

基于CORDIC的梯度跟踪算法的FPGA实现

王 韬¹, 罗明阳^{1,2}, 吴 军¹, 刘 波¹, 张建慧¹

(1. 重庆大学通信工程学院, 重庆 400044; 2. 重庆金美通信有限责任公司, 重庆 400044)

摘要: 梯度跟踪算法在多飞行器测控及卫星移动通信中有广泛的应用, 但现有的实现方案还存在很多缺点。该文以均匀圆阵为接收天线阵, 在分析梯度跟踪算法原理及CORDIC算法原理的基础上, 提出一种基于CORDIC算法与数字波束形成技术相结合的FPGA实现方案, 给出电路实现原理图并使用Quartus软件进行了仿真分析, 仿真结果表明该方案是正确、可行和有效的。

关键词: 梯度跟踪算法; 数字波束形成技术; CORDIC算法; 现场可编程逻辑阵列

FPGA Implementation of Gradient Tracking Algorithm Based on CORDIC

WANG Tao¹, LUO Ming-yang^{1,2}, WU Jun¹, LIU Bo¹, ZHANG Jian-hui¹

(1. Communication Engineering College, Chongqing University, Chongqing 400044;

2. Chongqing Jinmei Communication Co. Ltd., Chongqing 400044)

【Abstract】 Gradient tracking algorithm is widely used in multiple-spacecraft TT&C system and satellite mobile communication system, but the existing hardware implementation method has many drawbacks. This paper analyzes the basic principle of gradient tracking based on uniform circular array and the principle of CORDIC, and puts forward a new FPGA implementation method using CORDIC algorithm and digital beam forming technology. The circuit diagram is designed and the simulation result by using quartus software is also carried out. The simulation proves that the method is correct, more effective and feasible.

【Key words】 gradient tracking algorithm; Digital Beam Forming(DBF); CORDIC algorithm; FPGA

1 概述

在航天测控或移动卫星通信中, 地面站大多利用一个大的窄波束(高增益)天线捕获并跟踪空中目标, 待跟踪稳定后, 再对目标进行测控或通信。其跟踪过程为: 由传感器检测出天线电轴与激光陀螺惯性轴之间的角误差, 驱动伺服电机, 最终使天线电轴对准激光陀螺惯性轴, 从而实现天线对空中目标的自动跟踪^[1-2]。文献[2-3]提出了梯度跟踪算法并设计了单片微机控制电路来驱动伺服系统, 但由于引导和伺服系统不可能使庞大的硬天线在多个航天器之间迅速跳动跟踪, 因此很难实现对多个目标同时进行跟踪测控。数字波束形成(Digital Beam Forming, DBF)技术可解决这一问题。文献[1,4-5]提出了基于天线阵的多目标跟踪系统, 并用Matlab和mathcad软件进行了仿真分析。文献[5]研究了梯度跟踪算法的DSP实现, 它将各天线阵元的基带数据串行送入TI公司的DSP处理器, 在DSP处理器上进行数字波束形成及梯度跟踪算法运算, 但这种方法的缺点是处理速度慢, 不能并行形成多波束同时跟踪多目标。文献[6]给出了一种数字波束形成处理器结构, 使用FPGA实现数字波束形成运算, 使用DSP处理器进行自适应跟踪权值调整, 但这种波束形成方法需要用到很多复数乘法器, 占用FPGA芯片资源多, 当需要跟踪多个目标时, 这种资源消耗所付出的代价会更加明显。针对以上这些问题, 本文提出了一种基于CORDIC的梯度跟踪算法的FPGA实现方案。

2 基于均匀圆阵的梯度跟踪算法原理

本文拟采用二维均匀圆阵作跟踪天线, 设8个全向阵元均匀地分布在半径为r的圆周上, 如图1所示。

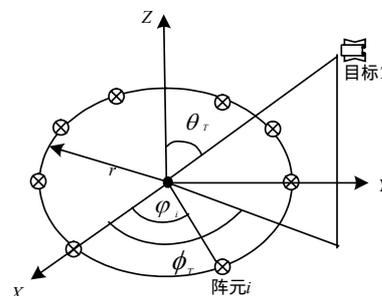


图1 8元均匀圆阵

圆阵的阵因子可以表示为

$$U(\theta, \phi, \alpha_n) = \sum_{n=1}^N \exp \left\{ j \left[\alpha_n + \frac{2\pi}{\lambda} r \sin \theta \cos(\phi - \phi_n) \right] \right\}$$

为使 $U(\theta, \phi, \alpha_n)$ 对准方向 (θ_r, ϕ_r) , 即天线阵获得最大接

收电平, 应使 $\alpha_n = -\frac{2\pi}{\lambda} r \sin \theta_r \cos(\phi_n - \phi_r)$ 。

圆阵方向图的梯度可以表示为

$$\nabla_{\phi} = \frac{d|U(\theta, \phi, \alpha_n)|}{d\phi} = \frac{d}{d\phi} \left| \sum_{n=1}^N \exp \left\{ j \left[\alpha_n + \frac{2\pi}{\lambda} r \sin \theta \cos(\phi - \phi_n) \right] \right\} \right|$$

基金项目: 重庆市院士基金资助项目(102075220050138); 重庆大学大学生创新基金资助项目(2007047)

作者简介: 王 韬(1977 -), 男, 讲师、博士, 主研方向: 嵌入式系统设计及应用, 运载器测控及遥感信息传输技术; 罗明阳, 工程师、硕士; 吴 军、刘 波、张建慧, 本科生

收稿日期: 2007-08-25 **E-mail:** wtccce@163.com

$$\nabla_{\theta} = \frac{d|U(\theta, \phi, \alpha_n)|}{d\theta} = \frac{d}{d\theta} \left| \sum_{n=1}^N \exp \left\{ j \left[\alpha_n + \frac{2\pi}{\lambda} r \sin \theta \cos(\phi - \varphi_n) \right] \right\} \right|$$

梯度跟踪的过程就是不断调整 (θ, ϕ) ，迭代关系式为

$$\nabla_{\phi} = \frac{U_2 - U_1}{\Delta \phi}$$

$$\phi_2 = \phi_1 + k_{\phi} \nabla_{\phi}$$

$$\nabla_{\theta} = \frac{U_2 - U_1}{\Delta \theta}$$

$$\theta_2 = \theta_1 + k_{\theta} \nabla_{\theta}$$

在实际应用中步长控制因子 k_{ϕ} 与 k_{θ} 在实验中调整,取值太大,容易发散,取值太小收敛速度慢。当 $|\nabla_{\phi}| < \varepsilon_{\phi}$ 即可认为完成了方位角跟踪,当 $|\nabla_{\theta}| < \varepsilon_{\theta}$ 即可认为完成俯仰角跟踪, ε_{ϕ} 和 ε_{θ} 为预先设置的任意小常数。

梯度跟踪的过程按时分复用方式进行,可先进行方位角跟踪后再进行俯仰角跟踪,这样可以节省波束形成模块。

3 CORDIC 算法的基本原理

文献[7]于 1959 年在美国航空控制系统的设计中提出了 CORDIC 算法, J.Walther 在 1974 年用它研究了一种能计算出多种超越函数统一算法。其基本思想是用一系列角度不断偏摆从而逼近所需旋转的角度。选取固定的旋转角度恰好使得计算只有移位和相加,因此,该算法比较适合使用 FPGA 实现。CORDIC 算法能有效地实现三角函数的计算以及极坐标和直角坐标系的变换^[8]。

3.1 CORDIC 算法的旋转模式

为避免除法和均方根操作,人们提出了一个改进的方法,即“double-rotation”法,通过旋转 2 次 $z_0/2$ (z_0 为要旋转的角度)来解决这个问题,旋转公式如下:

$$\begin{cases} x_{i+1} = (1-2^{-2i})x_i - d_i \cdot y_i \cdot 2^{-i+1} \\ y_{i+1} = (1-2^{-2i})y_i + d_i \cdot x_i \cdot 2^{-i+1} \\ z_{i+1} = z_i - d_i \tan^{-1}(2^{-i}) \end{cases} \quad (1)$$

$$d_i = -1 \text{ if } z_i < 0; +1 \text{ otherwise}$$

经过有限次旋转后,得到如下结果

$$\begin{cases} x_n = A_n^{(ii)}(x_0 \cos z_0 - y_0 \sin z_0) \\ y_n = A_n^{(ii)}(y_0 \cos z_0 + x_0 \sin z_0) \\ z_n = 0 \end{cases}$$

$$A_n^{(ii)} = \{A_n\}^2 = \prod_{i=0}^n 1 + 2^{-2i}$$

新的计算增益为 $A_n^{(ii)}$ 。则新的缩放因子为

$$K_n^{(ii)} = \frac{1}{\{A_n\}^2} = \prod_{i=0}^n 1 + 2^{-2i} \approx \frac{1}{2} \prod_{i=1}^n (1 + 2^{-4(i-2)})$$

新的缩放因子可以通过移位与加法器来实现。

3.2 CORDIC 算法的矢量模式

$$\begin{cases} x_{i+1} = x_i - d_i \cdot y_i \cdot 2^{-i} \\ y_{i+1} = y_i + d_i \cdot x_i \cdot 2^{-i} \\ z_{i+1} = z_i - d_i \tan^{-1}(2^{-i}) \end{cases}$$

$$d_i = 1 \text{ if } y_i < 0; -1 \text{ otherwise}$$

在 VECTOR 模式中,CORDIC 旋转器将输入向量旋转任意需要角度,而使结果向量与 X 轴重合。它完成直角坐标到极坐标的变换。

CORDIC VECTOR 模式下,迭代公式如下:

$$\text{经过 } n \text{ 次迭代后, } \begin{cases} x_n = A_n \sqrt{x_0^2 + y_0^2} \\ y_n = 0 \\ z_n = z_0 + \tan^{-1}(y_0/x_0) \\ A_n = \prod_{i=1}^n \sqrt{1 + 2^{-2i}} \end{cases}$$

其中, z_n 即为得到的相角; x_n/A_n 即为模值, $1/A_n \approx 0.60725$ 。

3.3 三角函数值的求解

当 CORDIC 工作于 ROTATION 模式时,能同时计算输入角度的正弦和余弦函数,这时设置输入矢量的 y 部分为 0,设置 $x_0 = 1/A_n$, z_0 为输入的角度值。

经过 n 次旋转后,结果为

$$\begin{cases} x_n = A_n x_0 \cos z_0 \\ y_n = A_n x_0 \sin z_0 \end{cases} \quad (2)$$

4 基于 CORDIC 的梯度跟踪算法的 FPGA 实现

4.1 基于 CORDIC 算法的数字波束形成模块

设第 i 个阵元接收到的信号电平为 $X_i = X_{ii} + jX_{Qi}$ 。

其复加权系数为 $w_i = w_{ii} + jw_{Qi} = \exp(j\alpha_i)$ 。

复加权后的输出为

$$X'_i = X_{ii} + jX'_{Qi} = X_i * w_i^* = (X_{ii} + jX_{Qi}) * \exp(-j\alpha_i)$$

其中,

$$\alpha_i = \frac{2\pi r}{\lambda} \sin \theta \cos(\phi - \varphi_i) = \frac{1}{2} \pi \times \left(\cos\left(\frac{\pi}{2} - \theta + \phi - \varphi_i\right) + \cos\left(\frac{\pi}{2} - \theta - \phi + \varphi_i\right) \right)$$

$$\varphi_i = \frac{2(i-1)\pi}{N}, i = 1, 2, \dots, N$$

通过上述的转换,可以看出接收天线阵的每一路复数加权,通过处理可以使用 3 个 CORDIC 算法模块实现:即 2 个求余弦的操作 $\cos(\frac{\pi}{2} - \theta + \phi - \varphi_i)$, $\cos(\frac{\pi}{2} - \theta - \phi + \varphi_i)$ 和 1 个 CORDIC 旋转模块(将每个阵元接收到的复信号旋转 $-\frac{2\pi r}{\lambda} \sin \theta \cos(\phi - \varphi_i)$)。图 2 是单路复加权模块的 FPGA 内部实现原理图。

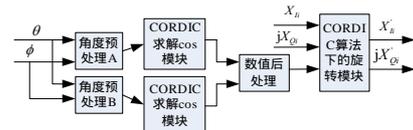


图 2 单路阵元复加权实现模块

要使用梯度算法实现对目标的跟踪,需要将各阵元的复加权后的 I, Q 两路信号进行并行加法操作,得到阵列合成后的总输出 $X'_{\Sigma} = X'_{I\Sigma} + jX'_{Q\Sigma}$,需计算改变 (θ, ϕ) 前后的变化。这里可以使用 CORDIC 算法的 VECTOR 模式,直接求出合成后的信号电平 $|X'_{\Sigma}|$,图 3 是 8 元圆阵数字波束形成模块的原理图。

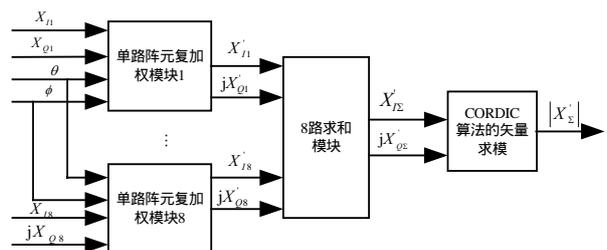


图 3 8 路数字波束形成模块

4.2 梯度算法完成模块

梯度算法完成模块主要包括 4 个模块：

(1) 梯度算法时钟控制模块。用于将时钟(Clk)分割成两部分(Clk_g & Clk_d), 分别驱动梯度算法模块和数字波束形成模块, 完成时钟的分时复用。

(2) 梯度算法角度控制模块。完成不同跟踪模式(方位角 ϕ 跟踪模式和俯仰角 θ 跟踪模式)下的角度分配工作, 从而完成 2 个跟踪模式的角度切换。

(3) 梯度算法模块。是整个梯度算法的核心控制模块。整个跟踪过程分 2 种模式完成, 先进行方位角 ϕ 跟踪, 完成后再启动俯仰角 θ 跟踪。2 种跟踪分时完成, 模块共用。每完成一次计算, 根据第 2 节中的梯度迭代计算公式, 计算出 $\phi_2 = \phi_1 + k_\phi \nabla_\phi$ 或 $\theta_2 = \theta_1 + k_\theta \nabla_\theta$, 并设置相应的控制线, 然后输出给角度控制模块, 由其分配给数字波束形成模块进行下次求解。满足 $\nabla_\phi < \varepsilon_\phi$ 后, 方位角跟踪结束, 启动俯仰角跟踪, 直到满足 $\nabla_\theta < \varepsilon_\theta$, 整个跟踪完成, 控制信号关闭时钟控制模块, 可由 Start 上升沿重新启动。

程序中 ϕ, θ 的初始步长可调, 可以设置为不同的数, 步长控制因子 k_ϕ, k_θ 也可以设置为不同的系数, 关于步长控制因子的选择, 本文不作讨论。

(4) 波束形成模块。已在 4.1 节中进行了详细说明, 此处不再累述。

图 4 为梯度算法模块原理框图。

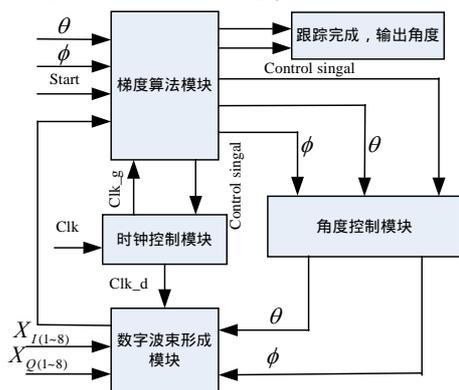


图 4 梯度算法模块

5 梯度跟踪算法的 FPGA 实现仿真分析

本文设计的梯度跟踪算法的仿真结果(见表 1)给出了俯仰角和方位角随迭代次数增加的变化过程。其仿真条件是半径 $r = \lambda/2$ 的 8 元均匀圆阵, 起始跟踪时的方位角和俯仰角分别为 40.48° (16 进制数为 1CC77)和 60.42° (16 进制数为 2AF5B)(在实际应用中这些角度是通过使用UCA-ESPRIT或者UCA-MUSIC算法计算得到的^[7-8])。假设当前目标的实际方位角和俯仰角分别为 44° 和 56° , 方位角跟踪的步长控制因子 $k_\phi = 19.32$, 方位角跟踪的收敛条件是 $|\nabla_\phi| < 0.0028$, 俯仰角跟踪的步长控制因子 $k_\theta = 33.81$, 俯仰角跟踪的收敛条件为 $|\nabla_\theta| < 0.0028$ 。表 1 中的 ϕ 和 θ 反映了方位角和俯仰角在跟踪过程中的变化情况, 上面一行是用 16 进制表示的角度, 下面一行是对应的 10 进制数表示的角度。梯度跟踪模块首先进行方位角跟踪(ϕ 变化, θ 不变), 当方位角跟踪完成后进入俯仰角跟踪阶段(θ 变化, ϕ 不变)。跟踪过程结束后跟踪到的目标方位角为 43.74° (16 进制数为 1F18F)和俯仰角为 56.51° (16 进制数为 282D7), 比较接近目标的实际方位, 即

方位角为 44° 和俯仰角 56° 。

整个基于数字波束和梯度算法的跟踪系统占用 17 474 个 LE, 6 163 个寄存器, 338 个管脚, 可以选用 Altera 公司的 CYCLONE 系列 FPGA 芯片 EP2C35F672C6, 该芯片有 33 216 个 LE 和 475 个管脚。

梯度计算模块需要 13 个时钟, 数字波束形成需要 68 个时钟, 因而完成 1 次梯度迭代运算需要 81 个时钟。在本例中完成梯度跟踪过程共消耗了 29 个周期, 计 2 349 个系统时钟。梯度跟踪过程消耗的总周期数与初始步长, 步长控制因子, 收敛门限设置以及跟踪起始角度与实际目标方位差等有关。

表 1 俯仰角和方位角的跟踪变化情况

次数	ϕ	θ	次数	ϕ	θ
1	1CC77	2AF5B	16	1F18F	2B4EF
	40.47	60.41		43.73	60.90
2	1D7D7	2AF5B	17	1F18F	2AE51
	41.47	60.41		43.73	60.32
3	1DD9F	2AF5B	18	1F18F	2A98F
	41.98	60.41		43.73	59.90
4	1E217	2AF5B	19	1F18F	2A3B5
	42.37	60.41		43.74	59.39
5	1E4BF	2AF5B	20	1F18F	2A027
	42.60	60.41		43.74	59.08
6	1E87F	2AF5B	21	1F18F	29C37
	42.93	60.41		43.74	58.73
7	1EA9F	2AF5B	22	1F18F	298A9
	43.12	60.41		43.74	58.42
8	1EC7F	2AF5B	23	1F18F	294B9
	43.29	60.41		43.74	58.07
9	1EE9F	2AF5B	24	1F18F	2912B
	43.47	60.41		43.74	57.76
10	1F07F	2AF5B	25	1F18F	28E37
	43.64	60.41		43.73	57.50
11	1EF6F	2AF5B	26	1F18F	28D11
	43.54	60.41		43.73	57.40
12	1ED8F	2AF5B	27	1F18F	28A0F
	43.38	60.41		43.73	57.13
13	1EFAF	2AF5B	28	1F18F	288E9
	43.57	60.41		43.73	57.03
14	1F18F	2AF5B	29	1F18F	282D7
	43.73	60.41		43.73	56.51
15	1F18F	2BABB	30	1F18F	282D7
	43.73	61.41		43.73	56.51

6 结束语

本文提出了一种基于CORDIC的梯度跟踪算法的FPGA实现方案, 给出了电路原理图, 并使用Quartus软件进行了仿真分析。仿真结果表明所提方案是可行的。由于波束形成模块部分采用了迭代结构的CORDIC算法, 从而节省了大量乘法器的使用, 并且方位角跟踪和俯仰角跟踪是时分复用的, 从而节省了FPGA芯片的大量资源。在同一块FPGA芯片内部实现整个跟踪算法, 与基于FPGA和DSP相结合的方案相比减少了芯片间的通信开销, 加上芯片内部没有使用乘法器, 所以该系统具有更快的处理速度, 完成一次梯度迭代运算仅需 81 个时钟周期。本文讨论的是跟踪单个目标的情况, 如要跟踪多个目标, 只需并行添加上述多个并行的波束形成模块即可。

参考文献

- [1] 杨士中, 杨力生, 谭晓衡. 多飞行器测控技术的研究[J]. 宇航学报, 2002, 23(6): 12-18.
- [2] 杨士中. 移动卫星通信系统[P]. 中国专利, 02126611.5. 2004-01-21.
- [3] 邓晓琴. 梯度法——移动卫星通信天线自动跟踪的新方法[J]. 信息与电子工程, 2003, 1(3): 33-37.
- [4] 余平, 廖明, 黄天聪. 基于数字波束形成的自动跟踪系统[J]. 电讯技术, 2004, 44(2): 94-98.
- [5] 韩庆文. 基于多飞行器测控系统的数字波束形成研究[D]. 重庆: 重庆大学, 2003.

(下转第 31 页)