

## 2 飞行控制实验平台设计

本章介绍飞行控制实验平台的设计。由于固定翼飞机载重大、飞行时间长、速度快、灵活性好，非常适用于做导航算法的实验与研究。根据导航算法对计算性能的要求设计了计算能力强，并且有多传感器的飞行控制控硬件电路。为了保证算法的实时性并且能够更好地处理多个复杂任务，选用了 QNX 实时操作系统。

### 2.1 固定翼飞行器

固定翼飞行器是靠螺旋桨或者涡轮发动机产生的推力作为飞机向前飞行的动力，主要的升力来自机翼与空气的相对运动。固定翼飞行器具有很多优点，它的巡航面积大，飞行高度高，飞行速度快，也可以做大机动动作，因此能采集到更全面的目标对象的数据。同时，固定翼飞行器的载重大，可将实验飞控电路板放入仓内进行挂飞实验。它的续航时间长，方便采集到足够量的飞行实验数据。

本文选用的是如图 2-1 所示的固定翼无人机，型号是天捷力 X-UAV，机体使用的是 EPO 泡沫，内部有碳杆做支撑，动力是电力驱动。



图 2-1 天捷力 X-UAV  
Figure 2-1 Tianjieli X-UAV

整机参数如表 2-1 所示：

表 2-1 整机参数  
Table 2-1 Parameter of the X-UAV

材料	EPO 泡沫	
翼面积	54.5	dm <sup>2</sup>
翼展	1718	mm
翼载荷	55~66	g/dm <sup>2</sup>
推力	3000	g
飞行重量	2500~3000	g
机身长度	1100	mm

动力系统的参数如表 2-2 所示:

表 2-2 动力系统的参数

Table 2-2 Parameter of the X-UAV dynamic system

电机	2820 KV1100	
电池	14.8V	Li-polymer
电调	70	A
桨叶尺寸	9	寸
舵机	9g (V 尾) 和 17g (机翼)	g
空机重量	1050	g
尾翼碳杆	6 (外径) ×4 (内径)	
副翼碳杆	10 (外径) ×8 (内径)	

## 2.2 飞控硬件电路设计

### 2.2.1 硬件电路设计需求

目前常见的低成本小型飞控电路板,使用的是 AVR 单片机或基于 ARM Cortex 内核的微控制器,它们的缺点是矩阵运算能力和浮点运算能力有限。在面对复杂的导航算法的时,它的实时性就会受到严重的影响。考虑到后续研究,在设计的飞控硬件电路上需要留有一些扩展口,方便进一步的算法的优化和功能的增加。总结一下,设计的飞控硬件板需要满足下面的几个要求:

(1) 微小型无人机的主要特点就是体积小、重量轻、灵活性大,相应的飞控硬件应该符合体积小,重量轻,功耗低的要求。

(2) 本文要求主控制器能够满足导航算法对运算能力的要求,所选择的控制器应该具有符合要求的运算能力和处理速度。最好可以运行实时操作系统,能够保证多任务下算法的实时性。

(3) 由于需要完成飞行控制的功能,并且做多传感器的数据融合,要求要有丰富的外设接口。

(4) 考虑到后续的研究工作,需要在硬件上留有足够的扩展接口。

(5) 在选择传感器和其他的模块时,需要考虑到性能和成本的问题。

### 2.2.2 主控制器和传感器选型

为了满足算法的运算要求,并且需要丰富的外设接口,主控制器选用的是 Xilinx Zynq-7000 All Programmable SoC XC7Z010。

Zynq-7000 系列使用 SoC 技术将 ARM 处理器和 FPGA 架构紧密结合,拥有两颗 ARM CortexTM-A9 核组成的处理核心部分(Processing System, PS),以及一颗 Xilinx 系列 FPGA

核心 Artix-7 所构成的可编程逻辑部分 (Programmable Logic, PL)。其 SoC 架构图如图 2-2 所示。该系列芯片中, ARM 拥有相对于市面上其他处理器更强大的计算速度, 所以完全满足复杂算的计算要求, FPGA 拥有完全可编程能力, 可以用来作为丰富的外设接口和用于预留接口。

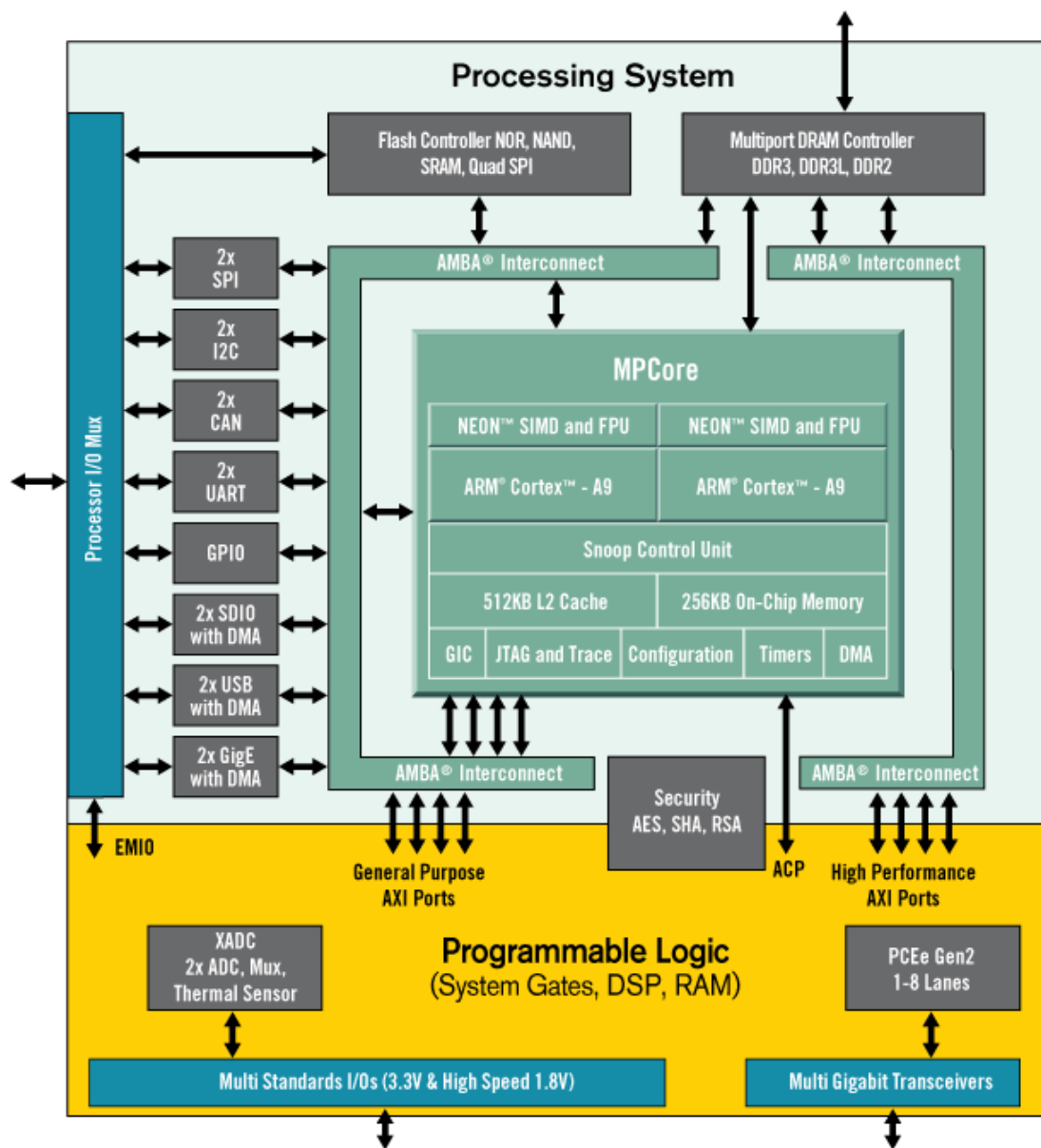


图 2-2 ZYNQ-7000 系列 SoC 架构

Figure 2-2 The SoC architecture of ZYNQ-7000

➤ 处理器系统单元 (PS):

- 处理器核心: Dual ARM Cortex<sup>TM</sup>-A9 MPCore<sup>TM</sup> with CoreSight<sup>TM</sup>
- 最高主频: 866MHz
- L1 Cache: 32KB 指令 Cache
- L2 Cache: 512KB
- 片内内存: 256KB

- 片外接口: DDR3, DDR3L, DDR2, LPDDR2
- 存储接口:  $2 \times$  Quad-SPI, NADA, NOR
- DMA 通道: 8 (其中四个 PL 专用)
- 外设:  $2 \times$  UART,  $2 \times$  CAN 2.0B,  $2 \times$  I2C,  $2 \times$  SPI,  $4 \times$  32b GPIO

➤ 可编程逻辑单元 (PL)

- Logic 核心: Xilinx 7 Series Artix-7 FPGA
- 可编程逻辑单元: 2.8 万逻辑单元 (约 43 万 ASIC 逻辑门)
- LUTs: 17600
- RAM: 240KB
- 触发器: 35200
- DSP slice: 80

本文的导航算法涉及的传感器有三轴陀螺仪 (L3GD20), 三轴加速度计 (BMA280), 三轴磁力计 (HMC5883L), 气压计 (MS5611), 空速计 (MPXV7002DP), 另外还有 GPS 接收模块 (u-blox MAX-M8)。选择这些器件需要满足一定的精度要求, 还要考虑到它的价格。所以, 综合考虑选择低成本的 MEMS 传感器。

相关传感器的主要指标如表 2-3 所示:

表 2-3 相关传感器的主要技术指标  
Table 2-3 The main indicator of sensors

三轴陀螺仪 (L3GD20)	量程: $\pm 250/500/2000$ dps, 灵敏度: $8.75/17.50/70$ mdps 零位偏移: $\pm 10/15/75$ dps, 噪声: $0.03\text{dps}/\sqrt{\text{Hz}}$
三轴加速度计 (BMA280)	量程: $\pm 2/4/8/16$ g, 灵敏度: $4096/2048/1024/512$ LSB/g 零位偏移: $\pm 50$ mg, 噪声: $120\mu\text{g}/\sqrt{\text{Hz}}$
三轴磁力计 (HMC5883L)	量程: $\pm 8$ gauss, 灵敏度: $\pm 0.2\%$ FS/gauss
气压计 (MS5611)	气压量程: $10\sim 1200$ mbar, 灵敏度: $\pm 1.5$ mbar 温度量程: $-40\sim +85$ °C, 灵敏度: $\pm 0.8$ °C
空速计 (MPXV7002DP)	压力量程: $\pm 2.0$ kPa, 灵敏度: $1.0\text{V/kPa}$
GPS (u-blox MAX-M8)	更新率: 5Hz, 水平精度: 2.5m 速度精度: $0.1\text{m/s}$ , 航向精度: $0.5^\circ$

### 2.2.3 硬件电路结构和原理图设计

硬件电路主要包括 Xilinx 的 ZYNQ-7000 系列的 XC7Z010, 256MB 的 DDR3, 16MB 的 SPIFLASH, 10/100/1000Mb/s 以太网, 8GB Micro SD 卡, 三轴陀螺仪 (L3GD20), 三轴加速度计 (BMA280), 三轴磁力计 (HMC5883L), 气压计 (MS5611), GPS 模块 (u-

blox MAX-M8), 无线数传模块 HM-TRP, 华为的 MG301 GSM LCC 模块。

所有的传感器都是通过 IIC 总线与 ZYNQ 相连, GPS 模块是用串口与 ZYNQ 相连。GPS 模块是使用的 u-blox 的 MAX-M8, 并且留有外接天线接口, SMA 接口。无线数传收发模块 HM-TRP 也是通过串口与 ZYNQ 相连, 也留有 SAM 天线接口。

图 2-3 是系统结构框图, 其中 PS 端接的是 DDR3、千兆网、SD、CAN、QSPI Flash、UART, PL 端接的是无线数传、GPRS、PWM 输出、S.BUS 输出、所有传感器、扩展口。

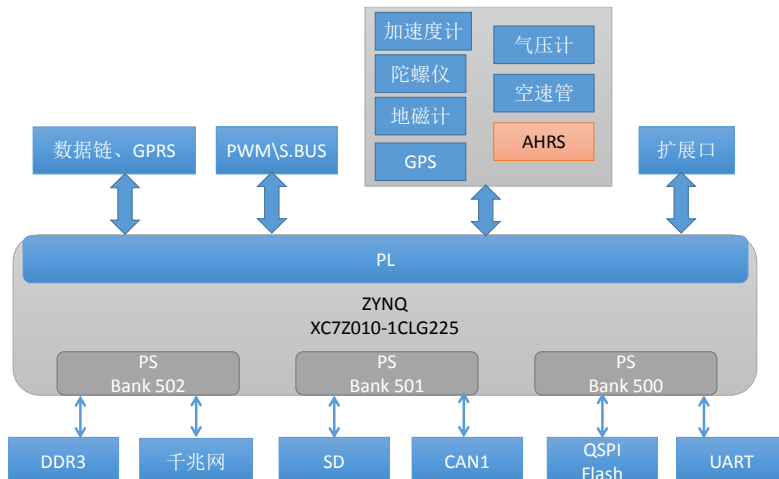


图 2-3 硬件电路结构框图

Figure 2-3 The structured flowchart of circuit

原理图见本文附录页。

## 2.2.4 硬件电路实物

如图 2-4 所示, 板子的尺寸为 120×55×48mm, 功耗小于 3W, 总重是 200g, 适用温度范围是-40℃到 80℃之间。



图 2-4 电路板实物图

Figure 2-4 The physical map of circuit board

### 2.2.5 功耗测试

由于微小型无人机受载重的限制，无人机的电池容量通常不会太大，这就需要飞控硬件必须具有低功耗的特点。可以根据每个器件的数据手册，可以计算出理论的功耗，如表 2-4 所示，计算出的理论最大功耗大约为 4000mW。而在实际使用的过程中，除非是在极度恶劣情况下才会满载运行，才有达到这个理论值的可能性。在实际的使用中，正常 7V 供电后，电流达到 0.22A，功耗不超过 2W，不到理论值的一半。

表 2-4 硬件的理论最大系统功耗

Table 2-4 Theoretical maximum power dissipation of the circuit board

器件名称	功耗 (mW)
ZYNQ XC7Z010	2000 (Max)
u-blox MAX-M8	240 (Max)
L3GD20	20
BMA280	0.42
MS5611	5
HMC5883	1
DDR3	480
HM-TRP	100 (Max)
GPRS	1330 (Max)
总计	4000 (Max)

## 2.3 飞控软件设计

### 2.3.1 QNX 实时操作系统

飞行控制器本身应该具有良好的实时性，需要实时采集传感器数据和实时进行飞行控制等，也需要实时计算出飞行器的导航信息。由于传统的单任务顺序结构资源的利用率太低，且执行的实时性太差，大量的 CPU 时间被浪费在等待中。为了保证实时性，本文使用了微内核操作系统 QNX，主要的优势如下：

(1) QNX 是一个实时操作系统，能够提供一个用户可控、优先级驱动、急者优先级抢占的调度方式。并且，QNX 实时操作系统自身开销小、上下文切换快，能够在同样的硬件条件下为实时应用留下更多的余地。

(2) QNX 是一个可嵌入的操作系统，由微内核和一组共操作的进程构成，具有高度可伸缩性，可灵活的裁剪，如图 2-5 所示。最小的配置只占用几十个 KB 内存。

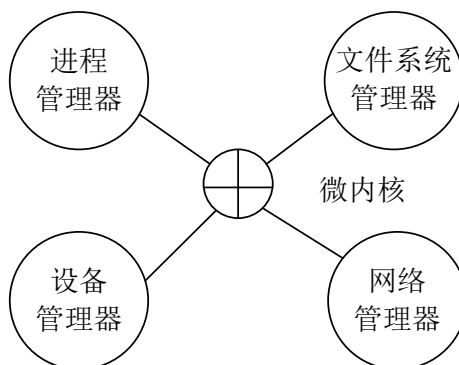


图 2-5 QNX 微内核协调系统管理器之间的工作

Figure 2-5 QNX microkernel coordination between system manager work

(3) 开发环境是使用的 QNX Momentics 套件。QNX Momentics 包含了开发 QNX Neutrino 实时操作系统应用程序时所需要的所有工具：高度集成的开发环境（IDE）、编译器、连接器、库函数以及其他的 QNX Neutrino 组件。QNX Momentics 如图 2-6 所示。

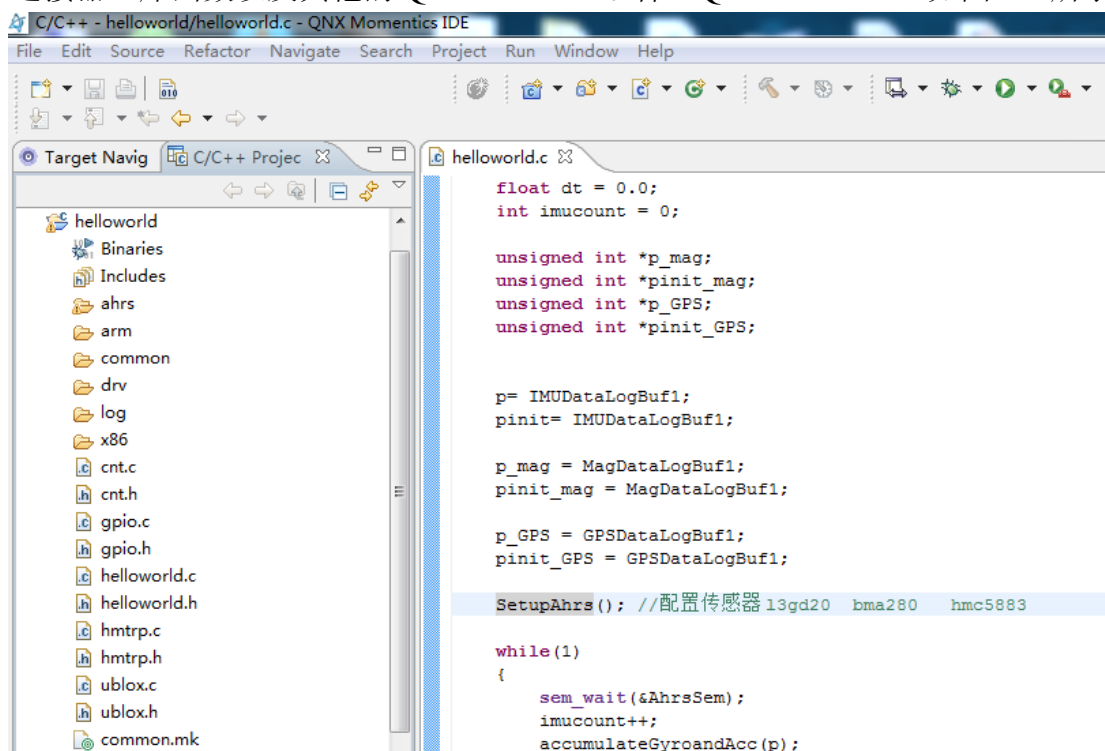


图 2-6 开发环境 QNX Momentics IDE

Figure 2-6 The development environment of QNX

### 2.3.2 软件结构

整个代码从 `int main(int argc, char *argv[])` 函数开始进入运行。在主函数中，主要任务是创建五个线程，包括了 u-blox 串口接收 GPS 数据，无线数传收发数据，定时器定时功能，SD 卡数据写入，导航算法。具体如图 2-7 所示：



```
pthread_create(NULL, NULL, ubx_int_thread, NULL);
pthread_create(NULL, NULL, hmtrp_int_thread, NULL);
pthread_create(NULL, NULL, timer_int_thread, NULL);
pthread_create(NULL, NULL, Data_log_thread, NULL);
pthread_create(NULL, NULL, ahrs_thread, NULL);
```

图 2-7 创建线程

Figure 2-7 Creating of thread

由于本文设定每 2ms 采集一次 IMU（陀螺仪和加速度计）数据。采用定时器去定时间，用信号量来标志。在 main() 中初始化信号量 sem\_init(&AhrsSem, 0, 1)，然后在定时器中每过 2ms 增加一次信号量的值 sem\_post(&AhrsSem)，在导航算法中等待 2ms 的到来 sem\_wait(&AhrsSem)。这样就完成了传感器的定时采样。在导航算法这个线程中，将采集到的传感器数据存到 buffer 中，每 20 秒将采集到的数据做均值滤波，将滤波后的值作为导航算法的数据来源。

使用乒乓操作将数据写入 SD 卡。由于 SD 卡写入数据可以批量快速写入，不需要一个一个数据写入，所以采用乒乓操作。先定义两个内存空间 LogBuffer1 和 LogBuffer2，并且初始化信号量 sem\_init(&LogAhrsSem, 0, 0)，首先将数据放入 LogBuffer1 中，当判断 LogBuffer1 中的数据已经写满就将信号量加 1，sem\_post(&LogAhrsSem)。SD 卡写的线程中，等待到信号量就开始将 LogBuffer 中的数据写入 SD 卡，sem\_wait(&LogAhrsSem)。同时在导航算法线程中，将解算的数据和传感器数据继续写入 LogBuffer2 中，当 LogBuffer2 写满后就将其写入 SD 卡，同时将后面的数据存入 LogBuffer1 中。按这种乒乓操作来进行 SD 卡的写入，记录飞行数据。

在 GPS 接收的线程中，用串口中断去读取 GPS 模块收到的数据包并将解析后的数据放入到内存中，GPS 接收的频率为 50Hz，所以每次收到数据后写一个标志，表示已经收到 GPS 数据，u-blox MAX-M8 中的数据已经是通过了滤波处理，所以直接可以传输到导航算法中使用。

在无线数传线程中，使用串口对数据进行数据的发送和接收。由于需要相对精确的时间对数据进行采样，所以启用定时器线程执行该过程。导航算法线程中主要负责读出各个传感器的数据并且做算法。图 2-8 是整个软件系统结构图。

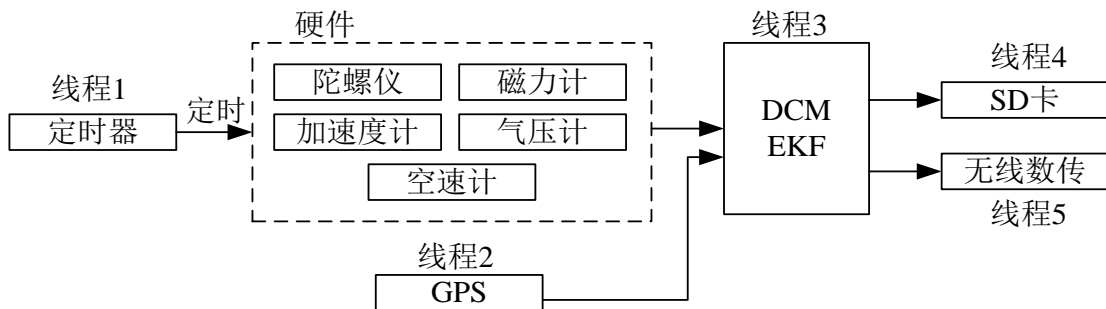


图 2-8 软件系统结构图

Figure 2-8 Software system architecture



### 2.3.3 运算性能测试

软件平台搭建完成并加入导航算法后，对其运算性能进行了简单的测试。不同算法的代码运行的时间列表如下：

表 2-5 性能测试

Table 2-5 Performance testing

任务名称	实际执行时间	执行时限	执行频率
DCM_update()	0.02ms	20ms	50Hz
EKF_update()	0.83ms	20ms	50Hz
AhrsUpdate()	0.90ms	20ms	50Hz

表 2-5 中,DCM\_update()表示的是互补滤波算法在飞控板上运行的时间,EKF\_update()表示的是扩展卡尔曼滤波数据融合算法在飞控板上的运行时间, AhrsUpdate()表示的是整个导航算法的运行时间。执行时限指的是在 20ms 内必须完成本次滤波算法,执行频率为 50Hz 指的是每 20 毫秒执行一次滤波算法。从表中数据可以明显看出,该算法在飞控板上能够很好的运行,可见该平台的运算潜能是比较大的。

## 2.4 本章小结

本章介绍了本文所设计的整个飞行实验平台。首先介绍了所使用固定翼飞机。然后设计了基于 ZYNQ-7000 和 MEMS 传感器的低成本、高性能的飞行控制板,可以满足算法的运算需求。选择 QNX 实时操作系统能够更有效和合理使用飞行控制板的资源,提高代码的运行效率。



### 3 传感器测量数据预处理

在本文所研究的导航算法中，需要使用到三轴陀螺仪、三轴加速度计、三轴磁力计、气压计和空速计。这些传感器测量值的物理意义如下表所示：

表 3-1 传感器和对应的测量值

Table 3-1 The value that sensors measured

传感器	测量值	单位
三轴陀螺仪	角速度	rad/s
三轴加速度计	加速度	m/s <sup>2</sup>
三轴磁力计	地磁磁通量	gauss
气压计	海拔高度	m
空速计	空速	m/s

这里使用的 MEMS 传感器存在误差，其中包括系统误差和随机误差。在捷联惯性导航中，姿态、速度、位置的求解都是基于积分运算实现的，以上误差会使导航计算结果随时间迅速发散。所以必须对这些传感器的误差进行补偿，保证测量值的准确性。系统误差主要包括刻度因数误差、零位偏置误差、非正交性误差等，这些误差可以通过各种误差补偿方法进行误差的消除。随机误差也认为是噪声，指的是传感器误差中随时间进行无规则变化的部分，由于它的随机特性决定了其不可能被完全消除，但可以使用滤波器将部分噪声去掉。

本章将主要讲述几个主要的 MEMS 传感器的数据预处理算法。其中包括三轴陀螺仪，三轴加速度计，三轴磁力计，气压计和空速计，对它们进行相应的误差补偿和噪声滤波。

#### 3.1 MEMS 陀螺仪

##### 3.1.1 MEMS 陀螺仪介绍

MEMS 陀螺仪固定在飞行器上，测量的是飞行器在机体坐标系下每个轴上的角速度，单位是 rad/s。

MEMS 陀螺仪的工作原理是利用科里奥利效应测量运动物体的角速率。如图 3-1，根据科里奥利效应，当一个物体在参考系内线性振动，沿参考系与线性运动的轴相互垂直的轴转动，就会产生一个既垂直于振动轴，又垂直于转动轴的哥氏力。然后从一个电容感应结构就可以检测到科里奥利效应最终产生的位移。

目前市面上的 MEMS 陀螺仪多数采用音叉结构，它是由两个震动并不断做反向运动的物体组成，当施加角速率时，两个物体上的科里奥利效应产生相反的力，从而引起电容的变化。电容的差值与角速率成正比，因此通过测量电容值，就可以得到角速率。