

连续系统仿真语言 MSL

陈友君 (中国科技大学管理学院)

夏双金 (中国科技大学研究生院)

摘要: MSL 仿真语言系统引入了降阶技术,从而允许用户以微分方程的原始形式作为仿真模型直接写入仿真程序;语法制导编辑技术的采用打破了常规的高级语言源程序编辑和编译的过程,使程序的语法和语义的正确性在程序一经输入即得以保证;MSL 使仿真的各概念清晰,实现了实验与模式显式分开,支持多模型,子模型和自定义函数等概念,使语言仿真描述模型能力强、逼真。

一、引言

所谓仿真,就是用模型代替实际系统进行实验和研究,计算机仿真则是在计算机上构造与实际系统相符合的模型来进行实验和研究。因此,宇宙飞船、卫星、导弹、轮船和核反应堆等,凡是需要用模型进行实验的,都可以用计算机仿真来研究系统本身的各种特性,选择最佳参数和设计最合理的系统方案。对于社会经济等非工程领域,若采用直接实验方法,有时后果将不堪设想,为此仿真技术则成为研究这些领域内问题的必要手段。

我们研制的 MSL 是一种全新的连续系统仿真语言,它的研制得到了中国科学院 MTIPS 项目的资助。其主要特点是:

1. MSL 是一种使用户可以直接向系统输入仿真模型以及控制运行程序从而获得所需数值解或图表的高层软件。它溶入了微分方程的降阶技术,因此可接受与数学表示完全一致的仿真模型,从而免去任何人工进行的常规降阶变换工作;

2. MSL 系统打破了传统的计算机解题所需的编辑和编译步骤。我们所采用的先进的语法制导编辑策略体现了仿真源程序编辑、修改、编译过程的全交互特点与高效率。系统利用用户输入源程序慢,时间长的特点,有效而周到地为用户在输入源程序过程中就进行词法和语

法检查服务,对病句指出错误的类型和位置。引导用户立即修改。这样,用户输入完源程序就意味着程序的词法分析和语法分析已经结束,高效地获得了正确程序,从而大大方便了用户。

3. 模型与实验显式分开,仿真概念清晰,模型管理简便。系统允许定义多个模型及多个实验,可以将一个实验作用到不同模型上,也可以将不同的实验作用到不同的模型上;并且允许定义用户函数,允许定义子模型;模型可由子模型构成,系统所提供的这些手段为用户精确地仿真带来了极大方便。

二、MSL 仿真语言概貌

用 MSL 语言编定的仿真源程序可含有五种类型的块:函数块 Function、子模型块 Submodel、模型块 model、实验块 experiment 和控制块 controlstructure。其中模型块和控制块是必须有的,其它块可以为空。

函数块为用户提供了自定义具有特殊功能函数的手段,自定义函数既可以是一般函数,也可以是列表函数。利用列表函数,用户可使用线性内插公式对自变量的每个值产生分段线性近似值。不仅如此,MSL 还对双变量函数提供了同样的便利。

基于客观世界实际系统中的大系统一般由多个子系统构成的特点,MSL 系统首先提供模型块和子模型块,允许用户根据问题需要在模型块中定义模型(甚至允许定义多个),在子模型块中定义一至多个子模型。在语法形式上,子模型定义是模型定义的子集。进而采用允许

* 本项目属多目标集成化可移植软件环境——MTIPS 的子课题,MTIPS 已获 1993 年中科院科技进步一等奖。

模型调用子模型的机制实现模型由多个子模型构成的思想。从而便客观世界的大系统得到精确仿真。为实现模型与子模型的信息交换,MSL 设置了控制块,在该块作用下通过参数传递完成这一功能。

实验块定义模型运行的环境。某些特定的变量如 T (独立变量),Integralmethod(积分算法)等是事先定义好的,并具有隐含初值。在实验块,用户可修改这些变量的值。为了实验的灵活性,实验可以有一些参数传入,从而提高了实验的通用性。MSL 允许用户定义多个实验,这些实验可作用到同一个模型,也可作用到不同的模型上。

控制块是仿真运行的逻辑控制部分,MSL 在此允许用户使用各种程序语句,如赋值语句,条件语句和循环语句等,其中循环语句包括 for 语句、repeat 语句和 while 语句。

特别地,该块还提供了一条将实验和模型连接的语句 commit。

commit <实验名>(参数表)on <模型名>(参数表)

它使由 <模型名> 指定的模型在 <实验名> 所指定的实验环境下运行一次。

限于篇幅,本文用形式化和非形式化说明相结合的方法描述 MSL 语言,从而使读者对后面的例子有个大致的了解。

<程序> ::= program <程序名> <程序体>
> Endprogram

<程序体> ::= { <函数块> }
 { <子模型块> }
 <模型块> { <模型块> }
 <实验块> }
 <控制块>

<函数块> ::= <一般函数块> | <列表函数块>

<列表函数块> ::= Tabular Function <列表函数名>

(<数据行>, <函数形参表>) : <

类型>

begin <数据语句>

 <插值语句>

end

<子模型块> ::= SubModel <子模型名>
 <子模型体>

EndsubModel

<子模型体> ::= <静态结构>

 <动态结构>

 <初始条件>

<模型块> ::= Model <模型名>

 <模型体>

 EndMidel

<模型体> ::= <静态结构>

 <动态结构>

 <初始条件>

 <数据收集段>

 [<终止段>]

<静态结构> ::= staticstructure

 <参数说明>

 <输出变量说明>

 <状态变量说明>

 <辅助变量说明>

Endstaticstructure

<动态结构> 是对仿真模型方程的描述,模型方程分为微分方程、输出方程和子模型调用方程三种。即:

<动态结构> ::= Dynamic <方程组>

<方程组> ::= <方程> { <方程> }

<方程组> ::= <微分方程> | <输出方程> | <子模型调用方程>

<初始条件> 描述微分方程的初始条件。即:

<初始条件> ::= <初值条件> | <边值条件>

<数据收集段> 给出一个收集数据的条件,当满足条件时,该段才执行,以减少计算和输入输出,<终目段> 描述一次仿真运行结束条件,当该段缺省时,仿真运行到终止时间。

<实验块> ::= Experiment <实验名>

 <实验体>

 End Experiment

<实验体> ::= begin <时间设置语句表>

 <跟踪语句>

 <积分法选择句>

 end

<控制块> ::= Controlstructure <说明表>

<复合句>

Endcontrolstructure

三、仿真语言的实现

MSL 系统主要包括文件管理、模板编辑器、编译、链接运行和仿真结果输出子系统,其系统结构如图 1:

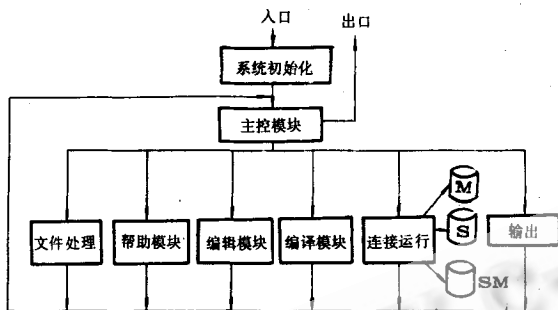


图 1

其中, M 为 MSL 仿真数学库; S 为求解微分方程的数学软件库; SM 为标准数学库。

现将系统的主要部分作一介绍:

1.MSL 编辑器

MSL 系统提供给用户的编辑程序是语法制导的,它具有 MSL 语言的语法知识,因而仿真程序一经编辑其语法就正确了。不仅如此,它还能够对程序的语义作一些检查,并且可以比较方便地变化语义检查的范围。在编辑仿真程序时,MSL 编辑器先产生一个程序的模板,用户对模板上的结构不断地扩展进行和填充,从而完成程序的编辑过程。模型的编辑的模板如下所示,其中以开头的是待扩展结构。

```
MODEL $ mdel-id
      $ Staticstructure
      $ Dynamic
      $ Initial / Boundary
      $ datacollection
      $ Terminal
```

ENDMODEL

这种编辑方式对初级用户和老用户边思考边编辑的场合都是非常有效的。但是,由于表达式类的结构的层次很深,用这种方式编辑显得十分繁琐。为此,MSL 编辑器提供了表达式类的局部正文编辑。也就是说,在不放弃

语法制导的原则下,在扩展表达式结构时可在上述编辑方式和正文编辑方式之间转换。

2.MSL 编译器

MSL 编译器实际上是一个翻译器,它将 MSL 仿真程序翻译成 Fortran77 代码源程序,继而再通过标准的 Fortran 编译程序和链接程序进行编译和链接,便产生了运行速度快的可执行代码。本编译器的特点是:引入了微分方程的降阶技术,从而使用户输入模型与实际数学微分方程一致,易于理解,仿真程序可读性好;既可自动获得微分方程组的 Yacobi 矩阵,也可对边值问题的边值条件进行相应处理;对实验块中的参数实现了动态修改;编译生成的目标代码已与现成的 Fortran 数学程序库连接起来。

3.输出子系统

MSL 的仿真结果处理构成了该子系统,它包括:列表显示、列表打印、二维绘图显示、二维绘图打印、三维绘图。对于列表显示和打印,MSL 允许取出不同数据文件(可达 10 个)中的不同变量以表格的形式输出,既可以间隔显示或打印,也可以控制显示或打印的长度。二维绘图对仿真结果的处理十分有效,它可选择横轴,绘制任意两个变量之间的曲线关系,以直观地分析曲线的趋势等特性。最大同时处理的文件个数为 10。

4.与数学库的连接

MSL 系统采用的积分数学软件库是 STYR / MATH 的 D 章(STYRD)。STYR / MATH 是国家“六·五”科技攻关项目,由中科院计算中心等八个单位经三年多的开发而完成的。STYRD 提供的积分算法有:

初值问题

- DGEAR: 变阶 Adams 预求法
- DREBS: 外插法
- DVERK: R-K-V 五阶和六阶方法

边值问题

DTPTB: 多点打靶法解常微分方程(组)两点边值问题。

虽然选用了 STYRD 这一数学库,但由于取出了微分方程(组)的所有特征量,所以,MSL 系统并不是限制在这一数学库上,事实上,它同样可方便地利用别的积分数学库。

例: MSL 仿真程序 1

```
!An example of msl program
PROGRAM exampleofharmonic
```

注释句

```

SUBMODEL tvarf
SUBMODEL tvarf
  Staticstructure
    Parameter Real init;
    Output Real force;
    State Real force;
  EndStaticstructure
  Dynamic
    d[1]force / d[1]t = sin(2 * t);
  Initial
    force = init;
  ENDSUBMODEL

MODEL os
  Staticstructure
    Parameter Real subinit,xinit,dxinit;
    State Real x;
    Auxiliary Variable Real extraforce;
  Endstaticstructure
  Dynamic
    d[2]x / d[2]t = 2 * d[1]t - 13 / 4 * x + extraforce;
    extraforce <- tvarf(subinit);
  Initial
    x = xinit;
    d[1]x / d[1]t = dxinit;
  DisplayCondition
    DataCollection
      Title(t,x,extraforce);
      Save(t,x,25000.0 * extraforce);
  terminal condition
  ENDMODEL

EXPERIMENT general
Begin
  Tbeg = 0.0; Tend = 13; Tdeltaa = 0.01
  IntegralMethod1;
ENDEXPERIMENT

CONTROLSTRUCTURE
Var Realsubinit,xinit,dxinit;
Begin
  subinit = 1.0
  xinit = 1.0
  dxinit = 0.0
  commit general on os(subinit,xinit,dxinit);
ENDCONTROLSTRUCTURE

ENDPROGRAM
    
```

MSL 仿真程序实例

这个程序包含了 MSL 语言的许多特性(已在源程序清单上---注明)。我们注意到, PROGRAM 和 ENDPROGRAM 是程序括号,遵循先定义后引用的原则,由模型调用的子模型先定义。在本例中,只有唯一的

子模型 tvarf,是在子模型块中定义的;在模型块中定义的模型为 OS;接着在实验块中定义了实验 general,它描述了模型运行的条件;最后是仿真控制块 CONTROLSTRUCTURE,其中语句。

commit general on os(subinit,xinit,dxinit)

表示在实验 general 的指定条件下运行模型 OS,而 subinit,xinit,dxinit 则是模型 OS 的实在参数。

模型 OS 的定义由五部分组成:静态结构 staticstructure、动态结构 Dynamic、初始条件 Initial、数据收集 Datacollection 和仿真终止(结束)Terminal。动态部分定义了数学模型,这是一个二阶的系统,方程为

$$X'' = 2 * X' - 13 / 4 * X + \text{extraforce}$$

已无需降阶而直接写入模型块,它研究此系统在外加作用力 extraforce 驱动下的行为,而 extraforce 是由子模型 tvarf 产生的,语句

extraforce --- tvarf(subinit)

表示子模型作用于实参 subinit 得到子模型的输出 extraforce。

例:MSL 仿真程序二

这是一个三阶伺服系统的仿真,其控制系统的框图如图 2 所示。

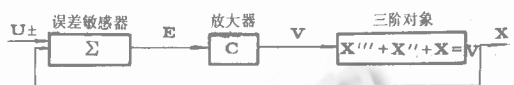


图 2 一个三阶控制系统的框图

其被控对象的微分方程为

$$X''' + X'' + X' = V \quad (T > 0)$$

增益为 C 的误差放大器用来实现闭环控制,使对象的输出 X 跟踪系统的输入 U,即

$$V = C(U - X)$$

下面给出的是求五种 C 值(c = 0.25,0.30,0.35,0.40,0.45)下阶跃响应的程序。

A servo system analysis

PROGRAM step

MODEL stm

```

StaticStructure
  Parameter Real c1,c2,c;
  State Real x;
  AuxiliaryVariable Real e,u;
Endstaticstructure
    
```

```

Dynamic
  lc is gain constant
  dz[3]x / d[3]t = -d[2]t - d[1]x / d[1]t + c * e;
!form error
e = u - x;
!general ramp or step input
u = c1 + c2 * t;
Initial
  x = 0.0;
  d[1]x / d[1]t = 0.0;
  d[2]x / d[2]t = 0.0;
Displaycondition true
  DataCollection
    title(t,x);
    save(t,x);
!Terminal
ENDMODEL

EXPERIMENT ste
Begin
  tbeg = 0.0; tend = 20.0; tdelta = 0.01;
ENDEXPERIMENT

CONTROLSTRUCTURE
Var
  Real c1,c2,c;
Begin
  c1 = 1.0; c2 = 0.0;
  for c = 0.2 to 0.45 by 0.05 do
    Commit ste on stm(c1,c2,c);
ENDCONTROLSTRUCTURE
ENDPROGRAM
    
```

三阶伺服系统的阶跃响应程序

四、结束语

在 MSL 系统的研制中我们考虑了以下设计原则:

1.可扩充性

在设计 MSL 系统时充分考虑了 MSL 系统提供的函数以及输入输出子程序可能满足不了各类专业用户的要求,因此允许用户将自定义函数和自编子程序加入到系统中;

2.可适应性

整个系统以 C 语言实现(数学库用 FORTRAN77 实现),可以方便地移植到不同的机型上;

3.可实用性

不同熟悉程度的人员可以在不同级别上运用语言的功能,对于一个几乎没有高级语言编程经历的用户编制,MSL 仿真程序也没有太大的困难。

MSL 系统还提供了不同层次的操作方式。对初学者,系统提供了窗口和菜单的工作方式,对用户的每一步操作都给出提示;对老用户则提供了行命令的操作方式。

随着人们所涉及的系统规模日益增大,用实际系统作实验开销庞大,而且往往是不可能的,因此计算机仿真已是系统分析与设计的重要手段。MSL 仿真语言的研究成功,对于非计算机专业的科学家和专业人员分析,研制和开发系统,并且十分有效的,参考文献:(略)