify Disease Genes?
No
Yes
Study Treatments

Fig. 1. The flowchart for genomic signal processing and drug discovery

The sigmoidal function can be defined as

$$f(x) = \frac{1}{1+e^{-x}}. \quad (1)$$

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

It may be used to model nonlinear systems that exhibit "saturation" phenomina. Sigmoidal function model has been applied in many studies such as cell control processes [2], wind speed prediction [3] and even weakly singular and near-singular boundary element integrals problems [6]. Sigmoidal function model has also been proposed for modeling GRNs. In [18], the author applied a sigmoidal model using weight matrices. In [2], an artificial neural network is used to model gene expressions, which is based on sigmoidal functions. However, the number of the parameters is too large to be computed correctly using limited time series gene expression data. In [8], sigmoidal function is applied together with simulated annealing for modeling GRNs. Clustering is applied first to make sure the whole process can be applied to large-scale gene expression data.

In this paper, we model GRNs using sigmoidal functions with *noise*. Both system noise and measurement noise are included in the model explicitly. Assuming there are N genes of interest and x_i denotes the state (such as the microarray reading) of the i^{th} gene. Then the dynamics of the GRN may be modeled as

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \cdots, x_N) \quad i = 1, 2, \cdots, N. \tag{2}$$

In this study we assume the functions f_i ($\forall i$) are in the form

$$f_i = \frac{C_{i,1}}{1 + e^{-(\mathbf{r}_i + \beta_i) + \nu_i}} - C_{i,2} x_i, \tag{3}$$

where

$$\mathbf{r}_i = \sum_{j=1}^{n} ((w_{ij} + \mu_{ij}) x_j + v_{ij} u_j). \tag{4}$$

$C_{i,1}$ and $C_{i,2} > 0$ are two parameters. w_{ij} is the weight value for gene j on gene i. β_i is an offset parameter. u_i is an input and v_{ij} is the weight value for input j on gene i. μ_{ij} and ν_i are intrinsic noise and external noise, respectively. In this study, it is assumed that there is no additional input other than the genes of interests, i.e., $u_i = 0$, $\forall i$. Then the GRN model using sigmoidal functions can be rewritten as

$$\frac{dx_i}{dt} = \frac{C_{i,1}}{1 + e^{-\{\sum_{j=1}^{n} [(w_{ij} + \mu_{ij}) x_j] + \beta_i\} + \nu_i}} - C_{i,2} x_i \qquad i = 1, 2, \cdots, N. \tag{5}$$

The proposed model includes all the major characteristics of a gene regulatory network: it is nonlinear, dynamic, and noisy. To the best of our knowledge, no previous work has used the same model. The noisy nature of GRNs is modeled explicitly. The deterministic model (without noise) corresponds to the nominal case, while the various stochastic effects are included as noise disturbances. The inherent stochasticity of biochemical processes (transcription and translation) is modeled as noise in the parameters (μ_{ij}), which corresponds to the "intrinsic noise" mentioned in the literature [4]. Other effects, such as those from genes not been included in the microarray, the amount of RNA polymerase, levels of regulatory proteins, and the effects of mRNA and protein degradation, are modeled by the external noise (ν_i) [4]. Previous work has modeled these noise types by Gaussian white noise processes [5]. The inclusion of noise also enables the proposed model to provide interpretation of the fact that GRNs are robust to noise, by which it is meant that the relationships among genes are not greatly affected by small changes caused by noise.

The parameters (w_{ij}, β_i, $C_{i,1}$ and $C_{i,2}$) need to be identified from time-series microarray measurements such that the identification error is minimized and the simplest model structure is selected. In this paper, the criteria of selecting the parameters are represented by a fitness function and modeling a GRN becomes a nonlinear optimization problem (minimization of fitness functions). We provide a framework to infer the proposed nonlinear model with noise using time-series data, where Genetic Algorithm and Kalman filtering are applied. Both synthetic data and experimental data from microarray measurements are used to evaluate the proposed method.

The remainder of the paper is organized as follows: The proposed framework and the iterative algorithm are illustrated in Section II. Simulation results are given in Section III. Section IV contains some concluding remarks.

II. Joint Genetic Algorithm and Kalman Filtering

The task of identifying GRNs may be considered as an optimization problem. The goal is to minimize the identification error and keep the model as simple as possible, which may be achieved by minimizing the following fitness function

$$\mathrm{fitness} = \sum_{\mathrm{i}=1}^{\mathrm{N}} [\sum_{\mathrm{k}=1}^{\mathrm{M}} (\mathrm{x_i(k)} - \mathrm{x_i^{tar}(k)})^2] \tag{6}$$

where M is the number of data points, x_i^{tar} is the target time series and x_i is the obtained time series given by the obtained sigmoidal function model.

Since it is a global nonlinear optimization problem, a nested optimization structure is adopted, where genetic algorithm is applied to determine the terms (global optimization) while Kalman filter is employed to estimate the corresponding parameters (local optimization) in each iteration. Such a decomposition of the problem into a structural part solved by genetic algorithm and a parameter optimization part solved by Kalman filtering reduces the complexity significantly and speeds up convergence. The detailed procedures of the proposed iterative algorithm is illustrated in Fig. 2. The genetic algorithm has four operations: reproduction, crossover, mutation and selection. Kalman filtering is employed to estimate the parameters for every generation.

A. Genetic Algorithm

Genetic algorithm can be described by a function $GA(f, \gamma, \delta, \zeta, \xi, \lambda)$ [14]. The general parameters in genetic algorithm are defined as follows.

- ϕ : the initial population randomly generated by computer program
- f: the fitness function for each individual
- γ: the fitness threshold to terminate the loops

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

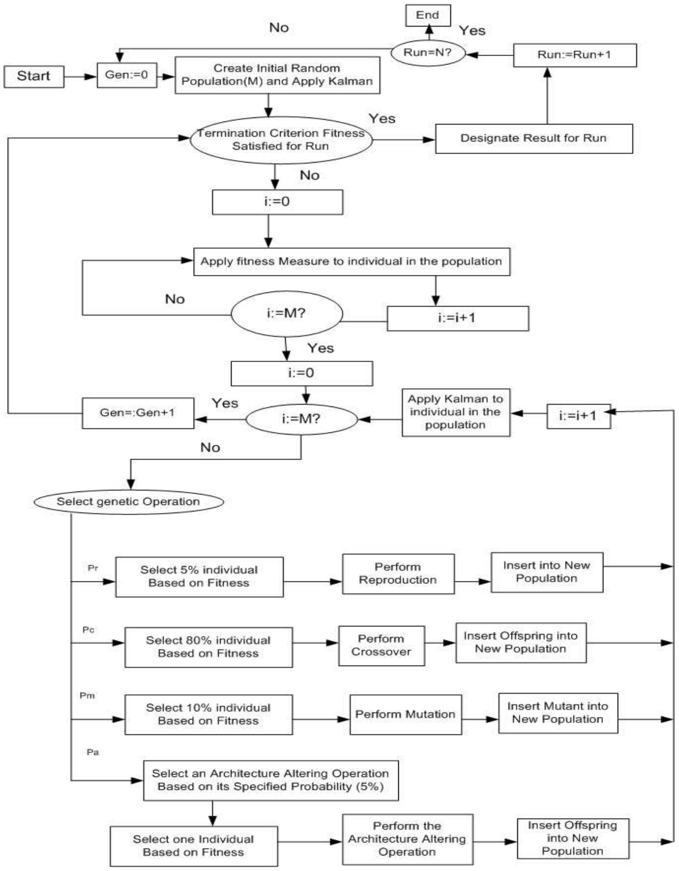
End
No
Yes
Run=N?
Run:=Run+1
Start
Gen:=0
Create Initial Random Population(M) and Apply Kalman
Yes
Termination Criterion Fitness Satisfied for Run
Designate Result for Run
No
i:=0
Apply fitness Measure to individual in the population
No
i:=M?
i:=i+1
Yes
i:=0
Yes
Gen=:Gen+1
i:=M?
Apply Kalman to individual in the population
i:=i+1
No
Select genetic Operation
Pr
Select 5% individual Based on Fitness
Perform Reproduction
Insert into New Population
Pc
Select 80% individual Based on Fitness
Perform Crossover
Insert Offspring into New Population
Pm
Select 10% individual Based on Fitness
Perform Mutation
Insert Mutant into New Population
Pa
Select an Architecture Altering Operation Based on its Specified Probability (5%)
Select one Individual Based on Fitness
Perform the Architecture Altering Operation
Insert Offspring into New Population

Fig. 2. The parameters search by genetic programming and RLS/Kalman filter estimation

- δ: the size of ϕ
- ζ: the crossover factor of ϕ in each generation
- ξ: the mutation rate
- λ: alternate termination threshold

The goal of the whole process is to find the generation that minimize the fitness function and get the minimize value of $GA(f, \gamma, \delta, \zeta, \xi, \lambda)$.

B. Kalman Filter

The sigmoidal model, equation (5), can be further rewritten as

$$\dot{x_i} + C_{i,2}x_i = \frac{C_{i,1}}{1 + e^{-(\mathbf{r}_i + \beta_i) + \nu_i}}. \tag{7}$$

In order to apply a linear estimator, the above equation can be re-arranged as

$$\mathbf{r}_i = -ln(\frac{C_{i,1}}{\dot{x}_i(t) + C_{i,2}x_i(t)} - 1) - \beta_i + \nu_i. \tag{8}$$

Equation (8) is now *linear* in parameters $w_i \triangleq [w_{i1} \quad w_{i2} \quad \cdots w_{iN}]^T$. Then a linear estimator may be applied to estimate w_i in each iteration. When noise is not considered, a Recursive Least Square (RLS) estimator may be used. If noise is modeled by Gaussian white noise processes [5] with known statistics, Kalman filter may be used to get the optimal estimate of w_i.

Kalman filter offers optimal filtering when the linear system model parameters and power spectral density of the noise are known [11]. The corresponding state and measurement equations are

$$w_i(n) = w_i(n-1) + \mu_i(n-1) \tag{9}$$
$$d_i(n) = x_i(n)w_i(n) + \nu_i(n) \tag{10}$$

where $\mu_i \triangleq [\mu_{i1} \ \mu_{i2} \ \cdots \mu_{iN}]^T$ is the process noise of gene i. Its covariance matrix is

$$E[\mu_i(n)\mu_i^T(k)] = \begin{cases} Q_i(n), & n = k \\ 0, & n \neq k \end{cases}$$

ν_i is the measurement noise. In the sigmoidal model, it is noise from the environment. Its covariance matrix is

$$E[\nu_i(n)\nu_i(n)^T(k)] = \begin{cases} R_i(n), & n = k \\ 0, & n \neq k \end{cases}$$

The noise vectors μ_i and ν_i are statistically independent.

The implementation of the Kalman filter model is given by serial of computation from the following set of expressions:

$$\hat{w}^-(n) = \hat{w}^+(n-1) \tag{11}$$
$$P^-(n) = P^+(n-1) + Q(n-1) \tag{12}$$
$$\hat{w}^+(n) = \hat{w}^-(n) + K(n)[d(n) - C(n)\hat{w}^-(n)] \tag{13}$$
$$K(n) = P^-(n)C^T(n)[C(n)P^-(n)C^T(n) + R(n)]^{-1} \tag{14}$$
$$P^+(n) = P^-(n) - K(n)C(n)P^-(n) \tag{15}$$

where $K(n)$ is the Kalman filter gain and P is the covariance matrix of the error. The superscripts $^-$ and $^+$ indicate the *a priori* and *a posteriori* values of the variables, respectively. $\hat{w}^-$ and $\hat{w}^+$ are the prior and posterior estimates, respectively. Q and R are the covariance matrices of the parameter noise and external noise, respectively. The initial conditions are $\hat{w}(0 \mid d_0) = E[\hat{w}(0)]$ and $P_0 = E[w(0)w^T(0)]$.

III. Simulation Results

A. Synthetic Data

In this part of the simulation, a synthetic sigmoidal function model is assumed. Multi-variable Runge-Kutta algorithm is used to generate the time-series data. Both genetic algorithm plus RLS and genetic algorithm plus Kalman filter are applied to infer the GRN.

The original synthetic sigmoidal function model without noise is given by

$$\begin{aligned}
\dot{x_1} &= \frac{0.1}{1 + e^{-(-20x_1 - 5x_2)}} - 0.1x_1 \\
\dot{x_2} &= \frac{0.1}{1 + e^{-(-25x_1 + 5x_2 + 17x_3)}} - 0.1x_2 \\
\dot{x_3} &= \frac{0.1}{1 + e^{-(-10x_2 - 20x_3 + 20x_4)}} - 0.1x_3 \\
\dot{x_4} &= \frac{0.1}{1 + e^{-(10x_3 - 5x_4)}} - 0.1x_4
\end{aligned}$$

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

Multi-variable Runge-Kutta Algorithm Runge-Kutta algorithm is the numerical solution to differential equation system. Even for multi-variable differential system it is very accurate and efficient. Suppose there are n variables with n differential equations, the synthetic time-series data can be obtained by [10]

$$\begin{aligned} a_{j,i}^1 &= f_j(x_{1,i}, x_{2,i}, ...x_{n,i}) \quad j = 1...n \\ a_{j,i}^2 &= f_j(x_{1,i} + a_{1,i}^1 \frac{h}{2}, x_{2,i} + a_{2,i}^1 \frac{h}{2}, ..., x_{n,i} + a_{n,i}^1 \frac{h}{2}) \\ a_{j,i}^3 &= f_j(x_{1,i} + a_{1,i}^2 \frac{h}{2}, x_{2,i} + a_{2,i}^2 \frac{h}{2}, ..., x_{n,i} + a_{n,i}^2 \frac{h}{2}) \\ a_{j,i}^4 &= f_j(x_{1,i} + a_{1,i}^3 h, x_{2,i} + a_{2,i}^3 h, ..., x_{n,i} + a_{n,i}^3 h) \\ x_{j,i+1} &= x_{j,i} + (a_{j,i}^1 + 2a_{j,i}^2 + 2a_{j,i}^3 + a_{j,i}^4)\frac{h}{6} \end{aligned}$$

where $x_{j,i}$ is the state of the j^{th} gene at the time instant i, $j = 1, 2, ..., n$. h is the step size. We do not include the noise terms in the above data generation equations for simplicity of presentation.

RLS algorithm The RLS algorithm is given as follows [11].

$$\begin{aligned} K(n) &= \frac{\lambda^{-1}P(n-1)u(n)}{1 + \lambda^{-1}U^T(n)P(n-1)u(n)} \\ e(n) &= d(n) - \hat{w}^T(n-1)u(n) \\ \hat{w}(n) &= \hat{w}(n-1) + k(n)e(n) \\ P(n) &= \lambda^{-1}P(n-1) - \lambda^{-1}k(n)u^T(n)P(n-1) \end{aligned}$$

For GRN inference, the input $u(n)$ will be the time series data $x_i(n)$. The initial condition of $P(0) = \delta^{-1}\mathbf{I}$ and $w(0) = 0$. δ is a small constant value.

The trajectory of gene expression levels are shown in Fig. 3. We observe that under noisy condition, genetic algorithm plus Kalman filter performs much better than genetic algorithm plus RLS, as expected. Because the noise level of microarray data is usually non-negligible, Kalman filter is a better choice than RLS in the proposed method for GRN inference.

B. Microarray Measurements

We consider time-series gene-expression data corresponding to yeast protein synthesis. Here, the data for five genes (HAP1(x_1), CYB2(x_2), CYC7(x_3), CYT1(x_4), COX5A(x_5)) are picked because the relations among them have been revealed by biological experiments. For example, HAP1 represses the nuclear encoding cytochrome gene CYC7 under the anaerobic condition; CYB2 activates CYC7. The branch pathway model shown in Fig. 4 is obtained by the proposed GA plus Kalman filter method using the time-series data from [15]. The trajectory is also shown in Fig.5. We observe that HAP1 represses CYC7, and CYB2 activates CYC7. It is also observed that HAP1 activates COX5A and CYT1. These observations are in agreement with the biological experiment findings in [12], [13].

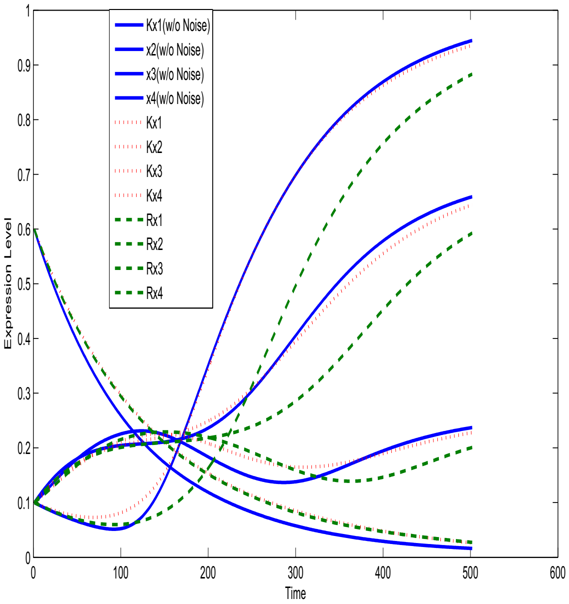

Fig. 3. Trajectory of gene expression levels. $x1, x2, x3, x4$ are the data without noise. $Kx1, Kx2, Kx3, Kx4$ are results from GA plus Kalman filter. $Rx1, Rx2, Rx3, Rx4$ are results from GA plus RLS.

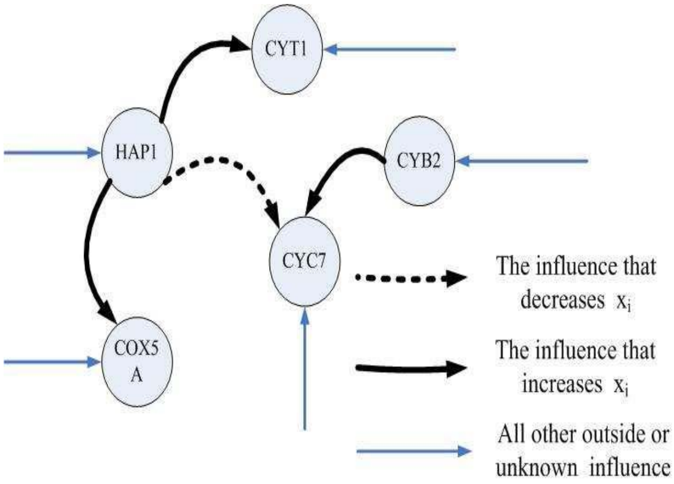
CYT1
HAP1
CYB2
CYC7
COX5
A
The influence that decreases x_i
The influence that increases x_i
All other outside or unknown influence

Fig. 4. The branch pathway model of the 5 genes in yeast.

IV. Conclusions

In this paper, the problem of genetic regulatory network inference from time series data is considered. A noisy sigmoidal function based model is proposed to include both intrinsic noise and external noise. In order to solve this nonlinear identification problem, a genetic algorithm plus Kalman filtering approach is proposed. Genetic algorithm is applied to minimize the fitness function and Kalman filter is employed to estimate the weight parameters in each iteration. The effectiveness of the proposed method is demonstrated by using both synthetic data and real microarray measurements. We also show that genetic algorithm plus Kalman filtering performs

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

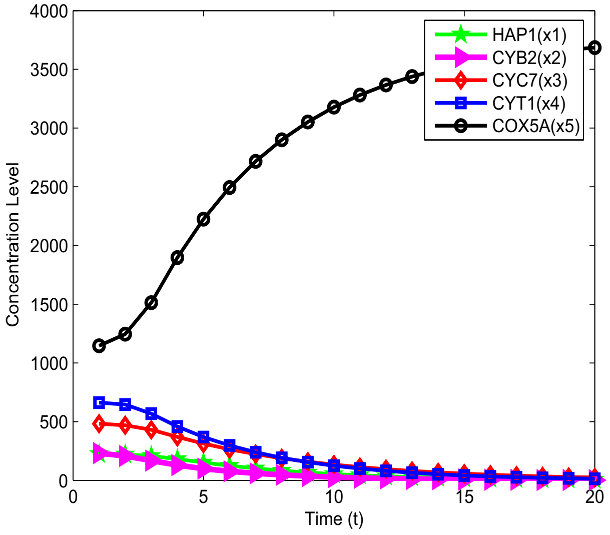

Fig. 5. The simulation result from microarray data

much better than genetic algorithm plus RLS when the data is noisy, which is the case in microarray measurements. The convergence and steady state of sigmoidal function based models will be investigated in our future work.

References

[1] H.de Jong, "Modeling and simulation of genetic regulatory systems: a literature review", *Journal of Computational Biology*, 9(1):67 - 103, 2002.

[2] Jiri Vogradsky " Neural Model of the Genetic Network", *The Journal of Biological chemistry*,Vol. 276 No. 39,pp. 36168-36173,2001.

[3] P. Flores, A. Tapia, and G. Tapia, "Application of a control algorithm for wind speed prediction and active power generation ", *Renewable Energy*,Vol 30 Issue 4,pp. 523-536, 2005.

[4] P. Swain, M. Elowitz, and E. Siggia, "Intrinsic and extrinsic contributions to stochasticity in gene expression", *Proc. Natl. Acad. Sci. USA*, 99:12795-12800, 2002.

[5] J. Hasty, J. Pradines, M. Dolnik, and J. J. Collins, "Noise-based switches and amplifiers for gene expression", *Proc. Natl. Acad. Sci. USA*, 97:2075-2080, 2000.

[6] Peter R. Johnston " Application of Sigmoidal Transformations to Weakly Singular and Near-singular Boundary Element Integrals", *International journal for Numberical Methods in Engineering*, Vol 45, pp. 1333-1348,1999.

[7] D.C Weaver " Modeling Regulatory Networks With Weight Matrices", *Pacific Symposium on Bimcomputing*,4:112-123, 1999.

[8] E.Mjolsness et al. "From Coexpression to Coregulation: An Approach to Inferring Transcriptional Regulation Among Gene Classes from Large-Scale Expression Data", *Advances in Neural Information Processing Systems*, 12:928-934,1999.

[9] L.F.A Wessels, E.P.Van Someren and M.J.T. Reinders, " A Comparison of Genetic Network Models", *pacific Symposium on Biocomputing* Vol 6 pp. 508-519,2001.

[10] W. Press, et. al, *Numerical Recipes in C*, 2nd. Ed, Cambridge University Press, 1992.

[11] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1992.

[12] P. Woolf and Y. Wang, "A fuzzy logic approach to analyzing gene expression data", *Physiol. Genomics*, 3: 9-15, 2000.

[13] J. Schneider and L. Guarente, "Regulation of the Yeast CYTI Gene Encoding Cytochrome cl by HAP1 and HAP2/3/4", *Molecular and Cellular Biology*, 11(10): 4934-4942, 1991.

[14] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT press, 1992.

[15] http://sgdlite.princeton.edu/download/yeast_datasets

[16] T. Chen, H.L. He, and G.M. Church, "Modeling gene expression with differential equations", *Pac. Symp. Biocomputing*, 4:29 - 40, 1999.

[17] M.K.S. Yeung, J. Tegnãr, and J.J. Collins, "Reverse engineering gene networks using singular value decomposition and robust regression", *Proc. Natl. Acad. Sci. USA*, 99:6163 - 6168, 2002.

[18] D.C. Weaver, C.T. Workman, G.D. Stormo, "Modeling regulatory networks with weight matrices", *Pac. Symp. Biocomputing*, 4: 112 - 123, 1999.

[19] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear Modeling of mRNA expression levels during CNS development and injury", *Pac. Symp. Biocomputing*, 4: 41 - 52, 1999.

[20] E. Mjolsness, D.H. Sharp, and J. Reinitz, "A connectionist model of development", *J Theor Biol.*, 152(4):429 - 53, Oct 1991.

[21] I. Tabus, C.D. Giurcaneanu, and J. Astola, "Genetic networks inferred from time series of gene expression data", *First International Symposium on Control, Communications and Signal Processing*, pp. 755 - 758, Hammamet, Tunisia, 2004.

[22] M.J.L. de Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano, "Inferring gene regulatory networks from time-ordered gene expression data of Bacillus subtilis using differential equations", *Pac. Symp. Biocomputing*, 8: 17 - 28, 2003.

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE