# Linear Constant-Coefficient Difference Equations

Maxx Seminario

University of Nebraska-Lincoln

September 3, 2025





## Introduction to Difference Equations

- **Important Class of LTI Systems**: Systems where input $x[n]$ and output $y[n]$ satisfy a difference equation
- **General Form**:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m]$$

- **Parameters**:
  - $N$: Order of the system (highest delay in output)
  - $a_k$: Output coefficients (constant)
  - $b_m$: Input coefficients (constant)
  - $M$: Highest delay in input terms
- **Why Important?**:
  - Provides computational algorithms for LTI systems
  - Foundation for digital filter implementation
  - Connects time-domain and system analysis

## Example 1: The Accumulator System

**Problem**: Find the difference equation for the accumulator system.
**System Definition**:

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

**Approach**:

- Rewrite the sum to separate current and past inputs
- Use the relationship between $y[n]$ and $y[n-1]$





## Example 1: Accumulator Solution

**Step 1**: Rewrite the accumulator equation:

$$y[n] = x[n] + \sum_{k=-\infty}^{n-1} x[k]$$

**Step 2**: Recognize that the sum is $y[n-1]$:

$$y[n-1] = \sum_{k=-\infty}^{n-1} x[k]$$

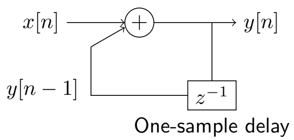
**Step 3**: Substitute to get the difference equation:

$$y[n] = x[n] + y[n-1]$$

**Standard Form**:

$$y[n] - y[n-1] = x[n]$$

# Block Diagram: Recursive Accumulator




**Recursive Implementation**:

- Each output value computed using previously computed values
- $y[n] = x[n] + y[n-1]$: Add current input to previous output
- Requires initial condition (e.g., $y[-1] = 0$)





# Example 2: Moving Average System

**Problem**: Find difference equation for causal moving average system.

**System Definition** (with $M_1 = 0$, so the system is causal):

$$y[n] = \frac{1}{M_2 + 1} \sum_{k=0}^{M_2} x[n-k]$$

**Two Approaches**:

1. **Direct (Non-recursive)**: Use convolution form directly
2. **Recursive**: Express as cascade of simpler systems

ECEN 463/863

Maxx Seminario

## Moving Average: Direct Implementation

**Direct Form**:

$$y[n] = \frac{1}{M_2+1}\sum_{k=0}^{M_2} x[n-k]$$

**Standard Form**:

$$y[n] = \sum_{k=0}^{M_2} \frac{1}{M_2+1} x[n-k]$$

**Disadvantages of Direct Implementation**:

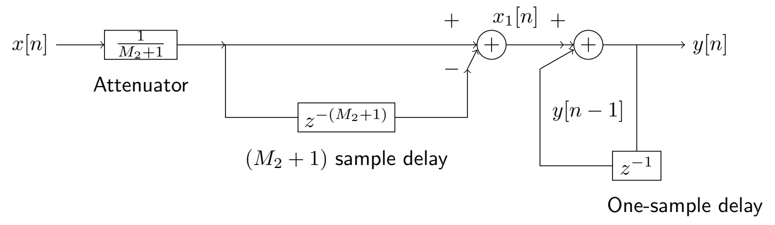
- Requires $(M_2+1)$ multiplications per output sample
- Must store $(M_2+1)$ input samples in memory
- $O(M_2)$ : Computational cost grows linearly with window size $M_2$



ECEN 463/863

Maxx Seminario

## Block Diagram: Recursive Moving Average




**Signal Flow**:

- Input $x[n] \to$ Attenuator $\frac{1}{M_2+1}$
- Attenuated signal splits: direct path and $(M_2+1)$ sample delay
- Sum: $x_1[n] = \frac{1}{M_2+1}[x[n] - x[n-(M_2+1)]]$
- $x_1[n] \to$ Accumulator $\to$ Output $y[n]$

## Moving Average as Difference Equation

ECEN 463/863

Maxx Seminario

**Intermediate signal**:

$$x_1[n] = \frac{1}{M_2+1}[x[n] - x[n-M_2-1]]$$

**Accumulator relation**: $y[n] = x_1[n] + y[n-1]$

**Final recursive form**:

$$y[n] - y[n-1] = \frac{1}{M_2+1}[x[n] - x[n-M_2-1]]$$

Note: There is an unlimited number of distinct difference equations to represent an LTI I/O relation.



## General Solution of Difference Equations

ECEN 463/863

Maxx Seminario

**Key Issue**: Difference equation alone does not uniquely specify output!

**General Solution Structure**:

$$y[n] = y_p[n] + y_h[n]$$

- $y_p[n]$: Particular solution (satisfies original equation)
- $y_h[n]$: Homogeneous solution (satisfies homogeneous equation)

**Homogeneous Equation (x[n] = 0)**:

$$\sum_{k=0}^{N} a_k y_h[n-k] = 0$$

**Homogeneous Solution Form**:

$$y_h[n] = \sum_{m=1}^{N} A_m z_m^n$$

where $z_m$ are roots of characteristic polynomial $A(z) = \sum_{k=0}^{N} a_k z^{-k} = 0$

## Auxiliary Conditions

**Need for Auxiliary Conditions**:

- $N$ undetermined coefficients $A_m$ in homogeneous solution
- Need $N$ auxiliary (boundary) conditions for unique solution

**Types of Auxiliary Conditions**:

1. **Fixed Values**: Specify $y[-1], y[-2], \ldots, y[-N]$
2. **Initial Rest**: If $x[n] = 0$ for $n < n_0$, then $y[n] = 0$ for $n < n_0$





## Forward and Backward Computation for Specific Class of Difference Equations

**Specific Class of Difference Equations**:

- Inputs $x[n]$ and Outputs $y[n]$ satisfy an $N$th-order linear constant coefficient difference equation:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$$

$$a_0 y[n] + a_1 y[n-1] + \cdots + a_N y[n-N] = b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M]$$

**Recursive Computation** (forward):

$$y[n] = -\sum_{k=1}^{N} \frac{a_k}{a_0} y[n-k] + \sum_{k=0}^{M} \frac{b_k}{a_0} x[n-k]$$

**Recursive Computation** (backward):

$$y[n-N] = -\sum_{k=0}^{N-1} \frac{a_k}{a_N} y[n-k] + \sum_{k=0}^{M} \frac{b_k}{a_N} x[n-k]$$

## Example 3: First-Order Difference Equation



**Problem**: Solve the difference equation with given input and initial condition.

**Difference Equation**:

$$y[n] = ay[n-1] + x[n]$$

**Input**:

$$x[n] = K\delta[n]$$

**Auxiliary Condition**:

$$y[-1] = c$$

**Find**: The complete solution $y[n]$ for all $n$.



## Example 3: Solution for $n > 0$



**Method**: Use recursive computation starting from $n = 0$.

**For** $n = 0$:

$$\begin{aligned} y[0] &= ay[-1] + x[0] && (1)\\ &= ay[-1] + K\delta[0] && (2)\\ &= ac + K \cdot 1 && (3)\\ &= ac + K && (4) \end{aligned}$$

**For** $n = 1$:

$$\begin{aligned} y[1] &= ay[0] + x[1] && (5)\\ &= a(ac + K) + K\delta[1] && (6)\\ &= a^2c + aK + K \cdot 0 && (7)\\ &= a^2c + aK && (8) \end{aligned}$$

# Example 3: Solution for $n > 0$



**For** $n = 2$:

$$y[2] = ay[1] + x[2] \tag{9}$$
$$= a(a^2c + aK) + K\delta[2] \tag{10}$$
$$= a^3c + a^2K + 0 \tag{11}$$
$$= a^3c + a^2K \tag{12}$$

**For** $n = 3$:

$$y[3] = ay[2] + x[3] \tag{13}$$
$$= a(a^3c + a^2K) + 0 \tag{14}$$
$$= a^4c + a^3K \tag{15}$$

**Pattern Recognition**: For $n \geq 0$: $y[n] = a^{n+1}c + a^nK$



# Example 3: Solution for $n < 0$



**For** $n < 0$: Use backward recursion.
**Rearrange difference equation**:

$$y[n-1] = \frac{y[n] - x[n]}{a}$$

**For** $n = 0$ **(find** $y[-1]$**)**:

$$y[-1] = \frac{y[0] - x[0]}{a} = \frac{(ac + K) - K}{a} = \frac{ac}{a} = c$$

This confirms our auxiliary condition
**For** $n = -1$ **(find** $y[-2]$**)**:

$$y[-2] = \frac{y[-1] - x[-1]}{a} = \frac{c - 0}{a} = \frac{c}{a}$$

**General pattern for** $n < 0$:

$$y[n] = ca^{n+1} = \frac{c}{a^{n+1}}$$

# Example 3: Complete Solution

**Final Answer**:

$$y[n] = \begin{cases} ca^{n+1}, & n < 0 \\ a^n(ac + K), & n \geq 0 \end{cases}$$

**Alternative Compact Form**:

$$y[n] = ca^{n+1} + Ka^n u[n]$$

where $u[n]$ is the unit step function.

**Physical Interpretation**:

- **Homogeneous part**: $ca^{n+1}$ (due to initial condition)
- **Forced response**: $Ka^n u[n]$ (due to impulse input)
- **System behavior**: Exponential with base $a$
- **Stability**: System stable if $|a| < 1$



# Summary: Linear Constant-Coefficient Difference Equations

- **Definition**:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m]$$

  Describes the relationship between input $x[n]$ and output $y[n]$ in LTI systems.
- **Key Components**:
  - $N$ and $M$: Order of the system
  - $a_k, b_m$: Constant coefficients
  - Recursive and direct computation methods
- **General Solution**:

$$y[n] = y_p[n] + y_h[n]$$

  - $y_p[n]$: Particular solution (due to input $x[n]$)
  - $y_h[n]$: Homogeneous solution (due to initial conditions)
  - Auxiliary conditions: Necessary to determine a unique / particular solutions.
- **Applications**: Digital filters, computational algorithms for LTI systems, and time-domain analysis.