

LTE 系统的 Viterbi 译码算法仿真及 DSP 实现

罗友宝, 李小文

(重庆邮电大学 通信与信息工程学院, 重庆 400065)

摘要:文章基于长期演进(LTE)系统的咬尾卷积码,通过对多种 Viterbi 译码算法的仿真比较,为时分长期演进(TD-LTE)测试系统选择了一种最优的 Viterbi 译码算法,并在 TMS320C64x 数字信号处理器(DSP)中实现了这种算法。提出了一些具体的软件优化策略和技巧,对加比选蝶形运算进行了大量简化。通过译码程序在 CCS3.3 中的运行结果,验证了该算法及优化策略和技巧的可行性和有效性。

关键词:长期演进; Viterbi 译码; 软判决; 数字信号处理器实现

中图分类号: TN929.5 **文献标识码:** A **文章编号:** 1005-8788(2010)03-0067-04

Simulation and DSP realization of Viterbi decoding algorithm in LTE systems

Luo Youbao, Li Xiaowen

(School of Communication and Information Engineering, Chongqing University
of Posts and Telecommunications, Chongqing 400065, China)

Abstract: On the basis of the tailbiting convolutional code for Long Term Evolution (LTE) systems, this paper chooses an optimal Viterbi decoding algorithm for the TD-LTE test system after comparing multiple Viterbi algorithms by simulation and realizes it in the TMS320C64x DSP. It presents a number of specific strategies and techniques for software optimization and greatly simplifies the addition comparison selection butterfly computations. The running results of the decoding program in CCS3.3 verify the feasibility and effectiveness of this algorithm and the optimization strategies and techniques.

Key words: LTE; Viterbi decoding; soft-decision; DSP realization

在长期演进(LTE)系统中,为了获得正确无误的数据传输,采用了差错控制编码技术。咬尾卷积编码和 Viterbi 译码就是一种有效的前向纠错方法,它具有一定的克服突发错误的能力。与传统的卷积码^[1]相比,咬尾卷积码没有结尾清零比特,编码时使用码块的最后 m 个信息比特来初始化编码寄存器,使编码器的初始状态和结束状态相同,从而可以提高编码效率,节约带宽资源。Viterbi 译码算法是由 Viterbi 于 1967 年提出的一种最大似然译码算法,通过计算累计码距,在相应卷积码网格图上寻找最大似然路径,再根据这条路径所通过的延时器状态来重构发送序列。

数字信号处理器(DSP)具有精度高、灵活性好、可靠性高、时分复用和易于大规模集成等优点,得到了广泛的应用。在现代通信系统中,它占据着非常重要的地位,可以说在通信系统中越来越多的功能部件都是采用 DSP 技术实现的。本文研究了一种适用于时分一长期演进(TD-LTE)综合测试仪表系统的最优的 Viterbi 译码算法,并重点介绍了其在 TI 公司 DSP 芯片 TMS320C64x 上的实现。

1 TD-LTE 系统中的咬尾卷积码

LTE 作为准 4G 技术,以正交频分复用(OFDM)和多输入多输出(MIMO)技术为基础,下行采用 OFDM 多址技术,上行采用单载波频分多址(SC-FDMA)技术,在 20 MHz 频谱带宽下能够提供下行 100 Mbit/s、上行 50 Mbit/s 的峰值速率。

LTE TDD (亦称 TD-LTE)系统采用的是结构为(3,1,7)的咬尾卷积编码方案^[2],如图 1 所示,它使用码块的最后 6 bit 信息来初始化 6 个编码移位寄存器,使编码器的初始状态和结束状态相同。

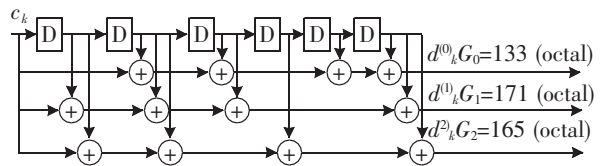


图 1 TD-LTE 系统的咬尾卷积编码器结构

2 Viterbi 译码算法的仿真与选择

与传统的 Viterbi 译码不同,咬尾卷积码的 Viterbi 译码不知道编码器的初始状态和结束状态,因

此,译码难度增加。目前咬尾卷积码 Viterbi 译码算法主要有循环维特比译码算法(CVA)和环绕维特比译码算法(WAVA)。通过研究分析可知,在众多的算法研究文献中,文献[3]提及的一种 M 比特重复循环维特比译码算法($N+M$ -CVA)可以说是后来很多改进算法的基础;文献[4]提出了一种有界距离循环维特比译码算法(BDD-CVA),这种算法通常只需将码块部分重复就可得到最终的译码结果,其优点是译码计算复杂度较低;WAVA^[5]通常要将码块重复多次才会得到最终的译码输出,其优点是误比特率相对降低了,但译码计算复杂度却相应增加了。在所有的 Viterbi 译码算法中,最大似然维特比译码算法(ML-VA)的误比特率是最低的,但是由于其惊人的译码计算复杂度,导致其几乎很难用于实际系统中。图 2、图 3 为以上各种译码算法的误比特率仿真比较,码块大小分别为 40 和 150 bit。其中 L 次迭代环绕维特比译码算法(L -WAVA)是在 WAVA 基础上稍加改进得到的。

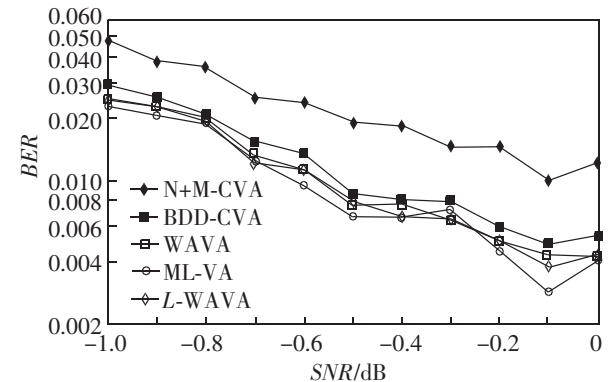


图 2 高斯信道下各算法误比特率(40 bit)

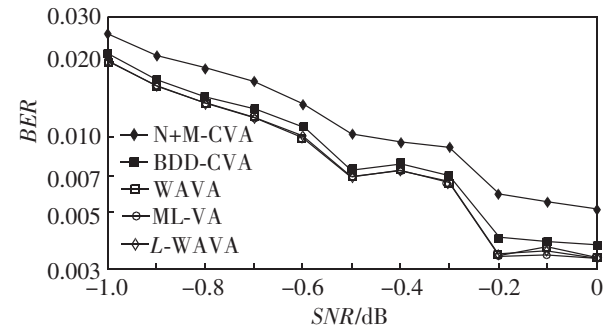


图 3 高斯信道下各算法误比特率(150 bit)

从图中可以看出不管码块的大小如何, L -WAVA 的译码误比特率都最接近 ML 算法。作为仪表系统的开发,系统对精度的要求比一般的终端系统更高,同时考虑到系统硬件配置也比一般的终端产品好很多的情况,系统最终采用了译码性能最优的 L -WAVA 算法,其算法描述如下:

(1) 初始化所有路径度量值为 0,对码块执行 VA 译码算法,第 1 次迭代结束后找出具有最大度量的路径,回溯,检查首末状态是否相同,相同则译码输出,译码结束;否则,进入下一步。

(2) 在第 i ($i > 1$) 次迭代开始时,用第 $i-1$ 次迭代结束时的状态度量信息初始化第 i 次迭代的初始状态度量,并继续进行 VA 译码算法;迭代结束时,选择最优路径回溯到初始时刻(0 时刻),比较第 1 次迭代到第 i 次迭代各次迭代首末状态是否相同,只要其中有一个相同,则译码输出对应迭代的码块,否则进行第 $i+1$ 次迭代,直到达到最大迭代次数 L 。

(3) 若在第 L 次迭代做完 VA 译码算法后没有发现任何一次迭代首末状态相同,则译码输出第 $[L/2]+1$ 次迭代的译码数据,将其作为最终的译码输出。

实验表明,本系统中 L 取值为 3 最为合适,既可使译码误比特率足够低,又不至于使译码延迟过大。可在保证系统高精度性能的前提下,尽可能地优化系统对其他资源的消耗,从而使整个系统性能最优。

3 Viterbi 算法的 DSP 实现

3.1 硬件简介

TMS320C6000 系列 DSP 是 TI 公司 1997 年 2 月推向市场的高性能 DSP,综合了目前 DSP 性价比高、功耗低等优点。TMS320C64x 系列在 TMS320C6000 DSP 芯片中处于领先水平,它不但提高了时钟频率,而且在体系结构上采用了 Velocity-TI 超长指令字(VLIW)结构^[6]。片内有 8 个独立功能单元的内核,每个周期可以并行执行 8 条 32 bit 指令,最大峰值速度 4 800 MI/s,2 组共 64 个 32 bit 通用寄存器,32 bit 寻址范围,支持 8/16/32/40 位的数据访问,片内集成大容量静态随机存取存储器(SRAM),最大可达 8 Mbit。由于具有出色的运算能力、高效的指令集、大容量的片内存储器和大范围的寻址能力,该系列特别适用于无线基站、测试仪表等对运算能力和存储量有高要求的应用场合。

3.2 Viterbi 译码算法处理流程

Viterbi 算法根据可能的状态转移进行译码,状态转移可用网格图表示。Viterbi 译码过程就是根据接收的数据符号,按最大似然译码准则找出编码器在网格图上所走的路径,其处理过程如图 4 所示。

由图 4 可以看出 Viterbi 算法的主要实现过程可分为 4 大部分:分支度量计算、加比选、幸存路径存储和回溯译码输出。

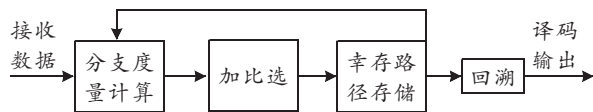


图 4 Viterbi 译码算法处理流程

3.3 Viterbi 译码算法程序实现

3.3.1 程序初始化

(1) 定义一个路径度量表 `state_metric`[64][2], 初始化清零。为了防止溢出, 表中每个单元内存大小分配一个字(32 bit), 在 VA 译码加比选的过程中交替更新为当前时刻的累积度量, 如偶时刻时第 2 列为被更新的度量, 奇时刻时第 1 列为被更新的度量, 这样在加比选结束时根据时刻的奇偶性即可判断最终的路径度量存储的位置, 从而查找出最优路径的末状态。

(2) 定义一个幸存路径状态表 `survivor_state`[64][3 * K], K 为码块的长度, 此表用于存储在整个译码过程中记录的所有可能的幸存状态, 因为状态值的范围为 0~63, 所以表中每个单元内存大小分配一个字节(8 bit)即可, 一个字节为 TMS320C64x 芯片可直接访问的最小数据单元。

(3) 定义一个幸存路径状态序列列表 `state_sequence`[3 * K], 与(2)同理, 表中每个单元内存大小分配一个字节(8 bit)即可, 回溯时将最优路径的各个状态存储到此表中, 最后判断各码块首末状态是否相同, 从而判断是否有译码输出, 结束译码。

(4) 定义一个输出码字常量表 `output_table`[32], 由于输出码字的范围为 0~7(8 进制), 因此, 为了便于访问, 表中每个单元内存大小分配一个字节(8 bit), 用于存储 j 状态在输入比特 0 时的输出码字, 以便计算各状态的分支度量。

3.3.2 度量值更新计算

Viterbi 算法的运算时间大部分消耗在距离量度的更新上, 每一个符号间隔期间, 距离量度需更新一次, 因此优化度量值的计算是降低整个译码过程时间消耗的关键, 整个实现方法的优劣性也都集中体现在此度量更新计算过程中。此过程可以分为以下 4 个方面:

(1) 分支度量简化。Viterbi 译码的分支度量计算分为硬判决和软判决两种, 硬判决用汉明距离表示, 软判决用欧氏距离表示。由于在误比特性能方面软判决比硬判决优 2 dB^[7], 因此本系统采用软

判决方法来计算分支度量。设符号间分支度量用 BM 表示, 根据欧氏距离计算公式有 $BM(n, i) = \sum_k [S_k(n) - G_k(n)]^2$, 式中, $S(n)$ 为接收数据序列, $G(n)$ 为网格期望输出码字, $k \in \{\text{与输入相关的编码输出比特}\}$, n 为时间, i 为要计算的路径。将 BM 表达式展开可得: $BM(n, i) = \sum_k [S_k^2(n) - 2S_k(n)G_k(n) + G_k^2(n)]$, 因为 $\sum_k S_k^2(n)$ 和 $\sum_k G_k^2(n)$ 在给定的符号期间对于所有的分支来说都是一样的, -2 也是一个常数, 在进行各路径度量值比较时可以不考虑。因此分支度量可简化为 $BM_1(n, i) = \sum_k S_k(n)G_k(n)$, 于是寻找具有最小分支度量 $BM(n, i)$ 的过程就转化为寻找具有最大分支度量 $BM_1(n, i)$ 的过程。

(2) 软比特量化。在接收端采用软解调, 使得接收数据在译码器的输入端量化为 3 bit, 量化范围为 $-4 \sim 3$, 即将 0 量化为 3, 1 量化为 -4 ; 将网格输出码字进行映射, 0 映射为 $+1$, 1 映射为 -1 , 这样在计算分支度量时就将乘法简化为加减法, 减少了运算量, 提高了运算速度。如接收到的码字为 $(-4, +3, +3)$, 网格输出码字为 $(1, 0, 1)$, 则分支度量为 $-(-4) + (+3) - (+3) = +4$ 。

(3) 蝶形运算。根据卷积编码的性质, 状态转移具有图 5 所示的蝶形结构。图 5 表明连续的两个状态有相同的转移状态, 各状态输入比特 0 与输入

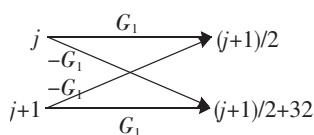


图 5 TD-LTE 系统中 (3, 1, 7) 卷积码蝶形结构

比特 1 有相反的分支度量, 且 j 状态与 $j+1$ 状态对称, 其中 j 应满足 $j \bmod 2 = 0$ 。因此, 利用此对称性, 在计算分支度量时只需计算状态 j 在输入比

特 0 时的分支度量 G_1 , 则输入比特 1 时的分支度量为 $-G_1$, $j+1$ 状态在输入比特 0 和 1 的分支度量分别为 $-G_1$ 和 G_1 , $-G_1$ 可通过 'NEG' 指令将 G_1 取反得到。这样, 每次计算一个分支度量就等于计算了两个状态、4 个输入比特的分支度量, 计算量降低了 $3/4$ 。

(4) 比较选择、保存最大累积度量和最优状态。将 j 和 $j+1$ 状态的累积度量与各自到达状态 $(j+1)/2$ 的分支度量相加, 得到 $(j+1)/2$ 处的两个累积度量, 进行比较, 将大的度量值存储到

state_metric,并将其相应的状态 j 或 $j+1$ 存储到幸存路径 survivor_state 中;到达状态 $(j+1)/2+32$ 的处理方法相同。

3.3.3 路径回溯

(1) 确定末状态。在加比选操作完成了一次迭代后,根据算法此时应回溯一次。首先查看当前时刻的奇偶性,如果为奇时刻,则在 state_metric[64][2]的第1列中查找最大度量值和其对应的状态,并将此状态作为回溯首状态,亦即最优路径的末状态;如果为偶时刻,则在 state_metric[64][2]的第2列中查找。

(2) 根据末状态回溯。将末状态存储到 state_sequence 中的对应位置,其值指示的为前一时刻 survivor_state 中的幸存状态的位置,依次向前回溯,取出最优路径的各状态存储到状态序列 state_sequence 中。

(3) 译码输出。在回溯完毕后,检查 state_sequence 中各次迭代的首末状态是否有相同。如果有相同,则译码输出相应码块的数据,将其作为最终的译码输出数据;如果没有相同,但是迭代次数达到了3次,则译码输出第2个码块的数据。译码输出实现方法如下:若(下一状态-当前状态 $\gg 1$) > 0 则输出1,否则输出0。

4 性能分析

在DSP软件实现中,通过指令并行,尽量优化程序^[8]循环体,减少或消除程序中的“NOP”指令,在码块大小为50 bit时,通过在CCS3.3上仿真运行程序,在理想信道情况下,整个译码过程耗时47 993个指令周期,此时因为输入数据没有错误,译码程序只迭代一次就得到了正确的译码输出;在信道条件极差的情况下,译码程序需迭代3次才能译码输出,此时译码耗时为142 545个指令周期。TMS320C64x芯片的主频一般为1 GHz,一个指令周期耗时为1 ns,因此本文中的DSP实现可达到约350.75~1 041.7 kbit/s的译码速率,且误比特

率相当低,满足系统的译码要求。

5 结束语

文章从理论仿真出发,根据TD-LTE系统特性,通过仿真比较为系统选择了一种最优的Viterbi译码算法,并在TMS320C64x芯片上实现了这种算法。重点讲述了Viterbi算法在DSP中的实现方法,程序运行结果表明,本算法具有较低的译码误比特率和较高的译码速率,能够满足LTE系统的需要,具有可行性和高效性。

参考文献:

- [1] 张宗橙. 纠错编码原理和应用 [M]. 北京:电子工业出版社,2003.150-198.
- [2] 3GPP TS 36. 212 v8. 7. 0-2009. Multiplexing and channel coding (Release 8) [S].
- [3] CoxAn Richard V, Sundberg Carl-Erik W. An Efficient Adaptive Circular Viterbi Algorithm for Decoding generaliiized Tailbiting convolutional Codes [J]. IEEE Transactions on Communications,1994,43(1):57-68.
- [4] Anderson John B, HladikStephen M. An Optimal Circular Viterbi Decoder for the Bounded Distance Criterion [J]. IEEE Transactions on Communications, 2002, 50(11):1 736-1 742.
- [5] Shao Rose Y, Lin Shu, Fossorier Marc P C. Two Decoding Algorithms for Tailbiting Codes [J]. IEEE Transactions on Communications,2003,51(10):1 658-1 665.
- [6] Texas Instruments Incorporated. TMS320C64x/C64x +DSP CPU and Instruction Set Reference Guide [EB/OL]. <http://www.ti.com.cn>,2008.
- [7] 张力军,张宗橙,郑宝玉,等(译),Proakis John G(著). 数字通信(第四版) [M]. 北京:电子工业出版社,2003.340-371.
- [8] 田黎育,何佩琨,朱梦宇(译). Texas Instruments Incorporated. TMS320C6000 系列 DSP 编程工具与指南 [M]. 北京:清华大学出版社,2006.32-50.