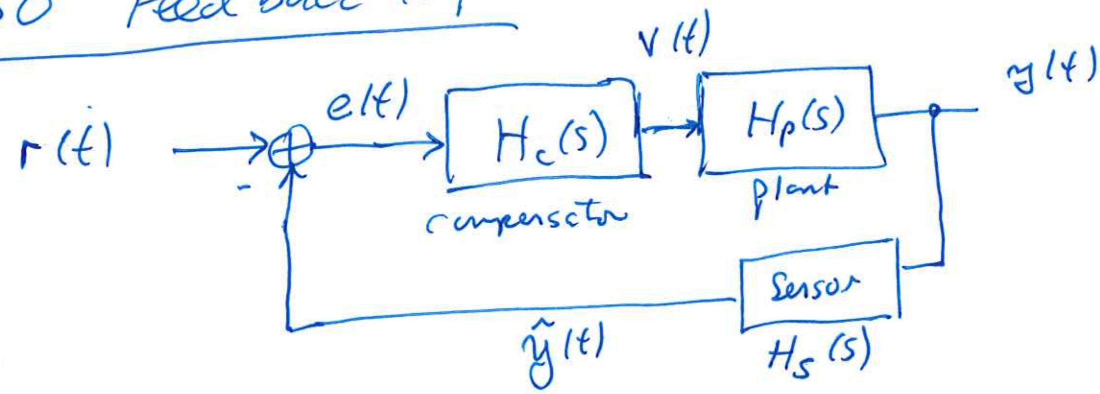
Lecture notes on Digital Control Unit 5

References: ① Unit 5 notes posted on D2L
② Textbook Nise Chapter 13

Recommend that you work through these as we go through this material.

SISO Feedback loop



$r(t)$ - input reference

$e(t)$ - error

$v(t)$ - plant drive

$y(t)$ - plant output

$\hat{y}(t)$ - measurement of $y(t)$
- usually approximate $\hat{y}(t) = y(t)$ ideal sensor feedback.

Digital sensor feedback

$$y(t) \longrightarrow \boxed{H_s(s)} \longrightarrow \boxed{ADC} \longrightarrow \hat{y}_k$$

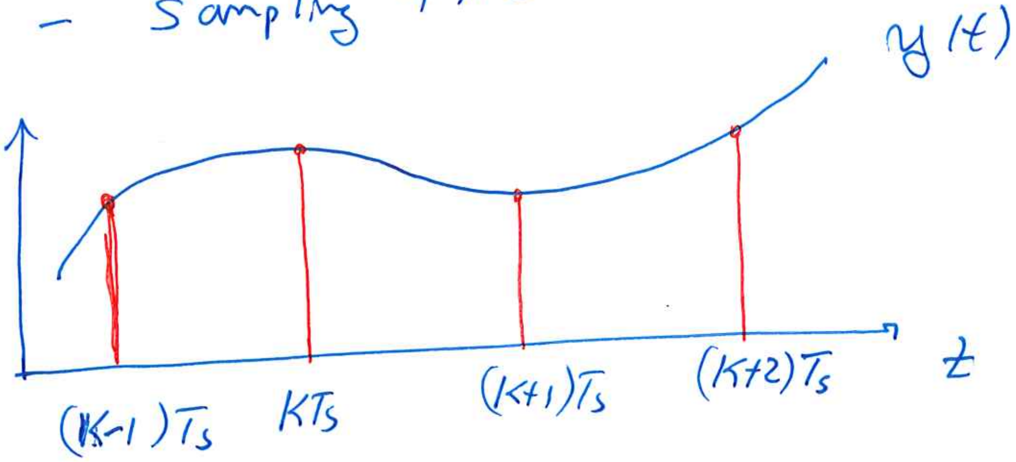
$\hat{y}_k$ discrete time sampled and quantized version of the signal $y(t)$.

For the present ignore the transfer function $H_s(s)$ or set $H_s(s) = 1$

$$\hat{y}_k \rightarrow y_k \quad \text{'drop the } \wedge \text{ notation'}$$

$$y_k = y(t)\Big|_{t = kT_s}$$

T_s - sampling time

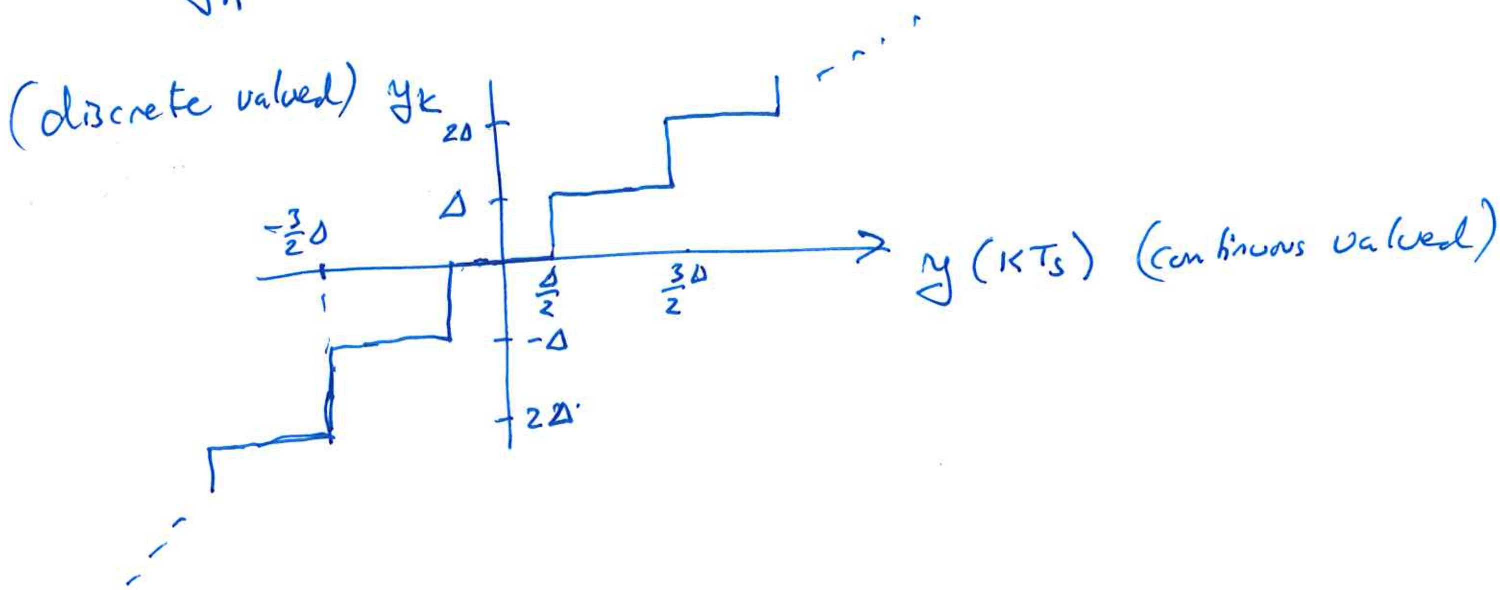


y_k is discrete time sampled representation of the continuous time signal $y(t)$

y_k is further quantized in amplitude by the ADC. Represented by rounding off the real value sample into an integer.

Suppose we have a quantization step of Δ then

$$y_k = \Delta \, \text{round}\left(\frac{y(kT_s)}{\Delta} \right)$$



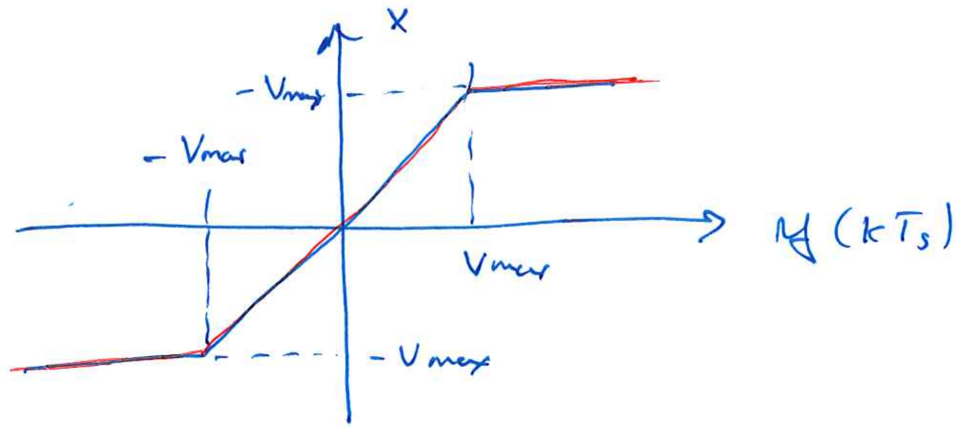
Real signal quantizers further have a saturation level.

upper limit V_{max}
lower limit $-V_{max}$

Model of saturation:

$$x = \max\left(\min\left(y(kT_s), V_{max}\right), -V_{max}\right)$$

This operation limits the output to within $-V_{max}$ to V_{max}



Saturation of ADC, DAC, sensors and compensator outputs is an issue for control systems.

Overall Model of ADC

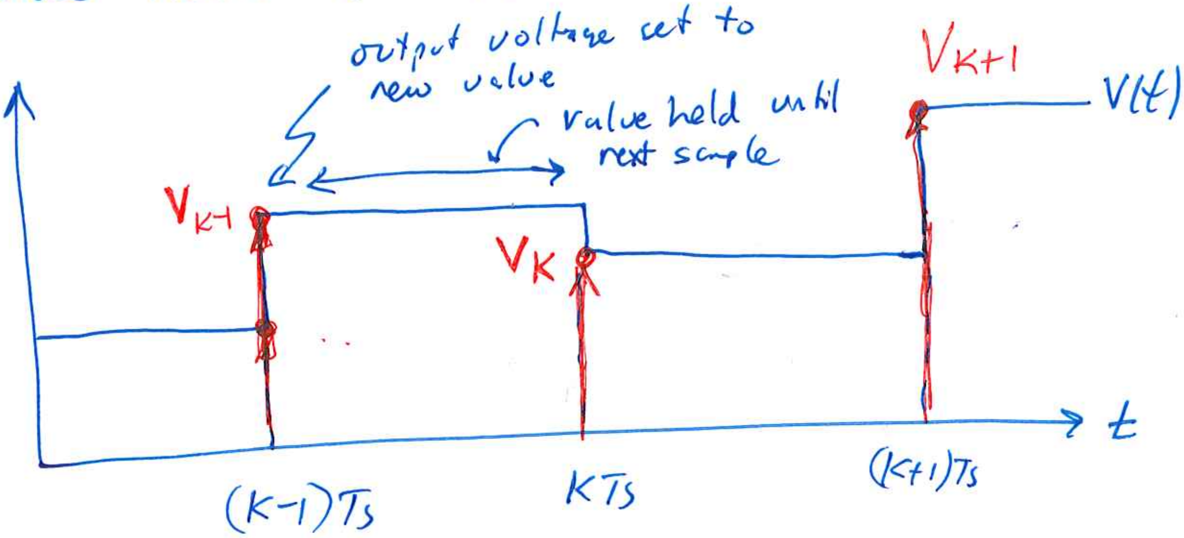
$$y_k = \Delta \,\mathrm{round}\left(\frac{1}{\Delta}\max\left(\min\left(y(kT_s), V_{max}\right), -V_{max}\right)\right)$$

DAC model (Digital to Analog Converter)

We will have a compensator output that is a discrete time sample denoted by v_k

DAC maps this into a continuous time signal.

Generally we assume a model of the DAC as a 'Zero order hold' (ZOH).



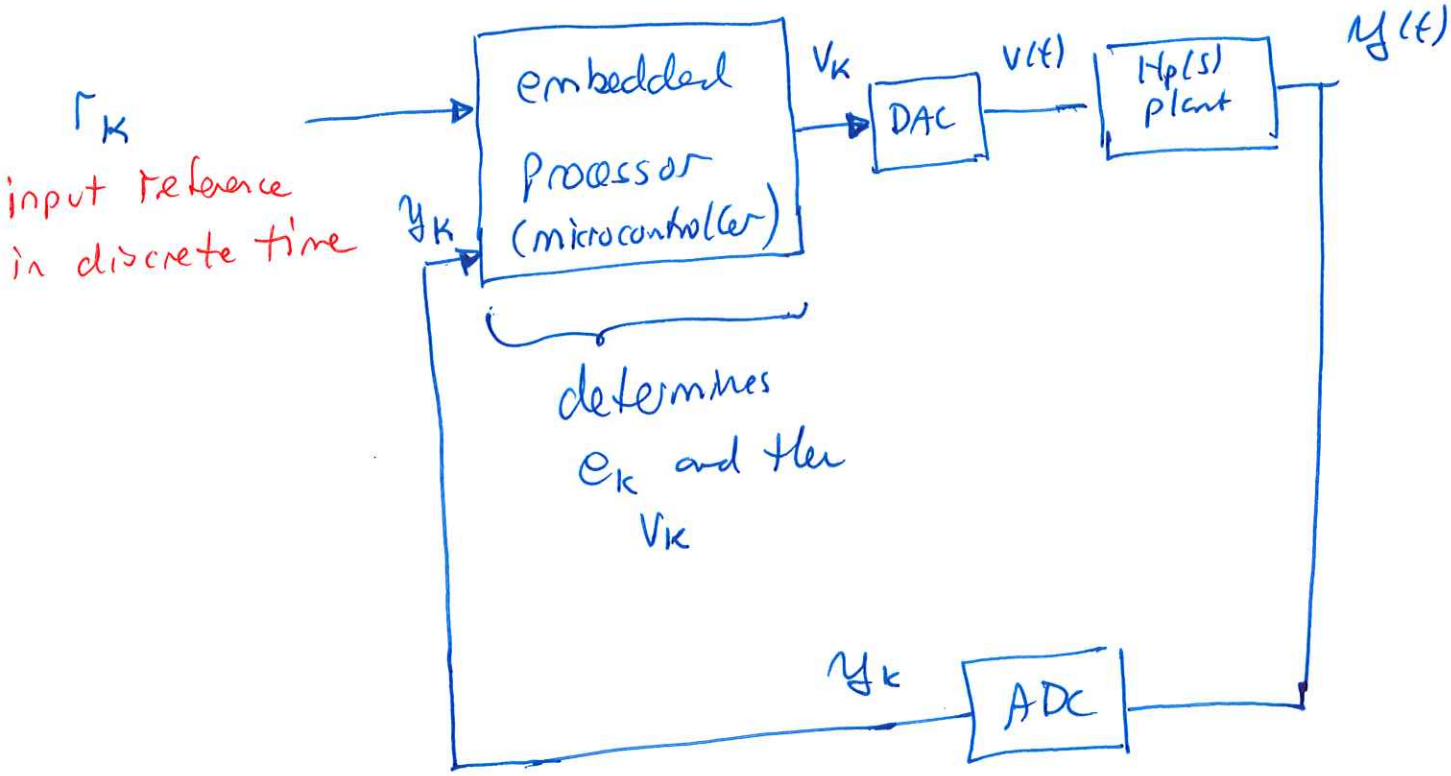
$$v(t) = v_{\text{floor}(t/T_s)}$$

eg. $T_s = 0.1$ seconds

$t = 1.01$ seconds

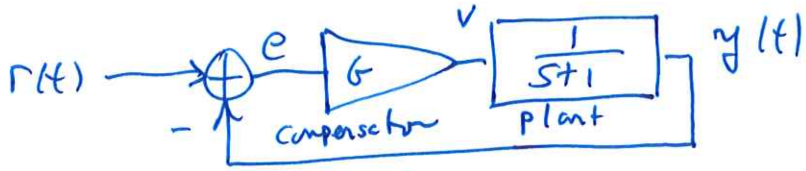
$$v(t) = v_{10}$$

Discrete time sampled control loop



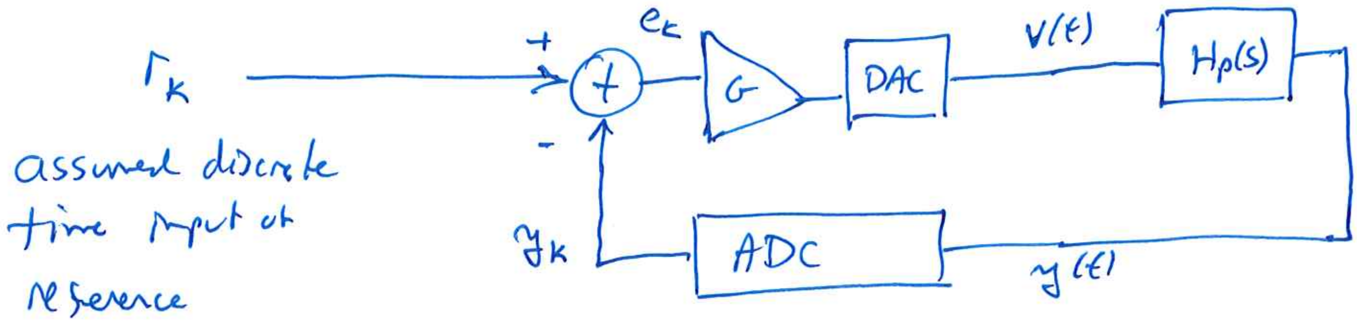
Mixed signal loop with discrete time signals in microcontroller, ADC, DAC. Continuous time signal in plant.

Digital Proportional Feedback loop example



Suppose we have basic feedback loop of proportional feedback as in this example.

We want to implement this with a digital processor

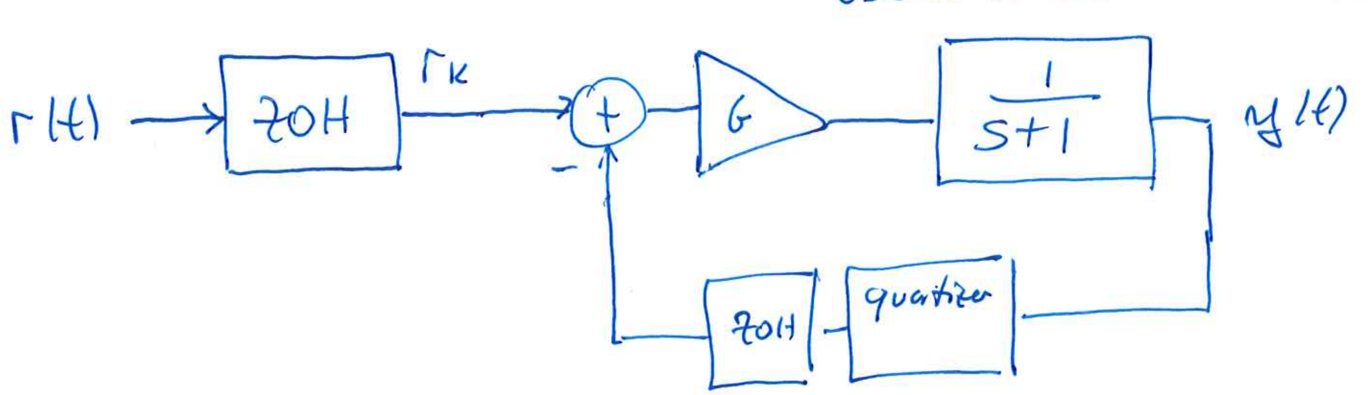


$$y_K = \max(\min(y(KT_s), V_{max}), V_{min})$$

$$e_K = r_K - y_K \quad , \quad V_K = G e_K$$

$$V(t) = G \, e_{floor\left(\frac{t}{T_s}\right)} = V_{floor\left(\frac{t}{T_s}\right)}$$

We can put this into a Simulink simulation as shown in the unit 5 notes.



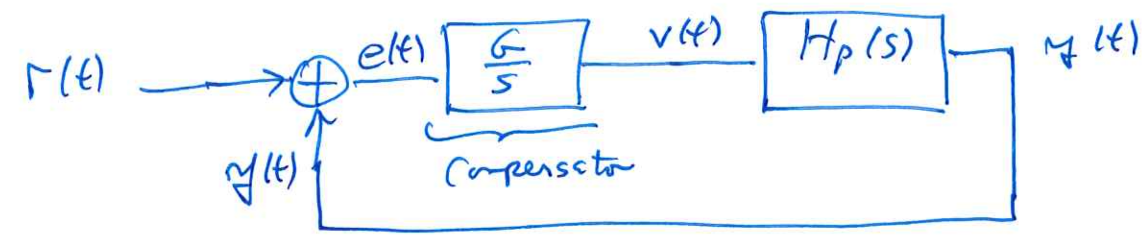
ZOH blocks are specified with a sampling time of T_s

In this simple example all the processor really does is the update of

$$v_k = G\,(r_k - y_k)$$

where r_k and y_k are inputs.

Consider next the compensator consisting of an integrator.



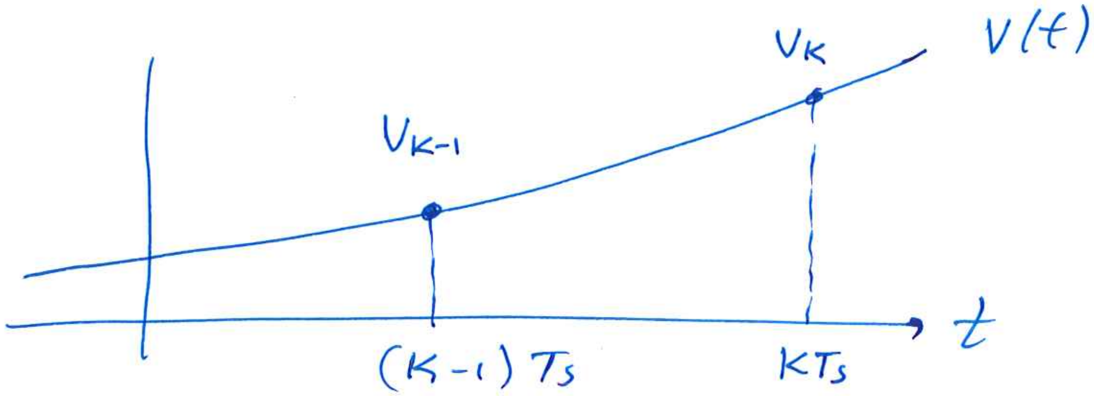
Converting into a discrete time compensator we need to implement

$$v(t) = G\int_0^t e(t')\,dt'$$

To convert this into a discrete time representation of the integral operation start with converting this into a D.E.Q. by taking derivative of both sides.

$$\frac{dv(t)}{dt} = G\,e(t)$$

Now consider an Euler approximation for the derivative of $v(t)$.



$$V_K = V(KT_s)$$

$$\left.\frac{dV}{dt}\right|_{t = KT_s} \approx \frac{V_K - V_{K-1}}{T_s}$$

Hence the recursion equation of

$$\frac{V_K - V_{K-1}}{T_s} = G\,e_K$$

Or $$V_K = V_{K-1} + T_s G\,e_K$$

Now we have the operation for the embedded controller to do.

$$e_k = r_k - y_k$$

$$V_k = V_{k-1} + T_s G e_k$$

which can be trivially implemented by the controller.

Now consider converting this into a z transform representation. The z-transform does not help in determining the microcontroller code but is useful for analysis of the loop.

Take the recursive equation $V_k = V_{k-1} + T_s G e_k$ and transform to z domain.

All we need is the identity of

$$V_k \Leftrightarrow V(z)$$

$$V_{k+n} \Leftrightarrow V(z) z^n$$

Hence $V(z) = V(z) z^{-1} + T_s G E(z)$

You may have to review some of your notes from previous courses regarding z transform identities.

if $v_k \Leftrightarrow V(z)$ this means

$$V(z) = \sum_{k=-\infty}^{\infty} v_k z^{-k}$$

note then that for v_{k+n}

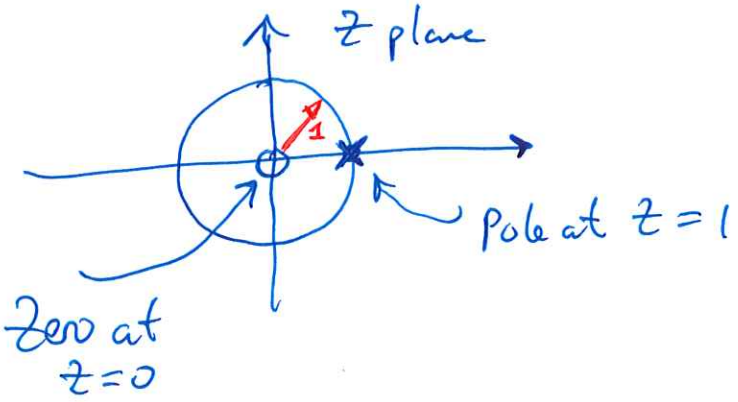
$$\sum_{k=-\infty}^{\infty} v_{k+n} z^{-k} = \underbrace{\sum_{k=-\infty}^{\infty} v_{k+n} z^{-(k+n)}}_{V(z)} z^{n}$$

$$= z^{n} V(z)$$

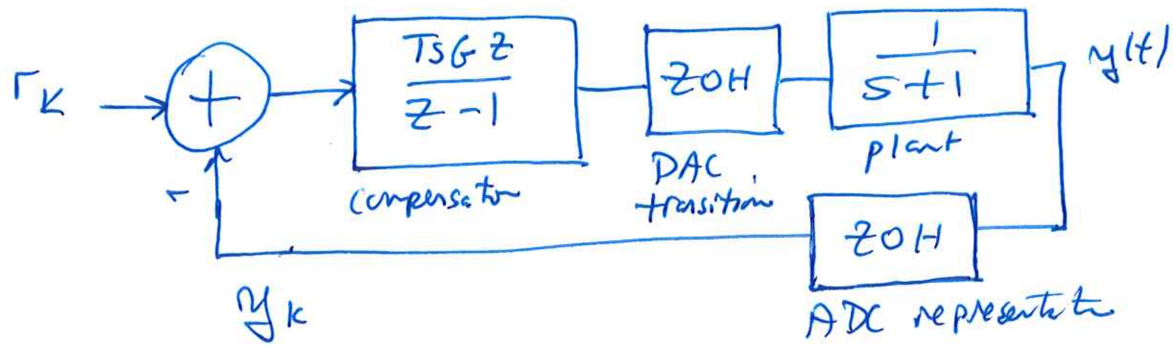
$\therefore \quad v_{k+n} \Leftrightarrow z^{n} V(z)$

So back to $V(z) = V(z) z^{-1} + T_s G E(z)$

we have $\dfrac{V(z)}{E(z)} = \dfrac{T_s G}{1 - z^{-1}} = \dfrac{T_s G z}{z - 1}$



Simulink model



To get a more accurate representation of the DAC and ADC operation we may want to add in limiters and quantization.

We also have a 'fixed-point' model of the processor evaluation of the difference equation which can be built into the Z transform representation.

Discrete Time approximation of DEQ

Note the DEQ in this case was

$$\frac{dv(t)}{dt} = G e(t)$$

which we approximated as

$$\frac{v_K - v_{K-1}}{T_S} = G e_K$$

But this is an approximation and not unique,

We could approximate as

$$\frac{V_K - V_{K-1}}{T_s} = G\, e_{K-1} \qquad \left(\begin{array}{l}\text{forward Euler instead}\\ \text{of backward Euler.}\end{array}\right)$$

In this case we are using a previous value of $e(t)$.
This adds additional delay and as discussed earlier
can lead to a less stable feedback loop.

For this case $\quad \dfrac{V(z)}{E(z)} = \dfrac{T_s G}{z-1}$

Single pole at $z = 1$ and no zero.
Having the delay is more realistic in that the
sensor data is always slightly delayed.

Matlab can compute the z transform directly from
the continuous time transfer function using <u>c2d( )</u>.
This is an extremely useful function for designing
discrete time control systems.

As an example, say we have

$$H(s) = \frac{1}{s}$$

Then the commands

```
H = tf(1, [1,0])
Ts = 0.1
Hd = c2d(H, Ts)
```

results in H_d as a discrete time Z transform of the integrator given as

$$H_d = \frac{0.1}{z-1}$$

c2d approximates the derivative as a forward Euler, and hence assumes the delay of the input.

This is because c2d assumes a default of 'ZOH' on the inputs. Other options are available.

For the present purposes, we will assume the default 'ZOH' option as being the most applicable, as it is realistic that the input is delayed by ZOH operation.

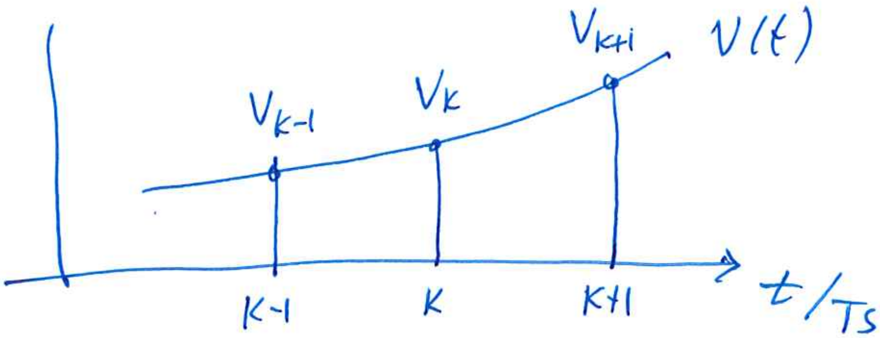
If Ts is small enough relative to the time constants of the poles in $H_c(s)$ and $H_p(s)$ then it does not really matter what method is used in c2D.

Note also that the conversion from the discrete time recursion equation for the compensator to the Z transform equivalent is only really necessary for analysis and simulation. What is implemented in the embedded processor is the difference equation.

If Ts is larger then we have to be careful how the derivatives are mapped into discrete time. As an example go back to the DEQ equation

$$\frac{dv(t)}{dt} = G\, e(t)$$

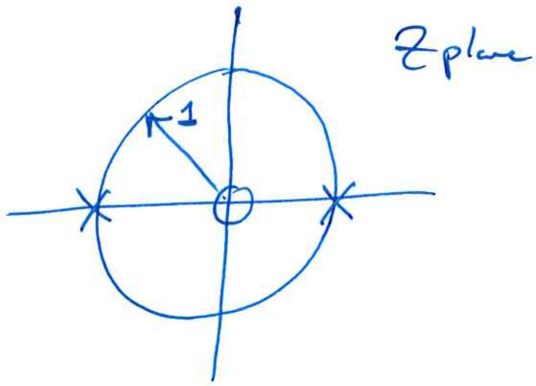
Suppose we use a central difference approximation



We can approximate as

$$\frac{V_{K+1} - V_{K-1}}{2T_s} = G e_K$$

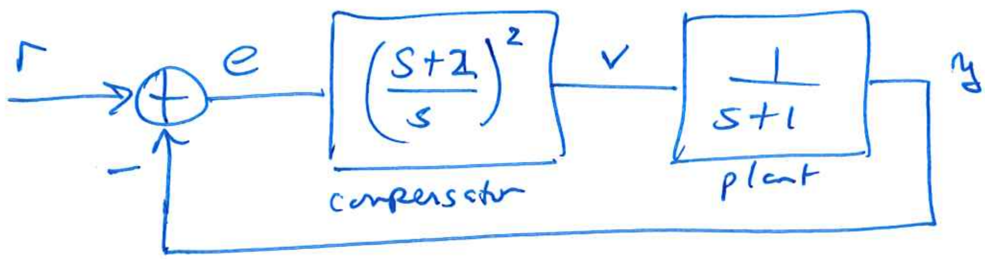
$$\frac{V(z)}{E(z)} = \frac{2T_s G}{z - z^{-1}} = \frac{2T_s G z}{z^2 - 1}$$



This is now a second order z transfer function with 2 poles and one zero. The additional pole at $z = -1$ does not influence the behavior for low frequency signals.

We will leave this point here but just be aware the mapping from continuous to discrete time is approximate and is therefore not unique.

Example convert to discrete time compensator and determine the recursive equation.



$$\frac{V(s)}{E(s)} = \frac{s^2+4s+4}{s^2}$$

$$V(s)\, s^2 = E(s)\, s^2 + 4E(s)\, s + 4\, E(s)$$

$$\frac{V_{k+1} - 2V_k + V_{k-1}}{T_s^2} = \frac{e_{k+1} - 2e_k + e_{k-1}}{T_s^2}$$

$$+ 4\, \frac{e_{k+1} - e_k}{T_s}$$

$$+ 4\, e_k$$

$$V_{k+1} = 2V_k - V_{k-1} + e_{k+1}\left(1 + 4T_s\right)$$

$$+ e_k\left(-2 - 4T_s + 4T_s^2\right)$$

$$+ e_{k-1}$$

What is the required difference recursive equation that can be directly implemented.

Note that the central difference was used here for the second derivative.

$$V(s)\, s^2 \Rightarrow \frac{d^2 v(t)}{dt^2}$$

$$\left.\frac{d^2 v(t)}{dt^2}\right|_{t=kT_s} \approx \frac{V_{k+1} - 2V_k + V_{k-1}}{T_s^2}$$

For this example we matched the derivatives of the difference equations.

C2d works differently using the approximation of an invariant impulse response with a ZOH for the DAC model.

Which model to use depends on the details of the problem.