Optimal Feedback Control of Nonlinear Systems with Control Vector Constraints

Josip Kasač, Branko Novaković*

Abstract

This paper presents the derivation of the learning algorithm for feedback one-layer neural network controller for optimal control of nonlinear multivariable systems with control vector constraint. Derived is the off-line algorithm for the feedback optimal control problem with fixed terminal time and the algorithm for the time optimal control problem. Derivation of the algorithm is based on backpropagation through time (BPTT) algorithm, which is used as a learning algorithm for dynamic neural networks, and it is not based on the calculus of variations. The derived algorithms are used for feedback minimum-time control of the robot with two degree of freedom. The paper also considers the possibilities of on-line optimal regulation.

Keywords: Feedback optimal control, minimum-time control, neural networks, robot control, on-line control

1 Introduction

Generally speaking, controllers can be divided into two main classes: regulators and terminal controllers, [1]. Regulators maintain the state of the system about some known reference state despite external disturbances and internal uncertainties, [2]. Terminal controllers drive the system to some final state at which point the control task is terminated. These terminal controllers have received less attention in the literature than regulators, [3].

The only exact way to find the control law for the general nonlinear optimal feedback control problem is solving the Hamilton-Jacobi-Bellman (HJB) partial differential equation. However, the cost of solving HJB equation "exactly" (in a finite element approximation) rises exponentially with the number of state variables in the system. The only special case where we actually can solve the HJB equation exactly is time-varying linear systems with quadratic performance criteria.

One approach in solving the HJB equation is the approximation of the solution by some kind of universal function approximator - a neural network. Then we can try to adapt the weights in that network to satisfy the performance criteria. The ability of neural networks to learn arbitrary nonlinear mappings can be effectively utilised to represent the controller nonlinearity.

*Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, I. Lučića 5, Zagreb, Croatia, e-mail: josip.kasac@fsb.hr, branko.novakovic@fsb.hr

Especially interesting, but also very difficult is the problem of Time Optimal Control (TOC) with control vector constraints. There are many cases where this is applied in industry, for example robotic manipulators, where increasing the speed of a maneuver is of primary importance, and minimum-time control is an attractive control strategy for this purpose.

One of the approaches in solving this problem is an assumption in the form of a bang-bang solution, which is generally sub-optimal but facilitates the resolution and implementation. Using a three-layer backpropagation neural network, a controller that learns the switching time and then uses it in minimum-time control was designed, [4].

This paper presents the derivation of a learning algorithm for the adjustment of weight parameter of the one-layer neural network controller for terminal control of nonlinear systems. The algorithm derivation is based on the backpropagation-through-time (BPTT) algorithm. BPTT algorithm, [5, 6], is time generalization of the backpropagation algorithm (BP), in case when the error which is minimised is given along the specified time interval. The essence of the BP algorithm is a simple and precise calculation of derivations of the cost function in relation to system parameters and the adjustment of parameters in line with those derivations in only one transfer through the network. BPTT algorithm expands this method through application to dynamic systems for which a direct calculation of derivations can be very complex. A solution to this problem lies in the chain rule for order derivations [5, 7, 8], which results in error backpropagation, i.e. parameter adjustment backward in time.

2 Problem formulation

We consider the control of the continuous time-invariant deterministic nonlinear plant, represented by

$$\dot{\mathbf{x}}(t) = \hat{\mathbf{f}}(\mathbf{x}(t),\ \mathbf{u}(t)), \tag{1}$$

where $\mathbf{x}(t) \in \Re^n$ and $\mathbf{u}(t) \in \Re^m$ respectively denote the state vector and control vector. Function $\hat{\mathbf{f}}(\cdot)$ is n - dimensional known function which is the first order differentiable in regard to $\mathbf{x}(t)$ and $\mathbf{u}(t)$.

Further, the initial and final state are given

$$\mathbf{x}(t_0) = \mathbf{x}_0, \qquad \mathbf{x}(t_f) = \mathbf{x}_f, \tag{2}$$

and control vector constraints

$$|\mathbf{u}(t)| \leq \mathbf{u}^{\max}. \tag{3}$$

For the feedback control law we assumed the following form

$$\mathbf{u}(t) = \mathbf{g}\left(\mathbf{x}(t), \mathbf{r}(t)\right), \tag{4}$$

where $\mathbf{g}(\cdot)$ is an m - dimensional, appropriate generally nonlinear function to be determined according to some performance criteria, and $\mathbf{r}(t)$ is exogenous reference input (in our case - final state $\mathbf{x}_f$). We can use a neural network to approximate nonlinear function $\mathbf{g}(\cdot)$ and what follows is

$$\mathbf{u}(t) = \mathbf{g}\left(\mathbf{x}(t), \mathbf{r}(t), \mathbf{W}\right), \tag{5}$$

where $\mathbf{W}$ is the constant weight matrix of neural network, which can be determined according to the performance criteria

$$J = \min_{\mathbf{W}} \int_{t_0}^{t_f} F\left(\mathbf{x}(t),\ \mathbf{u}(t)\right) dt, \tag{6}$$

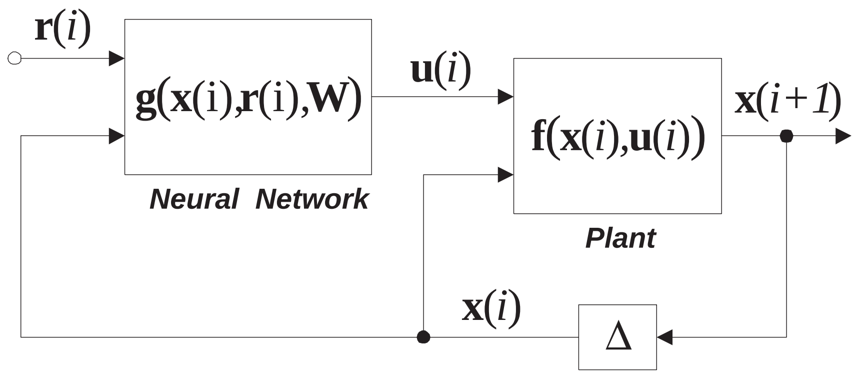

Figure 1: Feedback control with the neural network controller.

where $F(\cdot)$ is the first order differentiable scalar function. The basic structure of this neural network controller is schematically shown in Fig. 1.

In this paper we consider a special form of function $\mathbf{g}(\cdot)$, the one-layer perceptron, which doesn't have the ability to generalize arbitrary nonlinear function, but in our example (minimum-time control of robot with two-degree of freedom) this isn't significant. On the other hand, one advantage is the minimum number of weight parameters which we need to adjust, and another advantage is avoidance of the local minimum problem which is a characteristic of multi-layer perceptrons. Therefore, we use the following form of the feedback control law

$$u_j(t) = S_j(z_j), \qquad j = 1, 2, ..., m \tag{7}$$

where

$$z_j(t) = w_{j0} + \sum_{k=1}^{n} w_{jk} x_k(t), \tag{8}$$

$$S_j(z_j) = \begin{cases} u_j^{\max}; & z_j > u_j^{\max} \\ z_j; & -u_j^{\max} \le z_j \le u_j^{\max} \\ -u_j^{\max}; & z_j < -u_j^{\max} \end{cases} \tag{9}$$

Function $S_j(z_j)$ is the activation function in the form of a saturation limiter, which has the threshold of saturation in $\pm u_j^{\max}$, (Fig. 2). This form of the activation function is the guaranty of the constraint satisfaction (3). The parameters w_{jk} (for j = 1, 2, ..., m , k = 1, 2, ..., n) are synaptic weights, and w_{j0} (for j = 1, 2, ..., m) is a bias or offset.

3 Learning algorithm for weights adjustment in neurocontroller

The next step is the derivation of the learning algorithm for weights and bias adjustment according to the performance criteria (6) and constrains in the form of the plant equation (1) and the initial and final boundary condition (2). The form of the activation function (7)-(9) satisfies the control constraints.

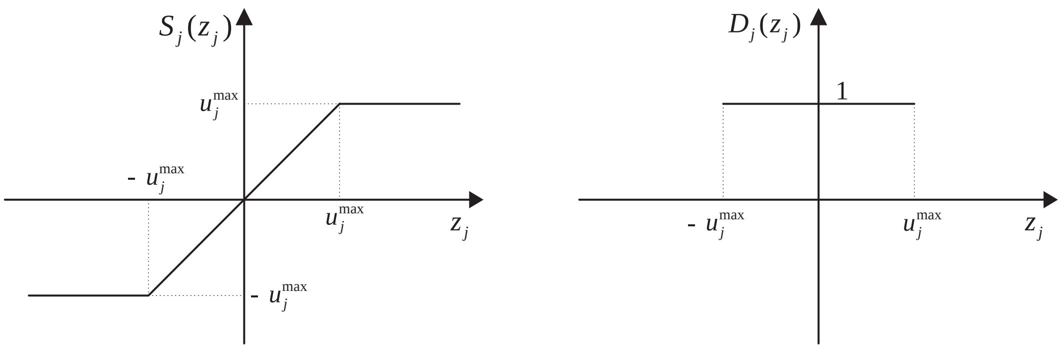

Figure 2: Activation function $S_j(z_j)$ and derivation $D_j(z_j)$.

3.1 Optimal feedback control with fixed terminal time

The discrete time form of overall performance criteria with the penalty function for final boundary conditions is

$$J(\mathbf{W}) = T\sum_{i=0}^{N-1} F(\mathbf{x}(i),\, \mathbf{u}(i)) + G(\mathbf{x}(N)), \tag{10}$$

where

$$G = K\sum_{k=1}^{n} (x_k(N) - x_k(t_f))^2, \tag{11}$$

and where K is the coefficient of the penalty function for final boundary conditions. The discrete time form of the plant equation (1) is

$$x_j(i+1) = f_j(\mathbf{x}(i), \mathbf{u}(i)), \quad j = 1, 2, ..., n \tag{12}$$

where

$$f_j(\mathbf{x}(i),\, \mathbf{u}(i)) = x_j(i) + T\hat{f}_j(\mathbf{x}(i),\, \mathbf{u}(i)), \tag{13}$$

for $i = 0, 1, ..., N$, where N is the number of sampling intervals, $x_j(i) \equiv x_j(t_0 + iT)$, $T = (t_f - t_0)/N$. The equations (7) and (8) can be expressed as

$$u_j(i) = S_j(z_j(i)), \quad z_j(i) = w_{j0} + \sum_{k=1}^{n} w_{jk} x_k(i). \tag{14}$$

The performance criteria (10) implicitly depend on w_{jk} and w_{j0} (equations (14)), and we can use the gradient algorithm for minimization of the performance criteria according to weights and biases

$$w_{pq}^{(l+1)} = w_{pq}^{(l)} - \eta\frac{\partial J}{\partial w_{pq}^{(l)}}, \tag{15}$$

for $p = 1, ..., m$, $q = 0, ..., n$, $l = 1, ..., M$ where η is the learning coefficient which determines the speed of convergence and M is the number of iteration of the gradient algorithm. The gradient of the performance criteria is

$$\frac{\partial J}{\partial w_{pq}} = T\sum_{i=0}^{N-1}\left(\sum_{r=1}^{n}\frac{\partial F(i)}{\partial x_r(i)}\frac{\partial x_r(i)}{\partial w_{pq}} + \sum_{r=1}^{m}\frac{\partial F(i)}{\partial u_r(i)}\frac{\partial u_r(i)}{\partial w_{pq}}\right) + \sum_{r=1}^{n}\frac{\partial G(\mathbf{x}(N))}{\partial x_r(N)}\frac{\partial x_r(N)}{\partial w_{pq}}. \tag{16}$$

Further, obtained from (12) is

$$\frac{\partial x_r(i)}{\partial w_{pq}}=\sum_{j=1}^{n}\frac{\partial f_r(i-1)}{\partial x_j(i-1)}\frac{\partial x_j(i-1)}{\partial w_{pq}}+\sum_{j=1}^{m}\frac{\partial f_r(i-1)}{\partial u_j(i-1)}\frac{\partial u_j(i-1)}{\partial w_{pq}}, \tag{17}$$

and from (14) it follows that

$$\frac{\partial u_r(i)}{\partial w_{pq}}=D_r\left(z_r(i)\right)\frac{\partial z_r(i)}{\partial w_{pq}}=D_r\left(z_r(i)\right)\left[\frac{\partial w_{r0}}{\partial w_{pq}}+\sum_{j=1}^{n}\left(\frac{\partial w_{rj}}{\partial w_{pq}}x_j(i)+w_{rj}\frac{\partial x_j(i)}{\partial w_{pq}}\right)\right], \tag{18}$$

where $D_r\left(z_r(i)\right)$ is the derivation of function $S_r\left(z_r(i)\right)$

$$D_r\left(z_r(i)\right)=\begin{cases}0; & z_r(i)>u_j^{\max}\\ 1; & -u_j^{\max}\le z_r(i)\le u_j^{\max}\\ 0; & z_r(i)<-u_j^{\max}\end{cases} \tag{19}$$

($D_r(z_r(i))$ is shown in Fig. 2). Further, it follows that

$$\frac{\partial u_r(i)}{\partial w_{pq}}=D_r\left(z_r(i)\right)\left[\delta_{rp}\delta_{q0}+\sum_{j=1}^{n}\left(\delta_{rp}\delta_{qj}x_j(i)+w_{rj}\frac{\partial x_j(i)}{\partial w_{pq}}\right)\right], \tag{20}$$

where

$$\delta_{rp}=\begin{cases}1; r=p\\ 0; r\neq p\end{cases} \tag{21}$$

Finally, obtained is

$$\frac{\partial u_r(i)}{\partial w_{pq}}=D_r\left(z_r(i)\right)\left(\delta_{rp}\delta_{q0}+\xi_q\left(x_q(i)\right)\delta_{rp}+\sum_{j=1}^{n}w_{rj}\frac{\partial x_j(i)}{\partial w_{pq}}\right), \tag{22}$$

where

$$\xi_q(x_q(i))=\begin{cases}0; q=0\\ x_q(i); q>0\end{cases} \tag{23}$$

The equations (17) and (20) presents recurrent relations which we need for the gradient calculation of performance criteria. The next step is the determination of the initial condition for the above-mentioned recurrent relations. The initial condition of the state vector is independent weights w_{jk}, so that

$$\frac{\partial x_r(0)}{\partial w_{pq}}=0 \tag{24}$$

for $r=1,2,...,n$, $p=1,2,...,m$, $q=0,1,...,n$. From (24) and (22) obtained is the second set of initial condition

$$\frac{\partial u_r(0)}{\partial w_{pq}}=\left(\delta_{rp}\delta_{q0}+\xi(x_q(0))\delta_{rp}\right)D_r\left(z_r(0)\right), \tag{25}$$

for $r=1,2,...,m$, $p=1,2,...,m$, $q=0,1,...,n$.

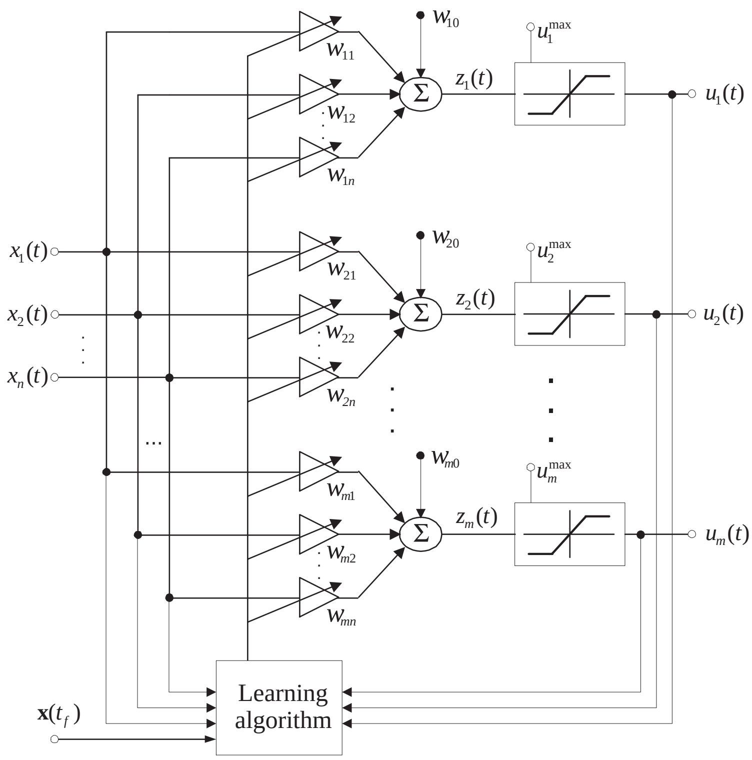

Figure 3: Architecture of the single-layer neurocontroller in the learning process.

3.2 Optimal feedback control with variable terminal time

For the problem of TOC, i.e. control in a situation which requires the minimization of the time control interval in accordance with the given performance criteria and constraints, we use the following heuristic algorithm, [9]

$$\tau^{(k+1)} = \tau^{(k)} - \Delta\tau \, H^{-} \left(G(\mathbf{x}(N)) - G_{\min} \right), \tag{26}$$

where

$$H^{-}(z) = \begin{cases} 0\,; & z \geq 0 \\ 1; & z < 0 \end{cases} \tag{27}$$

$\tau^{(k)}$ is the sampling interval in k-th iteration of the gradient algorithm, $G_{\min}$ is the measure of accuracy of the solution of the TOC problem for the given $\tau^{(k)}$, and $\Delta\tau$ is the constant of the sampling interval decrease.

In other words, the next iteration of sampling interval $\tau^{(k)}$ will ensue only when the value of function $G(\mathbf{x}(N))$ achieves the given value $G_{\min}$, and this means that $\tau^{(k)} > \tau_{\min}$. As $\tau^{(k)}$ decreases, $G(\mathbf{x}(N))$ converges at an increasingly slow pace toward $G_{\min}$ until it reaches a certain value of $\tau^{(k)}$ for which $G(\mathbf{x}(N))$ will not be able to reach $G_{\min}$ or will converge toward it very slowly, which means that $\tau^{(k)} < \tau_{\min}$. In that case, $\tau_{\min} \approx \tau^{(k-1)}$ is taken as an approximation for $\tau_{\min}$.

This algorithm structure guarantees stability and convergence toward $\tau_{\min}$, because it does not change the value of $\tau^{(k)}$ until the value of function $G(\mathbf{x}(N))$ falls below the given, sufficiently low value of $G_{\min}$.

This form of algorithm for the TOC problem enables a simple generalisation of the algorithm (15)-(17), (22) (with the given fixed terminal time) through the expansion of the algorithm with the expression (26).

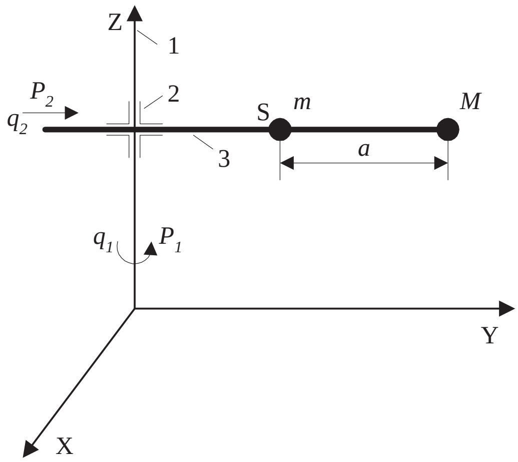

Figure 4: Robot with two degrees of freedom - rotation and translation.

4 Application of the algorithm on optimal robot control

This section presents the application of the derived algorithms for optimal feedback control and TOC to the minimum-time feedback control of the robot with two degrees of freedom.

4.1 Dynamics of the robot with two degrees of freedom

The non-linear dynamic model of the manipulator with two degrees of freedom, [10, 11] is presented through cylindrical coordinates in the form of

$$\begin{bmatrix} M_{11}(\mathbf{q}) & 0 \\ 0 & M_{22}(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} N_1(\mathbf{q},\dot{\mathbf{q}}) \\ N_2(\mathbf{q},\dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} P_1(t) \\ P_2(t) \end{bmatrix} \tag{28}$$

where

$$\begin{aligned} M_{11} &= I_1 + I_2 + (m+M)q_2^2 + 2Maq_2 + Ma^2, \\ M_{22} &= m + M, \\ N_1 &= 2[(m+M)q_2 + Ma]\dot{q}_1\dot{q}_2, \\ N_2 &= -[mq_2 + M(a+q_2)]\dot{q}_1^2, \end{aligned} \tag{29}$$

where $\mathbf{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T$ are cylindrical coordinates of the center of the mass of manipulator 3, (Fig. 4), M is the total mass (manipulator hand and load), m is the stick mass, J_1 is the total moment of inertia of sticks 1 and 2 in relation to axis Z, J_2 is the moment of inertia of stick 3 in relation to the axis which is parallel to axis Y and goes through point S, a is the distance between the center of mass M and point S. $P_1(t)$ stands for the control moment of rotation freedom of motion q_1, while $P_2(t)$ is the control force of the translation freedom of motion q_2.

The numerical values of the above-mentioned parameters are:

$M = 50kg$; $m = 97kg$; $J_1 + J_2 = 193kgm^2$; $a = 1.1m$; $P_{1\max} = 600Nm$; $P_{2\max} = 500N$, where $P_{1\max}$ is the maximum allowed moment and $P_{2\max}$ is the maximum allowed force. If the above-mentioned system of the second-order differential equations is to be transformed into the system of the first-order differential equations, the following coordinate transformation is introduced:

$q_1 = x_1; \dot{q}_1 = x_2; q_2 = x_3; \dot{q}_2 = x_4$; $P_1(t) = u_1(t)$; $P_2(t) = u_2(t)$,
and, for the sake of a more elegant expression, the following constants are introduced:

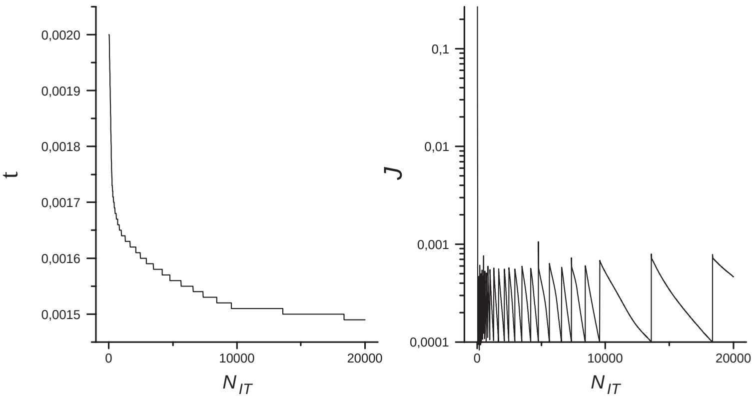

Figure 5: Dependence of τ and J on the number of iterations.

$A_1 = J_1 + J_2 + Ma^2$; $A_2 = 2Ma$; $A_3 = m + M$,
so that

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{-2A_3x_2x_3x_4 - A_2x_2x_4}{A_1 + A_2x_3 + A_3x_3^2} + \frac{u_1}{A_1 + A_2x_3 + A_3x_3^2} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= x_3x_2^2 - \frac{A_2}{2A_3}x_2^2 + \frac{u_2}{A_3} \end{aligned} \tag{30}$$

Here we have transformed the dynamics of the robot with two degrees of freedom into a system of four non-linear first-order differential equations.

4.2 Numerical results

Considered here is the minimum-time control problem, i.e. the problem of minimum time necessary for the robot to go from the initial to the final state with the given control vector constraints. In other words, one should determine weight parameters and biases for the transformation of the robot state from the initial one

$x_1(0) = 0\,rad$, $x_2(0) = 0\,rad \cdot s^{-1}$, $x_3(0) = 0\,m$, $x_4(0) = 0\,m \cdot s^{-1}$,
to the final state

$x_1(t_f) = \pi/2\,rad$, $x_2(t_f) = 0\,rad \cdot s^{-1}$, $x_3(t_f) = 1\,m$, $x_4(t_f) = 0\,m \cdot s^{-1}$,
for minimum time $t_{min} \equiv t_f$, with control constraints

$$\begin{aligned} \mid u_1(t) \mid &\leq u_{1max}, \\ \mid u_2(t) \mid &\leq u_{2max}, \end{aligned} \tag{31}$$

where $u_{1max} = 600\,N \cdot m$, $u_{2max} = 500\,N$.

The method of the conjugated gradient has been used here because of good convergence properties and algorithm stability which do not depend on the choice of penalty function coefficients. The values of the constants are $N = 1000$, $M = 20000$, $K = 1$, $\Delta\tau = 0.00001$, $G_{\min} = 0.0001$. As can be seen in Fig. 5, obtained is the minimum time $t_{\min} = 1.5s$, i.e. $\tau_{\min} = 0.0015s$. The results for that (fixed) time are obtained with the help of the algorithm (15)-(17), (22), with $M = 10000$. These results are shown in Fig. 6 - 9. Weight parameters and biases after $M = 10000$ iterations are

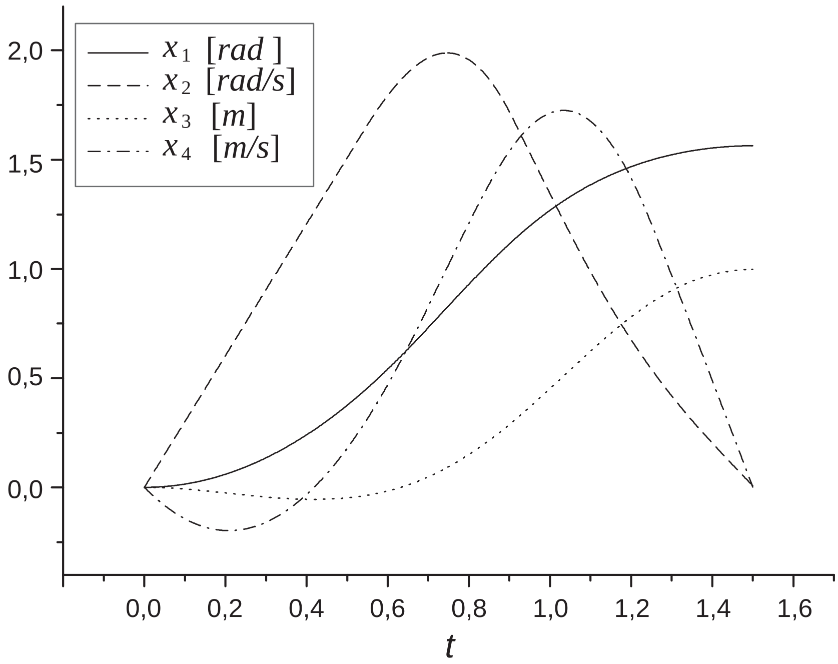

Figure 6: Time dependence of the state variables.

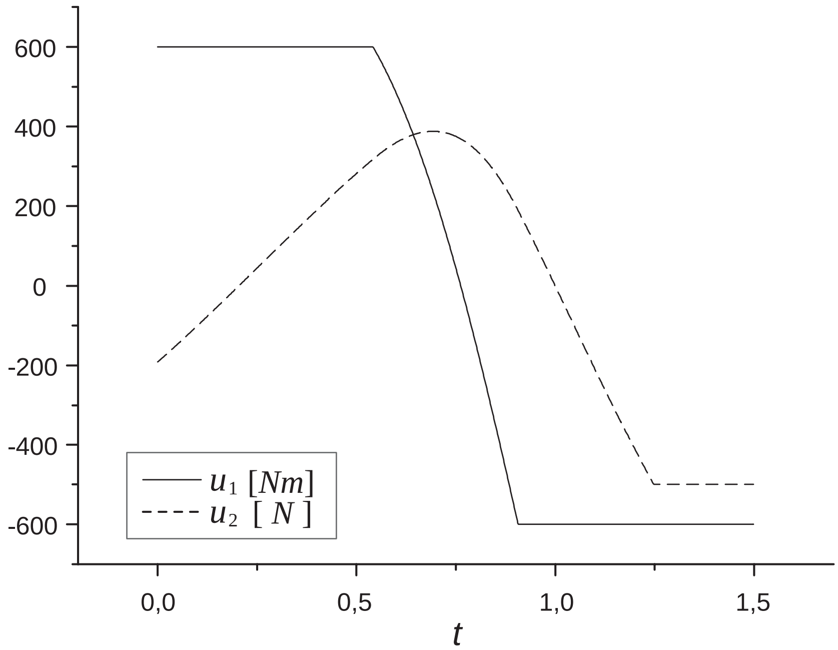

Figure 7: Time dependence of the control variables.

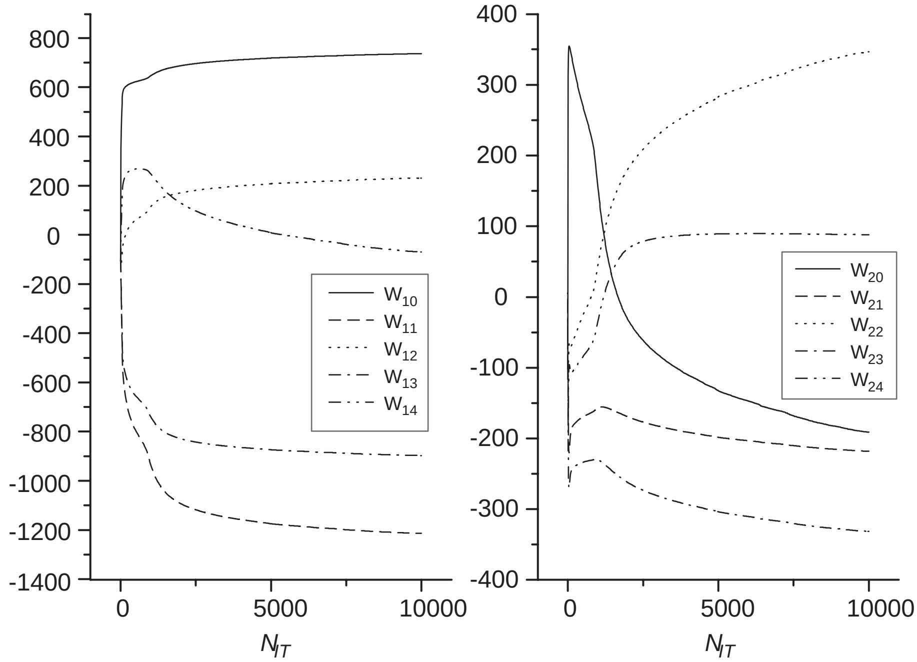

Figure 8: Dependence of the weight parameters on the number of iterations.

$$
\begin{array}{rclrcl}
w_{10} & = & 738.8, & w_{20} & = & -191.2, \\
w_{11} & = & -1211.4, & w_{21} & = & -218.1, \\
w_{12} & = & 233.5, & w_{22} & = & 346.6, \\
w_{13} & = & -894.1, & w_{23} & = & -331.5, \\
w_{14} & = & -67.4, & w_{24} & = & 88.0.
\end{array} \tag{32}
$$

5 Possibilities of on-line optimal regulation

The previous section address the feedback terminal control problem. In this section considered is the time-optimal regulation problem, i.e. retrieving the desired referent state for the minimum time.

We considered the example in the previous section, but with the exchanged initial and final state, i.e.

$$
\begin{array}{rclcrcl}
x_1(0) & = & \pi/2 & rad, & x_1(t_f) & = & 0 \quad rad, \\
x_2(0) & = & 0 & rad \cdot s^{-1}, & x_2(t_f) & = & 0 \quad rad \cdot s^{-1}, \\
x_3(0) & = & 1 & m, & x_3(t_f) & = & 0 \quad m, \\
x_4(0) & = & 0 & m \cdot s^{-1}, & x_4(t_f) & = & 0 \quad m \cdot s^{-1}.
\end{array} \tag{33}
$$

This problem may be considered as returning the robot hand in its initial position after the termination of the task. In Fig. 10 shown is the speed of convergence of the algorithm in this case. In comparison with Fig. 9. the speed of convergence in the regulation case is obviously higher, i.e. less iterations are needed for the same level of accuracy. The reason for this difference in the speed of convergence is the exponentially decreasing behaviour of the state vector in the regulation case.

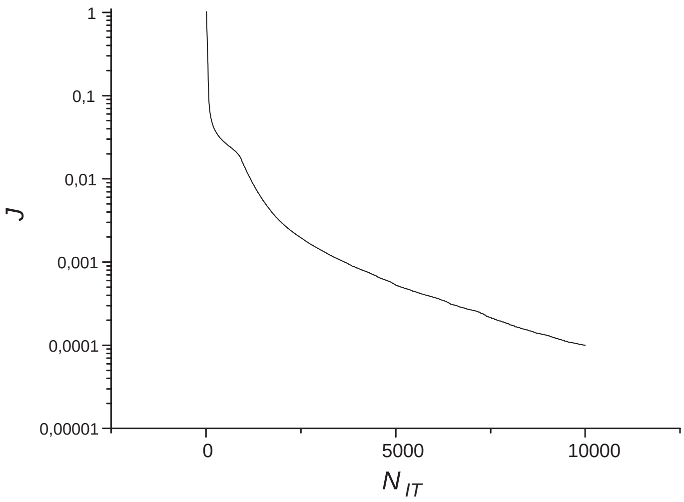

Figure 9: Dependence of J on the number of iterations.

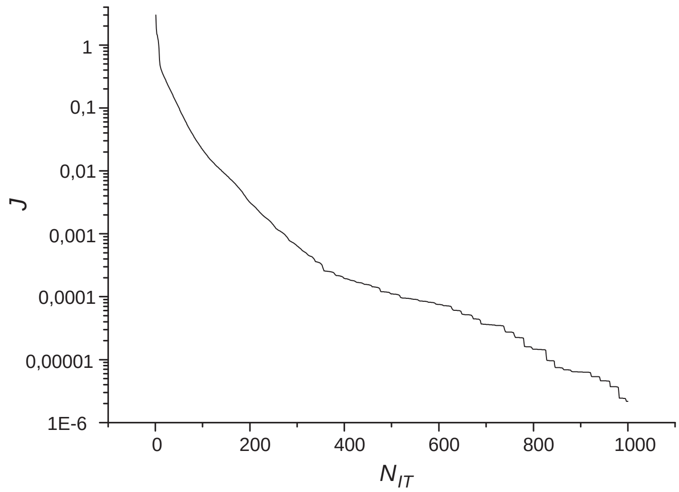

Figure 10: Dependence of J on the number of iterations for the on-line regulation.

Based on these properties we may consider the possibility of on-line optimal regulation which would be founded on the gradient (i.e. weight parameters) calculation during the sampling interval of the measuring signal. In Fig. 11. shown is the difference of the on-line solution and the off line solution (the solution with constant weight parameters for $M = N = 1000$).

We can see a small deviation which vanished at the end of the terminal time, which confirms the possibility of on-line optimal regulation. What remains to be determined is the required number of calculation during a sampling interval. On the basis of the algorithm (15)-(17), (22), obtained is the approximate number of calculations during the sampling interval

$$N_{OP} \approx N\left(n^3 m + 2n^2 m^2\right) \tag{34}$$

For $N = 1000$, $n = 4$, $m = 2$, obtained is $N_{OP} \approx 256000$ during the sampling interval $T = 0.0015s$, i.e. the required time for the execution of one instruction is $T_I = 5.8 \cdot 10^{-9}s$. For example, microprocessor TM5320C67x has such computational capabilities.

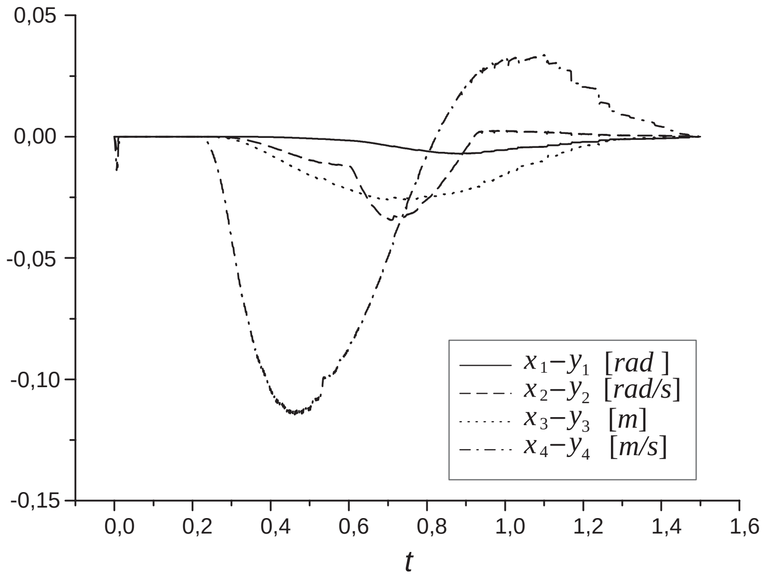

Figure 11: Differences of the off-line solutions and the on-line solutions.

6 Conclusion

The main goal of this paper is the research of possibilities of the one-layer neural network optimal feedback control of nonlinear systems with control constraints. Even though one-layer neural networks haven't general approximation capability, in the case of minimum-time robot control we obtain good result for the minimum time $t_{\min} = 1.5s$, the same as in feedforward optimal robot control, [9].

A good side of the one-layer neural network controller is a small number of weight parameters which have to be adjusted, which results in good convergence properties of the algorithm, especially for the optimal regulation which is possible to realise on-line. Furthermore, good convergence properties are the result of choosing the activation function in form of the saturation limiter function, which is how we avoid introducing additional penalty function for control vector constraints.

A next useful research step could be the application of a similar approach on the dynamic neural network controller which promise even better convergence properties. Furthermore, it would be interesting to research the considered problem with the state vector constraint introduced.

Appendix: Symbols

- a - distance between the center of mass M and point S, m
- J_1 - total moment of inertia of sticks 1 and 2 in relation to axis Z, kgm2
- J_2 - moment of inertia of sticks 3 in relation to the axis which is parallel to axis Y and goes through point S, kgm2
- K - coefficient of the penalty function, 1
- m - mass of stick, kg

- M - total mass (manipulator hand and load), kg
- $P_1(t)$ - control moment of rotation freedom of motion q_1, Nm
- $P_2(t)$ - control force of the translation freedom of motion q_2 , N
- $\mathbf{q}(t)$ - vector of generalised coordinates, rad, m
- $\dot{\mathbf{q}}(t)$ - vector of speed of generalised coordinates, rad/s, m/s
- t_0 - starting time-point, s
- t_f - final time-point, s
- T - constant sampling period, s
- $\mathbf{u}(t)$ - control vector, Nm, N
- w_{ij} - the elements of the weight matrix,
- $\mathbf{x}(t)$ - state vector, rad, rad/s, m, m/s
- η - constant coefficient of convergence, 1

References

[1] A. E. BRYSON, Y. C. HO, *Applied Optimal Control*, John Wiley, New York, 1969.

[2] K. S. NARENDRA, K. PARTHASARATHY, "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Trans. on Neural Networks, vol. 1, pp. 4-27, Mar 1990.

[3] E. S. PLUMER, "Optimal Control of Terminal Processes Using Neural Networks", IEEE Trans. on Neural Networks, vol. 7, no. 2, pp. 408-418, 1996.

[4] D. D. EGBERT and M. S. FADALI, "Time-optimal control with neural networks", IASTED Internat. Conf. on Robotic & Automation, Santa Barbara, May 25-27, 1988.

[5] P. J. WERBOS, "Backpropagation through time: What it does and how to do it", Proc. IEEE, vol. 78, pp. 1550-1560, 1990.

[6] D. E. RUMELHART and J. L. McCLELLAND, *Parallel Distributed Processing*, Vol. 1., MA: The MIT Press, 1986.

[7] S. W. PICHE, "Steepest Descent Algorithms for Neural Network Controllers and Filters", IEEE Trans. on Neural Networks, vol. 5, no. 2, pp. 198-212, 1994.

[8] F. BEAUFAYS and E. A. WAN, "Relating real-time backpropagation and backpropagation through time: An application of flow graph interreciprocity", Neural Computation, vol. 6, no. 2, 1994.

[9] J. KASAČ, *Optimal Control of Nonlinear Systems Using Neural Networks*, Master's of Science Thesis, University of Zagreb, Zagreb, 1998.

[10] B. HEIMAN, H. LOOSE, G. SCHUSTER, "Contribution to optimal control of an industrial robot", 4th CISM-IFTOMM Symp. on Theory and Pract. of Rob. And Manip., pp. 211-219, Zaborow, Poland, 1981.

[11] B. NOVAKOVIĆ, *Control Methods in Technical Systems: Applications in Robotics, Flexible Manufacturing Systems, and Processes*, Školska knjiga, Zagreb, 1990.