

捷联惯导系统的软件开发

赵 勇

(北京航空航天大学自动控制系, 北京, 100083)

THE SOFTWARE DEVELOPMENT OF STRAPDOWN INERTIAL NAVIGATION SYSTEM

Zhao Yong

(Dept. of Automatic Control, Beijing Univer. of Aero. & Astro. Beijing, 100083)

摘 要 本文介绍在 PC 机上开发捷联惯导系统软件并进行数字仿真的方法, 讨论了从分配软件任务, 到开发实时软件、管理软件、仿真软件, 直至固化运行的全过程。还介绍了 ASM-86 语言、PL/M-86 语言与 C 语言混合编程的技巧与优点。该方法与传统的在 86/386 开发系统上开发捷联系统软件的方法相比, 具有良好的经济效益与实用价值。同时, 本方法具有通用性, 可用在以 8086 为基体的单板机软件开发上。

关键词 捷联惯导系统, 软件开发, 实时系统

Abstract This paper introduces a method to develop the software of strapdown inertial navigation system in personal computers, and discusses the whole process from assigning the task of software, developing real-time software, management software, simulation software to running in the special computer systems. The paper also discusses the mixed programming method and its advantages using the C, PL/M-86, and ASM-86 programming languages. The method discussed in the paper is more economical and practical than the ordinary methods which are used in 86/386 open systems. The method can be spread for general use to develop the software for 8086-based computers.

Key words strapdown inertial navigation systems, software development, real-time systems

捷联系统的应用软件运行环境为 iSBC86/20 单板机, 该单板机将作为弹上计算机系统的主模块。iSBC86120 单板机为 INTEL 公司 16 位单板机系列产品, 其 CPU 为 8086-2, 并配有可进行超越函数运算的协处理器 8087 芯片, 装有 128kB 的动态 RAM, 可满足捷联系统的计算要求。

1 软件的开发特点

软件编制的最大特点是采用了 ASM-86 语言、PL/M-86 语言和 C 语言(iC86 编译版本)的互相混合编程技巧, 充分发挥了各自的优点, 起到了取长补短的作用。同时, 本文软件在 PC 机上通过仿真并在实际系统中获得应用, 能满足实时要求与精度要求。

1.1 3 种语言的特点与任务分配

ASM-86 语言属于 8086/8087 汇编语言, 是针对计算机硬件的, 能直接参与计算机的存储空间分配, 直接使用计算机的各种资源, 但用汇编语言编程, 存在编写效率低, 出

1991 年 3 月 20 日收到, 1991 年 12 月 27 日收到修改稿

错多、查错费时、费力等缺点。故 ASM-86 语言主要用于系统硬件初始化、系统可编程芯片(如 8251、8253、8255、8259 等)的编程和一小部分实时程序。

PL/M-86 语言是一种非常适用于开发系统软件与应用软件的高级语言,具有可维护性好和可靠性高的优点,故使用 PL/M-86 语言开发了捷联惯导系统的大部分弹上运行软件(实时软件与管理软件)。但 PL/M-86 语言直接提供给用户的 I/O 功能较差,我们用 C 语言来实现这部分功能。

C 语言具有表达能力强且灵活和可移植性好等特点。如 C 语言可以直接处理字符、数字、地址。在本方案中,C 语言主要用做主框架程序来调用 PL/M-86 模块和 ASM-86 模块。

1.2 从 ASM-86 语言程序中调用 PL/M-86 程序

对于数据段,在 PL/M-86 语言中声明为

```
declare data-pl0 byte public,
        data-pl1 real public;
```

在汇编语言中,做如下声明即可引用:

```
data segment public 'DATA'
    extrn data-pl0: byte
    extrn data-pl1: dword
data ends
```

对于 PL/M-86 语言编写的程序段做如下声明:

```
programpl: procedure public;
end programpl;
```

在汇编语言中做如下声明即可引用。

```
extrn programpl: near
```

PL/M-86 语言与 ASM-86 语言相互调用按常规。

1.3 从 C 语言调用 ASM-86 语言及 PL/M-86 语言

如变量 data-ap 在 ASM-86 或 PL/M-86 中声明为实数,在 C 语言中做如下声明即可引用:

```
extern float data-ap;
```

如程序段 programap 在 ASM-86 或 PL/M-86 中声明为公用模块,则在 C 语言中做如下声明即可调用。

```
extern programap( );
```

这部分详细内容可见有关文献。

2 捷联系统软件的开发与仿真

根据软件的任务,将捷联系统的软件分为实时软件与非实时软件两部分,又称非实时软件为管理软件。为了能在 PC 机上调试系统软件,用 C 语言编写一个主模块做为仿真软件。

2.1 实时软件的开发

实时软件开发的依据为捷联惯导系统的数学模型^[1]。

约定: $X_i Y_i Z_i$ 表示惯性坐标系, $X_e Y_e Z_e$ 表示地球坐标系, $X_n Y_n Z_n$ 表示地理坐标系, $X_b Y_b Z_b$ 表示弹体系。

λ 、 φ 分别表示当地的经度和纬度, 则位置矩阵 C_e^n 为:

$$C_e^n = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\varphi\cos\lambda & -\sin\lambda\sin\varphi & \cos\varphi \\ \cos\varphi\sin\lambda & \cos\varphi\cos\lambda & \sin\varphi \end{bmatrix} \quad (1)$$

将从弹体系向地理系的转换矩阵称为捷联矩阵 T , 选用四元数法作为捷联矩阵的即时修正算法。设弹体系相对地理系的转动四元数为 $Q = q_0 + q_1 i_b + q_2 j_b + q_3 k_b$, Q 的即时修正可通过求解下面的四元数微分方程来实现。

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_{nbx}^b & -\omega_{nby}^b & -\omega_{nbz}^b \\ \omega_{nbx}^b & 0 & \omega_{nbz}^b & -\omega_{nby}^b \\ \omega_{nby}^b & -\omega_{nbz}^b & 0 & -\omega_{nbx}^b \\ \omega_{nbz}^b & \omega_{nby}^b & -\omega_{nbx}^b & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (2)$$

由 (2) 式求得 q_0, q_1, q_2, q_3 以后, 用下式求得捷联矩阵 T

$$T = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3)$$

求解得到 T 后, 根据 T 的 θ, γ, ψ 表达式, 即可由 T 阵求得姿态角 θ, γ, ψ 值, 本文不再详述。

速度与位置的计算, 可以通过求解下面的方程来实现。首先进行比力的坐标转换, 可以将加速度计测量的比力 \vec{f}^b 通过矩阵 T 转换为 \vec{f}^n , 即

$$\begin{bmatrix} f_x^n \\ f_y^n \\ f_z^n \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{bmatrix} f_x^b \\ f_y^b \\ f_z^b \end{bmatrix} \quad (4)$$

地速 \vec{V} 的即时修正可以通过解下面的微分方程来实现。

$$\begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{V}_z \end{bmatrix} = \begin{bmatrix} f_x^n \\ f_y^n \\ f_z^n \end{bmatrix} + \begin{bmatrix} 0 & 2\omega_{iez}^n + \omega_{enz}^n & -(2\omega_{iey}^n + \omega_{eny}^n) \\ -(2\omega_{iez}^n + \omega_{enz}^n) & 0 & 2\omega_{iey}^n + \omega_{eny}^n \\ 2\omega_{iey}^n + \omega_{eny}^n & -(2\omega_{iey}^n + \omega_{eny}^n) & 0 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (5)$$

其中

$$\begin{bmatrix} \omega_{ie x}^n \\ \omega_{ie y}^n \\ \omega_{ie z}^n \end{bmatrix} = \begin{bmatrix} 0 \\ \omega_{ie} \cos\varphi \\ \omega_{ie} \sin\varphi \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} \omega_{en x}^n \\ \omega_{en y}^n \\ \omega_{en z}^n \end{bmatrix} = \begin{bmatrix} -V_y / (R_m + h) \\ V_x / (R_n + h) \\ V_x \operatorname{tg}\varphi / (R_n + h) \end{bmatrix} \quad (7)$$

其中 h 为导弹飞行高度, R_m 为参考椭球子午面内的曲率半径, R_n 为垂直子午面内的法线平面内的曲率半径。

由于捷联系统的惯性组件直接固连到弹体上, 陀螺与加速度计将直接感受弹体的角运动与线加速度, 对于一般的飞机与导弹来说, 其姿态速率 $\bar{\omega}_{nb}^b$ 的数值可能较高 (可高达 $100^\circ / \text{s} \sim 400^\circ / \text{s}$), 故选用阶数较高的四阶龙格-库塔法做为四元数微分方程的更新算法。并且, 在计算机硬件速度允许和在软件程序尽可能优化的情况下, 采用足够小的迭代周期。四元数微分方程的迭代周期用 T_f 表示, 本文取 $T_f = 0.025$ 。龙格库塔法要求给出采样值中间点的 $\bar{\omega}_{nb}^b$ 值, 故陀螺输出数据的提取要求每 $T_f / 2$ 进行一次。对(5)式、(6)式、(7)式进行分析可知, 在飞行器以约 $1000\text{m} / \text{s}$ 的速度飞行时, 其速度微分方程中 $\bar{\omega}_{en}^n$ 的变

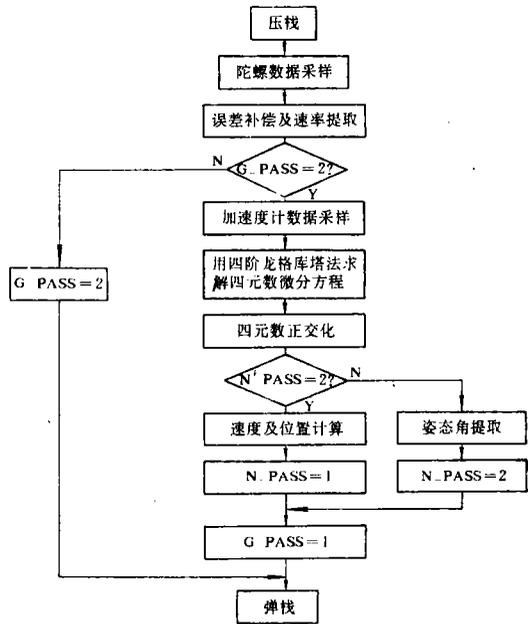


图 1 实时软件程序框图

化量在 mT_f (其中 m 为正整数, 取值为 2 至 5) 时间内也是不大的, 故速度微分方程用一阶欧拉法求解, 并且其步长为 mT_f (取 $m=2$)。对于与速率有关的位置矩阵的更新, g 的计算, 高度的计算等任务, 其迭代周期亦取 $2T_f$ 。根据实时软件的任务与时间分配的基本情况, 使用 PL/M-86 语言及少量汇编语言编写了捷联系统的实时软件程序, 其框图如图 1 所示。

2.2 管理软件的开发

本方案中, 管理软件的主要任务是建立实时程序的运行环境并对实时程序进行调配。在实验室中, 管理软件的另一个任务是通过人机对话的方式对弹上程序进行控制来完成不同的任务。

实时软件每 T_f 调用一次, 这就要求计算机产生时间基准信号, 本方案中, 时间基准

的确定是通过对 8253-5PIT 的编程来实现的。弹上计算机晶振出来的经整形后的方波加到 8253-5 计数器 1 的输入端，通过对 8253-5 编程设置后，其输出端可产生周期为 10ms 的方波，将此信号接到 8259A 的输入端，可定时产生中断请求信号。如 8253-5 计数器 1 的输入频率为 153.6kHz，那么计数初始值置为 600Hz 即可。

实时软件做为中断服务程序被调用，中断的管理是通过对 8259A 中断控制器的编程来实现的。8259A 允许 8 个中断源输入，其输入端为 IR0-IR7。在管理程序中，对 8259A 写相应的控制字，就可完成对中断的管理方式，对中断源输入的屏蔽情况等功能的控制。同时，在硬件上，要将 8253-5PIT 的计数器 1 的输出端接到 IR0-IR7 中间的一个上(本文选 IR1)。在管理程序中，加入对应的中断向量地址表。

管理软件的另一大任务是实现与 PC 机通讯，从 PC 机上接收指令，向 PC 机传送计算结果。通讯功能利用 86/20 板的串行口与 PC 机的串行口实现串行异步通讯，弹上机的通讯软件用 PL/M-86 语言与 ASM-86 语言混合编程，PC 机上的通讯软件用 Turbo C 语言与 MASM 语言混合编程。有关这两种语言的使用情况可参见有关资料。管理软件的基本流程图见图 2。

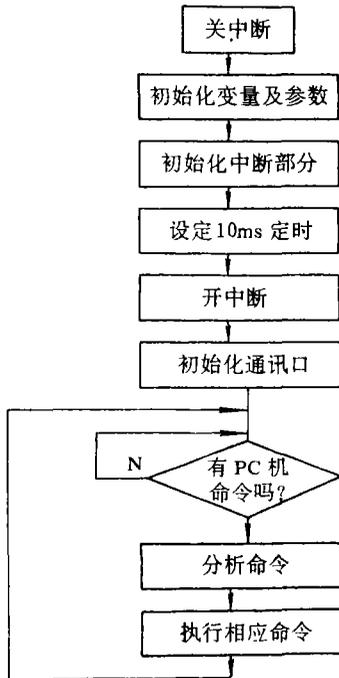


图 2 管理软件程序框图

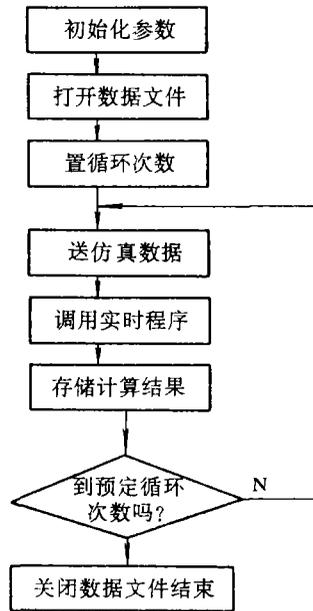


图 3 仿真软件程序框图

2.3 仿真软件的开发

为了验证实时软件的正确性，并考察计算精度，必须进行仿真分析。方法是给出一条理想的飞行轨迹，用其对应的陀螺与加速度计的输出值作为实时软件运算的输入值。

产生理想轨迹的程序在 PC 机上用 FORTRAN 语言编写。程序的输入值有目标点与发射点的经纬度，导航频率与发射仰角等值，输出值为该弹道对应的陀螺与加速度计的输出值，同时还有对应的导航参数。仿真软件在 PC 机上用 C 语言来编写，以 C 语言为主程序来调用实时应用程序。C 语言有很强的文件管理能力，可以很方便地存储并处理计算结果。仿真软件程序框图见图 3。

3 用户软件向弹载计算机上的移植

在开发捷联系统软件时, PC 机只是一个开发工具, 用户软件最终要在弹载计算机上运行, 这就要求把在 PC 机上仿真通过的用户软件移植到弹载计算机上。

用户软件在 PC 机上仿真时, 有 PC 机的操作系统(如 DOS)的支持, PC 机硬盘可以用来存储文件, 但用户软件在弹载计算机上运行时, 无操作系统支持, 也不挂硬盘, 最终程序固化在 EPROM 芯片中直接运行。由于存在上述差异, 在编写用户软件时, 就要求不调用系统命令, 用户程序自成模块。程序的开发可以分为以下二个部分。

(1) 在 PC 机上建立开发环境 在 PC 机上安装 ASM-86 语言, PL/M-86 语言, IC-86 语言的编译程序。用这些编译程序对相应的源程序进行编译, 产生目标代码。再安装工具程序 LINK86, LOC86, LIB86。LINK86 把 8086 目标模块连接在一起, 形成可浮动定位模块, 该模块在 iRMX86 操作系统支持下可运行, 也可在 PC 机上经过格式转换后, 在 DOS 系统下运行。LOC86 将可浮动定位目标模块转换成绝对目标模块, 绝对目标模块包含起始地址设定, 段址设定和各段的大小等信息, 可直接在弹载计算机上运行。LIB86 产生并维护程序库, 同时, 还要在 PC 机上安装可供调用的基本库文件。这样, 就在 PC 机上建立了一个既能对用户程序进行仿真, 又能经定位, 固化将用户程序移植到弹载计算机的软件开发的支持系统。

(2) 在 PC 机上开发用户程序的过程 在 PC 机上建立了开发环境以后, 就可按下面的基本步骤开发软件。首先是按各子任务编写不同功能的程序模块, 这样可以分别调试, 分别编译, 提高编程效率。源文件经编译与连接后, 产生可浮动定位模块。对该模块进行适用于 PC 机 DOS 操作系统的转换, 可在 PC 机上进行仿真运行。对同一模块再进行定位以后, 可产生绝对目标码, 使用编程器即可将用户软件固化运行。其基本过程见图 4。

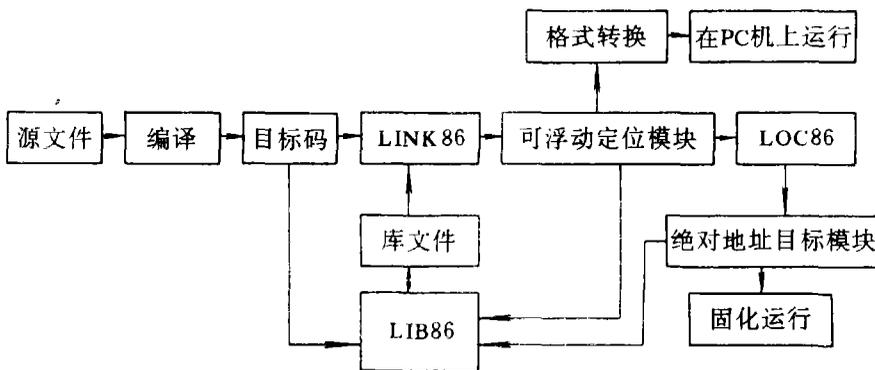


图 4 用户软件开发过程

4 结 论

用本文介绍的方法开发出的捷联系统的软件已用于实际系统, 并已在实验室条件下和靶试条件下证明运行是正确可靠的。该方法与传统的在 86/380 开发系统上开发软件的方法

法相比,有良好的经济效益与实用价值。一般而言,PC 机比 86/380 系统使用的较为广泛,其价格为 86/380 系统的 $1/10 \sim 1/3$,其速度(对 386 机型)约为后者的 4~8 倍,可大大缩短软件开发时间。同时,PC 机比 86/380 有更多的支持软件,便于进行数据处理。本方法具有通用性,可用于开发以 8086 为基体的单板机用户软件。

参 考 文 献

- 1 崔中兴. 惯性导航系统. 北京: 国防工业出版社, 1982. 188~203
- 2 陈哲. 捷联惯导系统原理. 北京: 宇航出版社, 1986. 144~161
- 3 赵勇. 双机捷联惯导系统管理软件研究. 北京航空航天大学学报, 1991, (2): 42~50
- 4 赵勇. 地对地导弹捷联惯导系统实时软件仿真. 梁炳成编, 91年全国航空航天系统仿真学术交流会议论文集, 重庆, 1991, 北京. 168~175
- 5 Intel Corporation. iAPX86, 88 Family Utilities User Guide for DOS' Systems, 1985, USA, 1~10