3.5 Frequency-domain Identification Method

■ Frequency identification of transfer functions

The frequency characteristic is a non-parametric model describing a dynamic system, which can be measured by experimental methods.

This section discusses the method of finding the system transfer function when the frequency characteristics have been measured.

When the controlled plant is described by frequency characteristics, the general expression is

$$G(j\omega) = \left.\frac{Y(s)}{U(s)}\right|_{s=j\omega} = \frac{Y(j\omega)}{U(j\omega)}$$

where $Y(s)$ is the Laplace transformation of the output of the identified plant, and $U(s)$ is the Laplace transformation of the input.

♦ **Calculating a transfer function using Bode plot properties**

If the frequency response data of the system are measured experimentally, the logarithmic frequency characteristic curve can be made according to the frequency characteristic, so as to obtain the transfer function.

A minimum phase system can usually be described by the following:

$$G(s) = \frac{K\Pi_{i=1}^{p}(T_{1i}s+1)\Pi_{i=1}^{q}\left(T_{2i}^{2}s^{2}+2T_{2i}\xi_{1i}s+1\right)}{s^{n}\Pi_{i=1}^{r}(T_{3i}s+1)\Pi_{i=1}^{l}\left(T_{4i}^{2}s^{2}+2T_{4i}\xi_{2i}s+1\right)}$$

where T_{1i} and T_{3i} are the time constants of the *i*-th first-order differential unit and inertial unit, ξ_{1i} and ξ_{2i} are the damping ratios of the *i*-th second-order differential unit and oscillation unit, T_{2i} and T_{4i} are the time constants of the *i*-th second-order differential unit and oscillation unit.

After measuring the frequency response of the system through experiments, Table 1 can be used to obtain the corresponding basic characteristics, thereby obtaining the transfer function.

The specific method is to use some straight lines with slopes of 0, $\pm 20\mathrm{dB/dec}, \pm 40\mathrm{dB/dec}, \ldots$ to approximate the magnitude-frequency characteristics, and try to find frequency inflection points, then the transfer function of the system can be obtained.

Table 1 Asymptotic characteristics of the frequency response of the basic units

Unit	$\omega \ll \frac{1}{T}$		$\omega = \frac{1}{T}$		$\omega \gg \frac{1}{T}$	
	M	P	M	P	M	P
K	$20\log K$	$0°$	$20\log K$	$0°$	$20\log K$	$0°$
s^n	$n\times 20\mathrm{dB}$	$n\times 90°$	$n\times 20\mathrm{dB}$	$n\times 90°$	$n\times 20\mathrm{dB}$	$n\times 90°$
$Ts+1$	0dB	$0°$	3dB	$45°$	20dB	$90°$
$\frac{1}{Ts+1}$	0dB	$0°$	−3dB	$-45°$	−20dB	$-90°$
$T^2s^2+2\xi Ts+1$	0dB	$0°$	因ξ而异	$90°$	40dB	$180°$
$\frac{1}{T^2s^2+2\xi Ts+1}$	0dB	$0°$	因ξ而异	$-90°$	−40dB	$-180°$
$e^{-\tau s}$	0dB	$-\frac{180°}{\pi}T\omega$	0dB	$-\frac{180°}{\pi}T\omega$	0dB	$-\frac{180°}{\pi}T\omega$

3

Taking the third line of Table 1 as an example, if the magnitude and phase at low frequency are 0dB and 0 degrees, respectively, and the magnitude and phase at high frequency are 20dB and 90 degrees, respectively, and when the phase is 45 degrees, the magnitude is 3dB, it means that the basic unit is Ts+1, and T can be obtained by

$$\omega = 1/T$$

For a minimum phase system, the transfer function $G(s)$ of a system can be determined by the magnitude frequency response of the system.

If the obtained phase angle of G(s) does not match the experimental results, and the difference between the two is a constant angular frequency change rate, it means that the obtained transfer function is G(s) and the controlled plant contains the delay unit.

If the transfer function of the controlled plant is $G(s)\mathrm{e}^{-\tau}$, then for high frequency there is

$$\lim_{\omega\to\infty} \frac{d}{d\omega} \angle G(s)\mathrm{e}^{-j\omega\tau} = -\tau$$

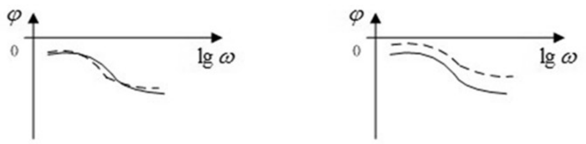
4

Therefore, the delay time τ of the delay unit can be determined according to the phase angle change rate of the experimentally obtained phase-frequency characteristics when the frequency ω tends to infinity.

However, it is difficult to measure the experimental data of the phase-frequency characteristics at high frequencies.

So, the following method can be used in engineering to determine the pure delay of the system.

Logarithmic frequency characteristic curves:



Without delay	With delay

As shown above, the solid line in the figure is the logarithmic phase-frequency curve obtained by the experiment, and the dotted line is the logarithmic phase-frequency characteristic determined by the fitted transfer function $G'(s)$.

If the dashed and solid lines are very close, the system contains no delay element.

If the dotted and solid lines differ by much, then there is a pure delay in the system.

Select a number of frequencies $\omega_k (k = 1,2,\cdots,n)$, and corresponding to each ω_k can find the phase difference $\Delta\varphi_k = \varphi_k' - \varphi_k$ between the measured curve and the fitted curve , so

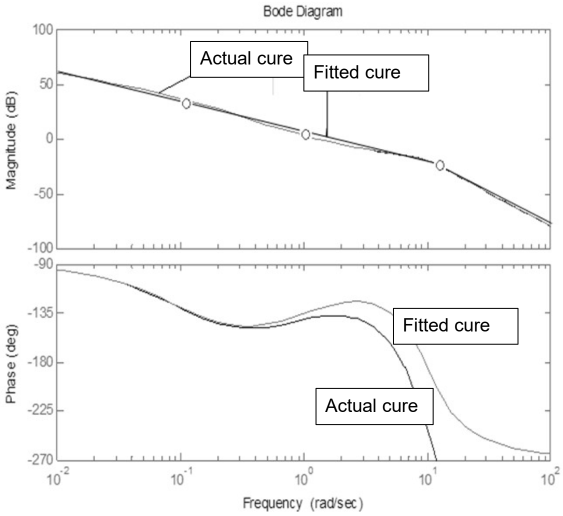
$$\tau_k = \frac{\Delta\varphi_k}{\omega_k} = \frac{\varphi_k' - \varphi_k}{\omega_k}, \qquad k = 1,2,\cdots,n$$

Then find the average value,

$$\tau = \frac{1}{n}(\tau_1 + \tau_2 + \cdots + \tau_n)$$

which can be used as the pure delay of the system.

Example: Suppose the experimental frequency response curve of a system is shown below. Try to determine the transfer function of the system.



(1) According to the slope of the approximate logarithmic magnitude-frequency curve at low frequency is $-20\mathrm{dB/dec}$, it can be seen from Table 1 that the measured plant contains an integral unit.

$$s^n (n = 1)$$

(2) The approximate logarithmic magnitude-frequency curve has 3 corner frequencies, namely 0.1rad/sec, 1rad/sec and 10rad/sec, which are determined according to the slope change at the corner frequency and the resonance peak near the corner frequency 10rad/sec, damping ratio and time constant of the transfer function.

The corresponding standard form: $$G(s) = \frac{K(Ts+1)}{s^n(T_1 s+1)[(T_2 s)^2 + 2T_2\zeta s + 1]}$$

where

$$\omega_1 = 0.1, \qquad \omega = 1, \qquad \omega_2 = 10$$

$$T_1 = \frac{1}{\omega_1} = 10, \qquad T = \frac{1}{\omega} = 1.0, \qquad T_2 = \frac{1}{\omega_2} = 0.10$$

Find out that the resonant peak from the figure. From

$$\mathrm{M}_r = \begin{cases} \dfrac{1}{2\zeta\sqrt{1-\zeta^2}}, \text{ for } \zeta \le 0.707 \\ 1, \qquad\qquad \text{for } \zeta > 0.707 \end{cases}$$

calculate the damping ratio, *e.g.*

$$\zeta = 0.5$$

Then the transfer function of the system under test can be written as

$$G(s) = \frac{K(s+1)}{s(10s+1)\left[\left(\frac{s}{10}\right)^2 + \frac{s}{10} + 1\right]}$$

According to $\omega = 0.01$, the magnitude is 60dB, that is,

$$20\,\mathrm{lg}|G(j\omega)|_{\omega=0.01} = 60$$

which gives $$20\,\mathrm{lg}\left|\frac{K(0.01j+1)}{0.01(0.1j+1)\left[\left(\frac{0.01j}{10}\right)^2 + \frac{0.01j}{10} + 1\right]}\right| = 60$$

Then, the proportional gain of the system under test can be approximated as K=10.

Through the above analysis, the transfer function of the actual model can be obtained as

$$G(s) = \frac{10(s+1)}{s(10s+1)\left[\left(\frac{s}{10}\right)^2 + \frac{s}{10} + 1\right]}$$

The above is only a transfer function obtained according to the magnitude-frequency characteristic, so it is only tentative.

It can be seen from the figure that the asymptotic phase curve $\angle G$ is inconsistent with the experimental result.

The fitted phase-frequency curve does not match the actual one.

When ω=1, the difference between the experimental curve and it is about -5 degrees.

When ω=10, the difference between the experimental curve and it is about -60 degrees.

This shows that the actual transfer function contains a delay unit.

Considering $G(s)e^{-\tau s}$,

$\tau = 5$ is consistent with the phase-frequency characteristics of the experimental curves.

So, the transfer function of the system under test can be modified as

$$G(s) = \frac{10(s+1)\mathrm{e}^{-5s}}{s(10s+1)\left[\left(\frac{s}{10}\right)^2 + \frac{s}{10} + 1\right]}$$

11

♦ **Using MATLAB tools to find the system transfer function**

The transfer function for a continuous system is

$$\mathrm{G}(s) = \frac{\beta_0 + \beta_1 s + \beta_2 s^2 + \ldots + \beta_m s^m}{1 + \alpha_1 s + \alpha_2 s^2 + \ldots + \alpha_n s^n}$$

Given a discrete frequency sampling point $\{\omega_i\}, i = 1,2,\cdots,N,$

it is assumed that the frequency response data of the tested system are $\{P_i, Q_i\}$

$$H_i = P_i + jQ_i$$

12

In the MATLAB signal processing toolbox, a function invfreqs() for identifying the system transfer function model is given.

The calling format of this function is

[B,A]=invfreqs(H,W,m,n)

where W is a vector composed of discrete frequency points, m and n are the numerator and denominator orders of the system to be identified, H is a complex vector, and its real and imaginary parts are the real and imaginary parts used in identification.

The returned B and A are the coefficient vectors of the numerator and denominator of the identified transfer function, respectively.

The transfer function can be obtained through A and B.

13

Matlab explanation of function **invfreqs():**

- >> help invfreqs
- INVFREQS Analog filter least squares fit to frequency response data.
- [B,A] = INVFREQS(H,W,nb,na) gives real numerator and denominator coefficients B and A of orders nb and na respectively, where H is the desired complex frequency response of the system at frequency points W, and W contains the frequency values in radians/s.
- INVFREQS yields a filter with real coefficients. This means that it is sufficient to specify positive frequencies only; the filter fits the data conj(H) at -W, ensuring the proper frequency domain symmetry for a real filter.

14

The following two examples illustrate the Matlab function to identify transfer functions:

Examples 1: The transfer function identification of the first-order continuous system verifies function invfreqs():

$$G(s) = \frac{1}{s+5}$$

freqs function

H = FREQS(B,A,W) returns the complex frequency response vector H of the filter B/A:

```
         B(s)
H(s) =  -------
         A(s)
```

Simulation program:

```
close all;
w= logspace(-1,1)
num = [1]
den = [1,5]
H=freqs(num,den,w)
[num,den] = invfreqs(H,w,0,1);
G=tf(num,den)
```


Example 2: Assuming that the frequency response value of the system measured on the frequency range w is H, the frequency range w and the frequency response value H are obtained as follows:

- w= logspace(-1,1)
- H = [0.9892 - 0.1073i 0.9870 - 0.1176i 0.9843 - 0.1289i 0.9812 - 0.1412i 0.9773 - 0.1545i 0.9728 - 0.1691i 0.9673 - 0.1848i 0.9608 - 0.2017i 0.9530 - 0.2200i 0.9437 - 0.2396i 0.9328 - 0.2605i 0.9198 - 0.2826i 0.9047 - 0.3058i 0.8869 - 0.3301i 0.8662 - 0.3551i 0.8424 - 0.3805i 0.8150 - 0.4060i 0.7840 - 0.4310i 0.7491 - 0.4549i 0.7103 - 0.4771i 0.6677 - 0.4968i 0.6216 - 0.5133i 0.5725 - 0.5258i 0.5210 - 0.5335i 0.4680 - 0.5361i 0.4144 - 0.5331i 0.3613 - 0.5242i 0.3099 - 0.5098i 0.2613 - 0.4900i 0.2164 - 0.4654i 0.1762 - 0.4370i 0.1413 - 0.4057i 0.1121 - 0.3728i 0.0886 - 0.3393i 0.0706 - 0.3064i 0.0577 - 0.2753i 0.0489 - 0.2466i 0.0436 - 0.2210i 0.0406 - 0.1987i 0.0391 - 0.1796i 0.0383 - 0.1635i 0.0377 - 0.1499i 0.0369 - 0.1385i 0.0356 - 0.1287i 0.0339 - 0.1201i 0.0318 - 0.1123i 0.0293 - 0.1051i 0.0266 - 0.0983i 0.0239 - 0.0919i 0.0212 - 0.0857i];
- logspace(X1, X2) generates a row vector of 50 logarithmically equally spaced points between decades 10^X1 and 10^X2.

Simulation program:

```
clear all;
close all;
w= logspace(-1,1);
H = [ 0.9892 - 0.1073i   0.9870 - 0.1176i   0.9843 - 0.1289i   0.9812 - 0.1412i
0.9773 - 0.1545i   0.9728 - 0.1691i   0.9673 - 0.1848i   0.9608 - 0.2017i   0.9530 -
0.2200i   0.9437 - 0.2396i   0.9328 - 0.2605i   0.9198 - 0.2826i   0.9047 - 0.3058i
0.8869 - 0.3301i   0.8662 - 0.3551i   0.8424 - 0.3805i   0.8150 - 0.4060i   0.7840 -
0.4310i   0.7491 - 0.4549i   0.7103 - 0.4771i   0.6677 - 0.4968i   0.6216 - 0.5133i
0.5725 - 0.5258i   0.5210 - 0.5335i   0.4680 - 0.5361i   0.4144 - 0.5331i   0.3613 -
0.5242i   0.3099 - 0.5098i   0.2613 - 0.4900i   0.2164 - 0.4654i   0.1762 - 0.4370i
0.1413 - 0.4057i   0.1121 - 0.3728i   0.0886 - 0.3393i   0.0706 - 0.3064i   0.0577 -
0.2753i   0.0489 - 0.2466i   0.0436 - 0.2210i   0.0406 - 0.1987i   0.0391 - 0.1796i
0.0383 - 0.1635i   0.0377 - 0.1499i   0.0369 - 0.1385i   0.0356 - 0.1287i   0.0339 -
0.1201i   0.0318 - 0.1123i   0.0293 - 0.1051i   0.0266 - 0.0983i   0.0239 - 0.0919i
0.0212 - 0.0857i];
[num,den] = invfreqs(H,w,3,4);
G=tf(num,den)
```

17

Using the above frequency response data, the identified transfer function is obtained as

```
 1.001 s^3 + 6.812 s^2 + 22.89 s + 20.59
-----------------------------------------------------------
s^4 + 9.816 s^3 + 33.29 s^2 + 45.2 s + 20.58
```

It can be seen from the simulation results that the transfer function identification result can be obtained by using the invfreqs() function.

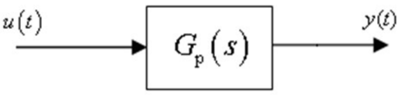
18

- **Identification of the transfer function of an open-loop linear system**

Basic principle: The open-loop transfer function can be identified by fitting the Bode plot. The block diagram of the open-loop transfer function test is shown below.

Let the input signal of the open-loop system be

$$u(t) = A_m \sin(\omega t)$$



where A_{m} and ω are the magnitude and angular velocity of the input signal, respectively.

Assuming the open-loop system is linear, its output can be expressed as:

$$\begin{aligned} y(t) &= A_f \sin(\omega t + \phi) \\ &= A_f \sin(\omega t)\cos(\phi) + A_f \cos(\omega t)\sin(\phi) \\ &= [\sin(\omega t) \quad \cos(\omega t)] \begin{bmatrix} A_f \cos(\phi) \\ A_f \sin(\phi) \end{bmatrix} \end{aligned} \tag{3.5.1}$$

where A_f and ϕ are the magnitude and phase of the output, respectively.

Take the sampling time

$$t = 0, h, 2h, \cdots, nh$$

Let

$$Y^{\mathrm{T}} = [y(0) \quad y(h) \quad \cdots \quad y(nh)]$$

$$\Psi^{\mathrm{T}} = \begin{bmatrix} \sin(\omega 0) & \sin(\omega h) & \cdots & \sin(\omega nh) \\ \cos(\omega 0) & \cos(\omega h) & \cdots & \cos(\omega nh) \end{bmatrix}$$

$$c_1 = A_f \cos(\phi) \qquad c_2 = A_f \sin(\phi)$$

From equation (3.5.1),

$$Y = \Psi \cdot \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{3.5.2}$$

From eq (3.5.2), according to the least squares principle, c_1 and c_2 can be obtained as:

$$\begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \end{bmatrix} = \left(\Psi^{\mathrm{T}} \Psi\right)^{-1} \Psi^{\mathrm{T}} Y$$

For the angular frequency ω, the magnitude and phase of the output signal of the open-loop system are as follows:

$$A_{\mathrm{f}} = \sqrt{\hat{c}_1^2 + \hat{c}_2^2} \qquad\qquad \phi = \mathrm{tg}^{-1}\left(\frac{\hat{c}_2}{\hat{c}_1}\right)$$

The phase is the phase difference between the output signal and the input signal.
The magnitude is expressed in decibels as the ratio of the steady-state output magnitude to the input magnitude.

Since the phase shift of the input signal $u(t) = A_{\mathrm{m}}\sin(\omega t)$ is zero, the phase and magnitude of the open-loop system are:

$$\phi_{\mathrm{e}} = \phi_{out} - \phi_{in} = \phi - 0 = \mathrm{tg}^{-1}\left(\frac{\hat{c}_2}{\hat{c}_1}\right)$$

$$M = 20lg\left(\frac{A_f}{A_m}\right) = 20lg\left(\frac{\sqrt{\hat{c}_1^2 + \hat{c}_2^2}}{A_m}\right)$$

21

Identification steps of a transfer function:

① The angular frequency sequence $\{\omega_i\}, i = 0,1,\cdots,n$, is taken in the frequency segment to be measured.

② For each angular frequency point, the phase and magnitude are calculated by the above method, and the frequency characteristic data of the open-loop system can be obtained.

③ The open-loop transfer function can be realized by using the Matlab frequency domain function invfreqs(.).

Example: The transfer function of the plant is:

$$G_{\mathrm{p}}(s) = \frac{133}{\mathrm{s}^2 + 25\mathrm{s} + 10}$$

The sampling period is 1ms, namely $h = 0.001$. The input signal is a sinusoidal frequency signal $u(t) = 0.5\sin(2\pi f t)$ with an magnitude of 0.5, the starting frequency is 1.0Hz, the ending frequency is 10Hz, and the step size is 0.5Hz.

For each frequency point, run 20,000 sampling times and record the data of the sampling interval of $[10000, 15000]$.

22

After calculating the phase and magnitude of the actual open-loop system at each frequency point, the complex representation of the frequency characteristics of the open-loop system can be written, namely .

$$h_{\mathrm{p}} = M(\cos(\phi_{\mathrm{e}}) + j\sin(\phi_{\mathrm{e}}))$$

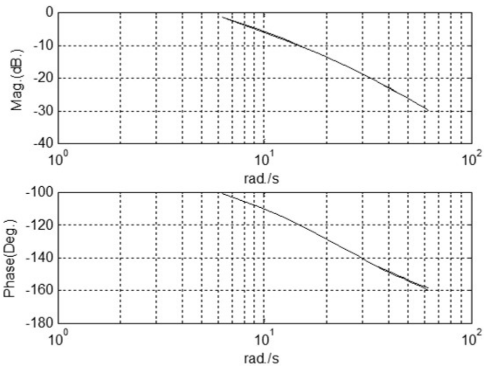
Taking $\omega = 2\pi F$, using the Matlab function $\mathrm{invfreqs}(h_{\mathrm{p}}, w, nb, na)$, the numerator *bb* and denominator *aa* coefficients of the transfer function corresponding to the complex frequency characteristic h_{p} with numerator and denominator orders of nb and na, respectively, are obtained, and thus the transfer function for the open-loop system is obtained.

Using the Matlab function $\mathrm{freqs}(bb, aa, w)$ to verify the identified transfer function, the complex frequency representation of the open-loop transfer function whose numerator and denominator orders are *bb and aa,* respectively, and can be obtained, so as to obtain the magnitude and phase plots of the fitted open-loop system transfer function.

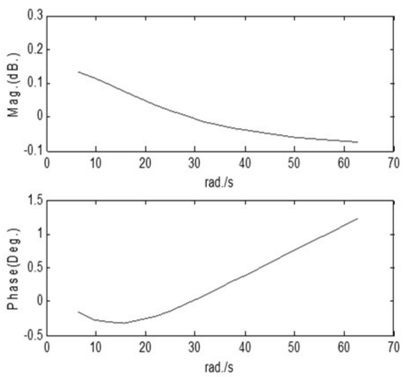
Through simulation, the open-loop transfer function can be obtained as

$$G_{\mathrm{c}}(s) = \frac{131.3}{s^2 + 24.28s + 10.08}$$

The simulation results are shown below.

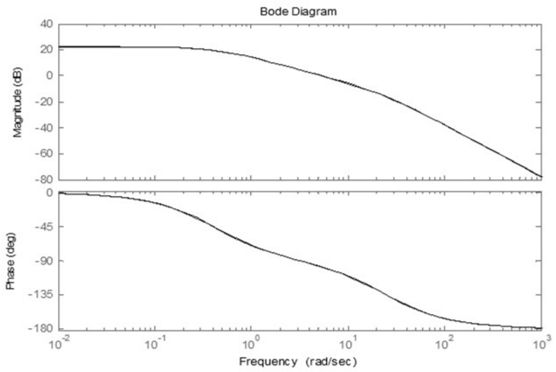


Comparison of Bode plots between the actual test and the fitted transfer function



Frequency characteristic fitting error curve

Bode plot comparison between actual plant and fitted transfer function



Open-loop system identification simulation program

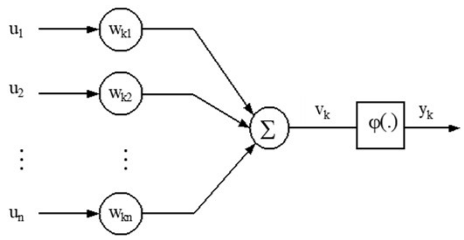
It can be seen that the algorithm can calculate the magnitude and phase of the open-loop transfer function very accurately, so that the identification of the open-loop transfer function can be realized accurately.

3.6 System Identification Based on Neural Networks

■ Neuron model

Neuron is the basic information processing unit of neural network operation.

There are three basic elements in the neuron model: link, adder and activation function.



Each link has its own weighting.

The adder sums the weighted input signals.

The activation function limits the amplitude of neuron output.

In mathematical terms, neurons can be described by the following two equations:

$$v_k = \sum_{j=1}^{n} w_{kj} u_j$$
$$y_k = \varphi(v_k)$$

where

u_j: the input signal

w_{kj}: the weight of the neuron

v_k: the linear combiner link

$\varphi(.)$: the activation function

y_k: the output signal of neurons

27

The activation function determines the output of neurons according to the activity level of inputs.

There are many types of activation functions.

Two basic activation functions: threshold function and s-curve function

$$\varphi(v) = \begin{cases} 1 & \text{if} \quad v \geq 0 \\ 0 & \text{if} \quad v < 0 \end{cases} \qquad\qquad \varphi(v) = \frac{1}{1 + e^{-av}}$$

where a is the slope parameter of the s-shaped curve function.

The s-shaped curve functions with different slopes can be obtained by changing the parameter a.

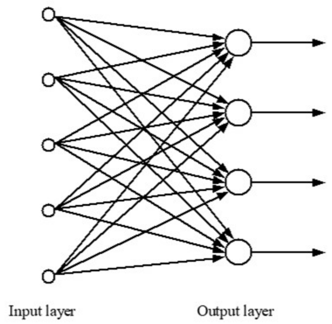
When a approaches infinity, the s-shaped curve function becomes a simple threshold function.

The s-shaped curve function is differentiable, while the threshold function is nondifferentiable.

28

■ **Neural network structure**

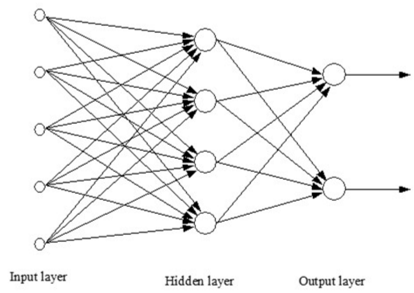
♦ **Single-layer neural networks**



The network composed of one layer of neurons is called single-layer neural network.

Single-layer neural network is a strict feedforward type.

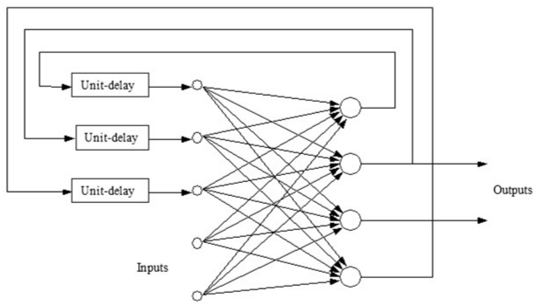
♦ **Multilayer neural networks**



The multilayer neural network has an input layer, one or more hidden layers and an output layer.

Each layer is composed of multiple neurons, and each neuron is connected to the neurons of the next layer.

♦ **Recurrent neural networks**



The recurrent neural network has at least one feedback loop, which distinguishes itself from the feedforward neural network.

The cyclic network may consist of single or multiple neurons, and each neuron can feed back its output signal to all other neurons as an input.

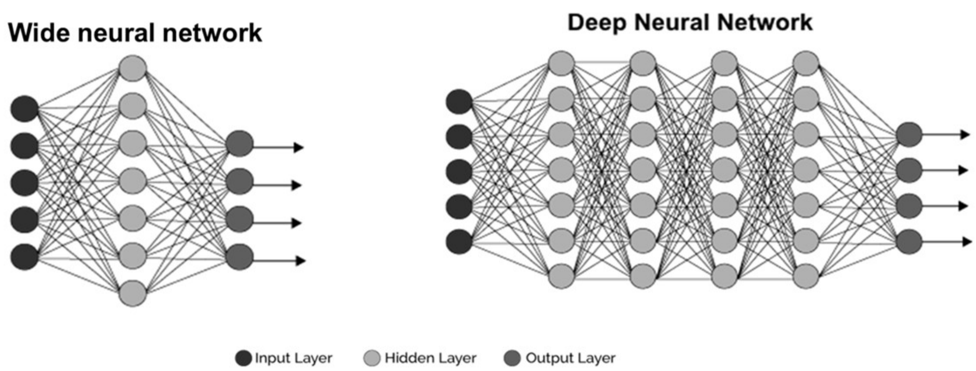
31

♦ **Wide neural networks**

There are a great number of neurons in one single layer.

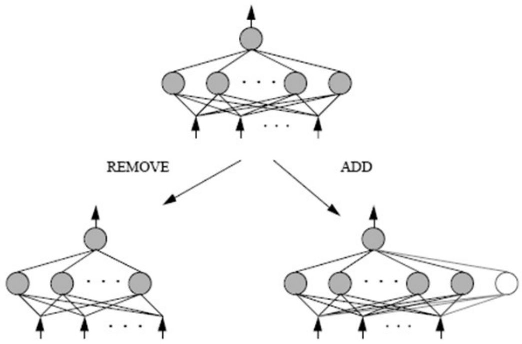
♦ **Deep neural networks**

There are a great number of hidden layers.



32

♦ Variable neural networks



The variable neural network has the following characteristics:

According to the design strategy, the number of neurons in the network can increase or decrease with time.

♦ General form of neural network

The function of neural network is mainly used to describe the nonlinear relationship between input and output.

The neural network is parameterizable and can be described as a linear combination of basis functions.

$$f(u, w) = \sum_{k=1}^{m} w_k \varphi_k(u)$$

where w_k is the weighting coefficient, $\varphi_K(.)$ base function (or activation function), m is the number of base functions used in the network mapping.

Self Assessment Question

For single input single output system:

$$2y(t) + 4y(t-1) + 6y(t-2) + 3r(t-1) = 0$$

The above system is described by the following neural network.

$$y(t) = \sum_{k=1}^{3} w_k \varphi_k\big(u(t)\big)$$

where

$$u(t) = [\, y(t-1),\, y(t-2),\, r(t-1)]$$

Determine the weight coefficients and basis functions of the network, and sketch the block diagram of the network.

■ Neural network learning

Neural networks learn from the examples provided to them, which appear as input-output pairs.

Neural networks map input modes to output modes, that is:

$$f:\ u \mapsto y$$

For these learning examples, one assumption is that they are consistent with the underlying mapping, such as $f^*(.)$.

The relationship between input and output can then be expressed as

$$y = f^* + v$$

where v is the measurement noise, which is an unknown random signal.

For a given function f(u), find the function $f(u;w)$ with the minimum approximation error, that is,

$$\min\|f(u) - f(u;w)\|_2$$

Based on the given set of learning examples D, the learning approximation accuracy of a neural network is measured by the following function

$$J(D,w) = \sum_{k=1}^{n} \|y_k - f(u_k;w)\|_2$$

The learning approximation problem of a neural network can be described as

$$w^* = \arg\min_{w} J(D;w)$$

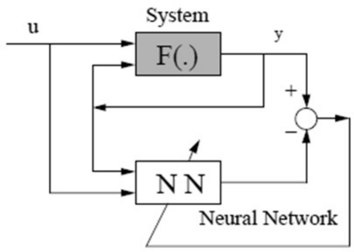
■ Neural network identification

Based on the input-output relationship of the system, discrete non-linear systems can also be represented by a non-linear autoregressive (NARX) model.

$$y_t = F(y_{t-1}, y_{t-2}, \ldots, y_{t-n}, u_{t-1}, u_{t-2}, \ldots, u_{t-m})$$

where *F*(.) is a non-linear function and n and m are the corresponding maximum delays.

♦ **A nonlinear identification structure based on a neural network**



It is assumed that the non-linear function F(.) in the NARX model is approximated by a single-layer neural network composed of a linear combination of basis functions.

$$\hat{F}(x_t) = \sum_{k=1}^{N} w_k \varphi_k(x_t)$$

where $x_t = [y_{t-1}, y_{t-2}, \ldots, y_{t-n}, u_{t-1}, u_{t-2}, \ldots, u_{t-m}]$

39

♦ **Example**

A common NARX model

$$\begin{aligned}\hat{F}(x_t) = {} & w_1 + w_2 y_{t-1} + \ldots + w_{n+1} y_{t-n} + \\ & w_{n+2} u_{t-1} + w_{n+3} u_{t-2} + \ldots + w_{n+m+1} u_{t-m} + \\ & w_{n+m+2} y_{t-1}^2 + w_{n+m+3} y_{t-1} y_{t-2} + \ldots + w_N u_{t-m}^r \\ = {} & \sum_{k=1}^{N} w_k \varphi_k(x_t)\end{aligned}$$

where

$$[\varphi_1, \varphi_2, \ldots, \varphi_{n+1}, \varphi_{n+2}, \varphi_{n+3}, \ldots, \varphi_{n+m+1}, \varphi_{n+m+2}, \varphi_{n+m+3}, \ldots, \varphi_N] =$$
$$[1, y_{t-1}, \ldots, y_{t-n}, u_{t-1}, \; u_{t-2}, \; \ldots, u_{t-m}, \;\; y_{t-1}^2, \;\; y_{t-1} y_{t-2}, \ldots, u_{t-m}^r]$$

40

♦ **Direct identification method**

According to the general approximation theorem, there are finite basis functions, which enable the neural network to approximate the nonlinear function accurately.

Estimation for discrete nonlinear systems is

$$\hat{y}(t|\theta) = \Phi^T(t)\theta$$

where

$$\theta = [w_1,\ w_2,\ \cdots,\ w_N]^T$$

$$\Phi(t) = [\varphi_1(x_t),\ \varphi_2(x_t),\ \cdots,\ \varphi_N(x_t)]^T$$

41

■ Example of nonlinear system identification

The identified nonlinear system is

$$y_t = \frac{y_{t-1}y_{t-2}y_{t-3}u_{t-2}(y_{t-3}-1)+u_{t-1}}{1+y_{t-2}^2+y_{t-3}^2}$$

Selected basis functions and initial weighting factors are given in the right table.

The input signal is

$$u_t = \begin{cases} \sin\left(\dfrac{\pi t}{125}\right), & t \le 500 \\ 0.8\sin\left(\dfrac{\pi t}{125}\right) + 0.2\sin\left(\dfrac{2\pi t}{25}\right), & t > 500 \end{cases}$$

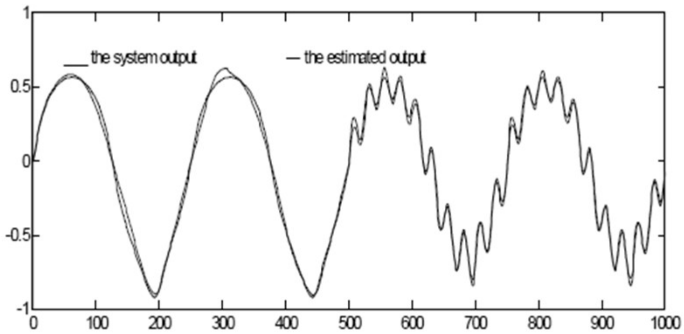
Priority Order i	VPBF φ_i^o	Weight w_i^o
1	u_{t-1}	0.9907
2	$y_{t-2}^2 u_{t-1}$	-0.6967
3	$y_{t-3}^2 u_{t-1}$	-0.7903
4	$y_{t-2}y_{t-3}$	-0.1437
5	$y_{t-2}u_{t-2}^2$	0.4680
6	$y_{t-1}^2 u_{t-3}$	-0.4705
7	$y_{t-2}u_{t-2}u_{t-3}$	0.3687
8	$y_{t-3}u_{t-3}$	0.0819
9	$y_{t-1}y_{t-2}u_{t-3}$	-0.3732
10	$y_{t-2}y_{t-3}u_{t-1}$	0.0244
11	$y_{t-2}^2 u_{t-3}$	-0.0304
12	$y_{t-1}y_{t-2}$	0.0052
13	$u_{t-1}u_{t-2}u_{t-3}$	0.2794
14	$y_{t-3}u_{t-2}^2$	-2.9104
15	$y_{t-3}u_{t-1}^2$	-0.0191
16	y_{t-2}	0.0052
17	$y_{t-3}u_{t-3}^2$	0.0177
18	$y_{t-3}u_{t-1}$	-0.0046
19	$y_{t-1}^2 y_{t-3}$	-4.2739
20	$y_{t-1}y_{t-3}u_{t-2}$	7.0854

42

The neural network model

$$\hat{y}_t = \sum_{k=1}^{20} w_{k,t} \varphi_k(x_t)$$

The online recursive least squares method is used to identify w_k with the following results:



Self Assessment Question

Consider a non-linear system below.

$$y(t) = 0.1y(t-1)(1+2\sin(u(t-1))) + 0.2y^2(t-2)(u(t-2)+3e^{-y(t-3)})$$

The above system can be described by the following neural network.

$$y(t) = \sum_{k=1}^{4} w_k \varphi_k(x_t)$$

where

$$x_t = [y(t-1),\ y(t-2),\ y(t-3),\ u(t-1),\ u(t-2]$$

Determine the weight coefficients and basis functions of the neural network.

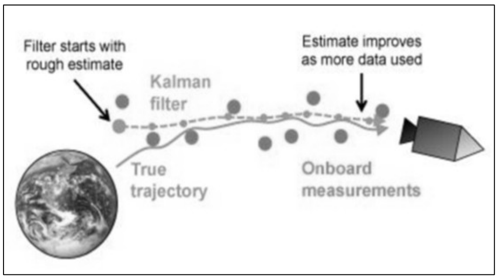
3.7 Parameter Estimation Using the Kalman Filter

■ History of the Kalman filter

Named after Rudolf E. Kalman ,who published his famous paper describing a recursive solution for the linear filtering problem using discrete data in 1960.

The first application of the Kalman filter was in the 1960s was the Apollo project, where the Kalman filter was used to estimate the trajectory of spacecraft to the moon and back.

- Filter starts with rough estimate
- Kalman filter
- True trajectory
- Estimate improves as more data used Onboard measurements



■ Background Knowledge

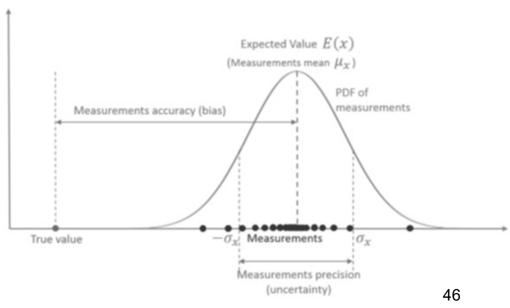
- Measurement is a random variable, described by the Probability Density Function (PDF).
- The measurement mean is the Expected Value of a random variable.
- The offset between the measurement mean and the true value is the measurement accuracy (or bias or measurement error).
- The dispersion of the distribution is known as precision or (measurement noise or measurement uncertainty).

Mean $$E(X) = \mu_x = \frac{1}{N}\sum_{n=1}^{N} x_n$$

Variance $$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

Gaussian PDF $$f\left(x; \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Measurements- Gaussian



- **Main characteristics of measurements**
  - **Accuracy** – how well the sensor measures the environment in an absolute sense. That is how good the data is when compared with a recognized standard. e.g., a temperature sensor accurate to 0.001º C is expected to agree within 0.001º C with a temperature standard.
  - **Resolution** – the ability of a sensor to see small differences in readings. e.g., a temperature sensor may have a resolution of 0. 01º C, but only be accurate to 0.1º C.
  - **Repeatability** – This is the ability of a sensor to repeat a measurement when put back in the same environment. It is often directly related to accuracy, but a sensor can be inaccurate, yet be repeatable in making observations.

- **Specifications of a measurement system**

  This may include factors such as:
  - The various types and ranges of measurements, i.e., performance characteristics
  - The quality of performance (accuracy, precision etc…)
  - Technical details, such as electric power supply requirements etc.

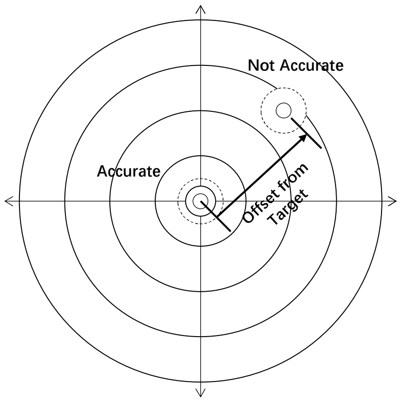
- Two types of specifications are associated with measurement systems:
  - Static characteristics
  - Dynamic characteristics

- **Static vs dynamic characteristics**
  - Static characteristics - steady state reading, i.e., reading taken after the instrument settles down.
  - Dynamic characteristics of an instrument describe its behaviour between the time when a measured quantity (measurand) changes values and the time when the instrument output attains a steady value in response.

- **Some key specifications**
  - Accuracy
  - Precision (Repeatability)
  - Resolution (Threshold)

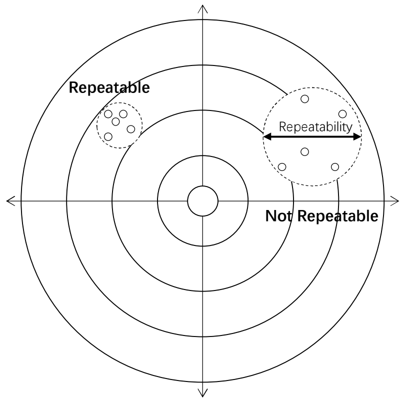
- ♦ **Accuracy**

Accuracy is the extent to which a reading can be wrong or closeness of output to input.

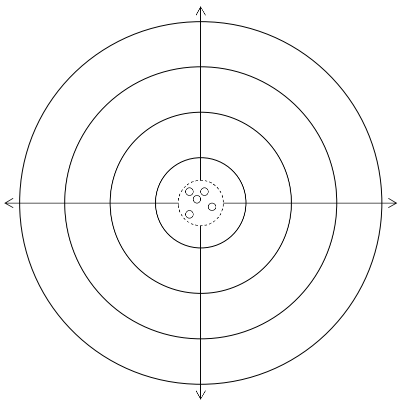


- ♦ **Precision**

Defined in terms of repeatability, which is the measure of an instrument's ability to reproduce a reading from a given input.



Understanding accuracy and repeatability is an important step to analyse system and transducer performance.



Both accurate and repeatable

Self Assessment Question

Which is correct in the following:

A. Dynamic characteristics means steady state reading.

B. The accuracy and repeatability of measurement are the same matter.

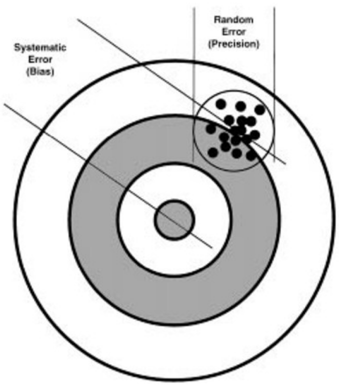
C. The precision and repeatability of measurement are the different matter.

D. A measurement instrument is a device or mechanism for measuring the value of a variable quantity under observation.

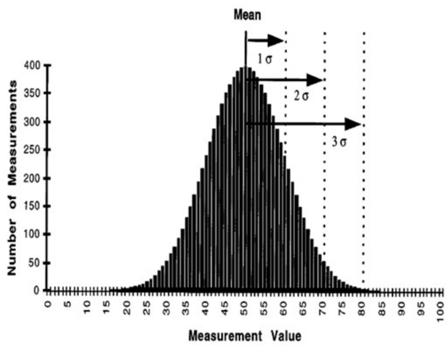
♦ **Accuracy vs precision**

High precision, but low accuracy.

There is a systematic error.



High accuracy means that the mean is close to the true value, while high precision means that the standard deviation is small.

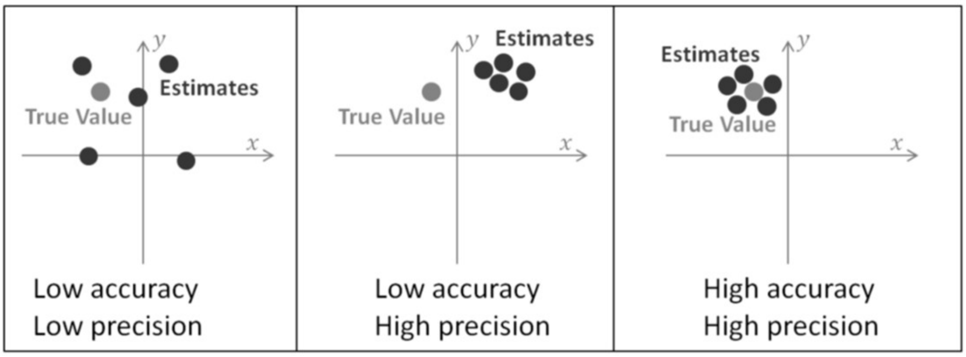


Precise instruments agree well but may all display considerable error.

It is possible to have high precision with poor accuracy.

There are various combinations between accuracy and precision.

Accuracy & Precision

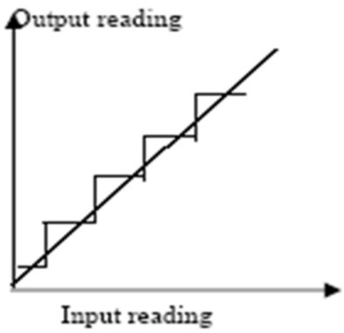


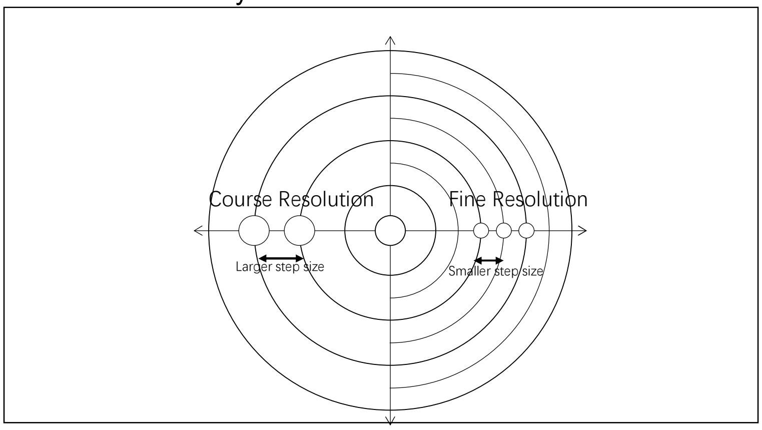
53

- **Resolution / threshold**

  The minimum level of input required to bring about a large enough magnitude of detectable change in output reading.

Resolution, also called step size, is the smallest possible change in a variable that the system (sensor) can detect. The resolution is determined by capability of the system and feedback.





54

Self Assessment Question

Which is correct in the following:

A. High precision means that the mean is close to the true value.

B. High accuracy means that the standard deviation is small.

C. High precision means high accuracy.

D. The resolution is determined by capability of the system and feedback.

■ Introduction to the Kalman filter

♦ What is a Kalman filter

A Kalman filter is an optimal estimator – *i.e.*, infers parameters of interest from indirect, inaccurate and uncertain observations.

It is recursive so that new measurements can be processed as they arrive.

♦ Optimal in what sense

- If Noise is Gaussian: the Kalman filter minimizes the mean square error of the estimated parameters.
- If Noise is NOT Gaussian: the Kalman filter is still the best linear estimator. Non-linear estimators may be better.
- Gauss-Markov Theorem – Optimal among all linear, unbiased estimators
- Rao–Blackwell theorem – Optimal among non-linear estimators with Gaussian noise

♦ **An estimator**: Optimal under linear or Gaussian and is on-line.

♦ **Why is the Kalman filter so popular:**

- Good results in practice due to optimality and structure
- Convenient form for online real time processing
- Easy to formulate and implement given a basic understanding
- Measurement equations need not be inverted

♦ **Why is the word "Filter" used**

- The process of finding the "best estimate" from noisy data amounts to "filtering out" the noise.
- The Kalman filter doesn't just clean up the data measurements, but also projects them onto the state estimate.

57

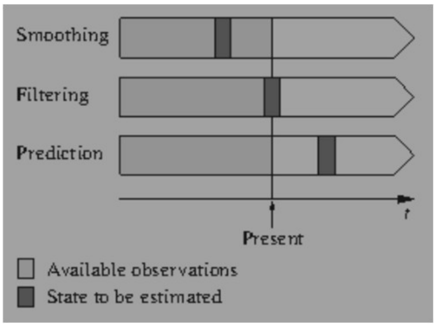
♦ **The Kalman filter: smoothing, filtering, prediction**

Real-time optimal estimation is desired when new data arrives.

There are three functions:

- Smoothing (take advantage of noise reduction)
- Filtering
- Prediction (extrapolate based on model)

Applications: controllers, tracking, etc.



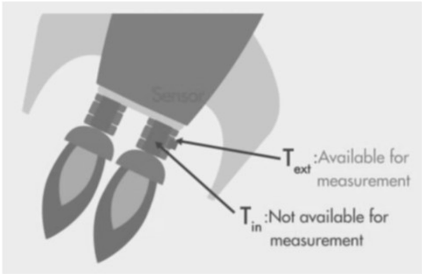
♦ **The Kalman filter: mechanism**

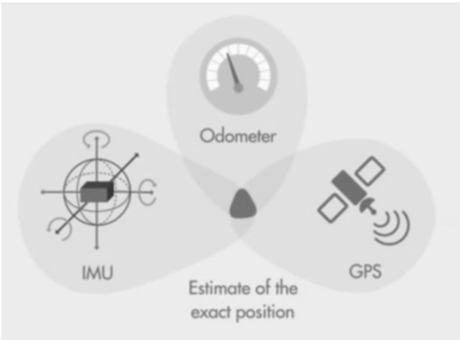
- Required: the system model and observations.
- The model may be uncertain and the measurements may be noisy
- Prediction-correction framework: optimal combination of the system model and observations

58

■ When is the Kalman filter used ?

The variables of interest can only be measured indirectly.





Measurements are available from various sensors but might be subject to noise.

♦ **The Kalman filter**

Recursive data processing algorithm
It is an optimal estimation algorithm that predicts the parameters of interest such as location, speed and direction in the presence of noisy measurements.

Optimal in what sense ?
If all noises are Gaussian and the system is linear , then Kalman Filter minimises the mean square error of estimated parameters, *i.e.*, give the best estimates based on previous measurements.

Recursive ?
Doesn't need to store all previous measurements and reprocess all data each time step.

♦ **Application example of the Kalman filter**

In spacecraft we need to monitor the internal temperature of the combustion chamber if it's get too high it could damage the mechanical component of jet engine.

This is not an easy task, since a sensor placed inside the chamber would melt.

Instead, it needs to be placed on a cooler surface close to the chamber.

The problem we're facing here is that we want to measure internal temperature of the chamber but we can't.

Instead, we have to measure external temperature.

In this situation, we can use a Kalman filter to find the best estimate of the internal temperature from an indirect measurement.

♦ **How does the Kalman filter work ?**

This algorithm is a recursive two-step process : prediction, and update.

The prediction step produces estimates of the current variables along with their uncertainties .

These estimates are based on the assumed model of how the estimates change over time.

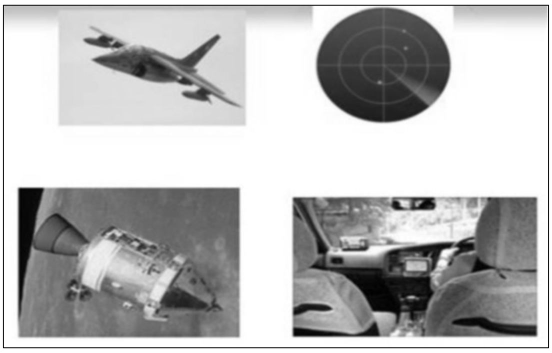
The update step is done when the next measurements (subject to noise) is observed.

In this step , the estimates (let's call it state from here on) are updated, based on the weighted average of the predicted state and the state based on the current measurement.

♦ **Application of the Kalman filter**

A common application is for guidance, navigation, and control of vehicles, particularly aircraft and spacecraft.

Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and econometrics.



63

■ The Luenberger observer

Given the linear dynamical system:
$$x(k+1) = Fx(k) + Gu(k)$$
$$y(k) = Hx(k)$$

where $x(k)$ is the n-dimensional state vector (unknown)
$u(k)$ is the m-dimensional input vector (known)
$y(k)$ is the p-dimensional output vector (known, measured)
F, G, H are appropriately dimensioned system matrices (known)

The Luenberger observer:
$$\hat{x}(k+1) = F\hat{x}(k) + Gu(k) + L(y(k) - H\hat{x}(k))$$
$$\hat{y}(k) = H\hat{x}(k)$$

where $\hat{x}(k)$ and $\hat{y}(k)$ are the estimated state vector and output vector, respectively
L is the observer gain matrix ($F - LH$ must be stable)

64

■ What does a Kalman filter do?

Given the linear dynamical system:

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$$

$$y(k) = H(k)x(k) + w(k)$$

where

$x(k)$ is the n-dimensional state vector (unknown)
$u(k)$ is the m-dimensional input vector (known)
$y(k)$ is the p-dimensional output vector (known, measured)
$F(k), G(k), H(k)$ are appropriately dimensioned system matrices (known)
$v(k), w(k)$ are zero-mean, white Gaussian noise with (known)
covariance matrices $Q(k), R(k)$

The Kalman Filter is a recursion that provides the "best" estimate of the state vector *x.*

65

♦ **What's so great about that?**

- Noise smoothing (improve noisy measurements)
- State estimation (for state feedback)
- Recursive (computes next estimate using only most recent measurement)

♦ **How does it work?**

1) Prediction based on last estimate: $\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k)$

$$\hat{y}(k+1) = H(k)\hat{x}(k+1|k)$$

2) Calculate correction based on prediction and current measurement:

$$\Delta_x(k+1) = f(y(k+1), \hat{x}(k+1|k))$$

3) Update prediction:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \Delta_x(k+1)$$

66

♦ **Kalman filter for stochastic processes**

Let's see an example of a car system

$u(k)$ →

$$x(k+1) = Fx(k) + Gu(k) + v(k)$$
$$y(k) = Hx(k) + w(k)$$

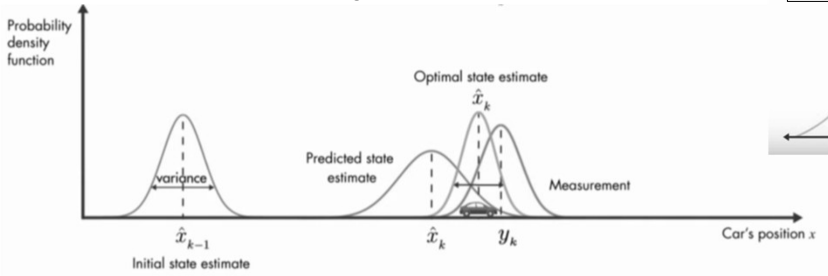
Car dynamics → $y(k)$

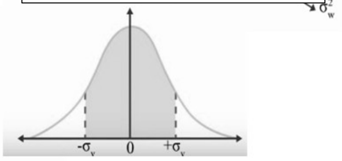
$$\hat{x}(k+1) = F\hat{x}(k) + Gu(k)$$
$$\hat{y}(k) = H\hat{x}(k)$$

Car model → $\hat{x}(k)$

Kalman filter combines the measurement and the prediction to find the optimal estimate of the car's position.

Noises: $v \sim N(0, Q)$, $w \sim N(0, R)$





♦ **A simple state observer**

System:
$$x(k+1) = Fx(k) + Gu(k) + v(k)$$
$$y(k) = Hx(k)$$

Observer:

1. prediction: $\boxed{\hat{x}(k+1|k) = F\hat{x}(k|k) + Gu(k)}$
2. compute correction: $\boxed{\Delta_x(k+1) = H^T\left(HH^T\right)^{-1}\left(y(k+1) - H\hat{x}(k+1|k)\right)}$
3. update: $\boxed{\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \Delta_x(k+1)}$

■ A better state observer

Assume $v(k)$ is a white noise with $N(0, Q)$.

We can create a better state observer following the same three steps, but now we must also estimate the covariance matrix *P*.

We start with $\hat{x}(k|k)$ and $P(k|k)$.

Step 1: Prediction

$$\hat{x}(k+1|k) = F\hat{x}(k|k) + Gu(k)$$

What about the covariance matrix P? From the definition,

$$P(k|k) = E\{(x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))^T\}$$

and

$$P(k+1|k) = E\{(x(k+1) - \hat{x}(k+1|k))(x(k+1) - \hat{x}(k+1|k))^T\}$$

69

To make life a little easier, let's shift the notation slightly:

$$\boxed{\begin{aligned} P_{k+1} &= P(k+1|k+1) \\ P_{k+1}^- &= P(k+1|k) \end{aligned}} \qquad \boxed{\begin{aligned} \hat{x}_{k+1} &= \hat{x}(k+1|k+1) \\ \hat{x}_{k+1}^- &= \hat{x}(k+1|k) \end{aligned}}$$

$$\begin{aligned}
P_{k+1}^- &= E\{(x_{k+1} - \hat{x}_{k+1}^-)(x_{k+1} - \hat{x}_{k+1}^-)^T\} \\
&= E\{(Fx_k + Gu_k + v_k - (F\hat{x}_k + Gu_k))(Fx_k + Gu_k + v_k - (F\hat{x}_k + Gu_k))^T\} \\
&= E\{(F(x_k - \hat{x}_k) + v_k)(F(x_k - \hat{x}_k) + v_k)^T\} \\
&= E\left\{F(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T F^T + 2F(x_k - \hat{x}_k)v_k^T + v_k v_k^T\right\} \\
&= FE\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\}F^T + E\left\{v_k v_k^T\right\} \\
&= FP_kF^T + Q
\end{aligned}$$

$$\boxed{P(k+1|k) = FP(k|k)F^T + Q}$$

70

Step 2: Computing the correction

From step 1, we get $\hat{x}(k+1|k)$ and $P(k+1|k)$.

Now we use these to compute $\Delta_x(k+1)$:

$$\Delta_x(k+1) = P(k+1|k)H\left(HP(k+1|k)H^T\right)^{-1}(y(k+1) - H\hat{x}(k+1|k))$$

For ease of notation, let

$$\Delta_x(k+1) = We$$

where

$$W = P(k+1|k)H\left(HP(k+1|k)H^T\right)^{-1}$$
$$e = y(k+1) - H\hat{x}(k+1|k)$$

Step 3: Update

$$\boxed{\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + We}$$

$$P_{k+1} = E\left((x_{k+1} - \hat{x}_{k+1})(x_{k+1} - \hat{x}_{k+1})^T\right)$$
$$= E\left((x_{k+1} - \hat{x}_{k+1}^{-} - We)(x_{k+1} - \hat{x}_{k+1}^{-} - We)^T\right)$$

(just take my word for it…)

$$\boxed{P(k+1|k+1) = P(k+1|k) - WHP(k+1|k)H^TW^T}$$

Just take my word for it…

$$
\begin{aligned}
P_{k+1} &= E\{(x_{k+1} - \hat{x}_{k+1})(x_{k+1} - \hat{x}_{k+1})^T\} \\
&= E\{(x_{k+1} - \hat{x}_{k+1}^- - We)(x_{k+1} - \hat{x}_{k+1}^- - We)^T\} \\
&= E\{((x_{k+1} - \hat{x}_{k+1}^-) - We)((x_{k+1} - \hat{x}_{k+1}^-) - We)^T\} \\
&= E\{(x_{k+1} - \hat{x}_{k+1}^-)(x_{k+1} - \hat{x}_{k+1}^-)^T - 2We(x_{k+1} - \hat{x}_{k+1}^-)^T + We(We)^T\} \\
&= P_{k+1}^- + E\{-2WH(x_{k+1} - \hat{x}_{k+1}^-)(x_{k+1} - \hat{x}_{k+1}^-)^T + WH(x_{k+1} - \hat{x}_{k+1}^-)(x_{k+1} - \hat{x}_{k+1}^-)^T H^T W^T\} \\
&= P_{k+1}^- - 2WHP_{k+1}^- + WHP_{k+1}^- H^T W^T \\
&= P_{k+1}^- - 2P_{k+1}^- H^T \left(HP_{k+1}^- H^T\right)^{-1} HP_{k+1}^- + WHP_{k+1}^- H^T W^T \\
&= P_{k+1}^- - 2P_{k+1}^- H^T \left(HP_{k+1}^- H^T\right)^{-1} \left(HP_{k+1}^- H^T\right)\left(HP_{k+1}^- H^T\right)^{-1} HP_{k+1}^- + WHP_{k+1}^- H^T W^T \\
&= P_{k+1}^- - 2WHP_{k+1}^- H^T W^T + WHP_{k+1}^- H^T W^T \\
&= P_{k+1}^- - WHP_{k+1}^- H^T W^T
\end{aligned}
$$

73

♦ **Better state observer** (without output noise)

System: $x(k+1) = Fx(k) + Gu(k) + v(k)$
$y(k) = Hx(k)$

where $v(k)$ is a white noise with $N(0, Q)$.

Observer

1. Prediction

$$\boxed{\begin{aligned} &\hat{x}(k+1|k) = F\hat{x}(k|k) + Gu(k) \\ &P(k+1|k) = FP(k|k)F^T + Q \end{aligned}}$$

2. Correction

$$\boxed{\begin{aligned} &W = P(k+1|k)H\left(HP(k+1|k)H^T\right)^{-1} \\ &\Delta_x(k+1) = W(y(k+1) - H\hat{x}(k+1|k)) \end{aligned}}$$

3. Update

$$\boxed{\begin{aligned} &\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \Delta_x(k+1) \\ &P(k+1|k+1) = P(k+1|k) - WHP(k+1|k)H^T W^T \end{aligned}}$$

74

♦ The Kalman Filter

System: $x(k+1) = Fx(k) + Gu(k) + v(k)$
$y(k) = Hx(k) + w(k)$

where $v(k)$ is a white noise with $N(0, Q)$ *and* $w(k)$ is a white noise with $N(0, R)$.

Kalman Filter

1. Prediction

$$\hat{x}(k+1|k) = F\hat{x}(k|k) + Gu(k)$$
$$P(k+1|k) = FP(k|k)F^T + Q$$

2. Correction

$$S = HP(k+1|k)H^T + \mathrm{R}$$
$$W = P(k+1|k)HS^{-1}$$
$$\Delta_x(k+1) = W(y(k+1) - H\hat{x}(k+1|k))$$

3. Update

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \Delta_x(k+1)$$
$$P(k+1|k+1) = P(k+1|k) - WSW^T$$

75

♦ The Kalman Predictor

System: $x(k+1) = Fx(k) + Gu(k) + v(k)$
$y(k) = Hx(k) + w(k)$

where $v(k)$ is a white noise with $N(0, Q)$ *and* $w(k)$ is a white noise with $N(0, R)$.

Predictor:

$$\hat{x}(k+1|k) = F\hat{x}(k|k-1) + Gu(k) + K(k)(y(k) - H(k)\hat{x}(k|k-1))$$
$$K(k) = FP(k|k-1)H^T(k)\left(H(k)P(k|k-1)H^T(k) + R\right)^{-1}$$
$$P(k+1|k) = FP(k|k-1) - K(k)\,H(k)P(k|k-1) + Q$$

76

■ Parameter estimation using the Kalman Filter

Difference equation of a system

$$y(k) + a_1 y(k-1) + \cdots + a_{n_a} y(k-n_a) \\ = b_1 u(k-1) + \cdots + b_{n_b} u(k-n_b) + v(k)$$

where $v(k)$ is a white noise sequence

Let the parameters of the system be defined as

$$\begin{aligned} a_1(k+1) &= a_1(k) + w_1(k) \\ &\vdots \\ a_{n_a}(k+1) &= a_{n_a}(k) + w_{n_a}(k) \\ \mathrm{b}_1(k+1) &= b_1(k) + w_{n_a+1}(k) \\ &\vdots \\ b_{n_b}(k+1) &= b_{n_b}(k) + w_{n_a+n_b}(k) \end{aligned}$$

where $w_i(k)$ is a white noise sequence.

77

Let

$$x(k) = \begin{bmatrix} a_1(k) \\ \vdots \\ a_{n_a}(k) \\ b_1(k) \\ \vdots \\ b_{n_b}(k) \end{bmatrix} \qquad H^T(k) = \begin{bmatrix} -y(k-1) \\ \vdots \\ -y(k-n_a) \\ u(k-1) \\ \vdots \\ u(k-n_b) \end{bmatrix}$$

$x(k)$ denotes the parameters to be estimated.

The estimate equation of the parameter can be described as

$$x(k+1) = x(k) + w(k)$$

where $w = \{w_i(k)\}$ is a white noise vector.

The difference equation of a system can be written as a state space form:

$$y(k) = H(k)x(k) + v(k)$$

78

Application of the Kalman predictor leads to the following recursive parameter estimation:

$$\hat{x}(k+1|k) = \hat{x}(k|k-1) + K(k)(y(k) - H(k)\hat{x}(k|k-1))$$
$$K(k) = P(k|k-1)H^T(k)\left(H(k)P(k|k-1)H^T(k) + R_2(k)\right)^{-1}$$
$$P(k+1|k) = P(k|k-1) - K(k)H(k)P(k|k-1) + R_1(k)$$

where

$$R_1(k) = E\{w(k)w^T(k)\}$$
$$R_2(k) = E\{v^2(k)\}$$

The initial values: $\hat{x}(0|-1) = x(0), P(0|-1) = P(0)$.

$\bar{x}(0), P(0), R_1(k)$ and $R_2(k)$ need to be detemined according to the measurement of the used sensors.

79

Self Assessment Question

For the difference equation of a system

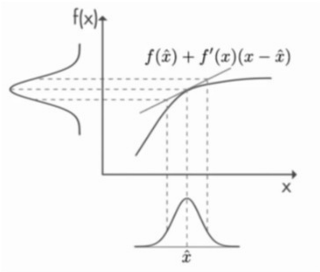
$$y(k) + a_1 y(k-1) + \cdots + a_{n_a} y(k-n_a)$$
$$= b_1 u(k-1) + \cdots + b_{n_b} u(k-n_b) + v(k)$$

how to obtain the following recursive parameter estimation formula given in the previous section?

$$\hat{x}(k+1|k) = \hat{x}(k|k-1) + K(k)(y(k) - H(k)\hat{x}(k|k-1))$$
$$K(k) = P(k|k-1)H^T(k)\left(H(k)P(k|k-1)H^T(k) + R_2(k)\right)^{-1}$$
$$P(k+1|k) = P(k|k-1) - K(k)H(k)P(k|k-1) + R_1(k)$$

80

■ The extended Kalman filter



Nonlinear system:

$$x_{k+1} = f(x_k, u_k, w_k, k)$$
$$y_k = h(x_k, v_k, k)$$

where $v_k \sim (0, R), w_k \sim (0, Q)$

Linearisation:

$$F = \frac{\partial f(x_k, u_k, w_k, k)}{\partial x_k}|_{\hat{x}_k, \hat{u}_k} \qquad G = \frac{\partial f(x_k, u_k, w_k, k)}{\partial u_k}|_{\hat{x}_k, \hat{u}_k} \qquad H = \frac{\partial h(x_k, v_k, k)}{\partial x_k}|_{\hat{x}_k}$$

Linearised system:

$$\boxed{\begin{aligned} \Delta x_{k+1} &= F\Delta x_k + G\Delta u_k + w_k \\ \Delta y_k &= H\Delta x_k + v_k \end{aligned}}$$

Then, the Kalman filter is applied to the above linearised system.

81

♦ Application of extended Kalman filters to parameter estimation

Consider a nonlinear system:

$$y_k = f\left(y_{k-1}, \dots, y_{k-n_a}, u_{k-1}, \dots, u_{k-k-n_b}, w_k, k\right)$$

Linearisation:

$$a_i = \frac{\partial f}{\partial y_{k-i}}|_{\hat{Y}_{k-1}, \hat{U}_{k-1}}, \qquad i = 1, 2, \dots, n_a$$

$$b_j = \frac{\partial f}{\partial u_{k-j}}|_{\hat{Y}_{k-1}, \hat{U}_{k-1}}, \qquad j = 1, 2, \dots, n_b$$

where the operating point is $\hat{Y}_{k-1} = \left[\hat{y}_{k-1}, \hat{y}_{k-2}, \dots, \hat{y}_{k-n_a}\right]$, $\hat{U}_{k-1} = \left[\hat{u}_{k-1}, \hat{u}_{k-2}, \dots, \hat{u}_{k-n_b}\right]$

The linearised system

$$\boxed{\Delta y_k \approx \sum_{i=1}^{n_a} a_i \Delta y_{k-i} + \sum_{j=1}^{n_b} b_j \Delta u_{k-j} + w_k}$$

So, the parameters of the nonlinear system can also be estimated by Kalman filter.

82

♦ **Drawbacks to using extended Kalman filters (EKFs):**

- It is difficult to calculate the Jacobians matrices (if they need to be found analytically)
- There is a high computational cost (if the Jacobians can be found numerically)
- EKF only works on systems that have a differentiable model
- EKF is not optimal if the system is highly nonlinear

Exercise 3.2

Consider the following system:

$$y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) + b_2 u(k-2) + v(k)$$

The input-output data are simulated by assuming

1) The system parameters are $a_1 = 1.6,\ a_2 = 0.7,\ b_1 = 1.0,\ b_2 = 0.4;$
2) The input signal adopts a 5^{th}-order M-sequence with an amplitude of 1;
3) The variance of the white noise $v(k)$ is 0.4.

Use the Kalman filter method to estimate the parameters of the above system if the parameter estimation noise $w(k)$ is assumed to be a white noise with a variance of 0.2.