



Y1760799

# The Design and Realization of GNSS Software Receiver Algorithm Based on FPGA

A Thesis Submitted to

Southeast University

For the Academic Degree of Master of Engineering

BY

Tang Xin-hua

Supervised by

Proefssor Chen Xi-yuan

School of Instrument Science and Engineering

Southeast University, Nanjing, P.R.China

June 2010

## 东南大学学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：汤新华 日期：2010.6.3

## 东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学研究生院办理。

研究生签名：汤新华 导师签名：阿西 日期：2010.6.3

## 摘要

在美国 GPS 的现代化、欧洲 Galileo 系统的出现、俄罗斯 GLONASS 的复兴和中国“北斗二代”部署的大背景下，软件接收机的研究已经称为一个必然的趋势。

本文以 GPS 为对象，阐述了 PRN(伪随机码)生成、GPS 信号调制过程、接收机前端的下变频、采样等基本原理，同时详细分析了 GPS 信号捕获、跟踪过程。

在 GPS 信号捕获环节中，为了能使算法适用于实际的硬件开发，改进了 GPS 捕获算法包括补零计算、循环相关改进、降采样。实验结果显示，补零计算可以弥补算法中采样点不基于  $2^N$  的不足，循环相关改进可以减少卷积的计算量，降采样可以减少整体计算数据量。三种改进算法的误差都在理论范围之内，不影响捕获的最终结果。

在 GPS 信号跟踪环节中，主要推导了锁相环 (PLL)、延迟锁定环 (DLL)、锁频环 (FLL) 的模型，而且同时改进了跟踪环路，采用了 FLL 辅助 PLL 完成跟踪的方法，前期 FLL 与 PLL 共同工作，后期 PLL 独立工作。实验结果显示了该辅助方法的可行性与优越性 (FLL 能够辅助 PLL 快速跟踪上目标卫星以防失锁)。最后提出了各环节编程实现过程中的要点问题。

最后探讨了 Virtex2p FPGA 开发板原理,详细说明了 System Generator 中的 XILINX 模块的结构及参数配置。为了能在 FPGA (可编程门阵列) 上实现 GPS 信号并行捕获算法，采用了前面补零计算来弥补算法中采样点不基于  $2^N$  的不足，降采样来减少整体数据量，并且在 Simulink 环境下运用基于 FPGA 的应用软件 System Generator 进行了并行捕获算法的 FFT 模块、虚数乘法模块，平方模块等搭建编程，同时采用了 Matlab 中的 M 文件将采样数据仿真成模拟信号导入完成仿真实验，并将仿真结果与 Matlab 结果进行了相应的比较和分析，得到与 Matlab 结果同样的捕获频率，虽码相位具有误差，但通过具体分析,该码相位误差由补零计算和降采样引起，证实了 System Generator 在 FPGA 实现捕获算法可行性。

本文研究的基于 System Generator 的 FPGA 开发，为以后软件接收机的后序硬件开发奠定了基础。

**关键字：GPS 接收机；捕获；跟踪；FLL；FPGA；System Generator**

## ABSTRACT

Under the background of the modernization of the GPS, the present of European Galileo system, the revival of Russian GLONASS and the construction of the Chinese second-generation Beidou system, the study of software receiver has become a trend.

In this paper, GPS is took as study object .the basic knowledge of pseudo random noise(PRN), the modulation of GPS signal, the technology involved in signal conditioning including frequency down conversion and sampling are studied, meanwhile, the processes of GPS signal acquisition and tracking are analyzed.

In GPS signal acquisition, in order to adapt the algorithm to the implementation in hardware, the algorithm is improved by the methods of zero padding, modified acquisition by circular correlation, reducing sampling rate. Through the corresponding experiments, it shows that zero padding can solve the limitation of data which is not based on  $2^N$ , modified acquisition by circular correlation can reduce the numbers involved in convolution, reducing sampling rate can lighten the whole loan of program computation. And the errors from the three advanced algorithm are acceptable in theory which won't influence the final result.

In GPS tracking, the models of PLL, DLL, FLL are proved. At the same time the PLL is improved using FLL to assist PLL in tracking. At beginning, FLL and PLL work together, after a while, PLL works singly without FLL. The result shows the feasibility and advantage of this modified method (FLL can assist PLL to track the satellite quickly to avoid the case not in a lock state). Later, the key points in programming are discussed.

At last, the principle of Virtex2p is discussed. The structure and the parameter configuration of XINLIX models in System Generator are studied. In order to implement the GPS parallel acquisition algorithm in FPGA, the method of zero padding is used to solve the limitation of data which is not based on  $2^N$ , and reducing sampling rate is used to decrease the total number of data. based on the Simulink environment ,the software System Generator is used to set up the FFT module, multiplier of imaginary number, square module of the improved acquisition algorithm, the input signal is generated by M file from the sample data, the two results from Matlab and field programmable gate array (FPGA) are compared and analyzed , it shows that the same acquisition frequency can be got in System Generator, Though the code phase deviates a little, it is demonstrated that the error is caused by zero adding and reducing sampling rate, in all, algorithm realization feasibility based on FPGA is verified.

**Key words:** GPS receiver; acquisition; tracking; FLL; FPGA; System Generator

## 目 录

摘要.....	I
ABSTRACT.....	II
目 录.....	III
第一章 绪 论.....	1
1.1 研究背景与意义.....	1
1.2 国内外发展现状.....	2
1.3 本文主要内容与结构.....	3
第二章 GPS 接收机原理.....	5
2.1 GPS 信号.....	5
2.1.1 C/A 码.....	5
2.1.2 GPS 载波信号与调制.....	10
2.2 GPS 接收机前端.....	12
2.2.1 下变频.....	12
2.2.2 采样.....	13
2.3 GPS 信号捕获与跟踪.....	13
2.3.1 GPS 信号捕获.....	14
2.3.2 GPS 信号跟踪.....	15
2.4 小结.....	15
第三章 GPS 信号捕获算法及改进.....	16
3.1 GPS 捕获算法.....	16
3.1.1 中频采样器.....	16
3.1.2 二维捕获.....	16
3.1.3 串行捕获.....	17
3.1.4 并行捕获.....	18
3.1.5 模型理论推导.....	19
3.2 精细频率估计.....	20
3.3 捕获算法编程实现.....	22
3.4 捕获算法的改进及实验结果.....	23
3.4.1 补零计算.....	23
3.4.2 循环相关改进.....	24
3.4.3 降采样.....	27
3.5 小结.....	27
第四章 GPS 信号跟踪算法研究.....	29
4.1 GPS 信号跟踪.....	29
4.1.1 线性模型.....	29
4.2 锁相环 (PLL).....	30
4.2.1 PLL 系统模型.....	30
4.2.2 等效噪声宽度(Noise Equivalent Bandwidth).....	34
4.2.3 PLL 鉴相器.....	35
4.2.4 PLL 编程实现.....	36
4.3 延迟锁定环 (DLL).....	37
4.3.1 DLL 鉴相器的选择.....	37

4.3.2 DLL 编程实现.....	38
4.4 锁频环 (FLL) .....	38
4.4.1 FLL 鉴相器.....	39
4.4.2 FLL 编程实现.....	39
4.4.2 FLL 与 PLL 结合 .....	39
4.5 实验结果分析.....	40
4.5.1 纯 PLL 跟踪卫星结果.....	40
4.5.2 等效噪声带宽、阻尼比对跟踪结果影响.....	41
4.5.3 FLL 辅助 PLL 跟踪实验分析 .....	43
4.6 小结.....	46
第五章 捕获算法的 FPGA 仿真实现.....	48
5.1 FPGA 开发平台 .....	48
5.2 基于 System Generator 的 FPGA 开发.....	50
5.2.1 FPGA 开发优点 .....	50
5.2.2 System Generator 开发.....	51
5.2.3 System Generator 模块介绍.....	51
5.3 实验测试.....	54
5.3.1 FFT 模块测试.....	54
5.3.2 捕获算法仿真.....	57
5.4 小结.....	59
第六章 结论与展望.....	60
6.1 本文工作总结.....	60
6.2 工作展望.....	60
致谢.....	62
参考文献.....	63
攻读硕士期间发表论文 .....	65

# 第一章 绪 论

## 1.1 研究背景与意义

1901年无线电信号以莫尔斯码方式成功发送字母“S”跨越大西洋（从英格兰到达纽芬兰），预示着无线电世纪的到来，1903年美国怀特兄弟实现了航空飞行，航空也慢慢发展起来，如何应用无线电导航成为了一个热点研究方向<sup>[1]</sup>。

GPS系统是基于美国在60年代研制使用的“子午仪（transit）卫星导航系统”和“时间导航（Timation）系统”，于70年代初投入开发的。从GPS计划的提出到该系统的建成使用，历经20余年，耗资数百亿美元<sup>[2]</sup>。截至2004年7月23日，在过去的近30年时间里，美国已经发射了50多颗GPS导航卫星，形成了目前为止世界上最为完善、应用最广泛的全球卫星导航系统，并且在海湾战争等一系列局部战争中发挥了重要的作用，使全世界重新开始认识卫星导航系统的战略意义<sup>[3]</sup>。

近几年，美国GPS的现代化，俄罗斯GLONASS的复兴和欧洲Galileo系统的出现即将带来下一代全球导航卫星定位系统（GNSS）。同样中国的北斗系统（BNTS）也正在慢慢发展中，BUTS覆盖范围有限，只能为中国国内及周边地区的用户提供服务，以后也会通过更多的卫星组成区域性或者全球性的导航系统<sup>[4]</sup>。如今，国内导航领域对北斗系统（BNTS）的全面发展已经开始准备，为了能顺利设计、掌握本国自己的导航系统，接收机的研发已经义不容辞，现在国内高校、公司等开始慢慢研究接收机技术并努力将技术将用于实际的实践开发。

自从1992年5月，在美国电信系统会议上，来自Mitre公司的Joe Mitola首先提出了“Software Radios”概念以来，软件接收机就被受到广泛的关注<sup>[5]</sup>。软件无线电的基本概念就是在一个通用硬件平台上，通过软件加载的方式用软件来实现所有无线电系统的通信功能。软件无线电的基本思想是将宽带模数变换器(A/D)及数模变换器(D/A)尽可能地靠近射频天线，建立一个具有“A/D-DSP-D/A”模型的通用的、开放的硬件平台，在这个硬件平台上尽量利用软件技术来实现电台的各种功能模块。目前GPS接收机由射频前端，用于信号处理的专用集成电路(ASIC)和进行位置解算的CPU组成。软件可以下载到CPU改变接收机的性能参数，然而预先设计的跟踪通道，相关器和控制环路参数已固化到ASIC中，其灵活性不强。相反，GNSS软件接收机通过软件实现其信号捕获和跟踪处理而不是硬件，由于软件接收机的开放性、全面可编程性、和灵活性的特点，在不需更改硬件的前提下，通过对软件模块的调整和升级就可以满足不同用户的需要，因此，在整合GNSS各频段卫星导航数据中，有深远的意义<sup>[6-10]</sup>。

FPGA（Field Programmable Gate Array）是一类高集成度的可编程逻辑器件，1985年起源于美国的XILINX公司，它是一种采用单元型结构的新型PLD器件，采用CMOS、SRAM工艺制作，在结构上与简单的阵列型PLD不同，它的内部由许多独立的可编程逻辑单元构成，各逻辑单元之间可以灵活地相互连接，具有密度高、速度高、编程灵活、可重新配置等优点。

在这二十几年的发展过程中, FPGA 硬件系统和软件开发工具不断完善且日趋成熟。从最初的 1200 个可用门到现在的数百万至上千万的单片 FPGA 芯片。FPGA 技术结合了微电子技术、电路技术、EDA 技术, 与专用的集成电路 ASIC 相比, FPGA 具有灵活性高、设计周期短、成本低、风险小等优势, 因而得到了广泛的应用<sup>[11]</sup>, 而且用 FPGA 替代一般的 ASIC 芯片进行设计已经成为一种发展趋势。

一方面 FPGA 可以通过编写代码来实现复杂逻辑电路, 另一方面其内部是由规则的逻辑阵列所组成, 这样就可以开发出适应的专用芯片。最近几年里, FPGA 已近成为数字信号处理系统的核心器件, 尤其在数字通信、网络、视频和图像应用领域中<sup>[12]</sup>。如今的 FPGA 不仅包含查表法、寄存器、多路复用器、分布式块存储器, 而且嵌入了专用的快速加法器、乘法器和输入/输出设备。更重要的一点是 FPGA 具有实现高速并行运算的能力, 这就使得 FPGA 成为高性能的数字信号处理方面的理想器件, 如在数字滤波、快速傅立叶变换等方面, FPGA 具有可重新编程的优点, 比 ASIC 更实用<sup>[13]</sup>。

传统的数字信号处理系统核心器件较多使用 DSP 处理器, 因此数字信号处理系统的设计者和底层技术人员通常对 C 语言和汇编语言很熟悉, 但是对硬件描述语言 VHDL 或 Verilog 并不熟悉, 而且硬件描述语言最终要转换成硬件实现, 这就要求工程师对硬件有一定的了解, 使用 XILINUX System Generator for DSP 可以使上述问题迎刃而解, System Generator 是一款理想的 FPGA 开发软件<sup>[14]</sup>。

本文主要以 GPS 为对象, 详细研究了 GPS 信号捕获、跟踪以及基于软件 System Generator 的 FPGA 开发, 其思想可以贯通于其余导航系统, 包括中国自制的北斗系统, 与传统的接收机相比, 软件接收机的优点主要有一下几点<sup>[15]</sup>:

- 1、实时功率谱分析。可用于监视天线信号的频谱, 检测窄带连续波干扰。
- 2、中频数据重演。用户可以经常重复使用采集的原始射频信号, 以优化算法和分析结果。
- 3、多传感器组合。在软件层面上组合传感器数据; 例如, 允许用户利用惯性测量器件等的所有信息, 获得“超紧耦合”的 GPS / INS 组合导航解。
- 4、多相关器输出。允许用户采用先进的鉴相器与滤波器组合跟踪环路; 在相关器层面对反射信号与多径干扰进行分析。

总而言之, 软件接收机为以后其他导航系统提供了方便的平台, 将研究成本最低化, 并显示出巨大的灵活性和商业潜力。而且为创新提供了一个极好的通道, 同时为普通研究者一个理想的用于教学与研究的平台, 能让更多的学者认知并投入其中, 使之具有一个美好的未来。

## 1.2 国内外发展现状

第一批 GPS 接收机研制于 20 世纪 70 年代, 供 DoD (美国国防部) 验证 GPS 的可行性。

1982 年, 为精密测量大地而设计的两款 GPS 接收机进入市场。麻省理工学院 Charles C. Counselman III 设计、Steinbrecher 公司制造。

1989 年 DoD 开始发射数量足以布满星座的 Block II 卫星, 并发布了宣布系统运行的计划。从此, 注意力开始向信号处理转移, 专用集成电路 (ASIC) 被采用, 最终实现了微型化发展<sup>[1]</sup>。

经过多年的发展, 基于 ASIC 硬件接收机技术早已成熟, 近几年众多国内外学者开始研究软

件接收机,并同时也取得了一系列明显的成就。理论知识已经慢慢趋于成熟,具有代表性的是 Pratap Misra (Lincoln Laboratory)和 Per Enge(Stanford University)的 Global Positioning System (Signals, Measurements, and Performance),对 GPS 的基础、信号、接收机等做了详细的解释和理论推导<sup>[15]</sup>。Kai Borre、Dennis M.Akos 等所著的 A Software-Defined GPS and Galileo Receive 从信号处理的基本知识到在软件接收机的具体应用都描述的非常仔细<sup>[16]</sup>,国内也开始紧跟国外的理论发展。

在基础知识之上,国外开始研究算法改进和及其理论分析。1997年8月,Ohio大学的 Dennis M. Akos 在他的博士论文中讨论了 GNSS 软件接收机前端模拟信号的接收和模数转化 (Front-End) 的两种设计方案,并且事后处理并验证了卫星信号捕获的 FFT 算法的可行性和跟踪环路的稳定性。JAMES BAO-YEN TSUI 在 Fundamentals of Global Positioning System Receivers A Software Approach 结合了 matlab 分析了软件接收机。并且提出了一种新的信号跟踪方法——BASS(Block Adjustment of Synchronizing Signal)。

基于理论知识的研究的深入,对算法的具体实现的研究也开始慢慢热起来,2005年在美国 Long Beach ,California 的界国际卫星导航会议中 Hun-Soo Cho Sung-Hyuck Im Gyu-In Jee 提出了基于 System Generator 的软件接收机实现,提出软件在 pc 机中无法达到实时性,并对 FPGA 中实现理论分析。同样在最近的 2007 年的国际全球卫星导航会议中 Tao Huang Bin Guo Matthew Trinkle (澳大利亚)也类似的发了基于 System Generator 的软件接收机实现问题的文章,主要做了跟踪环路的系统搭建并把它用于教学中。算法的移植和实时性技术仍待提高和改进<sup>[17]</sup>。

在国内,对软件接收机的研究也开始慢慢增多,我国电子集团 54 所、航天 704 所、北京航空航天大学、国防科技大学、武汉大学、中科院等高校以及多家从事卫星导航设备开发的公司,都在 GPS 接收机的研制方面投入了较大的人力物力,并相继尝试开发 GPS 接收机,其中北航的软件接收机在 pc 机上的软件实现技术已经成熟。同样国内将 FPGA 大多应用于硬件接收机,也就是纯粹的用 FPGA 来实现 ASIC 电路,在用 FPGA 实现具体的软件接收机算法这块研究较少。目前 GPS 在我国的科研应用主要有如下明显缺点:(1)主要采用进口 OEM 板作为 GPS 接收机内核,并非自主研制;(2)主要是进行产品的二次开发,如 GPS/GIS 地理信息系统。总而言之,国内 GNSS 软件接收机的研究还处于起步水平<sup>[18-21]</sup>。

### 1.3 本文主要内容与结构

本文主要介绍了 GPS 接收机发展背景、GPS 接收机原理、FPGA 芯片结构以及开发原理。在此基础上通过具体编程实验进行了进一步分析与改进。其中包括 GPS 信号捕获算法的改进,GPS 信号跟踪环路的改进,加入了 FLL 来辅助 PLL 进行跟踪,最后在 Simulink 平台下用 System Generator 软件仿真实现了捕获算法。

全文共分六章,各部分结构如下:

第一章:绪论 介绍GNSS软件接收机、FPGA开发的研究背景、意义和国内外研究现状,概述论文主要内容和论文结构。

第二章:GPS接收机原理 对GPS信号的产生原理做了详细理论推导,包括C/A码发生器结构,GPS信号调制,接收机前端。并介绍了GPS信号捕获与跟踪,为后面的信号处理打下了基础。

第三章：GPS信号捕获算法及改进 讨论了GPS C/A码信号捕获传统方法，包括串行、并行捕获，并在此基础上提出了三大改进方法，补零计算、循环相关改进、降采样。并用实验进行了验证。

第四章：GPS信号跟踪算法研究 主要分析了锁相环（PLL）、延迟锁定环（DLL）、锁频环（FLL）结构，详细推导了各自鉴别器理论依据。并改进了跟踪环路，采用了FLL辅助PLL方式。最后用相应实验进行了论证。

第五章：捕获算法的FPGA仿真实现 主要介绍了FPGA芯片结构原理，并详细论述了XILINX模块的参数配置，最后完成了相应实验。

第六章：总结与展望 论文完成工作的总结及后续研究的个人意见。

## 第二章 GPS 接收机原理

## 2.1 GPS 信号

GPS 卫星在不停的发射着两种波段的无线电波，分别是 L1 和 L2 波段，L 波段覆盖的频率波段主要在 1GHz 和 2GHz 之间，L1 和 L2 波段的中心频率分别是：

$$L1: f_{L1} = 1575.42\text{MHz}, L2: f_{L2} = 1227.60\text{MHz}.$$

其中 L1 中包含了两种信号，一个是给平面用的，另一个是需 DoD (department of defense) 批准的用户，L2 上的信号则是专为 DoD (department of defense) 批准的用户服务。

GPS 信号主要有 3 种组成：数据码 (D 码，或者基带信号)、测距码 (C/A 码、P 码) 和载波信号 (L1 和 L2)。

用户是用 GPS 接收机来接收 GPS 卫星信号，然后经过捕获跟踪等一系列相关处理来测出信号传播的时间延迟，从而得到相应的伪距 (因为含有误差)，从而完成定位等一系列功能。

## 2.1.1 C/A 码

GPS 卫星信号主要有两种伪随机码 (pseudo-random noise, PRN)，他们分别是 C/A(coarse/acquisition)码和 P 码，本文中主要讨论 C/A 码，C/A 码是用于分址、捕获卫星信号和粗测距，具有一定的抗干扰能力，提供给民用。

GPS C/A 码是属于 Gold 码的伪随机码，信号是两个 1023 位的 PRN 序列 G1 和 G2 产生的，其中，G1 和 G2 都是由 10 位最大长线性移位寄存器组成，并且又同一个 1.023MHz 的时钟来驱动。

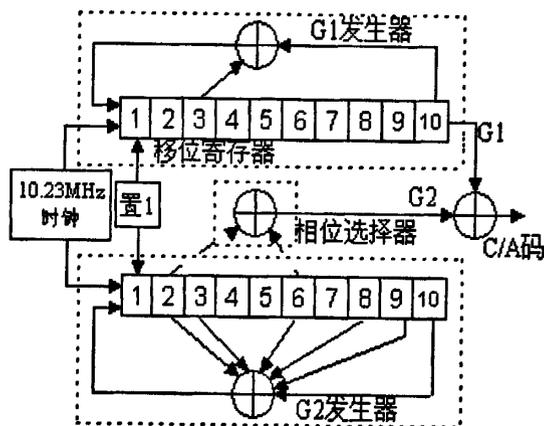


图 2.1 GPS C/A 码发生器

两个产生器工作的基本原理一样，由带适当反馈的移位寄存器构成，如果移位寄存器位数为  $n$ ，则产生的序列长度为  $2^n - 1$ 。G1 和 G2 都是由 10 位寄存器组成，所以对应产生的序列长度为  $2^{10} - 1 = 1023$ 。反馈电路是由模 2 加法实现的。两个移位寄存器 G1 和 G2 的初值是全“1”且预先置入寄存器。卫星表示号 (ID) 是根据 G2 发生器的两个输出位置来决定的 (图 2.1)。总共有 37 个不同的输出序列，其中 32 个序列可为 32 颗卫星的 C/A 码所利用，且目前只有 24 颗卫星在运作，另外 5 个 C/A 码保留作为其他用途。

表 2.1 C/A 码相位分配规定

卫星 PRN 号	C/A 码 抽头 选择						
1	2 ⊕ 6	11	3 ⊕ 4	21	5 ⊕ 8	31	3 ⊕ 8
2	3 ⊕ 7	12	5 ⊕ 6	22	6 ⊕ 9	32	4 ⊕ 9
3	4 ⊕ 8	13	6 ⊕ 7	23	1 ⊕ 3	33	5 ⊕ 10
4	5 ⊕ 9	14	7 ⊕ 8	24	4 ⊕ 6	34	4 ⊕ 10
5	1 ⊕ 9	15	8 ⊕ 9	25	5 ⊕ 7	35	1 ⊕ 7
6	2 ⊕ 6	16	9 ⊕ 10	26	6 ⊕ 8	36	2 ⊕ 8
7	1 ⊕ 8	17	1 ⊕ 4	27	7 ⊕ 9	37	4 ⊕ 10
8	2 ⊕ 9	18	2 ⊕ 5	28	8 ⊕ 10		
9	3 ⊕ 10	19	3 ⊕ 6	29	1 ⊕ 6		
10	2 ⊕ 3	20	4 ⊕ 7	30	2 ⊕ 7		

这组C/A码的码长、周期和数码率均相同。即：

码长： $N=2^{10}-1=1023\text{bit}$

周期： $T=1\text{ms}$

数码率： $1.023\text{Mbit/s}$

由此可以推断出 码元宽度为： $t_0 = \frac{1}{1023}\text{ms} = 0.97752\mu\text{s}$ ，这样一个码元对应的距离为：

$$3 \times 10^8 \times 0.97752 \times 10^{-6} = 293.3\text{m}。$$

由于C/A码相对于P码比较短，所以易于捕获。在GPS定位中，为了捕获C/A码，以测定卫星信号的传播延迟，需要对所有的C/A码逐个搜索，通过C/A码捕获卫星后，获得导航电文后，便可以很容易地捕获P码（也是利用P码进行定位的思路）。C/A码的码元宽度较大，一个码元误差对应的测距误差为293.3m，由于精度较低，故C/A码也称为粗码。

在用C语言或者Matlab进行C/A码发生器编程的时候，用两个长度为10的一维数组代替G1和G2即可，然后根据具体的C/A码发生器结构进行具体的反馈操作，模2加法采用除2取余算法即可，如图2.1所示。

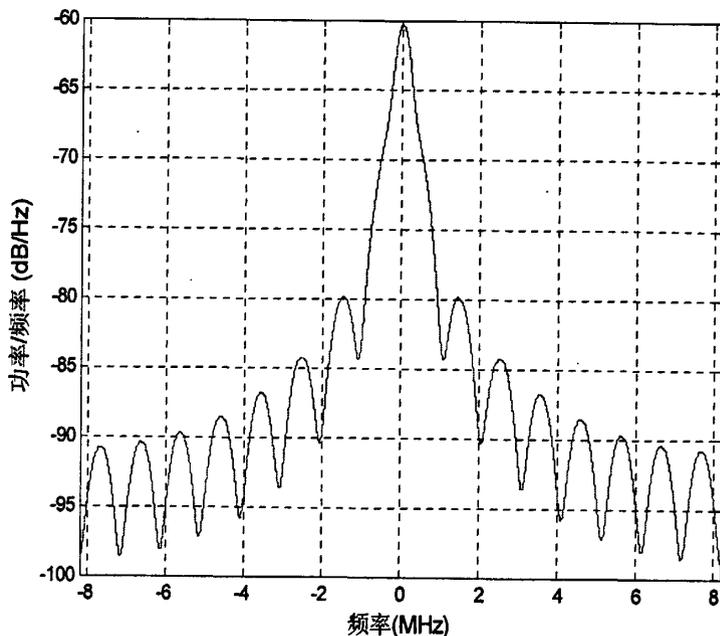


图2.2 C/A码的频谱图(将1023位扩成16367位后的C/A码)

GPS信号采用C/A码是利用了其相关特性，该特性类似于数学中的三角函数的正交性，所谓的三角函数系<sup>[22]</sup>：

$$1, \cos x, \sin x, \cos 2x, \sin 2x, \dots, \cos nx, \sin nx, \dots$$

在区间 $[-\pi, \pi]$ 上正交，就是在三角函数系中任何不同的两个函数的乘积在区间 $[-\pi, \pi]$ 上的积分等于零，即：

$$\int_{-\pi}^{\pi} \cos kx \sin nx = 0 (k \neq n, k, n = 1, 2, 3, \dots) \quad (2-1)$$

其中区间 $[-\pi, \pi]$ 可以改成任何长度为 $2\pi$ 的区间。

前面提过在对GPS信号进行捕获的时候，需要对所有的C/A码逐个搜索，这样做的目的是为了搜索到可见星，因为不同的C/A码对应着不同的卫星，相当于一个对号入座的过程，恰恰类似与三角函数正交性的C/A码的相关特性提供了基础。

如果码是正交的，则互相关的值就是零，Gold码虽然不是正交，但是接近于正交，这意味着互相关值不为零，但是非常小。

从第K颗卫星得到的码的时间平均时间相关函数 $x^{(k)}(t)$ 如下：

$$R(\tau) = \frac{1}{T_{code}} \int_0^{T_{code}} x^{(k)}(t)x^{(k)}(t-\tau)dt \quad (2-2)$$

下面根据时间偏移和码元周期的关系来进行讨论<sup>[17]</sup>：

首先，来看下时间偏移是码元周期的整数倍的情况，即 $\tau = \ell T_c$ ：

$$\begin{aligned}
 R(\tau = iT_C) &= \frac{1}{T_{code}} \int_0^{T_{code}} x^{(k)}(t)x^{(k)}(t - iT_C)dt \\
 &= \frac{T_C}{T_{code}} \sum_{n=0}^{N-1} x_n x_{n+i} \\
 &= \frac{T_C}{T_{code}} (\#agreements - \#disagreements) \\
 &= \frac{1}{N} (\#agreements - \#disagreements)
 \end{aligned} \tag{2-3}$$

计算结果只是简单的计数过程，即偏移后的波形和原波形极性相同的次数与极性不同的次数之差。很明显可以得到：

$$\begin{aligned}
 R(\tau) \leq R(0) &= \frac{1}{T_{code}} \int_0^{T_{code}} x^2(t)dt \\
 &= \frac{1}{N} (\#agreements - \#disagreements) \\
 &= 1
 \end{aligned} \tag{2-4}$$

当  $\tau \neq iT_C$  时，即时间延迟  $\tau$  落在  $iT_C$  和  $(i+1)T_C$  之间：

$$\begin{aligned}
 R(\tau) &= \frac{1}{T_{code}} \int_0^{T_{code}} x(t)x(t-\tau)dt \\
 &= \frac{1}{T_{code}} \sum_{n=0}^{N-1} \int_{iT_C}^{(n+1)T_C} x(t)x(t-\tau)dt \\
 &= \frac{1}{T_{code}} \sum_{n=0}^{N-1} \left( \int_{iT_C}^{(n+1)T_C - (\tau - iT_C)} x(t)x(t-\tau)dt + \int_{(n+1)T_C - (\tau - iT_C)}^{(n+1)T_C} x(t)x(t-\tau)dt \right) \tag{2-5} \\
 &= \frac{1}{T_{code}} \sum_{n=0}^{N-1} (x_n x_{n+i} * (T_C - (\tau - iT_C)) + x_n x_{n+i+1} * (\tau - iT_C)) \\
 &= R(iT_C)(i+1 - \frac{\tau}{T_C}) + R((i+1)T_C)(\frac{\tau}{T_C} - i)
 \end{aligned}$$

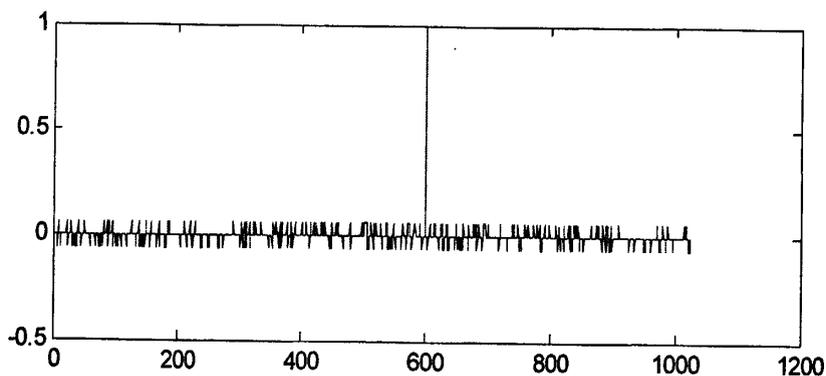
可见，相关函数  $R(iT_C < \tau < (i+1)T_C)$  落在  $R(iT_C)$  与  $R((i+1)T_C)$  的连线上。其中一定要注意公式(2-5)中的\*乘号。

表 2.2 Gold 码的互相关性能

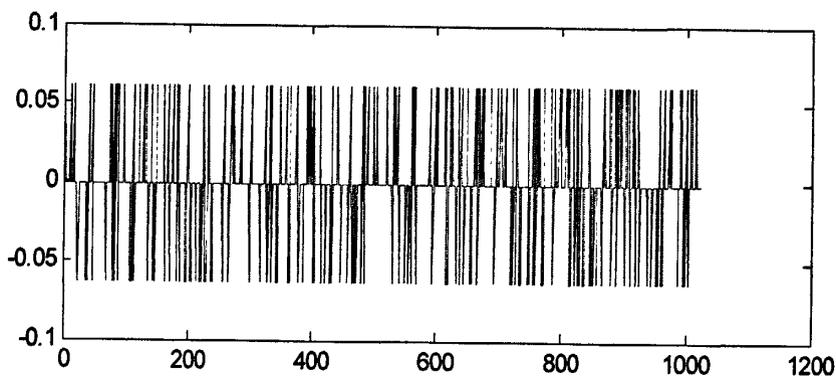
码周期	移位寄存器级数	归一化的互相关值	值出现的概率
$p = 2^n - 1$	n 为奇数	$\frac{2^{(n+1)/2} + 1}{P}$	0.25
		$-\frac{1}{P}$	0.5
		$\frac{2^{(n+1)/2} - 1}{P}$	0.24
$p = 2^n - 1$	n 为偶数	$\frac{2^{(n+2)/2} + 1}{P}$	0.125
		$-\frac{1}{P}$	0.75
		$\frac{2^{(n+2)/2} - 1}{P}$	0.125

由表 2.2 可得, 当  $n=10$  时,  $p=1023$ , 得到的相关值为:  $-65/1023$  (12.5%)、 $-1/1023$  (75%)、 $63/1023$  (12.5%)。

用 Matlab 编程可得如图 2.3 结果:



(a) 第 3 颗 GPS 卫星的自相关, 起始点移位了 600 个码元



(b) 第 3 颗和第 5 颗 GPS 卫星的互相关

图 2.3 C/A 码的自相关和互相关

## 2.1.2 GPS 载波信号与调制

根据前面的讨论,可以知道 GPS 信号主要有 3 种组成:数据码(D 码,或者基带信号)、测距码(C/A 码、P 码)和载波信号(L1 和 L2)。这三种信号分量,都是在同一个基本频率  $f_0 = 10.23\text{MHz}$  的控制下产生的。

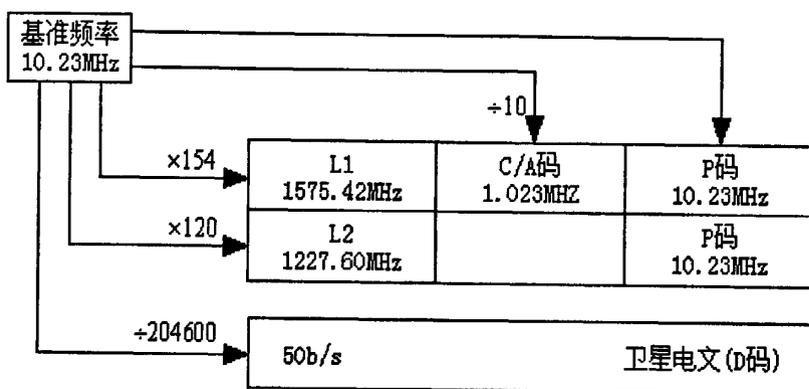


图2.4: GPS信号的产生

GPS 广播的导航信号如下式:

$$S_{L1}(t) = \sqrt{2P_c} D(t)x(t) \cos(2\pi f_{L1} + \theta_{L1}) + \sqrt{2P_{Y1}} D(t)y(t) \sin(2\pi f_{L1} + \theta_{L1}) \quad (2-6)$$

$$S_{L2}(t) = \sqrt{2P_{Y2}} D(t)y(t) \sin(2\pi f_{L2} + \theta_{L2})$$

式(2-6)所示的任何一种信号都是四部分的乘积:振幅  $\sqrt{2p}$ ; 导航数据  $D(t)$ ; 扩频码  $x(t)$  或  $y(t)$ , 也就是 C/A 码或者 P 码; 射频(RF)载波  $\cos(2\pi ft + \theta)$  或者  $\sin(2\pi ft + \theta)$ 。

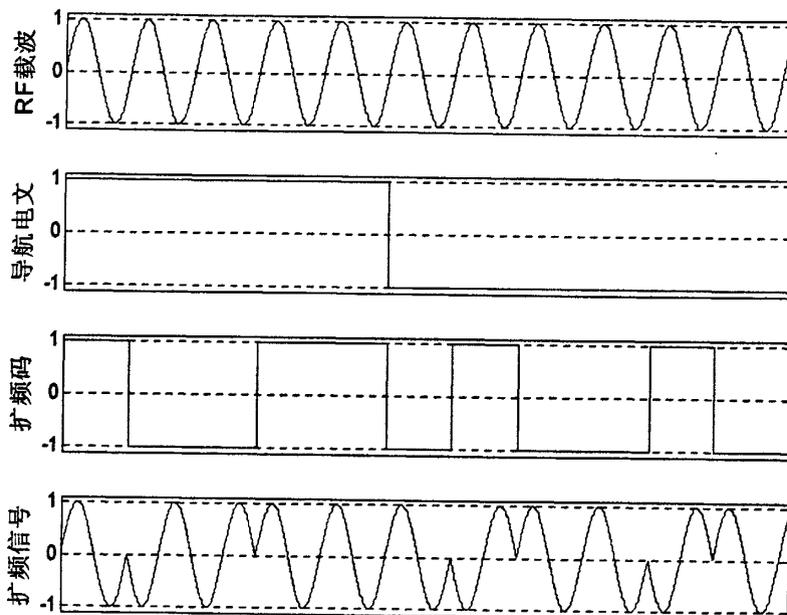


图2.5 导航电文和扩频码调制RF载波示意图

由图 2.5 可知, 数据和扩频码 (C/A 码和 P 码) 都是使用二进制相移键控 (BPSK) 调制在传输的载波上的, 其中数据码和扩频码可以用具体的公式表达出来。

$$\text{数据码可以表示为: } Ap\left(\frac{t-\tau}{T}\right), \text{ 其中 } p(t) = \begin{cases} 1, & -\frac{1}{2} \leq t \leq \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}$$

$$\text{C/A 码可以表示为: } x(t) = \sum_{n=0}^{N-1} x_n p\left(\frac{t-nT_c}{T_c}\right)$$

$$N=1023$$

GPS 的出色性能主要来自于扩频技术, 二进制码元宽度  $T=20\text{ms}$ , 对应基带的信号宽度为  $\Delta F = \frac{1}{T} = 50\text{Hz}$  (码率=50bit/s), 如伪随机码 C/A 码的码率为  $f_{C/A} = 1.023\text{MHz}$ , 相应的的码元宽度  $t_0 = 1\mu\text{s}$  左右, 将  $D(t)$  调制到 C/A 码上, 即将二者模 2 相加或者直接波形相乘, 得到了  $1.023\text{MHz}$  的信号, 基带信号  $D(t)$  的频率从  $50\text{Hz}$  被扩展到  $1.023\text{MHz}$ , 这就是伪码扩频。

现多数无线电系统, 都是将信息调制到射频载波中进行传输, 这样, 简简单单的载波在其中担任了巨大的作用, 相当于媒介将信息从发射器传送到接收器。

当 GPS 信号在频率上表示时, 由数字信号处理基础知识:

$$A(f) = F\{a(t)\} = \int_{-\infty}^{\infty} a(t) \exp(-j2\pi ft) dt$$

$$a(t) = F^{-1}\{A(f)\} = \int_{-\infty}^{\infty} A(f) \exp(j2\pi ft) df \quad (2-7)$$

$$\exp(j2\pi ft) = \cos(2\pi ft) + j \sin(2\pi ft)$$

当用频率去分析 GPS 信号的时候, 可以得到:

$$F\{\sqrt{2P_c}x(t)\cos(2\pi f_{L1})\} = \frac{\sqrt{2P_c}}{2} X(f-f_{L1}) + \frac{\sqrt{2P_c}}{2} X(f+f_{L1}) \quad (2-8)$$

式 (2-8) 说明了信号在进行载波调制的时候, 频谱被分成两部分, 一部分是加了  $f_{L1}$ , 另一部分则是减少了  $f_{L1}$ 。现从理论上分析  $X(f)$ , 可以近似的认为:

$$F\{x(t)\} = F\left\{p\left(\frac{t}{T_c}\right) * \sum_{n=0}^{N-1} x_n \delta(t-nT_c)\right\}$$

$$= F\left\{p\left(\frac{t}{T_c}\right)\right\} F\left\{\sum_{n=0}^{N-1} x_n \delta(t-nT_c)\right\}$$

$$= T_c \frac{\sin(\pi T_c f)}{\pi T_c f} \int_{-\infty}^{\infty} \sum_{n=0}^{N-1} x_n \delta(t-nT_c) \exp(-2\pi ft) dt \quad (2-9)$$

$$= T_c \frac{\sin(\pi T_c f)}{\pi T_c f} \sum_{n=0}^{N-1} x_n \exp(-2\pi fnT_c)$$

$$= T_c \frac{\sin(\pi T_c f)}{\pi T_c f} DFT_x(f)$$

## 2.2 GPS 接收机前端

GPS 信号从卫星传播到地球, 当到达接收机的时候, 用对应的能量体现出来, 事实上, GPS 信号此时的能量非常弱, 只有  $-160\text{dBW}$  ( $10^{-16}\text{W}$ ),  $L_1$  波段的载波频率为  $1575.42\text{MHz}$ , 根据 GPS C/A 码  $2\text{MHz}$  的带宽, 实际接收到的 GPS 信号能量在热噪声之下<sup>[23]</sup>:

$$\begin{aligned} p_{\text{thermalNoise}} &= kTB = 1.38 \times 10^{-23} \times 290 \times 2 \times 10^6 \\ &= 8.004 \times 10^{-15} \text{W} \\ &= 10 \log_{10}(8.004 \times 10^{-15}) \text{dBW} = -140.91 \text{dBW} \end{aligned} \quad (2-10)$$

除了噪声, 还有其他因素的影响, 如其他卫星上发射的信号等, 即便如此, 接收机仍然有能力捕获这个微弱的信号, 并且从中解调出需要的导航电文, 来进行相应的计算。

为了达到这个目的, 接收机的前端需要将相邻频段的干扰信号出去, 并且放大 GPS 有用信号功率, 同时还得将载波频率降低, 这样可以将接收机设前端计成如图(2-6)所示:

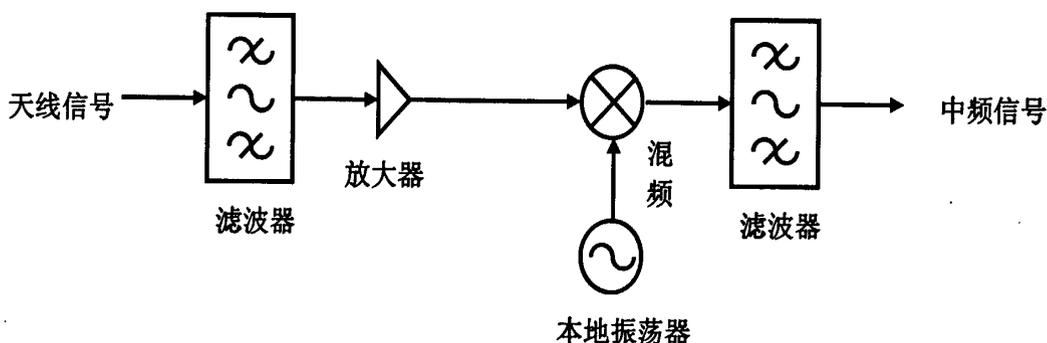


图 2-6 前端信号处理

## 2.2.1 下变频

下变频基于的三角函数等式:

$$(A \cos \alpha)(B \cos \beta) = \frac{AB}{2} (\cos(\alpha + \beta) + \cos(\alpha - \beta)) \quad (2-11)$$

接收机接到的信号可以表示成:

$$Y_{\text{REC}} = \sqrt{2C} x(t-\tau) D(t-\tau) \cos(2\pi(f_L + f_D)t + \theta) \quad (2-12)$$

本地参考信号表示成:

$$Y_{\text{ref}} = \sqrt{2} \cos(2\pi(f_L - f_{\text{IF}})t + \theta_{\text{IF}}) \quad (2-13)$$

这样将式(2-11)与式(2-12)相乘可得

$$\begin{aligned} Y_{\text{REC}} * Y_{\text{ref}} &= \sqrt{C} x(t-\tau) D(t-\tau) \cos(2\pi(2f_L - f_{\text{IF}} + f_D)t + \theta + \theta_{\text{IF}}) + \\ &\quad \sqrt{C} x(t-\tau) D(t-\tau) \cos(2\pi(f_{\text{IF}} + f_D)t + \theta - \theta_{\text{IF}}) \end{aligned} \quad (2-14)$$

一般,  $2f_L \gg f_{\text{IF}}$ , 再通过一个带通滤波器 (BPF), 只允许  $f_{\text{IF}}$  附近的信号通过。可以得出剩下的信号, 如式(2-14)所示:

$$\sqrt{C}x(t-\tau)D(t-\tau)\cos(2\pi(f_{IF}+f_D)t+\theta-\theta_{IF}) \quad (2-15)$$

这个过程称作为混频，但是在实际的操作过程中，不可能直接使用  $f_{IF}$  频率参考的余弦波去做混频，因为这样对硬件的要求太高，所以在实际中，大多数接收机都采用了多级变频，放大的过程被分配到多个频率中。这样就减少了对硬件的要求，而且提高可靠性。

### 2.2.2 采样

信号采样是很关键的一步，在变换器中，接收到的信号是连续的模拟量。在变换完成以后，模拟信号就变成了数字信号，一般 GPS 接收机采用了 2 位的 A/D，如东方联星的前端 NewStar210A，也采用了 2 位。这样，可以将信号量化成 4 个级别，这样接收机就不会遭受更多的信噪比损耗，但是它们需要一个 AGC (automatic gain control)，AGC 是用来控制输入信号振幅的，来确保振幅分布于 A/D 之间。

当用采样波形  $y(t)$  对被采样信号  $x(t)$  进行采样时，如果采样率足够高时，那么采样就能重现  $x(t)$ 。如果  $x(t)$  在频率  $f_{up}$  之上没有能量，那么它能够通过高于  $2f_{up}$  的采样率被重构。这就是基带采样。

在 GPS 接收机中，越来越多采用了带通采样，它是根据带通信号的统一采样定理，将采样任务和下变频任务进行了组合。如果  $x(t)$  的上限频率为  $f_{up}$ ，带宽为  $B$ ，那么它能够通过  $2f_{up}/m$  的统一采样率来重构信号，其中  $m = \lfloor f_{up}/B \rfloor$ ， $\lfloor \quad \rfloor$  表示不超过变量的最大整数。采样率现在又跟信号的带宽有关系，并不是完全有它的最高频率来决定。

著名的香农采样定理也说明了连续信号  $x_a(t)$  的最高频率为  $\Omega_m$ ，则最低的采样频率  $\Omega_s \geq 2\Omega_m$  [24]。

## 2.3 GPS 信号捕获与跟踪

当信号经过调整，已经放大到能激发 A/D 转换器的水平，最后生成的数字数据可以表示如下：

$$S(t) = \sqrt{C}D(t-\tau)x(t-\tau)\cos(2\pi(f_{IF}+f_D)t+\delta\theta_{IF})+n(t) \quad (2-16)$$

这样，后面的数字信号处理是本论文的重点，若要正确得到某卫星的导航电文，需要经过捕获、跟踪这两个重要环节。如图 2.7 所示。

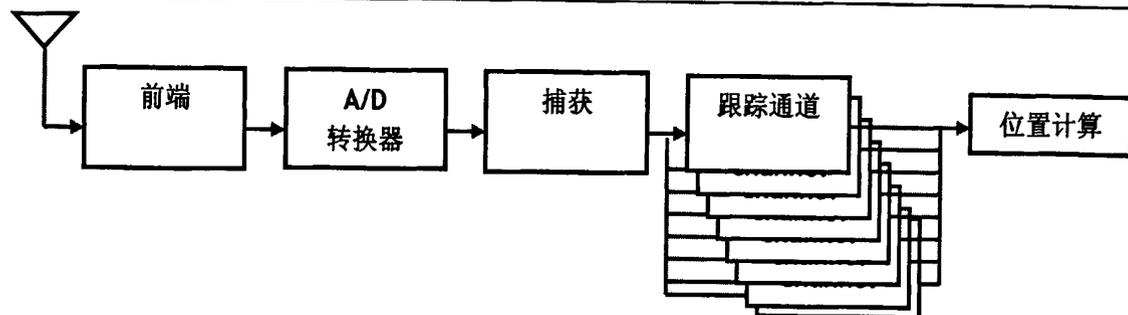


图 2.7 接收机示意图

### 2.3.1 GPS 信号捕获

GPS 信号捕获的目的是为了确定可见星，如果一颗卫星为可见，捕获就能得到以下信号的两个特点：

**载波频率：**从一颗卫星上得到的信号载波频率跟设定的值是有变化的，在信号经过下变频之后 L1 波段的理论频率是  $f_{IF}$ ，然而信号是受卫星的运动影响的，也就是多普勒频移，多普勒频移所造成的影响大概在 10KHz 左右，对一个地球上的静止接收机，多普勒频移不会超过 5KHz。

**码相位：**码相位可以表示数据块中的 C/A 码的起点位置，如果分析 1ms 数据块，就可以包含一个完整的 C/A 码周期，并且找到一个 C/A 码的起点。

当然捕获的方法有好多种，串行捕获，并行捕获等一系列，这些捕获方法都是基于 GPS 信号的特点，在这里 C/A 码的相关性就起到了很大的作用。

接收机接受到的信号 S 是好多颗可见星的混合信号：

$$s(t) = s^1(t) + s^2(t) + \dots + s^n(t)$$

当在捕获卫星 K 的时候，输入的信号 S 与本地生成的 C/A 码 (K 星) 进行相关，这样其他卫星的信号就可以通过这个相关的操作去除掉。但是为了能正确的得到对应的可见星，C/A 码必须跟输入的信号对齐，也就是说必须得到正确的码相位。

在与本地生成的伪随机码相乘以后，信号必须再跟本地载波相乘，这样是为了剥离信号中的载波。为了能正确的剥离载波，本地生成的载波频率应尽可能地接近 GPS 信号中的载波频率，前面已经提过，由于多普勒频移的影响，GPS 信号载波频率会在正常频率的  $\pm 10\text{Hz}$  范围内，所以各种不同的频率可能性必须都要去测试，可以设置 500Hz 为步长去测试。

在经过码相位和载波频率的估测之后，最后就是将信号所有分量平方求和产生一个数值，如果该值超过了预期的阈值，就可以定义为该星可见。如图 2.8 所示，可见星有一个明显的峰值，这显示了很好的相关性，也就是证明该星为可见。

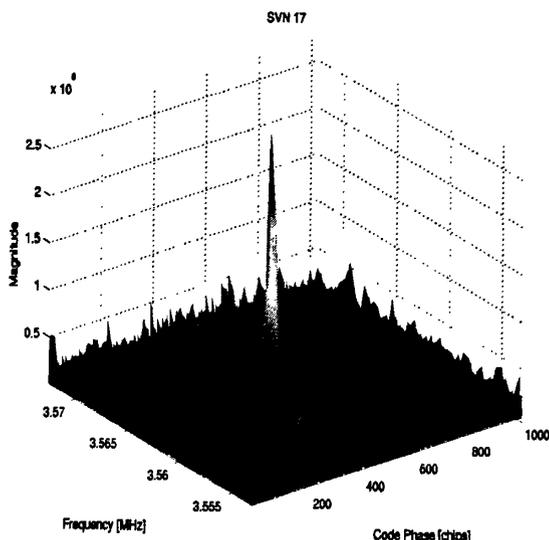


图 2.8 17 号可见星捕获结果

### 2.3.2 GPS 信号跟踪

GPS 信号捕获得到码相位和载波频率的粗值之后,随着时间的变化,为了能顺利跟踪这些已捕获到的 GPS 信号,跟踪环就被采用了,码相位的精确度直接影响到后面的伪距计算,跟踪主要包含了以下两个部分:

**码跟踪:**码跟踪一般是用 DLL 环来实现的, DLL 中一般有三路本地码来与输入的 GPS 信号进行相关,它们分别是超前、当前、滞后。一般采用半个码元为间距。

**载波跟踪/相位跟踪:**跟踪的另一部分则是载波跟踪,一般通过 2 个方法完成,通过跟踪信号的相位或者载波的频率。这也是跟踪环路中最具有疑惑的地方,因为按照理论跟踪中需要确定三个参数,但是在实际反馈过程中,只修改 2 个参数。

跟踪是为了能跟上随着时间变化的 GPS 信号,如果出现失锁,那么需要重新对该星进行捕获。如何运用 FLL 辅助 PLL 跟踪将在后章节介绍。

## 2.4 小结

本章概述了 GPS 信号的产生机理,并且详细的用数学推导了整个扩频、调制的过程,在接收端主要从接收机前端和 GPS 的后序基带信号处理两方面来介绍概述,同时用理论公式推导了下变频、采样等前端的关键技术,最后阐述了捕获、跟踪的整体思路与结构。

## 第三章 GPS 信号捕获算法及改进

### 3.1 GPS 捕获算法

由第二章内容介绍可得, GPS 卫星是通过 32 个不同的 PRN (伪随机码) 来区别的, 捕获的目的是为了确定某一时刻的可见星, 并且得到该星的两个参数: 码相位、载波频率。

本文实验中主要采用了东方联星公司的采样前端, 通过前端的采样和滤波之后, 得到的信号可以表示如下:

$$S_l = \sqrt{P_c} D(t_l - \tau) x(t_l - \tau) \cos[2\pi(f_{IF} + f_D)t_l + \delta\theta]$$

$$t_l = lT_s, l = 0, 1, 2, 3, \dots \quad (3-1)$$

由式 (3-1) 可知, 通过捕获和跟踪对参数  $(\tau, f_D, \delta\theta)$  进行估计, 而在信号的捕获阶段需解决的是参数  $(\tau, f_D)$  粗略值。

#### 3.1.1 中频采样器

本文中的实验采用了北京东方联星公司的中频信号采样器 NewStar210A 作为 GPS 信号采集前端, 通过 USB2.0 把数据上传至 PC 机进行信号处理。由前面介绍得 C/A 码带宽只有 2M 左右, 根据香农定律, 只要高于 4M 采样即可。此前端的中频采样率为 16.367MHz, 数字中频为 4.123MHz。采样时通常采用 1 位或 2 位进行量化。低的量化位数节省数据缓存, 降低传输数据负担, 减小处理的复杂度, 而对 GPS 信号处理的性能的影响却不大。此前端采用 2 位量化, 获得的 GPS 中频数字信号以二进制格式存储, 每个字节含有四个采样点, 第一个采样点放在最高两位(Bit7, Bit6)存储顺序如下所示<sup>[25]</sup>:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Sign	Mag	Sign	Mag	Sign	Mag	Sign	Mag
第一个采样点		第二个采样点		第三个采样点		第四个采样点	

其中 Sign 为符号位, 0 表示正值, 1 表示负值; Mag 为幅度位, 0 代表 1, 1 代表 3。因此在 NewStar210A 原始数据中, “00”代表+1, “01”代表+3, “10”代表-1, “11”代表-3。这样将信号分成了 4 个级别。

#### 3.1.2 二维捕获

在捕获的过程中, 有两个不确定量  $(\tau, f_D)$ , 也就是在两个变量对应的范围内找到匹配的确参数, 在实际编程过程中, 可以先解调后解扩, 也可以先解扩后解调, 也就是估计变量  $\tau$  和  $f_D$  先后顺序可以互换。

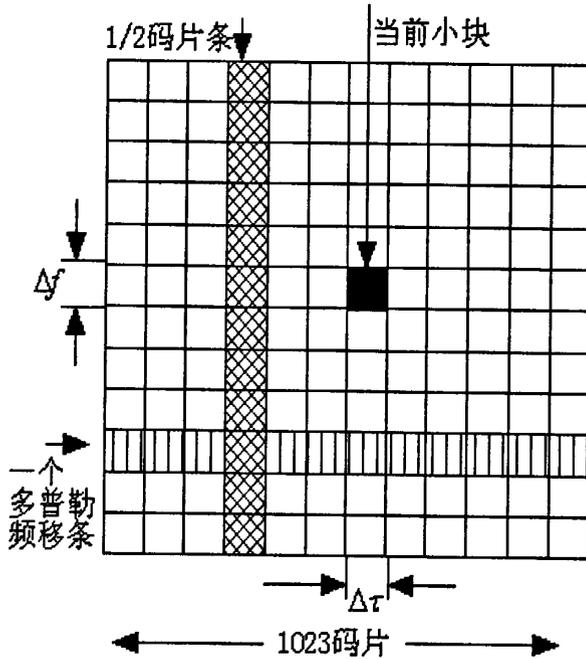


图 3.1 二维捕获示意图

如图 3.1 所示在二维捕获中，主要涉及到两个步长：

$\Delta\tau$  表示 C/A 码域的搜索步长。由前面介绍的 C/A 码的自相关特性可以得到，当输入的 C/A 码信号和本地产生的 C/A 码信号相差 1 个码元或更多码元时，则两个信号不相关；如果这两个信号在 1 个码元之内，则部分相关；如果两个信号完全对准，则可以得到最大的相关值。一般要求这两个信号保持在 0.5 个周期之内。所以一般 C/A 码的最大搜索步长为 1/2 码元，理论上在 1023 个码元范围依次搜索。在本文中，将 1023 位 C/A 码扩展到了 16367 位(前端的采样频率决定)。则码元范围变成了 16367 位。

$\Delta f$  表示估计多普勒频率偏移域里的搜索步长。它与捕获使用的信号长度紧密有关。如果电文长度为 1ms，一个 1kHz 的信号将在 1ms 内变换一个周期，为了保持 1ms 内的最大频率相离在 0.5 个周期，最大步进频率为 1kHz。这样，输入信号与本地复现信号之间最大的频率间隔必在 500Hz 之内。但是使用信号长度为 10ms，最大采用 100Hz 的步进频率，既步进频率间隔与信号长度成反向关系。故本系统采用 1kHz 步进频率，在  $L1=1575.42\text{MHz}$  的左右 -5kHz 到 +5kHz 范围搜索。

### 3.1.3 串行捕获

串行捕获一般用在 CDMA 系统中，GPS 也是一个 CDMA 系统。如图 3.2 所示。串行捕获算法是通过先与本地 PRN 码乘积然后再与本地载波乘积，其中本地 PRN 码有一定的码相位偏移，一般是 0-1023 个码元（本实验中采样率为 16.367Mhz，1ms 数据有 16367 个采样点，这样需要考虑 0-16367 个偏移，数据量更大），在与本地载波频率相差的时候，采用了 I 和 Q 两路（Q 路在 I 路基础上加了  $90^\circ$  相位）。

I、Q 两路都是积分 1ms，这取决于 C/A 码的周期，理想状况下，信号能量应该在 I 路，因为在 GPS 信号调制的时候，C/A 码调制在余弦载波中。然后，一般 I 路信号不与解调的 I 对应，这是因为输入信号的相位不明确（式(3-1)中的  $\delta\theta$ ），所以为了能检测到该信号，需要在正交的 I、Q 两路同时进行，最后将两路的相关值的能量相加，如果该能量值超过预期的阈值，表示已被捕获。

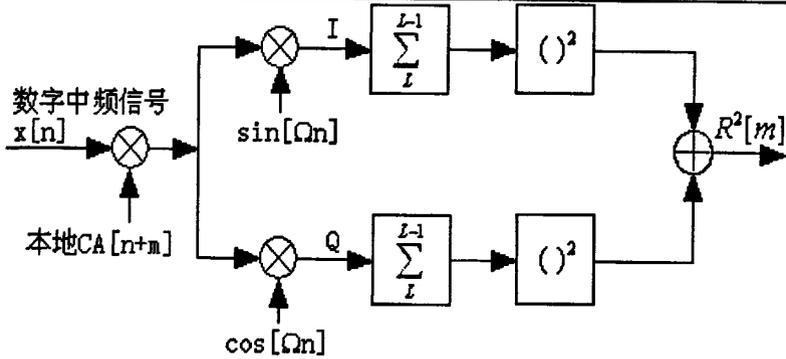


图 3.2 串行捕获结构图

串行捕获主要是进行二维扫描：基于步长 1kHz 的±10kHz 的频率扫描和 1023 个码相位扫描。综合起来的复杂度如下：

$$1023 * (2 * \frac{10000}{1000} + 1) = 1023 * 21 = 21483$$

很明显，这是个运算量非常大的算法，所以这种方法的耗时性成为了弱点。在硬件接收机一般采用多个相关器并行处理，所以此方法适用于硬件实现。此算法采用非相干相关器。

### 3.1.4 并行捕获

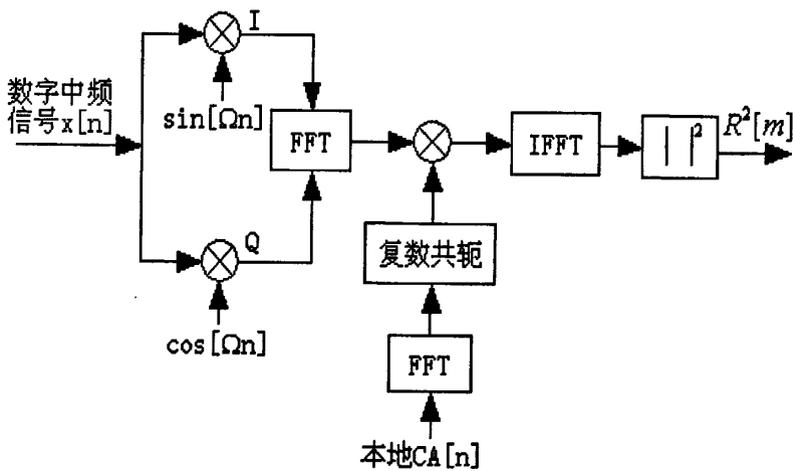


图 3.3 并行捕获结构图

并行捕获里面同样可以细分为：频率并行、码相位并行。这取决于 FFT 和 IFFT 用在哪个环节。本实验中采用了码相位并行。如图 3.3 所示，并行捕获区别于串行捕获重要一点就是引入了循环卷积，这样可以通过简单的 FFT 和 IFFT 来完成 0-1023 种所有本地 PRN 码与输入信号的相关。

设有两长度为 N 的有限长度序列 x(n) 和 y(n)，其离散傅立叶变换为：

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad Y(k) = \sum_{n=0}^{N-1} y(n)e^{-j2\pi kn/N} \quad (3-2)$$

x(n) 与 y(n) 的互相关函数表示如下：

$$z(n) = \frac{1}{N} \sum_{m=0}^{N-1} x(m)y(m+n) = \frac{1}{N} \sum_{m=0}^{N-1} x(-m)y(m-n) \quad (3-3)$$

接下来将系数  $\frac{1}{N}$  省去。

$z(n)$  的  $N$  点离散傅立叶变换可以表示如下：

$$\begin{aligned} Z(k) &= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(-m)y(m-n)e^{-j2\pi kn/N} \\ &= \sum_{m=0}^{N-1} x(m)e^{j2\pi km/N} \sum_{n=0}^{N-1} y(m+n)e^{-j2\pi k(m+n)/N} \\ &= X^*(k)Y(k) \quad (\text{其中 } X^*(k) \text{ 是 } X(k) \text{ 的共轭}) \end{aligned} \quad (3-4)$$

该方法大大减小了算法的循环次数，其循环次数就是  $(2 * \frac{10000}{1000} + 1) = 21$ 。但是前提条件是离散傅立叶算法也就是 DFT 算法必须速度很快<sup>[26]</sup>，不然实时性也很差。

### 3.1.5 模型理论推导

串行、并行捕获模型已讨论，现在从理论上证明该模型的可行性：  
I 路的本地载波：

$$\cos[2\pi(f_{IF} + \hat{f}_D)t_1 + \hat{\theta}] \quad (3-5)$$

Q 路的本地载波：

$$\sin[2\pi(f_{IF} + \hat{f}_D)t_1 + \hat{\theta}] \quad (3-6)$$

通过低通滤波器后（积分），I、Q 两路得到的信号：

$$\begin{aligned} &\sqrt{P_c}D(t_1 - \tau)x(t_1 - \tau) \cos(2\pi\Delta f_D t_1 + \Delta\theta) \\ &\sqrt{P_c}D(t_1 - \tau)x(t_1 - \tau) \sin(2\pi\Delta f_D t_1 + \Delta\theta) \\ &\Delta f_D = f_D - \hat{f}_D \quad \Delta\theta = \delta\theta - \hat{\theta} \end{aligned} \quad (3-7)$$

式(3-7)也可以称为去载波，因为最后得到的信号不再是原先调制的载波，得到的载波频率变为  $\Delta f_D$ ，也就是本地估计的多普勒值与真实值的相差值，接下来就可以清楚的看到捕获可以得到很好的多普勒估值而不能得到很好的相位估计值  $\Delta\theta$ 。

接下来在通过码剥离后，得到一下信号：

$$\begin{aligned} S_I &= \frac{\sqrt{P_c}D}{L} \sum_{l=1}^L x(t_1 - \tau)x(t_1 - \hat{\tau}) \cos(2\pi\Delta f_D t_1 + \Delta\theta) \\ S_Q &= \frac{\sqrt{P_c}D}{L} \sum_{l=1}^L x(t_1 - \tau)x(t_1 - \hat{\tau}) \sin(2\pi\Delta f_D t_1 + \Delta\theta) \end{aligned} \quad (3-8)$$

这里的数学推导有个前提，在没有出现数据的翻转，也就是  $D$  的值为 1 或者 -1。如果数据位翻转，那么得到的能量就会下降，这样捕获的结果就会出现误差。

最后就是模糊度函数的选择（阈值选择）：

如果把本地生成的载波表示成:

$$\exp\{-j[2\pi(f_{IF} + \hat{f}_D)t_i + \hat{\theta}]\} \quad (3-9)$$

把 I、Q 两路得到的信号写成虚数形式:

$$\sqrt{P_c} D(t_i - \tau) x(t_i - \tau) \exp[j(2\pi\Delta f_D t_i + \Delta\theta)] \quad (3-10)$$

再经过与本地 PRN 码相关后:

$$\begin{aligned} \tilde{S} &= S_L + jS_Q \\ &= \sqrt{P_c} D \exp(j\Delta\theta) \tilde{R}(\Delta\tau, \Delta f_D) \\ \tilde{R}(\Delta\tau, \Delta f_D) &= \frac{1}{L} \sum_{l=1}^L x(t_l - \tau) x(t_l - \hat{\tau}) \exp(j2\pi\Delta f_D t_l) \\ &\approx \frac{1}{T} \int_0^T x(t - \tau) x(t - \hat{\tau}) \exp(j2\pi\Delta f_D t) dt \end{aligned} \quad (3-11)$$

为了得到  $(\Delta\tau, \Delta f_D)$ , 可以通过  $\tilde{S}$  的幅度, 并且要去除两个不能估计的变量  $(D, \Delta\theta)$ , 最后选择下面的模糊度函数:

$$\begin{aligned} |\tilde{S}|^2 &= S_L^2 + S_Q^2 \\ &= P_c |\tilde{R}(\Delta\tau, \Delta f_D)|^2 \end{aligned} \quad (3-11)$$

这样如式(3-11)所示, 通过模糊度函数之后最后就剩下需要的评估的两个变量  $(\Delta\tau, \Delta f_D)$ 。

### 3.2 精细频率估计

由 1ms 数据得到的频率分辨率大约为 1KHz, 这对于跟踪环来说过于粗糙。需要的频率分辨率应当在几十赫兹之内。通常, 跟踪环的带宽仅为几个赫兹。用 DFT 寻找精细频率不太合适, 因为要得到 10Hz 的分辨率, 需要 100ms 的数据 (前面已经论证), 如本实验中采用了 16.367MHz, 1ms 数据有 16367 个采样点, 这样 100ms 数据量过大, FFT 操作耗时太久, 而且耗资源。

通过相对关系可以得到精细频率分辨率。一旦将 C/A 码从输入信号中剥离掉, 输入信号就变成了载波信号 (cw)。如果 1ms 数据中的最高频率分量在 m 时刻为  $X_m(k)$ , k 表示输入信号的频率分量。从 DFT 输出中得到输入信号的初始相位为<sup>[27]</sup>:

$$\theta_m(k) = \tan^{-1} \left( \frac{\text{Im}(X_m(k))}{\text{Re}(X_m(k))} \right) \quad (3-12)$$

Im 和 Re 分别表示实部和虚部。假定在 m 时刻后很短时间的 n 时刻, 1ms 输入信号的 DFT 分量  $X_n(k)$  也是最强分量, 因为输入分量在很短时间内不会变化。N 时刻输入信号的初始相位角和频率分量 k 为:

$$\theta_n(k) = \tan^{-1} \left( \frac{\text{Im}(X_n(k))}{\text{Re}(X_n(k))} \right) \quad (3-13)$$

使用式(3-12)与式(3-13)两个相位角计算精频:

$$f = \frac{\theta_n(k) - \theta_m(k)}{2\pi(n-m)} \quad (3-14)$$

式(3-14)给出了一个比从 DFT 精确的多得频率分辨率。为了保证其值得唯一性， $\theta_n(k) - \theta_m(k)$  得相位差必须小于  $2\pi$ 。如果相位差为最大值  $2\pi$ ，带宽就是  $1/(n-m)$ 。其中， $(n-m)$  是里两组输入信号之间的延时。

如果每毫秒取 DFT 的第  $k$  个分量，他们之间相隔 1kHz，精频测量的不确定宽度也是 1kHz。如图 3.4(a)给出了 5 个频率分量，他们间隔 1kHz。如果输入信号如图 3.4(b)落在两个频率分量之间，由于系统噪声的影响，将不能确定其相位。

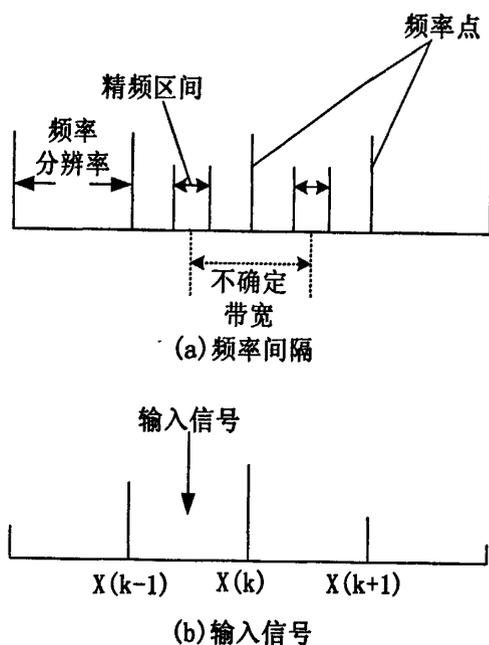


图 3.4 精频捕获的不确定带宽

为了去除这种不确定性，有两种解决方法。第一个是加速 DFT 的操作过程。如果每 0.5ms 执行一次 DFT 操作，不确定带宽是 2kHz。在频率分辨率为 1kHz 和不确定带宽为 2kHz 的条件下，精频测量就不存在不确定性了。但这种方法使 DFT 操作增加一倍。第二种方法是，当输入连续信号时，采用一种幅值比较方案。如图 3.4(b)所示，输入信号落在两个频率点之间，假定  $X(k)$  的幅值略高于  $X(k-1)$ ，将  $X(k)$  代入上式，得到精细频率。输入信号频率与  $X(k)$  相差近 500Hz。正确的结果是输入频率比  $X(k)$  低 500Hz，由于噪声的存在，可能错误地认为输入比  $X(k)$  高了 500Hz。如果输入最高频率点是  $X(k)$ ，且相位计算在不确定带宽内，则这个区域接近于  $X(k)$  与  $X(k-1)$  的中心或者  $X(k)$  与  $X(k+1)$  的中心，这时，如果  $X(k-1)$  大于  $X(k+1)$ ，输入频率就低于  $X(k)$ ，否则，认为输入信号频率高于  $X(k)$ 。如此，精频的准确性由相位决定，但频率差值的符号有最高频率分量邻近的两个频率分量的幅值决定<sup>[27]</sup>。

然而，考虑到两组连续输入信号可能由于导航电文引起了  $180^\circ$  的相位偏移，这样，输入信号就不能看作是连续信号了。对 1ms 的输入信号，当频移为  $\pm 250\text{Hz}$  时，由上式得其相应的相位角变化为  $\pm\pi/2$ 。假设频率偏移 250Hz，其相应的相位角为  $\pi/2$ ，但由于导航数据导致了相位的翻转，相位变为  $-\pi/2$  ( $\pi/2 - \pi$ )，相当于频率偏移了 -250Hz。假设不考虑导航数据导致的相位翻转，结果将有 500Hz 的误差。

为了避免这个问题, 频率不确定的最大值必须小于 250Hz。由前人的实验经验, 本系统最大频移设为  $\pm 200\text{Hz}$ , 其相应的相位角差为  $\pm 2\pi/5$ 。若无相位翻转, 则相位角差在  $2\pi/5$  之内; 如有相位翻转, 则产生相位角差绝对值为  $3\pi/5$  ( $|\pm(2\pi/5) \mp \pi|$ ), 这个值比  $2\pi/5$  大得多。

如果这样, 减去  $\pi$  就可以保证频率差小于  $2\pi/5$ , 得到正确的相位差。为了保证最大频移在 200Hz 以内,  $X(k-1)$ ,  $X(k)$ ,  $X(k+1)$  的频率点间隔最大必须为 400Hz。

最后由上两式将  $X(k)$  的实部和虚部转化为相位角, 通常相位角范围在  $\pm\pi$  之间, 两个相位角的角度差可以是  $0\sim 2\pi$  间的任意值。由于 200kHz 允许的最大相位差是  $2\pi/5$ , 这个角度差必须等于或小于  $2\pi/5$ 。如果结果大于  $2\pi/5$ , 加上或减去  $2\pi$ , 其绝对值就小于  $2\pi/5$  了。如果将噪声考虑进去,  $2\pi/5$  的门限值可以适当放宽, 如用  $2.3\pi/5$ , 这样处理后还大于门限值, 就意味着两个连续毫秒数据间有相位偏移了, 须从结果中减去  $\pi$ 。当然, 得到的结果还要过加减  $2\pi$ , 以保证最后得到的值小于门限。采用精捕获算法后测得的多普勒频率偏移得到很大提高, 可以顺利进入跟踪环。

### 3.3 捕获算法编程实现

本文中主要用 Matlab 和 C++ 实现了捕获算法, 编程思路与图 3.3 的思路略有变化, 也就是前面提过的解扩与解调的顺序选择问题, 在编程过程中, 采用了同时进行, 也就是先把本地载波乘以本地 C/A 码, 然后再与输入 GPS 信号进行卷积操作。整体的思路如下:

#### 1: 初始化

初始化包括了: 本地 C/A 码初始化, 将原先 1023 位的采样成 16367 位, 得到  $32*16367$  二维数组 gold。

初始化数字中频频率:  $f_{IF}=4.123968\text{MHz}$ 。中频采样频率为:  $163.367667\text{MHz}$ 。(由前端的具体参数而设置)。

#### 2: 转化数据

因为前端得到的采样点的存储方式为:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Sign	Mag	Sign	Mag	Sign	Mag	Sign	Mag
第一个采样点		第二个采样点		第三个采样点		第四个采样点	

根据“00”代表+1, “01”代表+3, “10”代表-1, “11”代表-3 原则将 1 字节转换成对应的 4 个采样点得到新的数据 data.dat。

3: 读入 6ms 数据存放在一个数组中。

4: 设置多普勒频移步长

$\Delta f=1\text{KHz}$ , 搜索范围为  $\pm 10\text{KHz}$ , 总共有 21 种值。

5: 将 C/A 码乘以本地载波得到本地信号, 将本地信号与输入 GPS 信号进行卷积运算, 得到高峰值, 并求出高峰值与旁边剩下的所有值的平均值的比值, 判断是否超过预设的阈值。改变多普勒频移, 循环至 21 种多普勒值结束。

6: 针对以捕获的星进行精频捕获。

现分别用 Matlab 与 C++ 对前端采集的数据进行捕获运算编程得, 如图 3.5, 图 3.6 所示。

```

C:\>
输入字节数: 30000

*****捕获到的卫星参数*****
星号-----码相位-----载波频率
2星:          16291          4125795
4星:          4274          4123401
5星:          357           4124447
10星:         7524          4126039
13星:         6419          4122391
17星:         7008          4121292
30星:         8976          4124515
*****
    
```

图 3.5 C++计算结果

```

** Channel PRN Frequency Doppler Exact CA Offset Status **
*****
** 1 2 4.125796e+006 1.827611e+003 76 P **
** 2 4 4.123401e+006 -5.669822e+002 12093 P **
** 3 5 4.123534e+006 -4.338817e+002 16010 P **
** 4 10 4.126039e+006 2.071368e+003 8843 P **
** 5 13 4.122641e+006 -1.326866e+003 9948 P **
** 6 17 4.121292e+006 -2.675922e+003 9359 P **
** 7 30 4.124516e+006 5.477565e+002 7391 P **
    
```

图 3.6 Matlab 结果

仔细将图 3.5, 图 3.6 对比可以发现(注意码相位只要相加为 16367 或者相同, 都可以表示参数一样), 有两个卫星得到的最终结果不同, 5 号星与 13 号星, 不通的地方出现在载波频率, 这是因为在捕获算法过程中, C++中定义的变量格式与 Matlab 不同, 而且其实这 2 颗星的信号信噪比底, 在后面的跟踪就没有跟踪上, 也表示了虚捕获星。

### 3.4 捕获算法的改进及实验结果

以上介绍了捕获算法的原理结构, 但是在实际的运用中, 需要综合多种情况来计划对应的方法, 如 FFT 模块的编程, 数据量的优化等, 需要根据具体的实际情况来具体优化或者改动。

#### 3.4.1 补零计算

本文中的实验采用了北京东方联星公司的中频信号采样器 NewStar210A 作为 GPS 信号采集前端, 此前端的中频采样率为 16.367MHz, 1ms 有 16367 个采样点, 不是基于 2 的 N 次方或 4 的 N 次方, 在用 C++程序的时候采用了麻省理工的 FFT 函数 DLL 动态连接库, 这样就可以不用考虑采样点的具体数目, 因为该库可以很快的计算出任意多点的 FFT 结果, 但是在将算法进行移植, 如 DSP、FPGA 移植时, 就要考虑 FFT 算法的实现问题, 虽然 DSP 有自己的 FFT 核, 但是前提条件是采样点的总数基于 2 的 N 次方或 4 的 N 次方, 下面在 matlab 中对补零计算进行验证, 进行补 0 操作, 将 16367 个采样点加上 17 个 0 点, 变成 16384。

根据上面捕获结果, 图 3.7 表示了补零前先对可见星 10 号星进行捕获结果 (右图), 和补零后的捕获结果 (左图)。

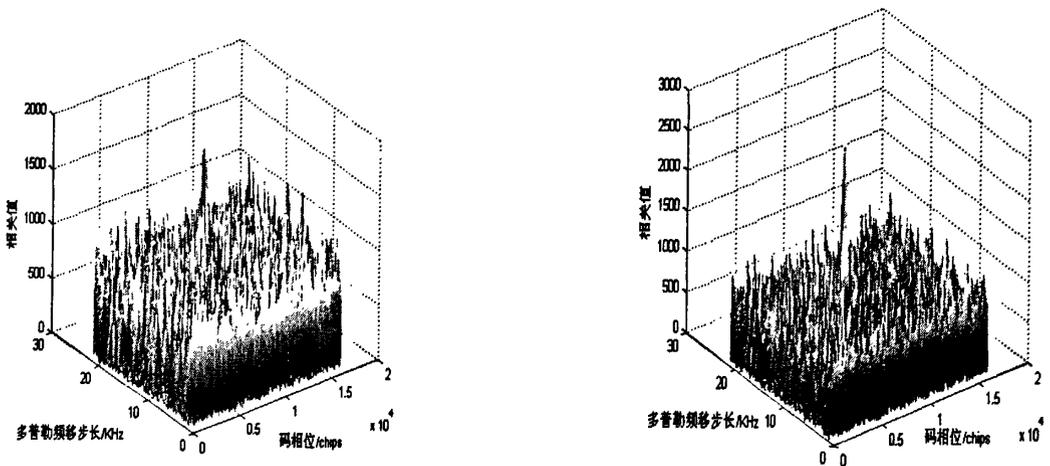


图 3.7 10 号星捕获结果 (补零)

虽然，峰值下降（相关值与旁边值的平均值比值下降），但是最后得到了一样的参数：码相位  $\Delta\tau=7524$ ，载波频率  $f=4126039\text{Hz}$

对可见星依然可以明显的捕获到，但是捕获的结果码相位跟补零前的有 0~17 个采样点误差（基于具体的实验），如图 3.8 对可见星 2 的补零结果：

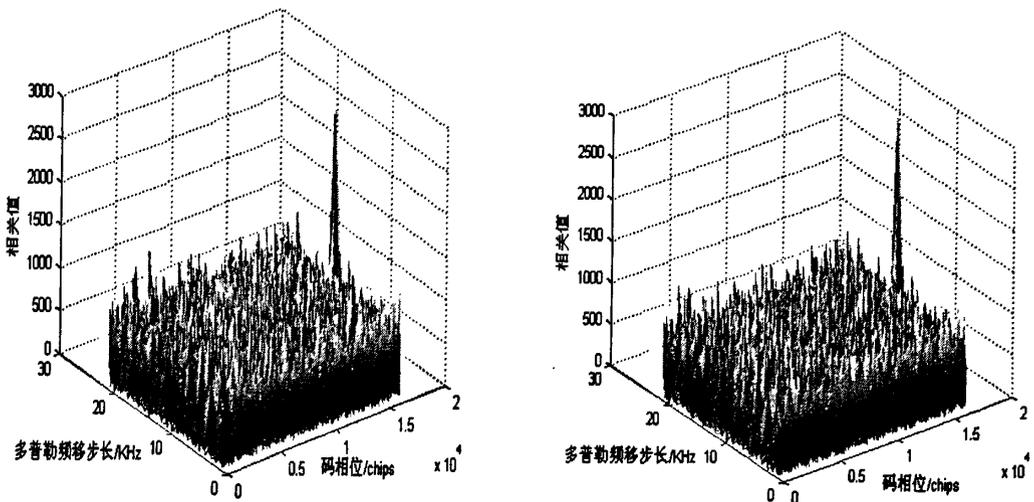


图 3.8 2 号星捕获结果 (补零)

2 号可见星的补零结果(左)与补零前(右)不一样，补零前： $\Delta\tau=16291$ ， $f=4125796\text{Hz}$ 。

而补零后的结果是： $\Delta\tau=16308$ ， $f=4125796$ 。可见，码相位相差了 17 个采样点，这是允许范围内的误差。

该方法的优点是它弥补了采样点数不基于 2 的 N 次方或 4 的 N 次方这一瑕疵，这样为以后移植调用现成 FFT 算法库打下基础。

### 3.4.2 循环相关改进

循环相关是并行捕获算法的核心和优于传统串行捕获算法之处，循环相关改进主要是基于信号特征：

特征 1: GPS 输入信号为实数, 这样在频谱上是对称的, 也就是将输入信号 1ms 数据进行 FFT 变换得, 可见在频谱上是完全对称的关系, 如图 3.9 所示:

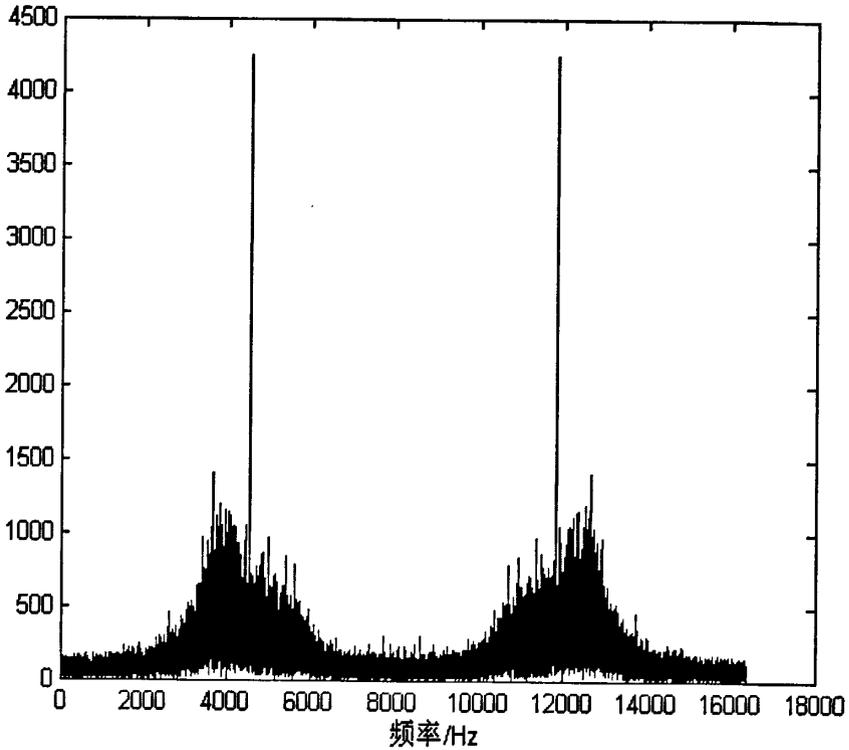


图 3.9 GPS 输入信号频谱分析 (FFT 变换)

特征 2: 除了 GPS 输入信号的特点, 本地信号 (采用 2 号卫星进行示例分析) 的频谱也具有一个重要的特点, 可见信号的信息大部分处于前半频段内。如图 3.10 所示。

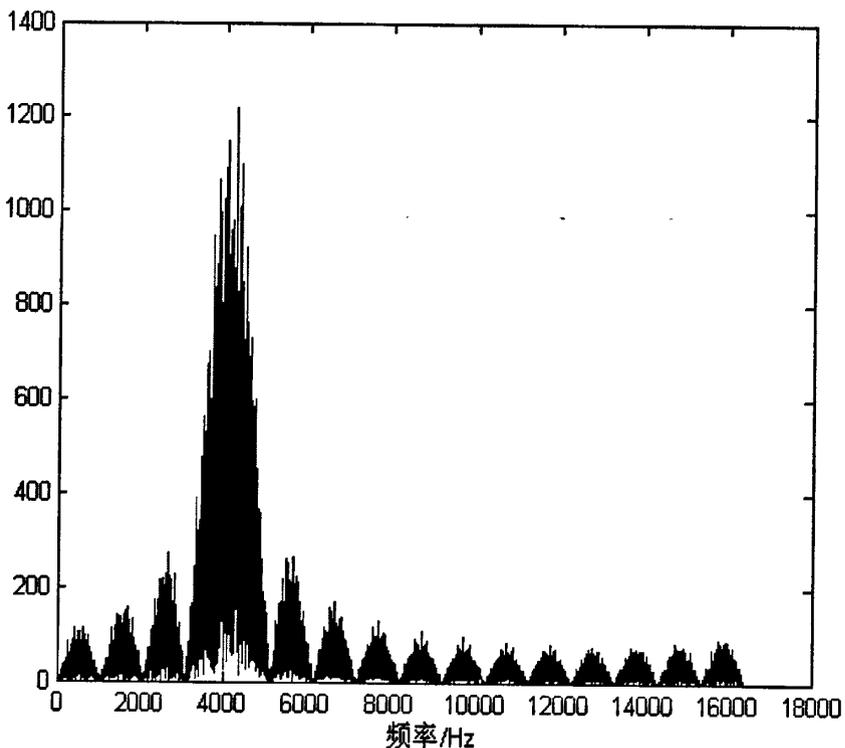


图 3.10 本地信号频谱分析(FFT 变换)

有了上述的两个重要特点, 现在将对整个程序进行改进:

- 1: 对 1msGPS 输入数据  $x(n)$  进行 FFT 运算得到对应的  $X(k)$ ,  $n=0\sim 16367$ .
  - 2: 在  $X(k)$  中取前半得  $X(k)$ ,  $k=0\sim 8184$ , 取复共轭, 得到  $X(k)^*$ .
  - 3: 在本地生成本地信号的时候, 需经过 21 个步长, 生成 21 组  $L_f(n)$  数据, 每组都有 16367 个点.
  - 4: 对一组 16367 采样点 FFT, 转换为频率中的  $L_f(k)$ .
  - 5: 取  $L_f(k)$  前半段, 即:  $L_f(k)$ ,  $k=0\sim 8184$ . 然后将  $L_f(k)$  与  $X(k)^*$  逐点相乘, 得到结果  $R(k)$ ,  $k=0\sim 8184$ .
  - 6: 然后对  $R(k)$ ,  $k=0\sim 8184$  进行 IFFT 操作, 得到时域中的  $r(n)$ . 经过 21 个循环得到  $21\times 8184$  二维数组.
  - 7: 最后进行阈值判断.
- 最后得到的载波频率分辨是 1KHz, 没有改变, 但是得到的码相位的分辨率变成  $\frac{2}{16367}\times 1ms$ , 现对 2 号卫星进行改进的循环捕获, 如图 3.11 所示.

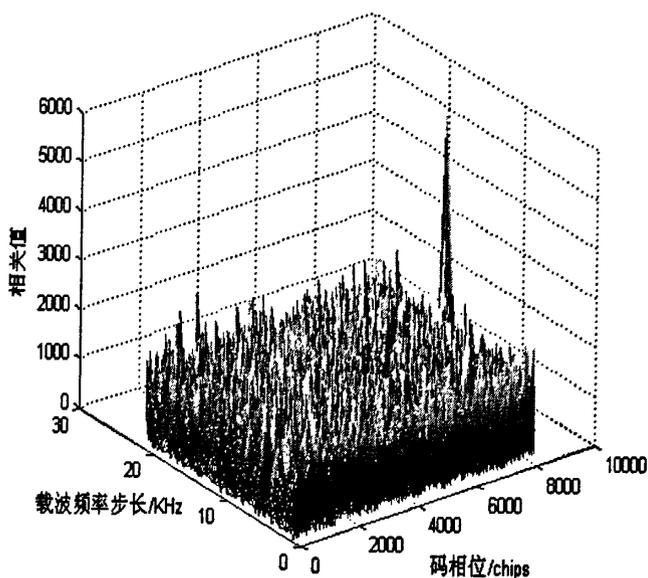


图 3.11 循环相关改进 2 号卫星捕获结果

得到的载波频率为:  $f=4125796\text{Hz}$ ,  $\Delta\tau=8147$ , 将原先  $\Delta\tau=16291$  除以 2 得  $\Delta\tau'=8146$ , 可见误差在理论允许范围之内, 这是由于该方法最后的码相位精度比原先方法低引起的.

该方法的优点是在  $L_f(k)$  与  $X(k)^*$  相乘的时候, 都只用了一半的数据, 而且最后做 IFFT 运算的时候也相应的减少了一半的数据输出, 这样节省了时间. 为以后的算法移植打下基础.

## 3.4.3 降采样

由前面介绍可知,本文中的实验采用了北京东方联星公司的中频信号采样器 NewStar210A 作为 GPS 信号采集前端,此前端的中频采样率为 16.367MHz, 1ms 有 16367 个采样点,在对采样点进行 FFT 计算的时候,可以发现数据量过大,如果在进行算法移植的时候,16367 个采样点会极大影响整个系统的运行速度,甚至出现系统瘫痪等现象。

由于 C/A 码带宽只有 2M 左右,根据香农定律,只要高于 4M 采样即可。所以可以对 16367 个采样点降采样,这样就能减少数据量的大小,现在对 1ms 数据进行 8.1835MHz 进行采样,也就是从第一个数据开始,中间每隔一个数据进行记录下来。把原来 16367 个采样点降采样为 8184 个新采样点,同样当 GPS 输入数据改变的时候,本地生成的信号也要对应降采样,就是把采样时间放大 2 倍。

下面对 2 号可见星进行降采样,结果如图 3.12 所示。

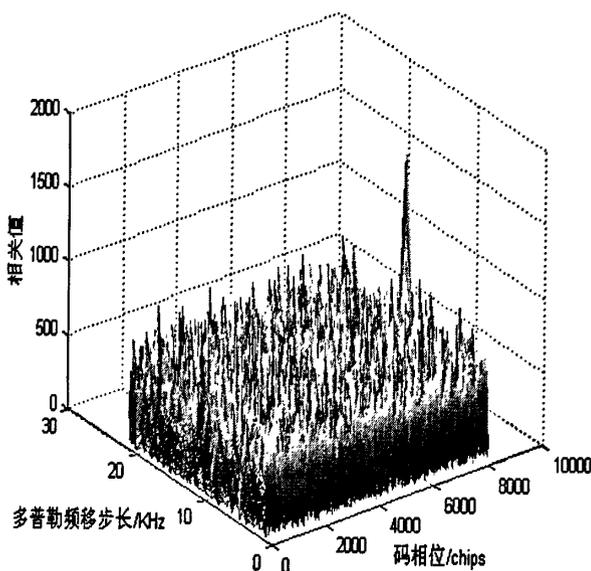


图 3.12 降采样 2 号星捕获结果

最后得到的两个参数大小分别为:  $f=4125796\text{Hz}$ ,  $\Delta\tau=8148$ , 因为采样频率的下降直接

导致码相位精度下降,变成  $\frac{2}{16367} \times 1\text{ms}$ , 将原先  $\Delta\tau=16291$  除以 2 得  $\Delta\tau'=8146$ , 可见误差为  $\frac{4}{16367} \times 1\text{ms}$ 。在误差范围之内。

此方法的优点并不是在于将数据量大大减少,提高了计算速度,更重要的影响在于为以后前端 A/D 转换器的采样频率打下了基础,同时为软件接收机的后序数据处理提供了改变采样点的方法。

## 3.5 小结

本章首先从 GPS 信号机制出发说明了信号捕获过程的作用,主要为了得到参数  $(\tau, f_D)$  粗略值。介绍了实验前端东方联星公司的中频信号采样器 NewStar210A 的数据存储格式,同时从数学理论详细推导了整个捕获模型,以及精细频率的过程,为以后的算法实现与改进打

下基础。

分别用 Matlab 与 C++ 实现了并行捕获算法, 阐述了算法编程实现的框架以及注意点, 并将两者结果进行了对比。最后详细说明了捕获算法的改进, 包括补零计算与循环相关改进, 这些改进方法都是从算法移植的角度进行考虑, 补零计算弥补了采样点数不基于  $2^N$  的缺点, 虽然在得到的载波频率依然没有改变, 但是码相位出现 0~17 个采样点的误差。同样, 循环相关改进减少了相关过程中的数据量, 提高了运行速度, 得到相同的载波频率参数, 但是码相位的精度有所下降, 变为  $\frac{2}{16367} \times 1ms$ , 是原先的 2 倍。最后根据时域采样定理 (香农采样定理), 通过降采样来改变前端的采样率, 大大减少处理数据量, 得到的分辨率为  $\frac{2}{16367} \times 1ms$ 。

同时用实际数据进行了系统的仿真实验。

该章节为以后的跟踪与算法移植从理论上和实际算法编程实现打下了良好基础。

## 第四章 GPS 信号跟踪算法研究

## 4.1 GPS 信号跟踪

本文实验中主要采用了东方联星公司的采样前端，通过前端的采样和滤波之后，得到的信号可以表示如下：

$$S_l = \sqrt{P_c} D(t_l - \tau) x(t_l - \tau) \cos[2\pi(f_{IF} + f_D)t_l + \delta\theta]$$

$$t_l = lT_s, l = 0, 1, 2, 3, \dots$$

(4-1)

GPS 信号捕获提供码相位与载波频率即参数  $(\tau, f_D)$  粗略值，GPS 信号跟踪的目的是为了得到更精确地得到这些参数，继续跟踪可见星并解调出导航电文。

跟踪信号的基本方法是根据具体的输入信号构造一个窄带滤波器，当输入信号的频率随时间变化时，滤波器的中心频率必须跟随信号而对应的变化，这样才能动态的得到具体的参数来剥离载波与随机码得到最后的导航电文。在实际的跟踪过程中，窄带滤波器的中心频率是固定的，但是本地信号要随着输入信号频率的变化而变化。通过鉴相器对输入信号与本地信号的相位进行比较，鉴相器的输出结果连接到一个窄波滤波器。由于跟踪电路的宽带很窄，与捕获方法相比，灵敏度相对要高。

跟踪一个 GPS 信号一般需要两个跟踪环，一个环是用来载波频率，称为载波环；另一个用来跟踪 C/A 码，称为码环。分别用锁相环 (PLL) 和延迟锁定环 (DLL) 来完成。DLL 使码相位的初始相位更加精确，并跟踪未来的变化。为此，DLL 生成了卫星发射的 PRN 码的复制码，并使它和接收码对准。PLL 作为载波跟踪环，不但保持本地信号频率与载波频率一致，而且同时也对载波相位进行估计。为此它必须调整复制信号的频率和相位，使之与输入信号的载波相位一致。同样，FLL 是为了使多普勒频率的初始值更加精确，并跟踪未来的变化，一般用于高动态。

## 4.1.1 线性模型

在对 GPS 信号跟踪进行分析的时候，首先要基于数学模型的理论推导，现将 DLL、PLL、FLL 抽象成一个控制模型，如图 4.1 所示：

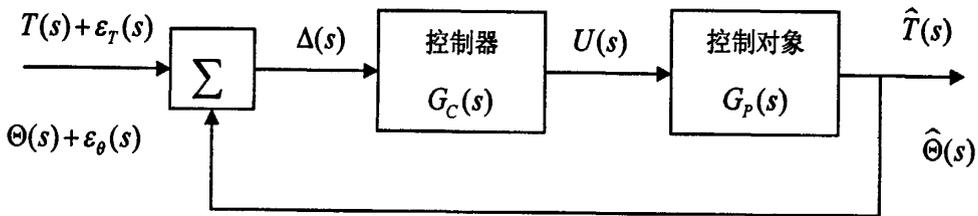


图 4.1 控制器模型

其中， $T(s) = L\{\tau(t)\}$ ， $\Theta(s) = L\{\theta(t)\}$ ， $\varepsilon_T(s) = L\{\varepsilon_T(t)\}$ ， $\varepsilon_\Theta(s) = L\{\varepsilon_\Theta(t)\}$ 。 $T(s)$  为码延迟

$\tau(t)$  的拉普拉斯变换,  $\Theta(s)$  为载波相位的变换,  $\varepsilon_T(s)$  和  $\varepsilon_\Theta(s)$  分别为码延迟和载波相位测量相关的噪声项。

如 PLL 中, 复制码的相位就是  $\hat{\Theta}(s)$ 。它检测复制码的相位与接收机信号相位之差  $\Delta(s)$ 。产生控制量  $U(s)$ , 传给 NCO 来减小该误差。

由经典控制理论<sup>[28]</sup>可知:

$$\frac{\hat{T}(s)}{T(s)} = H(s) = \frac{G_C(s)G_P(s)}{1 + G_C(s)G_P(s)} \quad (4-2)$$

$H(s)$  为该 DLL 的闭环转换函数, 同样也可以得到  $\Delta(s)$  的转换函数:

$$\begin{aligned} \Delta(s) &= T(s) - \hat{T}(s) \\ &= T(s) \left(1 - \frac{\hat{T}(s)}{T(s)}\right) \end{aligned} \quad (4-3)$$

$$\begin{aligned} \frac{\Delta(s)}{T(s)} &= 1 - H(s) \\ &= \frac{1}{1 + G_C(s)G_P(s)} \end{aligned} \quad (4-4)$$

这样, 通过式(4-2), 式(4-4)能够从控制理论去分析跟踪环路, 如阻尼系数、带宽的选择等。

## 4.2 锁相环 (PLL)

锁相环的目的是为了调整本地振荡器的频率使之与输入信号频率相匹配, 表面上违背了 PLL 的初衷, PLL 理论上是用调整载波相位的, 但是在实际中 PLL 还要调整载波频率, 下面将一一分析。

### 4.2.1 PLL 系统模型

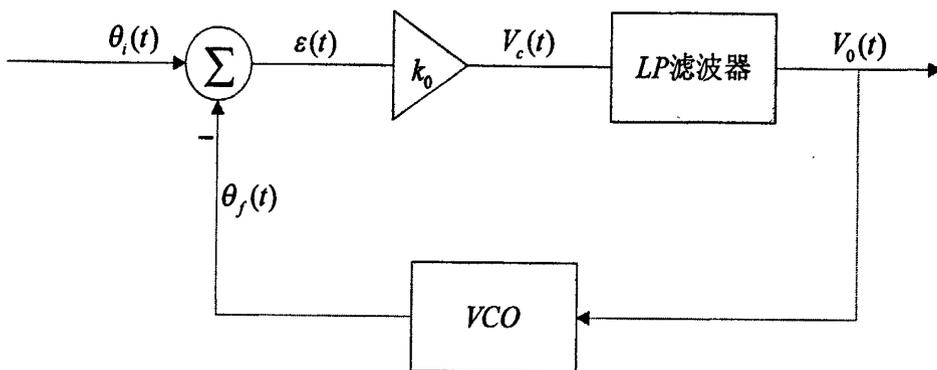


图 4.2 时域系统结构

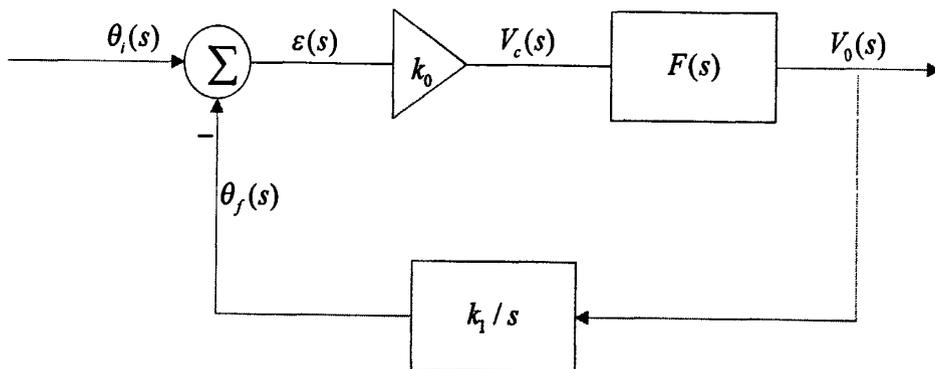


图 4.3 S 域系统结构

图 4.2 是锁相环的时域结构，而图 4.3 则是对应的频域结构。输入信号是  $\theta_i(t)$ ，压控振荡器 (VCO) 输出为  $\theta_f(t)$ 。 $\Sigma$  充当了鉴相器的功能，鉴相器的增益放大为  $k_0$ ，低通滤波器为了限制噪声。

VCO 时域上模型，输入电压  $V_0(t)$  控制输出频率<sup>[27]</sup>：

$$w_{out}(t) = w_0 + k_1 u(t) v_0 \quad (4-5)$$

式中， $w_0$  是 VCO 中心频率， $k_1$  为 VCO 增益， $u(t)$  是单位阶跃函数

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t > 0 \end{cases} \quad (4-6)$$

对式 (4-5) 两边积分得到

$$\begin{aligned} \int_0^t w_{out}(t) dt &= w_0 t + \theta_f(t) = w_0 t + \int_0^t k_1 u(t) v_0 dt \\ \theta_f(t) &= \int_0^t k_1 u(t) v_0 dt \end{aligned} \quad (4-7)$$

$\theta_f(t)$  拉普拉斯变换为

$$\theta_f(s) = V_0(s) \frac{k_1}{s} \quad (4-8)$$

由式 (4-8) 得到 VCO 的频域模型  $\frac{k_1}{s}$ 。根据经典控制理论可得

$$H(s) = \frac{\theta_f(s)}{\theta_i(s)} = \frac{k_0 k_1 F(s)}{s + k_0 k_1 F(s)} \quad (4-9)$$

误差转移函数为

$$H_e(s) = 1 - H(s) = \frac{s}{s + k_0 k_1 F(s)} \quad (4-10)$$

在本文中，采用二阶锁相环，锁相环的阶数主要由滤波器决定的，现取滤波器为

$$F(s) = \frac{\tau_2 s + 1}{\tau_1 s} \quad (4-11)$$

将式 (4-11) 代入式 (4-9), 转移函数变为

$$H(s) = \frac{\frac{k_0 k_1 \tau_2 s}{\tau_1} + \frac{k_0 k_1}{\tau_1}}{s^2 + \frac{k_0 k_1 \tau_2 s}{\tau_1} + \frac{k_0 k_1}{\tau_1}} \equiv \frac{2\xi w_n s + w_n^2}{s^2 + 2\xi w_n s + w_n^2} \quad (4-12)$$

由式 (4-12) 可得固有频率  $w_n = \sqrt{\frac{k_0 k_1}{\tau_1}}$ , 阻尼因数  $\xi = \frac{w_n \tau_2}{2}$ 。同时, 可以将 s 域的结构图

细化如图 4.4。

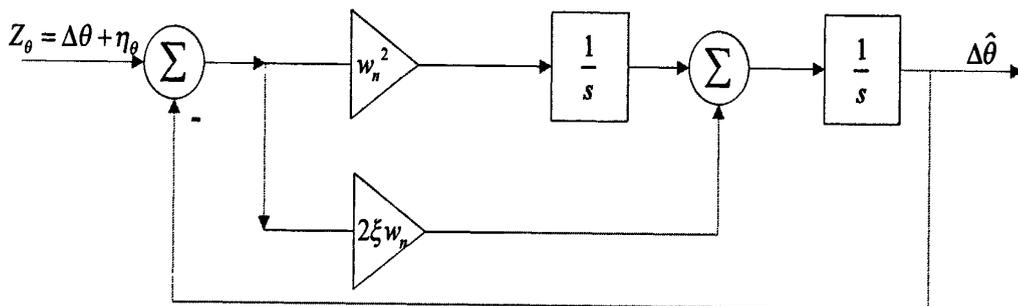


图 4.4 频域系统结构

现输入单位阶跃函数  $\theta_i(s) = \frac{1}{s}$  与调频信号  $\theta_i(s) = \frac{\Delta w}{s^2}$  得最后的稳态误差都是 0, 这意味着二阶环路能跟踪调频信号, 使鉴相器特征值回到零。

频域讨论完了后, 下面介绍怎么将该模型应用到具体的实际当中, 在实际中, 信号并非连续信号, 一般都是经过采样得到的离散信号, 这样, 整个系统必须向离散系统转换。

大部分参考书通过双线性变换, 使连续的 s 域模型转换为离散的 z 域模型

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}, \quad T \text{ 为采样间隔时间} \quad (4-13)$$

现在先直接从结构图 4.4 将 s 域模型转换为离散的 z 域模型如图 4.5,

$$Z_{\theta, m+1} = \Delta \theta_{m+1} + \eta_{\theta, m+1}$$

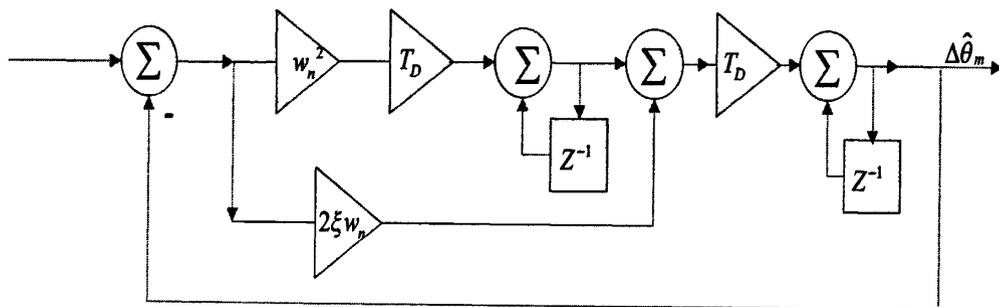


图 4.5 离散系统结构

将图 4.4 与图 4.5 相比可以知道, 只是将积分器  $\frac{1}{s}$  用离散积分器替代了。

现将图 4.5 简化成图 4.6

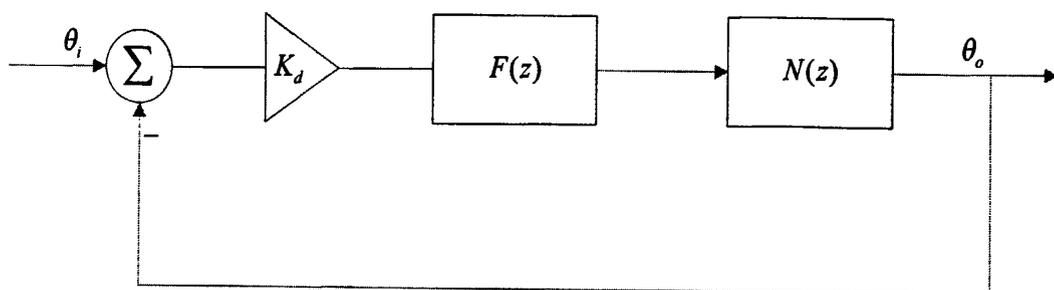


图 4.6 离散线性简化系统

将图 4.5 与 4.6 比较并假设

$$F(z) = \frac{(C_1 + C_2) - C_1 z^{-1}}{1 - z^{-1}} \quad (4-14)$$

$$N(z) = \frac{k_0 z^{-1}}{1 - z^{-1}} \quad (4-15)$$

$F(z)$  具体机构如下图 4.4 所示:

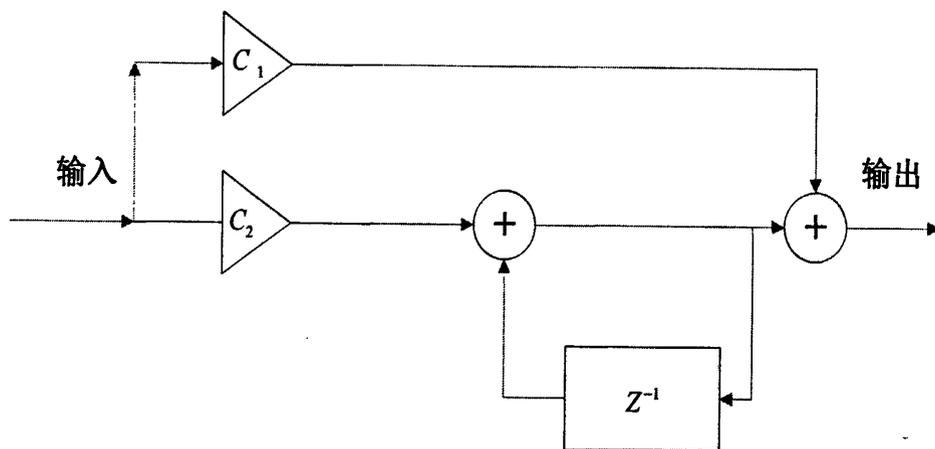


图 4.7 环路滤波器

将式 (4-13) 代入式 (4-12) 得

$$H_1(z) = \frac{(4\xi w_n T + (w_n T)^2) + 2(w_n T)^2 z^{-1} + ((w_n T)^2 - 4\xi w_n T) z^{-2}}{(4 + 4\xi w_n T + (w_n T)^2) + (2(w_n T)^2 - 8) z^{-1} + (4 - 4\xi w_n T + (w_n T)^2) z^{-2}} \quad (4-16)$$

由图 4.6 得

$$\begin{aligned} H(z) &= \frac{K_d F(z) N(z)}{1 + K_d F(z) N(z)} \\ &= \frac{K_d K_0 (C_1 + C_2) z^{-1} - K_0 K_d C_1 z^{-2}}{1 + (K_0 K_d (C_1 + C_2) - 2) z^{-1} + (1 - K_0 K_d C_1) z^{-2}} \end{aligned} \quad (4-17)$$

将式 (4-16) 与 (4-17) 对比可得

$$C_1 = \frac{1}{K_0 K_d} \frac{8\xi w_n T}{4 + 4\xi w_n T + (w_n T)^2} \quad (4-18)$$

$$C_2 = \frac{1}{K_0 K_d} \frac{4(w_n T)^2}{4 + 4\xi w_n T + (w_n T)^2} \quad (4-19)$$

式(4-18)与式(4-19)是关键公式,为编程实现的时候提供了参数修改理论依据。式中, $K_0 K_d$ 为环路增益, $\xi$ 是阻尼比, $w_n$ 是自然频率。关于参数选择下面将分析。

#### 4.2.2 等效噪声宽度(Noise Equivalent Bandwidth)

由帕塞瓦尔定理得<sup>[27]</sup>

$$\varepsilon_{L,U} = 2 \int_{f_L}^{f_U} |X(f)|^2 df \quad (4-20)$$

式(4-20)该结果对所有实 $x(t)$ 都成立, $|X(f)|^2$ 称为能量谱密度。由4.2.1节可知,PLL可以模拟成一个线性系统,则

$$Y(f) = H(f)X(f) \quad (4-21)$$

将式(4-21)两边平方可得

$$\begin{aligned} Y(f)^2 &= X(f)^2 \\ S_Y(f) &= |H(f)|^2 S_X(f) \end{aligned} \quad (4-22)$$

$S_Y(f)$ 为输出信号的频谱密度,同样 $S_X(f)$ 为输入信号频谱密度。

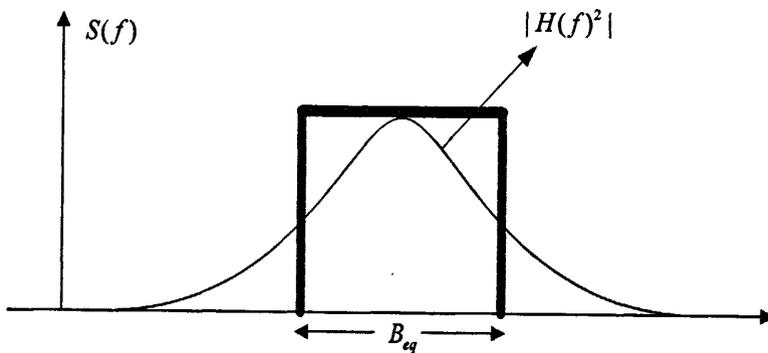


图 4.8 等效噪声宽带

PLL的一个重要特性就是噪声等效宽带,如图4.8所示,噪声等效宽带为boxcar滤波器的宽带,该滤波器与研究中的滤波器可以让等量的噪声通过。通常,噪声等价带宽为

$$B_{eq} = \frac{1}{2|H(j0)|^2} \int_0^{\infty} |H(j2\pi f)|^2 df \quad (4-23)$$

将公式(4-23)应用到公式(4-12)得噪声等效带宽为

$$\begin{aligned} B_{eq} &= \frac{(4\xi^2 + 1)w_n}{8\xi} \\ w_n &= \frac{8\xi B_{eq}}{4\xi^2 + 1} \end{aligned} \quad (4-24)$$

这样, 就可以通过选择等效噪声带宽  $B_{eq}$  与阻尼比  $\xi$  来确定  $w_n$ , 再根据 (4-18), (4-19) 可以确定  $C_1$  与  $C_2$ 。

### 4.2.3 PLL 鉴相器

鉴相器在整个跟踪环路中是最难设计的一部分, 它主要起到了对应误差的对比输出, 由 3.1.5 节可知输入信号经过本地载波、本地 C/A 码相关之后得

$$\begin{aligned}\tilde{S} &= S_L + jS_Q \\ &= \sqrt{P_C} D \exp(j\Delta\theta) \tilde{R}(\Delta\tau, \Delta f_D) \\ \tilde{R}(\Delta\tau, \Delta f_D) &= \frac{1}{L} \sum_{l=1}^L x(t_l - \tau) x(t_l - \hat{\tau}) \exp(j2\pi\Delta f_D t_l) \\ &\approx \frac{1}{T} \int_0^T x(t - \tau) x(t - \hat{\tau}) \exp(j2\pi\Delta f_D t) dt\end{aligned}\quad (4-25)$$

其中, 复数相关值的实部来自于同相相关器, 虚部来自于正交相关器, 在  $t = nT_s$  时相关器的输出记为

$$\begin{aligned}\tilde{Z}_n &= \tilde{S}_n + \tilde{\eta}_n \\ &= S_{I,n} + jS_{Q,n} + \eta_{I,n} + j\eta_{Q,n} \\ Z_{I,n} &= \text{Re}\{\tilde{Z}_n\} \\ &= S_{I,n} + \eta_{I,n} \\ Z_{Q,n} &= \text{Im}\{\tilde{Z}_n\} \\ &= S_{Q,n} + \eta_{Q,n}\end{aligned}$$

同样, 超前相关结果为

$$\begin{aligned}\tilde{Z}_E &= \tilde{S}_E + \tilde{\eta}_E \\ \tilde{S}_E &= \sqrt{P_C} D \exp(j\Delta\theta) \tilde{R}(\Delta\tau - T_c/2, \Delta f_D)\end{aligned}\quad (4-26)$$

滞后相关结果为

$$\begin{aligned}\tilde{Z}_L &= \tilde{S}_L + \tilde{\eta}_L \\ \tilde{S}_L &= \sqrt{P_C} D \exp(j\Delta\theta) \tilde{R}(\Delta\tau + T_c/2, \Delta f_D)\end{aligned}\quad (4-27)$$

鉴相器是为了能突显待估参数, 又能同时抑制其他未知参数的影响。PLL 鉴相器必须对载波相位误差  $\Delta\theta$  非常敏感, 但对其他未知参数如数据位 D 不敏感。这样, 可以选择如式(4-29)鉴相器。

$$\begin{aligned}
\tilde{Z}_\theta &= \operatorname{Re}\{\tilde{Z}_P\} \operatorname{Im}\{\tilde{Z}_P\} \\
&= Z_{I,P} Z_{Q,P} \\
&= S_{I,P} S_{Q,P} + \tilde{\eta}_\theta \\
&= P_C R^2(\Delta\tau) \frac{\sin 2\Delta\theta}{2} + \tilde{\eta}_\theta \\
&\approx \tilde{Z}_\theta(\Delta\theta = \Delta\tau = 0) + \left. \frac{\delta \tilde{Z}_\theta}{\delta \Delta\theta} \right|_{\Delta\theta=0} \Delta\theta + \tilde{\eta}_\theta \\
&= P_C \Delta\theta + \tilde{\eta}_\theta
\end{aligned} \tag{4-28}$$

但是, 在实际编程过程中, 采用了著名的 Costas 环路, 其中采用的 PLL 鉴相器为

$$\tilde{Z}_\theta = \frac{\operatorname{Im}\{\tilde{Z}_P\}}{\operatorname{Re}\{\tilde{Z}_P\}} \tag{4-29}$$

因为当码相位对齐的时候, I 路出来的信号简单表示为

$$D^k(n) \cos(w_i n) \cos(w_i n + \varphi) = \frac{1}{2} D^k(n) \cos(\varphi) + \frac{1}{2} D^k(n) \cos(2w_i n \varphi) \tag{4-30}$$

其中  $\varphi$  为输入信号与本地信号的相位差, 同样在 Q 路乘积得到的信号可以简单表示为

$$D^k(n) \cos(w_i n) \sin(w_i n + \varphi) = \frac{1}{2} D^k(n) \sin(\varphi) + \frac{1}{2} D^k(n) \sin(2w_i n \varphi) \tag{4-31}$$

当这两路信号经过积分和累加器(等效于低通滤波器), 滤除近似  $2w_i$  的第二项信号, 得到

$$I^k = \frac{1}{2} D^k(n) \cos(\varphi) \tag{4-32}$$

$$Q^k = \frac{1}{2} D^k(n) \sin(\varphi) \tag{4-33}$$

将式 (4-33) 与式 (4-32) 相比可得

$$\frac{Q^k}{I^k} = \frac{D^k(n) \sin(\varphi)}{D^k(n) \cos(\varphi)} = \tan(\varphi) \tag{4-34}$$

则  $\Delta\theta = \tan^{-1}\left(\frac{\operatorname{Im}\{\tilde{Z}_P\}}{\operatorname{Re}\{\tilde{Z}_P\}}\right)$ , 对数据位的翻转  $D$  不敏感。

#### 4.2.4 PLL 编程实现

在用 Matlab 或者 C++ 编程实现的时候, 难点主要有两个:

1. 参数  $C_1$  与  $C_2$  的确定, 在编程过程中, 前 30ms 数据选择  $\xi = 0.900, B_{eq} = 150$ , 30ms 以后选择  $\xi = 0.700, B_{eq} = 15$ , 这样做的目的是为了一开始能在比较大的带宽下跟踪上对应目标卫星, 然后再缩小跟踪带宽, 使精度变高。但是由公式 (4-18) 与公式 (4-19) 可知, 为了能确定  $C_1$  与  $C_2$ , 还需要解决系数  $\frac{1}{K_0 K_d}$  的大小, 这也是靠具体实验来确定。

2. 反馈方法，从PLL鉴相器得到载波相位误差后，怎么去反馈到本地载波发生器中，这是比较困难的一点，理论上得到的相位误差应该直接反馈到载波相位上，但实际编程的时候，先是通过得到的相位误差去反馈到载波频率，然后再通过载波频率去修改下一个毫秒的相位。它正是用了频率跟相位的关系，相位的导数是频率，频率的积分是相位。这样，只要用一个比例系数将相位误差映射（求导的过程）成频率误差（系数  $\frac{1}{K_0 K_d}$  选择也需考虑这点）。

### 4.3 延迟锁定环 (DLL)

DLL 为延迟锁定环，如图 4.9<sup>[18]</sup>所示，在信号跟踪中采用 DLL 的目的是为了消除超前相关器与滞后相关器的差距，也就是使当前的码相位估计值与实际值对齐。

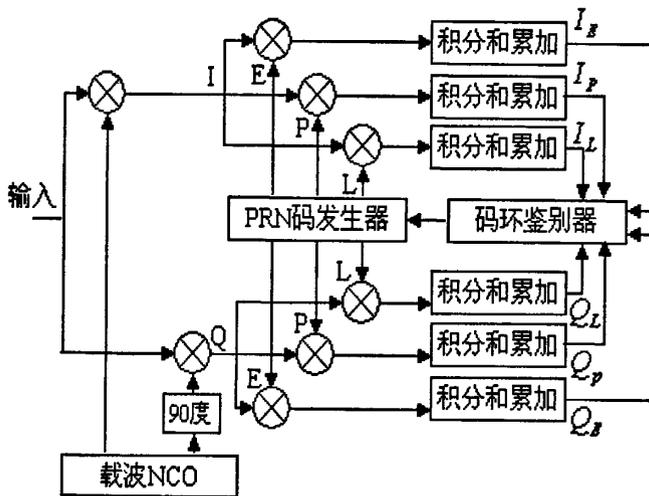


图 4.9 DLL 结构图

#### 4.3.1 DLL 鉴相器的选择

DLL 鉴相器必须对码延迟误差  $\Delta\tau$  非常敏感，但对未知的载波相位和数据位不敏感。这样，可以选择如式 (4-30) 非相干鉴相器。

$$\begin{aligned}
 \tilde{Z}_\tau &= |\tilde{Z}_E|^2 - |\tilde{Z}_L|^2 \\
 &= |\tilde{S}_E|^2 - |\tilde{S}_L|^2 + \tilde{\eta}_\tau \\
 &= P_C R^2 (\Delta\tau - T_C / 2) - P_C R^2 (\Delta\tau + T_C / 2) + \tilde{\eta}_\tau \\
 &\approx \tilde{Z}_\tau (\Delta\tau = 0) + \left. \frac{\delta \tilde{Z}_\tau}{\delta \Delta\tau} \right|_{\Delta\tau=0} \Delta\tau + \tilde{\eta}_\tau \\
 &= 0 + \frac{2P_C}{T_C} \Delta\tau + \tilde{\eta}_\tau \\
 Z_\tau &= \tilde{Z}_\tau \frac{T_C}{2P_C} \approx \Delta\tau + \eta_\tau
 \end{aligned} \tag{4-35}$$

同样在实际中，采用了的具体鉴相器为

$$D = \frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)} \quad (4-36)$$

式 (4-31) 中  $I_E$  为  $I$  路的当前路;  $Q_E$  为  $Q$  路的当前;  $I_L$  为  $I$  路的滞后路;  $Q_L$  为  $Q$  路的滞后路。当然还有其他的鉴相器选择, 在这里不细分析。

#### 4.3.2 DLL 编程实现

DLL 编程实现是三个环路中最容易, 也是最能好理解的环路, 本文中主要将得到的码相位误差与一个预设好的阈值  $k$  进行比较 ( $k$  的选择基于反复实验的验证), 如果当 DLL 输出的码相位值  $\Delta\tau > k$  或者  $\Delta\tau < -k$  时, 就相应地调整本地 C/A 码的位置。

#### 4.4 锁频环 (FLL)

锁频环 (FLL) 为通过复现近似的频率以完成载波剥离过程, 如图 4.10<sup>[18]</sup>, 该结构同时可以使用于 FLL 与 PLL, 两者的区别就是载波环鉴别器的选择不同。所以两者可以统称为载波跟踪环 (CTL)。

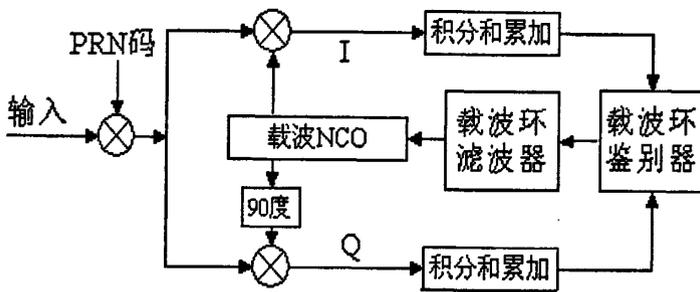


图 4.10 载波跟踪环路

由公式 (3-5) 至式 (3-8) 得  
I 路的本地载波:

$$\cos[2\pi(f_F + \hat{f}_D)t_1 + \hat{\theta}]$$

Q 路的本地载波:

$$\sin[2\pi(f_F + \hat{f}_D)t_1 + \hat{\theta}]$$

通过低通滤波器后 (积分), I、Q 两路得到的信号:

$$\sqrt{P_c} D(t_1 - \tau) x(t_1 - \tau) \cos(2\pi\Delta f_D t_1 + \Delta\theta)$$

$$\sqrt{P_c} D(t_1 - \tau) x(t_1 - \tau) \sin(2\pi\Delta f_D t_1 + \Delta\theta)$$

$$\Delta f_D = f_D - \hat{f}_D \quad \Delta\theta = \delta\theta - \hat{\theta}$$

也就是锁频环使参数  $\Delta f_D = f_D - \hat{f}_D$  接近于零, 典型情况下, 允许输入载波信号相位的翻转。因此有时候锁频环也称为自动频率控制环 (ACF), 但是为了使 GPS 接收机锁频环对 I 和 Q 信号中的  $180^\circ$  翻转不敏感, I 和 Q 信号的采样时间不应该跨越数据比特的翻转, 但是, 在初始信号捕获的时候, 接收机并不明确具体的数据跳变边界在哪里, 在完成比特同步的同

时,一般保持频率锁定比相位锁定更容易,这是因为,锁频环鉴相器对某些 I 和 Q 信号确实跨越了数据比特跳变并不那么敏感。当预测积分时间较小时,受影响的积分区域就会很少。

#### 4.4.1 FLL 鉴相器

同样 FLL 鉴相器的选择,类似与前面 PLL 鉴相器、DLL 鉴相器。要对参数载波频率,而对其余的参数不敏感。同样在 FLL 鉴相器选择的时候,也有几种选择,在本文试验中,按需要选择的鉴相器如式

$$D_{FLL} = \frac{\tan^{-1}\left(\frac{I_{E,1} \times Q_{E,2} - I_{E,2} \times Q_{E,1}}{I_{E,1} \times I_{E,2} + Q_{E,1} \times Q_{E,2}}\right)}{t_2 - t_1} \quad (4-37)$$

其中,  $I_{E,1}, Q_{E,1}$  表示  $t_1$  时刻的采样, 同样  $I_{E,2}, Q_{E,2}$  表示稍后时刻  $t_2$  的采样, 这两者相邻的采样应该在同一个数据比特时间区内。

在码相位对准的时候, 假设  $t_1$  时刻的相位为  $\varphi_1$ ,  $t_2$  时刻的相位为  $\varphi_2$ , 则式 (4-36) 可以表示为

$$\begin{aligned} D_{FLL} &= \frac{\tan^{-1}\left(\frac{\cos(\varphi_1)\sin(\varphi_2) - \cos(\varphi_2)\sin(\varphi_1)}{\cos(\varphi_1)\cos(\varphi_2) + \sin(\varphi_1)\sin(\varphi_2)}\right)}{t_2 - t_1} \\ &= \frac{\varphi_2 - \varphi_1}{t_2 - t_1} \end{aligned} \quad (4-38)$$

再结合式 (3-8), 这样就能得到参数  $\Delta f_D$ 。

只要相邻的 I 和 Q 采样是在同一个数据比特时间内完成的, 那么在 GPS 接收机中, 锁频环鉴相器便不会因数据跳变而引起频率模糊度。但是在高动态环境中, 锁频环可能会锁定在错误的频率上, 因此对于最初的锁频环闭合很短的预测检测积分时间是很重要的。

#### 4.4.2 FLL 编程实现

在编程实现 FLL 鉴相器的时候, 根据 (4-37) 来具体实现, 其中主要注意的地方

1.  $t_1$  时刻对应数据  $I_{E,1}, Q_{E,1}$ ,  $t_2$  时刻对应数据  $I_{E,2}, Q_{E,2}$ 。
2. 还有就是预检测积分时间的选择, 由上面讨论可以知道, 预检测积分时间尽量不要出现数据位跳转的时刻, 同时应该较小, 在本文实验中, 选取  $t_2 - t_1 = 1ms$ 。

3. 通过鉴相器得到的载波频率误差滤波器的选择, 可以参考 PLL 中的滤波器, 因为 PLL, DLL, FLL 三个环路的理论控制系统是一样的。

#### 4.4.2 FLL 与 PLL 结合

由前面讨论可知, FLL 与 PLL 都是载波跟踪, 只是各自鉴相器目标参数不同, 一个是载波频率, 另一个则是载波相位, 两者的联系非常紧密, 就像载波频率与载波相位的关系一样

密切，从某个角度可以认为两个环路是性质一样的两个环路。

在锁频环 (FLL) 中，两个相邻时刻  $t_1$ 、 $t_2$  时刻的  $I, Q$  组成的矢量差直接正比于频率误差的速率旋转，当实际达到真正的实际频率锁定之后，该矢量停止旋转，但是可能停止在与  $I$  轴成任意角的地方。因此对锁频环不可能实现相干的码跟踪，因为其依赖于相位锁定。

在相同的噪声宽带和  $C/N_0$  下，FLL 比 PLL 动态应力性能要好的多。例如在噪声宽带为 10Hz， $C/N_0$  为 dB-Hz 下，FLL 能够适应高达 240g/s 的加加速度，而 PLL 只能适应  $4g/s^{[17]}$ 。

所以在可靠 GPS 设计的时候，在环路初始闭合期间和在高动态应力期间失去相位锁定的时候，用 FLL 做 PLL 的备份<sup>[29]</sup>，当低至中等动态的稳态期间则返回纯 PLL，以产生最高精度的载波多普勒频移相位测量。而其 FLL 鉴相器要求在一个数据跳变间隔内有两个采样点，所以 FLL 最大的预检测积分时间不超过 10ms。

也就是说一般 FLL 其实是充当了辅助作用<sup>[30]</sup>，如图 4.11<sup>[31]</sup>所示，1 阶 FLL 辅助 2 阶 PLL 方案（滤波器部分）。

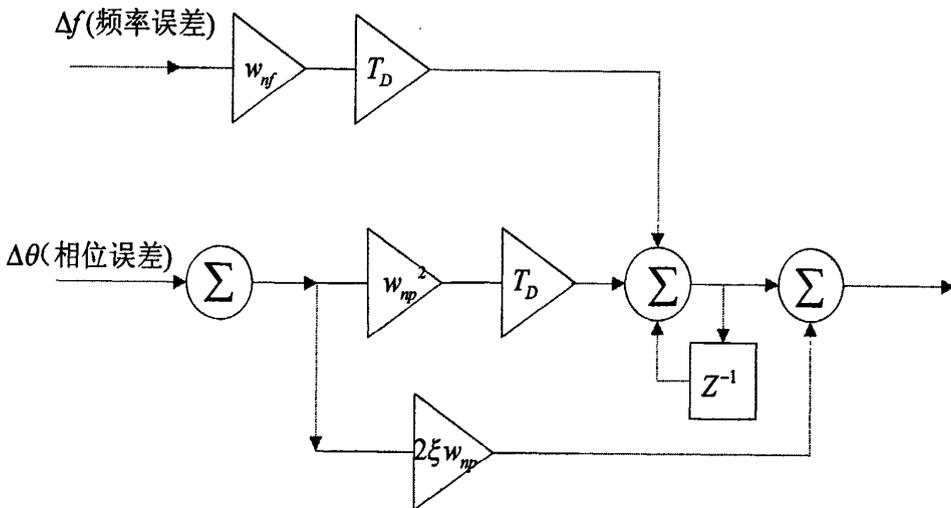


图 4.11 一阶 FLL 辅助二阶 PLL

如果 PLL 的误差变为 0，那么图中的滤波器就变成纯 FLL，相反，如果将 FLL 的误差输入置为 0，则滤波器就变成了纯 PLL，当 FLL 辅助的 PLL，一开始对两个鉴相器的误差输入信号同时加以应用，一直到获得相位锁定之后，再转为纯 PLL，一直到下一次相位锁定丢失。

## 4.5 实验结果分析

本节主要对跟踪环路进行了对比实验分析。

### 4.5.1 纯 PLL 跟踪卫星结果

对 17 可见星跟踪结果（跟踪上），如图 4.12 为 I 路当前输出，图 4.13 为 Q 路输出

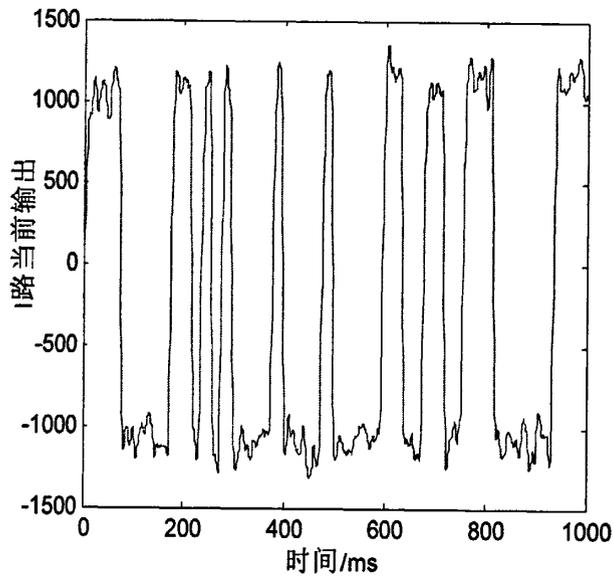


图 4.12 I 路输出

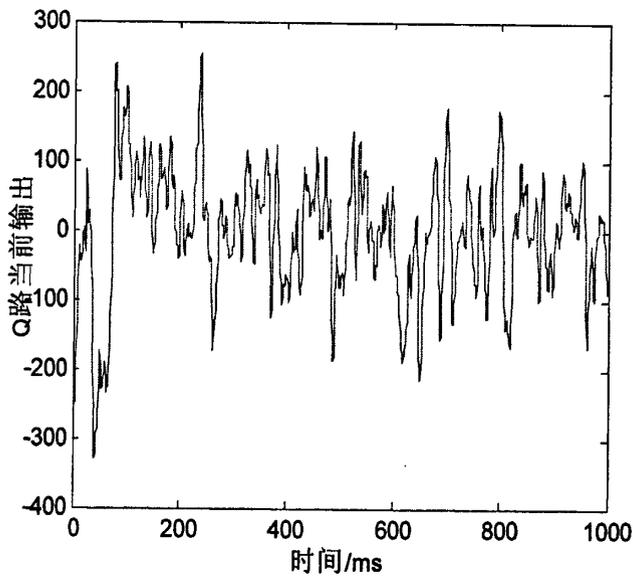


图 4.13 Q 路输出

#### 4.5.2 等效噪声带宽、阻尼比对跟踪结果影响

1. 在整个过程中设置  $\xi = 0.900$ ,  $B_{eq} = 150$  不变, 对 17 号可见星跟踪得到载波频率误差如图 4.14 所示。

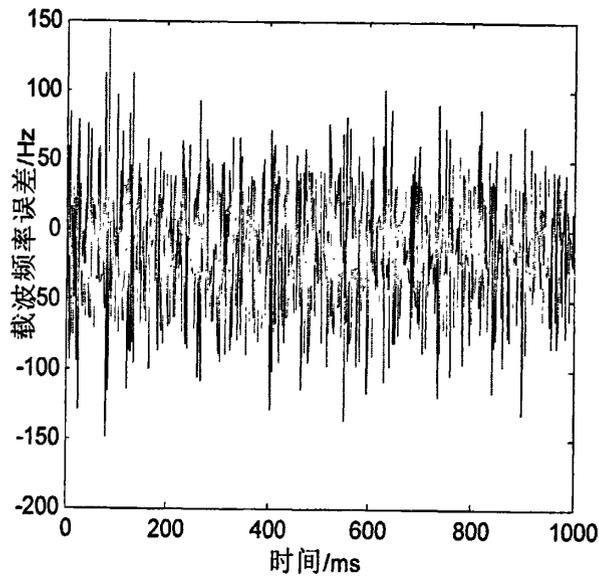


图 4.14 载波频率误差

2.在整个过程中设置 $\xi = 0.700, B_{eq} = 5$ 不变,对 17 号可见星跟踪得到载波频率误差如图 4.15 所示。

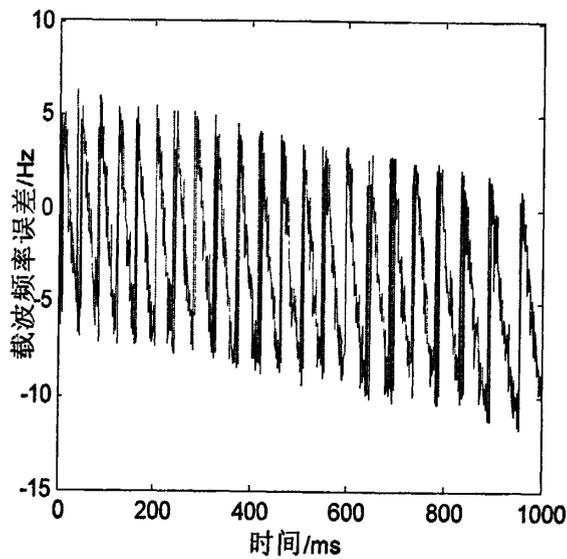


图 4.15 载波频率误差

其实已经失锁,说明过小的带宽将导致最后无法跟踪上目标卫星。

3.设置前 30ms 数据选择 $\xi = 0.900, B_{eq} = 150$ , 30ms 以后设置 $\xi = 0.700, B_{eq} = 15$ ,对 17 号可见星跟踪得到载波频率误差如图 4.16 所示。

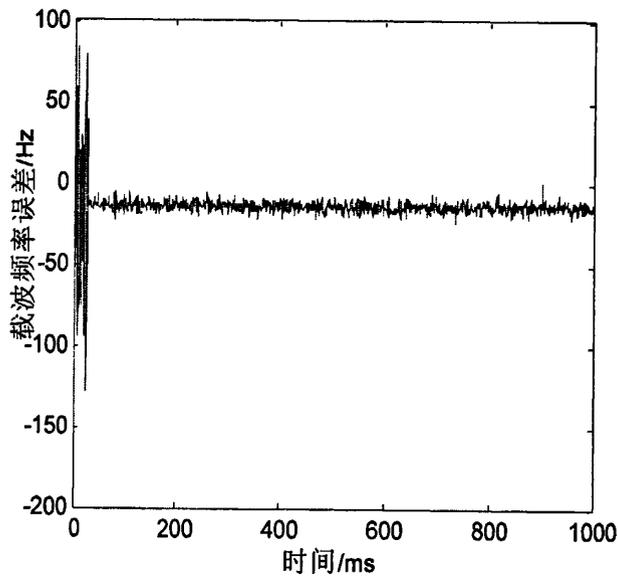


图 4.16 载波频率误差

将图 4.16 与图 4.14 对比可得, 小的等效噪声带宽会使噪声能量变小, 得到明显精度更高的载波频率估计值。但是过小的等效噪声带宽会直接导致失锁, 如图 4.15 所示, 所以等效噪声带宽与最后的持续跟踪需要权衡的。

#### 4.5.3 FLL 辅助 PLL 跟踪实验分析

1. 先用纯 PLL (前 30ms  $\xi = 0.900, B_{eq} = 150$ , 30ms 以后设置  $\xi = 0.700, B_{eq} = 15$ ) 跟踪, 得到参数载波频率误差如图 4.17 所示。

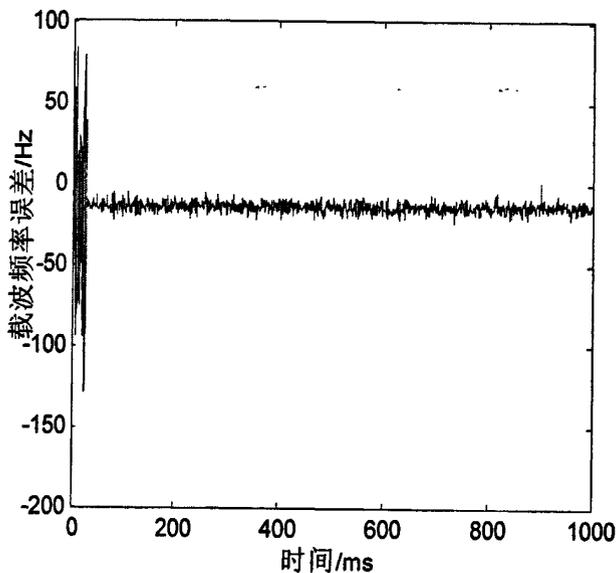


图 4.17 载波频率误差

同样得到的载波相位误差如图 4.18 所示:

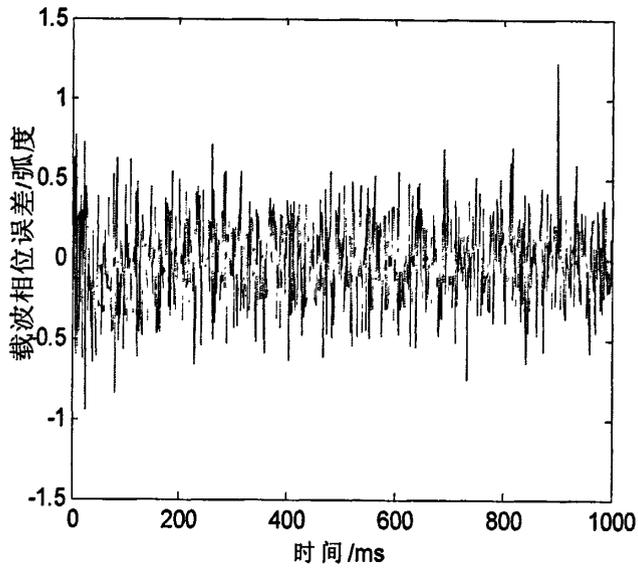


图 4.18 载波相位误差

2.FLL+PLL(1),其中前 30msFLL 单独工作, 30ms 后 PLL 单独工作, 得到的参数载波频率误差如图 4.19 所示。

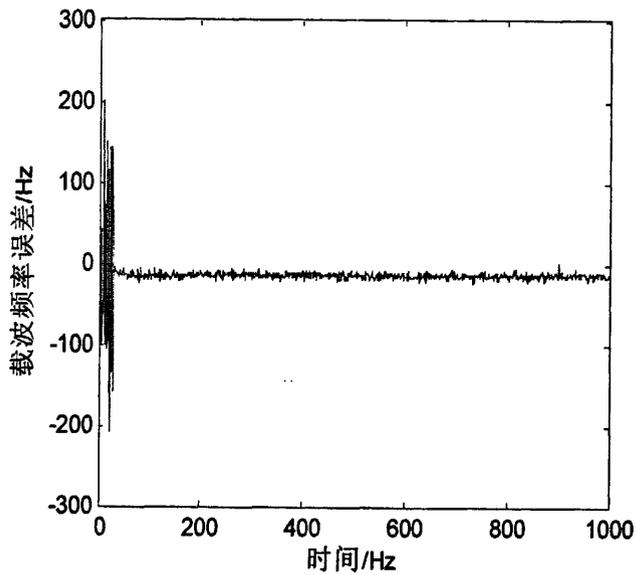


图 4.19 载波频率误差

同样得到的载波相位误差, 如图 4.20 所示。

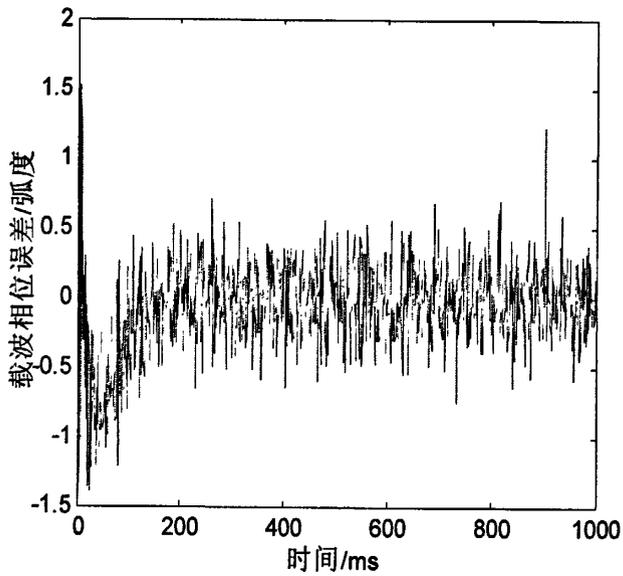


图 4.20 载波相位误差

3.FLL+PLL (2), 其中前 30ms 同时利用 FLL 和 PLL 的信息, 也就是 FLL 与 PLL 同时工作, 后 30ms PLL 单独工作得到的参数载波频率误差如图 4.21 所示。

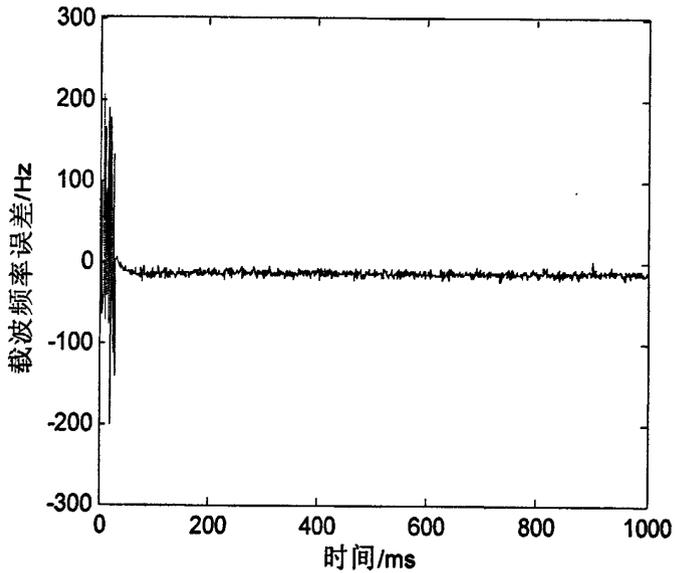


图 4.21 载波频率误差

同样得到的载波相位误差如图 4.22 所示。

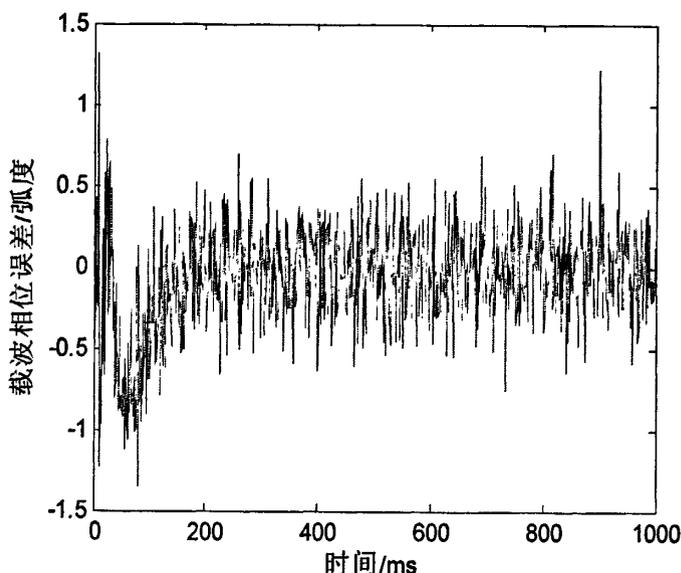


图 4.22 载波相位误差

FLL+PLL 实验结论：通过纯 PLL 工作（前 30ms  $\xi = 0.900, B_{eq} = 150$ ，30ms 以后设置  $\xi = 0.700, B_{eq} = 15$ ）、FLL+PLL（前 30ms FLL 单独工作，30ms 后 PLL 单独工作）、FLL+PLL（前 30ms FLL 与 PLL 同时工作，后 30ms PLL 单独工作）3 中情况，可得到结论是

1. 在本实验数据中，PLL 控制的精度比 FLL 高，纯 PLL 通过得到的相位差反馈载波频率最后得到的精度比较高。
2. 在本实验中，前 30ms FLL 的功能等效于前 30ms 的（ $\xi = 0.900, B_{eq} = 150$ ）PLL，帮助后面窄带宽的 PLL 找到载波频率。因为如果直接上来用精细的 PLL 来跟踪，有可能导致完全失锁，如图 4.15 所示
3. 纯 FLL 一般不能用来全程的跟踪，因为 FLL 最后还要依赖于相位稳定（锁相位）

## 4.6 小结

本章主要详细阐述了 GPS 信号跟踪环的结构，推导了锁相环 PLL、延迟锁定环 DLL、锁频环 FLL 的理论模型及各自鉴相器的选择理论依据。根据整个理论体系，分析了对应编程实现的难点和注意点，最后用具体的实验分析了跟踪环的跟踪结果、等效噪声带宽、阻尼比对跟踪环的影响、FLL 辅助 PLL 跟踪。

在整个章节中，强调突出了本文中跟踪环出现的疑点与难点。

1. 跟踪环整体的架构。由第三章内容，可以知道在跟踪环中，主要是处理（ $\tau, f_D, \delta\theta$ ）三个参数，这样也许对应的会想到分别用三个环路去单独控制这三个参数，但在具体编程实践的过程中，主要用 PLL 和 DLL 去完成跟踪环（高动态另行考虑），其中的原因在 4.5 节中已经分析。

2. PLL 的反馈问题，在 PLL 鉴相器得到载波相位误差后，应该怎么去反馈的问题，理论上应该用载波相位误差去修改本地信号发生器的载波相位，使载波相位误差越来越趋近于零，但在实际编程中并没有去反馈载波相位，真正反馈的是载波频率，利用了载波频率与载波相

位的关系，相位的求导即频率，频率的积分即相位。这样得到的精确度更高。

3. FLL 辅助 PLL 目的。明确 FLL 辅助 PLL 的目的是非常重要的，在本文中，主要是用 FLL 来辅助 PLL 找到载波频率，这样为以后的精度相对高的 PLL（等效噪声带宽低）提供了基础，防止失锁。

最后本章同样细解了各鉴相器选择的思路，为以后的问题思考方式打下了良好的基础。

## 第五章 捕获算法的 FPGA 仿真实现

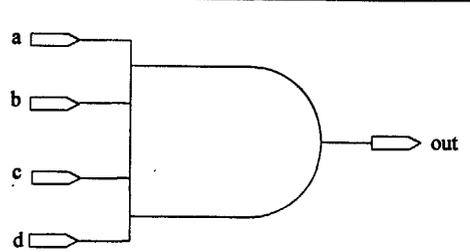
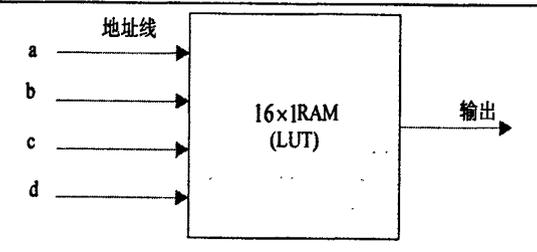
GPS 软件接收机已经被提出十余年之久,目前标准的 GPS 接收机主要基于 ASIC 和快速的微型处理器。ASIC 优点主要在于它的高效率,但是灵活性不强不易重构,修改或改进算法将会非常耗资源。GPS 软件接收机刚好弥补了这一缺点,它具有易重构、适应性强等优点。目前对软件接收机的算法基本都是基于 PC 机在进行研究,对算法的芯片移植,工程化研究甚少。在以 FPGA 为核心器件设计信号处理系统时,使用 XILINUX System Generator for DSP 可以减小 FPGA 开发的难度,对工程师的硬件知识要求下降,它主要用于建立数字信号处理系统的抽象算法,并将抽象的算法转化成可靠的硬件实现。这样为复杂算法的 FPGA 开发提供了良好的平台。本文利用该软件进行了相应的仿真开发。

### 5.1 FPGA 开发平台

FPGA 是 1985 年美国 Xilinx 公司推出的一种采用单元型结构的新型 PLD 器件。Xilinx 把基于查找表(Look Up Table, LUT)技术、SRAM 工艺、要外挂配置用的 EEPROM 的 PLD 称为 FPGA<sup>[10]</sup>。

查找表(Look Up Table)简称 LUT,LUT 本质上就是一个 RAM。目前 Xilinx FPGA 中多采用 4 输入的 LUT 结构,所以每一个 LUT 可以看成是一个有 4 为地址的 $16 \times 1$ 的 RAM。当用户通过原理图或者 HDL 语言描述一个逻辑电路后,PLD/FPGA 开发软件会自动计算逻辑电路的所有可能结果,并把结果事先写入 RAM 中。这样,每输入一个信号进行逻辑运算就等于输入一个地址进行查表,找出地址对应的内容,然后输出结果即可<sup>[32]</sup>。表 5.1 是一个 4 输入与门的例子。

表 5.1 四输入与门逻辑电路的 LUT 实现

实际逻辑电路		LUT 实现方式	
			
a,b,c,d 输入	逻辑输出	地址	RAM 中存储的内容
0000	0	0000	0
0001	0	0001	0
...	0	...	0
1111	1	1111	1

Xilinx FPGA 器件是基于 LUT 结构设计的可编程逻辑芯片。

简化的 FPGA 基本由 6 部分组成:可编程输入/输出单元、基本可编程逻辑单元、嵌入式块 RAM、丰富的布线资源、底层嵌入功能单元和内嵌专用硬核<sup>[32]</sup>。

#### 1. 可编程输入/输出单元

输入/输出简称 I/O 单元,FPGA 的 I/O 单元被设计为可编程模式,即通过软件的灵活设置,可以匹配不同的电气标准与 I/O 物理特性,可以调整匹配阻抗特性、上下拉电阻、可以调整输出驱动电流大小。

## 2. 基本可编程逻辑单元

基本可编程逻辑单元是可编程逻辑的主体,可以根据设计灵活地改变其内部的连接与配置,完成不通的逻辑功能。FPGA 一般基于 SRAM 工艺,其基本可编程单元通常是由查表法(LUT)和寄存器(Register)组成的。Xilinx 可编程逻辑单元叫做 Slice,由上下两部分构成,每个部分都由一个 Register 加一个 LUT 组成。

## 3. 嵌入式块 RAM

目前大多数 FPGA 都有内嵌的块 RAM(Block RAM)。FPGA 内部嵌入可编程 RAM 模块,大大地拓展了 FPGA 的应用范围和使用灵活性。一般可以灵活的配置为单口 RAM、双口 RAM、FIFO 等常用存储器结构。

## 4. 丰富的布线资源

布线资源连通 FPGA 内部所有单元,连线的长度和工艺决定着信号在连线上的驱动能力和传输速度。设计者通常不需要直接选择布线资源,实现过程中一般是由布局布线器自动根据输入的逻辑网表的拓扑结构和约束条件选择可用的布线资源连通所用的底层单元模块,所以设计者通常忽略布线资源。

## 5. 底层嵌入功能单元

主要的功能模块为那些通用程度较高的嵌入式功能模块。比如 PLL、DLL、DSP 和 CPU 等。越来越多的高端 FPGA 产品将包含 DSP 或 CPU 等软处理核,从而 FPGA 将由传统的硬件设计手段逐步过渡到系统级设计工具。如本文试验中用的 Virtex-II Pro 开发板内部集成了 Power PC 450 的 CPU Core。这些 CPU 或者 DSP 处理模块的硬件主要由一些加、乘、快速进位链、Pipelining 和 Mux 等结构组成,加上用逻辑资源和块 RAM 实现的软核部分就组成了功能强大的软计算中心。这种 CPU 或 DSP 比较适合实现 FIR 滤波器、编码解码和 FFT(快速傅立叶变换)等运算。

## 6. 内嵌专用硬核

内嵌专用硬核是相对于前面的“底层嵌入单元”而言的,它主要指那些通用性相对较差,不为大多数 FPGA 器件所包含的硬核(Hard Core)。FPGA 主要有两类,一方面是通用性较强的,价格适中的。另一方面则是针对性较强的,价格比较昂贵的。如为了提高 FPGA 性能,适用高速通信总线与接口标准,很多高端 FPGA 集成了 SERDES(串并收发器)等专用 Hard Core。

本文主要采用 Virtex2p xc2vp30-7ff896 芯片的开发板,如图 5.1 所示。

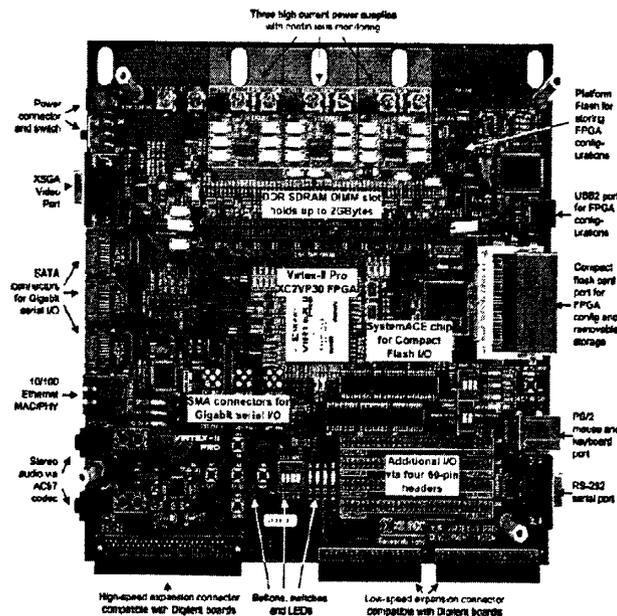


图 5.1 XUP Virtex-II Pro 开发板

XUP Virtex-II Pro 开发板具有非常出色的硬件平台，它具有丰富的外围设备，这样就能很好地体现出 Virtex-II Pro FPGA 芯片的性能优越性。该系列开发板资源如表 5.2 所示

表 5.2 Virtex-II Pro 开发板资源

资源名称	XC2VP30(型号)
Slices	13969
Array Size	80×46
Distribute RAM	428Kb
Multiplier Blocks	136
Block RAMs	2448Kb
DCMs(数字时钟管理器)	8
PowerPC RISC Cores	2
Multi-Gigabit Transceivers	8

## 5.2 基于 System Generator 的 FPGA 开发

System Generator 是一款理想的 FPGA 开发软件，它通过建立数学信号处理系统的系统构架，并将抽象算法转化成可靠的硬件实现，是数字信号处理高层系统与 Xilinx FPGA 实现的“桥梁”。它在 MATLAB/Simulink 的环境下完成算法的建模，然后生成相应的工程。这样可以弥补对 VHDL 或者 Verilog 语言不熟悉的弱点，在 System Generator 生成工程后，后序的开发与 HDL 语言开发基本一样，ISE 可对工程进行仿真、综合、最后完成算法的硬件化，当然也可以直接由 System Generator 直接生成位流文件，并下载到 FPGA<sup>[33]</sup>。

### 5.2.1 FPGA 开发优点

随着 FPGA 的市场推广，研究人员技术的提高，从性能上面，FPGA 可以完全替代当今市场流行的 DSP、ARM 等处理器，采用 FPGA 开发的主要优点有以下几点：

#### 1. 并行性

传统的数字信号处理系统大多用 DSP 处理器完成，这是由于 DSP 处理器有较高的主频以及专用的乘法、加法计算模块等资源，也是相当于一个小型的 CPU。但是 DSP 处理在运行的时候是按指令顺序执行的，且数据位宽也是固定的，因而资源的利用率不高，限制了处理器的数据吞吐量，而且对信号频率也有限制，不可能来处理高于 DSP 本身系统频率的信号。其实 DSP 就相当于小型的单核奔腾 CPU。

但是，用 FPGA 在处理数据时候是并行方式执行的，当然，这根据你具体的配置来完成的，也可以配置成跟 DSP 一样的流水线操作，如果在并行处理数据的时候，相对于前面的流水线式大大提高了数据吞吐量。类似于多 CPU 处理。

#### 2. 灵活性

灵活性是 FPGA 的一个重大特点，这也类似于软件接收机区别与硬件接收机一样，灵活性来自于 FPGA 比较底层，因为在编程的时候直接接触到底层的资源分配与配置问题，而传统的 DSP、ARM 芯片一般基于高级语言 C 开发，越高级的语言虽带来了方便，但是在灵活性方面就逊色于 FPGA，如 FPGA 可以改变具体数据的位宽，这样不仅节省资源而且提高运行速度，但是在传统的芯片开发，就不可以随意更改数据位宽。

## 5.2.2 System Generator 开发

System Generator 开发环境是基于 Matlab 中的 Simulink 平台, Simulink 提供了一个可以创建和仿真动态系统的可视化的环境, 在此环境下可以很方便地进行系统级的设计。而 System Generator 作为一个工具箱包含在 Simulink 中, 包括了 XILINX Blockset、XILINX Reference Blockset 和 XILINX XtremeDSP Kit 三个库, 由 XILINX 公司和 MathWorks 公司合作开发。使用 System Generator 可以很方便的在 Simulink 中搭建系统模型并仿真, 而且根据设计模型生成对应的 ISE 工程及可下载到 FPGA 的位流文件。如图 5.2 所示, 说明了 System Generator 整个系统开发过程<sup>[12]</sup>。

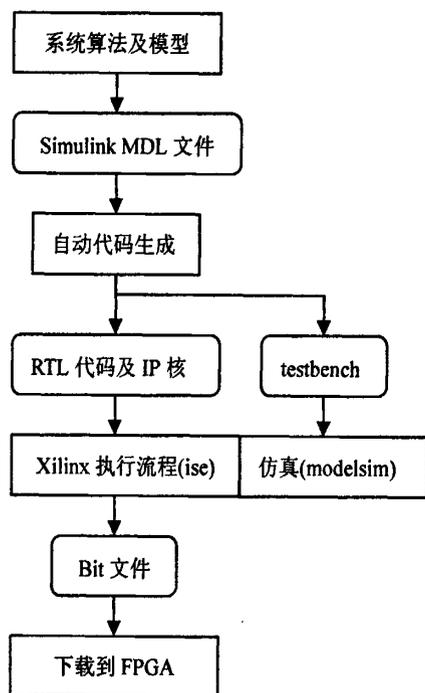


图 5.2 System Generator 开发过程

## 5.2.3 System Generator 模块介绍

XILINX Blockset、XILINX Reference Blockset 和 XILINX XtremeDSP Kit 三个库中包含了许多模块, 这些模块就是 XILINX 模块。System Generator 能够基于设计人员用 HDL 语言开发的功能模块和 XILINX Smart-IP 核完成数字系统的 FPGA 实现。这也许多对 HDL 语言不熟悉的开发着采用 System Generator 软件的原因, 利用现成的模块去搭建自己的数学算法。但是重要的一点是只有 XILINX 模块才会被在 FPGA 硬件中实现, 也就是任何其他模块, 如 Simulink 中本身自带的模块不可以被翻译成最后的 Bit 文件(位流文件)。

在本文实验中, 主要设计的模块有 System Generator、Gateway 模块、FFT 模块、虚数乘法模块(编程实现)、数据源模块(非 XILINX 模块)。

## 一. System Generator 模块

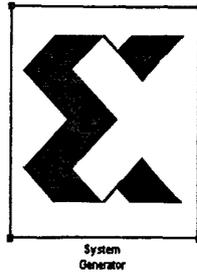


图 5.3 System Generator 模块

如图 5.3 所示，System Generator 模块是一个最基本的模块，该模块主要的重点参数设置包括有以下几点：

1. **Compilation:** 设定编译类型，设置最后代码生成器的生成结果。
2. **Part:** 选择所使用的目标 FPGA 芯片类型。
3. **Hardware description language:** 设定设计编译时生成的硬件描述语言，可以选择 VHDL 或者 VerilogHDL。
4. **FPGA clock period:** 以 ns 为单位设置硬件的时钟周期。
5. **Clock pin location:** 定义硬件时钟的管脚位置。
6. **Simulink system period:** 以秒为单位定义 Simulink 仿真系统周期。

## 二. Gateway 模块

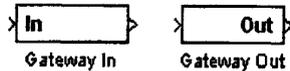


图 5.4 Gateway 模块

如图 5.4 所示，Gateway 模块用来作为 XILINX 与非 XILINX 模块的接口，使 XILINX 可以很方便的在 Simulink 中方便的完成仿真，Gateway 模块有输入和输出两种，该模块的主要参数有

1. **Number of bits:** 定点数的位数。
2. **Binary point:** 小数点的位置。
3. **overflow:** 当数据超出定义的数据范围后，溢出后的操作。
4. **quantization:** 当数据小于定义的范围后，即量化操作。
5. **sample period:** 设定特定的采样率对数据流进行进行处理，通常每个模块按照采样率对输入信号进行采样并按照采样率输出。

## 三. FFT 模块

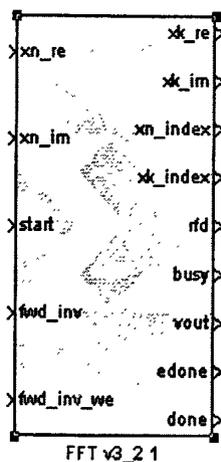


图 5.5 FFT 模块

如图 5.5 所示，FFT 模块是本文实验中最重要模块，也是最难设置的模块，该模块可以完成不同的模块功能（基于不同的设置），如 FFT，IFFT。该模块的主要设置参数有

输入端：

1. xn\_re: 输入数据流的实数部分。
2. xn\_im: 输入数据流的虚数部分。
3. start: 一组数据输入开始标记。当检测到一组输入数据中出现脉冲信号或某个一直为高的信号，则认为出现 start 信号。该信号必须是布尔型。
4. fwd: 如果为 0，则是离散傅立叶反变换，如果是 1，则是离散傅立叶变换。信号 fwd 必须是布尔型。
5. fwd\_we: 当该信号有效时，读入 fwd 端口信号，决定下一帧输入数据的处理方式，该信号必须是布尔型。

输出端：

1. xk\_re: 输出数据流的实数部分。
2. xk\_im: 输出数据流的虚数部分。
3. vout: 标志着输出的数据是否有效，也就是输出的数据是否为处理好的数据。该信号为布尔型。

#### 四. 虚数乘法模块（编程实现非 XILINX 自带）

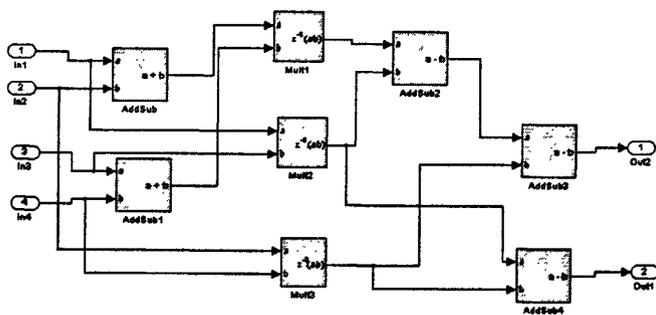


图 5.6 虚数乘法模块

如图 5.6 所示，该虚数模块是运用 XILINX 提供的基本模块编写的，顺利地完成了虚数的乘法模块。

#### 五. 数据源模块<sup>[34]</sup>（Simulink）

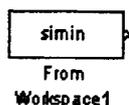


图 5.7 数据源模块 (Simulink)

如图 5.7 所示, 数据源模块是 Simulink 中的模块, 不属于 XILINX 模块, 所以不能被编译成位流文件, 但是在仿真的时候, 可以充当了信号源的作用, 而且这个模块的数据来源是一个当前工作环境下的一个具体变量, 这样输入信号可以通过具体的 M 文件生成。为仿真系统提供了一个良好的接口。

### 5.3 实验测试

在对 System Generator 里面的模块分析之后, 需要进行必要的测试与分析, 下面主要对 FFT 模块、最后的捕获算法进行仿真实现。

#### 5.3.1 FFT 模块测试

实验一:

第一个实验采用了 Simulink 中的正弦函数发生器作为信号发生源, 然后采用了如图 5.8 的系统搭建编程。具体的参数设置的难点就是采样时间的设置。

首先, 将 System Generator 模块里面的 FPGA clock period 设置为 10ns, Simulink system period 时间设为  $\frac{1}{256}$  s (这是由于 FFT 模块选择的是 256 点的采样点, 也就是 FFT 模块设置了主要采集到 256 个点就进行 FFT 计算一次, 所以时间周期需是  $\frac{1}{256}$  的倍数), 这个参数是最难设置的 (后面 M 文件导入数据更复杂), 这样对应的所有仿真采样时间都变为  $\frac{1}{256}$  s。

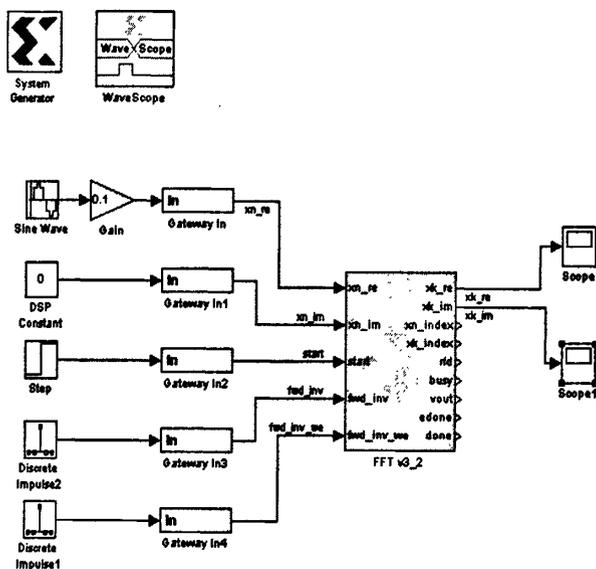


图 5.8 FFT 测试系统 1

在运行的时候, 将运行时间设为 5s (主要考虑到 FFT 模块的延迟问题, 放宽时间), 最后通过 WaveScope 观测信号所得结果如图 5.9 所示。

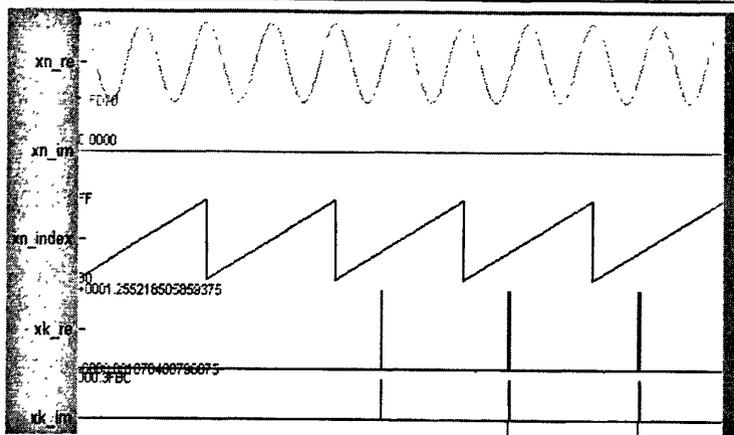


图 5.9 FFT 测试系统 1 结果

可以发现这次运算 5s 时间内, FFT 模块计算了 3 次 256 点的 FFT 计算, 而且很明显看出  $xk\_re$  实部输出的延迟, 也就是 FFT 模块的延迟时间。很遗憾的一点, 就是结果并不是像 Matlab 仿真一样可以看出信号的频率分布, 因为没有具体的坐标来区分。这也说明了 System Generator 是个为硬件服务的软件, 它只管信号的处理, 但是对于信号的物理量并不关心, 而且经过 FFT 运算出来的信号幅值是经过一个比例系数的 (这个参数需自己去测量) 放大或缩小的, 下面的问题是怎么样把该模块用于实际的数字信号处理中。

实验二:

第二个实验是将数据通过 Simulink 中的 From Workspace 数据源模块导入进去, 现在没有发现具体的参考书对此做过解析, 本文实验参数设置是通过具体的例程总结出来。

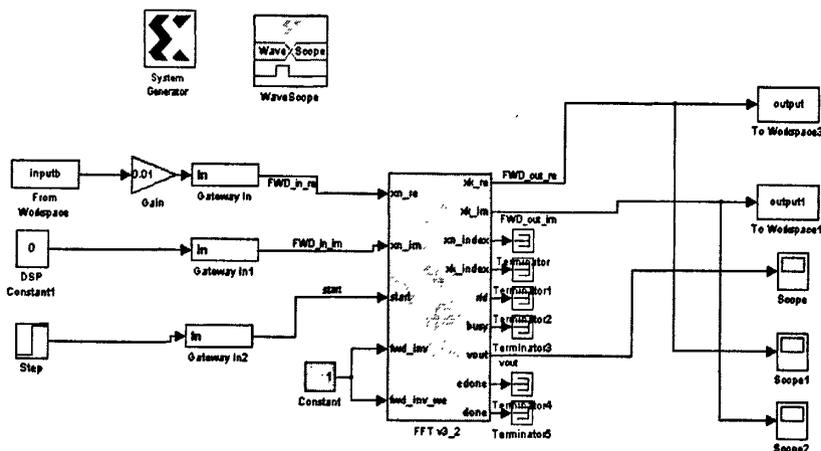


图 5.10 FFT 测试实验系统 2

如图 5.10 为 FFT 测试实验系统 2 的结构图, 通过图 5.8 与图 5.10, 可以看出实验二与实验一的明显区别就是数据源的不同, 实验二的数据源来自于当前工作环境的具体变量。实验二的具体步骤如下:

1. 将 FFT 模块中的采样位数设置为 1024 位 (就是采集到 1024 个采样点就做一次 FFT 计算)。
2. 在当前 Matlab 工作环境中, 读入 1023 位的 1 号卫星 C/A 码放入全局变量 code 里面, 最后通过补零得到 1024 位的全局变量 codeb (因为 FFT 模块只支持基于  $2^N$  的数据计算, 在前面 3.1.4 的补零计算已经论证该方法的可行性)。输入数据可以任意选择。
3. 在得到  $1024 \times 1$  维的 codeb 变量以后, 在变量 codeb 里面加入采样时间标注  $[0:1023]$ , 变成  $1024 \times 2$  维的变量 inputb, 其中时间标注由模块 Gateway 中的 sample period 参数决定, 如果 Gateway 中的 sample period 参数设置为  $\frac{1023}{1024}$ , 则对应的时间标注为  $[0: \frac{1}{1024}: \frac{1023}{1024}]$ , 采样

时间步长应该与 Gateway 对应, 这样为了能让 FFT 模块得到预定的 1024 位采样点。这样变量 inputb 为  $1024 \times 2$  维, 其中 inputb[:,1](也就是第一列数据)为时间标注数据。为了方便在本实验中时间标注设为 [0:1023]。

4. 设置模块 From Workspace 变量为 inputb, 在进入 FFT 模块之前加入增益模块, 增益系数设为 0.01 (将数据缩小 100 倍, 因为 FFT 只支持小数计算, 本实验中 FFT 中支持小数位数设置为 16 位)。

5. 其余简单参数设置, 统一整个系统 sampling period 等。运行时间设为 4000 (4000 个 Simulink 仿真周期)。

6. 最后设置的是输出结果变量, 从 To Workspace 模块中设置, 分别设置为变量 output 与变量 output1。

最后得到的 Scope 结果截图如图 5.11

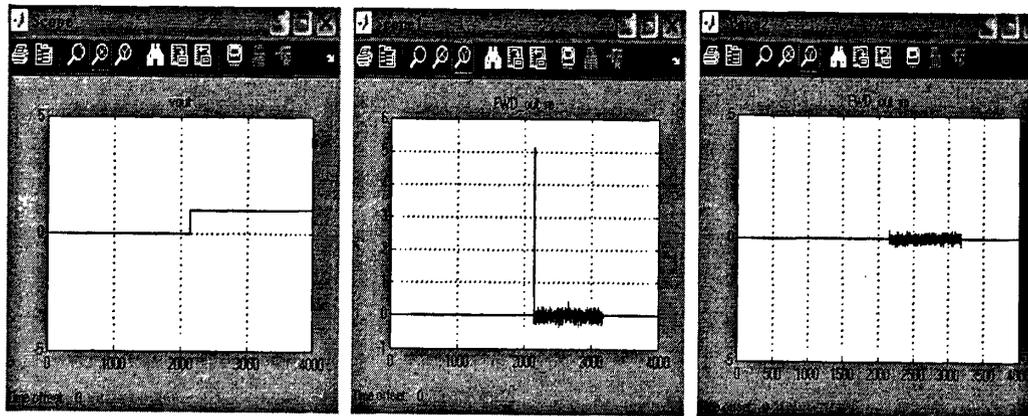


图 5.11 Scope 显示结果

如图 5.11 结果, Scope 显示的 vout 信号表示了时间  $t=2155s$  (这里的时间单位是相对的, 因为实际仿真的时候便没有跑了 4000s, 只是代表了 4000 个 Simulink 仿真周期) 时, FFT 模块运算完了 1024 个采样点的傅立叶变换, 便开始将结果输出, 这样就可以通过这个重要的参数来分析后面的数据。

Scope1 显示的结果是傅立叶变换的实部结果 (从  $t=2156s$  到  $t=3180s$  为有效数据), 同样 Scope2 显示的结果为傅立叶变换的虚部结果。

从当前工作环境中生成了新的两个变量 output 和 output1, 都是  $4001 \times 1$  维 (运行时间为 4000s, 采样间隔时间为 1s, 这样就得到 4000 个点加上最后的延迟一个点, 再次强调仿真中的时间秒数都是相对的), 其中根据图 5.11 中的 Scope 可以知道 output[2156:3180], output1[2156:3180] 为最后的有效结果。现将这两个有用结果用 plot 画出, 如图 5.12 所示。

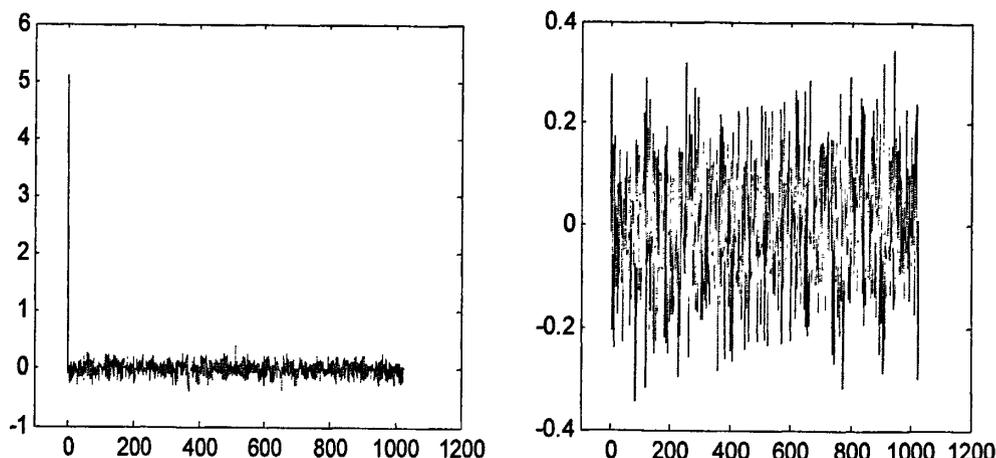


图 5.12 System Generator 导出数据结果(横坐标为数组位数)

现在 Matlab 中用 `fft()` 语句对 `inputb` 进行傅立叶变化结果, `a=fft(inputb)`, 最后对 `a` 的实部和虚部 `plot()` 得到结果如图 5.13 所示。

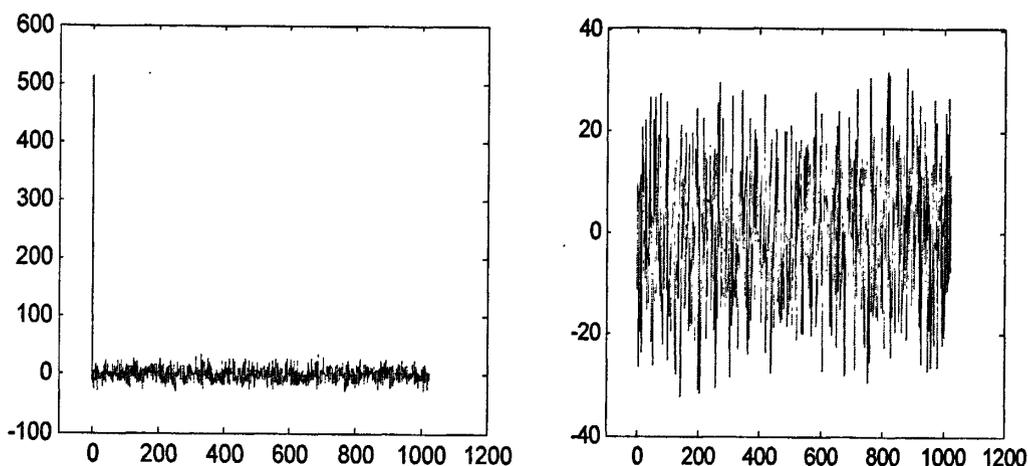


图 5.13 傅立叶变换理论结果(横坐标为数组位数)

由图 5.13 与图 5.12 对比可知, 幅值相差了 100 倍, 这是由于 System Generator 里面的 FFT 只支持输入为小数的数据, 在前面参数设置的时候将整个数据乘以了 0.01 增益系数, 所以导致了最后的 FFT 结果对应缩小了 100 倍, 同时得到的结果略有误差 (在实际编程中, 用得到的具体的数据进行直接比较所得), 但不影响后面捕获算法的仿真实验。

### 5.3.2 捕获算法仿真

捕获算法实现的基础实验前面都一一分析, 包括数据降采样、补零计算、System Generator FFT 模块的应用等。根据这些实验基础结论得到一下实验步骤:

1. 对前端采集的数据进行读入, 并进行降采样, 把原来 16367 个采样点降采样为 8184 个新采样点。
2. 采用补零法, 将 8184 个点补成 8192 个点, 这样数据个数基于  $2^N$ , 可以利用 System Generator FFT 模块进行傅立叶变换计算。
3. 有了输入数据, 对捕获算法结构开始用 System Generator 搭建。其中的注意点就是控制中间的数据格式, 5.3.1 实验已经说明 FFT 模块只支持小数计算。
4. 对最后仿真得到的数据进行放大恢复, 因为前面在 FFT 模块中对数据进行缩小处理。

5. 最后用 Matlab 分析数据。

根据 3.1.4 并行捕获结构用 System Generator 搭建如图 5.14。

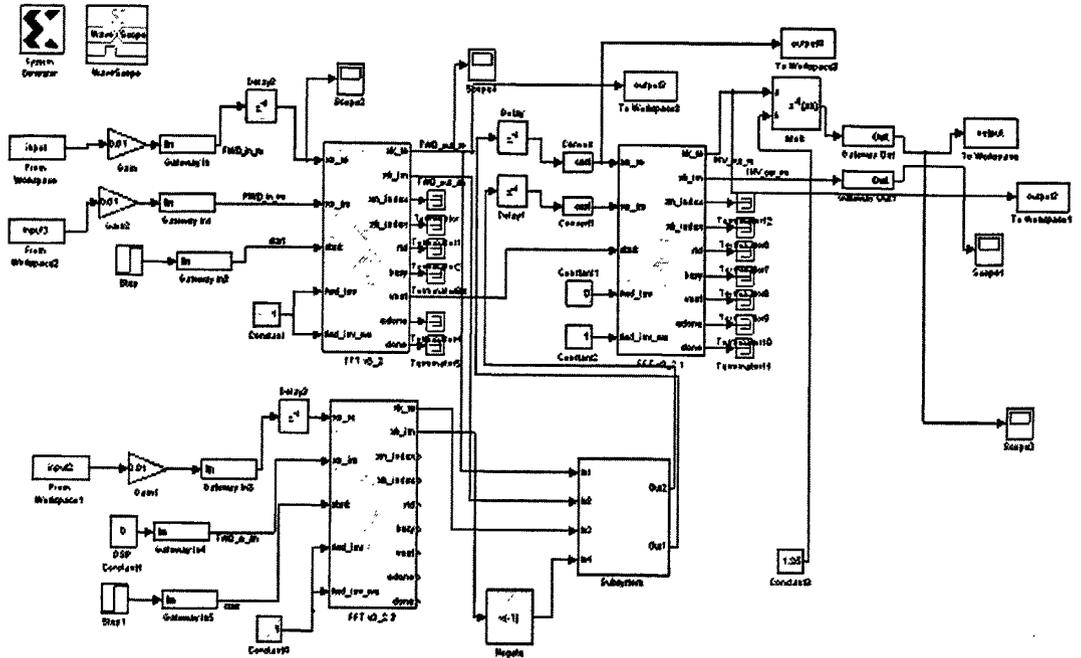


图 5.14 System Generator 系统搭建

System Generator 运行结果会自动保存到工作环境中的变量中，具体的参数变量可以进行设置，但是数据结果是整个系统运行时间的结果包括延迟导致的结果，所以对其中无用的数据进行删除处理，然后对有用数据结果用 Matlab 绘图结果如图 5.15 所示。

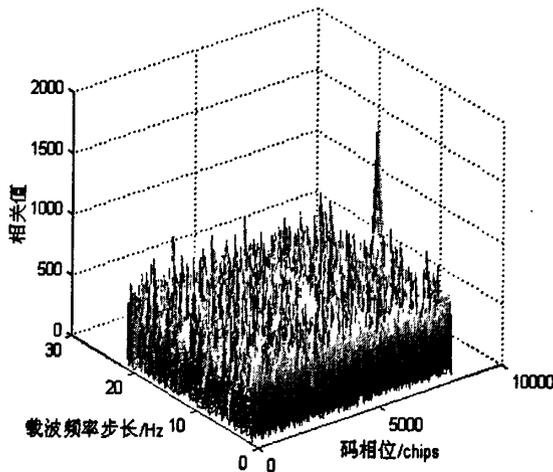


图 5.15 System Generator 导出数据结果

从导出的变量 output 和变量 output2（总共 21 组）综合得到的参数为  $f=4125796\text{Hz}$ ,

$\Delta\tau = 8156$ ，与原先 Matlab 的结果  $\Delta\tau=16291$ ， $f=4125796\text{Hz}$ ，的误差主要是  $\Delta\tau$ 。原因如下：

1. 采用了降采样方法，说明两个  $\Delta\tau$  代表的精度不一样，System Generator 的出来的数据精度为  $\frac{2}{16367} \times 1\text{ms}$ ，而原先的精度为  $\frac{1}{16367} \times 1\text{ms}$ ，其实相互转换结果为  $16291/2=8146$ ，这样还

是有 10 个采样点的误差。

2. 由于补零引起的误差，补零引也会引起码相位的误差，得到的对齐位并不是真正的对齐，就是由于补零引起的。

3. System Generator 的误差，因为整个 System Generator 的系统是在小数下完成的，所以在幅值不断转换的时候也会引起误差，但是这误差对捕获的结果是没有影响的，只会影响具体峰值的大小而已。

## 5.4 小结

本章主要介绍了 FPGA 芯片的特点，以及 FPGA 开发软件 System Generator 的特点与优点，同时介绍了本文实验过程中涉及到的单元模块，最后在单元模块测试实验、前面的理论补零、降采样等实验的基础上完成了捕获算法的 FPGA 仿真实验。

在整个章节的内容中，主要强调了 System Generator 的模块配置，与数据源导入问题的解决。

1. 在 System Generator 配置过程中的时间参数理解问题上，System Generator 模块中的 FPGA clock period 与 Simulink system period 参数的区别，其中 FPGA clock period 为硬件时钟周期，这个毫无疑问，难点是后面的 Simulink system period 参数，该参数的数值并不是真正代表了实际的采样周期，其实该参数的设置要基于 Gateway 模块的参数设置，Gateway 模块是 XILINX 模块与非 XILINX 模块的接口，决定了后序进入 XILINX 模块的数据采样点的格式。

2. 本章重点实验在分析 FFT 模块上面，其实后面的捕获算法都是基于 FFT 模块的扩展，在 FFT 模块开发的时候，要理解导入数据的格式要求(没有使用说明，通过具体的例程分析而得)，也就是采样时间的标注问题，然后也要理解导出数据的格式，它是基于具体的仿真周期决定的，一仿真时间周期对应一个输出数据。

3. XILINX 模块的延迟问题，这个整个系统同步以及对输入输出数据分析时候是非常重要的。

## 第六章 结论与展望

美国 GPS 的现代化,俄罗斯 GLONASS 的复兴和欧洲 Galileo 系统的出现即将带来下一代全球导航卫星定位系统 (GNSS)。同样中国的北斗系统 (BNTS) 也正在慢慢发展中,国内导航领域对北斗系统 (BNTS) 的全面发展已经开始准备,为了能顺利设计、掌握本国自己的北斗导航系统,接收机的研发已经义不容辞,现在国内高校、公司等开始慢慢研究接收机技术并努力将技术将用于实际的实践开发。在此背景下,本文主要研究了 GPS 软件接收机,以便为以后的北斗系统打下良好的基础。

### 6.1 本文工作总结

本文在 GPS 接收机发展背景、GPS 接收机原理、FPGA 芯片结构以及开发原理的基础上对软件接收机算法的具体实现进行了研究,完成的主要工作如下:

1. 软件接收机理论模型推导。比较了 GPS 串行捕获、并行捕获算法差别,分析了捕获目标参数的选择依据;在跟踪环节中,分别对锁相环(PLL)、延迟锁定环(DLL)、锁频环(FLL)的鉴别器选择做了详细的推导;最后对跟踪环路整个控制系统进行了理论推导,并建立了 FLL 辅助 PLL 跟踪模型。

2. 在系统编程实现过程中,主要用 C++语言完成了捕获算法。

3. 在 Matlab 平台上完成了捕获算法改进仿真、跟踪环算法的改进仿真,在 GPS 捕获环节中,用 Matlab 完成了算法的改进包括补零计算、循环相关改进、降采样。分别在 Matlab 完成了半物理仿真实验,实验结果显示,补零计算可以弥补算法中采样点不基于  $2^N$  的不足,

循环相关改进可以减少卷积的计算量,降采样可以减少整体计算数据量。并同时做了详细的误差数据分析。在 GPS 跟踪环路实现中,主要完成了 FLL 模块的建立与仿真;以及完成参数等效噪声带宽、阻尼比的影响实验;各种鉴别器的修改,如 PLL 反馈方式等;最后用 Matlab 实现了 FLL 辅助 PLL 模块,尝试了多种辅助方式,最后选择 FLL 辅助 PLL 的前期(具体在实验中取了 30ms),就是同时利用 FLL 与 DLL 得到的误差信息进行反馈,在 FLL 辅助完之后,并由 PLL 独立完成后面的跟踪,一直到下次失锁。

4. 用 System Generator 软件完成了 FPGA 的捕获算法仿真。在捕获算法的 FPGA 实现过程中,在 Simulink 环境下用软件 System Generator 进行了 GPS 捕获算法的 FPGA 仿真,在整个开发过程中,采用了从简单到复杂,底层到上层的思路进行了相应的开发,完成了 XILINX 模块学习,最后创造性的将采集的数据通过 Workspace 导入到系统中进行仿真,解决了信号源问题,而且没有失去仿真的真正意义,采用了真实的采样数据导入。其中的参数设置是通过具体的例程归纳总结而得。最后从 FFT 算法实现入手完成捕获算法的仿真实现。

5. 论文中所有的程序都采用东方联星的前端采集实际的数据进行了实验验证,实验结果表明算法和所开发程序的正确性。

在完成相应的实验后,论文还进行了具体的误差分析,特别在捕获算法改进、FLL 辅助 PLL 实验以及 GPS 捕获 FPGA 仿真的实验中,分析了具体的误差来源,提出可改进之处。

### 6.2 工作展望

如今软件接收机研究越来越热,在本文实验的基础就软件接收机本身提出一下几点展望。

1. 虽然顺利完成了 GPS 信号捕获跟踪,但在具体实现过程中,基带信号数据处理计算量

偏大,以后可以向粗捕获——FLL——PLL 模型尝试,取消精捕获过程。

2. 在本地信号发生器中,原封不动的按照具体的公式产生本地信号,可以考虑结合程序设计平台优化编程,向实际接收机靠拢(借鉴实际接收机原理),采用查表法等技术减少乘法运算,以及中间变量的反复存储,提高实时性。

3. 改进跟踪环节的理论模型,在跟踪环在 S 域向 Z 域转换的过程中,鉴别器的最优化选择可考虑用更精确的离散积分环节去替代 S 域中的积分环节。同样尝试同时进行载波频率纠正与载波相位纠正措施(虽已经尝试,但是处于精度问题,没有达到具体的精度要求)。

4. 虽然用 System Generator 完成了 GPS 捕获算法的 FPGA 仿真,可以考虑具体的产品开发,用底层语言 HDL 语言开发等措施来完成最后的产品<sup>[35-42]</sup>。

## 致谢

衷心地感谢我的导师陈熙源教授在我读研期间对我的悉心指导和培养。为我的学习提供良好的平台和交流机会，使我大大开阔了视野。为以后的课题研究打下了良好的基础。在日常生活中，陈老师以自己严谨的工作作风、丰富的学识和实践经验来教育我，使我受益匪浅。他对工作的严格要求，对学生的关心和他热情爽朗的性格，都给我留下了深刻的印象。在此对陈老师表示感谢和敬意。

感谢直序扩频接收机技术群里的各位老师和学者为我在理论知识理解方面提供的大量帮助。

感谢导航课题组的徐晓苏教授、程向红教授和吴峻研究员，为整个实验室及学科发展所做的贡献，为我课题研究提供一个良好的环境。

感谢张昆鹏、顾明武、陈建锋、祝雪芬等师兄师弟，在两年多的学习生活中给了我很多鼓励与帮助。同时感谢读研期间学习，生活过的所有朋友，与你们一起度过的美好时光让我终身难忘。

最后要特别感谢我的父母和家人，感谢你们多年来的养育之恩，感谢你们物质和精神上对我无私的支持！谢谢！

## 参考文献

- [1] Elliott D.Kaplan, Christopher J.Hegarty. Understanding GPS[M],电子工业出版社, 2007
- [2] Peter Rinder, Nicolaj Bertelsen. Design of a single frequency GPS software receiver[M], Aalborg University, 2004.
- [3] Wayne Soehren , Wes Hawkinson. Prototype Personal Navigation System[J], IEEE A&E SYSTEMS MAGAZINE, APRIL 2008
- [4] OLEKSIY V, KORNIYENKO ,MOHAMMAD S, SHARA WI.GPS software receiver implementation[C]. Potentials, IEEE. May-June 2007: 42 – 46.
- [5] D. Akopian. Fast FFT based GPS satellite acquisition methods[J]. IEE Proc.-Radar Sonar Navig., Vol. 152, No. 4, August 2005:277-286
- [6] Nomi Sonowal,Rajeev Yadav and S. kannan .Real Time GPS Software Receiver with New Fast Signal Tracking Method[J], IEEE, 2008
- [7] W.H.Lin, W.L.Mao, H.W.Tsao,F.R.Chang ,W.H.Huang. Acquisition of GPS Software Receiver Using Split-Radix FFT[J], 2006 IEEE International Conference on Systems, Man, and Cybernetics October 8-11, 2006
- [8] A. G. Dempster.New GNSS Signals: Receiver Design Challenges[J], The 2004 International Symposium on GNSS/GPS, 6–8 December 2004
- [9] L. L. Liou, D. M. Lin, J. B. Tsui J. Schamus. Frequency Calibration of A/D Converter in Software GPS Receivers[J], IEEE ,2005
- [10] 沈涛, 李传志等. Xilinx FPGA/CPLD设计初级教程[M],西安: 西安电子科技大学出版社,2009
- [11] 纪志成, 高春能. FPGA数字信号处理设计教程[M],西安: 西安电子科技大学出版社, 2008
- [12] 李云松, 宋锐等. Xilinx FPGA设计基础[M],西安: 西安电子科技大学出版社, 2008
- [13] 张洪润, 张亚凡等. FPGA/CPLD应用设计200例[M],北京: 北京航空航天大学出版社, 2009
- [14] 张 雷, 邓江平, 王建宇. GPS软件接收机的模块设计与信号处理[J],计算机工程, 2009
- [15] 袁安存.软件 GNSS 接收机[J]. 船用导航雷达, 2005.
- [16] K J Parkinson, A G Dempster, P Mumford, C Rizos.FPGA based GPS receiver design considerations[C], The 2005 International Symposium on GNSS/GPS, 8-10 December 2005
- [17] Pratap Misra, Per Enge. Global Positioning System[M],Ganga-Jamuna Press,2001
- [18] Kai Borre, Dennis M.Akos, Nicolaj Bertelsen , et al. A Software-Defined GPS and Galileo Receiver[M].NewYork:Birkäuser Boston ,2007.
- [19] Tao Huang ,Bin Guo , Matthew Trinkle. FPGA Implementation of GPS Carrier and Code tracking loops[J],IGNSS Symposium 2007.
- [20] JAMES BAO,YEN TSUI.GPS软件接收机基础[M]. 陈军, 潘高峰等译.北京: 电子工业出版社, 2007.9
- [21] Thorsten Lück, Michael Bodenbach, Thomas Pany. Software versus FPGA based realtime kinematic GNSS receiver - a comparison[J]
- [22] 张国玳, 徐海燕等.高等数序[M].南京: 南京航空航天大学, 2004
- [23] Pratap Misra,Per Enge.全球定位系统[M].第二版.罗鸣, 曹冲等译.北京: 电子工业出版社, 2008
- [24] 东南大学自控系.数字信号处理[M].
- [25] OlinkStar.NewStar210A GPS中频信号采样器[s].北京东方联星科技有限公司.
- [26] 华涛,丁阳,石玉.基于FFT伪码快速捕获方法及其性能分析[J].现代电子技术, 2007
- [27] JAMES BAO-YEN TSUI.Fundamentals of Global Positioning System Receivers A Software Approach[M]. USA:A Wiley-interscience publication.2000
- [28] 胡寿松.自动控制原理[M].北京: 科学出版社, 2001

- [29] 沈锋, 徐定杰等.基于FLL/PLL相结合的载波跟踪在导航接收机中的应用[J].中国航海,2004
- [30] 王惠南.GPS导航原理与应用[M].北京: 科学出版社, 2003
- [31] Elliott D.Kaplan, Christopher J.Hegarty.GPS原理与应用[M].第二版.寇艳红译.北京: 电子工业出版社, 2007
- [32] 王诚, 薛小刚等.FPGA/CPLD设计工具 Xilinx ISE适用详解[M].北京: 人民邮电出版社, 2005
- [33] Bob Stewart, Jeff Weintraub. The Xilinx DSP primer [M]. Scotland: Xilinx program, August 2007
- [34] 张志涌, 杨祖樱等.MATLAB教程[M].北京: 北京航空航天大学出版社, 2006
- [35] Ershen Wang , Shufang Zhang ,etc .Implementation of an Embedded GPS Receiver Based on FPGA and MicroBlaze[J].IEEE,2008
- [36] 刘钝, 曹冲.GPS软件接收机的发展前景[J].全球定位系统, 2003: 23-26
- [37] Heinrich Meyr,Marc Moeneclaey.Digital communication receivers[M].USA:A Wiley-interscience publication.
- [38] 魏敬法, 胡永辉.GPS接收机中采样平均技术的FPGA实现[J].时间频率学报, 2004, 27(1):16-21
- [39] 马薇薇, 黄其华等.基于DSP+FPGA的扩频接收机快捕技术[J].电讯技术, 2003
- [40] 王尔申, 张淑芳等.GPS接收机相关器技术研究及FPGA实现[J].系统仿真学报, 2008, 20(13)
- [41] Hun-Soo Cho,Sung-Hyuck Im. A FPGA-based Software GPS Receiver Implemntation Using Simulink and Xilinx System Generator[J].ION GNSS.2005
- [42] Clifford Kelley,Joel Barnes.OpenSource GPS open source software for learning about GPS[J].

## 攻读硕士期间发表论文

汤新华.接收机捕获算法的 DSP 实现.东南大学校庆学术报告 2009.