4.6 Practical Work

A system for the automatic management of two valves, placed 500 m (around 1600 feet) away of an industrial process, sends a codified signal with four variables {x_1, x_2, x_3 and x_4} that are necessary for the operation of the valve actuator. As illustrated in Fig. 4.7, the same communication line is used for activating both valves, so the switch placed next to them must decide whether the signal is for valve *A* or *B*.

However, during the communication, the signals suffer interferences that modify the content of the originally transmitted information. To avoid this problem, the team of engineers and scientists decided to train an ADALINE to classify the noisy

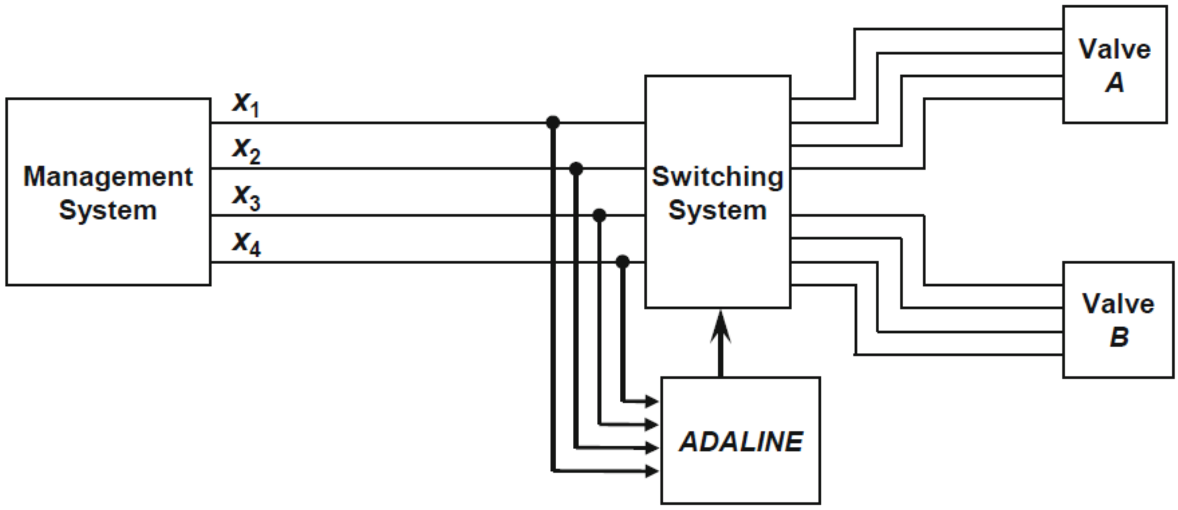

Fig. 4.7 Schematic structure of the valve management system

signals, in order to ensure to the switching system what data must be forwarded to valve *A* or *B*.

Thus, using the measurement of some noisy samples, the team compiled the training set presented in Appendix B, with the following convention: value −1 for the signals that must be sent to valve *A*; and value 1 for the signals that must be sent to valve *B*. The ADALINE structure proposed to this task is illustrated in Fig. 4.8.

Using the learning algorithm Delta rule for pattern classification with the ADALINE, perform the following activities:

1. Execute 5 training processes for the ADALINE, initializing the weight vector {$\mathbf{w}$} with random values between zero and one for each training process. If necessary, update the random number generator in each training so that the initial elements composing the vector are different in each process. Use a learning rate value {η} equal to 0.0025 and a precision {ε} equal to 10^{-6}. The training set can be found in Appendix B.
2. Register the results from the 5 training processes on Table 4.2

3. For the first two training processes, plot a graph showing the mean squared error on each training epoch, and analyze the behavior of both graphs. Discuss if the classes involved with the problem can be considered linearly separable.
4. For all training processes previously executed, present the values registered in Table 4.3 to the ADALINE (already trained), in order to classify (indicate to the switch) if the given signals should be sent to valve *A* or *B*.
5. Knowing that although the number of epochs of each training process from item 2 may be different, explain why the final values of the weights remain almost the same.

Fig. 4.8 Topology of the ADALINE used in the practical work

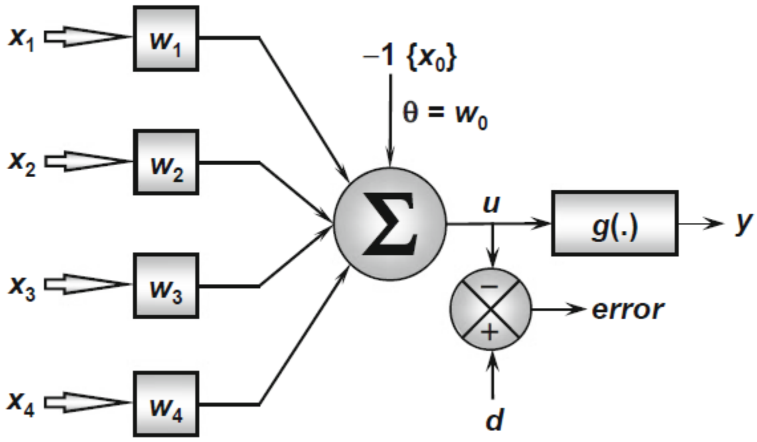

Table 4.2 Results from the ADALINE training

Training	Vector of weights (initial)					Vector of weights (final)					Number of epochs
	w_0	w_1	w_2	w_3	w_4	w_0	w_1	w_2	w_3	w_4	
#1 (T1)											
#2 (T2)											
#3 (T3)											
#4 (T4)											
#5 (T5)											

Table 4.3 Signal samples for classification by the ADALINE

Sample	x_1	x_2	x_3	x_4	y (T1)	y (T2)	y (T3)	y (T4)	y (T5)
1	0.9694	0.6909	0.4334	3.4965					
2	0.5427	1.3832	0.6390	4.0352					
3	0.6081	−0.9196	0.5925	0.1016					
4	−0.1618	0.4694	0.2030	3.0117					
5	0.1870	−0.2578	0.6124	1.7749					
6	0.4891	−0.5276	0.4378	0.6439					
7	0.3777	2.0149	0.7423	3.3932					
8	1.1498	−0.4067	0.2469	1.5866					
9	0.9325	1.0950	1.0359	3.3591					
10	0.5060	1.3317	0.9222	3.7174					
11	0.0497	−2.0656	0.6124	−0.6585					
12	0.4004	3.5369	0.9766	5.3532					
13	−0.1874	1.3343	0.5374	3.2189					
14	0.5060	1.3317	0.9222	3.7174					
15	1.6375	−0.7911	0.7537	0.5515					

Appendix B

Training set relating to Sect. 4.6

Sample	x_1	x_2	x_3	x_4	d
1	0.4329	−1.3719	0.7022	−0.8535	1.0000
2	0.3024	0.2286	0.8630	2.7909	−1.0000
3	0.1349	−0.6445	1.0530	0.5687	−1.0000
4	0.3374	−1.7163	0.3670	−0.6283	−1.0000
5	1.1434	−0.0485	0.6637	1.2606	1.0000
6	1.3749	−0.5071	0.4464	1.3009	1.0000
7	0.7221	−0.7587	0.7681	−0.5592	1.0000
8	0.4403	−0.8072	0.5154	−0.3129	1.0000
9	−0.5231	0.3548	0.2538	1.5776	−1.0000
10	0.3255	−2.0000	0.7112	−1.1209	1.0000
11	0.5824	1.3915	−0.2291	4.1735	−1.0000
12	0.1340	0.6081	0.4450	3.2230	−1.0000
13	0.1480	−0.2988	0.4778	0.8649	1.0000
14	0.7359	0.1869	−0.0872	2.3584	1.0000
15	0.7115	−1.1469	0.3394	0.9573	−1.0000
16	0.8251	−1.2840	0.8452	1.2382	−1.0000
17	0.1569	0.3712	0.8825	1.7633	1.0000
18	0.0033	0.6835	0.5389	2.8249	−1.0000
19	0.4243	0.8313	0.2634	3.5855	−1.0000
20	1.0490	0.1326	0.9138	1.9792	1.0000
21	1.4276	0.5331	−0.0145	3.7286	1.0000
22	0.5971	1.4865	0.2904	4.6069	−1.0000
23	0.8475	2.1479	0.3179	5.8235	−1.0000
24	1.3967	−0.4171	0.6443	1.3927	1.0000
25	0.0044	1.5378	0.6099	4.7755	−1.0000
26	0.2201	−0.5668	0.0515	0.7829	1.0000
27	0.6300	−1.2480	0.8591	0.8093	−1.0000
28	−0.2479	0.8960	0.0547	1.7381	1.0000

(continued)

I.N. da Silva et al., *Artificial Neural Networks*,
DOI 10.1007/978-3-319-43162-8

(continued)

29	−0.3088	−0.0929	0.8659	1.5483	−1.0000
30	−0.5180	1.4974	0.5453	2.3993	1.0000
31	0.6833	0.8266	0.0829	2.8864	1.0000
32	0.4353	−1.4066	0.4207	−0.4879	1.0000
33	−0.1069	−3.2329	0.1856	−2.4572	−1.0000
34	0.4662	0.6261	0.7304	3.4370	−1.0000
35	0.8298	−1.4089	0.3119	1.3235	−1.0000