Circular Convolution in MATLAB

# Introduction

Circular convolution is used in digital signal processing when signals are considered periodic. It is especially important in applications involving the Discrete Fourier Transform (DFT).

# Mathematical Definition

Given two sequences x[n] and h[n] of length N, their circular convolution y[n] is defined as:  
  
 y[n] = ∑ x[k] \* h[(n - k) mod N] for k = 0 to N-1  
  
This assumes both sequences are of the same length N (padded with zeros if necessary).

# MATLAB Code for Circular Convolution

Below is the MATLAB code to perform circular convolution manually:

clc; clear;  
  
% Input sequences  
x = [1 2 3]; % Input signal  
h = [4 5 6]; % Impulse response  
  
% Make both sequences of equal length by zero-padding  
N = max(length(x), length(h));  
x = [x, zeros(1, N - length(x))];  
h = [h, zeros(1, N - length(h))];  
  
% Circular convolution using the definition  
y = zeros(1, N);  
for n = 1:N  
 for k = 1:N  
 index = mod(n - k, N);  
 if index == 0  
 index = N;  
 end  
 y(n) = y(n) + x(k) \* h(index);  
 end  
end  
  
% Display results  
disp('Input signal x[n]:');  
disp(x);  
disp('Impulse response h[n]:');  
disp(h);  
disp('Output y[n] = Circular Convolution of x[n] and h[n]:');  
disp(y);  
  
% Plot the results  
n = 0:N-1;  
subplot(3,1,1);  
stem(n, x, 'filled'); title('Input Signal x[n]'); xlabel('n'); ylabel('x[n]'); grid on;  
  
subplot(3,1,2);  
stem(n, h, 'filled'); title('Impulse Response h[n]'); xlabel('n'); ylabel('h[n]'); grid on;  
  
subplot(3,1,3);  
stem(n, y, 'filled'); title('Circular Convolution y[n]'); xlabel('n'); ylabel('y[n]'); grid on;

# Example Output

Given:  
 x = [1 2 3]  
 h = [4 5 6]  
  
After circular convolution, the result is:  
 y = [32 31 28]