

Experiment no. : 1

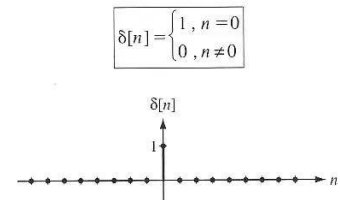
Experiment name: Generate unit impulse, unit step and ramp signal and convert one to another in MATLAB.

Theory:

Unit Impulse Signal ($\delta[n]$)

The unit impulse signal is defined as:

- $\delta[n] = 1$ for $n = 0$
- $\delta[n] = 0$ for $n \neq 0$



Unit Step Signal ($u[n]$)

The unit step signal is defined as:

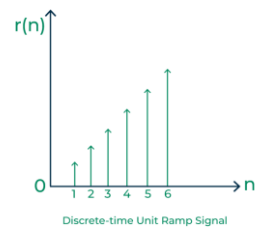
- $u[n] = 1$ for $n \geq 0$
- $u[n] = 0$ for $n < 0$



Ramp Signal ($r[n]$)

The ramp signal is defined as:

- $r[n] = n$ for $n \geq 0$
- $r[n] = 0$ for $n < 0$



Signal Relationship

1. **Unit step from unit impulse:** $u[n] = \sum_{k=-\infty}^n \delta[k]$ (cumulative sum)
2. **Unit impulse from unit step:** $\delta[n] = u[n] - u[n-1]$ (difference)
3. **Ramp from unit step:** $r[n] = \sum_{k=-\infty}^n u[k]$ (cumulative sum)
4. **Unit step from ramp:** $u[n] = r[n] - r[n-1]$ (difference)

Procedure:

1. The MATLAB environment was initialized and sample indices were defined
2. A unit impulse signal with value 1 at $n=0$ was created
3. A unit step signal with value 1 for $n \geq 0$ was created
4. A ramp signal with value n for $n \geq 0$ was created

5. Signal conversions were performed using cumulative sum and difference operations
6. All original and converted signals were plotted
7. The conversions were verified by comparing the plots

MATLAB CODE:

<pre>% Clear workspace and command window clear all; clc; close all; % Define time vector n = -10:20; % Sample indices % 1. Generate Unit Impulse Signal delta = zeros(size(n)); delta(n == 0) = 1; % Impulse at n=0 % 2. Generate Unit Step Signal u = zeros(size(n)); u(n >= 0) = 1; % Step function % 3. Generate Ramp Signal r = zeros(size(n)); r(n >= 0) = n(n >= 0); % Ramp function % 4. Convert Unit Impulse to Unit Step (Cumulative Sum) u_from_delta = cumsum(delta); % 5. Convert Unit Step to Unit Impulse (Difference) delta_from_u = [0, diff(u)]; % Add zero at beginning % 6. Convert Unit Step to Ramp (Cumulative Sum) r_from_u = cumsum(u); % 7. Convert Ramp to Unit Step (Difference) u_from_r = [0, diff(r)]; % Add zero at beginning % Plotting figure('Position', [100, 100, 1200, 800]);</pre>	<pre>% Original signals subplot(3,3,1); stem(n, delta, 'b', 'LineWidth', 2); title('Unit Impulse Signal $\delta[n]$'); xlabel('n'); ylabel('Amplitude'); grid on; subplot(3,3,2); stem(n, u, 'r', 'LineWidth', 2); title('Unit Step Signal u[n]'); xlabel('n'); ylabel('Amplitude'); grid on; subplot(3,3,3); stem(n, r, 'g', 'LineWidth', 2); title('Ramp Signal r[n]'); xlabel('n'); ylabel('Amplitude'); grid on; % Converted signals subplot(3,3,4); stem(n, u_from_delta, 'b--', 'LineWidth', 2); title('Unit Step from Impulse'); xlabel('n'); ylabel('Amplitude'); grid on; subplot(3,3,5); stem(n, delta_from_u, 'r--', 'LineWidth', 2); title('Unit Impulse from Step'); xlabel('n'); ylabel('Amplitude'); grid on; subplot(3,3,6); stem(n, r_from_u, 'g--', 'LineWidth', 2); title('Ramp from Step'); xlabel('n'); ylabel('Amplitude'); grid on;</pre>
---	--

```
subplot(3,3,7);
stem(n, u_from_r, 'm', 'LineWidth', 2);
title('Unit Step from Ramp');
xlabel('n'); ylabel('Amplitude');
grid on;
```

```
% Display signal values
fprintf('Signal Values at key points:\n');
fprintf('n = -2, -1, 0, 1, 2, 3, 4, 5\n');
fprintf('Delta: '); fprintf('%d ', delta(n
>= -2 & n <= 5)); fprintf('\n');
fprintf('Step: '); fprintf('%d ', u(n >= -2
& n <= 5)); fprintf('\n');
fprintf('Ramp: '); fprintf('%d ', r(n >= -2
& n <= 5)); fprintf('\n');
```

Output:

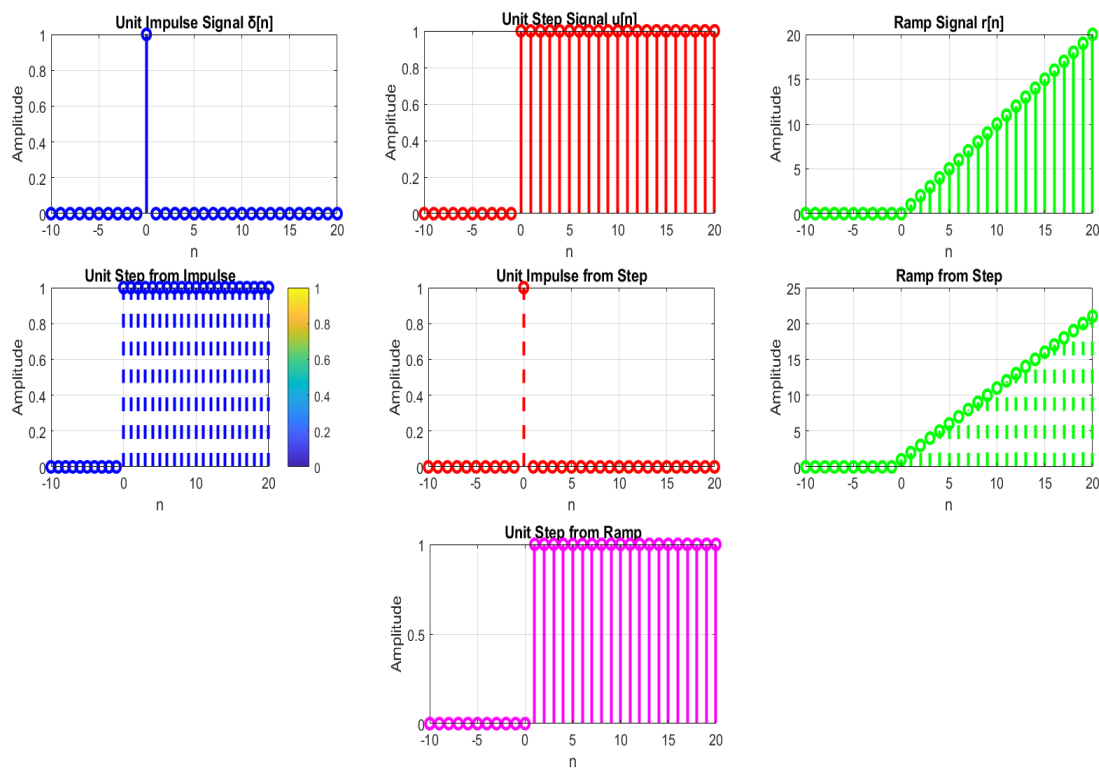


Figure 1: All generated original signals and converted signals

```
Command Window
New to MATLAB? See resources for Getting Started.
Signal Values at key points:
n = -2, -1, 0, 1, 2, 3, 4, 5
Delta: 0 0 1 0 0 0 0 0
Step: 0 0 1 1 1 1 1 1
Ramp: 0 0 0 1 2 3 4 5
fx >> |
```

Figure 2: Terminal output

Discussion:

Observations

1. **Unit Impulse Signal:** Shows a single spike at $n=0$ with amplitude 1, zero everywhere else.
2. **Unit Step Signal:** Shows zero values for $n<0$ and unity values for $n\geq 0$, creating a step discontinuity at $n=0$.
3. **Ramp Signal:** Shows zero for $n<0$ and linearly increasing values for $n\geq 0$.

Signal Conversions

1. **Impulse to Step:** The cumulative sum of the unit impulse gives the unit step signal, demonstrating the integration property.
2. **Step to Impulse:** Taking the difference of consecutive samples of the unit step gives the unit impulse, demonstrating the differentiation property.
3. **Step to Ramp:** The cumulative sum of the unit step creates the ramp signal.
4. **Ramp to Step:** The difference operation on the ramp signal recovers the unit step signal.

Mathematical Verification

- The conversions follow the fundamental relationships in discrete-time signal processing
- Integration (cumulative sum) and differentiation (difference) are inverse operations
- The results match theoretical expectations

Practical Applications

- These basic signals are building blocks for more complex signal analysis
- Understanding these conversions is crucial for filter design and system analysis
- They form the basis for convolution and system response calculations

Conclusion:

The experiment successfully demonstrated

1. **Signal Generation:** All three fundamental discrete-time signals (unit impulse, unit step, and ramp) were generated correctly in MATLAB.
2. **Signal Conversions:** The mathematical relationships between these signals were verified through cumulative sum and difference operations.

3. **MATLAB Implementation:** The code effectively used built-in functions like `cumsum()` and `diff()` to perform signal conversions.
4. **Visualization:** The `stem()` plots clearly showed the discrete nature of the signals and the success of the conversion operations.
5. **Theoretical Validation:** The results matched the expected theoretical behavior of these fundamental signals.

This lab provides a foundation for understanding discrete-time signal processing concepts and prepares for more advanced topics like convolution, filtering, and system analysis. The ability to convert between these basic signals is essential for digital signal processing applications.

References:

1. TechnicalSource9. "How do I generate a unit impulse signal in MATLAB?" *Medium*, Available: <https://technicalsource9.medium.com/how-do-i-generate-a-unit-impulse-signal-in-matlab-d17dc67ceb8b> [Accessed: July 26, 2025].
2. Fiber Optics 4 Sale. "The Unit Impulse and Unit Step Functions," *Wave Optics Blog*, Available: <https://www.fiberoptics4sale.com/blogs/wave-optics/the-unit-impulse-and-unit-step-functions> [Accessed: July 26, 2025].
3. GeeksforGeeks. "Unit Ramp Signal," *Electronics Engineering*, Available: <https://www.geeksforgeeks.org/electronics-engineering/unit-ramp-signal/> [Accessed: July 26, 2025].