Generalization of Backpropagation for LQ Control Tasks

Dušan Krokavec, Anna Filasová

Technical University of Košice, Faculty of Electrical Engineering and Informatics,
Department of Cybernetics and Artificial Intelligence
Letná 9/B, SK-042 00 Košice, Slovak Republic
e-mail: dusan.krokavec@tuke.sk, anna.filasova@tuke.sk

Abstract: The questions addressed in this paper concern the applicability of dynamic system LQ control tasks using neural networks as well as a method how the exposed problems can be reduced to a standard formulation using dual heuristic dynamic programming and back-propagation training. There are presented some background materials on the recursive least-square methods, formulated in terms of generalized LQ control and task-determination of neural network structure. The results arise with such neural representation and their training capability, which could be useful in uncertain dynamical system control. This application can be considered as a task concerned the class of problems referred to as reinforcement learning and is based on the existence of the dynamic model of systems.

Keywords: discrete-time LQ control, gradient optimization, Q-learning, heuristic dynamic programming.

1 Introduction

One of the principal reasons for introducing feedback into a control system is to obtain relative insensibility to changes in plant parameters and to disturbances. It is well-known fact that linear quadratic (LQ) optimal control yields a stable closed-loop system and the minimal value of the criterion for any initial (non-zero) condition. The disadvantage of applying these ideas, however, is that the computational burden associated with the implementation is proportional to the cube of system order.

The LQ control design can be cast as an optimization problem that involve matrix equation, where this equation have the form of discrete or algebraic Riccati equation. In general, optimization problem solving need methods that can capture high order complexity and uncertainty of the system. One class of this method is

Adaptive Critic Design (ACD). ACD approximate dynamic programming for optimal decision making in noisy, non-stationary or non-linear environments.

Using neural network a typical ACD include action, critic and model modules. Each module can be a neural network or, alternatively, any differentiable system. Heuristic dynamic programming is so a neural network approach to solve Bellman equation, where for neural LQ control structure three neural nets can be used –one used to train the control gain (action), one for functioning as the Lyapunov function observer (critic), and a third could be trained to copy the system model. Good knowledge of the derivatives of an optimization criterion is a prerequisite to find a solution.

The paper present an algorithm to solve the optimization tasks concerning with neural structure design of the discrete-time LQ control, where dual heuristic programming is used for realization of this closed form structure. Dual heuristic dynamic programming have an important advantage since its critic module produces a representation for parameter derivatives being explicitly trained on them.

The most applicable publications which have dealt with the above mentioned problem are presented in paper References.

2 Discrete-time LQ Control

In general, a discrete-time multi-variable system can be considered as

$$\mathbf{x}(i+1) = \mathbf{F}\mathbf{x}(i) + \mathbf{G}\mathbf{u}(i) \tag{1}$$

$$\mathbf{y}(i) = \mathbf{C}\mathbf{x}(i) \tag{2}$$

where $\mathbf{x}(i)$ ™ P^n, $\mathbf{u}(i)$ ™ P^r, $\mathbf{y}(i)$ ™ P^m, are state, input and output vectors, respectively, and system matrices $\mathbf{F}$ ™ P^{nxn}, $\mathbf{G}$ ™ P^{nxr}, $\mathbf{C}$ ™ P^{mxn}, are finite valued ones.

For such an system (1), (2), which must be controllable, the optimal control design task is to determine the control

$$\mathbf{u}(i) = -\mathbf{K}(i)\mathbf{x}(i) \tag{3}$$

that minimizes the quadratic cost function

$$J_N = \mathbf{x}^T(N)\mathbf{Q}^*\mathbf{x}(N) + \\ + \sum_{i=0}^{N-1} (\mathbf{x}^T(i)\mathbf{Q}\mathbf{x}(i) + \mathbf{x}^T(i)\mathbf{S}\mathbf{u}(i) + \mathbf{u}^T(i)\mathbf{S}^T\mathbf{x}(i) + \mathbf{u}^T(i)\mathbf{R}\mathbf{u}(i)) \tag{4}$$

where N is finite, $\mathbf{Q}$ ™ P^{nxn} and $\mathbf{Q}^*$ ™ P^{nxn} are symmetric positive semi-definite matrices, $\mathbf{R}$ ™ P^{mxm} is a symmetric positive definite matrix, $\mathbf{S}$ ™ P^{mxm} is a constant matrix and $\mathbf{K}(i)$ ™ P^{rxm} is the optimal control gain matrix.

Using (3) criterion (4) can be rewritten as

$$J_N = \mathbf{x}^T(N)\mathbf{Q}^*\mathbf{x}(N) + \sum_{i=0}^{N-1} \mathbf{x}^T(i)\mathbf{J}_J(i)\mathbf{x}(i) \tag{5}$$

$$\mathbf{J}_J(i) = \mathbf{Q} - \mathbf{S}\mathbf{K}(i) - \mathbf{K}^T(i)\mathbf{S}^T + \mathbf{K}^T(i)\mathbf{R}\mathbf{K}(i) \tag{6}$$

or

$$J_N = \mathbf{x}^T(N)\mathbf{Q}^*\mathbf{x}(N) + \sum_{i=0}^{N-1} q(\mathbf{x}(i), \mathbf{u}(i)) \tag{7}$$

$$q(\mathbf{x}(i), \mathbf{u}(i)) = \mathbf{x}^T(i)\mathbf{Q}\mathbf{x}(i) + \mathbf{x}^T(i)\mathbf{S}\mathbf{u}(i) + \mathbf{u}^T(i)\mathbf{S}^T\mathbf{x}(i) + \mathbf{u}^T(i)\mathbf{R}\mathbf{u}(i) \tag{8}$$

The Lyapunov function may be used in the procedure of LQ optimal control design. For the best obtainable Lyapunov function

$$V(\mathbf{x}(i)) = \mathbf{x}^T(i)\mathbf{P}(i-1)\mathbf{x}(i) \tag{9}$$

the function difference is given by

$$\Delta V(\mathbf{x}(i)) = \mathbf{x}^T(i+1)\mathbf{P}(i)\mathbf{x}(i+1) - \mathbf{x}^T(i)\mathbf{P}(i-1)\mathbf{x}(i) \tag{10}$$

where $\mathbf{P}(i)$ ™ P^{nxn} is a positive definite matrix, $\mathbf{P}(-1) = \mathbf{P}(0)$ and $\Delta V(\mathbf{x}(i) < 0$. Using (1), in accordance with (3), for the difference of the Lyapunov function follows

$$\begin{aligned} \Delta V(\mathbf{x}(i)) &= (\mathbf{F}\mathbf{x}(i) + \mathbf{G}\mathbf{u}(i))^T \mathbf{P}(i)(\mathbf{F}\mathbf{x}(i) + \mathbf{G}\mathbf{u}(i)) - \mathbf{x}^T(i)\mathbf{P}(i-1)\mathbf{x}(i) = \\ &= v(\mathbf{x}(i), \mathbf{u}(i)) = \mathbf{x}^T(i)\mathbf{J}_V(i)\mathbf{x}(i) \end{aligned} \tag{11}$$

$$v(\mathbf{x}(i), \mathbf{u}(i)) = (\mathbf{F}\mathbf{x}(i) + \mathbf{G}\mathbf{u}(i))^T \mathbf{P}(i)(\mathbf{F}\mathbf{x}(i) + \mathbf{G}\mathbf{u}(i)) - \mathbf{x}^T(i)\mathbf{P}(i-1)\mathbf{x}(i) \tag{12}$$

$$\mathbf{J}_V(i) = (\mathbf{F} - \mathbf{G}\mathbf{K}(i))^T \mathbf{P}(i)(\mathbf{F} - \mathbf{G}\mathbf{K}(i)) - \mathbf{P}(i-1) \tag{13}$$

respectively.

The Lyapunov function value at the time point $i = N$-1 for a non-zero initial system state vector is

$$V = \sum_{i=0}^{N-1} \Delta V(\mathbf{x}(i)) = \mathbf{x}^T(N)\mathbf{P}(N-1)\mathbf{x}(N) - \mathbf{x}^T(0)\mathbf{P}(0)\mathbf{x}(0) \tag{14}$$

and using (11) – (13)

$$V = \sum_{i=0}^{N-1} \Delta V(\mathbf{x}(i)) = \sum_{i=0}^{N-1} v(\mathbf{x}(i), \mathbf{u}(i)) = \sum_{i=0}^{N-1} \mathbf{x}^T(i)\mathbf{J}_V(i)\mathbf{x}(i) \tag{15}$$

Adding (15) and subtracting (14) (i.e. adding zero value) to (7), the cost function for the linear feedback control law (3) can be expressed as

$$\begin{aligned} J_N = {} & \mathbf{x}^T(N)\mathbf{Q}^*\mathbf{x}(N) - \mathbf{x}^T(N)\mathbf{P}(N-1)\mathbf{x}(N) + \mathbf{x}^T(0)\mathbf{P}(0)\mathbf{x}(0) + \\ & + \sum_{i=0}^{N-1} (q(\mathbf{x}(i), \mathbf{u}(i)) + v(\mathbf{x}(i), \mathbf{u}(i))) \end{aligned} \tag{16}$$

Setting

$$\mathbf{P}(N-1) = \mathbf{Q}^* \tag{17}$$

the cost function (16) takes the form

$$J_N = \mathbf{x}^T(0)\mathbf{P}(0)\mathbf{x}(0) + \sum_{i=0}^{N-1} (q(\mathbf{x}(i), \mathbf{u}(i)) + v(\mathbf{x}(i), \mathbf{u}(i))) \tag{18}$$

$$J_N = \mathbf{x}^T(0)\mathbf{P}(0)\mathbf{x}(0) + \sum_{i=0}^{N-1} \mathbf{x}^T(i)\mathbf{J}(i)\mathbf{x}(i); \quad \mathbf{J}(i) = \mathbf{J}(i)_J + \mathbf{J}_V(i) \tag{19}$$

respectively. In this form the cost function is explained in dependency on a non-zero initial system state vector values.

3 Control Law Optimization

It is presumed that the optimal control law optimized (18), i.e. the followed modified criterion

$$\begin{aligned} & p(\mathbf{x}(i), \mathbf{u}(i)) = q(\mathbf{x}(i), \mathbf{u}(i)) + v(\mathbf{x}(i), \mathbf{u}(i)) = \\ & = \begin{bmatrix} \mathbf{x}^T(i) & \mathbf{u}^T(i) \end{bmatrix} \left(\begin{bmatrix} \mathbf{Q} + \mathbf{F}^T\mathbf{P}(i)\mathbf{F} - \mathbf{P}(i-1) & \mathbf{S} + \mathbf{F}^T\mathbf{P}(i)\mathbf{G} \\ (\mathbf{S} + \mathbf{F}^T\mathbf{P}(i)\mathbf{G})^T & \mathbf{R} + \mathbf{G}^T\mathbf{P}(i)\mathbf{G} \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}(i) \\ \mathbf{u}(i) \end{bmatrix} \end{aligned} \tag{20}$$

has to be minimized. It is evident, that the condition

$$\begin{aligned} & \frac{\partial p(\mathbf{x}(i), \mathbf{u}(i))}{\partial \mathbf{u}^T(i)} = \\ & = \begin{bmatrix} \mathbf{x}^T(i) & \mathbf{u}^T(i) \end{bmatrix} \left(\begin{bmatrix} \mathbf{Q} + \mathbf{F}^T\mathbf{P}(i)\mathbf{F} - \mathbf{P}(i-1) & \mathbf{S} + \mathbf{F}^T\mathbf{P}(i)\mathbf{G} \\ (\mathbf{S} + \mathbf{F}^T\mathbf{P}(i)\mathbf{G})^T & \mathbf{R} + \mathbf{G}^T\mathbf{P}(i)\mathbf{G} \end{bmatrix} \right) \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} = \mathbf{0} \end{aligned} \tag{21}$$

imply

$$(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})^T\mathbf{x}(i)+(\mathbf{R}+\mathbf{G}^T\mathbf{P}(i)\mathbf{G})\mathbf{u}(i)=\mathbf{0} \tag{22}$$

$$\mathbf{u}(i)=-(\mathbf{R}+\mathbf{G}^T\mathbf{P}(i)\mathbf{G})^{-1}(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})^T\mathbf{x}(i)=-\mathbf{K}(i)\mathbf{x}(i) \tag{23}$$

$$\mathbf{K}(i)=(\mathbf{R}+\mathbf{G}^T\mathbf{P}(i)\mathbf{G})^{-1}(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})^T \tag{24}$$

respectively, and condition

$$\begin{aligned}&\frac{\partial p(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i)}=\\ &=\begin{bmatrix}\mathbf{x}^T(i)\ \mathbf{u}^T(i)\end{bmatrix}\left(\begin{bmatrix}\mathbf{Q}+\mathbf{F}^T\mathbf{P}(i)\mathbf{F}-\mathbf{P}(i-1) & \mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G}\\ (\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})^T & \mathbf{R}+\mathbf{G}^T\mathbf{P}(i)\mathbf{G}\end{bmatrix}\right)\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}=\mathbf{0}\end{aligned} \tag{25}$$

give

$$(\mathbf{Q}+\mathbf{F}^T\mathbf{P}(i)\mathbf{F}-\mathbf{P}(i-1))\mathbf{x}(i)+(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})\mathbf{u}(i)=\mathbf{0} \tag{26}$$

$$(\mathbf{Q}+\mathbf{F}^T\mathbf{P}(i)\mathbf{F}-\mathbf{P}(i-1)-(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})\mathbf{K}(i))\mathbf{x}(i)=\mathbf{0} \tag{27}$$

$$\begin{aligned}&\mathbf{P}(i-1)=\mathbf{Q}+\mathbf{F}^T\mathbf{P}(i)\mathbf{F}-(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})\mathbf{K}(i)=\\ &=\mathbf{Q}+\mathbf{F}^T\mathbf{P}(i)\mathbf{F}-(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})(\mathbf{R}+\mathbf{G}^T\mathbf{P}(i)\mathbf{G})^{-1}(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})^T\end{aligned} \tag{28}$$

If the control gain matrix is given by (24) and $\mathbf{P}(i)$ is a solution of the Riccati equation (28) then, using (27) and (24), the value of (19) can be expressed as

$$\begin{aligned}&\mathbf{J}(i)=\mathbf{Q}-\mathbf{S}\mathbf{K}(i)-\mathbf{K}^T(i)\mathbf{S}^T+\mathbf{K}^T(i)\mathbf{R}\mathbf{K}(i)+\\ &+(\mathbf{F}-\mathbf{G}\mathbf{K}(i))^T\mathbf{P}(i)(\mathbf{F}-\mathbf{G}\mathbf{K}(i))-\mathbf{P}(i-1)=\\ &=\mathbf{Q}+\mathbf{F}^T\mathbf{P}(i)\mathbf{F}-(\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})\mathbf{K}(i)-\mathbf{P}(i-1)-\\ &-\mathbf{K}^T(i)((\mathbf{S}+\mathbf{F}^T\mathbf{P}(i)\mathbf{G})^T-(\mathbf{R}+\mathbf{G}^T\mathbf{P}(i)\mathbf{G})\mathbf{K}(i))=\mathbf{0}\end{aligned} \tag{29}$$

and the minimal value of the criterion (18), (19) is

$$J_N=\mathbf{x}^T(0)\mathbf{P}(0)\mathbf{x}(0) \tag{30}$$

The design progresses backward in time from time point $i = N$-1, using the final value of the $\mathbf{P}(N\text{-}1)$, with the optimal gain matrix $\mathbf{K}(i)$ defined by (24) and with the matrix sequence $\mathbf{P}(i\text{-}1)$ obtained from Riccati equation (28). It is evident that (28) form a set of nonlinear difference equations, which may be solved recursively starting from $\mathbf{P}(N\text{-}1) = \mathbf{Q}^*$.

4 Parameterization of Control Design Task

The control design objective is to construct a feedback controller $\mathbf{u}(i) = -\mathbf{K}(i)\mathbf{x}(i)$ such that the quadratic performance index (4) is minimized. This problem is equivalent to finding for system state $\mathbf{x}(i)$ the control function $\mathbf{u}(i) = g(\mathbf{x}(i))$ and the Lyapunov function $V(\mathbf{x}(i))$ of given special structure.

Assuming, that system is on the form of discrete-time state-space description and the performance index is (7), then the design task conditions can be rewritten as

$$f(\mathbf{x}(i),\mathbf{u}(i)) = \mathbf{F}\mathbf{x}(i) + \mathbf{G}\mathbf{u}(i) = \mathbf{x}(i+1) \tag{31}$$

$$q(\mathbf{x}(i),\mathbf{u}(i)) = \mathbf{x}^T(i)\mathbf{Q}\mathbf{x}(i) + \mathbf{u}^T(i)\mathbf{R}\mathbf{u}(i) + \mathbf{x}^T(i)\mathbf{S}\mathbf{u}(i) + \mathbf{u}^T(i)\mathbf{S}^T\mathbf{x}(i) \tag{32}$$

$$g(\mathbf{x}(i)) = -\mathbf{K}(i)\mathbf{x}(i) = \mathbf{u}(i) \tag{33}$$

and for stabilization the Lyapunov function $V(\mathbf{x}(i))$

$$V(\mathbf{x}(i)) = \mathbf{x}^T(i)\mathbf{P}(i-1)\mathbf{x}(i) \tag{34}$$

$$v(\mathbf{x}(i),\mathbf{u}(i)) = \mathbf{x}^T(i+1)\mathbf{P}(i)\mathbf{x}(i+1) - \mathbf{x}^T(i)\mathbf{P}(i-1)\mathbf{x}(i) = \Delta V(\mathbf{x}(i)) \tag{35}$$

is used. Using the dynamic programming principle the actual error minimization can be considered as the minimization of the function

$$p(\mathbf{x}(i),\mathbf{u}(i)) = q(\mathbf{x}(i),\mathbf{u}(i)) + v(\mathbf{x}(i),\mathbf{u}(i)) = q(\mathbf{x}(i),\mathbf{u}(i)) + \Delta V(\mathbf{x}(i)) \tag{36}$$

The Pontryagin minimum principle implies, if there is no bounds on $\mathbf{u}(i)$, the minimizing $\mathbf{u}(i)$ must be such, that

$$\begin{aligned}\frac{\partial p(\mathbf{x}(i),\mathbf{u}(i))}{\partial g(\mathbf{x}(i))} &= \frac{\partial q(\mathbf{x}(i),\mathbf{u}(i))}{\partial g(\mathbf{x}(i))} + \frac{\partial v(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i+1)}\frac{\partial f(\mathbf{x}(i),\mathbf{u}(i))}{\partial g(\mathbf{x}(i))} = \\ &= \frac{\partial q(\mathbf{x}(i),\mathbf{u}(i))}{\partial g(\mathbf{x}(i))} + \frac{\partial V(\mathbf{x}(i+1))}{\partial \mathbf{x}(i+1)}\frac{\partial f(\mathbf{x}(i),\mathbf{u}(i))}{\partial g(\mathbf{x}(i))} = 0\end{aligned} \tag{37}$$

$$\begin{aligned}\frac{\partial p(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i)} &= \frac{\partial q(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i)} + \frac{\partial v(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i+1)}\frac{\partial f(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i)} = \\ &= \frac{\partial q(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i)} + \frac{\partial V(\mathbf{x}(i+1))}{\partial \mathbf{x}(i+1)}\frac{\partial f(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i)} - \frac{\partial V(\mathbf{x}(i))}{\partial \mathbf{x}(i)} = 0\end{aligned} \tag{38}$$

Applying this, the value of $p(\mathbf{x}(i),\mathbf{u}(i))$ be zero.

5 Action Network Structure

In the sense of the Pontryagin minimum principle, the target for an action network minimization can be defined as zero and the network output error is

$$\mathbf{e}_a(i) = 0 - \frac{\partial p(\mathbf{x}(i),\mathbf{u}(i))}{\partial g(\mathbf{x}(i))} = -\left(\frac{\partial q(\mathbf{x}(i),\mathbf{u}(i))}{\partial g(\mathbf{x}(i))} + \frac{\partial V(\mathbf{x}(i+1))}{\partial \mathbf{x}(i+1)} \frac{\partial f(\mathbf{x}(i),\mathbf{u}(i))}{\partial g(\mathbf{x}(i))} \right) \tag{39}$$

Using the criterion

$$W_a(i) = \tfrac{1}{2}\mathbf{e}_a^T(i)\mathbf{e}_a(i) = \tfrac{1}{2}\sum_{h=1}^{r} e_{ah}^2(i) \tag{40}$$

for the action neural network training, a steepest-descent discrete gradient method, based on error back-propagation algorithm, can be applied to solve this minimization problem, i.e.

$$\begin{aligned} \Delta w_{rs}(i) &= -\mu_a \frac{\partial W_a(i)}{\partial w_{rs}(i)} = -\mu_a \left(\frac{\partial q(i)}{\partial \mathbf{u}(i)} + \frac{\partial V(i+1)}{\partial \mathbf{x}(i+1)} \frac{\partial \mathbf{x}(i+1)}{\partial \mathbf{u}(i)} \right) \frac{\partial \mathbf{u}(i)}{\partial w_{rs}(i)} = \\ &= -\mu_a \sum_{k=1}^{r} \left(\frac{\partial q(i)}{\partial u_k(i)} + \sum_{j=1}^{n} \frac{\partial V(i+1)}{\partial x_j(i+1)} \frac{\partial x_j(i+1)}{\partial u_k(i)} \right) \frac{\partial u_k(i)}{\partial w_{rs}(i)} \end{aligned} \tag{41}$$

where

$\frac{\partial x_j(i+1)}{\partial u_k(i)}$ - are calculated from analytical equation of the system model,

$\frac{\partial V(i+1)}{\partial x_j(i+1)}$ - are approximated by the critic network,

$\frac{\partial q(i)}{\partial u_k(i)}$ - are calculated as a derivative of performance index.

Variable n and r designate the number of state and input variables, respectively and this target is fixed trough whole control time.

Those, the full-connected input/output action network structure have to be used

$$\mathbf{w}_{AI}^T(i) = \left[\mathbf{x}^T(i) \;\; \mathbf{u}^T(i) \right]; \quad \mathbf{w}_{AO}^T(i) = \left[\mathbf{0} \right] \tag{42}$$

with linear neuron activation functions.

Trained action neural network presents a non-parametric representation of gain vector $\mathbf{K}(i)$, where matrix $(\mathbf{S} + \mathbf{F}^T\mathbf{P}(i)\mathbf{G}^T)$ is determined by synaptic weight products between $\mathbf{x}^T(i)$ and $\mathbf{w}_{AO}(i)$ and matrix $(\mathbf{R} + \mathbf{G}^T\mathbf{P}(i)\mathbf{G})$ is given as a synaptic weigh products between $\mathbf{u}(i)$ and $\mathbf{w}_{AO}(i)$.

6 Critic Network Structure

Method of heuristic dynamic programming use a critic network, where the critic neural network is trained using the assumption of the optimal response. The critic is trained forward in time, which is of the great importance for real-time operation in LQ control based on neural network structure.

Since (38) implies

$$\frac{\partial V(\mathbf{x}(i))}{\partial \mathbf{x}(i)} = \frac{\partial q(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i)} + \frac{\partial V(\mathbf{x}(i+1))}{\partial \mathbf{x}(i+1)} \frac{\partial f(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{x}(i)} \tag{43}$$

the right side of (43) can be considered as a desired vector of partial derivatives of the Lyapunov function with respect to the state vector $\mathbf{x}(i)$, which gives for the j-th desired output of the critic neural network at all control point i

$$\begin{aligned} c_j^o(i) &= \frac{\partial q(\mathbf{x}(i),\mathbf{u}(i))}{\partial x_j(i)} + \frac{\partial q(\mathbf{x}(i),\mathbf{u}(i))}{\partial \mathbf{u}(i)} \frac{\partial \mathbf{u}(i)}{\partial x_j(i)} + \frac{\partial V(\mathbf{x}(i+1))}{\partial \mathbf{x}(i+1)} \frac{\partial \mathbf{x}(i+1)}{\partial x_j(i)} = \frac{\partial q(i)}{\partial x_j(i)} + \\ &+ \sum_{k=1}^{r} \frac{\partial q(i)}{\partial u_k(i)} \frac{\partial u_k(i)}{\partial x_j(i)} + \sum_{h=1}^{n} \frac{\partial V(i+1)}{\partial x_h(i+1)} \frac{\partial x_h(i+1)}{\partial x_j(i)} + \sum_{k=1}^{r} \sum_{h=1}^{n} \frac{\partial V(i+1)}{\partial x_h(i+1)} \frac{\partial x_h(i+1)}{\partial u_k(i)} \frac{\partial u_k(i)}{\partial x_j(i)} \end{aligned} \tag{44}$$

where

$$c_j(i) \approx \frac{\partial V(i)}{\partial x_j(i)} \tag{45}$$

and

$\frac{\partial x_j(i+1)}{\partial u_k(i)}$ - are calculated from analytical equation of the error model,

$\frac{\partial V(i+1)}{\partial x_j(i+1)}$ - are approximated (predicted) by the critic network,

$\frac{\partial q(i)}{\partial u_k(i)}$ - are calculated as a derivative of performance index,

$\frac{\partial q(i)}{\partial x_j(i)}$ - are calculated as a derivative of performance index,

respectively. Variable r and n designate the number of state and input variables.

The values

$\frac{\partial u_k(i)}{\partial x_j(i)}$

are given as the product of synaptic weights on the path from the *j*-th input to *k*-th output of the action neural network.

The training criterion for critic neural network can be defined as

$$W_c(i) = \frac{1}{2}\sum_{j=1}^{N}(c_j(i) - c_j^0(i))^2 \tag{46}$$

and the neural network optimization procedure is given by

$$\Delta w_{rs}(i) = -\mu_c \frac{\partial W_c(i)}{\partial w_{rs}(i)} \tag{47}$$

It is evident that the basic strategy to update the networks can be given by the straight application of (41) and (43), (46). The better critic neural network approximate criterion the better the action neural network will approximate an optimal control.

Also the critic network is the full-connected input/output network structure

$$\mathbf{w}_{CI}^T(i) = \left[\mathbf{x}^T(i) \;\; \mathbf{u}^T(i)\right]; \quad \mathbf{w}_{CO}^T(i) = \left[\mathbf{c}^{oT}(i)\right] \tag{48}$$

with linear neuron activation functions.

Using presented strategy the targets (needed for the critic network training) are typically calculated by running the critic network one more computational cycle to provide its next-in-time output, and then use this value to compute the target for the present-time cycle. The error term is calculated and the critic network update is performed in the usual way. Since the critic network that calculates the target is changing with each update, it provides a moving target for critic neural network training.

Conclusions

The paper presents some background material on the LQ control design, an overview of the heuristic dynamic programming problem and a survey of techniques considered from the point of feed-forward multi-layer perceptron neural network training.

Presented application, based on the existence of a complete model of the environment and the system model, involved then back-propagation utilities with system response parameterization. This approximation of the gradient algorithms for parameter updating in the sense of the mean value for given training set is a basic one for implementation of presented tasks using adaptive critic design for neuro-control, which is suitable for learning in noisy and non-stationary environments.

Applications can be considered as a task concerned the class of problems referred to as reinforcement learning algorithms. Reinforcement learning is a general way to formulate complex learning problems. The goal of the system is to maximize a

long terms sum of an instantaneous reward (provided by the teacher). It is a decision process based on system environment simulation and in its extremum form it only requires that the teacher can provide a measure of success.

Acknowledgements

The work presented in this paper was supported by Grant Agency of Ministry of Education and Academy of Science of Slovak Republic VEGA under Grant No. 1/2173/05.

References

[1] Bellman, R.E. - Kalaba, R *Dynamic Programming and Modern Control Theory*. New York : Academic Press, 1965

[2] Chichocky, A. - Unbehauen, R. *Neural Networks for Optimization and Signal Processing*, New York : Wiley, 1993, ISBN 0-471-93010-5

[3] Krokavec, D. - Filasová, A. *Optimal Stochastic Systems.* Košice : Elfa, 2002, ISBN 80-89066-52-6. (in Slovak)

[4] Krokavec, D. - Filasová, A. Application of heuristic dynamic programming to dynamic system stabilization. In *State of Art in Computational Intelligence* / P. Sinčák, J. Vačšák, V. Kvasnička, R. Mesiar (eds.). Heidelberg : Physica Verlag, 2000, pp. 196-201. (Advances in Soft Computing). ISBN 3-7908-1322-2, ISSN 1615-3871

[5} Krokavec, D. - Filasová, A. Neural Network Implementation of Robust Kalman Predictors. *Acta Electrotechnica at Informatica*, 2002, Vol. 2, No. 1, pp. 60-64. ISSN 1335-8243

[6] Krokavec, D. - Filasová, A. Quadratically stabilized discrete-time robust LQ control. In *Control System Design 2003 CSD '03: A proceedings volume from the 2nd IFAC Conference, 7-10 September, 2003, Bratislava, Slovak Republic,* / Š. Kozák, M. Huba (eds.). Elsevier, Oxford, 2004, s. 375-380. ISBN 0-08-044175-0, ISSN 1474-6670

[7] Krokavec, D. - Filasová, A. Closed form of neural system state estimator for stochastic environment. In *Proceedings of the 2nd Slovakian – Hungarian Joint Symposium on Applied Machine Intelligence SAMI 2004, 16-17 January, 2004, Herľany, Slovak Republic,* / I.J. Rudas, P. Sinčák (eds.), Budapest Polytechnic, Budapest, Hungary, 2004, pp. 103-110. ISBN 963-7154-23-X

[8] Pekař, J. - Havlena, V. The efficient robust MPC algorithm: Simulation results. In: *Proceedings of the 6th International Scientific-Technical Conference Process Control ŘIP 2004, Kouty nad Desnou, Czech Republic*, [CD ROM] / S. Krejčí (ed.). University of Pardubice, Czech Republic, pp. R146 1-6. ISBN 80-7194-662-1

[9] Prokhorov, D.V. - Santiago, R.A. - Wunsch, D.C. Adaptive critic design: a case study for neurocontrol. *Neural Networks,* 1995, Vol. 8, No. 9, pp. 1367-1372. ISSN 0893-6080

[10] Prokhorov, D. - Wunsch, D. Adaptive critic design. *IEEE Transactions on Neural Networks*, 1997, Vol. 8, No. 5, pp. 997-1007. ISSN 1045-9227

[11] Výtečková, M., Výteček, A. Multiple dominant poles method and its verification. *Reprints of the 5th International Scientific -Technical Conference Process Control ŘÍP 2002, Kouty nad Desnou, Czech Republic* [CD-ROM] / S. Krejčí at all. (eds). ISBN 80-7194-452-1

[12] Werbos, P. J. Consistency of HDP applied to a simple reinforcement learning problem. *Neural Networks*, 1990, Vol. 3, No. 2, pp. 179-189. ISSN 0893-6080