UNIT- III

State space Analysis and the concepts of controllability and observability:

Conventional Methods such as root-locus and Frequency-response methods are useful for dealing with single-input-single output systems. and also they are applicable only to LTI systems.

The state-space methods for the analysis and synthesis of control systems are best suited for dealing with MIMO systems.

State space methods enable the engineer to include initial conditions in the design. This is very convenient and useful feature that is not possible in the conventional design Methods.

* It gives the Future behaviour of the system based on the present input and past history of the system.
* The past history of the system described by state variable. past history is nothing but the initial states or initial condition.
* The resistive circuit don't have any state variable because output is depends on Input and present value of the system.
* Inductors and capacitors are the state variables.
* No. of state variables can be determined by order of system.

Eg. $y^4 + 5y^3 + y + 6 = 0$

order = 4

Hence 4 state variables are required.

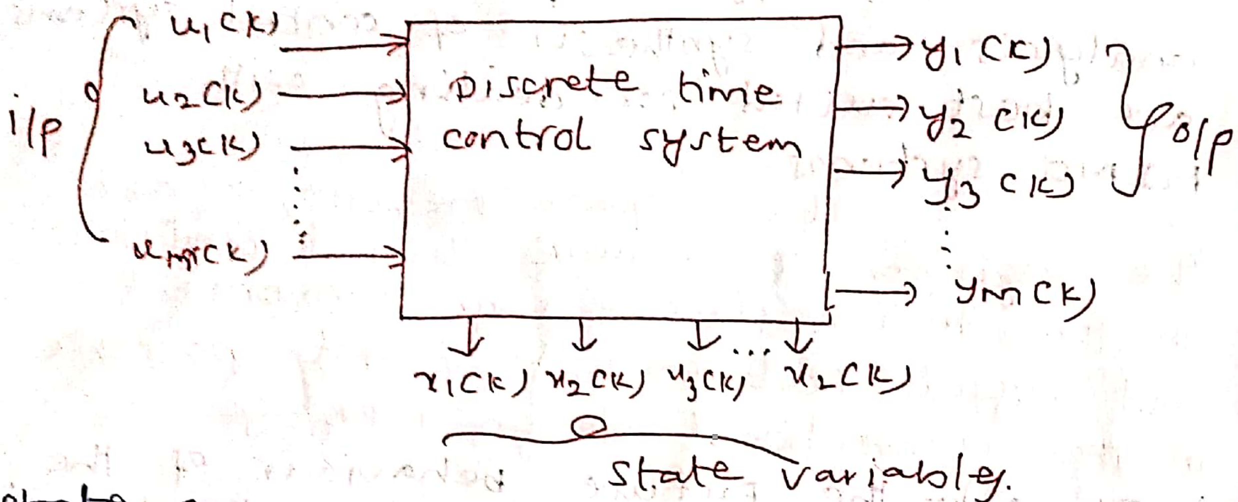
State: It completely determines the behavior of the system for any time $t \geq t_0$.

Scanned by CamScanner

State variable: Determine the state of the dynamic system.

State vector: It determines the system state $x(t)$ for any time $t \geq t_0$, once the state at $t = t_0$ is given and the input $u(t)$ for $t \geq t_0$ is specified.

State space: The n-dimensional space whose coordinate axes consist of the x_1 axis, x_2 axis ... x_n axis is called a state space.



State space equations:

For linear time varying continuous-time systems, the state equation and output equations are

$$x'(t) = A(t)\,x(t) + B(t)\,u(t)$$
$$y(t) = C(t)\,x(t) + D(t)\,u(t)$$

If the system is time invariant, then the last two equations (above equations) can be written as.

$$x'(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

Similarly for linear time-varying discrete time systems, the state equation and output equation can be written as.

$$x(k+1) = G(k)\,x(k) + H(k)\,u(k)$$
$$y(k) = C(k)\,x(k) + D(k)\,u(k)$$

~~Input~~ state equation. — output equation.

Scanned by CamScanner

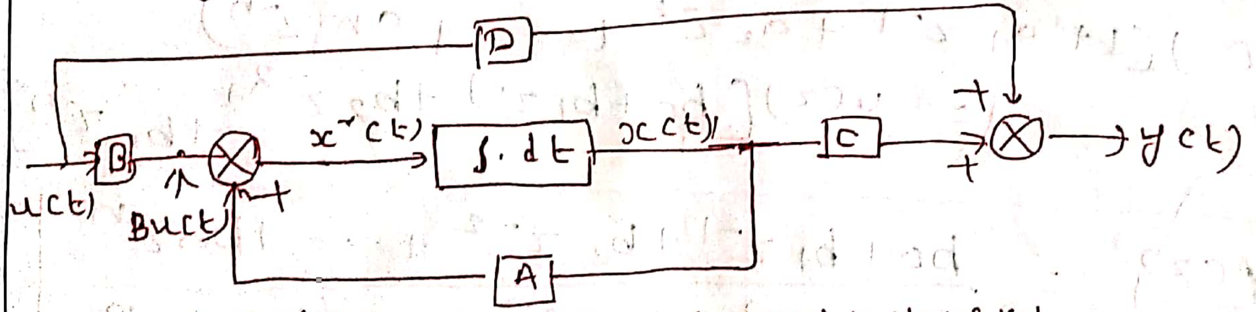
$x(k)$ = state vector (n-vector)
$y(k)$ = output vector (m-vector)
$u(k)$ = Input vector (r-vector)
$G(k)$ = state matrix (n×n matrix)
$H(k)$ = Input matrix (n×r matrix)
$C(k)$ = output matrix (m×n matrix)
$D(k)$ = direct transmission matrix (m×r matrix)

The variable k represents time varying.
If the system is time invariant, then the equations becomes:

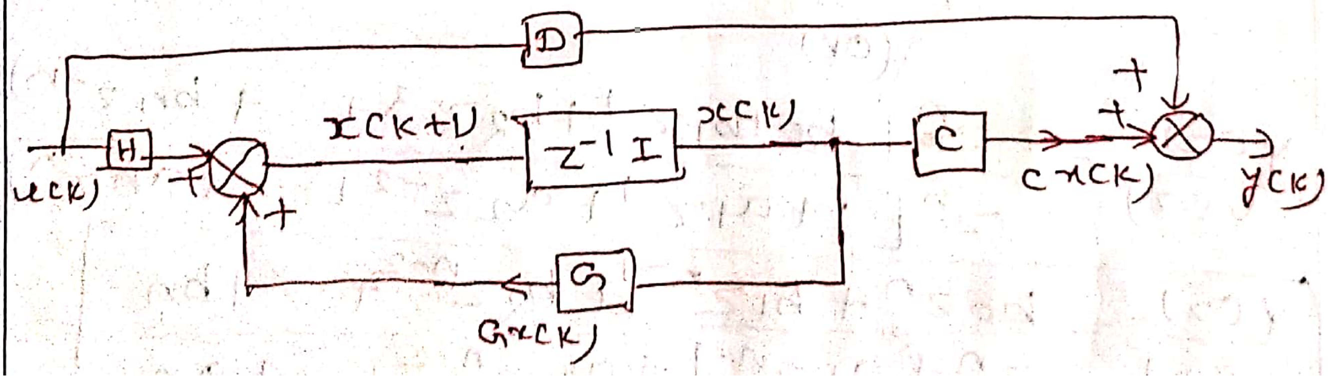
$$x(k+1) = Gx(k) + Hu(k)$$
$$y(k) = Cx(k) + Du(k)$$ *

Block Diagram Representation of discrete-time control system defined by equations.

For: $\dot{x}(t) = Ax(t) + Bu(t)$
$y(t) = Cx(t) + Du(t)$



For Discrete: $x(k+1) = Gx(k) + Hu(k)$
$y(k) = Cx(k) + Du(k)$



Scanned by CamScanner

State-space Representations of Discrete-Time Systems:

Canonical forms for Discrete-Time state space equations:

Consider the discrete-time system described by.

$$y(k) + a_1 y(k-1) + a_2 y(k-2) + \dots + a_n y(k-n)$$
$$= b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) + \dots + b_n u(k-n) \quad \text{--- (1)}$$

$u(k)$ = Input
$y(k)$ = output
k = sampling instant.
a and b are coefficients.

Apply z-transform for eqn (1):

$$Y(z) + a_1 z^{-1} Y(z) + a_2 z^{-2} Y(z) + \dots + a_n z^{-n} Y(z)$$
$$= b_0 U(z) + b_1 z^{-1} U(z) + b_2 z^{-2} U(z) + \dots + b_n z^{-n} U(z)$$

$$Y(z)(1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n})$$
$$= U(z)[b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}]$$

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}$$

↳ pulse transfer Function.

(or)

$$\frac{Y(z)}{U(z)} = \frac{z^n [b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}]}{z^n [1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}]}$$

$$\frac{Y(z)}{U(z)} = \frac{b_0 z^n + b_1 z^{n-1} + b_2 z^{n-2} + \dots + b_n}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \dots + a_n}$$

Scanned by CamScanner

There are many ways to realize state-space representations for DTS.

1) controllable canonical form.
2) observable canonical form
3) Diagonal canonical form.
4) Jordan canonical form.

<u>controllable canonical form:</u> It can be derived by direct programming method.

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

controllable canonical form can be represented by:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \dots & & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & & -a_1 \end{bmatrix}}^{G} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{(H)} u(k) \longrightarrow (1)$$

$$y(k) = C x(k) + D u(k)$$

$$y(k) = \begin{bmatrix} b_n - a_n b_0 & : & b_{n-1} - a_{n-1} b_0 & : & \dots & : & b_1 - a_1 b_0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + b_0 u(k) \longrightarrow (2)$$

equation (1) is the state equation.

equation (2) is the output equation.

Scanned by CamScanner

problem: consider the following system

$$\frac{Y(z)}{U(z)} = \frac{z+1}{z^2+1.3z+0.4}$$

obtain controllable canonical form.

sol:

$$\frac{Y(z)}{U(z)} = \frac{b_0+b_1 z^{-1}+b_2 z^{-2}+\cdots+b_n z^{-n}}{1+a_1 z^{-1}+\cdots+a_n z^{-n}} \rightarrow ①$$

controllable canonical form is:

state equation:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k) \rightarrow ②$$

The given problem: $\frac{Y(z)}{U(z)} = \frac{z+1}{z^2+1.3z+0.4}$

order of the system is ② hence we need two state equations. $x_1(k+1)$ and $x_2(k+2)$

The equation ② becomes:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_n & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

write $\frac{Y(z)}{U(z)}$ as equation ①

Divide N.r and D.r by z^2

$$\frac{Y(z)}{U(z)} = \frac{\frac{z+1}{z^2}}{\frac{z^2+1.3z+0.4}{z^2}} = \frac{z^{-1}+z^{-2}}{1+1.3z^{-1}+0.4z^{-2}}$$

$$\frac{Y(z)}{U(z)} = \frac{z^{-1}+z^{-2}}{1+1.3z^{-1}+0.4z^{-2}} \rightarrow ③$$

compare equations ① and ③

$b_0 = 0 \quad b_1 = 1, \quad b_2 = 1$

$a_1 = 1.3 \quad a_2 = 0.4$

Scanned by CamScanner

n = 2 (state variables)

so that state equation:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

[G] [H]

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.4 & -1.3 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

matrix $G = \begin{bmatrix} 0 & 1 \\ -0.4 & -1.3 \end{bmatrix}$, $H = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

For output equation:

$$y(k) = [b_n - a_n b_0 : b_{n-1} - a_{n-1} b_0 : \cdots b_1 - a_1 b_0] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + b_0 u(k)$$

$$y(k) = [b_2 - a_2 b_0 \quad b_1 - a_1 b_0] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + b_0 u(k)$$

~~y(k) = [1-(1.3×0) 1-(1-3~~

$$y(k) = [1-(0.4\times 0) \quad 1-(1.3\times 0)] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + 0 u(k)$$

$$y(k) = [1 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + 0$$

$$= [1 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

$C = [1 \quad 1]$ $D = [0]$

observable canonical form:

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & -a_n \\ 1 & 0 & \cdots & 0 & 0 & -a_{n-1} \\ 0 & 1 & \cdots & 0 & 0 & -a_{n-2} \\ \vdots \\ 0 & 0 & \cdots & 1 & 0 & -a_2 \\ 0 & 0 & \cdots & 0 & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} b_n - a_n b_0 \\ b_{n-1} - a_{n-1} b_0 \\ \vdots \\ b_2 - a_2 b_0 \\ b_1 - a_1 b_0 \end{bmatrix}$$

Scanned by CamScanner

$$y(k) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + b_0 u(k)$$

Problem: $\frac{Y(z)}{U(z)} = \frac{z+1}{z^2+1.3z+0.4}$

Sol: $\frac{Y(z)}{U(z)} = \frac{z^{-1}+z^{-2}}{1+1.3z^{-1}+0.4z^{-2}}$

$$\frac{Y(z)}{U(z)} = \frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_nz^{-n}}{1+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}}$$

compare: $b_0=0$, $b_1=1$, $b_2=1$

$a_1=1.3$, $a_2=0.4$.

observable canonical form:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & -a_2 \\ 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} b_2-a_2b_0 \\ b_1-a_1b_0 \end{bmatrix} u(k)$$

$$= \begin{bmatrix} 0 & -0.4 \\ 1 & -1.3 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1-(0.4\times 0) \\ 1-(1.3\times 0) \end{bmatrix} u(k)$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & -0.4 \\ 1 & -1.3 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + b_0 u(k)$$

$$= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + [0] u(k)$$

$$\boxed{y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}}$$

Diagonal canonical form:

If the poles of the pulse transfer function given by:

$$\frac{Y(z)}{U(z)} = \frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_nz^{-n}}{1+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}}$$

are all distinct, then the state-space representation in Diagonal form is

Scanned by CamScanner

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+2) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} p_1 & 0 & \cdots & \cdots & 0 \\ 0 & p_2 & \cdots & \cdots & 0 \\ 0 & 0 & p_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & & p_n \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + b_0 u[k]$$

transfer

Scanned by CamScanner

State variable feedback

We have previously discussed systems described by the linear state-space equations

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

with $x(t) \in \mathbb{R}^n$ the internal state, $u(t) \in \mathbb{R}^m$ the control input, and $y(t) \in \mathbb{R}^p$ the measured output.

Matrix A is the system or plant matrix, B is the control input matrix, C is the output or measurement matrix, and D is the direct feed matrix.

We learned that the open-loop poles are given by the roots of

$$\Delta(s) = |sI - A|$$

and the open-loop transfer function is given by

$$H(s) = C(sI - A)^{-1}B + D\,.$$

In previous lectures we discussed how feedback controllers (which could be of PID-type) can be designed for the system given in transfer function form, using the Root-Locus method. This method allowed calculation of the parameters of a PID-type controller such that the poles of the closed loop system are placed in desired positions in the s-plane corresponding to the desired closed loop performances.
It has to be mentioned here that this is a general method which could be used in conjunction with any sort of controller having a transfer function representation.

In the following we will look a one method which allows derivation of feedback controllers using the state description (A,B,C,D) of a system.

The STATE-VARIABLE FEEDBACK (SVFB) control law is a basic control scheme which is based on the assumption that ALL the states of the system can be measured as outputs.
Thus we would like to determine a controller in the form of a gain matrix *K* which calculates the control input to be sent to the system based on the measured state.

$$u = -Kx + v$$

For the moment we will consider that the signal *v* is zero. This signal plays the role of the reference for the closed loop system, and choosing it as 0 only means that the desired performance for the system is one of stabilization (i.e. the states of the system need to be transferred to the equilibrium point x=0).

The feedback matrix *K* is $m \times n$ so that there are now mn control loops.

Assuming SVFB, with $u = -Kx$ the closed-loop system is given as

$$\begin{cases} \dot{x} = (A - BK)x = A_c x \\ y = (C - DK)x = C_c x \end{cases}$$

where A_c is the CLOSED-LOOP SYSTEM MATRIX and C_c is the closed-loop output matrix.

The closed-loop poles are given by the roots of

$$\Delta_c(s) = |sI - A_c| = |sI - (A - BK)|$$

One asks now the very good question:
Does such matrix *K*, which stabilizes the system states, exist?
Or, in other words, does this problem of finding the matrix *K* which stabilizes the system dynamics have a solution?

To answer this let us remember the notion of **reachability** of a state variable system:
The system (*A*,*B*,*C*) is called **reachable** if the control input can be selected to drive any initial state to any desired final state at some final time.

Thus we can solve the stabilization problem if and only if the system is reachable, since only then the controller would be able to drive any initial state of the system to the equilibrium point.

The following provides a **test for reachability**.
A system is reachable if and only if the **reachability matrix**
$U = [B \quad AB \quad \ldots \quad A^{n-1}B]$
has full rank *n*.

We also remember that reachability is equivalent to the absence of input-decoupling zeros.

At this point, provided that the system is reachable, we would like to proceed to deriving a methodology which allows the calculation of a matrix *K* which places the poles of the closed loop system in desired locations given by the desired performances of the closed loop system.

SVFB Pole Placement with Ackermann's Formula

In the case of SVFB the output y(t) plays no role. This means that only matrices A and B will be important in SVFB.
We would like to choose the feedback gain K so that the closed-loop characteristic polynomial $\Delta_c(s) = |sI - A_c| = |sI - (A - BK)|$ has prescribed roots (which the engineer determines based on the given desired performances). This is called the POLE-PLACEMENT problem.

An important theorem says that the poles may be placed arbitrarily as desired iff (A,B) is reachable.

If the system is reachable, there are many techniques to find a suitable K that guarantees stability and/or places the poles.
One technique that works for the single input case, i.e. m=1, is ACKERMANN'S FORMULA

$$K = e_n U^{-1} \Delta_D(A),$$

where $e_n = [0 \quad 0 \quad \cdots \quad 0 \quad 1]$ is the last row of the $n \times n$ identity matrix, and $\Delta_D(s)$ is the DESIRED characteristic polynomial for the closed loop system. Note that $\Delta_D(A)$ is a MATRIX POLYNOMIAL.

Derivation of Ackermann's formula

We would like to choose the feedback gain K so that the closed-loop characteristic polynomial $\Delta_c(s) = |sI - A_c| = |sI - (A - BK)|$ has prescribed roots, i.e. $\Delta_c(s) = \Delta_D(s)$. Thus we would like that the characteristic polynomial of the closed loop system matrix is $\Delta_D(s)$ (again we mention here that this polynomial is specified by the engineer as it gives the desired locations for the poles of the closed loop system).

We remember Cayley-Hamilton theorem which says that a matrix satisfies its own characteristic polynomial $\Delta_D(A_c)$=0. Starting from this condition Ackermann determined his famous formula.

As a nice exercise we develop in the following the derivation of Ackermann's formula for the case of a system with 3 states.

Say that the desired characteristic polynomial for the closed loop system (which has as solutions the desired values for the poles of the closed loop system) is given by

$$\Delta_D(s) = s^3 + q_2 s^2 + q_1 s + q_0 .$$

Then the closed loop system matrix $A_c = A - BK$ must satisfy $\Delta_D(A_c) = 0$.

Thus we can write

$$\Delta_D(A_c) = A_c^3 + q_2 A_c^2 + q_1 A_c + q_0 I_2$$

$$\begin{aligned} A_c^2 = (A - BK)^2 &= A(A - BK) - BK(A - BK) \\ &= A^2 - AB\,K - B\,K(A - BK) \\ A_c^3 = (A - BK)^3 &= [A^2 - ABK - BK(A - BK)](A - BK) \\ &= A^3 - A^2 B\,K - AB\,K(A - BK) - B\,K(A - BK)^2 \end{aligned}$$

$$\begin{aligned} \Delta_D(A_c) = {} & A^3 - A^2 B\,K - AB\,K(A - BK) - B\,K(A - BK)^2 \\ & + q_2[A^2 - AB\,K - B\,K(A - BK)] \\ & + q_1(A - BK) \\ & + q_0 I_2 \end{aligned}$$

We can rewrite and arrange to obtain

$$\begin{aligned} \Delta_D(A_c) = {} & A^3 + q_2 A^2 + q_1 A + q_0 I_2 \\ & - A^2 B\,K \quad - AB\,K(A - BK) - B\,K(A - BK)^2 \\ & \qquad\qquad - AB\,q_2 K \qquad - B\,q_2 K(A - BK) \\ & \qquad\qquad\qquad\qquad\qquad - B\,q_1 K \end{aligned}$$

But $A^3 + q_2 A^2 + q_1 A + q_0 I_2 = \Delta_D(A)$

Thus we can write

$$\Delta_D(A_c) = \Delta_D(A) - \begin{bmatrix} B & AB & A^2B \end{bmatrix} \begin{bmatrix} q_1K + q_2K(A-BK) + K(A-BK)^2 \\ q_2K + K(A-BK) \\ K \end{bmatrix}$$

Remember that we started with the assumption that $\Delta_D(A_c) = 0$ (i.e. the controller that we are looking for places the poles of the system in the positions given by the roots of the equation $\Delta_D(s) = 0$) then we get

$$\Delta_D(A) - \begin{bmatrix} B & AB & A^2B \end{bmatrix} \begin{bmatrix} q_1K + q_2K(A-BK) + K(A-BK)^2 \\ q_2K + K(A-BK) \\ K \end{bmatrix} = 0$$

which is

$$\Delta_D(A) = \begin{bmatrix} B & AB & A^2B \end{bmatrix} \begin{bmatrix} q_1K + q_2K(A-BK) + K(A-BK)^2 \\ q_2K + K(A-BK) \\ K \end{bmatrix}$$

We also see that

$\begin{bmatrix} B & AB & A^2B \end{bmatrix}$ is the controllability matrix $U = \begin{bmatrix} B & AB & A^2B \end{bmatrix}$ and we can write

$$\Delta_D(A) = U \begin{bmatrix} q_1K + q_2K(A-BK) + K(A-BK)^2 \\ q_2K + K(A-BK) \\ K \end{bmatrix}$$

Now, provided that the system is controllable (i.e. U has an inverse) we can write

$$U^{-1}\Delta_D(A) = \begin{bmatrix} q_1K + q_2K(A-BK) + K(A-BK)^2 \\ q_2K + K(A-BK) \\ K \end{bmatrix}.$$

Here we notice that the matrix $U^{-1}\Delta_D(A)$ has on the last line exactly the controller that we were set to determine. Thus determining the controller gain K reduces to simply extracting the last line of this matrix:

$$K = [0 \quad 0 \quad 1]U^{-1}\Delta_D(A).$$

And this is exactly Ackermann's formula for the case of three states.

We can easily extend this to the case of *n* states and see that

$$K = [0 \quad 0 \quad \ldots \quad 1]U^{-1}\Delta_D(A)$$

where $U = [B \quad AB \quad \ldots \quad A^{n-1}B]$.

Example 1

The angle subsystem of the inverted pendulum is given for some specific values of parameters by

$$\dot{x} = Ax + Bu = \begin{bmatrix} 0 & 1 \\ 9 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -2 \end{bmatrix} u$$

where the state is $x = [\theta \quad \dot{\theta}]$ (angle and angular velocity).

The design specifications are to design a state variable feedback gain K which will give a closed-loop POV of 4% with a settling time of τ_s= 1 sec. This will make the rod of the inverted pendulum balance upright.

The open-loop poles are given by the roots of

$$\Delta(s) = |sI - A| = \begin{vmatrix} s & -1 \\ -9 & s \end{vmatrix} = s^2 - 9 = (s+3)(s-3),$$

which are s=-3, s=3. The system is unstable, with natural modes e^{-3t}, e^{3t}.

The design specifications allow one to compute the desired closed-loop poles. In fact, since

$$\tau_s \approx 5\tau = 5/\alpha \text{ and } POV = 100e^{-\pi\zeta/\sqrt{1-\zeta^2}}$$

one may find the required real-part and damping ratio of the closed loop poles to be $\alpha = 5$, $\zeta \approx 1/\sqrt{2} = 0.707$. Thus, the natural frequency is $\omega_n = \alpha/\zeta$ = 7.072 so that the desired characteristic polynomial is

$$\Delta_D = s^2 + 2\alpha s + \omega_n^{\ 2} = s^2+10s+50.$$

To use Ackermann's formula, one first verifies reachability by computing the reachability matrix

$$U = [B \quad AB] = \begin{bmatrix} 0 & -2 \\ -2 & 0 \end{bmatrix}$$

and checking that it is indeed nonsingular.
Computing the quantities needed for Ackermann's formula now yields

$$U^{-1} = \begin{bmatrix} 0 & -0.5 \\ -0.5 & 0 \end{bmatrix}$$

$$\Delta_D(A) = A^2 + 10A + 50I = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix} + \begin{bmatrix} 0 & 10 \\ 90 & 0 \end{bmatrix} + \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix} = \begin{bmatrix} 59 & 10 \\ 90 & 59 \end{bmatrix}.$$

Substituting now into Ackermann's formula yields the required SVFB of

$$K = e_n U^{-1} \Delta_D(A) = [0 \quad 1] \begin{bmatrix} 0 & -0.5 \\ -0.5 & 0 \end{bmatrix} \begin{bmatrix} 59 & 10 \\ 90 & 59 \end{bmatrix} = [-29.5 \quad -5].$$

This solves the problem.

To check the design, one should compute the actual closed-loop poles using

$$A_c = A - BK = \begin{bmatrix} 0 & 1 \\ 9 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ -2 \end{bmatrix} [-29.5 \quad -5] = \begin{bmatrix} 0 & 1 \\ -50 & -10 \end{bmatrix}$$

$$\Delta_c(s) = |sI - A_c| = \begin{vmatrix} s & -1 \\ 50 & s+10 \end{vmatrix} = s^2 + 10s + 50 .$$

This is indeed equal to $\Delta_D(s)$.

Note that for this problem where the meaning of the state variables is angle and angular velocity one may write the P SVFB as

$$u = -Kx = -\begin{bmatrix} -29.5 & -5 \end{bmatrix} x = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = k_1\theta + k_2\dot{\theta} ,$$

so that in effect a proportional-plus-derivative (PD) control is produced, with the proportional gain given by k_1 and the derivative gain by k_2.
In this case the output of the system is the angle which is desired to become 0 (i.e. the pendulum should be in the upright position).

One now sees that this method allows calculation of both the proportional and the derivative gains at the same time, which allows placing the poles of the system in any location in the s-plane in **a single design step**. Thus state variable feedback design offers a major advantage compared with a root locus method of design.
But do not forget that this method requires measurement of **all** the system states.

One now asks another very good question:

What do we do in the case in which we only have available a transfer function model for the system? Can we obtain a state space model which will allow us to use this nice method of design?

The answer is yes.
Even more, when going from transfer function model to state space model of a system, one can obtain an infinite number of representations in the state-space for the same single transfer function.

In what follows we will look at three popular methods of determining a state space representation for a given transfer function.

Realization and canonical forms

The problem of finding a state variable or block diagram representation given a prescribed transfer function is called the ***realization problem***.

A transfer function can be realized as a block diagram in series form or parallel form. We now introduce two series forms that are very convenient for solving the block diagram realization problem for single-input/single-output (SISO) systems. We will then introduce a parallel form realization.

1. Block Diagram realization of a transfer function – series forms

Consider for example a general description of a third-order transfer function

$$G(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

For realization, it is important to ensure that highest order term in the denominator has a coefficient of 1. If this is not true then divide the numerator and denominator by this coefficient to put the transfer function in the desired form.

The transfer function must also have relative degree of 1 or more. If the relative degree is zero (e.g. same power of *s* in the numerator as the denominator), then divide the denominator by the numerator in one step of long division to write *H(s)* as a constant term plus a term whose relative degree is at least one. The constant term is a direct feedthrough term, and the procedures below may be carried out to realize the remainder term.

A transfer function is said to be *proper* if its relative degree is greater than or equal to zero, and *strictly proper* if the relative degree is greater than or equal to one.

We use a third-order system to illustrate the approach, which works for any *n*-th order rational, monic, strictly proper transfer function.

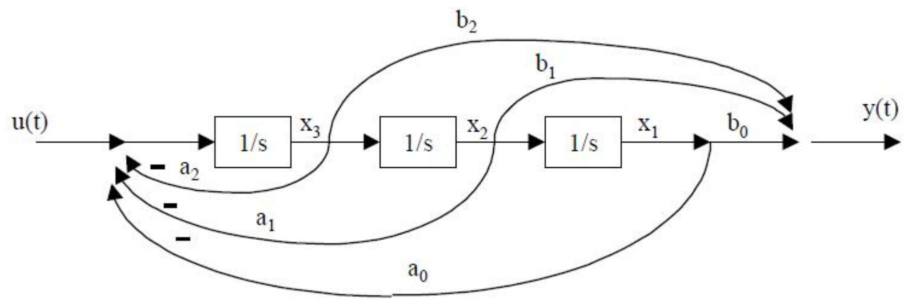
To find a BD realization of *H(s)*, divide by the highest power of *s* to obtain

$$G(s) = \frac{b_2 s^{-1} + b_1 s^{-2} + b_0 s^{-3}}{1 + a_2 s^{-1} + a_1 s^{-2} + a_0 s^{-3}} = \frac{b_2 s^{-1} + b_1 s^{-2} + b_0 s^{-3}}{1 - (-a_2 s^{-1} - a_1 s^{-2} - a_0 s^{-3})}$$

Now think of Mason's Formula. To draw a BD we can use three feedforward paths and three loops if we select the correct transmissions and loop structure.

We give two series forms that have a convenient structure for realizing SISO systems. Note particularly that Mason's Formula is very easy to use if there are no disjoint loops, and all loops touch all feedforward paths. Then, the determinant *D(s)* is simply 1 minus the sum of the loop gains, and all cofactors are equal to one.

1.A. Reachable Canonical Form (RCF)



Reachable Canonical Form

Note that all loops and all feedforward paths have the left-hand integrator in common, so all cofactors are equal to 1 and the determinant has no higher-order terms.
Applying Mason's Formula to this BD gives the transfer function that we considered.

Each integrator output is labeled as a state.
The rule used in this course for labeling states will be:
Label the states from right to left, from top to bottom.

We will see some examples of this to clarify it.
With the states labeled as shown, one may write down directly the state equations

$$\begin{cases} \dot{x}_3 = -a_2 x_3 - a_1 x_2 - a_0 x_1 + u \\ \dot{x}_2 = x_3 \\ \dot{x}_1 = x_2 \\ y = b_2 x_3 + b_1 x_2 + b_0 x_1 \end{cases}$$

which can now be arranged in the nice form

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y = \begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{cases}$$

As an exercise, one may find the transfer function

$$G(s) = C(sI - A)^{-1} B + D$$

and verify that it is the same as the one we started with.

This development gives a very easy way to realize SISO system in state variable form. One notes that it is easy to write down the state space representation directly from the transfer function without having to draw the block diagram. In fact, simply take the denominator of *H(s)*, turn the coefficients backwards, make them negative, and place them into the bottom row of the *A* matrix. Take the coefficients of the numerator, turn them backwards, and place them into the *C* matrix.
The *A* matrix in (3) is known as a *bottom companion matrix* for the characteristic polynomial

The superdiagonal 1's in *A* and the lower 1 in *B* mean simply that the three integrators are connected in series. Look at:

$$[A \quad B] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a_0 & -a_1 & -a_2 & 1 \end{bmatrix}$$

Example 2. Realize the given transfer function as reachable state variable system

$$G(s) = \frac{s^2 + 2s - 1}{s^3 + 2s^2 + 3s + 4}$$

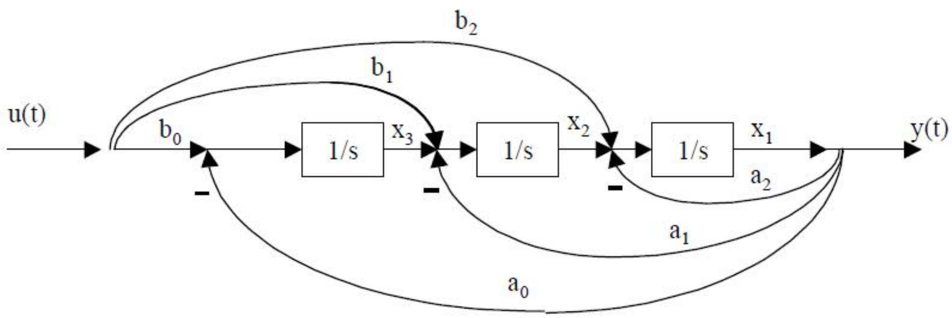
The SV equations are directly written down as

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y = \begin{bmatrix} -1 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{cases}$$

Now one may analyze the system including simulation, finding output given an input and initial conditions, etc.

1.B. Observable Canonical Form (OCF)

A BD satisfying this condition is drawn below.



Observable Canonical Form

Note that all loops and all feed forward paths have the right-hand integrator in common, so all cofactors are equal to 1 and the determinant has no higher-order terms.
Applying Mason's Formula to this BD gives the transfer function that we started with.
With the states labeled from right to left as shown, one may write down directly the state equations

$$\begin{cases} \dot{x}_1 = -a_2 x_1 + x_2 + b_2 u \\ \dot{x}_2 = -a_1 x_1 + x_3 + b_1 u \\ \dot{x}_3 = -a_0 x_1 + b_0 u \\ y = x_1 \end{cases}$$

which can now be arranged in the nice form

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{cases}$$

As an exercise, one may find the transfer function

$$G(s) = C(sI - A)^{-1} B + D$$

and verify that it is the same as the one we started with.

Again, one notes that it is easy to write down the state space description directly from the transfer function, without having to draw the block diagram. In fact, simply take the denominator of *H(s)*, stack the coefficients on end, make them negative, and place them into the first column of the *A* matrix. Take the coefficients of the numerator, stack them on end, and place them into the *B* matrix.

Note that this OCF state-space form is not the same as RCF, though both have the same transfer function.
In fact, RCF and OCF are related by a *state-space transformation,* which we shall not discuss in this course (it is discussed in Linear Systems). In fact the states of one representation can be written as a linear combination of the states of the other representation.

The A matrix in this case is known as a *left companion matrix* for the characteristic polynomial.

The superdiagonal 1's in A and the left-hand 1 in C mean simply that the three integrators are connected in series.

Example 3. Realize Transfer Function as OCF SV System

Let there be given the same transfer function for the system

$$G(s) = \frac{s^2 + 2s - 1}{s^3 + 2s^2 + 3s + 4}$$

The state variable equations are directly written down as

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -2 & 1 & 0 \\ -3 & 0 & 1 \\ -4 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{cases}$$

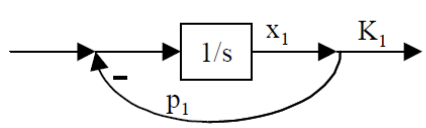
2. Block diagram realization of transfer functions in parallel form

To realize a system in parallel form, one performs a partial fraction expansion of the transfer function to obtain

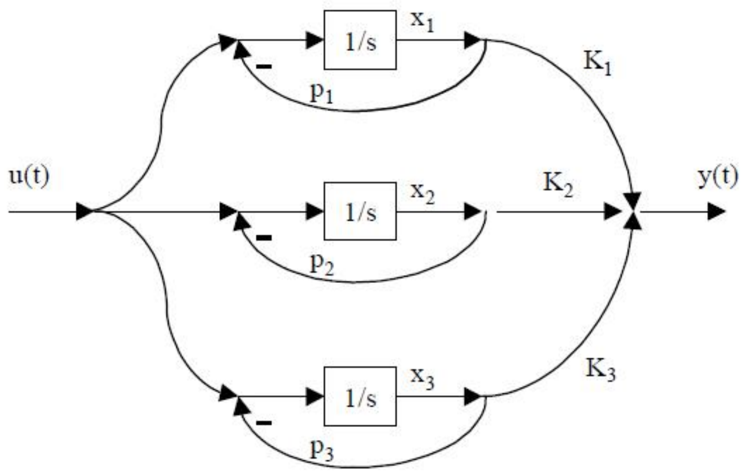
$$G(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} = \frac{k_1}{s - p_1} + \frac{k_2}{s - p_2} + \frac{k_3}{s - p_3}$$

where the poles are at p_1, p_2, p_3 and the residues are k_1, k_2, k_3.

Now note that a single term of this form can be realized using the simple block diagram shown in the next figure.



The complete transfer function with three parallel paths can be realized as shown in the next figure.



This realization is known as parallel form. If there are repeated poles, then the transfer function has higher-order poles in the partial fraction expansion. In this event, some parallel paths will contain multiple integrators.

A system which has a PFE with no higher-order poles is called *simple*.
The parallel form is known as *Jordan Normal Form* in mathematics. The case of higher-order pole factors, corresponding to multiple integrators in some paths, corresponds to what is known as *eigenvector chains* in those paths.

With the states labeled from top to bottom as shown in the figure, one may write down directly the state equations
With the states labeled from right to left as shown, one may write down directly the state equations

$$\begin{cases} \dot{x}_1 = -p_1 x_1 + u \\ \dot{x}_2 = -p_2 x_2 + u \\ \dot{x}_3 = -p_3 x_3 + u \\ y = k_1 x_1 + k_2 x_2 + k_3 x_3 \end{cases}$$

which can now be arranged in the nice form

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -p_1 & 0 & 0 \\ 0 & -p_2 & 0 \\ 0 & 0 & -p_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u \\ y = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{cases}$$

The residues can be placed on the input paths in the figure above. In fact, one can split the residues between input and output paths and thus one can have

$$\begin{cases}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}-p_1 & 0 & 0\\ 0 & -p_2 & 0\\ 0 & 0 & -p_3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}b_1\\ b_2\\ b_3\end{bmatrix}u\\ y=\begin{bmatrix}c_1 & c_2 & c_3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}\end{cases}$$

where

$$k_i = c_i b_i \text{ ,} i=\overline{1,3}$$

Note that this parallel state-space form is not the same as RCF or OCF, though all three have the same transfer function. RCF, OCF, and the Jordan form are related by *state-space transformations,* which we shall not discuss in this course.

Example 4. Determine state feedback controllers using Ackermann's formula for the system $G(s)=\dfrac{s^2+2s-1}{s^3+2s^2+3s+4}$ considering the reachable canonical form and observable canonical form realizations, such that the poles of the closed loop system are the solutions of -5 + i, -5-i and -5.

The poles of the system to be controlled are -1.6506, -0.1747 + 1.5469i and -0.1747 - 1.5469i.

a. reachable canonical form

$$\begin{cases}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -4 & -3 & -2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}u\\ y=\begin{bmatrix}-1 & 2 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}\end{cases}$$

$$K_r=\begin{bmatrix}13 & -21.75 & -11.75\end{bmatrix}$$

b. observable canonical form

$$\begin{cases}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}-2 & 1 & 0\\ -3 & 0 & 1\\ -4 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}1\\ 2\\ -1\end{bmatrix}u\\ y=\begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}\end{cases}$$

$$K_c=\begin{bmatrix}2 & -0.25 & -11.5\end{bmatrix}$$

Notice that depending on the canonical form that was used, a different state feedback controller was determined. In order to make the state feedback control schemes work there are different states that need to be measured in each case.

The bad news is that often times the states of the system can not be measured directly from the system. And without the measurements of the states one can not implement the state feedback controller.

The good news is that we have a model of the system available. And this model could give the values of the states in response to the same input.

The bad news is that even if we have a model, to obtain the correct values for the state (at all times) we also need to know the exact values of the initial values of the states. And these values may not always be known.

For this reason a state estimation scheme needs to be developed such that the estimated state gets close to the real state of the system even in the case of a wrong initialization of the state of the model system. This way one can implement the state feedback controller using the values of the estimated state.