NEURO – ADAPTIVE OPTIMAL CONTROL SYSTEM FOR AIRCRAFTS

ROMULUS LUNGU
Avionics Department, University of Craiova
Decebal str. No.5

ROMANIA
MIHAI LUNGU
Avionics Department, University of Craiova
Decebal str. No.5

ROMANIA
NICOLAE JULA
Military Technical Academy of Bucharest
ROMANIA

COSTIN CEPISCA
Electrical Engineering Faculty, University POLITEHNICA of Bucharest
313 Splaiul Independentei, sector 6
ROMANIA
www.electro. pub.ro

Abstract: - The paper presents neuro–adaptive optimal control system with direct application to the aircrafts' control. The system has two neural networks, one with command role function of difference between leading system's output and reference signal and the other which has as training signal the difference between leading system state (nonlinear) and the linear model's state. The command law has an optimal component, which must compensate deviations of the linear model from optimal trajectory. For solving adjunct vector equation one evaluates exterior perturbation by its influence on deviation of the leading system's output from reference signal.

Key-Words: - command, optimal, adaptive, neural network, aircraft

1 Structure of the neuro – adaptive optimal control system

Flying object dynamic A is described by nonlinear state equation

$$\dot{x} = f\big(x(t), u(t), p(t), t\big), t \in [t_0, t_1], x(t_0) = x_0, \quad (1)$$

where x is the state vector $(n \times 1)$, u - command vector $(m \times 1)$ and p - perturbations vector $(n \times 1)$.

One presents a control system with a neural network

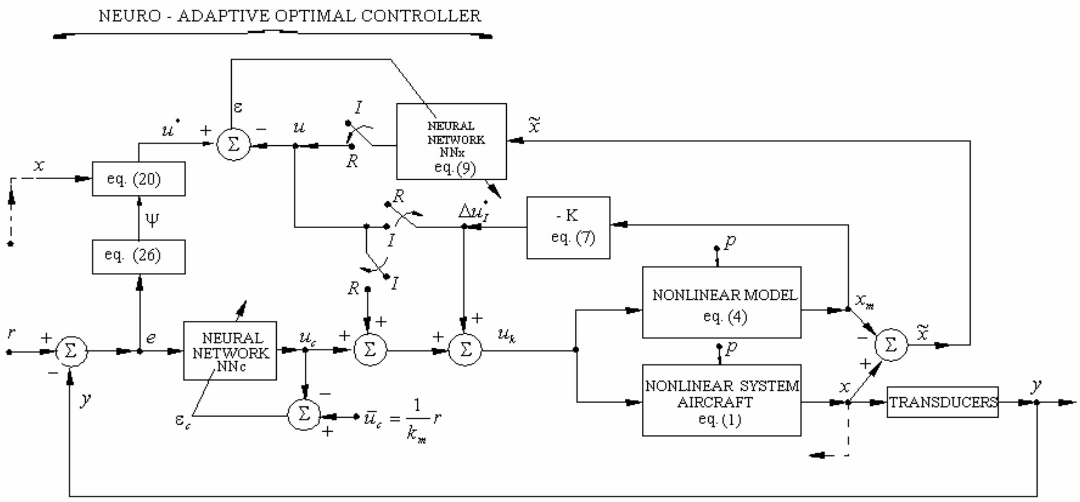

Fig.1 Neuro - adaptive system

NN_x (fig.1), which is trained in two steps. In the first step (initialization one) one obtains neural network's output function of gain matrix's elements; x_m is nonlinear model state of A, and ΔU_I^* - initialization command of NN_x (I expresses initialization stage and R expresses control stage).

In the second stage, neural network's train is made on – line because of the variables' time variations (deviation of real trajectory of A from linear model trajectory).

Neural network train is made, for example, using optimal criteria based on Pontreaghin maximum principle or based on Bellman dynamic propagation principle.

2 Control system project

Optimal control law project is a problem which consists of command vector $u \in U$ determination; it leads system (1) from initial state $x_0 = x(t_0)$ in the final state $x_1 = x(t_1)$ so that criterion [2]

$$J(x_0\, t_0, u, x_1, t_1) = M(x_1, t_1) + \int_{t_0}^{t_1} L(x(\tau), u(\tau), \tau)\,\mathrm{d}\tau \tag{2}$$

takes minimum value on array U.

Let x^*, u^* be the components on optimal trajectory and x_0^*, u_0^* -components on reference trajectory; then

$$x^*(t) = x_0^*(t) + \Delta x(t), u^*(t) = u_0^*(t) + \Delta u(t). \tag{3}$$

In **initialization stage** of the neural network NN_x one may use linear control theory for nonlinear control system command; linear system is described by equation [3]

$$\Delta\dot{x} = A\Delta x + B\Delta u + D\Delta p, \Delta x(t_0) = \Delta x_0. \tag{4}$$

Optimal command Δu^* determination may be done using condition that system passes in the final state so that quadratic criterion

$$\bar{J} = \frac{1}{2}\int_0^\infty \left[\Delta x^{\mathrm{T}}(\tau) Q \Delta x(\tau) + \Delta u^{\mathrm{T}}(\tau) R \Delta u(\tau) + 2\Delta x^{\mathrm{T}}(\tau) N \Delta u(\tau)\right]\mathrm{d}\tau \tag{5}$$

takes minimum value; matrix $Q(n \times n)$ is symmetric and positive semi defined, $R(m \times m)$ - nonsingular matrix, symmetric and positive defined, $N(n \times m)$ - symmetric matrix.

Generally, project problem solving consists in optimal command determination [4]

$$\Delta u_t^* = \Delta u^* + \Delta u_p^*, \tag{6}$$

which assures simultaneously optimal amortization of dynamic linear processes; it means to assure the convergence of linear model's trajectory to the reference one (component Δu^*) and compensation of exterior perturbations (component Δp^*)

Component Δu^* is expressed as it follows

$$\Delta u^* = -K\Delta x = -R^{-1}\left[B^{\mathrm{T}} P + N\right]\Delta x, \tag{7}$$

where K is gain matrix and P the solution of algebraic matricidal Riccati equation [3]

$$A^{\mathrm{T}} P + PA - PBR^{-1}B^{\mathrm{T}} P + Q = 0. \tag{8}$$

In **training stage** neural network calculates command law function of state vector $\tilde{x}$ (fig.1). One may use a neural network with only a hidden layer; equation shaped by a neural network is [5]

$$u = V^{\mathrm{T}} g[Wx + d] + b, \tag{9}$$

where W is the input weight vector, V output weight vector, d and b - the input and output biases and g - sigmoid functions' vector.

If in (2) $M(x_1, t_1) \to 0$ and t is an intermediary moment of time, $t \in [t_0, t_1]$, in which the system has $x(t)$ state on considered trajectory; one may define functional [2]

$$J(x(t), u(t), t) = \int_t^{t_1} L(x(\tau), u(\tau), t)\mathrm{d}\tau, \tag{10}$$

which attaches trajectory parts corresponding to the interval $[t, t_1)$ a number $I(x(t))$ well calculated; $u(t) = u(x(t))$,

$$I(x(t), t) = -J(x(t), u(t), t). \tag{11}$$

Then,

$$\frac{\mathrm{d}J}{\mathrm{d}t} = -L = -\frac{\mathrm{d}I}{\mathrm{d}t} = -\frac{\partial I}{\partial t} - \frac{\partial I}{\partial x}\dot{x} \tag{12}$$

and, taking into account equation (1) and the fact that the derivation is made on optimal trajectory, it results

$$\frac{\partial I(x,t)}{\partial t} + H\left(x(t), u(t), \frac{\partial I(x,t)}{\partial x}, t\right) = 0, \tag{13}$$

or Hamilton – Jacobi equation

$$\frac{\partial I(x,t)}{\partial t} + H\left(x(t), u(t), \frac{\partial I(x,t)}{\partial x}, t\right) = 0, \tag{14}$$

where H is the Hamilton function

$$H\left(x(t), u(t), \frac{\partial I(x,t)}{\partial x}, t\right) = -L(x(t), u(t), t) + \underbrace{\left[\frac{\partial I(x,t)}{\partial x}\right]^{\mathrm{T}}}_{\psi^{\mathrm{T}}} f(x(t), u(t), t). \tag{15}$$

Equation (14) is available for any trajectory which starts from x_0 and arrives in x_1, determined by an admitted command $u \in U$, inclusively along optimal trajectories; that means

$$\frac{\partial I^*\left(x^*(t),t\right)}{\partial t}+H\left(x^*(t),u^*(t),\frac{\partial I^*\left(x^*(t),t\right)}{\partial x},t\right)=0. \qquad (16)$$

If function $I^*\left(x^*(t),t\right)$ admits second order partial derivatives, by equation (16) derivation, one obtains equation

$$\dot{\psi}=-\left[\frac{\partial H\left(x(t),u(t),\psi(t),t\right)}{\partial x}\right]^*, \qquad (17)$$

For optimal command u^* determination one uses equation

$$\frac{\mathrm{d}H(x,u,\psi,t)}{\mathrm{d}u}=0 \Leftrightarrow \frac{\partial H}{\partial u}+\frac{\partial H}{\partial x}\frac{\partial x}{\partial u}=0 \qquad (18)$$

or, from equation (17), it results equation

$$\frac{\partial H}{\partial u}-\dot{\psi}\frac{\partial x}{\partial u}=0. \qquad (19)$$

By discretizating this equation, it results

$$\frac{\partial H[x(k),u(k)]}{\partial u(k)}-\psi(k+1)\frac{\partial x(k+1)}{\partial u(k)}=0, \qquad (20)$$

For calculus of the optimal command $u^*(t_k)$, one solves equation (20) using, for example, a Netwon – Raphson method, after calculus of $\psi(k+1)$. Though, considering that linear model (4), one obtains equation

$$\Delta\dot{x}=G\Delta x+B\Delta u_W^*+D\Delta p, \qquad (21)$$

where

$$G=A-BR^{-1}\left[B^{\mathrm{T}}P+N\right]. \qquad (22)$$

Component Δu_p^* of the optimal command (6) may be determined so that initial state x_0 passes to final state $x_1=0$ so that quadratic criterion

$$\bar{J}_p=\int_0^\infty\left[\Delta x^{\mathrm{T}}Q_y\Delta p+\frac{1}{2}\Delta u_p^{\mathrm{T}}R_y\Delta u_p\right]\mathrm{d}t \qquad (23)$$

takes minimum possible value; matrix $Q_y(n\times n)$ has elements $q_y^{ij}=\mathrm{sgn}(x_i\Delta p_j)$ for $x_i\neq 0,\Delta p_j\neq 0$, with $i,j=\overline{1,n}$, $q_y^{ij}=\mathrm{sgn}(\Delta p_j)$ for $x_i=0$ and $q_y^{ij}=\mathrm{sgn}(x_i)$ for $\Delta p_j=0$; $R_y(m\times m)$ may be chosen as unity matrix $(m\times m)$ [4].

With equation (17), one obtains

$$\dot{\psi}(t)=-G^{\mathrm{T}}\psi(t)+Q_y\Delta p(t). \qquad (24)$$

From stationary condition of function H, from equation (18) it results equation

$$\Delta u_p^*(t)=R_y^{-1}B^{\mathrm{T}}\psi(t). \qquad (25)$$

Calculus equation for adjunct vector ψ is (24), which assumes exterior perturbation's knowing. But, in general, this isn't possible; in equation (24) one may replace term $Q_y\Delta p$ by other term which expresses perturbation's effect. Perturbation affects output vector y of the nonlinear system and vector y_m of the linear model. As a consequence, the perturbation affects system error $e=r-y, r$ being reference vector (imposed). So, equation (24) may be expressed as it follows

$$\dot{\psi}(t)=-G^{\mathrm{T}}\psi(t)+k(r-y), \qquad (26)$$

where k is a gain coefficient with a variable value; one may choose $k=1$. If $k\neq 1$, the effect is equivalent with supplementary perturbation's appearance, which is compensated by a feedback loop after output vector y. Hence, ψ may be calculated by integrating of equation (26), with $k=1$.

NN_c neural network models an equation with form (9) and is trained by error's minimization

$$\varepsilon_c=\left\|\bar{u}_c-u_c\right\|^2, \qquad (27)$$

where $\bar{u}_c$ may be calculated with equation

$$\bar{u}_c=\frac{1}{k_m}r, \qquad (28)$$

k_m being direct way subsystem's gain coefficient of the control loop with feedback after output vector y. When $e\to 0(y\to r, u_c\to\bar{u}_c, x\to x_s=\mathrm{ct.})$, $\psi\to 0$, $u^*\to u_s^*=\mathrm{ct.}=h(x_s)$, $u\to u_s^*=\mathrm{ct.}$ and $\varepsilon\to 0$, $u(k)\to u_s(k)=\mathrm{ct.}$ and, consequently, $\tilde{x}\to 0$ and $x_m\to x_{ms}=\mathrm{ct.}\to x_s$.

4 Conclusions

One chooses a linear model very closed to the nonlinear one of the leading system, whose state $x_m\to x$ in regime without disturbances.

In the initialization stage one leads model in x_m state $(\Delta x_m=0)$ by use of optimal control law $\Delta U_I^*=-K\Delta x_m$.

In the training stage of the neural network NN_x

$$u_k=u_c+\Delta U_I^*+u, \qquad (29)$$

where u_c is the output of neural network NN_c (neural regulator), whose input is the difference between leading system's output and the reference signal r. The imposed value is $\bar{u}_c=\frac{1}{k_m r}$, because at equilibrium $y=k_m u_c\to k_m\bar{u}_c \Leftrightarrow \bar{u}_c=\frac{1}{k_m}y\to\frac{1}{k_m}r$; adaptive component u of the command law is thhe input of neural network NN_x, whose input is the difference $\widetilde{x} = x - x_m$ because of disturbance P and deviation of model A from the reference one. Imposed value u^* of NN_x 's output is calculated function of adjunct vector ψ , which is the solution of equation (26).

References:

[1] Narendra, K.S., Parthasaranthy, K. *Identification and Control of dynamical systems using neural networks.* I.E.E.E. Transaction Neural networks, Vol.1, pag. 4-27, 1990.

[2] Călin, S., Belea, C. *Sisteme automate adaptive şi optimale*. Editura Tehnică, Bucureşti, 1971.

[3] Călin, S., Belea, C. *Sisteme automate complexe.* Editura Tehnică, Bucureşti, 1983.

[4] Lungu, R. *Automatizarea aparatelor de zbor.* Editura Universitaria, Craiova, 2002.

[5] Ferrari, S., Stengel, R.F. *An adaptive Global Controller.* American Control Conference, May 2002.