

Hanoi University of
Science and Technology

BÁO CÁO BÀI TẬP LƯỢNG TỬ HÓA

MÔN HỌC: XỬ LÝ TÍN HIỆU

HỌC KỲ 2024.1

Giáo viên: Nguyễn Hồng Quang

Cao Văn Bảo 20225166

Ngày 2 tháng 10 năm 2024

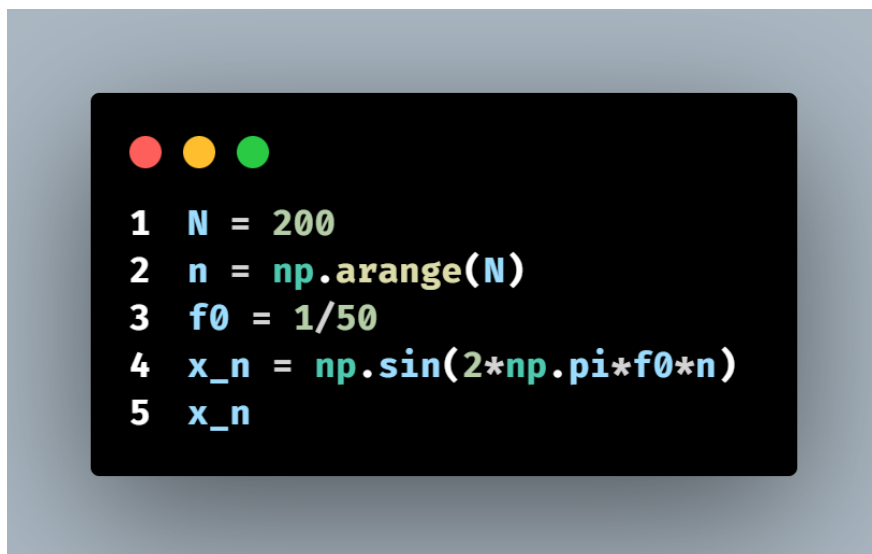
=

Mục lục

1	Xác định lỗi lượng tử	2
2	Mã nguồn	4

1 Xác định lỗi lượng tử

Tạo 1 tín hiệu hình sin: $x_n = \sin(2\pi f_0 n)$ với $N = 200$ và $f_0 = 1/50$



```
1 N = 200
2 n = np.arange(N)
3 f0 = 1/50
4 x_n = np.sin(2*np.pi*f0*n)
5 x_n
```

Tính các giá trị lượng tử với các mức lượng tử: 64, 128, 256 lần lượt là step64, step128, step256

```

1 max_val = np.max(x_n)
2 min_val = np.min(x_n)
3 step64 = (max_val-min_val)/(64-1)
4 step128 = (max_val-min_val)/(128-1)
5 step256 = (max_val-min_val)/(256-1)
6 print(step64, step128, step256)
7 max_val

```

Tiếp theo, dựa vào các bước lượng tử tính được, chúng ta sẽ xác định các tín hiệu sau khi lượng tử $x_q(n)$ bằng hàm determine xq_n

```

1 def determine_xq_n(step):
2     xq_n = np.round((x_n-min_val)/step) *step + min_val
3     return xq_n

```

Sau đó, chúng ta tính SQNR trên thực tế và lý thuyết, chuyển qua đơn vị db để so sánh

```

1 SNQR64 = max_val**2 / (np.sum(e_n64 * e_n64)/200 * 2)
2 SNQR128 = max_val**2 / ( 2* np.sum(e_n128 * e_n128)/200)
3 SNQR256 = max_val**2 / (2 * np.sum(e_n256 * e_n256)/200)
4 print(SNQR64, SNQR128, SNQR256, sep='\n')
5 SNQR64_lt = 3/2 * 2**(2*6)
6 SNQR128_lt = 3/2 * 2**(2*7)
7 SNQR256_lt = 3/2 * 2**(2*8)
8 print(SNQR64_lt, SNQR128_lt, SNQR256_lt)

```

Cuối cùng, vẽ 2 tín hiệu: $x(n)$ và $xq(n)$ bằng thư viện matplotlib, zoom 1 cự ly thích hợp để thấy rõ các tín hiệu $x(n)$ và $xq(n)$



```
1 plt.plot(n, x_n, label = "x(n)")
2 plt.plot(n, xq_n64, "o", label = "xq(n)")
3 ticks = np.linspace(min_val, max_val, 64)
4 plt.legend()
5 plt.yticks(ticks=ticks)
6 plt.ylim(-0.4, 0.5)
7 plt.xlim(71, 79)
8 plt.grid(True)
9 plt.show()
```

Conclusion: Tỷ lệ tín hiệu nhiễu SQNR có sự khác biệt nhỏ giữa lý thuyết và thực tế. Giải thích cho sự khác biệt đó là do tín hiệu phân bố không đều và sai số khi làm tròn

2 Mã nguồn

Chi tiết mã nguồn, mình để ở: [GitHub/repo:Signal Processing](#)