基于模型的设计及其嵌入式实现

刘 杰 编著

北京航空航天大学出版社

基于模型的设计及其嵌入式实现

刘　杰　编著

北京航空航天大学出版社

内 容 简 介

本书以基于模型的设计为主线，讲述了 M 代码和 Embedded MATLAB 代码的快速编写与调试、浮点 Simulink/Stateflow 模型的建立、调试与验证以及用户驱动模块的创建；详细介绍了基于模型设计的全过程，主要包括：需求的验证与跟踪、模型的系统测试与设计验证、浮点到定点模型的转换、模型嵌入式 C 代码的自动生成以及软件/处理器/硬件在环测试。整个过程满足DO-178B航空电子规范，可显著提高工作效率、降低开发成本，并且增加了代码的安全性与鲁棒性，避免了产品开发的潜在市场风险。

本书可作为汽车电子、航天军工、通信与电子信息、电力等领域的工程师从事嵌入式系统开发的技术手册，也可作为高校电子类专业嵌入式系统开发与基于模型设计的教材，另外也是一本MATLAB高级建模与模型验证的参考书。

图书在版编目(CIP)数据

基于模型的设计及其嵌入式实现 / 刘杰编著. -- 北京：北京航空航天大学出版社，2010.9

ISBN 978-7-5124-0213-3

Ⅰ. ①基… Ⅱ. ①刘… Ⅲ. ①航空电气设备—设计 Ⅳ. ①V242.2

中国版本图书馆 CIP 数据核字(2010)第 176287 号

基于模型的设计及其嵌入式实现

刘 杰 编著

责任编辑 刘 星

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) http://www.buaapress.com.cn

发行部电话：(010)82317024 传真：(010)82328026

读者信箱：emsbook@gmail.com 邮购电话：(010)82316936

保定市中画美凯印刷有限公司印装 各地书店经销

*

开本：787×1 092 1/16 印张：27.75 字数：710 千字

2010 年 9 月第 1 版 2010 年 9 月第 1 次印刷 印数：4 000 册

ISBN 978-7-5124-0213-3 定价：59.00 元

前　言

在目前的市场上，想找到一款不包含嵌入式控制器件的电子、机电产品已经很难了，含有嵌入式系统的产品已深入到了我们工作、生活的方方面面。同时，人们对于产品的安全性、实时性、可操作性、特定功能等要求也越来越高，这也就大大增加了嵌入式系统的复杂性和开发难度。

为了在日趋激烈的市场竞争中占据有利地位、开发出高质量的产品，采用传统的项目开发方法已很难满足这些需求。

传统项目开发的方法一般分为 4 个步骤：

① 需求分析与技术规范阶段。

一般用纸质文档或电子 Word 文档写成，系统工程师团队以此进行概念和算法研究，评估技术规范的可行性。对于厚厚的技术文档，每个系统工程师对需求和技术规范的理解难免存在偏差。NASA 的研究报告指出："在需求分析阶段产生的错误占整个开发错误的 50% 以上"，这给后期的项目开发带来了诸多隐患。

② 设计阶段。

硬件开发工程师团队根据系统工程师的评估报告，设计制作原型样机（如汽车、航空航天器、电路板等），项目的前期投入巨大、开发周期长，并且不能保证所制作的原型样机一定能满足技术规范要求的技术指标。

③ 实现阶段。

软件开发工程师团队根据需求与技术规范，在原型样机上，手工编写 C 代码或汇编代码，实现技术规范所要求的技术指标。这一阶段需要精通软件的编程人员花费大量时间来编制程序、查错、调试、验证，明显增加了工作量，延长了研制周期。此外，手工编制的代码良莠不齐，降低了软件运行的可靠度，增加了代码错误的可能性，给新产品上市带来风险。

④ 测试与验证阶段。

原型样机制造完成后，才能对产品进行测试与验证，只要上述任何一个过程出现偏差都会导致产品开发的失败，这也是传统项目开发最大的弊病。因此，传统的项目开发方法难免重复多次才能成功，开发风险巨大。

为了解决这些问题，工程师必须找到一种更快速度、更有效率的开发产品的方法，基于模型的设计就是解决该问题的一种选择。它始于 20 世纪 90 年代初的汽车制造和航空航天工业，这些行业需要使用大量的微处理器单元，因此工程师们最先发现了采用建模与仿真的方法来开发嵌入式系统的巨大优势；到了 90 年代中期，控制算法仿真技术的发展催生了自动代码生成技术。模型仿真和自动代码生成技术在这些行业得到的成功应用，使人们清楚地认识到

它在嵌入式系统开发中的经济和高效。基于模型的设计为工程师们提供了一种通用的开发与测试平台，使具有不同工程背景的工程师之间建立起更好的联系，使开发具有高集成度的复杂系统成为可能。

目前国际上流行的基于模型设计的软件主要有 SCADE 和 MATLAB，它们都成功地应用于大型项目的开发上，例如，欧洲的空客 380、美国的 GM 混合动力车、诺-马公司的联合攻击机等项目。MATLAB 已经成为一种近乎完美的高度集成化的开放式开发平台，在科学计算与建模方面处于不可替代的领先地位，加之其拥有国内众多的用户，因此，本书也选用 MATLAB软件为例来讲述基于模型设计的方法。

Mathworks 公司的 Simulink / Stateflow / Embedded MATLAB 等工具使得工程师可以在一个可视化的交互开发测试平台上进行基于模型的设计，工程师还可以利用直观的模块图对系统模型和子系统设计进行可视化处理。

基于模型的设计对应传统项目开发的方法同样分为 4 个步骤：

① 可执行、可跟踪的技术规范。

在基于模型的设计方法中，系统工程师首先要建立一个系统模型，即通过数学模型来精确、无歧义地描述用户的需求，创建一个可执行、可跟踪的技术规范。工程师可以通过这个系统模型，动态地确认系统性能。

相对于用传统的纸质或电子 Word 文档来描述的需求与技术规范，这种方法具有明显的优势，它使得开发团队中的每个成员都能够无歧义地理解并运行该模型，从而可以更加专注于开发主要模型的各个部分，不会因理解的不同而造成需求的丢失、冗余或冲突。

② 生成定点模型。

系统模型与需求之间可建立双向链接，在整个开发过程中，软件工程师可以对模型进行需求追踪和测试，将产品的缺点暴露在产品开发的初期。根据具体的嵌入式器件和实现条件，对系统模型进行细化与功能重分区；此后重新进行系统测试、设计测试和模型助手测试，验证是否满足需求与技术规范，判断是否还存在缺失的需求，验证是否符合特殊的行业标准（如 DO-178B、IEC-61508、MAAB 等）；之后再对模型做定点转换，形成简捷、高效的定点模型。

③ 嵌入式代码的自动生成。

Mathworks 公司的 Real-Time Workshop Embedded Coder 可以将 Simulink / Stateflow 中的模型自动转换为嵌入式 C 代码，大大降低嵌入式系统的开发门槛。开发人员可以在 Simulink / Stateflow、Embedded MATLAB 中建立系统模型、构思解决方案，然后使用 RTW-EC 自动生成优化的、可移植的、自定义的产品级 C 代码，并根据特定的目标配置自动生成嵌入式系统实时应用程序。这样就缩短了开发周期，同时避免了人为引入的错误。

④ 连续的测试和验证。

基于模型的设计在整个设计过程中都在不断地进行测试和验证，工程师利用测试用例追踪系统级模型和需求，检测设计变更导致的系统输出变化，并快速追踪到变更的来源；通过测试用例还能够了解系统模型的功能覆盖度。

对于嵌入式系统，还需测试其实时性，工程师可以使用硬件在环测试检测嵌入式代码的实时性。通过测试，可以收集实时数据，修改代码参数。硬件在环检测能确保在开发早期就完成

嵌入式软件的测试，这样在系统整合时，嵌入式软件测试就可以比传统方法检测得更彻底、更全面，从而可以及早地发现问题，大大降低解决问题的成本。

本书分为3个部分，第1～4章为第1部分，主要介绍了Simulink/Stateflow模型的建立、调试与验证，是基于模型设计的基础。

第1章主要包括：MATLAB部分新功能、基于cell的M-code快速编写、M-Lint实时代码验证器的使用、基于R2010a的Embedded MATLAB编程规范等；

第2章主要包括：Simulink建模与调试、创建模型测试用例、模型的验证与覆盖度分析、基于采样与基于帧的信号分析等；

第3章主要包括：Stateflow建模与调试、简单的应用实例等；

第4章主要包括：编写S-function（C MEX S-function和level-2 M S-funcion）、生成S-Function Builder用户模块、编写Embedded MATLAB模块、利用代码继承工具集成现存C代码到Simulink模型的方法、编写TLC文件等。

第5章为第2部分，介绍了CCS 3.3集成开发环境的使用，讨论了嵌入式代码的快速生成、MATLAB与CCS的交互式开发、传统滤波器设计与基于模型设计的比较等。

第6章为第3部分，包括：传统项目开发方法与基于模型设计的比较；DO-178B航空电子规范的简介，符合DO-178B规范的基于模型设计工作流程；需求与模型间的双向跟踪，模型的系统测试与验证，模型的设计测试，模型覆盖度分析；浮点模型到定点模型的转换；代码的自动生成；生成代码的验证；模型与生成代码的双向跟踪；代码的实时性分析；综合硬件测试等。

本书从策划到完成经历了两年半的时间，得到了两家公司的资助，书中使用的实验设备大多由它们提供，在此表示感谢。撰写过程中，作者阅读了超过数万页的外文资料和技术文档，做了大量的验证实验，有些例子是作者开发实例的总结，可以直接用于生产实践。

书中很多内容由作者及其团队独立完成，不少内容比较新颖，也是首次在国内的公开出版物中出现。由于基于模型的设计涉及知识太多，无法在短短的几百多页图书中得到充分论述，加之时间紧且作者的水平有限，书中的错误或遗漏在所难免，敬请读者批评指正。欢迎读者参加对基于模型的设计技术的讨论，有兴趣的朋友可以发送邮件到：liuyu3594@yahoo.com.cn与本书作者沟通；也可以发送邮件到：emsbook@gmail.com，与本书策划编辑进行交流。

翁公羽、孙瑶瑶、周宇博全程参与了本书的资料整理和撰写工作，郑仁富、杨元廷、罗兵、陈添丁、郑明魁、李涵、胡步发参与了本书的策划和个别小节的编写工作，张华君、刘大茂、郭里婷、李恭伟、张玮、陈智宾、陈声登、李天建对部分程序进行了调试和修改，刘高阳、万方、孙昕、郑红武、陈乐武、伍悦参与了部分画图工作，在此一并表示感谢。

刘 杰

2010年5月于怡园

目 录

第1章

MATLAB基础

MATLAB是一种用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境。对比传统的编程语言，MATLAB可以更快地解决技术计算问题；广泛地应用于信号和图像处理、通信、控制系统设计、测试和测量、财务建模和分析以及计算生物学等众多领域；附加的工具箱(MATLAB函数集)扩展了MATLAB环境，可用于解决这些应用领域内特定类型的问题。

更重要的是，使用MATLAB编程或开发算法的速度将大大提高，这得益于MATLAB无须执行诸如声明变量、指定数据类型以及分配内存等低级管理任务，例如很多情况下，用户无须使用for循环。因此，一行MATLAB代码经常等效于几行C/C++代码。

同时，MATLAB还提供了传统编程语言的所有功能，包括算法运算符、流控制、数据结构、数据类型、面向对象编程(OOP)以及调试功能。利用MATLAB，无须执行编译和链接即可一次执行一个或一组命令，这样就可以迅速迭代到最佳解决方案。

MATLAB的主要功能如下：

- 用于技术计算；
- 对代码、文件和数据进行管理；
- 交互式工具可以按迭代的方式探查、设计及求解问题；
- 数学函数可用于线性代数、统计、傅立叶分析、筛选、优化以及数值积分等；
- 二维和三维图形函数可用于可视化数据；
- 各种工具可用于构建自定义的图形用户界面；
- 各种函数可将基于MATLAB的算法与外部应用程序和语言(如C、C++、Fortran、Java、COM以及Microsoft Excel)集成。

Mathworks公司自2006年起，每年发布两次软件更新，以R200x a/R200x b命名。

- R2008b新增了函数浏览器，用户在编辑器或命令窗口输入时，可得到相关的函数提示；用户输入函数时，提供自变量提醒。
- R2009a扩展了M-Lint代码检查器功能，对于MATLAB编辑器内部的警告和错误可以提供更为详细的解释说明；对FFT等MATLAB函数提供了多核支持。
- R2009b增强了帮助浏览器的功能，可按产品和结果类型显示详细的搜索结果，支持从MATLAB直接访问MATLAB Central文件交换。

- R2010a 新增了 zip 文件压缩与解压功能，为 50 多个函数提供了多核支持并增强其性能，扩展了 Image Processing Toolbox 中对大型图像的支持。

根据本书的定位，本章介绍 MATLAB 开发环境新增或增强的功能，着重说明 M 文件的结构，如何使用 M-Lint 和 cell 加快 M 文件的编写与调试，并利用代码效率分析器优化代码，最后介绍 Embedded MATLAB。

本章主要内容包括：

- MATLAB 开发环境新功能；
- M 文件结构；
- 加快 M 文件的编写——M-Lint；
- 加快 M 文件的调试——cell；
- 代码效率分析；
- Embedded MATLAB。

1.1 MATLAB 开发环境新功能

1.1.1 函数浏览器

① 单击对话框左侧的 f_x 图标，函数浏览器显示各种常规函数、工具箱及模块集，逐次打开目录，可得到对应的函数，单击某函数或将鼠标停留在该函数，函数浏览器将显示函数的详细描述，单击右上角的 More Help，可链接到帮助文档，如图 1.1.1 所示。

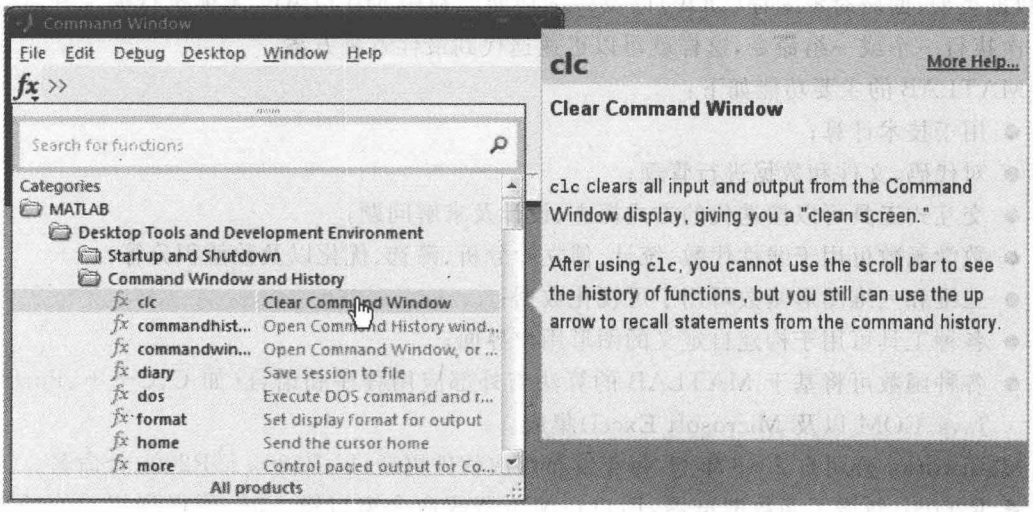

图 1.1.1 函数浏览器显示方式 1

② 用户可以在函数浏览器上方的文本框内输入关键词，系统将自动显示与该关键词有关的所有函数，该关键词可以是部分或完整的函数名，也可以是函数的功能描述。

输入 convolutional，函数浏览器列出了各种与卷积相关的函数，而 convolutional 本身并

不是一个函数名，如图 1.1.2 所示。

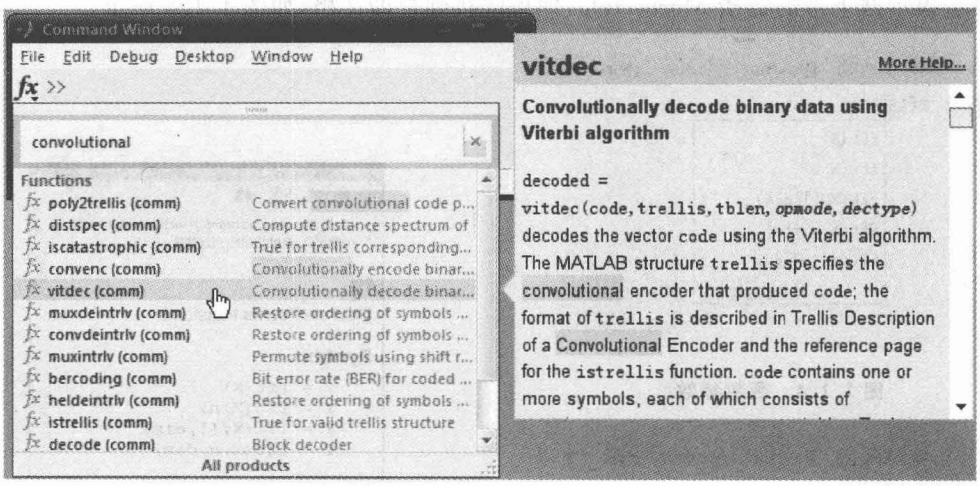

图 1.1.2　函数关键词搜索

③ 用户也可以在命令行窗口输入关键字并选中，单击窗口左侧的 fx 图标，函数浏览器即显示与所选文字有关的所有函数，如图 1.1.3 所示。

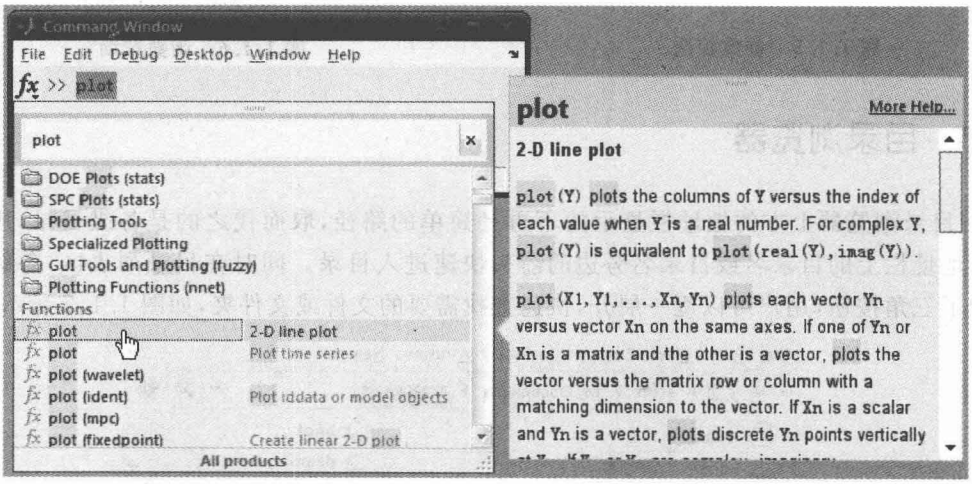

图 1.1.3　函数浏览器显示方式 2

在文件编辑对话框 Editor 的工具栏上也能找到函数浏览器 fx 图标，功能与上述一致。

④ 双击函数，将其添加到当前命令行。

1.1.2　函数提示

用户若事先已明确某一函数的函数名，但不确定该函数的输入参数，则可以在命令行窗口输入完整的函数名，并加上一个左圆括弧，函数浏览器即显示该函数各种可能的表达式格式，如图 1.1.4 所示。

提示框的内容将根据用户后续的输入不断筛选显示，如图 1.1.5 所示。

单击提示框下部的链接 More Help，即可链接到帮助文档，如图 1.1.6 所示。

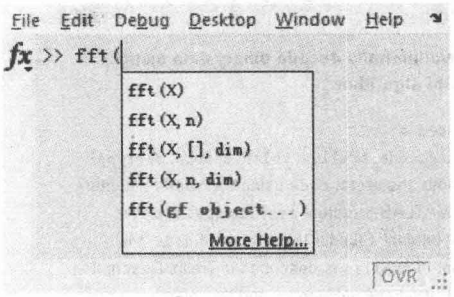

图 1.1.4　函数参数

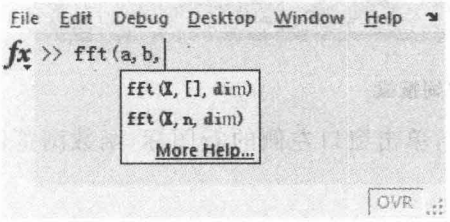

图 1.1.5　参数筛选

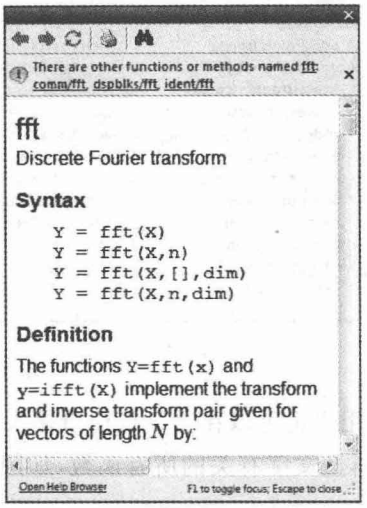

图 1.1.6　函数帮助

1.1.3　目录浏览器

① 目录浏览器上方的地址栏显示的不再是简单的路径，取而代之的是各级目录，用户可以单击地址栏上的目录名或目录名旁边的箭头快速进入目录。同时在目录列表区，各级目录前增加了三角按钮，用户可以逐一展开，快速查找需要的文件或文件夹，如图 1.1.7 所示。

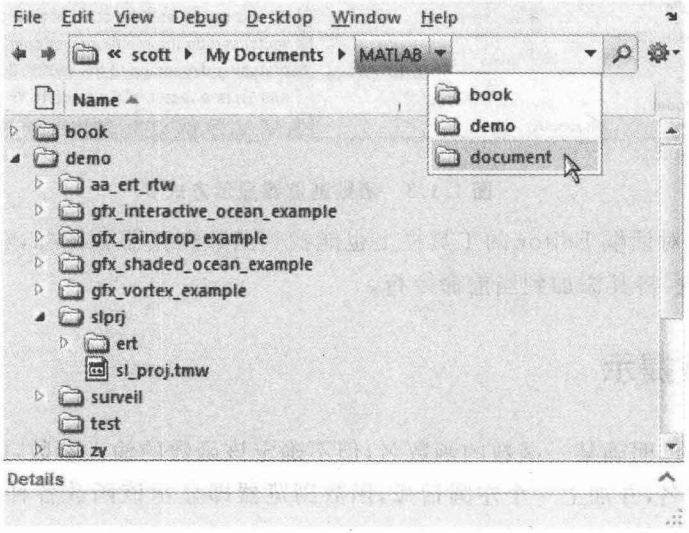

图 1.1.7　地址栏

② 单击地址栏末尾的空白处，则地址栏的内容显示为传统的路径，便于用户直接输入路径，如图 1.1.8 所示。

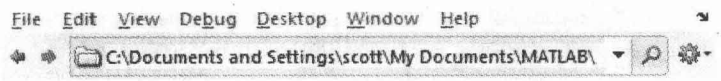

图 1.1.8　直接输入地址

③ 右击目录区上方的□按钮，则显示数据分类菜单，如图 1.1.9 所示，用户可以根据需要启用目录列表栏目或按类型显示。

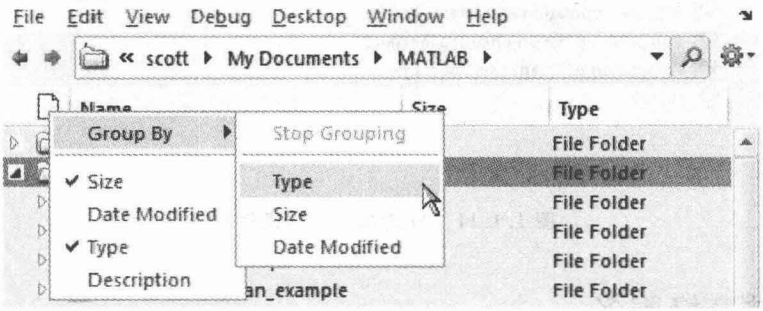

图 1.1.9　文件分类选项

④ 目录浏览器下方有个详细信息区，右侧的箭头用于展开或折叠。选中某个文件，该区域则显示文件的详细信息，用户可以根据这些信息初步判断文件的性质或用途。

对于 M 函数，信息区显示函数名及参数，如图 1.1.10 所示；定义了 cell 的 M 脚本文件，信息区显示脚本的各个 cell，如图 1.1.11 所示。

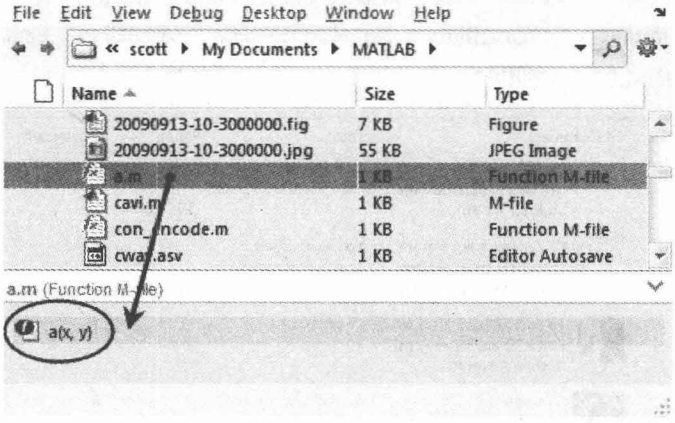

图 1.1.10　M 函数文件详细信息