

# 当算法说"你的射频有问题"，你却找不到话反驳

原创 皮诺曹 射频通信链 2026年2月6日 11:05 江苏



射频通信链

让射频学习不再困难，学射频，学通信，就看射频通信链。

433篇原创内容

公众号

在射频集成系统的开发中，你是否曾遇到这样的场景：算法工程师递来一张布满毛刺和杂散的频谱图，斩钉截铁地指出你的射频链路性能不达标？你想辩解，却不知从何说起——因为你面对的是完全看不懂的复数 I/Q 数据。

## 一、为什么时域与频域必须联动？

算法工程师的视角聚焦在时域。他们关心采样点、波形变化、I/Q 轨迹的形态。他们担心时钟抖动会影响眼图质量，担心功率放大器（PA）饱和导致信号峰值被削平，担心本振泄漏在星座图中引入偏移。他们关注的，是从数字模拟转换器（DAC）输出到模数转换器（ADC）输入之间的真实模拟信号，以及一帧帧的数字采样点。

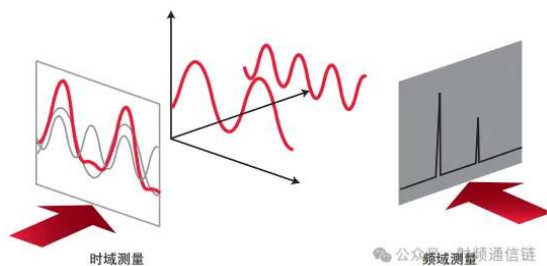
而射频工程师，则更习惯在频域中思考。我们分析史密斯圆图、S参数、相位噪声、邻道功率比（ACPR）等。我们习惯用稳态、周期性的视角去评估电路。频谱分析仪上那根信号强度的曲线，往往就是我们的“判决书”。

但问题的关键在于：射频问题虽然最终体现在频域，其根源却常常深藏于时域之中。

算法工程师采集到的 I/Q 数据，本质上是电压波形的数字化采样。我们之所以看不懂，仅仅是因为它们被进行了一层快速傅里叶变换（FFT）。

## 二、核心理论：FFT——时域到频域

快速傅里叶变换是完成这一转换的核心算法。它将一个时域信号分解成一系列不同频率、不同幅度、不同相位的正弦波之和。



基本公式： $X(f) = \text{FFT}(x(t))$  其中  $x(t)$  是时域信号， $X(f)$  是其频域表示（复数，包含幅度和相位信息）。

## 三、MATLAB实战：从时域波形到精准频谱分析

下面我们通过几个典型场景，演示如何利用 MATLAB 实现从时域到频域的转换，并进行深入分析。

场景一：分析一个简单的单音信号

% 参数设置

Fs = 100e6; % 采样率 100 MHz

T = 1e-5; % 信号时长 10 us

t = 0:1/Fs:T-1/Fs; % 时间向量

f0 = 10e6; % 信号频率 10 MHz

```
A = 1; % 信号幅度
```

```
% 1. 生成时域信号 (理想单音)
```

```
x_ideal = A * cos(2*pi*f0*t);
```

```
% 2. 执行FFT
```

```
N = length(x_ideal); % 信号点数
```

```
X_ideal = fft(x_ideal); % FFT计算
```

```
X_ideal = fftshift(X_ideal); % 将零频移动到频谱中心
```

```
% 3. 计算频率轴
```

```
f = (-N/2:N/2-1) * (Fs/N);
```

```
% 4. 计算幅度谱 (dBm)
```

```
P_ideal_dBm = 20*log10(abs(X_ideal)/sqrt(2)/50e-3); % 假设50欧姆负载
```

```
% 5. 绘图
```

```
figure;
```

```
subplot(2,1,1);
```

```
plot(t*1e6, x_ideal); % 时域, 时间单位us
```

```
xlabel('Time (us)'); ylabel('Amplitude (V)');
```

```
title('时域波形 - 理想单音');
```

```
grid on;
```

```
subplot(2,1,2);
```

```
plot(f/1e6, P_ideal_dBm); % 频域, 频率单位MHz
```

```
xlabel('Frequency (MHz)'); ylabel('Power (dBm)');
```

```
title('频域频谱 - 理想单音');
```

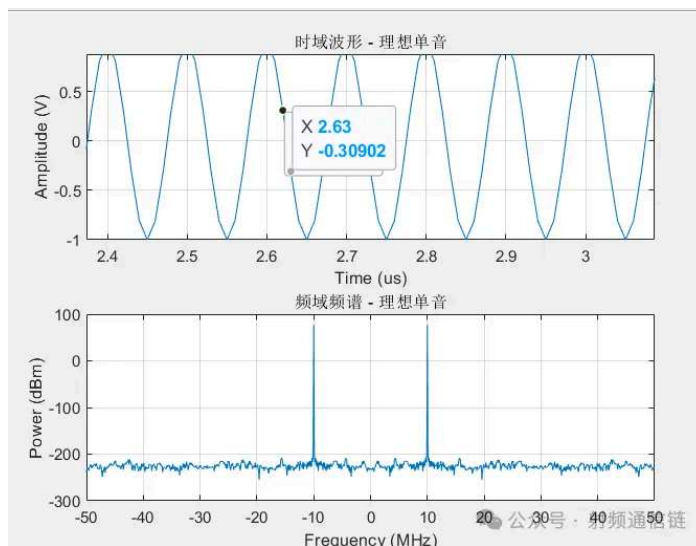
```
xlim([-Fs/2/1e6, Fs/2/1e6]); % 显示整个奈奎斯特区间
```

```
grid on;
```

这段代码能让你：

在时域看到一个纯净的正弦波。

在频域看到一个尖锐的单根谱线，其位置正好在10MHz。



场景二：引入射频损伤——看看放大器非线性

现在，我们模拟一个真实射频功率放大器的非线性效应。

% 在上一段代码的基础上，生成一个有非线性的信号

% 使用一个简单的无记忆多项式模型： $y = x - 0.1 \cdot x^3$

```
x_distorted = x_ideal - 0.1 * (x_ideal).^3;
```

% 执行相同的FFT和绘图流程

```
X_distorted = fft(x_distorted);
```

```
X_distorted = fftshift(X_distorted);
```

```
P_distorted_dBm = 20*log10(abs(X_distorted)/sqrt(2)/50e-3);
```

```
figure;
```

```
subplot(2,1,1);
```

```
plot(t*1e6, x_distorted);
```

```
xlabel('Time (us)'); ylabel('Amplitude (V)');
```

```
title('时域波形 - 存在非线性失真');
```

```
grid on;
```

```
subplot(2,1,2);
```

```
plot(f/1e6, P_distorted_dBm);
```

```
xlabel('Frequency (MHz)'); ylabel('Power (dBm)');
```

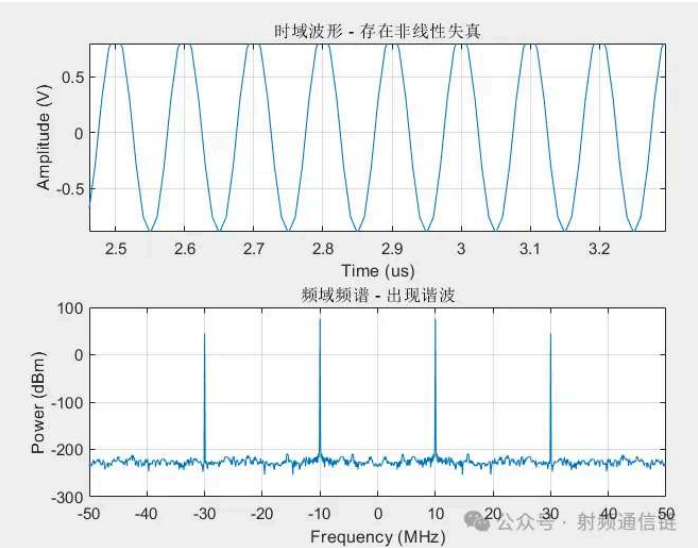
```
title('频域频谱 - 出现谐波');
```

```
xlim([-50, 50]); % 放大看关键区域
```

```
grid on;
```

这时你会看到：

时域：波形不再是完美的正弦波，顶部出现了轻微的平缓（压缩）。



频域：除了10MHz的主信号，在30MHz（3次谐波）处出现了一个新的频率分量！这就是非线性失真的直接证据。

结论

掌握时域与频域之间的转换，我们可以看懂算法的输出，结合算法去分析定位电路的问题：

主动诊断：将算法工程师提供的原始 I/Q 数据导入 MATLAB，分析其频域表现，并与射频指标进行关联，从而定位问题根源。

精准定位：当频域出现异常时，你能迅速判断是算法生成信号的问题（如峰均比过高），还是射频硬件的非线性、相位噪声、杂散等引起的。

高效沟通：通过展示时域与频域之间的内在联系，与算法工程师展开更有效的交流，推动问题快速解决。

在现代射频集成系统中，能够自由穿梭于时域与频域的工程师，才是系统设计的核心力量。

你在联调现场被算法“怼”过吗？最后怎么解决的？

最后的话：

如果你对射频方案设计把不准，想系统的学习，扫描下方二维码，手把手带你从头开始设计，还有一对一答疑：



皮诺曹

“ 射频工程师加油 ”

喜欢作者

阅读原文

