

DSP Homework 08

Xu, Minhuan

October 23, 2022

Contents

1	Videos	1
1.1	Boston Dynamics	1
1.1.1	Types of Robots in Boston Dynamics	1
1.1.2	Pros and Cons of Using Robots in the Military	2
1.2	Bone Density	2
1.2.1	How Our Body Change Bone Density	2
1.2.2	How We Can Raise Bone Density	2
1.3	My Thoughts	2
2	Comparison Between Two Sampling Methods	2
2.1	Local Adaptivity	2
2.2	Circuits	2
2.3	Flexible Error	2
2.4	Flexible Bandwidth	3
2.5	My Understanding of Wan Sampling Method	3
3	Conclusion	3
A	Code Listing	3

Abstract

1 Videos

1.1 Boston Dynamics

This video introduced the robots in Boston Dynamics briefly and mainly discussed about why somebody says that those robots can make soldiers obsolete. The robots' reliability, ruggedness, and most important, intelligence make that sound reasonable.

1.1.1 Types of Robots in Boston Dynamics

The robots in Boston Dynamics have 3 types, see the list below.

- Automatic Robots: These robots must have inputs from human, which means they are just the extension of human arms and legs.
- Automated Robots: These robots can have their own decision when they are facing a judgement which decides what they should do next. That kind of decision are in fact made by the program written by people in advance.
- Autonomous Robots: These robots have the ability of not only making decisions but learning how to make decisions. They are more like those things with intelligence or so-called AI.

However, if somebody wants robots to work like soldiers, this person means that the robots can complete a whole mission all by the robots themselves. To make this situation come true, there must be a synergy between mechanical parts and AI-powered programmed brain. Obviously, among those robots, only Autonomous Robots will have the change to replace the soldiers.

1.1.2 Pros and Cons of Using Robots in the Military

Also, we can know from the video what advantages and disadvantages those military concept robots have.

- Advantages: No human will die in war; Adjusting to severe environment easily,
- Disadvantages: vulnerable to hacking; AI out of control; moral issues like people who have access to the technology oppress those who don't.

1.2 Bone Density

1.2.1 How Our Body Change Bone Density

The bones are made of some compact bone tissues outside and some spongy bone tissues inside. There are two kind of cells in bones, and one form new bone tissues, and the other absorb the old bone tissues. With the help of these two cells, the inside spongy bone tissues are able to adapt the architecture to the environment.

1.2.2 How We Can Raise Bone Density

In this video, they said that once pushing force and pulling force are applied to bones, the bone will let cells produce more bone tissues to raise the bone density. Therefore, more force the bone bears, the more density it will have. What's more, almost every kind of exercises we usually do works in that way. In one word, exercise strengthen our bones.

1.3 My Thoughts

2 Comparison Between Two Sampling Methods

2.1 Local Adaptivity

We can know from the class that the reconstructed function of the wan Sampling Method usually be like

$$\hat{x}(t) = x(nT) + x'(nT)(t - nT) \quad T \in [n - \frac{1}{2}, n + \frac{1}{2}) \quad (1)$$

Therefore, a certain time $\hat{x}(t)$ can be expressed only using near sampling points.

But, in the Shannon/Nyquist Sampling Method, we have derived the reconstructed function many times and that is

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \cdot \text{sinc}(\frac{t - nT}{T}) \quad (2)$$

That means for a certain point t_0 we have

$$x(t_0) = \sum_{n=-\infty}^{\infty} x(nT) \cdot \text{sinc}(\frac{t_0 - nT}{T}) \quad (3)$$

and this specific value is associated with all sampling points.

Therefore, if the receiver got a piece of signal from the sender, it's hard to determine the sampling rate in advance in Shannon/Nyquist Sampling Method. However, in Wan Sampling Method we can use (1) to sample this little piece of signal.

2.2 Circuits

In Shannon/Nyquist Sampling Method, see (3), mathematical methods we must use are *infinite* summation, multiple, sinc function. It is easy to find that the summation and the sinc function are difficult to implement.

In Wan Sampling Method, see (1), mathematical methods we must use are just *finite* summation, multiple, differential calculus. In Analog Circuit, it is easy to do the plus, multiple, differential calculus using the operational amplifier.

2.3 Flexible Error

In Shannon/Nyquist Sampling Method, if we have determined the sampling rate in advance, the error is not predictable when the frequency of the signal is beyond the maximum frequency of Shannon/Nyquist Sampling Method.

In Wan Sampling Method, see subsection.2.5, no matter what signal the receiver get, as long as we know the order of the reconstructed signal take, we can always find the maximum of the error. Moreover, if we take different order of the reconstructed signal,

2.4 Flexible Bandwidth

In Shannon/Nyquist Sampling Method, even there's only a little piece of the signal are relatively high-frequency, the sampling rate still should be set to the twice of the maximum frequency, otherwise there will be unpredictable error in the reconstructed function.

In Wan Sampling Method, because the reconstructed function $x(t)$ are only associated values within the interval of $[n - \frac{1}{2}, n + \frac{1}{2}]$. The signal bandwidth can be variable, which means we can choose wide-band channel when signal are wide-band, and choose narrow-band channel when signal are narrow band.

2.5 My Understanding of Wan Sampling Method

The Taylor Theorem is expressed as below.

$$f(x) = \sum_{i=0}^N \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i + R_N(x)$$

Applying to our signal $x(t)$, we can make $x_0 = nT, N = 2$ here and get the equation below. In this case, we assume that the value of $R_2(t)$ here is small enough compared with $\frac{x''(t_0)}{2}(x - t_0)^2$, so just ignore it.

$$x(t) = x(nT) + x'(nT)(x - nT) + \frac{x''(t_0)}{2}(x - t_0)^2 \quad (4)$$

What exactly we get after sampling is only the values of $x(nT)$, but it is easy to do derivation. Therefore, using simple circuit, we know the values of $x'(nT), x''(nT), x'''(nT), \dots$

So, the reconstructed signal $\hat{x}(t)$ can be expressed as

$$x(t) = \overbrace{x(nT) + x'(nT)(x - nT)}^{\hat{x}(t)} + \frac{x''(t_0)}{2}(x - t_0)^2 \quad (5)$$

but also as

$$x(t) = \overbrace{x(nT)}^{\hat{x}(t)} + x'(nT)(x - nT) \quad (6)$$

The difference of the above two lies on the error of the reconstructed signal. It time to do the quantitative analysis, we assume that

$$\begin{aligned} |\hat{x}(t) - x(t)| &< \epsilon \\ |x^{(n)}(t)| &< \eta_n \\ |x - nT|^n &< \frac{T}{2} \end{aligned} \quad (7)$$

So, to ensure the accuracy, we need to make the sampling period T satisfy the equations below.

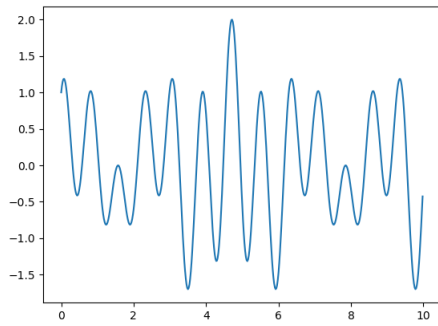
$$T < 2 \sqrt[n]{\frac{n! \epsilon}{\eta_n}} \quad (8)$$

About the value of T , I plotted a signal of $\cos(8t) + \sin(5t) \cos(2t)$ and the corresponding sampling period $T' = 2 \sqrt[n]{\frac{n!}{\eta_n}}$ (ϵ are ignored only to see the monotonicity of T about n), see Fig. 1.

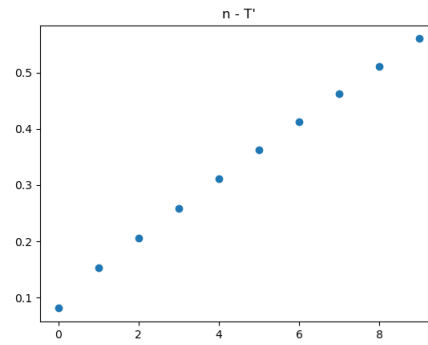
Analyzing these digits, I found that with the same error ϵ , if we raise the order of the $\hat{x}(t)$, the sampling rate can be set higher. In other words, it is the order of the reconstructed signal and the sampling rate decide the maximum of the error, and the two variables are negatively correlated when the error ϵ is fixed. So, if the accuracy is needed to be less, then what can be done is to raise the sampling rate or to raise the order of $\hat{x}(t)$.

3 Conclusion

Appendix A Code Listing



(a) Original Signal



(b) $n - T'$

Figure 1: The pictures I drew

```
import numpy as np
from matplotlib import pyplot as plt

N = 10
t = np.arange(0, 20, 0.1)
func = np.cos(8*t) + np.sin(5*t) * np.cos(2*t)

fig1 = plt.figure()
plt.plot(t, func)
plt.show()

maxs = []
intv = [i for i in range(N)]

for i in range(1, N + 1):
    df = np.diff(func, i)
    df.resize(len(t))
    df /= (t[1] - t[0])**i
    nl = 1
    for j in range(1, i + 1):
        nl *= j
    maxs.append((nl/np.max(df))**(1/i))

fig2 = plt.figure()
plt.scatter(intv, maxs)
plt.show()
```