LEARNING ASYMMETRY EFFECT FOR THE NEURON NET CONTROL SYSTEMS

Alexander M. Astapkovitch, PhD

Head of Student Design Center, Saint-Petersburg State University of Aerospace Instrumentation, Saint-Petersburg, Russia
skb@aanet.ru, http://www.guap.ru

Abstract

The benchmark model of the differential drive mobile robot SOFA-2009 with the neuron net control system is described. Results of the supervised learning for different cases were compared for «moving to prescribed point of 2D surface» robot behavior under the control of two channel neuron net system. It was shown that the learning asymmetry effect exist for this type of behavior. Effect results in the asymmetry of the weight values of the symmetry control channels.

It was shown, that to avoid this effect the symmetrical learning set has to be used. One step learning procedure on base of Lagrange multipliers method is proposed, that provide possibility to take into account the existence of the linear relations for weights and avoid asymmetry effects also.

Key words: two wheel robot, benchmark model SOFA-2009, supervised learning, learning asymmetry effect, modified one step learning procedure, Lagrange multipliers method, computational robotics

INTRODUCTION

Computational Robotics methodology is supposed that the virtual robot models and the virtual environment models are used for research purposes.

SOFA-2009 is as simple as possible benchmark model of the two wheels differential drive mobile robot developed to support the student research project Phoenix-3 [1, 2]. The model reflects the most important features of the two wheels robot and consists from seven ordinary differential equations and includes description of the robot cinematic, dynamic, gears and motors.

For numerical experiments, described in this article, SOFA-2009 model was added with simplified neuron net control system. «Movement to prescribed point with slowing down» behavior was investigated. The one step supervised learning procedure was used, described in the details [3, 4]. In fact, it was shown, that very simple neuron net can be learned to solve optimal control problem of movement on 2D surface to prescribed position.

The results of learning for different samples have been compared and the existence of the learning asymmetry effect was demonstrated. This effect means that weights for two channel neuron net controller with ideal symmetry, received after supervised learning, will differ, if the asymmetry learning set was used.

It was shown also, that that symmetry solution could be received if the modified one step learning procedure is used, that takes into account the linear relations for weights. The proposed learning procedure implements the Lagrange multipliers method and it is the universal tool of the neuron net theory development.

1. BENCHMARK ROBOT MODEL SOFA–2009

SOFA is a virtual two wheels differential drive mobile robot. Every wheel is a rigid body disk with diameter $\mathbf{D_w}$. Distance between wheels is $\mathbf{L_r}$ and motors, gears, battery and control system are inside this space. Each wheel is driven independently with own motor. Motors have identical electromechanical structure. Robot forward/back motion is produced by both wheels been driven at the same rate. Turning is achieved by rotating the wheels with different rates.

SOFA kinematic model is illustrated by fig.1.1. At any given moment of the time SOFA moving is an arc with center at the $\mathbf{R_0(t)}$. The position of the robot center is described with vector $\mathbf{R_c}$ **(t)** and angle $\boldsymbol{\varphi}$**(t),** that is the robot axe inclination angle, measured from the axe x.

Robot movement is described with three ordinary differential equations concluded from basic assumption of SOFA model:

$$\varphi\,(t+\Delta t) - \varphi\,(t) = \Delta\,\varphi\ (\text{for left wheel}) = \Delta\,\varphi\ (\text{for right wheel}) \qquad (1.1)$$

$$R_0\,(t) = R_0\,(t+\Delta t) \qquad (1.2)$$

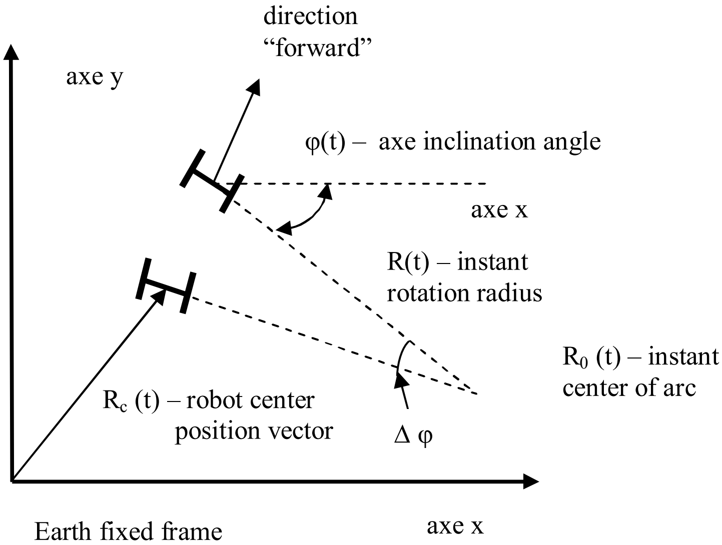


Fig. 1.1. SOFA kinematics model

Whole model is described with system of 7 ordinary differential equations (1.3–1.9). Gear for every wheel is described with equations 1.6 and 1.7 with free parameters – k_{12} , k_{21} Model takes into account moment of inertia of wheel J_r. Coefficients k_{11}, k_{21} were introduced to provide flexibility to SOFA model.

$$\begin{cases} \left.\frac{dR_c}{dt}\right|_x = \frac{D_w(\omega_1+\omega_2)}{4}\left[-\sin\varphi\right] \\ \left.\frac{dR_c}{dt}\right|_y = \frac{D_w(\omega_1+\omega_2)}{4}\left[\cos\varphi\right] \\ \frac{d\varphi}{dt} = \frac{D_w}{2Lr}\cdot(\omega_2-\omega_1) \\ \frac{d\omega_1}{dt} = \frac{-k_{11}}{J_r}\omega_1 + \frac{k_{12}}{J_r}I_1(t) \\ \frac{d\omega_2}{dt} = \frac{-k_{22}}{J_r}\omega_2 + \frac{k_{21}}{J_r}I_2(t) \\ \frac{dI_1}{dt} = -\frac{R_m}{L_m}I_1 - \frac{k_{13}}{L_m}\omega_1 + \frac{1}{L_m}U_1(t) \\ \frac{dI_2}{dt} = -\frac{R_m}{L_m}I_2 - \frac{k_{23}}{L_m}\omega_2 + \frac{1}{L_m}U_2(t), \end{cases} \quad (1.3 – 1.9)$$

Model SOFA-2009 is defined with parameters set : $D_w = 0.3$, $L_r = 0.5$, $J_r = 0.25$, $k_{11} = k_{22} = 75$, $k_{12} = k_{21} = 10$, Lm = 0.01, Rm = 0.1, $k_{13} = k_{23} = 1.5$ abs(Vmax)=12 .

SOFA-2009 model was realized and verified with Mathcad 14. The ODE solver function Rkadapt (.,.,) was used for robot dynamic simulation.

2. SIMPLIFIED NEURON NET CONTROL SYSTEM

To have the virtual robot valuable the model of the control system has to be added. The model depends on the research task. The simple two channels neuron net control system will be considered in this part.

Structure of the neuron net control system is presented in fig. 2.1. This system has 3 sensors and regulates the robot axe inclination angle. This system was learned to rotate robot on prescribed angle π/4 and slow down at final point with the one step supervised learning procedure [3]. Learning means that 6 weights (3 for every channel) have to be calculated.

Supervised learning procedure includes three steps: learning set establishing, weights calculation, control system quality estimates. The last step is the simulation of robot dynamic under neuron net control for different angle.

The same robot model SOFA-2009 was used for all steps with control system that are different for 1 and 3 steps.

The same robot model SOFA-2009 was used for all steps with control system that are different for 1 and 2, 3 steps.

During learning set establishing, the control voltage graphs have to be created ”by hand”. To get rotation behavior left and right motors have to rotate in different directions with the same speed. For angle as large as π/4 it is naturally to use the maximal voltages VMAX and the voltages applied to left and right motors have the different signs. Duration of pulses was selected iteratively in such way that wheels rotation speed have to be 0 when angle is reaching **π/4.** Control voltage graphs are presented by fig. 2.3(a) and graph 2.3(b) presents the robot behavior under such type of “virtual operator” control.

During second step the variant of the one step learning procedure in one channel approximation is used.

One step learning procedure is interpreted as task of the approximation for graph that is described as

$$
\begin{matrix}
S_1(T_1)\ S_2(T_1)\ \dots\ S_{Nsen}(T_1) & & w_1 & & A_1(T_1) \\
\dots & & \dots & & \dots \\
S_1(T_k)\ S_2(T_k)\ \dots\ S_{Nsen}(T_k) & * & w_2 & = & A_1(T_2) \\
\dots & & \dots & & \dots \\
S_1(T_{Np})\ S_2(T_{Np})\ \dots\ S_{Nsen}(T_{Np}) & & w_{Nsen} & & A_1(T_{Np})
\end{matrix} \qquad (2.1)
$$

where $\mathbf{S_i(T_k)}$ – **i**-sensor measurements at time moment $\mathbf{T_k}$, **Nsen** – number of sensor, $\mathbf{A_m(T_k)}$ – sample control voltage for **m-**actor at time moment $\mathbf{T_k}$, $\mathbf{w_i}$ – weight of the **i**-sensor, **Np** – number of time points in learning sample.

More complex possibilities of the one step learning procedure, included nonlinear approximations, are described in details in [3].

In matrix form system (2.1) is expressed as

$$\mathbf{S\ w = Ua} \qquad (2.2)$$

where **S –** rectangular matrix of sensor measurements **Np* Nsen**, **w** – weights vector with dimension **Nsen**, **Ua –** the learning sample control voltage vector has dimension **Np.**

In described example the two channels system is used and (2.2) has to be solved twice for different control voltage samples (left and right motors) and the same sensor matrix. In common case system (2.2) is the singular one. So, to get correct solution in accordance with Adamar definition the Tichonov regularization procedure has to be used [5].

Regularization procedure can be interpreted as linear programming problem for function **F(w)** of neuron net weights

$$\min_{\mathbf{w}} \mathbf{F(w) = (Sw - Ua, Sw – Ua)} + \gamma\ \mathbf{(w,w)} \qquad (2.3)$$

where γ – is regularization coefficient, that has a small value.

In this case learning procedure is the solution of the linear equations system and the weights can be calculated with expression

$$\mathbf{w} = (\mathbf{S}^T \mathbf{S} + \gamma\ \mathbf{E})^{-1} \mathbf{S}^T \mathbf{Ua} \qquad (2.\ 4)$$

where **E** – the units square matrix with dimension **Nsen*Nsen** .

Some results of the described example are presented in fig. 2.3. Three sensors were used (Nsen=3): sensor of difference between robot inclination angle at final point and current one, sensor of instant angler velocity and sensor of difference of prescribed angle velocity at final point and instant one. One hundred time point was used to form the learning sample (Np =100).

Results of the learning :

- weight vector for left channel W^t = [17.5, 0.5, -0.5];
- weight vector for right channel W^t = [-17.5, -0.5, 0.5].

Absolute values of the measurement are the same for second and third sensor, because it is supposed that at final point the angle velocity equal 0. So, learning procedure gives the same absolute value for those weights. In additional, every channel of control system has the identical parameters, so absolute values for the corresponding weights of different channel have to be the same in sense of the absolute value.

Results of preliminary estimation of the learning are presented at fig. 2.3(c). For estimation the matrix S was multiplied by weight vector for every channel to get a voltage that will be applied to motors. As it can be seen the calculated voltage accedes the maximal accumulator voltage VMAX.

To overcome this situation the actor neurons were introduced to control net structure as it can be seen from fig. 2.1. The actor neuron imposes limits on the output voltage that depends on parameters Vmax and Vmin.

Fig 2.3(d) presents the results of simulation of robot rotation for the angle $\boldsymbol{\pi/4}$ (the same value that was used for learning), under neuron net control with and without actor neuron. In the last case the actor neuron limits were Vmax =12V and Vmin =0V.

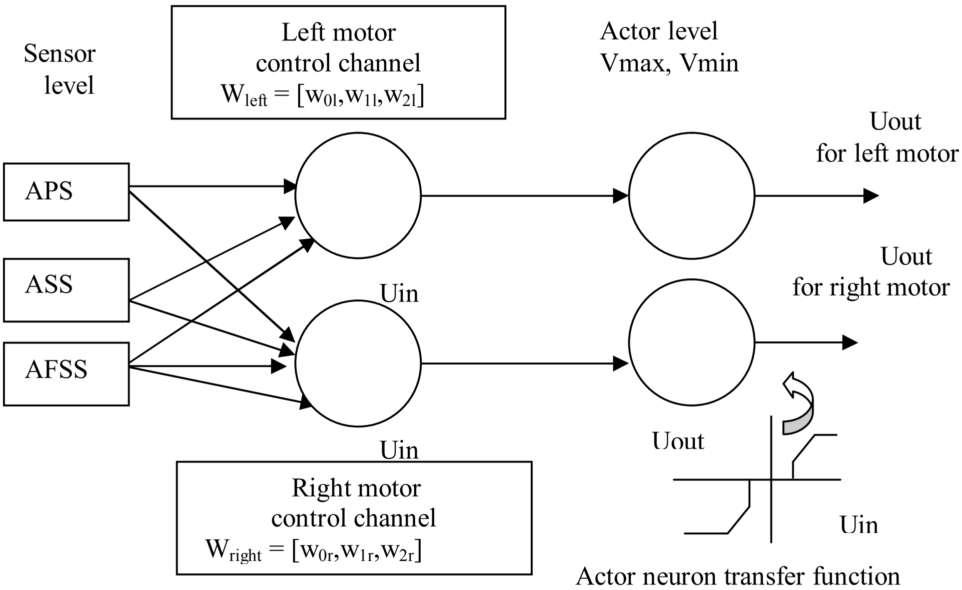


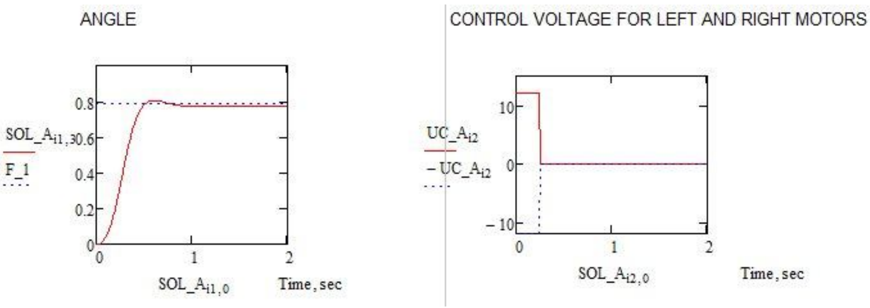
Fig. 2.1. Neuron net control system structure
APS –angle sensor, ASS –angle velocitysensor, AFSS – final angle velocity sensor

hird step is the estimation of quality of learning and the rotation for angle that differs from learning sample value. The rotation under neuron net control on the angle $\boldsymbol{\pi}$ is presented by fig. 2.4. The rotations were simulated for case s with and without voltage limits.

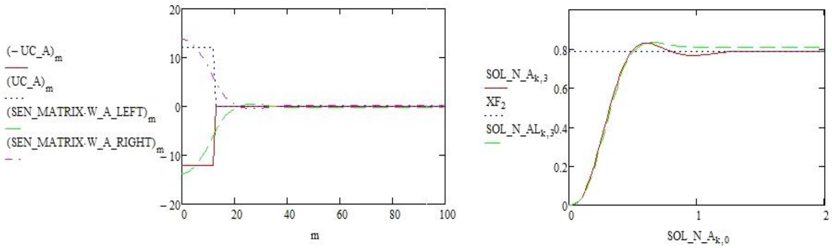
As it can be seen from presented graphs neuron net can solve this problem and robot reaches the prescribed point in the phase space. Under limits on maximal voltage robot have accelerated during initial phase and have slowed down during final one.

From this simple example of the learning some conclusion can be made. Certainly, actor neuron is a solution, but universal approach has to include description of such type limits in learning procedure.

Examples of the numerical experiments with SOFA-2009 model are described in [1, 2] and can be found on the site [6].

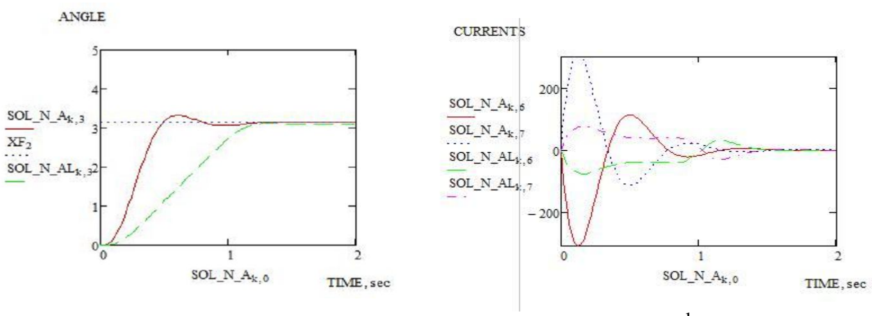


a b

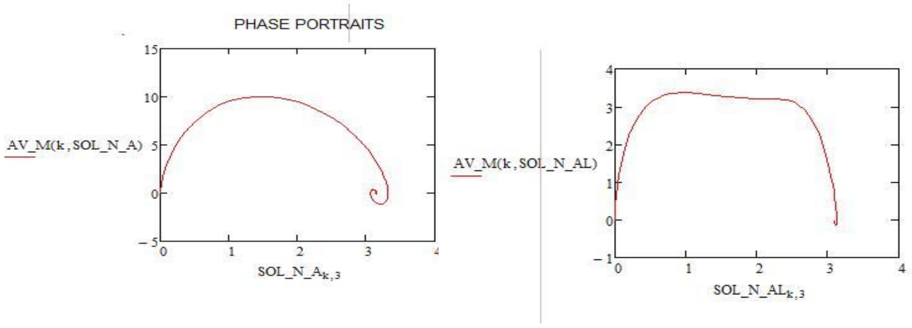


c d

Fig. 2. 3 Simple neuron net control … n learning
a)Learning sample "rotation to left on … nd slowing down "; b) Sample control voltage; (… ning quality preliminary estimation; d) Robot with neuron net control system rotation on π/4



a b



c d

Fig. 2.4 Rotation to left on π with slow down at final point
a) Angle for rotation with and without limit on Vmax b) Motor currents c) Phase portrait for system without limits on Vmax (initial point (0,0), final (3.14,0)) d) Phase portrait for system with limits on Vmax (initial point (0,0), final (3.14,0)

3. LEARNING ASYMMETRY EFFECT

The described above approach was used to investigate the simple neuron net control system of the robot SOFA-2009, that has to be learned to reach the prescribed point of 2D plane. The robot control system has the same structure as it presented in Fig. 2.1, but number of sensors is increased up to 9. Every channel uses the same sensor set, that are described with table 3.1.

In table 3.2 the weights for left and right channel are presented. The weights are results of the learning with the different learning samples. The initial position of the robot was the same for different learning samples, namely, robot is motionless at point (0,0) and is looking forward along axe Y (axe inclinations angle between robot axe and axe X is 0).

First two samples were used to for learning to simple behavior: rotation to left and moving forward. Third, forth and fifth samples were used to learn for more complex behavior – to reach the prescribed point of 2D plane where robot has to be slowed down and has the prescribed angle.

The presented in table 3.2 table data for sample 3 demonstrates the existence of the learning asymmetry effect. The effect leads to situation that asymmetry of learning sample is translated to weights. Because the virtual robot has the symmetrical control channels then a difference in the weights means that the reason is the learning sample asymmetry. It means that robot behavior will differ for moving to points that are situated in left and in the right half planes.

Table 3.1

Sensor description table

Name	Sensor description
DF	Distance to final point robot center position from instant position **D(F)-D(Rc(t))**
DV	Instant linear velocity **(w1(t)+ w2(t))/2**
AF	Difference between the robot inclination angle at the final point and the instant one **φ (F)- φ (t)**
AV	Instant robot axe inclination angle rotation speed **D_w/2*Lr*(w2(t)- w1(t))**
AVF	Difference between the robot angle speed at the final point and the instant one $\left.\frac{d\varphi}{dt}\right\|_{(F)}$ **- D_w/2*Lr*(w2(t)- w1(t))**
W1	Instant rotation speed for left wheel **w1(t)**
W1F	Difference between rotation speed at the final point for left wheel and the instant one **w1(F)- w1(t)**
W2	Instant rotation speed for right wheel **w2(t)**
W2F	Difference between rotation speed at the final point for right wheel and the instant one **w2(F)- w1(t)**

Note: *F* denotes the final point and D (*) is a function that calculates the distance from instant point to start one

Table 3.2

Weights for different learning samples

	S C	DF	DV	AF	AV	AVF	W1	W1F	W2	W2F
1	L	0.0	0.0	-17.5	0.093	-0.093	-0.155	0.155	0.155	-0.155
	R	0.0	0.0	17.5	-0.093	0.093	0.155	- 0.155	- 0.155	0.155
2	L	0.59	0. 382	0.0	0.0	0.0	0. 382	-0. 382	0. 382	-0. 382
	R	0.59	0. 382	0.0	0.0	0.0	0. 382	-0. 382	0. 382	-0. 382
3	L	0.239	0. 422	-18.5	0.077	-0.077	0.294	-0.294	0.55	-0.55
	R	0.251	0. 416	16.0	-0.143	0.143	0.654	-0.654	0.178	-0.178
4	L	0.196	0. 424	-17. 3	0.109	-0.109	0.243	-0.243	0.605	-0.605
	R	0.196	0. 424	17. 3	-0.109	0.109	0.605	-0.605	0.243	-0.243
5	L	0.196	0. 424	-17. 3	0.109	-0.109	0.243	-0.243	0.605	-0.605
	R	0.196	0. 424	17. 3	-0.109	0.109	0.605	-0.605	0.243	-0.243

1. Rotation in place on π/ 4 to left.
2. Moving from the point (0.0) to the point(- 4,4).
3. Rotation in place on π/ 4 to left and moving from point (0.0) to the point (- 4,4).
4. Rotation in place on π/ 4 to left and moving from point (0.0) to the point (- 4,4) and rotation in place on π/ 4 to right and moving from point (0.0) to the point (4,4).
5. Rotation in place on π/ 4 to left and moving from point (0.0) to the point (- 4,4). Robot was Learned with using modified learning procedure with Lagrange multipliers to provide the symmetry for weights absolute meanings

Two possible solutions were investigated to get a symmetry weights values, that are described as cases 4 and 5.

For 4 case from table 3.2 the learning procedure used the complex sample that was formed from two samples: moving to point in left half plane (-4,4) and moving to symmetry point in right half plane (4,4). As it can be seen from table 3.2 for this learning the weights demonstrate the symmetry.

In fact, this result is the confirmation for the learning asymmetry effect.

It can be pointed out, that one step learning procedure (2.1) – (2.4) imposes no limits on number of learning samples. To learn the real robot in accordance with this methodology it will be necessary some sets of the experimental data (sensors and actors graphs), received from the operator control moving. For more complex situation with increased number of the freedom of the possible robot movements the forming the symmetry learning set can be not trivial problem.

As one more solution to avoid the learning asymmetry effect, it is possible to use the modified one step learning procedure. Main idea is in the using the Lagrange multipliers method to implement the explicit relations between the weights. This approach gives possibility to solve problems, related with learning asymmetry effects. The results of learning with modified one step procedure are presented in table 3.2 and demonstrate the workability of the modified procedure.

The modified one step learning procedure has the own value and will be described in the next part.

For real robot the situation is more complex. It is clear, also, that the experimental data will put on weight the experimental asymmetry effect. It means, that the learning asymmetry effect and the experimental asymmetry effect influences have to be separated from the real asymmetry of control channels

4. MODIFIED ONE STEP LEARNING PROCEDURE

The one possible way to solve asymmetry problem is using the description of the ties between the weights in explicit form. For learning asymmetry there are exist linear relation between the same weights in the different channels, so it is possible to use Lagrange multiplier method.

To realize this approach it is necessary to use the more common formulation for problem (2.1) – (2.3), that takes into account mutual influence of channel. The simplified form of modified one step learning procedure is described below.

Let W_k is the weight vector for k-th control channel

$$\mathbf{W_k = [\, w_{k1},\ w_{k2},\ \ldots\ldots\ w_{k\,Nsen}\,]^T} \qquad (4.1),$$

where Nsen – number of sensors that are used to produce control voltage.

In this case the weight vector for whole system can be expressed as

$$\mathbf{W = [\, W_1\ W_2\ W_k\ W_k\ W_{Nc}\,]^T} \qquad (4.2),$$

where Nc – number of the control channels in whole system.

Actor vector can be expressed with the same manner

$$\mathbf{Ua_k = [\, Ua_{k1}\ \ Ua_{k2}\ \ldots\ldots\ Ua_{k\,Np}\,]^T} \qquad (4.3),$$

$$\mathbf{Ua = [\, Ua_1\ Ua_2\ \ldots\ \ Ua_{Nc}\,]^T} \qquad (4.4).$$

where Np – is number of the time points in the learning sample.

Learning asymmetry problem can be solved if one take into account the two type of linear relations between weights

$$\mathbf{w_{kj} = w_{mj}} \text{ or } \mathbf{w_{kj} = - w_{mj}} \qquad (4.5),$$

where maximal value for i is equal to number of sensors (weights), that are used in the every control channel. For one layer neuron net this number is Nsen.

In common way the set of this relation can be presented as

$$\mathbf{L\,W = b} \qquad (4.6),$$

where **L** – rectangular matrix.

With the introduced above vectors Lagrange multipliers method can be formulated as linear programming optimization problem

$$\mathbf{\min_{W,\mu} F(W) = (SW - Ua, SW - Ua) + \gamma\,(W,W) + D_{\mu}(LW-b)} \qquad (4.7)$$

where D_μ – diagonal matrix, formed from Lagrange multipliers μ_i for every relation (4.5) or (4.6).

The elegant form of the one step learning procedure exists, if the modified vectors **W** and **Ua** are used to rewrite (4.7).

Let **μ** is the vector that is formed from Lagrange multiplier

$$\mathbf{\mu = [\mu_1,\ \mu_2,\ \ldots\ldots\ \mu_{Nsen}\,]^T} \qquad (4.8)$$

and let us introduce the modified vector of the independent variables for problem (4.7)

$$\mathbf{W\mu = [\, W_1\ W_2\ W_k\ W_k\ W_{Nc}\ \mu]^T} \qquad (4.9),$$

that includes this vector μ. Vector **U**a has to be modified with the same manner and the resulted vector will denote as **Uμ**. For the describing case this vector is formed from vector **Ua** and the added vector **b** from (4.6).

With this vectors the one step learning procedure can be expressed as

$$\mathbf{W\mu = (S\mu^T S\mu + \gamma\, E\mu)^{-1}\, S\mu^T\ \ U\mu} \qquad (4.10).$$

This method was tested for the effect of asymmetry learning problem, described above. The results of the learning with (4.11) are presented in the table 3.2 (learning number 5).

As it is described above the system has two symmetrical channels and the every of each uses the same sensor set. For this simple case the special formulation for Lagrange method can be proposed, that takes into account the special structure of L.

For this case matrixes Sμ and Eμ have the structures

Sμ	=	$S^T S$	0	E
		0	$S^T S$	E1
		E	E1	0
Eμ	=	γ E	0	0
		0	γ E	0
		0	0	0

In this form, **E** and **E1** are the blocs of the matrix **L**,

$$\mathbf{L = [E\ \ E1]} \qquad (4.11),$$

that describes the whole set of the symetry relations (4.7) in form

$$\mathbf{L\,W = 0} \qquad (4.12).$$

The first bloc **E** is the diagonal matrix, with units for every relation (4.5), (4.6). Block **E1** is the sparse matrix with elements: 0, 1 (weights have the equal absolute values and the different signs), -1 (weights have the equal absolute values and the same signs). Vector **Uμ** is formed from vector **Ua** and vector **b** from (4.6) and it is the vector with zero elements.

This variant of the modified one step learning procedure was realized with Mathcad 14 and used to investigate the asymmetry learning effect. The results are presented in table 3.2, that confirms the workability of proposed approach.

5. CONCLUSION

As it seems, that the using virtual robot model SOFA-2009 have demonstrated the effectiveness for researching and developing the theory of control system on base of neuron nets. It was discovered in theory that learning asymmetry effect has to exist. It is clear, that for real robot the experimental asymmetry effect has to exist also.

It means, that the learning asymmetry effect and the experimental asymmetry effect influences have to be separated from the real asymmetry of control channels. Some additional investigations have to be done as the next step in this direction.

The received results pointed that subclass of the neuron net control system with linear ties for variables have to be studied as from point of the building the new learning procedures, also as possibility to build the new methods of applied mathematics. It can be stated at least, that the formulation of the learning procedure as linear programming problems with limits (4.7) increases number of tools that can be used researches and engineers who work with control system on neuron net base.

Any interested person can find Mathcad 14 problem set, that includes the SOFA-2009 model with neuron net control system and one step learning procedure, on site guap.ru > student design center>project SOFA-2009.

REFERENCES

[1] Alexander M.Astapkovitch Virtual mobile robot SOFA-2009

[2] Proc. International forum “Information and communication technologies and higher education - prioriries of modern society development”, p.7-15,SUAI Saint-Petersburg,2009

[3] A.Burdukov Autonomous robot "Phoenix-3"

[4] Proc. International forum “Information and communication technologies and higher education - prioriries of modern society development”, p.41-47,SUAI Saint-Petersburg,2009

[5] Astapkovitch A.M. One step learning procedure for neural net control system. Proc. International forum “Information systems. Problems, perspectives , innovation approaches” „ p.3-9,SUAI Saint-Petersburg,2007

[6] Д.А.Астапкович, А.А.Гончаров, А.С.Дмитриев, А.В.Михеев. Проект “Робот Феникс-1”. Сборник докладов пятьдесят девятой международной студенческой научно-технической конференции ГУАП., Часть I, Технические науки, 18-22 апреля 2006, г. Санкт-Петербург, с.211-216.

[7] Тихонов А.Н, Арсенин В.Я. Методы решения некорректных задач, М., Наука, главная редакция физико-математической литературы, 1979, с. 286.

[8] http://guap.ru/guap/skb