

参赛密码 _____

(由组委会填写)



“华为杯”第十四届中国研究生 数学建模竞赛

题 目 面向下一代光通信的 VCSEL 激光器仿真模型

摘 要：

激光器是现代光通信中的关键器件之一，激光器的性能极大地影响了光通信系统的整体性能。VCSEL 激光器有潜力成为符合下一代光通信要求的关键器件。本文围绕着 VCSEL 激光器的多种性能，通过建立数学模型，修正数学模型，采取多种算法对典型实验数据进行拟合，得到了更广泛条件下 VCSEL 激光器的工作性能和关键参数。本文的工作包括但不限于建立了 VCSEL 激光器工作电流与输出光功率强度关系模型（L-I 模型），分析了输出光功率对于温度的依赖性；在分析 L-I 模型精度和误差产生原因的基础，通过考虑更高阶的温度依赖因子，改进 L-I 模型，并进行分析；在速率方程稳态解基础上引入小信号，建立小信号响应模型，分析环境温度与偏置电流对于小信号响应的影响，并提出拓宽 10dB 带宽的方案；对于小信号响应模型中变量选择性近似处理，在保证精度的前提下优化模型并提高了模型的运算速度。

在问题一中，本文在忽略激光器装换效率受温度影响的情况下，建立了 L-I 模型，在此基础上，比较讨论了局部优化，matlab 自带的全局优化以及自己编写的 PSO 算法，以剩余误差平方和最小为主，以是否具备物理意义为辅，运用两种算法计算得到了模型的具体参数并做了相应评估，认为 PSO 算法优于局部优化算法。在求解模型获得模型参数后，讨论了 VCSEL 激光器对于温度的依赖性。

在问题二中，本文在分析 L-I 模型的误差产生原因后，通过两种方式优化 L-I

模型：一则考虑转换效率 $\eta(T)$ 对于温度的依赖性，二则增加泄漏电流高阶的优化项。同样，从剩余误差平方和与物理意义两方面对模型的改进进行了评估，发现在同时优化转换效率项与泄漏电流项时对模型的改进最明显。同时我们定性的研究了 FDTD 模型，用微分法定性的分析 P-I 模型，通过曲线的一致性，我们可以通过进一步拟合函数中的参数可以得到较好的比对值。FDTD 模型可以进一步观察内部微观状态，从而更好调优激光器的性能。

在问题三中，本文在激光器速率方程稳态解的基础上，引入小信号，建立了激光的带宽响应模型。在求解模型时，提出三个误差约束条件：理论驱动电流 I_{cur} 与实测驱动电流 I 误差最优；理论光功率 P_s 与实测激光器光功率 P_0 误差最优；理论小信号响应 $G(f)$ 与 S21 中的实测数据误差最优。由于参数数量级差别大的关系，由于 Y, Z 作为独立参数单独独立拟合，可以避免算法循环调用中复杂度的增加，进一步大大加快拟合速度和效率，从而调优参数，那么本文提出分两步拟合参数：第一步利用小波变换调优数据，进一步使用 PSO 算法拟合得到 Y 与 Z 的值，第二步，由于约束条件的作用我们引入了罚函数去规范约束 I, P, 同时融合协调函数，去协调两个 Y, Z 之间的相互影响作用（Y, Z 差了 11 个数量级），转换参数使得在统一数量级，利用光滑函数 BP 求解不会出现断点的特性，同时去改进了目标函数为光滑函数，再进一步利用了拉格朗日乘子法 PH 以及 PHR 模型，同时设定初始值，并且同时采用调用 PSO，拟合得到模型 8 个参数，分析对比取得最好的数值，进一步拟合参数。在此基础上，讨论了环境温度和偏置电流对于小信号响应曲线的影响，并发现增大模型中的 Z 参数可以有效增大激光器带宽，增大模型中的 Y 参数可以在特定条件下使 3dB 以内带宽响应曲线更为平坦。

在问题四中，本文针对问题三中的小信号响应模型做了进一步优化。通过完善模型理论基础，在理论推导过程中对模型参数进行了合理近似，获得了更为精确和简洁的新模型。简化模型中，响应曲线参数有了更为简洁的表达式，

$$Y = \frac{(G_0/k)P_s}{1+(\varepsilon/k)P_s}, \quad Z = \frac{Y}{\tau_p}, \quad \text{从而在模型整体误差小于 1\%的基础上大大减少了模}$$

型参数和求解复杂度。在此基础上，将改进模型推导出的激光器性能与原模型相应结果进行了对比，充分验证了改进模型的可靠性。

本文的创新点在于：第一，充分利用了系统稳态工作条件，即各参数对时间偏导为 0，大大简化求解思路；第二，灵活使用了多种拟合算法，对不同问题选用了更为精确的算法，例如在问题三中拟合参数数量级相差太大的情况下，为了保证拟合精确度并降低求解模型难度，分两部分求解参量，并改进了拉格朗日乘子算法；第三，所有数学过程保持了对参数物理意义的关注，使整体结果不偏离模型的物理背景；第四，物理推导严格，数学模型可靠，例如在问题二中，将问题一中被忽略的转换效率随温度变化也纳入了考虑，在问题四中，推导了比常用公式更为精确和简洁的模型。

关键词：VCSEL 激光器，小信号响应模型，PSO 算法，拉格朗日乘子法。

目 录

一、	问题重述.....	5
1.1	提出问题.....	5
1.2	问题要求.....	5
二、	基本假设和符号说明.....	6
2.1	基本假设.....	6
2.2	符号说明.....	6
三、	问题 1: VCSEL 的 L-I 模型.....	7
3.1	问题分析.....	7
3.2	VCSEL 的 L-I 模型建立.....	7
3.3	算法介绍.....	8
3.4	模型求解与分析.....	9
3.5	环境温度对用户使用的影晌.....	13
四、	问题 2: VCSEL 的 L-I 模型改进.....	15
4.1	问题 1 中误差分析及解决方案.....	15
4.2	考虑 $\eta(T)$ 对于温度是缓变函数.....	15
4.3	增加泄漏电流项优化因子.....	17
4.4	考虑 $\eta(T)$ 对于温度是缓变函数并增加优化因子.....	19
4.5	改进模型的分析.....	20
4.6	FDTD 建立模型.....	20
五、	问题 3: VCSEL 的带宽模型.....	22
5.1	问题分析.....	22
5.2	模型建立.....	22
5.3	算法介绍.....	25
5.4	曲线平滑.....	30
5.5	求解模型.....	32
5.6	分析带宽响应曲线.....	35
5.7	拓宽带宽的方案.....	35
5.8	平坦 3dB 以内的的带宽曲线.....	36
六、	问题 4: VCSEL 的带宽模型的改进.....	39
6.1	问题分析.....	39
6.2	模型建立.....	40
6.3	模型求解.....	45
6.4	模型分析.....	46
七、	参考文献.....	48

一、 问题重述

1.1 提出问题

垂直腔面发射激光器（Vertical-Cavity Surface-Emitting Lasers）是在 1977 年提出的新型半导体激光器，具有极低的阈值、较小的远场发散角、调制频率高等优越性。VCSEL 在光互连、光开关和光显示等领域都有广泛的应用前景。[1]在不久的将来将会在中等距离光互连传输中占据主要地位。激光器是光纤通信中的核心器件，VCSEL 的发展将会促进了光纤通信的发展。

在 VCSEL 中，有源区中结温度的增加会影响双异质结结构半导体激光器的输出特性，温度对 VCSEL 的输出特性起着决定性作用。所以建立数学模型描述激光器输出功率的功率强度与温度的关系，对 VCSEL 在光通信中的应用有着重要的指导作用。

随着互联网技术的发展，人们对网络传播速度的要求势必越来越高。光纤传播技术早已开始融入到每家每户的生活中，，便利了我们的生活的方方面面。因为光纤传输的激光信号带宽远大于电信号的传输带宽，这才使得光纤通信开始慢慢替代电通信。带宽标识了信息的传输能力，如何进一步提高光纤中激光传输带宽也将成为一大研究热点。VCSEL 因其较短的谐振腔，有着弛豫振荡频率和调制带宽高的特性。为了将 VCSEL 更好地应用于光通信中，拓宽 VCSEL 的传输带宽是必不可少的。

1.2 问题要求

本研究主要解决两个问题：根据实测数据建立并改进激光器工作电流与输出光功率强度关系模型；建立并改进 VCSEL 的带宽数学模型。

问题 1：附件“L-I-20C.mat”提供了室温在 20℃下采集的光功率 P、实测驱动电流 I、实测电压 U 和实测温度 Ta，基于该实测数据，参考附录中给出的经验公式，建立数学模型拟合出模型参数 (η , I_{th0} , R_{th} , a_0 , a_1 , a_2 , a_3 , a_4)，从而得出不同温度下 VCSEL 的 L-I 特征曲线，并根据 L-I 模型推测 VCSEL 激光器工作的最高环境温度。

问题 2：分析问题 1 中由经验公式建立的拟合模型的误差产生原因，解决问题，并根据改进的数学模型画出 10℃, 20℃, 30℃,, 90℃等温度下的 VCSEL 的 L-I 特征曲线。

问题 3：在速率方程稳态解的基础上，引入小信号，建立小信号响应模型，并考虑环境温度和偏置电流对于小信号响应模型的影响，其后讨论拓宽带宽的方法。

问题 4：在原有建模的基础上，通过考虑不同的忽略项来简化模型，探讨运算速度更快的带宽模型，或者改变建模方式，在相同温度和偏置电流下，获得更宽的 10dB 带宽。

二、基本假设和符号说明

2.1 基本假设

- (1)假设系统中没能转化为光输出的能量全部转化为热量;
- (2)假设系统都是工作在稳态情况下, 此时物理量对时间偏导为 0。

2.2 符号说明

符号	符号说明
P_0	激光器输出光功率, mW
I	注入到激光器的外部驱动电流, mA
$\eta(T)$	转换效率
$I_{th}(N, T)$	阈值电流, mA
N	载流子数
R_{th}	VCSEL 热阻抗
T_0	环境温度, °C
V	输入电压, mV
η	$\eta(T)$ 近似等于的常数
η_i	注入效率
N_0	透明载流子数
$I_{off}(T)$	与温度相关的偏置电流, mA
τ_n	载流子复合寿命, ns
τ_p	光子寿命, ps
G_0	增益系数, s ⁻¹
S	光子数
β	受激辐射耦合系数
ε	增益压缩因子
k	光功率与光子数的比例因子, W

三、 问题 1：VCSEL 的 L-I 模型

3.1 问题分析

VCSEL 激光器的阈值电流、泄漏电流、输出功率的热饱和效应等对于温度依赖性很大。本题为一个简单的 VCSEL 的 L-I 经验模型。在有 20°C 下的 L-I 实测数据的情况下，根据经验模型，确定模型的参数，获取其他温度的 L-I 特征曲线。首先需要通过一定的模型假设，简化得到 L-I 模型，然后通过最小优化算法，确定模型参数，使得理论 L-I 曲线与实测曲线的拟合度最高，最后进行模型分析。在已有模型的基础上，探讨 VCSEL 的最高工作环境温度，并对其做出一定分析。

3.2 VCSEL 的 L-I 模型建立

根据附录一提供的经验公式：

$$P_0 = \eta(T)(I - I_{th}(N, T))$$

其中， P_0 为激光器输出的光功率， I 为注入电流， $I_{th}(N, T)$ 为阈值电流，与载流子数和温度相关。

在此基础上，我们有如下的假设：

1. 工作在稳态状况下，即不考虑 dT/dt ；
2. $\eta(T)$ 受温度影响较小，假设 $\eta(T)$ 近似于常数 η ；

在稳态下，载流子数保持不变，则可认为阈值电流只与温度相关，可写为

$$I_{th}(N, T) = I_{th0} + I_{off}(T)$$

其中， I_{th0} 为常数， $I_{off}(T)$ 可用检验公式表示为 $I_{off}(T) = \sum_{n=0}^{\infty} a_n T^n$ 。

温度 T 在受到 V-I 特性的影响下，可以表示为[2]

$$T = T_0 + (IV - P_0)R_{th}$$

从而，得到简化的经验模型如下

$$P_0 = \eta \left\{ I - I_{th0} - \sum_{n=0}^{\infty} a_n [T_0 + (IV - P_0)R_{th}]^n \right\}$$

其中， T_0 为环境温度， I 为偏置电流 I_b （输入电流）， V 为输入电压， a_0 ，

a_1 ， a_2 ， a_3 ， a_4 为泄漏电流优化因子。

需要注意的一点是，在上述模型中需要确认的模型参数有八个， η ， I_{th0} ， R_{th} ， a_0 ， a_1 ， a_2 ， a_3 ， a_4 。而实际上只有 6 个自由度。虽然此处将 I_{th0} 与 a_0 分别作为参数进行模型求解， I_{th0} 与 a_0 的效果是等同的，故而在本文中，若出现拟合参数 I_{th0} 为负数也并非求解模型出现问题，在此处的经验模型中，单独的 I_{th0} 没有物理意义，保证 I_{th0} 与 $I_{off}(T)$ 和为正数即可。而理论上 R_{th} 可以保持 2600 不变，将泄漏电流只取到二阶为例证之。泄漏电流取到四阶的情形可以类比得到。

泄漏电流只取到二阶可以表示为

$$I_{off} = a_0 + a_1(T_0 + (IV - P_0)R_{th}) + a_2(T_0 + (IV - P_0)R_{th})^2$$

必然可以找到另一组参数 $(a'_0, a'_1, a'_2, R'_{th})$ 来描述 $I_{off}(T)$ ，假设 $R'_{th} = kR_{th}$ ，则

有

$$\begin{aligned}
I_{\text{off}} &= a'_0 + a'_1(T_0 + (IV - P_0)R'_{th}) + a'_2(T_0 + (IV - P_0)R'_{th})^2 \\
&= a'_0 + a'_1k \left(T_0 + (IV - P_0)R_{th} \right. \\
&\quad \left. - \left(1 - \frac{1}{k} \right) T_0 \right) + a'_2k^2 \left(T_0 + (IV - P_0)R_{th} - \left(1 - \frac{1}{k^2} \right) T_0 \right)^2 \\
&= a'_0 + a'_1k(T_0 + (IV - P_0)R_{th}) - a'_1k \left(1 - \frac{1}{k} \right) T_0 + a'_2k^2(T_0 + (IV - P_0)R_{th})^2 \\
&\quad - 2a'_2k^2 \left(1 - \frac{1}{k^2} \right) T_0(T_0 + (IV - P_0)R_{th}) + a'_2k^2 \left(1 - \frac{1}{k^2} \right)^2 T_0^2
\end{aligned}$$

则有

$$\begin{cases} a_2 = a'_2k^2 \\ a_1 = a'_1k - 2a'_2k^2 \left(1 - \frac{1}{k^2} \right) T_0 \\ a_0 = a'_0 - a'_1k \left(1 - \frac{1}{k} \right) T_0 + a'_2k^2 \left(1 - \frac{1}{k^2} \right)^2 T_0^2 \end{cases}$$

也就是在 $R'_{th} = kR_{th}$ 的情况下, a'_0, a'_1, a'_2 都可以用 k 和 T_0 来表示。也就是无论 R_{th} 取多少, 最终都可以将其变换到 2600, 此时也能有一组参数来描述模型。

3.3 算法介绍

在拟合 L-I 曲线, 确立参数中, 本文考虑了局部优化算法和 PSO 算法。

(1) 局部优化算法

所使用的局部优化算法就是 Matlab 工具箱中的 `fmincon` 函数, 可以求最小约束的非线性多变量函数, 可以从最初的估计中找到一个有限的标量函数所限制的最小值。函数 `fmincon` 通常被称为约束非线性优化或者非线性规划。

寻求最小值问题的方程可以写作

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases}$$

`fmincon` 函数也有一定的局限性, `fmincon` 是一种基于梯度的方法, 被设计用于解决目标和约束函数都是连续的一阶导数问题。在遇到实际问题不可行时, `fmincon` 会尝试将最大的约束值最小化。`fmincon` 函数中“信任域反射”算法不能取上下边界为相同值, 若出现“`lb==ub`”的状况, 将会出错。

(2) PSO 算法[3]

由于本文研究的八种随机参数问题, 在需要更好稳定性以及更好精度, 需要采取别的方式研究。而以遗传算法 GA 为代表的智能算法特点就是全局随机搜索, 不依赖于函数本身性质和初始状态的影响。对于超越函数求解此类方式再合适不过。粒子群 (particle swarm optimization, 简称 PSO) 算法起源于简单的社会模拟, 来源于模拟鸟类觅食的过程, 由 1995 那年 Eberhart 和 Kennedy 对鸟群觅食行为的研究演化而来, 是模拟生物社会系统的算法。与其他的传统算法中, 此算

法可以无集中的控制约束，通过非间接通信进行合作，群体中相互作用的个体是分布，执行时间较短等优点。对于这种较高的并行高效的优化算法可以解决各种超越函数。PSO 具有很多优点，形式简洁、收敛快速，并且参数调节灵活，既有全局搜索功能，又能寻求局域最优点。但 PSO 算法也存在一些算法固有缺陷，主要表现在函数出现了局域最优解，导致了此算法并不能全部应用于所有的超越函数的求解。为了探索本文要解决的超越函数是否适用，需要探究搜寻点和搜寻目标之间的关系。探寻搜寻目标过程中要采用不同的飞行速度，通过寻找最佳飞行速度，也就是在最短的时间内找得到最优的解，结合遗传算法理论，找寻最佳速度。[4]

PSO 算法简单的思想如下：

假设在一个 D 维的目标搜索空间中，有 m 个粒子组成的一个群体，其中第 i 个粒子表示为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ，即其位置为 X_i ，换言之，每个粒子的位置就是一个潜在的解，将 X_i 代入目标函数，就可以计算出其适应值，根据适应值的大小衡量优劣，经历的最好位置记作 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ ，整个群体所有粒子经历的最好位置记作

$$P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$$

粒子 i 的速度表示为

$$V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$$

粒子群算法采用下列公式对粒子进行操作：

$$\begin{aligned} v_{ij} &= wv_{ij} + c_1r_1(p_{ij} - x_{ij}) + c_2r_2(p_{gj} - x_{ij}) \\ x_{ij} &= x_{ij} + av_{ij} \end{aligned}$$

其中， $i = 1, 2, \dots, m; j = 1, 2, \dots, D; w$ 为非负数，称作惯性因子； c_1, c_2 为非负数。称作学习因子； r_1, r_2 为 $[0, 1]$ 内变化的随机数； a 称作约束因子，控制粒子速度。

粒子的速度是需要被一个最大速度限制的，若当时刻粒子的某维速度 v_{ij} 更新后超过该维的最大速度，则当前时刻的该维的速度被限制。在优化算法中，一般选取预定最大迭代次数或粒子群目前为止搜索到的最优位置满足预定最小适应阈值为算法终止条件。

3.4 模型求解与分析

将附录中提供的模型参考初值代入到已建立的模型中，计算得到 L-I 特性曲线，如图 1 所示，拟合所得的曲线与实测数据曲线差别很大，但是基本趋势是相符合的，模型的建立有可以拟合的空间。下面将由局部优化算法和 PSO 算法分别进行模型求解。

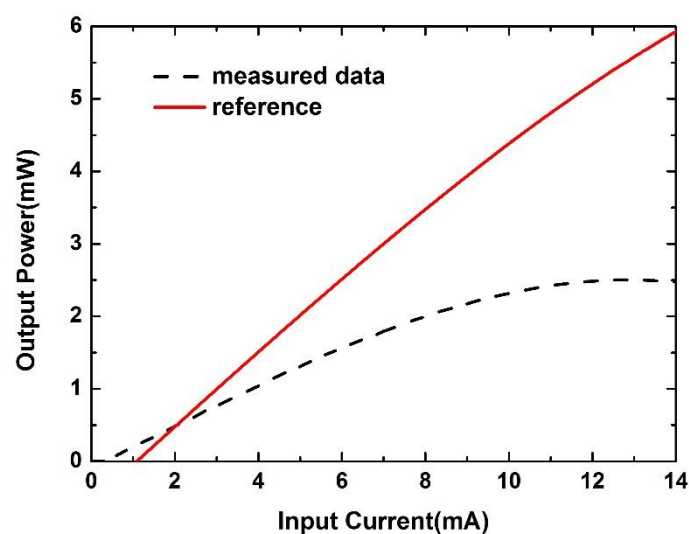
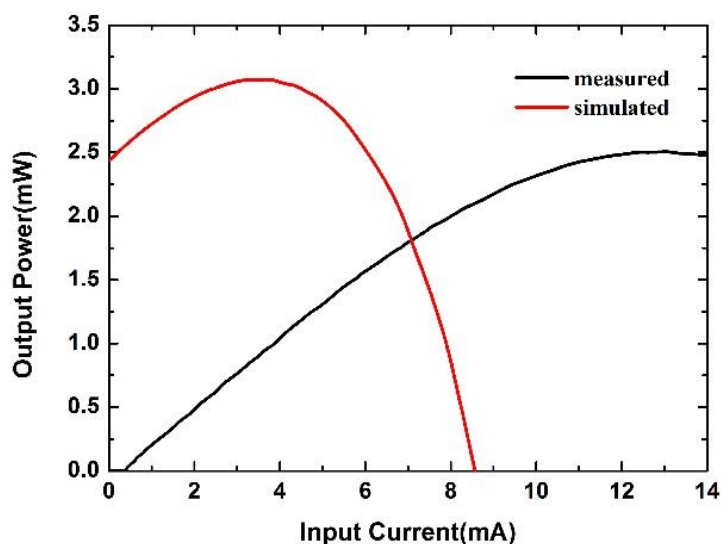


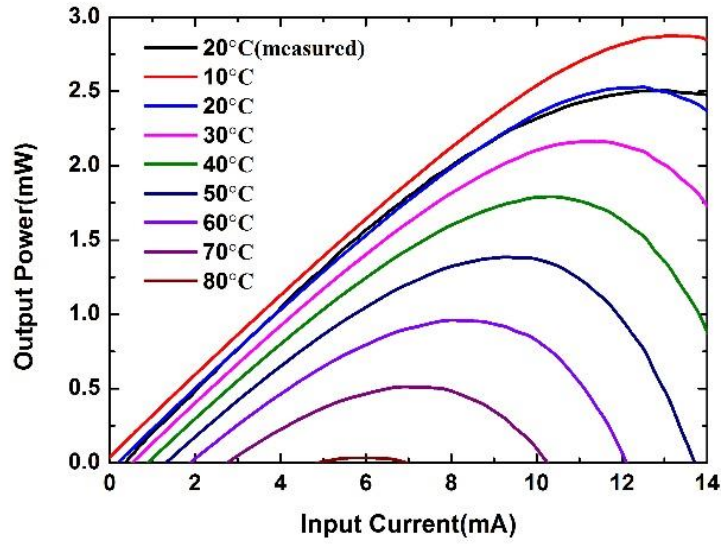
图 1 参考值为参数 L-I 特征曲线与实测曲线对比图

(1) 局部优化算法

在已有实测数据下,通过 matlab 优化工具箱中局部优化算法可以拟合理论数据,确定八个模型参数。在局部优化的过程中,有无设置优化约束条件,将会对拟合的准确性有极大的影响。图 2(a)为随机输入初始值的拟合与实测数据对比图,图 2(b)输入附录中参考值为初始值拟合参数所得不同温度下 L-I 曲线图。表 1 为 L-I 模型参考初值和局部优化参数值。



(a)随机设置初始值



(b)参考值为初始值

图 2 局部优化有无初始值拟合 L-I 特征曲线对比

表1 L-I 模型参考初值和局部优化参数值

参数	参考初值	随机初始值	参考初始值	单位
η	0.5	0.382	0.318	-
I_{th0}	0.3E-3	-0.708E-3	2.248E-3	A
R_{th}	2.6E3	2909.83	2600	°C/W
a_0	1.246E-3	-5.883E-3	-2.643E-3	A
a_1	-2.545E-5	-1.20E-05	2.84E-05	A/K
a_2	2.908E-7	2.22E-06	0	A/K ²
a_3	-2.531E-10	1.62E-09	-5.06E-10	A/K ³
a_4	1.022E-12	-4.83E-12	2.04E-11	A/K ⁴

在优化过程中,以剩余误差的平方和(RSS)来评价拟合程度,记为 SS_{res} 。随机初始值情况下, $SS_{res} = 1.89 \times 10^{-2}$; 以参考值为初值的情况下, $SS_{res} = 1.06 \times 10^{-6}$ 。记标准误差为 Re, 公式为

$$Re = \sqrt{\frac{1}{n} SS_{res}}$$

下表为局部优化随机初始值和参考值为初值情况误差表。

表2 随机初始值和参考值为初值情况误差

	RSS	Re
随机初始值	$1.89 * 10^{-2}$	$3.67 * 10^{-3}$
参考值为初值	$1.06 * 10^{-6}$	$2.7571 * 10^{-5}$

实测数据的输出功率是 mW 级，故而随机初始值误差极大，即使在设置参考值为初值的情况下，拟合曲线与实测数据契合度依旧不够高，残差的稳定性不足。因此局部优化对于初始值的设置过于敏感。为了适应此处的数学模型，将全局优化和局部优化相结合或是较佳的解决方法。

通过 PSO 算法设定 $c1=1.2$ ， $c2=0.052$ ， $D=8$ ， $N=1500$ ，迭代次数为 1000，优化所得的模型参数 (η , I_{th0} , R_{th} , a_0 , a_1 , a_2 , a_3 , a_4) 如表三，该算法下， $SS_{res} = 2.07 * 10^{-7}$ ， $Re = 11.215 * 10^{-5}$ 。

表3 PSO 算法拟合模型参数

η	I_{th0}	R_{th}	a_0	a_1	a_2	a_3	a_4
0.28983	-0.00214	1798.1012	0.003073	-4.53E-05	6.95E-07	4.59E-09	-3.77E-12

PSO 算法拟合所得 20℃时 L-I 曲线，较之局部优化算法，误差值和残差稳定性都得到了很大提升。图 3 显示了从 38mA 到 1400mA 的 PSO 算法和设置参考值为初值的局部优化算法的残差绝对值比较图。从实测数据中可以看出，0-38mA 的输入电流下，没有输出功率，故而不考虑 0-38mA 的残差绝对值分布，没有物理意义。从图 3 中可以看出，PSO 的残差稳定性远优于局部优化算法下的残差稳定性。

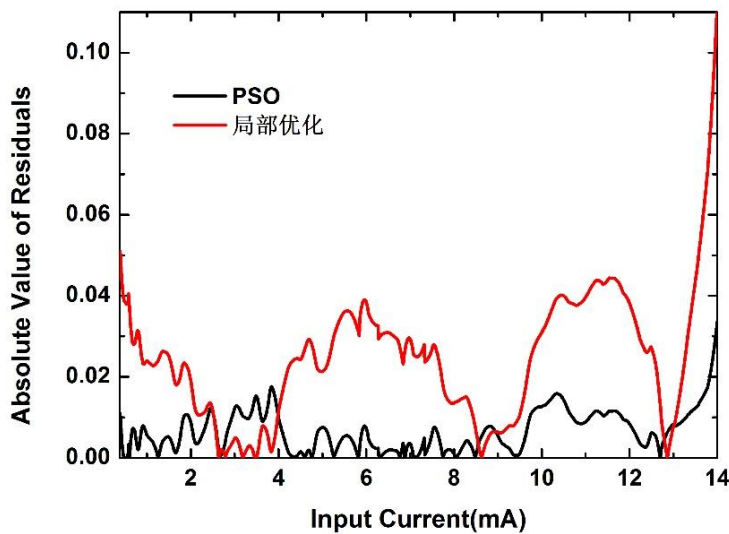


图 3 PSO 与局部优化算法残差绝对值对比图

根据 PSO 算法拟合所得的模型参数 (η , I_{th0} , R_{th} , a_0 , a_1 , a_2 , a_3 , a_4), 画出 10°C, 20°C, 30°C,, 90°C 等温度下的 L-I 特征曲线图如图 4。其中, 输出功率小于 0 的曲线不予显示, 输出功率小于 0 时, 不具有物理意义。故而温度为 70°C, 80°C, 90°C 的曲线已经没有物理意义, 即温度高于 70°C 时, 无论输入电流多少, 都不会有输出功率。由图可以看出, 输出光功率在高温和大输入电流的情况下会出现热饱和而下降的现象。

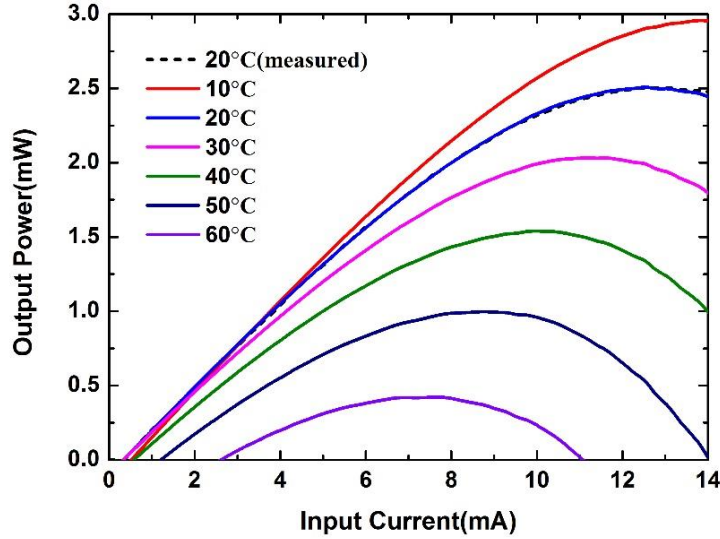


图 4 不同温度下 L-I 特征曲线

3.5 环境温度对用户使用的影

当 VCSEL 激光器输出光功率过低时, 检测器无法检测到信号, 过低的输出光功率没有实际应用意义。设定激光器在直流输入时输出的平均光功率低于 2mW 时, 检测器就无法检测到信号了。根据 PSO 算法建立的 L-I 模型, 以 0.02°C 的步长进行计算, 得到 VCSEL 激光器工作的最高环境温度为 30.68°C。图 5 为输出功率才大于 2mW 时 10°C, 20°C, 30°C 温度下的 L-I 特征曲线, 从图中可以看出, 在 30°C 温度时, 只有约 10mA 到 12.5mA 的输入电流时, 输出功率才大于 2mW, 并且该温度下, 输出功率几乎已经接近 2mW, 实际应用效率已然不高。故而实际的 VCSEL 激光器工作环境温度会比所求最高温度 30.68°C 更低。

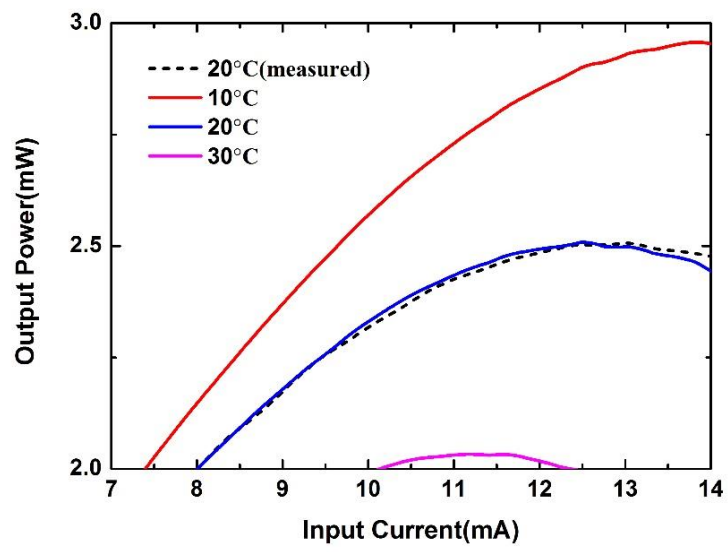


图 5 10°C, 20°C, 30°C温度下的 L-I 特征曲线

四、 问题 2：VCSEL 的 L-I 模型改进

4.1 问题 1 中误差分析及解决方案

对比上述的拟合可以发现，虽然拟合过程本身都尽可能精确，但拟合结果之间仍然有一些差别，尤其是当拟合模型被用于推算其他温度条件下的 L-I 曲线时，这种差别不可避免地被进一步放大。这种误差的主要来源在于，一方面，拟合所根据的 L-I 曲线仅仅是单个温度下的结果，本身所包含的温度相关信息是极其有限的，非常依赖模型本身中温度相关的假设，另一方面，拟合中自由度很高，而且大部分参数都与温度相关，参数的小变动都可能造成不同温度下结果大相径庭，而且拟合结果未必符合物理情境，很难从物理角度对拟合过程加以限制。

从图 2 可以看出，输入电流为 10mA 到 12.5mA 之间时，理论输出功率残差绝对值较大，模型仍然存在改进空间。考虑更多的温度依赖项能够减小问题 1 中出现的误差，有两处温度依赖项可以进一步改进。第一点，问题 1 中假设转换效率受温度影响较小，即 $\eta(T)$ 对于温度是缓变函数，即 $\eta(T)$ 近似于常数 η ，而事实上，输入电流和输出功率都相对较小，转换效率于温度的变化也会对输出功率产生一定的影响。第二点，在问题 1 中，选取的优化因子达到了四阶，优化因子描述的是由温度控制的泄漏电流项。VCSEL 对的增益对于温度依赖性很大，VCSEL 的输出功率在大电流下存在热饱和和下降。VCSEL 内部的温度会因注入电流的增大而提高，然后有源层中更多载流子能量增大脱离量子阱的束缚，从而载流子复合辐射产生的光子数减少，输出光功率骤降。[5]故而，优化因子对于输出光功率的影响度较高，可以通过增加优化因子的阶数，来进一步提高拟合优度。

4.2 考虑 $\eta(T)$ 对于温度是缓变函数

考虑到 $\eta(T)$ 对于温度是缓变函数，假定

$$\eta(T) = \sum_{n=0}^{\infty} b_n T^n$$

其中， $T = T_0 + (IV - P_0)R_{th}$ 。考虑到三阶的影响因子,也就是只可考虑 b_0 , b_1 , b_2 项的影响。同时，对于 a_n 项的影响因子，如问题 1 一样，只考虑 a_0 , a_1 , a_2 , a_3 , a_4 五个优化因子。故而 L-I 模型可写为

$$P_0 = \sum_{n=0}^{n=2} b_n (T_0 + (IV - P_0)R_{th})^n \left\{ I - I_{th0} - \sum_{n=0}^{n=4} a_n [T_0 + (IV - P_0)R_{th}]^n \right\}$$

使用 PSO 算法，确定模型参数 (b_0 , b_1 , b_2 , I_{th0} , R_{th} , a_0 , a_1 , a_2 , a_3 , a_4)。

拟合的模型参数结果如下表。由于 $\eta(T)$ 对于温度是缓变函数，故而在优化过程中，必须约束 b_0 , b_1 , b_2 项保证其为缓变函数。

表4 对于温度是缓变函数时拟合模型参数

b_0	b_1	b_2	I_{th0}	R_{th}
0.264801	0.003746	-5.8E-05	0.002045	1007.382
a_0	a_1	a_2	a_3	a_4
-0.00234	-6.42E-07	1.31E-06	3.69E-09	2.04E-11

在上述模型参数下，20℃的拟合曲线与实测曲线的剩余误差的平方和 $SS_{res} = 1.89 * 10^{-7}$ ， $Re = 1.16 * 10^{-5}$ 。画出 10℃，20℃，30℃，.....，90℃等温度下的 L-I 特征曲线图如图 6。

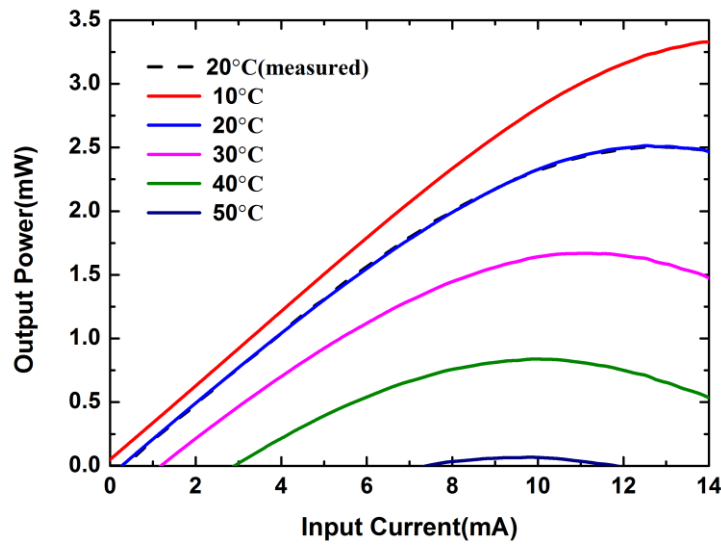


图6 $\eta(T)$ 缓变函数时不同温度下的 L-I 特征曲线

图6表明将 $\eta(T)$ 设置为缓变函数时,拟合效果较佳,VCSEL对于环境温度的工作要求也提高了。图7为 $\eta(T)$ 设置为缓变函数与 $\eta(T)$ 设置为常数时残差绝对值对比图。从图中可以明显看出,改进后的模型的拟合数据与实测数据的残差数值更加稳定。通过设置 $\eta(T)$ 为缓变函数来改进模型很有成效。从图中可以看出,温度为 10℃,理论曲线较之其他温度情况下不是特别符合实际,在没有输出电流的情况下,不会出现输出光功率。

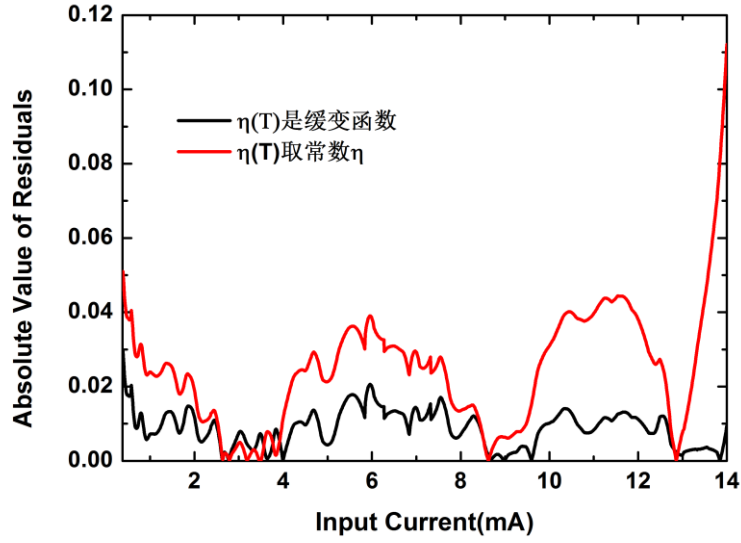


图 7 $\eta(T)$ 为缓变函数和常数时残差绝对值对比图

4.3 增加泄漏电流项优化因子

若没有泄漏电流项，输出功率将随着注入电流的增大无限增加，热饱和和下降现象将会消失，这不符合物理规律。泄漏电流项对于输出功率至关重要，也较为敏感。考虑增加泄漏电流优化项，将其扩充为六阶，即增加两个优化因子 a_5, a_6 。则 L-I 模型为

$$P_0 = \eta \left\{ I - I_{th0} - \sum_{n=0}^{n=6} a_n [T_0 + (IV - P_0)R_{th}]^n \right\}$$

使用 PSO 算法，确定模型参数 ($\eta, I_{th0}, R_{th}, a_0, a_1, a_2, a_3, a_4, a_5, a_6$)。拟合的模型参数结果如下表。

表5 增加泄漏电流项优化因子时拟合模型参数

η	I_{th0}	R_{th}	a_0	a_1
0.257989	0.00299	2522.1699	-0.00182	-5.09E-05
a_2	a_3	a_4	a_5	a_6
3.64E-07	3.74E-09	-9.56E-12	1.82E-15	-4.40E-17

从数据可以看出, a_5 , a_6 数量级分别达到了 10^{-15} 和 10^{-17} , 但由于本身输出功率就是 mW 级的, 即使 a_5 , a_6 比较小, 仍然会对输出功率带来一定的影响。优化算法中的评估量 $SS_{res} = 1.32 * 10^{-7}$, $Re = 9.7169 * 10^{-6}$ 。剩余误差平方和较改进前更小。此组模型参数下, 画出 10°C , 20°C , 30°C ,, 90°C 等温度下的 L-I 特征曲线图如图 8。

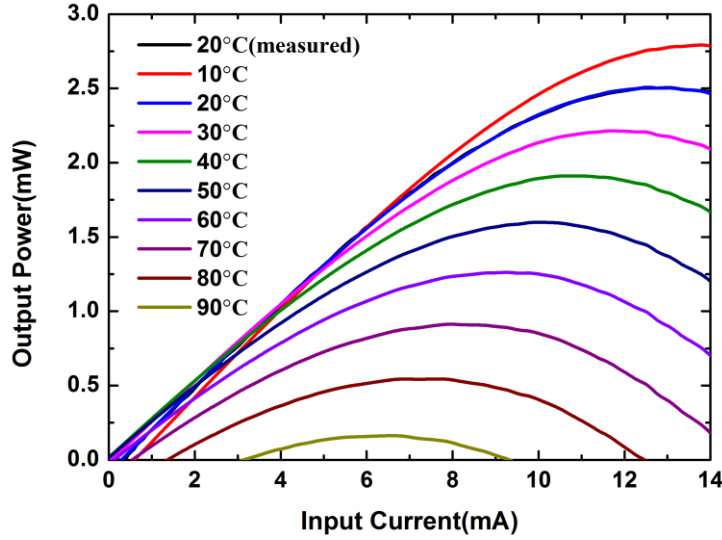


图 8 增加 a_5 , a_6 时不同温度 L-I 特征曲线

为了表明改进是有效的, 对比了增加泄漏电流项优化因子, $\eta(T)$ 设置为缓变函数和未改进前的残差绝对值, 如图 9。图中, 增加泄漏电流项优化因子和 $\eta(T)$ 设置为缓变函数这两次改进的残差绝对值曲线基本重合, 说明这两次改进后, 拟合曲线的稳定性已经大大提升。尤其是降低了注入电流 14 毫安附近的输出功率残差值。但是从 L-I 特征曲线来看, 增加 a_5 , a_6 时, 在注入电流 0-6mA 段, 10°C 的 L-I 特征曲线与其他温度的 L-I 特征曲线出现了交叉的现象, 可能是由泄漏电流对于低温时相对不敏感导致的。从物理意义上来讲, $\eta(T)$ 设置为缓变函数的改进结果较之此次改进结果更有效。

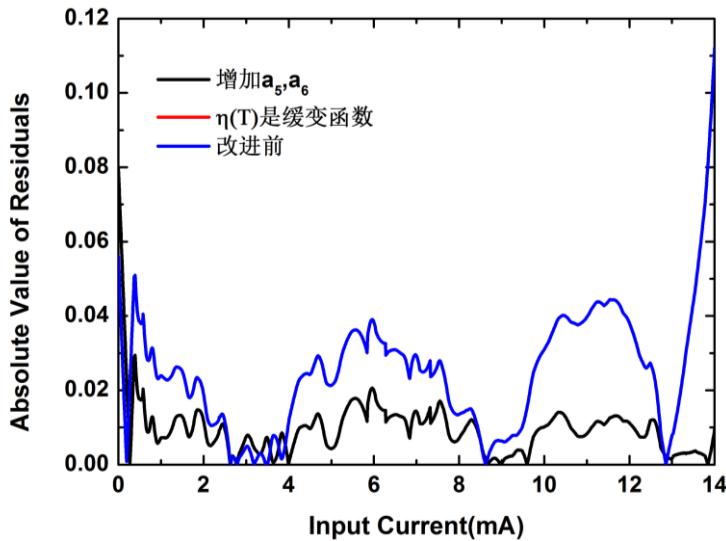


图 9 两次改进与改进前的残差绝对值对比图

4.4 考虑 $\eta(T)$ 对于温度是缓变函数并增加优化因子

在上述两种改进模型方法的基础上，思考将两种改进方法相结合，希望能够得到更好的结果。在改进中，在原模型的基础上，对于 $\eta(T) = \sum_{n=0}^{\infty} b_n T^n$ ，考虑 b_0, b_1, b_2 项的影响；对于泄漏电流优化项，将其扩充为六阶，即增加两个优化因子 a_5, a_6 。故而模型公式可以写为

$$P_0 = \sum_{n=0}^{n=2} b_n (T_0 + (IV - P_0)R_{th})^n \left\{ I - I_{th0} - \sum_{n=0}^{n=6} a_n [T_0 + (IV - P_0)R_{th}]^n \right\}$$

使用 PSO 算法，确定模型参数 ($I_{th0}, R_{th}, b_0, b_1, b_2, a_0, a_1, a_2, a_3, a_4, a_5, a_6$)。拟合的模型参数结果如下表。

表6 十二个拟合模型参数

b_0	b_1	b_2	I_{th0}	R_{th}	a_0
0.276183	0.0030574	-3.95E-05	-0.00208	1000	0.002061
a_1	a_2	a_3	a_4	a_5	a_6
-2.55E-05	1.89E-06	4.98E-09	2.04E-11	2.49E-15	2.47E-16

从拟合数据中可以看出， a_5 和 a_6 数量级都很小，若后期再增加泄漏电流优化阶数意义不是很大。同时， b_2 的数量级也相对较小，再增加阶数同样意义不大。在确立的这十二个模型参数的基础上，画出了 10°C，20°C，30°C，.....，90°C等温度下的 L-I 特征曲线图如图 10。

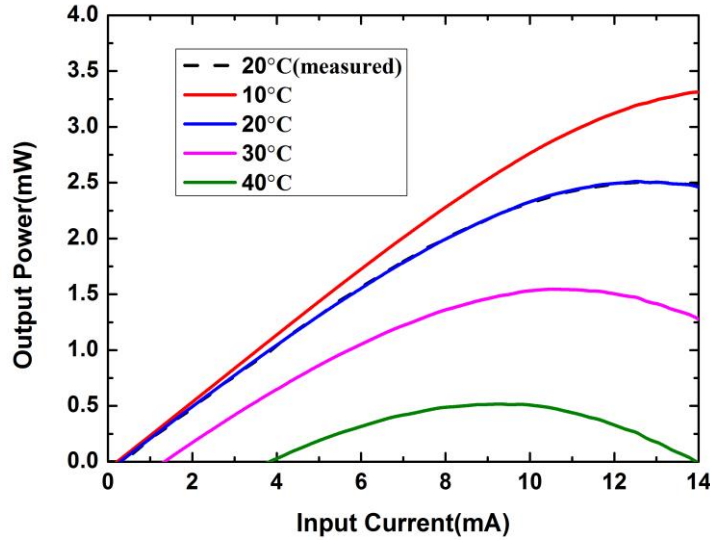


图 10 增加 b_1, b_2, a_5, a_6 时不同温度 L-I 特征曲线

图 10 显示的其他温度下的 L-I 特征曲线与前两组改进后的 L-I 特征曲线是有一些差异的，50°C 及以上温度下的 L-I 特征曲线均无物理意义。可能由于 VCSEL 激光器的 L-I 特性对于温度的依赖性很强，带来微小改变时，就能引起很大的变化。此次模型的评估量 $SS_{res} = 1.62 \times 10^{-7}$ ， $Re = 1.08 \times 10^{-5}$ 。增加 b_1, b_2, a_5, a_6 与只增加 b_1, b_2 项残差绝对值对比图如图 11。

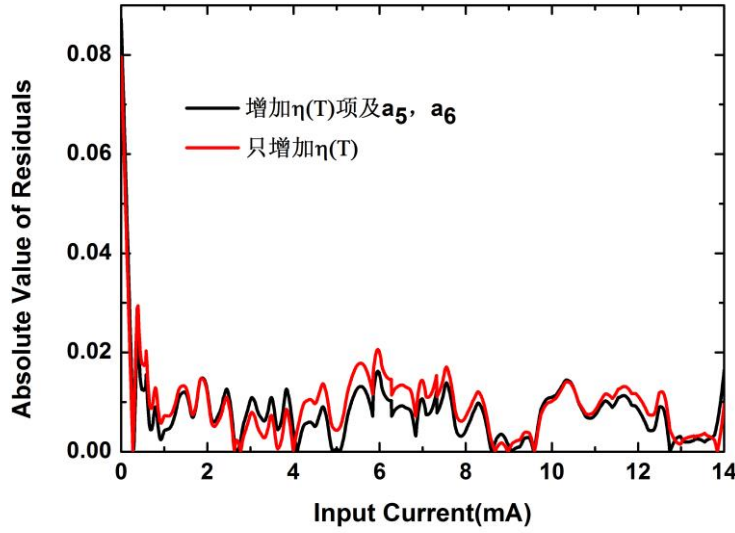


图 11 增加 b_1, b_2, a_5, a_6 与只增加 b_1, b_2 项残差绝对值对比图

从图 11 中可以看出，两种优化的残差值相差并不多，优化都是有效的。 a_5, a_6 两个优化因子的效果可能影响程度相对较低。

4.5 改进模型的分析

只增加 b_1, b_2 的改进模型，存在的问题，从图 6 中可以看出，温度为 10°C 时，VCSEL 激光器的 L-I 特征曲线没有截止电流，当输入电流为 0mA 时，输出功率大于 0，而这在物理上是不成立的。而对于只增加 a_5, a_6 的改进模型，也存在一定的问题，温度为 10°C 时的 L-I 特征曲线与其他温度下的 L-I 特征曲线有交叉。出现这两个问题的可能原因是，模型对于温度低的情况契合度不够高，但对于温度高的情况，模型准确度很好。而第三次改进模型的结果中，虽然对于 20°C 的理论曲线与实测曲线的残差值没有很大的提升，但是实际效果有了很大提升。

4.6 FDTD 建立模型

有限时域差分法利用麦克斯韦方程组在时间和空间进行差分化，从而解决问题。本节考虑使用 FDTD 的方法来建立数学模型。

在半导体激光器中，由于电流注入和器件结电阻以及体电阻等原因会使得器件产生热量，流动产生温度分布。假设单位面积热传递效率为 $g(x,y,z)$ 。产生单位体积热速率 $Q(x,y,z)$ ，矩形热电流传导过程建模如下，热传导过程以及拉普拉斯变换如下：

$$\begin{cases} \nabla g(x, y, z) + \rho C_p \nabla T = Q + \nabla(k(x, y, z, T) \nabla T) \\ g(x, y, z) = -k(x, y, z) \nabla T(x, y, z) \end{cases}$$

其中 k 是热电导， Q 是焦耳热， σ 为电导值。通过追赶法进行求解

热传导的过程中，代入热导率得到传导方程

$$\nabla(-k(x, y, z) \nabla T(x, y, z)) + \rho C_p \nabla T = Q + \nabla(k(x, y, z, T) \nabla T)$$

半导体的热导率和各个分布的温度有关，假设激光器的顶部，底部以及侧面没有能量损耗，热流以及二维方式流入温度 T 的传导热方程边界条件，温度流连续，边界条件如下。

$$\begin{aligned}
T(z=0) &= T \\
\left. \frac{\partial T}{\partial z} \right|_{z=H} &= \left. \frac{\partial T}{\partial r} \right|_{r=0} = \left. \frac{\partial T}{\partial r} \right|_{r=r_s} = 0 \\
k_i \left. \frac{\partial T}{\partial z} \right|_{z=z_i} &= k_j \left. \frac{\partial T}{\partial z} \right|_{z=z_j}
\end{aligned}$$

其中， k 为热导率， H ， r_s 为 VCSEL 的高度和半径。 Q 为热流。有源层热源的非辐射复合以及自发吸收， Q 如下表达式：

$$Q_i(r) = j^2(r)\rho_i$$

采用有限元求解热传导方程，稳态解（ t 趋近于无穷），确定 k 是热电导，是关于 T （温度）的二次函数。

$$\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} + \frac{1}{k(T)} Q(r, t) = 0$$

近似差分求解：

$$\begin{aligned}
\frac{\partial^2 T}{\partial r^2} &= \frac{T(i-1, j) - 2T(i, j) + T(i+1, j)}{\Delta r^2} \\
\frac{\partial^2 T}{\partial z^2} &= \frac{T(i, j-1) - 2T(i, j) + T(i, j+1)}{\Delta z^2}
\end{aligned}$$

耦合后求自洽，采用有限元追赶法求解热传导方程，对其进行定性的仿真结果如下。

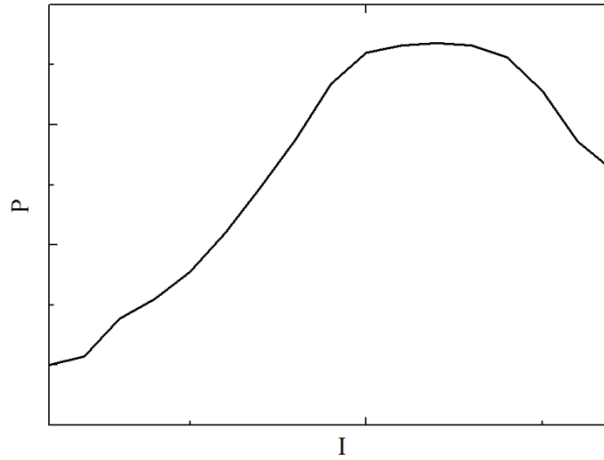


图 12 FDTD 模型下 L-I 定性分析

五、 问题 3：VCSEL 的带宽模型

5.1 问题分析

VCSEL 激光器因其较短的谐振腔，它的弛豫振荡频率和调制带宽都很高。传输带宽在通信中是很重要的概念。如何拓宽传输带宽是实际应用中的关键问题。本题的目的就是围绕着 VCSEL 的带宽问题而展开。题中所提出的小信号模型就是要给 VCSEL 激光器输入幅度小的多频信号。一般小信号响应模型的解决思路是先由速率方程求得稳态解，进而引入小信号，得到小信号幅频响应曲线，确立小信号响应模型的参数。并在确立参数的基础上，画出不同环境温度 and 不同偏置电流下的带宽响应曲线，对其进行相应的分析。在此基础上思考如何优化其带宽性质。

5.2 模型建立

本次建模的思想主要就是在基于速率方程稳态解的基础上，加入小信号，给出小信号幅频响应模型，并在理论驱动电流和实际驱动电流误差值，理论光功率与实际光功率误差值约束条件下，确立最佳的模型参数，进而获得不同温度 and 不同驱动电流下的小信号幅频响应曲线。

在此次模型中有如下假设：

- (1) 忽略二阶小量的影响；
- (2) 调制频率较高时，增益、透明载流子数、漏电流以及温度随小信号波动幅度微小，忽略不计；
- (3) 假设稳态响应可以写为稳态值与交变值之和；
- (4) 假设小信号响应为线性频率响应。

根据附录中的速率方程的思想，将偏置电流和注入激光器的外部驱动电流代入激光器速率方程，得到：

$$\begin{cases} \frac{dN}{dt} = \frac{\eta_i(I - I_{th0} - I_{off}(T))}{q} - \frac{N}{\tau_n} - \frac{G_0(N - N_0)S}{1 + \varepsilon S} \\ \frac{dS}{dt} = -\frac{S}{\tau_p} + \frac{\beta N}{\tau_n} + \frac{G_0(N - N_0)S}{1 + \varepsilon S} \end{cases}$$

且 VCSEL 输出的光功率与光子数成正比，即 $P_0 = kS$ 。

在稳态情况下，没有驱动信号，载流子数和光子数都是稳定的， $\frac{dN}{dt} = \frac{dS}{dt} = 0$ ，可以求得：

$$N_s = \frac{\frac{P_0}{k\tau_p} + G_0 N_0 P_0 / (k + \varepsilon P_0)}{\frac{\beta}{\tau_n} + G_0 P_0 / (k + \varepsilon P_0)}$$

根据载流子浓度 N_s 可以表示转换成光能的激光驱动电流 I_{cur} 和激光器稳态下的光子数 S_s ：

$$I_{cur} = \frac{q}{\eta} \left(\frac{N_s}{\tau_n} + \frac{G_0(N_s - N_0)P_0}{k + \varepsilon P_0} \right) + I_{th0} + I_{off}(T)$$

$$S_s = \frac{\eta_i (I - I_{th0} - I_{off}(T)) / q - N_s / \tau_n}{G_0(N_s - N_0)}$$

其中，在推导得到 S_s 的表达式时，认为 $\varepsilon S_s \ll 1$ ，将其忽略不计。

由 $P_s = kS_s$ 可以得到理论上的出光功率的表达式

$$P_s = k \frac{\eta_i (I - I_{th0} - I_{off}(T)) / q - N_s / \tau_n}{G_0(N_s - N_0)}$$

接下来，加载小信号，将稳态响应写为稳态值与交变值之和：

$$I(t) = I_s + i(f) e^{j2\pi ft}$$

$$N(t) = N_s + n(f) e^{j2\pi ft}$$

$$S(t) = S_s + s(f) e^{j2\pi ft}$$

其中， $i(f), n(f), s(f)$ 为小信号引入了与频率相关的电流、载流子和光子数，并且这三个值足够小。

在引入小信号的情况下，原来速率方程中电流、载流子、光子数也需要替换为稳态值与小信号引入的交变值之和。可得下式：

$$\begin{aligned} \frac{d(N_s + n(f) e^{j2\pi ft})}{dt} &= \frac{\eta_i (I_s + i(f) e^{j2\pi ft} - I_{th0} - I_{off}(T))}{q} - \frac{N_s + n(f) e^{j2\pi ft}}{\tau_n} - \frac{G_0 (N_s + n(f) e^{j2\pi ft} - N_0) (S_s + s(f) e^{j2\pi ft})}{1 + \varepsilon (S_s + s(f) e^{j2\pi ft})} \\ \frac{d(S_s + s(f) e^{j2\pi ft})}{dt} &= -\frac{S_s + s(f) e^{j2\pi ft}}{\tau_p} + \frac{\beta (N_s + n(f) e^{j2\pi ft})}{\tau_n} + \frac{G_0 (N_s + n(f) e^{j2\pi ft} - N_0) (S_s + s(f) e^{j2\pi ft})}{1 + \varepsilon (S_s + s(f) e^{j2\pi ft})} \end{aligned}$$

对上式进行下一步化简

$$\begin{aligned} \frac{d(N_s + n(f) e^{j2\pi ft})}{dt} &= \frac{\eta_i (I_s + i(f) e^{j2\pi ft} - I_{th0} - I_{off}(T))}{q} - \frac{N_s + n(f) e^{j2\pi ft}}{\tau_n} - \frac{G_0 (N_s + n(f) e^{j2\pi ft} - N_0) (S_s + s(f) e^{j2\pi ft})}{1 + \varepsilon (S_s + s(f) e^{j2\pi ft})} \\ (j2\pi f) n(f) e^{j2\pi ft} &= \frac{\eta_i (I_s - I_{th0} - I_{off}(T))}{q} - \frac{N_s}{\tau_n} + \frac{\eta_i i(f) e^{j2\pi ft}}{q} - \frac{n(f) e^{j2\pi ft}}{\tau_n} \\ &\quad - G_0 \cdot \frac{(N_s - N_0) S_s + S_s n(f) e^{j2\pi ft} + (N_s - N_0) s(f) e^{j2\pi ft} + n(f) s(f) e^{j2\pi(2f)t}}{1 + \varepsilon S_s + \varepsilon s(f) e^{j2\pi ft}} \\ &= \frac{\eta_i (I_s - I_{th0} - I_{off}(T))}{q} - \frac{N_s}{\tau_n} - \frac{G_0 (N_s - N_0) S_s}{1 + \varepsilon S_s + \varepsilon s(f) e^{j2\pi ft}} \\ &\quad - \frac{G_0 n(f) s(f) e^{j2\pi(2f)t}}{1 + \varepsilon S_s + \varepsilon s(f) e^{j2\pi ft}} \\ &\quad + \frac{\eta_i i(f) e^{j2\pi ft}}{q} - \frac{n(f) e^{j2\pi ft}}{\tau_n} - \frac{G_0 S_s n(f) e^{j2\pi ft} + G_0 (N_s - N_0) s(f) e^{j2\pi ft}}{1 + \varepsilon S_s + \varepsilon s(f) e^{j2\pi ft}} \end{aligned}$$

其中，由于已经假设二阶小量可以忽略，故而 $\varepsilon s(f)$ 和 $e^{j2\pi(2f)t}$ 都足够小，进行化简时均可以忽略不计，有下式

$$\begin{aligned} (j2\pi f) n(f) e^{j2\pi ft} &= \frac{\eta_i (I_s - I_{th0} - I_{off}(T))}{q} - \frac{N_s}{\tau_n} - \frac{G_0 (N_s - N_0) S_s}{1 + \varepsilon S_s} \\ &\quad + \frac{\eta_i i(f) e^{j2\pi ft}}{q} - \frac{n(f) e^{j2\pi ft}}{\tau_n} - \frac{G_0 S_s n(f) e^{j2\pi ft}}{1 + \varepsilon S_s} - \frac{G_0 (N_s - N_0) s(f) e^{j2\pi ft}}{1 + \varepsilon S_s} \end{aligned}$$

上式右边前三项为稳态时情况，均为0，化简得

$$(j2\pi f)n(f) = \frac{\eta_i i(f)}{q} - \frac{n(f)}{\tau_n} - \frac{G_0 S_s n(f)}{1 + \varepsilon S_s} - \frac{G_0 (N_s - N_0) s(f)}{1 + \varepsilon S_s}$$

同理，可得

$$(j2\pi f)s(f) = -\frac{s(f)}{\tau_p} + \frac{\beta n(f)}{\tau_n} + \frac{G_0 (N_s - N_0) s(f)}{1 + \varepsilon S_s} + \frac{G_0 S_s n(f)}{1 + \varepsilon S_s}$$

至此，可得 $n(f)$ 和 $s(f)$ 的关系

$$n(f) = \frac{j2\pi f + 1/\tau_p - G_0 (N_s - N_0)/(1 + \varepsilon S_s)}{\beta/\tau_n + G_0 S_s/(1 + \varepsilon S_s)} s(f)$$

依据假设，小信号响应为线性频率响应 $h(f)$ ，按照定义表示为小信号带来的输出

信号 $p(f)$ 与输入信号 $i(f)$ 的比值

$$h(f) = \frac{p(f)}{i(f)} = k \frac{s(f)}{i(f)}$$

代入数值，化简可得

$$\begin{aligned} h(f) &= k \frac{s(f)}{i(f)} \\ &= \frac{k\eta_i}{q} \cdot \frac{1}{\left[\frac{j2\pi f + 1/\tau_p - G_0 (N_s - N_0)/(1 + \varepsilon S_s)}{\beta/\tau_n + G_0 S_s/(1 + \varepsilon S_s)} \right] \cdot \left[\frac{j2\pi f + 1/\tau_n + G_0 S_s/(1 + \varepsilon S_s)}{\beta/\tau_n + G_0 S_s/(1 + \varepsilon S_s)} \right] + G_0 (N_s - N_0)/(1 + \varepsilon S_s)} \end{aligned}$$

可简写为

$$h(f) = \frac{X}{(j2\pi f)^2 + (j2\pi f)Y + Z}$$

其中，

$$Y = \frac{1}{\tau_p} + \frac{1}{\tau_n} + \frac{G_0 P_s}{k + \varepsilon P_s} - \frac{G_0 (N_s - N_0)}{(1 + \varepsilon P_s/k)^2}$$

$$Z = \frac{1}{\tau_p \tau_n} + \frac{G_0 P_s}{\tau_p (k + \varepsilon P_s)} - \frac{(1 - \beta) G_0 (N_s - N_0)}{\tau_n (1 + \varepsilon P_s/k)^2}$$

最终归一化的信号响应为

$$H(f) \equiv \frac{h(f)}{h(0)} = \frac{Z}{(j2\pi f)^2 + (j2\pi f)Y + Z}$$

将归一化的信号响应化为分贝为单位，则有

$$G(f) = 20 \lg H(f) = 20 \lg \left(\frac{Z}{(j2\pi f)^2 + (j2\pi f)Y + Z} \right)$$

在本次的模型中，需要拟合的模型参数有 8 个， $\eta_i, \beta, \tau_n, k, G_0, N_0, \tau_p, \varepsilon$ 。在

此次模型中，有三个误差约束条件：

- (1) 理论驱动电流 I_{cur} 与实测驱动电流 I （7.5mA）误差最优；
- (2) 理论光功率 P_s 与实测激光器光功率 P_0 误差最优；
- (3) 理论小信号响应 $G(f)$ 与 S21 中的实测数据误差最优。

由于方程中参数的数量级差别非常大，如果将八个参数放入最终模型中进行求解，会给方程的求解造成一定的困难。故而对于参数的确立，可以分两个步骤：根据误差约束条件(3),由附件 S21 的信息与 $G(f)$ 模型进行协同优化，确立 Y 和 Z 的参数；再利用误差约束条件(1)和(2)，以及确立的 Y 和 Z 参数，再参考值的数量级基础上，改进拉格朗日乘法，拟合八个模型参数 $(\eta_l, \beta, \tau_n, k, G_0, N_0, \tau_p, \varepsilon)$ 。

具体求解模型流程如下图。

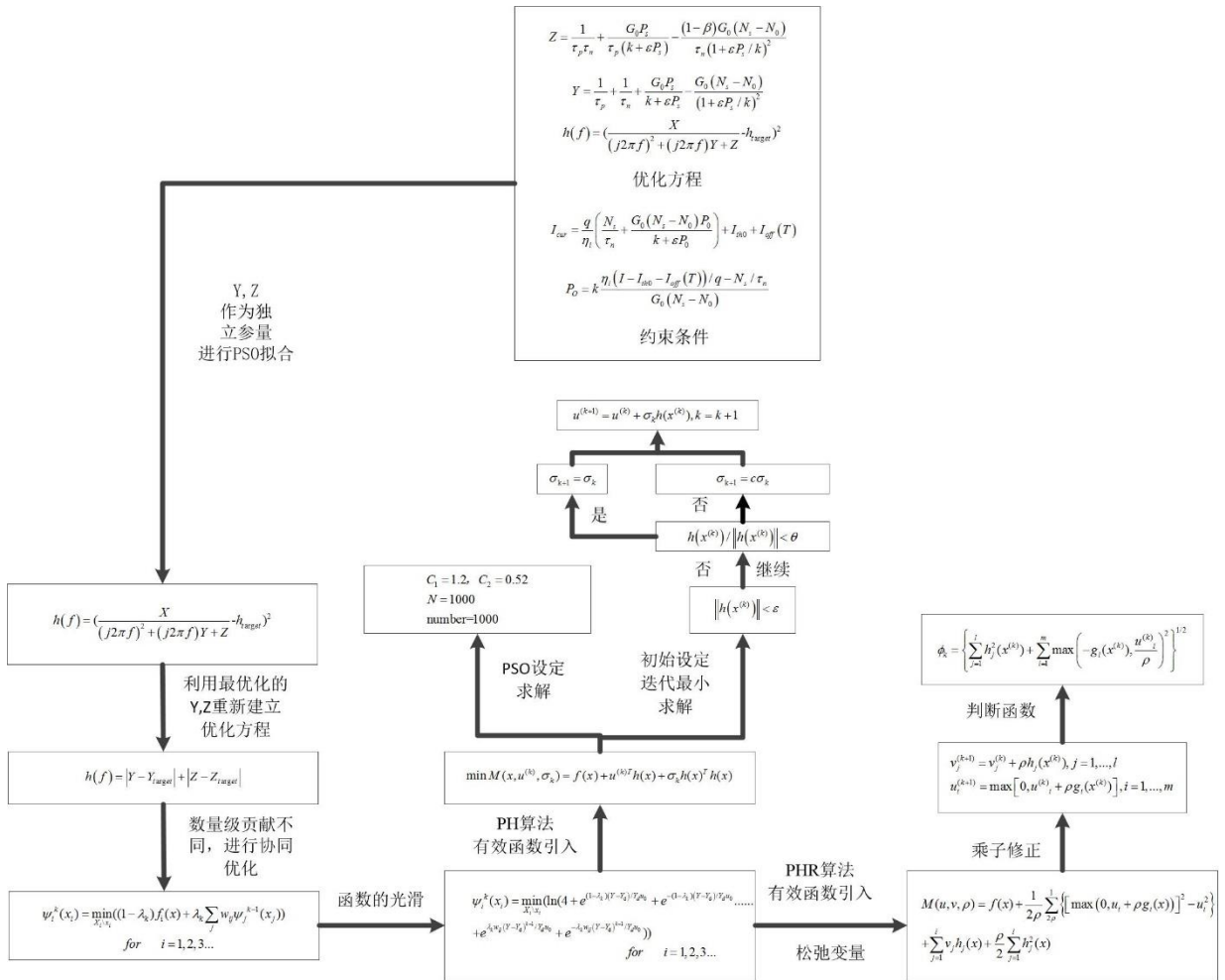


图 13 求解模型流程图

5.3 算法介绍

(1)小波变换

小波变换在图像处理、模式识别、计算机视觉领域多方面应用普遍。小波变换对于不同的频率在时域上的取样步长可调节，符合低频信号变化缓慢而高频信

号变化迅速的特点，也就是在低频时小波变化的时间分辨率较低，频率分辨率较高；在高频时小波变换的时间分辨率较高，而频率分辨率较低。利用小波的多分辨率的特点，可以进行图像处理，降低图像中的噪声干扰，提取有用的信息数据，并且能够对信号进行多尺度细化分析。[6]

设 $\psi(t) \in L(R) \cap L^2(R)$, 其傅里叶变换为 $\hat{\psi}(\omega)$, 若傅里叶变换满足如下条件

$$C_\psi = \int_R \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty$$

则将 $\psi(t)$ 称为基本小波，基本小波在尺度伸缩和平移作用下可以得到下列函数族，称为由基本小波生成的连续小波：

$$\psi_{a,b} = |a|^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right), a \in R, a \neq 0, b \in R$$

其中， a 称作尺度因子， b 称为平移因子。上式为连续小波的定义，也可以将连续小波变换离散化得到离散小波变换。

小波变换的多分辨率功能被称为数学显微镜。而小波变换的“显微镜”功能就是通过尺度因子来实现的。当尺度因子较大时，视野宽而分辨率低，适合粗观察；当尺度因子较小时，视野窄而分辨率高，适合细观察。小波变换的多分辨率特点有如下的理论背景。

定义空间 $L^2(R)$ 中的一列闭子空间 $\{V_j\}_{j \in Z}$ 称为 $L^2(R)$ 的一个多分辨率的分析，

同时，这一列空间需要满足如下条件：

伸缩性： $f(t) \in V_{m-1}$ 等价于 $f(2t) \in V_m, \forall m \in Z$;

平移不变性： $f(t) \in V_m$ 等价于 $f(t - 2^{-m}k) \in V_m, \forall k \in Z$;

渐进完全性： $\bigcap_{m \in Z} V_m = \{0\}, \overline{\bigcup_{m \in Z} V_m} = L^2(R)$;

一致单调性：对于任意 $m \in Z, V_m \subset V_{m+1}$;

Riesz 基：存在 $\phi(t) \in V_0$ ，使得 $\{\phi(t - k)\}_{k \in Z}$ 构成 V_0 的标准正交基；

函数空间的逐级剖分性，即

$$V_0 = V_1 \otimes W_1, V_1 = V_2 \otimes W_2, \dots, V_m = V_{m+1} \otimes W_{m+1}$$

因此，空间列 $\{V_m\}_{m \in Z}$ 中任一空间 V_m 的基都可以通过其中另一个空间 V_m 的基经过伸缩变换得到。 $\phi(t)$ 可称为尺度函数。

(2)改进拉格朗日乘子法

对于求取约束条件的优化问题，拉格朗日和 KKT 条件是两个非常重要的求取方法，对于等式约束的优化问题，应用朗格朗日乘子计算；对于不等式的约束条件，应用 KKT 条件。在这我们应用拉格朗日求解最优化问题。

约束优化问题的非线性规划问题为：

$$\begin{aligned} \min & f(x), x \in R^n \\ \text{s.t.} & g_i(x) \leq 0, i = 1, 2, \dots, m \\ & h_j(x) = 0, j = 1, 2, \dots, l \end{aligned}$$

其中 $f, g_i, h_j : R^n \rightarrow R$

基本思想就是通过引入效用函数的方法将约束优化问题转换为无约束问题，再不断更新效用函数，收敛算法。

研究者发现构造一种神经网络的最优化问题都需要目标函数和约束函数的光滑性,因此学习解决构造目标函数的光滑问题是极其重要的。无论待解决的优化问题是否光滑我们都是在其中添加了惩罚因子,但在惩罚函数的构造上也存在不少问题,在这提出了解决这一致命问题的方法,通过把约束合并到一个修正目标函数中来解决拉格朗日乘子问题。

在复杂的设计过程中,需要考虑各个参量之间的相互作用及其印象,为了解决这个问题,我们提出了协同优化的方法,充分探索利用工程中的协同作用优化,其结构简单算法收敛性较强,得到了广泛的重视以及应用。

协同方法设计到两个部分一个是整体的系统还有一个是并行的协同过程,系统的并行过程分别满足约束条件,使得之间相互约束相互制衡,得到的函数可以一致性优化,最后达到一个一致协调优化的作用。

1) 罚函数法

罚函数法(内点法)的主思想是:可行域就像一个“无限深势阱”,边界处的目标函数极大,迭代点无法穿越边界,最优解始终被“困”在可行域之内了。

它只适用于不等式约束:

$$\begin{aligned} \min f(x), x \in R^n \\ s.t. \quad g_i \leq 0, i=1,2,\dots,m \end{aligned}$$

它的可行域为:

$$D = \{x \in R^n \mid g_i(x) \leq 0, i=1,2,\dots,m\}$$

对上述约束问题,其可行域的内点可行集 $D_0 \neq \emptyset$ 的情况下,引入效用函数:

$$\min B(x, r) = f(x) + r\tilde{B}(x) \text{、}$$

$$\text{其中 } \tilde{B}(x) = \sum_{i=1}^m -\frac{1}{g_i(x)} \text{ 或 } \tilde{B}(x) = \sum_{i=1}^m |\ln(-g_i(x))|$$

算法的具体步骤如下:

给定控制误差 $\varepsilon > 0$, 惩罚因子的缩小系数 $0 < c < 1$ 。

步骤 1: 令 $k=1$, 选定初始点 $x^{(0)} \in D_0$, 给定 $r_1 > 0$ (一般取 10)。

步骤 2: 以 $x^{(k)}$ 为初始点, 求解无约束

$$\min B(x, r) = f(x) + r_k \tilde{B}(x)$$

$$\text{其中 } \tilde{B}(x) = \sum_{i=1}^m -\frac{1}{g_i(x)} \text{ 或 } \tilde{B}(x) = \sum_{i=1}^m |\ln(-g_i(x))|, \text{ 得最优解 } x^{(k)} = x(r_k)$$

步骤 3: 若 $r_k \tilde{B}(x^{(k)}) < \varepsilon$, 则 $x^{(k)}$ 为其近似最优解, 停; 否则, 令 $r_k = cr_k, k=k+1$,

转步骤 2.

2) 协调函数设定

对于目标函数 $f(x_1, x_2, \dots, x_n)$ 简写成 $f(x)$, 协同算法将其分解成几个 n

的目标函数。[7]

$$f(x) = f_1(x) + f_2(x) + \dots + f_n(x)$$

对于单独优化每一个目标函数，则他们的结果很难一致。将变量包含最优解很难相同，对于 $i=1,2$ 我们最优解可以表示成 \tilde{x}_i

$\tilde{x}_i = \arg \min_{x_i} \min_{X_i \setminus x_i} f_i(x)$, 我们很难体现出原函数的最优，那么我们需要子目标函数的优化一致。同时考虑子函数目标优化

$$\psi_j(x_j) = \min_{X_j \setminus x_j} f_j(x)$$

我们的两个目标函数如下（其数量级差了 11 个数量级）

$$f_1(x) = |Y - Y_d|, f_2(x) = |Z - Z_d|$$

我们具体做法是利用其它子目标优化结果通过数值加权修正子目标：

$$(1 - \lambda) f_1(x) + \lambda_k \sum_j w_{ij} \psi_j(x_j)$$

在这里， $\lambda_k \geq 0$ ， $w_{ij} \leq 1$ 是加权系数，其中

我们每一个函数进行加权优化叠加放入修正后的子目标函数中，协同算法叠加如下。

$$\psi_i^k(x_i) = \min_{X_i \setminus x_i} ((1 - \lambda_k) f_i(x) + \lambda_k \sum_j w_{ij} \psi_j^{k-1}(x_j)), \text{ for } i = 1, 2, 3 \dots$$

在此我们的协调函数保持一致，解得最优值。

3) 光滑函数设定[8]

对于 $R^n \rightarrow R$ 是 Lipschitz 函数，我们把函数 $\tilde{f}(x, \mu): R^n \times [0, +\infty)$ 的函数 g 叫做光滑函数，此函数满足条件

- 给定 $u > 0$ 上，在 R^n 上是连续可微的，并且给定任意的 $x \in R^n$ 都 $\tilde{f}(x, \mu)$ 可微。
- 对于给定任意的 $x \in R^n$ ， $\lim_{u \downarrow 0} \tilde{f}(x, \mu) = f(x)$
- 对于正常数 $K_g > 0$ 有以下关系

$$|\nabla_u \tilde{f}(x, \mu)| \leq K_g, \quad \forall \mu \in [0, +\infty), x \in R^n$$

- $\left\{ \lim_{z \rightarrow x, u \downarrow 0} |\nabla_u \tilde{f}(x, \mu)| \subseteq \partial \tilde{f}(x) \right\}$

对于给定的 $x \in R^n$ ，我们可以构造光滑函数，本文使用光滑函数 $\tilde{f}(x, \mu)$ 近似非光滑函数 $f_1(x) = |Y - Y_d|$ ， $f_2(x) = |Z - Z_d|$ ，利用光滑函数逼近技术，将非线性

函数 $f(x)$ 转化成光滑函数 $\tilde{f}(x, \mu)$ 。选择 $\mu(t) = e^{-t} / (1 + e^{-t})^2$

当 $\mu_0 > 0$ 时, 光滑函数为

$$f(s, u_0) = \int_{-\infty}^{+\infty} (s - u_0 t) + u(t) dt = u_0 \ln(1 + e^{s/u_0})$$

那么此时 $|Y - Y_d| = (Y - Y_d)_+ + (-(Y - Y_d))_-$ 可以得到光滑函数为:

$$f(s, u_0) = u_0 \ln(2 + e^{s/u_0} + e^{-s/u_0})$$

那么我们结合协同函数代入

$$\begin{aligned} \psi_i^k(x_i) = \min_{X_i \setminus x_i} & (\ln(4 + e^{(1-\lambda_k)(Y-Y_d)/Y_d u_0} + e^{-(1-\lambda_k)(Y-Y_d)/Y_d u_0} \dots \\ & + e^{\lambda_k w_{ij}(Y-Y_d)^{k-1}/Y_d u_0} + e^{-\lambda_k w_{ij}(Y-Y_d)^{k-1}/Y_d u_0})) \\ & \text{for } i = 1, 2, 3 \dots \end{aligned}$$

4) Pso 调用参数

假设一个 8 维的搜索空间, 存在 1000 个粒子组成了一个种群, 其中第 i 个粒子表示为 8 维向量, $x = (x_1, x_2, \dots, x_8)$, 那么我们设定种群最优位置, 不断更新最优位置和速度, 我们引入两个加速因子分别 1.2, 0.5 和惯性权值为 0.005。

5) 拉格朗日乘子法统一计算[9]

拉格朗日乘子法统一计算一般包含 PH 算法和 PHR 算法。

PH 算法一般指约束为等式的情况引入:

效用函数为

$$\min M(x, u^{(k)}, \sigma_k) = f(x) + u^{(k)T} h(x) + \sigma_k h(x)^T h(x)$$

判断函数为

$$\phi_k = \|h(x^{(k)})\|$$

当 $\phi_k = \phi(x^{(k)}) < \varepsilon$ 时迭代停止。

步骤 1: 选定初始点 $x^{(0)}$, 初始拉格朗日乘子向量 $u^{(1)}$, 初始罚因子 σ_1 及其放大系数 $c > 1$, 控制误差 $\varepsilon > 0$ 与常数 $\theta \in (0, 1)$, 令 $k = 1$ 。

步骤 2: 以 $x^{(k+1)}$ 为初始点, 通过 PSO 算法求解无约束问题:

$$\min M(x, u^{(k)}, \sigma_k) = f(x) + u^{(k)T} h(x) + \sigma_k h(x)^T h(x)$$

得到无约束问题最优解 $x^{(k)}$

步骤 3: 当 $\|h(x^{(k)})\| < \varepsilon$ 时, $x^{(k)}$ 为所求的最优解, 停; 否则转步骤 4。

步骤 4: 当 $h(x^{(k)})/\|h(x^{(k)})\| < \theta$ 时, 转步骤 5; 否则令 $\sigma_{k+1} = c\sigma_k$, 转步骤 5.

步骤 5: 令 $u^{(k+1)} = u^{(k)} + \sigma_k h(x^{(k)})$, $k = k + 1$, 转步骤 1。

PHR 算法一般约束形式的松弛变量法和指数形式法:
松弛变量法:

$$M(u, v, \rho) = f(x) + \frac{1}{2\rho} \sum_{i=1}^l \left\{ \left[\max(0, u_i + \rho g_i(x)) \right]^2 - u_i^2 \right\} \\ + \sum_{j=1}^l v_j h_j(x) + \frac{\rho}{2} \sum_{j=1}^l h_j^2(x)$$

乘子的修正公式为:

$$v_j^{(k+1)} = v_j^{(k)} + \rho h_j(x^{(k)}), j = 1, \dots, l \\ u_i^{(k+1)} = \max[0, u_i^{(k)} + \rho g_i(x^{(k)})], i = 1, \dots, m$$

判断函数为:

$$\phi_k = \left\{ \sum_{j=1}^l h_j^2(x^{(k)}) + \sum_{i=1}^m \max \left(-g_i(x^{(k)}), \frac{u_i^{(k)}}{\rho} \right)^2 \right\}^{1/2}$$

当 $\phi_k = \phi(x^{(k)}) < \varepsilon$ 时迭代停止。

5.4 曲线平滑

附件中 S21 实测数据经过 FFT 变换去噪尝试, 小频率部分数据波动较大, 数据可能存在比较大的实验误差。而从物理意义上分析, 在频率为 0 时, 小信号响应应该为 0, 且在频率较小时, 小信号响应理应接近于 0。实测数据曲线的抖动比较大, 在用模型拟合求解的过程中, 会增加难度。

附录中有提供参考初值, 如表 7。将参考初值代入到已经建立的模型中, 可以得到的小信号响应曲线如图 14。从图 14 可以看出, 参考值代入建立的小信号模型中得到的小信号响应曲线与实测数据偏离较大。

表7 小信号模型参考值

参数	参考初值	单位
η_i	0.7	-
β	1E-5	-
τ_n	9.6E-9	s
k	1.5E-8	W
G_0	1.8E6	s ⁻¹
N_0	4.97E5	-
τ_p	3.8E-12	s
ε	4.7E-8	-

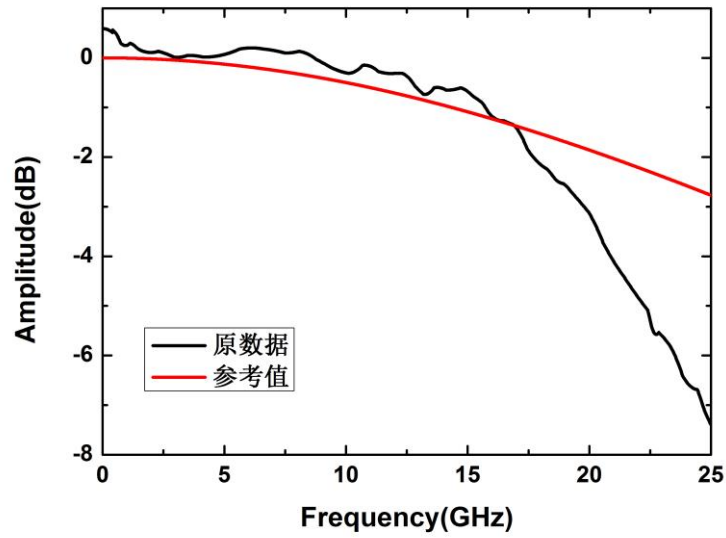


图 14 参考值代入小信号模型后理论曲线

由小信号模型得到的小信号响应曲线比较平滑,图 14 中也可以看出这一点。而实测数据抖动大,实现较高的拟合度响度比较难。所以在进行求解模型之前,考虑使用离散小波变换的多分辨率特点,将实测数据曲线进行平滑处理,得到效果图如图 15。

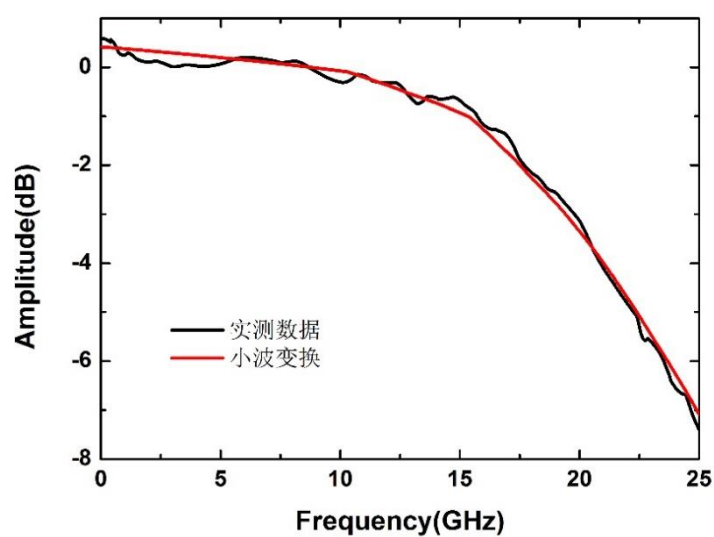


图 15 小波变换平滑效果图

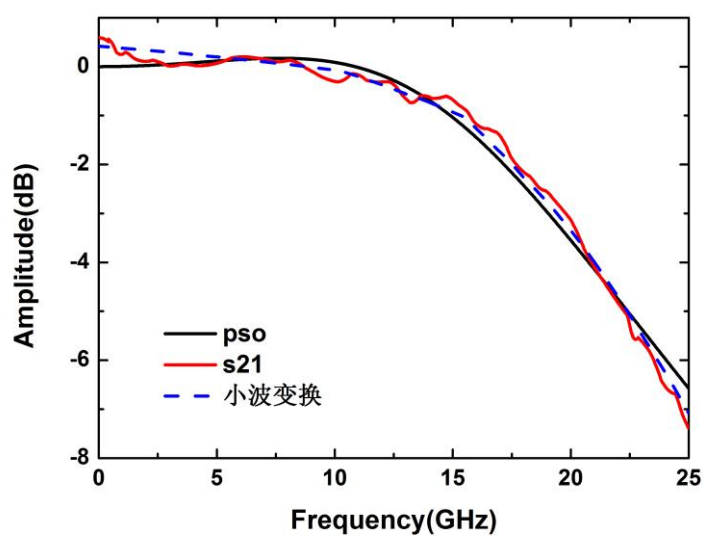
5.5 求解模型

(1) 确立 Y 和 Z

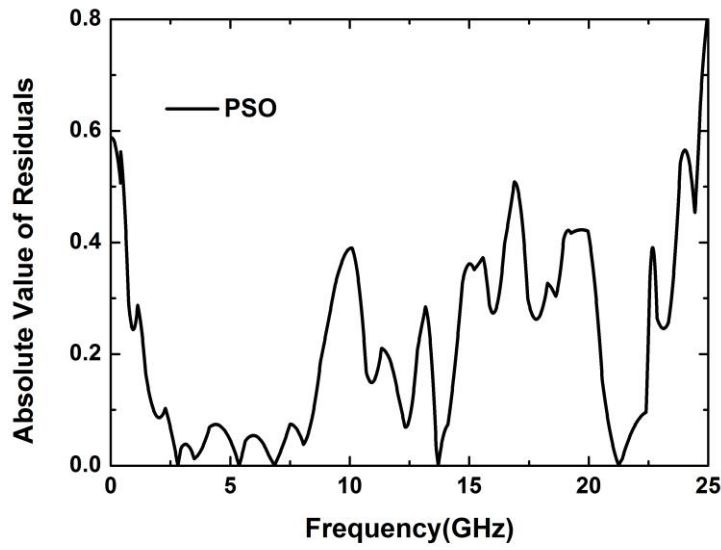
首先，根据模型

$$G(f) = 20\lg\left(\frac{Z}{(j2\pi f)^2 + (j2\pi f)Y + Z}\right)$$

使用问题 1 中 PSO 算法，以小波变换平滑后的数据为目标，可以拟合得到模型参数 Y,Z。模型的评估量 $SS_{res} = 99.2$, $Re = 0.199$ ，误差值较小。图 16(a)为拟合曲线与实测数据曲线的对比图。图 16(b)为带宽模型下拟合所得带宽响应曲线与实测带宽响应曲线的残差绝对值图。最终确立的 Y 为 $1.378e+11$, Z 为 $1.1778e+22$ 。



(a)PSO 算法拟合曲线



(b) 拟合所得带宽响应曲线与实测带宽响应曲线的残差绝对值图

图 16 拟合带宽响应曲线及残差值绝对值图

(2) 拟合模型参数 $(\eta_I, \beta, \tau_n, k, G_0, N_0, \tau_p, \varepsilon)$

在确立 $Y = 1.378e + 11/s, Z = 1.1778e + 22/s^2$ 的情况下，运用拉格朗日乘子法，在理论驱动电流 I_{cur} 与实测驱动电流 I (7.5mA) 以及率 P_s 与实测激光器光功率 P_0 误差最优的情况下，获取最优参数值。拟合所得模型参数

$(\eta_I, \beta, \tau_n, k, G_0, N_0, \tau_p, \varepsilon)$ 如表 8:

表8 小信号模型拟合参数

参数	拟合值	单位
η_i	0.3	-
β	1E-9	-
τ_n	3.1E-7	s
k	1.02E-4	W
G_0	7.381E9	s^{-1}
N_0	4.232E9	-
τ_p	1.167E-11	s
ε	1.5E-9	-

在此参数值下，有 $I_{cur} = 0.0078A, P_s = 0.0019W$ ，实际测量值 $I =$

$0.0075A, P_0 = 0.0019W$ 。理论输出功率和实际输出功率没有误差，理论输入电流与实际偏置电流相差 $0.0003A$ ，相差值很小，满足了误差约束条件。此次建立的小信号响应理论模型是可靠的。

保持环境温度不变，始终为 $20^{\circ}C$ ，改变偏置电流分别为 $4.5mA$, $6mA$, $7.5mA$, $9mA$ ，作出相同温度不同偏置电流下的带宽响应曲线如图 17；保持偏置电流 $7.5mA$ 不变，改变环境温度分别为 $20^{\circ}C$, $30^{\circ}C$, $40^{\circ}C$, $50^{\circ}C$ ，作出相同偏置电流不同温度下的带宽响应曲线如图 18。

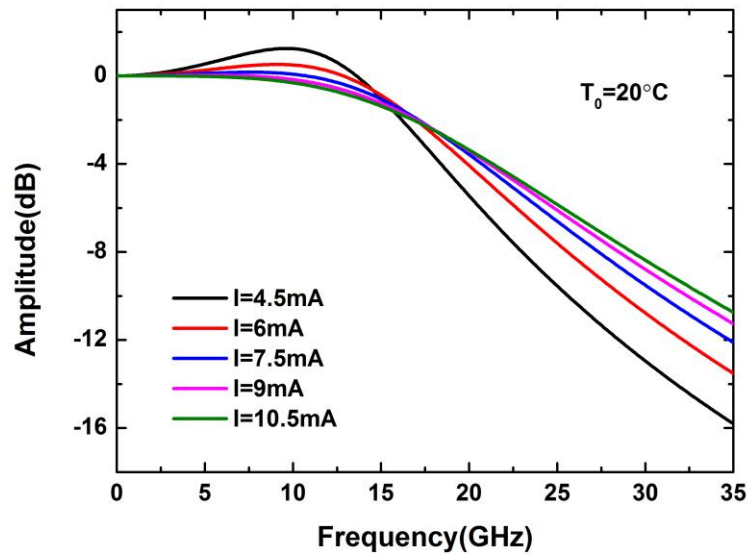


图 17 温度 $20^{\circ}C$ 不同偏置电流带宽曲线

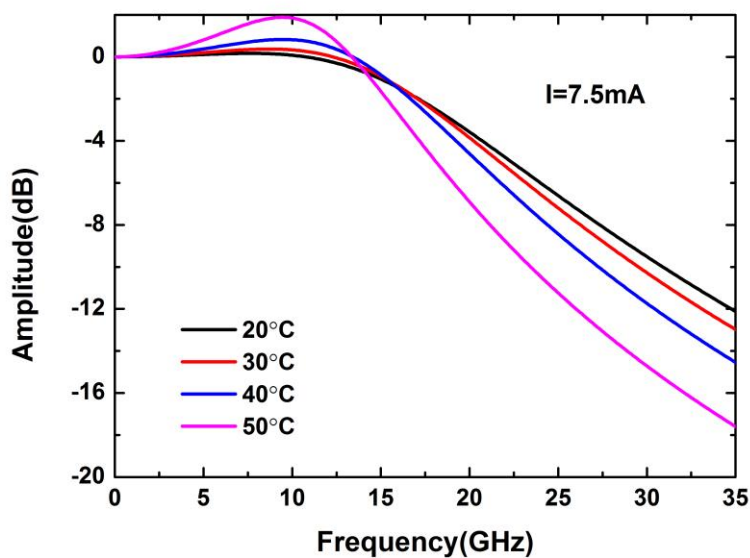


图 18 偏置电流 $7.5mA$ 不同温度带宽曲线

5.6 分析带宽响应曲线

根据拟合带宽响应曲线的数据，可得到-10dB 带宽，如表 9：

表9 不同温度及偏置电流下的-10dB 带宽

$T_0=20^{\circ}\text{C}$	$I=4.5\text{mA}$	25.6GHz
	$I=6\text{mA}$	28.7GHz
	$I=7.5\text{mA}$	30.9GHz
	$I=9\text{mA}$	32.4GHz
	$I=10.5\text{mA}$	33.4GHz
$I=7.5\text{mA}$	$T_0=20^{\circ}\text{C}$	30.9GHz
	$T_0=30^{\circ}\text{C}$	29.6GHz
	$T_0=40^{\circ}\text{C}$	27.3GHz
	$T_0=50^{\circ}\text{C}$	23.5GHz

从上表中，能够总结出下面两天规律：

- (1) 在相同温度下，带宽随着偏置电流的增大而增大；
- (2) 在相同偏置电流下，带宽随着温度的升高而减小。

偏置电流和温度决定了输出功率，该模型的假设下，小信号响应曲线本质上仅仅受到输出功率的影响。

5.7 拓宽带宽的方案

根据之前对器件带宽响应曲线的分析，对器件带宽影响最大的因素仍然是温度和偏置电流。对于温度而言，恒温箱是保持器件处于较低温度的办法之一，另一方法是采用更高效率的降温方式，例如水冷、半导体制冷等等。对于偏置电流而言，7.5mA 并不是能保证器件带宽的最优选择，进一步增大电流即可获得大的带宽，当然也意味着更大的功耗，需要均衡取舍。

另一方面，审视响应模型计算公式：

$$H(f) \equiv \frac{h(f)}{h(0)} = \frac{Z}{(j2\pi f)^2 + (j2\pi f)Y + Z}$$

换算成分贝形式为：

$$20\log|H(\nu)| = 10\log \frac{Z^2}{(\nu^2 - Z)^2 + \nu^2 Y^2}$$

其中 $\nu=2\pi f$ 。

10dB 处的解满足：

$$\begin{aligned} (\nu^2 - Z)^2 + \nu^2 Y^2 &= 10Z^2 \\ \nu^2 &= \left(2Z - Y^2 + \sqrt{(2Z - Y^2)^2 + 36Z^2} \right) / 2 \end{aligned}$$

那么要使这个解尽可能大，只要在增大模型参数中的 Z 的同时而尽可能减小 Y 即可。

对于参数 Y 和 Z ，可以有近似结果：

$$Y = \frac{(G_0/k)P_s}{1+(\varepsilon/k)P_s} \text{ 和 } Z = \frac{Y}{\tau_p}$$

该近似结果的推导会在第六部分中具体讨论。

由于 Y 和 Z 成正相关，因而直接推导模型参数对带宽的影响较为复杂，而且各参数有自己的取值范围，很难说明在这一区间中的影响程度。但可以清楚看到的是， τ_p 对 Z 有更大的影响，而对 Y 的影响可以忽略不计，那么在尽可能保持其他参数不变的同时，尽量减小 τ_p 也即光子寿命，就可以使 Z 大幅增大而 Y 保持不变，从而获得一个更大的器件带宽。

为了验证上述推论，选取 $Y=1 \times 10^{11} \text{s}^{-1}$ ，改变 Z 的取值（ $Z=1.0 \times 10^{22} \text{s}^{-2}$ ， $Z=1.1 \times 10^{22} \text{s}^{-2}$ ， $Z=1.2 \times 10^{22} \text{s}^{-2}$ ），绘制带宽响应曲线进行比较，如下图所示。

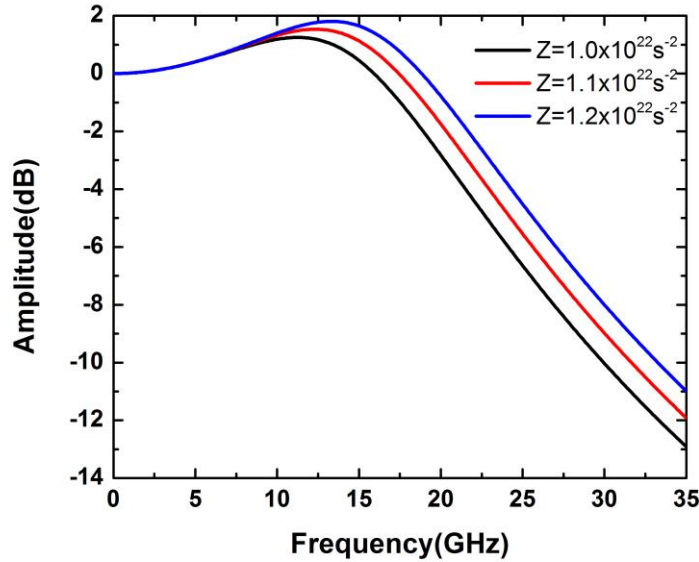


图 19 不同 Z 值下带宽对比图

5.8 平坦 3dB 以内的的带宽曲线

引用之前所推导的分贝形式的响应模型公式：

$$20\log|H(\nu)| = 10\log \frac{Z^2}{(\nu^2 - Z)^2 + \nu^2 Y^2}$$

$$20\log|H(\nu)| = 10\log \frac{Z^2}{\left(\nu^2 - Z + \frac{Y^2}{2}\right)^2 + Y^2 Z - \frac{Y^4}{4}}$$

当 $Z - \frac{Y^2}{2} > 0$ 时, 该曲线事实上是以 $\nu = \sqrt{Z - \frac{Y^2}{2}}$ 为轴的近似对称的曲线, 且在

$\nu = \sqrt{Z - \frac{Y^2}{2}}$ 处取得最大值 $10 \log \frac{Z^2}{Y^2 Z - \frac{Y^4}{4}} > 0$, 也即该处的曲线幅度大于 0 频位置。

考虑 $\nu = \sqrt{Z - \frac{Y^2}{2}}$ 附近一点 ($\nu^2 = Z - \frac{Y^2}{2} + \Delta$) 处的响应曲线斜率, 将响应曲线

对 ν 求导:

$$\frac{\partial (20 \log |H(\nu)|)}{\partial \nu} = -\frac{10}{\ln(10)} \frac{1}{\left(\nu^2 - Z + \frac{Y^2}{2} \right)^2 + Y^2 Z - \frac{Y^4}{4}} \cdot 2 \left(\nu^2 - Z + \frac{Y^2}{2} \right) \cdot 2\nu$$

代入 $\nu^2 = Z - \frac{Y^2}{2} + \Delta$ 得到峰值附近斜率为

$$-\frac{10}{\ln(10)} \frac{1}{\Delta^2 + Y^2 Z - \frac{Y^4}{4}} \cdot 2\Delta \cdot 2\nu$$

由于 Δ 是个极小量, 因此该斜率正比于 $\frac{1}{Y^2 Z - \frac{Y^4}{4}} \cdot 2\sqrt{Z - \frac{Y^2}{2}}$

如果想要让这部分曲线越平坦越好, 就需要这部分曲线斜率尽可能趋向于 0,

亦即 $\frac{1}{Y^2 Z - \frac{Y^4}{4}} \cdot 2\sqrt{Z - \frac{Y^2}{2}}$ 尽可能小。

对于参数 Y 和 Z , 可以有近似结果:

$$Y = \frac{(G_0/k)P_s}{1 + (\varepsilon/k)P_s} \text{ 和 } Z = \frac{Y}{\tau_p}$$

该近似结果的推导会在第六部分中具体讨论。

代入 $Z = \frac{Y}{\tau_p}$, 那么峰值附近曲线斜率可化为 $\frac{1}{Y^2 \sqrt{Y}} \frac{1}{\left(\frac{1}{\tau_p} - \frac{Y}{2} \right) + \frac{Y}{4}} \cdot 2\sqrt{\frac{1}{\tau_p} - \frac{Y}{2}}$

因为 $Z - \frac{Y^2}{2} > 0$, 因此当上式中 Y 尽可能大或 $Z - \frac{Y^2}{2}$ 尽可能趋于 0 时, 该值总

体趋于 0。但由于 $Z - \frac{Y^2}{2}$ 会影响带宽大小，减小 $Z - \frac{Y^2}{2}$ 反而不利于器件性能，因而

最佳设计情况在于保持合理的 $Z - \frac{Y^2}{2}$ 值的同时，尽可能增大 Y 。

为了验证上述推论，选取 $Z = 1.0 \times 10^{22} \text{s}^{-2}$ ，改变 Y 的取值（ $Y = 1.0 \times 10^{11} \text{s}^{-1}$ ， $Y = 1.1 \times 10^{11} \text{s}^{-1}$ ， $Y = 1.2 \times 10^{11} \text{s}^{-1}$ ），绘制带宽响应曲线进行比较，如下图所示。

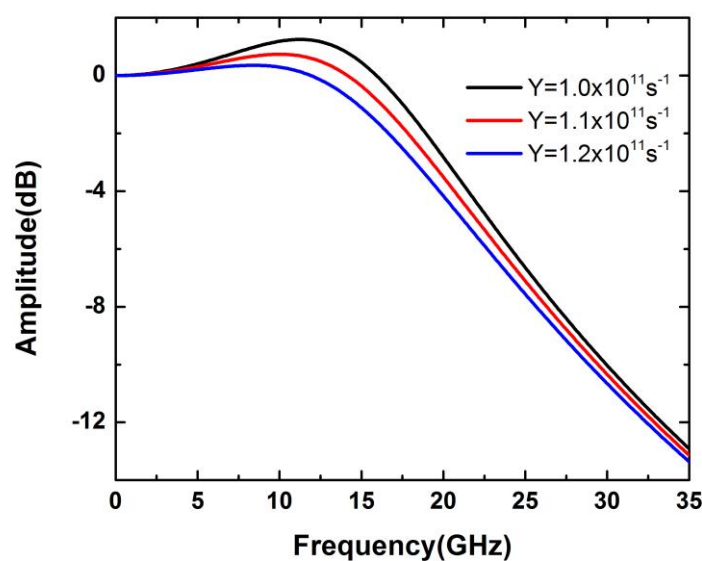


图 20 不同 Y 值下 3dB 内带宽曲线平坦度对比图

由近似计算式 $Y = \frac{(G_0/k)P_s}{1 + (\varepsilon/k)P_s}$ 可知，增大激光器的增益系数、增大激光器工作功率都可以使 Y 增大，亦即使高于 0 频部分的曲线更加平坦。

六、 问题 4： VCSEL 的带宽模型的改进

6.1 问题分析

对于第五部分中所采用的模型，实际上是不完善的，由于各参数之间关联性较大而约束条件的作用有限，很容易得到一个拟合结果取值偏离实际物理意义的情况，另一方面，该模型所采用的速率方程中，也有一些不自洽的地方。

如果直接将问题一中得到的偏置电流和注入激光器的外部驱动电流代入激光器速率方程，得到：

$$\begin{cases} \frac{dN}{dt} = \frac{\eta_i(I - I_{th0} - I_{off}(T))}{q} - \frac{N}{\tau_n} - \frac{G_0(N - N_0)S}{1 + \varepsilon S} \\ \frac{dS}{dt} = -\frac{S}{\tau_p} + \frac{\beta N}{\tau_n} + \frac{G_0(N - N_0)S}{1 + \varepsilon S} \end{cases}$$

VCSEL 输出的光功率与光子数成正比，假定比例因子为 k ，

$$P_o = kS$$

当外部驱动电流保持恒定，载流子数 N 与光子数 S 将在一段时间的瞬态后达到稳定状态，此时 N 和 S 对时间求导为 0（稳态下，载流子和光子数变化为 0），那么速率方程可以写为

$$\begin{aligned} \frac{\eta_i(I - I_{th0} - I_{off}(T))}{q} - \frac{N}{\tau_n} - \frac{G_0(N - N_0)S}{1 + \varepsilon S} &= 0 \\ -\frac{S}{\tau_p} + \frac{\beta N}{\tau_n} + \frac{G_0(N - N_0)S}{1 + \varepsilon S} &= 0 \end{aligned}$$

现在研究在已知输出光功率 P_o 情况下激光器中载流子的浓度问题。将

$P_o = kS$ 和 $P_o = \eta(I - I_{th0} - I_{off}(T))$ 关系式带入上式，此时求出的 N 为稳态下载流子浓度，记为 N_s ，上两式可以变换为

$$\begin{cases} \frac{\eta_i k S}{\eta q} - \frac{N_s}{\tau_n} - \frac{G_0(N_s - N_0)S}{1 + \varepsilon S} = 0 \\ -\frac{S}{\tau_p} + \frac{\beta N_s}{\tau_n} + \frac{G_0(N_s - N_0)S}{1 + \varepsilon S} = 0 \end{cases}$$

可以发现，上式中第三项相同，那么应有

$$\begin{aligned} \frac{G_0(N_s - N_0)S}{1 + \varepsilon S} &= \frac{\eta_i k S}{\eta q} - \frac{N_s}{\tau_n} = \frac{S}{\tau_p} - \frac{\beta N_s}{\tau_n} \\ (\frac{\eta_i k}{\eta q} - \frac{1}{\tau_p})S &= \frac{(1 - \beta)N_s}{\tau_n} \end{aligned}$$

那么 S 与 N_s 成正比，可设

$$S = mN_s, m = \frac{1 - \beta}{\tau_n \left(\frac{\eta_i k}{\eta q} - \frac{1}{\tau_p} \right)}$$

代回稳态方程组中的第二式，并约去一个 N_s 可得

$$\frac{G_0 m (N_s - N_0)}{1 + \varepsilon m N_s} = \frac{m}{\tau_p} - \frac{\beta}{\tau_n}$$

$$(G_0 m \tau_p \tau_n - \varepsilon m^2 \tau_n + \varepsilon m \beta \tau_p) N_s = G_0 m \tau_p \tau_n N_0 + m \tau_n - \beta \tau_p$$

为使上式对不同的稳态工作情况都成立，所有系数应为 0。那么有

$$\begin{cases} G_0 m \tau_p \tau_n - \varepsilon m^2 \tau_n + \varepsilon m \beta \tau_p = 0 \\ G_0 m \tau_p \tau_n N_0 + m \tau_n - \beta \tau_p = 0 \end{cases}$$

$$\begin{cases} G_0 \tau_p \tau_n = \varepsilon m \tau_n - \varepsilon \beta \tau_p \\ G_0 m \tau_p \tau_n N_0 + m \tau_n - \beta \tau_p = 0 \end{cases}$$

将第一式代入第二式将得到

$$(m \tau_n - \beta \tau_p) \varepsilon m N_0 + m \tau_n - \beta \tau_p = 0$$

$$(m \tau_n - \beta \tau_p) (\varepsilon m N_0 + 1) = 0$$

$$m \tau_n = \beta \tau_p$$

这个等式是不可接受的，否则代回方程 $G_0 m \tau_p \tau_n N_0 + m \tau_n - \beta \tau_p = 0$ 将得到

$G_0 m \tau_p \tau_n N_0 = 0$ 的情况，与实际的物理条件矛盾。这一矛盾并非来源于数学推导，

而是因为一开始的假定就出现了错误， $(I - I_{th0} - I_{off}(T))$ 仅能应用于第一、二问题中探讨 VCSEL 性能的宏观经验公式，并不能作为探讨 VCSEL 微观模型时电流中的有效部分，在该模型中，由于电流分布不均、界面损耗等因素，激光器内偏置电流会与之前 L-I 模型中经验公式拟合获得的结果有所区别。因此需要对激光器速率方程模型做出修改。

6.2 模型建立

基于上述讨论的结果，对激光的速率方程中电流项做一定的修改：

$$\begin{cases} \frac{dN}{dt} = \frac{\eta_i(I - I_{th2})}{q} - \frac{N}{\tau_n} - \frac{G_0(N - N_0)S}{1 + \varepsilon S} \\ \frac{dS}{dt} = -\frac{S}{\tau_p} + \frac{\beta N}{\tau_n} + \frac{G_0(N - N_0)S}{1 + \varepsilon S} \end{cases}$$

其中 I_{th2} 是新的激光器内偏置电流修正项，该修正项与问题一中的偏置电流有所区别。

重新计算稳态条件下激光器中载流子浓度 N_s ：

$$\begin{aligned} & \left. \begin{aligned} -\frac{S}{\tau_p} + \frac{\beta N_s}{\tau_n} + \frac{G_0(N_s - N_0)S}{1 + \varepsilon S} &= 0 \\ P_0 &= kS \end{aligned} \right\} \Rightarrow \\ & -\frac{P_0}{k\tau_p} + \frac{\beta N_s}{\tau_n} + \frac{G_0 N_s P_0}{k + \varepsilon P_0} - \frac{G_0 N_0 P_0}{k + \varepsilon P_0} = 0 \\ & \left(\frac{\beta}{\tau_n} + \frac{G_0 P_0}{k + \varepsilon P_0} \right) N_s = \frac{G_0 N_0 P_0}{k + \varepsilon P_0} + \frac{P_0}{k\tau_p} \\ & \frac{\beta(k + \varepsilon P_0) + \tau_n G_0 P_0}{\tau_n(k + \varepsilon P_0)} N_s = \frac{k\tau_p G_0 N_0 P_0 + P_0(k + \varepsilon P_0)}{k\tau_p(k + \varepsilon P_0)} \\ & N_s = \frac{\tau_n}{\beta(k + \varepsilon P_0) + \tau_n G_0 P_0} \cdot \frac{k\tau_p G_0 N_0 P_0 + P_0(k + \varepsilon P_0)}{k\tau_p} \\ & N_s = \frac{G_0 N_0 P_0 + P_0(k + \varepsilon P_0)/(k\tau_p)}{\beta(k + \varepsilon P_0)/\tau_n + G_0 P_0} \\ & N_s = \frac{P_0/(k\tau_p) + G_0 N_0 P_0/(k + \varepsilon P_0)}{\beta/\tau_n + G_0 P_0/(k + \varepsilon P_0)} \\ & N_s - N_0 = \frac{P_0/(k\tau_p) - \beta N_0/\tau_n}{\beta/\tau_n + G_0 P_0/(k + \varepsilon P_0)} \end{aligned}$$

其中， P_0 为通过仪器测量到的激光器的实测出光功率， $(k, \tau_p, \tau_n, G_0, N_0, \varepsilon, \beta)$ 为参数。根据实测光功率和参数值，从上述等式可以计算出激光器中载流子浓度 N_s 。

但由于另一稳态条件修改为

$$\frac{\eta_i(I - I_{th2})}{q} - \frac{N}{\tau_n} - \frac{G_0(N - N_0)S}{1 + \varepsilon S} = 0$$

其中 η_i 和 I_{th2} 是互相关联的未知参数，从现有数据无法拟合得到准确结果，因此无法通过从 N_s 计算得到理论驱动电流和理论出光功率作为拟合的误差约束

条件，需要重新考虑模型中参数的获得方法。

给 VCSEL 加载上小信号后，原来速率方程中的电流、载流子数、光子数则表示为稳态下的值与小信号下引入信号变化的值的和。

$$N(t) = N_s + n(f)e^{j2\pi ft}$$

$$S(t) = S_s + s(f)e^{j2\pi ft}$$

其中， $n(f)$ ， $s(f)$ 足够小。

将加载小信号后的速率方程进行化简：

$$\begin{aligned} \frac{d(N_s + n(f)e^{j2\pi ft})}{dt} &= \frac{\eta_i(I_s + i(f)e^{j2\pi ft} - I_{th0} - I_{off}(T))}{q} - \frac{N_s + n(f)e^{j2\pi ft}}{\tau_n} - \frac{G_0(N_s + n(f)e^{j2\pi ft} - N_0)(S_s + s(f)e^{j2\pi ft})}{1 + \varepsilon(S_s + s(f)e^{j2\pi ft})} \\ (j2\pi f)n(f)e^{j2\pi ft} &= \frac{\eta_i(I_s - I_{th0} - I_{off}(T))}{q} - \frac{N_s}{\tau_n} + \frac{\eta_i i(f)e^{j2\pi ft}}{q} - \frac{n(f)e^{j2\pi ft}}{\tau_n} \\ &\quad - G_0 \cdot \frac{(N_s - N_0)S_s + S_s n(f)e^{j2\pi ft} + (N_s - N_0)s(f)e^{j2\pi ft} + n(f)s(f)e^{j2\pi(2f)t}}{1 + \varepsilon S_s + \varepsilon s(f)e^{j2\pi ft}} \\ &= \frac{\eta_i(I_s - I_{th0} - I_{off}(T))}{q} - \frac{N_s}{\tau_n} - \frac{G_0(N_s - N_0)S_s}{1 + \varepsilon S_s + \varepsilon s(f)e^{j2\pi ft}} \\ &\quad - \frac{G_0 n(f)s(f)e^{j2\pi(2f)t}}{1 + \varepsilon S_s + \varepsilon s(f)e^{j2\pi ft}} \\ &\quad + \frac{\eta_i i(f)e^{j2\pi ft}}{q} - \frac{n(f)e^{j2\pi ft}}{\tau_n} - \frac{G_0 S_s n(f)e^{j2\pi ft} + G_0(N_s - N_0)s(f)e^{j2\pi ft}}{1 + \varepsilon S_s + \varepsilon s(f)e^{j2\pi ft}} \end{aligned}$$

增益压缩因子本身较小，它与小信号的乘积 $\varepsilon s(f)$ 值较小，现考虑在等式中假定

可以忽略其影响，即 $\varepsilon s(f) \rightarrow 0$ ；再考虑等式中第四项，该项中包含的 $e^{j2\pi(2f)t}$ 相比其他项为 2 倍频高次项，高频信号在系统响应中不在考虑范围内，可忽略不考虑。由此，等式可简化为

$$\begin{aligned} (j2\pi f)n(f)e^{j2\pi ft} &= \frac{\eta_i(I_s - I_{th0} - I_{off}(T))}{q} - \frac{N_s}{\tau_n} - \frac{G_0(N_s - N_0)S_s}{1 + \varepsilon S_s} \\ &\quad + \frac{\eta_i i(f)e^{j2\pi ft}}{q} - \frac{n(f)e^{j2\pi ft}}{\tau_n} - \frac{G_0 S_s n(f)e^{j2\pi ft}}{1 + \varepsilon S_s} - \frac{G_0(N_s - N_0)s(f)e^{j2\pi ft}}{1 + \varepsilon S_s} \end{aligned}$$

根据稳态结果，上式的前三项为 0，在略掉因子 $e^{j2\pi ft}$ 后，可简化为

$$(j2\pi f)n(f) = \frac{\eta_i i(f)}{q} - \frac{n(f)}{\tau_n} - \frac{G_0 S_s n(f)}{1 + \varepsilon S_s} - \frac{G_0(N_s - N_0)s(f)}{1 + \varepsilon S_s}$$

用同样的方法，可有

$$(j2\pi f)s(f) = -\frac{s(f)}{\tau_p} + \frac{\beta n(f)}{\tau_n} + \frac{G_0(N_s - N_0)s(f)}{1 + \varepsilon S_s} + \frac{G_0 S_s n(f)}{1 + \varepsilon S_s}$$

综合可得：

$$n(f) = \frac{j2\pi f + 1/\tau_p - G_0(N_s - N_0)/(1 + \varepsilon S_s)}{\beta/\tau_n + G_0 S_s/(1 + \varepsilon S_s)} s(f)$$

根据前面的描述，信号的响应等于对应频率处输出与输入信号的比值，假定小信号响应为线性频率响应 $h(f)$ ，按照定义表示为小信号带来的输出信号 $p(f)$ 与输入信号 $i(f)$ 的比值

$$h(f) = \frac{p(f)}{i(f)} = k \frac{s(f)}{i(f)}$$

$$h(f) = k \frac{s(f)}{i(f)} = \frac{k\eta_i}{q} \cdot \frac{1}{\frac{[j2\pi f + 1/\tau_p - G_0(N_s - N_0)/(1 + \varepsilon S_s)] \cdot [j2\pi f + 1/\tau_n + G_0 S_s/(1 + \varepsilon S_s)]}{\beta/\tau_n + G_0 S_s/(1 + \varepsilon S_s)} + G_0(N_s - N_0)/(1 + \varepsilon S_s)}$$

通常，我们在通信系统里面的信号响应有如下形式

$$h(f) = \frac{X}{(j2\pi f)^2 + (j2\pi f)Y + Z}$$

那么对应有：

$$Y = \frac{1}{\tau_p} + \frac{1}{\tau_n} + \frac{G_0 S_s}{1 + \varepsilon S_s} - \frac{G_0(N_s - N_0)}{1 + \varepsilon S_s}$$

$$Z = \left[\frac{1}{\tau_p} - \frac{G_0(N_s - N_0)}{1 + \varepsilon S_s} \right] \cdot \left[\frac{1}{\tau_n} + \frac{G_0 S_s}{1 + \varepsilon S_s} \right] + \left[\frac{\beta}{\tau_n} + \frac{G_0 S_s}{1 + \varepsilon S_s} \right] \cdot \frac{G_0(N_s - N_0)}{1 + \varepsilon S_s}$$

$$= \frac{1}{\tau_p \tau_n} + \frac{G_0 S_s}{\tau_p (1 + \varepsilon S_s)} - \frac{(1 - \beta)G_0(N_s - N_0)}{\tau_n (1 + \varepsilon S_s)}$$

将 $P_s = kS_s$ 带入上述两个式子，将稳态下的光子数 S_s 用光功率 P_s 表示

$$Y = \frac{1}{\tau_p} + \frac{1}{\tau_n} + \frac{G_0 P_s}{k + \varepsilon P_s} - \frac{kG_0(N_s - N_0)}{k + \varepsilon P_s}$$

$$Z = \frac{1}{\tau_p \tau_n} + \frac{G_0 P_s}{\tau_p (k + \varepsilon P_s)} - \frac{k(1 - \beta)G_0(N_s - N_0)}{\tau_n (k + \varepsilon P_s)}$$

为了进一步提高精度，假定只忽略高阶项，而不忽略 $\varepsilon s(f)$ 项，那么

$$\frac{1}{1 + \varepsilon S_s + \varepsilon s(f)} = \frac{1}{1 + \varepsilon S_s + \varepsilon s(f)} \cdot \frac{1 + \varepsilon S_s - \varepsilon s(f)}{1 + \varepsilon S_s - \varepsilon s(f)} = \frac{1 + \varepsilon S_s - \varepsilon s(f)}{(1 + \varepsilon S_s)^2 - (\varepsilon s(f))^2}$$

而其中 $(\varepsilon s(f))^2$ 是高阶项仍然可以忽略。

将 $\frac{1}{1+\varepsilon S_s+\varepsilon s(f)} = \frac{1+\varepsilon S_s-\varepsilon s(f)}{(1+\varepsilon S_s)^2}$ 应用于参数计算过程可以得到：

$$Y = \frac{1}{\tau_p} + \frac{1}{\tau_n} + \frac{G_0 P_s}{k + \varepsilon P_s} - \frac{G_0 (N_s - N_0)}{(1 + \varepsilon P_s / k)^2}$$

$$Z = \frac{1}{\tau_p \tau_n} + \frac{G_0 P_s}{\tau_p (k + \varepsilon P_s)} - \frac{(1 - \beta) G_0 (N_s - N_0)}{\tau_n (1 + \varepsilon P_s / k)^2}$$

最后，响应函数需要进行归一化处理，因此不必关心刚才等式里面的 X 表达式是什么，在归一化处理时 X 被抵消掉。归一化小信号响应为

$$H(f) \equiv \frac{h(f)}{h(0)} = \frac{Z}{(j2\pi f)^2 + (j2\pi f)Y + Z}$$

将上述的 Y 和 Z 带入上式，即可获得 VCSEL 的小信号响应数学表达式。对于表达式：

$$Y = \frac{1}{\tau_p} + \frac{1}{\tau_n} + \frac{G_0 P_s}{k + \varepsilon P_s} - \frac{k G_0 (N_s - N_0)}{k + \varepsilon P_s}$$

$$Z = \frac{1}{\tau_p \tau_n} + \frac{G_0 P_s}{\tau_p (k + \varepsilon P_s)} - \frac{k(1 - \beta) G_0 (N_s - N_0)}{\tau_n (k + \varepsilon P_s)}$$

将 $N_s - N_0$ 的值代入 Y 的表达式可得：

$$Y = \frac{1}{\tau_p} + \frac{1}{\tau_n} + \frac{G_0 P_s}{k + \varepsilon P_s} - \frac{k G_0}{k + \varepsilon P_s} \cdot \frac{P_s / (k \tau_p) - \beta N_0 / \tau_n}{\beta / \tau_n + G_0 P_0 / (k + \varepsilon P_s)}$$

$$Y = \frac{1}{\tau_p} + \frac{1}{\tau_n} + \frac{G_0 P_s}{k + \varepsilon P_s} - \frac{G_0 P_s / \tau_p - \beta k G_0 N_0 / \tau_n}{\beta (k + \varepsilon P_s) / \tau_n + G_0 P_s}$$

$$Y = \frac{G_0 P_s}{k + \varepsilon P_s} + \frac{1}{\tau_n} + \frac{(k + \varepsilon P_s) / (\tau_n \tau_p) - k G_0 N_0 / \tau_n}{\beta (k + \varepsilon P_s) / \tau_n + G_0 P_s} \beta$$

其中 $\frac{1}{\tau_n}$ 与 $\frac{1}{\tau_p}$ 相比是一个很小的量，虽然 $\frac{(k + \varepsilon P_s) / (\tau_n \tau_p) - k G_0 N_0 / \tau_n}{\beta (k + \varepsilon P_s) / \tau_n + G_0 P_s}$ 与

$\frac{G_0 P_s}{k + \varepsilon P_s}$ 量级相近，但由于 β 是个小量，因此相乘以后也可以忽略不计。

因而可以得到一个近似公式 $Y = \frac{G_0 P_s}{k + \varepsilon P_s} = \frac{(G_0 / k) P_s}{1 + (\varepsilon / k) P_s}$ ，近似所带来的误差小

于 1%。

同理将 $N_s - N_0$ 的值代入 Z 的表达式近似可得：
$$Z = \frac{G_0 P_s}{\tau_p (k + \varepsilon P_s)} = \frac{Y}{\tau_p}$$

虽然通过上述表达式无法获得除 $\tau_p, G_0/k, \varepsilon/k$ 以外的任何参数值 $(k, \tau_n, G_0, N_0, \varepsilon, \beta)$ ，但并不影响其他功率下响应曲线的获得。

而对于更加精确的参数表达式：

$$Y = \frac{1}{\tau_p} + \frac{1}{\tau_n} + \frac{G_0 P_s}{k + \varepsilon P_s} - \frac{G_0 (N_s - N_0)}{(1 + \varepsilon P_s / k)^2}$$

$$Z = \frac{1}{\tau_p \tau_n} + \frac{G_0 P_s}{\tau_p (k + \varepsilon P_s)} - \frac{(1 - \beta) G_0 (N_s - N_0)}{\tau_n (1 + \varepsilon P_s / k)^2}$$

同样将 $N_s - N_0$ 代入可得

$$Y = \frac{G_0 P_s}{k + \varepsilon P_s} - \frac{\varepsilon P_s}{k + \varepsilon P_s} \frac{1}{\tau_p}$$

$$Z = \frac{G_0 P_s}{\tau_p (k + \varepsilon P_s)} - \frac{\varepsilon P_s}{k + \varepsilon P_s} \frac{1}{\tau_p \tau_n}$$

上述结果与之前的表达式相比仅仅是相差了与 $\frac{\varepsilon P_s}{k + \varepsilon P_s}$ 相乘的两个小项，总体

上是一致的，也证明了推导无误。

由于高阶近似结果相差不大，可直接采用：

$$Y = \frac{(G_0 / k) P_s}{1 + (\varepsilon / k) P_s}, \quad Z = \frac{Y}{\tau_p}$$

6.3 模型求解

该模型需由 PSO 算法确立 Y 与 Z 的值，此处与原模型相同，可直接取原模型确立的 Y 和 Z 的参数。

之前的拟合已得在 $I_b = 7.5\text{mA}$, $T_a = 20^\circ\text{C}$ 条件下， $P = 1.904\text{mW}$ ，幅频响应参

数 $Y = 1.38 \times 10^{11} \text{s}^{-1}$ ， $Z = 1.18 \times 10^{22} \text{s}^{-2}$ ，代入近似表达式 $Y = \frac{(G_0 / k) P_s}{1 + (\varepsilon / k) P_s}$ 和 $Z = \frac{Y}{\tau_p}$

得到 $G_0 / k = 7.28 \times 10^3 \text{W}^{-1} \text{s}^{-1}$, $\varepsilon / k = 2.35 \text{W}^{-1}$, $\tau_p = 1.17 \times 10^{-11} \text{s}$ 。其中 $\varepsilon / k = 2.35 \text{W}^{-1}$

可能有一定误差，但是由于 $(\varepsilon / k) P$ 是一个小量，所以并不会对最终模型带来很大影响，误差仍然在 1% 以内。

在此模型参数下，保持环境温度不变，始终为 20°C ，改变偏置电流分别为

4.5mA, 6mA, 7.5mA, 9mA, 作出相同温度不同偏置电流下的带宽响应曲线如图 21; 保持偏置电流 7.5mA 不变, 改变环境温度分别为 20°C, 30°C, 40°C, 50°C, 作出相同偏置电流不同温度下的带宽响应曲线如图 22。

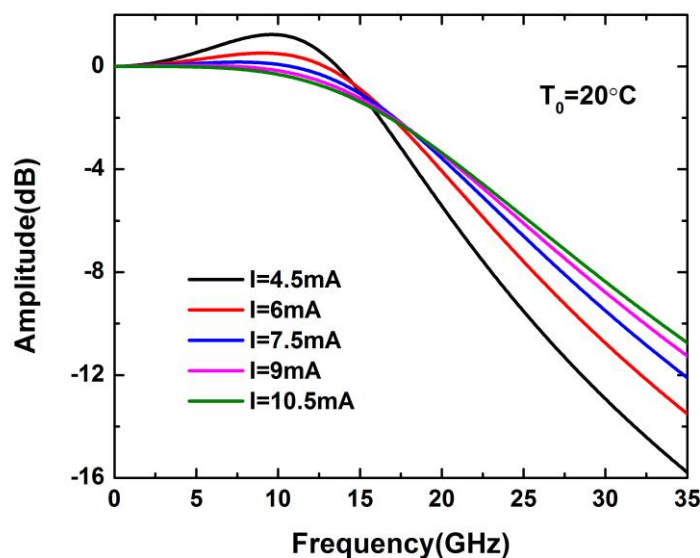


图 21 温度为 20°C 不同偏置电流带宽曲线

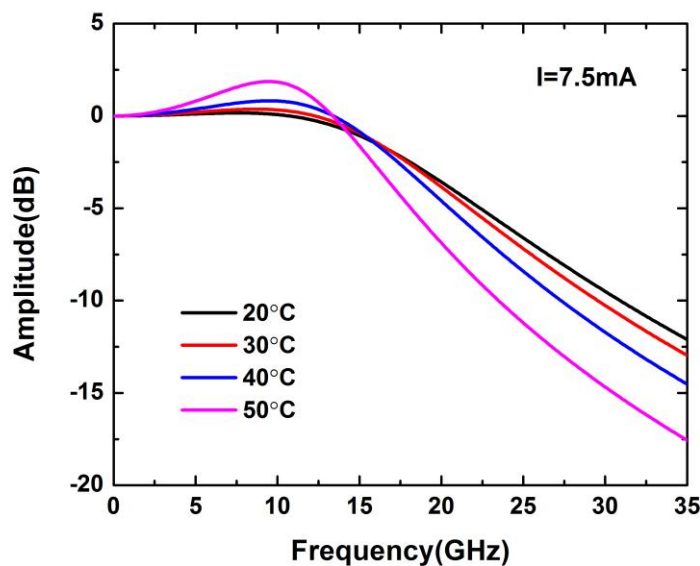


图 22 偏置电流 7.5mA 不同温度带宽曲线

6.4 模型分析

改进模型中, 在模型求解的第二部分, 完全不再需要使用多个误差约束条件进行参数拟合, 仅仅从一条带宽响应曲线拟合得到的 Y 和 Z 就可以直接推算得到新模型的所有参数, 且整体误差控制在 1% 以内。

为了将改进模型与原模型进行对比, 选取了偏置电流为 7.5mA 下两个不同温

度拟合曲线的新旧模型推算的响应曲线作为对比，如下图所示。

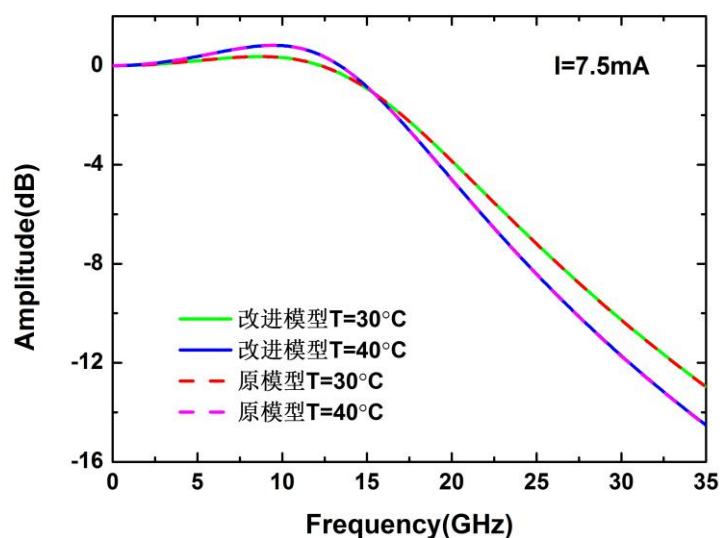


图 23 改进模型前后 $I=7.5\text{mA}$, $T_0=30^\circ\text{C}$ 和 40°C 带宽曲线

可见改进模型准确性非常高，而在模型计算的工作量上，改进模型体现出了非常大的优势。

相较原模型，改进模型的劣势在于无法获得大部分参数的具体值，但由于原模型的理论基础并不充分，实际上在拟合过程中反而可能给模型参数带来较大误差，拟合得到的参数值可能是赝值。相比之下，改进模型虽然得到不到大部分具体参数，但是维持了模型整体的精确性，并且节省了计算量，可能很快推导出其他条件下的频率响应曲线。

七、 参考文献

- [1] 郑世奇, 基于速率方程的垂直腔面发射激光器数值模拟与特性研究,北京邮电大学, 2006;
- [2] J Gao, X Li, J Flucke, G Boeck, Direct parameter-extraction method for laser diode rate-equation model, *Lightwave Technology Journal of*, 22 (6):1604-1609, 2004;
- [3] Carlisle A, Dozier G. An off-the-shelf PSO[C]// The Workshop on Particle Swarm Optimization. 2001;
- [4] 卢明,刘明,基于遗传算法的水文地质参数推求,《科技传播》,(4):168, 2011
- [5] Mena P V, Morikuni J J, Kang S M, et al. A Comprehensive Circuit-Level Model of Vertical-Cavity Surface-Emitting Lasers[J]. *Journal of Lightwave Technology*, 17(12):2612-2632, 2002.
- [6] 王丽荣, 基于小波变换的目标检测方法研究,《吉林大学》, 2006
- [7] Hai-Yana L I, Jing Y W, Ming-Xua M A, et al. Collaborative optimization algorithm based on dynamic penalty function method[J]. *Control & Decision*, 2009, 24(6):911-868.
- [8] Ferrari S, Stengel R F. Smooth function approximation using neural networks[J]. *IEEE Transactions on Neural Networks*, 2005, 16(1):24.
- [9] Cottle R W, \dagger S G D, Zikan K. A lagrangean relaxation algorithm for the constrained matrix problem[J]. *Naval Research Logistics Quarterly*, 33(1):55–76, 1986,

附录（文中所用程序）：

第一题：

1

N=12;

```
L=[0,-1*10^-2,-2.4920*10^-2,-5.09*10^-5,0,-5.0620*10^-9,-  
2.044*10^-10,100,-2.4920*10^-14,-5.09*10^-2,-5.09*10^-4,-  
2.4920*10^-15];
```

```
H=[1,1*10^-2, 2.4920*10^-2,5.09*10^-4,  
5.816*10^-5,5.0620*10^-9,2.044*10^-10,2000,2.4920*10^-14,  
5.09*10^-2,5.09*10^-4,2.4920*10^-15];
```

```
%L=[0.5,0.3*10^-3,1.246*10^-3,-2.545*10^(-6),2.908*10^(-7)  
, -2.531*10^(-10),1.022*10^(-12),2600];
```

```
%H=[0.5,0.3*10^-3,1.246*10^-3,-2.545*10^(-6),2.908*10^(-7)  
, -2.531*10^(-10),1.022*10^(-12),2600];
```

```
%L=[0,-1,-1,-1,-1,-1,-1,-1,200];
```

```
%H=[1,1,1,1,1,1,1,1,5000];
```

```
[X,Fx,StopMessage] = pso(@L_I,N,L,H);
```

2

```
function [X,Fx,StopMessage] =  
pso(ObjectiveFunction,Nvars,LB,UB)
```

```
%% Default Parameters
```

```
%C1 = 00.082;
```

```
%C2 = 0000.52;
```

```
%C1 = 0000001.2;
```

```
%C2 = 0000.002;
```

```
C1 = 01.2;
```

```
C2 = 00.52;
```

```
MaxIterations = 1000;
```

```
N =1000;
```

```
W = 0.0005;
```

```
%W = 0.0004;
```

```
StopMessage = 'Stopping PSO : Reached Maximum Number of  
Iterations';
```

```
%% Check Input
```

```
if nargin < 4
```

```
    error('PSO : Not Enough Input Arguments');
```

```
end
```

```
if ~isequal(Nvars,size(LB,2),size(UB,2))
```

```
    error('PSO : Invalid Arguments :
```

```
isequal(Nvars,size(LB,2),size(UB,2)) should be true ');
```

```
end
```

```
%% Initial Position and Velocity
```



```

CurrentPosition = zeros(N,Nvars); % Initial Position
for i = 1:Nvars
    CurrentPosition(:,i) = random('unif',LB(i),UB(i),N,1);
end
Velocity = W.*rand(N,Nvars) ; % Initial Velocity
%% Evalute Initial Position
CurrentFitness = zeros(N,1); % Fitness Value
for i = 1:N
    CurrentFitness(i) =
ObjectiveFunction(CurrentPosition(i,:));
end
%% Update Local Best
LocalBestPosition = CurrentPosition; % Local Best
LocalBestFitness = CurrentFitness;
%% Update Global Best
[GlobalBestFitness,index] = min(LocalBestFitness);
GlobalBestPosition =
repmat(LocalBestPosition(index,:),N,1); % Global Best
%% Update Velocity and Position
R1 = rand(N,Nvars); % random Number 1
R2 = rand(N,Nvars); % random Number 2
Velocity = W.*Velocity +
C1.*(R1.*(LocalBestPosition-CurrentPosition))...
+ C2.*(R2.*(GlobalBestPosition-CurrentPosition));
CurrentPosition = CurrentPosition + Velocity ;
%% Bound Data LB UB
for i = 1:Nvars
    indexes = CurrentPosition(:,i) < LB(i).*ones(N,1);
    CurrentPosition(indexes,i) = LB(i);
    indexes = CurrentPosition(:,i) > UB(i).*ones(N,1);
    CurrentPosition(indexes,i) = UB(i);
end
%% Iterate to Achive Best
Iter = 0;
while (Iter < MaxIterations)
    Iter = Iter+1;
    %%Evalute Current Position
    for i = 1:N
        CurrentFitness(i) =
ObjectiveFunction(CurrentPosition(i,:));
    end
    %% Update Local Best
    indexes = find(CurrentFitness < LocalBestFitness);

```

```

        LocalBestFitness(indexes) = CurrentFitness(indexes);
        LocalBestPosition(indexes,:) =
CurrentPosition(indexes,:);
        %% Update Global Best
        [GlobalBestFitnessNew,index] = min(LocalBestFitness);
        if GlobalBestFitnessNew < GlobalBestFitness
            GlobalBestFitness = GlobalBestFitnessNew;
            GlobalBestPosition =
repmat(LocalBestPosition(index,:),N,1);
            MM=GlobalBestPosition(1,:);
        end
        %% Update Velocity and Position
        R1 = randn(N,Nvars); % random Number 1
        R2 = randn(N,Nvars); % random Number 2
        Velocity = W.*Velocity +
C1.*(R1.*(LocalBestPosition-CurrentPosition))...
        + C2.*(R2.*(GlobalBestPosition-CurrentPosition));
        CurrentPosition = CurrentPosition + Velocity ;
        %% Bound Data LB UB
        for i = 1:Nvars
            indexes = CurrentPosition(:,i) < LB(i).*ones(N,1);
            CurrentPosition(indexes,i) = LB(i);
            indexes = CurrentPosition(:,i) > UB(i).*ones(N,1);
            CurrentPosition(indexes,i) = UB(i);
        end
        %% Print Result
        M = mean(LocalBestFitness);
        if (Iter==1) || (rem(Iter,100)==0)
            fprintf('Iteration-%5.0d | Best F(x) = %d | Mean F(x)
= %d\n',Iter,GlobalBestFitness,M);
        end
    end
end
%% Output
X = GlobalBestPosition(1,:)
Fx = GlobalBestFitness
end

```

```

3.
%eta*(I-I_th0-a0-a1*(T0+(I*V-P0)*R_th)-a2*(T0+(I*V-P0)*R_th)^2-a3*(T0
+(I*V-P0)*R_th)^3-a4*(T0+(I*V-P0)*R_th)^4)-P0=0;
%eta x1
%I_th0 x2
%a0 x3
%a1 x4

```

```

%a2 x5
%a3 x6
%a4 x7
%R_th x8
%L=[0,0,0,-5.09*10^-5,0,-5.0620*10^-10,0,0];
%H=[1,0.6*10^-3, 2.4920*10^-3,0, 5.816*10^-7,0,2.044*10^-12,5.2*10^3];
syms P1
T0=0;
X=[0.381966011,-0.000708204,-0.005882814,-1.20E-05,2.22E-06,1.62E-09,
-4.83E-12,2909.830056];
for n=1:9
    T0=T0+10;
    for k=1:1401
        % (X(1)*(I(k)/1000-X(2)-X(3)-X(4)*(T0+(I(k)/1000*U(k)-P(k)/1000)*X(8))
        -X(5)*(T0+(I(k)/1000*U(k)-P(k)/1000)*X(8))^2-X(6)*(T0+(I(k)/1000*U(k)
        -P(k)/1000)*X(8))^3-X(7)*(T0+(I(k)/1000*U(k)-P(k)/1000)*X(8))^4)-P(k)
        /1000
        x(n,k)=
        fzero(@(P1)X(1)*(I(k)/1000-X(2)-X(3)-X(4)*(T0+(I(k)/1000*U(k)-P1/1000
        ) *X(8))-X(5)*(T0+(I(k)/1000*U(k)-P1/1000)*X(8)).^2-X(6)*(T0+(I(k)/100
        0*U(k)-P1/1000)*X(8)).^3-X(7)*(T0+(I(k)/1000*U(k)-P1/1000)*X(8)).^4)-
        P1/1000,0.003);
    end
    plot(I,x);
    hold on;
end
    %plot(I,P,'*');hold on;
axis([0,14,0,4])

```

```

4.
X=[0.289829968,-0.002135034,0.003073175,-4.53E-05,6.95E-07,4.59E-09,-
3.77E-12,1798.101195];

```

```

a=0;
for k=1000:1300
    for T0=30:0.02:31
        x=
        fzero(@(P1)X(1)*(I(k)/1000-X(2)-X(3)-X(4)*(T0+(I(k)/1000*U(k)-P1/1000
        ) *X(8))-X(5)*(T0+(I(k)/1000*U(k)-P1/1000)*X(8)).^2-X(6)*(T0+(I(k)/100
        0*U(k)-P1/1000)*X(8)).^3-X(7)*(T0+(I(k)/1000*U(k)-P1/1000)*X(8)).^4)-
        P1/1000,0.002);
        if x>=2;
            if T0>a
                a=T0;
            end
        end
    end
end

```

```

        end
    end
    end
end

```

第三题

1.

```

function f1=obj(x)
q=1.6*10^-19;
P0=1.904172/1000;
%load('S21_5')
Ns=(P0/(x(4)*x(7))+x(5)*x(6)*P0/(x(4)+x(8)*P0))/(x(2)/x(3)
)+x(5)*P0/(x(4)+x(8)*P0));
Ps=1.904172/1000;
%Ps=x(4)*(x(1)*(I_cur-I_th-I_off)/q-Ns/x(3))/(x(5)*(Ns-x(
6)))
Y=1/x(7)+1/x(3)+x(5)*Ps/(x(4)+x(8)*Ps)-x(5)*(Ns-x(6))/(1+
x(8)*Ps/x(4))^2
Z=1/(x(7)*x(3))+x(5)*Ps/(x(7)*(x(4)+x(8)*Ps))-(1-x(2))*x(
5)*(Ns-x(6))/(x(3)*(1+x(8)*Ps/x(4))^2)
%f1=sum(20*log(abs(Z./((2i*pi*f).^2+(2i*pi*f)*Y+Z)))-S21)
^2;
u=1;
f1=u*(log(4+exp((Y-2.064017391930334e+11)/2.0640173919303
34e+11)+exp(-(Z-2.273316831356820e+22)/2.273316831356820e
+22)+exp((Z-2.273316831356820e+22)/2.273316831356820e+22)
+exp(-(Y-2.064017391930334e+11)/2.064017391930334e+11)));
%f1=(log((Y-137659035280.771)^2))^2+abs(log(((Z-1.1811542
86486651e+22))^2))^2;
%f1=((Y-137659035280.771)^4)+(((Z-1.181154286486651e+22
)^4*10^(-40)*5));

end

```

2.

```

function [ h1 , h2 ]=constrains(x)
I_cur=7.5/1000;
%P0=1.904172;
q=1.6*10^-19;
I_th=3.41*10^-04;
I_off=0;
P0=1.904172/1000;

```

```

Ns=(P0/(x(4)*x(7))+x(5)*x(6)*P0/(x(4)+x(8)*P0))/(x(2)/x(3)
)+x(5)*P0/(x(4)+x(8)*P0));
%x(4)*(x(1)*(I_cur-I_th-I_off)/q-Ns/x(3))/(x(5)*(Ns-x(6))
);
h1=x(4)*(x(1)*(I_cur-I_th-I_off)/q-Ns/x(3))/(x(5)*(Ns-x(6)
)))-1.904172/1000;
h1=h1^2*10^20;
h2=q/x(1)*(Ns/x(3)+x(5)*(Ns-x(6))*P0/(x(4)+x(8)*P0))-7.5/
1000;
h2=h2^2*10^16;
end

```

3.

```

function [X,FVAL]=AL_main(x_al,r_al,N_equ)
global r_al pena N_equ ;
pena=10;
c_scale=2;
cta=0.5;
e_al=0.005;
e_al2=5;
max_itera=2;
out_itera=1;
%=====
while out_itera<max_itera
    x_al0=x_al;
    r_al0=r_al;

    compareFlag=compare(x_al0);

    N=8;

    L=[0.3,1*10^-9,9.6*10^-11,1.5*10^-12,1.8*10^2,4.97*10^1,3
.8*10^-14,4.7*10^-12];

    H=[1,1*10^-1,9.6*10^-7,1.5*10^-4,1.8*10^10,4.97*10^9,3.8*
10^-10,4.7*10^-4];

    % [X,FVAL]=fminunc(@AL_obj,x_al0);
    % [X,FVAL]=fmincon(@AL_obj,x_al0,[],[],[],[],[0.3,1*10^
-9,9.6*10^-11,1.5*10^-12,1.8*10^2,4.97*10^1,3.8*10^-14,4.
7*10^-12],[1,1*10^-1,9.6*10^-7,1.5*10^-4,1.8*10^10,4.97*1
0^9,3.8*10^-10,4.7*10^-4]);
    [X,Fx,StopMessage] = pso(@AL_obj,N,L,H);
    x_al=X;
    %μÃµµÐÃµü´úµã
    %ÃÐ¶İİİÖ¹İö¼p
    if compare(x_al)<e_al && obj(x_al)<e_al2

```

```

        disp('we get the opt point');
        break
    end

    if compare(x_al)<cta*compareFlag
        pena=pena;
    else
        pena=min(1000,c_scale*pena);
        % disp('pena=2*pena');
    end

    [h1,h2]=constrains(x_al);
    %for i=1:N_equ

        r_al(1)=r_al(1)+pena*h1;
        r_al(2)=r_al(2)+pena*h2;
    %end
    %for i=1:N_inequ

        %r_al(i+N_equ)=max(0,(r_al(i+N_equ)+pena*g(i)));
    %end
    out_itera=out_itera+1;
end
disp('!!!!!!!!!!!!!!!!!!!!!!!!the iteration
over!!!!!!!!!!!!!!!!!!!!!!!!');
disp('the value of the obj function');
obj(x_al)
disp('the value of constrains');
compare(x_al)
disp('the opt point');
X=x_al;
FVAL=obj(X);
4.
function f=AL_obj(X)
global r_al pena N_equ ;
h_equ=0;

[h1,h2]=constrains(X);
%for i=1:N_equ

h_equ=h_equ+h2*r_al(2)+(pena/2)*h2.^2+h1*r_al(1)+(pena/2)
*h1.^2;
%end

```

```
f=obj(X)+h_equ;
```

5.

```
function f=compare(X)
global r_al pena N_equ ;
h_equ=0;

[h1,h2]=constrains(X);
%for i=1:N_equ
    h_equ=h_equ+h2.^2+h1.^2;
%end

f=sqrt(h_equ);
```

6.

```
function f1=obj1(X1)
load('S21_5.mat')
load('xd2')
%f=f(200:2300);
%xd2=xd2(200:2300);
f1=sum((20*log10(abs(X1(2)./(0*(2i*pi*f).^3+(2i*pi*f).^2*
X1(3)+(2i*pi*f)*X1(1)+X1(2))))-(xd2)).^2);
%5.867875648000000e-01
end
```

7.

```
function [X,Fx,StopMessage] =
pso(ObjectiveFunction,Nvars,LB,UB)
%% Default Parameters
%C1 = 00.082;
%C2 = 0000.52;
%C1 = 0000001.2;
%C2 = 0000.002;
C1 = 01.2;
C2 = 00.52;
MaxIterations = 1000;
N =1000;
W = 0.0005;
%W = 0.0004;
StopMessage = 'Stopping PSO : Reached Maximum Number of
Iterations';
%% Check Input
if nargin < 4
    error('PSO : Not Enough Input Arguments');
end
```

```

if ~isequal(Nvars,size(LB,2),size(UB,2))
    error('PSO : Invalid Arguments :
isequal(Nvars,size(LB,2),size(UB,2)) should be true ');
end
%% Initial Position and Velocity

CurrentPosition = zeros(N,Nvars); % Initial Position
for i = 1:Nvars
    CurrentPosition(:,i) = random('unif',LB(i),UB(i),N,1);
end
Velocity = W.*rand(N,Nvars) ; % Initial Velocity
%% Evaluate Initial Position
CurrentFitness = zeros(N,1); % Fitness Value
for i = 1:N
    CurrentFitness(i) =
ObjectiveFunction(CurrentPosition(i,:));
end
%% Update Local Best
LocalBestPosition = CurrentPosition; % Local Best
LocalBestFitness = CurrentFitness;
%% Update Global Best
[GlobalBestFitness,index] = min(LocalBestFitness);
GlobalBestPosition =
 repmat(LocalBestPosition(index,:),N,1); % Global Best
%% Update Velocity and Position
R1 = rand(N,Nvars); % random Number 1
R2 = rand(N,Nvars); % random Number 2
Velocity = W.*Velocity +
C1.*(R1.*(LocalBestPosition-CurrentPosition))...
    + C2.*(R2.*(GlobalBestPosition-CurrentPosition));
CurrentPosition = CurrentPosition + Velocity ;
%% Bound Data LB UB
for i = 1:Nvars
    indexes = CurrentPosition(:,i) < LB(i).*ones(N,1);
    CurrentPosition(indexes,i) = LB(i);
    indexes = CurrentPosition(:,i) > UB(i).*ones(N,1);
    CurrentPosition(indexes,i) = UB(i);
end
%% Iterate to Achive Best
Iter = 0;
while (Iter < MaxIterations)
    Iter = Iter+1;
    %%Evaluate Current Position
    for i = 1:N

```



```

        CurrentFitness(i) =
ObjectiveFunction(CurrentPosition(i,:));
    end
    %% Update Local Best
    indexes = find(CurrentFitness < LocalBestFitness);
    LocalBestFitness(indexes) = CurrentFitness(indexes);
    LocalBestPosition(indexes,:) =
CurrentPosition(indexes,:);
    %% Update Global Best
    [GlobalBestFitnessNew,index] = min(LocalBestFitness);
    if GlobalBestFitnessNew < GlobalBestFitness
        GlobalBestFitness = GlobalBestFitnessNew;
        GlobalBestPosition =
repmat(LocalBestPosition(index,:),N,1);
        MM=GlobalBestPosition(1,:);
    end
    %% Update Velocity and Position
    R1 = randn(N,Nvars); % random Number 1
    R2 = randn(N,Nvars); % random Number 2
    Velocity = W.*Velocity +
C1.*(R1.*(LocalBestPosition-CurrentPosition))...
    + C2.*(R2.*(GlobalBestPosition-CurrentPosition));
    CurrentPosition = CurrentPosition + Velocity ;
    %% Bound Data LB UB
    for i = 1:Nvars
        indexes = CurrentPosition(:,i) < LB(i).*ones(N,1);
        CurrentPosition(indexes,i) = LB(i);
        indexes = CurrentPosition(:,i) > UB(i).*ones(N,1);
        CurrentPosition(indexes,i) = UB(i);
    end
    %% Print Result
    M = mean(LocalBestFitness);
    if (Iter==1) || (rem(Iter,100)==0)
        fprintf('Iteration-%5.0d | Best F(x) = %d | Mean F(x)
= %d\n',Iter,GlobalBestFitness,M);
    end
end
%% Output
X = GlobalBestPosition(1,:)
Fx = GlobalBestFitness
end
8.
function f1=obj1(X1)
load('S21_5.mat')

```

```

load('xd2')
%f=f(200:2300);
%xd2=xd2(200:2300);
f1=sum((20*log10(abs(X1(2)./(0*(2i*pi*f).^3+(2i*pi*f).^2*
X1(3)+(2i*pi*f)*X1(1)+X1(2))))-(xd2)).^2);
%5.867875648000000e-01
end
9.
r_al=[1,1];      %³öÊ¼À-,ñÀÊÈÕ³Ë×Ó
x_al=[0.3000000000000000,1.000000000000000e-09,3.0095066353
7008e-07,0.000101997974596610,7381246949.64758,4231719817
.44920,1.16675401489621e-11,1.49901154542345e-09];
%x_al=[0.999999999976693,2.29151117295465e-07,1.600368061
82917e-10,9.53039531779461e-08,4032587.69686457,38745.653
0726966,3.79999999999989e-10,4.70082151820282e-10];
N_equ=2;
%N_inequ=1;
[X,FVAL]=AL_main(x_al,r_al,N_equ);

8.
%T0=20;I=10.5/1000;P0=2.375074221/1000;U=2.680639134/1000;
%T0=20;I=4.5/1000;P0=1.182955919/1000;U=2.225163645/1000;
%T0=20;I=6/1000;P0=1.568837/1000;U=2.355/1000;
%T0=20;I=9/1000;P0=2.173222/1000;U=2.575589/1000;
%T0=20;I=7.5/1000;P0=1.904172/1000;U=2.464597008/1000;
%T0=30;I=7.5/1000;P0=1.686205611/1000;U=2.464597008/1000;
%T0=40;I=7.5/1000;P0=1.378409/1000;U=2.464597008/1000;
T0=50;I=7.5/1000;P0=0.961602/1000;U=2.464597008/1000;

%etai x(1) 0.7
%beta x(2) 1E-5
%tn x(3) 9.6E-9
%k x(4) 1.5E-8
%G0 x(5) 1.8E6
%N0 x(6) 4.97E5
%tp x(7) 3.8E-12
%epsilon x(8) 4.7E-8

R_th=1790;
T=T0+(I*U-P0)*R_th;

I_tho=-0.00214;a0=0.003073;a1=-4.53E-05;a2=-4.53E-05;a3=4.59E-09;a4=-
3.77E-12;
I_th=I_tho+a0+a1*T+a2*T^2+a3*T^3+a4*T^4;

```

```

%x=[0.333639261,1.00E-09,9.60E-07,0.000149556,11226285.83,4970000000,
1.08E-11,9.25E-06];
x=[0.3,1.00E-09,3.01E-07,0.000101998,7381246950,4231719817,1.17E-11,1
.50E-09];

%x=[0.7,1*10^-5,9.6*10^-9,1.5*10^-8,1.8*10^6,4.97*10^5,3.8*10^-12,4.7
*10^-8] %cankaozhi

f=0:10^8:35*10^9;
q=1.6*10^-19;
I_th=3.41*10^-04;
I_off=0;
Ns=(P0/(x(4)*x(7))+x(5)*x(6)*P0/(x(4)+x(8)*P0))/(x(2)/x(3)+x(5)*P0/(x
(4)+x(8)*P0));
I_cur=q/x(1)*(Ns/x(3)+x(5)*(Ns-x(6))*P0/(x(4)+x(8)*P0))+I_th;
Ps=x(4)*(x(1)*(I_cur-I_th-I_off)/q-Ns/x(3))/(x(5)*(Ns-x(6)));
%Ps=P0;
Y=1/x(7)+1/x(3)+x(5)*Ps/(x(4)+x(8)*Ps)-x(5)*(Ns-x(6))/(1+x(8)*Ps/x(4)
)^2;
Z=1/(x(7)*x(3))+x(5)*Ps/(x(7)*(x(4)+x(8)*Ps))-(1-x(2))*x(5)*(Ns-x(6))
/(x(3)*(1+x(8)*Ps/x(4))^2);
H=Z./((2i*pi*f).^2+(2i*pi*f)*Y+Z);
M=20*log10(abs(H));
plot(f,M);

```