Biotechnological Processes Identification Using Dynamic Neural Networks

Marian BARBU, Sergiu CARAMAN, Emil CEANGA

"Dunarea de Jos University of Galatz,

Domneasca Street, No. 47, 800008 Galatz, ROMANIA

e-mail: {Marian.Barbu; Sergiu.Caraman; Emil.Ceanga}@ugal.ro

Phone/Fax: +40 236 460182

Abstract: The paper deals with the identification of a batch biotechnological process using dynamic neural networks. The process considered in the paper is the growth process of Candida lipolytica population on an ammonium sulfate substrate and its model includes a mean age equation. The neural network used for identification is trained at every hour, based on the experimental data from the process and the process parameters are given by the neural network weights determined at every training step. The mean age model has been validated based on the fact the parameters of the mean age equation are the same with the ones from the other model equations (biomass, substrate and product).

1 Introduction

The biotechnological processes are very complex and strong non-linear. That is why the traditional methods for modeling and control do not offer good results. The problems are more difficult in the case of the discontinuous biotechnological processes. Unlike the continuous processes, the batch processes are totally isolated from the outside environment. During the process no additional substrate is fed from the outside, the microorganisms population growing only from the substrate supplied at the beginning of the batch. A number of uncertainties arises right from the beginning of the process with respect to the substrate preparation and nature (the composition of the natural substrates is not known).

A method to control a biotechnological process is based on the physiological state of the cells, using the mean age as indicator [1], [8]. If the aim of the process is to obtain a reaction product, than someone can consider that each species is characterized by an optimal mean age, corresponding to the maximum productivity. Basically, the control method consists in command determining, so that the microorganisms population to be brought to an age accordingly to the maximum productivity. In [8] is developed the theory of the age and mean age models for a microorganisms population.

The paper deals with the identification of a mean age model for the growth process of *Candida lipolytica* population on an ammonium sulfate substrate, which is a batch process. The validation of mean age equation is based on the fact that in this equation the same parameters from the equations of biomass, substrate, and product do appear. As a consequence, a correct identification of the equations mentioned above leads to an age equation, which expresses the real evolution of the mean age. For the identification, a dynamic neural network was used. It has been trained at every hour, based on the experimental data from the process. The model parameters represents the weights of the neural network.

The paper structure is as follows: the second section presents the model of the growth process of *Candida lipolytica* population on an ammonium sulfate substrate, the firth section deals with a brief presentation of the dynamic neural networks, the fourth section presents the identification procedure, in the fifth section the simulation results are shown and the last is dedicated to the conclusions.

2 The mean age model of an enzyme biosynthesis process

In this paper the growth process of *Candida lipolytica* population on an ammonium sulfate substrate is considered. The process model is given by the following reaction scheme [4]:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} S - E \xrightarrow{k_{+2}} M + E \tag{1}$$

where *S* is the substrate, *E* is the enzyme, *S-E* represents the enzyme-substrate complex, *M* is a metabolite and $k_i, i \in \{+1, -1, +2\}$ are the kinetic rate constants.

Two assumptions are made about the process variables:

1. The concentration of the total enzyme *E_c(t)* which is the sum of enzyme-substrate complex *Cp(t)* and sole enzyme *E(t)*, is proportional to the biomass concentration *X(t)*:

$$E_0(t) = \varepsilon X(t) \tag{2}$$

2. The growth rate of the microorganisms population $\dot{X}(t)$ is proportional to the rate of metabolite formation $\dot{M}(t)$:

$$\dot{X}(t) = \psi \dot{M}(t) \tag{3}$$

The model is given by the following equations:

$$\dot{X}(t) = \mu \cdot C_p(t) \quad (4)$$

$$\dot{S}(t) = -\nu\{S(t)(\varepsilon X(t) - C_p(t)) - K_S C_p(t)\} \quad (5)$$

$$\dot{C}_p(t) = \nu\{S(t)(\varepsilon X(t) - C_p(t)) - K_m C_p(t)\} \quad (6)$$

where

$$\mu = \psi k_{+2}\,,\ \nu = k_{+1}\,, K_S = k_{-1}/k_{+1}\,,\ K_m = (k_{-1} + k_{+2})/k_{+1}\,,\ K_m \rangle K_S \quad (7)$$

In equations (4) – (6), which describe the dynamics of the growth process of *Candida lipolytica* population on an ammonium sulfate substrate, μ is the specific growth rate, ν represents the specific consumption rate and K_S and K_m are saturation constants. The model is complete if the mean age equation is added:

$$\frac{dm(t)}{dt} = 1 - \mu(t) \cdot m(t) \quad (8)$$

where *m(t)* is the mean age of the microorganisms population.

The equation (8) was determined from the notions provided by Ranta [8].

1. the total biomass concentration

$$X(t) = \int_0^\infty x(t,\tau)d\tau \quad (9)$$

2. the first and n^th moments

$$\Phi^1(t) = \int_0^\infty \tau x(t,\tau)d\tau = <\tau> \quad (10a)$$

$$\Phi^n(t) = \int_0^\infty \tau^n x(t,\tau)d\tau = <\tau^n> \quad (10b)$$

3. the mean age

$$m(t) = \frac{<\tau>}{<1>} = \frac{\Phi^1(t)}{X(t)} \quad (11)$$

where τ is the age coordinate.

Based on equations (9) – (11) the mean age derivative can be determined:

$$\frac{dm(t)}{dt} = \frac{1}{X(t)} \cdot \frac{d\Phi^1(t)}{dt} - \frac{\Phi^1(t)}{X^2(t)} \cdot \frac{dX(t)}{dt} \quad (12)$$

and taking into account the fact that the process is batch, equation (8) does result.

3 The parameter identification using recurrent neural networks

The neural networks can be classified in two categories: feedforward and recurrent networks [7]. In feedforward neural networks the processing elements are connected such that all signals are going in a single direction, from the input to the output units. The recurrent networks contain both feedforward and feedback paths.

The feedforward networks have been successfully applied to the dynamic systems identification. For the representation of a dynamic system the use of the delay lines has been adopted. This method involves the consideration of the current and previous inputs and outputs of the system as inputs of the network. The system output at the next moment is used as training signal. One of the biggest drawbacks of this method consists in the great number of calculus generated by the number of neurons from the input layer. As a consequence the identification is extremely sensitive to the external noise.

Due to their structure the recurrent networks don't have the same drawbacks. They can be classified in total or partial recurrent networks and can have two kind of connections: feedforward and feedback, all connections being trainable. The main structure of the partial recurrent networks is of feedforward type. The feedback connections contribute to the storage of the output layer statements that are not trainable.

Jordan proposes a structure, where the feedback connections are restricted from the output to the hidden layer, such that an additional layer named *State Layer* is created [3]. In this layer, each neuron gets as input an output signal of the neural network and its own output at the previous moment in the regime "own feedback". The structure proposed by Jordan is capable to store the information regarding the previous outputs for an infinite period.

Elman suggests a neural network with a very restricted feedback [2]. The recurrence is given by a Context Layer which makes a copy of the internal states of the hidden layer and then the internal states are introduced in the network at the next moment. The context neurons have linear activation functions, such that they don't produce signal processing (as the inputs neurons do). They do behave as memories. The recurrent weights from the hidden layer to the context one are set to 1, such that they permit the training using the backpropagation algorithm.

The Functional Links Neural Network (FLNN) has been proposed as an alternative architecture for the multilayer Perceptron network (MLP) for solving approximation problems of non-linear functions and classification problems. FLNN forward propagates the input variables and differs from the MLP network because it doesn't contain internal layers [5]. They are replaced by a number of knots named functional links. The functional links perform supplementary transformation of the network input space giving supplementary inputs to the network neuron. Basically, the functional links extend the original input space in a superior dimension aiming to reduce the loading of the computing unit in the training phase of the neural network. The functional link acts on an element of the input vector or on every input vectors generating a set of linear independent functions. Then these functions are evaluated. One can notice no new information has been introduced in the process. However the representation was certainly increased and the separation becomes possible in the increased space. As a consequence, both the training period and the training error are improved.

4 The experimental identification of the bioprocess

The identification scheme is given in figure 1. It contains the following elements: linear neurons; elements with internal dynamics (integrators); functional links (multiplying type). The scheme from figure 1 is used for the identification of the process described by equations (4) – (6), presented in section 2.

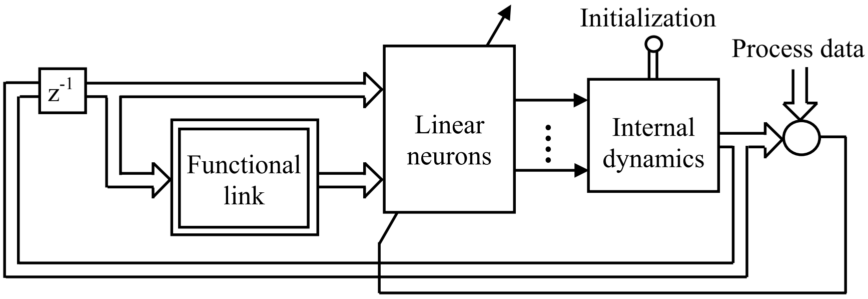

Fig. 1. The general scheme of the identification process

For the parameter adjustment the square error criterion between the experimental data and the model outputs is defined:

$$I = \int_0^t e^2(\tau) d\tau \tag{13}$$

where: $e = X_m - X$, with X_m - the measured data and X - the model outputs. The parameter vector $\underline{p} = [\mu \ \ \nu \ \ \varepsilon \ \ k_S \ \ k_M]^{\mathrm{T}}$ at the step k is calculated by a gradient method:

$$\underline{p}(k) = \underline{p}(k-1) - \nabla_{\underline{p}} I \tag{14}$$

where $\nabla_p I$ is defined for each component p_i:

$$\frac{\partial I}{\partial p_i} = \int_0^t 2e(\tau) \frac{\partial e(\tau)}{\partial p_i} \mathrm{d}\tau \tag{15}$$

The sensitivity of the biomass X with respect to each parameter p_i (s_{Xp_i}) is given by equation (16) [6]:

$$\frac{\partial e(\tau)}{\partial p_i} = \frac{\partial\left(X_m(\tau)-X(\tau)\right)}{\partial p_i} = -\frac{\partial X(\tau)}{\partial p_i} = -s_{Xp_i} \qquad (16)$$

Taking into account equations (15) and (16), the equation (14) becomes:

$$\underline{p}(k) = \underline{p}(k-1) + h\int_0^t e(\tau)s_{X\underline{p}}\,\mathrm{d}\tau \qquad (17)$$

where h is the length of the searching step.

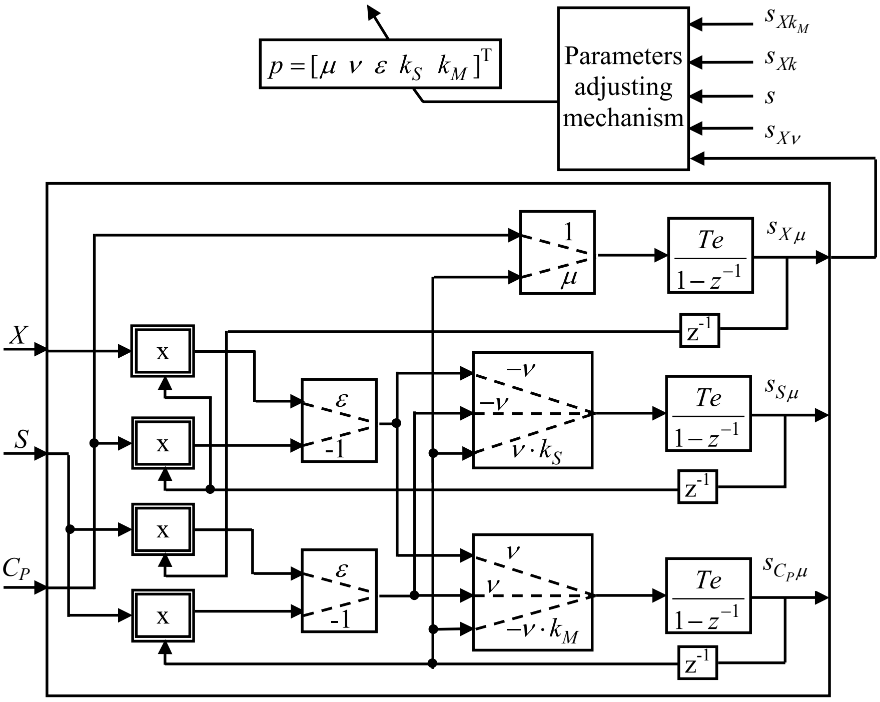

Fig. 2. The block for the calculus of the biomass sensitivity with respect to the parameter μ

One can notice that although the sensitivities of each state variable must be calculated with respect to the parameter vector p, in the equation where the parameters are adjusted the biomass sensitivities in relation with the parameter vector p do appear. The computing mechanism of the biomass sensitivity with respect to the parameter μ is presented in figure 2. The other sensitivities (S_{Xk_M} , S_{Xk_S} , $S_{X\varepsilon}$, $S_{X\nu}$) are calculated by similar blocks. The equations that define the sensitivities mentioned above are presented in appendix 1.

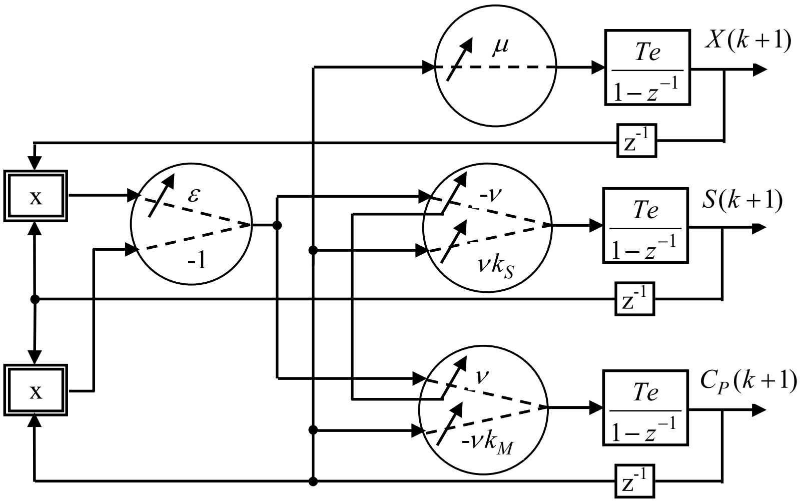

Fig. 3. The dynamic neuronal network structure

The neural network structure used for the bioprocess identification is of Elman type and is presented in figure 3. It contains four linear neurons to which, by the weights adjustment, the process parameters are obtained. The weight adjustment that determines the parameter ν(to the neurons 3 and 4) is synchronous done.

5 Simulation results

The simulation begins with a parameter set randomly chosen. After 10 hours of functioning, the adjustment procedure of the parameters does begin. The adjustment is done at every hour. The one hour period has been chosen based on two elements: the biotechnological processes are very slow (the sample period in the case of the process considered in the present paper is about 0.1 hour) and the experimental data are difficult to obtain (by taking proves and lab analyses). In one hour a new set of parameters is determined (by network training), based on the previous parameter set. In figure 4 are given the evolutions of biomass, substrate, enzyme-substrate complex and the mean age obtained with the identification procedure presented in section 3.

The smallest errors are obtained in the case of the biomass because it is the variable according to which the parameter adjustment is made. One can notice a good convergence of the model variables evolutions with respect to the real ones. The convergence is better if the parameter adjustment is done on a bigger horizon. The validation of the mean age evolution can be considered correct because in the mean age equation there are the parameters from the other model equations (biomass, substrate and product), which have been adjusted to the real values.

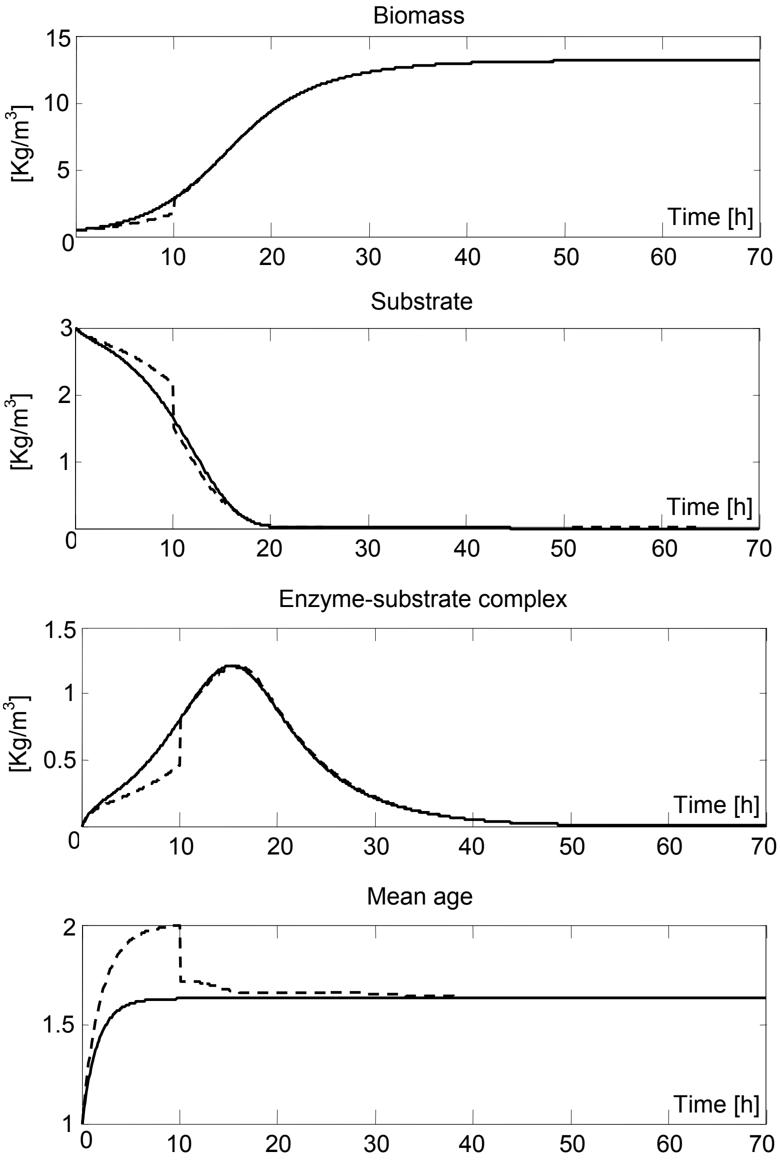

Fig. 4. The comparison between the evolution of the process and model variables (process variables – continuous line and model variables – dotted line)

Conclusions

The paper clearly presents the difficulty to obtain a “good model” for a biotechnological process. The difficulty is bigger if the model includes the mean age of the microorganisms population.

The authors tried to show that a dynamic neural network is a good tool to model such a complex process. In fact, the process identification consists in the determining the process parameters, which are weights of the dynamic neural network. The convergence of the parameter values to the real ones is very good. It becomes better if the time horizon for the parameter adjustment is bigger.

The parameters of the mean age equation are also determined. They are the parameters of the other equations of the model, such that the validity of the mean age equation is assured.

References

[1] S. Caraman, E. Ceangã, L. Frangu, G. Mencinicopschi, "Modelling and Controlling Techniques of Microorganisms Mean Age in Biotechnological Processes", European Control Conference, Karlsruhe, CD-ROM Proc., 1999

[2] J.L. Elman, "Finding structure in time", Cognitive Science, No.14, pp. 179-211, 1990

[3] M.I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine", Proc. of 8th Conf. of the Cognitive Society, USA, pp. 531–546, 1986

[4] M. Katoh, H. Kanoh, M. Masubuchi, "Analysis and State Estimation of Enzyme Kinetic Model for Microbial Growth Processes", 8th IFAC World Congress, Kyoto, Japan, pp. 2803-2808, 1981

[5] L. Mirea, T. Marcu, "System Identification Using Functional-Link Neural Networks With Dynamic Structure", 15th IFAC World Congress, Barcelona, Spain, CD-ROM Proceedings, 2002

[6] C. Penescu, G. Ionescu, M. Tertisco, E. Ceanga, "Experimental Identification of Automatized Processes", Technical Editure, Bucharest, Romania, 1971

[7] D.T. Pham, X. Liu, "Neural Networks for Identification, Prediction and Control", Springer-Verlag, London, Great Britain, 1995

[8] Ranta, I., "On the Mathematical Modelling of Microbial Age Dynamic and Some Control Aspects of Microbial Growth Processes", Acta Polytechnica Scandinavica, Mathematics and Computer Science Series, nr. 35, Helsinki, 1982

Appendix 1 – The sensitivity functions

$$s_{X\nu}(k+1) = s_{X\nu}(k) + Te \cdot \mu \cdot s_{C_P\nu}(k)$$

$$s_{X\mu}(k+1) = s_{X\mu}(k) + Te\left(C_P(k) + \mu \cdot s_{C_P\mu}(k)\right)$$

$$s_{X\varepsilon}(k+1) = s_{X\varepsilon}(k) + Te \cdot \mu \cdot s_{C_P\varepsilon}(k)$$

$$s_{Xk_S}(k+1) = s_{Xk_S}(k) + Te \cdot \mu \cdot s_{C_Pk_S}(k)$$

$$s_{Xk_M}(k+1) = s_{Xk_M}(k) + Te \cdot \mu \cdot s_{C_Pk_M}(k)$$

$$\begin{aligned} s_{S\nu}(k+1) &= s_{S\nu}(k) + Te\Big(-\Big(S(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) - k_S \cdot C_P(k)\Big) - \\ &\quad - \nu \cdot \Big(s_{S\nu}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + S(k)\big(\varepsilon \cdot s_{X\nu}(k) - s_{C_P\nu}(k)\big) - k_S \cdot s_{C_P\nu}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{S\mu}(k+1) &= s_{S\mu}(k) + Te\Big(-\nu \cdot \Big(s_{S\mu}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + \\ &\quad + S(k)\big(\varepsilon \cdot s_{X\mu}(k) - s_{C_P\mu}(k)\big) - k_S \cdot s_{C_P\mu}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{S\varepsilon}(k+1) &= s_{S\varepsilon}(k) + Te\Big(-\nu \cdot \Big(s_{S\varepsilon}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + \\ &\quad + S(k)\big(X(k) + \varepsilon \cdot s_{X\varepsilon}(k) - s_{C_P\varepsilon}(k)\big) - k_S \cdot s_{C_P\varepsilon}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{Sk_S}(k+1) &= s_{Sk_S}(k) + Te\Big(-\nu \cdot \Big(s_{Sk_S}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + \\ &\quad + S(k)\big(\varepsilon \cdot s_{Xk_S}(k) - s_{C_Pk_S}(k)\big) - C_P(k) - k_S \cdot s_{C_Pk_S}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{Sk_M}(k+1) &= s_{Sk_M}(k) + Te\Big(-\nu \cdot \Big(s_{Sk_M}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + \\ &\quad + S(k)\big(\varepsilon \cdot s_{Xk_M}(k) - s_{C_Pk_M}(k)\big) - k_S \cdot s_{C_Pk_M}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{C_P\nu}(k+1) &= s_{C_P\nu}(k) + Te\Big(\Big(S(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) - k_M \cdot C_P(k)\Big) + \\ &\quad + \nu \cdot \Big(s_{S\nu}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + S(k)\big(\varepsilon \cdot s_{X\nu}(k) - s_{C_P\nu}(k)\big) - k_M \cdot s_{C_P\nu}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{C_P\mu}(k+1) &= s_{C_P\mu}(k+1) + Te\Big(-\nu \cdot \Big(s_{S\mu}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + \\ &\quad +S(k)\big(\varepsilon \cdot s_{X\mu}(k) - s_{C_P\mu}(k)\big) - k_M \cdot s_{C_P\mu}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{C_P\varepsilon}(k+1) &= s_{C_P\varepsilon}(k) + Te\Big(-\nu \cdot \Big(s_{S\varepsilon}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + \\ &\quad + S(k)\big(X(k) + \varepsilon \cdot s_{X\varepsilon}(k) - s_{C_P\varepsilon}(k)\big) - k_M \cdot s_{C_P\varepsilon}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{C_Pk_S}(k+1) &= s_{C_Pk_S}(k) + Te\Big(-\nu \cdot \Big(s_{Sk_S}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + \\ &\quad + S(k)\big(\varepsilon \cdot s_{Xk_S}(k) - s_{C_Pk_S}(k)\big) - k_M \cdot s_{C_Pk_S}(k)\Big)\Big) \end{aligned}$$

$$\begin{aligned} s_{C_Pk_M}(k+1) &= s_{C_Pk_M}(k) + Te\Big(-\nu \cdot \Big(s_{Sk_M}(k)\big(\varepsilon \cdot X(k) - C_P(k)\big) + \\ &\quad + S(k)\big(\varepsilon \cdot s_{Xk_M}(k) - s_{C_Pk_M}(k)\big) - C_P(k) - k_M \cdot s_{C_Pk_M}(k)\Big)\Big) \end{aligned}$$