On Fundamental and practical limitations in SISO control

(1) Fundamental Limitations of SISO control

★ **Theorem (Bode's integral formula):** Let the loop transfer function of a feedback system satisfy $\lim_{s\to\infty} sL(s) = 0$, and have K unstable poles P_k. Then

$$\int_0^\infty \log|S(j\omega)|\,d\omega = \int_0^\infty \log\left|\frac{1}{1+L(j\omega)}\right| d\omega = \pi\sum P_k$$

⇒ This is a conservation law: "Principle of conservation of dirt" (also) {Waterbed effect}

★ **Theorem (Second waterbed formula):** Suppose that $L(s)$ has a single real RHP zero z or a complex conjugate pair of zeros $z = x \pm jy$ and has N_p RHP-poles, P_i. Let $\overline{P_i}$ denote the complex conjugate of P_i.

↳ Then for closed loop stability the sensitivity function must satisfy:

$$\int_0^\infty \ln|S(j\omega)|\cdot\omega(z,\omega)\,d\omega = \pi \prod_{i=1}^{N_p}\left|\frac{P_i+z}{\overline{P_i}-z}\right|$$

⇒ If the zero is real:

$$C(z,\omega) = \frac{2z}{z^2+\omega^2}$$

⇒ If the zero is complex:

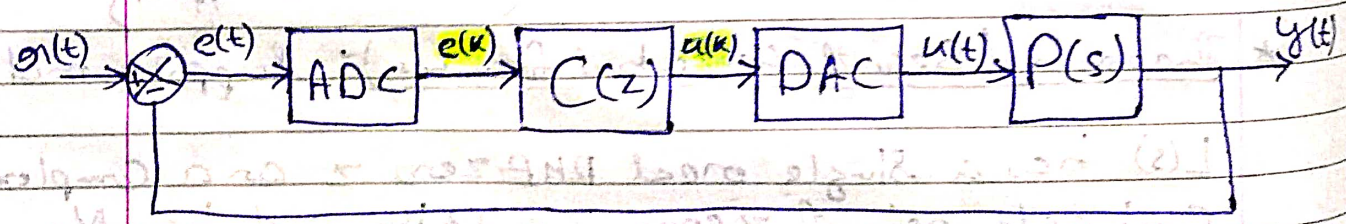
$$C(z,\omega) = \frac{x}{x^2+(y-\omega)^2} + \frac{x}{x^2+(y+\omega)^2}$$

(2) Discrete Time Implementation

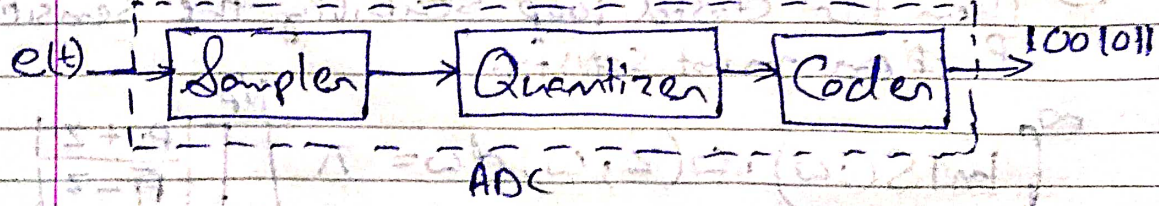
⇒ Two ways to obtain a discrete time controller:

- Discrete-time Synthesis
- Emulation

{Design your system for continuous time and then discritize the controller}



★ Analog to Digital (Uniform Sampling)



Sampling period : T_s

Sampling frequency : $f_s = T_s^{-1}$

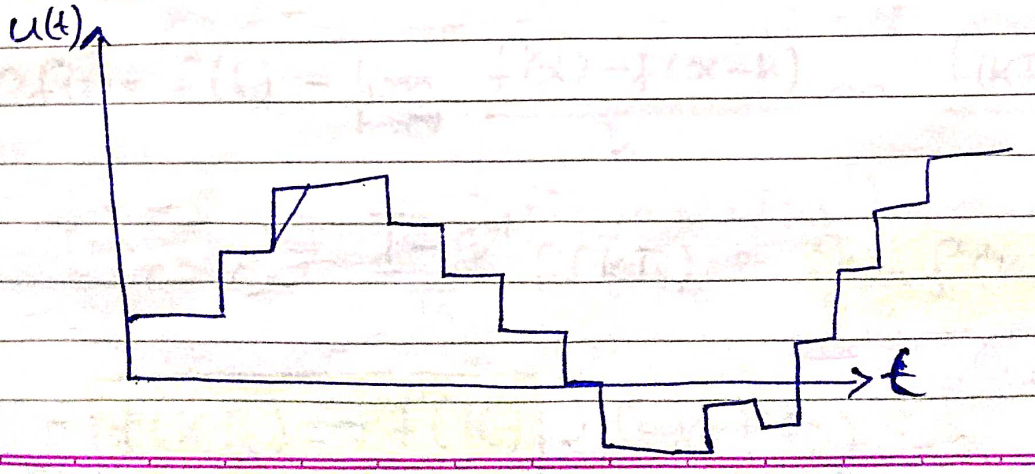
★ Controller implementation

1. Waint for clock interrupt
2. Read input from sensor (ADC)
3. Compute Control signal
4. Write output to the actuator (DAC)
5. Update controller variables
6. Repeat

★ Digital to Analogue : Sample and Hold

Zero order hold

→ Keep sampled value constant until next sample is obtained.



★ Effect of (Uniform) Sampling

Sampling theorem: If a signal $s(t)$ contains no frequencies higher than f_b, it is completely determined by giving ordinates at a series of points spaced [at most] $1/2f_b$ seconds apart.

$$T_s \leq \frac{1}{2\omega_b} \qquad \boxed{f_s \geq 2f_b}$$

$f_{Nyquist}$

⇒ **In practice:** $f_s \geq 5 f_{Nyquist}$

★ Aliasing

⇒ When a signal is not sampled fast enough, aliasing occurs.

↳ Mapping of high frequency oscillation in continuous time to lower frequency oscillation in discrete time.

③ Discrete Time System

★ Finite differences

⇒ Let p be a "derivative operator" such that:

$$pf(t) = \dot{f}(t)$$

⇒ Let q be a "forward shift operator", such that:

$$qf(KT_s) = f(KT_s + T_s)$$

⇒ Forward difference

$$pf(t) = \dot{f}(t) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$\approx \frac{f(KT_s + T_s) - f(KT_s)}{T_s}$$

$$\approx \frac{q-1}{T_s} f(KT_s)$$

⇒ Backward difference

$$pf(t) = \dot{f}(t) = \lim_{h \to 0} \frac{f(x) - f(x-h)}{h} \approx \frac{f(KT_s) - f(KT_s - T_s)}{T_s}$$

$$= \frac{1 - q^{-1}}{T_s} f(KT_s) = \frac{q-1}{qT_s} f(KT_s)$$

$$f(KT_s) = f(K) \quad \{\text{convention}\}$$

★ Numerical Integration

$$\dot{f}(t) = s(t)$$

Euler forward: $f(KT_s + T_s) = f(KT_s) + T_s\, s(KT_s)$

Euler backwards: $f(KT_s + T_s) = f(KT_s) + T_s\, s(KT_s + T_s)$

Tustin (trapezoids) method:

$$f(KT_s + T_s) = f(KT_s) + T_s \frac{s(KT_s) + s(KT_s + T_s)}{2}$$

★ Discritizing transfer functions

	Time	Frequency
Continuous	p	s
Discrete	q	z

Euler Forward: $s \simeq \frac{z-1}{T_s}$

Euler Backwards: $s \simeq \frac{z-1}{zT_s}$

Tustin (trapezoids): $s \simeq \frac{2}{T_s}\left(\frac{z-1}{z+1}\right)$

Page No.
Date: / /

Discrete Controller:

$$C(z) = C(s)\Big|_{s=f(z)}$$

★ Emulation

1. Design a continuous time controller for the continuous time plant
2. Choose a sampling rate that is at least twice (five times in practice) the crossover frequency.
3. If required, design an anti aliasing filter.
4. Modify your controller to take into account the phase lag introduced by the discretization.
5. Emulate the Controller (use Euler / Tustin)
6. Implement the controller.

★ LTI discrete time system {Implicit representation}

⇒ Continuous time differential equation:

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

⇒ Discrete time difference equation:

$$x(k+1) = A_d x(k) + B_d u(k)$$
$$y(k) = C_d x(k) + D_k u(k)$$

Where,

$$A_d = e^{AT_s}$$

$$B_d = \int_0^{T_s} e^{A\lambda}\, d\lambda\, B = A^{-1}(e^{AT} - I)B$$

$$C_d = C A_d$$

$$D_d = D$$

★ LTI discrete time system : Explicit representation

⇒ Continuous time:

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)} Bu(\tau)\, d\tau$$

$$y(t) = Ce^{At}x(0) + C\int_0^t e^{A(t-\tau)} Bu(\tau)\, d\tau + Du(t)$$

⇒ Discrete time:

$$x(k) = A_d^k x(0) + \sum_{i=0}^{k-1} A_d^{k-1-i} B_d u(i)$$

$$y(k) = CA_d^k x(0) + \sum_{i=0}^{k} W(k-i)u(i)$$

$$\left\{ W(k) = \begin{cases} D & k=0 \\ CA_d^{k-1}B_d & k \geq 1 \end{cases} \right\}$$