# Signal Flow Graphs Revision

It is common to represent digital system signal processing routines as a visual **signal flow** graphs.

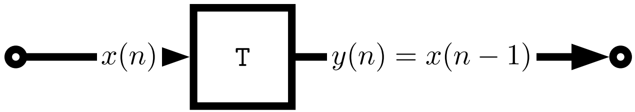
We use a simple *equation* relation to describe the algorithm.

We will need to consider *three* basic components:

- Delay
- Multiplication
- Summation

# Signal Flow Graphs: Delay

- We represent a delay of one sampling interval by a block with a `T` label:




- We describe the algorithm via the equation: $y(n) = x(n-1)$

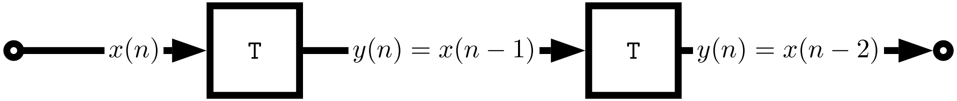
# Signal Flow Graphs: Delay Example

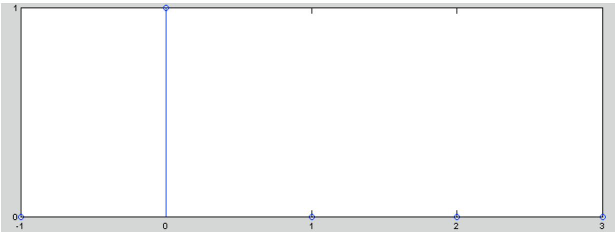
A delay of the input signal by **two** sampling intervals:

- We can describe the **algorithm** by:

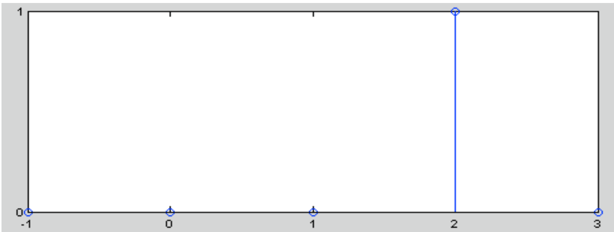
$$y(n) = x(n-2)$$

- We can use the block diagram to represent the **signal flow graph** as:



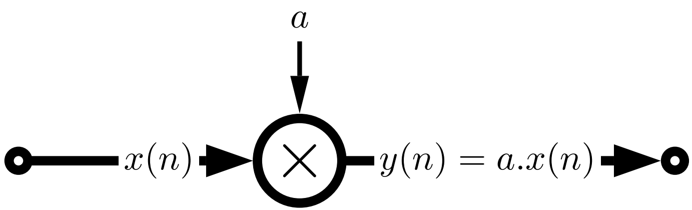

$x(n)$



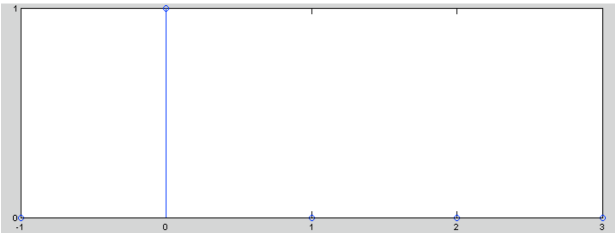

$y(n) = x(n-2)$

# Signal Flow Graphs: Multiplication

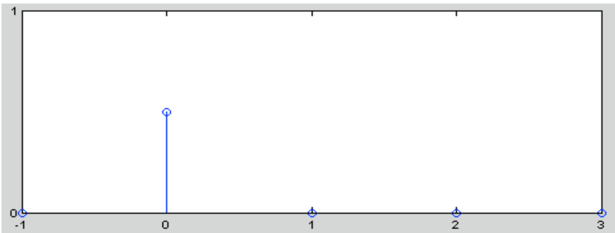
- We represent a multiplication or weighting of the input signal by a circle with a $\times$ label.
- We describe the algorithm via the equation: $y(n) = a.x(n)$




$e.g.\ \ a = 0.5$




$x(n)$

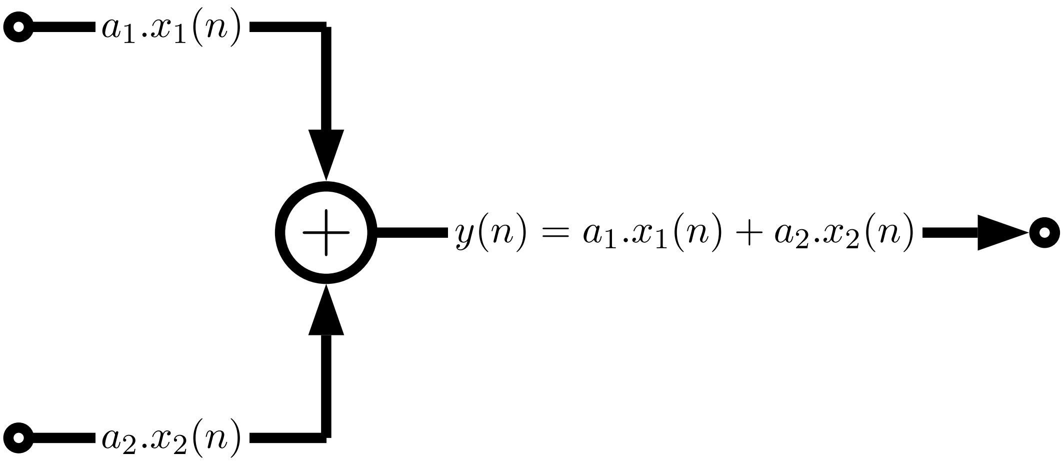



$y(n) = 0.5x(n)$

# Signal Flow Graphs: Addition

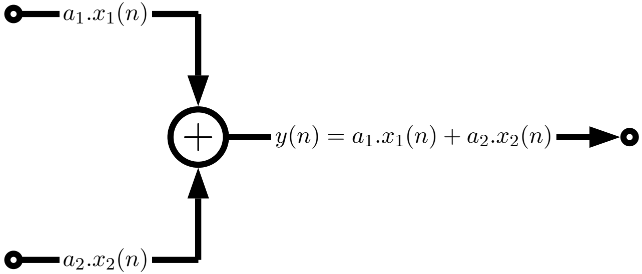
- We represent a addition of two input signal by a circle with a $+$ label.
- We describe the algorithm via the equation:

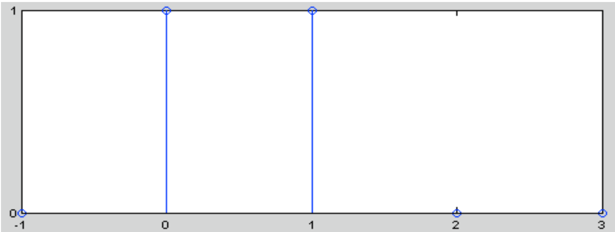
$$y(n) = a_1.x_1(n) + a_2.x_2(n)$$



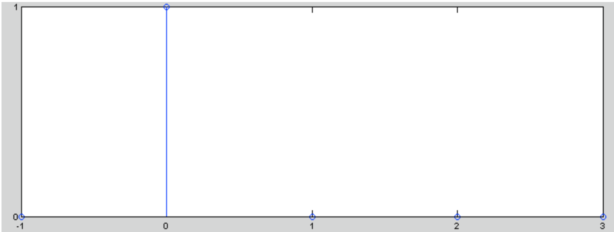
# Signal Flow Graphs: Addition Example

In the example, set $a_1 = a_2 = 1$:

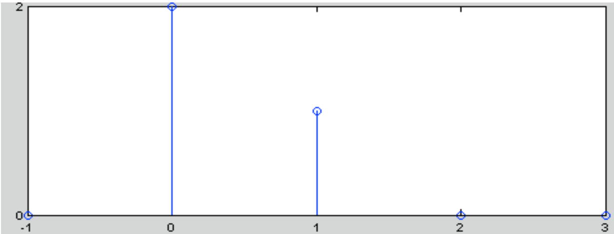



$x_1(n)$




$x_2(n)$



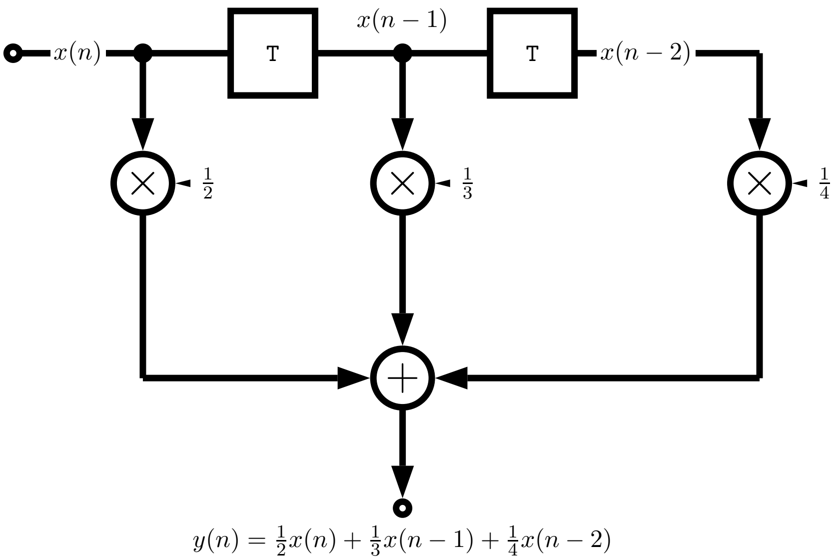

$$y(n) = x_1(n) + x_2(n)$$

# Signal Flow Graphs: Complete Example

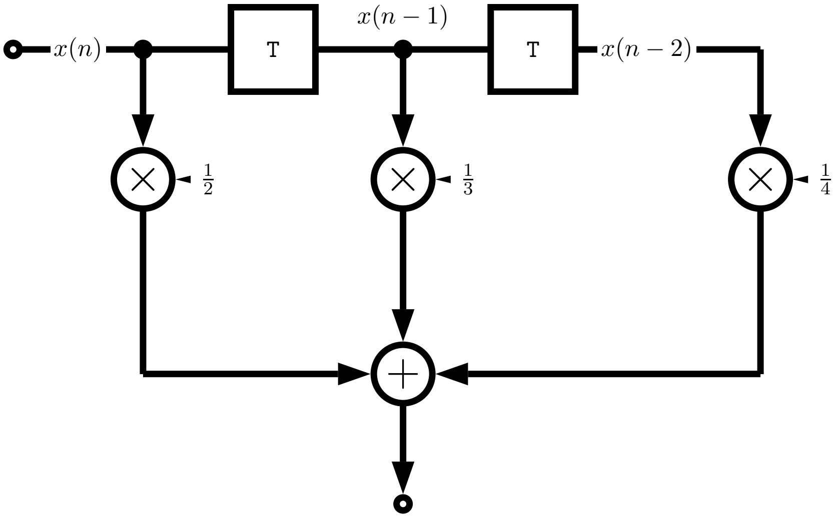
We can combine all above algorithms to build up more complex algorithms:

$$y(n) = \frac{1}{2}x(n) + \frac{1}{3}x(n-1) + \frac{1}{4}x(n-2)$$

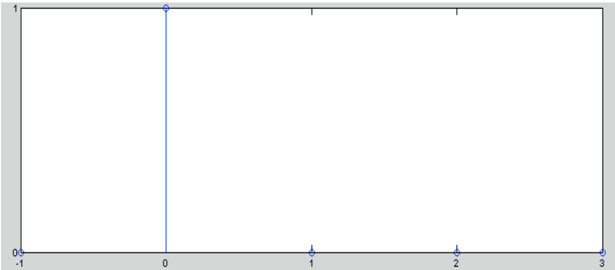
- This has the following signal flow graph:



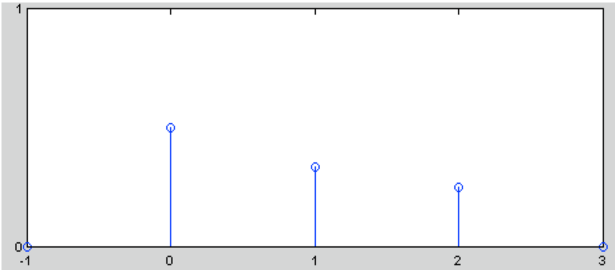
# Signal Flow Graphs: Complete Example Impulse Response




$$y(n) = \tfrac{1}{2}x(n) + \tfrac{1}{3}x(n-1) + \tfrac{1}{4}x(n-2)$$




$x(n)$




$$y(n) = \tfrac{1}{2}x(n) + \tfrac{1}{3}x(n-1) + \tfrac{1}{4}x(n-2)$$

# Hints for Constructing Signal Flow Graphs

Apart from the three basic building blocks of *Delay, Addition and Multiplication* there are two other tools that we can exploit:

- Feedback loops — merged back with *Delay, Addition and/or Multiplication*.

  Frequently (In many of our examples) we tap the output $y(n)$ and then delay *etc.* this.

  - $y(n-1)$ *etc.* then appears in the equation (right hand side), $y(n)$ on left hand side.

- Subproblem — break problem into smaller Signal flow graph components. **Useful for larger problems**

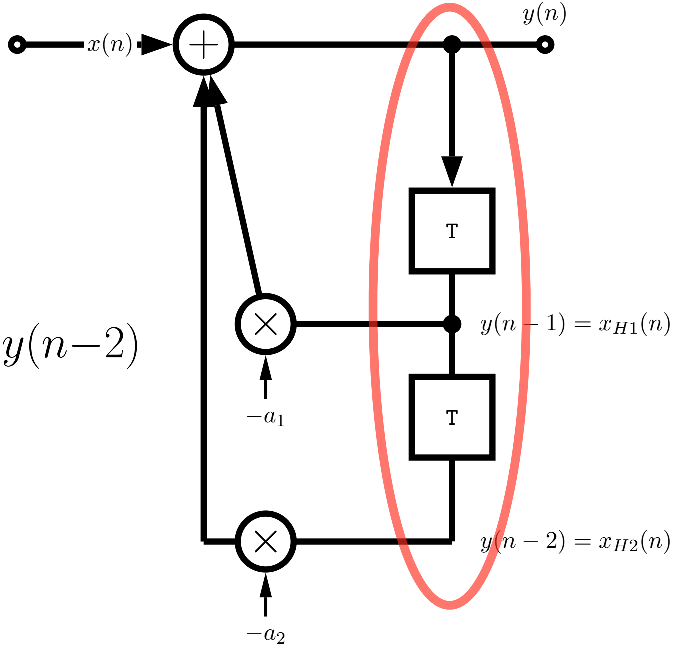
# Simple Feedback Loop Example

(Simple IIR Filter)

- The algorithm is represented by the difference equation:

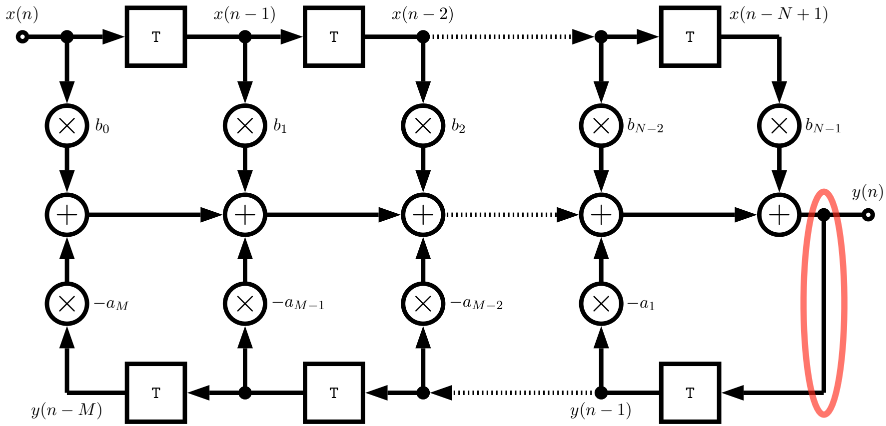
$$y(n) = x(n) - a_1 y(n-1) - a_2 y(n-2)$$

- This produces the opposite signal flow graph



# More Complex Feedback Loop Example

(General IIR Filter)




We can represent the IIR system algorithm by the difference equation:

$$y(n) = -\sum_{k=1}^{M} a_k\, y(n-k) + \sum_{k=0}^{N-1} b_k\, x(n-k)$$

# Signal Flow Graph Problem Decomposition

(Shelving Filter)

$$
\begin{aligned}
y_1(n) &= a_{B/C}x(n) + x(n-1) - a_{B/C}y_1(n-1) \\
y(n) &= \frac{H_0}{2}(x(n) \pm y_1(n)) + x(n)
\end{aligned}
$$

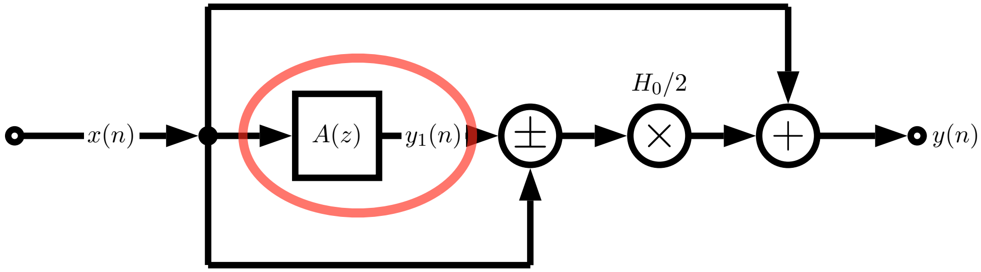
The gain, $G$, in dB can be adjusted accordingly:

$$H_0 = V_0 - 1 \text{ where } V_0 = 10^{G/20}$$

and the cut-off frequency for **boost**, $a_B$, or **cut**, $a_C$ are given by:

$$
\begin{aligned}
a_B &= \frac{tan(2\pi f_c/f_s) - 1}{tan(2\pi f_c/f_s) + 1} \\
a_C &= \frac{tan(2\pi f_c/f_s) - V_0}{tan(2\pi f_c/f_s) - V_0}
\end{aligned}
$$

# Shelving Filters Signal Flow Graph




where $A(z)$ is given by: