Optimal Boundary Control Synthesis for a Class of Distributed Parameter Systems Applying Adaptive Critics

Daniela G. Toshkova¹, Petar D. Petrov² and G. P. Toshkov³

¹,²,³Technical University of Varna
daniela_toshkova@abv.bg

Abstract—In the present paper optimal boundary control problem for a class of distributed parameter systems of parabolic type is considered. The simplified adaptive critic methodology is applied for the optimal control synthesis. The methodology requires the distributed parameter system to be reduced to lumped parameter system, as well as to be subsequently discretized. The approximation to finite-dimensional system is realized by using the generalized finite integral transform technique.

I. INTRODUCTION

The optimal control problem for distributed parameter system is well studied subject from the theoretical point of view. The complexity of the models, described by partial differential equations, however can lead to difficulties in realization of the obtained theoretical results regarding the methods for optimal control synthesis. That is why it is necessary new engineering approaches to be developed. Combing the analytical methods and soft computing methods, in particular neural networks could be such a possibility.

Recently a large number of papers considering neural networks based optimal control for linear and nonlinear lumped parameter systems appeared. The adaptive critics methodology for optimal control synthesis for distributed parameter systems is also developed [3, 4]. The optimality conditions are derived by applying the dynamic programming method and the obtained functions are approximated using neural networks. The optimal control law of linear or nonlinear system is determined through consecutively adapting of two neural networks – action neural network and critic neural network. The action neural network captures the relationship between the state and the control and the critic neural network captures the relationship between the state and the co-state. Later the methodology is simplified and the action neural network drops off [6]. Examples of the methodology applications are developed in [2, 5, 8], which show that the methodology is applicable for different classes of distributed parameter systems. In these publications however an example of applying of adaptive critics in the case of boundary control is not given. Moreover the requirement the first derivative of the state with respect to the control to be different from zero makes the methodology inapplicable for boundary control cases.

In the present paper this restriction is surmounted for a class of plants, governed by parabolic partial differential equations by reducing the distributed parameter system to lumped parameter system applying the generalized finite integral transform technique. After that the optimal control is synthesized by using the adaptive critics methodology.

II. PROBLEM STATEMENT

In the present publication a class of plants, governed by parabolic partial differential equations with spatial configuration of plate, cylinder and sphere type. The dynamics is described by the equation

\[
\frac{\partial \theta(\zeta, t)}{\partial t} = a \left[ \frac{\partial^2 \theta(\zeta, t)}{\partial \zeta^2} + \frac{\Gamma}{\zeta} \frac{\partial \theta(\zeta, t)}{\partial \zeta} \right] \zeta \in \Omega; \\ (1)
\]

The initial condition has the following form:

\[
\theta(\zeta, 0) = \phi(\zeta) \ \ \zeta \in \overline{\Omega} \ ; \ t = 0 \ \ (2)
\]

The boundary conditions are as follows:

\[
\alpha_1 \frac{\partial \theta(\zeta, t)}{\partial \zeta} + \beta_1 \theta(\zeta, t) = u(t) \ \ \zeta \in S \ ; \ t \in (0, \infty) \ \ (3)
\]

\[
\alpha_2 \frac{\partial \theta(\zeta, t)}{\partial \zeta} + \beta_2 \theta(\zeta, t) = 0 \ \ \ (4)
\]

where

\[
\Omega \in \mathbb{R}^n - \text{domain (it may be unbounded) of generalized spatial variable } \zeta \text{ variation;}
\]

\[
S \text{ – smooth boundary of the domain } \Omega;
\]

\[
\overline{\Omega} = \Omega \cup S;
\]

\[
\theta(\zeta, t) \in C^1_{1/2}((0, T) \times \Omega) \cap C([0, T] \times \overline{\Omega}) \ - \text{ function, describing the plant state at point } \zeta \text{ and time } t;
\]

\[
u(t) \in C([0, T] \times S), \sup u < \infty \ - \text{ function, describing the control actions, which are distributed along the boundary } S;
\]

\[
\alpha, \beta, a, b, c, \gamma \ - \text{ given nonnegative coefficients or smooth enough nonnegative functions } \in C^2([0, \infty)) \cap C([0, \infty));
\]

\[
\Gamma \in L_2(\Omega) \cap C(\Omega) \ - \text{ function, describing the interaction between the state and the co-state};
\]

\[
\zeta(t) \in C^1_{1/2}((0, T) \times \Omega) \ - \text{ function, describing the control's trajectory}.
\]
The Dirichlet, Neumann and Robin boundary conditions in (3) and (4) are formed in the following way:

- if $\alpha_i = 0, \beta_i \neq 0$ - Dirichlet boundary condition
- if $\alpha_i \neq 0, \beta_i = 0$ - Neumann boundary condition
- if $\alpha_i \neq 0, \beta_i \neq 0$ - Robin boundary condition

The constant $\Gamma$ has the following values:

- $0$ - space configuration plate
- $1$ - space configuration cylinder
- $2$ - space configuration sphere

The problem is to find such a control, which minimizes the performance index:

$$J = \sum_{k=1}^{N-1} \frac{1}{2} \left( \theta_k^T Q \Delta t \theta_k + R \Delta u_k^2 \right)$$

(8)

where $Q$ and $R$ are weight coefficients.

### III. APPROXIMATION AND DISCRETIZATION

The adaptive critic methodology requires system dynamics to be discrete. Another requirement of the methodology is the first derivative of the state equation with respect to the control to exist and to be different from zero. In order to surmount the restriction the derivative is equal to zero. In case of boundary control however the derivative is assumed to be differentiable with respect to $\theta_k$ and $u_k$.

The aim is to find feedback control $u_k$ as a function of $\theta_k$ by following the dynamic programming method [1].

By using the formula for the cost function in (8), the cost function for the time $k$ has the following form:

$$J_k = \sum_{k=1}^{N-1} \frac{1}{2} \left( \theta_k^T Q \Delta t \theta_k + R \Delta u_k^2 \right)$$

(9)

where $\Psi_k$ and $J_{k+1}$ represent are utility function at time step $k$ and cost-to-go from time $k+1$ to $N$ respectively. The co-state vector with dimensions $n \times 1$ is specified by the expression

$$\lambda_k = \frac{\partial J_k}{\partial \theta_k}$$

(11)

The necessary optimality conditions through which the optimal control can be found are given by

$$\frac{\partial J_k}{\partial u_k} = 0$$

(12)

After transforming [3, 4] of the Eq. (12) and taking into consideration Eq. (10) the optimal control equation can be written as follows:

$$u_k = \frac{C_\mu}{a_\mu R \Delta t} \lambda_{k+1}$$

(13)

After developing the Eq. (11) [3, 4], the co-state equation takes the following form:

$$\lambda_k = Q \Delta t \theta_k + \frac{1}{(1 - a_\mu)} \lambda_{k+1}$$

(14)

In order to obtain the optimal control it is necessary Eqs. (7), (13) and (14) to be solved simultaneously taking into consideration the corresponding boundary conditions.
V. OPTIMAL CONTROL SYNTHESIS BY APPLYING THE SIMPLIFIED ADAPTIVE CRITIC METHODOLOGY FOR LUMPED PARAMETER SYSTEMS

Main characteristics of simplified adaptive critic methodology for lumped parameter systems is that the critic networks capture the relationship between $\theta_k$ and $\lambda_{k+1}$. The methodology can be applied for a large class of problems where the optimal control is explicitly solvable in terms of $\theta_k$ and $\lambda_{k+1}$. $u_k$ can be calculated easily after successful training the neural networks as it is supposed that $u_k$ is known function of $\theta_k$ and $\lambda_{k+1}$. The main steps of the simplified adaptive critics approach for optimal control synthesis are given below (Fig.1).

A. State profile generation for neural network training

In the optimal control synthesis based on adaptive critics first the time step is fixed. The set $S = \{ \theta_k : \theta_k \in \text{domain of interest} \}$, for which the neural networks will be trained, is defined. The set has to be chosen in such a way that its elements approximately cover entirely the subdomain of the state space, in which the trajectory is supposed to lie at time $k$ starting from any initial state belonging to the domain of interest.

For regulator problems it is known [1] that as time increases the states tend to zero. Therefore the set $S$ has to contain states with different magnitudes including ones close to zero. In order this requirement to be met as well as the requirement the set to include as many points from the trajectory as possible, the procedure below can be followed.

For $i = 1, 2$, the set $S_i$, which is defined as $S_i = \{ \theta_k : \| \theta_k \|_\infty \leq C_i \}$, where $C_i$ is positive constant.

For $C_1 \leq C_2 \leq C_3, \ldots, S_1 \subseteq S_2 \subseteq S_3 \subseteq \ldots$ Therefore for given $i = I, S_I$ will include the whole domain of interest for the initial conditions. The synthesis procedure starts with fixing small value to the constant $C_1$ and the networks are trained for the states generated for the set $S_1$. After the proper convergence of the networks is reached, a new constant $C_2$ close to $C_1$ is chosen and the networks are trained again for the profiles within $S_2$ and so on. The constant $C_i$ is increased in this way till the whole domain of interest for the initial conditions is included in the set $S_I$.

B. Neural network initialization

In order to obtain a stable set of initial weights it is necessary the neural networks to be initialized. This is fulfilled by using the liner optimal control theory [9].

The discretized system is described by the following equations:

$$\theta_{k+1} = A_d \theta_k + B_d u_k,$$

where

$$A_d = \begin{bmatrix}
    (1-a\mu_1^2) & 0 & \cdots & 0 \\
    0 & (1-a\mu_2^2) & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & (1-a\mu_n^2)
\end{bmatrix},$$

and

$$B_d = \begin{bmatrix}
    a\mu_1 \\
    a\mu_2 \\
    \vdots \\
    a\mu_n \\
\end{bmatrix}.$$
The control is given by the equation [9]

\[ u_k = -K_d \theta_{k+1} \]  

(16)

Co-state can be written as

\[ \lambda_k = S_d \theta_k \]  

(17)

where the \( n \times n \) matrix \( S_d \) is the solution of the following Reiccati equation:

\[ A_d^T S_d A_d - S_d - A_d^T S_d B_d \left( R_d + B_d^T S_d B_d \right)^{-1} B_d^T S_d A_d + Q_d = 0 \]  

(18)

One can note that \( S_d \) gives the relationship between \( \lambda_k \) and \( \theta_k \) while the neural networks to be trained are supposed to approximate the functional relationship between \( \theta_k \) and \( \lambda_{k+1} \). Using the state equation (15), the optimal control equation (16) and the co-state equation (17) one can establish the following relationship:

\[ \lambda_{k+1} = S_d \theta_{k+1} = S_d (A_D - B_D K_D) \theta_k \]  

(19)

Eq. (19) is used for pre-training of the neural networks. Arbitrary values are assigned to \( \theta_k \) and the corresponding values of \( \lambda_{k+1} \) are generated.

C. Neural network training

Simplified adaptive critic algorithm for optimal control synthesis contains following steps.

1. The set \( S_i \) is generated. For each element of \( S_i \) the steps below are followed.
   - Input \( \lambda_i \) to the neural networks to get \( \lambda_{k+1,i} \). Let it be denoted by \( \lambda_{k+1}^{a,i} \)
   - Calculate \( u_i \) knowing \( \lambda_{k+1,i} \) from the optimal control equation (13)
   - Using \( u_i \) and \( \theta_k \), \( \theta_{k+1,i} \) is obtained from the state equation (7)
   - Input \( \theta_{k+1,i} \) to the neural networks to get \( \lambda_{k+2,i} \).
   - Calculate \( \lambda_{k+1,i} \) from the co-state equation. Let it be denoted by \( \lambda_{k+1}^{t,i} \)

2. The neural networks are trained with all \( \theta_k \) as input and all corresponding \( \lambda_{k+1}^{t,i} \) as output.

3. Check for convergence as described below.

4. If proper convergence is achieved, stop and revert to step 1 with \( i = i + 1 \). If not – go to step 1 and re-train the neural networks.

5. The process continues till \( i = L \).

To minimize the chance of getting trapped in a local minimum, the network may be trained for all of the elements of \( S_i \) together.

D. Convergence conditions

Before changing the value of the constant \( C_i \) to \( C_{i+1} \) and generating new profiles for further training of neural networks it should be checked if proper convergence is achieved for the constant value \( C_i \). It can be done in the following manner.

1. Assign to the constant \( C_i \) same values, which are used for neural networks training. Generate the set \( S_i^{C_i} \) of profiles in the same way as \( S_i \) is generated. This set will be used for network convergence check.

2. Fix a value to the tolerance \( tol \).

3. Using the profiles from \( S_i^{C_i} \) generate the target outputs as it is described in the previous subsection. Let the outputs are \( \lambda_1^{C_i}, \lambda_2^{C_i}, \ldots, \lambda_n^{C_i} \).

4. Generate the actual output from the networks by simulating the trained networks with the profiles from \( S_i^{C_i} \). Let the values are \( \lambda_1^p, \lambda_2^p, \ldots, \lambda_n^p \).

5. Check if \( \| \lambda_i^p - \lambda_i^{C_i} \|_2 < tol \), \( \forall p = 1, 2, \ldots, n \). If yes it is assumed that proper convergence of the networks is achieved.

For faster convergence the convex combination \( \beta \lambda_{k+1} + (1 - \beta) \lambda_{k+1}^{C_i,0} \) can be used as target output where \( 0 < \beta < 1 \) is the learning rate for neural networks training.

E. Neural network structure

One of the most important stages is the choice of both the type and the structure of the neural networks. The choice of neural network with small number of neurons and hidden layers can lead to non-adequate determination of the function. The choice of neural network with large number neurons and hidden layers can lead to delay of the training, which is connected to the possibility of getting trapped in local minima. As far as the author knows there is not a certain method for determining neural network structure. The number of the neurons in each layer and the number of layers can vary depending on the problem.

Here a feedforward neural network is chosen. The neural network consists of four layers—input, output and two hidden ones. The first (input) has five neurons with log-sigmoid activation function. The second and the third (hidden) layers again consist of five neurons with log-sigmoid activation function. The forth (output) layer has one neuron with purelin activation function.

V. SIMULATION STUDIES

For the simulation the plant is chosen to be a plate. It has the following characteristics: heat conduction coefficient \( a = 1,04.10^{-6} \); relative heat capacity \( c = 1,13 \) kJ/(kg.K); density \( \gamma = 1452 \) kg/m³; relative heat exchange coefficient \( H = 6,83 \) m; width \( L = 0,04 \) m.

The distributed parameter system is reduced to lumped parameter system by means the generalized finite integral transform technique. Then the system is discretized. The optimal control is synthesized by using the simplified adaptive critic methodology.

In Fig. 3 the temperature profiles at time \( t_1 = 1800 \) s, \( t_2 = 3600 \) s and \( t_3 = 5400 \) s respectively are given. Fig. 4 shows the optimal trajectories for the layer \( \zeta = 0,01 \) m, layer \( \zeta = 0,02 \) m.
0.02 m, layer \( \zeta = 0.03 \) m and layer \( \zeta = 0.04 \) m respectively. In Fig. 5 the optimal boundary control, which is synthesized by using the simplified adaptive critics methodology is depicted.

In order to check the accuracy and the applicability of the methodology the function LQR of Matlab is used for comparison. In Fig. 6, Fig. 7 and Fig. 8 the temperature profile, the optimal trajectories in the different layers and the optimal boundary control, synthesized by using the function LQR from Matlab are shown.
V. CONCLUSIONS

On the basis of the simulation results the following conclusions may be drawn. There is a good coincidence between the results obtained by means of the approach using the adaptive critic methodology and the approach using the LQR theory. Therefore the approach is applicable for synthesis of optimal control for distributed parameter systems of parabolic type with boundary conditions of Dirichlet, Neuman or Robin type and boundary control. The choice of the number of the hidden layers and the number of neurons in each layer have an effect on the exactness of the synthesis of the optimal control. As far as the author knows there are not exact rules for choice of neural network structure, then it is done on the basis of series of experiments and the most exact variant is chosen.

REFERENCES