6.0 Demonstration of Feed-Through Neural Network on Boiler Plant Simulation

A boiler is used to generate electricity by creating stream to drive a turbine. A simplified power generation process can be seen in Figure 6.1. A PID controller works very well maintaining the drum level until changes are implemented on the nonlinear system. The boiler plant makes a good nonlinear system for trying the feed-through neural network (F-T N/N) and the two update algorithms that were developed in Chapter 5. The computer simulation was provided to Dr. Van Landingham by the American Electric Power Company. The simulation models the nonlinearities of a thermal power plant, including the disturbances inherent in the process.

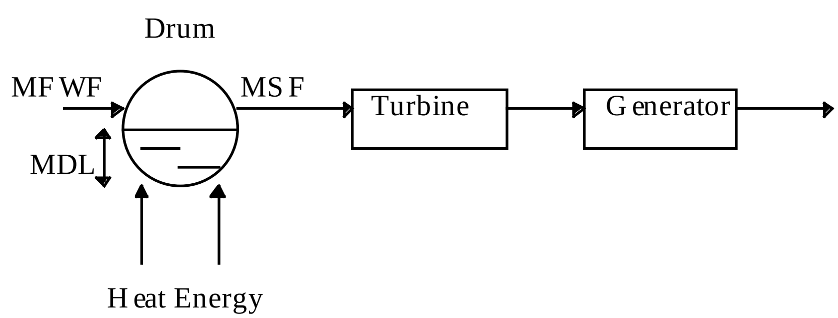

Figure 6.1 Power Generation Process

MFWF is the Measured Feedwater Flow;
MDL is the Measured Drum Level;
MSF is the Measured Steam Flow; and
SP is the Set Point for the Drum Level.

Feedwater enters the drum through a valve. The heat energy converts the water in the drum into steam. The steam drives the turbine and the generator coupled to the turbine produces electricity. One important consideration is to keep the level of water in the drum at a desired set point because there is a risk that the waters could completely evaporate. The standard three-element PID control works well under steady-state conditions. However, there may be changes in equilibrium conditions because, occasionally, the set point is changed. Tripathi, Tran, and VanLandingham (1995) did the research into the neural network control of the boiler plant that is being used in this chapter and their results are shown in Section 6.4.

A controller was placed on the boiler plant to maintain a commanded level of water in the drum, where heat is added to the water to create the steam needed to turn the turbine. The controller commands the amount that the feedwater flow valve is open. A step input can be commanded to the water level in the drum to increase or decrease the amount of water in the drum. This change in equilibrium can cause the PID controller to become detuned. The feed-through neural network can be used to compliment the existing fixed-gain PID controller. The PID implemented on the boiler plant was a three-element controller with drum level, stream flow, and feedwater flow being the variables involved, as seen in Figure 6.2.

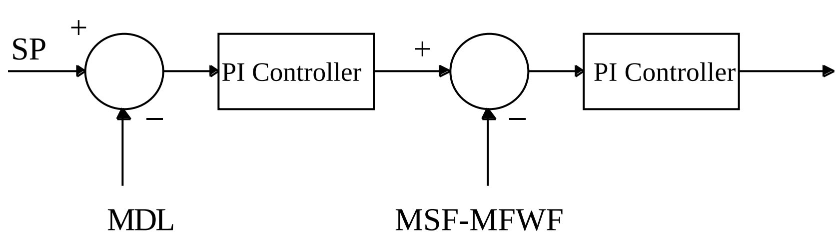

Figure 6.2 The Three-Element PID Controller

The implementation of the feed-through neural network on the boiler plant with the existing PID controller was done the same way for all three update algorithms: back propagation, the first update algorithm, and the second update algorithm. The neural network was placed between the existing controller and the plant, as seen in Figure 6.3. The feed-through neural network, initially, would allow the fixed-gain controller to control the system with the exact same performance it would have without the neural network. A reference model or desired response is used to train the neural network. The desired response of the drum level, used to train the neural network, was a response with zero rise time and no overshoot. The neural network improved the performance of the system as it converged. For the feed-through neural network, the weights were predetermined in order to give the system its initial performance. Because the initial weights were predetermined, the neural network only needs to be converged a single time.

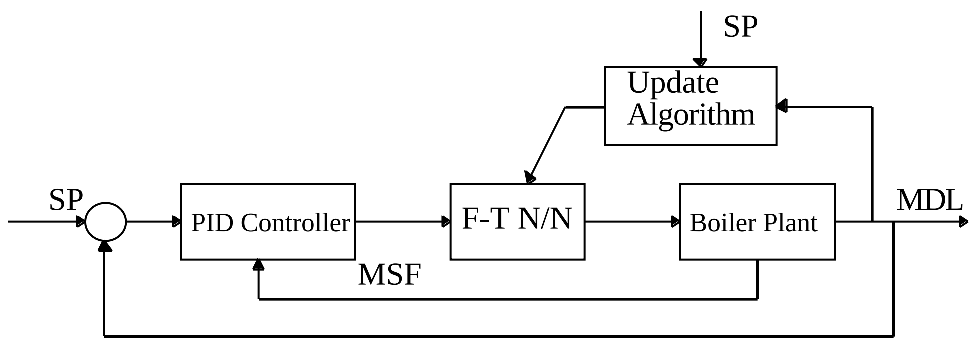

Figure 6.3 Block Diagram for Boiler Plant with F-T N/N

The feed-through neural network was applied to the boiler plant three separate times, shown in Sections 6.1, 6.2, and 6.3. The difference between these is the update algorithm used to converge the weights of the neural network. In Section 6.1, back propagation was used to obtain the results. After that, the following sections contain the results of the two update algorithms developed in Chapter 5 applied to the boiler plant. The results from all three sections are similar and follow the same pattern established in Chapter 5. The back propagation algorithm had excellent results, but it took much longer for the weights to converge than the other two algorithms. The first update algorithm converged quickest of the three but had a larger summed squared error than the other two algorithms. The second update algorithm converged quickly and had low summed squared error. The results of all three algorithms can be seen in the next three sections.

6.1 Results of the Back Propagation Algorithm

The back propagation algorithm was originally derived for the open-loop scheme as shown in Section 2.2. For boiler plant, the neural network was placed inside the closed loop. Although the back propagation was not originally derived for the updating of the neural network inside the closed loop, the algorithm did work very well. With the weights initially set in the feed-through configuration, the back propagation algorithm slowly converged the weights to improve the performance of the system.

The back propagation algorithm is not dependent on a plant model. This makes back propagation a very desirable update algorithm. In the boiler plant example, the PID controller was designed and implemented without a plant model. In the situation when a plant model is not available, the back propagation is the best choice as an update algorithm for the control methodology.

Four different steps were applied to the boiler plant. The boiler plant was set in an equilibrium position, and the step inputs were applied from this equilibrium position. The step input varied from one to four feet off of the twenty-foot equilibrium position.

The inputs to the neural network were a tap delay line of the outputs of the PID controller. The problem of implementing the tap delay line on the system was that the plant was sampled at a high rate compared to the time constants of the plant. Upon training the neural network with white noise, it was discovered that drum level was unaffected by the white noise. However, when a series of random-sized steps drove the system, the drum level was changed accordingly.

The inputs to the neural network were changed from every data point to every fifth data point. This was often enough to capture all of the pertinent dynamics without the neural network becoming too large for the computer system. The neural network was trained with a series of random steps because the system was unaffected by white noise.

The results of the boiler plant show that back propagation can converge the neural network to improved performance. The drum level with the converged neural network is the solid line. The drum level of the plant without the neural network is the dotted line. The minimum summed squared error that can be obtained is 71. The results of the step input from 20 to 21 can be seen in Figure 6.4. The summed squared error (SSE) between the desired drum level and the actual drum level for the case without the neural network was 286, and for the case with the converged neural network, it was 100, which is almost a three-fold decrease in SSE. The results of the step input from 20 to 22 can be seen in Figure 6.5. The SSE between the desired drum level and the actual drum level for the case without the neural network was 911, and was 172 for the case with the converged neural network, which is a five-fold decrease. The results of the step input from 20 to 23 can be seen in Figure 6.6. The SSE for the case without the neural network was 2036, and was 392 for the case with the converged neural network, another five-fold decrease. The results of the step input from 20 to 24 can be seen in Figure 6.7. The SSE for the case without the neural network was 3870, and was 566 for the case with the converged neural network, almost a seven-fold decrease. The back propagation greatly reduced the summed squared error of the drum level and increased the performance of the boiler plant.

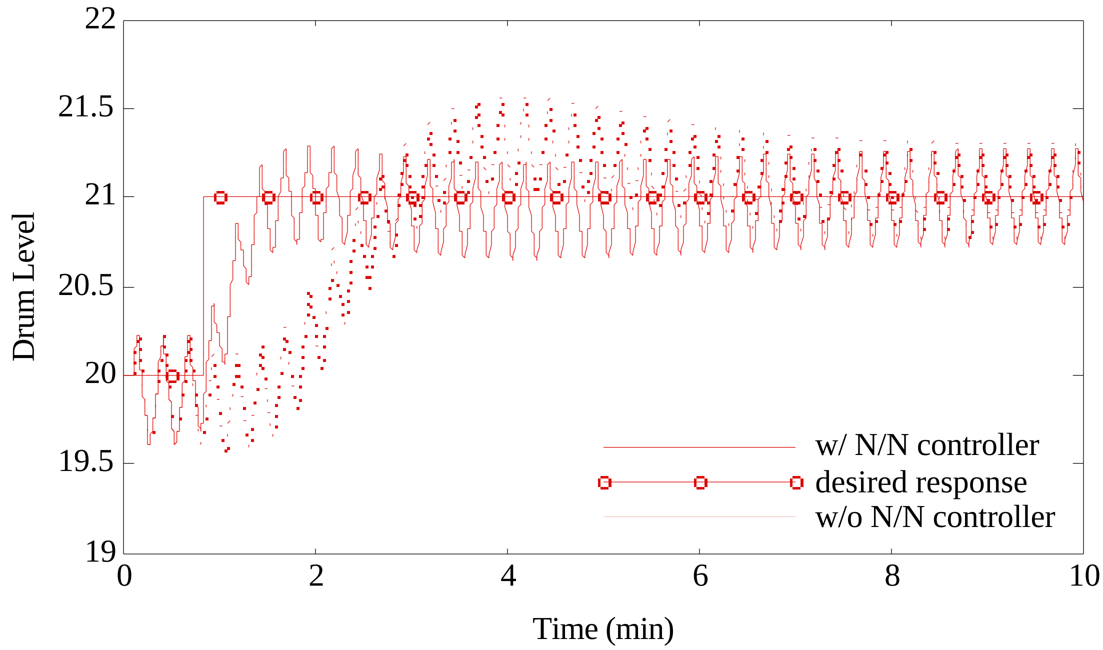

Figure 6.4 Results of One-Foot Step Input with Back-Propagation Training

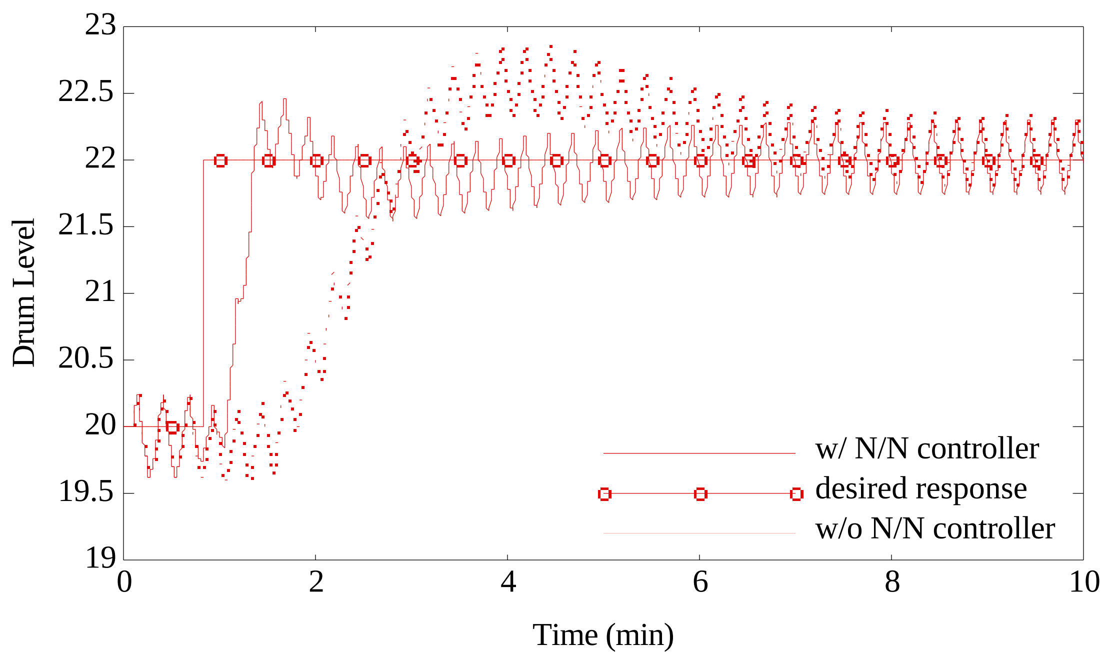

Figure 6.5 Results of Two-Foot Step Input with Back-Propagation Training

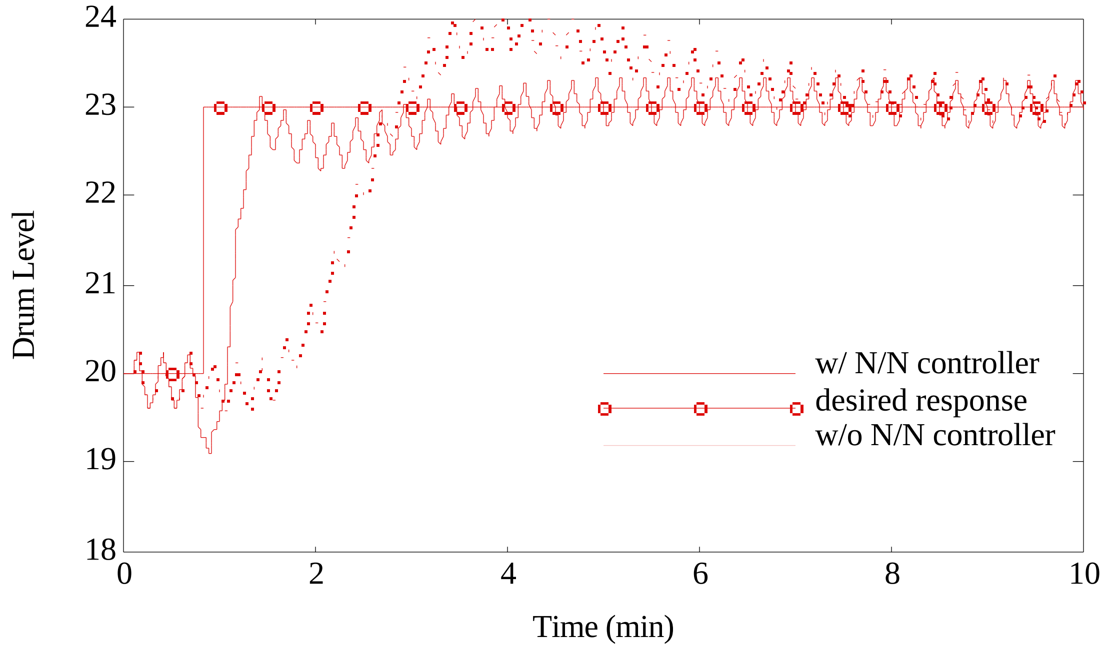

Figure 6.6 Results of Three-Foot Step Input with Back-Propagation Training

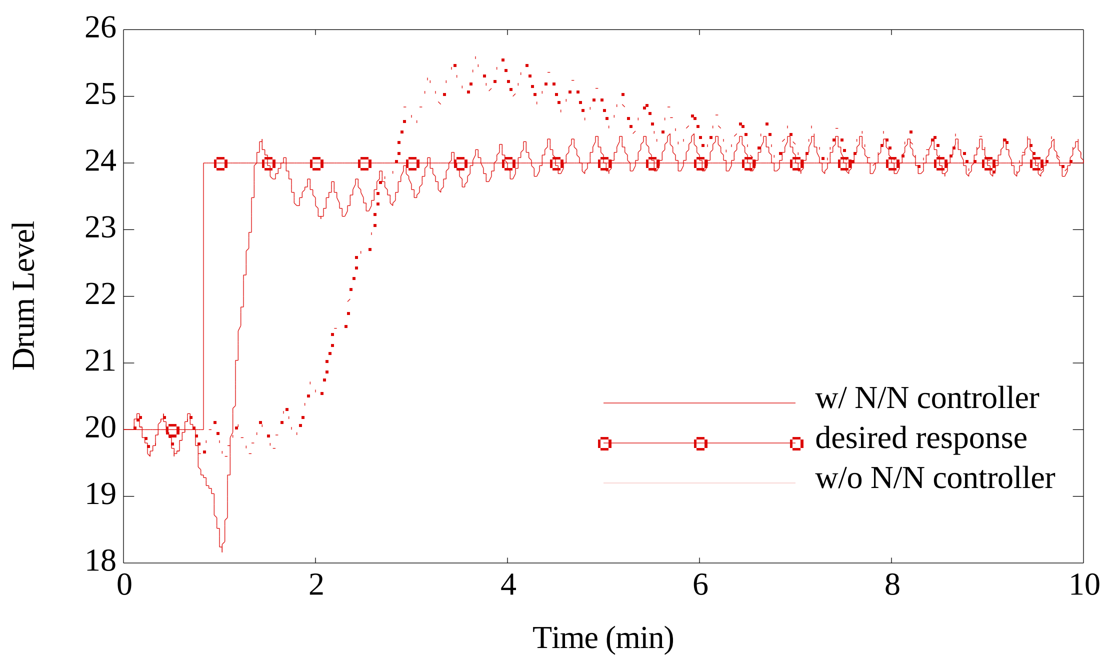

Figure 6.7 Results of Four-Foot Step Input with Back-Propagation Training

Back propagation algorithm worked very effectively to converge the weights of the neural network and increase the performance of the boiler plant. The PID controller worked very well when the boiler was at equilibrium. However, the PID controller did not perform well when a step input was commanded to the system. The neural network converged slowly to its weights because back propagation was not derived for the closed-loop system. The magnitude and phase of the plant can interfere with the convergence process. The back propagation algorithm took over a 100,000 iterations to converge. The convergence time can be reduced by using the two update algorithms developed in Chapter 5.

6.2 Results of the First Update Algorithm

The first update algorithm was derived to minimize the difference between the output of the neural network and the ideal output of the neural network. The algorithm requires a model of the system. A model of the boiler system was taken from Choi and Van Landingham (1995). The first update algorithm was very successful increasing the performance of the systems in Chapter 5, even though the primary variable of interest was not minimized. The difference between the output of the neural network and the ideal output of the neural network was minimized to reduce the effects of the plant's magnitude and phase on the convergence process.

The model for the boiler plant was a sixth-order system with two zeros outside the unit circle. The state space model of the system can be seen in Equations 6.1 - 6.4. The third output of the state-space equations is the drum level. The calculation of the ideal output of the neural network requires an inverse model of the plant. The inverse model can be difficult to use if the zeros are outside of the unit circle. This can be dealt with effectively by reflecting the zeros outside the unit circle to inside the unit circle. This was done for the two zeros in the boiler plant model using the method shown in Lu and Yahagi (1993).

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -0.7222 & 0.0467 & -0.5581 & 1.713 & -0.0512 & 0.5575 \\ 0.0523 & 0.3990 & 2.6559 & -0.0489 & 0.5901 & -2.7782 \\ 0.0045 & 0.0016 & -0.8259 & -0.0041 & -0.0013 & 1.8261 \end{bmatrix} \tag{6.1}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.2}$$

$$B = \begin{bmatrix} 0.3104 \\ 2.2622 \\ -0.0060 \\ 0.1227 \\ -0.9127 \\ -0.0102 \end{bmatrix} \tag{6.3}$$

$$D = \begin{bmatrix} 0.3491 \\ -1.4560 \\ 0.0395 \end{bmatrix} \tag{6.4}$$

The results for the first update algorithm were very good. The algorithm had similar results to the Back Propagation algorithm. The convergence time was much better with the first update algorithm. It converged within 15, 000 iterations, which compares to the 100,000 iterations with the back propagation algorithm.

The results of the closed-loop system show that the first update algorithm converges the weights of the neural network. The drum level with the converged neural network is the solid line. The drum level of the plant without the neural network is the dotted line. The results of the step input from 20 to 21 can be seen in Figure 6.8. The summed squared error between the desired drum level and the actual drum level for the case without the neural network was 286, and was 154 for the case with the converged neural network, almost a two-fold decrease in summed squared error. The results of the step input from 20 to 22 can be seen in Figure 6.9. The SSE between the desired drum level and the actual drum level for the case without the neural network was 911, and was 360 for the case with the converged neural network, which is almost a three-fold decrease. The results of the step input from 20 to 23 can be seen in Figure 6.10. The SSE for the case without the neural network was 2036, and was 602 for the case with the converged neural network, which is a three-fold decrease. The results of the step input from 20 to 24 can be seen in Figure 6.11. The SSE for the case without the neural network was 3870, and was 770 for the case with the converged neural network, which is a five-fold decrease. The first update algorithm greatly reduced the SSE of the drum level and increased the performance of the boiler plant.

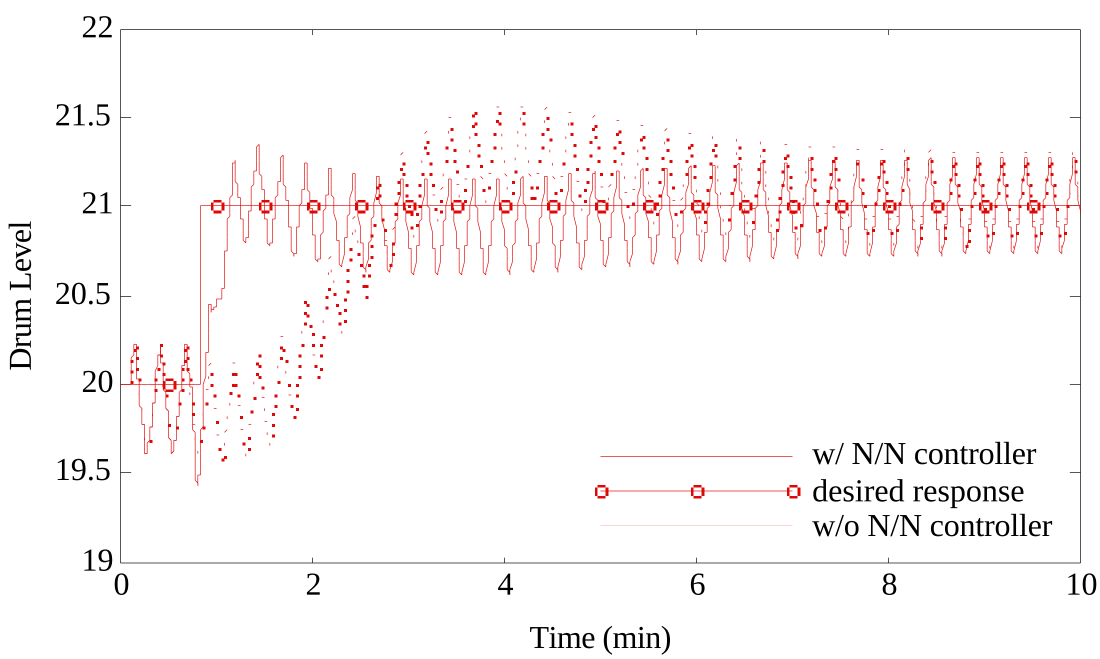

Figure 6.8 Results of One-Foot Step Input with First Update Algorithm Training

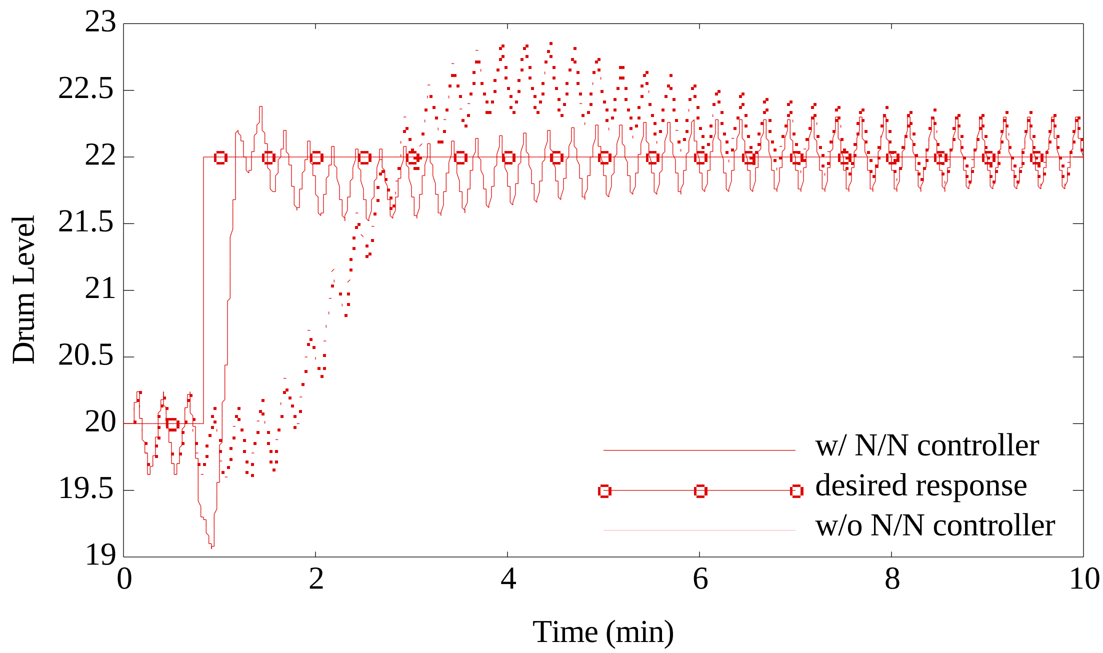

Figure 6.9 Results of Two-Foot Step Input with First Update Algorithm Training

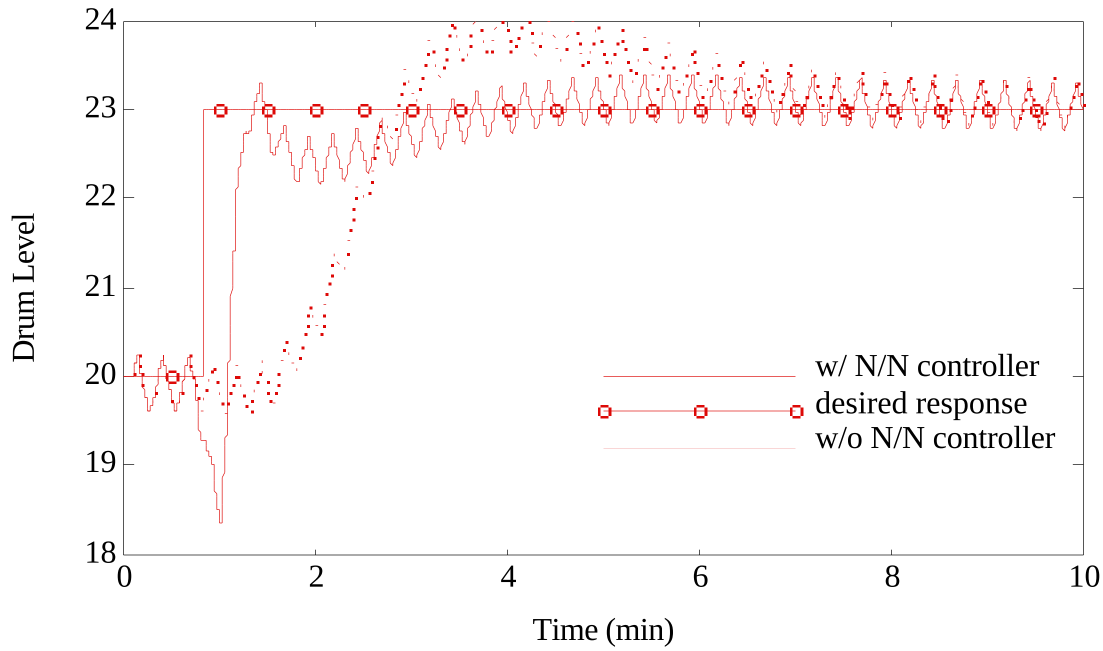

Figure 6.10 Results of Three-Foot Step Input with First Update Algorithm Training

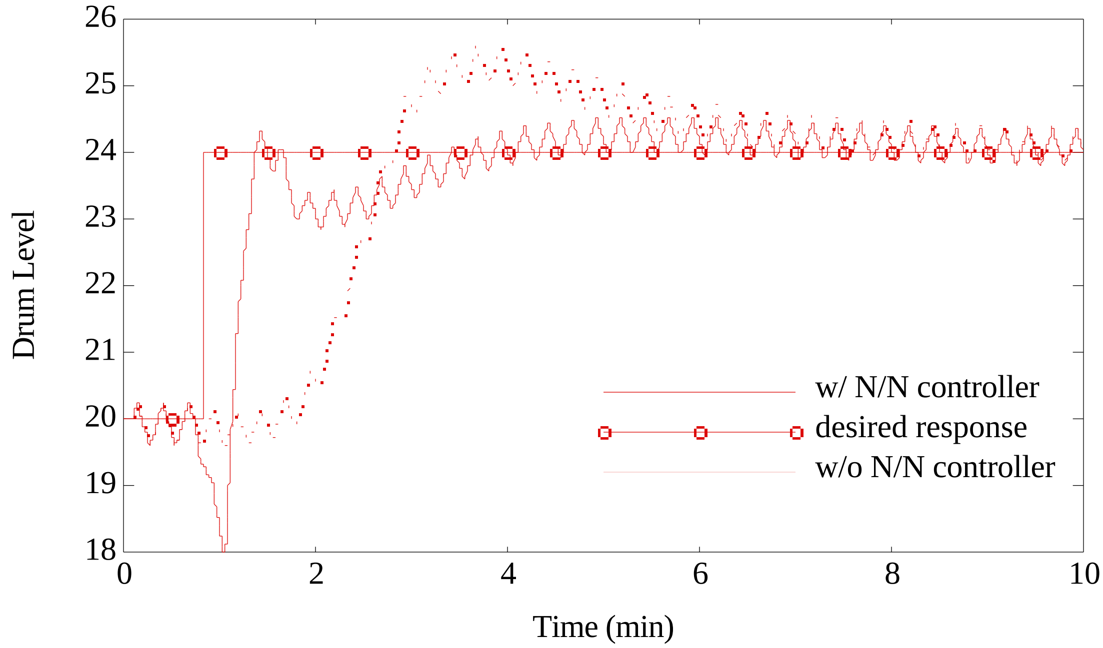

Figure 6.11 Results of Four-Foot Step Input with First Update Algorithm Training

With the exact model being unknown, a sixth-order model of the plant was developed through system identification. The first update algorithm used the model to effectively converge the neural network. The summed squared error was reduced by factors ranging from 2 to 5. The results show that the first update algorithm works effectively on nonlinear plants with unknown linearities. The first update algorithm improved the performance of the boiler plant. The results of the first update algorithm were comparable to the results with back propagation, but the first update algorithm converged faster. The second update algorithm can achieve the equivalent results of back propagation with the same quick convergence of the first update algorithm.

6.3 Results of the Second Update Algorithm

The second update algorithm minimizes the output of the plant and the desired output for the closed-loop system. The primary objective of the control system is to mimic the reference model or the desired output of the system. It is reasonable to minimize the primary variable of interest. Unlike the first update algorithm, the second update algorithm does not require an inverse plant model; however, it still does have the plant's dynamics effecting the convergence process.

The second update algorithm was implemented on the boiler plant in a similar fashion as the previous two update algorithms. The model required by the second update algorithm was the same model used by the first update algorithm, as seen in Equations 6.1 - 6.4.

The results of the second update algorithm show the improvement of the performance due to the convergence of the neural network's weights. The drum level with the converged neural network is the solid line. The drum level of the plant without the neural network is the dotted line. The results of the step input from 20 to 21 can be seen in Figure 6.12. The SSE between the desired drum level and the actual drum level for the case without the neural network was 286, and was 105 for the case with the converged neural network, almost a three-fold decrease. The results of the step input from 20 to 22 can be seen in Figure 6.13. The SSE between the desired drum level and the actual drum level for the case without the neural network was 911, and was 251 for the case with the converged neural network, almost a four-fold decrease. The results of the step input from 20 to 23 can be seen in Figure 6.14. The SSE for the case without the neural network was 2036, and was 421 for the case with the converged neural network, almost a five-fold decrease. The results of the step input from 20 to 24 can be seen in Figure 6.15. The SSE between the desired drum level and the actual drum level for the case without the neural network was 3870, and was 539 for the case with the converged neural network, which is a seven-fold decrease. The second update algorithm greatly reduced the SSE of the drum level and increased the performance of the boiler plant.

The second update worked as effectively on the boiler plant as the back propagation algorithm. The summed squared errors were very comparable to that of the back propagation. The convergence process took about 15,000 iterations, which is comparable to the first update algorithm. The second update algorithm is the best of the three algorithms with both performance and a quick convergence rate.

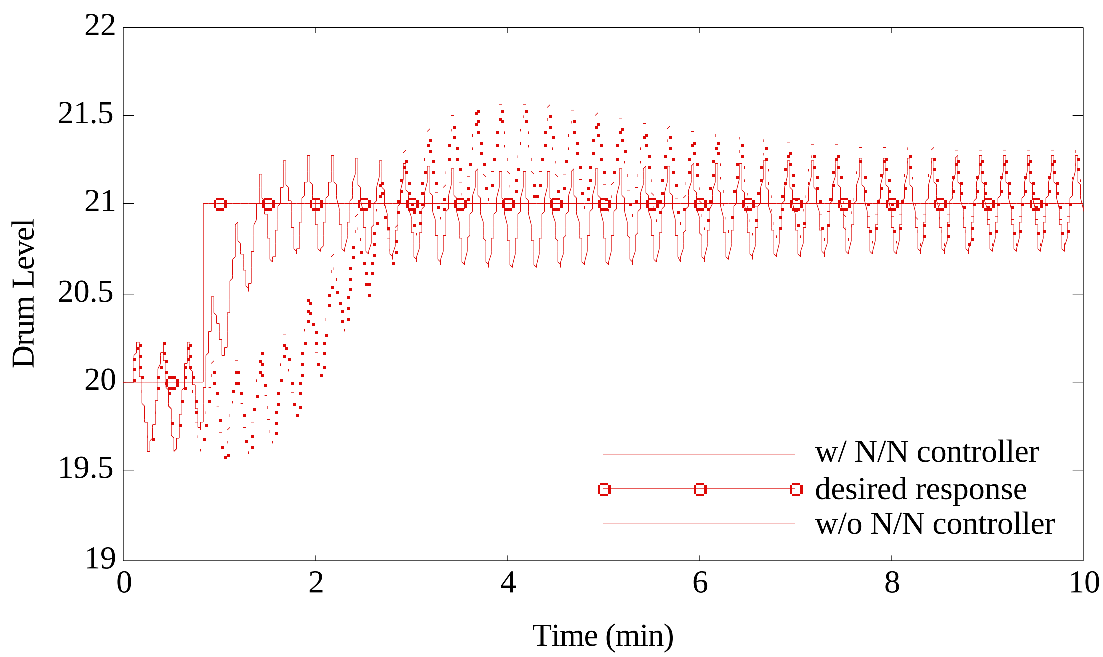

Figure 6.12 Results of One-Foot Step Input with Second Update Algorithm Training

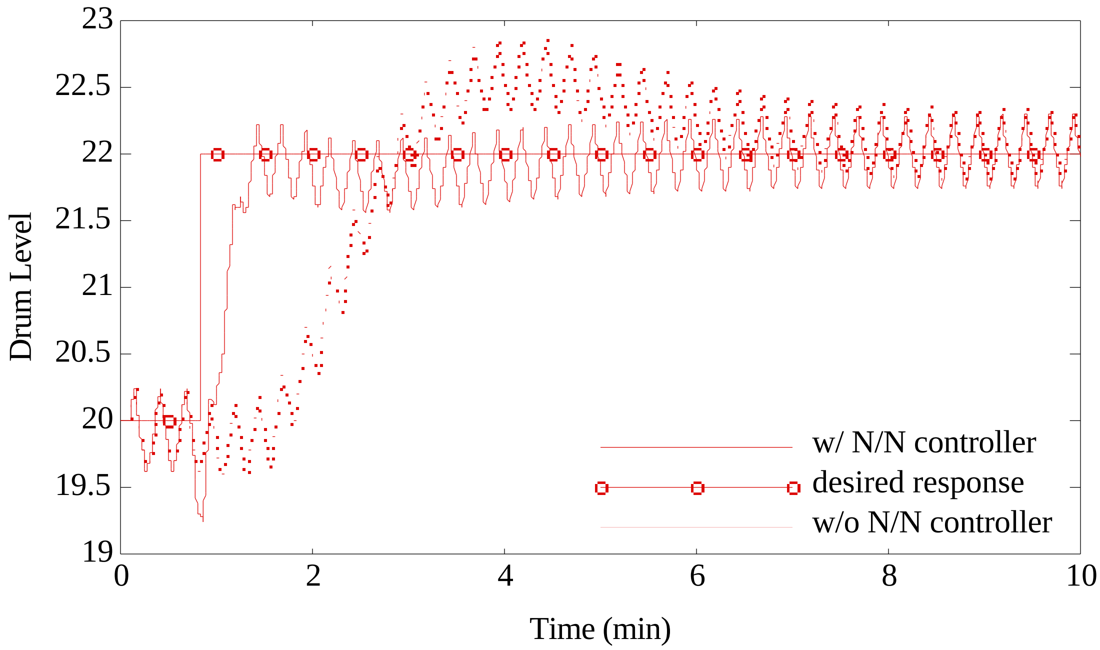

Figure 6.13 Results of Two-Foot Step Input with Second Update Algorithm Training

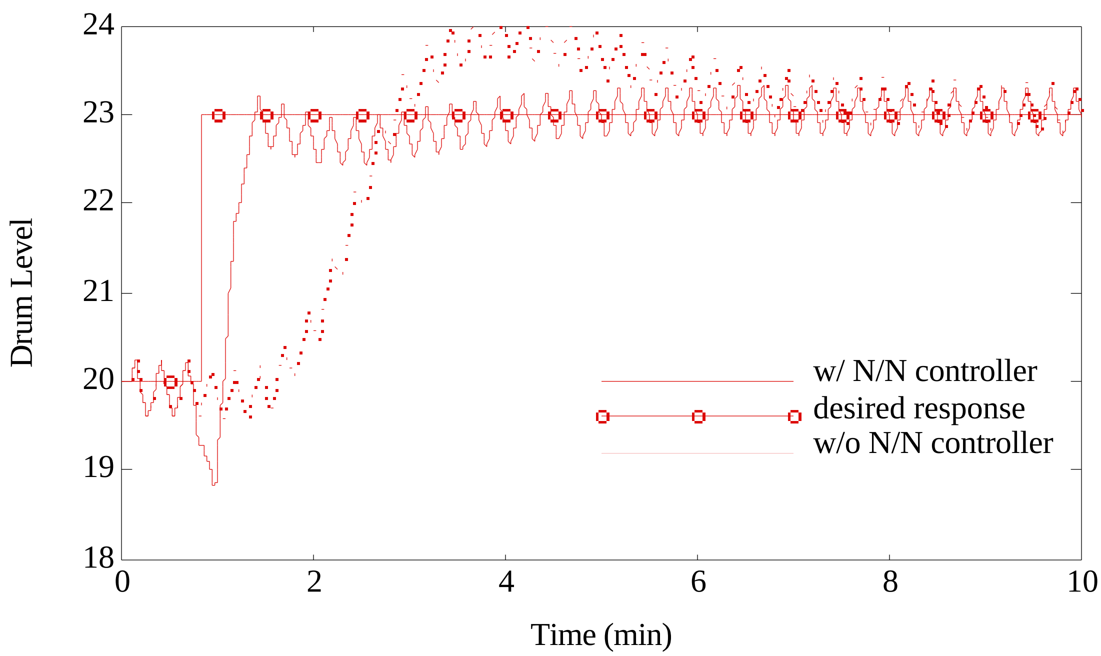

Figure 6.14 Results of Three-Foot Step Input with Second Update Algorithm Training

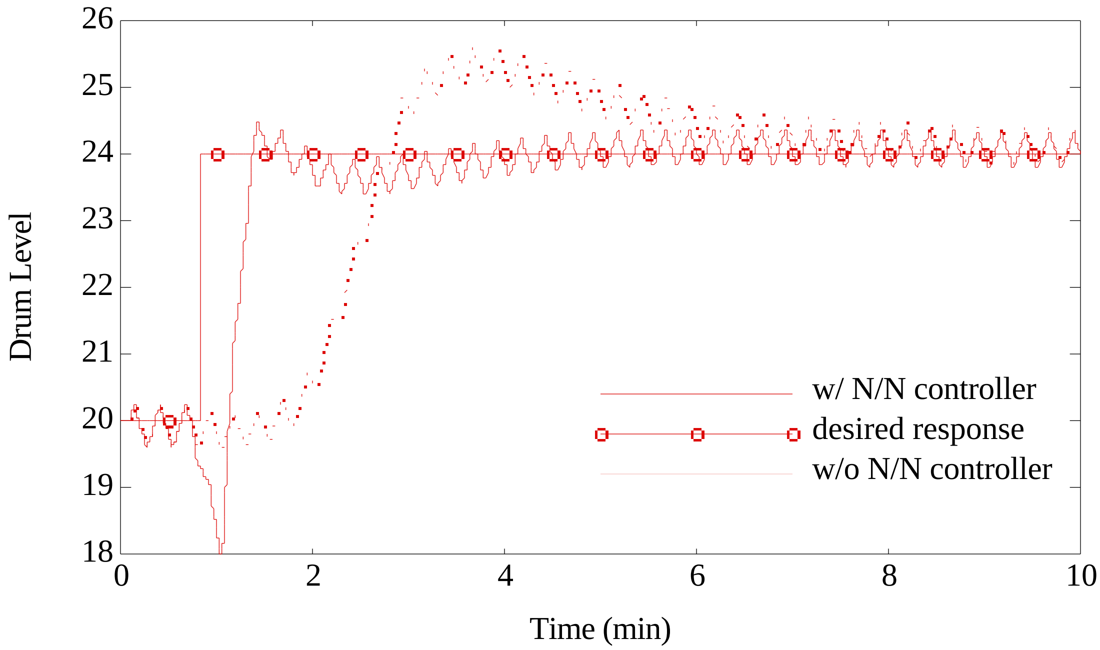

Figure 6.15 Results of Four-Foot Step Input with Second Update Algorithm Training

6.4 Summary

All three update algorithms worked effectively to improve the performance of the boiler system and decrease the summed squared error. Compared to back propagation, the two update algorithms developed in Chapter 5 specifically for closed-loop systems had a greatly-reduced convergence time. Back propagation and the second update algorithm had slightly reduced summed squared errors compared to the first update algorithm. Finally, the final performance of the controlled system, after the neural network was converged, was very similar for all three algorithms.

The direct comparison of the converged results for the three update algorithms can be seen in Figures 6.16 through 6.19. The solid line is for the back propagation training. The dashed line is for the first update algorithm. The dotted line is for the second update algorithm. The three lines are difficult to distinguish except for the initial transient. The use of the exact same neural network configuration for all three could explain part of the similarities. However, the largest similarity is that the two new update algorithms are very similar to the back propagation algorithm, except for a correction term for the closed-loop that caused the weights to converge much faster than with back propagation.

Table 6.1 is the summary of the summed squared errors (SSE) for the three update algorithms, BP, AL1, and AL2, as well as the results from Tripathi, Tran and Van Landingham (1995). They developed a neural network controller for this boiler plant and their results are presented in Table 6.1. The baseline, BL(TTV), is slightly different than the baseline found in this work. The data shows that the feed-through neural network has converged to a lower SSE than that for the neural network controller, N/N(TTV), developed by Tripathi et al. The results of the three update algorithms are similar, with back propagation and the second update algorithm being better than the first update algorithm.

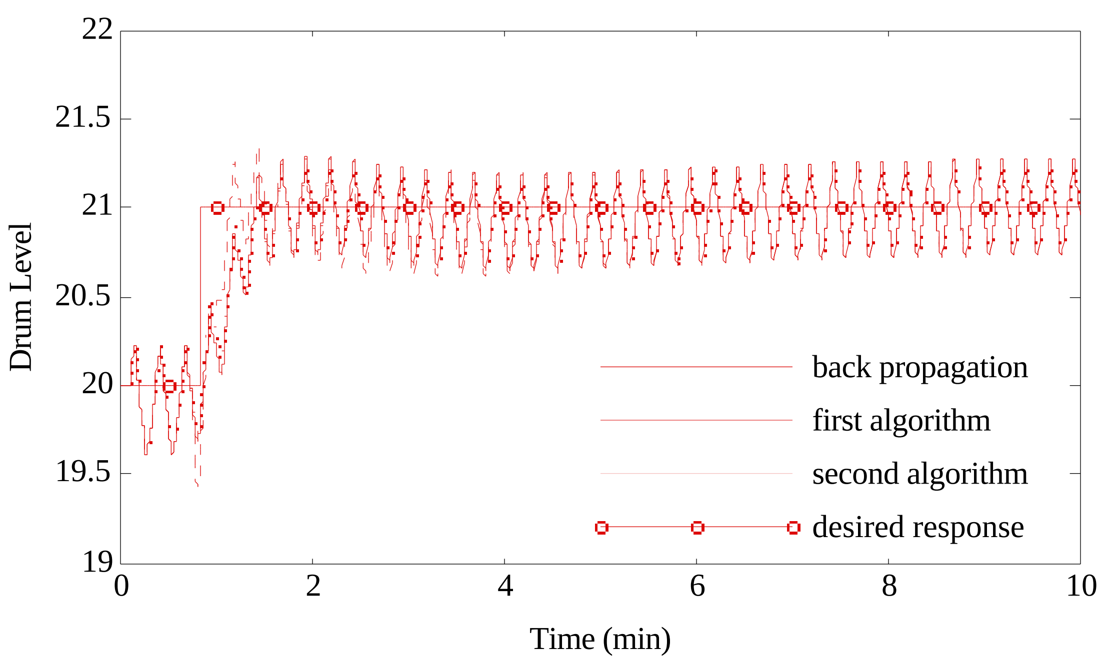

Figure 6.16 Comparison of Results for the Three Algorithms of One-Foot Step Input

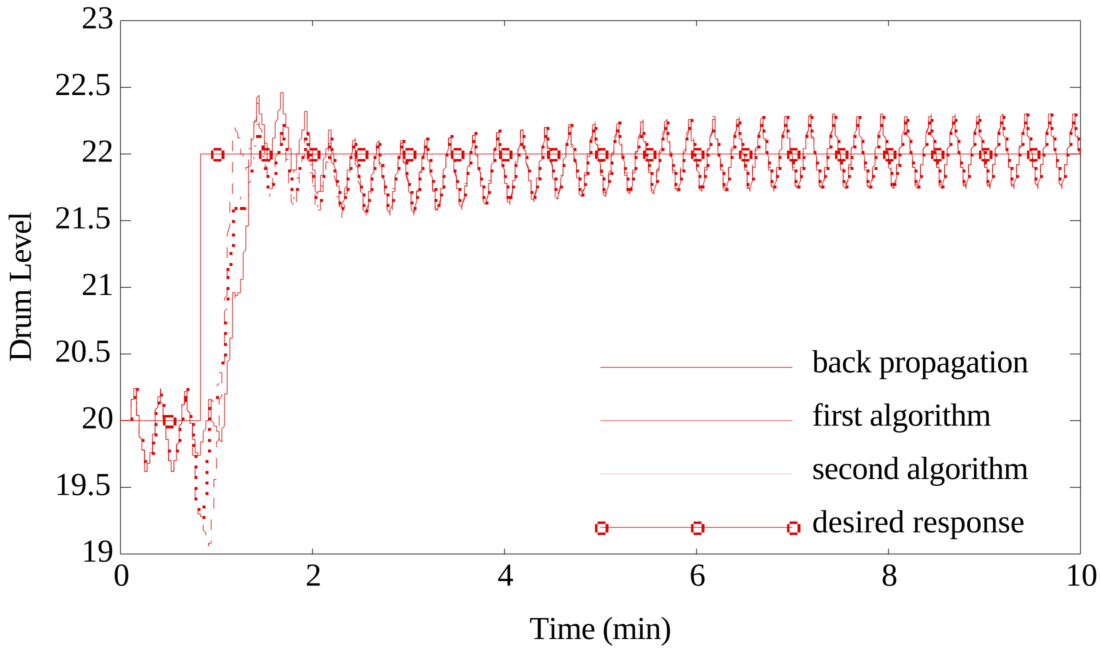

Figure 6.17 Comparison of Results for the Three Algorithms of Two-Foot Step Input

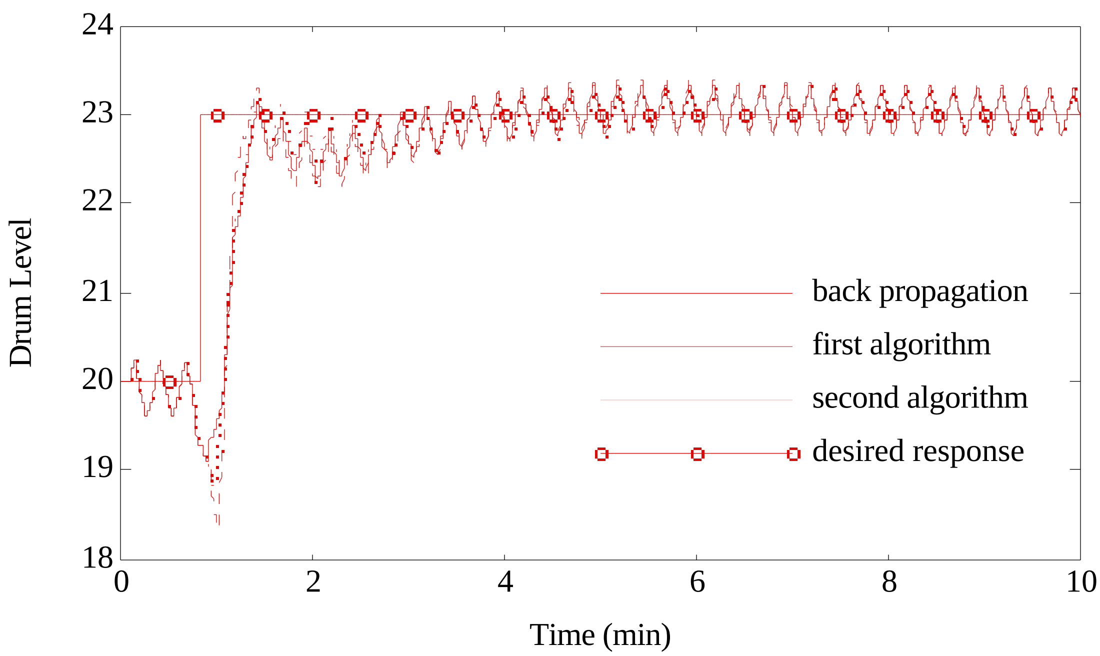

Figure 6.18 Comparison of Results for the Three Algorithms of Three-Foot Step Input

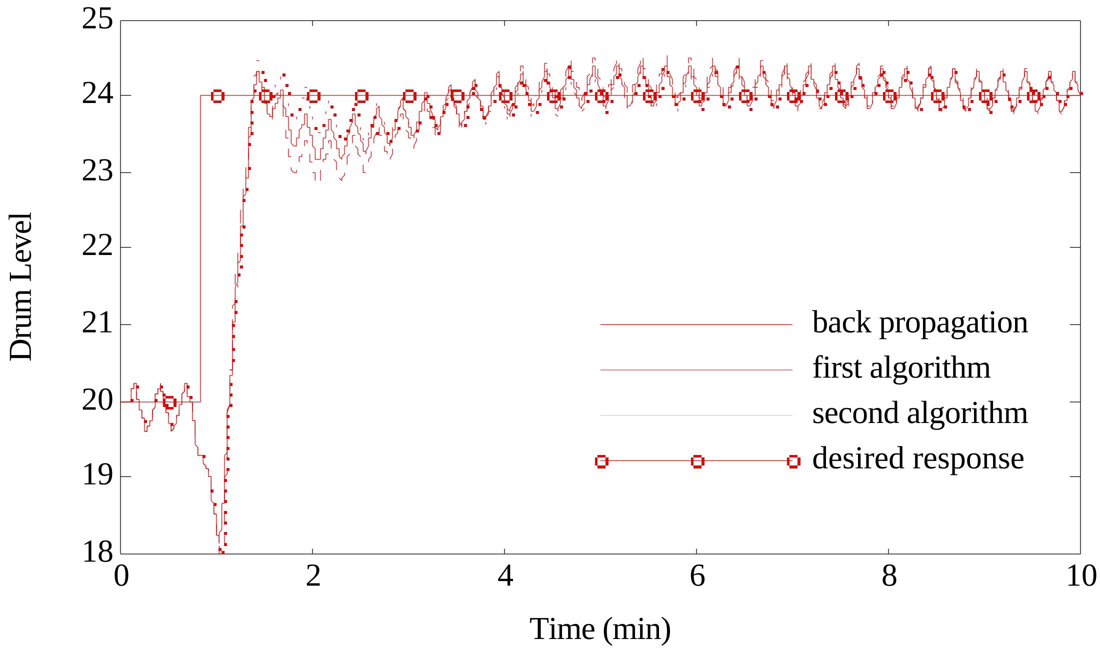

Figure 6.19 Comparison of Results for the Three Algorithms of Four-Foot Step Input

Table 6.1 Comparison of SSE for the Three Algorithms and Other Results

	Baseline	BP	AL1	AL2	BL (TTV)	N/N (TTV)
SP 20 to 21	286	100	154	105	320	180
SP 20 to 22	911	172	360	251	953	700
SP 20 to 23	2036	392	602	421	2050	1671
SP 20 to 24	3870	566	770	539	3858	3179

The three update algorithms improved the performance of the boiler system. The two new update algorithms converge much quicker than back propagation. The first update algorithm has the worst performance of the three algorithms but has the fastest convergence rate. This can be attributed to the use of an imperfect plant model. The imperfect model does remove a portion of the plant's dynamics. However, it does not make up for the primary variable of interest not being minimized in the process. The second update algorithm is the best of the three algorithms; it has the performance of back propagation and the convergence time of the first update algorithm. However, all three could greatly increase the performance of the boiler system.