**深 圳 大 学 实 验 报 告**

**课程名称：­ 信号与系统**

**实验名称： 信号的卷积实验（程序部分）**

**学 院： 电子与信息工程学院**

**专 业： 电子信息工程**

**指导教师： 郑能恒**

**报 告 人 ： 陈应权 学 号：2022280297 班 级： 文华班**

**实验时间： 2024.4.11-2024.4.18**

|  |
| --- |
| 一、Experiment purpose and requirement：   1. Understand the principles and applications of signal convolution: Through the experiment, comprehend the convolution operation between two signals and its applications in analog circuits and digital signal processing. 2. Verify the consistency of hardware and software results: Compare hardware and software experiment results to validate whether the implementation of signal convolution is consistent. 3. Design and implement software experiments: Use MATLAB programming language to design and implement the experiment for signal convolution. The input signal should be a periodic square wave signal, and the impulse response signal should be the impulse response of the RC circuit. |
| 二、Design ideas   1. Set Signal Parameters: 2. Determine the parameters of the input signal, such as the frequency, amplitude, and offset of the square wave signal. 3. Determine the parameters of the RC circuit, such as the resistance, capacitance, and time constant. 4. Generate Signals: 5. Use MATLAB to generate a periodic square wave signal, ensuring that the frequency and amplitude meet the experimental requirements. 6. Generate the impulse response signal of the RC circuit by calculating the impulse response function based on the RC circuit parameters. 7. Perform Convolution Operation: 8. Use MATLAB's conv function to perform convolution between the input signal and the impulse response signal. 9. To match the hardware experimental results, extract one cycle length of the convolution result. 10. Repeat the Cycle:   Repeat the cycle length of the convolution result to simulate multiple cycle repetitions.   1. Plot Graphs:   Use MATLAB to plot graphs of the input signal, impulse response signal, and convolution result.  Ensure that the time axis range is within the experimental requirements to better observe the experimental results.   1. Compare and Analyze: 2. Compare the software simulation of the convolution result with the hardware experimental results to verify their consistency. 3. Analyze the relationship between the input signal, impulse response signal, and convolution result to understand the circuit or system's influence on the input signal. |
| 三、Experimental process  Initialization parameters:  We first set the circuit and signal parameters, including capacitance, resistance, signal frequency, offset, amplitude, and total observation time.  Create a time series: We have created a time vector t that will be used to generate input signals and calculate impulse responses.  Generate input signal:  We used the square function to generate a square wave signal and made necessary adjustments to meet the experimental requirements.  Calculate impulse response:  According to the theory of RC circuits, we have calculated the impulse response h (t).  Perform convolution: We use the conv function to convolve the input signal and impulse response to obtain the output signal y (t).  Plot result: We used the plot function to plot the input signal, output signal, and impulse response in two subgraphs. We also set the size of the graphics window to display the results more clearly. |
| 四、Experimental result  Actual experimental results:      MATLAB simulation results: |
| 五、Experimental analysis   1. Convolution Operation: In the experiment, we observed the relationship between the convolution result and the input signal and impulse response signals by simulating the convolution operation between the square wave signal x(t) and the impulse response of the RC circuit h(t). This convolution operation simulates the output signal after the input signal passes through the circuit. 2. Signal Comparison: By comparing the convolution result with the input signal, we can analyze the effect of the convolution operation on the input signal. In the simulation experiment, the convolution result showed a shape and amplitude different from those of the input signal, reflecting the impact of the convolution operation. 3. Comparison between Hardware and Software Experiments: By comparing the results of the hardware experiment and the software experiment, we found that they were essentially consistent in terms of the convolution result. This indicates that the software simulation can accurately predict the behavior of the hardware circuit. 4. Impulse Response: The impulse response $h(t)$ of the RC circuit showed an exponential decay, which was consistent with expectations. This confirms that the RC circuit model in the simulation experiment is accurate. 5. Result Verification: By comparing the results of the hardware experiment and the software experiment, we obtained good consistency between the two. This validates the effectiveness of software simulation in signal processing and circuit analysis. 6. Conclusion: The experiment demonstrates that programming simulation can effectively reproduce the behavior of a hardware circuit and verify that the results of the hardware experiment and the software experiment are consistent. This provides a solid foundation for further research in signal processing and circuit design.   There are some problem:  In MATLAB simulation, the input signal is a square wave with a specific frequency and amplitude. In the simulation, this signal is idealized, without noise and distortion. Compared to actual experimental signals, there will always be some noise and measurement errors.  The dynamic characteristics of actual circuits may be more complex than simulation models, and it is also possible that the impulse signal in actual experiments is a pulse sequence, rather than the unit pulse in MATLAB simulation. MATLAB simulation is usually based on linear time invariant systems and uses unit pulse h (t).  Regarding the issue of pulse signals approaching zero at around 50e-5s, but convolutional signals taking about 0.0025 seconds to return to zero, this means that the system's processing of the input signal has been "broadened" in time, which is determined by the physical or electronic properties of the system.  Because the response signal h will almost be 0 when it is greater than this time, then the convolution will be disturbed, which is different from our experiment, so it is truncated here.  This can very well restore the phenomenon of pulse cycle after pulse cycle in the actual experiment |
| 六、指导教师批阅意见：  成绩评定：  指导老师签名：  年 月 日 |

% Parameter settings

C = 10e-9; % Capacitance value (farads)

R = 2.2e3; % Resistance value (ohms)

f = 10e3; % Square wave signal frequency (Hz)

V\_offset = 0; % Square wave signal offset (V)

V\_amplitude = 1; % Square wave signal amplitude (V)

T\_period = 1/f; % Square wave period (seconds)

t\_total = 1e-3; % Total time (seconds), observe for 1 millisecond

sampling\_interval = 50e-9; % Sampling interval (seconds)

% Time vector

t = 0:sampling\_interval:t\_total;

% Generate square wave signal x(t)

x = (V\_amplitude \* square(2 \* pi \* f \* t, 50) + 1) + V\_offset;

% Compute impulse response h(t) of the RC circuit

h = 1/(R \* C) \* exp(-t / (R \* C));

% Perform convolution operation

y = conv(x, h, 'same') \* sampling\_interval;

% Define cutoff time and index

cutoff\_time = 0.5e-3; % Time limit to retain

cutoff\_index = find(t > cutoff\_time, 1); % Find index corresponding to cutoff time

% Truncate time vector, input signal, convolution result, and impulse response

t\_truncated = t(1:cutoff\_index);

x\_truncated = x(1:cutoff\_index);

y\_truncated = y(1:cutoff\_index);

h\_truncated = h(1:cutoff\_index);

% Visualize results

figure;

% Subplot 1: Input signal and truncated convolution result

subplot(2,1,1);

plot(t\_truncated, x\_truncated, 'b', 'LineWidth', 2); % Input signal, blue

hold on;

plot(t\_truncated, y\_truncated, 'Color', [0.8500, 0.3250, 0.0980], 'LineWidth', 2); % Truncated convolution output, orange

hold off;

% Add title and labels

title('Input Signal and Truncated Convolution Result');

xlabel('Time (s)');

ylabel('Amplitude (V)');

% Add legend

legend('Input Signal x(t)', 'Convolution Output y(t)');

% Set plot limits

xlim([0 cutoff\_time]);

% Subplot 2: Impulse response

subplot(2,1,2);

plot(t\_truncated, h\_truncated, 'k', 'LineWidth', 2); % Impulse response, black

% Add title and labels

title('Impulse Response h(t)');

xlabel('Time (s)');

ylabel('Amplitude');

% Add legend

legend('h(t)');

% Set plot limits

xlim([0 cutoff\_time]);

% Ensure appropriate size of the plot window

set(gcf, 'Position', [100, 100, 800, 600]);