RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

LICA Open ended experiment report

Aahana Nischal
1RV23ET001

Simulation of astable multivibrator using MATLAB

Contents

Aim

To simulate an astable multivibrator using MATLAB and analyze its behavior by observing the output waveforms. The simulation will demonstrate the generation of a continuous square wave signal through the oscillations of the circuit.

Working

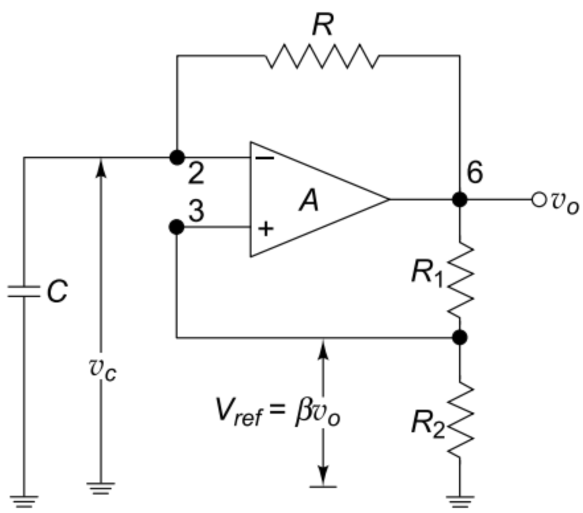
The astable multivibrator works as a square wave generator. The output of the opamp is fed back to its non-inverting terminal - Resistors R1 and R2 form a feedback network and a fraction of the output(β=R2/R1+R2) is fed back as the input.

The capacitor C charges to +Vsat through resistor R. The voltage at the non inverting terminal of the opamp is held at +βVsat as indicated by the use of the R1-R2 potential divider network.

The charging continues until the voltage at the negative terminal of the opamp is just greater than the voltage at the non-inverting terminal (+βVsat). When this happens the output voltage switches down to -Vsat.

The voltage at the inverting terminal becomes more negative and at one point exceeds -βVsat and then the output switches back to +Vsat. This cycle repeats.

Circuit Diagram



Equations

$$v_c(t) = +V_{sat} + (-\beta V_{sat} - V_{sat})\, e^{\frac{-t}{RC}}$$

$$\beta V_{sat} = V_{sat}\left(1 - (1+\beta)\, e^{\frac{-T_1}{RC}}\right)$$

$$(1-\beta) = (1+\beta)\, e^{\frac{T_1}{RC}}$$

$$T_1 = RC \ln \frac{1+\beta}{1-\beta} = RC \ln \frac{R_1 + 2R_2}{R_1}$$

$$T = 2T_1 = 2RC \ln \frac{1+\beta}{1-\beta}$$

$$T = 2\, RC \ln\left(\frac{R_1 + 2R_2}{R_1}\right)$$

$$f_o = \frac{1}{T} = \frac{1}{2\, RC \ln\left(\frac{1+\beta}{1-\beta}\right)}$$

$$f_o = \frac{1}{2RC \ln 3} = \frac{1}{2.2RC}$$

MATLAB Code

```
% Parameters for the astable multivibrator circuit
R1 = 483.2e3;        % Resistor 1 value (Ohms)
R2 = 483.2e3;        % Resistor 2 value (Ohms)
C = 1e-6;            % Capacitor value (Farads)
Vcc = 5;             % Supply voltage (Volts)
Vth_low = 1.5;       % Lower threshold voltage (Volts)
Vth_high = 3.5;      % Upper threshold voltage (Volts)

% Time settings
Tmax = 5;         % Maximum time to simulate (seconds)
dt = 1e-5;           % Time step (seconds)
t = 0:dt:Tmax;       % Time vector

% Initial conditions
V_C = 0;             % Initial capacitor voltage (Volts)
output = zeros(size(t)); % Output square waveform
charge = true;       % Flag for charging the capacitor

% Store capacitor voltage for plotting
V_C_values = zeros(size(t));

% Simulation of charging and discharging behavior
for i = 2:length(t)
    % Charging phase
    if charge
        V_C = V_C + (Vcc - V_C) / (R1 * C) * dt;  % Exponential charge
        if V_C >= Vth_high  % Switch to discharge when voltage reaches upper threshold
            charge = false;
            output(i) = 0;  % Output goes low during discharge
        else
            output(i) = 1;  % Output goes high during charge
        end
    % Discharging phase
    else
        V_C = V_C - V_C / (R2 * C) * dt;  % Exponential discharge
        if V_C <= Vth_low  % Switch to charge when voltage reaches lower threshold
            charge = true;
            output(i) = 1;  % Output goes high during charge
        else
            output(i) = 0;  % Output stays low during discharge
        end
    end
    % Store the capacitor voltage for plotting
    V_C_values(i) = V_C;
end

% Adjust both the capacitor voltage and output to oscillate from -5V to +5V
V_C_values_adjusted = V_C_values - 2.5;  % Shift capacitor voltage to oscillate around 0V
output_adjusted = (output - 0.5) * Vcc;  % Shift output to oscillate between -2.5V and +2.5V

% Plot capacitor voltage and output square waveform in the same plot
figure;
plot(t, V_C_values_adjusted, 'LineWidth', 1.5, 'DisplayName', 'Capacitor Voltage');
hold on;
plot(t, output_adjusted, 'LineWidth', 1.5, 'DisplayName', 'Output Square Wave'); % Output wave scaled
xlabel('Time (seconds)');
ylabel('Voltage (V)');
title('Capacitor Voltage and Output Square Waveform');
legend('show');
grid on;

% Set y-axis limits from -5V to +5V
ylim([-5, 5]);
```

Simulation Results