

关于液调配器的阻抗匹配算法的研究

潘亚平 赵燕平 秦成明

(中科院等离子体物理研究所 安徽 合肥 230031)

摘要 本文给出了液调配器的阻抗匹配算法,对算法进行深入的分析与研究,且给出两个典型实例,通过 Matlab 绘图对该算法进行了正确性验证。

关键词 阻抗匹配算法 Matlab

THE RESEARCH FOR IMPEDANCE MATCHING ALGORITHM
OF LIQUID STUB TUNER

Pan Yaping Zhao Yanping Qin Chengming

(Institute of Plasma Physics, Academia Sinica, Hefei Anhui 230031, China)

Abstract This paper talks about impedance matching algorithm of liquid stub tuner, and makes a thorough analysis about it. By making a rough sketch of two representative examples with Matlab, this paper verifies the correctness of the algorithm.

Keywords Impedance matching algorithm Matlab

1 引言

阻抗匹配是离子回旋加热系统 ICRF (Ion Cyclotron Range of Frequency) 中非常重要的一部分,阻抗匹配使发射机的功率通过天线辐射更加有效地耦合到等离子体中^[1]。阻抗匹配算法的正确性、高效性、稳定性、鲁棒性对 ICRF 系统起着决定性的作用。

液调配器是一种新型的调配器,其调配快速、高效、安全,避免了普通调配器高功率高频下机械操作的危险性^[2]。

根据微波传输线理论,经过一系列复杂的计算与推导,可得出当整个 ICRF 系统匹配时关于液调配器液面高度 x 的函数 $f(x)$ ^[3]:

$$f(x) = \tan(al * x) * \tan(ag * (1 - x)) - k - b * (\tan(al * x) + k * \tan(ag * (1 - x))) \quad (1)$$

式(1)中的参数说明如表 1 示:

表 1 参数说明

$al: 2\pi f\sqrt{\epsilon}/c$	$ag: 2\pi f/c$
f : 发射机的频率	c : 光速 3×10^8 米/秒
ϵ : 2.72 是硅油的介电常数	x : 液调配器的液面高度
l : 液调配器的总长度,在实际工程中取值为 6	b : 支节的输入导纳

寻求 x 使:

$$f(x) = 0 \quad (2)$$

由于(2)为超越方程,其求解可分两步进行:

首先,确定根的初始近似值或根的所在区间;然后,对方程的根进一步精确化。

为求方程(2)根的近似值,首先必须根据方程的性质,分析

根的大致分布情况,最好能先给出一个粗略的近似值或者预先确定出根所在的区间,然后在此基础上再进行加工,使之逐步精确化。常用方法有图解法和计算机扫描法。我们采用的是计算机扫描法,并结合图解法来验证计算机扫描法的正确性。

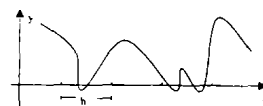


图 1

在对方程根进行扫描的过程中,步长 h 的选择应根据具体情况而定。如果步长 h 选得过大,就有可能扫描不到实根或者在一个小区间内根不唯一,如图 1 所示。但如果 h 选得过小,则需要扫描的步数就会增多,从而使计算工作量过大。因此,选择合适的步长 h 是扫描的关键^[4]。

显然,在扫描过程中,只要步长 h 选得足够小,就可以直接得到满足一定精度的根的近似值。但这种方法一般不可取。我们的做法是先利用逐步扫描法得到一个初始近似值或根的所在区间,有了根的初始近似值或根的所在区间后,再进行第二步,即通过某种“细加工”的过程,使之逐步精确化。在“细加工”过程中采用的是二分法。

深入分析(2)式,存在两种情况:

- $f(x)$ 不经过 x 轴,即 $f(x) = 0$ 无解,此时取 $f(x)$ 最接近 x 轴时的 x 值作为所求的结果。

- $f(x)$ 经过 x 轴,即 $f(x) = 0$ 时的 x 值即为所求的结果。

由于 $f(x)$ 是关于 x 的正切函数,其存在间断点 θ ,使 \tan 无解 ($\theta = \pi/2 + k\pi, k = 0, \pm 1, \pm 2, \dots$)。在计算机扫描的过程中,应剔除掉这些间断点。

2 算法描述

用 C 语言编写的具体算法如下:

```

#include "math.h"
double fun(x)
double x;
{double f, k, al, ag, pi, y, t;
f = 30000000; //发射机频率
pi = 3.1415926;
k = sqrt(2.72);
al = 2 * pi * f * k / 300000000;
ag = 2 * pi * f / 300000000;
t = 1.059602;
y = tan(al * x) * tan(ag * (6 - x)) - k - t * (tan(al * x) + k * tan(ag * (6 - x)));
return(y);
}
main()
{
double s0, s1, s2, s3, s4, n1, nn1, nnn1, nr2, nr3, err;
s0 = fun(0);
n1 = 0;
err = 0.001;
while(fabs(s0) > 1e-8 && n1 < 6)
{
nn1 = n1 + err;
//有些情况下方程没有解,取|fun(x)|值最小时的x值
s1 = fun(n1);
if(fabs(s0) > fabs(s1))
nnn1 = n1;
s2 = fun(nn1);
if(fabs(s0) > fabs(s2))
nnn1 = nn1;
//满足s1 * s2 < 0条件时,有两种情况:方程的解就
//在此范围内;tgθ中的θ为π/2, 3π/2
if(s1 * s2 < 0)
{
nr2 = n1;
nr3 = nn1;
while(fabs(nr3 - nr2) > 1e-12 && fabs(s0) > 1e-8)
{
s3 = fun(nr2);
nnn1 = (nr2 + nr3) / 2;
s0 = fun(nnn1);
s4 = fun(nr3);
if(s3 * s0 < 0)
nr3 = nnn1;
else if(s0 * s4 < 0)
nr2 = nnn1;
}
if(fabs(s0) < 1e-8)
{
printf("nnn1 = %lf \ n", nnn1);
break;
}
}
//有可能s1, s2就为方程的解
else if(fabs(s1) < 1e-8)
{
printf("n1 = %lf \ n", n1);
}
}
}

```

```

break;
else if(fabs(s2) < 1e-8)
{
printf("nn1 = %lf \ n", nn1);
break;
}
n1 = nn1;
}
printf("nnn1 = %lf \ n", nnn1);
printf("ok \ n");
}
}

```

3 算法验证

• 第一种情况

$f(x)$ 不经过 x 轴,即 $f(x) = 0$ 无解,此时只能取 $f(x)$ 最接近 x 轴时的 x 值作为所求的结果。令 $13 = 5.7$ (13 为第三根支节的液面高度),则可算出 $b = 36.894677$ (b 为第二支节导纳)。用 c 语言描述的算法求解时,结果为 $x = 6$ 。

从 Matlab 所绘图形中可看出方程的解为 $x = 6$,与算法算出的结果是一致的。

图 2 为 $f(x)$ 函数在 $[0, 6]$ 区间内的分布图,图中出现波峰与波谷的点为 $\tan \theta = 0$ 的无穷间断点,应去除掉这些间断点。这些点的值为 $x = 1.515847, 3.500000, 4.547543$ 。

图 3 显示了 $f(x)$ 在 $[5.5, 6]$ 区间逐渐逼近 x 轴,取最接近 x 轴时的 x 值作为所求结果,即 $x = 6$ 。

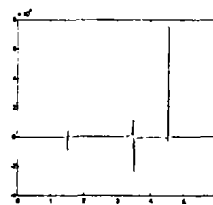


图 2

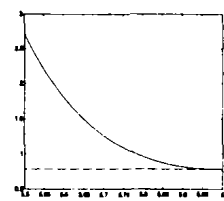


图 3

• 第二种情况

$f(x)$ 经过 x 轴,即 $f(x) = 0$ 时的 x 值即为所求的结果。令 $13 = 5.7$ (13 为第三根支节的液面高度),则 $b = 2.167614$ (b 为第一支节导纳)。用该算法求解的结果为 $x = 4.766196$ 。从 Matlab 所绘图形中可以验证该结果是正确的。

图 4 为 $f(x)$ 函数在 $[0, 6]$ 区间内的分布图,图中出现波峰与波谷的点为 $\tan \theta = 0$ 的无穷间断点,应去除掉这些间断点。这些点的值为 $x = 1.515847, 3.500000, 4.547543$ 。

图 5 显示了 $f(x)$ 在 $[4.7, 4.8]$ 区域经过 x 轴, $f(x)$ 经过 x 轴时的 x 值为 $x = 4.766196$ 。

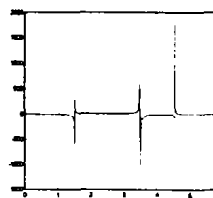


图 4

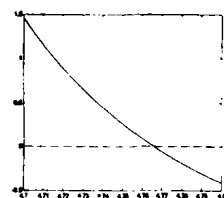


图 5

4 算法分析

液调配器阻抗匹配算法的基本思想是进行递增式的顺序搜

索,每次搜索的区间长度为 err ,判断方程的根是否落在该搜索区间内,若在该搜索区间内,则对该区间采用二分搜索;若不在该搜索区间内,则判断该区间的两个端点是否是方程的根。

算法的最好情况是在第一个递增搜索区间即 $[0, err]$ 内,进行第一次二分搜索时即 $(0 + err)/2$ 为所求的结果。

算法的最差情况是在整个区间搜索完毕即外循环搜索至 $[6 - err, 6]$ 区间,内循环采用二分法搜索完毕,算法的搜索次数为 $\frac{6}{err} * \lceil \log_2(n+1) \rceil$ 。如算法验证中给出的第一种情况,其整个区间 $[0, 6]$ 搜索完毕,非线性方程 $f(x) = 0$ 无解,取 $f(x)$ 最接近 x 轴时的 x 值作为方程的解。

算法的平均搜索次数^[5]为:

$$O((6/err + 1)/2) * O(\log_2(n + 1))$$

5 结束语

在实际工程中,我们采用基于 Labview 的开发平台来开发液调配的采集与控制系统。Labview 提供 CIN 节点来调用 C 语言编写的液调配器阻抗匹配算法的源代码。CIN 节点使基于 Labview 的编程更加简洁、方便、快速。

参 考 文 献

- [1] 廖承恩,微波技术基础,西安:西安电子科技大学出版社,1994.
- [2] R. Kumazawa, Yanping Zhao etc. Liquid stub tuner for ion cyclotron heating. Review of scientific instruments. volume 70, number 6, June 1999.
- [3] 秦成明、赵燕平等, plasma science & technology. volume 5 number 3, science press, 2003.
- [4] 徐士良,计算机常用算法,北京:清华大学出版社.
- [5] 潘金贵、顾铁成等,现代计算机常用数据结构和算法,南京:南京大学出版社,1994.
- [6] 谭浩强,C 程序设计,北京:清华大学出版社,1999.

(上接第 44 页)

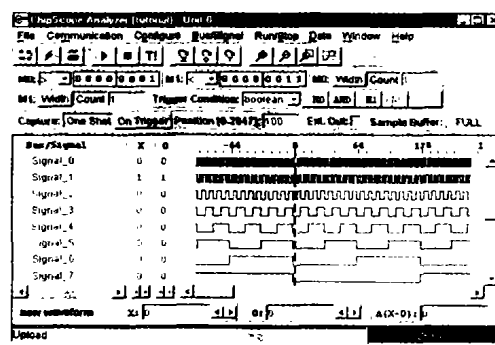
```
);
end top;
architecture behave of top is
signal counter: std_logic_vector(31 downto 0); = (others => '0');
signal control_bus: std_logic_vector(41 downto 0);
component icon; 例化分析器接口控制软核 icon
port
(
CONTROL0: out std_logic_vector(41 downto 0)
);
end component;
signal data: std_logic_vector(15 downto 0);
signal trig: std_logic_vector(7 downto 0);
component ila_dd256_dw16_tw8_e2; 例化信号捕捉控制软核 ila
port
(
CONTROL : in std_logic_vector(41 downto 0);
CLK: in std_logic;
DATA: in std_logic_vector(15 downto 0);
TRIG: in std_logic_vector(7 downto 0)
);
end component;
begin
process (clk)
```

```
begin
if (clk'event and clk = '1') then
counter <= counter + 1;
end if;
end process;
cnt <= counter(31 downto 28);
i_icon : icon
port map
(
CONTROL0 => control_bus; 连接 icon 总线
);
data <= counter(15 downto 0); 16 位可观察数据宽度
trig <= counter(7 downto 0); 8 位可触发数据宽度
i_ila_dd256_dw16_tw8_e2: ila_dd256_dw16_tw8_e2
port map; 连接 ila 总线
(
CONTROL => control_bus,
CLK => clk,
DATA => data,
TRIG => trig
);
end behave;
```

接下来要对已插入分析软核的原码进行综合。一旦把 ILA 模块例化到设计中,用 HDL 综合工具会把设计原码综合成 EDIF 或 XST 网表文件。在网表中 ILA 实际上是一个黑盒子,即只是一个网表。在实现时,编译软件会调用相应的模块库。

综合之后,再做布局布线,产生下载 FPGA 的比特流文件。将并行电缆连接到 FPGA 的 JTAG 口上,使用 ChipScope Pro 分析器,即可完成类似普通逻辑分析仪的工作了。

下图为软件分析器所抓取的计数器工作波形。



4 结束语

随着 FPGA 器件不断深入到数字电子的设计领域,FPGA 的逻辑容量不断增大,其封装方式不断向 BGA, FBGA 等方向发展,对于复杂 FPGA 的调试难度也越来越大。因此,像 ChipScope Pro 这样类似逻辑分析仪的软件工具的出现,大大方便了 FPGA 逻辑设计工作师的调试工作,也节约了很多的仪器成本。对 ChipScope Pro 工具的认识与熟练使用,将十分有效地提高逻辑与硬件的调试效率。

参 考 文 献

- [1] Virtex-II Pro/Virtex-II Pro X Complete Data Sheet; XILINX DS083 (v4.1) November 17, 2004.
- [2] ChipScope Pro Software and Cores User Manual; XILINX UG029 (v6.3.1) October 4, 2004.