CHAPTER 7

DEVELOPMENT OF KALMAN ESTIMATOR AND SOFT SENSORS TECHNIQUE FOR CHEMICAL PROCESS

7.1 INTRODUCTION

The aim of this chapter is to find the estimation errors and observation errors for an interacting system, consisting of liquid level of the tank and liquid temperature which needs to be maintained at a desired level. Previous chapters presented three different decoupling and linearization algorithms. After the linearization of Level-Temperature MIMO process, estimation is carried out using the Kalman filter. Also soft sensor techniques are implemented for an approximated model of an interacting thermal non-linear process. The soft sensor techniques such as the Luenberger observer, Kalman observer and the Unknown Input Observer (UIO) are designed to estimate the system parameters namely level of the tank and temperature of the tank for a non interacting linear MIMO system. From the study, it has been found that performance of UIO is better. The estimated error for the plant using UIO observer is less when compared to Luenberger and Kalman observer.

7.2 DEVELOPMENT OF KALMAN ESTIMATOR

After applying Decoupling and Linearization (Hirschorn’s algorithm), Kalman estimator is applied to the Level-Temperature cascaded process and the performance is analyzed. The Kalman filter estimates the state
of a dynamic system, if the precise form of the system is unknown. The Kalman Filter estimates a process by using a feedback (Gunter Krapf et al 2011). The operation can be described as the process estimated by the filter at some point of time and the feedback is obtained in the form of noisy measurements. Figure 7.1 shows the Kalman Filter Prediction Estimation Cycle.

![Figure 7.1 Kalman filter prediction estimation cycle](image)

**Figure 7.1 Kalman filter prediction estimation cycle**

The Kalman filter equations can be divided into two categories: time update equations and measurement update equations. To obtain the apriori estimates for the next time step, the time update equations project forward (in time) the current state and error covariance estimates. The measurement update equations get the feedback to obtain an improved aposteriori estimate incorporating a new measurement into the a priori estimate.
The Kalman filter has two steps, such as

- Prediction
- Correction

In the first step, the system state is predicted with the aid of the dynamic model of the system. In the second step, it is corrected with the observation model, so that the error covariance of the estimator is minimized.

Estimation technique is implemented for the linear state equation of a liquid level temperature process to measure the level and temperature difference between real output and desired output. Estimated output of a temperature and level parameter follows the real process output. Applying Kalman estimation is easy to design a controller for future prediction due to any deviation of the estimated output from true output (Gerasimos Rigatos 2009). Block diagram representation of Kalman Filter is shown in Figure 7.2. The Kalman filter is a recursive predictive filter that is based on the state space techniques and recursive algorithms, i.e., only the state from the previous time step and the current measurement are needed to compute the estimate of the future state. The Kalman filter operates by propagating the mean and covariance of the state through time. The notation $\hat{x}_{nm|m}$ represents the estimate of the state vector ‘x’ at time ‘n’ for observations up to, and including time ‘m’ (Gibon et al 2000).
The state of the filter is represented by two variables:

- $\hat{x}_{k|k}$, the a posteriori state estimate at time ‘k’ for observations up to and including time ‘k’

- $P_{k|k}$, the a posteriori error covariance matrix (a measure of the estimated accuracy of the state estimate)

The prediction phase uses the state estimate from the previous time step to produce an estimate of the state at the current time step. This predicted state estimate is also known as the a priori state estimate because, although it is an estimate of the state at the current time step ‘k’. In the correction phase, the current a priori prediction is combined with current observation information to refine the state estimate. This improved estimate is termed as a posteriori state estimate. Consider a linear time invariant discrete system given by the following equation

$$X_k = F x_{k-1} + B u_k$$  \hspace{1cm} (7.1)

$$Z_k = H x_k$$  \hspace{1cm} (7.2)
where,

‘F’ is the state transition matrix. ‘B’ is the control input matrix. ‘H’ is the observation matrix. u_k is the control input. ‘X’ is the state of the system. ‘Z’ is the measured output

a) Prediction (Time update) Equations

Predicted state estimate

\[ \hat{x}_{k|k-1} = F \hat{x}_{k-1|k-1} + Bu_k \]  \hspace{1cm} (7.3)

Predicted estimate covariance

\[ P_{k|k-1} = FP_{k-1|k-1}F^T + Q_k \]  \hspace{1cm} (7.4)

b) Correction (Measurement update) Equations

Innovation or measurement residual

\[ \tilde{y}_k = z_k - H \hat{x}_{k|k-1} \]  \hspace{1cm} (7.5)

Innovation (or residual) covariance

\[ S_k = HP_{k|k-1}H^T + R_k \]  \hspace{1cm} (7.6)

where \( Q_k \) and \( R_k \) are Covariance matrix.
Optimal Kalman gain

\[ K_k = P_{k|k-1} H^T S_k^{-1} \]  \hspace{1cm} (7.7)

Updated (a posteriori) state estimate

\[ \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \]  \hspace{1cm} (7.8)

Updated (a posteriori) estimate covariance

\[ P_{k|k} = (I - K_k H) P_{k|k-1} \]  \hspace{1cm} (7.9)

Thus, the Kalman estimation is carried out for a MIMO process. If the measurement noise is large, ‘k’ will have a small value and will not give much weight to the measurement in the estimation of the state. If the measurement noise is small, ‘k’ will be large and the measurement will give a large weight when estimating the state. The results that are obtained by the Kalman estimator technique are shown in Figures 7.7 and 7.8.

7.3 DEVELOPMENT OF SOFT SENSORS

Most of the state feedback control scheme assumes full order state feedback. However, this is not the case in many industrial processes. So, the presence of unknown state becomes a serious drawback when implementing a state feedback control law. Such difficulty can be overcome through the design of an appropriate observer to estimate the missing states of the system from the knowledge of its input/output.
In control theory, a state observer is a system that models a real system to provide an estimate of its internal state, given measurements of the input and output of the real system without the use of actual sensors. It is typically a computer-implemented mathematical model (Abderraouf and Faouzi 2009). Knowing the system state is necessary to solve many control theory problems. In most practical cases, the physical state of the system cannot be determined by direct observation. Instead, indirect effects of the internal state are observed by way of the system outputs. If a system is observable, it is possible to fully reconstruct the system state from its output measurements using the state observer (Stefen Hui and Stanislaw Zak 2005).

Observers use the plant input and output signals to generate an estimate of a particular plant’s state which is then employed to close the control loop. Observers are used to augment or replace sensors in a control system. Chemical reactors are examples of industrial processes that are concerned with measurement problems: some concentrations may be not easily measurable (especially for intermediate components). In this case, estimation theory via observer synthesis will cope with this difficulty. Soft sensors are algorithms based on the dynamic model of the process described by differential equations and the knowledge of the control variables, the parameters of the model and some measured dynamical variables.

The principle of an observer is that by combining a measured feedback signal with knowledge of the control system components (primarily the plant and feedback system), the behavior of the plant can be known with greater precision than by using the feedback signal alone. As shown in Figure 7.3, the observer augments the sensor output and provides a feedback signal to the control laws.
In some cases, the observer can be used to enhance system performance. It can be more accurate than sensors or can reduce the phase lag inherent in the sensor. Observers can also provide observed disturbance signals, which can be used to improve disturbance rejection. In other cases, observers can reduce system cost by augmenting the performance of a low cost sensor so that the two can provide performance equivalent of a higher cost sensor together. In the extreme case, observers can eliminate a sensor altogether, reducing sensor cost and the associated wiring. State observers can be implemented to linear and non linear systems and are respectively called as linear and non linear observers. Here, only linear observers have been discussed (Elom et al 2011).

The linear observers which are available to estimate the state of the systems are Luenberger observer, Kalman observer, Unknown Input observer, Augmented Robust observer, High gain observer and more, of which only three observers have been discussed.
The observers which are considered in detail are

1. Luenberger observer
2. Kalman observer
3. Unknown Input observer

7.3.1 Luenberger Observer

In early days, this observer has been developed to implement in a deterministic continuous time invariant systems but later it has been extended by many researchers to implement in discrete, time invariant systems. The relative simple design of this observer makes it an attractive general design technique. For discrete time linear system described by Equations (7.10) and (7.11) the Luenberger observer has been implemented. It is assumed that the system has been influenced by additive faults on system states, \( f_i(k) \) and also on the output, \( f_m(k) \).

\[
x(k+1) = Ax(k) + Bu(k) + Lf_i(k) \quad (7.10)
\]

\[
y(k) = Cx(k) + Du(k) + Mf_m(k) \quad (7.11)
\]

where,

\( x(k) \) - Plant states

\( u(k) \) - Control input

\( y(k) \) - Output of the system

\( L, M \) - Constant Matrices

For such linear system, the state equation and output equation of the observer are given by,
\[
\hat{x}(k+1) = AX(k) + Bu(k) + Hr(k) \\
\hat{y}(k) = C\hat{x}(k) \\
r(k) = y(k) - \hat{y}(k)
\]

where

\( r(k) \) is the error in the output.

H- Observer gain matrix

Hence, the state equation of the observer is modified as

\[
\hat{x}(k+1) = (A - HC)\hat{x}(k) + Bu(k) + Hy(k)
\]

Figure 7.4 shows the block diagram of Luenberger observer
The observer gain is found using pole placement technique. ‘K’ in the block diagram is the state feedback gain matrix which is used in determining the observer gain matrix. The results that are obtained by using the Luenberger Observer technique are shown in Figures 7.9 and 7.10.

7.3.2 Implementation of Kalman Observer

Kalman Observer is a recursive predictive filter based on the use of state space techniques and recursive algorithms, i.e. only the estimated state from the previous time step and the current measurement are needed to compute the estimate of the current state. The Kalman filter operates by propagating the mean and covariance of the state through time. The notation $\hat{x}_{n|m}$ represents the estimate of the state vector ‘X’ at time n given observations till m.

The state of the filter is represented by two variables

- $\hat{x}_{k|k}$, a posteriori state estimate at time ‘k’. The given observation is up to and including at time ‘k’.

- $P_{k|k}$, a posteriori error covariance matrix which measure the estimated accuracy of the state.

The Kalman filter has two distinct phases, prediction and correction. The prediction phase uses the state estimate from the previous time step to produce an estimate of the state at the current time step. This predicted state estimate is also known as the apriori state estimate because, although it is an estimate of the state at the current time step, it does not include observation information from the current time step. In the correction
phase, the current apriori prediction is combined with current observation information to refine the state estimate. This improved estimate is termed the a posteriori state estimate. The block diagram in Figure 7.5 represents the state estimator.

\[
x(k|k-1) = Fx(k-1|k-1) + Bu(k)
\]

\[
x(k|k) = x(k|k-1) + K(k)e(k)
\]

\[
y(k|k-1) = y(k-1|k-1)
\]

\[
y(k|k) = y(k|k-1) + Hx(k|k)
\]

**Figure 7.5 Block diagram of Kalman observer**

Typically, the two phases alternate, with the prediction advancing the state until the next scheduled observation, and the correction incorporating the observation. However, this is not necessary, if an observation is unavailable for some reason, the update may be skipped and multiple prediction steps can be performed. Consider a linear time invariant discrete system given by the following equations,

\[
X_k = FX_{k-1} + Bu_k + W_k
\]  \hspace{1cm} (7.16)

\[
Z_k = HX_k + V_k
\]  \hspace{1cm} (7.17)
where,

- $F$ is the state transition matrix,
- $B$ is the control input matrix,
- $W_k$ is the process noise with zero mean multivariate normal distribution having covariance $Q_k$.
- $H$ is the observation matrix,
- $V_k$ is the observation noise which is zero mean Gaussian white noise having covariance $R_k$.
- $U_k$ is the control input.

**a) Prediction (Time update) Equations**

Predicted state estimate

$$\hat{X}_{k|k-1} = F \hat{X}_{k-1|k-1} + B u_k$$ (7.18)

Predicted estimate covariance

$$P_{k|k-1} = FP_{k-1|k-1} F^T + Q_k$$ (7.19)

**b) Correction (Measurement update) Equations**

Innovation or measurement residual

$$\hat{y}_k = Z_k - H \hat{X}_{k|k-1}$$ (7.20)
Innovation (or residual) covariance

\[ S_k = H P_{k|k-1} H^T + R_k \]  (7.21)

Optimal Kalman gain

\[ K_k = P_{k|k-1} H^T S_k^{-1} \]  (7.22)

Updated (a posteriori) state estimate

\[ \hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k \hat{y}_k \]  (7.23)

Updated (a posteriori) estimate covariance

\[ P_{k|k} = (I - K_k H) P_{k|k-1} \]  (7.24)

The results that are obtained by the Kalman observer technique are shown in Figures 7.10 and 7.11.

**7.3.3 Implementation of Unknown Input Observer**

UIO is an estimator that is decoupled from the unknown input (disturbance, faults) that acts on the system. UI observer is capable of estimating the states with unknown inputs. The unknown inputs may be a combination of unmeasurable or unmeasured disturbances, unknown control action or unmodeled system dynamics. This observer is very useful when dealing with problem of instrument fault detection. It can be implemented as reduced order observer or full order observer (Krzeminski and Kaczone 2004).
Consider a continuous linear time invariant state space model of the system (Pau Lo Hsu et al 2001)

\[ \dot{x}(t) = Ax(t) + Bu(t) + Ed(t) \quad (7.25) \]

\[ y(t) = Cx(t) \quad (7.26) \]

\( x \in \mathbb{R}^{nx1} \) represents the state vector, \( u \) represents input vector, \( y \) represents sensor output, \( A \) represents system coefficient matrix, \( B \) represents input coefficient matrix, \( C \) represents output coefficient matrix, \( d \in \mathbb{R}^{q*1} \) represents the unknown input vector, and \( E \in \mathbb{R}^{nxq} \) represents the unknown input distribution matrix. \( Ed(t) \) can also be used to represent uncertainties, which may incorporate unknown nonlinearity and uncertain coefficients.

The structure of the UIO is described as (Yan et al 2004),

\[ z(k+1) = Rz(k) + Tu(k) + Ky(k) \quad (7.27) \]

\[ \hat{x}(k) = z(k) + Hy(k) \quad (7.28) \]

\[ \hat{y}(k) = C\hat{x}(k) \quad (7.29) \]

\( \hat{x} \in \mathbb{R}^{nx1} \) represents the estimated state vector and \( T \in \mathbb{R}^{nxn} \), \( K \in \mathbb{R}^{nxn} \) and \( H \in \mathbb{R}^{nxn} \) are matrices satisfying requirements. The full order observer is illustrated in Figure 7.6.
Figure 7.6 Structure of a full-order UIO

The error vector is given by

\[ e(k) = \hat{x}(k) - x(k) \]  \hspace{1cm} (7.30)

Error vector is obtained as

\[ e(k) = x(k) - \hat{x}(k) = x(k) - z(k) - Hy(k) \]  \hspace{1cm} (7.31)

\[ = x(k) - z(k) - HCx(k) \]  \hspace{1cm} (7.32)

\[ = (I - HC)x(k) - z(k) \]  \hspace{1cm} (7.33)

The following relations hold true

\[ (HC - I)E = 0 \]  \hspace{1cm} (7.34)

\[ T = (I - HC) \]  \hspace{1cm} (7.35)
\[ F = A - HCA - K_1CK_2 \]  \hspace{1cm} (7.36)

\[ K_2 = RH \]  \hspace{1cm} (7.37)

\[ K = K_1 + K_2 \]  \hspace{1cm} (7.38)

\( F, \ H, \ K_1, \ K_2 \) are the coefficients matrices with appropriate dimensions. The desired observer response can be achieved by assigning suitable poles through the design of \( K_1 \) and \( K_2 \). The results that are obtained by the UIO observer technique is shown Figure 7.13 and 7.14.

### 7.4 SIMULATION RESULTS AND DISCUSSION

The following simulation results prove that the desired objectives of achieving decoupling and linearization were accomplished. Simulation results obtained after applying Hirschorn’s algorithm for Level-Temperature process with Kalman estimator are shown in Figure 7.7. Estimated output of the temperature process follows the real process output.

![Figure 7.7 Estimated error using Kalman estimator for temperature parameter](image)

**Figure 7.7 Estimated error using Kalman estimator for temperature parameter**

Results of Decoupling and Linearization (Hirschorn’s algorithm) with Kalman estimator for level process are shown in Figure 7.8. Estimated output of the level process is same as the measured process output (level).
Figure 7.8 Estimated error using Kalman estimator for level parameter

Figure 7.9 illustrates the level tracking error between plant output and Luenberger observer output. Luenberger observer is designed for the state space model of a level process. Actual level output $y$ and observer level output $\hat{y}$ are obtained directly from simulation model of the plant and state estimation error is calculated. The level error varies between -0.075 to +0.075 cm when the set point of the level is raised from 1 to 60 cm. So 0.25% error occurs as mentioned in Table 7.1.

Figure 7.9 Estimated error using Luenberger observer for level parameter
Figure 7.10 illustrates the temperature tracking error between plant output and Luenberger observer output. Luenberger observer is designed to track the temperature of the process represented by the state space model. Actual temperature output $y$ and observer temperature output $\hat{y}$ are obtained from simulation model of the plant and state estimation error is calculated. The temperature error varied between $-0.05$ to $+0.05^\circ C$ when the setpoint of the temperature is raised from 1 to 30$^\circ C$. So 0.33% error occurs as mentioned in Table 7.1.

![Figure 7.10 Estimated error using Luenberger observer for temperature parameter](image)

Figure 7.11 illustrates the level tracking error between plant output and Kalman observer output. Kalman observer is designed for the state space model of a level process. Actual level output $y$ and observer level output $\hat{y}$ are obtained from simulation model of the plant and state estimation error is calculated. The level error varied between $-2$ to $+2$ cm when the setpoint of the level is raised from 1 to 60 cm. In Luenberger observer estimation error is improved as 0.25% than 0.66% achieved by Kalman observer as mentioned in Table 7.1.
Figure 7.11 Estimated error using Kalman observer for level parameter

Figure 7.12 illustrates the temperature tracking errors for plant output and Kalman observer output. Kalman observer is designed for the state space model of a temperature process. The state estimation error is calculated from the actual output $y$ and observer temperature output $\hat{y}$. The temperature error varied between -2 to +2°C when the setpoint of the temperature is raised from 1 to 30°C. In Luenberger observer, estimation error is improved as 0.33% than 13.3% achieved by Kalman observer as mentioned in Table 7.1.

Figure 7.12 Estimated error using Kalman observer for temperature parameter
Figure 7.13 illustrates the level tracking errors for plant output and Unknown Input observer output. Unknown Input observer is designed for the state space model of a level process. Actual level output of \( y \) and observer level output \( \hat{y} \) are obtained directly from the simulation model of the plant and state estimation error is calculated. The level error varied between +0.02 to -0.02 cm when the setpoint of the level is raised from 1 to 60 cm. In UI observer, estimation error is improved as 0.06% than 0.25% achieved by Luenberger observer as mentioned in Table 7.1.

Figure 7.13 Estimated error using UI observer for level parameter

Figure 7.14 illustrates the temperature tracking errors for plant output and Unknown Input observer output. Unknown Input observer is designed to estimate the temperature parameter of the system. Actual temperature output of \( y \) and observer temperature output \( \hat{y} \) are obtained directly from simulation model of the plant and the state estimation error is calculated. The temperature error varied between +0.02 to -0.02ºC when the setpoint of the temperature is raised from 1 to 30ºC. In UI observer estimation error is improved as 0.13% than 0.33% achieved by Luenberger observer as mentioned in Table 7.1.
Figure 7.14 Estimated error using UI observer for temperature parameter

Table 7.1 shows the performance of three soft sensors in Level-Temperature process. It can be inferred that UIO observer gives less error for both level and temperature parameter

Table 7.1 Comparative performance of observers

<table>
<thead>
<tr>
<th>Observer type</th>
<th>Level parameter error</th>
<th>Temp parameter error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luenberger</td>
<td>0.25%</td>
<td>0.33%</td>
</tr>
<tr>
<td>Kalman</td>
<td>6.66%</td>
<td>13.3%</td>
</tr>
<tr>
<td>UIO</td>
<td>0.06%</td>
<td>0.13%</td>
</tr>
</tbody>
</table>

7.5 SUMMARY

In this chapter, the detailed explanation of Kalman estimator and three observers (soft sensors) are presented. Implementation of Kalman estimator and observers are carried out for the liquid Level-Temperature process. By using Hirschorn’s algorithm, the non linear interacting MIMO process (level and temperature) is decoupled and linearized. After getting
linear state equations, output parameters are estimated using Kalman estimation.

Soft sensor observers like Luenberger observer, Kalman observer, and Unknown Input Observer (UIO) are then applied for the same linear state model. From the simulation results, it is observed that the performance of a UIO gives less error between the real and observed output.