EXPERIMENT REPORT

of

Digital Signal Processing

GIBBS PHENOMENON GUI BY USING MATLAB

SUBMITTED BY

Xu Yining *2014141453202*

Professor Zhang Hong

School of Electronic and Information Engineering, Sichuan University

November, 2016

1 Gibbs Phenomenon

1.1 Background of Gibbs

In mathematics, the Gibbs phenomenon, discovered by Henry Wilbraham (1848) and rediscovered by J. Willard Gibbs (1899), is the peculiar manner in which the Fourier series of a piecewise continuously differentiable periodic function behaves at a jump discontinuity. The nth partial sum of the Fourier series has large oscillations near the jump, which might increase the maximum of the partial sum above that of the function itself. The overshoot does not die out as n increases, but approaches a finite limit. This sort of behavior was also observed by experimental physicists, but was believed to be due to imperfections in the measuring apparatuses.

1.1.1 Overview of Gibbs

When the original signal contains discontinuities when the original signal is recovered or approximated from the Fourier transform of the signal, the overshoot spike occurs at each discontinuity, a phenomenon known as the Gibbs phenomenon.
Gibbs phenomenon is when the harmonic components of the signal to express the sum of the waveform with a break point, and can be observed.
Complex periodic signals like square wave through a certain mathematical tools, you can get a series of different frequencies of harmonics.
According to Gibbs phenomenon, we can create a square wave.

In the square wave case the period L is 2π, the discontinuity x_0 is at zero, and the jump a is equal to $\pi/2$. For simplicity let us just deal with the case when N is even (the case of odd N is very similar). Then we have

$$S_N f(x) = \sin(x) + \frac{1}{3}\sin(3x) + \cdots + \frac{1}{N-1}\sin((N-1)x)$$

Substituting x_0, we obtain

$$S_N f(0) = 0 = \frac{-\frac{\pi}{4} + \frac{\pi}{4}}{2} = \frac{f(0^-) + f(0^+)}{2}$$

as claimed above. Next, we compute

$$S_N f\left(\frac{2\pi}{2N}\right) = \sin\left(\frac{\pi}{N}\right) + \frac{1}{3}\sin\left(\frac{3\pi}{N}\right) + \cdots + \frac{1}{N-1}\sin\left(\frac{(N-1)\pi}{N}\right)$$

If we introduce the normalized sinc function $sinc(x)$, we can rewrite this as

$$S_N f\left(\frac{2\pi}{2N}\right) = \frac{2}{\pi}\left[\frac{2}{N}\sin\left(\frac{1}{N}\right) + \frac{2}{N}\sin\left(\frac{3}{N}\right) + \cdots + \frac{2}{N}\sin\left(\frac{(N-1)}{N}\right)\right]$$

But the expression in square brackets is a numerical integration approximation to the integral $\int_0^1 sinc(x)dx$ (more precisely, it is a midpoint ruleapproximation with spacing $2/N$). Since the sinc function is continuous, this approximation converges to the actual integral as $N \to \infty$. Thus we have

$$\lim_{N\to\infty} S_N f\left(\frac{2\pi}{2N}\right) = \frac{\pi}{2}\int_0^1 sinc(x)dx$$

$$= \frac{1}{2}\int_{x=0}^1 \frac{\sin(\pi x)}{\pi x} d(\pi x)$$

$$= \frac{1}{2}\int_0^\pi \frac{\sin(t)}{t} dt = \frac{\pi}{4} + \frac{\pi}{2}\cdot(0.089489872236\ldots)$$

which was what was claimed in the previous section. A similar computation shows

$$\lim_{N\to\infty} S_N f\left(-\frac{2\pi}{2N}\right) = -\frac{\pi}{2}\int_0^1 sinc(x)dx = \frac{\pi}{4} + \frac{\pi}{2}\cdot(0.089489872236\ldots)$$

For period square wave x(t):

$$x(t) = \begin{cases} A & ,0 < t < \frac{T}{2} \\ -A & ,\frac{T}{2} < t < T \end{cases}$$

where T is the period of square wave. Its Fourier series expansion is available.

$$x(t) = \frac{4A}{n\pi}\sin(n\pi t)\,, n = 1,3,5,\ldots, \omega = \frac{2\pi}{T}$$

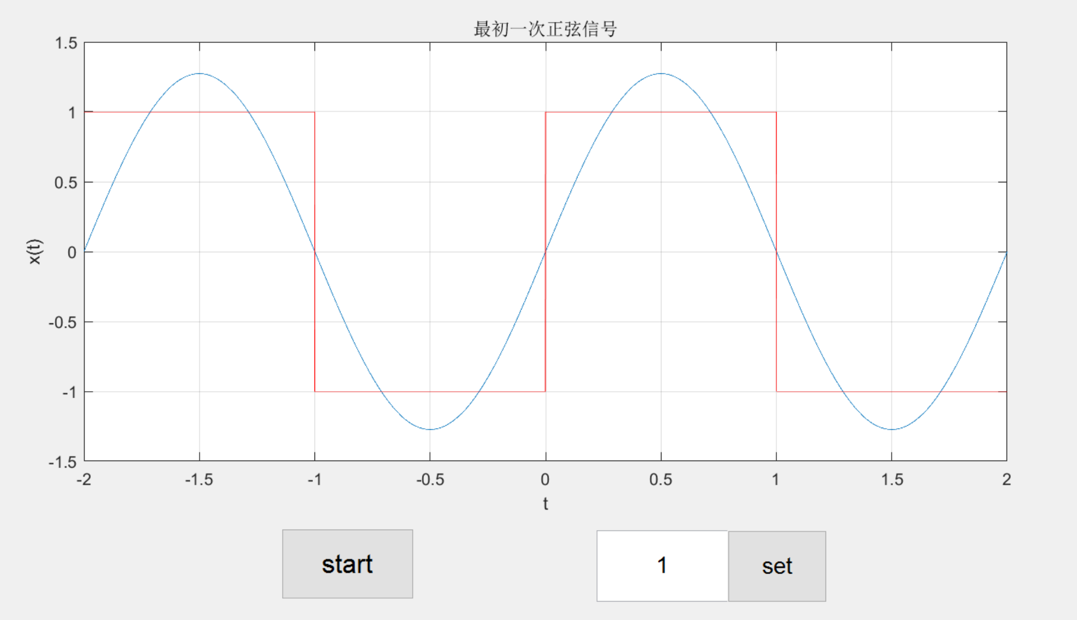
and when we coded in MATLAB, we will find that, with the increase the value of n, Gibbs phenomenon is more and more significant.

1.1.2 MATLAB Codes (*with Notes*)

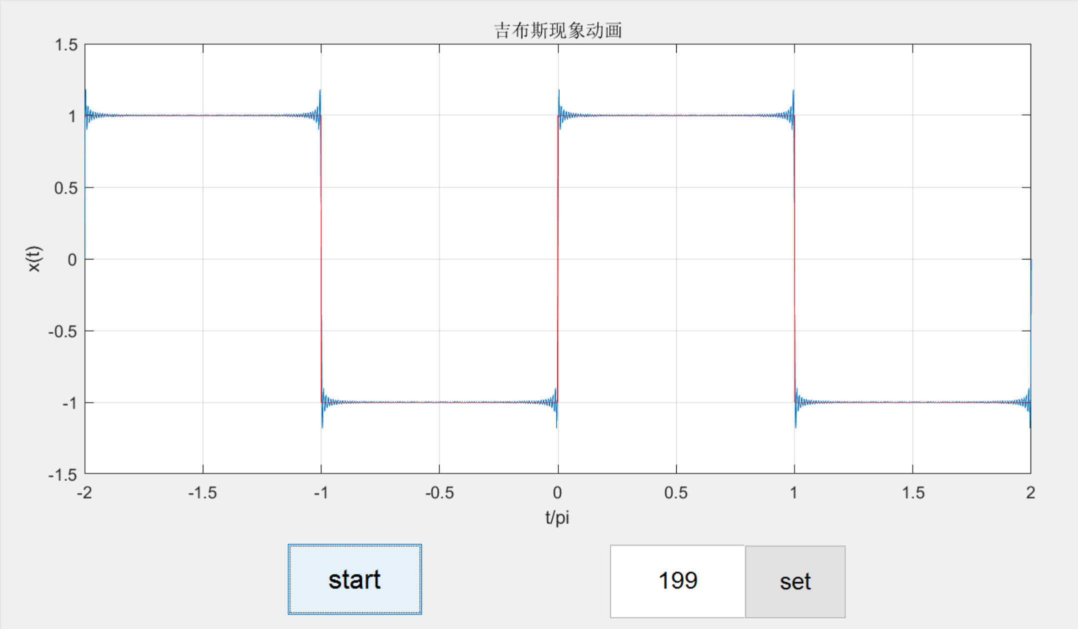
```
t = -2: 0.001: 2;
x = 4/pi * sin(t*pi);   % the first sin function

for n = 3: 2: N    % Nth harmonic superposition
    pause(2/n);    % pause delay decided by n
    x = x + (4/(n*pi) * sin(n*t*pi));
    y = square(x, 50);   % a square wave
    plot(t, x);          % the main signal wave
    hold on
    plot(t, y, 'r');      % the square wave
    hold off
    xlim([-2 2]);
```

```
    ylim([-1.5 1.5]);
    grid on;
    title('吉布斯现象');
    xlabel('t/pi');
    ylabel('x(t)');
end
```

1.1.3 Simulation Results and Illustrations



For the initial signal (1th harmonic superposition). It is a sin function.



For the 199th harmonic superposition. It is like a square wave.

1.1.4 Simple analysis and Summary

However, due to the phenomenon of periodic signals with discontinuous points, when the number of terms N of the selected Fourier series increases, the synthesized waveform is closer to the original function, but there is a fixed the higher the overshoot, the greater the overshoot, the greater the value of the overshoot, the closer to the discontinuity point, but the peak does not decrease, but approximately equal to the original function at the discontinuity point 9% of the jump value, and discontinuous point both sides of the form of attenuation oscillations.

2 Gibbs Phenomenon Using GUI

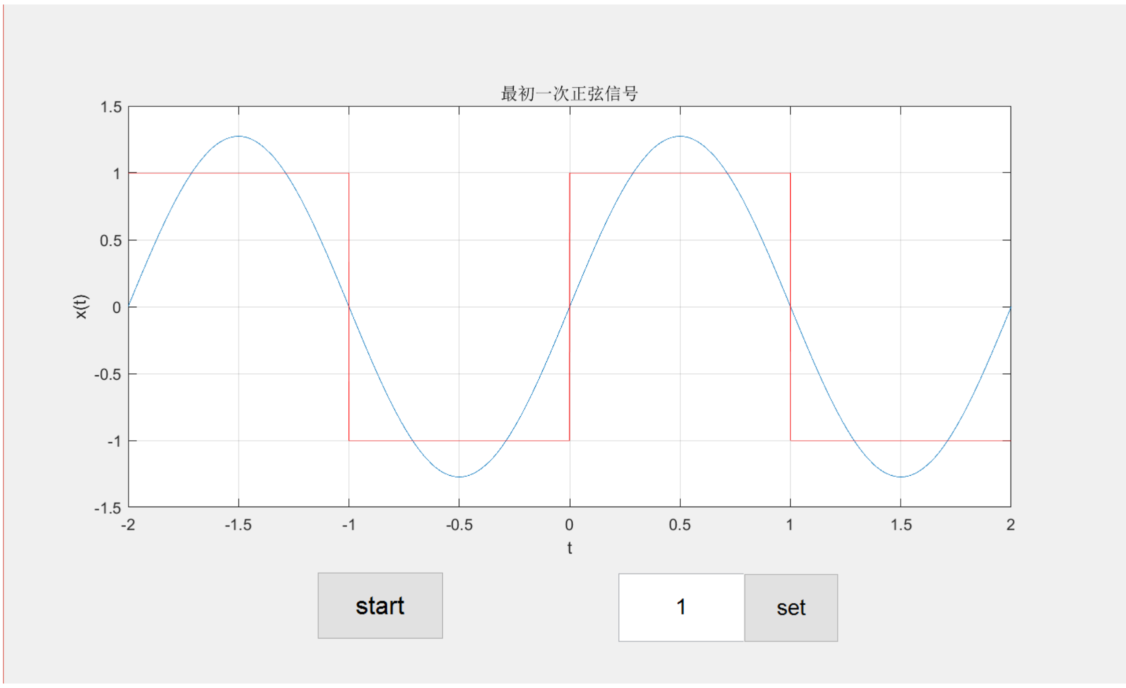
2.1 Introduce of GUI

In computer science, a graphical user interface, is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such assecondary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

2.2 Gibbs Phenomenon using GUI

2.2.1 Basic interface

As we can see in the figure as follow:



I used two buttons, a editable text and a axis to attained the GUI of Gibbs phenomenon.

2.2.2 MATLAB Codes (*with Notes*)

2.2.2.1 Button1

I used the botton1 to start the beginning of Gibbs phenomenon's actions, which can get the number N from handles, and loop overlay. The main code as follows:

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
N = str2double(handles.N);
t = -2: 0.001: 2;
x = 4/pi * sin(t*pi);    % the first sin function

for n = 3: 2: N    % Nth harmonic superposition
    pause(2/n);    % pause delay decided by n
    x = x + (4/(n*pi) * sin(n*t*pi));
    y = square(x, 50);   % a square wave
    plot(t, x);          % the main signal wave
    hold on
    plot(t, y, 'r');     % the square wave
    hold off
    xlim([-2 2]);
    ylim([-1.5 1.5]);
    grid on;
    title('吉布斯现象');
    xlabel('t/pi');
    ylabel('x(t)');
end
```

2.2.2.2 Button2

I using the button2 in order to update the value of N, which is the number of superimposed harmonics. The main code as follows:

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
handles.N = get(handles.edit2,'String');
```

```
guidata(hObject, handles);
```

2.2.2.3 Edit

I using the editable text due to let someone who is using this GUI to change the harmonics according to what he needed. If he want to get Nth harmonics superimpose, he can input a string N, then we can get this string, and using str2num or str2double to storage the number in handles. The main code as follows:

```
handles.N = get(handles.edit2,'String');
set(handles.edit2, 'String', num2str(handles.N));
```

2.3 Summary and Discussion

A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders and so forth. The GUI should behave in an understandable and predictable manner, so that a user knows what to expect when he or she performs an action. By using graphs in the form of Gibbs phenomenon will show the principle of more intuitive, and also you can see the benefits of GUI.

3 Summary and Conclusion

In signal processing, the Gibbs phenomenon is undesirable because it causes artifacts, namely clipping from the overshoot and undershoot, and ringing artifacts from the oscillations. In the case of low-pass filtering, these can be reduced or eliminated by using different low-pass filters. In MRI, the Gibbs phenomenon causes artifacts in the presence of adjacent regions of markedly differing signal intensity. This is most commonly encountered in spinal MR imaging, where the Gibbs phenomenon may simulate the appearance of syringomyelia.

Through this experiment, not only can it give us a more profound understanding of the Gibbs phenomenon, but also to understand the relevant knowledge of the GUI. Moreover, it is also strengthen our programming capabilities.