

基于搜索的 Simulink 测试数据生成*

邓绍鹏, 杨志义, 王宇英

(西北工业大学 计算机学院 陕西省嵌入式系统技术重点实验室, 西安 710129)

摘要: 为满足基于模型的测试需求,研究了基于搜索的 Simulink 模型测试数据生成方法。对于严格约束,该方法利用代价函数规则构建目标函数,针对 Simulink 模型复杂性的特点,采用模拟退火算法对目标函数求优,最终搜索出满足目标约束的测试数据。由此设计了一种综合考虑宽松约束和严格约束的 Simulink 测试数据生成框架,最后实现了该方法在典型案例的应用。实践证明,该方法快速有效地解决了 Simulink 模型的测试数据生成问题。

关键词: 测试数据生成; Simulink 测试; 基于搜索的测试; 基于模型的测试; 模拟退火

中图分类号: TP311.56 **文献标志码:** A **文章编号:** 1001-3695(2012)07-2527-04

doi:10.3969/j.issn.1001-3695.2012.07.033

Search-based test-data generation for Simulink

DENG Shao-peng, YANG Zhi-yi, WANG Yu-ying

(Shaanxi Key Laboratory of Embedded System Technology, School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract: In order to apply model-based testing, this paper focused on search-based approach of Simulink test-data generation. This method firstly constructed the objective function using the principle of cost function which met the requirements of particular target constraint. According to the complexity of Simulink model, it calculated the optimal value of the target function through the algorithm of simulated annealing and searchet the test-data which met the target constraint. And put forward a test-data generation framework which took loose constraints and tense constraints into consideration and illustrated the usage of this framework through a case study. Finally, it shows that search-based test-data generation is useful for Simulink model testing.

Key words: test-data generation; Simulink testing; search-based testing; model-based testing; simulated annealing

基于模型的设计已经成为嵌入式软件设计的一种标准模式^[1]。在模型驱动的软件设计思想下,可以通过编写模型来描述和分析系统软硬件的行为。在完成系统初步设计的同时,尽早地对模型进行测试,排除模型中的错误。如果系统可靠性较高,可以直接生成代码和系统程序,用于后期集成为更大更复杂的系统。这种嵌入式系统设计方法能够提高设计的质量,缩短后期测试时间,减少开发周期,同时还能保证系统的品质。

目前由模型自动生成代码,并自动生成基于这些代码的测试数据的方法已很成熟,但是还不能很好地从模型直接生成可靠性较高的测试用例来验证模型的正确性。所以,研究如何从模型生成测试用例对基于模型的开发技术具有重要意义。

Simulink 工具满足模型驱动架构(MDA)的设计要求,能够快速建立用户需要的物理和计算模型,可用于快速原型开发。本文研究和设计了一种基于搜索的 Simulink 模型测试数据生成方法。

1 模型测试数据生成

1.1 基于模型的测试

与基于模型的设计类似,基于模型的测试是与之配套的测试方法。该过程在以自上而下的方法设计各个模块的同时,按

照同样的顺序测试模块的逻辑、实现和功能代码,在多个模块组合集成为目标系统后,再由自下而上的方式进行软件系统测试硬件在线测试和全系统集成测试。基于模型的整个测试过程可以用一个 V 字图来描述^[2],如图 1 所示。

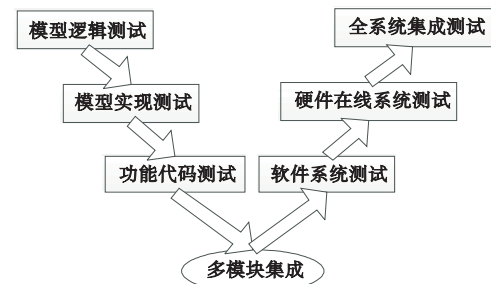


图1 基于模型的测试流程

本文介绍的方法主要针对图1中的模型逻辑测试阶段。从图1可以看出,该阶段的测试直接影响后续多个过程的设计和测试。不仅如此,通过测试用例复用,直接在后续阶段使用第一阶段产生的测试用例。由此可见,本文关注的研究对于基于模型的开发全过程都具有重要意义。

1.2 测试数据生成

软件测试包含测试用例生成、测试执行和测试结果分析。研究表明测试用例生成是最消耗时间和人力的过程^[3]。

收稿日期: 2011-12-12; 修回日期: 2012-01-16 基金项目: 国家“863”计划资助项目(2011AA010101); 西安市科技计划项目(CXY1008)

作者简介: 邓绍鹏(1988-),男,江西吉安人,硕士研究生,主要研究方向为嵌入式软件仿真测试(dspMint@gmail.com); 杨志义(1952-),男,教授,主要研究方向为分布计算、网络化嵌入式计算; 王宇英(1978-),女,讲师,博士,主要研究方向为嵌入式系统验证与仿真技术。

测试数据是测试用例中的测试输入,可以通过数据来描述。Simulink 测试用例中最重要的是相应测试数据的设计。

因此,如何自动高效地从 Simulink 模型产生需要的测试数据,是本文所要研究的内容。

测试用例生成就是解决如何生成特定的测试输入数据和测试输出数据的问题。当以其中某个测试输入作为输入数据时,被测系统中的特定结构或方法能够被执行,并产生对应的测试输出;而且还需要尽力地寻找足够多的测试用例,使测试可以满足测试用例充分性的要求。Simulink 测试用例中,最为重要的是测试输入数据。通常,生成特定要求下的测试数据问题可以转换为一个约束满足的求解问题。

目前,已经存在一些成熟的测试数据生成方法,考虑到 Simulink 模型本身的特点,本文采用了基于搜索的测试技术来生成测试数据。

2 基于搜索的 Simulink 测试技术

最近几年,基于搜索的软件工程(SBSE)正以引人注目的速度发展,基于搜索的优化求解算法贯穿于软件工程的整个生命周期^[4]。其中,基于搜索的软件测试技术(search-based software testing)是基于搜索的软件工程领域的研究者关注的焦点。该技术不仅仅用于基于搜索的测试中,由于搜索算法具有明确的目的性,同样也用于回归测试的用例生成。目前,基于搜索的测试方法已经广泛用于结构化测试、功能测试和非功能测试。

由于基于搜索的方法原理简单且具有很强的适应性,考虑到复杂多变的 Simulink 模型,本文采用了基于模拟退火算法的搜索测试技术。应用该方法必须解决目标函数中的代价函数定义规则设定和测试数据搜索中搜索算法选择两个问题,在此基础上,构建测试数据生成环境搜索测试数据。本章将研究该方法在 Simulink 模型测试中的应用。

2.1 代价函数

测试用例生成按照被测程序是否被执行可分为动态测试和静态测试。基于搜索的测试用例生成是一种动态测试方法,即测试数据在模型运行之后得出。因此,基于搜索的测试技术的重点在于有目的地运行模型,从而筛选出满足特定覆盖目的的测试用例。

一般地,基于搜索的测试数据生成的主要方法是通过搜索算法努力寻求测试输入数据以满足被测对象内部的特定约束。例如分支情况中,若需要测试某一条件为真的分支情况,此时的约束就是该分支判断条件为真,且满足分支运行的前置条件。要采用搜索算法来求解满足约束的输入值,必须设定一个目标函数(通常是最小化目标函数),该目标函数综合评价模型运行过程中所有约束被遵守和违反的程度。

为了定量表示目标函数,实际操作时,可以对于每个约束设定一个代价函数,定量表示相应约束条件不满足的程度。代价函数可理解为对该约束不被满足的一种惩罚。例如,若实际过程中,约束偏离较大时代价函数取较大的正值,而满足约束时代价函数取 0 值,甚至是负值。目标函数就是各约束满足情

况的汇总,可通过代价函数的组合构成搜索时需要的目标函数。在这样的理论下,测试数据生成可以转换为求解目标函数最小值的优化问题。通过优化搜索算法,找到输入域中满足目标函数最小的输入,即满足所有约束的测试用例。

如何设定合理的代价函数是成功运用搜索测试的关键。参考 Tracey^[5]和 Bottaci^[6]提出的规则,本文设计了代价函数定义基本规则,分为简单约束和复杂约束,列举为表 1、2。表 1 为简单约束的代价函数定义。其中 K 表示当约束不满足时的惩罚代价,可用于复杂约束时平衡各约束之间的差异。对于表 1、2,可总结为:约束满足时代价为 0,不满足时代价要根据不同的约束类型来计算。对于简单的布尔型约束,不满足时代价为 K ;而对于不等式和等式约束,不满足时,代价还需算上偏离约束的程度;如果是复合约束,则按照复合逻辑的一些特性,通过各子项约束的代价函数组合而成。复合约束的代价函数定义如表 2 所示。

表 1 简单约束的代价函数

约束表达式 E	约束满足时 $\text{cost}(E)$	约束不满足时 $\text{cost}(E)$
简单布尔型	0	K
$A \neq B$	0	K
$A < B$	0	$A - B + K$
$A = B$	0	$\text{abs}(A - B) + K$
$A \leq B$	0	$A - B + K$

表 2 复合约束的代价函数

复合约束表达式 E	代价函数 $\text{cost}(E)$
$E_1 \vee E_2$	$(\text{cost}(E_1) \times \text{cost}(E_2)) / (\text{cost}(E_1) + \text{cost}(E_2))$
$E_1 \wedge E_2$	$\text{cost}(E_1) + \text{cost}(E_2)$

从前文的解释可以得出,目标函数是各约束组合的复合约束的代价函数。举个例子,目标函数涉及的目标约束 E 与其三个子约束 E_1 、 E_2 、 E_3 的关系为 $E = E_1 \vee (E_2 \wedge E_3)$,则寻求满足这些约束的目标函数 obj 表达式应该为:

$$\text{obj} = \text{cost}(E) = \frac{\text{cost}(E_1) \times \text{cost}(E_2 \wedge E_3)}{\text{cost}(E_1) + \text{cost}(E_2 \wedge E_3)} = \frac{\text{cost}(E_1) \times [\text{cost}(E_2) + \text{cost}(E_3)]}{\text{cost}(E_1) + \text{cost}(E_2) + \text{cost}(E_3)}$$

其他复杂约束的目标函数也可按照上述组合规则分析和化简得到。

2.2 搜索算法

要求解目标函数的最优值,需要选用一种优化算法,用于搜索能够得出最优值时的模型输入值。搜索算法不仅可以从一些成熟的线性、非线性规划方法中选取,还可以利用启发式的优化算法。文献[7]用进化测试的概念来特指使用启发搜索算法自动生成测试数据。常用的启发式优化算法包括模拟退火算法、遗传算法和蚁群算法等。McMinn^[8]总结了通过启发式搜索来生成测试用例的方法。

由于 Simulink 模型输入常常十分复杂,而常用的非启发式算法都会在某些方面具有局限性。Bostrom 等人^[9]采用了 MATLAB 优化工具箱中提供的 fmincon 函数,该方法是一种常用的非线性优化方法,算法采用信赖域法或序列二次规划来逐步求优。但上述作者在进行大量与随机搜索算法对比实验

后指出,采用 `fmincon` 函数的局限性在于最终求出的测试数据往往陷入给定的初始数据附近的局部最优值。

对于搜索全局最优的测试数据,应该综合考虑采用效率较高的启发式搜索算法。经过对比发现,模拟退火算法在基于搜索的 Simulink 模型测试中有以下特点:

a) 由于退火算法相对较强的随机特性,对于在较大区域存在分散的极值的情况,算法找到全局最优值的可能性较大;

b) 当系统的极值在较小的区域存在个别极值时,找到全局最优目标极值的概率比较低,此时可以通过延长求解时间或者多次搜索来获取更高的搜索成功概率;

c) 整体来说,模拟退火算法对于 Simulink 模型中各种复杂情况具有更高的适应性,且相对其他启发式搜索算法在寻找全局最优方面具有更高的效率。

本文选择在测试中常用的模拟退火算法。该算法可以直接从 MATLAB 工具箱中,通过 MATLAB 编程环境调用,用于 Simulink 模型测试数据生成,十分方便。

2.3 Simulink 测试数据生成方法

基于以上理论,本文设计了一种 Simulink 测试数据自动生成方法,其生成流程和测试数据生成框架将在下文介绍。

需要指出的是,测试数据要满足覆盖要求,这也是测试数据生成的目的。Simulink 模型测试数据生成的终止条件是测试用例覆盖模型的各种情况。通常要考虑以下覆盖:

- 对于模型中判断分支结构,包括条件判断的覆盖;
- 对于含有状态机的模型,覆盖所有的简单状态;
- 边界值覆盖,使条件或状态改变的临界值及邻域值。

应用基于搜索的 Simulink 模型测试数据生成方法之前,需要按照已有的覆盖情况来确定每一次搜索的覆盖目标。对于相对容易的覆盖目标(称之为宽松约束),可通过随机数据生成来产生测试数据;反之,严格约束难以通过随机方法得到测试数据,就采用本文介绍的搜索优化方法。Simulink 测试数据生成框架可以描述为图2所示流程。采用这种方案可以节省搜索测试消耗的大量时间。

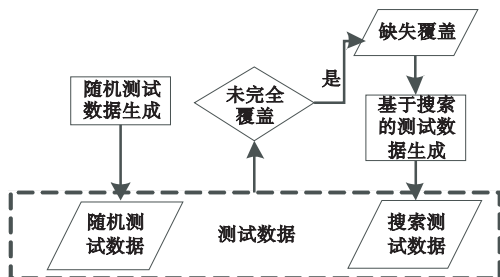


图2 Simulink模型测试数据生成框架

图2中基于搜索的 Simulink 模型测试数据生成过程可用图3的流程来表示,具体分为以下几个步骤:

a) 根据覆盖目标需要,确定要导出的数据,在模型中利用 To Workspace 模块,将数据流导出到 Workspace 以备目标函数计算使用。这一步称为插桩,导出的数据称为桩点变量。

b) 目标函数是由代价函数组合而成,代价函数是插桩得到的桩点变量的函数。利用前文中代价函数的定义,可以推导出目标函数。本文采用 MATLAB 的 `m` 函数编写。

c) 建立模型运行的必要环境,包括运行参数设定、目标函数

调用、搜索算法设定等,可以采用 MATLAB 脚本建立运行环境。

d) 搜索算法使用的初始输入数据可以采用随机得到,也可人为设定,一般固化在运行脚本中。

e) 在运行环境中运行搜索算法,计算最优输入,作为目标测试数据。

至此,一个完整的基于搜索的 Simulink 模型的测试数据生成方法已经设计完毕。第3章将用一个实例详细介绍该方法的应用。

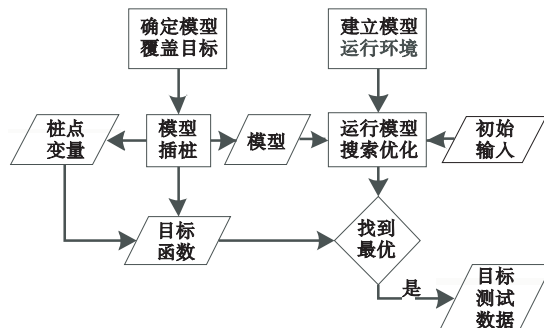


图3 基于搜索的测试数据生成流程

3 案例与实现

3.1 案例简介

在实际项目应用的 Simulink 模型中经常会包含控制系统和选择部件。为了说明本文的测试数据生成方法,用一个典型实例来简要介绍该方法的应用和实现。

图4为一个包含离散控制系统的 Simulink 模型,如图中虚线框标记所示,前面是一个离散控制系统,后面为一个选择输出部件。如果控制系统输出大于等于 1.04,整个系统输出是 1;反之,系统输出是 -1。图中的输入信号 `in` 是 $[0, 1]$ 范围内的离散序列,对于所有符合要求的输入,该离散控制系统的输出将有微小几率越过 1.04,即系统在大多数情况的输出全 -1 的序列。因此,条件开关选择 -1 作为输出情况的测试数据非常容易通过随机方法得到,但使系统中开关选择 1 的测试数据因为随机得到概率很小,需用搜索算法得出。

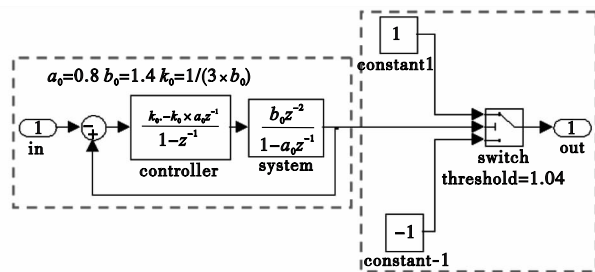


图4 被测Simulink系统模型

3.2 案例实现

本节将针对 3.1 节中的例子,实现了上述 Simulink 模型的搜索测试数据生成,具体如下:

a) 插桩。为了搜索到条件开关选择 1 的情况,离散控制系统需要突破 1.04。因此应以离散控制系统输出信号 `Out_system` 作为桩点变量,约束表达式为 $\max(\text{Out_system}) > = \text{threshold}$ 。模型插桩后如图5所示。

b) 编写目标函数。本例中的约束为 $\max(\text{Out_system}) > =$

threshold, 根据表 1 的代价函数定义方法, 目标函数可定义为: $\max(\text{Out_system}) \geq \text{threshold}$ 时取 0, 否则为 $\text{threshold} - \max(\text{Out_system}) + K$ 。因为本例中不涉及到其他约束, 可取 $K = 0$ 。目标函数可用 MATLAB 脚本编写, 参考如下脚本的写法。由定义可知, 目标函数最优值为 0。

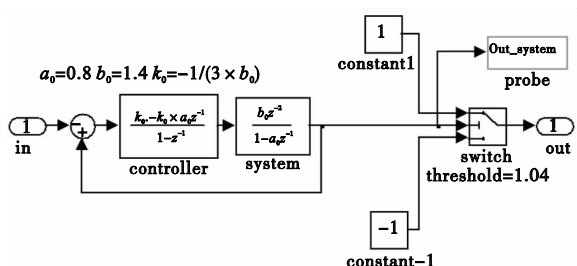


图5 插桩后的系统模型

```
function y = obj_function(x) % 变量 x 是需要优化的输入
global n;
% 构造输入矩阵 ut = [t, u], t 为时间变量, 输入信号 u 为 t 的函数
u = x'; t = [1:n]'; ut = [t, u];
model = 'control_system'; % 模型名称
sim(model, n, [], ut); % 以 ut 为输入运行仿真, n 为仿真时长
maxnum = max(Out_system(:, 1));
threshold = 1.04;
if maxnum >= threshold
y = 0; % 约束满足时代价函数取值
else
y = threshold - maxnum; % 约束不满足时代价函数取值
end
```

建立测试数据生成环境, 主要包括清理用户空间、设定模型运行所需的参数、设定输入数据上下限和初始值、指出调用的目标函数句柄、模拟退火优化算法参数设定等其他内容。用 MATLAB 编写的测试数据生成脚本如下:

```
a0 = 0.8; b0 = 1.4; k0 = -1/(3 * b0); % 模型参数的设定
global n; n = 10; % global 参数将在目标函数中使用
ObjectiveFunction = @obj_function; % 目标函数句柄设定
x0 = rand(1, n); % 输入值初始值
lb = 0 * ones(1, n); % 输入值下限
ub = 1 * ones(1, n); % 输入值上限
options = sgaoptimset('MaxIter', 500, 'StallIterLim', 500, 'TolFun', 1e-4, 'AnnealingFcn', @annealingboltz, 'InitialTemperature', 1000, 'TemperatureFcn', @temperatureexp, 'ReannealInterval', 20, 'PlotFcns', {@saplotbestx, @saplotbestf, @saplotx, @saplotf}); % 模拟退火算法参数设定
[x, fval, exitflag, output] = simulannealbnd(ObjectiveFunction, x0, lb, ub, options); % 用模拟退火算法求目标函数的最值 fval 和输入值 x
c) 设定搜索初始值。在上述脚本中初始值  $x_0$  是长度为 10 的随机序列。
```

运行搜索生成测试数据。运行第三步编写的脚本, MATLAB 将调用模拟退火算法自动生成最优的输入数据。当仿真时长为 10, 一次运行后的优化得到的输入序列为 [0.0001 0.9995 0.9990 0.9990 0.9896 0.9979 0.9794 0.9794 0.0094 0.0104], 可以验证对应的系统输出序列为 [-1 -1 -1 -1 -1 -1 -1 -1 -1 -1]。

运行时如图 6 所示, 模拟退火算法运行时绘制了优化过程中迭代 300 次得到的目标函数最优值。经过 236 次迭代计算,

目标函数已经达到最小值 0, 即已经找到需要的测试数据。按照上述测试数据运行模型, 可测试系统中开关选择 1 时的模型运行情况。

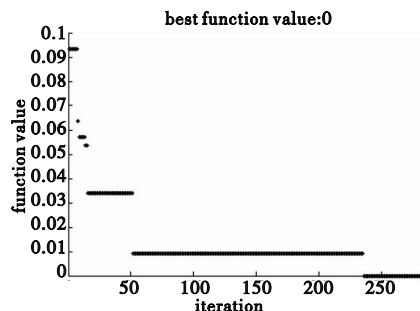


图6 搜索迭代次数与函数最优值

与文献[9]中方法相比, 本文由于采用模拟退火算法, 搜索结果不依赖于初始输入, 能够跳出初始输入附近的局部最优, 搜索到目标输入的稳定性更高, 所以本文介绍的方法同样适用于复杂环境下的 Simulink 模型测试数据生成。

4 结束语

针对基于模型的测试在嵌入式系统开发过程中的重要意义, 本文应用基于搜索的测试数据生成方法, 设计了一种 Simulink 测试数据生成框架。通过基于搜索的方法与随机生成方法的结合, 利用代价函数定义规则, 构建目标函数; 针对 Simulink 模型复杂性的特点, 选取模拟退火算法对目标函数求优, 能够快速准确地生成满足覆盖要求的测试数据。对比相关文献发现, 该方法快速有效地解决了 Simulink 模型的测试数据生成问题, 具有实用意义。

参考文献:

- [1] WINDISCH A. Search-based testing of complex simulink models containing stateflow diagrams [C]//Proc of the 31st International Conference on Software Engineering. Berlin: IEEE Press, 2009: 395-398.
- [2] LAMBERG K, BEINE M, ECHMANN M, et al. Model-based testing of embedded automotive software using Mtest, 2004-01-1593 [R]. Pennsylvania: SAE International, 2004.
- [3] BEIZER B. Software testing techniques [M]. London: Thomson Computer Press, 1990.
- [4] HARMAN M, MANSOURI S A, ZHANG Yuan-yuan. Search based software engineering: a comprehensive analysis and review of trends techniques and applications, TR-09-03 [R]. London: King's College London, 2009.
- [5] TRACEY N. A search-based automated test-data generation framework for safety critical software [D]. York, UK: University of York, 2000.
- [6] BOTTACI L. Predicate expression cost functions to guide evolutionary search for test data [C]//Proc of Genetic and Evolutionary Computation Conference. Berlin: Springer-Verlag, 2003: 2455-2464.
- [7] 赵庆兰, 艾丽蓉, 刘西洋. 基于进化算法的软件测试数据生成的自动化 [J]. 计算机测量与控制, 2006, 14(10): 1420-1423.
- [8] McMINN P. Search-based software test data generation: a survey [J]. Software Testing, Verification and Reliability, 2004, 14(2): 105-156.
- [9] BOSTROM P, BJORKQVIST J. Optimisation-based black-box testing of assertions in Simulink model, 711 [R]. 2005.