

复杂陈述式仿真模型的相容性分析*

丁建完⁺, 陈立平, 周凡利, 黄华

(华中科技大学 国家 CAD 支撑软件工程技术研究中心,湖北 武汉 430074)

Consistency Analysis of Complex Declarative Simulation Models

DING Jian-Wan⁺, CHEN Li-Ping, ZHOU Fan-Li, HUANG Hua

(National CAD Support Software Engineering Research Center, Huazhong University of Science and Technology, Wuhan 430074, China)

+ Corresponding author: Phn: +86-27-87543973, Fax: +86-27-87547405, E-mail: jwdingwh@etang.com, http://www.hust.edu.cn

Received 2004-09-28; Accepted 2005-06-02

Ding JW, Chen LP, Zhou FL, Huang H. Consistency analysis of complex declarative simulation models. *Journal of Software*, 2005,16(11):1868–1875. DOI: 10.1360/jos161868

Abstract: The use of mathematical modeling and simulation in engineering is rapidly increasing since modern products are increasingly complex and heterogeneous. Consistency analysis of simulation model is a crucial subject of multi-domain modeling of complex physical systems. In this paper, a structural analysis method based on graph theoretical approaches is presented. The method can not only determine whether a simulation model is consistent or not, but also decompose the overall system of equations into three distinct parts: over-constrained, under-constrained, and well-constrained part. A methodology is also proposed for detecting and repairing over-constrained and under-constrained situations. Equations and variables that cause the inconsistencies can be automatically detected and isolated without solving the system of equations, and meaningful repairing messages for users are given. The methodology can considerably enhance the error finding and correcting process by providing a broad range of errors, and is implemented in a modeling and simulation tool, named MWorks.

Key words: consistency analysis; declarative model; simulation; Modelica; multi-domain

摘要: 模型的相容性分析是复杂产品多领域建模面临的一个关键问题。研究了陈述式基于方程仿真模型的方程系统过约束或欠约束判定、检测与修正策略。首先,将方程系统表示为二部图,通过二部图分解判定模型的相容性,并分离出方程系统的过约束和欠约束部分。然后,通过检测、判定与缩减过程,自动判别出过约束或欠约束发生的大致范围,并给出修正方案。所提出的策略与算法能够显著地提高用户发现与排除过约束或欠约束问题的效率,已在多领域物理系统混合建模与仿真平台 MWorks 中实现。

* Supported by the National Natural Science Foundation of China under Grant No.60574053 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2003AA001031 (国家高技术研究发展计划(863)); the National Grand Fundamental Research 973 Program of China under Grant No.2003CB716207 (国家重点基础研究发展规划(973))

作者简介: 丁建完(1975 -),男,湖南桃江人,博士生,主要研究领域为多领域系统建模与求解;陈立平(1964 -),男,博士,教授,博士生导师,主要研究领域为变量几何,装配设计,参数化设计,智能 CAD,虚拟仿真;周凡利(1976 -),男,博士生,主要研究领域为多体系系统建模与仿真,多领域系统建模与求解;黄华(1975 -),男,硕士,主要研究领域为多领域系统建模与仿真。

关键词: 相容性分析;陈述式模型;仿真;Modelica;多领域

中图法分类号: TP311 文献标识码: A

数学建模与仿真作为分析产品技术性能的重要手段,在产品开发中发挥着越来越重要的作用.随着科技的高速发展,现代产品日趋复杂与异构,如汽车、机器人、航空航天器等高科技产品,通常是集机械、电子、液压、控制等多个学科领域功能单元于一体的复杂系统.完整地分析跨领域复杂系统的整体性能需要对系统进行多领域建模,现有的商用建模仿真工具还无法满足这一需要^[1].为此,欧洲仿真界的一群学者在统一和归纳先多种数学建模语言的基础上,推出了一种面向对象的、陈述式基于方程建模语言——Modelica^[2].该语言采用数学方程统一不同领域的模型描述,较好地解决了模型的交互和重用问题,非常适合于复杂多领域物理系统建模.

复杂多领域物理系统建模是一件非常复杂和费时的事情,建模工作量大且容易出错.对于陈述式基于方程仿真模型来说,问题主要表现为模型不相容,即模型中定义的方程过多或不够,致使模型过约束或欠约束,从而无法求解.解决欠、过约束问题的常用方法是补全遗漏的方程或者剔除冗余的方程.然而,对于复杂物理系统仿真,模型包含的方程数目十分庞大,一个由几百个组件构成的模型通常可以产生上万个方程.当这样的复杂模型出现欠、过约束问题时,模型修正将是十分困难和费时的.但如果仿真建模工具能够采取适当策略,自动地把引起问题的根源限定在较小的范围内并提示给用户,那么模型的修正效率将大为提高.故深入研究相容性问题具有理论意义和实用价值.目前,只有 Peter Bunus 等人^[3,4]针对 Modelica 语言提出了一个调试构架,对相容性问题进行了初步研究,所提出的方法对简单模型比较有效,对复杂模型的分析效率较低.本文将对此展开进一步探讨.

1 Modelica 建模语言简介

在 Modelica 中,模型以类的形式存在,一个模型由变量与方程两类成员构成.变量表示模型的属性,通常对应于某个物理量;方程定义模型的行为,描述模型属性的数值约束关系.模型可以通过继承机制从基类模型中获得变量与方程.由于方程具有陈述式非因果特性,故 Modelica 模型通常是非因果的,也就是说,模型的输入、输出数据流在定义时没有明确地指定,直到模型求解时才能确定.这一特性不仅简化了建模工作,同时也提升了模型的重用性.

Modelica 将模型与外界的通信接口定义为连接器,并用一种称作连接类(connector)的特殊类来描述.连接类中只允许定义变量.这些变量用于实现模型的交互.它们被划分为两种类型:流变量与势变量.流变量是一种“穿过(through)”型变量,表示某种能量流,如电流、力、力矩等,用关键字 flow 限定.势变量是一种“跨越(across)”型变量,如电压、位移、角度等.两个相同类型的连接器可以关联到一起构成一个连接.在模型示意图中连接对应着组件之间的连线.连接本质上也是通过方程来表达变量耦合关系,因而在模型编译时,连接会转化为方程.连接中的势变量耦合关系转化为“等值”形式的方程,即两个势变量的值相等,流变量耦合关系转化为“和零”形式的方程,即两个流变量的和为 0.

在 Modelica 中模型的构建分两种方式:底层基本功能元件模型通过定义方程与变量直接创建;上层复合元件模型通过连接已有组件模型组合构成.图 1 给出了一个简单振荡器,它由质量块 M、弹簧 SP、阻尼器 Dm、作用力 F 和固定架 Fx 五个组件构成.组件模型 Mass, Spring, Damper, Force 和 Fixed 直接构建,系统模型 Oscillator 组合组件模型构成.限于篇幅,本文只给出系统模型 Oscillator 的 Modelica 代码.

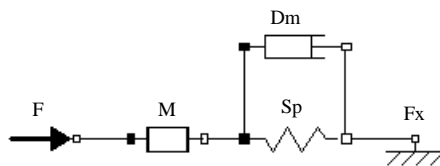


Fig.1 An oscillator model

图 1 振荡器模型

model Oscillator

Force F; Mass M; Spring Sp;
Damper Dm; Fixed Fx;

equation

connect(F,fa,M,fa); connect(M,fb,Sp,fa); connect(Sp,fa,Dm,fa);
connect(Sp,fb,Dm,fb); connect(Sp,fb,Fx,fa);

end Oscillator.

2 相容性分析

2.1 基本概念

定义 1. 如果无向图 $G = (V, E)$ 的顶点集合 V 能够划分为两个子集 V_1 和 V_2 , 并满足: $V_1 \cap V_2 = \emptyset, V_1 \cup V_2 = V$. 且对任意一条边 $e = (x, y) \in E$, 均有 $x \in V_1, y \in V_2$, 则称 G 为二部图^[5].

定义 2. 设 M 为图 G 的边集 E 的任意子集, 若 M 中任意两条边均不相邻, 则称 M 为图 G 的一个匹配. 匹配 M 中边的数目称为匹配 M 的基数. 具有最大基数的匹配称为最大基数匹配, 简称最大匹配.

定义 3. 设 v 是图 G 的一个顶点, 如果匹配 M 中存在边与 v 关联, 则称 v 是被 M 覆盖的. 没有被 M 覆盖的顶点, 称作自由顶点.

定义 4. 设 M 为图 G 的一个匹配, 如果 M 覆盖了 G 中的所有顶点, 则称 M 为图 G 的一个完美匹配.

定义 5. 设 P 为图 G 中的路径, M 为 G 的一个匹配, $E-M$ 表示 G 中不属于 M 的边的集合, 如果 P 的边交错地在 $E-M$ 和 M 中出现, 则称 P 为交错路径. 如果交错路径 P 的起点和终点均没有被不属于该路径的匹配边覆盖, 则称 P 为可行路径.

定义 6. 在有向无环图上, 从顶点 v 出发的所有路径上的顶点集合称为 v 的传播域. 可达顶点 v 的所有路径上的顶点集合称为 v 的先决域.

2.2 相容性判定方法

我们采用二部图 $G = (V_1, V_2, E)$ 来表示方程系统中方程与变量之间的约束依赖关系. 其中 V_1 表示方程集, V_2 表示变量集, 方程 $e \in V_1$ 与变量 $v \in V_2$ 之间的边表示变量 v 出现在方程 e 中. 二部图抽象地表示了方程与变量之间的依赖关系, 本文将这种二部图称作方程系统的约束表示图.

DM 分解(Dulmage-Mendelsohn decomposition)是 Dulmage 和 Mendelsohn 提出来的一种二部图规范分解方法^[6]. 该方法在某个最大匹配的基础上, 可将一个二部图 $G = (V_1, V_2, E)$ 分解成许多子图. 在这些子图中有两个子图比较特殊, 其中一个子图中属于 V_1 的顶点多于属于 V_2 的顶点, 另一个子图中属于 V_1 的顶点少于属于 V_2 的顶点.

文献[7]基于 DM 分解提出了一种判断参数化模型的欠、过和完整约束性的图论算法. 本文利用 DM 分解来判定仿真模型的相容性, 所分解出的两个特殊子图分别对应于模型的过约束部分和欠约束部分. 判定算法主要步骤如下:

步骤 1. 采用文献[8]中提出的方法找出约束表示图 $G = (V_1, V_2, E)$ 的一个最大匹配 M .

步骤 2. 在约束表示图 G 中, 将属于 M 的边变为双向边, 将不属于 M 的边定向为从方程指向变量, 最终生成有向图 $\bar{G} = (V_1, V_2, \bar{E})$.

步骤 3. 找出所有没有被最大匹配覆盖的方程, 计算它们的传播域, 并将传播域中的顶点从有向图中分离出来, 得到过约束子图 O_G^{k+} .

步骤 4. 找出所有没有被最大匹配覆盖的变量, 计算它们的先决域, 并将先决域中的顶点从有向图中分离出来, 得到欠约束子图 U_G^{k-} .

步骤 5. 在有向图 \bar{G} 中删除欠、过约束部分, 得到完整约束子图 W_G .

判定算法不但可以判定模型是否相容,而且能够分解出方程系统中的欠约束部分和过约束部分.算法的时间花费主要在第 1 步,其时间复杂度为 $O((m+n)\sqrt{m})$,其中 m 为表示图的顶点数, n 为表示图的边数.图 2 给出了一个 DM 分解示例,图中顶点 $e7$ 与 $v1$ 均为自由顶点.

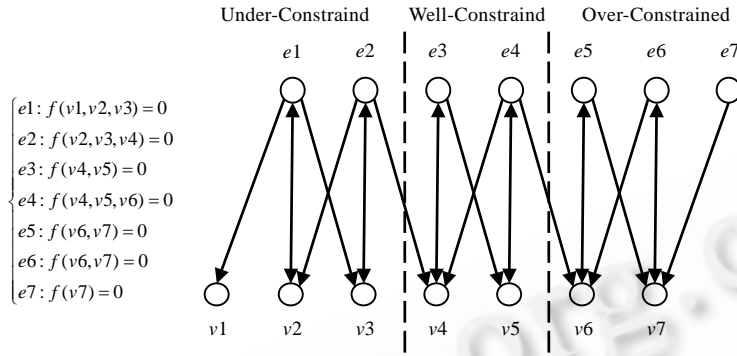


Fig.2 DM decomposition of a bipartite graph

图 2 二部图的 DM 分解

2.3 过约束处理策略

2.3.1 过约束方程的检测

为了得到一个过约束模型,我们对图 1 所示的振荡器模型进行修改.在组件模型 Spring 和 Damper 的基类模型 Compliant 中添加方程: $f=8$,在模型 Mass 中添加方程: $s=6$.修改后的振荡器模型 Oscillator 可转化为如下的微分代数方程组:

$e1: M.v = der(M.s)$	$v1: M.s$
$e2: M.a = der(m.v)$	$v2: M.v$
$e3: M.m * M.a = M.fa.f + M.fb.f$	$v3: M.a$
$e4: M.fa.s = M.s - M.L/2$	$v4: M.fa.s$
$e5: M.fb.s = M.s + M.L/2$	$v5: M.fb.f$
$e6: Sp.f = Sp.c * (Sp.s_rel - Sp.L)$	$v6: M.fb.s$
$e7: Sp.s_rel = Sp.fb.s - Sp.fa.s$	$v7: M.fb.f$
$e8: Sp.fb.f = Sp.f$	$v8: Sp.s_rel$
$e9: Sp.fa.f = -Sp.f$	$v9: Sp.f$
$e10: Dm.v = der(Dm.s_rel)$	$v10: Sp.fa.s$
$e11: Dm.f = Dm.d * Dm.v$	$v11: Sp.fb.f$
$e12: Dm.s_rel = Dm.fb.s - Dm.fa.s$	$v12: Sp.fb.s$
$e13: Dm.fb.f = Dm.f$	$v13: Sp.fb.f$
$e14: Dm.fa.f = -Dm.f$	$v14: Dm.s_rel$
$e15: F.f = 10 * \sin(2 * PI * 15.9155 * time)$	$v15: Dm.f$
$e16: F.fa.f = F.f$	$v16: Dm.v$
$e17: Fx.fa.s = 1.0$	$v17: Dm.fa.s$
$e18: F.fa.s = M.fa.f$	$v18: Dm.fa.f$
$e19: F.fa.f + M.fa.f = 0$	$v19: Dm.fb.s$
$e20: M.fb.s = Sp.fa.s$	$v20: Dm.fb.f$
$e21: Sp.fa.s = Dm.fa.s$	$v21: F.f$
$e22: M.fb.f + Sp.fa.f + Dm.fa.f = 0$	$v22: F.fa.s$
$e23: Sp.fb.s = Dm.fb.s$	$v23: F.fa.f$
$e24: Sp.fb.s = Fx.fa.s$	$v24: Fx.fa.s$
$e25: Sp.fb.f + Dm.fb.f + Fx.fa.f = 0$	$v25: Fx.fa.f$

- $e_{26}: M.s = 6$
- $e_{27}: Sp.f = 8$
- $e_{28}: Dm.f = 8$

在上述方程组中,方程 e_{26}, e_{27} 和 e_{28} 是我们特意添加的 3 个多余方程.将上述方程组表示成二部图,应用相容性判定算法对其进行分解,可得到如图 3 所示的过约束子图.在过约束子图中,双向边关联的方程 e 与变量 v 可以理解为由方程 e 求解变量 v 的值.如果某个方程不存在双向边与其关联,即没有被最大匹配覆盖,那么该方程是冗余的.图 3 中包含 3 个自由点 e_{12}, e_6 和 e_2 ,这表明振荡器模型存在 3 个冗余方程.要使模型变为完整约束系统,必须在过约束子图中删除 3 个自由点.但由于自由点的形成与最大匹配有关,而一个二部图的最大匹配通常是不唯一的,不同的最大匹配可能形成不同的自由点,从而使得过约束子图可能存在多种自由点组合.

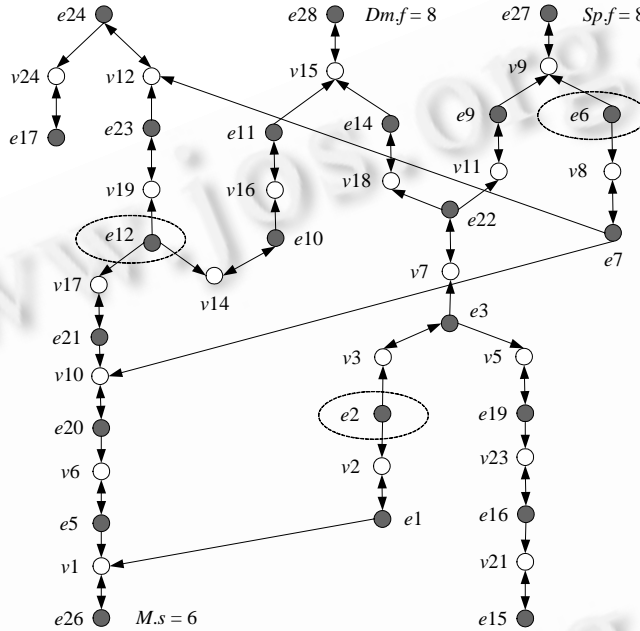


Fig.3 The over-constrained subgraph O_G^{3+}

图 3 过约束子图 O_G^{3+}

文献[9]提出了一种获取一个二部图的所有最大匹配的方法.该方法的基本思想是,在一个已有最大匹配的基础上,找出一条交错环路或长度值为偶数的可行路径,沿着该路径调整匹配关系,即将路径上的匹配边变为非匹配边,将非匹配边变为匹配边,从而获得一个新的最大匹配.沿着交错环路调整匹配关系得到的最大匹配形成的自由顶点与原匹配相同,沿着可行路径调整匹配关系得到的最大匹配形成的自由顶点与原匹配不同.本文只需找出所有形成不同自由点的最大匹配即可.

性质 1. 在过约束子图中,所有从自由点出发的路径均为交错路径.

性质 2. 在过约束子图中,所有从自由点出发,终止于任意方程的交错路径均是长度为偶数的可行路径.

从性质 2 可知,一个自由点的传播域中的所有方程均可以替代它成为自由点.在 O_G^{3+} 中,自由点 e_{12}, e_6 和 e_2 的传播域中的方程集合分别为

$$S_1 = \{e_{12}, e_{23}, e_{24}, e_{17}, e_{21}, e_{20}, e_5, e_{26}, e_{10}, e_{11}, e_{28}\},$$

$$S_2 = \{e_6, e_{27}, e_7, e_{24}, e_{17}, e_{20}, e_5, e_{26}\},$$

$$S_3 = \{e_2, e_1, e_{26}, e_3, e_{22}, e_{14}, e_{28}, e_9, e_{27}, e_{19}, e_{16}, e_{15}\}.$$

从集合 S_1, S_2, S_3 中各取一个元素,就可以构成过约束子图 O_G^{3+} 的一种自由点组合. S_1, S_2, S_3 的笛卡尔积,就是 O_G^{3+} 的所有可能的自由点组合,本文将其称作可能组合集,记为 PCS.但我们注意到,某些方程不只出现在一个

集合中,例如, e_{26} 既出现在 S_1 中,也出现在 S_2 和 S_3 中.组合 $\{e_{26}, e_{26}, e_{26}\}$ 显然不是 O_G^{3+} 的一种自由点组合.因此,我们必须判别 PCS 中每一种组合的有效性.有效性判别的基本思想是,在过约束子图 O_G^{3+} 中判断是否可以通过交换匹配关系使组合中的每一个方程均成为自由点.

通过有效性判别后,PCS 中还会存在一些重复组合.例如, $\{e_{24}, e_5, e_2\}$ 与 $\{e_5, e_{24}, e_2\}$, $\{e_{17}, e_{20}, e_3\}$ 与 $\{e_{20}, e_{17}, e_3\}$,它们包含同样的方程,只是方程在组合中出现的次序不同.接着,我们剔除 PCS 中的重复组合,并将剩下的 PCS 称作有效组合集,记为 VCS.

2.3.2 过约束方程的判定

VCS 中的每一种组合从数学上来说,都可以是一个有效的过约束方程组合,剔除组合中的方程会使仿真模型变成完整约束系统.然而,从建模的角度来说,有些方程具有特殊意义,它们是过约束方程的可能性很小,甚至不可能.例如,某个方程来自一个组件模型,该组件模型的相容性已得到证实,这样的方程肯定不是过约束方程.另一方面,模型的相容性分析是在方程系统上进行的,而用户对模型的修改只能在源代码上进行.然而,从源代码修改的角度来分析,有许多组合是不可以剔除的.为此,我们结合建模相关知识和 Modelica 语法规则,对 VCS 中的各个组合作进一步判定,将一些不可行的组合过滤掉.

条件 1. 如果方程组合 c 中的每个方程均不是来自标准模型,则称 c 为可剔除组合.

条件 1 所指的标准模型是指相容性已得到证实的模型.从 Modelica 标准模型库引入模型是标准模型,另外,某些已验证过的自定义模型也可以是标准模型.我们通过在建模仿真环境中设置重用模型库的方式来标识标准模型,用户可以将已验证过的自定义模型提交到重用模型库中.在模型转换时,编译器会将重用模型库和 Modelica 标准模型库中引入的模型认定为标准模型.

条件 2. 如果方程组合 c 满足条件:

- (a) 在过约束子图 O_G^{3+} 中,组合中的方程均没有被最大匹配覆盖,
- (b) 在二部图 G 中,剔除组合中的方程及其关联边,相应的无向图保持连通,

则称 c 为可剔除组合.

条件 2 要求一个前提,即仿真系统的数学模型对应的二部图是连通的.对物理系统建模仿真而言,这一点通常可以得到保证.因为仿真的目的是通过设置输入参数来观测系统属性值的变化.仿真模型中的每个属性变量要么直接或间接地决定其他属性变量,要么直接或间接地被其他属性变量所决定,整个模型的各个属性变量由输入输出数据流关联在一起,这使得其数学模型对应的二部图是连通的.

条件 3. 如果方程组合 c 满足条件:在仿真模型的 Modelica 源代码中删除或修改 c 中各方程对应的定义语句,不会导致有不属于 c 的方程被一起删除,则称 c 为可剔除组合.

条件 1 和条件 3 均需要获取方程的源代码信息.为了能在方程和源代码之间建立一种映射关系,借鉴文献[3]中的思想,我们为每个方程创建一个属性集,用它来记录方程的某些源代码信息.属性集包含如下属性:

- (1) 方程描述:数学模型中方程的表述形式.
- (2) 类名:表示方程由哪个类定义.
- (3) 定义语句:表示方程由哪条源代码语句定义.
- (4) 相关方程:这是一个集合,记录方程定义语句产生的所有方程.由于类之间存在继承和组合关系,因此一条定义语句可能产生多个方程.

(5) 柔性度:表示方程的相对重要性程度.它在 0~3 之间取整数值,值越小,柔性越低,表示方程越重要.本文规定:0 表示方程来自标准模型;1 表示定义该方程的类只定义了这一个方程;2 表示方程由连接产生;3 为缺省值.

方程属性集中各属性值的设置由编译器在模型转换过程中生成方程时自动完成.定义语句与方程的映射关系可以通过二部图来表示.在映射图中,方程与定义语句之间的关联边表示方程由该语句定义产生.

至此,我们可以构造一个判定算法,利用以上 3 个条件对 VCS 中的组合进行判定,将不满足条件的组合过滤掉.对每一个组合 c ,判定过程分如下 3 步:

步骤 1. 检查 c 中每个方程的柔性度,如果 c 包含柔性度为 0 的方程,则表明 c 不满足条件 1,将其从 VCS 中剔除;否则,继续判定.

步骤 2. 数学模型对应的二部图中,删除 c 中的方程及其关联边,测试图的连通性,如果图不连通,则表明 c 不满足条件 2,将其从 VCS 中剔除;否则,继续判定.

步骤 3. 根据定义语句与方程的映射关系,判断删除组合 c 的各方程的定义语句是否有不属于 c 的方程被一起删除,如果有,则表明 c 不满足条件 3,将其从 VCS 中剔除.

在示例模型中,组件模型 Force 和 Fixed 为标准模型,它们定义了方程 $e15, e16$ 和 $e17$. 条件 1 会将所有包含 $e15, e16$ 和 $e17$ 的组合过滤掉. 条件 2 和条件 3 也会过滤掉所有包含 $e3$ 、包含 $e19$ 、包含 $e24$ 的组合以及其他一些组合,这里不再一一列出. 最终 VCS 中还剩下 18 种组合,即还存在 18 种过约束剔除方案.

2.3.3 剔除方案的缩减

Modelica 模型通常是基于组件构成的,奇异的模型组件会导致整个系统模型奇异.也就是说,包含在过约束子图中的那些由相容组件产生的方程不是冗余的方程.为此,我们需要判定某些组件的相容性,这些组件中的某些方程出现在过约束子图中.

由于在 Modelica 模型中,组件之间的连接通常是非因果的,如果脱离组件的使用环境来单独判定一个组件的相容性,我们将无法确定关联在该组件上的连接产生的方程中有哪些应该包含到该组件对应的方程系统中.例如,在如图 1 所示的振荡器模型 Oscillator 中,弹簧 Sp 与质量块 Ms 之间的连接 $connect(Ms.fb, Sp.fa)$ 会产生两个方程: $Ms.fb.s = Sp.fa.s$ 和 $Ms.fb.f + Sp.fa.f = 0$. 如果方程 $Ms.fb.s = Sp.fa.s$ 的求解方向被确定为基于 $Ms.fb.s$ 求解 $Sp.fa.s$, 那么该方程应该包含到弹簧对应的方程系统中,否则,该方程应该包含到质量块对应的方程系统中. 方程 $Ms.fb.f + Sp.fa.f = 0$ 也是如此,它归属于哪些组件对应的方程系统取决于它的求解方向.

基于上述原因,在判定模型组件的相容性时,我们为连接器中的每个流变量添加一个虚构方程来替代那些由外部连接产生的方程.假设连接器类的一般形式如下:

```
connector generic_connector
  Real e;      //势变量
  flow Real f; //流变量
end generic_connector.
```

假设 $c1$ 和 $c2$ 是以上连接器类的两个实例.根据 Modelica 语义,连接声明 $connect(c1, c2)$ 会产生两个方程: $c1.e = c2.e$ 和 $c1.f + c2.f = 0$. 根据这两个方程,我们可得到一个新的方程 $c1.e * c1.f + c2.e * c2.f = 0$. 若假定 $c1.e * c1.f = C$, C 为某个常数,那么可得到 $c2.e * c2.f = -C$. 由于进行相容性判定只关注方程的结构信息,即方程包含哪些变量,并不关注它的具体表达形式,故可以用隐式方程 $f_1(c1.e, c1.f) = 0$ 表示方程 $c1.e * c1.f = C$, 用 $f_2(c2.e, c2.f) = 0$ 表示 $c2.e * c2.f = -C$. 然而,一个连接器类中可能存在多对匹配的流变量与势变量,故我们可以不失一般性地将为流变量添加的方程表示为隐式形式 $f(e_1, e_2, \dots, f_1, f_2, \dots) = 0$, 其中 e_i 表示势变量, f_i 表示流变量. 对于图 1 中弹簧 Sp 与质量块 M 之间的连接 $connect(M.fb, Sp.fa)$, 我们可以虚构两个方程 $f_1(M.fb.s, M.fb.f) = 0$ 和 $f_2(Sp.fa.s, Sp.fa.f) = 0$, 替代由该连接产生的两个方程 $M.fb.s = Sp.fa.s$ 和 $M.fb.f + Sp.fa.f = 0$. 其中, $f_1(M.fb.s, M.fb.f) = 0$ 是为流变量 $M.fb.f$ 添加的, $f_2(Sp.fa.s, Sp.fa.f) = 0$ 是为流变量 $Sp.fa.f$ 添加的.

在图 1 所示振荡器模型中,组件 M, Sp 和 Dm 均有方程出现在过约束子图中.通过用虚构方程替代由相应连接产生的方程,可以得到每个组件对应的方程组,采用第 2.2 节提出的判定方法,我们可以判定出 M, Sp 和 Dm 均是过约束的,每个组件中各存在一个冗余方程.这一结果表明,过约束方程组合中的 3 个方程应该分别由组件 M, Sp 和 Dm 定义产生.据此验证 VCS 中剩下组合的可行性,将不可行的组合删除.然后将剩下的组合根据柔性度(组合的柔性度为组合中方程的柔性度的最小值)从高到低排序,并结合方程属性将剔除方案映射为源代码修改信息提供给用户.最终得到如下 6 种剔除方案,方案 4 中的方程为我们此前特意添加的冗余方程.

- (1) $\{e1, e27, e28\}$: 从 Mass 中删除定义语句 $v = der(s)$; 从 Compliant 中删除定义语句 $f = 8$.
- (2) $\{e2, e27, e28\}$: 从 Mass 中删除定义语句 $a = der(v)$; 从 Compliant 中删除定义语句 $f = 8$.
- (3) $\{e5, e27, e28\}$: 从 Mass 中删除定义语句 $fb.s = s + L/2$; 从 Compliant 中删除定义语句 $f = 8$.
- (4) $\{e26, e27, e28\}$: 从 Mass 中删除定义语句 $s = 6$; 从 Compliant 中删除定义语句 $f = 8$.

(5) $\{e6, e10, e26\}$:从 Spring 中删除定义语句 $f = c * (s_rel - L)$;从 Damper 中删除定义语句 $v = der(s_rel)$;从 Mass 中删除定义语句 $s = 6$.

(6) $\{e6, e11, e26\}$:从 Spring 中删除定义语句 $f = c * (s_rel - L)$;从 Damper 中删除定义语句 $f = d * v$;从 Mass 中删除定义语句 $s = 6$.

2.4 欠约束处理策略

修正欠约束模型的常用方法是为模型中的每个自由变量添加一个方程,并要求新添的方程包含对应的自由变量.在欠约束子图中存在多少个自由变量点,就需要为模型增加多少个方程.与处理过约束问题类似,通过判定与欠约束子图相关的每个组件的相容性,我们可以找出模型中的不相容组件,即欠约束组件,并且能够确定出每个欠约束组件缺少多少个方程,据此提示用户为这些组件添加方程.本文在此不再举例分析.

3 结 语

模型相容是模型可以仿真求解的一个必要前提,因而相容性分析是陈述式模型求解预处理过程中的一个重要环节,它对于复杂系统模型尤为重要.本文针对 Modelica 语言对这一问题进行了深入研究,提出了一套有效的分析策略,并给出了相应算法.本文的主要贡献在于:(1) 可以有效地判定模型的相容性,分解出模型中过约束与欠约束部分;(2) 能够自动判别出过约束或欠约束问题发生的大致范围,并为用户提供修正提示,极大地提高了模型的修正效率;(3) 对其他陈述式仿真模型的相容性分析具有借鉴与参考价值.与文献[3,4]中提出的方法相比,本文的方法效率更高,特别是对复杂模型.对于文中给出的过约束问题,采用文献中提出的方法进行分析会得到 46 种剔除方案,而采用本文的方法只有 6 种.本文提出的分析策略及相应算法已在多领域物理系统混合建模与仿真平台 MWorks 中得以实现,效果良好.

References:

- [1] Xiong GL, Fan WH, Chen XB. Simulation technology of complex product development. *Journal of System Simulation*, 2004,16(2): 194–201 (in Chinese with English abstract).
- [2] Fritzson P, Engelson V. Modelica—A unified object-oriented language for system modeling and simulation. In: Eric J, ed. *Proc. of the 12th European Conf. on Object-Oriented Programming*. Brussels: Springer-Verlag, 1998. 67–90.
- [3] Bunus P, Fritzson P. Methods for structural analysis and debugging of modelica models. In: Otter M, ed. *Proc. of the 2nd Int'l Modelica Conf. Oberpfaffenhofen: Modelica Association*, 2002. 157–165.
- [4] Bunus P, Fritzson P. Semantics guided filtering of combinatorial graph transformations in declarative equation-based languages. In: Kawada S, ed. *Proc. of the 2nd IEEE Int'l Workshop on Source Code Analysis and Manipulation*. Montreal: IEEE Computer Society, 2002. 163–172.
- [5] Asratian AS, Denley T, Haggkvist R. *Bipartite Graphs and their Applications*. Cambridge: Cambridge University Press, 1998. 7–15.
- [6] Pothen A, Fan CJ. Computing the block triangular form of a sparse matrix. *ACM Trans. on Mathematical Software*, 1990,16(4): 303–324.
- [7] Jiang K, Gao XS, Yue JY. A decision method for under-, over- and well-constrainsness of parametric model. *Journal of Software*, 2003,14(12):2092–2097 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/2092.htm>
- [8] Hopcroft JE, Karp RM. An $n^2/2$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal of Computing*, 1973,2(4): 225–231.
- [9] Takeaki U. Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs. In: Leong HW, Hiroshi I, Sanjay J, eds. *Proc. of the 8th Int'l Symp. on Algorithms and Computation*. Singapore: Springer-Verlag, 1997. 92–101.

附中文参考文献:

- [1] 熊光楞,范文慧,陈晓波.复杂产品开发的仿真技术. *系统仿真学报*,2004,16(2):194–201.
- [7] 蒋鲲,高小山,岳晶岩.参数化模型欠、过和完整约束的判定算法. *软件学报*,2003,14(12):2092–2097. <http://www.jos.org.cn/1000-9825/14/2092.htm>