CHAPTER 5

Results

Performance Surface

Following the scheme shown in second chapter (Fig 2.12 & 2.13), the performance surface of the derived network (Fig. 3.39) is first empirically obtained for individual adaptive weight pairs (w_{O3} & w_{O4} with weight step, $\Delta w = 0.1$ between 4.5 to 6.5 and $\Delta w = 0.25$ else) as shown in Figures 5.1, 5.2 and 5.3. The performance surface is obtained with the E-N receiving B and S-stimulus but no D-stimulus. Thus, also called conditioned performance surface. Note that w_{O3} is the feeding field weight for ENU's in Eck3 (network end with additional drive stimulus) at the receiving end of sensory stimulus while w_{O4} is the adaptive weight for Eck4 ENU's (network end with just bias stimulus). The performance index is defined by

$$P = \left(\text{Desired}_{GN} - \text{Transformed}_{EN}\right)^2 \qquad (1).$$

Starting from initial weights, $w_{O3} = 0$ and $w_{O4} = 0$, the surface remains flat at maximum P but with increasing weight values the surface has regions of local minima (arrow head, Fig. 5.1) and a global minima (arrow, Fig. 5.1). With further increase in weight values the surface climbs, eventually reaching maximum P when adaptive weight for the dipole channel with inhibitory connection to the M-node increases beyond a certain value ($w_{O4} \geq 4.9$). It should be noted that beyond a certain value of adaptive weights (either w_{O3} or $w_{O4} > 8.4$) ENU's in Eck3 or Eck4 gets into saturated (unwanted) firing mode thus, causing loss of functional performance.

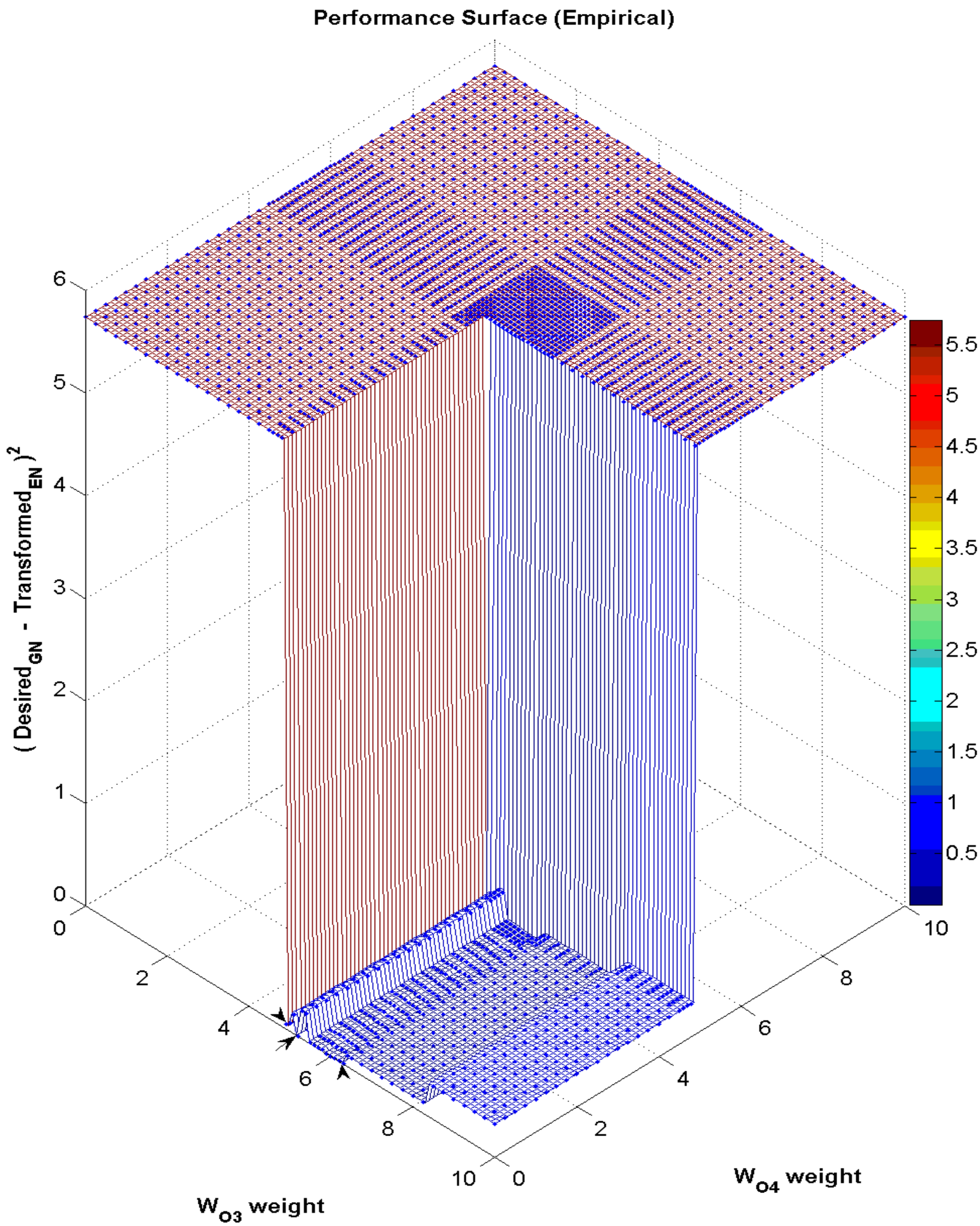

Figure 5.1. Performance surface (conditioned) of Eckhorn network obtained empirically by manually (weight step, $\Delta w = 0.1$ between 4.5 to 6.5 and $\Delta w = 0.25$ else) adjusting the adaptive weight pairs (w_{O3} & w_{O4}). Blue dots on the performance represent data points. Weight w_{O3} is the adaptive weight in ENU's of Eck3 while weight w_{O4} is for ENU's of Eck4. A global minimum (arrow) is situated between the two local minima (arrowheads). Note that for the E-DN in this particular E-N, Eck3 is on the channel with excitatory connection to M-node while Eck4 is on the channel with inhibitory connection.

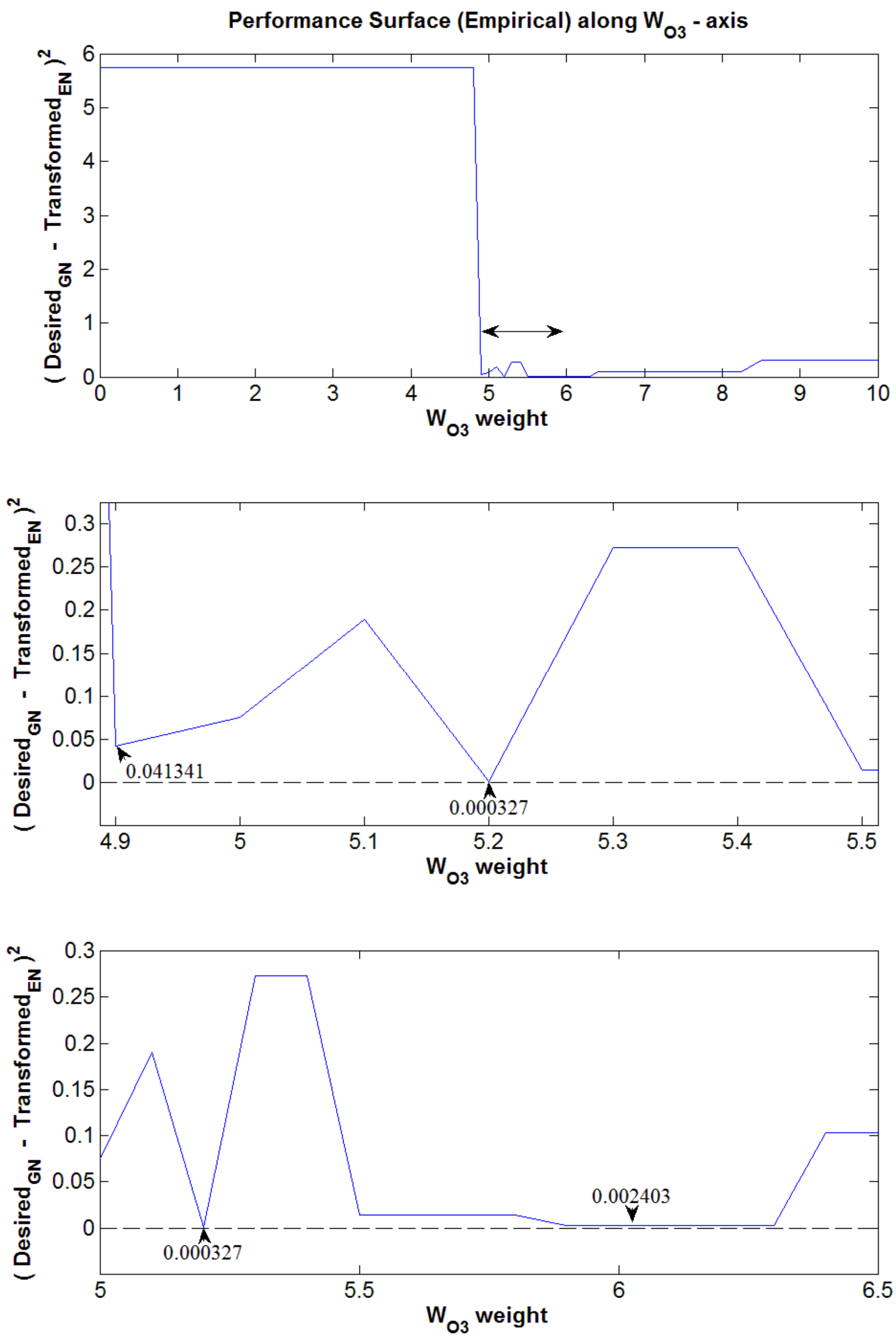

Figure 5.2. Side views along w_{O3} weight axes of the 3D-performance surface (Fig.5.1). Top: Performance index (P) decreases to local minima, global minima, local minima and then climbs such that beyond a certain w_{O3} value (>8.4) the surface becomes non P-selective. The regions of minima (double arrow) are enlarged in bottom two sub-figures. Middle: Shows the first local minima and the global minima with their respective P values. Bottom: Shows the global minima and second local minima with respective P values. The dashed horizontal line in lower two figures indicates P = 0 (for reference).

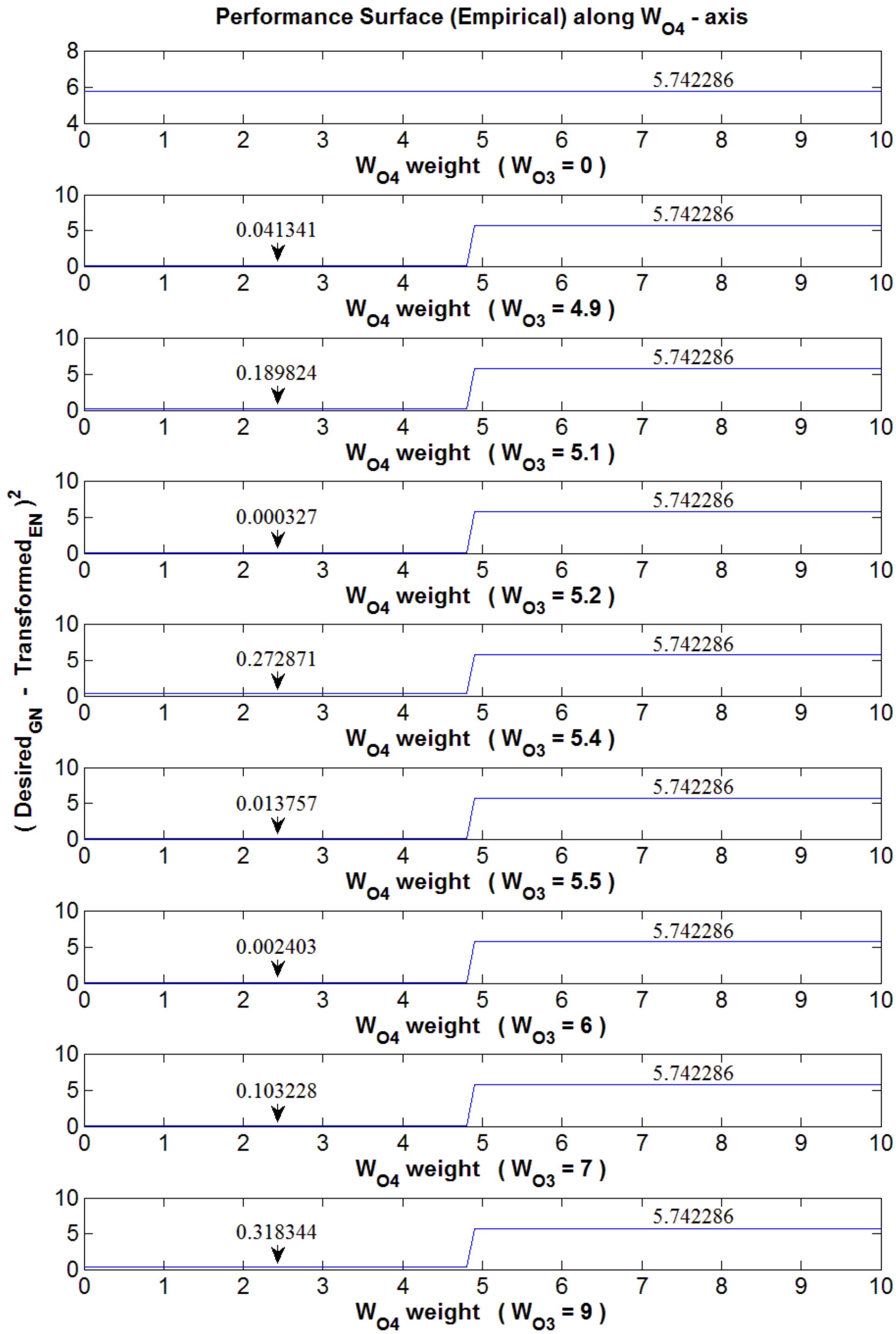

Figure 5.3. Side views along w_{O4} weight axes at various slices (different fixed w_{O3} values) of the 3D-performance surface (Fig.5.1). The numbered arrows give respective P values. The 2nd row figure ($w_{O3} = 4.9$) is the slicing at the region of first local minima (first arrow head, Fig. 5.1) within a certain range of w_{O4} beyond which P increases to its maximum value (P = 5.742286). Similarly, the 4th row figure ($w_{O3} = 5.2$) is the slicing at global minima (arrow, Fig. 5.1) and 7th row figure ($w_{O3} = 6$) the slicing at second local minima (second arrow head, Fig. 5.1).

Adaptation Algorithm

Though the global minimum is located between local minima, the minima values are very close to each other. For instance, difference between the global minimum and the larger local minima is of the order 10^{-2}. Thus adaptive weight values within the region of minima do not cause any practical difference whether it is local minima or global minimum. Under the set membership paradigm for adaptive systems, these miniscule differences place the minimum points within the same solution set [Zadeh 1963, Combettes 1993]. Thus steepest descent method can be implemented around the region of minima. However a gradient method cannot be used in regions with constant gradient and also between regions with different constant (gradient) regions due to sudden (steep) changes between them. Therefore, the principle of gain scheduling is implemented for performance outside the region of the minima. The performance surface is therefore divided into two regions:

- Region 1 (R1) when $|\text{Gradient}| \leq G_{\text{Minimum}}$, region of flat spot,
- Region 2 (R2) when $|\text{Gradient}| > G_{\text{Minimum}}$.

This checking for region of flat spot is given by the scalar,

$$\text{Gradient} = \frac{P(w_{O3}+\delta,\ w_{O4}+\delta) - P(w_{O3}-\delta,\ w_{O4}-\delta)}{2 \bullet \delta \bullet \sqrt{2}} \qquad (2)$$

where, performance index (P) is measured for both weight perturbation ($\delta = 5/10^3$).

The adaptation for the network is such that the adaptive procedure in R1 pushes the weights (w_{O3} & w_{O4}) a constant amount. The purpose of this adaptive procedure is to push the adaptive weights until performance gets to R2, where steepest decent method can be used.

The push-weight procedure is given by

$$W_{k+1} = W_k + Push_{Uniform} + Push_{Extra} \quad (3)$$

where, $W = \begin{bmatrix} w_{O3} \\ w_{O4} \end{bmatrix}$,

$Push_{Uniform} = \begin{bmatrix} c \\ c \end{bmatrix}$ and

$$Push_{Extra} = \begin{bmatrix} b \bullet (Conn_{5,M} - Conn_{4,M})_0^1 \\ b \bullet (Conn_{4,M} - Conn_{5,M})_0^1 \end{bmatrix}$$

such that the extra push parameter, b is determined by the connection function,

$$Conn_{i,M} = \begin{cases} +1, \text{ } if \text{ excitatory} \\ -1, \text{ } if \text{ inhibitory} \end{cases}$$

where *i* is the node end (Eck*i*) of the E-DN channel connected to M-node. For the reception of b by the adaptive weight in question, connection function of the opposing channel is subtracted from the function of channel carrying the adaptive weight whose outcome is given by the Heaviside step function,

$$(S)_0^1 = \begin{cases} 1, \text{ } if \text{ } S > 0 \\ 0, \text{ } if \text{ } S \le 0 \end{cases}.$$

Therefore the adaptive weight of the channel with excitatory connection receives b and hence $Push_{Extra}$. The need for $Push_{Extra}$ in the push-weight procedure arises because with just $Push_{Uniform}$, the increasing adaptive weight (wo3 & wo4) due to the push could pass diagonally across the performance surface (Fig. 5.1) missing the region of minima.

This problem of determining b and of invoking $Push_{Extra}$ is closely related to the "context-dependent choice" problem [Grossberg 1978]. That is, the same sensory cue can result in different responses depending upon the context. Here, the word context is used to mean events or processes (physical & mental) characteristic of a particular situation which has a behavioral impact [Reber 2001]. Grossberg articulated some possible approached to context-dependant choice. Figure 5.4 illustrates one of these [Wells 2011b].

The E-DN shown in this thesis is just part of a map or network system. Thus the connection types (excitatory and inhibitory) from E-DN to M-node occur in complementary pairs [James 1980, Plutchik 1980]. In other words, if for a particular context, the connection of E-DN channel with M-node is excitatory than in another context the connection for the same channel may be inhibitory (Fig. 5.4).

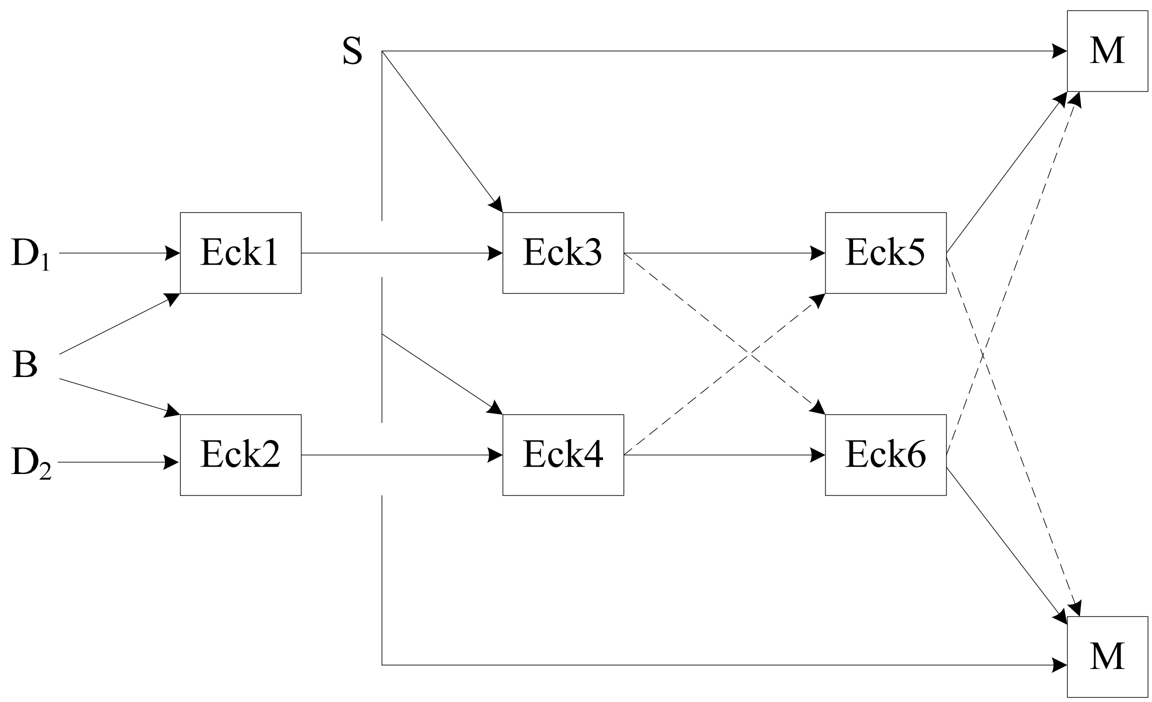

Figure 5.4. An architectonic solution to the "context-dependant choice". The figure shows contradictory unconditioned stimulus (D_1 & D_2) resulting in respective responses (M-node output). Thus adaptation of the network is dependent on which of the two dipole channels receive unconditioned stimulus. Solid lines represent excitatory connection while dashed lines represent inhibition.

During the learning stage, E0N receives a bias, drive and sensory stimulus. Thus Eck5 and Eck6 spiking will differ during dual-stimuli. An alternative $Push_{Extra}$ equation is

$$Push_{Extra} = \begin{bmatrix} b \bullet (Spikes_{ENU3} - Spikes_{ENU4})_0^1 \\ b \bullet (Spikes_{ENU4} - Spikes_{ENU3})_0^1 \end{bmatrix}$$

where, $Spikes_{ENUx}$ is the total spike output from the respective ENU node (EckX) during adaptation conditions (that is, when the E-N receives a bias, drive and sensory stimulus). Thus the $Push_{Extra}$ parameter follows a Hebbian rule. Either of the above $Push_{Extra}$ equations can be used in the push-weight procedure.

Finally, if the performance is located in region R2 on the performance surface, adaptation procedure switches over to steepest descent method which is given by

$$W_{k+1} = W_k + r \bullet \hat{\nabla} \qquad (4)$$

such that rate r = $5/10^6$ and the gradient estimate is

$$\hat{\nabla} = \begin{bmatrix} \dfrac{P(w_{O3}+\delta) - P(w_{O3}-\delta)}{2 \bullet \delta} \\ \dfrac{P(w_{O4}+\delta) - P(w_{O4}-\delta)}{2 \bullet \delta} \end{bmatrix}$$

where performance index (P) is measured for individual weight perturbations ($\delta = 5/10^3$). Since flat region R1 can also occur in region of minima triggering the push-weight procedure, an increase for P following the procedure is checked. Therefore, no P increase means it is not in region of minima. However, for P in region of minima, adaptive weights are set to the values responsible for P minima. That is, set to the weights prior to P increase. Figure 5.5 shows the flowchart implementing the adaptive procedure, i.e., adaptation algorithm for the Eckhorn network.

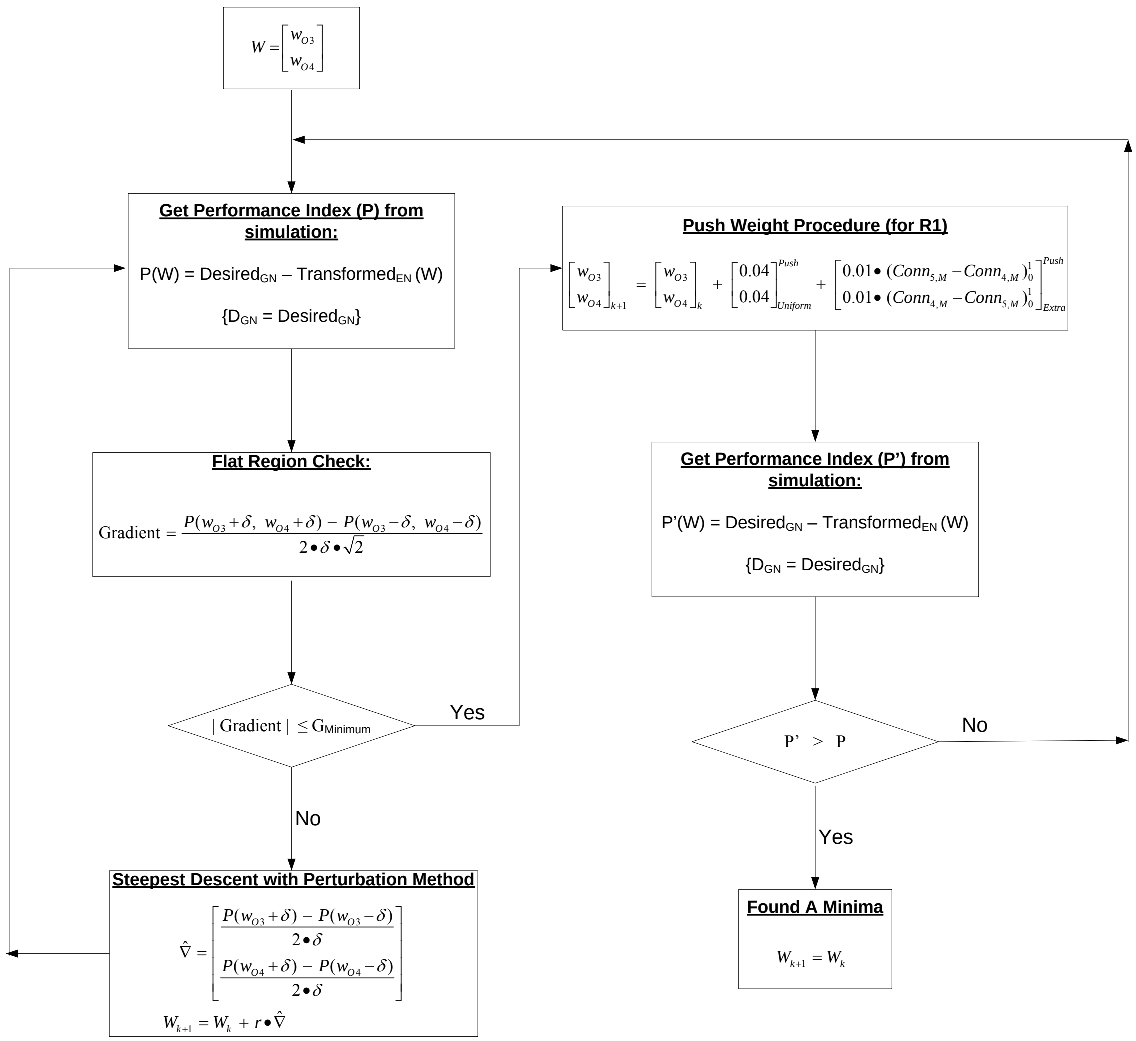

Figure 5.5. Flowchart showing the adaptive algorithm for the Eckhorn network. First, P is obtained from the initial weight inputs (W) which is then used for calculating the gradient for flat region check by simultaneously perturbing both the adaptive weights. Depending upon where P is located on the performance surface, flat region or not, either the Push-Weight procedure or the steepest descent method is chosen to update W (W_{k+1}) for the next weight input. In region of minima the P' caused by the push-weight is compared with P. Thus in minima, P' > P and algorithm is stopped with adaptive weights, $W_{k+1} = W_k$. Note that the parameters for Push-Weight procedures $Push_{Uniform}$ & $Push_{Extra}$ are different.

Learning Curve and Weight Curves

The performance surface of E-N during conditioning (i.e., with B, D and S-stimulus) is different from conditioned performance surface (Fig. 5.6). Though the later performance surface (Fig. 5.6a) is what matters for the adapted weights, the dynamics of the adapting weights depends on the former performance surface (Fig. 5.6b). That is, during adaptation and hence during conditioning the P used for estimating the gradient for flat region check and $\hat{\nabla}$ for the steepest-descent method (Fig. 5.5) is based on the conditioning performance surface (Fig. 5.6b). However, the P of the E-N with the adapted weight is evaluated against the conditioned performance surface (Fig. 5.6a).

Implementing the algorithm (Fig. 5.5) with initial weights $w_{O3} = 0$ and $w_{O4} = 0$ of E-N (Fig. 3.39) during conditioning, the learning curve was obtained (Fig. 5.8a). Figures 5.7 and 5.8 shows P in the learning curve settles at weight values which correspond to a solution set performance region (arrows, Fig. 5.7 & 5.8b) in both performance surfaces. Superimposing $P_{Conditioned}$ (Fig. 5.7b) and $P_{Conditioning}$ (Fig. 5.7d), one notices that regardless of the initial dip amount, the instant of the dip for both $P_{Conditioned}$ and $P_{Conditioning}$ coincides with the same weight values (arrow, Fig. 5.8b). As mentioned earlier, the weight values within the region of minima do not cause any practical difference whether it is local minima or global minimum of the conditioned performance surface. Similarly, though P in the learning curve dips and then rise to a plateau (arrow, Fig. 5.8a) the difference between the least P and P value at plateau is of the order 10^{-2}. The apparent rise of P from the minimum P in the learning curve to its plateau, especially when considered with respect to the conditioned performance surface, becomes less significant. Thus the minimum and plateau P values are within the solution set as per set membership.

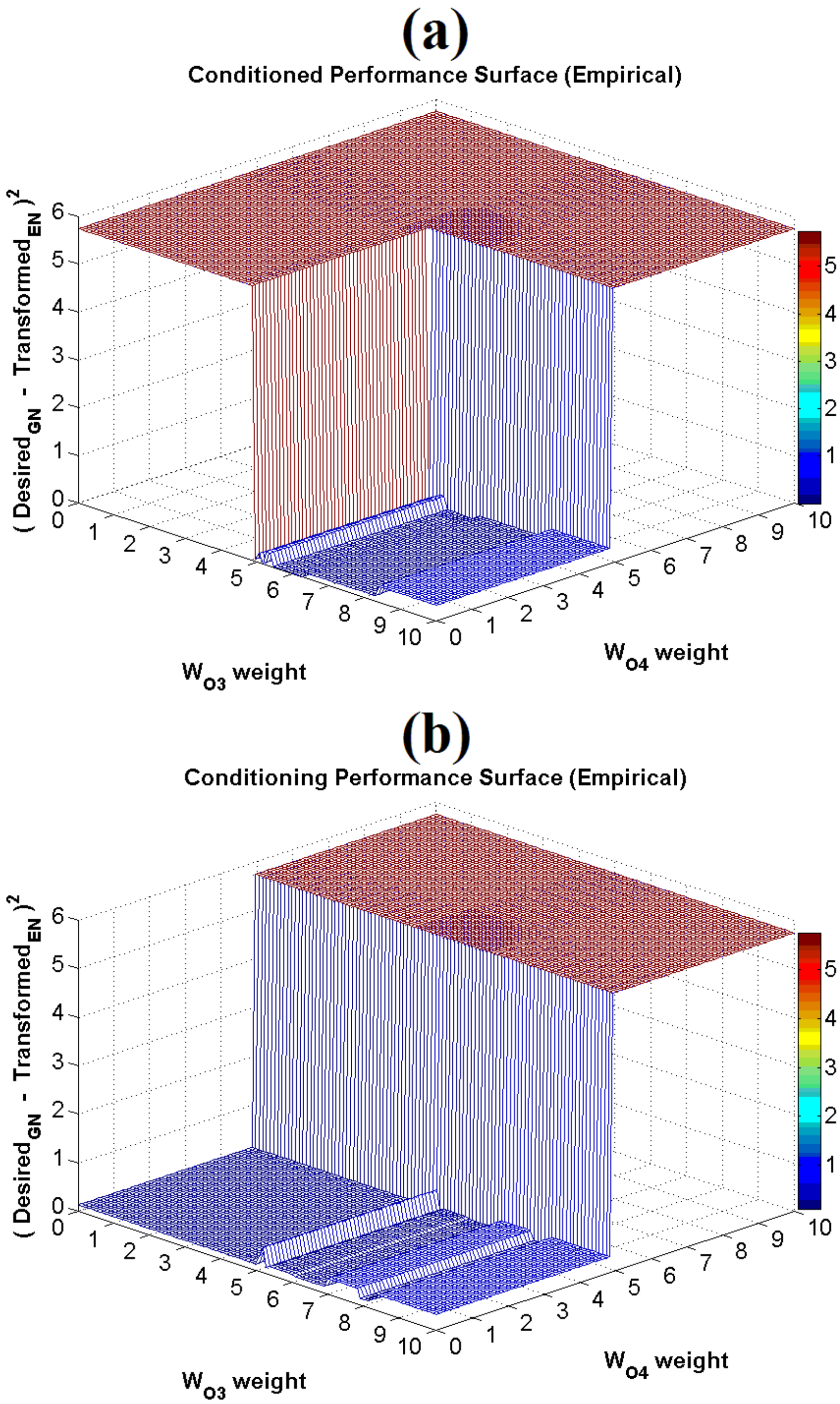

Figure 5.6. Conditioned (a) and Conditioning (b) Performance surface of Eckhorn network obtained empirically by manually (weight step, $\Delta w = 0.1$ between 4.5 to 6.5 and $\Delta w = 0.25$ else) adjusting the adaptive weight pairs (w_{O3} & w_{O4}). (a) Same as Figure 5.1 is the surface during E-N receiving B and S-stimuli while for (b) E-N receives all three stimuli, B, S and D. Weights w_{O3} and w_{O4} are adapting in (b) and adapted in (a). Thus, adaptation algorithm follows the conditioning performance surface (b).

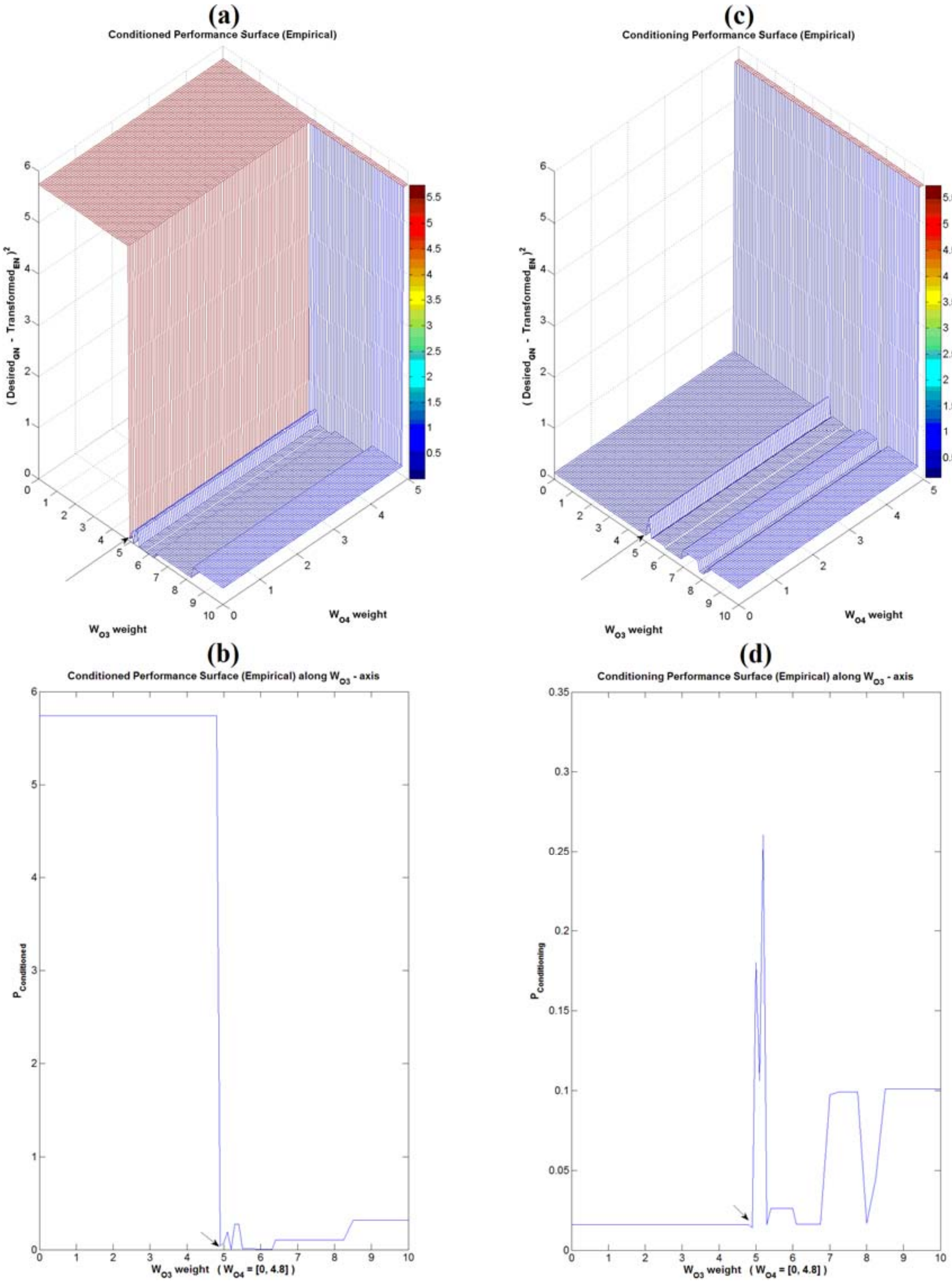

Figure 5.7. Conditioned and Conditioning Performance surface of Eckhorn network. (a) and (c) are conditioned and conditioning surface sections respectively from their whole surface (Fig. 5.6). The sections include weight ranges, $w_{O3} \in [0, 10]$ and $w_{O4} \in [0, 4.8]$. Bottom figures, (b) and (d) are sections of respective figures (a) and (c) taken along w_{O3} weight axes of the above 3D-surfaces. The arrows in all four figures indicate the dip from the initial plateau. The dip (arrow) correspond to when $w_{O3} \approx 4.9056$ and $w_{O4} \in [0, 4.8]$.

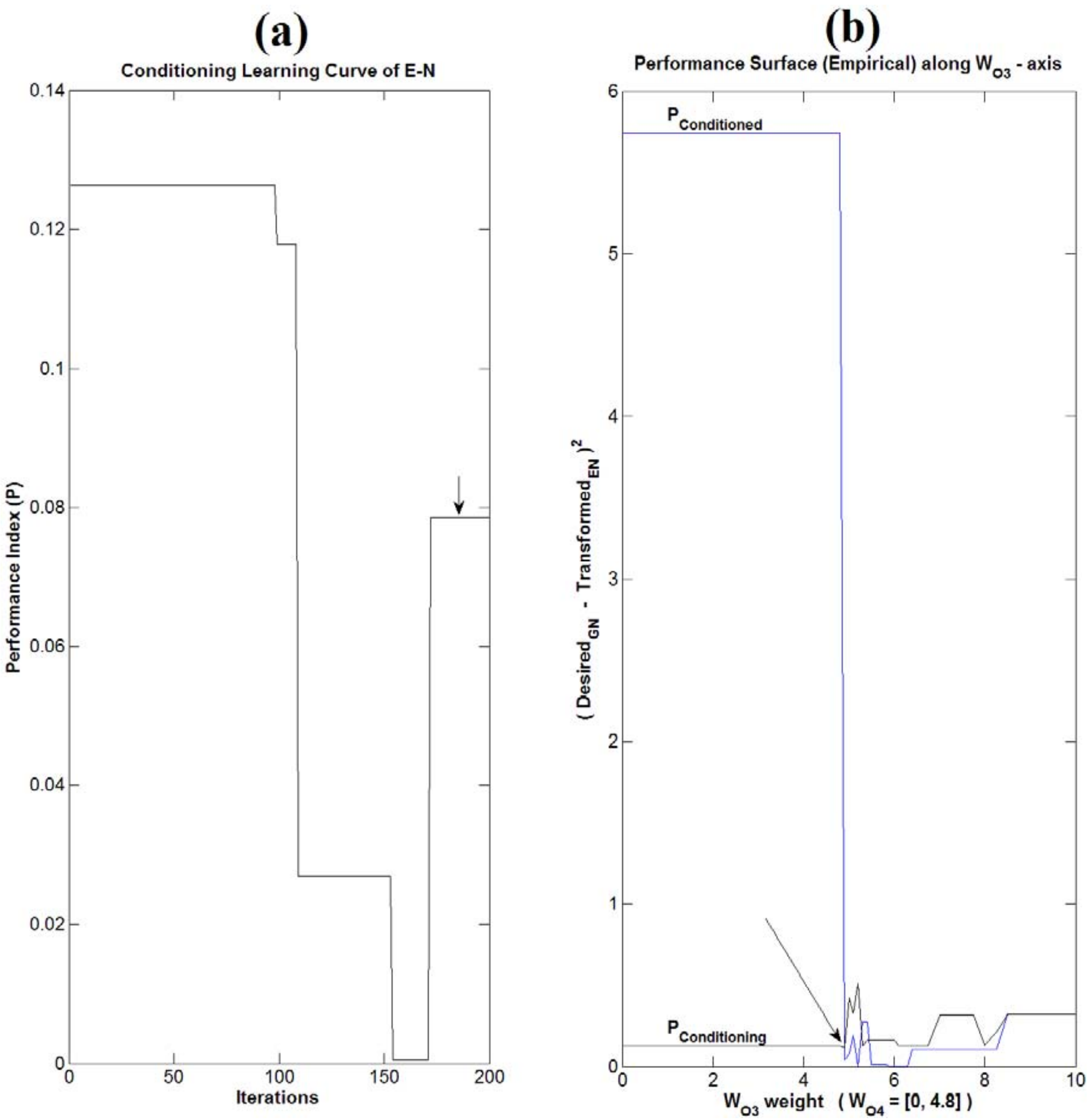

Figure 5.8. Conditioning learning curve (a) of E-N and performance surface (b) along w_{O3} - axis. (b) is obtained by superimposing Figures 5.7b and 5.7d.

(a). P starts out in the region of flat spot (R1) and remains in same R1 with increasing adaptive weights (push-weight procedure) until at 172nd iteration it moves down into new R1 (arrow) at the region of minima where the adaptation stops because beyond this point the performance gets worse (P increases). Note the y-axis scale (P value) in (a).

(b). P for the learning curve (a) follows $P_{Conditioning}$(b) during adaptation. The plateau (arrow (a)) in the learning curve after the fall in P corresponds to adaptive weight values at the dip (arrow (b)) in $P_{Conditioning}$. (b) also shows that the adaptive weights in learning curve plateau (arrow (a)) and hence the adapted weights also correspond to the dip of $P_{Conditioned}$.

The dynamics of the learning curve is shown in Figure 5.9. After the 99th iteration, |Gradient| > $G_{Minimum}$ (middle Fig. 5.9) thus, choosing steepest-descent method in the algorithm (Fig. 5.5). The steepest-descent method causes $\hat{\nabla}_{w_{O3}}$ to change (bottom Fig. 5.9) but $\hat{\nabla}_{w_{O4}}$ remains at zero. However after around 172nd iteration, the gradient estimate for flat region check and $\hat{\nabla}_{w_{O3}}$ fluctuates, though at different magnitudes. Therefore, after 172nd iteration onwards this $\hat{\nabla}_{w_{O3}}$ fluctuation results in an average w_{O3} value. The average w_{O3} (≈ 4.9056) and a non-changing w_{O4} (3.92) values causes the P plateau in the learning curve (double arrow in top Fig. 5.9). In other words, the P plateau in the learning curve (arrow, Fig. 5.10a) is due to the loop operation along the steepest-descent method path of the adaptive algorithm (Fig. 5.10b). This is merely the usual misadjustment property of gradient descent [Widrow & Stearns 1985]. The changes in the adaptive weights (w_{O3} & w_{O4}) responsible for the performance shift in the learning curve are shown in Figure 5.11.

Figures 5.12 and 5.13 shows the weight-learning curves for Grossberg's and Eckhorn's network respectively. The initial values for the adaptive weights are zero in Grossberg's network. The curves are plotted with same time axis (milliseconds) for one-on-one comparison. Due to the outstar-rule implemented in Grossberg's network, the adaptive weight for the node receiving B and no D stimulus ($w_{O4(Grossberg)}$) remains zero while $w_{O3(Grossberg)}$ (node with additional D stimulus) keeps getting bigger until it reaches a steady-state (optimal) value (≈ 0.4972). Adaptive weights in Grossberg's network reaches optimal value at around 110 sec or 1.83 min. However, adaptive weights in Eckhorn's network reaches optimal value at around 258 sec or 4.3min. That is, adaptation for the Eckhorn network is about two and half times slower (2.4 x). Investigation on optimization techniques for speeding up the adaptation is outside the scope of this thesis.

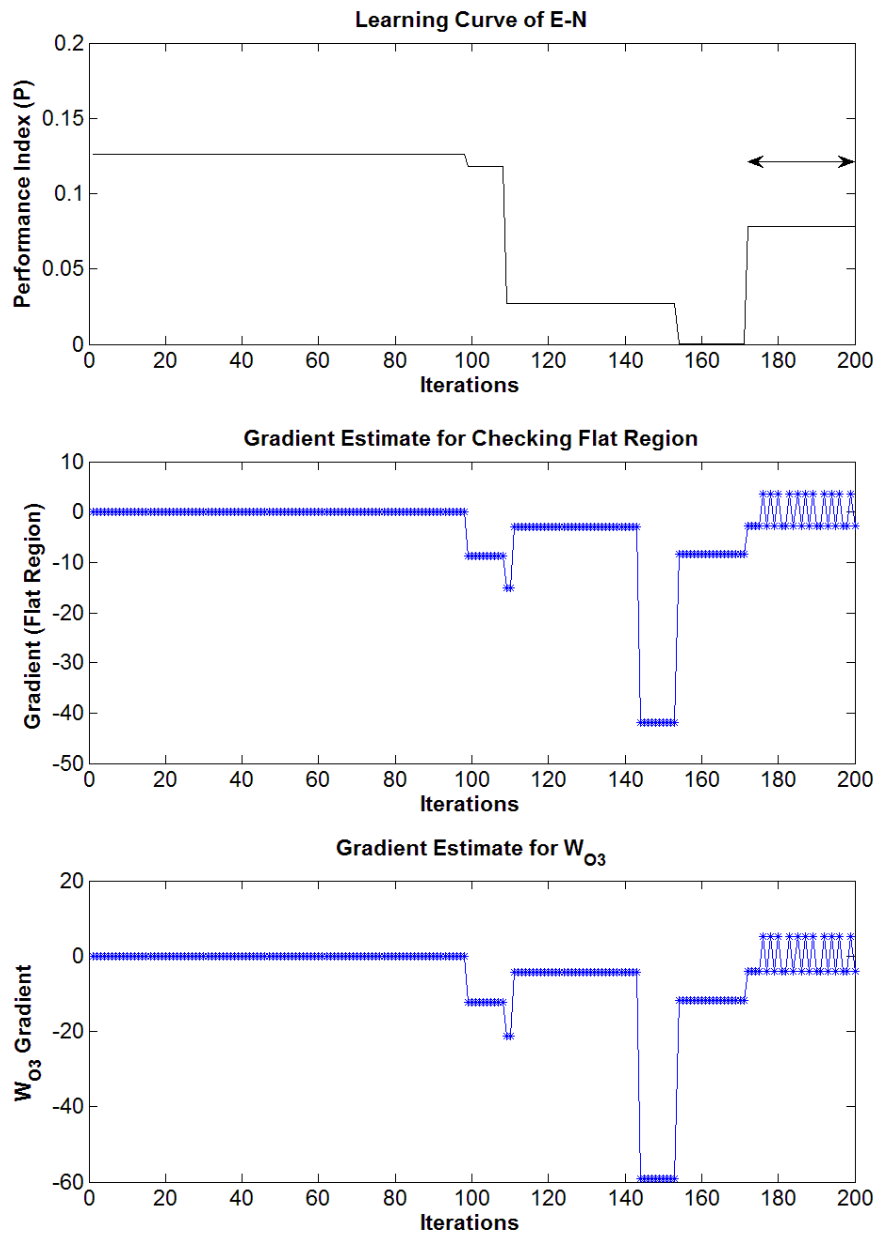

Figure 5.9. Learning curve of E-N with gradient estimates for flat region and w_{O3}. Learning curve or conditioning learning curve (a) is the same curve seen in Figure 5.8a. The double arrow indicates the plateau region (arrow, Fig. 5.8a) which begins at 172nd iteration. During the first 99 iterations the gradient estimate for flat region check (middle figure) is $|$ Gradient $| \leq G_{Minimum}$, thus adaptive weights follow push-weight procedure (Fig. 5.5). However during succeeding iterations $|$ Gradient $| > G_{Minimum}$, thus undergoing steepest descent procedure (Fig. 5.5). In the plateau region (double arrow), the gradient fluctuates by the same quantity. This fluctuation corresponds to those of w_{O3} gradient estimate ($\hat{\nabla}_{w_{O3}}$), bottom figure. $\hat{\nabla}_{w_{O4}} = 0$ (not shown) during the adaptation process.

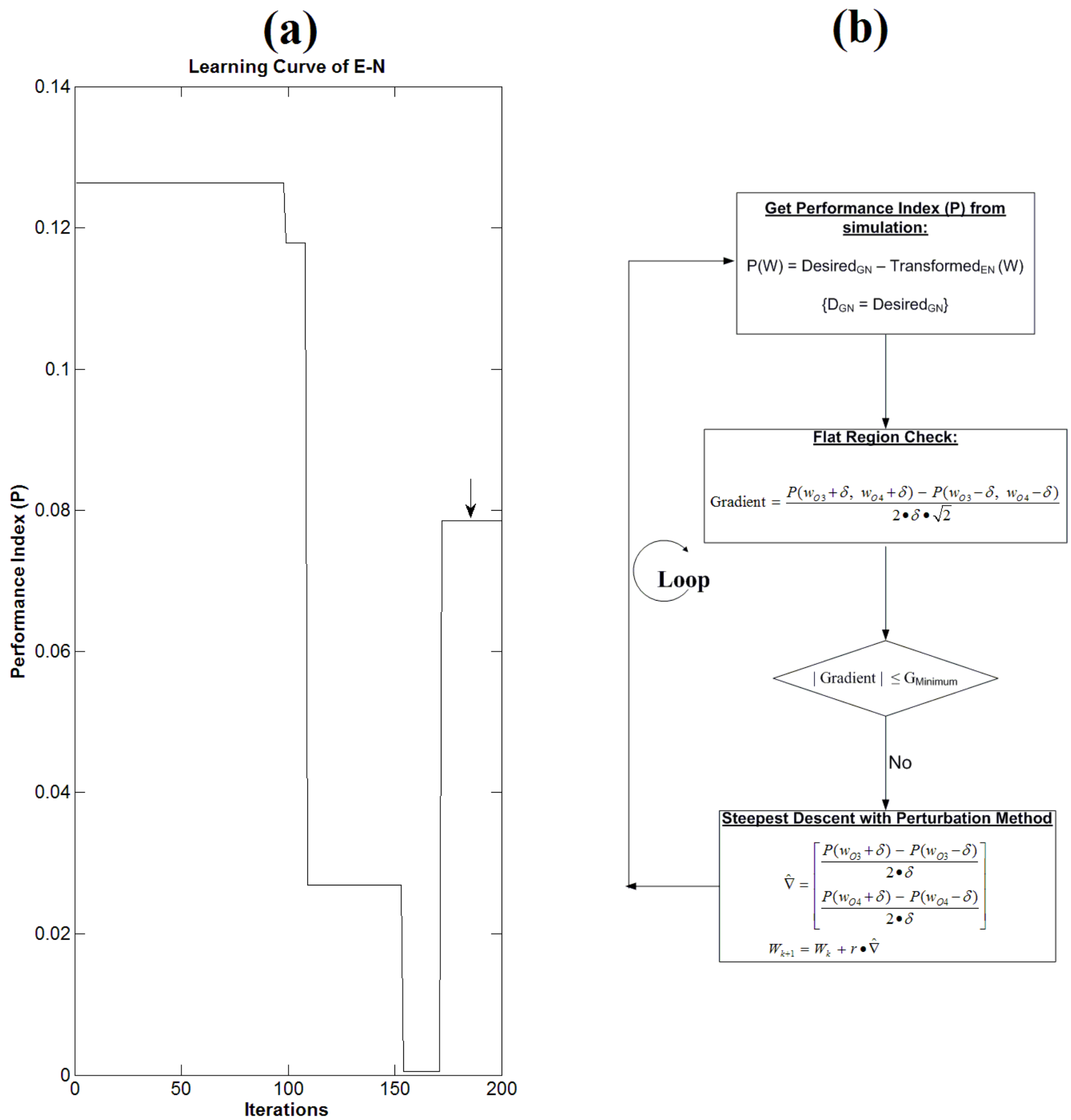

Figure 5.10. Learning curve of E-N. The plateau (arrow) in conditioning learning curve (a) beginning at 172nd iteration corresponds to fluctuation in gradient estimate for flat region check (middle Fig. 5.9), thus following the steepest descent method (b). During this plateau, $\hat{\nabla}_{w_{O3}}$ also fluctuates (bottom Fig. 5.9) resulting in an average w_{O3} value. The fluctuations in flat region gradient estimate and $\hat{\nabla}_{w_{O3}}$ results in repetition (b) along the steepest descent method of the adaptive algorithm.

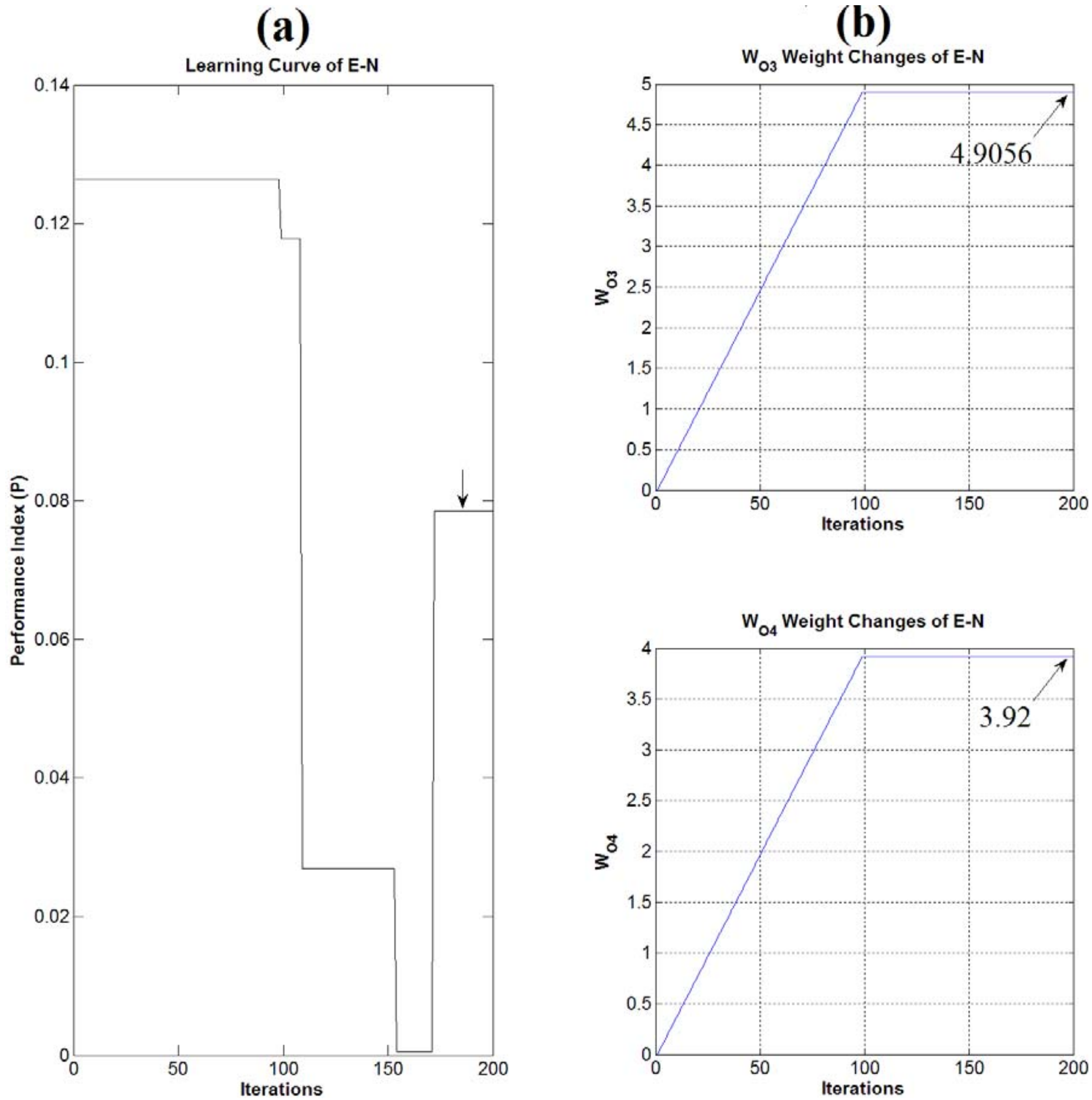

Figure 5.11. Weight changes (b) of Eckhorn network responsible for the learning curve (a). The adaptive weights (w_{O3} & w_{O4}) increases (push-weight procedure) in R1 of the initial plateau (Fig. 5.6) until it settles at the region of minima (arrow, (a)). The weights reach optimal values (arrows) at 172nd iteration. The labeled values ($w_{O3} \approx 4.9056$, $w_{O4} = 3.92$) corresponds to the P plateau seen in the learning curve (Fig. 5.10a). Note that $w_{O3} \approx 4.9056$ is an average value due to the $\hat{\nabla}_{w_{O3}}$ fluctuations (bottom Fig. 5.9).

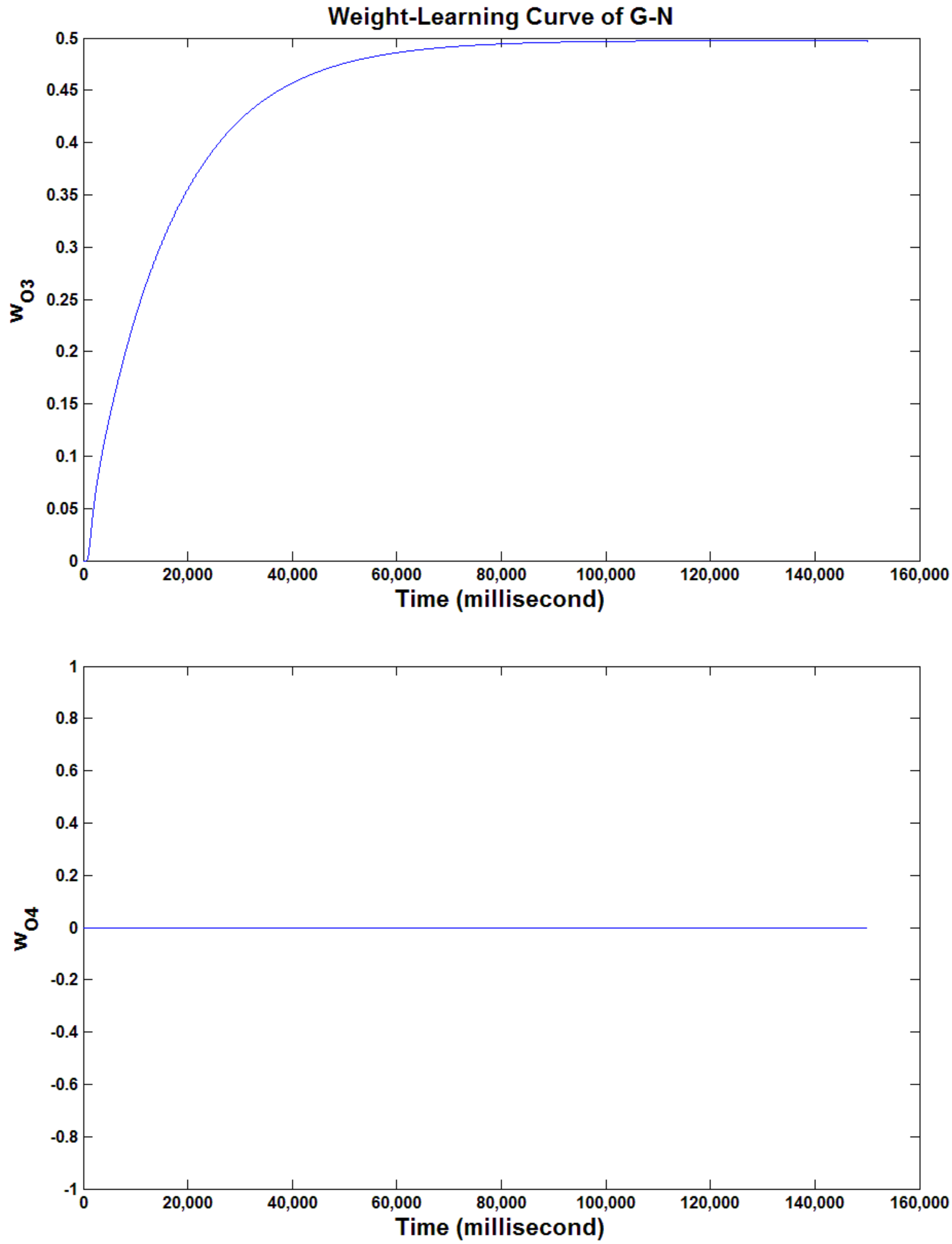

Figure 5.12. Weight-Learning curves of Grossberg's network plotting the adaptive weights ($w_{O3(Grossberg)}$ & $w_{O4(Grossberg)}$) against time in milliseconds. The weights reach optimal values ($w_{O3(Grossberg)} \approx 0.4972$) at 110 sec or 1.83 min.

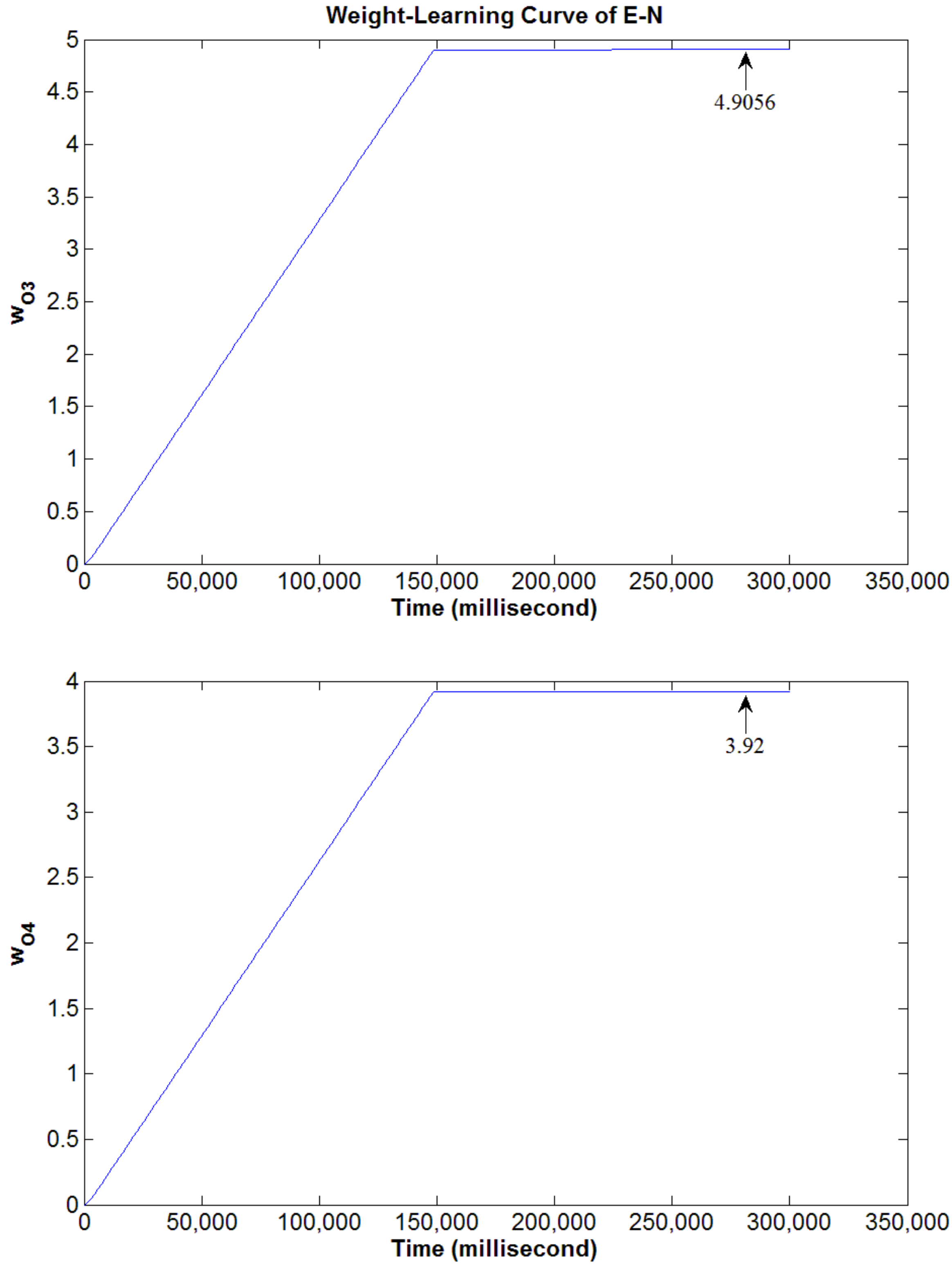

Figure 5.13. Weight-Learning curves of Eckhorn's network plotting the adaptive weights ($w_{O3(Eckhorn)}$ & $w_{O4(Eckhorn)}$) against time in milliseconds. The weights reach optimal values ($w_{O3(Eckhorn)} \approx 4.9056$ and $w_{O4(Eckhorn)} = 3.92$) at around 258 sec or 4.3 min (172nd iteration).