A review of the current literature on sensor noise elimination algorithms, reveals the existence of several adaptive noise reduction strategies. The well known Wiener filter, (Widrow and Stearns 1985) which is an optimum one, belongs to this category and removes the noise in the mean square error sense. The architecture of this adaptive filter is shown in Figure 6.1. In this, the noise \( n_1(k) \) is estimated as \( \hat{n}_1(k) \) using a secondary noise source \( n_2(k) \) and is then subtracted from \( s(k) \) to get an estimate of the desired signal. This scheme is suitable provided,

1. \( x(k), n_1(k) \) and \( n_2(k) \) are stationary,
2. \( n_1(k) \) and \( n_2(k) \) have known statistics and
3. \( n_2(k) \) is a noise signal that is correlated with \( n_1(k) \) (Hayes, 1996)

\[
s(k) = x(k) + n_1(k) + \hat{n}_1(k)
\]

**Figure 6.1** Adaptive noise cancellation network for stationary signals. \( x(k) \) is the true value of the process variable, \( n_1(k) \) is the sensor noise, \( n_2(k) \) is the secondary noise source and \( s(k) \) is the measured noisy value.
Extended adaptive filter techniques have also been reported for situations where $x(k)$, $n_1(k)$ and $n_2(k)$ can be non stationary and no apriori statistical information about them is available, except for the consideration that $n_2(k)$ and $x(k)$ are uncorrelated. This architecture is shown in Figure 6.2. These techniques, however, need a secondary noise source $n_2(k)$ to estimate $n_1(k)$. The desired signal $x(k)$ is extracted as before by subtracting the estimate from the observed signal $s(k)$.

A final class of noise elimination strategies, that are ideally suited for sensor noise cancellation in a real time environment favors a noise cancellation model in which the secondary noise source $n_2(k)$ is absent, while retaining the constraints that

(i) $x(k)$ and $n_1(k)$ can be nonstationary and
(ii) no apriori statistical information is required about $x(k)$ and $n_1(k)$.

\[ s(k) = x(k) + n_1(k) \]

\[ \hat{x}(k) = s(k) - \hat{n}_1(k) \]

Figure 6.2 Adaptive noise cancellation network to include non-stationary signals. $x(k)$ is the true value of the process variable, $n_1(k)$ is the immeasurable sensor noise and $n_2(k)$ is the secondary noise source correlated with $n_1(k)$.
The sensor noise removal algorithm to be presented in this chapter and to be used along with the intelligent control algorithms discussed previously, belong to this class of noise elimination technique. This network architecture is shown in Figure 6.3. The delay path serves to generate a reference signal, that can be used to estimate \( x(k) \). It has the advantage, that, it can remove the noise by observing the mixed signal alone. No apriori statistical information about the desired signal is required to be given as input to the network. Also, the source signal \( x(k) \) need not be stationary. This motivates the use of blind signal separation (BSS) algorithms to remove the sensor noise and extract the true value of the process from the noisy sensor readings. A neural network based independent component analysis (ICA) algorithm is incorporated.

Figure 6.3 Adaptive noise cancellation network to include non-stationary \( x(k) \), with no secondary noise source. (i.e. \( n_2(k) \) is not required)

- \( x(k) \) is the true value of the process variable
- \( n(k) \) is the immeasurable sensor noise

\[ \text{over sampled mixed signal} = x(k) + n(k) \]
into the existing system to remove the sensor noise. The proposed intelligent controller, (discussed previously) along with this ICA algorithm for sensor noise elimination, can be used to handle exceptional conditions, such as

(i) poor performance of controller due to faulty sensor readings and
(ii) reduced life of actuators due to frequent false actuating signals to the control valve. The second problem arises, since at steady state \( x(k) \) is stationary but the conventional control algorithms have access only to the noisy sensor readings \( s(k) \).

By using the sensor noise elimination algorithm proposed (ICA network based) at the input of each of the separately located low cost sensors, the noise rejection capability of the sensor is improved at no extra hardware costs.

6.1 **INDEPENDENT COMPONENT ANALYSIS**

The independent component analysis consists of three steps, as shown in Figure 6.4

(i) Whitening,
(ii) Separation and
(iii) Estimation
If the observed data vectors $x_k$ have a nonzero mean, it is desirable to remove it before further processing. This process is called whitening. The effects of second-order statistics can be removed by using the whitening transformation (Cardoso and Laheld, 1997). The whitening matrix is chosen so that the covariance matrix of the whitened vectors $\{v_k v_k^T\}$ equals the unit matrix $I_M$. After whitening, the components of the whitened vectors $v_k$ are mutually uncorrelated with zero mean and unit variance.

Uncorrelatedness is a necessary prerequisite for the stronger independence condition, so that after pre-whitening, the noise removal task from the observed mixed data usually becomes easier. There exist many methods for whitening the input data, provided that the number of observations of the noisy sensor readings exceed or at least equal the number of source signals.
6.1.1 Principal component analysis (PCA) based whitening algorithm

In this work, the standard PCA (Jutten and Cardoso et al. 1997 and Karhunen et al. 1997) based approach is used for whitening. This method has the advantage that it can simultaneously compress information optimally in the mean-square error sense and filter possible noise.

The following steps are used in the PCA based pre-whitening process:

1) Remove the mean value from the noisy sensor readings \( s[k] \)
2) Form the covariance matrix of the observed signal \( E\{s_k s_k^T\} \).
3) Extract the matrices \( D \) and \( E \),
   where \( D = \text{diag}[\lambda(1), \ldots, \lambda(M)] \) is a \( M \times M \) diagonal matrix,
   \( E = [c(1), \ldots, c(M)] \) is a \( L \times M \) matrix,
   \( \lambda(i) \) is the \( i_{th} \) largest eigenvalue of the data covariance matrix \( E\{s_k s_k^T\} \) and \( c(i) \) is the respective \( i_{th} \) principal eigenvector.
4) Form whitening matrix \( V \) using \( V = D^{1/2} \times E^T \)
5) Pre-whiten the inputs using the transformation \( v[k] = V^* s[k] \)

6.1.2 Separation process

The second stage in the ICA network is the separation of the source signals. This can be achieved by using suitable higher-order statistics. The separation process can be modeled as a single-layer neural network with an equal number of input and output nodes, where the coefficient \( w_{ij} \) of the separation matrix \( W \), are simply the weights from the input to output nodes. In
the present work, the noise is removed from the measured sensor readings at this stage.

Several types of separation algorithms have been in use, such as the HJ algorithm (Jutten et al 1991), PFS algorithm (Cardoso et al 1996), Bell’s and Sejnowski’s algorithm (Bell and Sejnowski 1995), bigradient algorithm (Wang and Karhunen 1995), nonlinear principal-component-analysis (PCA) subspace learning rule (Oja et al 1991) etc. The nonlinear PCA sub-space learning algorithm is used in the present work, as it has the advantage, that it can be realized using a simple modification of the one-layer standard symmetric PCA network (Karhunen et al 1995).

6.1.2.1 Neural network based separation algorithm

The neural network based separation algorithm is implemented by writing a ‘c’ language program. The pseudo-code of the algorithm is given below.

Step 1

while(count < number of training epochs)
{
    while(i < number of observed samples)
    {
        assign y[k] = W^T * v[k]
        assign \( \mu = 1/(\gamma / \mu[k-1] + |v[k]|^2) \)
        assign W[k+1] = W[k] + \mu[k] \{ v[k] - W[k] * g(y[k]) \} g(y^T[k])
Step 2

Estimate the source signals using $y[k] = W^T v[k] \sqrt{v/\text{(number of observed samples)}}$

6.1.3 Choice of activation function ‘g(.)’ used in the separation process

A proper choice of the activation function $g(.)$ is important for effective removal of noise present in the sensor readings. The activation function at the output nodes is used for the training mode only and plays a central role in blind signal separation (BSS). Its nature is defined by objective or contrast or score function. The maximum likelihood approach, (Yang, 1999) defines the score function as,

$$g_i(u_i) = -\frac{\partial \log p_i(u_i)}{\partial u_i} = \frac{-p'_i(u_i)}{p_i(u_i)}, \quad i = 1, \ldots, M_s$$
where $p_s(u)$ and $p'_s(u)$ are the probability density function (pdf) and its derivative, respectively, of the source signals. In this work, a sigmoidal activation function of the form $\tanh(.)$ is chosen.

6.2 MULTIPLE SENSOR COORDINATION MODEL (WITH SENSOR NOISE ELIMINATION SCHEME AND FEEDBACK SENSOR VALIDATION NETWORK)

The nonlinear plant with the noise added to the inflow, outflow and level measuring sensors is shown in Figure 6.5. The multiple-sensor coordination model proposed for use in an interacting multivariable nonlinear system, to accommodate single sensor fault and noise in the measured states that may occur independently or simultaneously, is shown in Figures 6.6a and 6.6b. The three independent ICA networks in Figure 6.6a remove the noise present in the inflow, outflow and level sensors, while the neural network or fuzzy trained estimator (discussed in chapter 5) in Figure 6.6b takes care of the feedback sensor failure.
Figure 6.5  Plant model with sensor noise added to inflow, outflow and level sensors

Unknown Transfer function
Figure 6.6a  Multiple-sensor coordination model with the ICA networks for sensor noise elimination in a multivariable interacting system. $G_c$ is the controller and FT is the flow transmitter.
Figure 6.6b Multiple-sensor coordination model with the ICA networks for sensor noise elimination and the neural or fuzzy trained estimator for feedback sensor validation in a multivariable interacting system. \( G_c \) is the controller and \( FT \) is the flow transmitter.
6.3 RESULTS

The focus of this section is to demonstrate the capability of the designed noise elimination algorithm to remove the sensor noise and extract the true value of the process, independent of the ensemble statistics of a signal. The distribution of the noise model is important since it is a representative of the physical source from where the noise originated (Middleton (1977), Garth and Poor (1994) and Adler et al (1998)). In the present work, the sensor noise was created during each of the trial, by filtering noise samples initially drawn from a Gaussian distribution. Though the proposed algorithm is designed to remove noise without regard to a particular distribution, the noise was assumed to have evolved from a Gaussian distribution, since it is sensible in terms of the central limit theorem (Papoulis 1991 and Kadota 1988).

Samples of the nonstationary signals inflow, outflow and level of sufficiently long duration was captured for study and many trials were performed. The samples were quantized with an 8-bit resolution. The noisy sensor readings were oversampled and presented to the noise elimination network of Figure 6.3. It was assumed throughout the period of study, that, the autocorrelation sequence of the signal can decay to zero at a slower rate than that of the noise. This assumption is reasonable and is typical of the one that would occur in a real time process control plant. The whitening and separation algorithms (section 6.1) were used and an estimate of the true value was obtained. The mean square error (MSE) criterion was used as a measure of performance index in each case. The normalized MSE was calculated as
\begin{align*}
\text{MSE} = \frac{1}{N} \sum_{k=1}^{N} [\text{true value of the measured state} - \text{estimated value}]^2
\end{align*}

where N is the number of samples and is chosen as 400.

6.3.1 Eliminating level sensor noise (during set point tracking)

The true variations of the process variable level, sensor noise, observed noisy sensor readings and the ICA network output for this case are shown in Figures 6.7 to 6.10 respectively. The combined plot is shown in Figure 6.11 for comparison.

![Figure 6.7 Magnified view of the true variations of process level in the experimental set point tracking response with FLC (i.e. level variations with sensor noise =0)](image)

Figure 6.7 Magnified view of the true variations of process level in the experimental set point tracking response with FLC (i.e. level variations with sensor noise =0)
Figure 6.8 Sensor noise added to level

Figure 6.9 Magnified view of the variations of the process level in the presence of sensor noise in the experimental set point tracking response with fuzzy controller
Figure 6.10 Magnified view of the estimated variations of process level in the experimental set point tracking response with fuzzy controller
6.3.2 Eliminating level sensor noise (during load variations)

The true variations of the process variable level, the observed noisy sensor values and the estimated level variations for this case are shown in Figures 6.12 to 6.15 respectively. The combined plot is shown in Figure 6.16 for comparison.
Figure 6.12  Magnified view of the true variations of process level in the experimental regulatory response (response in the presence of perturbations in load variable outflow and with sensor noise = 0)
Figure 6.13 Sensor noise added to level

Figure 6.14 Magnified view of the variations of process level in the experimental regulatory response in the presence of sensor noise
Figure 6.15  Magnified view of the estimated variations of the process level in the experimental regulatory response

Figure 6.16  Magnified view of the combined plot
6.3.3 Eliminating inflow sensor noise

The true value of the inflow rate, added sensor noise, the observed noisy sensor readings and the estimated inflow variations for this case are shown in Figures 6.17 to 6.20. The combined plot of the true and estimated inflow is shown in Figure 6.21.

Figure 6.17 Magnified view of the variations in the true value of inflow (in %) (measured with sensor noise = 0)
Figure 6.18 Sensor noise added to inflow

Figure 6.19 Magnified view of the variations in inflow (in %) measured in the presence of sensor noise (true value mixed with sensor noise)
Figure 6.20 Magnified view of the variations in inflow (in %) estimated by the ICA network

Figure 6.21 Combined plot of true fluctuations in inflow, noisy sensor values and the estimated inflow (in %) about its nominal value of 50%
6.3.4 Eliminating outflow sensor noise

The true value of the load variable outflow, added sensor noise, the observed noisy sensor readings and the estimated outflow variations for this case are shown in Figures 6.22 to 6.25. The combined plot of the true and estimated outflow is shown in Figure 6.26.

![Figure 6.22 Magnified view of the variations in the true value of outflow (in %) (measured with sensor noise = 0)](image)

![Figure 6.23 Sensor noise added to outflow](image)
Figure 6.24 Magnified view of the variations in outflow (in %) measured in the presence of noise (true value mixed with sensor noise)

Figure 6.25 Magnified view of the variations in outflow (in %) estimated by the ICA network
6.4 PERFORMANCE OF THE SENSOR NOISE ELIMINATION NETWORK

The proposed sensor noise elimination scheme was implemented and tested for satisfactory performance on the real-time plant. The true sensor readings were corrupted with Gaussian noise and presented to the sensor noise elimination network. The estimated value is compared with the true value by computing the mean square error deviation (MSE). From the obtained MSE value, it can be concluded that the estimated values match well with the true values of the measured states. The ability of the network to process the noisy
level sensor readings and give a good estimate of the true value during the set point tracking response is significant and worth mentioning.

The proposed sensor noise elimination algorithm can be incorporated with ease into any of the existing process control loops at no extra hardware costs. This permits the use of a commercially available low cost sensor, with the sensor noise elimination taken care by the proposed algorithm. The algorithm can be implemented easily using the unsupervised neural learning architecture. The algorithms are designed to remove the sensor noise in a manner ideally suited for a real-time environment, where the sensor noise elimination process need to be performed independent of the ensemble statistics of the signal without regard to a particular class of signal model. The MSE values obtained for the different cases and the error plots are presented in chapter 7.

However, it may be mentioned here, that, in the area of blind signal separation, the theoretical properties, range of applicability and mutual comparisons, still remain mostly unexplored.