

一种基于 MATLAB 快速开发跨平台算法软件的方法

周希娃, 张国玉, 李 洋, 胡继军
(北京遥测技术研究所 北京 100076)

摘要: 利用 MATLAB 进行算法的研究、仿真和实现, 已经成为科研工作者的的重要手段之一。在 MATLAB 上开发算法便捷、高效, 但无法直接应用在其他平台, 此时需要软件人员编码转换进行二次开发。如果某个算法同时应用在 Windows 和 Linux 这两个系统中, 软件人员需要开发两个不同的软件, 它们功能相同只是代码不同。这不仅增加了软件开发的工作量, 同时也不利于后期维护。针对上述问题, 提出了一种基于 MATLAB 快速开发跨平台算法软件的方法。首先, 利用 MATLAB 开发算法的便捷性进行算法的调试与验证, 简化接口函数; 其次, 使用 MATLAB Coder 生成独立于 MATLAB 的 C/C++ 代码; 然后, 将生成的代码在不同平台下进行封装, 生成该目标平台的动态链接库; 最后, 不同平台通过调用该平台下的动态链接库, 实现了同一算法的跨平台软件开发。所提出的算法软件跨平台开发方法, 已成功应用于多个实际项目。通过本方法, 不仅缩短了软件的开发周期, 提高了软件编程效率, 减少了软件开发中的人为错误, 同时方便了软件的后期维护。

关键词: 算法软件; 跨平台软件开发; 自动代码生成; MATLAB Coder

中图分类号: TP311 **文献标识码:** A **文章编号:** CN11-1780(2022)05-0068-06

DOI: 10.12347/j.ycyk.20210910001

引用格式: 周希娃, 张国玉, 李洋, 等. 一种基于 MATLAB 快速开发跨平台算法软件的方法[J]. 遥测遥控, 2022, 43(5): 68-73.

A method of rapidly developing cross-platform algorithm software based on MATLAB

ZHOU Xiwa, ZHANG Guoyu, LI Yang, HU Jijun
(Beijing Research Institute of Telemetry, Beijing 100076, China)

Abstract: Using MATLAB for developing, simulating and implementing algorithm software has become one of primary means for researchers. Although the algorithm developed on MATLAB is convenient and efficient, it cannot be directly applied to other platforms, it has to be under secondary development. If one algorithm is to be applied in both Windows and Linux systems, the software engineers need to coder two software, which are same functional but not in same form. That not only largely increases the workload, but also inconvenient to subsequent maintenance. To solve those problems, this paper formulates a method for rapid development of cross-platform algorithm software based on MATLAB. Firstly, algorithms are developed and validated in MATLAB for its convenience; Secondly, MATLAB Coder is used to generate C/C++, which is independent of MATLAB; Then generated code are packaged into corresponding platform dynamic link library in target platform; Finally, by calling interface functions is able to achieve the requirement of the same algorithm software running on different platforms. The method mentioned above has been successfully applied to some engineering projects. During the period of software development, the method not only shortens the software development cycle, improves the efficiency of programming, reduces the errors of human, but also facilitates software post-maintenance.

Key words: Algorithm software; Cross-platform software development; Automatic generate code; MATLAB Coder

DOI: 10.12347/j.ycyk.20210910001

Citation: ZHOU Xiwa, ZHANG Guoyu, LI Yang, et al. A method of rapidly developing cross-platform algorithm software based on MATLAB [J]. Journal of Telemetry, Tracking and Command, 2022, 43(5): 68-73.

引 言

MATLAB^[1]具有高效的数值运算能力、完备的图形处理能力、丰富的工具箱和接近数学表达的语言等特点，为科学研究、算法开发提供了一种全面高效的解决方案，并让算法工程师在很大程度上摆脱了传统非交互式程序设计语言（如 C/C++）的编程模式^[2]。然而，直接利用 MATLAB 强大的函数库和工具箱开发的算法软件，无法直接应用在工程实现中，这就需要对 MATLAB 开发的算法进行人工二次编码，转换成独立于 MATLAB 软件的其他编程语言。由于编程语言的差异，在对算法的二次编码过程中，不仅不易保持 MATLAB 源码与其他编程语言的一致性，同时也容易出现人工手写代码的错误。

现代工程需求中，算法软件往往需要同时运行在不同的操作系统中，典型的如 Windows 系统、Linux 系统。不同操作系统，对应不同的开发语言。不同系统通过调用该系统提供的应用程序接口 API（Application Program Interface）来实现不同的功能需求，不同系统之间的 API 的实现方式差异很大。这使得跨平台开发面临着诸多问题^[3]，在某系统上开发的软件，无法直接应用于另一个系统中，需要编程人员作出大量修改方可移植。

目前，跨平台软件开发的主流编程语言是 Java 和 C/C++。Java 是将不同平台与操作系统的跨系统核心代码抽象出来，形成单独的 JVM 层（Java Virtual Machine），也就是 Java 的虚拟机^[4]。Java 语言设计本身不受平台的限制，Java 代码运行在 JVM 上。JAVA 跨平台运行的实质是不同平台有不同的 JVM 实现。只要针对不同平台开发相应的 JVM，代码就可以实现跨平台^[5]。因此，Java 付出了运行效率的代价，一般程序之间通过操作系统的 CPU 执行，而 Java 语言需要先通过 JVM 再映射到操作系统里，最后由操作系统执行。

QT 是一个跨平台的 C++ 开发库^[6]。QT 在不同操作系统下都有相应的底层类库，然后再通过一个公共的应用层接口实现跨平台，QT 的软件开发就是基于这个公共的应用层接口来完成的。使用 QT 编写的代码是跨平台的，而不是编译出来的文件跨平台^[7]。因此，同一份代码放到另一种平台上运行时，需要用该目标平台的类库进行重新编译。

但不管是用 Java 还是 C/C++ 进行算法软件开发，在算法的设计、调试阶段都不如 MATLAB 便捷。因此，本文实现了一种基于 MATLAB 快速开发跨平台算法软件的方法。该方法利用 MATLAB 的便捷性进行算法开发，通过自动代码生成标准的 C/C++ 代码，然后在不同运行环境下进行封装，实现不同平台的调用。这使得即使算法工程师并不熟悉跨平台的开发知识，也能够进行跨平台的算法软件开发，还可以把主要精力用于算法开发和系统测试。

1 跨平台算法软件开发流程

MATLAB 开发算法软件的优势非常明显：在算法设计阶段，通过利用数学意义明了的 M 代码进行算法研究、开发与验证，在结果得到验证后直接转换为目标平台的代码，然后封装和移植。基于 MATLAB 跨平台的算法软件开发流程如图 1 所示。根据输入需求及算法原理，用 MATLAB 进行算法设计、开发和验证，待算法满足功能和性能需求后，进行自动代码生成，并对生成的代码进行接口封装，满足不同平台下的移植需求。后期根据需求的扩展和变化不断优化、迭代和完善算法，再重新生成代码及封装，每次修改只需更改 MATLAB 中的源码，就能做到不同平台之间算法源码的一致性。

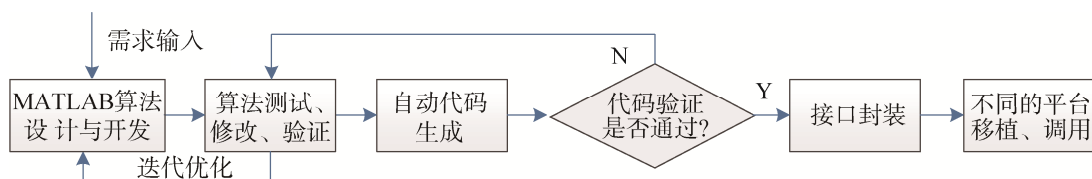


图 1 基于 MATLAB 跨平台算法软件开发流程图

Fig. 1 The flow chart of cross-platform algorithm software development based on MATLAB

2 自动代码生成技术

MATLAB Coder^[8]工具是 MathWorks 公司于 2011 年推出的一个重要产品，它可以将 MATLAB 函数直接生成可读、可移植的 C/C++ 代码。软件工程师们不再需要将设计的算法进行人工二次 C/C++ 语言开发，而是利用该软件中强大的函数库^[9]，按照一定的流程，直接进行代码自动生成。这种开发模式减少了复杂繁琐的代码转换、调试工作，避免了不必要的人工错误。

MATLAB Coder 进行自动代码生成时有一定的流程。简单的 M 函数^[1]可以直接进行代码生成，而面对复杂情况时，需要根据具体情况做出相应的修改与调整，才能达到自动代码生成的目的。在实际过程中，MATLAB 算法进行自动代码生成的具体流程如图 2 所示。

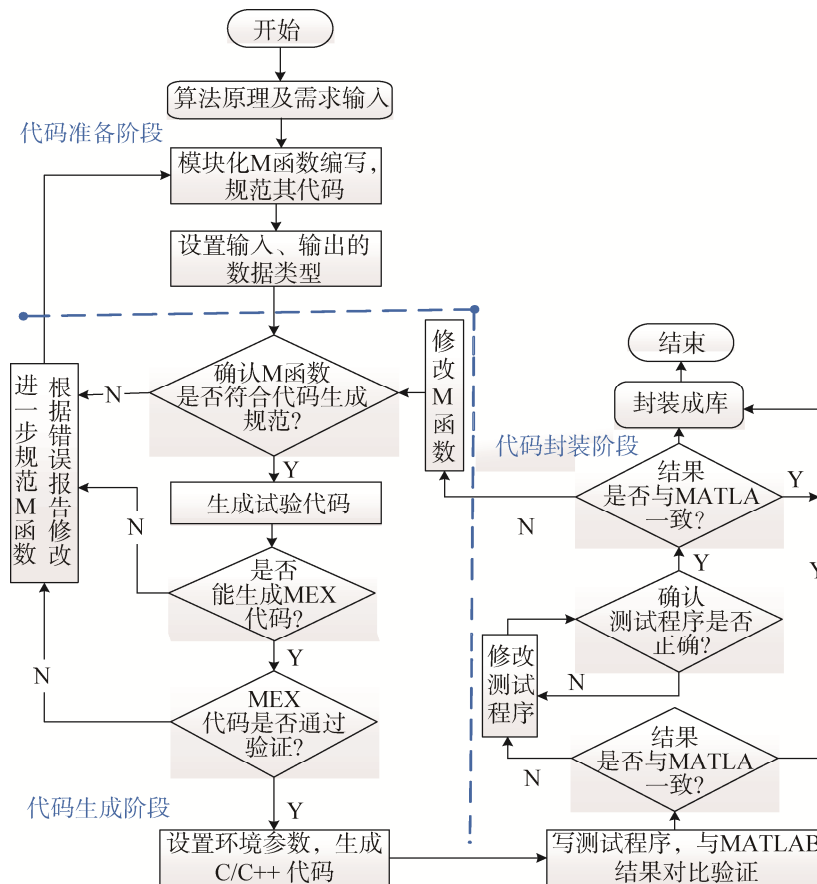


图 2 MATLAB Coder 自动代码生成流程图

Fig. 2 The flow chart of automatic code generation based on MATLAB Coder

图 2 中 MEX 代码验证主要有两个目的^[10]：一是修改和调整 M 函数，使其支持代码转换，并成功转换为目标代码；二是对转换好的目标代码进行验证，使 MEX 函数调用的结果与 M 文件运行的结果一致。MEX 文件相当于先转换为目标代码，并向 MATLAB 提供接口，使 MATLAB 直接调用并进行验证。因此，能很方便地验证转换代码编译后的执行结果。MEX 验证通过，意味着如果目标环境一致，生成的代码能够正确运行在目标环境中。

2.1 模块化 M 函数

在编写 M 函数时，要做到统筹编程，在编写时就注意自动代码生成的规范。按照模块化设计思想，将整个算法软件按功能划分成独立模块，不同模块实现不同功能，并将其实现的功能结果作为输出，将所需的数据作为输入，再将各个子模块组合封装成大系统中的一个配置项。该配置项除了有初始化、终止、数据类型定义等接口函数外，其他子函数在自动代码生成中均只有源文件 (*.c) 和头文件 (*.h)。自动代码生成后，对外接口函数调用只需关心该配置项函数的输入与输出。

2.2 错误检查及修改

在代码生成阶段,需要点击“Check for Issues”进行兼容性检查。对检查出的错误,可打开“Open Error Report”或“View Report”报告,查看错误提示信息和位置。根据相关提示信息进行修改,在不改变源码功能逻辑的情况下使得其符合自动代码生成规范,保存后再次进行检查。该过程需要多次迭代,以便逐个消除不兼容的情况,直到最终能够生成 MEX 代码。对于生成的 MEX 代码,会进行测试验证,在测试验证不成功时,同样会给出相应的报告文件,依据报告文件进行修改,直到 MEX 代码通过测试验证。此时表明可以生成 C/C++代码,并且该代码已预先通过了 MATLAB 的调用验证。

程序中影响自动代码生成的常见错误和应对方法总结如下:

① 数据类型赋值不一致或数据类型运行前后发生变化。如将 double 型数据赋值给 INT32 型数据,将 complex(复数)型数据赋值给 real(实数)型数据。可通过预先指定数据类型进行实现,如 `data=coder.nullcopy(zeros(N, 1, 'like', sig_in))`,指定 data 的数据类型与输入 sig_in 一致,如果 sig_in 为复数,则 data 为复数;sig_in 为实数,则 data 也会是实数型数据。

② 数据长度未提前声明或运行时长度发生变化。MATLAB 在运行过程中数据长度可以变化,但 C/C++则需确定的数据长度。因此,在使用数据时,需提前进行长度声明,如 `data=coder.nullcopy(zeros(N, 1, 'like', sig_in))`,coder.nullcopy 表示申明数组大小为 $N \times 1$,但不赋值,其中 N 为数组可能的最大长度。

③ 赋值语句左右两侧不一致。当数据长度不确定时,可用动态内存分配技术实现。推荐使用 `data=[]` 进行初始化;数据赋值时采用 `data=[data value]`,其中 value 为新的赋值数据,可以为单值,也可以是复杂的数据类型。其中 `data=[data value]` 为行向量;`data=[data; value]` 为列向量。通过该方式可实现数据的动态内存分配。

④ 变量在某些语句中未定义。经常出现在 if else 语句中并且该变量作为输出的情况,在这种情况下,使用变量之前先进行初始化,确保每条分支语句都有确定输出。

⑤ MATLAB 自带的一些函数不支持直接转换,如 decimate, interp, tabulate 等函数。因此,先修改再进行转换。比如插值 interp 函数,可提前将不同倍数的滤波器系数存储到文件里,运行时用关键字 coder.load 进行加载,通过首地址和数据长度进行滤波系数的提取,然后再进行插值运算。

2.3 快速 FFT 函数的生成

FFTW3^[11]是一个快速计算离散傅里叶变换的标准 C 语言程序集,号称最快的 FFT 算法^[12, 13]。在 MATLAB Coder 转成 C/C++语言时,并不会自动调用 FFTW3 库,因此生成的代码 FFT 运算不如 FFTW3 快。这里介绍一种能够直接调用 FFTW3 库的代码生成方法。首先,将 FFTW3 的头文件和库文件放在 MATLAB 安装路径>user>lib 中,使用 `coder.fftw.StandaloneFFTW3Interface` 进行 useMyFFTW 的类定义,具体使用参考 help 说明。然后在 Generate Code 中设置 Custom FFT library callback 为 useMyFFTW,则在生成的代码中,使用到 FFT 函数的地方,都会自动调用 FFTW3 库里的函数。

2.4 接口函数的简化

数据类型简单的接口函数,将会使函数调用非常方便。当接口函数输入和输出数据长度可变时,其接口函数的数据类型为可变数据长度 emxArray 的数据类型。这将会导致接口调用复杂,而且容易由于未事先申明变量导致函数调用不成功。因此,这里介绍一种能够生成简易接口函数的方法。首先,声明输出数组可能的最大长度;其次,在 MATLAB 中将输出数组作为输入进行调用;最后,在函数输出的地方进行数组的赋值操作。实例如下:

① 预先申明输出数组 fre_abs 和 fre_cplx 可能的最大长度,假设为 L, sig_in 为输入的复数型数据
`fre_abs=coder.nullcopy(zeros(L, 1, 'like', real(sig_in)));`
`fre_cplx=coder.nullcopy(zeros(L, 1, 'like', sig_in));`

② 将输出数组作为接口函数的输入、输出进行调用, len 表示实际有效数据长度
`function [fre_abs, fre_cplx, len]=FunExample(sig_in, fre_abs, fre_cplx) %codegen`

③ 在函数输出时,进行数据赋值

```
fre_cplx(1:len)=outData(1:len);  
fre_abs(1:len)=abs(fre_cplx);
```

按照以上步骤进行操作, 其生成代码的接口函数数据类型会相当简单。

2.5 代码生成

当 MEX 代码通过测试验证时, 就可以点击代码生成了, 代码生成支持定制化操作。比如生成的类型可以选择源文件、MEX 文件、静态库、动态库以及可执行文件; 可选择硬件版、编译工具; 同时支持文件路径、运行速度、存储特性、代码风格、调试设置等定制化设置; 一般情况下选择默认设置即可。

自动代码生成后, 会有对应函数的源文件 (*.c) 和头文件 (*.h), 其文件一一对应。可对 MATLAB 源程序和生成代码进行双向追踪。除此之外, 需要关注一些接口类的文件, 比如初始化类函数 *.initial.c, 终止类函数 *.terminate.c; *.initial.c 和 *.terminate.c 一般为空, 在封装接口时可以根据需要自行定义; *.types.h 文件中定义了输入、输出的数据结构; *.emxAPI.c 函数为 emxArray 数据类型 (动态分配数组) 的 API 接口函数; *.emxutil.c 函数可对每一 emxArray 型变量判断是否有足够的空间。同时在 examples 文件中的 main.c 函数展示了如何调用生成的代码, 包括数据的声明和初始化、程序的调用、终止和内存的释放, 是写接口函数的极好参考。

2.6 代码验证

原则上, 如果 MEX 验证通过了, 则生成的代码是不会有问题的。在对生成代码进行验证 (确保与 MATLAB 程序运行时的输入数据完全一致), 出现结果与 MATLAB 不一致时, 大部分原因是生成代码的调用程序不完善或不正确所导致的。比如直接使用没有初始化的 emxArray 数据类型, 导致程序崩溃。因此, 首先需确保验证程序本身不存在 BUG, 验证程序可参照 examples 中的 main 函数, 此时需将输入数据改成待验证的测试数据。在保证验证程序正确的情况下, 如果调用生成的代码出现崩溃或者结果与实际不符 (这种情况较少), 可通过对验证程序进行断点调试, 找到出错的位置, 并在 MATLAB 源码对应的位置进行修改。一旦修改了 MATLAB 源码, 则重新进行代码生成和验证。只要 MATLAB 源程序设计科学, 没有 BUG, 生成的代码在目标平台上也能正常运行。

3 生成代码的跨平台封装

自动代码生成的函数可以直接在目标平台上调用, 但为了规范化管理软件, 可对生成的代码用统一接口函数进行封装。其封装的接口函数可参照 examples 中的 main 函数进行修改。

不同运行环境下, 算法软件的封装方式不一样。比如 Windows 平台调用的话, 在统一接口函数引入关键字 __declspec(dllexport), 然后在 Visual Studio 下将统一接口函数封装成 .dll (Dynamic Link Library) 文件, 在目标平台调用时引入关键字 __declspec(dllimport) 即可^[14]; 而 Linux 平台中调用, 可直接在 Shell 中用 gcc 打包封装成 .so (Share Object) 库进行调用^[15]; 在安卓系统中通常采用 Java 进行编程, Java 调用标准 C/C++ 时采用关键字 JNIEXPORT 和 JNIEnv 对生成的代码进行统一接口函数的编写^[5], 再将统一接口函数打包封装成 .so 文件, 即可作为插件使用; 在 QT 中调用标准 C/C++ 时将统一接口函数封装成类^[6], 再打包成 .so 文件, 即可在 QT 作为插件使用。

代码封装后, 算法软件作为一个整体在调用的平台中进行软件测试。首先通过专门测试工具进行软件的单元测试; 测试时算法软件的功能, 应至少被一个正常测试用例或一个被认可的异常测试用例所覆盖; 算法软件测试时, 其输入应至少包括有效等价类值、无效等价类和边界数值; 测试算法软件运行在边界状态、异常状态和人为设定状态下时的功能和性能。当算法软件没有通过测试时, 定位问题, 然后返回 MATLAB 源码进行修改, 再进行自动代码生成、验证、封装和调用, 直到通过全部的软件测试为止。

4 实际应用

本文所提出的算法软件跨平台开发, 成功用于某大数据汇聚处理系统和某接收机项目。前期通过 MATLAB 进行算法设计、仿真和验证, 确保算法本身设计合理后, 进行自动代码生成。生成后的代码

在 Visual Studio 进行调用, 通过了 TestBed 的软件测试, 在测试过程中代码没有出现问题。然后将代码封装成.dll 文件, 在 Windows 系统下开发的上位机中成功调用该.dll 文件, 运行结果正确。在某大数据汇聚处理系统中, 由于软件均运行在国产化硬件平台的 Linux 系统中, 因此将自动生成的代码封装成.so 文件, 在该系统中成功调用了该.so 的算法插件, 实现了相应的功能。在某接收机项目中, 由于开发时间紧, 研发人员主要将工作放在 MATLAB 上的算法设计、验证和优化上, 后期通过代码生成和封装, 最终在 QT 中成功调用该算法软件, 实现了系统的联调。

5 结束语

本文利用 MATLAB 开发算法的便捷性和自动代码生成技术, 提出了一种基于 MATLAB 的快速开发跨平台算法软件的方法。在实际工程项目中, 应用了该算法的开发模式, 验证了本方法的有效性和高效性。算法研发是未来的核心竞争力, 开发人员通过本文的方法进行算法软件开发, 将主要精力集中在算法研发上, 所开发的算法软件只需适配性修改接口函数就可以实现在不同平台中调用, 提升了软件开发效率。本文提出的跨平台算法软件开发方法, 将会有更大的应用价值。

参考文献

- [1] 胡晓冬, 董辰辉, 等. MATLAB 从入门到精通 (第 2 版) [M]. 北京: 人民邮电出版社, 2018.
- [2] 张轶. MATLAB 信号处理—算法、仿真与实现[M]. 北京: 清华大学出版社, 2022.
- [3] 张弛. 基于 C++ 语言的跨平台软件开发的设计与实现[D]. 北京: 北京交通大学, 2010.
- [4] 周志明. 深入理解 JAVA 虚拟机[M]. 北京: 机械工业出版社, 2011.
- [5] 明日科技. JAVA 从入门到精通 (第 5 版) [M]. 北京: 清华大学出版社, 2019.
- [6] 彭源, 孙超超, 田秀霞, 等. Qt C++ 编程从入门到实践[M]. 北京: 清华大学出版社, 2022.
- [7] 徐野, 赵星宇, 黄海新. Qt 平台体系与应用-Qt5.5+核心方法、技巧与案例. [M]. 北京: 北京航空航天大学出版社, 2017.
- [8] Online MATLAB[EB/OL]. [2021-09-10]. <http://www.mathworks.com/help/coder>.
- [9] 周世钦, 王波涛. MATLAB 程序转 C 代码的方法研究[J]. 价值工程, 2018, 37(2): 182–185.
ZHOU Shiqin, WANG Botao. Method study of MATLAB program transforming to C code[J]. Value Engineering, 2018, 37(2): 182–185.
- [10] 杨耀宗. 基于 MATLAB 的 λ 能谱处理工具箱的研制[D]. 成都: 成都理工大学, 2014.
- [11] FFTW[EB/OL]. [2021-09-10]. <http://www.fftw.org>.
- [12] FRIGO M, JOHNSON S G. The design and implementation of FFTW3[J]. Proceedings of the IEEE, 2005, 93(2): 216–231.
- [13] 李佳伟. 软件化雷达系统实时性研究[D]. 西安: 西安电子科技大学, 2018.
- [14] 张铮, 张宝山, 周天立. Windows 程序设计 (第 3 版) [M]. 北京: 人民邮电出版社, 2022.
- [15] BLUM R, BRESNAHAN C. Linux 命令行与 shell 脚本编程大全 (第 3 版) [M]. 北京: 人民邮电出版社, 2016.

[作者简介]

- 周希娃 1988 年生, 硕士, 工程师, 主要研究方向为信息对抗。
张国玉 1987 年生, 硕士, 高级工程师, 主要研究方向为信息对抗。
李洋 1996 年生, 硕士, 助理工程师, 主要研究方向为信息对抗。
胡继军 1981 年生, 硕士, 研究员, 主要研究方向为信息对抗。

(本文编辑: 傅 杰)