14. COMPARISON BETWEEN CONTINUOUS AND DISCRETE TIME STATE-SPACE SYSTEMS

The following comparisons can be made :

(a) In a continuous time system *t* may take any value, whereas kT can only take integer values.

(b) The block diagram for a generalised discrete time system is similar to the continuous time system (see Figure 14-1) with the integrator replaced by a one step store (ie this could be a shift register, step in a program or a charged coupled device).

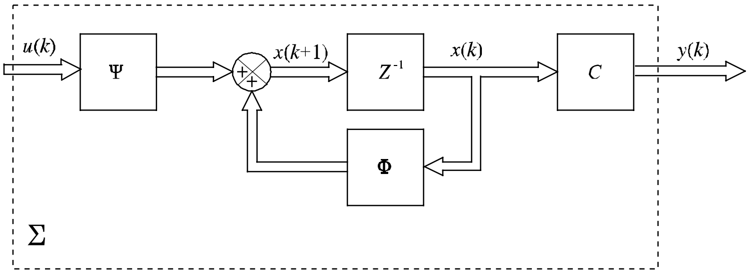

Figure 14-1 Discrete State Model Block Diagram.

(c) The definitions of state stability etc are identical, i.e. the eigenvalues of the continuous time system are the roots of $|sI - A| = 0$, and the corresponding values for the discrete time system are the roots of $|zI - \Phi| = 0$, for stability the roots of this equation must lie inside the unit circle on the *z*-plane.

(d) The transfer function matrix is the same as for the continuous time system, with *s* replaced by *z*, *A* and *B* by Φ and Ψ i.e.:

$$x(k+1) = \Phi x(k) + \Psi u(k) \quad .. \quad \text{taking } z\text{−Transforms} \quad \text{gives}$$

$$zX(z) - zX(0) = \Phi X(z) + \Psi U(z)$$

$$(zI - \Phi)^{-1} X(z) = zX(0) + \Psi U(z)$$

$$X(z) = z(zI - \Phi)^{-1} X(0) + (zI - \Phi)^{-1} \Psi U(z)$$

To obtain a transfer function matrix relating the input to the output we assume zero initial conditions so that $X(0) = 0$ then:

$$X(z) = (zI - \Phi)^{-1} \Psi U(z)$$

$$\Phi(z) = CX(z) + DU(z) = (C(zI - \Phi)^{-1} \Psi + D) U(z)$$

$$Y(z) = G(z) U(z) \quad \therefore \quad G(z) = C(zI - \Phi)^{-1} \Psi + D$$

$$\text{n.b.} \quad G(s) = C(sI - A)^{-1} B + D$$

(e) The controllability and observability matrices become:

$$Q = (\Psi \quad \Phi\Psi \quad \Phi^2\Psi \quad ....) \quad \Rightarrow \quad R = \begin{pmatrix} C \\ C\Phi \\ C\Phi^2 \\ .. \end{pmatrix}$$

(f) Pole shifting techniques apply where:

$$|zI - \Phi + \Psi K| = \text{Desired closed−loop roots}$$

(g) State estimation techniques apply where:

$$| zI - \Phi + MC | = \text{Desired estimator roots}$$

(h) Summarizing, since the analysis is basically the same, the same methods apply for both discrete and continuous state equations, i.e. find Φ and Ψ then apply as if a continuous form.

(i) State Estimated Feedback controller transfer function becomes:

$$K(z) = \frac{U(z)}{Y(z)} = -K(zI - \Phi + \Psi K + MC)\Psi$$

14.1. Stability of a discrete-time system in terms of the eigenvalues of Φ.

The discrete-time system equivalent to the continuous system $\dot{x} = Ax$, is $x(k+1) = \Phi x(k)$ where $\Phi = e^{AT}$ and the free motion of this discrete-time system is described by $x(k) = \Phi^k x(0)$, and this will be stable IFF $\Phi^k \to 0$ as $k \to \infty$.

Let $\{\lambda_i : i = 1,2, \ ... \ ,n\}$ and $\{\mu_i : i = 1,2, \ ... \ ,n\}$ be the eigenvalues of A and Φ respectively. Then since:

$$Au_i = \lambda_i u_i \;\Rightarrow\; A^k u_i = \lambda_i^k u_i \;\Rightarrow\; \Sigma \frac{A^K T^k}{k!} u_i = e^{\lambda_i T} u_i = \Phi u_i = e^{\lambda_i T} u_i$$

$$\mu_i = e^{\lambda_i T} \;\Rightarrow\; i = 1,2, \ ... \ ,n$$

and the eigenvalues of Φ^k will be:

$$\mu_i^k = e^{\lambda_i kT} \;\Rightarrow\; i = 1,2, \ ... \ ,n$$

Hence the discrete-time system will be stable IFF:

$$e^{\lambda_i kT} \rightarrow 0 \text{ as } t \rightarrow \infty \quad .. \quad i = 1,2, \quad .. \quad ,n$$

$$\text{Let: } \lambda_i = \sigma_i + j\,\omega_i \quad .. \quad i = 1,2, \quad .. \quad ,n$$

where some or all of the terms ω_i may of course be zero. Then:

$$|e^{\lambda_i kT}| = |e^{\sigma_i kT}|$$

and since, $kT > 0,$ we will have that:

$$|e^{\lambda_i kT}| \rightarrow 0 \quad \text{IFF} \quad \sigma_i < 0$$

We also have that since $(T > 0)$:

$$|\mu_i| = |e^{\lambda_i T}| = |e^{\sigma_i T}| \;,\; 1 \text{ if } \sigma_i \;,\; 0 \quad \text{and} \quad |\mu_i| = |e^{\lambda_i T}| = |e^{\sigma_i T}| = 1 \text{ if } \sigma_i \geq 0$$

It follows that the discrete-time system will be stable IFF all the eigenvalues of the matrix Φ lie within an origin-centred disc of unit radius in the complex plane.

Example 14.1 Continuous to Discrete State Model.

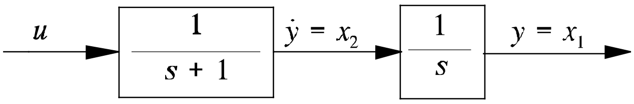

Figure 14-2 System for Example 13.1.

$$\text{Let:}\;\; x_1 = y \quad x_2 = \frac{dy}{dt} = \dot{x}_1 \;\Rightarrow\; \dot{x}_2 = -x_2 + u$$

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \quad .. \quad y = (1 \;\; 0)\, x$$

$$x(k+1) = \Phi(T)x(k) + \Psi(T)u(k) \quad .. \quad T = 1\,\text{second}$$

$$\Phi(s) = (sI - A)^{-1} = \frac{\begin{pmatrix} s+1 & 1 \\ 0 & s \end{pmatrix}}{s(s+1)} = \begin{pmatrix} \frac{1}{s} & \frac{1}{s(s+1)} \\ 0 & \frac{1}{s+1} \end{pmatrix}$$

$$\Phi(T) = \mathfrak{L}^{-1}\Phi(s) = \begin{pmatrix} 1 & 1-e^{-T} \\ 0 & e^{-T} \end{pmatrix} = \begin{pmatrix} 1 & 0.632 \\ 0 & 0.368 \end{pmatrix}$$

$$\Psi(s) = \frac{(sI - A)^{-1}B}{s} = \begin{pmatrix} \frac{1}{s^2} & \frac{1}{s^2(s+1)} \\ 0 & \frac{1}{s(s+1)} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{s^2(s+1)} \\ \frac{1}{s(s+1)} \end{pmatrix}$$

$$\Psi(T) = \mathfrak{L}^{-1}\Psi(s) = \begin{pmatrix} T - 1 + e^{-T} \\ 1 - e^{-T} \end{pmatrix} = \begin{pmatrix} 0.368 \\ 0.632 \end{pmatrix}$$

$$x(k+1) = \Phi(T)x(k) + \Psi(T)u(k)$$

$$x(k+1) = \begin{pmatrix} 1 & 0.632 \\ 0 & 0.368 \end{pmatrix} x(k) + \begin{pmatrix} 0.368 \\ 0.632 \end{pmatrix} u(k)$$

Note: due to choice of states, $x(k)$ is directly related to $x(t)$.

Example 14.2 Discrete State Model via *z*-Transforms.

The same system using *z*-Transforms, gives the following transfer function :

$$G(s) = G_{\text{hold}} G_{\text{sys}}(s) = \frac{1 - e^{-sT}}{s^2(s+1)} \quad .. \quad T = 1\,\text{second}$$

$$G(z) = \left(\frac{z-1}{z}\right) Z\left\{\frac{1}{s^2(s+1)}\right\} = \left(\frac{z-1}{z}\right) Z\left\{\frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1}\right\}$$

$$G(z) = \left(\frac{z-1}{z}\right)\left(\frac{Tz}{(z-1)^2} - \frac{z}{z-1} + \frac{z}{z-e^{-T}}\right) \quad \text{i.e. for } T = 1\,\text{second}$$

$$G(z) = \frac{e^{-1}z + (1 - 2e^{-1})}{(z-1)(z-e^{-1})} = \frac{0.368(z + 0.718)}{(z-1)(z-0.368)}$$

Using our discrete state model, we can evaluate $G(z)$ using:

$$G(z) = C(zI - \Phi)^{-1}\Psi + D = \frac{(1 \;\; 0)\begin{pmatrix} z - 0.368 & 0.632 \\ 0 & z-1 \end{pmatrix}\begin{pmatrix} 0.368 \\ 0.632 \end{pmatrix}}{(z-1)(z-0.368)}$$

$$\therefore \quad G(z) = \frac{(1 \;\; 0)\begin{pmatrix} 0.368\; z + 0.264 \\ 0.632(z-1) \end{pmatrix}}{(z-1)(z-0.368)} = \frac{0.368\,(z + 0.718)}{(z-1)(z-0.368)}$$

Clearly this leads to the same transfer function, hence the name ZOH equivalent in the discrete state format.

Note: It will be shown later that if a system can be represented by various different state equations, all will lead to the **same** transfer function, and hence the same output. However clearly the opposite will NOT be the case, so that if it is important to retain the same state links as is normally the case the ZOH method should be applied.

14.2. Realization of State equations from discrete forms

As stated previously, since the algebra of the continuous and discrete time state equations and transfer functions are identical, with *A*, *B*, *C*, *D* and *s* replaced with **Φ**, **Ψ**, *C*, *D* and *z*. The methods of realizing state equations from transfer functions are also identical.

SISO systems have transfer functions in the form :

$$G(z) = \frac{Y(z)}{U(z)} = \frac{Y(z)}{X(z)} \times \frac{X(z)}{U(z)} = \frac{a_q z^q + a_{q-1} z^{q-1} + \ldots + a_0}{z^p + b_{p-1} z^{p-1} + \ldots + b_0}$$

$$G(z) = \frac{a_{p-1} z^{p-1} + a_{p-2} z^{p-2} + \ldots + a_0}{z^p + b_{p-1} z^{p-1} + \ldots + b_0}$$

Using the rules for numerator dynamics as in continuous case gives :

$$x(k+1) = \begin{pmatrix} 0 & 1 & 0 & . \; . \; . & 0 \\ 0 & 0 & 1 & . \; . \; . & 0 \\ . & . & . & . \; . \; . & . \\ . & . & . & . \; . \; . & . \\ -b_0 & -b_1 & -b_2 & . \; . \; . & -b_{p\text{-}1} \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ 1 \end{pmatrix} u(k) \quad :$$

$$y(k) = (a_0 \quad a_1 \quad \ldots \quad a_{p-1}) x(k)$$

Example 14.3 *z*-Plane Model to State Model

$$G(z) = \frac{0.368(z+0.718)}{(z-1)(z-0.368)} = \frac{0.368(z+0.718)}{z^2 - 1.368z + 0.368}$$

By inspection : $x(k+1) = \begin{pmatrix} 0 & 1 \\ -0.368 & 1.368 \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(k)$

$$y(k) = (0.368 \quad 0.718) x(k)$$

Note: This method leads to different states to the original model, however the output will be the same.

Example 14.4 State Model from Difference Equations

A discrete system is governed by the difference equation below, derive the discrete state space model for the system.

$$y(k+3) - 2.4y(k+2) + 1.85y(k+1) - 0.45y(k) = 400u(k)$$

Typically:

Let

$$x_1(k) = y(k)$$

$$x_2(k) = y(k+1)$$

$$x_3(k) = y(k+2)$$

hence;

$$x(k+1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.45 & -1.85 & 2.4 \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 0 \\ 400 \end{pmatrix} u(k)$$

$$y(k) = (1 \quad 0 \quad 0)\, x(k)$$

Example 14.5 State Model from Computer Algorithm.

Consider the computer algorithm:

$$y(k) = u(k-1) + 0.5u(k-2) + y(k-1) - 0.5y(k-2)$$

Let: $x_1(k) = y(k)$

$$x_2(k) = x_1(k+1) - x_1(k) - u(k)$$

appropriate rearrangement leads to :

$$x(k+1) = \begin{pmatrix} 1 & 1 \\ -0.5 & 0 \end{pmatrix} x(k) + \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} u(k)$$

$$y(k) = (1 \quad 0)\, x(k)$$

14.3. Deadbeat Revisited

We shall introduce this topic via a simplified proof of controllability. Earlier we showed that:

$$x(k) = \Phi^k x(0) + \sum_{i=0}^{k-1} \Phi^{k-i-1}\Psi u(k) = \Phi^k x(0) + \Phi^{k-1}\Psi u(0) + \Phi^{k-2}\Psi u(1) + \;..\; + \Psi u(k-1)$$

$$\begin{pmatrix} x_1(k) \\ . \\ . \\ x_k(k) \end{pmatrix} - \Phi^k \begin{pmatrix} x_1(0) \\ . \\ . \\ x_k(0) \end{pmatrix} = (\Psi \;\; \Phi\Psi \;\; . \;\; . \;\; \Phi^{k-1}\Psi) \begin{pmatrix} u(k-1) \\ . \\ . \\ u(0) \end{pmatrix} = z = Qu \;\; .. \;\; [1]$$

i.e. we want to go from one state to another in finite time.

Q is termed the Controllability matrix. The system will be controllable if Q has an inverse, then we can generate an appropriate input sequence given by:

$$u = Q^{-1} z \;\; .. \;\; [2]$$

Hence any system $\sum$ is controllable IFF:

$$Q = (\Phi \;\; \Phi\Psi \;\; . \;\; . \;\; \Phi^{n-1}\Psi) \text{ is of rank } n$$

An important consequence of this in digital control is that any n^{th} order system which is controllable, can move from its current state to **any** other state in n time steps. In simple terms **deadbeat** control provides a way of computing this input in the form of state feedback.

14.4. Summary

A pole located @ $s = -\infty$ maps to $z = 0$, i.e. $(\Phi - \Psi K)^n = z^n = 0$. The system settling time will then be nT seconds. So that if we know our required settling time, set the sampling rate on **completion** of the design using: $T = \dfrac{T_{\text{settling}}}{n}$.

In general only use if:

(i) plant model is very accurate.

(ii) No disturbances acting.

(iii) Plant must be able to handle large discontinuities @ its input.

Example 14.6 Deadbeat/Controllability Revisited

Consider the system:

$$x(k+1) = \begin{pmatrix} 1.8 & 1 \\ 0.85 & 0 \end{pmatrix} x(k) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(k)$$

$$Q = (\Phi \quad \Phi\Psi) = \begin{pmatrix} 1 & 1.8 \\ 0 & 0.85 \end{pmatrix} \Rightarrow |Q| \neq 0$$

Hence the system is fully controllable.

given that:

$$x(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

We want want to move in 2 steps (**second order**) *to*:

$$x(2) = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

From [2]:

$$u = \begin{pmatrix} u(1) \\ u(0) \end{pmatrix} = Q^{-1}z = Q^{-1}\left(\begin{pmatrix} x_1(2) \\ x_2(2) \end{pmatrix} - \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix}\right)$$

$$= Q^{-1}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 1 & -2.1176 \\ 0 & 1.1765 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha - 2.1176\beta \\ 1.1765\beta \end{pmatrix}$$

From the state update equation, since $x(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$\Rightarrow \quad x(1) = \Phi x(0) + \Psi u(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(0) = \begin{pmatrix} 1.1765\beta \\ 0 \end{pmatrix}$$

Similarly: $x(2) = \begin{pmatrix} 1.8 & 1 \\ 0.85 & 0 \end{pmatrix}\begin{pmatrix} 1.1765\beta \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(1)$

$$= \begin{pmatrix} 2.1176\beta \\ \beta \end{pmatrix} + \begin{pmatrix} \alpha - 2.1176\beta \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$ checks out ok.

Example 14.7 Consider the sampled-data system:

$$x(k+1) = \begin{pmatrix} 0 & 0.9 \\ -0.9 & 0 \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(k)$$

(i) Generally, what is the minimum time for the system to be brought to rest from an arbitrary initial state? Are there some non-zero initial states that can be driven to zero in a shorter time?

(ii) Design a state feedback controller that brings the system to rest in minimum time from all initial conditions.

This question implies a deadbeat response, for a second order system this generally means we require a minimum of **2** steps.

However in **1** step:

$$x(1) = \Phi x(0) + \Psi u(0) \;\Rightarrow\; x(0) = 0 \;\; \text{IFF} \;\; x(0) = -\Phi^{-1}\Psi u(0) \;\; \text{for some } u(0).$$

In **2** steps:

$$x(2) = \Phi x(1) + \Psi u(1) = \Phi^2 x(0) + \Phi\Psi u(0) + \Psi u(1)$$

$$\text{For } x(2) = 0 \;\Rightarrow\; -\Phi^2 x(0) = (\Psi \;\; \Phi\Psi)\begin{pmatrix} u(1) \\ u(0) \end{pmatrix} = Qu = \begin{pmatrix} 0 & 0.9 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} u(1) \\ u(0) \end{pmatrix} \;\Rightarrow$$

since Q has full rank, we can always compute the $u(0)$, $u(1)$ necessary to make $x(2) = 0$

$$x(1) \text{ can only be made zero IFF: } x(0) = -\begin{pmatrix} 0 & 0.9 \\ -0.9 & 0 \end{pmatrix}^{-1}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\alpha \quad \text{where } \alpha \text{ is some value of } u(0)$$

$$x(0) = -\frac{1}{0.9^2}\begin{pmatrix} 0 & -0.9 \\ 0.9 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\alpha = \begin{pmatrix} \beta \\ 0 \end{pmatrix} \quad \text{for some constant } \beta.$$

Hence some initial states of this form, can be driven to zero in **1** time step, but the minimum time to drive an **arbitrary** state to zero is **2** steps for any second order system.

Deadbeat state feedback requires the matrix $(\Phi - \Psi K)$ to have all its eigenvalues @ zero:

$$|zI - \Phi + \Psi K| = z^2 = \begin{vmatrix} z & -0.9 \\ 0.9 + k_1 & z + k_3 \end{vmatrix} = z^2 + k_2 z + 0.9(0.9 - k_1) \;\; \text{by inspection: } K = (0.9 \;\; 0).$$

Example 14.8 Deadbeat again !!

Consider the system:

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \;\leftarrow \text{Z.O.H.}_{\text{equivalent}} \rightarrow\; x(k+1) = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} x(k) + \begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix} u(k)$$

Consider the problem of driving the state from $x(0) = x_0$ to $x(2T) = 0,$ discussing when it is possible and determine the **size** of inputs required.

$$x(2T) = \Phi x(T) + \Psi u(1) = \Phi^2 x_0 + \Psi\Phi u(0) + \Psi u(1)$$

$$x(2T) = 0 \;\Rightarrow\; z = Qu = (\Phi \;\; \Psi\Phi)\begin{pmatrix} u(1) \\ u(0) \end{pmatrix} = -\Phi^2 x_0 = -e^{2\Phi T} x_0$$

$$Q = \left(\begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix} \begin{pmatrix} \frac{3T^2}{2} \\ T \end{pmatrix} \right) \;\Rightarrow\; |Q| = -T^3 \neq 0 \;\Rightarrow\; \text{fully controllable}$$

$$\begin{pmatrix} u(1) \\ u(0) \end{pmatrix} = -Q^{-1} e^{2\Phi T} x_0 = -\frac{1}{-T^3} \begin{pmatrix} T & \frac{-3T^2}{2} \\ -T & \frac{T^2}{2} \end{pmatrix} \begin{pmatrix} 1 & 2T \\ 0 & 1 \end{pmatrix} x_0 = \frac{1}{T^3} \begin{pmatrix} T & \frac{T^2}{2} \\ -T & \frac{-3T^2}{2} \end{pmatrix} \begin{pmatrix} x_{01} \\ x_{02} \end{pmatrix}$$

Note that $\ddot{x} = u$ i.e. u is really acceleration here so for T small $\frac{1}{T^2}$ is large and may saturate.

This example illustrates that deadbeat controllers are based on the **design** parameter which is the sampling rate! i.e. you need to check that the control signal u will not saturate. Since we know the system will reach its goal in **2** steps this can easily be achieved analytically.

Example 14.9 Guess what ??

Consider the difference equation:

$$y(kT) = \alpha y((k-1)T) + \beta u((k-2)T)$$

Using the state vector x in which $x_1(kT) = u((k-1)T)$ and $x_2(kT) = y(kT)$

$$y(kT) = \alpha y(kT - T) + \beta u(kT - 2T)$$

$$x_1(kT) = u(kT - T) \;..\; x_2(kT) = y(kT) = \alpha y(kT{-}T) + \beta u(kT - 2T)$$

$$= \alpha x_2((k-1)T) + \beta x_1((k-1)T)$$

$$x(k+1)T = \begin{pmatrix} 0 & 0 \\ \beta & \alpha \end{pmatrix} x(kT) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(kT)$$

Show that the controller:

$$u(kT) = -\alpha u((k-1)T) - \frac{\alpha^2}{\beta} y(kT) + \frac{1}{\beta} r(kT)$$

where $r(kT)$ is a command signal, is a deadbeat controller.

$$u(kT) = -\alpha x_1(kT) - \frac{\alpha^2}{\beta} x_2(kT) + \frac{1}{\beta} r(kT)$$

$$x(k+1)T = \begin{pmatrix} 0 & 0 \\ \beta & \alpha \end{pmatrix} x(kT) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \left\{ \begin{pmatrix} -\alpha & -\frac{\alpha^2}{\beta} \end{pmatrix} x(kT) + \frac{1}{\beta} r(kT) \right\}$$

$$x(k+1)T = \begin{pmatrix} -\alpha & -\frac{\alpha^2}{\beta} \\ \beta & \alpha \end{pmatrix} x(kT) + \begin{pmatrix} \frac{1}{\beta} \\ 0 \end{pmatrix} r(kT)$$

$$|zI - \Phi + \Psi K|_{\text{deadbeat}} = z^2 = \begin{vmatrix} z + \alpha & \frac{\alpha^2}{\beta} \\ -\beta & z - \alpha \end{vmatrix} = z^2 - \alpha^2 + \alpha^2$$

Hence all closed-loop poles at 0 .. **deadbeat**:

$$Y(z) = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} z + \alpha & \frac{\alpha^2}{\beta} \\ -\beta & z - \alpha \end{pmatrix} \begin{pmatrix} \frac{1}{\beta} \\ 0 \end{pmatrix} R(z) = \frac{1}{z^2} \times (\beta) \times \frac{1}{\beta} \times W(z) = \frac{W(z)}{z^2}$$