Greta Oto 基带仿真模型设计文档

Jun Mo

Globsky Technology Inc.

2021/5/20

版权信息

1. 基带仿真模型设计

在对基带控制及位置解算 firmware 进行开发调试的时候，可以在实际的硬件平台上进行调试，但是硬件平台有若干固有的缺点：第一，在硬件平台上实际运行的时候，由于信号源不能中断，使得在进入断点后不能恢复程序的连续运行；第二，即使是采用模拟器的信号源，由于噪声的随机性，调试所需的中间结果数据是无法复现的；第三，嵌入式系统上的基于交叉编译器和 JTAG 的调试界面和调试手段往往不如 PC 上 IDE 工具提供的调试手段丰富和易用。

通过采集射频前端的中频采样数据或通过 Matlab 等工具生成的中频采样数据可以存储在文件中，在调试的时候以 C Model 作为底层处理中频采样数据，并通过抽象后的软/硬件接口提供相干累加结果给 firmware。通过这种方式，可以在 PC 平台上进行开发调试，并且规避了在实际硬件平台上调试的缺点，但是又引入了其他的缺点：第一，基于 C Model 的数据处理运算量大，处理速度慢；第二，如果需要仿真比较长的时间，那么中频采样数据的文件会非常大。

因此，如果既需要调试方便，又需要比较快的仿真速度，可以采用基于 SignalSim 的基带仿真平台。SignalSim 基带仿真平台是以 SignalSim 卫星信号仿真平台的底层作为支持，并在卫星信号和接收机轨迹仿真的基础上产生模拟的基带相干累加输出，然后通过抽象后的软/硬件接口提供给 firmware。通过这种方法可以直接计算相干累加结果，与 C Model 处理中频采样数据相比大大节省了计算时间。由于不受实际信号产生速度的限制，基于 SignalSim 的 firmware 仿真甚至可以达到几十倍于实际硬件平台的速度。

仿真平台 SignalSim 的设计思路详见《卫星信号仿真平台 SignalSim 设计说明》，其源代码和说明文档可以在 github 上进行下载：https://github.com/globsky/SignalSim

本文档主要讲述在 SignalSim 的平台上如何生成与 Greta Oto 基带硬件一致的仿真相干累加结果以及如何构造硬件抽象层。

2. 相关结果产生模型

无论是捕获引擎还是跟踪引擎的结果，都是先产生 1ms 的相关结果，然后再按照硬件完成相干累加以及非相干累加的方式，将 1ms 的相关结果进行累加运算，并最后输出。因此，首先需要计算 1ms 的相关结果。

根据卫星信号的特性，相关结果应该符合以下的公式：

$$S = A \cdot R(|\Delta t|) \cdot sinc\left(\Delta \bar{f} \cdot \mathrm{T}\right) \cdot e^{j\Delta\bar{\phi}} + \sigma \cdot noise$$

其中，A 是信号幅度，由信号功率决定，根据折算出来的信噪比和噪声功率计算。$R(\cdot)$是相关函数，表示相关峰的形状。Δt是码相位差，其中峰值相关器上的Δt是信号码相位和本地码相位的差，其他相关器的码相位差从峰值相关器根据相关器间隔折算。$\Delta \bar{f}$是积分时间段内的平均频差，

T=1ms 是积分时间。$\Delta\bar{\phi}$是积分时间段内的平均相位差，也就是平均的卫星信号相位减本地载波相位。σ是噪声幅度，*noise* 是标准高斯分布的复数白噪声。*sinc* 函数的表达式是$sinc(x) = \frac{\sin(\pi x)}{\pi x}$

2.1 噪声功率

一毫秒相关结果上的噪声功率（或噪声幅度）取决于输入信号的功率和信号处理增益。对于捕获引擎和跟踪引擎分别进行计算。1ms 相关结果上的噪声幅度计算公式为$\sigma_{1ms} = \sigma_{in} \cdot G_P \cdot \sqrt{N}$。其中$\sigma_{in}$是输入信号噪声的幅度，$G_P$是信号预处理阶段的增益，*N* 是参与累加的 1ms 内的样点个数。

对于捕获引擎，输入的信号为 2 比特的 sign/mag 格式的信号，在最优量化门限下，大约 2/3 的采样点量化为±1，1/3 的采样点量化为±3，此时信号的平均功率为$\frac{2}{3} \times 1^2 + \frac{1}{3} \times 3^2 = \frac{11}{3}$，或者说噪声的平均幅度$\sigma_{in} = \sqrt{\frac{11}{3}}$。在进行剥离多普勒的运算时，sin/cos 查找表的增益为 7，然后样点两两相加右移一位，增益为 1/2，因此，$G_P = \frac{7}{2}$。对于 1ms 的累加，共有 2046 个样点通过匹配滤波器累加，因此 *N*=2046。由此计算得到捕获引擎的 1ms 积分结果的噪声幅度约为 303。

对于跟踪引擎，输入的信号为 4 比特的 sign/mag 个数的信号，在最优量化门限下，大约 18.2% 的信号量化幅度位的最高比特为 1，对于高斯白噪声信号量化后的噪声幅度大约在 6 附近，即 $\sigma_{in} \approx 6.5$。在进行剥离多普勒的运算时，sin/cos 查找表的增益为 12，然后样点右移位数为 preshift+3。如果采样率为 4.113MHz，则 N=4113。由此计算得到在 preshift=0 时跟踪引擎的 1ms 积分结果的噪声幅度约为 625。

2.2 信号幅度

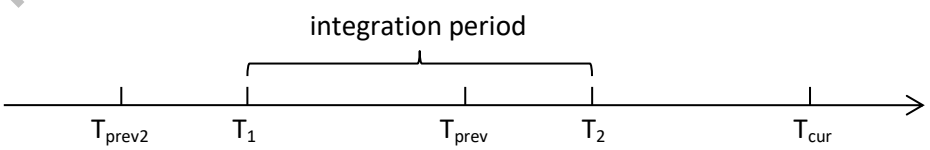
对于 1ms 的积分区间来说，噪声带宽为 1kHz。因此 CN0 和 SNR 的换算关系为 SNR=CN0-30。给定了噪声幅度后，信号的幅度按照以下公式进行计算：

$$\mathrm{SNR} = 10 \cdot \log 10 \frac{P_s}{P_{NI} + P_{NQ}} = 10 \cdot \log 10 \frac{A^2}{2\sigma^2}$$

因此可以得到$\mathrm{A} = \sqrt{2}\sigma \cdot 10^{SNR/20}$。

2.3 平均频差和平均相位差

积分时间段内的平均频率差$\Delta\bar{f}$和平均相位差$\Delta\bar{\phi}$通过以下方法进行计算：

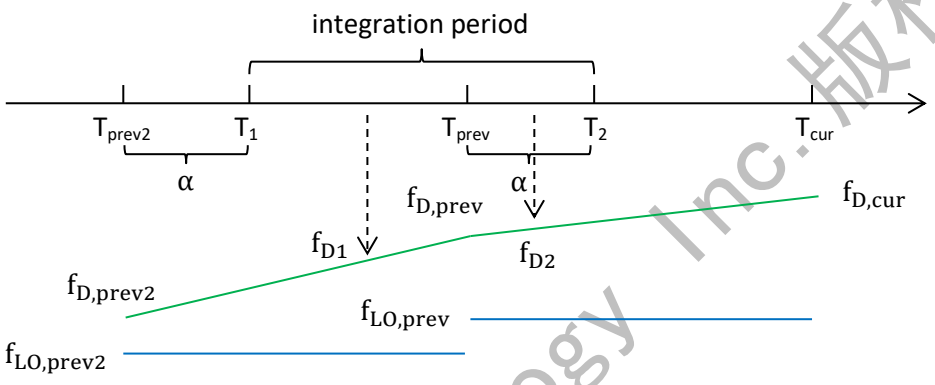


在上图中，T_{cur} 表示当前观测时刻，T_{prev} 表示前一毫秒的观测时刻，T_{prev2} 表示再前一毫秒的观测时刻。而 T_1 到 T_2 表示一个完整的码周期，其积分值的结果在 T_{cur} 时刻输出。因此知道 T_{cur} 时刻的码相位，就可以计算得到码周期起始时刻 T_1 和结束时刻 T_2 相对于 T_{cur} 及 T_{prev} 的关系。

尽管载波相位和载波频率是联系在一起的，为了简便计算，我们对平均的载波频率差和平均的载波相位差分别计算。

设$\alpha = \frac{T_2 - T_{prev}}{T_{cur} - T_{prev}}$表示积分区间在最近 1 毫秒内所占的比例，或者等效的当前时刻在 1ms 内的码相位为$1 - \alpha$（对于码长超过 1ms 的信号，当前码相位为码片计数对 1023 取模）。

频差的计算如下图所示：



如果在上述三个观测时刻分别计算得到的信号多普勒分别是 $f_{D,cur}$、$f_{D,prev}$ 和 $f_{D,prev2}$，并且认为信号频率是线性变化（如绿色线所示）的，则可以得到在 T_1 到 T_{prev} 和 T_{prev} 到 T_2 两个时间段上的平均信号多普勒分别是：

$$f_{D1} = f_{D,prev2} + \frac{1+\alpha}{2}\left(f_{D,prev} - f_{D,prev2}\right) = \frac{1-\alpha}{2} f_{D,prev2} + \frac{1+\alpha}{2} f_{D,prev}$$

$$f_{D2} = f_{D,prev} + \frac{\alpha}{2}\left(f_{D,cur} - f_{D,prev}\right) = \frac{2-\alpha}{2} f_{D,prev} + \frac{\alpha}{2} f_{D,cur}$$

如果在 T_{prev2} 和 T_{prev} 两个时刻，跟踪环路设置的本地载波频率分别是 $f_{LO,prev2}$ 和 $f_{LO,prev}$（如蓝色线所示，本地频率仅在观测时刻，或者说中断时刻会进行改变），则在 T_1 到 T_2 的时间段内，平均的频率差为：

$$\Delta\bar{f} = (1-\alpha)\left(f_{D1} - f_{LO,prev2}\right) + \alpha\left(f_{D2} - f_{LO,prev}\right)$$

同样的，如果在三个时刻，信号的相位分别是分别为$\Phi_{s,prev2}$、$\Phi_{s,prev}$和$\Phi_{s,cur}$，而本地的以周为单位的载波相位分别为$\Phi_{LO,prev2}$、$\Phi_{LO,prev}$和$\Phi_{LO,cur}$，忽略信号频率变化产生的高阶项，认为相位线性变化，则可以分别计算三个时刻以弧度为单位的相位差$\Delta\phi = 2\pi(\Phi_s - \Phi_{LO})$，则平均的相位差为：

$$\Delta\bar{\phi} = (1-\alpha)\left(\frac{1-\alpha}{2}\Delta\phi_{\text{prev2}} + \frac{1+\alpha}{2}\Delta\phi_{\text{prev}}\right) + \alpha\left(\frac{2-\alpha}{2}\Delta\phi_{\text{prev}} + \frac{\alpha}{2}\Delta\phi_{\text{cur}}\right)$$

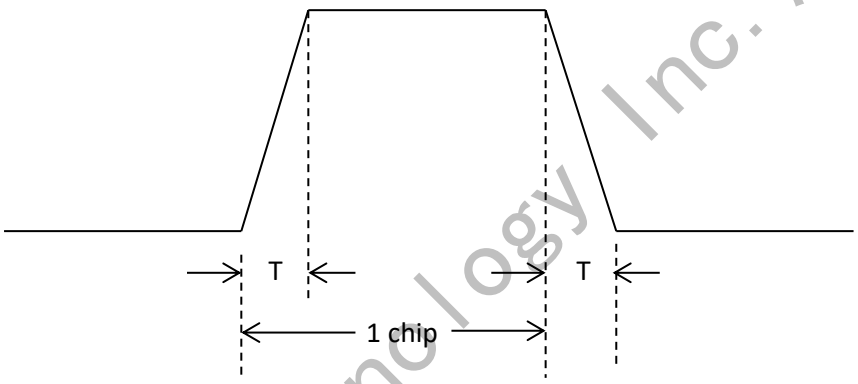
对于不同的相关器来说，$\Delta\bar{f}$和$\Delta\bar{\phi}$是相同的。但是，相关函数$R(\cdot)$的参数不同。

2.4 相关函数

在前端带宽无限的情况下，相关函数是一个三角形，其公式表达是：

$$R(\Delta t) = \begin{cases} 1 - \dfrac{\Delta t}{T_c}, \Delta t \le T_c \\ 0, \Delta t > T_c \end{cases}$$

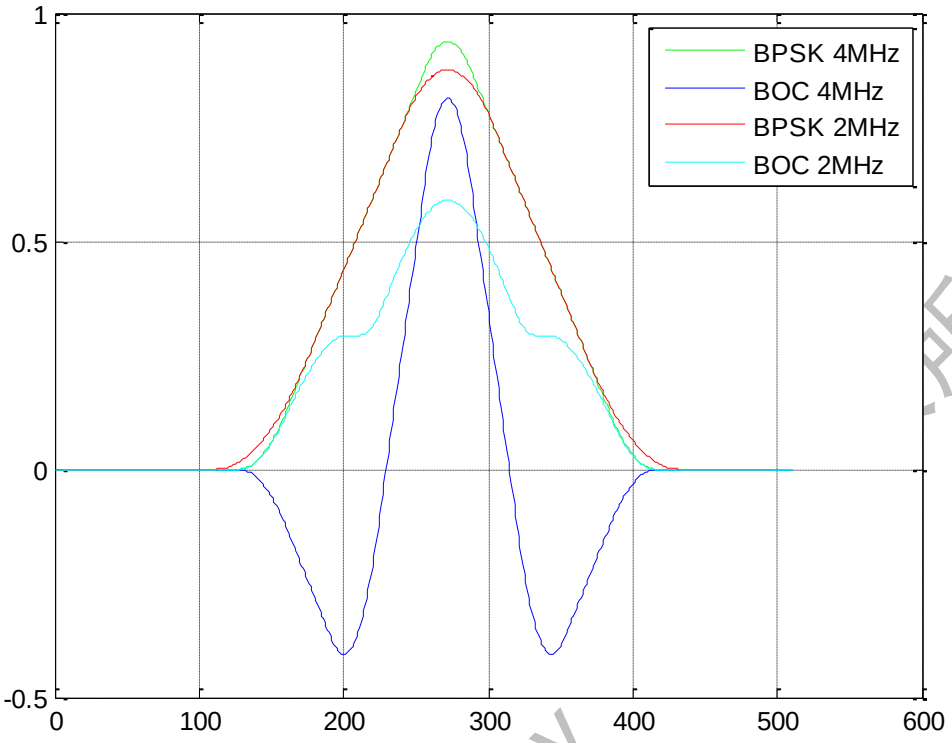
但是对于前端带宽受限的信号，由于信号变化的边沿会变得平缓，导致相关峰的形状会变得更加圆滑。通常对于造成这一现象的近似是将前端带宽受限信号的变化边沿变成一个固定斜率的斜坡，变化区间与前端带宽成反比，如下图所示：



其中$\mathrm{T} = \frac{1}{f}$，f 为前端带宽。需要注意的是，前端带宽包括射频前端的滤波器带宽和进行 AD 采样后进行数字处理的滤波器带宽。因此，对于捕获引擎，由于增加了额外的降采样滤波，带宽会变得更窄。对于跟踪引擎来说，带宽可以设计为 4MHz（4 倍码速率）。对于捕获引擎来说，带宽设计为 2MHz（2 倍码速率）。对于 BPSK 调制的信号和 BOC 信号，也需要计算不同的相关峰形状。另外捕获引擎对于 BOC 信号的单边带处理也需要考虑。

相比于计算相关峰形状的公式解，通过 Matlab 计算出不同延时的相关函数值并在使用时通过查表法计算是一个更加方便简单的方法，并且可以达到足够的近似精度（在模拟带宽受限信号时就已经进行过近似了）。

下图显示了两种不同的前端带宽以及 BPSK 和 BOC 调制信号的相关峰形状曲线（横坐标单位为 1/128 码片间隔）。



2.5 不同相关器上的信号和噪声

由于本地码频率和信号码频率的差别不会很大，同时在短时间内可以假设信号源的频率是固定的，因此我们可以近似在积分周期内本地码相位和信号码相位的差固定不变，也就是可以取观测时刻的码相位差作为峰值相关器的码相位差。其他相关器的码相位差根据相关器间隔进行折算。各个相关器的积分时间段基本一致，因此平均的载波频率差和载波相位差可以用同样的值。

不同相关器的信号幅度可以用信号幅度 A 乘以相关函数值R(·)进行计算，但是需要注意的是不同的相关器在加上加性高斯百噪声的时候，由于各自本地码之间具有相关性，因此各自的噪声并不是独立的。产生各个相关器的噪声的时候，各个相关器之间的噪声的实部或者虚部的相关矩阵为：

$$\mathbf{R}=\begin{bmatrix} 1 & 1-\Delta & 1-2\Delta & \cdots & 0 \\ 1-\Delta & 1 & 1-\Delta & \cdots & 0 \\ 1-2\Delta & 1-\Delta & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

也就是主对角元素为 1，表示自相关为 1；次对角元素为$1-\Delta$，表示相邻相关器之间噪声的相关系数为$1-\Delta$；再次对角元素为$1-2\Delta$，表示间隔 2 的相关器之间噪声的相关系数是$1-2\Delta$；

以此类推。其中Δ是相关器间隔与码片间隔的比值。如果相关器间隔为 1/2 码片，则$\Delta = \frac{1}{2}$；相关器间隔为 1/8 码片，则$\Delta = \frac{1}{8}$。

由于上述相关矩阵$\mathbf{R}$为对称正定阵，因此可以将上述矩阵做 Cholesky 分解，得到三角阵$\mathbf{C}$。于是有$\mathbf{R} = \mathbf{C} \cdot \mathbf{C}^{\boldsymbol{T}}$。当需要产生相关的 N 个复数噪声时，首先产生 N 对独立同分布的高斯白噪声形成 N 个复数噪声，然后将三角阵$\mathbf{C}$与产生的 N 个复数噪声进行相乘，就可以得到符合相关关系为$\mathbf{R}$的 N 个相关噪声了。

上述方法适用于跟踪引擎中不同相关器的噪声。对于捕获引擎中不同码相位的噪声，可以简单地产生 N 个独立同分布的噪声，然后相邻两两相加再除以$\sqrt{2}$即可。

3. 捕获引擎模型设计

上一节介绍了相关结果产生的数学模型，本节对捕获引擎的具体实现进行介绍。

当写寄存器开始填充 AE Buffer 的时候，模型实际实现的是对填充的时间所对应的各个卫星的信号特性进行计算，其中包括可见星的 SVID，对应的信号幅度、多普勒以及码相位。其中为了和捕获结果相对应，码相位表示为从当前开始填充时刻需要经过多少个码片到达码周期的起始位置，也即是码周期减去信号发射时刻对应的码相位。由于信号在填充到 AE Buffer 的时候需要经过一个 6 阶滤波器，因此还需要再加上额外的 2.5 个样点对应的延时。

由于捕获的信号会包含数据调制，因此对应的信号特性还包括数据调制的比特流。比特流信息包括了下一个码起始位置在当前数据比特中的位置，以及从当前数据比特开始后续的调制比特。对于 128ms 的 AE Buffer 来说，GPS L1C/A、Galileo E1、GPS L1C 和 BDS B1C 在 128ms 内的数据比特数最多分别为 8、32、14、14。

实际的捕获引擎运行流程除了每次产生 1ms 的相关结果，而不是 3 个 1/3 毫秒的相关结果相加以外，DFT 相干累加、非相干累加、码搜索和频率搜索的循环流程与 C Model 都是一致的，只是中间结果都是以浮点数表示，一直到最后填充结果到 Config Buffer 的时候才转为整数。

上一节中经过计算得到 1ms 相关结果噪声的平均幅度约为 303，在实际产生噪声的时候，采用 Box-Muller 算法产生独立分布的噪声，然后相邻的两两相加后，产生相邻相关系数为 1/2 的随机噪声。由于两两相加会将噪声的方差增加$\sqrt{2}$倍，因此产生的独立分布噪声的方差应该除以$\sqrt{2}$。又由于 C Model 在非相干累加计算幅度以后会有一个右移一位的操作，相当于所有信号的幅度减半，所以产生噪声的方差应该为$\frac{303}{2\sqrt{2}} \approx 107$。考虑到实际的中频采样信号并不是严格的正态分布，因此适当增加噪声方差到 120。

在每一次新的非相干累加开始时，计算当前位置对应的数据调制比特和当前位置在数据调制比特中的毫秒计数，然后每次的相干累加依次增加毫秒计数并确定调制比特。

最后，根据最大峰值的幅度可以放到 8 比特的结果中确定 GlobalExp，然后相应地计算出其他峰值的幅度以及 noise floor 的值，写回到 config buffer 中。

4. 跟踪引擎模型设计

跟踪引擎的模型主要处理各个计数器状态以及相关值的产生。下面按照跟踪引擎模型的功能对实现方法和需要注意的地方进行介绍。

4.1 寄存器和 TE Buffer 的读写

寄存器的读写和 C Model 中的基本一致，主要的不同是 TE Buffer 的读写。由于跟踪引擎的运行会改变 TE Buffer 中的内容，因此根据 TE Buffer 中不同的数据段，处理的方法也不相同。

首先对于控制和状态数据，除了将数据写入TE Buffer的存储空间外，还另外有一个数据结构ChannelConfig，在将控制信息和状态信息写入TE Buffer的时候，会对写入的数据进行解析，并赋值给数据结构中的相应变量。在读取的时候，由于控制数据不会变化，因此直接从TE Buffer中取值，而状态数据则会根据数据结构中的值进行相应的拼接。

对于部分累加和的8个DWORD，由于仿真模型和基带控制软件都不会用到相应的值，因此除了在初始化的时候清零以外不做其他的处理。

对于相干累加的结果，也是像C Model中一样处理，需要累加的时候从TE Buffer相应的地址取出结果，与新的相关结果累加后再写回TE Buffer相应的地址，因此基带控制软件无论是写还是读，也都是照常访问TE Buffer的相应地址内容。

需要注意的是，对于PRN码生成相关的配置和状态，在写入PRN_CONFIG的时候，会解析出系统频点选择以及相应的SVID，如果没有匹配的SVID，则会设置为0，在生成相关结果的时候不会加入信号部分。在写入PRN_COUNT的时候，会根据相应的系统选择计算得到当前的码计数值。因此，这也要求在写PRN_COUNT的时候，应该已经写入了PRN_CONFIG（这一点需要软件保证，但是一般来说由于PRN_CONFIG地址在前，因此是可以保证这一点的）。

另外为了简化模型，仿真模型对于PRN_CONFIG2和PRN_STATE2不做解析，缺省认为对于E1、B1C和L1C的信号第一个PRN生成器用于产生导频PRN码，第二个PRN生成器用于产生数据PRN码。

在写入PRN_CONFIG的时候，由于此时相当于重新配置了该通道的SVID，因此也会对控制通道数据生成结构体CarrierParameter进行赋值，以便后续正常计算平均的频率差和相位差。同时也会生成一整帧（或子帧）的数据调制比特。

4.2 计数器更新

观测量产生依赖于相关器中的各个计数器，因此每向前推进1ms的时候，也需要相应地更新各个计数器，包括载波相位、载波整周、码相位、码片计数以及相干累加计数、NH计数等。

由于相关器中各个不同延迟的相关结果是依次产生的，所以在当前的起始时刻有可能只产生了部分相关值，所以需要处理剩余部分相关值继续输出，以及推进1ms以后新的当前时刻也只生成部分相关值的情况。

模型还需要考虑到和实际硬件中一致的时序问题，比如NH计数值的增加是与Cor0相关结果输出同步的，而相干累加计数值的增加是与Cor7（即最后一个相关累加器）的相关结果输出同步的。

计数值更新中码计数依赖于码NCO溢出的次数，同时哪些相关累加器会输出相关累加值也是从溢出次数计算，因此计数器更新的同时还会计算需要输出的相关累加器序号以及和各个相关累加器对应的码相位（以1/2码片为单位，各个相关累加器依次减1）。

4.3 相关结果产生

相关结果的产生是按照第二章中介绍的相关结果产生模型计算的。首先产生具有相应相关系数的高斯白噪声，然后如果存在相应的可见卫星，则根据模型加入信号。

对于平均频率差和相位差的计算，会根据当前值首先更新数据结构CarrierParameter中的值，然后根据公式进行计算。这里需要注意的一点是，由于相位差可能跨过2π的边界，因此计算加权的平均值的时候对于这样的情况需要进行补偿。

信号的幅度值根据各个相关器对应的码相位以及信号的码相位做差后在查找表里面通过线性插值进行计算。对于有窄相关的情况，也要对相应的相关累加器调制对应的码相位（Cor2、Cor3、Cor5、Cor6）。

调制数据是根据计算出的信号发射时刻在比特数组中查找到相应的值。如果当前时刻已经进入到新的一帧，则需要产生新一帧的数据。计算当前比特的时候，因为实际计算的是当前已经结束的整毫秒的起始时刻对应的调制比特值，所以需要将发射时刻减1进行计算。

最终产生的浮点相关结果，经过PreShift和PostShift的调整，再进行相干累加，并将结果写入TE Buffer。

5. 基带顶层模型设计

基带顶层提供了和C Model基带顶层相似的函数接口，因此可以实现与基于C Model的仿真平台的替换。

寄存器的读写部分和C Model基本一样，唯一的区别是如果涉及到PRN_CONFIG的写，则需要对相应的通道进行初始化操作。

为了辅助捕获转跟踪的功能，基带顶层维护了一个简单的TE FIFO类模型，模型除了实现基本的寄存器读写接口外，还实现了简单的read address和write address的计数。每向前推进1ms，l两个计数值同步向前推进1ms对应的样点个数。

基带顶层模型每次运行函数Process()向前仿真1ms，首先根据SignalSim的仿真模型计算向前推进1ms的当前时间，并根据轨迹模型重新计算当前位置，重新计算各个卫星的相应信息。在整分钟的时刻，还会重新计算可见星列表。然后把相应的信息送给跟踪引擎进行数据更新，最后根据情况设置中断标志位。

在同样的基带接口下，与基于C Model是基带软件仿真平台一样，main()函数只需要以下面简单的方式构造就可以运行起来：

```
#include <stdio.h>
#include <math.h>
#include <string.h>

#include "HWCtrl.h"
```

```
extern "C" {
#include "FirmwarePortal.h"
}

void main()
{
	SetInputFile("test_obs2.xml");

	FirmwareInitialize();
	EnableRF();
}
```

函数 SetInputFile()指定仿真对象为配置文件 test_obs2.xml，仿真所需的输入文件（包括 XML 文件本身和星历文件）可以在 SignalSim 工程中找到。

函数 FirmwareInitialize()进行 firmware 的初始化，包括创建相应的线程和信号量，注册中断处理函数以及初始化变量等。

函数 EnableRF()使能射频前端，产生中频采样信号和采样钟从而启动基带硬件并触发相应的中断处理函数。而在仿真平台上会运行底层仿真功能。