北斗系统扩频信号捕获研究报告

陈佳榕

无 42 2014011050

摘要：本文主要研究了相关法在北斗系统扩频信号捕获中的利用，并分别给出了在时域以及频域的实现算法，然后使用 MATLAB 进行了算法实现，结果表明在频域的实现可以获得更好的性能。

0 引言

北斗系统的扩频信号可以看作按一定持续时间以方波形式周期循环其自身特有的一串 PRN 序列而构成的信号，地面上的导航仪接收到的卫星信号是包含着若干颗卫星的扩频信号的混合信号，因为不同的北斗卫星的 PRN 序列重构的方波信号彼此之间可以看成是正交的，所以就可以利用相关法从混合信号中判断特定 PRN 序列是否存在，从而判断是否接收到对应的卫星的信号。

1 相关法在时域的实现

记第 i 颗卫星的 PRN 序列为$\{b_i[0], b_i[1], \cdots, b_i[N-1]\}$，则由 PRN 序列重构得到的方波信号可以记为

$$c_i(t) = \sum_{n=0}^{N-1} b_i[n] p(t - nT_c)$$

其中$T_c = \frac{1}{2.046\times10^6} s$，且

$$p(t) = \begin{cases} 1, 0 \le t < T_c \\ 0, \quad else \end{cases}$$

而不同 PRN 序列重构得到的方波信号的正交性可表示如下：

$$\begin{cases} \int_{-\infty}^{+\infty} c_i(\tau)c_i(\tau-t)d\tau = Q(t) \\ \int_{-\infty}^{+\infty} c_i(\tau)c_j(\tau-t)d\tau \approx 0, i \neq j \end{cases}$$

其中 Q(t)的形状如下图所示，

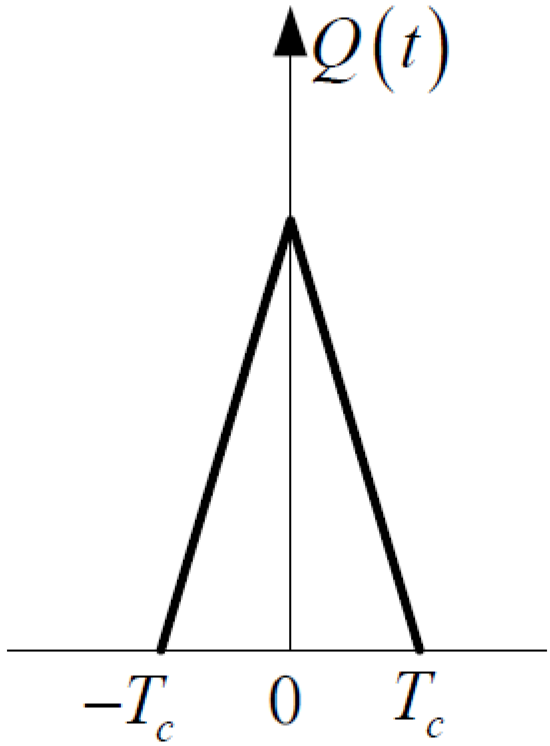


图 1 $c_i(t)$自相关函数示意图

再根据接收信号的表达式

$$r(t) = \sum_{i\epsilon v} A_i s_i(t-\tau_i) + n(t)$$

其中$s_i(t)$表达式如下，即为$c_i(t)$的周期延拓信号（也即扩频信号）

$$s_i(t) = \sum_{n=0}^{\infty} c_i(t-nT_c)$$

所以只要让$r(t)$与$c_i(t)$作相关，根据正交性并认为 n(t)服从零均值高斯分布，有

$$\int_{-\infty}^{+\infty} r(\tau)\, c_i(\tau-t)d\tau$$

$$
\begin{aligned}
&= \int_{-\infty}^{+\infty} \left[\sum_{j\epsilon\nu} A_j s_j\left(\tau-\tau_j\right)+n(\tau)\right] c_i(\tau-t)d\tau \\
&= \int_{-\infty}^{+\infty} \left[\sum_{j\epsilon\nu} A_j \sum_{n=0}^{\infty} c_j\left(\tau-\tau_j-nT_c\right)+n(\tau)\right] c_i(\tau-t)d\tau \\
&= \int_{-\infty}^{+\infty} \sum_{j\epsilon\nu} A_j c_i(\tau-t) \sum_{n=0}^{\infty} c_j\left(\tau-\tau_j-nT_c\right) d\tau \\
&\approx \int_{-\infty}^{+\infty} A_i c_i(\tau-t) \sum_{n=0}^{\infty} c_i(\tau-\tau_i-nT_c)\, d\tau \\
&= \sum_{n=0}^{\infty} \int_{-\infty}^{+\infty} A_i c_i(\tau-t)\, c_i(\tau-\tau_i-nT_c)d\tau \\
&= \sum_{n=0}^{\infty} \int_{-\infty}^{+\infty} A_i c_i(\tau)\, c_i(\tau-t+\tau_i+nT_c)d\tau \\
&= \sum_{n=0}^{\infty} Q(t-\tau_i-nT_c)
\end{aligned}
$$

结果即为图 1 所示波形向右平移τ_i之后再以T_c为周期进行周期延拓，因此利用相关不但可以检测出接收到的混合信号$r(t)$是否包含了某个特定的 PRN 序列对应的卫星发出的信号，还可以检测出其延时。

接下来是算法的设计:

①根据已有的 PRN 序列重构方波信号

②用重构得到的方波信号和接收信号作相关

③根据相关结果判断存在哪些卫星的信号

具体在实现的时候过程如下:

先根据接收信号的采样频率和T_c的关系确定重构方波信号所需要的延拓点数，因为 1ms 内采样点数为$10\text{M}\times\frac{1ms}{1s}=10000$，而 PRN 序列 1ms 对应的有 2046 个点，然后就可以确定延拓点数至少为$\left\lceil\frac{10000}{2046}\right\rceil=5$，做完延拓之后得到的重构方

波一共有5×2046=10230个点,所以还需要对采样信号重采样使得采样信号1ms内的采样点数也为10230。

然后就可以对采样信号和重构方波信号作相关了(相关实际上也可以利用卷积来实现，即先将一个信号作反褶再与另一个信号作卷积)，最后根据相关结果判断是否存在对应的卫星的信号。这里用来做相关的采样信号我只选取前几毫秒的，因为考虑到每个卫星信号的延时不会太大，这样子取前几毫秒既可以包含所有应该有的卫星信号而同时数据处理量又不至于太大。

判断是否存在对应的卫星的方法的出发点是如果存在的话会在相关结果的某个地方出现比较明显的峰值。基于此思想我一开始采用的是判断相邻峰值的距离是否与 PRN 长度相差不大的方法，因为如果存在对应卫星信号的话峰值会周期出现。但是实际上这种办法的效果并不好，因为噪声的影响使得峰值的位置难以准确判断，进而影响对信号存在与否的判断。因此后来我又采取了另一种判断方法,即根据相关结果的最大值和取绝对值之后的平均值的比值如果在阈值之上的话就判为存在，否则判为不存在。这样子的效果要比原来的做法好多了，因为这可以看成是一种基于功率的判断方法,当接收信号存在某个卫星的信号的时候该卫星对应的 PRN 重构方波信号会在相关过程的某处匹配成功，此时信号的功率会较大，信噪比也较高；若不存在某个卫星的信号，则对应的相关过程中不会出现特别尖锐的峰值，整个过程的最大值与取绝对值后的均值相差不会太大。因此根据这个来判断的话也会比较准确，这里面还包含了匹配滤波的思想。

运行的结果如下图所示:

```
>> result'

ans =

    1    2    8    9   11   12   17   18   22   23   25   27   30   33
```

一共检测出了 14 个卫星，编号如上图。

接下来分析算法的复杂度——

假设 PRN 重构方波信号的长度为 m，用来做相关运算的那部分采样信号的长度为 n，卫星的个数为 p，因为算法的计算复杂度主要在于相关计算，其他部分相对于此都是小量，所以只需分析相关运算的复杂度如下：

一般情况下采样信号的长度总是要长一些的，所以乘法次数为

$$1 + 2 + \cdots + m + (n - m) \times m + (m - 1) + (m - 2) + \cdots 1 = mn$$

加法次数为

$$\begin{aligned} &1 + 2 + \cdots + (m - 1) + (n - m) \times (m - 1) + (m - 2) + (m - 3) + \cdots 1 \\ &= (m - 1)(n - 1) \end{aligned}$$

而一共需要做 p 次相关运算，因此算法的计算复杂度为 O(mnp)。

至于存储复杂度，因为需要存储的是 PRN 重构方波信号和用来做相关运算的那部分采样信号，所以存储复杂度也只有 O(m+n)。

实际实现的时候运行时间如下：(代码见 timeDomain.m)

函数名称	调用次数	总时间	自用时间*	总时间图 (深色条带 = 自用时间)
timeDomain	1	3.666 s	0.130 s	
conv	37	2.957 s	2.957 s	
resample	1	0.456 s	0.003 s	
resample>uniformResample	1	0.450 s	0.028 s	
firls	1	0.317 s	0.084 s	
sinc	4	0.201 s	0.201 s	
close	1	0.098 s	0.021 s	
close>safegetchildren	1	0.075 s	0.003 s	
allchild	1	0.071 s	0.025 s	
kaiser	1	0.062 s	0.049 s	
onCleanup>onCleanup.onCleanup	1	0.042 s	0.042 s	
sigcasttofloat	6	0.041 s	0.018 s	
upfirdn	1	0.024 s	0.004 s	
sigcheckfloattype	6	0.024 s	0.024 s	
mean	37	0.019 s	0.019 s	
rat	1	0.019 s	0.019 s	

明显可以看到相关运算用去了最多的时间（注：我用卷积实现了相关运算），所以这里是优化算法的一个首选突破口。考虑到在时域做卷积相当于在频域做相乘，而从时域变到频域又可以选择计算复杂度较低的 FFT，因此这就引出了下一个实现——相关法在频域的实现。

2 相关法在频域的实现

这里实际上是对上一个算法的优化，关键就是对相关计算的优化。记 PRN 重构方波信号为 f(t)，采样信号为 g(t)，则相关结果为 c(t)=g(t)*f(-t)，在频域的话就是$C(j\Omega)=G(j\Omega)F^{*}(j\Omega)$，这里利用到了 f(t)为实信号的性质。

所以在上一个方法的基础上稍作改变，将相关的计算改为以下过程：

①对 PRN 重构方波信号作 FFT 变换，当然为了防止重叠而需要补上 length(g)-1 个零，可以得到 FFT 结果$F(j\Omega)$，再做一次共轭得到$F^{*}(j\Omega)$

②对采样信号作 FFT 变换，同样为了防止重叠需要补上 length(f)-1 个零，可以得到 FFT 结果$G(j\Omega)$

③频域相乘，得到$C(j\Omega)=G(j\Omega)F^{*}(j\Omega)$，最后做逆 FFT 变换，得到相关结果

算法的其他部分与上一个方法是一致的。

运行的结果如下图所示：

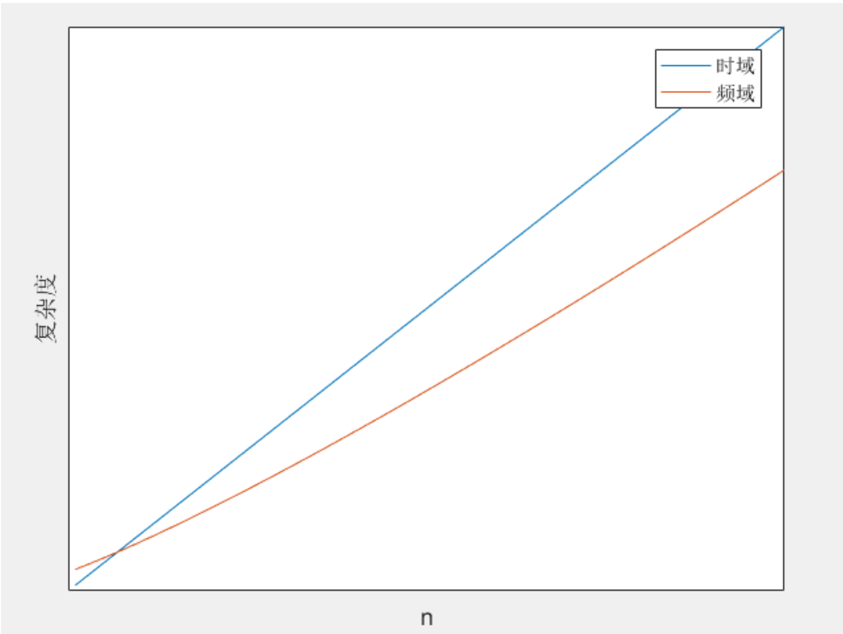
```
>> result'

ans =

    1    2    8    9   11   12   17   18   22   23   25   27   30   33
```

两种方法得到的结果是一样的。

接下来分析优化之后算法的复杂度——

同样假设 PRN 重构方波信号的长度为 m，用来做相关运算的那部分采样信号的长度为 n，卫星的个数为 p，因为 FFT 的复杂度为 O($N\log_2 N$)，其中 N 为作 FFT 运算的数据点数，而一共需要进行 2 次 FFT 和一次 IFFT，进行 FFT 和 IFFT 运算的数据点数为 m+n-1，同时频域相乘还需要 m+n-1 次乘法运算，最后加上 p 次循环，因此总的计算复杂度为 O($p(m+n)\log_2(m+n)$)。

将时域和频域的计算复杂度作比较如下图：



可见当数据量比较大的时候频域方法要优于时域方法，这种优势在数据量越大的时候会表现得越发明显。

然后再考虑存储复杂度，需要存储的信号为 PRN 重构方波信号和用来做相关运算的那部分采样信号，以及 FFT 变换结果和 IFFT 逆变换结果(都是 m+n-1)，实际上存储量会比时域方法偏大，但复杂度上都是 O(m+n)。所以这里的优化其实也可以看作以空间换时间，而且由于在时间上带来的增益要大于空间上带来的损耗，所以这种优化是可以接受的。

实际实现的时候运行时间如下：(代码见 freqDomain.m)

函数名称	调用次数	**总时间**	自用时间*	总时间图 (深色条带 = 自用时间)
freqDomain	1	1.649 s	1.319 s	
resample	1	0.271 s	0.004 s	
resample>uniformResample	1	0.264 s	0.025 s	
firls	1	0.146 s	0.081 s	
sigcasttofloat	6	0.049 s	0.020 s	
kaiser	1	0.049 s	0.035 s	
close	1	0.037 s	0.017 s	
rat	1	0.030 s	0.030 s	
sigcheckfloattype	6	0.029 s	0.029 s	
sinc	4	0.025 s	0.025 s	
mean	37	0.022 s	0.022 s	

明显有了性能上的提升，验证了理论分析的正确性。

这也显示了傅里叶变换的强大功能，在时域行不太通的可以变换到频域做，这也启示了我在思考问题的时候要懂得变换思路，当然因为经过各种有关信号处理课程的锻炼我们很容易会想到从时域到频域的变换，这也是一个电子人应该具有的基本素养。但我想到的是更多其他方面的思路变换，毕竟在以后的学习乃至研究生涯中遇到的是各种类型的问题，我想这种变换思路的方法会给予其很大的帮助。

3 结语

基于卫星本身特有 PRN 序列的良好的相关特性（近似为正交性），利用相关法可以从混合信号中识别出特定 PRN 序列的存在与否，达成扩频信号捕获的目的。基于相关法，最直接的想法是可以从时域做相关处理，因为相关处理复杂度较高进而得到一个新的想法—变换到频域进行处理。两种想法都可以较好地实现扩频信号的捕获，区别主要在于复杂度，在数据量较小的时候时域方法的复杂度要更低，在数据量较大的时候频域方法的复杂度更低，且随着数据量的增大优势越发明显。