

# 《非线性电路实验》实验报告

实验名称: Amplitude Demodulation 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: 2025.12.11 实验地点: 西实验楼(8号楼) 308 是否调课/补课: 否 成绩:   

---

## 目录

<b>1 实验目的</b>	<b>1</b>
<b>2 实验仪器</b>	<b>1</b>
<b>3 实验原理</b>	<b>2</b>
3.1 同步幅度解调基本原理 (Synchronous AM Demodulation) . . . . .	2
3.2 低通滤波器的作用 . . . . .	2
3.3 同步解调与包络检波的比较 . . . . .	2
3.4 实验电路简要分析 . . . . .	3
<b>4 实验内容与步骤</b>	<b>3</b>
4.1 实验前准备与连接 . . . . .	3
4.2 解调过程观测与调试 . . . . .	3
<b>5 实验结果与分析</b>	<b>4</b>
<b>6 思考题</b>	<b>5</b>
6.1 分析调幅波同步解调方法的优缺点。 . . . . .	5
6.2 为什么解调信号在经过滤波后出现了“反相”( $180^\circ$ 相移)? . . . . .	5
<b>附录 A 原始数据记录表</b>	<b>6</b>
<b>附录 B 实验预习报告</b>	<b>6</b>
<b>附录 C MATLAB Codes</b>	<b>10</b>

---

## 1 实验目的

- (1) 掌握基于集成模拟乘法器的幅度调制(AM)原理与实现方法。
- (2) 理解AM信号频谱结构、功率分配及调制系数(Modulation Index)的影响。
- (3) 掌握使用MC1496四象限模拟乘法器实现AM调制的电路配置与调试方法。
- (4) 学会在示波器上测量调制系数并分析基带信号(原始信号)、载波与已调波之间的关系。

## 2 实验仪器

- (1) 高频实验箱 - 乘法调幅/混频实验板 (031132201809392)
- (2) 示波器 RIGOL MSO2202A (080103201901376)

- (3) 信号发生器 GWINSTEK AFG-2225 (080102201901355)
- (4) 万用表 LINIT- UT61A (C181503983)

### 3 实验原理

#### 3.1 同步幅度解调基本原理 (Synchronous AM Demodulation)

设已接收的标准 AM 信号 (来自上一个实验的输出) 为:

$$v_{AM}(t) = A_c [1 + m \cos(\omega_s t)] \cos(\omega_c t) \quad (1)$$

其中,  $f_s$  为基带信号频率,  $f_c$  为载波频率,  $m$  为调制系数。同步解调的核心思想是: 在接收端产生一个与发射端载波同频率且同相位的本地载波 (Local Carrier, 也称恢复载波或参考信号)  $v_r(t)$ :

$$v_r(t) = A_r \cos(\omega_c t + \Delta\phi) \quad (2)$$

理想情况下, 相位差  $\Delta\phi = 0$ , 即完全同步。将调制信号  $v_{AM}(t)$  与恢复载波  $v_r(t)$  输入模拟乘法器进行相乘, 利用三角恒等式  $\cos^2(\omega_c t) = \frac{1}{2}[1 + \cos(2\omega_c t)]$ , 得到:

$$v_{out}(t) = k \cdot v_{AM}(t) \cdot v_r(t) = k A_c A_r [1 + m \cos(\omega_s t)] \cos^2(\omega_c t) \quad (3)$$

$$= \underbrace{\frac{k A_c A_r}{2} [1 + m \cos(\omega_s t)]}_{\text{低频分量 (包含直流与基带信号)}} + \underbrace{\frac{k A_c A_r}{2} [1 + m \cos(\omega_s t)] \times \cos(2\omega_c t)}_{\text{高频分量 (中心频率为 } 2f_c \text{)}}$$

如果相位不同步, 即  $\Delta\phi \neq 0$ , 则乘法器输出变为:

$$v_{out}(t) = k \cdot v_{AM}(t) \cdot v_r(t) = k A_c A_r [1 + m \cos(\omega_s t)] \cos(\omega_c t) \cos(\omega_c t + \Delta\phi) \quad (5)$$

$$= k A_c A_r [1 + m \cos(\omega_s t)] \cos(\omega_c t) [\cos(\Delta\phi) \cos(\omega_c t) - \sin(\Delta\phi) \sin(\omega_c t)] \quad (6)$$

$$= \cos(\Delta\phi) k A_c A_r [1 + m \cos(\omega_s t)] \cos^2(\omega_c t) - \sin(\Delta\phi) k A_c A_r [1 + m \cos(\omega_s t)] \cos(\omega_c t) \sin(\omega_c t) \quad (7)$$

$$= \underbrace{\frac{k A_c A_r \cos(\Delta\phi)}{2} [1 + m \cos(\omega_s t)]}_{\text{低频分量 (包含直流与基带信号)}} + \underbrace{\frac{k A_c A_r}{2} [1 + m \cos(\omega_s t)] \times [\cos(\Delta\phi) \cos(2\omega_c t) - \sin(\Delta\phi) \sin(2\omega_c t)]}_{\text{高频分量 (中心频率为 } 2f_c \text{)}}$$

由此看出相位误差  $\Delta\phi$  并不会从根本上影响解调结果, 但会使低频分量的幅度降低, 系统信噪比稍有下降。值得一提的是, 上述基于乘法器的同步解调方法适用于绝大多数 Amplitude Modulation 形式, 通用性较强。

#### 3.2 低通滤波器的作用

乘法器输出  $v_{out}(t)$  包含两个主要频率成分:

(1) **低频分量 (Low-Frequency Component):** 包含直流成分和原始基带信号  $\cos(\omega_s t)$ 。

(2) **高频分量 (High-Frequency Component):** 中心频率在  $2f_c$ , 其幅度受基带信号调制。

通过一个高边截止频率  $f_H$  满足  $f_s \ll f_H \ll 2f_c$  的低通滤波器, 可以很好地滤除高频分量 ( $2f_c$  附近):

$$v_{out, LPF}(t) = \frac{k A_c A_r}{2} [1 + m \cos(\omega_s t)] \quad (9)$$

该信号包含一个直流偏移  $\frac{k A_c A_r}{2}$  和放大后的原始基带信号  $\frac{k A_c A_r m}{2} \cos(\omega_s t)$ 。通过隔直电容或后级电路即可去除直流, 最终恢复出纯净的基带信号。

#### 3.3 同步解调与包络检波的比较

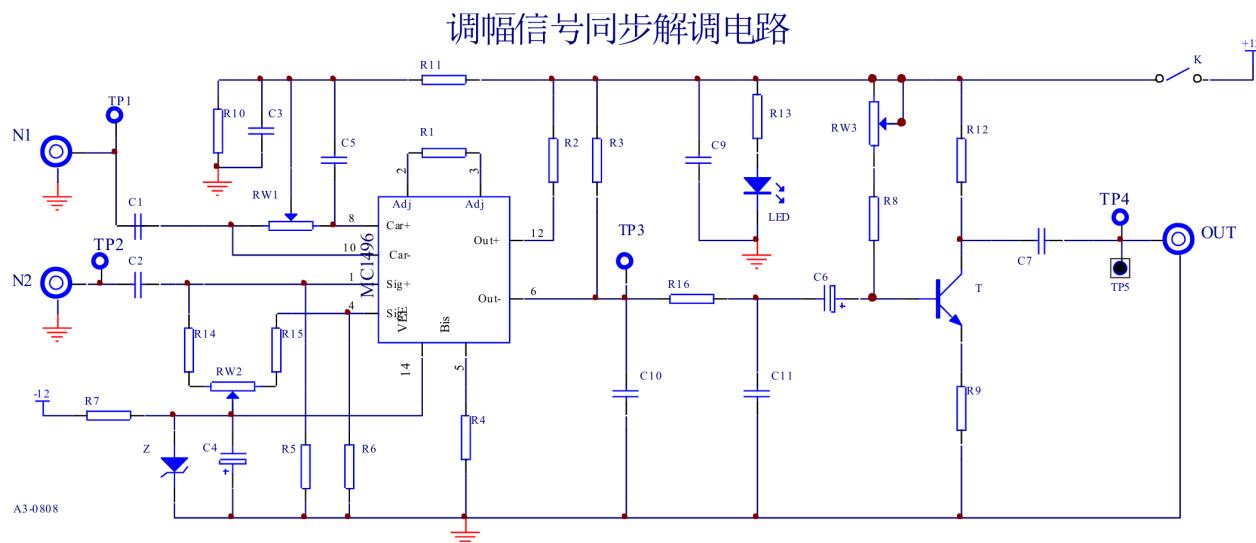
(1) **乘法器同步解调 (Synchronous Demodulation):** **优点:** 解调线性度好, 适用于低调制系数 ( $m \ll 1$ ) 及过调制 ( $m > 1$ ) 情况; 能解调所有类型的幅度调制信号 (包括 DSB-SC、SSB)。**缺点:** 需要产生与发射端严格同步的本地载波, 系统复杂度高。

(2) 二极管包络解调 (Envelope Detection): 优点: 电路极其简单(仅需一个二极管、电阻和电容), 无需本地载波。缺点: 仅适用于幅度较大的标准AM信号( $m \leq 1$ ); 解调效率低, 存在门限效应和失真; 无法解调DSB-SC或SSB信号。

### 3.4 实验电路简要分析

如 Figure 1 所示，本次实验的同步解调电路同样以 MC1496 为核心，结构与调制电路类似但功能不同：

- (1) **输入端口 IN1:** 接入本地恢复载波  $v_r(t)$  (将调整电路中所用的载波信号输入此端);
  - (2) **输入端口 IN2:** 接入待解调的 AM 信号  $v_{AM}(t)$  (来自调制实验板的 OUT 端);
  - (3) **核心乘法器:** MC1496 实现  $v_{AM}(t) \times v_r(t)$  的相乘操作;
  - (4) **低通滤波器:** 由 R16, C10, C11 构成无源低通滤波器, 用于滤除  $2f_c$  高频分量;
  - (5) **输出级:** 采用 Common Emitter 放大电路作为输出级, 提高信号幅度的同时隔离负载影响;
  - (6) **输出端口 TP3:** 乘法器直接输出端, 可观测未滤波的混合信号(包含低频和高频分量);
  - (7) **输出端口 TP4:** 经过低通滤波器后的输出, 即为恢复后的基带信号。



**Figure 1: MC1496-Based Synchronous AM Demodulation Circuit Used in the Experiment**

## 4 实验内容与步骤

#### 4.1 实验前准备与连接

- (1) 在实验箱上接入乘法器幅度调制实验板和调幅信号同步解调实验板;
  - (2) 按上一个实验的步骤, 在调制板上产生一个调制系数  $m \approx 0.5$  的 AM 信号;
  - (3) 将调制板 OUT 端产生的 AM 信号接入解调板的 IN2 端;
  - (4) 将调制板所用的 10.7 MHz 载波信号(信号源或晶体振荡器输出)接入解调板的 IN1 端作为本地载波。

## 4.2 解调过程观测与调试

- (1) 观测未滤波信号 (TP3): 将示波器通道 1 接 TP3, 调整解调板上的电位器 RW1 和 RW2, 使波形稳定。此时应观察到包含低频包络和高频载波分量的混合信号。
  - (2) 观测滤波后信号 (TP4): 将示波器通道 2 接 TP4, 观察经过低通滤波器后的输出, 应为一个频率为 1 kHz 的正弦波(可能带有少量残余高频纹波); 微调 RW1, RW2 使 TP4 端的正弦幅度最大、失真最小。
  - (3) 不同调制系数下的解调: 在幅度调制板中, 依次将调制系数  $m$  调节至约 0.3、0.5、0.8, 对每一个  $m$  值, 在解调板上记录 TP3 和 TP4 的波形; 对比不同  $m$  下, 解调输出波形(TP4)的幅度和噪声变化。

## 5 实验结果与分析

为解调电路设置合适工作点之后，调整幅度调制电路使输出法信号调幅系数约为 0.5，此时的调幅结果与解调结果如 Figure 2 所示：

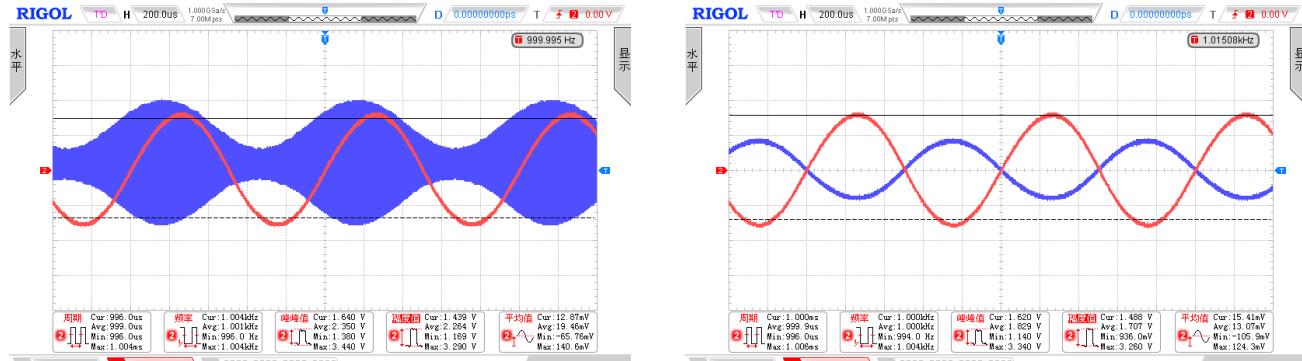


Figure 2: The measured AM modulation and demodulation results with moderate modulation index ( $m \approx 0.5$ ).

改变调幅系数后继续进行测量，结果如 Figure 3 和 Figure 4 所示：

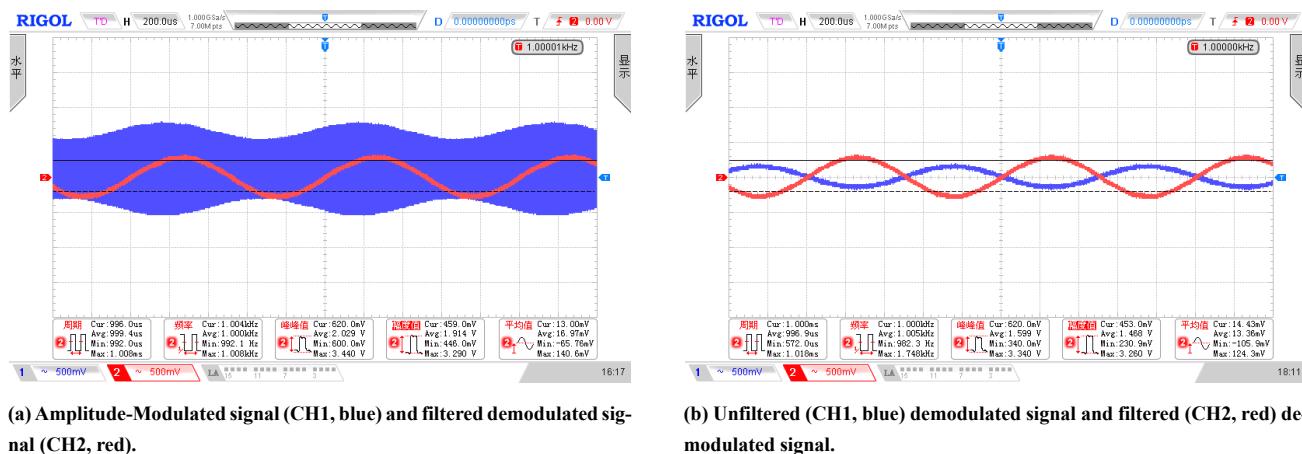


Figure 3: The measured AM modulation and demodulation results with small modulation index ( $m \approx 0.2$ ).

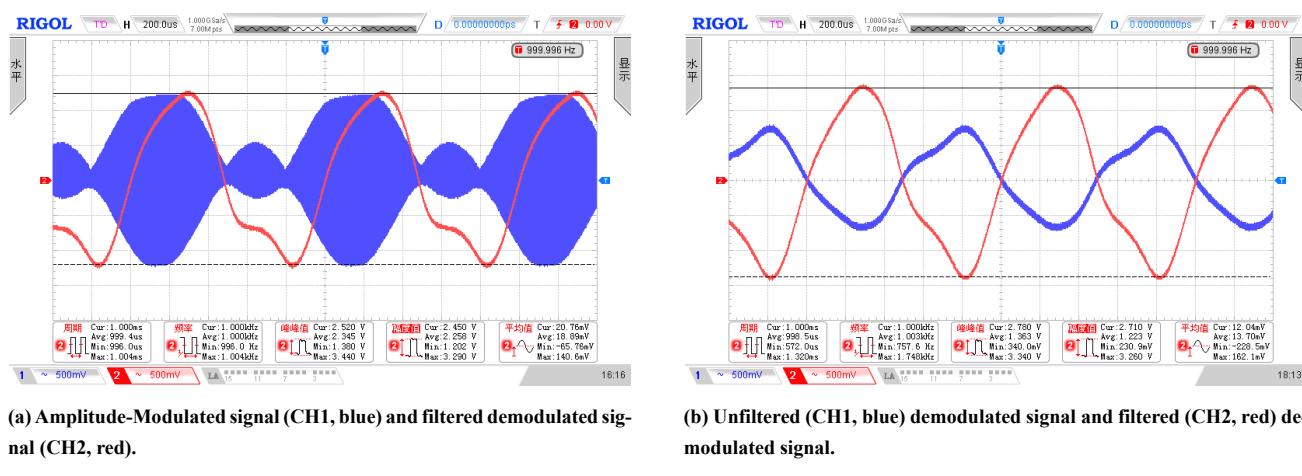


Figure 4: The measured AM modulation and demodulation results with large modulation index ( $m > 1$ ).

可以看到，经过低通滤波后的输出波形更加“纯净”，波形上的高频毛刺大多被消除，即附在信号上的高频分量基本被滤除。需要指出的是，通过低通滤波前的解调信号应该还含有直流分量，但我们测量时将示波器设置为了交流耦合，因此没有观察到直流分量。

在不同的调幅系数下信号均能被解调，但调幅系数过大或过小都会导致解调输出的质量下降。在调幅系数  $m > 1$  时，解调输出出现了明显失真，波形顶部被削平 (Clipping)；而在调幅系数较小时，解调输出的幅度较小，信噪比较低。

上图还可以观察到，解调后的信号经过低通滤波后，高频毛刺明显减少，信号更加纯净。

## 6 思考题

### 6.1 分析调幅波同步解调方法的优缺点。

调幅波同步解调方法具有一系列显著优点。首先，在理想同步条件下，其解调输出与输入调制信号之间呈现严格的线性关系，失真度极低，保证了高线性度与高保真度。其次，该方法的适用性非常广泛，不仅能解调标准 AM 信号，还能有效处理抑制载波的双边带 (DSB-SC) 和单边带调制 (SSB) 等复杂调制信号。此外，与包络检波相比，同步解调不存在“门限效应”，在低信噪比环境下依然能保持优良的接收性能。最后，该方法对过调制状态具有很好的鲁棒性，即使调制系数  $m > 1$ ，只要载波同步良好，依然能够正确解调出原始信号。

然而，同步解调方法也存在若干缺点，其中最主要的一点是需要实现精确的载波同步，即接收端必须产生一个与发射端载波严格同频同相的本地载波。这通常需要依赖锁相环 (PLL) 等复杂的载波生成电路，显著增加了系统的硬件成本与整体复杂度。其次，该方法的性能对相位噪声高度敏感，本地载波的相位噪声  $\phi_n$  会通过幅度系数  $\cos(\Delta\phi)$  直接耦合到解调后的基带信号中。最后，从电路实现层面看，同步解调需要模拟乘法器和低通滤波器等组件，其电路结构相比简单的包络检波解调电路要复杂得多。

总的来讲，同步解调主要应用于对解调质量要求严苛、信号形式复杂 (如 DSB-SC、SSB) 或工作于低信噪比环境的通信系统中，例如专业无线电、卫星通信以及数字通信的相干解调等场合。相比之下，包络检波则因其极简的电路结构，被广泛应用于对成本敏感、且只需处理标准 AM 信号的广播接收机等场景。

### 6.2 为什么解调信号在经过滤波后出现了“反相”( $180^\circ$ 相移)？

查看原理图发现，解调电路中的滤波部分内含一个 Common Emitter (CE) 放大器，放大解调信号在滤波后经过 CE 放大器，因此出现了  $180^\circ$  的相移。

## 附录 A 原始数据记录表

注：本次实验并未产生具体数据，仅通过示波器观察波形变化，已在正文部分给出相关图像，此处不再重复记录。

## 附录 B 实验预习报告

# 《非线性电路实验》预习报告

实验名称: Amplitude Demodulation 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: 2025.12.11 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) 掌握基于集成模拟乘法器的幅度调制(AM)原理与实现方法。
- (2) 理解AM信号频谱结构、功率分配及调制系数(Modulation Index)的影响。
- (3) 掌握使用MC1496四象限模拟乘法器实现AM调制的电路配置与调试方法。
- (4) 学会在示波器上测量调制系数并分析基带信号(原始信号)、载波与已调波之间的关系。

## 2 实验仪器

- (1) 高频实验箱 - 乘法调幅/混频实验板 (031132201809392)
- (2) 示波器 RIGOL MSO2202A (080103201901376)
- (3) 信号发生器 GWINSTEK AFG-2225 (080102201901355)
- (4) 万用表 LINIT- UT61A (C181503983)

## 3 实验原理

### 3.1 同步幅度解调(Synchronous AM Demodulation) 原理

设已接收的标准AM信号(来自上一个实验的输出)为:

$$v_{AM}(t) = A_c [1 + m \cos(\omega_s t)] \cos(\omega_c t) \quad (1)$$

其中,  $f_s$  为基带信号频率,  $f_c$  为载波频率,  $m$  为调制系数。同步解调的核心思想是: 在接收端产生一个与发射端载波同频率且同相位的本地载波(Local Carrier, 也称恢复载波或参考信号)  $v_r(t)$ :

$$v_r(t) = A_r \cos(\omega_c t + \Delta\phi) \quad (2)$$

理想情况下, 相位差  $\Delta\phi = 0$ , 即完全同步。将调制信号  $v_{AM}(t)$  与恢复载波  $v_r(t)$  输入模拟乘法器进行相乘, 利用三角恒等式  $\cos^2(\omega_c t) = \frac{1}{2}[1 + \cos(2\omega_c t)]$ , 得到:

$$v_{out}(t) = k \cdot v_{AM}(t) \cdot v_r(t) = k A_c A_r [1 + m \cos(\omega_s t)] \cos^2(\omega_c t) \quad (3)$$

$$= \underbrace{\frac{k A_c A_r}{2} [1 + m \cos(\omega_s t)]}_{\text{低频分量(包含直流与基带信号)}} + \underbrace{\frac{k A_c A_r}{2} [1 + m \cos(\omega_s t)] \times \cos(2\omega_c t)}_{\text{高频分量(中心频率为 } 2f_c \text{)}} \quad (4)$$

如果相位不同步, 即  $\Delta\phi \neq 0$ , 则乘法器输出变为:

$$v_{out}(t) = k \cdot v_{AM}(t) \cdot v_r(t) = k A_c A_r [1 + m \cos(\omega_s t)] \cos(\omega_c t) \cos(\omega_c t + \Delta\phi) \quad (5)$$

$$= k A_c A_r [1 + m \cos(\omega_s t)] \cos(\omega_c t) [\cos(\Delta\phi) \cos(\omega_c t) - \sin(\Delta\phi) \sin(\omega_c t)] \quad (6)$$

$$= \cos(\Delta\phi) k A_c A_r [1 + m \cos(\omega_s t)] \cos^2(\omega_c t) - \sin(\Delta\phi) k A_c A_r [1 + m \cos(\omega_s t)] \cos(\omega_c t) \sin(\omega_c t) \quad (7)$$

$$= \underbrace{\frac{k A_c A_r \cos(\Delta\phi)}{2} [1 + m \cos(\omega_s t)]}_{\text{低频分量(包含直流与基带信号)}} + \underbrace{\frac{k A_c A_r}{2} [1 + m \cos(\omega_s t)] \times [\cos(\Delta\phi) \cos(2\omega_c t) - \sin(\Delta\phi) \sin(2\omega_c t)]}_{\text{高频分量(中心频率为 } 2f_c \text{)}} \quad (8)$$

由此看出相位误差  $\Delta\phi$  并不会从根本上影响解调结果, 但会使低频分量的幅度降低, 系统信噪比稍有下降。值得一提的是, 上述基于乘法器的同步解调方法适用于绝大多数 Amplitude Modulation 形式, 通用性较强。

### 3.2 低通滤波器的作用

乘法器输出  $v_{out}(t)$  包含两个主要频率成分：

(1) **低频分量 (Low-Frequency Component)**: 包含直流成分和原始基带信号  $\cos(\omega_s t)$ 。

(2) **高频分量 (High-Frequency Component)**: 中心频率在  $2f_c$ , 其幅度受基带信号调制。

通过一个高边截止频率  $f_H$  满足  $f_s \ll f_H \ll 2f_c$  的低通滤波器, 可以很好地滤除高频分量 ( $2f_c$  附近):

$$v_{out, LPF}(t) = \frac{kA_c A_r}{2} [1 + m \cos(\omega_s t)] \quad (9)$$

该信号包含一个直流偏移  $\frac{kA_c A_r}{2}$  和放大后的原始基带信号  $\frac{kA_c A_r m}{2} \cos(\omega_s t)$ 。通过隔直电容或后级电路即可去除直流, 最终恢复出纯净的基带信号。

### 3.3 同步解调与包络检波的比较

- (1) **乘法器同步解调 (Synchronous Demodulation)**: **优点**: 解调线性度好, 适用于低调制系数 ( $m \ll 1$ ) 及过调制 ( $m > 1$ ) 情况; 能解调所有类型的幅度调制信号 (包括 DSB-SC、SSB)。**缺点**: 需要产生与发射端严格同步的本地载波, 系统复杂度高。
- (2) **二极管包络解调 (Envelope Detection)**: **优点**: 电路极其简单 (仅需一个二极管、电阻和电容), 无需本地载波。**缺点**: 仅适用于幅度较大的标准 AM 信号 ( $m \leq 1$ ); 解调效率低, 存在门限效应和失真; 无法解调 DSB-SC 或 SSB 信号。

### 3.4 实验电路简要分析

如 Figure 1 所示, 本次实验的同步解调电路同样以 MC1496 为核心, 结构与调制电路类似但功能不同:

- (1) **输入端口 IN1**: 接入本地恢复载波  $v_r(t)$  (将调整电路中所用的载波信号输入此端);
- (2) **输入端口 IN2**: 接入待解调的 AM 信号  $v_{AM}(t)$  (来自调制实验板的 OUT 端);
- (3) **核心乘法器**: MC1496 实现  $v_{AM}(t) \times v_r(t)$  的相乘操作;
- (4) **低通滤波器**: 由 R16, C10, C11 构成无源低通滤波器, 用于滤除  $2f_c$  高频分量;
- (5) **输出级**: 采用 Common Emitter 放大电路作为输出级, 提高信号幅度的同时隔离负载影响;
- (6) **输出端口 TP3**: 乘法器直接输出端, 可观测未滤波的混合信号 (包含低频和高频分量);
- (7) **输出端口 TP4**: 经过低通滤波器后的输出, 即为恢复后的基带信号。

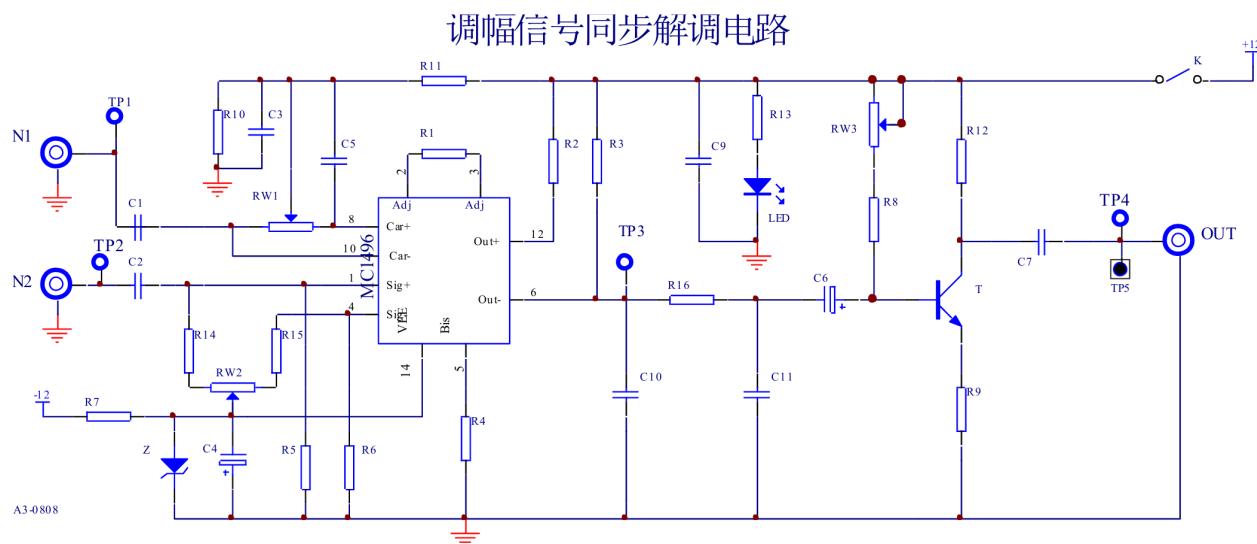


Figure 1: MC1496-Based Synchronous AM Demodulation Circuit Used in the Experiment

## 4 实验内容与步骤

### 4.1 实验前准备与连接

- (1) 在实验箱上接入乘法器幅度调制实验板和调幅信号同步解调实验板；
- (2) 按上一个实验的步骤，在调制板上产生一个调制系数  $m \approx 0.5$  的 AM 信号；
- (3) 将调制板 OUT 端产生的 AM 信号接入解调板的 IN2 端；
- (4) 将调制板所用的 10.7 MHz 载波信号（信号源或晶体振荡器输出）接入解调板的 IN1 端作为本地载波。

### 4.2 解调过程观测与调试

- (1) 观测未滤波信号 (TP3): 将示波器通道 1 接 TP3，调整解调板上的电位器 RW1 和 RW2，使波形稳定。此时应观察到包含低频包络和高频载波分量的混合信号。
- (2) 观测滤波后信号 (TP4): 将示波器通道 2 接 TP4，观察经过低通滤波器后的输出，应为一个频率为 1 kHz 的正弦波（可能带有少量残余高频纹波）；微调 RW1, RW2 使 TP4 端的正弦幅度最大、失真最小。
- (3) 不同调制系数下的解调: 在幅度调制板中，依次将调制系数  $m$  调节至约 0.3、0.5、0.8，对每一个  $m$  值，在解调板上记录 TP3 和 TP4 的波形；对比不同  $m$  下，解调输出波形 (TP4) 的幅度、失真度和信噪比的变化。

## 5 思考题

### 5.1 分析调幅波同步解调方法的优缺点。

调幅波同步解调方法具有一系列显著优点。首先，在理想同步条件下，其解调输出与输入调制信号之间呈现严格的线性关系，失真度极低，保证了高线性度与高保真度。其次，该方法的适用性非常广泛，不仅能解调标准 AM 信号，还能有效处理抑制载波的双边带 (DSB-SC) 和单边带调制 (SSB) 等复杂调制信号。此外，与包络检波相比，同步解调不存在“门限效应”，在低信噪比环境下依然能保持优良的接收性能。最后，该方法对过调制状态具有很好的鲁棒性，即使调制系数  $m > 1$ ，只要载波同步良好，依然能够正确解调出原始信号。

然而，同步解调方法也存在若干缺点，其中最主要的一点是需要实现精确的载波同步，即接收端必须产生一个与发射端载波严格同频同相的本地载波。这通常需要依赖锁相环 (PLL) 等复杂的载波生成电路，显著增加了系统的硬件成本与整体复杂度。其次，该方法的性能对相位噪声高度敏感，本地载波的相位噪声  $\phi_n$  会通过幅度系数  $\cos(\Delta\phi)$  直接耦合到解调后的基带信号中。最后，从电路实现层面看，同步解调需要模拟乘法器和低通滤波器等组件，其电路结构相比简单的包络检波解调电路要复杂得多。

总的来讲，同步解调主要应用于对解调质量要求严苛、信号形式复杂（如 DSB-SC、SSB）或工作于低信噪比环境的通信系统中，例如专业无线电、卫星通信以及数字通信的相干解调等场合。相比之下，包络检波因其极简的电路结构，被广泛应用于对成本敏感、且只需处理标准 AM 信号的广播接收机等场景。

## 附录 C MATLAB Codes

```

1  %% NCE-08 示波器数据读取与保存
2  clc, clear
3  address = 'USB0::0x1AB1::0x04B0::DS2F203700277::INSTR';
4  depth_level = 3; % 1 ~ 4
5  % rate_level = 2;
6
7  ch1 = 1;
8  ch2 = 1;
9
10 % MyOscilloscope_MS02202A_Read_TwoCh(address, 0);
11
12 if ch1 == 1
13     % CH1
14     flag_plot = 0;
15     ch = 1;
16     stc1 = MyOscilloscope_MS02202A_Read(address, ch, depth_level, flag_plot);
17     stc1.stc_spectrum.plot4.axes.XLim = [0.1e6, 0.5e9];
18     amplitude = stc1.stc_spectrum.STC.y_oneSided;
19     freq = stc1.stc_spectrum.STC.fAxis_oneSided;
20     power = stc1.stc_spectrum.STC.power_oneSided;
21     fr = stc1.stc_spectrum.STC.fr;
22     fs = stc1.stc_spectrum.STC.fs;
23
24     v_n1 = stc1.data';
25     t_n1 = stc1.time;
26     stc1 = MyAnalysis_Spectrum_hanningWindow(v_n1, fs, 1);
27 end
28
29 if ch2 == 1
30     % CH1
31     flag_plot = 0;
32     ch = 2;
33     stc2 = MyOscilloscope_MS02202A_Read(address, ch, depth_level, flag_plot);
34     stc2.stc_spectrum.plot4.axes.XLim = [0.1e6, 0.5e9];
35     amplitude = stc2.stc_spectrum.STC.y_oneSided;
36     freq = stc2.stc_spectrum.STC.fAxis_oneSided;
37     power = stc2.stc_spectrum.STC.power_oneSided;
38     fr = stc2.stc_spectrum.STC.fr;
39     fs = stc2.stc_spectrum.STC.fs;
40
41     v_n2 = stc2.data';
42     t_n2 = stc2.time;
43     stc2 = MyAnalysis_Spectrum_hanningWindow(v_n2, fs, 1);
44 end
45
46
47
48 %% NCE-07 幅度调制系数计算
49 warning('off')
50 %fileName = "D:\aa_MyExperimentData\Raw data backup\2025-12-11_14-26-01__NCE-07__Oscilloscope__ NCE-07-
51 %AM_base_100kHz_300mVpp_carrier_10.7MHz_500mVpp.txt";
52 %rawdata = readmatrix(fileName);
53
54 v_n = v_n1;
55 fs = 1/(t_n2(2) - t_n2(1))
56
57 f0 = 10.7e6
58 f_am = 20e3

```

```
59 delta_f = f_am/2*1.5;
60 A_LSB = [
61     MyAnalysis_Get_PowerAndFrequency(v_n, fs, f0 - 3*f_am, delta_f).target.A0;
62     MyAnalysis_Get_PowerAndFrequency(v_n, fs, f0 - 2*f_am, delta_f).target.A0;
63     MyAnalysis_Get_PowerAndFrequency(v_n, fs, f0 - 1*f_am, delta_f).target.A0;
64 ];
65 A_USB = [
66     MyAnalysis_Get_PowerAndFrequency(v_n, fs, f0 - 1*f_am, delta_f).target.A0;
67     MyAnalysis_Get_PowerAndFrequency(v_n, fs, f0 - 2*f_am, delta_f).target.A0;
68     MyAnalysis_Get_PowerAndFrequency(v_n, fs, f0 - 3*f_am, delta_f).target.A0;
69 ];
70 A_c = MyAnalysis_Get_PowerAndFrequency(v_n, fs, f0, delta_f).target.A0;
71 m = sqrt(sum(A_LSB(end).^2) + sum(A_USB(1).^2))/A_c
72 m = sqrt(sum(A_LSB.^2) + sum(A_USB.^2))/A_c
73 warning('on')
74
75
76 MyYYPlot(t_n1, v_n1, t_n2, v_n2);
77 xlim([0, 10e-06])
78 MyFigure_ChangeSize_2048x512
79
80
81
82 xlim([0, 1/(20e3)*5])
83
84 stc1.ax2.XLim = [f0 - 3*f_am, f0 + 3*f_am];
85 stc1.ax4.XLim = [f0 - 3*f_am, f0 + 3*f_am];
86 stc2.ax2.XLim = [f0 - 3*f_am, f0 + 3*f_am];
87 stc2.ax4.XLim = [f0 - 3*f_am, f0 + 3*f_am];
```