1. SS model $\dot{x}(t) = Ax + bu; y = Cx$;Laplace transform $\rightarrow sX(s) - x(0) = AX(s) + bU(s) \rightarrow X(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}bU(s)$Inverse Laplace transform$\rightarrow x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$;transfer function to time domain(look up table)

2. Discrete Sys$-x(k+1) = \Phi x(k) + \Gamma u(k); y = Cx; where \Phi = e^{Ah}; \Gamma = (\int_0^h e^{Av}dv)B$;Assume h = 1, to find e^A, we need to let f(t) = $e^{At}, then F(s) = L[f(t)] = (sI - A)^{-1}$, from the table do inverse L.T get f(t), let t = h = 1, to get $\phi = e^A = f(1).\Gamma = A^{-1}[e^{Ah} - I]B$
shift operator: qf(k)=f(k+1); $q^{-1}f(k) = f(k-1); z^2Y(z) \rightarrow y(k+2)$ For z domain, steady state gain is H(1). With H(z) = c(zI-Φ)$^{-1}\Gamma = \frac{Q(z)}{P(z)}$, the characteristic poly is P(z) = det(zI-Φ),which gives the poles z1, z2, if $|z1|\, and\, |z2|\, both < 1$, then stable,single z = 1 marginal stable, multiple $\rightarrow unstable$

3. Analysis of Discrete Time System$Discrete.SStoIOmodel : x(k+1) = \Phi x(k) + \Gamma u(k); y(k) = cx(k).Z.T : zX(z) - zx(0) = \Phi X(z) + \Gamma U(z); (zI - \Phi)X(z) = zx(0) + \Gamma U(z); \rightarrow X(z) = (zI - \Phi)^{-1}zx(0) + (zI - \Phi)^{-1}\Gamma U(z)$ and $Y(z) = cX(Z) = c(zI - \Phi)^{-1}zx(0) + c(zI - \Phi)^{-1}\Gamma U(z); T.F : H(z) = c(zI - \Phi)^{-1}\Gamma$
①$Stability : for \frac{1}{z^2+a_1z+a_2}$Using Jury's Stability Test yields $1 - a_2^2 > 0; \frac{(1-a_2)[(1+a_2)^2-a_1^2]}{1+a_2} > 0 \rightarrow -1 < a_2 < 1; a_2 > -1 + a_1; a_2 > -1 - a_1$;②$Controllability : W_c = [\Gamma \quad \Phi\Gamma \quad \Phi^2\Gamma \quad \cdots \quad \Phi^{n-1}\Gamma]$ is NONSINGULAR(full rank). ③ Observability: $W_o = \begin{bmatrix} C \\ C\Phi \\ C\Phi^2 \end{bmatrix}$

4. Pole Placement for State Space Model ①For a **deadbeat controller**, the desired poles are all zero. $A_m(z) = z^2$.
$W_c = [\Gamma \quad \Phi\Gamma], L = [0 \quad 0 \quad \cdots \quad 1]W_c^{-1}A_m(\Phi) = [0 \quad 0 \quad \cdots \quad 1]W_c^{-1}\Phi^2$.The deadbeat controller is u(k) = -Lx(k).
②For a **deadbeat observer**, the desired poles are all zero. $A_o(z) = z^2$.The observer can be expressed as $\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k)); \hat{y} = C\hat{x}(k); e(k+1) = \Phi e(k) - KCe(k) = (\Phi - KC)e(k)$; $W_o = \begin{bmatrix} C \\ C\Phi \end{bmatrix}. ; K = A_o(\Phi)W_o^{-1}[0 \quad 0 \quad \cdots \quad 1]^T = \Phi^2 W_o^{-1}[0 \quad 1]$;The deadbeat observer is $u(k) = -L\hat{x}(k)$;
③Disturbance Rejection:Augmented state vector : z(k)=$\begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix}$ z(k+1)=$\begin{bmatrix} \Phi & \Phi_{\omega x} \\ 0 & 1 \end{bmatrix}$z(k) +$\begin{bmatrix} \Gamma \\ 0 \end{bmatrix}$u(k) ,Choose u(k)=-Lx(k)-$L_w w(k).yields$ S.S model: x(k+1)=($\Phi - \Gamma L)x(k) + (\Phi_{x\omega} - \Gamma L_\omega)\omega(k); (zI - (\Phi - \Gamma L))X(z) = (\Phi_{x\omega} - \Gamma L_\omega)W(z); H_{x\omega}(z) = (zI - (\Phi - \Gamma L))^{-1}(\Phi_{x\omega} - \Gamma L_\omega)$;to reject constant disturbance, make $H_{x\omega}(1) = 0$ to get L_w.
④ **Deadbeat observer to estimate both states and disturbance + feedback controller to eliminate disturbance.** Define the augmented states : z(k)=$\begin{bmatrix} x(k) \\ w(k) \end{bmatrix}$, then the states space model turns to z(k+1)=$\begin{bmatrix} \Phi & \Phi_{xw} \\ 0 & \Phi_w \end{bmatrix}$z(k) +$\begin{bmatrix} \Gamma \\ 0 \end{bmatrix}$u(k); y(k) = [C 0]z(k). Where $\Phi_z = \begin{bmatrix} \Phi & \Phi_{xw} \\ 0 & 1 \end{bmatrix}.\Gamma_z = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}, C_z = [C0]$.The augmented observer is described as $\hat{z}(k+1) = \Phi_z z(k) + \Gamma_z u(k) + K(y - \hat{y}), \hat{y} = C_z\hat{z}(k)$. $W_o = \begin{bmatrix} C_z \\ C_z\Phi_z \\ C_z\Phi_z^2 \end{bmatrix}$The characteristic polynomial is f(z)=z^3, The feedback gain K is K = $\Phi_z^3 W_o^- 1[0 \quad 0 \quad 1]^T$;Then, to eliminate disturbance, u(k)=-Lx(k)-$L_w w(k), H_w(z) = C(zI - (\Phi - \Gamma L))^{-1}(\Phi_{x\omega} - \Gamma L_\omega)$;to reject constant disturbance, make $H_w(1) = 0$ to get L_w. The state feedback controller is u(k)=-L(k)-$L_w\hat{w}(k)$.

5. Pole Placement for IO Model No need observer in transfer function way. Goal is $G(z) = \frac{T(z)B(z)}{A(z)R(z)+S(z)B(z)} = \frac{B_m(z)A_o(z)B(z)}{A_m(z)A_o(z)B(z)} = \frac{B_m(z)}{A_m(z)}.A_{cl}(z) = A_m(z)A_o(z)$ Reference model: $\frac{B_m(z)}{A_m z}$. System: $\frac{B(z)}{A(z)}$. If the order of the plant is n, Deg $A_{cl} = 2n - 1; DegS = n - 1; DegR = n - 1$;①$DesignR(z).If$ $B_m(z)$ not contain B_z, then need to cancel zero. If need to cancel out zeros: R(z) must contain the zero polynomial B(z). Disturbance rejection: R must contain (z-1) for constant disturbance. **Conditions** $Deg(R) >= Deg(S)$ ② Design R and S by solving $AR + BS = A_{cl}$ ③ Choose T or H_{ff} at the final stage. $H_{fb} = \frac{S(z)}{R(z)}; H_{ff} = \frac{B_m(z)}{B(z)R(z)}$.The two-degree of freedom controller U(z)=-$H_{fb}(z)Y(z) + H_{ff}(z)U_c(z); chooseT(z) : G(z) = \frac{RB}{AR+SB}\frac{T}{R} = \frac{T(z)B(z)}{A(z)R(z)+S(z)B(z)}$.To make the steady-state is one, requires G(1) = 1. **Tracking**①$PerfectTracking$. B(z) is stable. ②One-step-ahead controller: derive u(k)(maybe need to shift),let y(k+1) = r(k+1), u(k) is the controller.

6. Kalman Filter

$K_f(k) = P(k|k-1)C^T(CP(k|k-1)C^T + R_2)^{-1}$
$K(k) = Ak_f(k) = (AP(k|k-1)C^T)(CP(k|k-1)C^T + R_2)^{-1}$
$\hat{x}(k|k) = A\hat{x}(k|k) + Bu(k) = \hat{x}(k|k-1) + K_f(k)(y(k) - C\hat{x}(k|k-1))$
$\hat{x}(k+1|k) = A\hat{x}(k|k-1) + Bu(k) + K(k)(y(k) - C\hat{x}(k|k-1))$
$P(k|k) = P(k|k-1) - P(k|k-1)C^T(CP(k|k-1)C^T + R_2)^{-1}CP(k|k-1)$
$P(k+1|k) = AP(k|k)A^T + R_1 = AP(k|k-1)A^T - K(k)(CP(k|k-1)C^T + R_2)K^T(k) + R_1$
$addtional : P(k+1|k) = E\{[x(k+1) - \hat{x}(k+1|k)][x(k+1) - \hat{x}(k+1|k)]^T\}; P(k|k) = E\{[x(k) - \hat{x}(k|k)][x(k) - \hat{x}(k|k)]^T\}$

7. Model Predictive Control

$x_p(k+1) = A_p x_p(k) + B_p u(k); y(k) = C_p x_p(k)$.

$$Augmentedstate-spacemodel : \overbrace{\begin{bmatrix} \Delta x_p(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} = \overbrace{\begin{bmatrix} A_p & o_p^T \\ C_pA_p & 1 \end{bmatrix}}^{A}\overbrace{\begin{bmatrix} \Delta x_p(k) \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_p \\ C_pB_p \end{bmatrix}}^{B}\Delta u(k); \quad y(k) = \overbrace{[o_p \quad 1]}^{C}\overbrace{\begin{bmatrix} \Delta x_p(k) \\ y(k) \end{bmatrix}}^{x(k)}$$

$R_s = \bar{R}_s r(k) = [1 \quad 1 \quad 1]^T \times 1 = [1 \quad 1 \quad 1]^T$

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_P} \end{bmatrix} \Phi = \begin{bmatrix} CB & 0 & 0 & 0 & 0 \\ CAB & CB & 0 & 0 & 0 \\ CA^2B & CAB & CB & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_P-2}B & CA^{N_P-3}B & CA^{N_P-4}B & \cdots & CA^{N_P-N_c-1}B \\ CA^{N_P-1}B & CA^{N_P-2}B & CA^{N_P-3}B & \cdots & CA^{N_P-N_c}B \end{bmatrix} K_r = \overbrace{[1 \quad 0 \quad 0 \quad \cdots \quad 0]}^{N_p}(\Phi^T\Phi + \bar{R})^{-1}\Phi^T\bar{R}_s$$

$K_{mpc} = K_r = \overbrace{[1 \quad 0 \quad 0 \quad \cdots \quad 0]}^{N_p}(\Phi^T\Phi + \bar{R})^{-1}\Phi^T F; \bar{R}_s = \overbrace{[1 \quad 1 \quad \cdots \quad 1]^T}^{N_p}; R_s = \bar{R}_s r(k) = [1 \quad 1 \quad 1]^T \times 1 = [1 \quad 1 \quad 1]^T$;
$\bar{R} = r_w I_{N_c x N_c}$,is a diagonal matrix of weights(r_w). $\Delta u(k) = K_r r(k) - K_{mpc}x(k)$

8. MPC with Constraint $\begin{bmatrix} -I \\ I \\ -C_2 \\ C_2 \\ -\Phi \\ \Phi \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \\ -U^{min} + C_1u(k-1) \\ U^{max} - C_1u(k-1) \\ -Y^{min} + Fx(k) \\ Y^{max} - Fx(k) \end{bmatrix} C_1 = \overbrace{[1 \quad 1 \quad \cdots \quad 1]^T}^{N_c} C_2 = \overbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}}^{N_c}$

Q1. For deadbeat controller, the desired poles are all zero. Thus, the closed-loop characteristic equation is $A_m(z) = z^2$.$W_c = [\Gamma \quad \Phi\Gamma]$, according Ackermann's formula, $L = [0 \quad 1]W_c^{-1}A_m(\Phi) = [0 \quad 1]W_c^{-1}\Phi^2$.The deadbeat controller is u(k) = -Lx(k).
For a **deadbeat observer**, the desired poles are all zero. $A_o(z) = z^2$.The observer can be expressed as $\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k)); \hat{y} = C\hat{x}(k); e(k+1) = \Phi e(k) - KCe(k) = (\Phi - KC)e(k)$; $W_o = \begin{bmatrix} C \\ C\Phi \end{bmatrix}$. ; $K = A_o(\Phi)W_o^{-1}[0 \quad 0 \quad \cdots \quad 1]^T = \Phi^2 W_o^{-1}[0 \quad 1]$;The controller is $u(k) = -L\hat{x}(k); Q2. controller R(q)u(k) = T(q)u_c(k) - S(q)y(k) Apply Z.T. \rightarrow U(z) = \frac{T(z)}{R(z)}U_c(z) - \frac{S(z)}{R(z)}Y(z)$
2rd order system: rise time:$t_r = \frac{1.8}{\omega_n}$,settling time: $t_r = \frac{4.6}{\xi\omega_n}$, overshoot $e^{\frac{-\pi\xi}{\sqrt{1-\xi^2}}}$ .8s 5 overshoot, $H(z) = \frac{0.67}{s^2+1.15s+0.67}$

Example 8

Consider the plant

$$x_p(k+1) = A_p x_p(k) + B_p u_p(k)$$
$$y(k) = C_p x_p(k)$$

where $A_p = 1$, $B_p = 1$, $C_p = 1$ and the augmented plant

$$x(k+1) = Ax(k) + B\Delta u(k)$$
$$y(k) = Cx(k)$$

which according to (2.7)

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} \Delta x_p(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_p(k) - x_p(k-1) \\ C_p x_p(k) \end{bmatrix}$$

and (2.5)

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

where $x_p(k) = y(k) = 0$ for $k \le 0$ and $u(k) = 0$ for $k < 0$.
Consider the MPC with $r_w = 5$, $N_c = 1$, $N_p = 2$ and

$$r(k) = \begin{cases} 0 & \text{for } k < 0 \\ 1 & \text{for } k \ge 0 \end{cases}$$

(a) Formulate a quadratic programming problem with constraint $\begin{bmatrix} y(k+1) \\ y(k+2) \end{bmatrix} \le Y^{max}$.

Equations (2.21), (2.26), (2.17) and (2.18) give

$$\bar{R} = r_w I_{N_c} = 5$$
$$R_s = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$
$$F = \begin{bmatrix} CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$
$$\Phi = \begin{bmatrix} CB \\ CAB \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Equations (3.15), (3.16), (3.17) and (3.18) give

$$H = 10$$
$$f = \begin{bmatrix} 5x_1(k) + 3x_2(k) - 3 \end{bmatrix}$$
$$M = \Phi = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$
$$\gamma = Y^{max} - Fx(k) = \begin{bmatrix} Y^{max} - x_1(k) - x_2(k) \\ Y^{max} - 2x_1(k) - x_2(k) \end{bmatrix}$$

The quadratic programming problem (3.13) with constraint (3.14) can be formulated as minimizing

$$J = \frac{1}{2}\Delta U^T H \Delta U + \Delta U^T f$$
$$M\Delta U \le \gamma$$

Substituting for H, f, M, γ and ΔU gives

$$J = 5\Delta u(k)^2 + (5x_1(k) + 3x_2(k) - 3)\Delta u(k)$$

with constraints

$$\Delta u(k) \le Y^{max} - x_1(k) - x_2(k) \qquad (3.20)$$
$$\Delta u(k) \le \frac{1}{2}(Y^{max} - 2x_1(k) - x_2(k)) \qquad (3.21)$$

(b) Given $u(0) = y(0) = \Delta u(0) = 0$, $x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ and $Y^{max} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$. Find $x(k)$, $\Delta u(k)$ and $y(k)$ for $k = 0, 1, \ldots 9$. □

$$\frac{dJ}{d\Delta u(k)} = 10\Delta u(k) + (5x_1(k) + 3x_2(k) - 3) = 0$$
$$\Delta u(k) = \frac{3 - 5x_1(k) - 3x_2(k)}{10} \qquad (3.22)$$

$k = 0$

Equation (3.22) gives $\Delta u(0) = 0.3$. Constaints (3.20) gives $\Delta u(0) \le 1$ and (3.21) $\Delta u(0) \le 0.5$. The minimum J is obtained from the minimum of $\Delta u(0) = 0.3$, 1, 0.5. Hence $\Delta u(0) = 0.3$. See Figure 3.1.

$k = 1$

$$x(1) = Ax(0) + B\Delta u(0) = \begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}$$

Equation (3.22) gives $\Delta u(1) = 0.06$. Constraints (3.20) give $\Delta u(1) \le 0.4$ and (3.21) $\Delta u(0) \le 0.05$. The minimum J is obtained from the minimum of $\Delta u(1) = 0.06$, 0.4, 0.05. Hence $\Delta u(1) = 0.05$.

$$\frac{Y(z)}{R(z)} = C\left[zI - (A - BK_{mpc})\right]^{-1} BK_r$$

Example 9

A driverless car of mass 1000 kg moving at 10 m/s detected a boy running acrossing the road at the traffic light 15 m ahead although the signal was green for vehicle and red for pedestrian. A MPC with $r_w = 1$, $N_c = 2$, $N_p = 3$ was used to compute the braking signal given that the maximun braking force is 5000 N. □

(a) Formulate a quadratic programming problem to compute the force.

Consider the Newton Law

$$\text{force} = \text{mass} \times \ddot{y}$$
$$\frac{\text{force}}{\text{mass}} = \ddot{y}$$
$$\Delta u = \ddot{y}$$

where y is distance and $\Delta u = \frac{\text{force}}{\text{mass}}$. We will compute Δu. The braking force can be obtained from force = mass × Δu.

Note: There are already 2 integrators in the process. Therefore we will not introduce an integrator into the controller.

We introduce y as the state x_1 and $\dot{y}$ as the state x_2. The state-space representation is then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\Delta u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Discretize by approximating the derivative using the forward difference with sampling interval h gives

$$\begin{bmatrix} \frac{x_1(k+1) - x_1(k)}{h} \\ \frac{x_2(k+1) - x_2(k)}{h} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\Delta u(k)$$
$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

Let $h = 1$ and rearranging gives

$$x(k+1) = Ax(k) + B\Delta u$$
$$y(k) = Cx(k)$$

where

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Formulate a quadratic programming problem with constraints

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ y(k+3) \end{bmatrix} \le Y^{max} = \begin{bmatrix} 15 \\ 15 \\ 15 \end{bmatrix}$$
$$\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} \ge \Delta U^{min} = \begin{bmatrix} -5 \\ -5 \end{bmatrix}$$

Equations (2.21), (2.26), (2.17) and (2.18) give

$$\bar{R} = r_w I_{N_c \times N_c} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$R_s = \overbrace{\begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \end{bmatrix}}^{N_p}{}^T r(k) = \begin{bmatrix} 15 \\ 15 \\ 15 \end{bmatrix}$$
$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$
$$\Phi = \begin{bmatrix} CB & 0 \\ CAB & CB \\ CA^2B & CAB \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 6 & 2 \\ 2 & 2 \end{bmatrix}$$
$$f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} -45 + 3x_1(k) + 8x_2(k) \\ -15 + x_1(k) + 3x_2(k) \end{bmatrix}$$
$$M = \begin{bmatrix} -I \\ \Phi \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$
$$\gamma = \begin{bmatrix} -\Delta U^{min} \\ Y^{max} - Fx(k) \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ 15 - x_1(k) - x_2(k) \\ 15 - x_1(k) - 2x_2(k) \\ 15 - x_1(k) - 3x_2(k) \end{bmatrix}$$

The quadratic programming problem (3.13) with constraint (3.14) can be formulated as minimizing

$$J = \frac{1}{2}\Delta U^T H \Delta U + \Delta U^T f$$
$$M\Delta U \le \gamma$$

Substituting for H, f, M, γ and ΔU gives

$$J = 3\Delta u(k)^2 + [2\Delta u(k+1) + f_1]\Delta u(k) + [\Delta u(k+1) + f_2]\Delta u(k+1) \qquad (3.23)$$

where

IE1: $\Delta u(k) \ge -5$ (3.24)
IE2: $\Delta u(k+1) \ge -5$ (3.25)
IE3: $0 \le 15 - x_1(k) - x_2(k)$ (3.26)
IE4: $\Delta u(k) \le 15 - x_1(k) - 2x_2(k)$ (3.27)
IE5: $2\Delta u(k) + \Delta u(k+1) \le 15 - x_1(k) - 3x_2(k)$ (3.28)

(b) Find $\Delta u(k)$ for $k = 0, 1, 2$ graphically. □

From (3.23)

$$\Delta u(k) = -\frac{2\Delta u(k+1) + f_1}{6} \pm \frac{\sqrt{[2\Delta u(k+1) + f_1]^2 - 12\{[\Delta u(k+1) + f_2]\Delta u(k+1) - J\}}}{6} \qquad (3.29)$$

$k = 0$

$$x(0) = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$
$$f_1 = 3x_1(0) + 8x_2(0) - 45 = 3(0) + 8(10) - 45 = 35$$
$$f_2 = x_1(0) + 3x_2(0) - 15 = 0 + 3(10) - 15 = 15$$

The constraints (3.24), (3.25), (3.27) $\Delta u(0) \le -5$ and (3.28) $\Delta u(0) \le -0.5\Delta u(1) - 7.5$ are superimposed on the $J = -100$ and -102 contours (3.29) in Figure 3.4 which also shows that the minimum $J = -100$ is at $\Delta u(0) = -5$.

$k = 1$

$$x(1) = Ax(0) + B\Delta u(0) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 10 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}(-5) = \begin{bmatrix} 10 \\ 5 \end{bmatrix}$$
$$f_1 = 3x_1(1) + 8x_2(1) - 45 = 3(10) + 8(5) - 45 = 25$$
$$f_2 = x_1(1) + 3x_2(1) - 15 = 10 + 3(5) - 15 = 10$$

The constraints (3.24), (3.25), (3.27) $\Delta u(1) \le -5$, (3.27) $\Delta u(1) \le -0.5\Delta u(2) - 5$ is superimposed on the $J = -50$ and -52 contours (3.29) in Figure 3.5 which also shows that the minimum $J = -50$ is at $\Delta u(1) = -5$. The constraints (3.24) and (3.27) are the tangents to the $J = -50$ contour.

$k = 2$

$$x(2) = Ax(1) + B\Delta u(1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 10 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}(-5) = \begin{bmatrix} 15 \\ 0 \end{bmatrix}$$
$$f_1 = 3x_1(1) + 8x_2(1) - 45 = 3(15) + 8(0) - 45 = 0$$
$$f_2 = x_1(1) + 3x_2(1) - 15 = 15 + 3(0) - 15 = 0$$

The constraints (3.24), (3.25), (3.27) $\Delta u(2) \le 0$, (3.27) $\Delta u(2) \le -0.5\Delta u(3)$ is superimposed on the $J = 0$ and 1 contours (3.29) in Figure 3.6 which also shows that the minimum $J = 0$ is at $\Delta u(2) = 0$.

$$A\dot{h}(t) = -q(t) + u(t)$$
$$= -\frac{1}{R}h(t) + u(t)$$

where the area $A = 10$ m^2 and constant $R = 0.5$ s/m^2. □

(a) Define the height $h(t)$ as the state $x_p(t)$ and output $y(t)$, obtain the discretized state-space model with sampling interval of 1 s. □

The state-space model is then

$$\dot{x}_p(t) = -\frac{1}{AR}x_p(t) + \frac{1}{A}u(t)$$
$$y(t) = x_p(t)$$

Discretize by approximating the derivative using the forward difference with sampling interval h gives

$$\frac{x_p(k+1) - x_p(k)}{h} = -\frac{1}{AR}x_p(k) + \frac{1}{A}u(k)$$
$$y(k) = x_p(k)$$

Substituting the values of A, R and h gives

$$x_p(k+1) = a_p x_p(k) + b_p u(k)$$
$$y(k) = c_p x_p(k)$$

where $a_p = 0.8$, $b_p = 0.1$, $c_p = 1$.

(b) Find $\Delta U(k)$, $\Delta u(k)$, $u(k)$ and $y(k)$ of the MPC with integrator control system for $k = 0$, 1, 2. It is given that $r_w = 0.01$, $N_p = 3$ and $N_c = 2$, $x_p(k) = y(k) = 0$, for $k \le 0$, $u(k) = 0$ for $k < 0$, and

$$r(k) = \begin{cases} 0 & \text{for } k < 0 \\ 1 & \text{for } k \ge 0 \end{cases}$$

$$A = \begin{bmatrix} 0.8 & 0 \\ 0.8 & 1 \end{bmatrix}$$
$$B = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$
$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$R_s = \bar{R}_s r(k) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T \times 1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$$
$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \end{bmatrix} = \begin{bmatrix} 0.8 & 1 \\ 1.44 & 1 \\ 1.952 & 1 \end{bmatrix}$$
$$\Phi = \begin{bmatrix} CB & 0 \\ CAB & CB \\ CA^2B & CAB \end{bmatrix} = \begin{bmatrix} 0.1 & 0 \\ 0.18 & 0.1 \\ 0.244 & 0.18 \end{bmatrix}$$
$$\bar{R} = r_w I_{N_c \times N_c} = 0.01\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$K_r = \overbrace{[1 \quad 0 \quad 0 \quad \cdots \quad 0]}^{N_p}(\Phi^T\Phi + \bar{R})^{-1}\Phi^T\bar{R}_s$$

$$K_{mpc} = K_r = \overbrace{[1 \quad 0 \quad 0 \quad \cdots \quad 0]}^{N_p}(\Phi^T\Phi + \bar{R})^{-1}$$

(d) Find $\Delta u(k)$ and $u(k)$ using K_r and K_{mpc} and compare with the results in (a).

Using (2.29)

$$\Delta u(0) = K_r r(0) - K_{mpc}x(0) = 4.9819$$
$$u(0) = u(-1) + \Delta u(0) = 4.9819$$
$$x(1) = (A - BK_{mpc})x(0) + BK_r r(0) = \begin{bmatrix} 0.4982 \\ 0.4982 \end{bmatrix}$$
$$\Delta u(1) = K_r r(1) - K_{mpc}x(1) = -0.4575$$
$$u(1) = u(0) + \Delta u(1) = 4.5244$$
$$x(2) = (A - BK_{mpc})x(1) + BK_r r(1) = \begin{bmatrix} 0.3528 \\ 0.8510 \end{bmatrix}$$
$$\Delta u(2) = K_r r(2) - K_{mpc}x(2) = -1.352$$
$$u(2) = u(1) + \Delta u(2) = 3.1724$$

•**How to check the stability without solving the equation?**

JURY'S STABILITY TEST: $A(z) = a_0 z^n + a_1 z^{n-1} + \ldots + a_n = 0$

Get the coefficients: $a_0 \quad a_1 \quad \cdots \quad a_{n-1} \quad a_n$

Reverse the order: $a_n \quad a_{n-1} \quad \cdots \quad a_1 \quad a_0 \qquad \times\frac{a_n}{a_0}$

Eliminate the last element a_n
By elementary row operation $\quad a_0^{n-1} \quad a_1^{n-1} \quad \cdots \quad a_{n-1}^{n-1}$

Repeat the process $\quad a_{n-1}^{n-1} \quad a_{n-2}^{n-1} \quad \cdots \quad a_0^{n-1} \qquad \times\frac{a_{n-1}^{n-1}}{a_0^{n-1}}$

Stop when there is only one element left $\quad a_0^0$

The system is stable if all the first elements

$a_0 \quad a_0^{n-1} \quad a_0^{n-2} \quad \cdots \quad a_0^0$ are positive!