

基于模型的预警无人机飞控系统开发与验证

罗喜霜, 宋亮, 雷玮, 郑亮, 金海洋

(北京经纬恒润科技有限公司, 北京 100191)

摘要: 为应对城市低空预警任务对固定翼无人机飞控系统自主飞行需求, 采用基于模型设计方法完成无人机飞控系统开发与验证; 开发过程中以飞控算法模型为中心, 逐级开展飞控算法模型设计与数学仿真、算法快速原型验证以及飞控系统硬件产品实现; 飞控算法模型在 Matlab/Simulink 平台中完成了构建及数学仿真验证, 结合 Higate 系统提供的实时仿真环境完成算法快速原型验证, 基于代码自动生成方式将算法模型自动转换为可在 DSP 中运行的实时代码, 下载到飞控计算机中, 进行硬件在回路仿真验证, 并进一步进行整机地面验证; 经过实践, 所提出的飞控系统开发与验证流程可以将系统设计中的问题缺陷在开发早期暴露出来并及时修正, 保证了系统研制进度、成本和品质。

关键词: 基于模型的系统设计方法; Higate 实时仿真系统; 硬件在回路; 快速原型; 代码自动生成

Model-based Development of Police UAV

Luo Xishuang, Song Liang, Lei Wei, Zheng Liang, Jin Haiyang

(Hirain Technologies Co., Ltd., Beijing 100191, China)

Abstract: For UAV-based urban airborne alert, a high reliability design and verification procedure is developed for the capability of autonomous path planning and autopilot controller of an unmanned aerial vehicle. Followed by model-based design pattern, the design and verification of flight control system is carried out step by step, control law software is designed and verified, then rapid-prototyping and hardware in the loop. The control law software is developed in terms of Simulink model and simulated synchronously. The confirmed software is then compiled and download in Higate real-time system for rapid-prototyping. When DSP-based controller board is manufactured, control law software is downloaded, and hardware in the loop simulation is conducted. At last, full scale UAV ground test is proceeded and verified. With the proposed design and verification method, most faults are explored to and modified timely by the designer early in the program, which guarantees the progress, cost reduction and high quality.

Keywords: model-based design; Higate; hardware in the loop; rapid prototyping; code generation

0 引言

随着无人机技术的普及和发展, 城市低空空域态势愈发复杂, 低空域事件具有突发、隐蔽、难处置及威胁大等特点, 容易成为城市安全隐患, 因此亟需完善低空空域应急预警手段。

考虑到响应快速性和使用灵活性, 采用中型固定翼无人机作为低空空域预警执行手段。飞控系统是无人机最核心的机载系统, 决定了其任务执行能力。特别的, 城市低空空域飞行限制较多, 存在飞行区域、飞行高度等诸多约束, 使得无人机可行飞行路径不是任意的, 存在禁飞区域。为了灵活应对突发事件, 无人机必须能够自主进行航路规划绕开禁飞区域执行任务。航路规划得到的航路后将由自动飞行控制执行, 实现航路跟踪, 是航路规划功能的执行基础, 决定了无人机飞行品质。所研无人机飞控系统的核心算法即由自适应航路规划和自动飞行控制两部分组成。

所研飞控系统主要包括飞控计算机(简称控制器)、传感器和执行机构。其中传感器、执行机构均采用货架产品, 通过任务需求分解出功能性能要求进行设备选型。控制器是飞控系统的核心, 通常小型消费级无人机飞控多采用开源飞控构架, 如 Arduio、APM 以及 PixHawk 等。然而开源飞控硬件性能

通常较低, 不能满足较高复杂度飞控算法的计算要求, 且其软件构架存在较多冗余, 无法充分利用硬件资源。因此一些涉及复杂算法或高级应用的无人机飞控系统常采用 DSP 作为核心计算单元^[1], 辅以外围电路与其他机载设备通信。所研无人机飞控计算机以两片高性能计算机为计算中心运行核心算法。

作为机载安全关键系统, 在整机试飞前的各个设计开发环节, 需要逐阶段对其进行全面的试验验证, 保证其功能性能符合设计要求。对于中大型无人机, 功能复杂性和可靠性要求的提高使得飞控系统也变得复杂, 为了高效进行新机型飞控系统的研发及成熟机型飞控的改进, 需要在系统逐步成型的过程中对设计和实现进行充分的验证, 保证在系统进入下一阶段研制流程前将所有潜在问题暴露出来并解决, 为下一阶段研制提供一个尽量干净的基础。然而这一需求目前并没有得到无人机行业的普遍重视。相比于无人机, 目前国内有人机各主机所在型号研发的过程中更加重视研发各个阶段的确认与验证工作, 以便在型号早期尽可能多地暴露设计、制造缺陷, 及时进行技术变更, 确保产品技术状态满足当前阶段的要求, 为后续工作奠定坚实基础, 进而达到缩短研发周期、降低研发成本的目的。

基于模型的飞控系统开发的核心是飞控算法软件, 其设计验证流程主要包括算法建立与数学仿真验证、半实物仿真验证(又可分为算法快速原型和硬件在回路)、整机地面联试与飞行试验。在控制系统设计领域, SCADE^[2]和 Simulink^[3]是应用较多的开发环境。SCADE 价格昂贵, 在机载软件适航性验证方面做得更好, 多为国内有人机主机所使用, 并在某些机型飞

收稿日期: 2017-04-25; 修回日期: 2017-05-21。

基金项目: 北京市科技计划项目基金(D161100005816002)。

作者简介: 罗喜霜(1976-), 女, 广西马山县人, 博士, 高级工程师, 主要从事飞行制导导航控制方向的研究。

控系统开发中强制使用；而 Matlab/Simulink 依托其在控制系统设计仿真方面全面的表现使其在中小型无人机飞控开发中有很好的应用前景。

由于飞控系统硬件设备多为货架产品，本文将围绕飞控核心算法的设计与开发进行展开，论文范围包括飞控算法模型建立与数学仿真验证、飞控算法快速原型、飞控设备硬件在回路、整机地面联试。在无人机飞控系统的开发过程中，实践了上述研发方法并结合所研固定翼直升机特点进行了针对性的改进，给出了各流程的详细方法和关键点，旨在探索一种高效的、可标准化的无人机飞控研发流程行业准则。

1 基于模型的无人机飞控开发与验证

采用 Matlab/Simulink 软件作为无人机飞控系统开发环境，充分利用其图形化设计、代码自动生成以及丰富的控制系统设计工具，快速可靠地完成飞控算法软件从数学仿真、快速原型到硬件在回路仿真的完整流程，逐层实现飞控系统功能性能验证，基于模型的预警无人机飞控系统开发验证流程如图 1 所示。各个研制阶段均以算法模型为基础，所有变更均直接体现在模型上，并基于模型生成各阶段所用的代码。

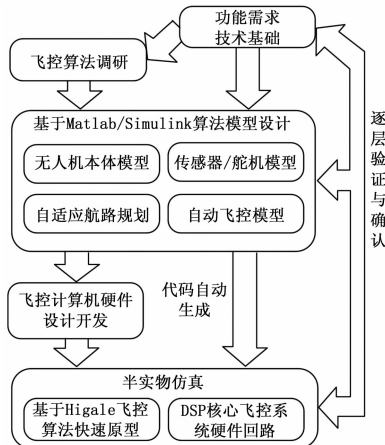


图 1 基于模型的无人机飞控系统开发验证流程

根据预警无人机需自主实现城市复杂环境下低空巡航与应急处置的需求，确定飞控系统应具有自适应路径规划和自动飞行控制功能。综合考虑已有技术储备和功能需求，确定了基于视通图法的自适应规划算法和常规自动飞行控制算法实现方案。在 Matlab/Simulink 环境下完成无人机本体模型、环境模型、传感器/舵机模型、自适应航路规划和自动飞控算法建模，并进行不同层级的仿真，验证算法模型的功能性能满足顶层需求。在项目前已经辨识出自适应航路规划算法模型需要消耗较多存储资源，在算法模型开发完成并经过数学仿真验证后，初步确定了其资源消耗需求，并基于此进行飞控计算机硬件选型。基于 Higate 实时仿真系统，通过代码自动生成技术实现 Simulink 算法模型的快速原型验证，验证算法模型功能性能，并初步验证系统接口。基于 DSP 的飞控计算机硬件开发完成后，采用代码自动生成方式快速生成可在 DSP 中运行的飞控软件代码，并基于 Higate 系统提供的闭环仿真环境，实现核心飞控系统硬件在回路仿真验证。飞控系统研制过程中，每个开发环节在充分验证后解决所有暴露问题后再进入下一环节，若通过验证发现前一阶段的某项设计存在缺陷，则需要追溯前一阶

段的设计对问题项进行改进。随着设计的进行，系统中潜在的缺陷均在其可能出现的早期进行修复，避免同一问题潜伏至开发后期造成成本和开发周期上更大的损失。

2 城市预警无人机飞控系统

2.1 系统组成

飞控系统功能模块主要包括控制器、传感器和执行机构。此外，飞控系统与地面站系统和探测系统存在闭环交联，系统组成原理如图 2 所示。

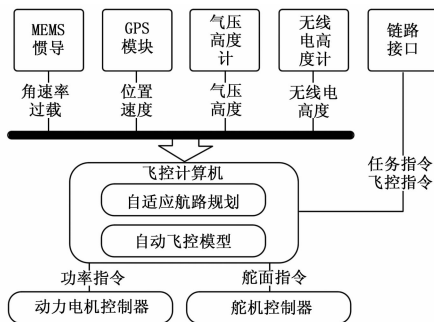


图 2 飞控系统组成框图

2.2 控制器

控制器核心飞控算法为自适应航路规划算法和自动飞行控制算法，由于算法复杂性较高，控制器电路板围绕 TMS320C28346 这款高性能微处理器搭建并辅以外围电路。基于模块化思想，考虑功能验证便利性，自适应航路规划模块和自动飞控模块在两块电路板中实现，并安装在同一结构件中。

2.3 传感器与执行机构

传感器和执行机构根据飞行任务分解得到的性能指标进行选型。自适应航路规划算法以无人机经纬高、速度、航向为输入，自动飞行控制算法以三轴姿态及角速度、航向、法向过载、速度为输入，因此确定传感器包括三轴陀螺+三轴加表 MEMS 惯导器件、卫星导航设备、无线电高度表、气压高度表及磁航向传感器。此外，自适应航路规划模块的障碍规避功能需要光学+雷达探测载荷的测量信息，这部分探测功能由专门的探测预警系统提供。考虑到所研无人机飞行包线较小，无需大范围的系统调参，故在基本配置中没有包含大气参数相关传感器。

2.4 飞控系统与通信链路接口

预警无人机以自主飞行为主要模式，手动控制模式备份。飞控系统通过链路终端与地面站通信，接收手动控制指令和地面任务设置指令。

3 飞控算法建模与数学仿真

所研飞控算法基于 Matlab/Simulink 开发环境，采用基于模型的设计范式，借助其图形化设计、早期验证、代码自动生成等特性和较为完备的建模与控制系统设计工具箱，提高开发质量与进度。飞控算法的核心是自动飞行控制算法和自适应航路规划算法，自动飞行控制直接作用于无人机本体，决定无人机飞行品质。自适应航路规划为自动飞控外环，两者串联实现无人机自主飞行功能。为了对核心飞控算法进行闭环验证，需要无人机本体模型、环境场模型、舵机以及传感器模型。

3.1 无人机本体数学建模

无人机本体模型是飞控算法的被控对象，其准确程度是算

法参数乃至算法结构有效与否的关键。一般地, 获取本体模型的方法包括理论分析、风动试验和飞行试验^[4]。风动试验和飞行试验能够获得较为准确的模型。但风动试验代价较大, 一些低成本项目无法承受。而飞行试验在无人机研发后期才能进行, 一般作为本体模型最终验证与改进的途径。理论分析方法可以在项目初期给出一个在较大范围内可信的本体模型用于飞控算法设计开发。

由于所研无人机的气动外形与动力系统均为全新设计, 且无风洞试验条件, 因此需要在研发前期采用理论分析方法进行本体建模。采用 Simulink 航空宇航工具箱中的固定翼飞机气动建模模块 DATCOM, 建立初步的自动飞控被控对象。DATCOM 模块使用 NASA 开源建模软件 digital DATCOM 解算的结果, 将所研无人机几何参数和质量参数输入 digital DATCOM 中即可得到 Simulink 中可用的气动力气动力矩计算模块。DATCOM 模块以迎角、侧滑角、Ma 数等飞行参数为输入, 输出对应该状态的气动力和气动力矩并作用到六自由度飞机方程, 作为无人机气动计算环节实现闭环控制仿真验证。

在数学仿真验证阶段, 基于 Digital DATCOM 得到的本体模型虽然准确度有限, 但足够作为飞控算法开发的基础。该本体模型将在飞行试验中通过系统辨识进行修正。

3.2 自适应航路规划算法建模

基于通视图法进行自适应航路规划算法设计, 并基于 Matlab 进行算法建模。自适应航路规划算法是飞控系统的最外环, 其输入条件为起始点和禁飞区信息, 不依赖于无人机本体和自动飞控模块, 对该算法的仿真验证可以开环进行。自适应航路规划算法的执行包括 5 个步骤:

1) 规范禁飞区: 将不规则几何形状的禁飞区都等效为圆形禁飞区;

2) 采用圆的内外公切线的方法, 求解起点位置到终点位置间的所有直线可行路径, 各直线可行路径与圆形禁飞区域的交点是可行航路点,

3) 根据所有直线可行路径求解从起点位置到终点位置间的所有可行航线, 将路径最短的可行航线确定为最优航线;

4) 根据最优航线与最优航线上可行航路点的属性, 确定起点位置到终点位置间的最优路径。

3.3 自动飞控算法建模

自动飞控算法模型以航路规划给出的航线信息控制无人机进行航线跟踪, 包括导引功能、自动驾驶功能和控制增稳功能。自动飞控算法需要保证除起飞降落模式以外的全包线飞行品质, 因此对非线性本体模型根据飞行包线范围选取若干工作点进行配平并线性化得到线性被控对象, 然后分别针对不同被控对象进行控制器设计。采用经典控制系统设计方法, 设计内容主要包括俯仰、滚转、偏航通道前向通路指令成型滤波器、角速率洗出滤波器及前向通道控制参数的设计, 具体构型参考^[5]。进行控制器设计时充分利用 Simulink 控制系统工具箱提高设计过程的自动化程度: 采用配平函数及工具直接对模型进行配平, 获得平衡点; 基于平衡点自动进行线性化得到线性被控对象; 采用 Simulink System Tuner 进行多通道耦合控制器优化, 获得控制器、滤波器和增益环节的参数。基于线性化模型设计的控制器首先与线性模型组成简单闭环, 进行仿真验证。针对所有平衡点完成控制器设计后, 将控制器与原非线性模型组成闭环, 针对单一平衡点进行飞控效果的验证。然后以

平衡点参数作为插值变量对所有控制器进行插值调用, 并针对全飞行包线进行闭环仿真验证, 确认控制器在全包线飞行时的功能与性能。

自动飞控按照纵侧向解耦方式实现垂直面和水平面控制。垂直面自动驾驶包括高度保持和定速爬升两种模式, 垂直导航系统通过升降舵在垂直面控制飞机按预定垂直速度爬升或下降进入某一高度, 并进行高度保持。垂直速度和高度保持的控制回路的内回路均为俯仰角控制。水平面控制通过副翼和方向舵两个通道的耦合控制飞机在水平面的轨迹运动。副翼通道为主通道, 方向舵为辅助通道, 后者只起到阻尼和协调转弯的作用, 通过副翼控制飞机转弯以便修正飞机的轨迹。

3.4 环境场/舵机与传感器模型

环境场主要为风场, 直接使用 Simulink 航空宇航工具箱中的风切变模型、剖面模型和紊流模型建立不同的风环境, 实现对算法的充分验证。传感器模型模拟真实传感器的动态特性和噪声特性, 本体模型输出的飞行参数经过该模型转换为传感器测量参数作为飞控算法模型的输入。传感器特性的引入可以使飞控算法的闭环仿真环境更贴近实际, 验证结果更加可信。传感器模型的搭建主要根据所选传感器标定文档所提功能的数据。舵机模型根据所选型号的动态特性进行建模, 需准确建模出舵回路的带宽。

3.5 闭环仿真验证

将自动飞行飞控算法模型、自适应航路规划算法模型与其他辅助模型组成飞控闭环, 在 Simulink 环境下进行任务级仿真验证, 验证完整飞控算法的功能和性能。

根据预警无人机的任务特性, 设置仿真场景。根据城市空管要求建立禁飞区域, 将常规巡航飞行模式和应急飞行模式结合, 选择不同的风场环境, 完成场景配置。在 Simulink 中进行非实时闭环仿真, 自适应航路规划方面主要验证所规划航线的合理性和最优性, 自动飞控方面主要考察实际飞行路线对规划航线的跟踪性能以及姿态变化的动态性。

4 飞控算法快速原型仿真

飞控算法快速原型是飞控核心算法与飞控系统硬件设备联试的第一步, 可以对实时环境下算法功能性能进行快速验证, 并对接口通信功能进行初步验证。快速原型中所用的飞控软件由经过数学仿真验证的算法模型自动生成, 并在 Higale 实时仿真系统中运行。在快速原型阶段, 需要为飞控算法提供实时、完整的闭环验证环境, 全面充分地激励算法特性。

4.1 Higale 实时仿真环境

HiGale 系统是一套基于半实物实时仿真技术的控制系统开发及测试的工作平台。无人机飞控系统的开发过程中的快速原型及硬件在回路仿真均以 Higale 系统为支撑环境开展。系统由仿真机及 IO 板卡等硬件, 和 HigaleTarget/Higale RTD/HigaleView 等软件组成。系统硬件基于高可靠性 cPCI/PXI 总线, 多套硬件设备可构成分布式仿真系统。系统基于 RTW 技术通过 HigaleTarget 与 Simulink 软件无缝连接, 提供仿真模型的代码生成、编译、下载等功能, 并支持 QNX 及 VxWorks 实时操作系统。HiGaleRTD 为实时驱动软件库, 提供多种 IO 通信驱动能力, 实现 Simulink 模型与硬件平台间的无缝衔接。HiGaleView 为实验管理软件, 内置丰富的虚拟仪器仪表控件, 可实现仿真数据实时显示、存储, 控制参数在线调整。

4.1.1 IO 接口模型

数学仿真阶段使用的模型中没有 IO 通信功能,需基于 HigaIeRTD 的 IO 接口驱动模块实现该功能。根据接口类型,在 HigaIeRTD 库中选择对应的通信模块,并根据通信协议进行总线数据的打包和通信参数设置。

4.1.2 飞控软件自动生成

基于 HigaIeTarget 将飞控算法模型直接转换成可在实时仿真系统中运行的、c 语言形式的飞控软件。在搭建模型时应充分考虑建模限制,所研飞控算法最终需要下载到 DSP 中运行,因此其对应的模型必须使用 discrete 离散求解器,相应的,算法模型及辅助模型必须满足离散求解器的要求,不能使用连续状态模块。此外,对于具有可变形参的 Matlab 函数,如取模函数 norm,在生成代码时会根据维数、数据类型的不同生成多个函数体实现,且会根据内部机制生成一个超限的中间变量,使得在 DSP 开发环境中编译错误。这种情况下应采用非向量操作的等价算法取代问题函数。

4.2 快速原型系统构架

预警无人机飞控算法快速原型以 HigaIe 实时仿真系统为核心,系统由三台实时仿真机、上位机、以太网交换机和无人机手动操作设备构成。实时仿真机由 NI PXI-1044 机箱、NI-8430 RS232 串口通信板卡和处理器板卡。

在实际使用时,自适应航路规划和自动飞控算法软件在两个独立的 DSP 中运行,为了尽可能模拟飞控系统真实物理环境,在快速原型阶段将两者分别在两台仿真机中下载。自适应航路规划与自动飞控均需接收传感器的数据,且自动飞控模块需向舵机伺服控制器发送控制指令,为了模拟真实接口,将相关模型下载到另一台仿真机中。仿真机分配所遵循的思想是,关键模块间的通信接口与实际使用接口一致,具有相同的接口类型、接口参数和通信协议,关键算法应具有与实际使用情况相同的独立运行环境。根据系统通信设计,传感器、自适应航路规划模块、自动飞控模块、舵机控制器均通过 RS232 进行数据传输,在快速原型系统中通过 NI-8430 板卡提供总线模拟。

在快速原型仿真中,HigaIeView、HigaIeTarget 以及 Matlab/Simulink 运行在上位机中,试验人员通过上位机控制试验的进行,包括模型代码自动生成与下载、实时仿真控制、在线调参与数据实时显示。

4.3 原型算法验证

对于所研无人机,由于条件所限,无法为验证系统配备完整的硬件环境,核心的外部激励参数均由实时仿真系统模拟。快速原型的目的是,在相同的飞行仿真条件下,确保实时运行的原型飞控算法控制结果与数学仿真阶段得到的结果在一定范围内是一致的。在符合建模规范的前提下,算法模型转换到实时仿真代码的过程不会造成显著的结果失配。造成结果不一致的可能原因主要包括模型拆分带来的模块执行顺序变化、物理接口引入带来的信号顺序变化等。尤其对于复杂模型,拆分后的输入输出端口被真实物理接口连接,在每一个仿真周期中有可能引入额外延迟。由于在进行飞控算法设计时考虑了系统的鲁棒性,延迟数据对计算部分影响较小,但对自主飞行导引逻辑的影响较大,可能会激发出潜在逻辑缺陷,而这些缺陷在原模型中是体现不出来的。

算法快速原型仿真中执行与数学仿真阶段相同的仿真算例,将得到的仿真结果与数学仿真结果进行对比,包括时域和

频域方面。时域方面,对比结果应有相似的趋势,且偏差应在给定范围内且没有显著的、非预期的超调,没有必要追求绝对的一致性;频域方面,控制内环、导引外环等各环节的稳定裕度应与数学仿真相似。由于 Simulink 的代码生成功能确保了模型与代码的一致性,若结果不满足以上要求,则应重点检查信号时序和逻辑方面的问题。比较好的检查方法是做飞控系统的极性与传动比试验,得到算法关键环节输入输出的关系,并与数学仿真阶段结果进行对比,确定问题发生的位置。

4.4 仿真结果

在快速原型仿真中,经过问题排查,最终得到的仿真结果与采用相同算例的数学仿真结果具有很高的一致性,验证了算法功能的正确性。图 3 给出了在某一测试算例下数学仿真(roll)和快速原型仿真(roll_rcp)中姿态角的结果对比,两者具有很好的一致性。

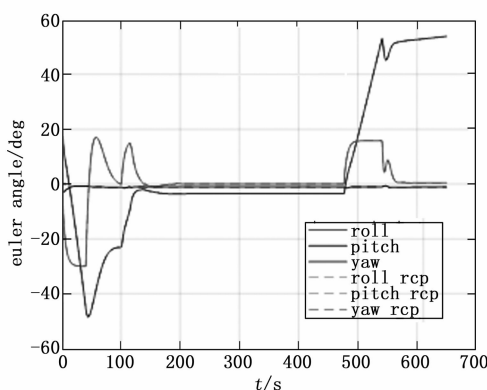


图 3 姿态角结果对比

5 飞控计算机硬件在回路仿真与整机地面试验

完整的飞控系统半实物仿真应包括飞控系统主要设备并通过物理效应模拟器将所有设备的真实工作环境激励出来,如使用转台模拟无人机姿态运动对陀螺进行激励,使用 GPS 模拟对 GPS 接收机进行激励,使用舵负载模拟器对舵机施加气动负载力矩。然而,由于中小型无人机项目经费一般较少,无法如大型飞机试验系统提供所有激励设备,传感器和舵机等设备仅参与完整飞控系统的接口验证,闭环仿真时仅飞控计算机接入。

5.1 飞控计算机硬件设计

飞控计算机主要包括自动飞控板和自适应航路规划板,分别独立运行自动飞控算法和自适应航路规划算法。

自适应航路规划板通过 232 串口接收地面站上传的指令及参数信息,接收各传感器反馈量,执行航迹规划算法,计算出各航点及控制指令,通过另一路 232 串口下发至自动飞控板。自动飞控板通过 232 串口接收自适应航路规划板下发的控制指令及传感器反馈信息,通过 FPGA 采集遥控器控制信号及电机转速反馈,DSP 处理器产生相应的控制信号控制舵机及动力电机执行相应动作,完成目标指令。在飞控系统运行时,自动飞控板实时存储飞行参数和关键的飞控算法中间参数,并存储在 SD 卡中,以供后续分析使用。飞控计算机功能如图 4 所示。

5.2 硬件回路仿真系统构架

相对于快速原型,硬件回路仿真中真实飞控计算机硬件将替代仿真机运行的飞控代码作为实际的控制单元。系统其他功

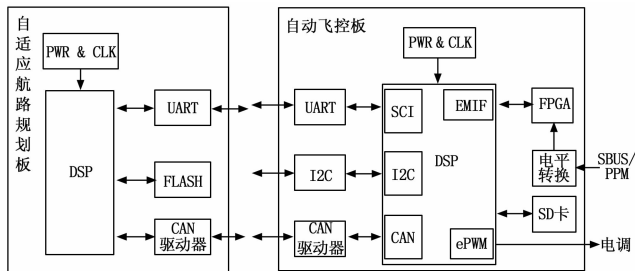


图 4 飞控计算机硬件组成框图

能与快速原型一致, 由 Higale 仿真机中的飞行仿真模型生成飞行参数激励, HigaleView 进行实时仿真监控。

硬件回路仿真中所需的验证科目与快速原型基本一致, 通过运行与前阶段相同的仿真算例, 验证系统的时域和频域特性。结合 HigaleView 和 CCS 实现代码调试、数据监控。基于 DSP 仿真器的调试功能在 CCS 中进行程序的调试操作, 步进程序排查问题; 将飞控算法中间变量从测试接口引出与输出变量发送到实时仿真机中, 借助 HigaleView 强大的数据可视化功能进行数据监控和保存, 保存后的数据在 Matlab 中进行进一步的对比分析。

5.3 针对 DSP 的飞控软件自动生成

联合 Matlab/Simulink 和 CCS (Code Composer Studio) 进行将飞控算法模型自动生成 DSP 中运行飞控代码, 并使用 CCS 手动进行必要的代码微调。

进行 DSP 飞控代码自动生成时, 飞控核心模型仍沿用先前成果, 但需要对 IO 通信接口驱动重新建模, 使用 Simulink Embedded Coder 工具箱中 C2834x 专用 IO 驱动模块替换快速原型中阶段使用的 Higale IO 驱动模块, 并使用 DSP 配置模块进行其他必要设置和资源分配。对于模块库中不提供的资源驱动, 需手动进行功能添加。所研飞控计算机采用大容量 SD 进行飞行数据存储, 模块库中没有现成的 SD 卡驱动, 在 CCS 环境下将 SD 卡驱动和读写逻辑手动编码并增加到算法软件工程中。模型由 Simulink 到 CCS 中 DSP 可执行代码的过程如图 5 所示^[6]。

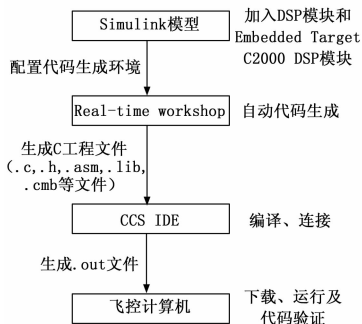


图 5 Simulink 自动生成 DSP 代码过程

5.4 整机地面试验

无人机各机载系统完成地面总装集成后, 对于飞控系统, 首先进行与其他系统的接口静态测试, 确保信号交联正确。然后进行核心的飞控系统极性测试, 即确定各个控制环节输入与舵面响应的关系, 保证符合设计要求。主要包括操纵杆——舵面、不同模式下自动飞控指令——舵面。手动飞行模式下, 无人机驾驶人员的操纵杆指令经过内环控制增稳后形成舵机控制

指令驱动舵面运动, 不同飞行状态下相同操纵指令舵面偏转角度甚至方向都是不同的, 通过算法代码中设置的测试模块静态设置若干典型飞行状态, 此时控制操纵杆进行预设的作动, 查看舵面偏转的极性, 并与标称结果进行对照。采用相似的方法进行自动飞控指令到舵面的极性检验类似。极性验证至关重要, 事实上, 相比于大型有人驾驶飞机, 由于中小型固定翼飞机的飞行包线较小、可控性好, 只要控制回路极性正确, 控制律参数在较大范围都能保证飞机飞行状态的稳定。整机地面试验阶段的闭环仿真中, 传感器和舵机等核心硬件都具备了试验状态, 但由于没有模拟器激励这些设备进行全包线仿真, 因此采用如下折中试验方式。仍使用 Higale 实时仿真系统进行飞行仿真, 将传感器等设备的仿真激励数据从飞控计算机测试接口输入; 对传感器进行静态激励使其输出静态值; 在飞控计算机的测试通道中将仿真激励数据与传感器静态测量输出值叠加, 作为仿真中实际使用的传感器等设备的输出值, 如此可以在有限的试验条件下最大限度地将在设备特性在闭环仿真中体现, 提高仿真真实性。由于实际传感器等设备特性的引入, 全包线闭环仿真结果与前几个阶段的结果相比有一定差异, 主要原因在于之前仿真中使用的模型不能完全准确模拟设备特性, 具体差异主要为噪声幅值和频谱差异和响应动态性差异。该原因导致如下现象: 处于平飞等稳定飞行状态时, 姿态回路在平衡位置处的震荡幅特性有差别; 处于机动状态时, 姿态、高度、速度、控制输出均有小幅的偏差。虽然一定差异, 但主要控制指标均得到满足, 达到试飞条件。如果整机地面试验中闭环仿真结果与先前结果有较大差异甚至飞控失效, 则应对传感器和舵机特性进行更准确的建模, 并依次重复上述各个研制阶段进行算法方面的调整, 直至问题解决。

事实上, 由于试验系统功能完整, 大型飞机飞控系统极性试验一般包含在硬件回路阶段 (铁鸟集成试验乃至更早的系统综合试验), 所研无人机受限于客观条件, 在先前阶段只能在信号层面进行极性试验, 直到整机集成后, 才有完整的硬件环境进行硬件级极性检验, 以及包括所有硬件的闭环仿真。

6 结论

应用基于模型的设计理念, 进行了某型城市预警固定翼无人机飞控系统开发。结合所研无人机飞行特点和研发条件, 优化了设计流程, 形成了适于无人机飞控系统快速、可靠的开发流程。经过数学仿真、快速原型、硬件回路仿真和整机地面试验, 完成了飞控系统试飞前的设计验证。

参考文献:

- [1] 刘 艳. 基于 DSP 的无人机飞控系统的研究 [D]. 重庆: 重庆大学, 2014.
- [2] 张雅妮, 李 岩, 李小勋. 基于 SCADE 的飞控软件的适航验证与确认 [J]. 飞行力学, 2012 (1): 34 - 37.
- [3] 云 超, 李小民, 郑宗贵. 基于 Matlab/Simulink 的硬件在回路无人机仿真系统设计 [J]. 计算机测量与控制, 2012 (12): 3306 - 3308, 3324.
- [4] 王冠林. 基于模式分割方法的小型无人直升机动力学频域辨识建模 [D]. 北京: 清华大学, 2012.
- [5] Thomas R. Yechout. Introduction to Aircraft Flight Mechanics [M]. AIAA, 2003.
- [6] 齐星刚, 赵 刚, 李 原. 在 MATLAB/Simulink 平台上 DSP 代码的自动生成 [J]. 中国测试技术, 2005 (1): 87 - 88, 106.