There are Multiple Methods of Discretization, Including:

- **Backward difference**
  - **Approximates derivative**
- **Forward Difference**
  - **Approximates integral**
- **Combined Forward/Backward Difference**
  - **Better derivative estimate**
- **Bilinear transform**
  - **End point trapezoidal integration**
- **Impulse invariance**
  - **Sampled version of impulse response**
- **Transition Matrix Method**
  - **Solve first order matrix differential equation**
- **Bootstrap Method (overkill)**

  a. Use x_k and x_{k-1} to compute x'_k
  b. Use forward difference to compute x_{k+1}
  c. Use backward difference to compute x'_{k+1}
  d. Use bilinear transform to recompute x_{k+1}

 1

Multiple Methods of Discretization Provide Slightly Different Difference Equations

- **Backward difference** $\dot{y}_k = \frac{y_k - y_{k-1}}{T}$ $\quad s = \frac{1 - z^{-1}}{T}$ $\quad z = \frac{1}{1 - s \cdot T}$
- **Forward Difference** $\dot{y}_k = \frac{y_{k+1} - y_k}{T}$ $\quad s = \frac{z - 1}{T}$ $\quad z = 1 + s \cdot T$
- **Combined Forward/Backward** $\dot{y}_k = \frac{y_{k+1} - y_{k-1}}{2 \cdot T}$ $\quad s = \frac{z - z^{-1}}{2 \cdot T}$
- **Bilinear transform** $\frac{\dot{y}_{k+1} + \dot{y}_k}{2} = \frac{y_{k+1} - y_k}{T}$ $\quad s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$ $\quad z = \frac{1 + \frac{T}{2} s}{1 - \frac{T}{2} s}$
- **Impulse invariance** $z = e^{s \cdot T}$
- **Transition Matrix Method** $\dot{y} = A \cdot y(t) + B \cdot x(t)$

$$y_{k+1} = e^{A \cdot T} \cdot y_k + \int_0^T e^{A \cdot (T - t)} \cdot B \cdot x(k \cdot T + t) \cdot dt = \Phi \cdot y_k + \Gamma_k$$

 2

P.O. Box 2875 • El Segundo, CA 90245 • (310) 607-9609 • Web: www.hasys.com

Example: A simple Integrator

- **Backward difference** $y_k = y_{k-1} + T \cdot x_k$
- **Forward Difference** $y_k = y_{k-1} + T \cdot x_{k-1}$
- **Bilinear transform** $y_k = y_{k-1} + \frac{T}{2} \cdot (x_k + x_{k-1})$
- **This can lead to:**
  - **Time wasting arguments**
  - **Confusing requirements**
  - **Questions about which transformation to use for each compensator**
  - **Incorporation of transformation into delivered code**
  - **Software design and coding errors**
  - **Wasted time in integration**

 3

Bilinear Transform Produces All First Order Compensators and Filters from the Same General Model

Continuous Time Transfer Function

$$H(s) = k \cdot \frac{\tau_z \cdot s + u_z}{\tau_p \cdot s + u_p} \quad \text{where: } \begin{array}{l} \tau_z, \tau_p \ge 0 \\ u_z, u_p = [0 \text{ or } 1] \\ k \ne 0 \end{array}$$

Bilinear Transformation

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \quad \text{where: } T = \text{update time}$$

Discrete Time Transfer Function

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1}}{1 - a_1 \cdot z^{-1}} \quad \text{where: } \begin{array}{l} a_1 = \frac{2 \cdot \tau_p - u_p \cdot T_s}{2 \cdot \tau_p + u_p \cdot T_s} \\ b_0 = k \cdot \frac{2 \cdot \tau_z + u_z \cdot T_s}{2 \cdot \tau_p + u_p \cdot T_s} \\ b_1 = -k \cdot \frac{2 \cdot \tau_z - u_z \cdot T_s}{2 \cdot \tau_p + u_p \cdot T_s} \end{array}$$

Difference Equation

$$y_n = a_1 \cdot y_{n-1} + b_0 \cdot x_n + b_1 \cdot x_{n-1}$$

 4

P.O. Box 2875 • El Segundo, CA 90245 • (310) 607-9609 • Web: www.hasys.com

Same Difference Equation Provides 6 Different First Order Compensators Depending on Model Parameters

Type:	k	τ_z	u_z	τ_p	u_p
Lead Compensation	≠ 0	> τ_p	1	> 0	1
Lag Compensation	≠ 0	> 0	1	> τ_z	1
Low Pass Filter (LPF)	≠ 0	= 0	1	> 0	1
Differential LPF	≠ 0	= 1	0	> 0	1
Integrator	≠ 0	= 0	1	1	0
Proportional plus Integral (PI)	= k_I ≠ 0	k_P/k_I ≠ 0	1	1	0

 5

Clamping and Preset is Accomplished by Modifying the Stored State Regardless of Type

- Accomplished by Changing the Stored Previous State Before Next Iteration

$$y_n = a_1 \cdot y_{n-1} + b_0 \cdot x_n + b_1 \cdot x_{n-1}$$

- Clamping:
  - Clamp y_n before output and save as previous value
  - Output is clamped
  - On next iteration, previous state is clamped (no build-up)
  - For PI, Equivalent to separating out Integrator and clamping its state
- Preset:
  - Compute initialization value
  - Store as previous state before first or next execution

 6