

《非线性电路实验》实验报告

实验名称: Small-Signal Resonant Amplifier 指导教师: 冯鹏 fengpeng06@semi.ac.cn
姓名: 丁毅 学号: 2023K8009908031 班级/专业: 2308/电子信息 分组序号: 2-06
实验日期: 2025.11.27 实验地点: 西实验楼(8号楼) 308 是否调课/补课: 否 成绩:

目录

1 实验目的	1
2 实验仪器	1
3 实验原理	2
3.1 小信号调谐放大器基本原理	2
3.2 小信号调谐放大器技术参数	3
4 实验内容与步骤	3
4.1 调整电路直流工作点	3
4.2 放大器的放大倍数及通频带的测试	3
5 实验结果与分析	3
6 思考题	7
6.1 小信号调谐放大器产生自激振荡的原因是什么? 如何避免产生自激振荡?	7
6.2 从频谱角度计算信号电压幅度时, 除积分法之外还有哪些方法?	7
附录 A 原始数据记录表	10
附录 B 实验预习报告	11
附录 C MATLAB Codes	15

1 实验目的

- (1) 掌握小信号谐振放大器的工作原理;
- (2) 掌握小信号谐振放大器的调试方法;
- (3) 掌握小信号谐振放大器各项技术参数测试(电压增益、通频带、矩形系数)。

2 实验仪器

- (1) 高频实验箱 - 小信号谐振放大器实验板 (031132201809392)
- (2) 示波器 RIGOL MSO2202A (080103201901376)
- (3) 信号发生器 GW INSTEK AFG-2225 (080102201901355)
- (4) 万用表 LINIT- UT61A (C181848639)

3 实验原理

3.1 小信号调谐放大器基本原理

小信号调谐放大器 (Small-Signal Resonant Amplifier) 的作用是有选择地对某一频率范围的高频小信号进行放大，这里的“小信号”通常指电压幅度 (amplitude) 在 $\mu\text{V} \sim \text{mV}$ 数量级的输入信号。本次实验所讨论的“小信号谐振放大器”，其实类似射频系统中的 LNA (Low Noise Amplifier, 低噪声放大器)，只是工作频率 (约 10 MHz) 远低于常见无线通信频段 (例如 2.4 GHz)。LNA 一般用于无线接收系统的前端，主要任务是对天线接收到的微弱射频信号进行放大，在保持可介绍信噪比 (SNR) 的情况下，将信号放大至足够大的功率给后续的混频器或解调器。

A9 参差调谐两级小信号放大器

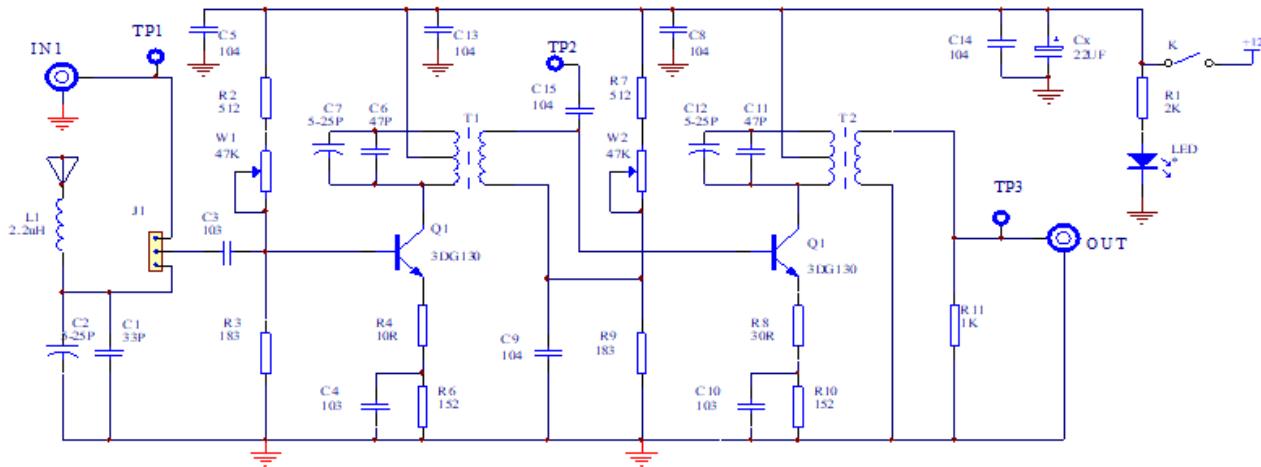


Figure 1: Small-Signal Resonant Amplifier Schematic

因此，我们完全可以将本实验中的小信号调谐放大器视为一种低频版的 LNA，其工作原理和技术参数与 LNA 十分相似，但对线性度和噪声系数的要求没有 LNA 那么高。

小信号调谐放大器 (后文简称为“Resonant Amplifier”) 的工作原理是：使用 RLC 谐振网络作为 BJT/MOS 放大电路的负载部分，谐振频率既为主要工作频率，使电路对带内信号 (in-band signal) 进行放大，同时有效抑制带外信号 (out-of-band signal)，这与 LNA 的工作原理类似。

本次实验的 Resonant Amplifier 由两级构成，根据噪声系数的理论公式，如果想尽可能降低噪声系数，就必须保持前级的噪声系数不太高。因此，在设计前级放大器时，常要求采用低噪声器件，合理地设置工作电流等，使放大器在尽可能高的功率增益下噪声系数最小。

对实验所用电路进行简单分析：

- (1) 这是一个二级放大器，从 IN1 (TP1) 端口输入小信号，由 C3 交流耦合到第一级放大器 Q1 的 Base；
- (2) 第一级：由 R2/W1/R3/R4/R6 提供直流偏置 (C4 为旁路电容)，C7/C6/T1 构成谐振负载网络 (C7 可调)，输出通过 T1 耦合到第二级放大器 Q2 的 Base，可通过 TP2 测量第一级输出信号；
- (3) 第二级：由 R7/W2/R9/R8/R10 提供直流偏置 (C10 为旁路电容)，C12/C11/T2 构成谐振负载网络 (C12 可调)，输出通过 T2 耦合到输出端口 OUT (TP3)。

3.2 小信号调谐放大器技术参数

R 调谐放大器的主要技术参数包括增益 (Voltage Gain)、带宽 (Bandwidth) 和矩形系数 (Shape Factor) 等：

$$\text{Voltage gain : } A_v := \frac{v_{out}}{v_{in}} \quad (1)$$

$$\text{Bandwidth : } \text{BW}_{-3\text{dB}} = f_{H,-3\text{dB}} - f_{L,-3\text{dB}} \quad (2)$$

$$\text{Shape Factor : } \text{SF}_{-20\text{dB}} := \frac{\text{BW}_{-20\text{dB}}}{\text{BW}_{-3\text{dB}}} \quad (3)$$

上式中 v_{in} 和 v_{out} 分别为放大器的输入和输出电压 (电压带有幅度和相位, 因此增益是复数), $\text{BW}_{-3\text{dB}}$ 和 $\text{BW}_{-20\text{dB}}$ 分别为放大器 voltage gain 下降 3 dB (最大值的 $\frac{\sqrt{2}}{2}$) 和 20 dB (最大值的 $\frac{1}{10}$) 时对应的频率范围宽度。注意 $\text{BW} = \frac{f_0}{Q}$, 因此放大器的带宽和矩形系数与谐振电路的品质因数 Q 密切相关。

4 实验内容与步骤

本次实验的注意内容为“放大器的频率特性及通频带的测量”。需注意：

- (1) 调整两级放大器的可调电容, 使两级放大器的谐振频率分别略低于和略高于 10 MHz, 但不能完全相等或相差过大; 完全相等会导致频率响应曲线过于尖锐, 难以测量通频带, 差距过大则会导致增益明显下降, 甚至出现中间窄两边高的“双峰现象”;
- (2) 为电路注入小信号时, 输入信号幅度 (amplitude) 应控制在 30 mV 以内, 过大会导致放大器进入非线性工作区, 输出波形严重失真;

4.1 调整电路直流工作点

- (1) 连接好电路, 为电路提供直流电源后, 从 IN1 端输入 $f_0 = 10 \text{ MHz}$ 小信号, 幅度在 10 mV 左右;
- (2) 在 OUT 端用示波器观测到放大后的输入信号, 调整两个电位器 W1/W2 来改变两个晶体管的静态工作点。微调电容 C7/C12, 使输出信号幅度最大且失真最小。

4.2 放大器的放大倍数及通频带的测试

- (1) 调整输入跳线 J1, 使电路输入端为 IN (TP1) 而不是天线;
- (2) 测量电压增益: 分别在 TP1 (IN) 和 TP3 (OUT) 处用示波器测量输入输出信号的幅度, 计算电压增益;
- (3) 测量带宽: 在中心频率 f_0 附近改变输入频率, 例如 $8 \text{ MHz} \sim 12 \text{ MHz} @ 0.1 \text{ MHz step}$, 重复步骤 (2), 对结果进行拟合后计算 -3dB 带宽;
- (4) 测量矩形系数: 根据步骤 (3) 得到的结果, 计算 -20dB 带宽, 进而计算矩形系数。

5 实验结果与分析

先设置信号发生器输出幅度为 $v_{sig,amp} = 25 \text{ mV}$ (50 mVpp) @ 10 MHz, 用示波器监测电路的输入输出情况。调整可变电阻和电容, 直到电路具有合适的直流工作点, 效果如 Figure 2 所示。

注意 Figure 2 中, 通道一测量的“Input Signal” v_{in} 是电路输入端实际接收到的信号, 与信号发生器直接输出的 v_{sig} 不同。具体而言, 假设电路的输入阻抗为 Z_{in} , 而信号发生器输出阻抗 $Z_s = R_s = 50 \Omega$, 我们有:

$$v_{in} = \frac{Z_{in}}{Z_{in} + Z_s} \times v_{sig} = \frac{Z_{in}}{Z_{in} + 50 \Omega} \times v_{sig} \quad (4)$$

将示波器两通道设置为“交流耦合”, 对不同输入频率下的输入输出电压进行采样。由于示波器上直接测量得到的 peak-to-peak value 不够准确, 我们将示波器采样数据导出到 MATLAB, 从频谱角度计算输入输

出电压幅度。具体而言，对于每次测量，将采样数据导出到 MATLAB 并作 DFT (Discrete Fourier Transform) 得到信号频谱，然后再由频谱计算出电压幅度，如 Figure 3, Figure 4 和 Figure 5 所示。

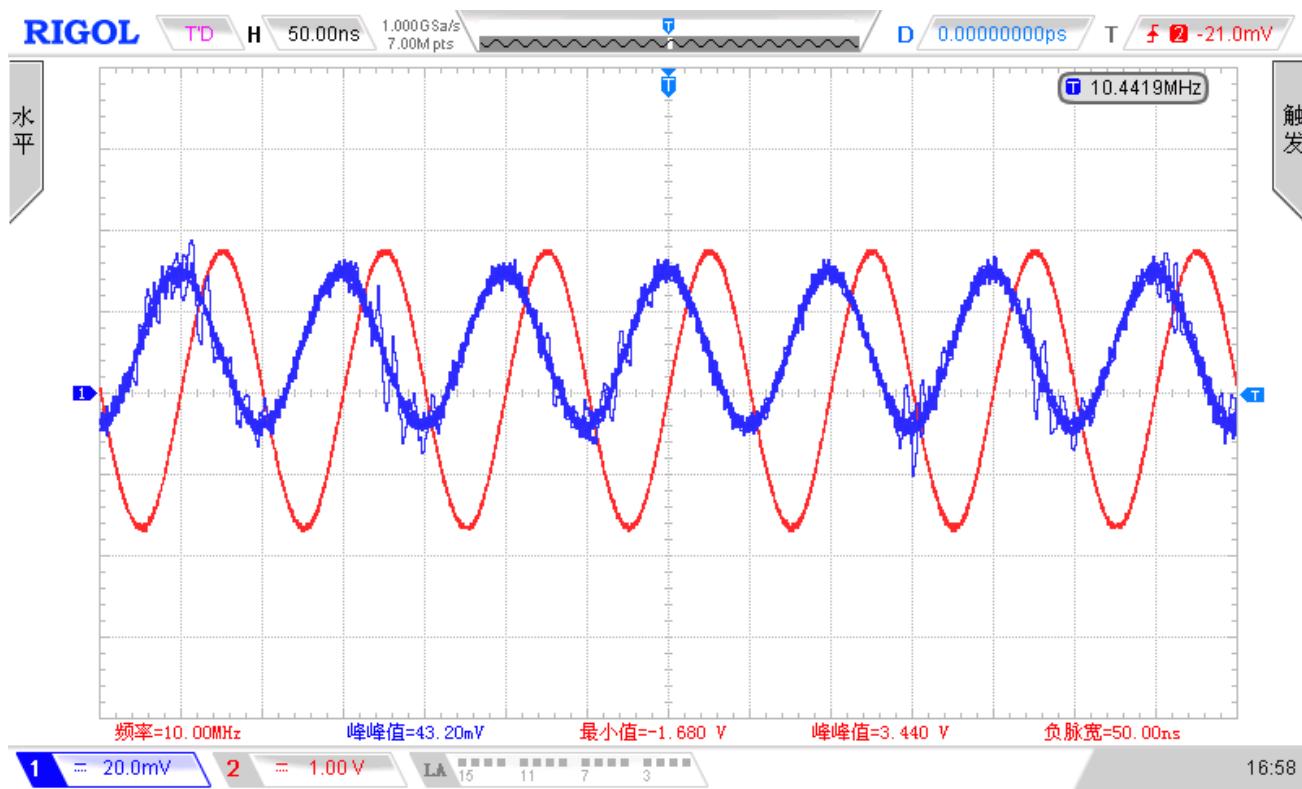


Figure 2: Input and Output Waveforms of The Resonant Amplifier @ Signal Amplitude $v_{sig,amp} = 25 \text{ mV}$ (50 mVpp). CH1 (Blue): Input Voltage v_{in} @ TP1; CH2 (Red): Output Voltage v_{out} @ TP3.

计算电压幅度时需注意：不能简单地将幅度最大的单个点作为信号幅度，因为频谱分辨率 $f_r = \Delta f$ 不一定与信号频率 f_0 匹配，当信号频率 f_0 不是频谱分辨率 Δf 的整数倍时，会出现“频谱泄露”现象。此时，需要将信号频率附近的区间考虑进去，在窄区间内对功率进行积分，再折算为电压幅度。具体代码详见“附录 B MATLAB Codes”。

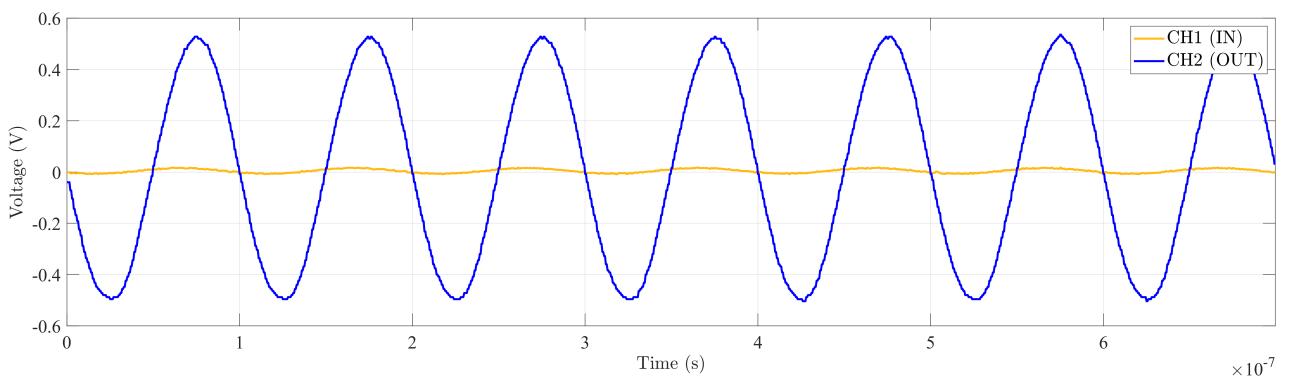


Figure 3: Exported Input and Output Waveforms from The Oscilloscope to MATLAB

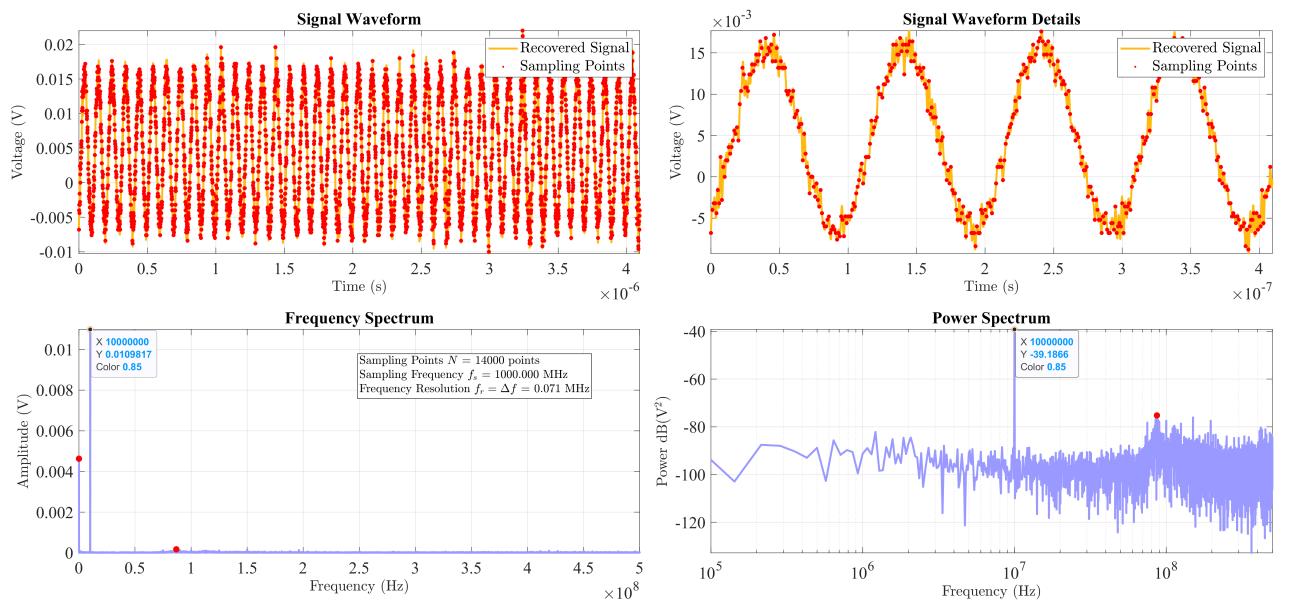


Figure 4: Calculate The Frequency Spectrum and Voltage Amplitude of The Input Voltage v_{in} in MATLAB

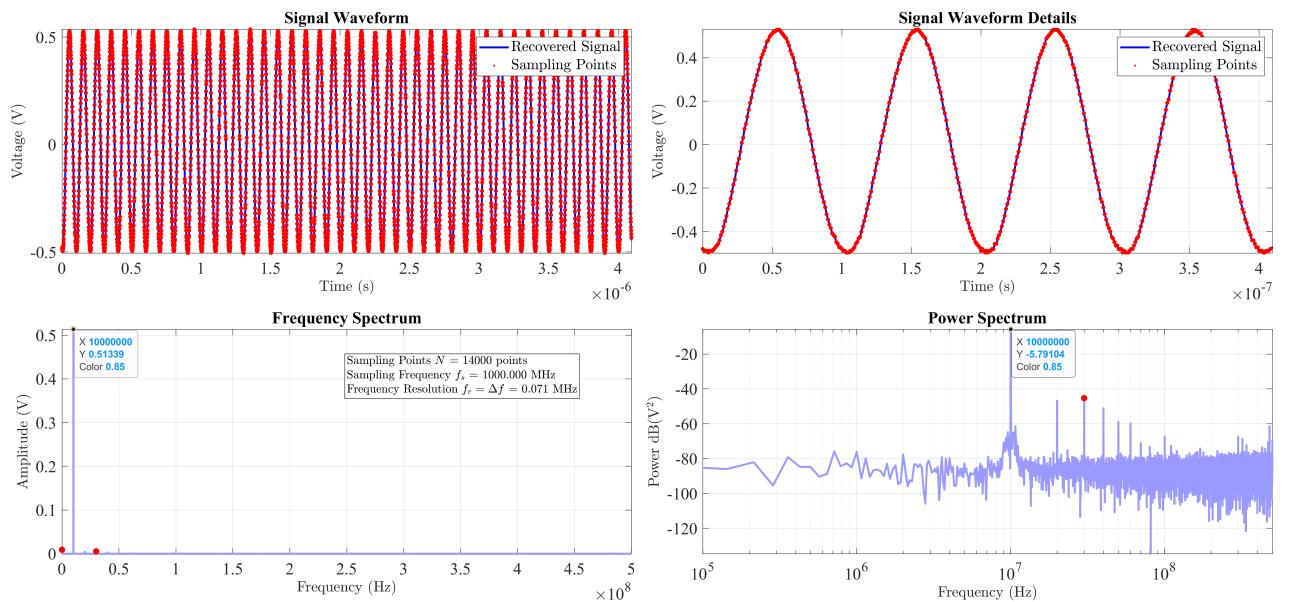


Figure 5: Calculate The Frequency Spectrum and Voltage Amplitude of The Output Voltage v_{out} in MATLAB

测量和计算得到的输入输出记录在 Table 1 中：

Table 1: Frequency Response of The Resonant Amplifier @ Signal Amplitude $V_{sig,amp} = 25 \text{ mV}$ (50 mVpp)

(信号发生器) f_{in} (MHz)	8.0	8.1	8.2	8.3	8.4	8.5	8.6	8.7	8.8	8.9	9.0	9.1	9.2	9.3	9.4	9.5	9.6	9.7
(AC of TP1) $V_{in,amp}$ (mV)	18.06	17.10	17.82	17.80	16.98	18.44	17.16	17.87	17.80	16.87	17.80	16.16	16.04	14.54	11.69	09.25	06.12	06.70
(AC of TP3) $V_{out,amp}$ (mV)	080.35	083.34	095.98	107.52	116.61	134.05	144.63	162.83	180.40	196.06	222.79	240.55	271.32	300.22	324.90	364.73	382.91	416.16
(信号发生器) f_{in} (MHz)	9.8	9.9	10.0	10.1	10.2	10.3	10.4	10.5	10.6	10.7	10.8	10.9	11.0	11.1	11.2	11.3	11.4	11.5
(AC of TP1) $V_{in,amp}$ (mV)	08.69	10.06	12.21	12.36	13.67	14.20	13.88	14.79	13.46	13.27	12.60	11.62	12.31	11.53	12.17	12.25	11.82	12.68
(AC of TP3) $V_{out,amp}$ (mV)	444.86	471.922	528.78	567.31	641.76	716.66	782.39	874.91	875.35	849.76	750.80	615.83	513.84	405.64	338.35	280.78	323.35	203.99

根据 Table 1 数据计算并拟合放大器的频率响应曲线，如 Figure 6 和 Figure 7 所示。然后进一步计算出放大器技术参数，包括 -3dB BW, -20dB BW, 矩形系数 SF 以及整个放大器的品质因数 $Q = \frac{f_0}{\text{BW}_{-3\text{dB}}}$ ：

$$A_{s,max} = A_s(f_0) = 17.5474 \text{ V/V} = 24.8843 \text{ dB}, \quad f_0 = 10.5683 \text{ MHz} \quad (5)$$

$$f_{-3\text{dB},L} = 10.177 \text{ MHz}, \quad f_{-3\text{dB},H} = 10.909 \text{ MHz} \implies \text{BW}_{-3\text{dB}} = 0.7314 \text{ MHz} \quad (6)$$

$$f_{-20\text{dB},L} = 8.286 \text{ MHz}, \quad f_{-20\text{dB},H} = 12.060 \text{ MHz} \implies \text{BW}_{-20\text{dB}} = 3.7735 \text{ MHz} \quad (7)$$

$$\text{SF}_{-20\text{dB}} = \frac{\text{BW}_{-20\text{dB}}}{\text{BW}_{-3\text{dB}}} = 5.1592, \quad Q = \frac{f_0}{\text{BW}_{-3\text{dB}}} = 14.4514 \quad (8)$$

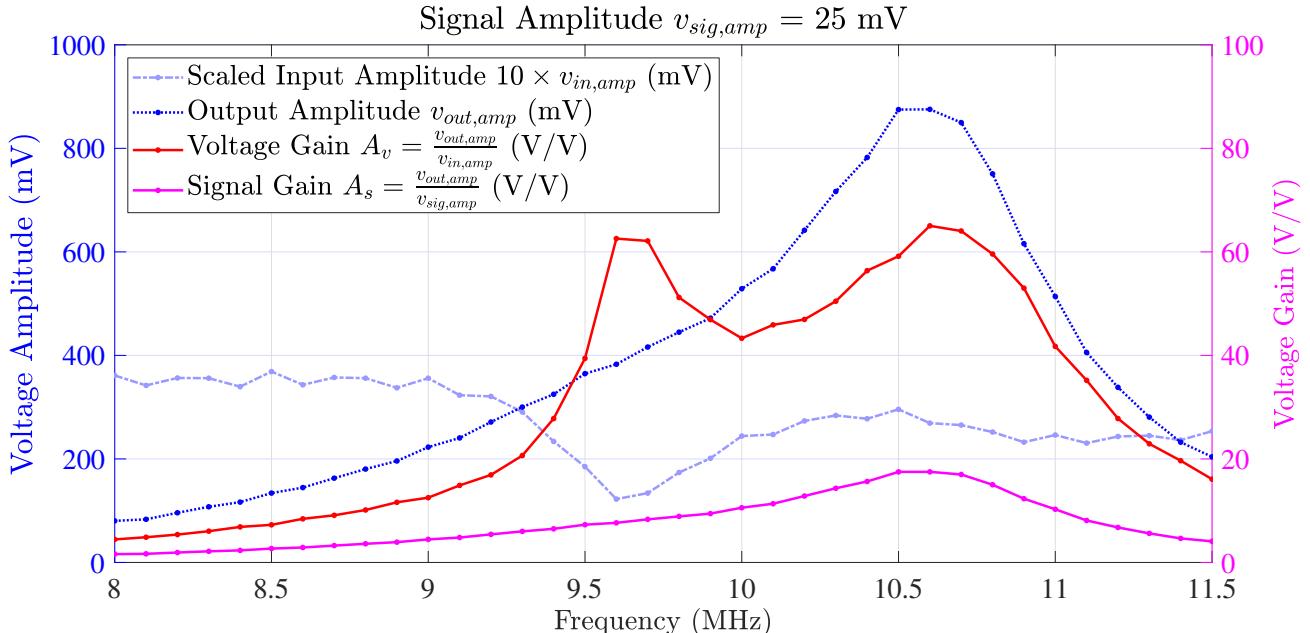


Figure 6: Frequency Response of The Resonant Amplifier (Signal Gain A_s and Voltage Gain A_v vs. Frequency f_{in})

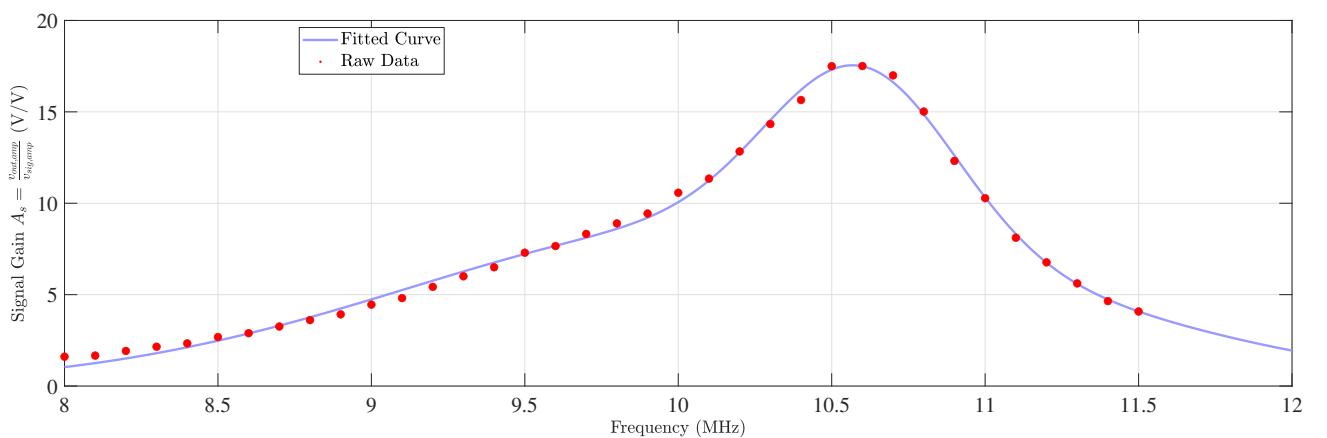


Figure 7: Fitted Signal Gain Frequency Response $A_s(f)$ of The Resonant Amplifier

从 Figure 6 中我们观察到 voltage gain $A_v = \frac{v_{out}}{v_{in}}$ 在 9.6 MHz 和 10.6 MHz 附近出现了两个峰值，猜测是由于两级放大器的谐振频率设置不当（过于远离）导致的“双峰现象”。随后多次尝试调整可变电容 C7/C12（例如 Figure 8），但都未能有效改善该现象。又注意到 9.6 MHz 出现峰值是因为输入信号幅度 $v_{in,amp}$ 异常降低，从 17 mVamp 降低到 7 mVamp 左右，导致计算得到的 voltage gain A_v 异常升高，因此原来的“谐振频率过于远离”猜测并不成立，估计是放大器第一级本身的输入阻抗导致，具体原因有待进一步探究。

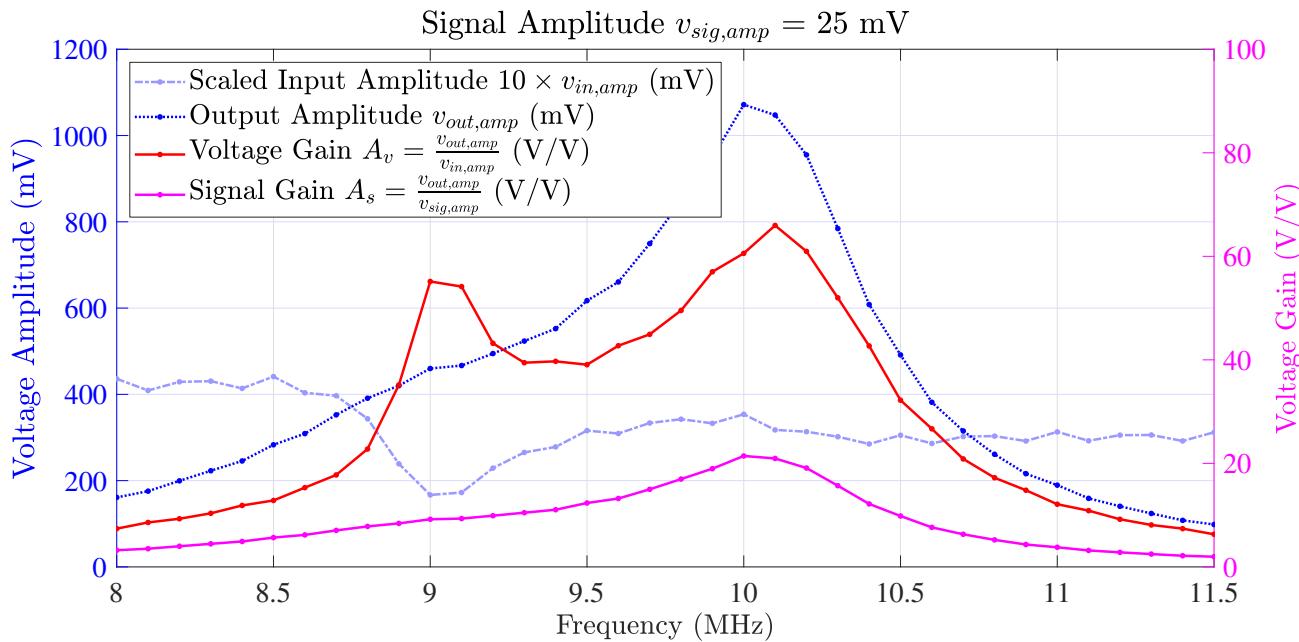


Figure 8: Frequency Response of The Resonant Amplifier After Adjusting C7/C12

6 思考题

6.1 小信号调谐放大器产生自激振荡的原因是什么？如何避免产生自激振荡？

本实验所用的小信号谐振放大器由两级构成，整体增益较高且 phase shift 较大。虽然电路中没有直接的反馈路径，但由于电路布局和元器件的寄生参数等原因（尤其晶体管的 parasitic base-collector capacitor C_π ，也常记为 C_{BE} ），仍然可能产生寄生反馈，一旦满足了自激振荡的起振条件 $|A_v(f_{osc})| = 1 @ \angle A_v(f_{osc}) = -180^\circ$ ，就会导致放大器产生自激振荡，输出端口出现持续的正弦波信号。此外，如果放大器的谐振频率设置不当（例如两级谐振频率完全相等），也可能导致频率响应曲线过于尖锐，增加了自激振荡的风险。

可通过在输入或输出端添加补偿网络（例如 RC 网络）来降低避免自激振荡的产生。此外，合理调整谐振频率，使两级谐振频率略有差异，也有助于降低自激振荡风险。

6.2 从频谱角度计算信号电压幅度时，除积分法之外还有哪些方法？

在本实验，我们采取的方法是：先计算未经窗函数处理的时域信号的频谱，再用积分法从频谱中计算信号功率，进而折算为电压幅度。这种方法的优点是简单直接，但缺点是频谱泄露现象会影响计算精度。

事实上，学界/业界最常用的方法是：在时域信号上先乘以一个窗函数（window function），例如 Hanning 窗、Hamming 窗或 Blackman 窗等，然后再计算频谱。加入窗函数后，频谱泄露现象会大大减弱，从而提高信号幅度计算的准确性。但是注意引入窗函数后，频谱幅度和功率有所衰减，需要对傅里叶结果进行矫正（correction），具体矫正系数取决于所用窗函数的类型。

作为一个例子，不妨设置被采样的原始信号为：

$$v(t) = (1 + A_{n1}(t)) \cdot \cos(2\pi f_0 t + \phi_{n1}(t)) + (0.1 + A_{n2}(t)) \cdot \cos(2\pi f_1 t + \phi_{n2}(t)) + V_n(t) \quad (9)$$

where $f_0 = 1$ kHz, $f_1 = 2$ kHz and A_n , ϕ_n , V_n denote amplitude, phase, voltage noise, respectively. $\quad (10)$

然后对比其在不同窗函数下的频谱结果，结果如 Figure 9 所示 (Uniform, Hanning and Flattop)。图中左半部分为时域信号乘以窗函数后的结果，右半部分为矫正后的功率谱 (Power Spectrum)。可以看出，三种不同窗函数 (Unitorm, Hanning, Flattop) 都能显示出 1 kHz 和 2 kHz 两个信号分量，但频谱泄露程度有所不同，Hanning 和 Flattop 窗函数的频谱泄露明显小于 Uniform (no window) 的结果。

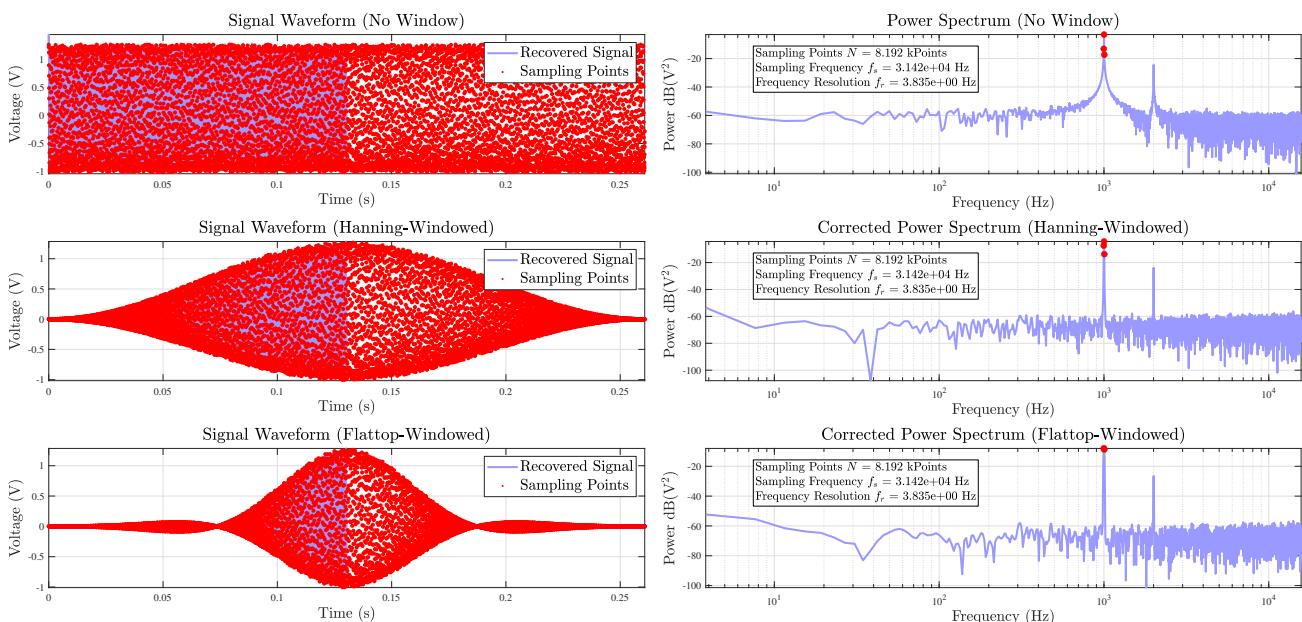


Figure 9: Comparison of Different Window Functions (Uniform, Hanning and Flattop) in Time-Domain and Frequency-Domain

从离散频谱结果计算信号频率/幅度/功率的理论详见下面这篇文献，我们这里直接给出关键计算公式：

[1] Brandt, A., 2011. Noise and vibration analysis: signal analysis and experimental procedures, John Wiley & Sons, doi: 10.1002/9780470978160.

$$\text{Extracted Frequency : } f_0 = \frac{\sum_{n=n_0-L}^{n_0+L} [P[n] \cdot (n\Delta f)]}{\sum_{n=n_0-L}^{n_0+L} P[n]} \quad (11)$$

$$\text{Corrected Power : } P_{0,cor} = \text{ECF} \times \sum_{n=n_0-L}^{n_0+L} P[n] \quad (12)$$

$$\text{Corrected Amplitude : } A_{0,cor} = \sqrt{2 \times P_{0,cor}} \quad (13)$$

where:

- (1) $P[n]$ is the single-sided power spectrum obtained from DFT;
- (2) n_0 is the index of the frequency bin closest to the target signal frequency f_0 ;
- (3) $\Delta f = f_s/N$ is the frequency resolution;
- (4) L denotes the half width of the integration window;
- (5) ECF is the Energy Correction Factor for the used window function, which is exactly 1 for Uniform window (no window).

三种窗函数的积分结果 (主信号频率/幅度/功率计算结果) 如 Figure 10 所示。图中可以看出，在当前信号和采样参数下，Hanning 窗函数收敛速度最快，计算结果最接近真实值，Flat Top 窗函数次之，Uniform 窗函数 (无窗函数) 收敛最慢，且频率收敛结果偏离真实值较多。

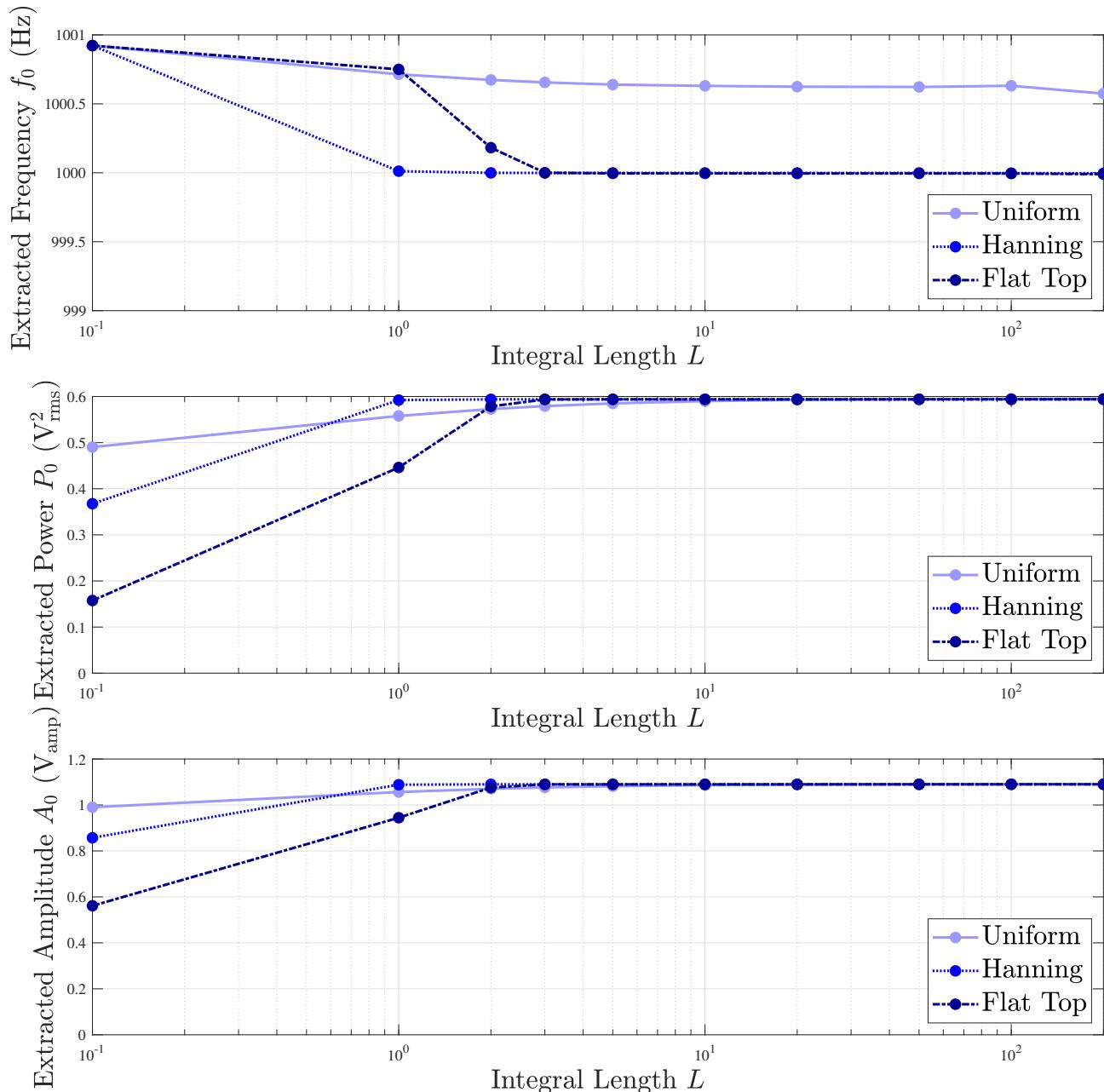
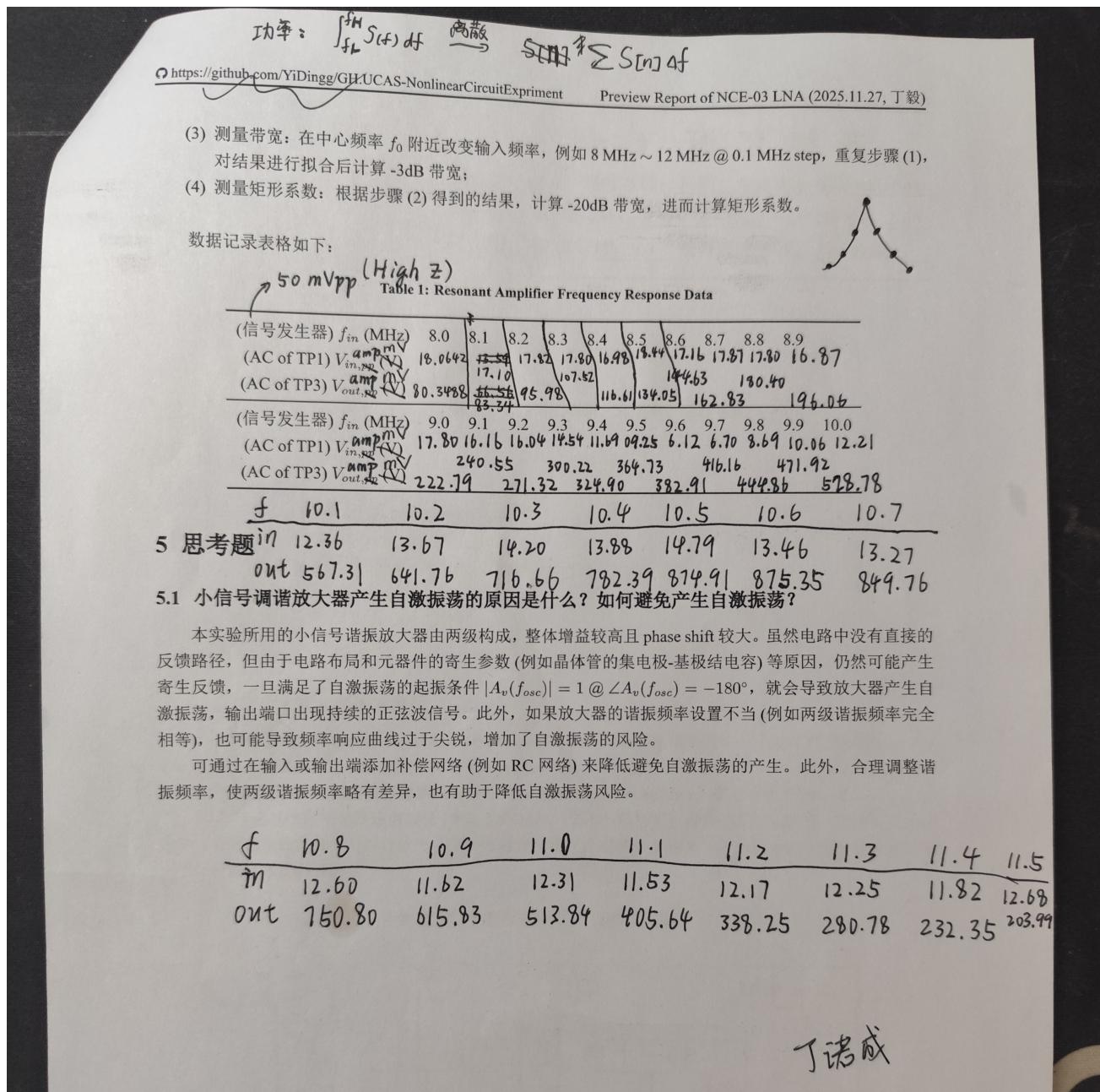


Figure 10: Comparison of Different Window Functions in Signal Frequency, Amplitude, and Power Calculation

除上述方法外，还有一些其他方法可以从频谱角度计算信号功率/幅度，比如“DFS-based method (直接在目标频率处对采样序列进行离散傅里叶级数展开)”等，这里不多赘述。

附录 A 原始数据记录表



附录 B 实验预习报告

《非线性电路实验》预习报告

实验名称: Small-Signal Resonant Amplifier 指导教师: 冯鹏 fengpeng06@semi.ac.cn
 姓名: 丁毅 学号: 2023K8009908031 班级/专业: 2308/电子信息 分组序号: 2-06
 实验日期: 2025.11.27 实验地点: 西实验楼(8号楼) 308 是否调课/补课: 否 成绩: _____

1 实验目的

- (1) 掌握小信号谐振放大器的工作原理;
- (2) 掌握小信号谐振放大器的调试方法;
- (3) 掌握小信号谐振放大器各项技术参数测试(电压增益、通频带、矩形系数)。

2 实验仪器

- (1) 小信号谐振放大器实验板 (序列号……)
- (2) 示波器 RIGOL MSO2202A (序列号……)
- (3) 信号发生器 GW INSTEK AFG-2225 (序列号……)
- (4) 万用表 LINIT- UT61A (序列号……)

3 实验原理

3.1 小信号调谐放大器基本原理

小信号调谐放大器 (Small-Signal Resonant Amplifier) 的作用是有选择地对某一频率范围的高频小信号进行放大，这里的“小信号”通常指电压幅度 (amplitude) 在 uV ~ mV 数量级的输入信号。本次实验所讨论的“小信号谐振放大器”，其实类似射频系统中的 LNA (Low Noise Amplifier, 低噪声放大器)，只是工作频率 (约 10 MHz) 远低于常见无线通信频段 (例如 2.4 GHz)。LNA 一般用于无线接收系统的前端，主要任务是对天线接收到的微弱射频信号进行放大，在保持可介绍信噪比 (SNR) 的情况下，将信号放大至足够大的功率给后续的混频器或解调器。

A9 参差调谐两级小信号放大器

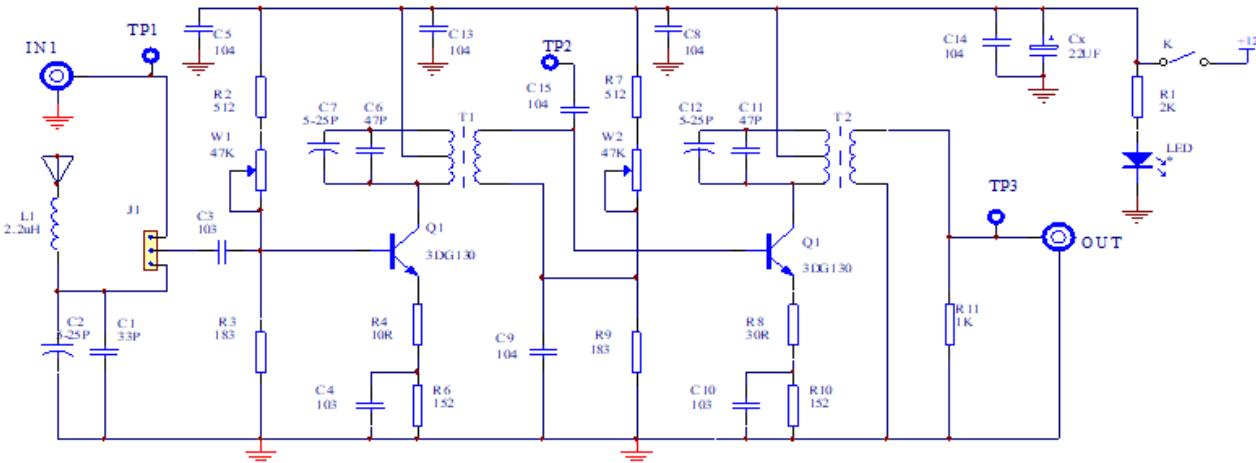


Figure 1: Small-Signal Resonant Amplifier Schematic

因此，我们完全可以将本实验中的小信号调谐放大器视为一种低频版的 LNA，其工作原理和技术参数与 LNA 十分相似，但对线性度和噪声系数的要求没有 LNA 那么高。

小信号调谐放大器(后文简称为“Resonant Amplifier”)的工作原理是：使用 RLC 谐振网络作为 BJT/MOS 放大电路的负载部分，谐振频率既为主要工作频率，使电路对带内信号 (in-band signal) 进行放大，同时有效抑制带外信号 (out-of-band signal)，这与 LNA 的工作原理类似。

本次实验的 Resonant Amplifier 由两级构成，根据噪声系数的理论公式，如果想尽可能降低噪声系数，就必须保持前级的噪声系数不太高。因此，在设计前级放大器时，常要求采用低噪声器件，合理地设置工作电流等，使放大器在尽可能高的功率增益下噪声系数最小。

对实验所用电路进行简单分析：

- (1) 这是一个二级放大器，从 IN1 (TP1) 端口输入小信号，由 C3 交流耦合到第一级放大器 Q1 的 Base;
- (2) 第一级：由 R2/W1/R3/R4/R6 提供直流偏置 (C4 为旁路电容)，C7/C6/T1 构成谐振负载网络 (C7 可调)，输出通过 T1 耦合到第二级放大器 Q2 的 Base，可通过 TP2 测量第一级输出信号；
- (3) 第二级：由 R7/W2/R9/R8/R10 提供直流偏置 (C10 为旁路电容)，C12/C11/T2 构成谐振负载网络 (C12 可调)，输出通过 T2 耦合到输出端口 OUT (TP3)。

3.2 小信号调谐放大器技术参数

Resonant Amplifier 的主要技术参数包括电压增益 (Voltage Gain)、通频带 (Bandwidth) 和矩形系数 (Shape Factor) 等：

$$\text{Voltage gain : } A_v := \frac{v_{out}}{v_{in}} \quad (1)$$

$$\text{Bandwidth : } \text{BW}_{-3\text{dB}} = f_{H,-3\text{dB}} - f_{L,-3\text{dB}} \quad (2)$$

$$\text{Shape Factor : } \text{SF}_{-20\text{dB}} := \frac{\text{BW}_{-20\text{dB}}}{\text{BW}_{-3\text{dB}}} \quad (3)$$

上式中 v_{in} 和 v_{out} 分别为放大器的输入和输出电压 (电压带有幅度和相位，因此增益是复数)， $\text{BW}_{-3\text{dB}}$ 和 $\text{BW}_{-20\text{dB}}$ 分别为放大器 voltage gain 下降 3 dB (最大值的 $\frac{\sqrt{2}}{2}$) 和 20 dB (最大值的 $\frac{1}{10}$) 时对应的频率范围宽度。注意 $\text{BW} = \frac{f_0}{Q}$ ，因此放大器的带宽和矩形系数与谐振电路的品质因数 Q 密切相关。

4 实验内容与步骤

本次实验的注意内容为“放大器的频率特性及通频带的测量”。需注意：

- (1) 调整两级放大器的可调电容，使两级放大器的谐振频率分别略低于和略高于 10 MHz，但不能完全相等或相差过大；完全相等会导致频率响应曲线过于尖锐，难以测量通频带，差距过大则会导致增益明显下降，甚至出现中间窄两边高的“双峰现象”；
- (2) 为电路注入小信号时，输入信号幅度 (amplitude) 应控制在 30 mV 以内，过大会导致放大器进入非线性工作区，输出波形严重失真；

4.1 调整电路直流工作点

- (1) 连接好电路，为电路提供直流电源后，从 IN1 端输入 $f_0 = 10$ MHz 小信号，幅度在 10 mV 左右；
- (2) 在 OUT 端用示波器观测到放大后的输入信号，调整两个电位器 W1/W2 来改变两个晶体管的静态工作点。微调电容 C7/C12，使输出信号幅度最大且失真最小。

4.2 放大器的放大倍数及通频带的测试

- (1) 调整输入跳线 J1，使电路输入端为 IN (TP1) 而不是天线；
- (2) 测量电压增益：分别在 TP1 (IN) 和 TP3 (OUT) 处用示波器测量输入输出信号的幅度，计算电压增益；

- (3) 测量带宽: 在中心频率 f_0 附近改变输入频率, 例如 8 MHz ~ 12 MHz @ 0.1 MHz step, 重复步骤(1), 对结果进行拟合后计算 -3dB 带宽;
- (4) 测量矩形系数: 根据步骤(2)得到的结果, 计算 -20dB 带宽, 进而计算矩形系数。

数据记录表格如下:

Table 1: Resonant Amplifier Frequency Response Data

(信号发生器) f_{in} (MHz)	8.0	8.1	8.2	8.3	8.4	8.5	8.6	8.7	8.8	8.9
(AC of TP1) $V_{in,pp}$ (V)										
(AC of TP3) $V_{out,pp}$ (V)										
(信号发生器) f_{in} (MHz)	9.0	9.1	9.2	9.3	9.4	9.5	9.6	9.7	9.8	9.9
(AC of TP1) $V_{in,pp}$ (V)										
(AC of TP3) $V_{out,pp}$ (V)										10.0

5 思考题

5.1 小信号调谐放大器产生自激振荡的原因是什么? 如何避免产生自激振荡?

本实验所用的小信号谐振放大器由两级构成, 整体增益较高且 phase shift 较大。虽然电路中没有直接的反馈路径, 但由于电路布局和元器件的寄生参数(例如晶体管的集电极-基极结电容)等原因, 仍然可能产生寄生反馈, 一旦满足了自激振荡的起振条件 $|A_v(f_{osc})| = 1 @ \angle A_v(f_{osc}) = -180^\circ$, 就会导致放大器产生自激振荡, 输出端口出现持续的正弦波信号。此外, 如果放大器的谐振频率设置不当(例如两级谐振频率完全相等), 也可能导致频率响应曲线过于尖锐, 增加了自激振荡的风险。

可通过在输入或输出端添加补偿网络(例如 RC 网络)来降低避免自激振荡的产生。此外, 合理调整谐振频率, 使两级谐振频率略有差异, 也有助于降低自激振荡风险。

附录 C MATLAB Codes

```

1 %% NCE-03 Small-Signal Resonant Amplifier (2025.11.24)
2
3 %% 示波器通信调试
4 address = 'USB0::0x1AB1::0x04B0::DS2F203700277::INSTR';
5 depth_level = 2; % 0 ~ 4
6 flag_plot = 0;
7 ch = 1;
8 stc = MyOscilloscope_MS02202A_Read(address, ch, depth_level, flag_plot);
9 stc.stc_spectrum.plot4.axes.XLim = [0.1e6, 0.5e9];
10 amplitude = stc.stc_spectrum.STC.y_oneSided;
11 freq = stc.stc_spectrum.STC.fAxis_oneSided;
12 power = stc.stc_spectrum.STC.power_oneSided;
13 fr = stc.stc_spectrum.STC.fr;
14
15 amplitude = amplitude(2:end); freq = freq(2:end); % 不考虑 DC 量
16 [ymax, ind] = max(amplitude);
17 index = (ind-1):1:(ind+1);
18 input_freq = freq(index);
19 input_amp_mV = round(sqrt(sum(amplitude(index).^2))*1e3, 2)
20
21
22 ch = 2;
23 stc = MyOscilloscope_MS02202A_Read(address, ch, depth_level, flag_plot);
24 stc.stc_spectrum.plot4.axes.XLim = [0.1e6, 0.5e9];
25 amplitude = stc.stc_spectrum.STC.y_oneSided;
26 freq = stc.stc_spectrum.STC.fAxis_oneSided;
27
28 amplitude = amplitude(2:end); freq = freq(2:end); % 不考虑 DC 量
29 [ymax, ind] = max(amplitude);
30 index = (ind-3):1:(ind+3);
31 output_freq_MHz = freq(index)/1e6
32 output_amp_mV = round(sqrt(sum(amplitude(index).^2))*1e3, 2)
33
34
35 % MyExport_png(300)
36
37 ch = 2;
38 plot_flag = 1;
39 stc = MyOscilloscope_MS02202A_Read_OneCh(address, ch, plot_flag);
40
41
42 % stc.data
43
44 flag_plot = 0;
45 stc = MyOscilloscope_MS02202A_Read_TwoCh(address, flag_plot)
46 stc.myplot.leg.String = [
47     "CH1 (IN)"
48     "CH2 (OUT)"
49 ];
50 stc.myplot.axes.YLim = [-0.6 0.6];
51 stc.myplot.axes.YTick = -0.6:0.2:0.6;
52 stc.myplot.leg.Location = 'northeast';
53
54 %% NCE-03 示波器数据读取
55 address = 'USB0::0x1AB1::0x04B0::DS2F203700277::INSTR';
56 depth_level = 2; % 0 ~ 4
57 flag_plot = 0;
58 ch = 1;
59 stc = MyOscilloscope_MS02202A_Read(address, ch, depth_level, flag_plot);

```

```

60 stc.stc_spectrum.plot4.axes.XLim = [0.1e6, 0.5e9];
61 amplitude = stc.stc_spectrum.STC.y_oneSided;
62 freq = stc.stc_spectrum.STC.fAxis_oneSided;
63 power = stc.stc_spectrum.STC.power_oneSided;
64 fr = stc.stc_spectrum.STC.fr;
65
66 amplitude = amplitude(2:end); freq = freq(2:end); % 不考慮 DC 量
67 [ymax, ind] = max(amplitude);
68 index = (ind-1):1:(ind+1);
69 freq_MHz = freq(index)/1e6;
70 input_amp_mV = round(sqrt(sum( amplitude(index).^2 ))*1e3, 2)
71
72
73 ch = 2;
74 stc = MyOscilloscope_MS02202A_Read(address, ch, depth_level, flag_plot);
75 stc.stc_spectrum.plot4.axes.XLim = [0.1e6, 0.5e9];
76 amplitude = stc.stc_spectrum.STC.y_oneSided;
77 freq = stc.stc_spectrum.STC.fAxis_oneSided;
78
79 amplitude = amplitude(2:end); freq = freq(2:end); % 不考慮 DC 量
80 [ymax, ind] = max(amplitude);
81 index = (ind-3):1:(ind+3);
82 freq_MHz = freq(index)/1e6;
83 output_amp_mV = round(sqrt(sum( amplitude(index).^2 ))*1e3, 2)
84
85 freq_MHz_now = freq_MHz_now + 0.1
86 in = [in, input_amp_mV]
87 out = [out, output_amp_mV]
88
89
90 %% NCE-03 數據作圖
91 clc, clear
92 freq = (8.0:0.1:11.5)*1e6;
93
94 % Vs = 50 mVpp = 25 mVamp
95 in_amp_mV = [
96 18.06 17.10 17.82 17.80 16.98 18.44 17.16 17.87 17.80 16.87 ...
97 17.80 16.16 16.04 14.54 11.69 09.25 06.12 06.70 08.69 10.06 12.21 ...
98 12.36 13.67 14.20 13.88 14.79 13.46 13.27 ...
99 12.60 11.62 12.31 11.53 12.17 12.25 11.82 12.68
];
100
101 out_amp_mV = [
102 080.35 083.34 095.98 107.52 116.61 134.05 144.63 162.83 180.40 196.06 ...
103 222.79 240.55 271.32 300.22 324.90 364.73 382.91 416.16 444.86 471.92 528.78 ...
104 567.31 641.76 716.66 782.39 874.91 875.35 849.76 ...
105 750.80 615.83 513.84 405.64 338.35 280.78 232.35 203.99
];
106
107 stc = MyPlot(freq/1e6, [in_amp_mV*20; out_amp_mV])
108 stc.plot.plot_1.LineStyle = '-.';
109 stc.axes.YLim = [0 1000];
110
111
112 yyaxis('left')
113 stc.label.y.String = 'Voltage Amplitude (mV)';
114 stc.axes.YColor = 'b';
115
116 yyaxis('right')
117 stc = MyPlot_ax(stc.axes, freq/1e6, [out_amp_mV./in_amp_mV; out_amp_mV./50]);
118 stc.plot.plot_1.Color = 'r';
119 stc.plot.plot_2.Color = 'm';
120 stc.plot.plot_2.LineStyle = '-';
121

```

```

122 stc.leg.String = [
123     "Scaled Input Amplitude $10 \times v_{in,amp} (mV)";
124     "Output Amplitude $v_{out,amp} (mV)";
125     "Voltage Gain $A_v = \frac{v_{out,amp}}{v_{in,amp}} (V/V)"
126     "Signal Gain $A_s = \frac{v_{out,amp}}{v_{sig,amp}} (V/V)"
127 ];
128 stc.leg.Location = 'northwest';
129 stc.axes.YLim = [0 100];
130 stc.axes.YColor = 'm';
131 stc.axes.Title.String = 'Signal Amplitude $v_{sig,amp} = 25 mV';
132 stc.label.y.String = 'Voltage Gain (V/V)';
133 stc.label.x.String = 'Frequency (MHz)';
134
135
136 MyFigure_ChangeSize([2.2, 1]*512)
137
138 %% 对 signal gain (A_s) 进行拟合, 求出 BW_-3dB, BW_-20dB 和 SF
139 %X = freq;
140 As = out_amp_mV./50;
141
142
143 [xData, yData] = prepareCurveData( freq, As );
144
145 % 设置 fittype 和选项。
146 ft = fittype('gauss2');
147 opts = fitoptions('Method', 'NonlinearLeastSquares');
148 opts.Display = 'Off';
149 opts.Lower = [-Inf -Inf 0 -Inf -Inf 0];
150 opts.MaxFunEvals = 2000;
151 opts.MaxIter = 1000;
152 opts.StartPoint = [17.507 10600000 289126.678825008 10.4663782976351 10100000 368362.996025349];
153 opts.TolFun = 1e-10;
154 opts.TolX = 1e-08;
155
156 % 对数据进行模型拟合。
157 [As_fit, gof] = fit( xData, yData, ft, opts );
158
159 freq_array = 8e6:10e3:12e6;
160 stc = MyPlot(freq_array/1e6, As_fit(freq_array));
161 stc = MyScatter_ax(stc.axes, freq/1e6, As);
162 stc.scatter.scatter_1.MarkerEdgeColor = 'r';
163 stc.scatter.scatter_1.SizeData = 500;
164 stc.axes.YLim = [0 20];
165 MyFigure_ChangeSize([3.5, 1]*512)
166 stc.leg.String = [
167     "Fitted Curve"
168     "Raw Data"
169 ];
170 stc.label.y.String = 'Signal Gain $A_s = \frac{v_{out,amp}}{v_{sig,amp}} (V/V)';
171 stc.label.x.String = 'Frequency (MHz)';
172
173 freq_array = linspace(8e6, 12e6, 3214)
174 [As_max, ind] = max(As_fit(freq_array))
175 As_max_dB = 20*log10(As_max)
176 f_maxAs = freq_array(ind);
177 f_maxAs_MHz = f_maxAs/1e6
178
179 f_3dB_L = fzero(@(x) As_fit(x) - As_max/sqrt(2), 8e6)
180 f_3dB_H = fzero(@(x) As_fit(x) - As_max/sqrt(2), 12e6)
181 f_20dB_L = fzero(@(x) As_fit(x) - As_max/10, 8e6)
182 f_20dB_H = fzero(@(x) As_fit(x) - As_max/10, 12e6)
183

```

```
184
185 BW_3dB_MHz = (f_3dB_H - f_3dB_L)/1e6
186 BW_20dB_MHz = (f_20dB_H - f_20dB_L)/1e6
187 SF_20dB = BW_20dB_MHz/BW_3dB_MHz
188 Q = f_maxAs_MHz/BW_3dB_MHz
189
190
191
192 %MyAnalysis_Spectrum_v1_20251124
193
194 %% NCE-03 调整工作点 (电阻和电容) 后的数据作图
195 clc, clear
196 in_amp_mV = [
197 21.8000
198 20.4500
199 21.4500
200 21.5300
201 20.6900
202 22.0600
203 20.1800
204 19.8400
205 17.1700
206 11.9500
207 8.3400
208 8.6200
209 11.4500
210 13.2700
211 13.9100
212 15.8000
213 15.4600
214 16.6800
215 17.1200
216 16.6400
217 17.6900
218 15.8800
219 15.6700
220 15.0900
221 14.2500
222 15.2600
223 14.3100
224 15.1100
225 15.1600
226 14.5900
227 15.6500
228 14.6200
229 15.2700
230 15.2900
231 14.6000
232 15.6000
233 ]';
234
235 out_amp_mV = 1e3*[
```

0.1611
0.1756
0.1995
0.2229
0.2456
0.2831
0.3090
0.3525
0.3910
0.4202

```
246     0.4599
247     0.4668
248     0.4946
249     0.5235
250     0.5524
251     0.6172
252     0.6606
253     0.7494
254     0.8482
255     0.9487
256     1.0716
257     1.0475
258     0.9552
259     0.7844
260     0.6083
261     0.4914
262     0.3817
263     0.3152
264     0.2611
265     0.2161
266     0.1895
267     0.1589
268     0.1405
269     0.1239
270     0.1079
271     0.0983
272 ];
273
274 freq = (8.0:0.1:11.5)*1e6;
275 stc = MyPlot(freq/1e6, [in_amp_mV*20; out_amp_mV])
276 stc.plot.plot_1.LineStyle = '-.';
277 stc.axes.YLim = [0 1200];
278
279
280 yyaxis('left')
281 stc.label.y.String = 'Voltage Amplitude (mV)';
282 stc.axes.YColor = 'b';
283
284 yyaxis('right')
285 stc = MyPlot_ax(stc.axes, freq/1e6, [out_amp_mV./in_amp_mV; out_amp_mV./50]);
286 stc.plot.plot_1.Color = 'r';
287 stc.plot.plot_2.Color = 'm';
288 stc.plot.plot_2.LineStyle = '-';
289 stc.leg.String =
290     "Scaled Input Amplitude $10 \times v_{in,amp} (mV)";
291     "Output Amplitude $v_{out,amp} (mV)";
292     "Voltage Gain $A_v = \frac{v_{out,amp}}{v_{in,amp}} (V/V)"
293     "Signal Gain $A_s = \frac{v_{out,amp}}{v_{sig,amp}} (V/V)"
294 ];
295 stc.leg.Location = 'northwest';
296 stc.axes.YLim = [0 100];
297 stc.axes.YColor = 'm';
298 stc.axes.Title.String = 'Signal Amplitude $v_{sig,amp} = 25 mV';
299 stc.label.y.String = 'Voltage Gain (V/V)';
300 stc.label.x.String = 'Frequency (MHz)';
301
302
303 MyFigure_ChangeSize([2.2, 1]*512)
304
305
306 %% 窗函数与目标信号幅度/功率/频率计算
307
```

```

308 clc, clear
309 f = 1e3;
310 T = 1/f;
311 omega = 2*pi*f;
312 Je_rms_mUI = 5;
313 % 带有 AM/PM noise + guass voltage noise + 杂七杂八噪声的信号
314 v_t = @(t) (1 + rand/10)*cos(omega * (t + randn*(Je_rms_mUI/1e3*T)) ) ...
315     + (0.1 + rand/10)*cos(2*omega * (t + randn*(Je_rms_mUI/1e3*T)) ) ...
316     + rand/10*cos(rand(size(t)) .* 10*omega .* (t + randn*(Je_rms_mUI/1e3*T)) ) + rand/10;
317
318 % 对信号进行采样
319 Je_rms_mUI = 10;
320 fs = f*(pi*10);
321 %fs = f*(32);
322 N = 2048*4;
323 t_n = 0:1/fs:(N-1)/fs;
324 v_n = v_t(t_n)
325
326 time_array = 0:0.1/fs:(N-1)/fs;
327 MyPlot(time_array(1:3000), v_t(time_array(1:3000)));
328
329 % 不加窗直接作 DFT (window = none, namely uniform)
330 stc1 = MyAnalysis_Spectrum_v1_20251124(v_n, fs, 1, 10);
331 stc1.plot4.axes.YLim = [-100 0];
332 %MyExport_png(100)
333
334 % 作 DFT 之前先对时域信号加 Hanning 窗
335 window = hannning(N, 'periodic');
336 v_n_hanning = v_n .* window';
337 % 注意需要乘上 ACF/ECF 才能得到修正后的 amplitude/power spectrum
338 stc2 = MyAnalysis_Spectrum_v1_20251124(v_n_hanning, fs, 1, 10);
339 stc2.plot4.axes.YLim = [-100 0];
340 %MyExport_png(100)
341
342
343 % 作 DFT 之前先对时域信号加 flat top 窗
344 window = flattopwin(N, 'periodic');
345 v_n_flattop = v_n .* window';
346 stc3 = MyAnalysis_Spectrum_v1_20251124(v_n_flattop, fs, 1, 10);
347 stc3.plot4.axes.YLim = [-100 0];
348 %MyExport_png(100)
349
350
351 % 功率谱对比
352 freq = stc1.STC.fAxis_oneSided;
353 y1 = stc1.STC.y_oneSided;
354 y2 = stc2.STC.y_oneSided;
355 y3 = stc3.STC.y_oneSided;
356 p1_dB = 10*log10(stc1.STC.power_oneSided);
357 p2_dB = 10*log10(stc2.STC.power_oneSided);
358 p3_dB = 10*log10(stc3.STC.power_oneSided);
359
360 stc = MyPlot(freq, [p1_dB; p2_dB; p3_dB])
361 stc.axes.XScale = 'log';
362 stc.label.y.String = 'Power $\mathbf{dB(V^2)}$';
363 stc.label.x.String = 'Frequency (Hz)';
364 stc.leg.String = [
365     "None (Uniform)"
366     "Hanning (Hann)"
367     "Flat Top"
368 ];
369 ylim([-100 0])

```

```

370 xlim([1e3/5, 1e3*5])
371 stc.leg.Location = 'northeast';
372
373 MyFigure_ChangeSize([2.2, 1]*512)
374 %MyExport_png(100)
375
376 %% 估计信号频率和幅度/功率
377 %% 积分法
378
379 L = 5;
380 B_en = [
381     1.00    % Uniform
382     1.50    % Hanning
383     3.77    % Flat Top
384 ];
385 num = 0;
386
387
388 %% amplitude/energy correction factor
389 % Calculate windowing correction factor
390 % https://www.mathworks.com/matlabcentral/answers/372516-calculate-windowing-correction-factor
391 N = stc1.STC.N;
392 i = 1;
393 for w = [1 + zeros(N, 1), hanning(N), flattopwin(N)]
394     if i == 1
395         ECF = 1/rms(w)^2;
396         ACF = 1/mean(w);
397     else
398         ECF = [ECF, 1/rms(w)^2];
399         ACF = [ACF, 1/mean(w)];
400     end
401
402     i = i + 1;
403 end
404 ECF
405 ACF
406 BW_n_nor = ACF.^2./ECF
407
408
409 disp('-----')
410 disp(['L = ', num2str(L)])
411 for stc = [stc1, stc2, stc3]
412     num = num + 1
413     [ymax, ind] = max(stc.STC.y_oneSided(2:end));
414     index = 1 + (ind-L:1:ind+L);
415     ymax_array = stc.STC.y_oneSided(index);
416     pmax_array = stc.STC.power_oneSided(index);
417     fr = stc.STC.fr;
418
419     %P0 = sum(pmax_array)/B_en(num)
420     P0 = ECF(num) * sum(pmax_array)
421     A0 = sqrt(2 * ECF(num) * sum(pmax_array))
422     sqrt(2*sum(pmax_array))
423     ACF(num) * sqrt(2*sum(pmax_array))
424     ACF(num) * sqrt(2*sum(pmax_array) / B_en(num) )
425     f0 = sum(pmax_array.*((index - 1)*fr)./sum(pmax_array));
426     f0 = vpa(f0)
427 end
428
429
430 L = 200; num = 0;
431

```

```

432 disp('-----')
433 disp(['L = ', num2str(L)])
434 for stc = [stc1, stc2, stc3]
435     num = num + 1
436     [ymax, ind] = max(stc.STC.y_oneSided(2:end));
437     index = 1 + (ind-L:1:ind+L);
438     ymax_array = stc.STC.y_oneSided(index);
439     pmax_array = stc.STC.power_oneSided(index);
440     fr = stc.STC.fr;
441
442     P0 = ECF(num) * sum(pmax_array)
443     A0 = sqrt(2*P0)
444     sqrt(2*sum(pmax_array))
445     ACF(num) * sqrt(2*sum(pmax_array))
446     ACF(num) * sqrt(2*sum(pmax_array) / B_en(num) )
447     f0 = sum(pmax_array.*((index - 1)*fr)./sum(pmax_array));
448     f0 = vpa(f0)
449     power_total = ECF(num) * sum( stc.STC.volt_n.^2)/N
450 end
451
452 %% 不同 window 用积分法的 amp 与 power 提取结果对比
453
454 % 实际数据处理版本 (需要您提供stc1, stc2, stc3变量)
455 L_values = [0, 1, 2, 3, 5, 10, 20, 50, 100, 200];
456 % B_en = [1.00, 1.50, 3.77]; % Uniform, Hanning, Flat Top
457 % window_names = {'Uniform', 'Hanning', 'Flat Top'};
458 window_names = ["Uniform", "Hanning", "Flat Top"]
459
460 % 初始化结果存储
461 results_table = table();
462 results_cell = cell(1, 5);
463 results_cell{1, 1} = 'L';
464 results_cell{1, 2} = 'Window';
465 results_cell{1, 3} = 'P0';
466 results_cell{1, 4} = 'A0';
467 results_cell{1, 5} = 'f0';
468
469 row_idx = 2;
470
471 for i = 1:length(L_values)
472     L = L_values(i);
473
474     num = 0;
475     for stc = [stc1, stc2, stc3]
476         num = num + 1;
477
478         [ymax, ind] = max(stc.STC.y_oneSided(2:end));
479         index = 1 + (ind-L:1:ind+L);
480         ymax_array = stc.STC.y_oneSided(index);
481         pmax_array = stc.STC.power_oneSided(index);
482         fr = stc.STC.fr;
483
484         P0 = ECF(num) * sum(pmax_array);
485         A0 = sqrt(2 * P0);
486         f0 = sum(pmax_array .* ((index - 1) * fr)) / sum(pmax_array);
487
488         % 添加到table
489         new_row = table(L, P0, A0, f0, ...
490                         'VariableNames', {'L', 'P0', 'A0', 'f0'});
491         if isempty(results_table)
492             results_table = new_row;

```

```

494     else
495         results_table = [results_table; new_row];
496     end
497
498     % 添加到cell
499     results_cell{row_idx, 1} = L;
500     results_cell{row_idx, 2} = window_names{num};
501     results_cell{row_idx, 3} = P0;
502     results_cell{row_idx, 4} = A0;
503     results_cell{row_idx, 5} = f0;
504
505     row_idx = row_idx + 1;
506
507     fprintf('-----\n');
508     fprintf('L = %d, Window = %s\n', L, window_names{num});
509     fprintf('P0 = %.6f\n', P0);
510     fprintf('A0 = %.6f\n', A0);
511     fprintf('f0 = %.6f\n', f0);
512 end
513
514 % 显示最终结果
515 fprintf('\n===== 最终结果汇总 ======\n');
516 disp(results_table);
517
518 fprintf('\n===== Cell格式结果 ======\n');
519 disp(results_cell);
520
521 % 作图
522 re = table2array(results_table)
523
524 P0_matrix = zeros(3, length(L_values));
525 A0_matrix = zeros(3, length(L_values));
526 f0_matrix = zeros(3, length(L_values));
527
528 for i = 1:3
529     P0_matrix(i, :) = re(i-1 + (1:3:end), 2)';
530     A0_matrix(i, :) = re(i-1 + (1:3:end), 3)';
531     f0_matrix(i, :) = re(i-1 + (1:3:end), 4)';
532 end
533
534 %A0_matrix
535
536 L_values(1) = 1e-1;
537
538 tiledlayout(3, 1)
539
540 ax = nexttile;
541 stc = MyPlot_ax(ax, L_values, f0_matrix);
542 stc.axes.XScale = 'log';
543 stc.leg.String = [
544     "Uniform"
545     "Hanning"
546     "Flat Top"
547 ];
548 stc.label.x.String = 'Integral Length $L$';
549 stc.label.y.String = 'Extracted Frequency $f_0 \ \mathrm{(Hz)}$ ';
550 ylim([1e3 - 1 1e3 + 1])
551 stc.leg.Location = 'southeast';
552
553 ax = nexttile;
554 stc = MyPlot_ax(ax, L_values, P0_matrix);
555 stc.axes.XScale = 'log';

```

```

556 stc.leg.String = [
557     "Uniform"
558     "Hanning"
559     "Flat Top"
560 ];
561 stc.label.x.String = 'Integral Length $L$';
562 stc.label.y.String = 'Extracted Power $P_0\ \mathrm{(V_{rms}^2)}$ ';
563 stc.leg.Location = 'southeast';
564
565 ylim([0 0.6])
566
567
568 ax = nexttile;
569 stc = MyPlot_ax(ax, L_values, A0_matrix);
570 stc.axes.XScale = 'log';
571 stc.label.x.String = 'Integral Length $L$';
572 stc.label.y.String = 'Extracted Amplitude $A_0\ \mathrm{(V_{amp})}$';
573 stc.leg.String = [
574     "Uniform"
575     "Hanning"
576     "Flat Top"
577 ];
578 stc.leg.Location = 'southeast';
579
580 ylim([0 1.2])
581
582 MyFigure_ChangeSize([2.2, 2.1]*512)
583
584 %% 构建多 window DFT 函数
585
586 stc = MyAnalysis_Spectrum_multiWindow(v_n, fs, 1);
587 % stc.STC2_hanning.volt_n
588
589
590 %% FS 方法
591
592 % 参数设置
593 f0 = 1000;           % 基频 1 kHz
594 fr = stc1.STC.fr;
595
596 % 生成测试信号 (包含基波和几次谐波)
597 t = 0:1/fs:0.01-1/fs; % 10ms 数据
598 N = length(t);
599
600 % 执行傅里叶级数展开
601 stc = MyAnalysis_DiscreteFourierSeries(v_n, fs, f0, fr);
602
603 % 计算幅度和相位
604 stc.a_0
605 stc.amplitude_k

```