4　オブザーバによる状態変数の復元

4.1　オブザーバとは？

制御対象が可観測であれば状態フィードバック制御により制御系の極（固有値）を任意に配置できる（ことが知られている）。ただし，この状態フィードバックを行うにはすべての状態変数を必要とする。しかし，現実には一部の状態変数が検出できないこともある。

「オブザーバ（観測器ともいう）」は，検出できるものを利用して状態変数の推定値を復元するものである。「オブザーバ」は，1964 年に D. G. Luenberger により提案された。「オブザーバ」とよく似ているものに「カルマンフィルタ」がある。「カルマンフィルタ」は，1960 年に R. E. Kalman によって提案された。両者の違いは，「オブザーバ」はノイズのない確定システムを対象とするのに対して，「カルマンフィルタ」は外乱および観測ノイズがある確率システムの出力からノイズを濾して状態を推定するものである。

本章では，検出できない状態変数を復元する方法に関して，順に，問題設定，出力微分による方法，シミュレータ，全次元オブザーバ，さらに，低次元オブザーバを概説し，最後に，制御と観測の分離定理を示す。

4.2　状態変数推定の問題設定

状態変数推定の問題とは，次のものである。
「次式の線形なシステムが与えられたとき，

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t)=\boldsymbol{Ax}(t)+\boldsymbol{Bu}(t)\quad : \boldsymbol{A}\ (n\times n),\ \boldsymbol{B}(n\times m) \cdots\cdots (1.a)$$

$$\boldsymbol{y}(t)=\boldsymbol{Cx}(t)\qquad : \boldsymbol{C}\ (\ell\times n) \cdots\cdots (1.b)$$

状態変数 $\boldsymbol{x}(t)$を復元せよ。ただし，**パラメータ** $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ と入出力 $\boldsymbol{u}(t), \boldsymbol{y}(t)$は既知である。 」◇

(1.a)の解は，初期値を

$$\boldsymbol{x}(0)=\boldsymbol{x}_0\qquad \text{(initial values)} \cdots\cdots (2)$$

とすれば，次式で与えられる。

$$\boldsymbol{x}(t)=\varepsilon^{\boldsymbol{A}t}\boldsymbol{x}_0+\int_0^t \varepsilon^{\boldsymbol{A}(t-\tau)}\boldsymbol{Bu}(\tau)\mathrm{d}\tau \cdots\cdots (3)$$

したがって，初期値 $\boldsymbol{x}_0$ が分かれば上式から状態変数 $\boldsymbol{x}(t)$を計算できる。ゆえに，状態変数推定の問題では，初期値 $\boldsymbol{x}_0$ をどのように推定するかがポイントとなる。

4.3　出力の微分を利用する方法

ここでは，出力を微分して状態変数を復元する方法を示す。出力 $\boldsymbol{y}(t)$を何回でも微分できるとすれば，状態変数 $\boldsymbol{x}(t)$を次のようにして復元できる。この方法は，理論的には大切である。なぜなら，可観測であるための必要十分条件が出てくるからである。しかし，微分を何回も行うとノイズを増幅するから実用的ではない。

まず，(1.b)を t で次々に微分して，(1.a)を利用すれば次のようになる。

$$\begin{aligned}
&\boldsymbol{y}(t)=\boldsymbol{Cx}(t)\\
&\dot{\boldsymbol{y}}(t)=\boldsymbol{C}\dot{\boldsymbol{x}}(t)=\boldsymbol{CAx}(t)+\boldsymbol{CBu}(t)\\
&\ddot{\boldsymbol{y}}(t)=\boldsymbol{CA}\dot{\boldsymbol{x}}(t)+\boldsymbol{CB}\dot{\boldsymbol{u}}(t)=\boldsymbol{CA}^2\boldsymbol{x}(t)+\boldsymbol{CABu}(t)+\boldsymbol{CB}\dot{\boldsymbol{u}}(t)\\
&\vdots\\
&\boldsymbol{y}^{(n-1)}(t)=\boldsymbol{CA}^{n-1}\boldsymbol{x}(t)+\boldsymbol{CA}^{n-2}\boldsymbol{x}(t)+\cdots+\boldsymbol{CBu}^{n-2}(t)
\end{aligned}$$

これらをまとめて行列で表せば，

$$\boldsymbol{Y}(t)=\boldsymbol{Ox}(t)+\boldsymbol{TU}(t) \cdots\cdots (4.a)$$

ただし，

$$\boldsymbol{Y}(t)=\begin{bmatrix} \boldsymbol{y}(t) \\ \dot{\boldsymbol{y}}(t) \\ \ddot{\boldsymbol{y}}(t) \\ \vdots \\ \boldsymbol{y}^{(n-1)}(t) \end{bmatrix} \cdots\cdots(4.b)$$

$$\boldsymbol{O}=\begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{n-1} \end{bmatrix} \cdots\cdots(4.c)$$

$$\boldsymbol{U}(t)=\begin{bmatrix} \boldsymbol{u}(t) \\ \dot{\boldsymbol{u}}(t) \\ \ddot{\boldsymbol{u}}(t) \\ \vdots \\ \boldsymbol{u}^{(n-1)}(t) \end{bmatrix} \cdots\cdots(4.d)$$

$$\boldsymbol{T}=\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \boldsymbol{CB} & 0 & \cdots & 0 & 0 \\ \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{CA}^{n-2}\boldsymbol{B} & \boldsymbol{CA}^{n-3}\boldsymbol{B} & \cdots & \boldsymbol{CB} & 0 \end{bmatrix} \cdots\cdots(4.e)$$

$\boldsymbol{x}(t)$を求めるため，(4.a)の両辺に左から $\boldsymbol{O}^T$ を乗じ，それを $\boldsymbol{x}$(t)について解けば，

$$\boldsymbol{x}(t)=(\boldsymbol{O}^T\boldsymbol{O})^{-1}\boldsymbol{O}^T(\boldsymbol{Y}(t)-\boldsymbol{TU}(t)) \cdots\cdots(5)$$

となる。ただし，上式のようにできるのは，逆行列$(\boldsymbol{O}^{\mathrm{T}}\boldsymbol{O})^{-1}$が存在するときだけである。その逆行列が存在するための必要十分条件は，(4.c)の $\boldsymbol{O}$ は $n\times\ell$ 行 n 列だから，

$$\mathrm{rank}\boldsymbol{O}=n \cdots\cdots(6)$$

である。この条件式が成り立てば，(5)から $\boldsymbol{x}(t)$が求められ，その初期値 $\boldsymbol{x}_0$ も求まる。

なお，制御理論で知られているように，(6)は(1.a, b)のシステムが可観測であるための必要十分条件である。ここで，可観測とは，0〜t の間の入力 $\boldsymbol{u}(t)$と出力 $\boldsymbol{y}(t)$とから初期値 $\boldsymbol{x}_0$ を一意に定められることをいう。

ただし，上に示した方法は，(4.b)のように出力の微分を何回も行うとノイズを増幅するから実用的には使われない。

4.4 微分を避ける工夫(シミュレータ)

前節のアプローチは，(1.a, b)のシステムの出力側から逆にたどって状態変数を求めようとしたから，微分操作が必要になった。それでは，(1.a, b)のシステムの入力側から順にたどって状態変数を求めれば微分操作を避けられるのではないか，と考えられる。これが以下に示すシミュレータの発想である。（図 4.1 を参照のこと。）

まず，入力 $\boldsymbol{u}(t)$から状態変数 $\boldsymbol{x}(t)$を復元するため，(1.a)を参考にして次のモデルをつくる。

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{x}}(t)=\hat{\boldsymbol{A}}\hat{\boldsymbol{x}}(t)+\hat{\boldsymbol{B}}\boldsymbol{u}(t) \quad \cdots\cdots (7)$$

上付きハットは推定値を示す。初期値は分からないから次のよう適当ににおく。

$$\hat{\boldsymbol{x}}(0)=\hat{\boldsymbol{x}}_0 \quad \cdots\cdots (8)$$

もし $\hat{\boldsymbol{A}}=\boldsymbol{A}$ ，$\hat{\boldsymbol{B}}=\boldsymbol{B}$ ならば，(1.a)と(7)は全く同じシステムであり，入力も同じだから，$\hat{\boldsymbol{x}}(t)$ と $\boldsymbol{x}(t)$ とは同じになりそうである。このようにして状態変数を推定するものをシミュレータという。

しかし，シミュレータでは，(8)のように初期値を適当においたから，初期値の誤差が状態変数推定値の誤差がどのように影響するかを調べておく必要がある。それを調べるために，次のようにする。$\hat{\boldsymbol{A}}=\boldsymbol{A}$ ，$\hat{\boldsymbol{B}}=\boldsymbol{B}$ として(7)から(1.a)を引けば，

$$\frac{\mathrm{d}}{\mathrm{d}t}(\hat{\boldsymbol{x}}(t)-\boldsymbol{x}(t))=\boldsymbol{A}(\hat{\boldsymbol{x}}(t)-\boldsymbol{x}(t)) \quad \cdots\cdots (9)$$

となる。ここで，推定誤差 $\boldsymbol{e}(t)$を(10)で定義すれば，上式は(11)のように書ける。

$$\boldsymbol{e}(t)\equiv\hat{\boldsymbol{x}}(t)-\boldsymbol{x}(t) \quad \cdots\cdots (10)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{e}(t)=\boldsymbol{A}\boldsymbol{e}(t) \quad \cdots\cdots (11)$$

(10)の解は，

$$\boldsymbol{e}(t)=\varepsilon^{\boldsymbol{A}t}\boldsymbol{e}(0) \quad \cdots\cdots (12)$$

である。よって，状態変数の真値と推定値の関係は，(10)から次のようになる。

$$\boldsymbol{x}(t)=\hat{\boldsymbol{x}}(t)-\varepsilon^{\boldsymbol{A}t}(\hat{\boldsymbol{x}}(0)-\boldsymbol{x}(0)) \quad \cdots\cdots (13)$$

上式において，$\boldsymbol{A}$ が安定なら t が大きくなれば $\varepsilon^{\boldsymbol{A}t}\to 0$ となり，上式右辺第 2 項の初期誤差が消えて $\hat{\boldsymbol{x}}(t)\to\boldsymbol{x}(t)$となることが分かる。ただし，この収束の速さは $\boldsymbol{A}$ の固有値に依存し，シミュレータではその収束の速さを設計者が自由に決めることができない。なお，シミュレータは，図 4.1 において $\boldsymbol{K}=0$ としたものである。

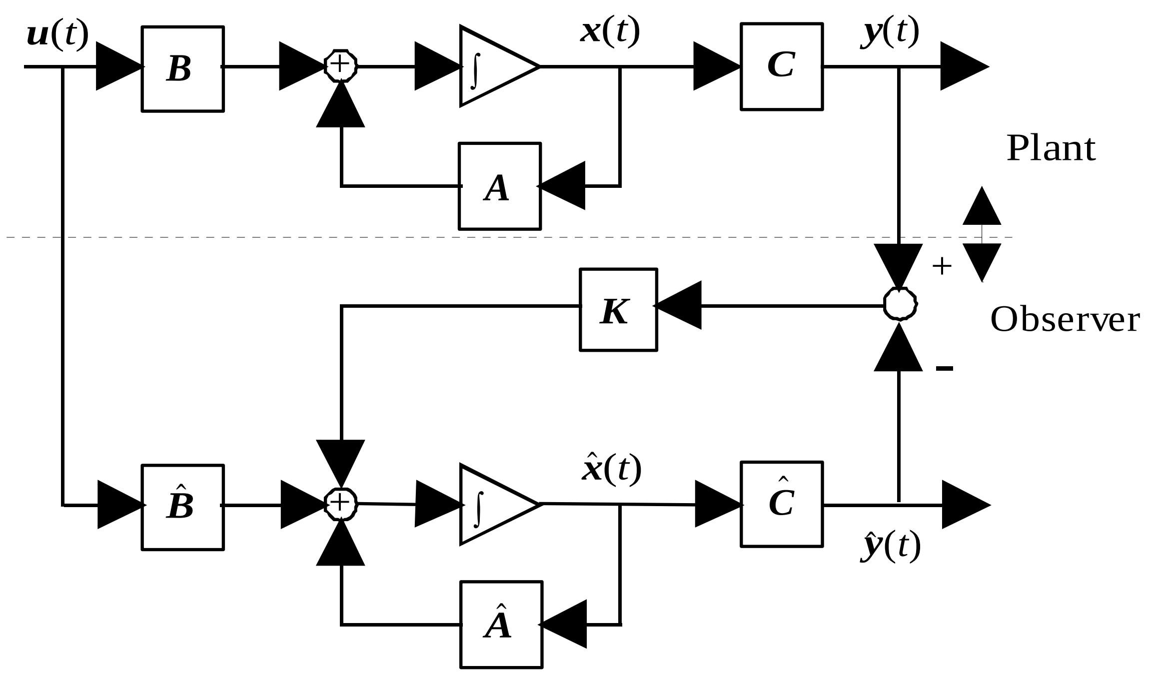

図 4.1 全状態オブザーバの構成

4.5　微分を避けて収束も速くする(全次元オブザーバ)

前節では入力 $\boldsymbol{u}(t)$だけを使ったが，まだ検出値 $\boldsymbol{y}(t)$は使っていない。実は，本節で示すように，検出値 $\boldsymbol{y}(t)$による修正項を付加すると状態変数の推定誤差の収束の速さを変えることができる。これが，オブザーバの発想である。

すなわち，(7)に検出値 $\boldsymbol{y}(t)$と推定値 $\hat{\boldsymbol{y}}(t)=\boldsymbol{C}\hat{\boldsymbol{x}}(t)$ の差による修正項を加え

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{x}}(t)=\hat{\boldsymbol{A}}\hat{\boldsymbol{x}}(t)+\hat{\boldsymbol{B}}\hat{\boldsymbol{u}}(t)+\boldsymbol{K}(\boldsymbol{y}(t)-\boldsymbol{C}\hat{\boldsymbol{x}}(t)) \quad \cdots\cdots (14)$$

とする。これを全状態オブザーバといい，その構造は図 4.1 のようになっている。

推定誤差の挙動を調べるため， $\hat{\boldsymbol{A}}=\boldsymbol{A}$， $\hat{\boldsymbol{B}}=\boldsymbol{B}$ として，(14)から(1.a)を引いて，(1.b)および(10)を使えば,

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{e}(t)=(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C})\boldsymbol{e}(t) \quad \cdots\cdots (15)<(20)$$

上式は，(11)と違い，誤差の収束を修正ゲイン $\boldsymbol{K}$ によって調整できることが分かる。

ここで，$\boldsymbol{K}$ によって $\boldsymbol{A}$-$\boldsymbol{KC}$ の固有値をどのくらい変えられるかについては，制御理論により次のことが知られている。すなわち，（$\boldsymbol{C}$, $\boldsymbol{A}$）が可観測であれば，$\boldsymbol{A}$-$\boldsymbol{KC}$ の固有値を任意の位置に配置でき，逆も成り立つ。

結局，（$\boldsymbol{C}$, $\boldsymbol{A}$）が可観測であれば，オブザーバにより微分操作を避け，かつ，初期値誤差の収束を任意に指定できることが分かった。

4.6　検出できないものだけ推定(低次元オブザーバ)

n 次の状態変数のうち m 個が出力 $\boldsymbol{y}(t)$として検出できるなら，$\boldsymbol{y}(t)$はそのまま使用し，検出できない残りの n-m 個の状態変数だけを復元してもよいと考えられる。これが，低次元オブザーバの発想である。

まず，状態変数 $\boldsymbol{x}(t)$を次のように分割して考える。

$$\hat{\boldsymbol{x}}(t)=\begin{bmatrix}\boldsymbol{y}(t)\\ \hat{\boldsymbol{z}}(t)\end{bmatrix}\quad \begin{matrix}(m)\text{th order}\\ (n-m)\text{th order}\end{matrix} \quad \cdots\cdots (16)$$

上のような分割を考慮して，(1.a)を次のように分割する。

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{y}(t)\\ \hat{\boldsymbol{z}}(t)\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}(t)\\ \hat{\boldsymbol{z}}(t)\end{bmatrix}+\begin{bmatrix}\boldsymbol{B}_1\\ \boldsymbol{B}_2\end{bmatrix}\boldsymbol{u}(t) \quad \cdots\cdots (17)$$

上式の 2 行目で $\boldsymbol{z}(t)$の予測をして，1 行目で誤差を修正するようにする。すなわち,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{z}}(t)=&\boldsymbol{A}_{22}\hat{\boldsymbol{z}}(t)+\boldsymbol{A}_{21}\boldsymbol{y}(t)+\boldsymbol{B}_2\boldsymbol{u}(t) && \text{estimation}\\ &+\boldsymbol{K}\left(\dot{\boldsymbol{y}}(t)-\boldsymbol{A}_{11}\boldsymbol{y}(t)-\boldsymbol{A}_{12}\hat{\boldsymbol{z}}(t)-\boldsymbol{B}_1\boldsymbol{u}(t)\right) && \text{correction}\end{aligned} \quad \cdots\cdots (18)$$

上式右辺の $\dot{\boldsymbol{y}}(t)$ は, 実際には微分しなくてもよい。それを示すため, 上式を次のように変形する。

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{z}}(t)=(\boldsymbol{A}_{22}-\boldsymbol{K}\boldsymbol{A}_{12})\hat{\boldsymbol{z}}(t)+(\boldsymbol{A}_{21}-\boldsymbol{K}\boldsymbol{A}_{11})\boldsymbol{y}(t)+(\boldsymbol{B}_2-\boldsymbol{K}\boldsymbol{B}_1)\boldsymbol{u}(t)+\boldsymbol{K}\dot{\boldsymbol{y}}(t) \quad \cdots\cdots (19)$$

上式に従って図 4.2 のようなブロック図を描いてみれば，右辺最終項の $\boldsymbol{y}(t)$は，実際に微分しなくてもよいことが分かる。

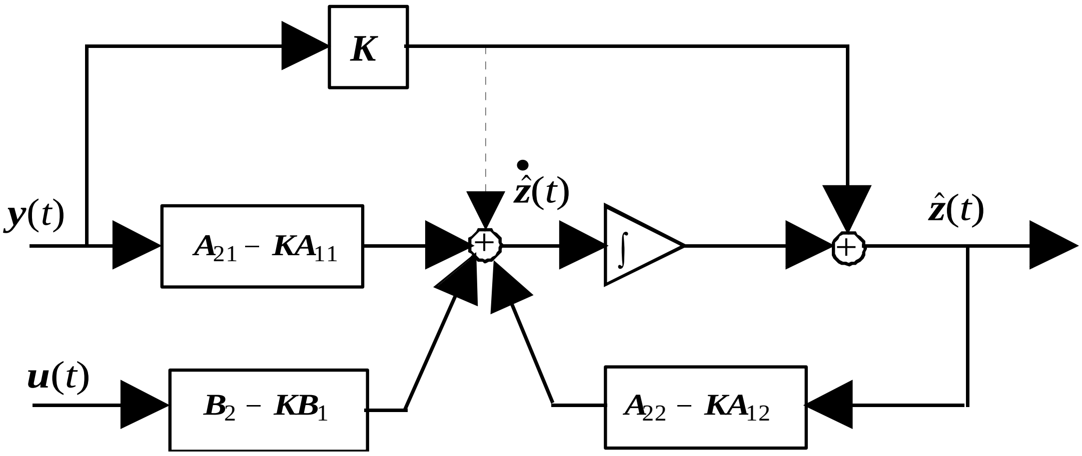

図4.2 低次元オブザーバの構成

4.7 オブザーバを併用した状態フィードバック(分離定理)

オブザーバで復元した状態変数を使えば，状態フィードバック系を構成できる。このとき，制御器の挙動とオブザーバの挙動とが，相互に絡み合って複雑になるようなことはないだろうか？このことについては，オブザーバの収束と状態フィードバックの収束を，それぞれ，独立に設計できることが，以下のようにして分かる。

まず，実際のシステムのパラメータ($\boldsymbol{C}, \boldsymbol{A}, \boldsymbol{B}$)とオブザーバに使うパラメータ($\hat{\boldsymbol{C}}, \hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}$)は，一致しているものと仮定でき，さらに，(1.a, b)で与えられるシステムは可制御，可観測であると仮定する。

次の状態フィードバックを考える。

$$\boldsymbol{u}(t) = \boldsymbol{F}\hat{\boldsymbol{x}}(t) + \boldsymbol{G}\boldsymbol{v}(t) \quad \cdots (20)$$

この式を，(1.a)と(14)に代入しすれば，それぞれ

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{BF}\hat{\boldsymbol{x}}(t) + \boldsymbol{BG}\boldsymbol{v}(t) \quad \cdots (21.a)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{x}}(t) = \boldsymbol{KC}\boldsymbol{x}(t) + (\boldsymbol{A} - \boldsymbol{KC})\hat{\boldsymbol{x}}(t) + \boldsymbol{BF}\hat{\boldsymbol{x}}(t) + \boldsymbol{BG}\boldsymbol{v}(t) \quad \cdots (21.b)$$

ここで，(21.a)は(10)を使って $\hat{\boldsymbol{x}}(t)$ を消去すれば，次式となる。

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = (\boldsymbol{A} + \boldsymbol{BF})\boldsymbol{x}(t) + \boldsymbol{BF}\boldsymbol{e}(t) + \boldsymbol{BG}\boldsymbol{v}(t) \quad \cdots (22.a)$$

一方，(21.b)から(21.a)を引いて，(10)を使って整理すれば，

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{e}(t) = (\boldsymbol{A} - \boldsymbol{KC})\boldsymbol{e}(t) \quad \cdots (22.b)$$

(22.a)と(22.b)をまとめて書くと，

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{x}(t)\\ \boldsymbol{e}(t)\end{bmatrix} = \begin{bmatrix}\boldsymbol{A} + \boldsymbol{BF} & \boldsymbol{BF}\\ 0 & \boldsymbol{A} - \boldsymbol{KC}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(t)\\ \boldsymbol{e}(t)\end{bmatrix} + \begin{bmatrix}\boldsymbol{BG}\\ 0\end{bmatrix}\boldsymbol{v}(t) \quad \cdots (23)$$

という拡大系の状態方程式が得られる。上式の右辺第一項の係数行列は上三角行列である。上三角行列の固有値は対角要素のブロック行列の固有値に等しいことが，線形代数で知られている。したがって，拡大系の固有値は， $\boldsymbol{A} + \boldsymbol{BF}$ および $\boldsymbol{A} - \boldsymbol{KC}$ の固有値に等しい。

ここで， $\boldsymbol{A} + \boldsymbol{BF}$ の固有値は(22.a)から状態フィードバック制御系の収束性を決める固有値であ

るが，状態フィードバックゲイン $\boldsymbol{F}$ で設定することができる。一方， $\boldsymbol{A}-\boldsymbol{KC}$ の固有値は(22.b)からオブザーバの収束性を決める固有値であるが，オブザーバの修正ゲイン $\boldsymbol{K}$ で設定することができる。このように，状態フィードバック制御系の収束性，および，オブザーバの収束性を，それぞれ，独立に設定することができる。ことを，**制御と観測の分離定理**という。

制御と観測の分離定理が成り立つと，制御器の設計とオブザーバの設計を別々に行ってよくなる。これは，制御系全体の設計という大きな問題を，制御器の設計とオブザーバの設計という小さな二つの問題に分割できることを意味する。一つの大きな問題より，二つの小さな問題のほうが取り扱いが易しい。

ただし，制御と観測の分離定理が成り立つのは，

① 制御対象のシステムが可制御・可観測であり，かつ,

② 実際のシステムのパラメータとオブザーバのパラメータが一致する

場合である。パラメータが不一致の場合には，フィードバック制御系の挙動とオブザーバの挙動が相互に干渉し，全体の挙動は大変複雑になる。

演習　パラメータ$(\boldsymbol{C}, \boldsymbol{A}, \boldsymbol{B})$と$(\hat{\boldsymbol{C}}, \hat{\boldsymbol{A}}, \hat{\boldsymbol{B}})$が等しくないとき，(23)はどうなるか？