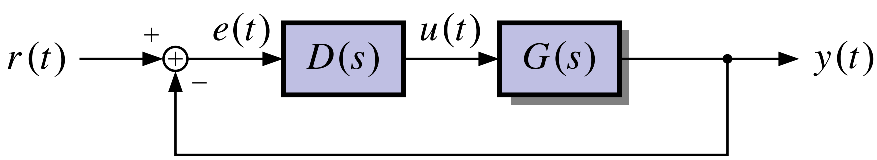
# DIGITAL CONTROLLER IMPLEMENTATION

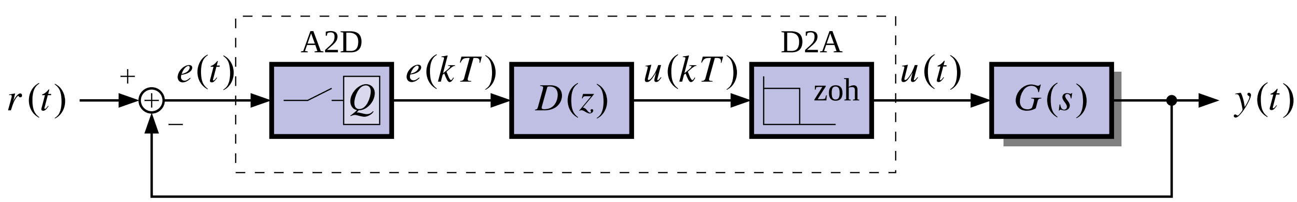
## 10.1: Some background in digital control

- We can implement our controllers with op-amp circuits (cf. Chap. 7).
- More commonly nowadays, we use digital computers (*e.g.*, microcontrollers) to implement our control designs.
- There are two main approaches to digital controller design:
  1. Emulation of an analog controller—we look at this here.
  2. Direct digital design—subject of more advanced course.
- Emulation is when a digital computer approximates an analog controller design.

**ANALOG:**




**DIGITAL:**




- Analog controller computes $u(t)$ from $e(t)$ using differential equations. For example,

$$\dot{u}(t) + bu(t) = k_0\dot{e}(t) + ak_0e(t).$$

- Digital controller computes $u(kT)$ from $e(kT)$ using difference equations. For example (we'll see where this came from shortly),

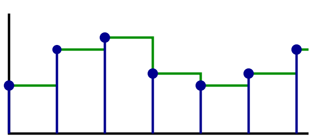
$$u(kT) = (1 - bT)u((k-1)T) + k_0(aT - 1)e((k-1)T) + k_0 e(kT).$$

- To interface the computer controller to the "real world" we need an analog-to-digital converter (to measure analog signals) and digital-to-analog converter (to output signals).
- Sampling and outputting usually done synchronously, at a constant rate. If sampling period $= T$, frequency $= 1/T$.
- The signals inside the computer (the sampled signals) are noted as $y(kT)$, or simply $y[k]$. $y[k]$ is a discrete-time signal, where $y(t)$ is a continuous-time signal.


- So, we can write the prior difference equation as

$$u[k] = (1 - bT)u[k-1] + k_0(aT - 1)e[k-1] + k_0 e[k].$$

- Discrete-time signals are usually converted to continuous-time signals using a zero-order hold:



*e.g.*, to convert $u[k]$ to $u(t)$.

## *Efficient implementation*

- We look at some efficient pseudo-code for an implementation of

$$u[k] = (1 - bT)u[k-1] + k_0(aT - 1)e[k-1] + k_0 e[k].$$

- Output of digital controller $u[k]$ depends on previous output $u[k-1]$ as well as the previous and current errors $e[k-1]$ and $e[k]$.

**Real-Time Controller Implementation**

$x = 0$. (initialization of "past" values for first loop through)

**Define constants:**

$\alpha_1 = 1 - bT$.

$\alpha_2 = k_0(aT - 1)$.

**READ** A/D to obtain $y[k]$ and $r[k]$.

$e[k] = r[k] - y[k]$.

$u[k] = x + k_0 e[k]$.

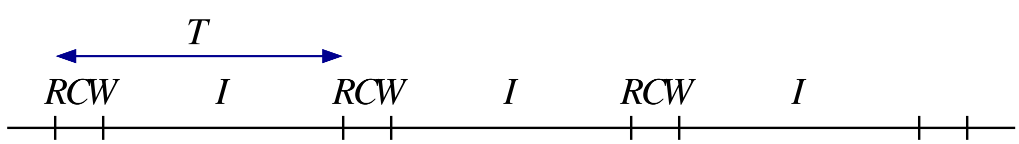
**OUTPUT** $u[k]$ to D/A and ZOH.

**Now compute $x$ for the next loop through:**

$x = \alpha_1 u[k] + \alpha_2 e[k]$.

**Go back to "READ" when $T$ seconds have elapsed since last READ.**

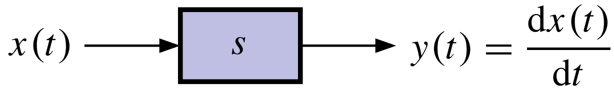
- Code is optimized to minimize latency between A2D read and D2A write.




$R$ = read. $W$ = write.
$C$ = compute. $I$ = idle.

## 10.2: "Digitization" (emulation of analog controllers)

- Continuous-time controllers are designed with Laplace-transform techniques. The resulting controller is a function of "$s$".




- So, "$s$" is a derivative operator. There are several ways of approximating this in discrete time.

*Forward-rectangular rule*

- We first look at the "forward rectangular" rule. We write:

$$\dot{x}(t) \triangleq \lim_{\delta t \to 0} \frac{\delta x(t)}{\delta t} = \lim_{\delta t \to 0} \frac{x(t+\delta t) - x(t)}{\delta t}.$$

- If sampling interval $T = t_{k+1} - t_k$ is small,[1]

$$\dot{x}(kT) \approx \frac{x((k+1)T) - x(kT)}{T} \qquad i.e., \qquad \dot{x}[k] \approx \frac{x[k+1] - x[k]}{T}.$$

*Backward-rectangular rule*

- We could also write $\dot{x}(t) \triangleq \lim_{\delta t \to 0} \frac{\delta x(t)}{\delta t} = \lim_{\delta t \to 0} \frac{x(t) - x(t-\delta t)}{\delta t}$.
- Then, if $T$ is small,

$$\dot{x}(kT) \approx \frac{x(T) - x((k-1)T)}{T} \qquad i.e., \qquad \dot{x}[k] \approx \frac{x[k] - x[k-1]}{T}.$$

*Bilinear (or Tustin) rule*

- We could also re-index the forward-rectangular rule as

$$\dot{x}[k-1] = \frac{x[k] - x[k-1]}{T}$$

to have the same right-hand-side as the backward-rectangular rule.

[1] Rule of thumb: Sampling frequency must be $\approx 30$ times the bandwidth of the analog system being emulated for comparable performance.

- Then, we average these two forms:

$$\frac{\dot{x}[k]+\dot{x}[k-1]}{2}=\frac{x[k]-x[k-1]}{T}.$$

*Digitizing a controller*

- Once we've chosen which rule to use, we "digitize" controller $D(s)$ by
  1. Writing $U(s)=D(s)E(s)$.
  2. Converting to differential equation: $\displaystyle\sum_{k=0}^{n}a_k\frac{\mathrm{d}^k u(t)}{\mathrm{d}t^k}=\sum_{k=0}^{m}b_k\frac{\mathrm{d}^k e(t)}{\mathrm{d}t^k}$.
  3. Replacing derivatives with differences.

**EXAMPLE:** Digitize the lead or lag controller $D(s)=\dfrac{U(s)}{E(s)}=k_0\dfrac{s+a}{s+b}$ using the forward-rectangular rule.

1. We write

$$\begin{aligned}U(s)&=k_0\frac{s+a}{s+b}E(s)\\(s+b)U(s)&=k_0(s+a)E(s)\end{aligned}$$

2. We take the inverse-Laplace transform of this result, term-by-term to get

$$\dot{u}(t)+bu(t)=k_0\dot{e}(t)+ak_0e(t).$$

3. Use "forward-rectangular rule" to digitize

$$\begin{aligned}\frac{u[k+1]-u[k]}{T}+bu[k]&=k_0\left(\frac{e[k+1]-e[k]}{T}+ae[k]\right)\\u[k+1]&=u[k]+\\&\quad T\left[-bu[k]+k_0\left(\frac{e[k+1]-e[k]}{T}+ae[k]\right)\right]\\&=(1-bT)u[k]+k_0(aT-1)e[k]+k_0e[k+1],\end{aligned}$$

or,

$$u[k] = (1 - bT)u[k-1] + k_0(aT-1)e[k-1] + k_0e[k].$$

- This is how we got the result at the beginning of this chapter of notes.

**EXAMPLE:** Digitize the lead or lag controller $D(s) = \dfrac{U(s)}{E(s)} = k_0\dfrac{s+a}{s+b}$ using the backward-rectangular rule.

1. As before, we have

$$(s+b)U(s) = k_0(s+a)E(s).$$

2. Again, we have

$$\dot{u}(t) + bu(t) = k_0\dot{e}(t) + ak_0e(t).$$

3. Use "backward-rectangular rule" to digitize

$$\begin{aligned}
\frac{u[k]-u[k-1]}{T} + bu[k] &= k_0\left(\frac{e[k]-e[k-1]}{T} + ae[k]\right) \\
u[k] &= u[k-1] + \\
&\qquad T\left[-bu[k] + k_0\left(\frac{e[k]-e[k-1]}{T} + ae[k]\right)\right] \\
(1+bT)\,u[k] &= u[k-1] + k_0(aT+1)e[k] - k_0e[k-1] \\
u[k] &= \frac{1}{1+bT}\left(u[k-1] + k_0(aT+1)e[k] - k_0e[k-1]\right).
\end{aligned}$$

- Notice that this is a different result from before.

**EXAMPLE:** Digitize the lead or lag controller $D(s) = \dfrac{U(s)}{E(s)} = k_0\dfrac{s+a}{s+b}$ using the bilinear rule.

1. As before, we have $(s+b)U(s) = k_0(s+a)E(s)$.
2. Again, we have $\dot{u}(t) + bu(t) = k_0\dot{e}(t) + ak_0e(t)$.

3. Using the bilinear rule is challenging since we need to have derivatives in a specific format. We'll use a trick here (ECE4540/5540 teaches more advanced techniques that don't need this trick).

- Re-index the differential equation:

$$\dot{u}(t-T) + bu(t-T) = k_0\dot{e}(t-T) + ak_0 e(t-T).$$

- Add this to the prior version, and divide by 2

$$\left[\frac{\dot{u}(t)+\dot{u}(t-T)}{2}\right] + b\left[\frac{u(t)+u(t-T)}{2}\right] = k_0\left[\frac{\dot{e}(t)+\dot{e}(t-T)}{2}\right] + ak_0\left[\frac{e(t)+e(t-T)}{2}\right]$$

$$\left[\frac{u[k]-u[k-1]}{T}\right] + \frac{b}{2}\left[u[k]+u[k-1]\right] = k_0\left[\frac{e[k]-e[k-1]}{T}\right] + \frac{ak_0}{2}\left[e[k]+e[k-1]\right].$$

- Rearranging,

$$\left(1+\frac{bT}{2}\right)u[k] = \left(1-\frac{bT}{2}\right)u[k-1] + k_0\left(1+\frac{aT}{2}\right)e[k] - k_0\left(1-\frac{aT}{2}\right)e[k-1]$$

$$u[k] = \frac{2-bT}{2+bT}u[k-1] + k_0\frac{2+aT}{2+bT}e[k] - k_0\frac{2-aT}{2+aT}e[k-1].$$

- Again, this is a different result from before.

## 10.3: The impact of the zero-order hold

- We illustrate the results of the prior three examples by substituting numeric values.
- Let $D(s) = 70\dfrac{(s+2)}{(s+10)}$, $\qquad G(s) = \dfrac{1}{s(s+1)}$.
- Choose to try a sample rate of 20 Hz and also try 40 Hz.
  (Note, BW of analog system is $\approx 1$Hz or so).

**FORWARD-RECTANGULAR RULE:** Digitizing $D(s)$ gives

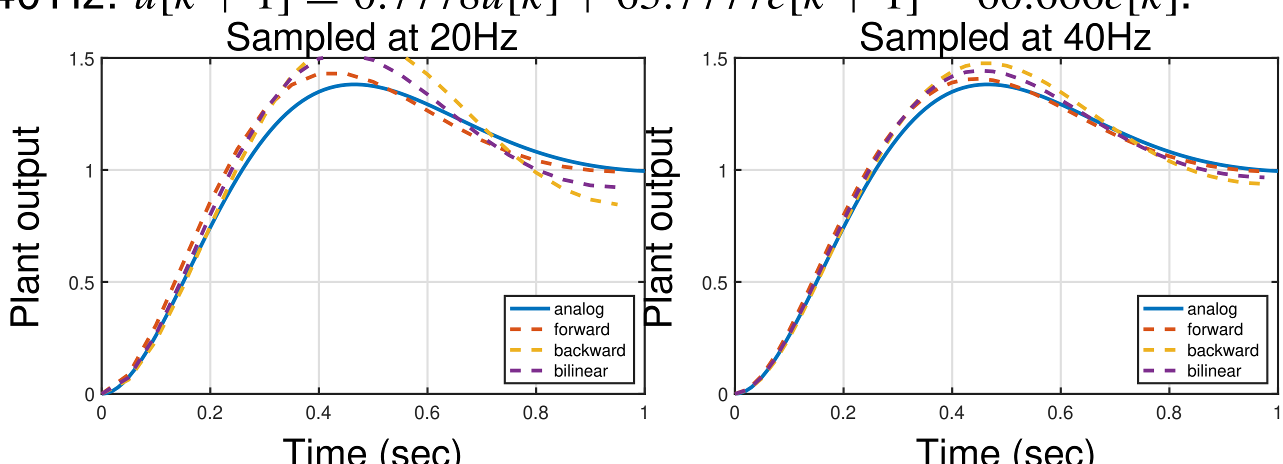
- 20 Hz: $u[k+1] = 0.5u[k] + 70e[k+1] - 63e[k]$.
- 40 Hz: $u[k+1] = 0.75u[k] + 70e[k+1] - 66.5e[k]$.

**BACKWARD-RECTANGULAR RULE:** Digitizing $D(s)$ gives

- 20 Hz: $u[k+1] = 0.6666u[k] + 51.3333e[k+1] - 46.6666e[k]$.
- 40 Hz: $u[k+1] = 0.8u[k] + 58.8e[k+1] - 56e[k]$.

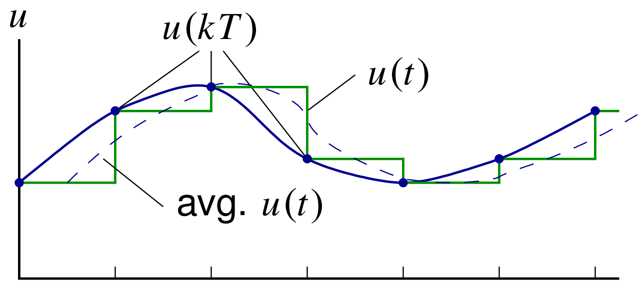
**BILINEAR RULE:** Digitizing $D(s)$ gives

- 20 Hz: $u[k+1] = 0.6u[k] + 58.8e[k+1] - 53.2e[k]$.
- 40 Hz: $u[k+1] = 0.7778u[k] + 63.7777e[k+1] - 60.666e[k]$.




- *IMPORTANT NOTE:* The digitized system has poorer damping than the original analog system. This will *always* be true when emulating an analog controller. We see why next . . .

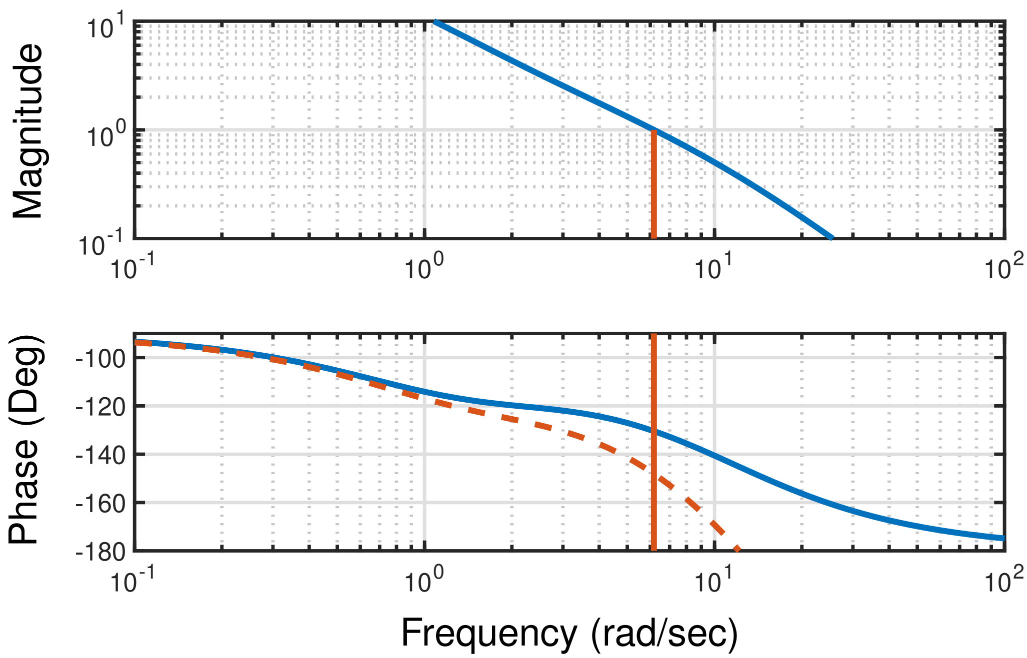
## The D2A "Hold" Operation




- Even if $u(kT)$ is a perfect re-creation of the output of the analog controller at $t = kT$, the "hold" in the D2A causes an "effective delay."
- The delay is approximately equal to half of the sampling period: $T/2$.
- Recall from frequency-response analysis and design, the magnitude of a delayed response stays the same, but the phase changes:

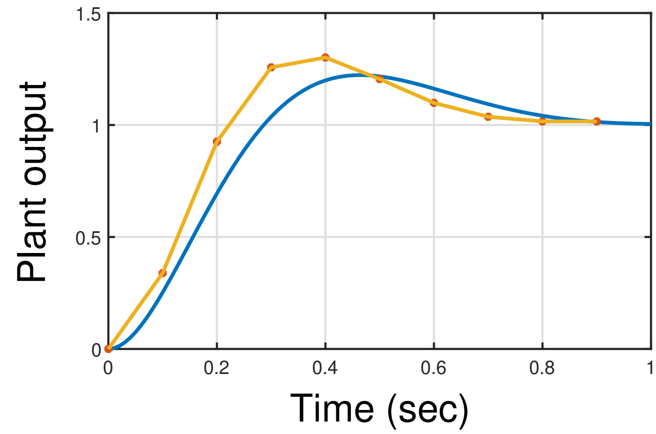
$$\Delta\text{phase} = -\omega\frac{T}{2}.$$

- For the previous example, sampling now at $10$Hz, we have:



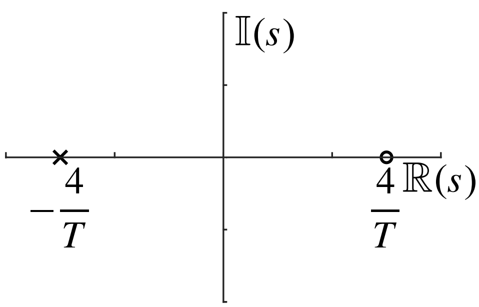

- The $PM$ has changed from $\approx 50°$ to $\approx 30°$.
- $\zeta \approx \dfrac{PM}{100}$ ... $\zeta$ changed from $0.5$ to $0.3$ ... much less damping.

- $M_p$ from about $20\%$ to about $30\%$ ...
- Faster sampling... smaller $T$ ... smaller delay... smaller change in response.



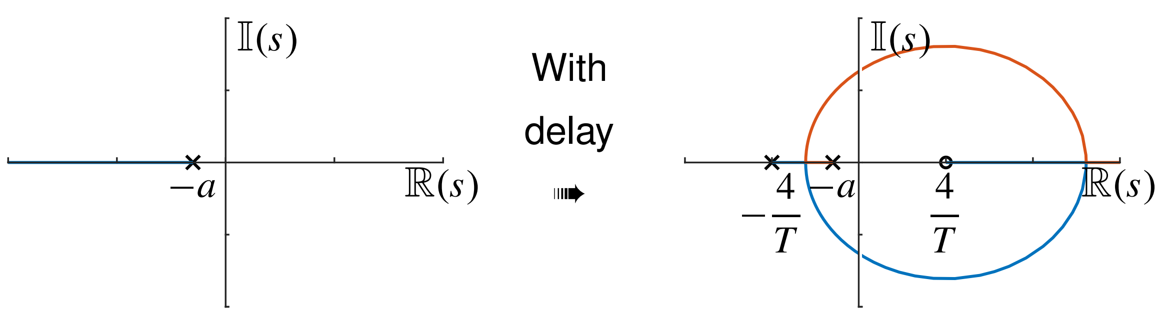

## Root-Locus View of the Delay

- Recall that we can model a delay using a Padé approximation.
- $\dfrac{T}{2}$ delay $\longrightarrow e^{-sT/2} \approx \dfrac{1 - sT/4}{1 + sT/4}$.
- Poles and zeros reflected about the $j\omega$-axis.




As $T \to 0$, delay dynamics $\to \infty$.

- Impact of delay: Suppose $D(s)G(s) = \dfrac{1}{s + a}$.




- Does the delayed locus make sense?

$$\frac{1 - sT/4}{1 + sT/4} = -\frac{(sT/4 - 1)}{(sT/4 + 1)}.$$

- Gain is negative! We need to draw a $0^\circ$ root locus, not the $180^\circ$ locus we are more familiar with.
- Conclusion: Delay destabilizes the system.

## PID Control via Emulation

$$\left.\begin{aligned} \text{P:}\quad u(t) &= Ke(t) \\ \text{I:}\quad u(t) &= \int_0^t \frac{K}{T_I} e(\tau)\,d\tau \\ \text{D:}\quad u(t) &= KT_D\dot{e}(t) \end{aligned}\right\} \begin{aligned} \text{PID:}\ u(t) &= K\left[e(t) + \frac{1}{T_I}\int_0^t e(\tau)\,d\tau + T_D\dot{e}(t)\right] \\ \text{or,}\quad \dot{u}(t) &= K\left[\dot{e}(t) + \frac{1}{T_I}e(t) + T_D\ddot{e}(t)\right]. \end{aligned}$$

- Convert to discrete-time (use fwd. rule twice for $\ddot{e}(t)$).

$$u[k] = u[k-1] + K\left[\left(1 + \frac{T}{T_I} + \frac{T_D}{T}\right)e[k] - \left(1 + \frac{2T_D}{T}\right)e[k-1] + \frac{T_D}{T}e[k-2]\right].$$

**EXAMPLE:**

$$G(s) = \frac{360000}{(s+60)(s+600)} \qquad K = 5,\ T_D = 0.0008,\ T_I = 0.003.$$

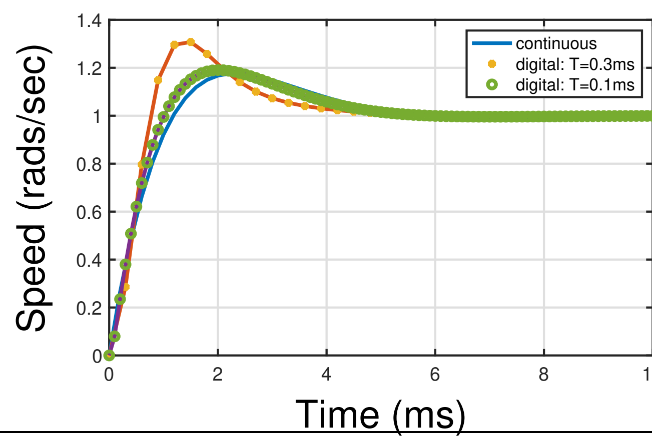
Bode plot of cts-time OL system $D(s)G(s)$ with the above PID $D(s)$ shows that BW $\approx 1800$ rad/sec, $\approx 320$Hz.

$$10 \times \text{BW} \quad \Longrightarrow \quad T = 0.3\,\text{ms}.$$

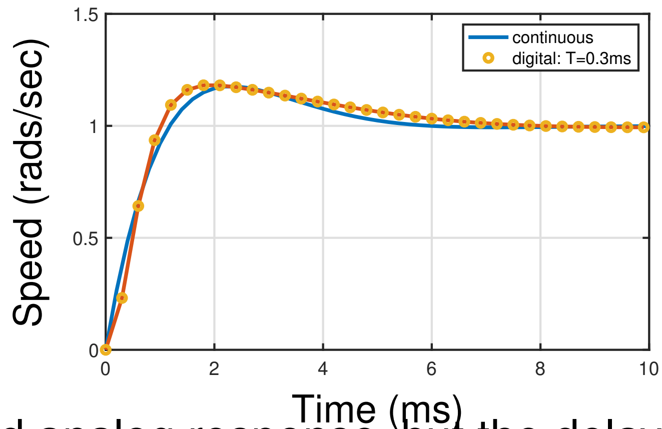
- From above,

$$u[k] = u[k-1] + 5\Big[3.7667e[k] - 6.333e[k-1] + 2.6667e[k-2]\Big].$$

- Step response plotted to the right.
- Performance not great, so tried again with $T = 0.1$ ms. Much better.



- Note, however, that the error is mostly due to the rise time being too fast, and the damping too low.
- *FIDDLE* with parameters
  ➠ increase $K$ to slow the system down; Increase $T_D$ to increase damping.
  ➠ New $K = 3.2$, new $T_D = 0.3$ ms.




**KEY POINT:** We can emulate a desired analog response, but the delay added to the system due to the D2A hold circuit will decrease damping. This could even destabilize the system!!!

- This delay can be minimized by sampling at a high rate (expensive).
- Or, we can change the digital controller parameters, as in the last example, to achieve the desired system performance BUT NOT BY emulating the specific analog controller $D(s)$.
- Hence the need for more advanced methods of digital control: ECE4540/5540.