

面向 FPGA 的惯性导航算法的研究与实现

刘强军, 张 岩

(哈尔滨工业大学 深圳研究生院, 广东 深圳 518055)

摘 要: 针对当前飞控系统中传感器多、运算量庞大, 传统微处理器已难以满足其日益增长的性能需求, 本文提出了一种面向 FPGA 的并行化惯性导航算法. 本算法采用扩展卡尔曼滤波对结果做最优估计, 四阶龙格库塔算法进行积分估计, 同时将惯性导航算法进行模块化分割, 采用流水线并行处理, 提高解算速度. 利用 FPGA 开发板驱动传感器, 并在该 FPGA 上实现对数据的解算, 解算结果由接口传输给微处理器, 降低了对微处理器接口和性能的要求. 最后仿真结果表明, 本算法在速度和精度上均有良好的表现, 验证了该算法在 FPGA 上实现的可行性.

关键词: 惯性导航算法; FPGA 开发板; 扩展卡尔曼滤波; 龙格库塔算法

中图分类号: TN47

文献标识码: A

文章编号: 1000-7180(2018)07-0058-04

Research and Implementation of Inertial Navigation Algorithm for FPGA

LIU Qiang-jun, ZHANG Yan

(Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China)

Abstract: Aiming at the problem that the sensor in the current flight control system is too many and the computational complexity is large, the traditional microprocessor has been difficult to meet its growing performance requirements. This paper presents a parallel adaptive inertial navigation algorithm for FPGA. In this algorithm, the extended Kalman filter is used to estimate the result. The fourth-order Runge-Kutta algorithm is used to estimate the integral. At the same time, the inertial navigation algorithm is modularized and the parallel processing is used to improve the solution speed. The FPGA development board is used to drive the sensor, and the data is solved on the FPGA. The result of the solution is transmitted from the interface to the microprocessor to reduce the interface and performance requirements of the microprocessor. Finally, the simulation results show that the proposed algorithm has good performance in terms of speed and accuracy, and verifies the feasibility of the algorithm on FPGA.

Key words: INS; FPGA; Extended Kalman Filter(EKF); Runge-Kutta algorithm

1 引言

21 世纪以来, 无人机高速发展, 其完成的功能也由以前单一的飞行变成航拍、巡航、遥测、灾害预警、投递物品等多种功能^[1-2], 而为了实现以上功能, 要求无人机携带的传感器也越来越多, 数据运算量也越来越大, 这大大的提高了对主控芯片性能的要求. 目前主流方案是采用基于 ARM 平台的 MCU

芯片^[3-4], 所有计算和控制过程全部由该芯片完成, 而 MCU 只能串行计算而且数据处理能力薄弱, 这极大地限制了飞控的发展.

本文将惯性导航系统在 FPGA 开发板上独立实现, 极大的减少主控芯片的计算压力. 同时现在大部分飞控中没有滤波算法或者是相对简单的互补滤波算法^[5], 抗噪声能力较低, 本文引入扩展卡尔曼滤波^[6]对姿态等信息做最优估计, 能极大的提高抗噪

声能力.最后充分利用 FPGA 的并行处理能力,引入流水线将算法作并行化设计,极大的提高了算法的执行速度.

2 惯性导航的算法背景

2.1 惯性导航模块的设计背景

飞控系统是无人机的核心,其性能直接决定无人机的性能,当前主流的飞控系统结构如图 1(a)所示.所有的传感器数据和控制处理都是微处理器执行,对微处理器的性能要求非常的高,且实时性得不到有效的保证.针对此,本文将运算量最高的惯性导航模块用 FPGA 实现,其在本文飞控结构中如图 1(b)所示.惯性导航模块主要负责从传感器的数据中解算出飞机的姿态信息,由于 FPGA 的并行处理能力强,能够更实时的为微处理器提供当前的姿态信息,微处理器只需处理控制信号,提高了飞控的反应速度.由于热噪声、测量噪声等的干扰,本文设计的惯性导航模块嵌入了一个 EKF 模型,实现对状态的最优估计.

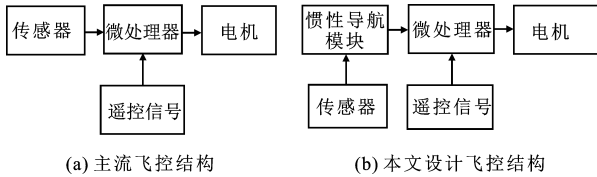


图 1 飞控结构对比

2.2 姿态解算的基本原理

为了描述飞机的运动和姿态,本文建立了一个六自由度的运动学模型,即惯性坐标系下的三维姿态角以及三维速度,本模型理论上可以用于任何一个运动设备.

由于传感器采集的数据均为机体坐标系数据,故需要将传感器数据转换到地理坐标系,因而本算法中一个重要的矩阵是旋转矩阵(式 1),用于将机体坐标系下的数据转换到地理(惯性)坐标系下.

$$\mathbf{R}_{be} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (1)$$

在矩阵 $\mathbf{R}_{be}$ 中,每一个元素都可以根据四元数计算得来,具体计算公式参考文献[7].

另一个矩阵用于更新四元数,此矩阵经常用于捷联式惯性导航系统.即 $\dot{q} = \mathbf{\Omega}(q) * \omega/2$, 其中 $\mathbf{\Omega}(q)$ 由四元数组成,具体定义如下:

$$\mathbf{\Omega}(q) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \quad (2)$$

本文设计了一个飞控状态矩阵 $\mathbf{X}$ 用于存储飞控的状态, $\mathbf{X} = \{P, V, q, b_w\}$. P 为三维位置, V 为三维速度, q 为四元数, b_w 为三维陀螺仪偏置.本系统的输入 u 为真实的传感器数据.由于在测量过程中存在噪声、传感器偏移以及重力等因素的影响,故会对输入数据进行一定的预处理.具体处理公式如式(3)所示:

$$u = \begin{bmatrix} \omega_m \\ a_m \end{bmatrix} = \begin{bmatrix} \omega - \omega_w + b_w \\ a - \omega_a - \mathbf{R}_{be}(0, 0, g)^T \end{bmatrix} \quad (3)$$

由于测得的都是角速度和加速度值,要获取确切的速度和角度需要对其进行积分,故在此基础上,增加了一个状态矩阵的导数矩阵 $\mathbf{X}$.

真实角速度和加速度计算方法如下所示:

$$a = a_m + \omega_a + \mathbf{R}_{be}(0, 0, g)^T \quad (4)$$

$$\omega = \omega_m + \omega_w - b_w \quad (5)$$

将上述公式带入可得:

$$\dot{\mathbf{X}} = f = \begin{bmatrix} V \\ \mathbf{R}_{be} * (a_m + \omega_a) + \mathbf{R}_{be}(0, 0, g)^T \\ 0.5 * \mathbf{\Omega} * (\omega_m + \omega_w - b_w) \\ W_b \end{bmatrix} \quad (6)$$

本动态模型的输出是传感器采集到的数据,输出给矫正更新模块.测量状态矩阵如下:

$$y = h(x) = \begin{bmatrix} P \\ V \\ B_b \\ A_b \end{bmatrix} = \begin{bmatrix} P \\ V \\ \mathbf{R}_{be} B_e \\ -P_z \end{bmatrix} \quad (7)$$

在式(7)中,由于磁力计测试的是地理坐标系的,为了变换到体坐标系中,将地理坐标系测量得到的值乘上旋转矩阵,转换到体坐标系中,而后将测量的值输入后续模块进行矫正.

2.3 EKF 算法的在惯性导航算法的应用

飞控系统是一个非线性系统,扩展卡尔曼滤波可以将非线性化模块线性化,故采用扩展卡尔曼滤波.线性化采用雅克比行列式法,将输入矩阵 $\mathbf{G}$ 、系统矩阵 $\mathbf{F}$ 以及测量矩阵 $\mathbf{H}$ 分别线性化之后就变成一个线性系统,然后进行卡尔曼滤波处理.其具体公式如下所示:

$$x_k = x_{k-1} + \int_{t_{k-1}}^{t_k} f(x_{k-1}, u_{k-1}) dt \quad (8)$$

$$P_k = (I + FT)P_{k-1}(I + FT)^T + T^2GQG^T \tag{9}$$

$$K = P_k H^T (HP_k H^T + R)^{-1} \tag{10}$$

$$x_k = x_k + K(Hx_k - y_k) \tag{11}$$

$$P_k = P_k - KH P_k \tag{12}$$

上式中, F, G, H 分别为 $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial W}, \frac{\partial h}{\partial x}$ 的雅克比行列式.

2.4 姿态融合算法的 MATLAB 模型

基于对算法的分析, 本文设计的 MATLAB 模型的计算流程如下图 2 所示.

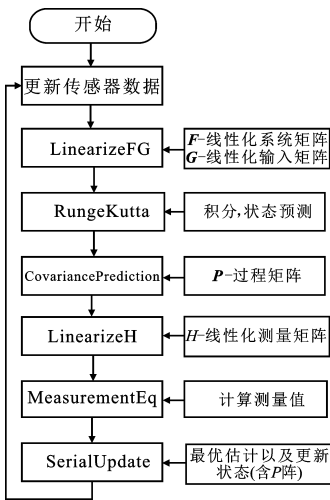


图 2 算法的 MATLAB 计算流程图

由于 MATLAB 自身的局限性, 只能够串行执行.

3 基于 FPGA 的并行化算法实现

3.1 算法的并行化结构设计

根据数据的依赖关系, 按照硬件的逻辑, 将算法作了并行化设计, 本文设计的完整算法框架如下图 3 所示.

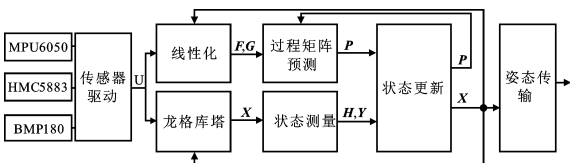


图 3 完整算法框架图

如图 3, 根据系统功能划分为传感器驱动、姿态解算、姿态传输三个主要模块. 其中姿态解算包含图中除传感器驱动和姿态传输的所有模块, 姿态传输采用 UART 接口与微处理器通讯.

3.2 总体流水线设计

在实现的过程中, 根据模块间的数据关联, 设计

了一条三级流水线, 即传感器驱动、姿态解算、姿态传输三级流水. 由于每级流水所需时间不等, 为保证流水之间数据传输的准确性, 每级流水间设计了数据同步电路, 如图 4 所示.

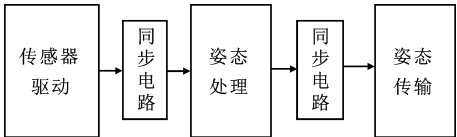


图 4 流水线结构设计

其中同步电路主要采用三个 D 触发器加一个与门构成, 如图 5 所示.

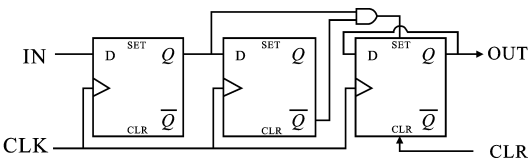


图 5 信号同步电路

其中前两级触发器为上升沿检测信号, 输入为上级的输出数据有效信号, 检测到上升沿后锁存需要传递的数据, 同时置位第三级触发器, 此时该“1”电平会被锁存, 表示此时同步器内的数据有效. 当下级流水处理完毕当前数据后, 会检测该信号是否使能, 如已经使能则会立即取出锁存的数据进行计算并清零该信号, 否则会持续等待该信号使能. 模块间通过这种方式进行数据交流, 不但可以保证数据的准确性, 同时也可以提高速度.

3.3 传感器驱动模块设计

本模块主要完成对陀螺仪、气压计以及磁力计的驱动和数据转换功能. 为简化硬件结构, 选用全部 IIC 协议的传感器 (MPU6050、HMC5883、BMP180), 挂载在同一条 IIC 总线上, 分时采集以上三个模块的数据.

由于传感器从接收到指令到输出数据需要一定的准备时间, 故采用先驱动准备时间长的气压计, 先读取准备时间短的磁力计, 减少时间消耗. 在读取完某个传感器数据后立即开始数据的转换, 同时进入下一个传感器的读取状态, 数据转换与读取并行执行.

3.4 姿态处理模块的并行化设计

姿态处理模块包含图 3 中的线性化、龙格库塔、过程矩阵预测、状态测量、状态更新一共 5 个模块, 实现了一个采用扩展卡尔曼滤波器的姿态解算模块. 而在扩展卡尔曼滤波中状态的预测需要计算公

式(8)的积分操作,而电路中无法实现数字积分操作,通过调研和分析,采用四阶龙格库塔法^[8]来实现积分估计。

由于 F 、 G 矩阵以及状态的估计均仅与当前状态有关,而状态估计采用四阶龙格库塔算法需要更长的时间,同时考虑到其结果的后续关联性,将 F 、 G 矩阵的线性化与龙格库塔算法并行设计。

过程矩阵预测的输入为 F 、 G 矩阵,而状态测量的输入为当前测量的传感器数据,彼此无关联,故也做并行化设计。最后同时输出给状态更新模块,由于此二者数据完成的时间不一致,故在姿态更新模块之间采用如图 5 所示的同步信号电路,状态更新模块需要同时检测到 2 个使能才会执行。具体并行化结构如图 3 所示。

3.5 其它说明

姿态传输模块主要包含一个 UART 接口模块和控制模块,按照要求顺序将姿态结果传输给微处理器。

在设计的过程中,内部数据全部采用单精度浮点数,通过状态机不断复用 IP 核,以减少资源的占用。同时在算法的实现过程中,将大部分重复计算的中间变量存储,当再次需要的时候直接读取该寄存器的值,提高运算效率。

4 电路仿真和测试结果

为了验证精度,首先以 MATLAB 为参考模型,先不加入传感器,通过指定相同的输入,比较 Modelsim 与 MATLAB 的仿真结果。为便于比较,将数据列入表 1 中。通过结果比较发现, X 、 Y 方向的数据完全一致,而 Z 方向由于重力加速度(注:在 Verilog 模块中使用是单精度浮点数,初始化已经存在误差)的存在导致误差,但在飞控中,此误差已经可以忽略不计。

表 1 Modelsim 与 Matlab 仿真结果对比

向量	Matlab	Modelsim	相对误差
Posx	7.488 75e-6	7.488 75e-6	0
Posy	-7.483 20e-6	-7.483 20e-6	0
Posz	-8.982 62e-9	-8.982 71e-9	1.001 9e-5
Velx	1.496 62e-3	1.496 62e-3	0
Vely	-1.494 96e-3	-1.494 96e-3	0
Velz	-0.900 12e-4	-0.900 11e-4	1.1109 6e-5

随后为验证传感器驱动模块和 UART 模块,设计了传感器的模型,由于传感器本身采用较慢的

IIC 接口、且传感器内部数据转换速度慢,故每一帧传感器数据接收用时达到了 1.95 ms,而姿态解算模块在接收到数据之后仅用了 0.1 ms 就计算出了当前的姿态信息。解算数据更新频率与其他文章对比如下表 2 所示。

表 2 数据更新频率对比

	平台	更新频率
本文	Virtex5	510 Hz
文献[9]	STM32F401RE	200 Hz
文献[10]	Virtex6	100 Hz

从表 2 中可知,数据更新频率达到 510 Hz,大于文献[9]中采用 ARM 核的 200 Hz。而与同样采用 FPGA 方案的文献[10]相比,速度也是其的 5 倍,可见在插入流水线和并行化处理之后,速度有很明显的提升。

5 结束语

本文提出并实现了一种面向 FPGA 的并行化惯性导航算法,在实现的过程中将该算法模块化,并加入流水线使其并行化处理,仿真结果表明,本算法的解算结果与 MATLAB 仿真结果一致,数据更新频率达到了 510 Hz,远远超过了导航系统需要的速度要求,且其数据速度仅受传感器自身数据采集速度的限制,解决了姿态运算速度的难题。其次本模块自主驱动传感器单元,不但优化了飞控系统的结构,也降低了对主控芯片接口的占用。最后算法中设计了扩展卡尔曼滤波器,使其抗干扰能力更高,能够满足更高精度的导航。

参考文献:

[1] Gerd Grau, Elisha J F, Vivek Subramanian. Printed unmanned aerial vehicles using paper-based electroactive polymer actuators and organic ion gel transistors [EB/OL]. [2017-10-15]. <http://www.nature.com/articles/micronano.201632?proof=twoe#f1>.

[2] 刘倩,梁志海,范慧芳. 浅谈无人机遥感的发展以行业运用[J]. 测绘与空间地理信息, 2016, 39(6): 167-169.

[3] Dryanovski Ivan, Valenti R G, Xiao J. An open-source navigation system for micro aerial vehicles[J]. Autonomous Robots, 2013, 34(3): 177-188.

[4] 高礼伟,朱小刚,陶波. 基于 STM32 的小型四旋翼飞控系统的设计[J]. 科技风, 2017, 29(6): 21-22.

图像的边缘检测所需的时间如表 1 所示:

表 1 NEON 处理和 CPU 处理性能对比

图像处理方法	CPU	NEON
时间/ms	1403	121

由表 1 可以看出使用 NEON 引擎计算速度,是 CPU 软件计算的 11.6 倍,速度有明显提升.对分辨率为 800×600 ,帧率为 60 fps 的输入图像处理速度达到了 8.3 fps,而使用 CPU 处理的速度仅为 0.7 fps.

本系统仅由 CPU 软件实现的算法程序是使用 C 语言编写的,NEON 方法的算法程序是使 NEON C 内联函数实现的.两者均为软件处理图像的方法.由于 NEON 协处理器由于拥有独立的寄存器系统和硬件执行单元,可以实现并行运算,大大提高了图像处理速度.本设计中 Sobel 算法并不复杂,程序代码不多,所以 NEON 的处理速度大约是传统软件方式处理的 11.6 倍,可以预见两者的差距会随着数据量的增加和算法复杂度的升级而加大.

5 结束语

本文针对嵌入式软件方式的视频处理研究,提出一种基于 NEON 优化技术的视频处理系统,尽量采用基于 FPGA 的片上资源,充分利用了 Avalon 总线特有的流传输功能,有效的缩短了数据传输的时间;采用 NEON 技术对算法进行优化,与仅用

CPU 软件实现算法的性能对比,从结果上可以看出 NEON 技术优化后的算法,不仅计算时间明显缩短,处理后视频流畅度较好.同时,本文所设计的视频处理系统很好地降低嵌入式视频处理系统的开发调试难度,节约了成本,为打算使用 NEON 技术进行算法加速的研究人员和应用开发者提供参考和借鉴.

参考文献:

[1] 陈泽新. 基于 NEON 引擎的 JM 解码器并行优化实现[D]. 成都:西南交通大学,2010.

[2] 简欢. 基于 ARM NEON 的 H. 265 解码器优化[J]. 电视技术,2014,38(15):102-105.

[3] 陈泽新,杨斌. ARM Cortex-A9 的 NEON 技术研究及应用[J]. 器件技术,2013(10):46-49.

[4] 杨新华,寇为刚. 基于 FPGA 的 Sobel 算子图像边缘检测技术[J]. 表技术与传感器,2013(1):102-104.

作者简介:

陈 炎 男,(1991-),硕士研究生.研究方向为 SOC 设计. E-mail: chenyan790779514@163.com.

袁国顺 男,(1966-),硕士,研究员,博士生导师.研究方向为数模混合集成电路设计.

刘小强 男,(1990-),博士研究生.研究方向为图像处理和 SOC 设计.

(上接第 61 页)

[5] 万晓凤,康利平,余运俊,等. 互补滤波算法在四旋翼飞行器姿态解算中的应用[J]. 测控技术,2015,34(2):8-11.

[6] 景丽. 基于卡尔曼滤波组合导航算法的计算量与精度分析[D]. 哈尔滨:哈尔滨工业大学,2014.

[7] 秦永元. 惯性导航[M]. 北京:科学出版社,2006:287-304.

[8] 王励扬,翟昆朋,何文涛,等. 四阶龙格库塔算法在捷联惯性导航中的应用[J]. 计算机仿真,2014,31(11):56-59.

[9] 徐泽远,伊国兴,魏振楠. MEMS-SINS/GPS 组合导航

系统设计[J]. 自动化仪表,2016,37(12):67-70.

[10] Bhogadi L R, Gottapu S B R, Konala V R. MicroBlaze implementation of GPS/INS integrated system on Virtex-6 FPGA[J]. Springerplus, 2015, 4(1):629.

作者简介:

刘强军 男,(1993-),硕士研究生.研究方向为数字 IC 设计.

张 岩(通讯作者) 男,(1969-),教授.研究方向为专用处理器设计. E-mail: ianzh@foxmail.com.