doi: 10.7690/bgzdh.2020.06.011

反正切函数的 CORDIC 算法及其 FPGA 实现

王　强，应　浩
（南京模拟技术研究所无人机事业部，南京 210016）

摘要：为解决传统的反三角函数求解方法存在速度低、精度低和不方便实现的问题，提出一种基于 FPGA 控制器实现运动平台的运动学逆解算法。根据 CORDIC 算法原理，以圆周坐标系统的旋转模式和向量化模式为例进行理论分析，构建基于 Nios Ⅱ 软核处理器的 SOPC 系统架构，利用 C 语言编写逆向运动学求解模块，并通过 ModelSim 仿真软件对逆向运动学理论进行仿真验证。测试结果表明，采用 FPGA 实现运动平台的逆向运动学求解算法是正确的。

关键词：反正切函数；FPGA；运动学逆解；CORDIC
中图分类号：TP302　　**文献标志码：**A

CORDIC Algorithm of Arctangent Function and Its FPGA Implementation

Wang Qiang, Ying Hao
(*Unmanned Aerial Vehicle Division*, *Nanjing Research Institute on Simulation Technique*, *Nanjing* 210016, *China*)

Abstract: In order to solve the problem of low speed, low precision and inconvenient realization of the traditional inverse trigonometric method, a kinematics inverse solution algorithm of motion platform based on FPGA controller. According to the principle of CORDIC algorithm, taking the rotating mode and vector quantization mode of the circular coordinate system as an example, the SOPC system architecture based on the Nios II soft core processor is constructed, and the reverse kinematics solution module is written using the C language. The theory of reverse kinematics is validated by the simulation software of Modelsim. The test results show that the reverse kinematics algorithm is correct by using FPGA.

Keywords: arctangent function; FPGA; kinematics inverse solution; CORDIC

0　引言

三角函数及反三角函数运算求解的应用领域非常广泛，包括机器人运动学求解领域[1]、通信载波的调制与解调领域、协处理器领域、雷达信号处理领域、图像处理领域、导航数字处理领域和计算器领域等。运动平台的逆向运动学求解方程中会存在大量的反三角函数的求解运算[2-3]，传统的反三角函数求解方法有 ROM 查找表法、多项式近似法和迭代法等，但存在速度低、ROM 硬件资源少、精度低和不方便硬件实现等缺点[4]；因此，急需一种兼容速度快、精度高和方便硬件资源实现的综合方法。坐标旋转数字计算机(coordinate rotation digital computer，CORDIC)[5]算法理论的提出解决了当时的问题，得到广泛的应用和推广。笔者分析了 CORDIC 算法的基本原理，提出将 CORDIC 算法应用于运动平台运动学逆解的反正切函数中，设计一种三角函数和反三角函数的 CORDIC 算法研究及 FPGA 实现。采用 CORDIC 算法替代传统的通过计算大量超越方程才能完成机械臂运动学正反解计算，使超越方程计算过程简化为只需要通过加、减法和移位等 FPGA(field programmable gate array)硬件便于实现的操作，从而提高了运动平台逆解模块的计算速度。笔者在 EP2C8Q208C8 的 FPGA 硬件平台上实现了该算法，并通过 ModelSim 仿真软件验证了该算法的正确性和有效性。

1　CORDIC 算法原理

CORDIC 是一种使用多次预定角度的不断旋转来逼近初值角度的近似逼近方法。1957 年，Jack Volder 最早提出 CORDIC 算法，并应用于实时导航的数字处理。John Walther 在 Volder 的基础上进行改进和推广，于 1971 年提出统一的 CORDIC 算法，具有更加广泛的应用范围，如三角函数、反三角函数、双曲线函数和超越函数等。如表 1 所示，CORDIC 算法包括旋转模式和向量化模式，可以在圆周坐标系统、线性坐标系统和双曲坐标系统中完成 2 种不同的模式，获得 6 种不同的结果。

收稿日期：2020-02-20；**修回日期：**2020-04-03
作者简介：王　强(1989—)，男，辽宁人，硕士，工程师，从事无人直升机航电系统、伺服控制系统、BLDC 控制等研究。E-mail: 174962512wq@163.com。

表 1 统一 CORDIC 算法的 3 种系统 2 种不同应用模式

S	旋转模式 $d_i=\text{sign}[z_i];z_i\to 0$	向量模式 $d_i=\text{sign}[X_i\,Y_i];Y_i\to 0$
圆周系统 S=1 u=1 $e_i=\tan^{-1}2^{-i}$ $k=\prod_{n=0}^{N}\sqrt{1+2^{-2n}}$	$x\to$ CORDIC $\to k(x\cos z-y\sin z)$ $y\to$ CORDIC $\to k(y\cos z+x\sin z)$ $z\to$ CORDIC $\to 0$ 基本模块 1：计算 cos$z$ 和 sin$z$，令 $x=1/K$，$y=0$	$x\to$ CORDIC $\to k(x^2+y^2)^{1/2}$ $y\to$ CORDIC $\to 0$ $z\to$ CORDIC $\to z+\tan^{-1}(y/x)$ 基本模块 2：计算 $\tan^{-1}y$，令 $x=1$，$z=0$
线性系统 S=0 u=0 $e_i=2^{-i}$	$x\to$ CORDIC $\to x$ $y\to$ CORDIC $\to y+(x-z)$ $z\to$ CORDIC $\to 0$ 基本模块 3：计算乘法，令 $y=0$	$x\to$ CORDIC $\to x$ $y\to$ CORDIC $\to 0$ $z\to$ CORDIC $\to z+(y/x)$ 基本模块 4：计算除法，令 $z=0$
双曲系统 S=−1 u=−1 $e_i=\tanh^{-1}2^{-i}$ $k^h=\prod_{n=1}^{N}\sqrt{1-2^{-2n}}$	$x\to$ CORDIC $\to k^*(x\cosh z-y\sinh z)$ $y\to$ CORDIC $\to k^*(y\cosh z-x\sinh z)$ $z\to$ CORDIC $\to 0$ 基本模块 5：计算 cosh z 和 sinh z，令 $x=1/k^k$，$y=0$	$x\to$ CORDIC $\to k^*(x^2-y^2)^{1/2}$ $y\to$ CORDIC $\to 0$ $z\to$ CORDIC $\to z+\tanh^{-1}(y/x)$ 基本模块 6：计算 $\tanh^{-1}y$，令 $x=1$，$z=0$

表中列出了统一 CORDIC 算法的 3 种坐标系统和 2 种模式，组成 6 种不同的 CORDIC 算法，分别用于求解不同的应用函数。由于 CORDIC 算法实现方式较多，笔者将以圆周坐标系统的旋转模式和向量化模式为例进行理论分析，并采用 FPGA 将其分别实现。

圆坐标系统下旋转模式的CORDIC算法可以求正弦函数 sin 和余弦函数 cos。如图 1 所示，假设在圆坐标系中，已知向量(X_1, Y_1)，经过逆时针旋转角度 θ 得到向量 X_2, Y_2。

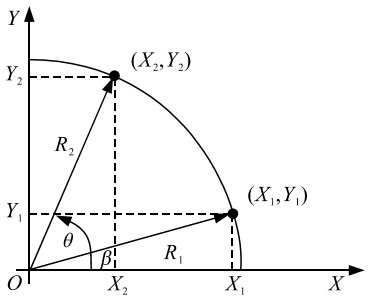

图 1 圆坐标系下的旋转模式

图中，$R_1=R_2=R$，经过向量旋转后得到：

$$\left.\begin{aligned}X_2&=R\cos(\theta+\beta)=X_1\cos\theta-Y_1\sin\theta\\Y_2&=R\sin(\theta+\beta)=X_1\sin\theta+Y_1\cos\theta\end{aligned}\right\}。\tag{1}$$

将上式写成矩阵形式：

$$\begin{bmatrix}X_2\\Y_2\end{bmatrix}=\begin{bmatrix}\cos\theta&-\sin\theta\\\sin\theta&\cos\theta\end{bmatrix}\begin{bmatrix}X_1\\Y_1\end{bmatrix}=\cos\theta\begin{bmatrix}1&-\tan\theta\\\tan\theta&1\end{bmatrix}\begin{bmatrix}X_1\\Y_1\end{bmatrix}。\tag{2}$$

如图 2 所示，如果想旋转任意角度 θ，可以通过 n 次连续旋转一系列预定的角度 θ_i 来完成。

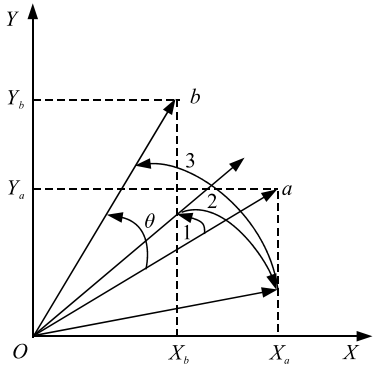

图 2 多次旋转图

图中每次旋转的方向为 d_i，顺时针旋转时 d_i=−1，而逆时针旋转时 d_i=1，则最终旋转的角度可以表示为：

$$\theta=\sum_{i=0}^{n-1}d_i\theta_i,\ d_i\in\{-1,1\}。\tag{3}$$

圆坐标系下的旋转模式 CORDIC 算法第 i+1 次迭代方程可以表示为：

$$\begin{cases}X_{i+1}=X_i-d_i2^{-i}Y_i\\Y_{i+1}=Y_i+d_i2^{-i}X_i\\Z_{i+1}=Z_i-d_i\arctan(2^{-i})\end{cases},\ d_i=\begin{cases}-1,Z_i<0\\+1,Z_i\geqslant 0\end{cases}。\tag{4}$$

式中 $\tan(\theta_i)=\pm 2^i=d_i\times 2^i$，$d_i$ 为旋转方向，由 z_{i+1} 角度累加值决定下一次旋转的方向。在这种模式下，CORDIC 旋转器对输入向量经过 n 次角度旋转，得到最终结果为

$$\left.\begin{aligned}X_n&=A_n\left[X_0\cos Z_0-Y_0\sin Z_0\right]\\Y_n&=A_n\left[Y_0\cos Z_0+X_0\sin Z_0\right]\\Z_n&=0\end{aligned}\right\}。\tag{5}$$

式中 $A_n=\prod_n\sqrt{1+2^{-2i}}\approx 1.647$。当 $X_0=1/A_n$，$Y_0=0$ 和 $Z_0=\text{angle\_input}$ 时，经过 n 次迭代后可直接得到 $X_n=\cos Z_0$，$Y_n=\sin Z_0$和Z_n=0 。以上就是采用 CORDIC 算法的旋转模式，求正弦函数和余弦函数的基本原理。

圆坐标系下的向量化模式的CORDIC算法可以求反正切函数 arctan 和向量的模值，其原理的推导过程与旋转模式类似。向量化模式与旋转模式不同之处在于：输入向量经过一些列旋转后，最终使向量与 x 轴对齐，且旋转方向 d_i 由 Y_i 决定。圆坐标系下的向量化模式的 CORDIC 算法第 i+1 次迭代方程可以表示为：

$$\begin{cases} X_{i+1}=X_i-d_i2^{-i}Y_i \\ Y_{i+1}=Y_i+d_i2^{-i}X_i \\ Z_{i+1}=Z_i-d_i\arctan(2^{-i}) \end{cases},d_i=\begin{cases}+1,Y_i<0\\-1,Y_i\geqslant 0\end{cases}。\quad (6)$$

在这种模式下，CORDIC 旋转器对输入向量经过 n 次角度旋转，得到最终结果为

$$\left.\begin{aligned} X_n&=A_n\sqrt{X_0^2+Y_0^2}\\ Y_n&=0\\ Z_n&=Z_0+\arctan(Y_0/X_0)\end{aligned}\right\}。\quad (7)$$

式中 $A_n=\prod_n\sqrt{1+2^{-2i}}\approx 1.647$。当 $X_0=x\_\text{input}$，$Y_0=y_\text{input}$和$Z_0=0$时，经过 n 次迭代后可直接得到向量(X_0,Y_0)的模值 $X_n=A_n\sqrt{X_0^2+Y_0^2}$ 和反正切函数 $Z_n=\arctan(Y_0/X_0)$。以上是采用 CORDIC 算法的向量化模式求反正切函数 arctan 和向量模值的基本原理。

分析 CORDIC 算法的 2 种模式发现，使用圆周坐标系、线性坐标系和双曲坐标系具有一定的相似性，通过模式变量 m 可以选择不同的坐标系统为：

$$\begin{cases} X_{i+1}=X_i-m\cdot d_i2^{-i}Y_i \\ Y_{i+1}=Y_i+d_i2^{-i}X_i \\ Z_{i+1}=Z_i-d_i\cdot e_i \end{cases},\text{其中}e_i=\begin{cases}\arctan(2^{-i}),m=1\\2^{-i},\quad m=0\\\arctan(2^{-i}),m=-1\end{cases}。\quad (8)$$

上式为统一的 CORDIC 算法迭代方程。当 m=1 时，选择圆周坐标系统；当 m=0 时，选择线性坐标系统；当 m=−1 时，选择双曲线坐标系统。

2　三角函数及反三角函数的 FPGA 实现

三角函数和反三角函数的传统求解方法包括查找表法、多项式近似法和迭代法等，与其他方法相比，CORDIC 算法的求解方法具有显著的优势。如运算的整个过程只包括加法、减法和移位运算，不包含乘法运算，节约了硬件乘法器模块；方便 FPGA 硬件实现，可采用流水线结构提高系统工作频率，也可采用循环迭代结构节约硬件逻辑资源；增加角度旋转次数提高求解精度等[6]。笔者分析了圆坐标系下 CORDIC 算法的旋转模式和向量化模式原理基础，使用 FPGA 实现统一的 CORDIC 算法的正弦函数、余弦函数和反正切函数的过程如下。

统一 CORDIC 算法的实现可以使用迭代法或流水线结构，根据系统的速度和资源的需求来选择合适的方法。如图 3 所示，笔者采用并行流水线结构实现统一的 CORDIC 算法。

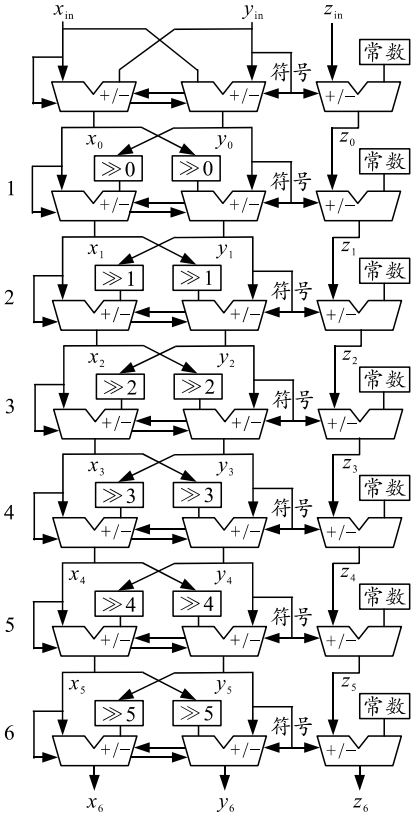

图 3　统一 CORDIC 算法的并行流水线结构

图中包括 6 Level 流水线结构，如果要提高运算结果的精度，则增加流水线级数即可。在 CORDIC 算法中，由于输入量、中间量和输出量都为浮点数，并且可能是有符号浮点数形式，将增加 FPGA 实现 CORDIC 算法的难度和降低运算精度；因此，必须考虑这些数据运算的格式问题[7]，一般采用十进制有符号浮点数转换为二进制有符号定点数方法。如表 2 所示，对输入量、中间量和输出量既可以使用角度值，又可以使用角度对应的量化值。量化值越大，流水线结构的级数越多，数据精度越高。

表 2　输出值和理论值比较

x_in	y_in	z_n 值	输出 z_n	弧度值 z_n	绝对误差	误差对应角度/(°)
400	6ED	1.047 2	2A2	1.034 4	0.012 8	0.73
C00	6ED	2.094 4	55D	2.107 2	0.012 8	0.73
C00	913	−2.094 4	AA3	−2.107 2	0.012 8	0.73
400	913	−1.047 2	D5E	−1.034 4	0.012 8	0.78
6ED	400	0.523 6	15E	0.537 2	0.013 6	0.77
913	400	2.617 8	6A1	2.604 4	0.013 4	0.78
913	C00	−2.617 8	95F	−2.604 4	0.013 6	0.73

利用 CORDIC 算法实现运动平台逆向运动学求解的工作流程如图 4 所示，说明 CORDIC 算法的三角函数和反三角函数在逆向运动学求解中的应用。

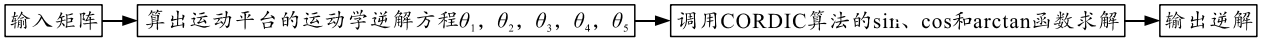

图 4　运动平台逆向运动学求解的工作流程

统一 CORDIC 算法的工作流程如图 5 所示。

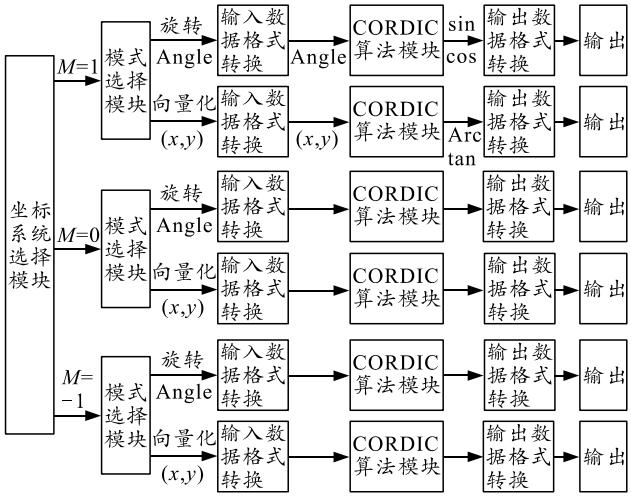

图 5 统一 CORDIC 算法的工作流程

3 ModelSim 仿真

为了验证 FPGA 实现 CORDIC 算法的正确性和有效性，笔者给出采用 CORDIC 算法实现正弦函数 sin、余弦函数 cos 和反正切函数的 ModelSim10.0d 仿真结果如图 6 和如图 7 所示。

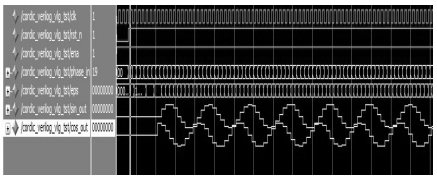

图 6 正弦函数和余弦函数 CORDIC 算法的 FPGA 实现仿真

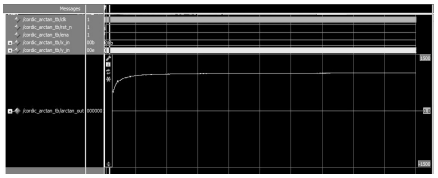

图 7 反正切函数 CORDIC 算法的 FPGA 实现仿真

图 6、图 7 中均使用模拟显示，更加方便观看。经过仿真数据验证了理论的正确性，并应用于运动平台的运动学逆解项目中，得到了良好的效果。

4 结论

笔者采用 FPGA 控制器作为硬件平台，构建 SOPC[8]系统，提出一种采用 SOPC 系统结构来完成运动学逆解的设计方案。测试和仿真结果验证了基于 FPGA 实现运动平台逆向运动学理论及其求解算法的正确性和可靠性。该方法具有计算速度快、效率高和便于实时控制等优点，可为逆解最优化和轨迹规划等算法模块后续工作提供铺垫。

参考文献：

[1] 韩建海. 工业机器人[M]. 武汉: 华中科技大学出版社, 2009: 61−63.

[2] 王朋月, 杨俊强, 毛征, 等. 动态背景下基于运动矢量补偿的目标检测[J]. 兵工自动化, 2019, 38(4): 6−10.

[3] OHLI D, OSVATIC M. Inverse kinematics of general 6R and 5RP serial manipulators[J]. Journal of Mechanieal Design, Transactions of the ASME, 1993, 4(6):922−931.

[4] 黄志雄, 何清华, 邹湘伏, 等. 一种求取运动学逆解的新算法[J]. 中国工程机械学报, 2009, 2(1): 7.

[5] OLDER J E. The CORDIC Trigonometric Computing Technique[J]. IRE Trans on Electronic Computers, 1959, 8(3): 330−334.

[6] 李兴格, 李刚, 熊思宇, 等. 一种新型反双曲正弦函数跟踪微分器设计[J]. 兵器装备工程学报, 2018, 39(12): 133−136.

[7] 胡欣, 魏龙, 陈怡君. 基于 FPGA 的雷达后端电路设计与实现[J]. 兵工自动化, 2019, 38(9): 39−43.

[8] 李兰英. Nios Ⅱ嵌入式软核 SOPC 设计原理及应用[M]. 北京: 北京航空航天大学出版社, 2009: 1−8.