

Stateflow 逻辑系统建模

张 威 编著

西安电子科技大学出版社

2007

内 容 简 介

本书详细介绍了 MATLAB 产品体系中用于动态逻辑系统建模仿真的 Stateflow 产品的应用方法。本书的前六章内容是 Stateflow 的基础,重点介绍了有限状态系统中状态、状态转移、连接节点、历史节点、动作、数据对象、事件等各种概念在 Stateflow 产品中的实现方法,还着重介绍了层次化系统建模、并行机制、事件广播、隐含事件和时间逻辑等有限状态系统的建模方法。本书的后三章内容涉及 Stateflow 的高级话题,涵盖了 Stateflow Coder 代码生成、真值表、Embedded MATLAB Function 以及 Stateflow API 的使用方法等内容。

本书内容丰富、全面、系统而且权威,对 Stateflow 有限状态系统中的每一种语法现象都进行了详尽的介绍,并列举了丰富的应用实例,便于读者掌握具体工具的使用方法。

本书既可作为需要完成动态逻辑系统建模和仿真的工程人员的参考书,也可作为在校本科生、硕士研究生和博士研究生的教材,还可作为 MATLAB 相应产品培训课程的教材。

图书在版编目(CIP)数据

Stateflow 逻辑系统建模 / 张威编著. —西安: 西安电子科技大学出版社, 2007.10

ISBN 978-7-5606-1852-4

I. S… II. 张… III. 计算机辅助计算—软件包, Stateflow IV. TP391.75

中国版本图书馆 CIP 数据核字(2007)第 083728 号

策 划 毛红兵

责任编辑 阎 彬 毛红兵

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029) 88242885 88201467 邮 编 710071

<http://www.xduph.com> E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2007 年 10 月第 1 版 2007 年 10 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 20.5

字 数 482 千字

印 数 1~4000 册

定 价 30.00 元

ISBN 978-7-5606-1852-4/TP·0963

XDUP 2144001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

MATLAB 软件最早成为商品化软件是在 1984 年，那时由 MathWorks 公司推出了 MATLAB 的第一个版本。在编写本书的时候，MATLAB 的最新版本是 7.3，即 MATLAB Release 2006b。MATLAB 产品提供了丰富的应用函数，并且具有易扩充的开放性结构。在不断地吸收各行各业专家、学者、工程师的经验之后，MATLAB 已成为了国际上优秀的工程应用软件之一。目前，该软件涵盖了控制系统设计与仿真、数字信号处理设计与仿真、数字图像处理、嵌入式系统设计与实现、通信、神经网络、小波理论分析、优化与统计、偏微分方程、动态系统实时仿真等多学科专业领域，其应用行业包括航空航天、汽车、兵器与国防工业、通信、医药卫生、化工工业、生物遗传工程、大学教育、国家政府机关以及金融财经等。在全球，MATLAB 的正式用户已经达到六十余万，遍布世界上 100 多个国家和地区。而自从 20 世纪 90 年代 MATLAB 进入中国起，MATLAB 软件在国内已经拥有了众多用户。目前在国内很多高校已经在本科教学阶段将 MATLAB 作为一门必(选)修课程，并把它作为应用数学、控制、信号处理、通信等专业博士研究生、硕士研究生、本科生必须掌握的工具软件之一。

基于模型的系统级设计是 MathWorks 公司近年来大力倡导的一种系统设计方法，它定义了从系统设计、仿真到具体实现的一系列工具以及应用方法。基于模型的系统级设计的核心工具是 MATLAB 软件族当中以 Simulink 为基础的模块集合和相应的工具。而 Stateflow 是 MATLAB 产品体系中非常重要的一个分支，它是在基于框图的动态系统建模仿真环境——Simulink 的基础上完成动态逻辑系统建模与仿真的可视化开发平台。Stateflow 能够对那些基于有限状态机理论的事件驱动系统进行建模和仿真，也能够针对复杂逻辑系统进行建模和仿真。结合 Simulink 的动态系统建模和仿真能力，MATLAB 产品为广大用户提供了多领域系统建模和仿真的统一开发环境。目前，Stateflow 产品主要被用来完成下列系统中动态逻辑部分的建模和仿真工作：

■ 嵌入式系统应用：

航空电子设备；

车辆设备，例如中控门锁逻辑、电动车窗逻辑等；

电信设备，例如路由算法；

通信系统，例如计算机外设；

可编程逻辑控制器(PLCs)；

工业机械设备。

■ 人机交互界面(MMI)：图形化用户界面(GUI)的控制逻辑。

■ 复杂系统：例如航空空管通信系统，其中包含了 DSP、控制系统以及人机界面的开发等。

本书详细介绍了使用 Stateflow 进行动态逻辑系统建模和仿真的方法。本书的内容组织如下：

第 1 章对 MATLAB 产品体系以及 Simulink 的应用进行简要的回顾，是学习 Stateflow 动态逻辑系统建模和仿真的基础。

第 2 章介绍 Stateflow 图形对象的创建方法，详细介绍状态、状态转移、连接节点等常用图形对象的使用方法。

第 3 章介绍 Stateflow 非图形对象——数据对象、事件等的使用方法以及如何完成简单事件驱动系统的建模和仿真。

第 4 章介绍 Stateflow 创建流程图的方法。

第 5 章介绍 Stateflow 层次化有限状态系统的建模与仿真方法。

第 6 章介绍 Stateflow 层次化框图的并行运行机制。

第 7 章介绍 Stateflow 的各种代码生成目标及 Stateflow Coder 产品的使用方法。

第 8 章介绍 Stateflow 的真值表和 Embedded MATLAB Function 的应用方法。

第 9 章介绍利用 Stateflow API 在 MATLAB 命令行中实现 Stateflow 框图模型的创建方法。

本书的附录中还详细总结了 Stateflow 的语法等内容，供大家在使用相应工具时参考。

本书的前六章内容是掌握 Stateflow 产品的必修内容，其中第 1~3 章是第 4~6 章的基础，而掌握了第 4~6 章的内容就基本掌握了各种动态逻辑系统的建模和仿真方法。从第 7 章开始的后三章内容涉及 Stateflow 的高级话题。建议初学者从第 1 章开始循序渐进地学习到第 6 章，这样就能够基本掌握 Stateflow 产品的应用方法。而后三章的内容可以快速地浏览一下，等需要时再仔细学习。另外，要掌握 Stateflow 并且能灵活地应用，还需要用户对 Simulink 产品的应用有足够的了解，因此，对于没有 Simulink 应用基础的读者，作者建议先学习掌握 Simulink 产品的应用方法。如果读者已经有 Simulink 甚至 Stateflow 的应用基础，则可以直接从本书的第 4 章开始学习。

本书是在作者收集了国内外大量的最新权威资料，总结了自身丰富的动态系统建模与仿真的开发经验和 MATLAB 产品技术支持、培训教学以及应用经验的基础上编写而成的，内容详实、全面、权威，示例丰富，便于读者学习具体工具的使用方法。

本书既可作为需要完成动态逻辑系统建模和仿真的工程人员的参考书，也可作为在校本科生、硕士研究生和博士研究生的教材，还可作为 MATLAB 相应产品培训课程的教材。

本书的编写前前后后历经了三年的时间，期间 MATLAB 软件几次升级，Stateflow 模块的特性也多次发生变化，而作者本人也经历了不少事情以及意外。在本书的编写过程中，得到了西安电子科技大学出版社毛红兵编辑的大力支持，在这里对她表示衷心的感谢。还要感谢北京九州恒润科技有限公司以及北京赛四达科技有限公司的全体员工，特别是这两家公司的工程技术人员。和他们在一起共同学习、钻研 MATLAB 软件以及研究实时仿真、系统仿真应用的日子让我终生难忘。更要感谢父母、兄长以及我的妻子余志鸿，我花费了太多的时间在计算机前而没能很好地尽到自己应尽的义务和责任，如果没有父母、兄长多年来对我的培养和教育，没有家人对我的关心、支持和鼓励，也就没有了今天这本书的出版。

Stateflow 涉及的内容比较丰富，在这样一本 300 多页的书籍中不可能将每个技术细节都介绍到。不过，希望此书能够发挥引路者的作用，带领大家进入 Stateflow 逻辑系统建模的大门。需要提醒读者的是，学习 Stateflow 的基础是 Simulink，如果读者对 Simulink 不甚

了解，可以阅读作者编写的《**Simulink** 动态系统建模与仿真基础》一书。

由于时间仓促，书中难免存在一些不妥之处，诚望广大读者谅解，并且提出宝贵的意见和建议，以便我们在再版时改进。

本书没有为读者提供任何示例源文件，希望读者在使用本书学习 **Stateflow** 的过程中能够边学习边动手，在实践的过程中掌握 **Stateflow** 产品的应用方法。如果对本书的内容有任何疑问或者想法，可以通过 E-mail 与作者直接联系：way.buaa@gmail.com，或者登录作者的博客 matlabworld.tianyablog.com。

作者
2007年8月

第 1 章 概 述

MATLAB 产品是用来解决工程与科学实际问题的应用软件，该产品包含了很多产品模块和工具箱。本书介绍的 Stateflow 是其产品体系中非常重要的产品之一。不过，在正式介绍 Stateflow 产品之前，作为本书的基础，首先简要回顾一下 MATLAB 产品的概况以及 Simulink 产品的使用方法。如果读者对 MATLAB 产品体系和 Simulink 模块的使用方法比较熟悉，则可以快速浏览本章的内容，甚至可以跳过本章，学习后面的内容。

本章的主要内容包括：

- MATLAB 产品简介。
- Simulink 回顾。
- Stateflow 概述。
- 安装配置 Stateflow。

1.1 MATLAB 产品简介

MATLAB 的名称源自 Matrix Laboratory，它的首创者是在数值线性代数领域颇有影响的 Cleve Moler 博士，他也是生产经营 MATLAB 产品的美国 MathWorks 公司的创始人之一。MATLAB 是一种科学计算软件，专门以矩阵的形式处理数据。MATLAB 将高性能的数值计算和强大的数据可视化功能集成在一起，提供了大量的内置函数，因而被广泛地应用于科学计算、控制系统、信息处理等领域的分析、仿真和设计工作中；而且利用 MATLAB 产品的开放式结构，可以非常容易地对 MATLAB 的功能进行扩充，从而在不断深化对问题的认识的同时，不断完善 MATLAB 产品以提高产品自身的竞争能力。

目前，MATLAB 产品族可以完成以下功能：

- 数值分析。
- 数值和符号计算。
- 工程与科学绘图。
- 控制系统的设计与仿真。
- 数字图像处理。
- 数字信号处理。
- 通信系统设计与仿真。
- 财务与金融工程。

编写本书时，作者使用的 MATLAB 版本为 MATLAB 7.3，MathWorks 公司将其称为 MATLAB Release 2006b。

提示:

对于 MATLAB 的版本,国内习惯以 MATLAB 产品体系中核心模块——MATLAB 模块的版本号作为整个产品体系的版本号。例如有的读者可能使用的 MATLAB 核心模块版本是 MATLAB 7.2,而 MathWorks 公司对 MATLAB 产品使用的是以产品发布次数计数的版本号,对应 MATLAB 7.2 的是 MATLAB Release 2006a,是 MATLAB 产品体系第 15 次正式发布版。每个 MATLAB 核心模块都对应一个完整的版本号,这里将常见的 MATLAB 版本对照总结如下:

MATLAB 5.3	MATLAB Release 11
MATLAB 6.0	MATLAB Release 12
MATLAB 6.1	MATLAB Release 12.1
MATLAB 6.5	MATLAB Release 13
MATLAB 6.5.1	MATLAB Release 13 Service Pack 1
MATLAB 6.5.2	MATLAB Release 13 Service Pack 2
MATLAB 7.0	MATLAB Release 14
MATLAB 7.0.1	MATLAB Release 14 Service Pack 1
MATLAB 7.0.4	MATLAB Release 14 Service Pack 2
MATLAB 7.1	MATLAB Release 14 Service Pack 3
MATLAB 7.2	MATLAB Release 2006a
MATLAB 7.3	MATLAB Release 2006b
MATLAB 7.5	MATLAB Release 2007a

根据 MathWorks 公司 2006 年初发表的声明,从 2006 年开始 MATLAB 每年将进行两次产品发布,以发布的年份作为版本号,3 月份发布的版本为 a 版本,9 月份发布的版本为 b 版本,因此 2006 年 3 月份该公司发布了 MATLAB Release 2006a,而 2006 年 9 月份该公司发布了 MATLAB Release 2006b。

请读者核对自己所使用的 MATLAB 产品版本,不同版本的 MATLAB 产品有诸多特性上的差别。如果需要了解特性上的差别,请读者自行查看相应版本的 Release Notes 信息。

MATLAB 产品由若干个模块组成,不同的模块可完成不同的功能。这些模块有:

- = MATLAB
- = MATLAB Toolboxes
- = MATLAB Compiler
- = Simulink
- = Simulink Blocksets
- = Real-Time Workshop (RTW)
- = Stateflow

这些产品大体上可以分为以 MATLAB 为基础的产品和以 Simulink 为基础的产品两大分支,由这些模块构成的 MATLAB 产品体系如图 1-1 所示。

其中, MATLAB 模块是 MATLAB 产品家族的基础,它提供了基本的数学算法,例如矩阵运算、数值分析算法等。MATLAB 集成了 2D 和 3D 图形功能,可以完成相应数值可视化的工作,并且提供了一种交互式的高级编程语言——M 语言。用户利用 M 语言可以通过编写脚本或者函数文件来实现自己的算法。

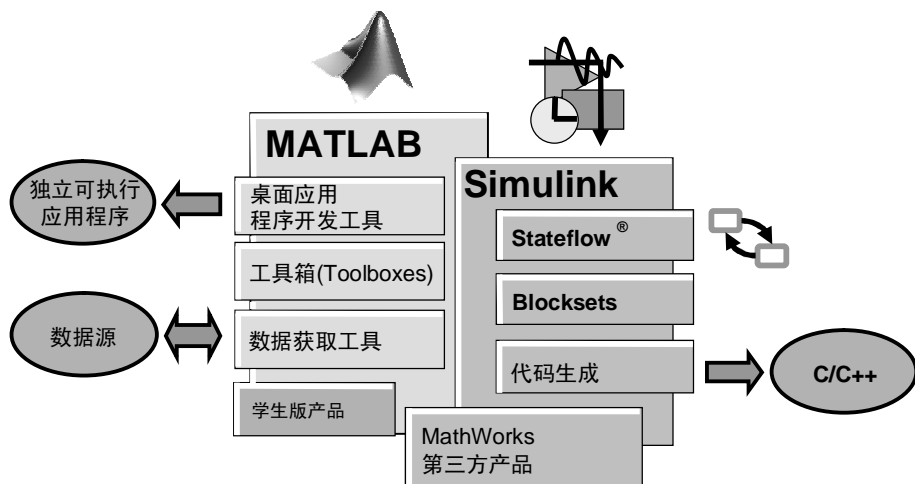


图 1-1 MATLAB 产品体系

MATLAB 的桌面应用程序开发工具是以 MATLAB Compiler 为核心的一组编译工具。MATLAB Compiler 能够将那些利用 MATLAB 提供的编程语言——M 语言编写的函数文件编译生成函数库或者可执行文件。这样就可以扩展 MATLAB 功能，使 MATLAB 能够同其他高级编程语言(例如 C/C++ 语言)进行混合应用，取长补短，以提高程序的运行效率，丰富程序开发的手段。从 MATLAB Release 14 开始，MATLAB Compiler 4 能够支持所有 M 语言特性，可以将大多数工具箱函数都编译生成成为独立可执行应用程序，这极大地提高了 MATLAB 的应用范围。

另外，MATLAB 除了能够和 C/C++ 语言集成开发以外，还提供了和 Java 语言接口的能力，并且还支持 COM 标准，能够和任何支持 COM 标准的软件协同工作。特别是从 Release 13 开始，包含了 MATLAB Compiler 的扩展产品——MATLAB Builder 能够将 MATLAB 的函数文件打包成 COM 组件或者 .NET 组件、Excel 插件，甚至可以打包成 Java 对象，这样就能够将 MATLAB 应用程序算法集成到相应的开发工具或者应用软件中。

MathWorks 公司及其他厂商利用 M 语言还开发了相应的 MATLAB 专业工具箱函数供用户直接使用。这些工具箱应用的算法是开放的、可扩展的，用户不仅可以查看其中的算法，还可以针对一些算法进行修改，甚至开发自己的算法来扩充工具箱的功能。目前，MATLAB 产品的工具箱有四十多种，分别涵盖了数据采集、科学计算、控制系统设计与分析、数字信号处理、数字图像处理、金融财务分析以及生物遗传工程等专业领域。

MATLAB 的数据采集工具可以将各种数据源提供的信号直接读入 MATLAB 环境，无论工程师使用的是 PC 插卡式的数据采集卡或图像采集卡，还是基于总线协议的高端仪器仪表设备以及各种关系型数据库，MATLAB 都提供了与其进行数据 I/O 的交互工具、函数等。利用 MATLAB 开放的集成环境还能够引入更加丰富的数据 I/O 能力。

MATLAB 的主要工具箱和产品模块包括：

- 数学与数据分析：
 - Optimization
 - Statistics

- Neural Network
- Symbolic Math
- Partial Differential Equation
- Mapping
- Spline
- Curve Fitting
- Bioinformatics
- Genetic Algorithm and Direct Search
- 数据采集与测量测试:
 - Data Acquisition
 - Image Acquisition
 - Instrument Control
 - Database
 - OPC Toolbox
 - Excel Link
- 信号处理与图像处理:
 - Signal Processing
 - Image Processing
 - Communication
 - System Identification
 - Wavelet
 - Filter Design
 - Filter Design HDL Coder
 - MATLAB Link for Code Composer Studio
 - Link for ModelSim
- 控制系统设计与分析:
 - Control system
 - Fuzzy Logic
 - Robust Control
 - Model Predictive Control
- 财经与金融:
 - Financial
 - Financial Time Series
 - GARCH
 - Datafeed
 - Financial Derivatives
 - Fixed Income
- 应用程序集成与发布:
 - MATLAB Compiler

- MATLAB Report Generator
- MATLAB Web Server
- MATLAB Builder for .NET
- MATLAB Builder for Excel
- MATLAB Builder for Java

这些产品一般作为整个 MATLAB 产品的基础，它们的共同特点是可通过 M 语言编程或者命令行窗体指令完成具体的功能，即需要一定的代码工作才能够完成算法的开发与实现。在所有 MATLAB 产品基础之上就形成了 Simulink 产品，Simulink 产品的特点是使用拖放方式开发，较少需要编写代码。

Simulink 也是两个单词的缩写——Simulation 和 Link，它是一个框图化的建模环境，能够针对各种复杂的动态系统建立相应的系统仿真模型。Simulink 利用其内建的数学求解器，能够针对任何一种使用数学手段建立的系统进行建模和仿真。目前该产品被广泛应用于控制系统建模与仿真、数字信号处理系统的建模与仿真等领域。

Simulink 的特点：

■ 交互式建模。Simulink 本身提供了大量的功能块以方便用户快速建立动态系统的模型。建模的过程只需要利用鼠标拖放功能块并将其连接起来即可完成。Simulink 的基本模块库如图 1-2 所示。

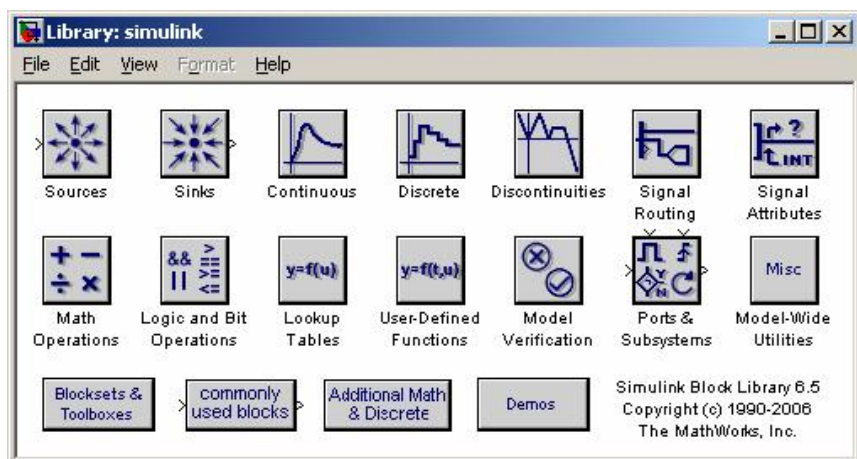


图 1-2 Simulink 的基本模块库

■ 交互式仿真。Simulink 的框图提供可交互的仿真环境，可以将仿真结果动态显示出来，并且可在各种仿真过程中调节系统的参数。

■ 任意扩充和定制功能。Simulink 的开放式结构允许用户扩充仿真环境的功能，可以将用户利用 C、C++、FORTRAN 语言编写的算法集成到 Simulink 框图中。

■ 与 MATLAB 工具集成。Simulink 的基础是 MATLAB，那么在 Simulink 框图中就可以直接利用 MATLAB 的数学、图形和编辑功能，完成诸如数据分析、过程自动化分析、优化参数等工作。

为了丰富该产品在不同领域内的应用，MathWorks 公司还专门开发了不同的功能模块(如图 1-3 所示)，用于特殊领域的应用。这些模块包括：

- = Signal Processing Blockset
- = Simulink Fixed-Point
- = Simulink Control Design
- = Simulink Parameter Estimation
- = Simulink Response Optimization
- = SimPowerSystems
- = Communications Blockset
- = CDMA Reference Blockset
- = SimMechanics
- = SimDriveline
- = SimHydraulic
- = Aerospace Blockset
- = Video and Image Processing Blockset

∧

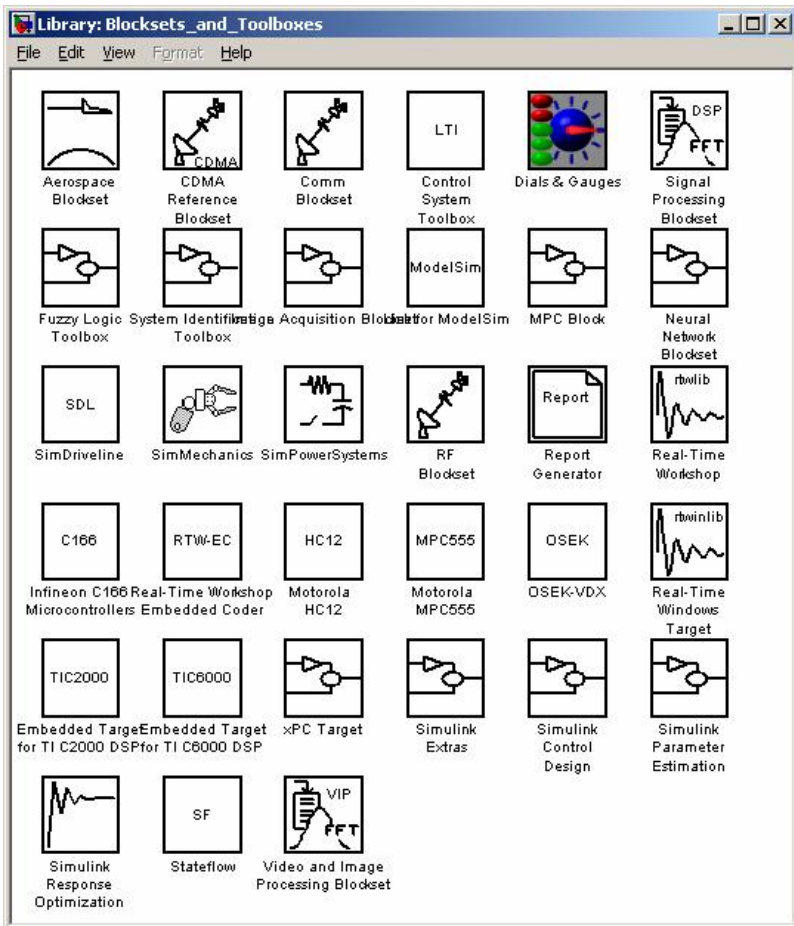


图 1-3 Simulink 的特殊模块库

在 MATLAB 产品体系中, Simulink 的代码生成工具是指 Real-Time Workshop 和 Real-Time Workshop Embedded Coder, 以及在其基础上开发出来的各种实时目标。Real-Time Workshop(RTW)是用来将 Simulink 框图化的模型转变成标准 C/C++语言的工具, 利用该工具转变出来的 C/C++语言代码, 可以进行实时仿真、产品快速原型开发等。而利用 RTW 的扩展——Real-Time Workshop Embedded Coder 可以直接生成用于产品的嵌入式代码。并且 MathWorks 公司也提供了相应的快速原型目标来辅助不同的应用开发。目前 MATLAB 产品中包含的系统目标有:

- = Embedded Target for Motorola MPC555
- = Embedded Target for Infineon C166
- = Embedded Target for Motorola HC12
- = Embedded Target for OSEK/VDX
- = Real-Time Windows Target
- = xPC Target
- = Embedded Target for TI C6000 DSP
- = Embedded Target for TI C2000 DSP

利用 MATLAB 产品的开放性体系, 在全世界范围内共有 350 多家第三方厂商、公司在 MATLAB 的基础上开发了第三方产品。其中, 有些厂商利用 M 语言开发了面向某一特定专业领域的工具箱, 例如美国 Princeton Satellite System 公司开发的面向航天器控制系统设计、分析应用的 Spacecraft Control Toolbox 和 Aircraft Control Toolbox; 而有些厂商则在 Simulink 的基础上开发了面向某一特定应用领域的 Blocksets, 例如英国 RadioScape 公司开发的面向 WCDMA 标准系统仿真应用的 RadioLap 3G 产品。此外还有很多公司在 Real-Time Workshop 的基础上开发了实时系统目标, 其中包括德国 dSPACE 公司开发的 dSPACE 系统、加拿大 Opal-RT Technologies 公司开发的 RT-LAB 产品、英国 PI Technology 公司开发的 OpenECU 等产品。这些第三方产品极大地丰富了 MATLAB 产品在某一特定领域内的应用能力, 让 MATLAB 逐渐成为了众多工程师、研发团体的首选科研软件平台。

在 Simulink 产品的基础之上就是本书即将介绍的产品 Stateflow, 该产品是基于有限状态机理论, 针对事件驱动模型进行建模和仿真的图形化环境。Stateflow 相当于 Simulink 中特殊的产品模块, 它运行的基础是 Simulink 环境。一般地, 可以使用 Stateflow 创建各种复杂的动态逻辑系统, 当系统需要针对大量的时序逻辑或者事件逻辑进行建模时, Stateflow 是良好的选择。

使用 Stateflow 进行建模和仿真需要安装如下产品:

- MATLAB。
- Simulink。
- 第三方标准 C 编译器, 例如 Visual Studio 或者 Borland C/C++。

利用 Real-Time Workshop 和 Stateflow Coder 可以将包含了 Stateflow 有限状态机的 Simulink 模型生成为标准 C 代码, 用于系统的实时仿真或者产品级嵌入式代码。

编写本书时, 作者使用的 Stateflow 产品的版本为 6.5(R2006b), Stateflow Coder 产品的版本也为 6.5(R2006b)。

1.2 基于模型的设计思想

近年来，MATLAB 软件产品族中以 Simulink 为基础的系列产品得到了迅速的发展，特别是 MATLAB 的自动代码生成技术以及围绕着代码生成的建模技术。那么究竟是什么使世界各地的工程师如此关注 MATLAB 以及 Simulink 建模、仿真和自动代码生成技术呢？归根结底，就是因为基于模型的设计思想(Model Based Design)正在被大家广泛地接受。所谓基于模型的设计，就是在系统的设计过程之中，所有信息传递、工作的核心与基础都是若干模型，所有工程师都利用统一的模型完成自己关注的开发任务。那么相对于传统的设计手段，基于模型的设计有什么样的好处呢？在下面的内容中将给出答案。

1.2.1 系统设计的基本过程

若是想了解基于模型的设计方法有什么样的优势，就需要看看当前系统设计的基本过程，对比一下传统的设计手段和基于模型的设计手段的不同。通常情况下，无论工程师所面对的系统有多么简单或者多么复杂，其设计方法都可以分为以下七个步骤。

1. 需求分析

在需求分析阶段，工程师通常需要根据客户的需要定义自己需要完成的工作，以及工作结果需要满足的某种特性要求。例如对于军用作战飞机，需要具有若干公里的作战半径，需要有若干吨的起飞重量，能够搭载若干类型的作战武器，这些都是军方向武器制造商提出的需求。武器制造商就需要根据军方的要求分解需求，分析出为了满足这些需求分别需要完成哪些工作，工作的步骤以及不同工作阶段的产出与验收手段等。

2. 系统规范

需求分析结束之后，工程师基本明确了工作阶段以及不同阶段具体的产出。那么此时需要完成的工作就是根据任务的要求详细定义工作的步骤以及解决问题的方法，同时制定给不同分系统设计实现人员的任务书。定义正确的设计规范是进行分系统设计的重要前提，只有规范是正确的，才能够保证设计的结果是可靠的。

3. 子系统设计

当工程师拿到设计规范以及任务书之后，就需要根据自己掌握的知识逐步完成子系统的设计。例如，对于控制系统设计工程师，需要建立被控对象的模型，然后分析对象特性，再根据任务书或者规范要求设计具体的控制算法以满足需求所要求的技术指标；而对于数字信号处理算法工程师，则需要根据处理信号的特点以及需求的要求，设计不同的滤波器算法，以满足需求提出的技术指标。不同专业的工程师解决自己的问题时，都会选择自己熟悉可靠的设计工具，快速完成算法设计。

4. 子系统实现

算法设计完毕之后最终需要实现到某些嵌入式系统中。无论是飞机的飞控系统、航电系统或者作战飞机的火控系统等，都包含了丰富的嵌入式软/硬件设备。工程师需要根据系统的特点选择合适的硬件设备，将算法具体实现在相应的嵌入式系统中。此时往往需要进

行大量的编程工作以及硬件设计实现工作。

5. 子系统集成与测试

子系统实现出来之后需要一步一步地进行集成。例如飞机起落架的电子控制单元首先需要与被控的起落架进行集成测试，而飞机飞控计算机也需要首先与飞行操纵系统进行相应的集成检测。

6. 分系统集成与测试

当子系统完成集成之后，需要进行分系统集成与测试。例如对于飞机的飞控系统，需要把不同的子系统——惯性导航设备、飞行控制计算机、舵系统等集成起来，构成完整的飞行控制系统以进行集成与测试，检查各个子系统是否能够正确地集成工作。

7. 全系统集成与测试

整个系统全部设计完毕之后就需要进行全系统的集成与测试，把所有分系统放在一起，构成最后产品的原理样机或者原型机，进行各种任务的测试工作，检查产品是否能够满足设计任务书的要求，是否满足了设计需求。

1.2.2 传统设计手段的缺陷

上述的设计过程与实际的设计手段是毫不相干的。传统的开发手段是自顶向下、逐步细化的“瀑布式”开发手段，可以用图 1-4 来表示。

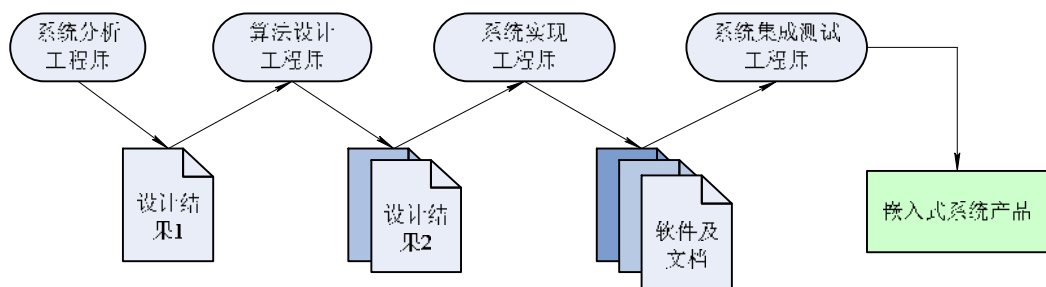


图 1-4 传统设计手段

传统的设计过程中，不同类型的工程师彼此交换自己的设计成果，逐步细化设计任务，直到完成最后的产品。这种自顶向下的“瀑布式”开发流程具有快速便捷、易于实现的特点。但是，因为嵌入式系统本身复杂程度的增加以及设计研发周期缩短和市场的压力等诸多原因，传统的设计开发手段逐渐暴露出下列问题。

1. 信息的交流依赖文档

传统的开发手段中，在不同阶段彼此之间传递的信息需要依赖文档，例如需求分析报告、系统详细设计规范、设计任务书、设计报告等。由于工程师对系统的认识与理解有一定的局限，因此在系统开发的过程中会引入一些缺陷甚至错误，而这些缺陷或者错误会一步一步地传递到下一个设计阶段。例如需求分析的不准确就会影响规范设计和任务书的编写，那么工程师依据不正确的任务书和规范设计实现的产品就包含了很多错误。此外，由于工程人员总会存在针对文字理解的二义性问题，因此即便是文档本身没有错误，可能也会由于工程人员理解上的误差而引起系统设计、实现的错误。

2. 早期设计阶段引入的错误要在开发后期才能发现

由于设计的嵌入式软件算法需要有专门的硬件生产出来之后才能够进行集成的测试，因此在设计阶段早期，例如在规范设计阶段，引入的一些设计错误要在软/硬件产品都具备之后才能够进行集成与测试。但是在开发后期发现的错误修正起来又不是那么简单容易，极有可能需要重新编写软件，甚至重新开发硬件才能够完成错误的修正。

3. 手写代码与手写文档

在产品的实现过程中，大多数控制系统、信号处理、图像处理系统最终都会实现到某种单片机或者 DSP 处理器上，成为嵌入式系统。传统的设计手段中，这一过程只能利用手写代码来完成。可能每个读者都有编写、调试手写 C/C++、汇编代码的痛苦经历。特别在完整的系统设计过程中，传递给产品实现工程师(通常为嵌入式软件工程师)的文档极有可能就已经包含了错误，再加上手写代码的错误，最终的产品就会包含很多问题，以致于无法完成最后的子系统集成、分系统集成和全系统集成工作。

特别是，当出现了需求更改的情况时，由于传统的设计手段里面存在着大量的手工代码工作，以致于细微的修改都需要子系统实现工程师重新修改并编译大量的代码，而在修改的过程中，极有可能再次引入新的错误，这样就成为了一个恶性循环，直到最后都无法满足产品需求，致使设计进度拖后，甚至整个项目被彻底取消。

下面的图形来自 Software Development Times 2002 年 12 月份的一个统计报告，他们统计了北美地区部分嵌入式系统设计项目的完成情况，得到的结果如图 1-5 所示。

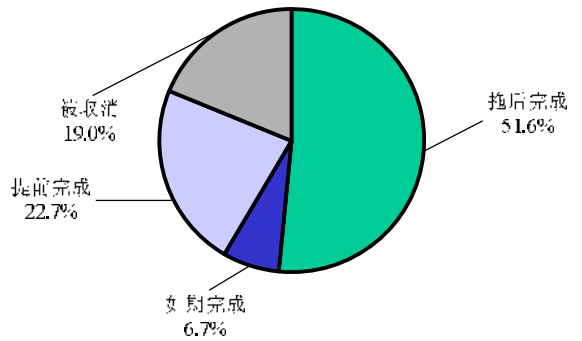


图 1-5 嵌入式系统项目的完成情况

从统计数字上可以看到，有一半以上的项目由于种种原因被迫推迟完成，而只有 6.7% 的项目如期完成了。具体原因统计如图 1-6 所示。

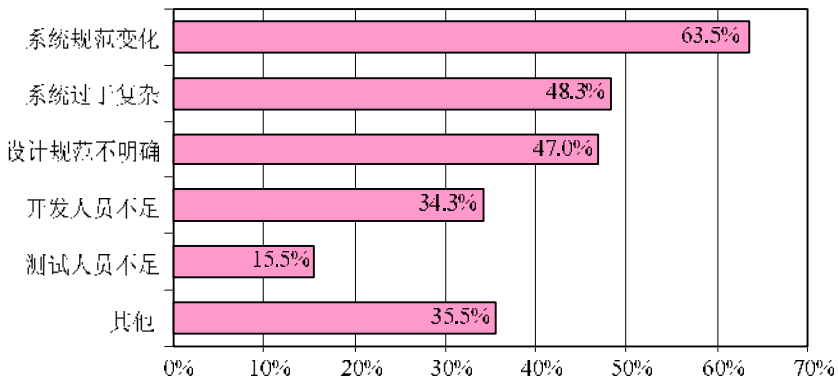


图 1-6 嵌入式系统项目延期完成的原因统计

从上面的统计数字可以看到，嵌入式系统项目延期的主要原因是由于设计规范造成的，系统设计规范的不明确以及规范在设计过程中发生的变化都为整个项目的进程造成了非常大的影响。归根结底的原因就是，大量存在手工工作和文档，不能在早期利用某种手段考察各个不同组件的细微修改对整体的影响，只有在最后阶段才能够进行全面的测试。一旦发现问题，不得不重新返工，浪费了很多时间和精力，从而增加了系统面市的压力，带来了巨大的成本浪费。

为了解决这些问题，近两年来，越来越多的工程师都在逐步采用基于模型的设计方法，以提高他们的设计效率。

1.2.3 基于模型的设计优势

如前所述，基于模型的系统设计就是在系统的设计过程之中，所有信息传递以及设计工作的核心与基础都是若干模型，所有工程师都利用统一的模型完成自己关注的开发任务。图 1-7 所示为基于模型的设计过程。

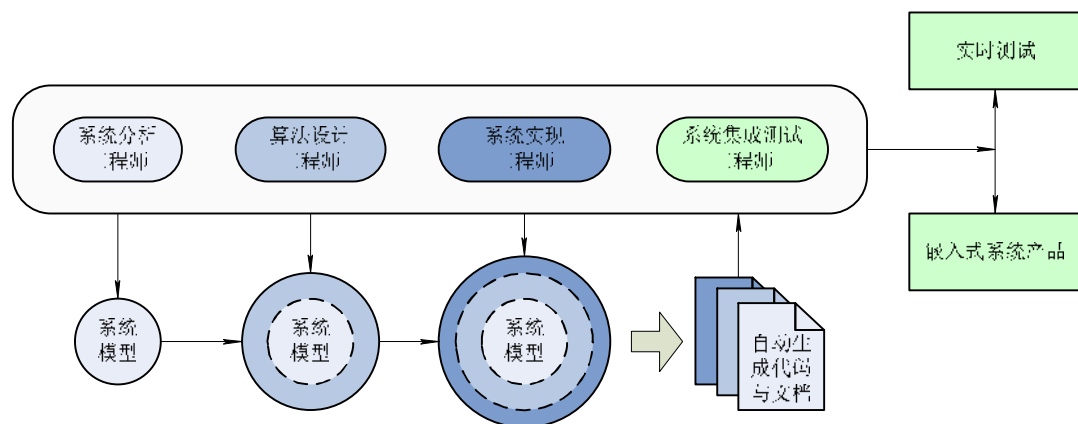


图 1-7 基于模型的设计过程

在基于模型的系统设计过程中，执行不同设计任务的工程师彼此在统一的环境下完成开发工作，在他们之间传递的不仅仅是设计文档，还包含系统模型。而随着工作不断的细化，越到开发后期，模型就越来越复杂，每个工程师都要把自己的设计工作增加到相应的模型中。当需要具体实现产品或者进行集成测试时，只要通过自动代码生成就可以迅速地完成代码生成、硬件实时仿真与测试，甚至还可以生成直接应用于产品的嵌入式产品级代码，避免了大量的手写代码工作。

采用基于模型的系统设计主要解决了传统设计手段中手写规范文档、测试工作在后期阶段的主要问题，它具有如下的优势：

(1) 模型就是可执行规范。在基于模型的系统设计过程中，所有过程遵循统一环境下的统一的模型。由于模型能够进行仿真计算，因而在设计初期就可以针对设计思想进行仿真验证，这相当于把测试手段从系统设计的初期就引入到了整个设计流程中。作为规范的模型能够通过执行仿真来验证自己的正确性，从而保证了规范的正确性。

(2) 连续不断的测试与验证。在整个基于模型的系统设计过程中，从设计初期就进行离

线的算法仿真测试，在产品实现之前通常还需要完成快速原型仿真、软件在回路仿真、处理器在回路仿真等步骤，而产品实现之后需要完成硬件在回路仿真、人在回路仿真等集成测试手段。这些测试都基于前面开发出来的系统模型，即在每个环节都引入了测试仿真手段。利用充分的仿真，可以考察系统不同组件对整个系统的影响。

(3) 自动代码生成。自动代码生成技术的逐步成熟是基于模型的系统设计被广大工程师逐步接受的重要突破。利用自动代码生成技术，工程师摆脱了繁琐的代码编写与调试工作。自动生成代码的源头为前面开发过程中一步一步经过测试验证过的模型，只要模型是正确的，则生成的代码也基本是正确的。自动代码生成技术使快速控制原型仿真和硬件在回路仿真成为了简便易行的测试手段。而嵌入式产品级代码生成技术则能够帮助工程师节约大约 70% 以上的工作时间，极大地提高了工作效率。一旦在测试过程中发现了模型算法还需要修改，只需直接回到模型中修改模型，然后在仿真之后生成产品代码即可。

关键概念：RCP(Rapid Control Prototyping)——快速控制原型

快速控制原型仿真是实时仿真的一种，它实现于产品研发的算法设计阶段与具体实现阶段之间。快速控制原型就是利用某种手段将工程师开发的算法下载到某个计算机硬件平台中，该计算机硬件平台在实时条件下运行，模拟控制器，通过实际 I/O 设备与被控对象实物连接，验证算法的可靠性和准确度。要实现快速控制原型，必须有集成良好、便于使用的建模、设计、离线仿真、实时开发及测试工具。用户选择的实时系统允许反复修改模型设计，进行离线及实时仿真。这样，就可以将错误及不当之处消除于设计初期，使设计修改费用减至最小。

使用 RCP 技术，可以在费用和性能之间进行折中；在最终的产品硬件投产之前，可以仔细研究诸如离散化及采样频率等的影响、算法的性能等问题；通过将快速原型硬件系统与所要控制的实际设备相连，可以反复研究使用不同传感器及驱动机构时系统的性能特征。

关键概念：HILS(Hardware-in-the-Loop Simulation)——硬件在回路半实物仿真

当新型控制系统设计结束并已制成产品控制器，需要在闭环下对其进行详细测试时，往往由于种种原因，如极限测试、失效测试或在真实环境中测试费用较昂贵等，使测试难以进行。在积雪覆盖的路面上进行汽车防抱死装置(ABS)控制器的小摩擦测试就只能在冬季冰雪天气进行；有时为了缩短开发周期，甚至希望在控制器运行环境不存在的情况下(如控制对象与控制器并行开发时)对其进行测试。于是就需要利用某种计算机硬件平台在实验室中实时解算模拟对象在实际工作条件下的运动过程，并且通过相应的 I/O 设备将信号提供给控制器。此时可以通过修改控制对象参数来模拟被控对象的各种工作状态，达到全面考察验证控制器开发质量以及控制算法可靠程度的目的。

在进行基于模型的设计过程中，建模成为了关键，只有正确的模型才能够成为正确的设计规范，才能得到正确的代码，并且需要选择合适的建模工具。Simulink 能够创建各种动态系统模型，无论是进行控制系统设计、数字信号处理系统设计还是进行视频处理系统设计，都可以利用 Simulink 创建系统模型，完成算法的设计。结合了 Stateflow 之后，整个 MATLAB 产品族就可以针对连续时序系统、离散时序系统、离散事件系统进行建模与仿真，得到正确的模型之后，利用自动代码生成工具——Real-Time Workshop，迅速完成快速原型仿真、硬件在回路仿真等实时仿真任务，而后在 Real-Time Workshop Embedded Coder 的支