

4.0 Update Algorithms For Linear Closed-Loop Systems

A controller design methodology has been developed that combines an adaptive finite impulse response (FIR) filter with feedback. FIR filters are used in the open-loop with the Least Mean Squares (LMS) algorithm for adaptation. They work very well for disturbance rejection. The FIR filter was chosen because it is the linear equivalent of the feed-forward neural network. Feedback can add damping to the system and works well for tracking. In our methodology, the FIR filter is adapted while inside the closed-loop of the system. With the FIR filter inside the closed-loop, the poles of the system can be moved. However, the structure of the FIR filter only allows zeros on the controller to be adapted. The poles of the FIR filter remain at zero.

An LMS-style algorithm is derived for the feedback control problem. The combination of the easy adaptability of the LMS algorithm with the ability to change the dynamics of the system can provide a useful means of developing controllers. The development of the weight update rule is based on minimizing the square of the error between the plant and the reference model. The algorithm is then derived from the feedback layout while trying to minimize the error. However, the weights of the tap delay line are combined with the dynamics of the plant in the same manner as the controller on a more traditional feedback controller.

This chapter develops two new update algorithms for converging an FIR filter inside a closed-loop configuration. The first update algorithm is derived to minimize the error between the output of the FIR filter and the desired reference output of the FIR filter. This error is the ideal error to facilitate a quick convergence of the system. The second update algorithm is derived to minimize the error between the output of the plant and the desired output of the plant; this is the variable of primary interest. The derivation of the two update algorithms for the linear system provides a better understanding for the development of the update algorithms for the nonlinear system.

Examples show various aspects of the design methodology. The first two examples have analytical solutions to them and have feedback with a proportional controller that has a gain of one. The first example is a stable plant; the second plant is unstable. Both require two weights in the FIR filter for the closed-loop system to mimic the reference model. The FIR filter adapts its first two weights to the required values, and the additional weights converge to zero. The third example is a stable plant, which is not modeled perfectly. The damping ratio of the actual plant is 50% less than what is modeled. The weights converge to a point at which the error between the reference model and the closed-loop system is minimized but not completely eliminated. The design methodology shows an 80% reduction in the mean squared error between the analytically-computed controller with the inaccurate model and the new adapted controller. Two additional examples, one stable and one unstable, are shown with fixed gain controllers and plants that have not been perfectly modeled. All of the examples are used with both update algorithms.

The design methodology only moves the zeros of the compensator, which decreases the potential effectiveness of the controller. The FIR filter is structured such that the weights are only in the numerator. The zeros of the filter can be varied. The poles are set at zero, which limits the number of variables to half of the potential variables of the compensator but greatly simplifies the adaptation algorithm. The adaptation algorithm does converge to the analytical solution when such a solution exists. When an analytical solution does not exist, the algorithm minimizes the mean square error in the same manner as the LMS algorithm.

4.1 First Update Algorithm

Terms used in Figure 4.1 and throughout this work are listed below:

x is the input into the system;

u is the output of the tap delay line;
 W is the weights of the tap delay line;
 P is the discrete plant model;
 P_m is the discrete reference model;
 μ is the learning rate;
 y is the output of the plant;
 y_m is the model output;
 u_m is the input to the plant that would result in y matching y_m ;
 L is the number of weights in the tap delay line;
 X is a vector of past values of x ; and
 Y is a vector of past values of y .

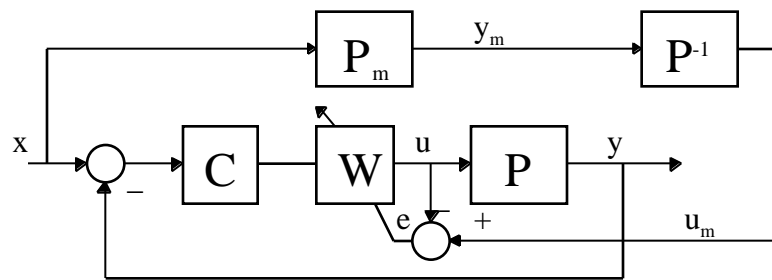


Figure 4.1 Block Diagram of Closed-Loop Control Scheme

The objective of this work is to get the output of the plant to mimic the output of the reference model. Therefore, the logical error used to adapt the tap delay line is the difference between the actual system output and the model system output ($y_m - y$). However, another possible error for the tap delay line (W) is the ideal output of the tap delay line minus the actual output of the tap delay line ($u_m - u$). If e_y , in Equation 4.1, is used, there is unnecessary phase and amplitude introduced by the plant dynamics that could affect the convergence, which may take much longer and require a smaller convergence factor.

The calculation of u_m can be difficult in some cases because the inverted plant could be unstable if the zeros of the plant are outside the unit circle. If this

occurs, a numerical technique can be applied: the zeros outside the unit circle can be replaced with zeros mirrored into the unit circle. By reflecting the zeros into the unit circle, a stable inverted model is created, as seen in Lu and Yahagi (1993).

Equations 4.2 through 4.6 define several key relationships needed to derive the weight update rule.

$$e_y = y_m - y \quad (4.1)$$

$$e = u_m - u \quad (4.2)$$

$$y_m = P_m x \quad (4.3)$$

$$y = Pu \quad (4.4)$$

$$u_m = P^{-1} y_m \quad (4.5)$$

$$u = CX^T W - CY^T W \quad (4.6)$$

The derivative of the weight update rule is parallel to the Widrow-Hoff equation and starts with minimizing the squared error with respect to the weight, as shown in Equation 4.7. The model controller output, u_m , is independent of the weights of the tap delay line.

$$\frac{e_k^2}{W_k} = 2e_k \frac{e_k}{W_k} = 2e_k \frac{1}{W_k} (u_m - u_k) = -2e_k \frac{u_k}{W_k} \quad (4.7)$$

In Equation 4.8, $\frac{u_k}{W_k}$ is broken down into its components.

$$\frac{u_k}{W_k} = \frac{1}{W_k} [CX^T W_k - CY^T W_k] = CX^T - CY^T - W_k C \frac{Y^T}{W_k} \quad (4.8)$$

Assume that the plant, P, has the difference equation description shown in Equation 4.9.

$$y(k) = -b_1 y(k-1) - b_2 y(k-2) \dots - b_l y(k-l) + a_0 u(k) + a_1 u(k-1) + \dots \quad (4.9)$$

In Equation 4.10, a substitution is made for Y^T using Equation 4.8, which allows the break-down of the vector, as seen in Equation 4.11.

$$\frac{Y^T}{W_k} = \frac{1}{W_k} \begin{matrix} y(k) \\ \cdot \\ \cdot \\ \cdot \\ y(k-L) \end{matrix} = \frac{1}{W_k} \begin{matrix} -b_1 y(k-1) - b_2 y(k-2) \dots + a_0 u(k) + \dots \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{matrix} \quad (4.10)$$

$$\frac{y(k)}{W_k} = -b_1 \frac{y(k-1)}{W_k} - b_2 \frac{y(k-2)}{W_k} \dots + a_0 \frac{u(k)}{W_k} + a_1 \frac{u(k-1)}{W_k} \quad (4.11)$$

The assumption that the direct feed-through term a_0 is zero eliminates $\frac{u(k)}{W_k}$ from the right side of Equation 4.11. The rest of the terms on the right side of Equation 4.11 have been calculated in previous iterations. The weight adjustment of Equation 4.7 can be added to the existing weights. All of the terms needed to update the weights have been calculated because the weights are adjusted to minimize the mean square error, as seen in Equation 4.12.

$$W_{k+1} = W_k + 2\mu e_k \frac{u_k}{W_k} \quad (4.12)$$

The weight update rule, as seen in Equation 4.12, has a form similar to the Widrow-Hoff update, but it is more complicated than the Widrow-Hoff weight update rule. The Widrow-Hoff weight update rule, seen in Equation 4.13, is a simple exchange of the X vector with the $\frac{u(k)}{W_k}$ term in the equation developed above. The individual parts of the $\frac{u(k)}{W_k}$ term can be calculated or have been calculated on previous iterations.

$$W_{k+1} = W_k + 2\mu e_k X_k \quad (4.13)$$

4.2 Results For First Update Algorithm

Two simple, second-order plants with a single zero are chosen for the examples. The first plant is stable; the second plant is unstable. The weights of the tap delay had to be initialized to a point that the closed-loop system is stable in order to allow stable convergence of the algorithm. The training input set is discrete white noise. For both cases, the model plants are determined by analytically finding solutions to the closed loop plants given an ideal, two-weight tap delay line. Two weights are chosen for the tap delay line, and the closed-loop solution is found, which then is used as the model plant.

The first example has a stable plant. The open-loop transfer function can be seen in Equation 4.14. Two weights for the tap delay are chosen, 0.9 and -0.7, respectively, as seen in Equation 4.15. The controller, C , for this example is a proportional gain equal to one. The closed loop transfer function for the chosen tap delay line and the model plant can be seen in Equation 4.16.

$$P = \frac{y}{u} = \frac{z - 0.4}{z^2 - 1.2z + 0.72} \quad (4.14)$$

$$W = \frac{u}{x-y} = \frac{0.9z - 0.7}{z} \quad (4.15)$$

$$P_m = \frac{y}{x} = \frac{y_m}{x} = \frac{0.9z^2 - 1.06z + 0.28}{z^3 - 0.3z^2 - 0.34z + 0.28} \quad (4.16)$$

Two weights are all that will be needed in order to make the closed-loop system behave like the model plant. However, for this example, a tap delay line with five weights is used to demonstrate that the extra weights in the tap delay would go to zero. The weights are initialized to zero.

The difference between the model output, y_m , and the plant output, y , can be seen in Figure 4.2. The difference converges to zero with a white noise input. The weights for the same simulation can be seen in Figure 4.3. The weights converge to their expected values of 0.9, -0.7, 0.0, 0.0, and 0.0, respectively.

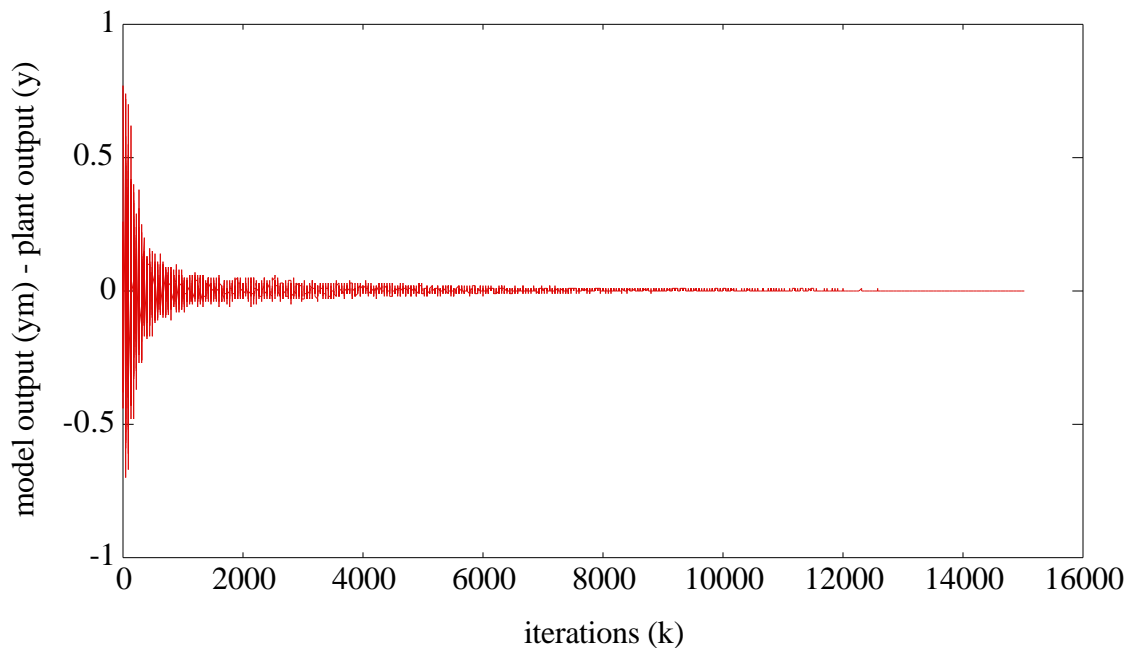


Figure 4.2 Convergence to Reference Model for the Stable Example

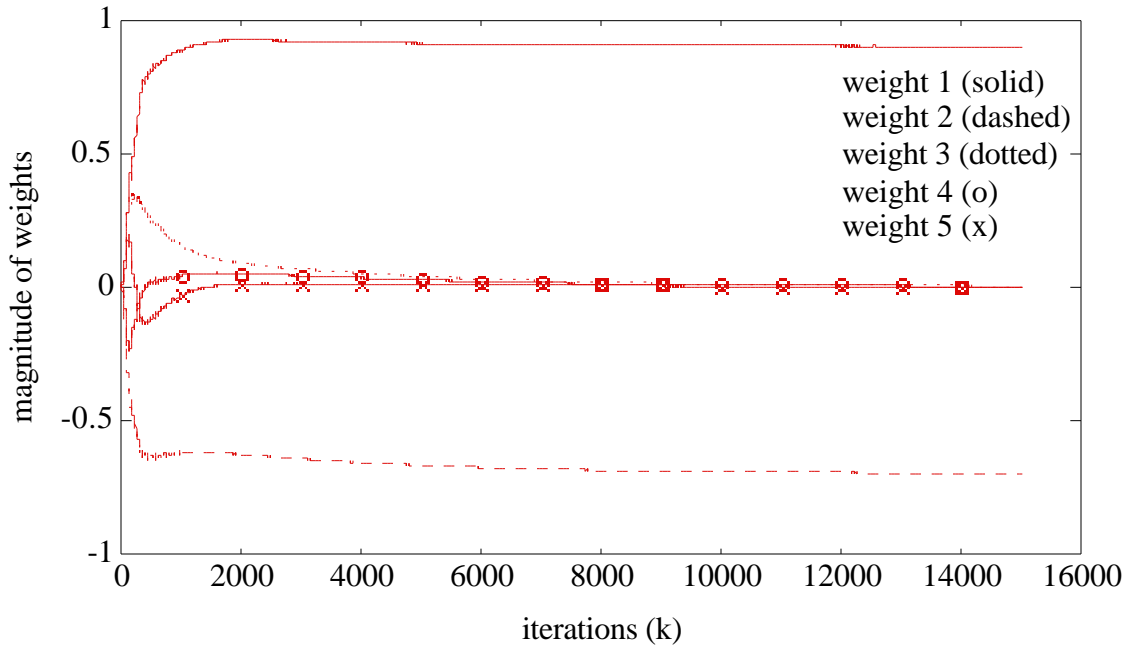


Figure 4.3 Convergence of Weights to Analytical Values for the Stable Example

The second example for the closed-loop scheme is an unstable plant. The transfer function for the plant can be seen in Equation 4.17. Like the first example, two weights are chosen for the tap delay line, and the model plant is the closed-loop dynamics of the plant and the tap delay line. The transfer function of the tap delay line can be seen in Equation 4.18. The fixed-gain controller, C , is a proportional gain of one. The transfer function for the model plant and the closed loop system can be seen in Equation 4.19. For this example, the weights cannot be initialized to zero because the plant is unstable. In order to stabilize the plant, the initial value for the first weight is 1.0. The other weights are initialized to 0.0.

$$\frac{y}{u} = \frac{z - 0.3}{z^2 - 1.6z + 1.28} \quad (4.17)$$

$$\frac{u}{x - y} = \frac{0.8z - 0.7}{z} \quad (4.18)$$

$$\frac{y}{x} = \frac{y_m}{x} = \frac{0.8z^2 - 0.94z + 0.21}{z^3 - 0.8z^2 + 0.34z + 0.21} \quad (4.19)$$

Two weights are required to get the closed loop system to mimic the model plant. Five weights are used to demonstrate that the extra weights would converge to zero. The difference between the model output, y_m , and the plant output, y , is seen in Figure 4.4, and the difference converges to zero. The weights converge to the expected values of 0.8, -0.7, 0.0, 0.0, and 0.0, respectively, seen in Figure 4.5.

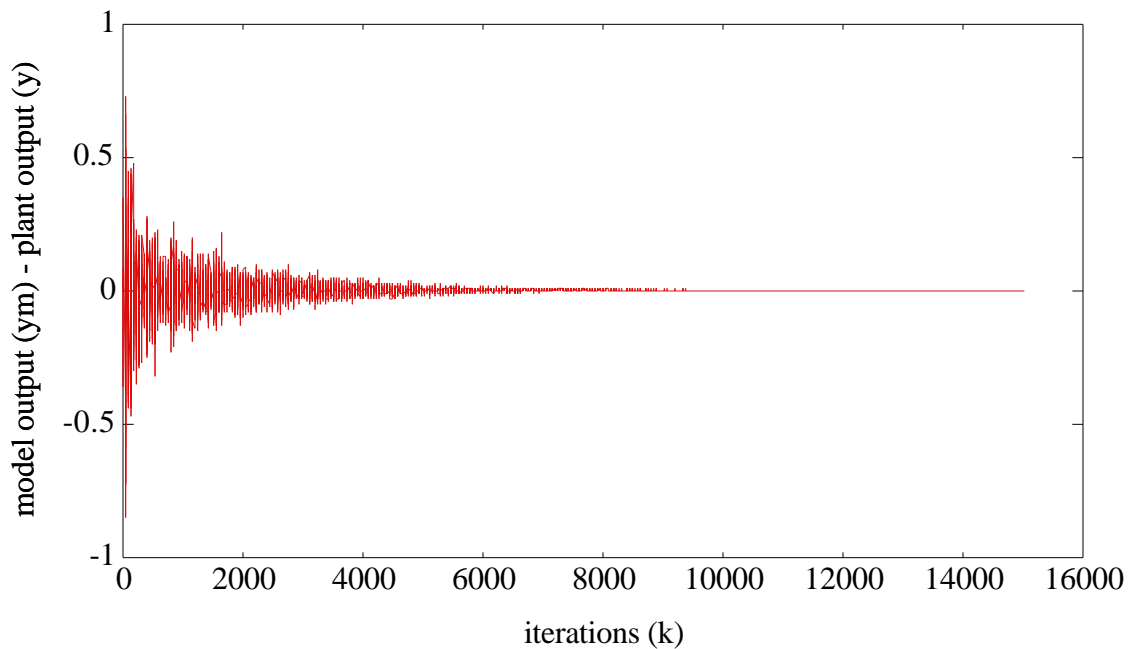


Figure 4.4 Convergence to Reference Model for the Unstable Example

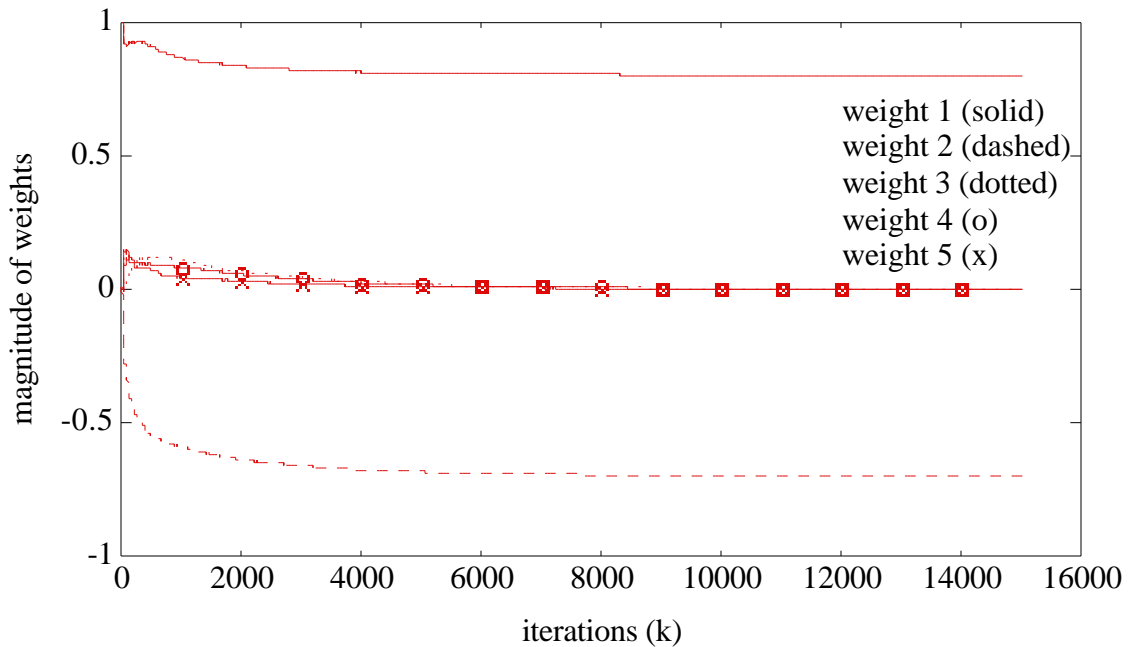


Figure 4.5 Convergence of Weights to Analytical Values for the Unstable Example

The examples used above assume the plants were known exactly; however, this is rarely the case. The stable example with the closed-loop scheme is run again with a change in the plant: a 50% reduction in the damping ratio. The transfer function for the actual plant is seen in Equation 4.20, and the transfer function for the inaccurate model of the plant is seen in Equation 4.14. The previous exact analytical solution will no longer be valid because of the 50% reduction in the damping ratio of the plant. The weights are started from two different positions. The first starting position for all five of the weights is zero. There is not an exact match between the reference model and the closed-loop adapted system, as seen in Figure 4.6. The difference for the FIR output ($u_m - u$) is driven to zero, as seen in Figure 4.7, while the difference between the plant outputs maintain an error. This is due to the inaccurate modeling of the system. The weights converge to 0.89, -0.71, 0.09, 0.00, and 0.00, as seen in Figure 4.8. The second starting position for the five weights is 0.9, -0.7, 0.0, 0.0, and 0.0, which is the solution to the previous stable example. The weights converge to the same solution as they did with the first position. This adaptive method can

start from the mathematically-calculated solution, which is limited because of inaccurate modeling, and can improve upon the existing controller. A comparison of the effectiveness of the old controller, as seen in Equation 4.15 and the newly-adapted controller can be seen in Figure 4.9. The solid line is the difference between the model output, y_m , and the plant output, y , associated with the old controller and the dashed line is the difference between the model output, y_m , and the plant output, y , associated with the new controller. The figure is generated with white noise as the input. The mean squared error between the model output and the plant output is reduced by more than 80%.

$$\frac{y}{u} = \frac{z - 0.4}{z^2 - 1.3z + 0.845} \quad (4.20)$$

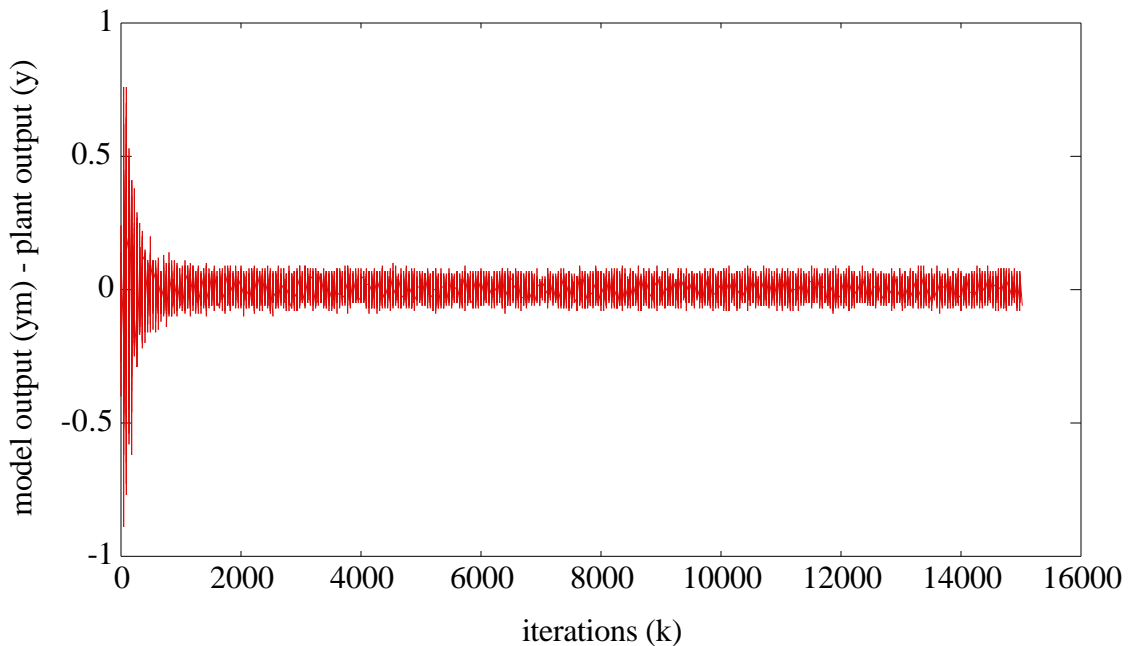


Figure 4.6 Convergence to the Reference Model for the Inaccurate Model Example

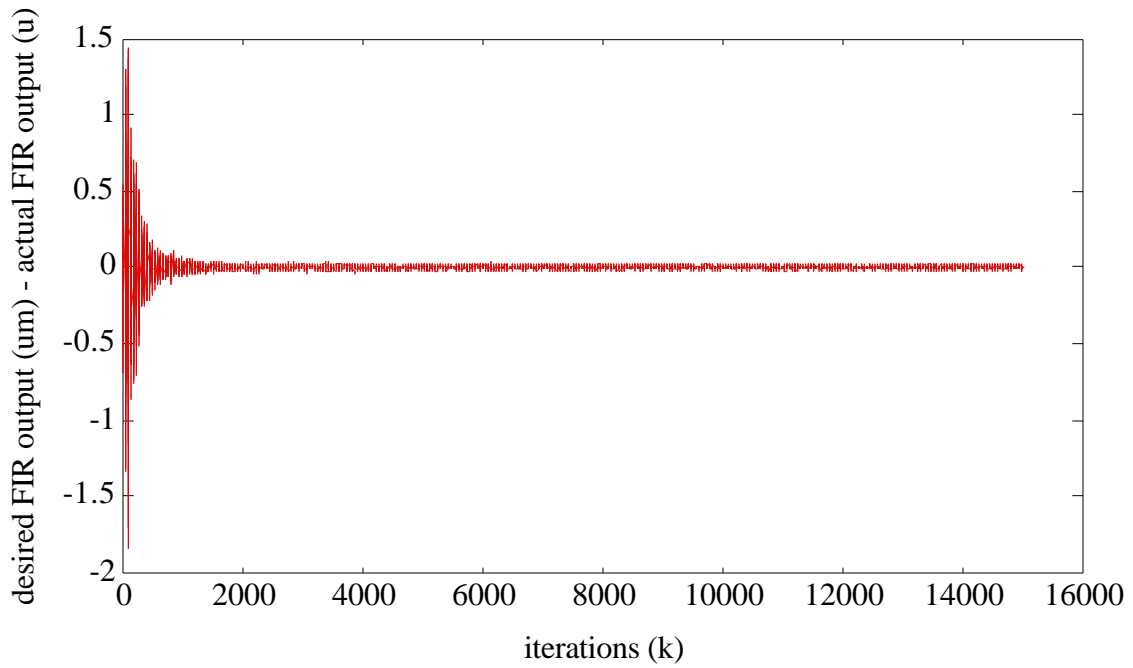


Figure 4.7 Convergence to the FIR Output for the Inaccurate Model Example

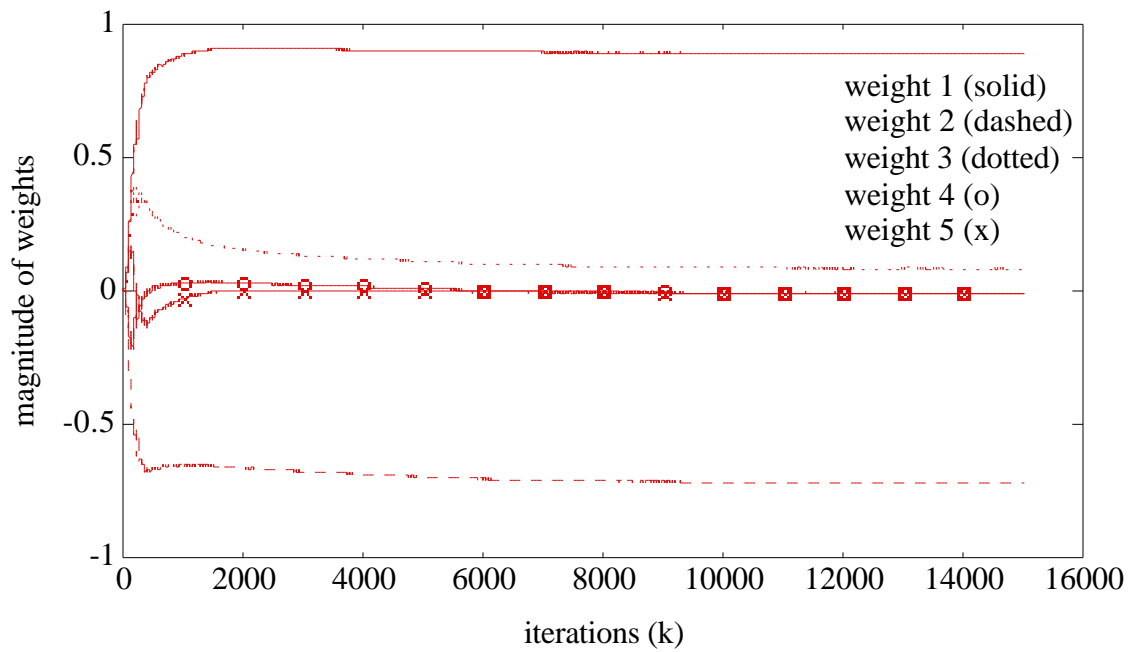


Figure 4.8 Convergence of the Weights for the Inaccurate Model Example

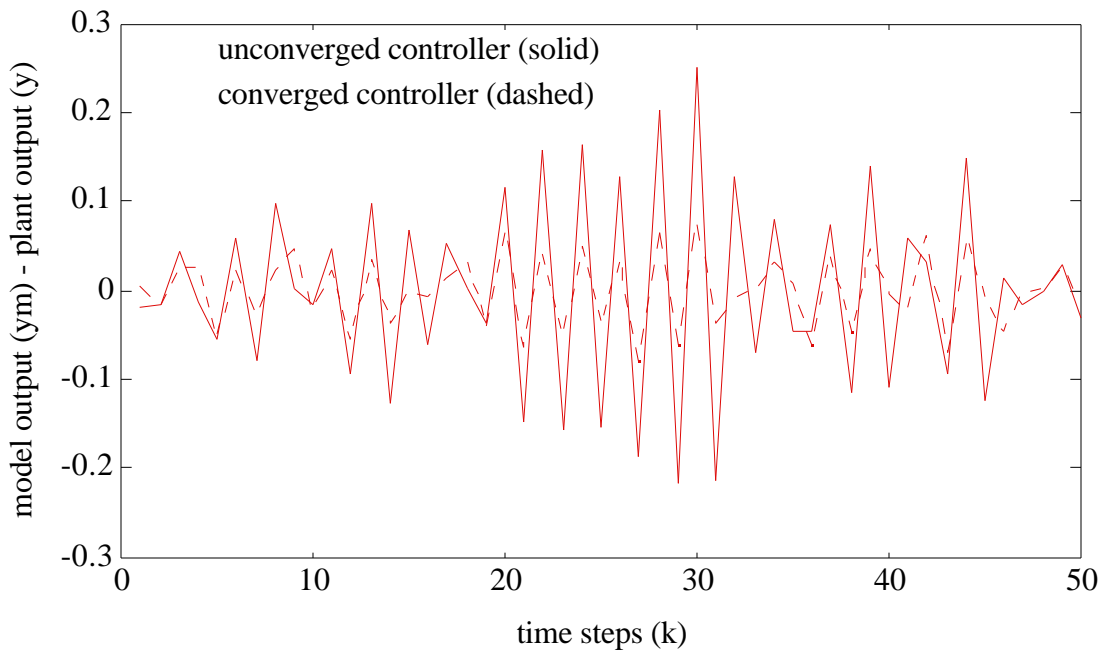


Figure 4.9 A Comparison of the Effectiveness of Two Controllers for the Inaccurate Model

The next two examples, one stable and one unstable, are developed for the scenario when the controller is designed with an inaccurate model. If the models are exact, the system will respond exactly like the reference model. Two fourth-order systems are used, and in each example, the damping ratio for the stable pairs of poles is reduced by 50%. The effectiveness of the fixed gain controller is then reduced. The FIR filter inside the closed loop adapts to reduce the error between the reference model and output of the plant.

The modeled transfer function for the stable example with fixed gain controller can be seen in Equation 4.21. The actual plant has a 50% reduction in damping in both sets of poles, as seen in Equation 4.22. The transfer function for the fixed gain controller can be seen in Equation 4.23. The reference model is the closed-loop combination of the fixed gain controller and the modeled plant, seen in Equation 4.24. If the plant had been modeled perfectly, the weights of the FIR filter would have remained at their initial values. Their initial values are set at zero except for the first weight, which was set at one. This is

the FIR filter equivalent of a feed-through neural network. Initially, the output of the FIR filter is equal to the input of the FIR filter.

$$P = \frac{y}{u} = \frac{z^3 - 1.55z^2 + 0.91z - 0.323}{z^4 - 1.2z^3 + 0.72z^2 - 1.2z + 0.01} \quad (4.21)$$

$$P_{ACTUAL} = \frac{y}{u} = \frac{z^3 - 1.55z^2 + 0.91z - 0.323}{z^4 - 1.4z^3 + 1.22z^2 - 0.456z + 0.1872} \quad (4.22)$$

$$C = \frac{f}{x - y} = \frac{0.2(z - 0.2)}{z - 0.9} \quad (4.23)$$

$$P_m = \frac{y}{x} = \frac{y_m}{x} = \frac{0.2(z^4 - 1.75z^3 + 1.22z^2 - 0.505z + 0.0646)}{z^5 - 1.9z^4 + 1.45z^3 - 0.524z^2 + 0.017z + 0.0039} \quad (4.24)$$

The results for the stable example with a fixed-gain controller were excellent. The closed-loop system converges to the reference model, as seen in Figure 4.10. This correlates to the convergence of the FIR filter output to the ideal FIR filter output, as seen in Figure 4.11. The final values of the weight can be seen in Figure 4.12. The weights further down on the tap delay line are very small, but without all twenty weights, the system did not converge to the reference model. When the weights are inside the closed loop, they have a greater effect on the system dynamics.

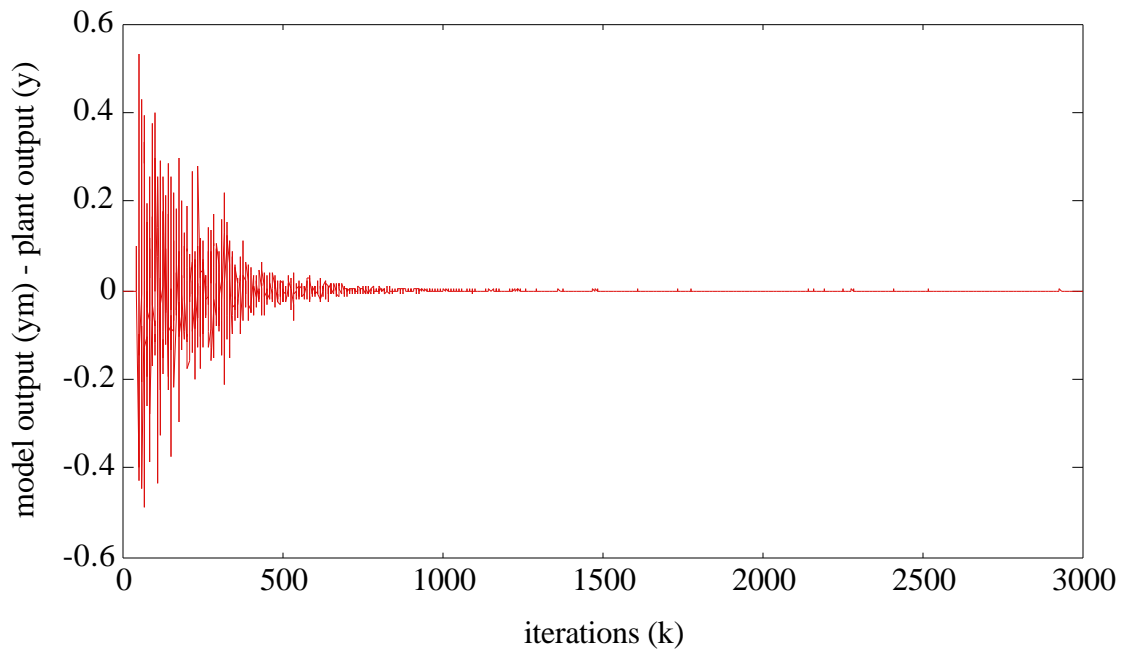


Figure 4.10 Convergence to Reference Model for Stable Example with Fixed Gain Controller

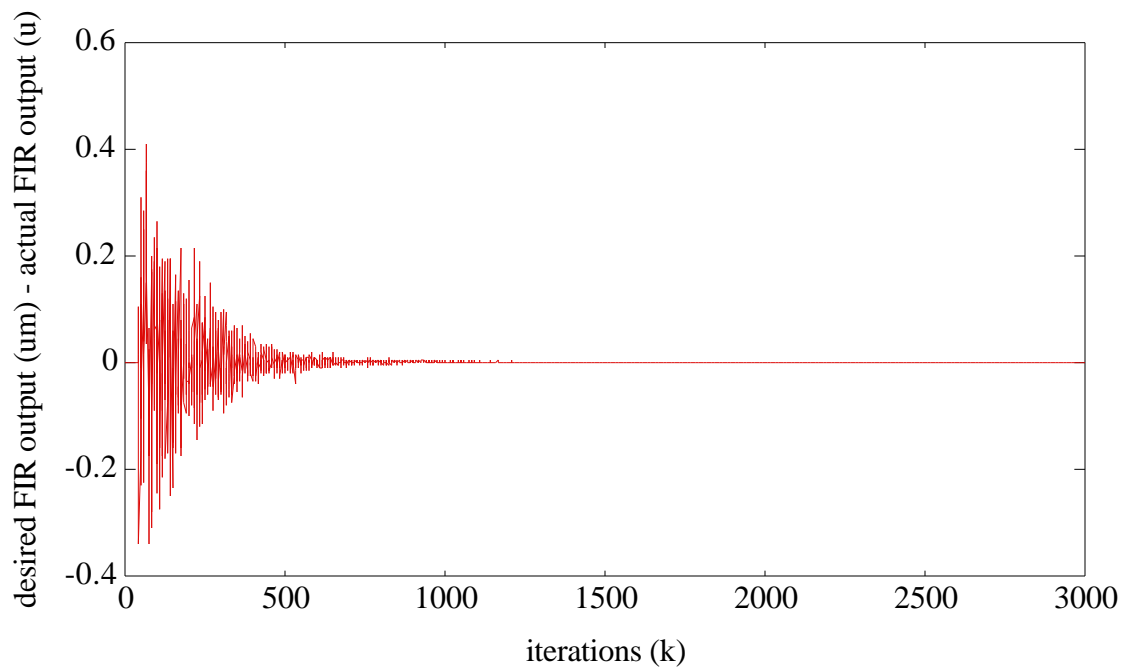


Figure 4.11 Convergence of FIR Output for Stable Example with Fixed Gain Controller

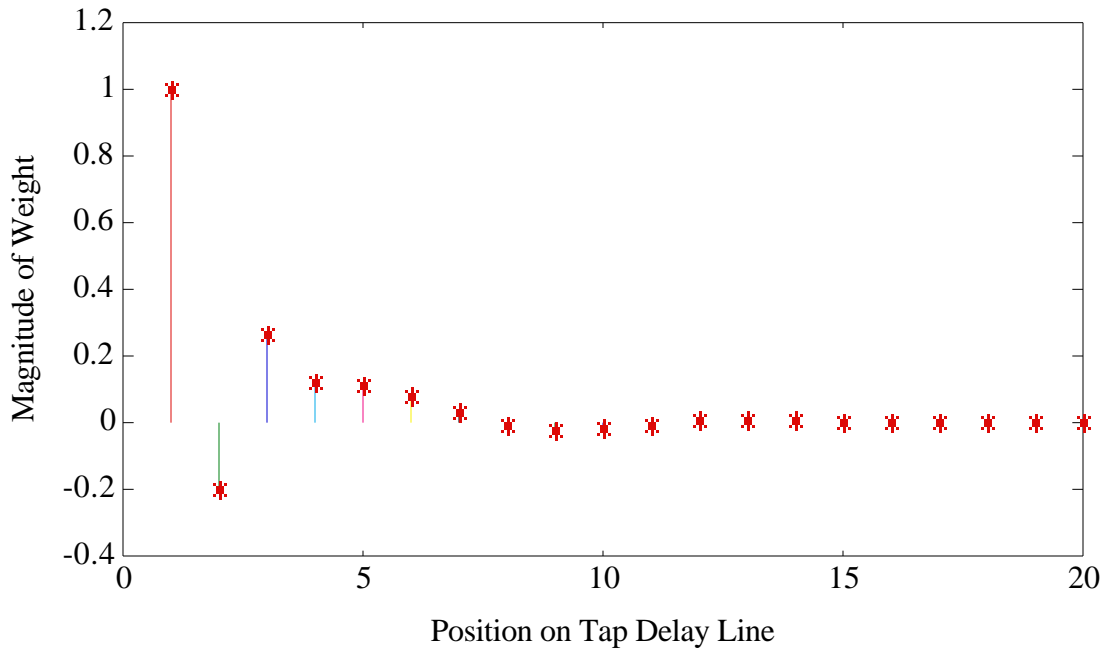


Figure 4.12 Final Weight Values for Stable Example with Fixed Gain Controller

The unstable example is a fourth order system with a fixed gain controller. The plant has one pair of poles outside the unit circle and one pair inside the unit circle. The damping ratio is reduced by 50% for the stable pair of poles. The transfer function for the imperfect plant model can be seen in Equation 4.25; the transfer function for the actual plant can be seen in Equation 4.26. The fixed-gain controller can be seen in Equation 4.27. The reference model for the closed-loop system can be seen in Equation 4.28. If the actual plant was modeled perfectly, there would be no need for the FIR filter because the reference model would match the closed-loop system.

$$P = \frac{y}{u} = \frac{z^3 - 1.55z^2 + 0.91z - 0.323}{z^4 - 1.7z^3 + 1.445z^2 - 0.255z + 0.0225} \quad (4.25)$$

$$P_{ACTUAL} = \frac{y}{u} = \frac{z^3 - 1.55z^2 + 0.91z - 0.323}{z^4 - 1.7z^3 + 1.685z^2 - 0.615z + 0.2925} \quad (4.26)$$

$$C = \frac{f}{x-y} = \frac{1.2(z-0.2)}{z-0.9} \quad (4.27)$$

$$P_m = \frac{y}{x} = \frac{y_m}{x} = \frac{1.2(z^4 - 1.75z^3 + 1.22z^2 - 0.505z + 0.0646)}{z^5 - 1.4z^4 + 0.875z^3 - 0.0915z^2 - 0.354z + 0.0573} \quad (4.28)$$

The results of the unstable example with a fixed gain controller are excellent. The closed-loop system almost converges exactly to the reference model, as seen in Figure 4.13. The FIR filter output corresponding converges to the ideal FIR filter output, as seen in Figure 4.14. The final weights for the FIR filter can be seen in Figure 4.15.

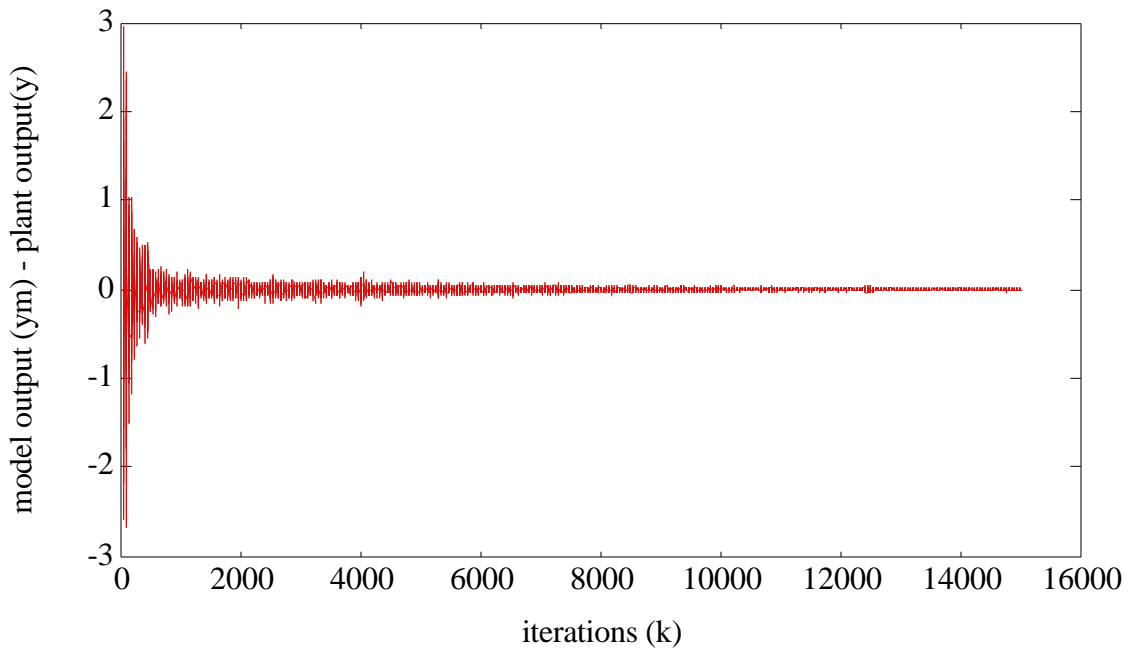


Figure 4.13 Convergence to Reference Plant for Unstable Example with Fixed Gain Controller

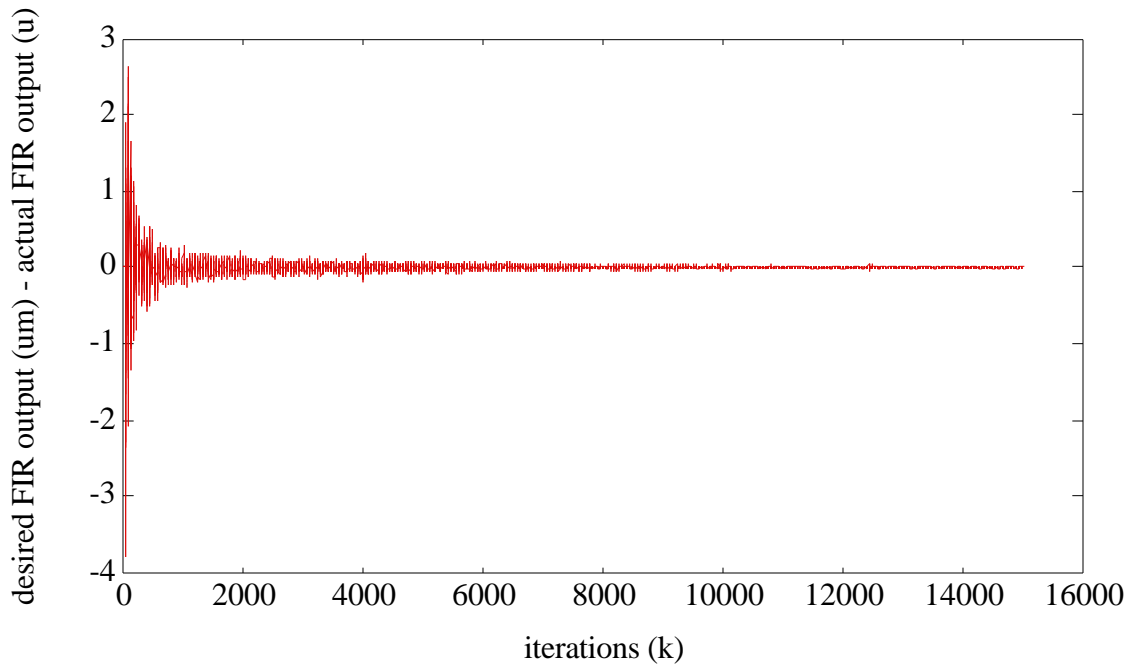


Figure 4.14 Convergence of the FIR Output for Unstable Example with Fixed Gain Controller

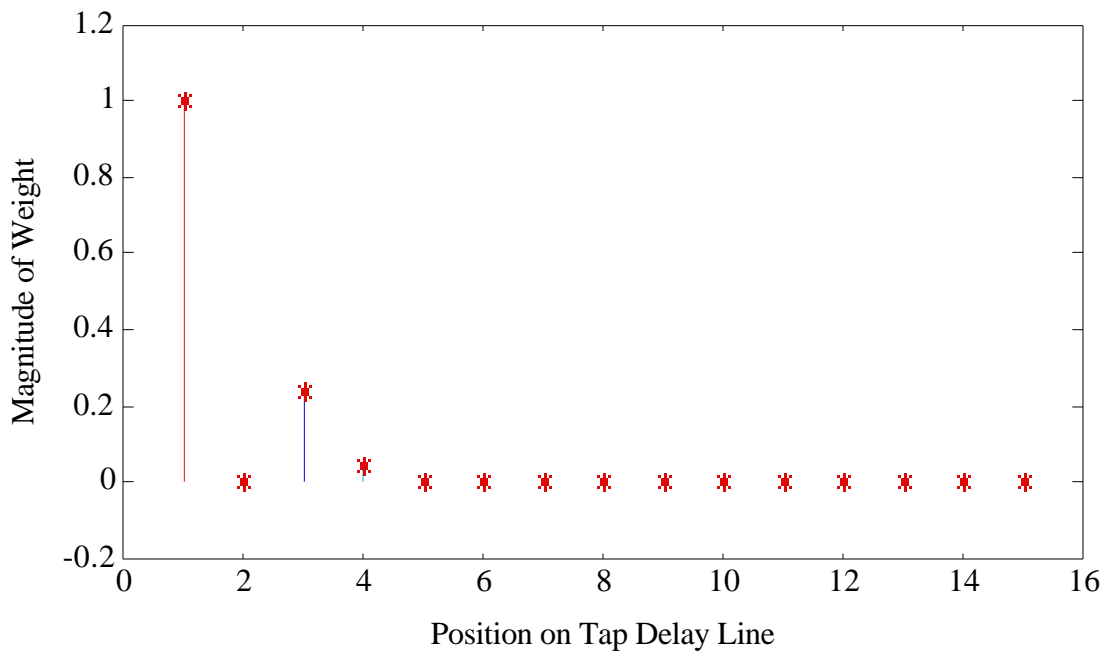


Figure 4.15 Final Weight Values for Unstable Example with Fixed Gain Controller

The new update algorithm works very well with one exception. It did not work well in the imperfectly-modeled plant example with no fixed-gain controller. The other examples had the closed-loop system converge to the reference model. The first update algorithm is based on the minimization of the difference between the FIR filter output and the Ideal FIR filter output. The logic is to improve convergence by eliminating the interference associated with the magnitude and phase of the plant. The system for all of the examples had the error associated with the FIR filter convergence, but this did not mean that the closed-loop system converges to the reference model, as was the case in the imperfectly-modeled example with no fixed gain controller. Although the difference between the FIR filter output and the ideal FIR filter output goes to zero, the error between the closed-loop system and the reference model does not converge to zero. There is a discrepancy between the two different errors because the plant was not perfectly modeled. The problem associated with minimizing this variable is that it is not the primary variable of interest when the system is not perfectly modeled. In the next section, another algorithm is derived for minimizing the error between the closed-loop system and the reference model with the FIR filter inside the closed-loop.

4.3 Second Update Algorithm

In the previous section, it was shown that an update algorithm could be developed to converge an FIR filter inside a closed loop. The problem with the update algorithm in the previous section was that the error between the closed-loop system and the reference model did not always converge to zero. This is because the first update algorithm was derived to minimize an error between the output of the FIR filter and an ideal output of the FIR filter. When the plant is not perfectly modeled, the error for the FIR filter can be zero, and there can still be an error between the closed-loop system and the reference model. The inverse model of the plant is required to find the ideal output of the FIR filter, and it can be unstable. These two problems can make the first update algorithm difficult to manage.

An alternative algorithm can be developed by minimizing the error between the plant output and the reference model output. This algorithm would minimize the primary variable of interest and would not require an inverse model. The negative aspect would be that there would still be some problems with amplitude and phase associated with the plant's dynamics. The algorithm would be valid for an FIR filter inside the closed loop. Figure 4.16 is the block diagram for the control methodology.

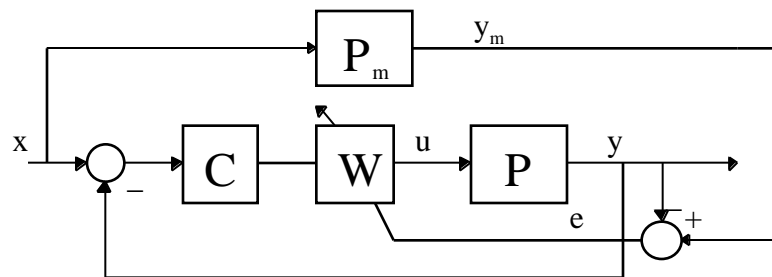


Figure 4.16 Block Diagram of Closed-Loop Control Scheme

The second algorithm is based on the difference between the output of the closed-loop system and the output of the reference plant, as seen in Equation 4.29. The update algorithm is derived for the FIR filter inside the closed loop. This is different than LMS because the LMS algorithm assumes independence between the input to the FIR filter and the weights of the FIR filter. This assumption is not true when the FIR filter is inside the closed loop. The derivation is very similar to the derivation of the first update algorithm. Equations 4.29 and 4.31 are key relationships needed for its derivation.

$$e_y = y_m - y \quad (4.29)$$

$$y_m = P_m x \quad (4.30)$$

$$y = Pu \quad (4.31)$$

The derivation of the weight update rule is parallel to the derivation of the Widrow-Hoff equation and will start with minimizing the squared error with respect to the weight, as shown in Equation 4.32. The reference model output, y_m , is independent of the weights of the tap delay line.

$$\frac{e_k^2}{W_k} = 2e_k \frac{e_k}{W_k} = 2e_k \frac{1}{W_k} (y_m - y_k) = -2e_k \frac{y_k}{W_k} \quad (4.32)$$

Assume that the plant P has the difference equation in the form shown in Equation 4.33. The partial differential is carried through the difference Equation, as seen in Equation 4.34.

$$y(k) = -b_1 y(k-1) - b_2 y(k-2) \dots - b_l y(k-l) + a_0 u(k) + a_1 u(k-1) + \dots \quad (4.33)$$

$$\frac{y(k)}{W_k} = -b_1 \frac{y(k-1)}{W_k} - b_2 \frac{y(k-2)}{W_k} \dots + a_0 \frac{u(k)}{W_k} + a_1 \frac{u(k-1)}{W_k} \quad (4.34)$$

The output of the FIR filter, u , which is the input into the plant, can be broken down into its components, as seen Equation 4.35.

$$u = CX^T W - CY^T W \quad (4.35)$$

The partial derivation of the FIR filter output with respect to the weights can be seen in Equation 4.36. The input into the closed-loop system, x , is assumed to be independent of the weights.

$$\frac{u_k}{W_k} = \frac{1}{W_k} [CX^T W_k - CY^T W_k] = CX^T - CY^T - W_k C \frac{Y^T}{W_k} \quad (4.36)$$

The substitution of Y^T into Equation 4.34 results in Equation 4.37.

$$\frac{Y^T}{W_k} = \frac{\begin{matrix} y(k) \\ \cdot \\ \cdot \\ \cdot \\ y(k-L) \end{matrix}}{W_k} = \frac{\begin{matrix} -b_1y(k-1) - b_2y(k-2) \dots + a_0u(k) + \dots \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{matrix}}{W_k} \quad (4.37)$$

The assumption that the direct feed through term a_0 is zero eliminates the $\frac{u(k)}{W_k}$ term from the right side of Equations 4.34 and 4.37. The rest of the terms on the right side of Equation 4.11 have been calculated in previous iterations. All of the terms needed to update the weights have been calculated because the weights are adjusted to minimize the mean square error, as seen in Equation 4.38.

$$W_{k+1} = W_k + 2\mu e_k \frac{y_k}{W_k} \quad (4.38)$$

The update algorithm is ready to be implemented on the same examples from previous section, allowing a direct comparison of the effectiveness of the two different algorithms.

4.4 Results For The Second Algorithm

The second algorithm is implemented on the same five examples as the first algorithm. The first two examples are modeled exactly and have a simple analytical solution. They demonstrate that the algorithm converges the weights to their expected values for stable and unstable plants. The third example is the first example, revisited with an inaccurately modeled plant. The fourth and fifth examples are systems with fixed gain controllers and inaccurately-modeled plants, which is similar to the previous section using the first update algorithm. The reference model for the last two examples is the expected closed-loop

system if the plant had been modeled accurately. The last two examples are the most realistic examples; often there is a loss of performance because the plant is not accurately modeled.

The first example is a stable plant. The open-loop transfer function can be seen in Equation 4.14. The reference model and the closed-loop system will be equal when the first two weights on the tap delay line converge to 0.9 and -0.7, respectively, as seen in Equation 4.16. Similar to the first algorithm, five weights are used on the tap delay line, and they are initially started at zero. The expected transfer function of the tap delay line can be seen in Equation 4.15.

The algorithm converges the weights to the expected values. The error between the reference model output and the plant output, which is the primary variable of interest, did converge to zero, as seen in Figure 4.17. The first two weights converged to their expected analytical values of 0.9 and -0.7, respectively. The additional weights ultimately converge back to zero as their final value, as seen in Figure 4.18.

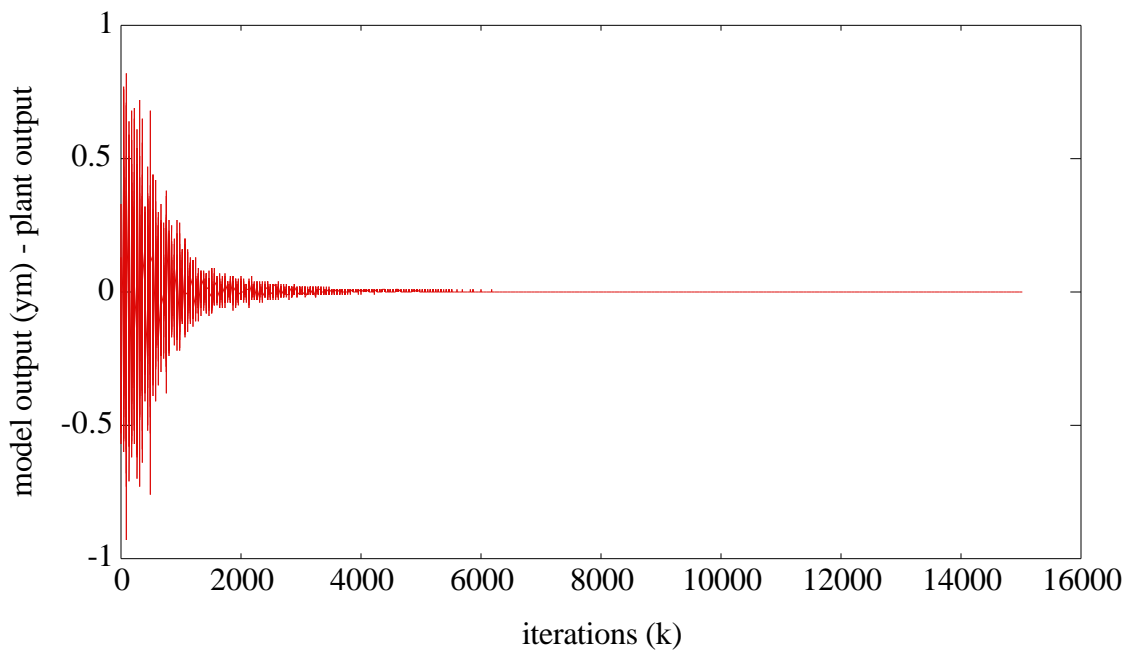


Figure 4.17 Convergence to Reference Model for the Stable Example

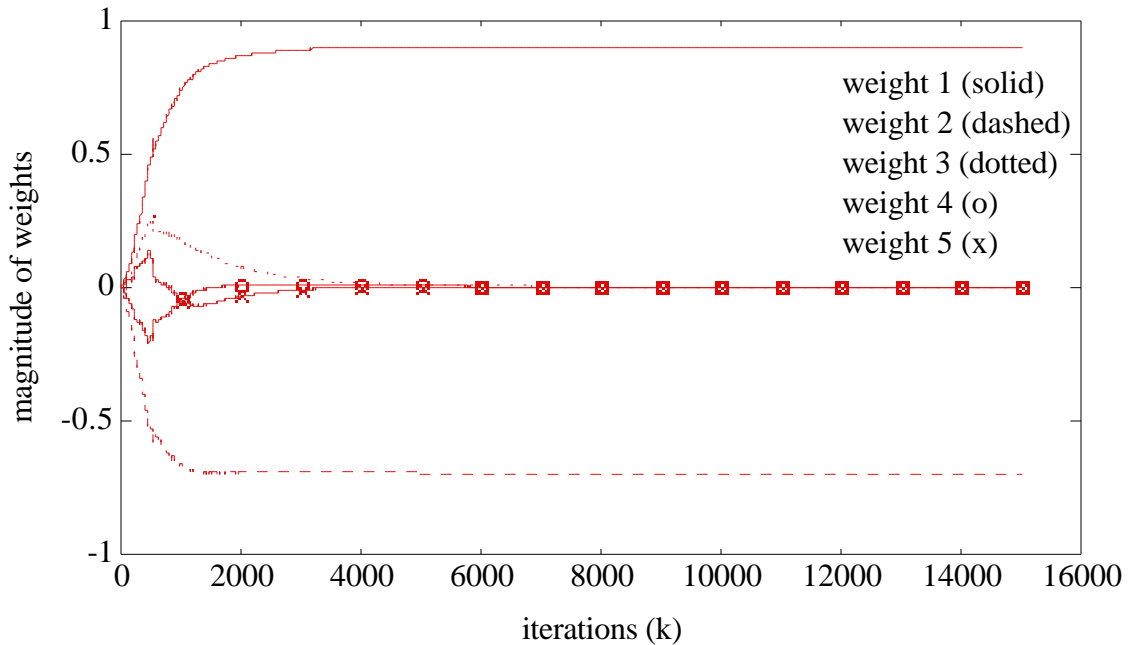


Figure 4.18 Convergence of Weights to Analytical Values for the Stable Example

The second example is an unstable plant with an analytical solution. The open-loop transfer function can be seen in Equation 4.17. The analytical values of the first two weights on the tap delay line are 0.8 and -0.7, respectively, as seen in Equation 4.18. The reference model is the closed-loop system when the tap delay line converges to its analytical values, as seen in Equation 4.19. Five weights are used in this example. The first weight is initially set to a value of one, and the rest are set at zero. This is necessary because the plant needs to be stabilized by feedback. With the weights initially set at zero, the feedback loop will have been effectively cut, and the unstable plant could have diverged before the weights had a chance to converge to the analytical solution.

The unstable example converges to the analytical values. The error between the reference model output and the plant output converges to zero, as seen in Figure 4.19; the weights converge to the expected values, as seen in Figure 4.20.

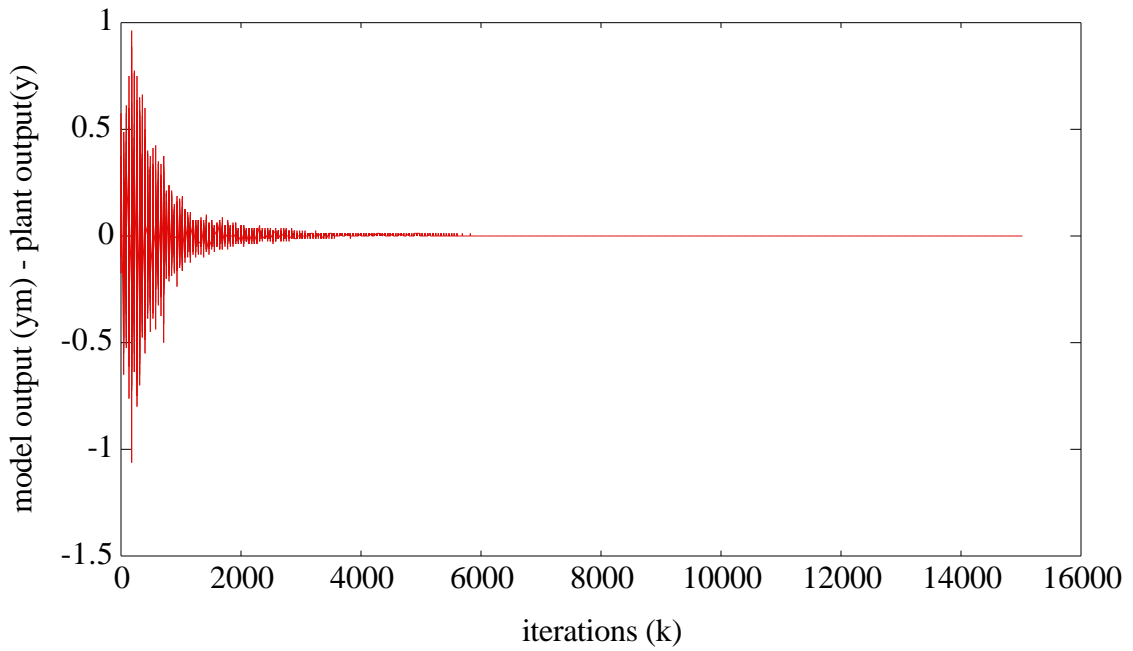


Figure 4.19 Convergence to Reference Model for the Unstable Example

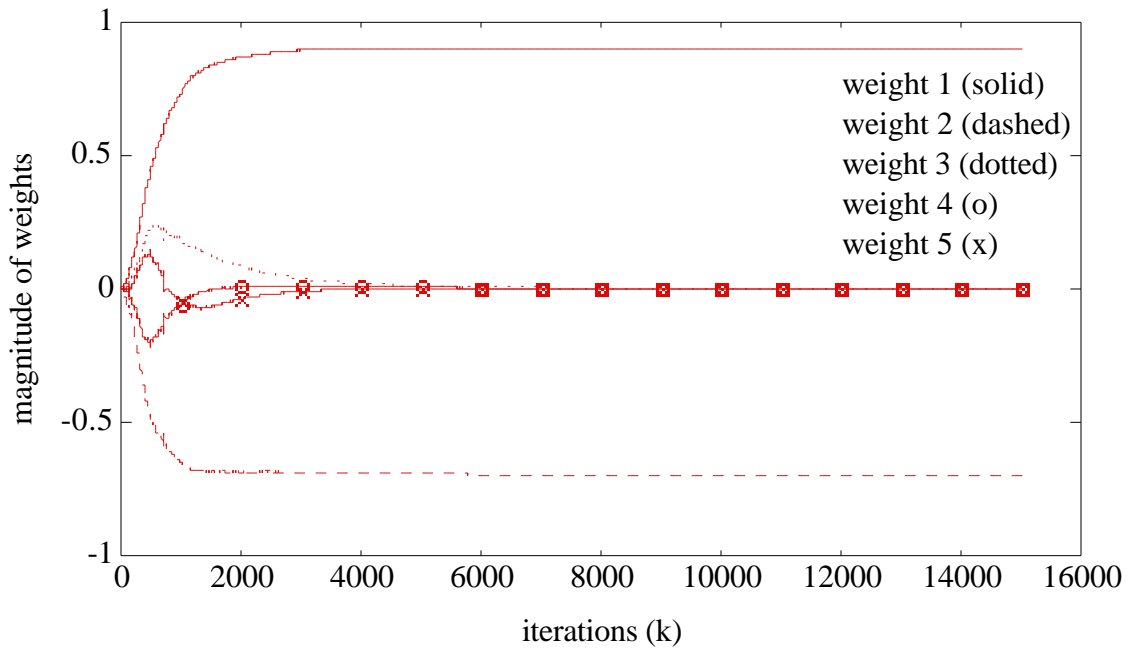


Figure 4.20 Convergence of Weights to Analytical Values for the Unstable Example

The third example is a stable plant that has been modeled inaccurately. From the modeled plant to the actual plant, the damping in the plant is reduced by 50%. The open-loop transfer function for the actual plant can be seen in Equation 4.20; the inaccurate transfer function can be seen in Equation 4.14. The expected values for this example would have been the same as the first stable example. These values are no longer valid because of the inaccurate model. Fifteen weights are used in the tap delay line to adapt to the reference model. All of the weights are initially started at zero.

The closed-loop system almost converged exactly to the reference model, as seen in Figure 4.21. This is a vast improvement over the performance of the first algorithm, which reduced the mean squared error by 80% but did not converge completely. The final weight values can be seen in Figure 4.22. The first two weights on the tap delay line from the previous stable example were 0.9 and -0.7, respectively. With the inaccurate model, the first weights on the tap delay line converged to 0.92 and -0.54, respectively. The algorithm performed very well with the inaccurate model.

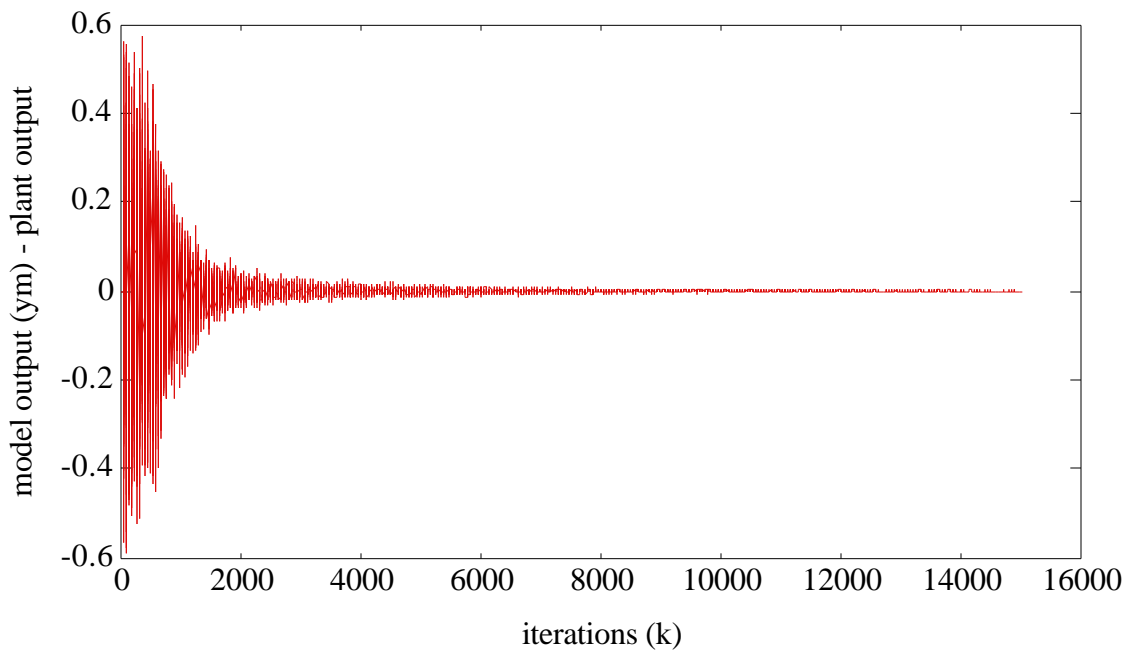


Figure 4.21 Convergence to Reference Model for the Inaccurate Model Example

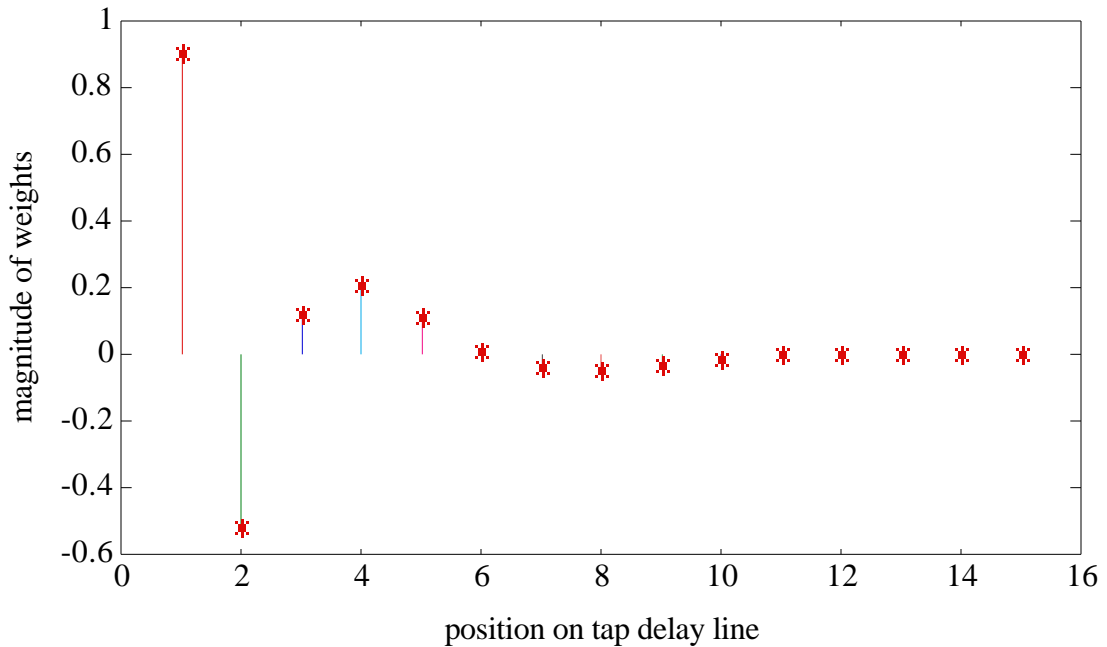


Figure 4.22 Final Value of Weights for the Inaccurate Model Example

A stable plant with a fixed gain controller is the fourth example. This plant is a fourth-order plant in which both pairs of poles have been inaccurately modeled. The transfer function for the plant can be seen in Equation 4.22. Each pair of poles has 50% less damping than had been modeled; the transfer function of the plant's model can be seen in Equation 4.21. If the system had been accurately modeled, the reference model and the closed-loop system would match exactly. The reference plant's transfer function and the fixed gain controller's transfer function can be seen in Equations 4.24 and 4.23, respectively. The first weight of the tap delay line is set to one. This has the same effect of the feed-through neural network and gives the closed-loop system the same initial performance as if the tap delay line did not exist.

The closed-loop system converges to the reference model, as seen in Figure 4.22. The tap delay line has twenty weights; their final values can be seen in Figure 4.23. The algorithm converges the weights of the tap delay line to eliminate the error between the reference model and the closed-loop system.

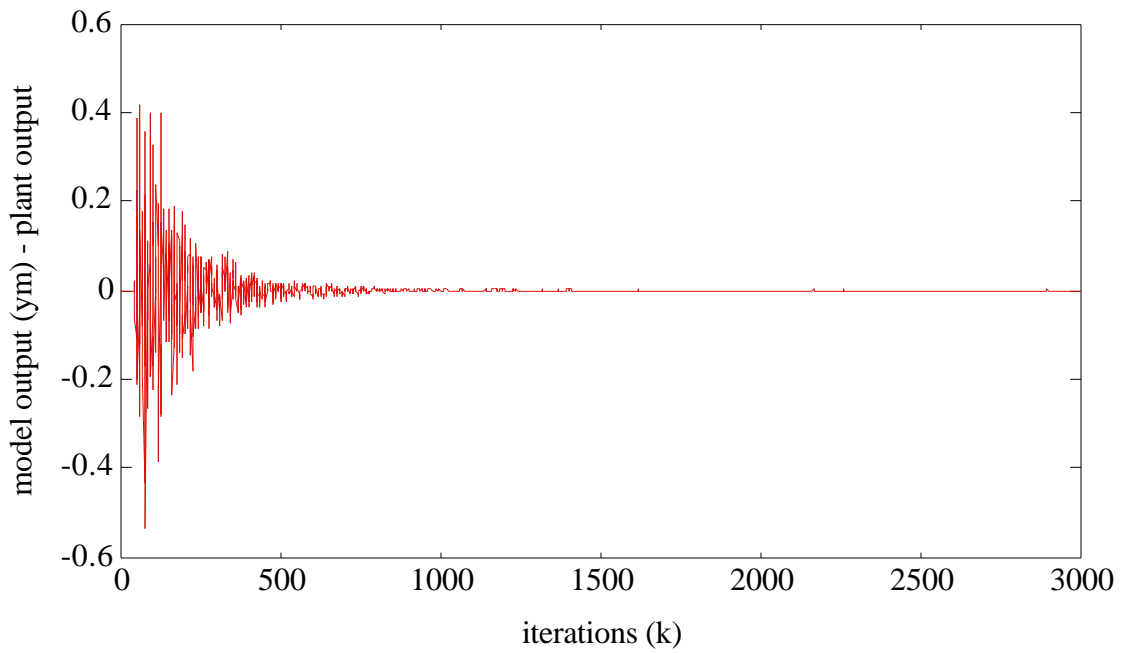


Figure 4.23 Convergence to Reference Model for the Stable Example with Fixed Gain Controller

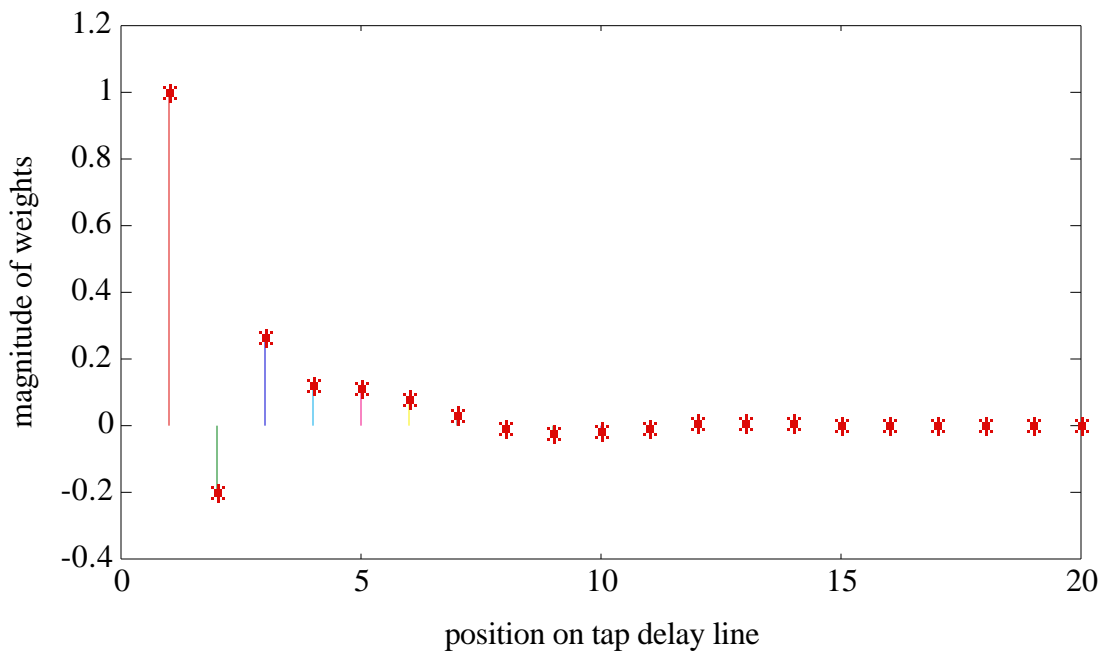


Figure 4.24 Final Value of Weights for the Stable Example with Fixed Gain Controller

The last example is an unstable plant with a fixed-gain controller. The transfer function of the plant can be seen in Equation 4.26. The plant is a fourth order plant with one pair of poles outside the unit circle. The second pair of poles is inside the unit circle. For the plant's model, the unstable poles are accurately modeled, and the stable pair of poles has the damping inaccurately modeled by 50%. The transfer function for the plant's model can be seen in Equation 4.25. The reference model would match the closed-loop system with the fixed gain controller exactly if the plant had not been inaccurately modeled. The reference model's and the fixed gain controller's transfer functions can be seen in Equations 4.28 and 4.27, respectively. The first weight of the tap delay line was set to one, initially. The example did have the requirement that the closed-loop system must be stable even though the system was inaccurately modeled.

The closed-loop system almost converges to the reference model, as seen in Figure 4.24. The final value of the weight of the tap delay line can be seen in Figure 4.25. Initially, the performance of the system would be equivalent to not having a tap delay line in the closed-loop. The error between the reference model and the closed-loop system is reduced to zero. While the weights are converging, the system is going through a constant increase in performance.

In all five examples, the second algorithm converges the weights of the tap delay line until the performance of the closed-loop system matches or comes very close to matching the reference model. The examples vary from an accurately-modeled stable example with a known analytical solution to an inaccurately-modeled unstable example with an unknown solution. Overall, the second algorithm worked very well.

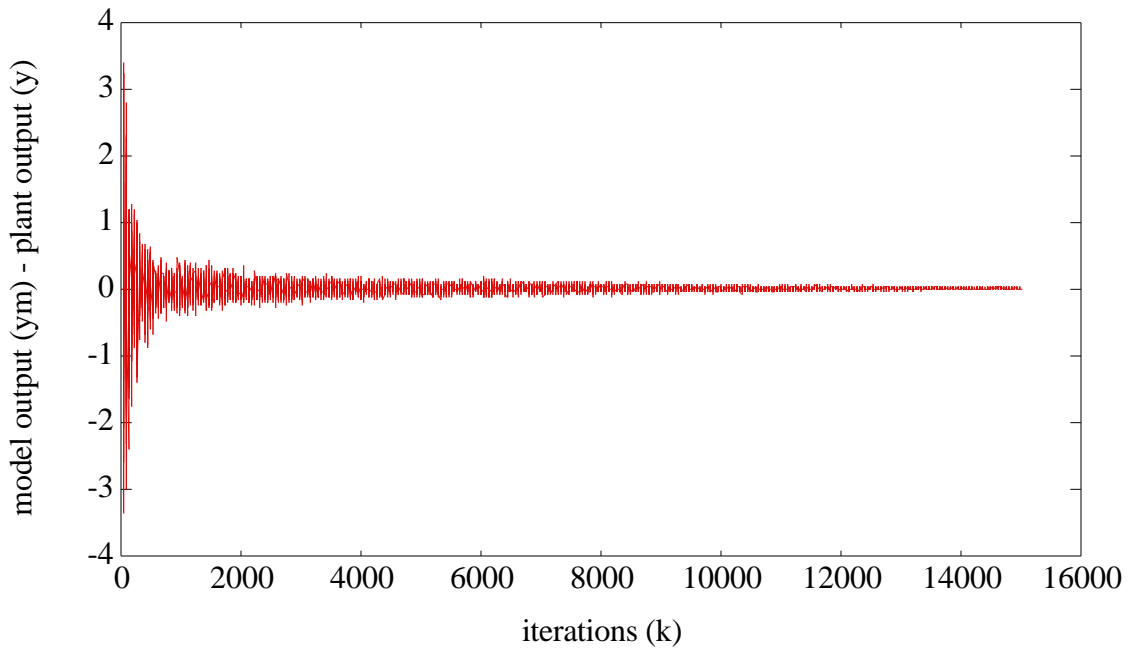


Figure 4.25 Convergence to Reference Model for the Unstable Example with Fixed Gain Controller

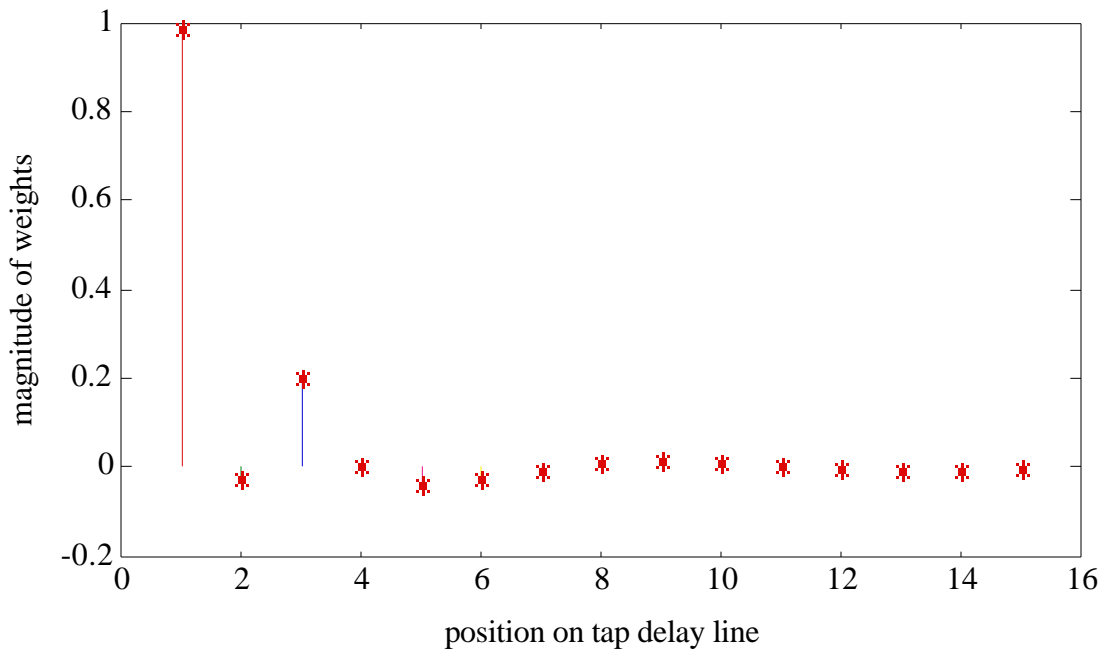


Figure 4.26 Final Value of Weights for the Unstable Example with Fixed Gain Controller

4.4 Summary

By placing an adaptive FIR filter inside a closed-loop, the performance of the closed-loop system can be adapted to a reference model. The standard LMS update algorithm does not accommodate the feedback through the FIR filter, and the open-loop configuration of the LMS does not lend itself to reference-model tracking and control of unstable plants. Two different algorithms have been developed for the FIR filter inside the closed-loop. The two update algorithms have been shown to work very well under a variety of situations.

The first update algorithm is derived to minimize the error between the output of the FIR filter and idealized output of the FIR filter. The idealized output of the FIR filter requires the inverse model of the plant. By using this error, the magnitude and phase associated with the plant does not have as great of an effect on the convergence process. The algorithm converges the weights of the tap delay line to the known analytical solution. Although the update algorithm works to varying degrees of success, it has two potential problems: the first problem is that there is not always a good inverse plant readily available to be used; the second problem is that the difference between the output of the FIR filter and the idealized output of the FIR filter is not the primary variable of interest. The primary variable of interest is the difference between the plant's output and the reference model's output, and this is the variable that is to be minimized. This can be seen in the stable example that was inaccurately modeled. Although the difference between the ideal output of the FIR filter and the actual output of the FIR filter did go to zero, the difference between the reference model's output and the plant's output did not go to zero. The overall success of the update algorithm does, however, make it a viable solution for this class of problems.

The second update algorithm is derived to minimize the error between the output of the plant and the reference model. When LMS is used in control applications, this same error is used. By minimizing this error, the primary variable of interest is being minimized. This update algorithm does not also

require an inverse model of the plant. The second update algorithm addresses the two problems associated with the first update algorithm. However, it does not address the problem associated with the magnitude and phase of the plant that can affect the convergence process. The second update algorithm is very successful in all of the examples. The difference between the reference model and the plant converges to zero for each example.

The two algorithms are successful with all of the examples given. Obviously, the second algorithm has more success with one of the five examples than the first algorithm. However, the first algorithm converges more quickly. The opportunity to develop adaptive controllers for unstable plants is one of the most promising aspects of these algorithms. Initially, an unstable plant must be stabilized with feedback. However, the performance of an unstable system can be improved greatly through this convergence process.

The algorithms are very successful with linear plants. New algorithms need to be derived for nonlinear plants. The combination of the new update algorithms with the feed-through neural network will make a very good control strategy when the nonlinear system has not been modeled accurately. It is very difficult to accurately model the nonlinearities of a plant. This control strategy can provide an improvement in performance for both stable and unstable nonlinear plants.