This code really implements a difference equation:

$$I_k = I_{k-1} + Ts*Error_{k-1}$$

Then, we can take the Z-transform of this difference equation for the integral computation. (We will assume zero initial conditions since we are trying to determine a transfer function.)

$$I[z] = z^{-1}I[z] + z^{-1}Ts*Error[z]$$

So,

$$I[z](1 - z^{-1}) = z^{-1}Ts*Error[z]$$

and:

$$I[z] = z^{-1}Ts*Error[z]/(1 - z^{-1})$$

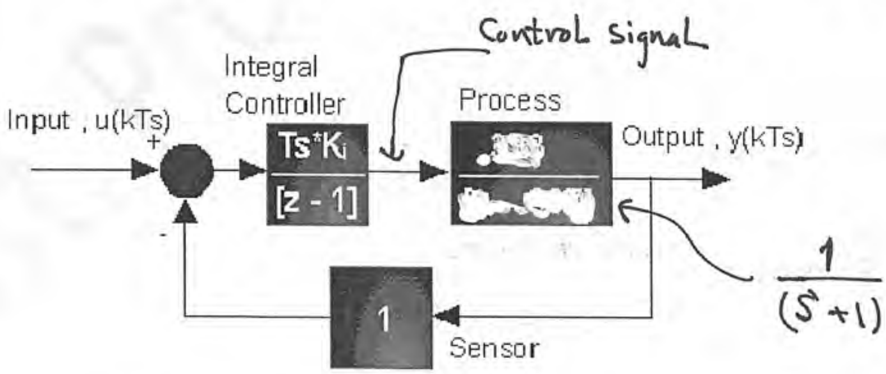
and:

$$I[z] = Error[z]*\{Ts/(z - 1)\}$$

The transfer function of this digital integrator is then given by:

$$G_{int}[z] = Ts/(z - 1)$$

How do we use this result? Well, if you implement digital control in a program - as above - then, the equivalent block diagram for that digital implementation is the one shown below.



In this system, the digital integration is shown with $G_{int}[z]$.

Next, we refer to the block diagram above, and we note that we need the equivalent transfer function for the continuous system in the sampled representation. That equivalent transfer function is given by:

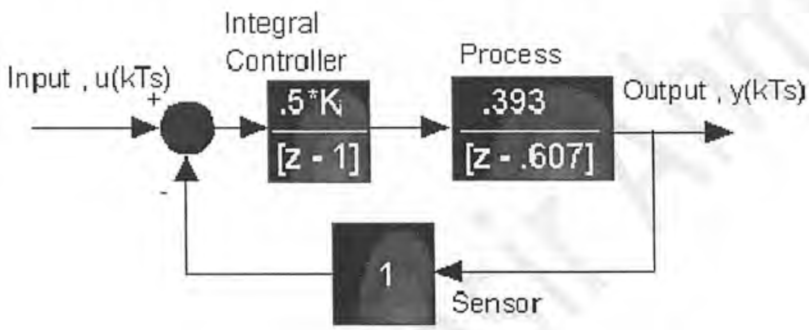
$$G[z] = (1 - e^{-Ts/\tau})/(z - e^{-Ts/\tau})$$

$$= (1 - z^{-1})\, Z \left\{ \frac{A}{s} + \frac{B}{s+1} \right\}$$

With:

- $Ts = 0.5$ sec, and;
- $\tau = 1$ sec.
- $G[z] = (1 - 0.607)/(z - 0.607)$

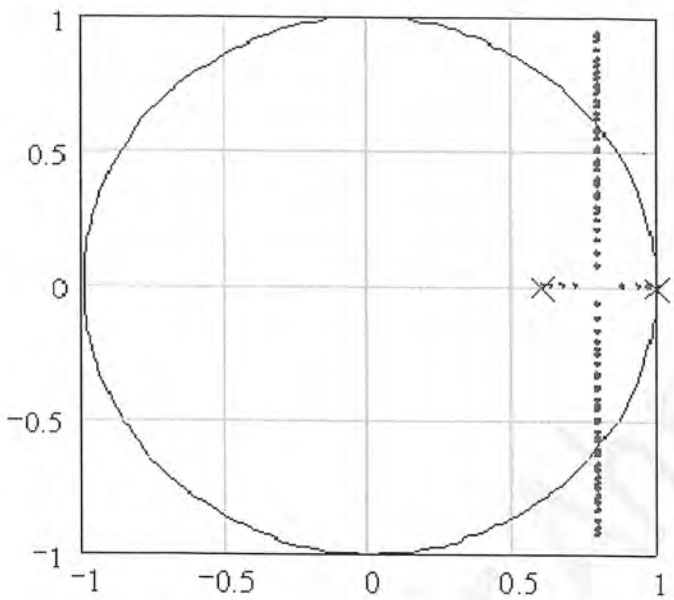
The block diagram for the equivalent system is the one shown below.



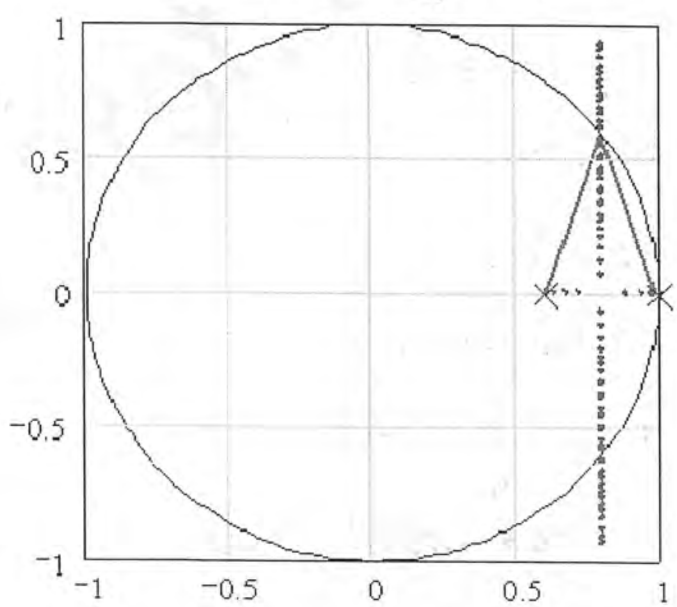
Note that the numerator of the integral controller block has a term $.5^*K_i$. That term is really Ts^*K_i, but the process transfer function, $G(s)$, was transformed assuming that the sampling period was 0.5 second. You have to use the same sampling period in the calculation of the integral.

Now, you can do some analysis of the equivalent system. Doing a root locus, we find some interesting things can happen in this system also. Here is the root locus for the control system :-

$$G(z)\,G_{int}(z) = \frac{\overbrace{0.5K_i * 0.393}^{\bar{K}}}{(z-1)(z-0.607)}$$



In this root locus, we again see that the system can become unstable. To calculate the gain at which the system becomes unstable, we first calculate the root locus gain as the product of pole distances to the point on the locus where the poles exit the unit circle. That's shown in the plot below. or using Jury Test.



We need to determine the length of the two lines shown in purple on the plot. Actually, they are both the same length, and each length is about 0.6. Taking that value, we calculate the root locus gain.

K_{cr} → $K_{RL} = (0.6)^2 = 0.36$

Then, we have (Check the block diagram above.):

$$K_{RL} = 0.5 * K_i * 0.393$$

So, the integrator gain, K_i, at which the system becomes stable is:

$$K_i = 0.36/(0.5 * 0.393)$$

or:

$$K_i = 1.83$$

Or

This value is the same Kcr

Note that: When using Jury Test

c/cs equ = $z^2 - 1.607z + 0.607 + \bar{K}$

Conditions: $1 > (0.607 + \bar{K})$

$\therefore \bar{K} < 0.393$ — while when using lengths $\bar{K} = 0.36$

$Q(1) > 0 \Rightarrow \bar{K} > 0$

$Q(-1) > 0 \Rightarrow \bar{K} > -3.214$ X

$\therefore 0 < \bar{K} < 0.393$

↓

$0.5 * 0.393 K_i < 0.393$

$\therefore$ $\boxed{K_i < 2}$ → like $K_i = 1.83$

Chapter 7

7.1 State-Space Representations of Discrete-time Systems

- Many ways to realize state-space representation
  - Controllable canonical form
  - Observable canonical form
  - Diagonal canonical form
  - Jordan canonical form

In control engineering, a state space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations.

Why state space equations ?

- dynamical systems where physical equations can be derived : electrical engineering, mechanical engineering, aerospace engineering, microsystems, process plants ....
- include physical parameters: easy to use when parameters are changed for design
- State variables have physical meaning.
- Easy to extend to Multi-Input Multi-Output (MIMO) systems
- Advanced control design method are based on state space equations (reliable numerical optimisation tools)

When a discrete system is composed of all digital signals, the state and output equations can be described by

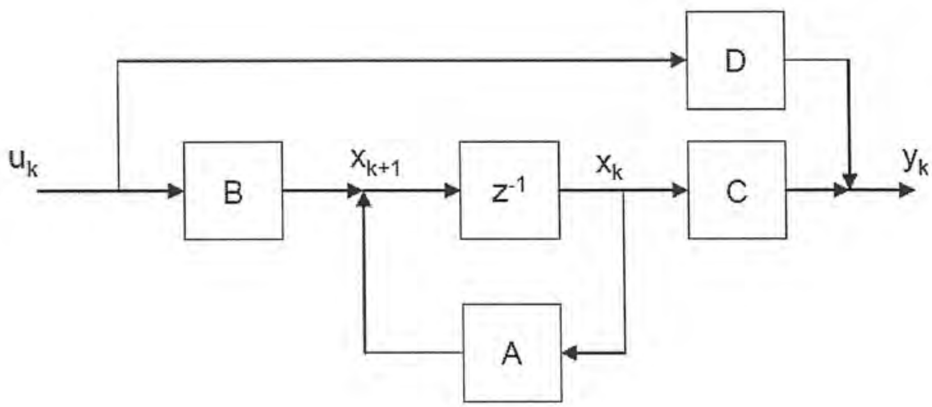
$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad \text{---}(1)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) \quad \text{---}(2)$$

where $\mathbf{x}(k)$ is an n-vector, $\mathbf{u}(k)$ is an r-vector, $\mathbf{y}(k)$ is a p-vector (as shown in Figure), and $\mathbf{A}(k)$, $\mathbf{B}(k)$, $\mathbf{C}(k)$, and $\mathbf{D}(k)$ are time-varying matrices of dimensions $n \times n$, $n \times r$, $p \times n$, and $p \times r$, respectively. If the system is time invariant, then the matrices in (1) and (2) are constant, and hence the equations reduce to

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (3)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (4)$$



Linear Discrete-Time State-Space System

We will discuss about the relation between transfer function and state space model for a discrete time system and various standard or canonical state variable models.

7.2 Various Canonical Forms

We have seen that transform domain analysis of a digital control system yields a transfer function of the following form.

$$G(z) = \frac{Y(z)}{U(z)} = \frac{\beta_0 z^m + \beta_1 z^{m-1} + \ldots + \beta_m}{z^n + \alpha_1 z^{n-1} + \ldots + \alpha_n} \quad m \leq n \qquad \text{----- (5)}$$

1- Controllable canonical form

Consider the transfer function as given in Eqn. (5). Without loss of generality, we assume m=n. Let

$$\frac{\bar{X}(z)}{U(z)} = \frac{1}{z^n + \alpha_1 z^{n-1} + \ldots + \alpha_n}$$

In time domain, the above equation may be written as

$$\bar{x}(k+n) + \alpha_1 \bar{x}(k+n-1) + \ldots + \alpha_n \bar{x}(k) = u(k)$$

Now, the output $Y(z)$ may be written in terms of $\bar{X}(z)$ as

$$Y(z) = \left(\beta_0 z^n + \beta_1 z^{n-1} + \ldots + \beta_n\right) \bar{X}(z)$$

or in time domain as

$$y(k) = \beta_0 \bar{x}(k+n) + \beta_1 \bar{x}(k+n-1) + \ldots + \beta_n \bar{x}(k)$$

The state equations are then written as:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_3(k) \\ \vdots &= \vdots \\ x_n(k+1) &= -\alpha_n x_1(k) - \alpha_{n-1} x_2(k) - \ldots - \alpha_1 x_n(k) + u(k) \end{aligned}$$

Output equation can be written as by following

$$y(k) = (\beta_n - \alpha_n\beta_0)x_1(k) + (\beta_{n-1} - \alpha_{n-1}\beta_0)x_2(k) + \ldots + (\beta_1 - \alpha_1\beta_0)x_n(k) + \beta_0 u(k)$$

In state space form, we have

$$\begin{aligned} \mathrm{x}(k+1) &= A\mathrm{x}(k) + Bu(k) \\ y(k) &= C\mathrm{x}(k) + Du(k) \end{aligned}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -\alpha_n & -\alpha_{n-1} & -\alpha_{n-2} & \ldots & -\alpha_1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$C = [\beta_n - \alpha_n\beta_0 \;\; \beta_{n-1} - \alpha_{n-1}\beta_0 \;\; \ldots \;\; \beta_1 - \alpha_1\beta_0] \qquad D = \beta_0$$

2-Observable Canonical Form

Equation (5) may be rewritten as

$$\left(z^n + \alpha_1 z^{n-1} + \ldots + \alpha_n\right) Y(z) = \left(\beta_0 z^n + \beta_1 z^{n-1} + \ldots + \beta_n\right) U(z)$$

or, $z^n[Y(z) - \beta_0 U(z)] + z^{n-1}[\alpha_1 Y(z) - \beta_1 U(z)] + \ldots + [\alpha_n Y(z) - \beta_n U(z)] = 0$

$$Y(z) = \beta_0 U(z) - z^{-1}[\alpha_1 Y(z) - \beta_1 U(z)] - \ldots - z^{-n}[\alpha_n Y(z) - \beta_n U(z)]$$

This can be rewritten in matrix form with

$$A = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & -\alpha_n \\ 1 & 0 & 0 & \ldots & 0 & -\alpha_{n-1} \\ 0 & 1 & 0 & \ldots & 0 & -\alpha_{n-2} \\ \vdots & \vdots & \vdots & \ldots & \vdots & \\ 0 & 0 & 0 & \ldots & 1 & -\alpha_1 \end{bmatrix} \quad B = \begin{bmatrix} \beta_n - \alpha_n\beta_0 \\ \beta_{n-1} - \alpha_{n-1}\beta_0 \\ \vdots \\ \beta_1 - \alpha_1\beta_0 \end{bmatrix} \quad C = [0 \; 0 \; \ldots \; 1] \quad D = \beta_0$$