

The Simulink-Based Box Diagram Solution to Delay Differential Equations

Bangxing Cao, Liubin Hu

Sontan College, Guangzhou University, Guangzhou Guangdong
Email: 478360653@qq.com

Received: Feb. 6th, 2019; accepted: Feb. 20th, 2019; published: Feb. 27th, 2019

Abstract

Being one of the most commonly-used tools in describing the dynamic system, the delay differential equation contains not only the current value of the signal but also the previous one. However, it cannot be directly solved by writing anonymous functions with Matlab if the delay differential equation is only one part of the complex system, and its input signal comes from the previous module; therefore, the simulation strategy on the basis of the box diagram is the best way to solve such problems. Based on the box diagram, Matlab Simulink is the ideal tool, which helps build the delay differential equation model. By this way, we can solve the numerical solution of this equation.

Keywords

Delay Differential Equation, Matlab, Transfer Function, Simulink Environment, Numerical Solution of Differential Equations

基于Simulink的延迟微分方程框图求解

曹邦兴, 华柳斌

广州大学松田学院, 广东 广州
Email: 478360653@qq.com

收稿日期: 2019年2月6日; 录用日期: 2019年2月20日; 发布日期: 2019年2月27日

摘 要

微分方程是描述动态系统的最常用工具之一; 延迟微分方程除包含有信号当前时刻的值外, 还含有信号以前的值; 若延迟微分方程只是复杂系统中的一部分, 其输入信号来自于前一个模块, 则无法用MATLAB

编写匿名函数来直接求解; 基于框图的仿真策略是解决这类问题的最好方法; Simulink是MATLAB下基于框图仿真方法的理想工具; 用Simulink搭建延迟微分方程模型, 并求出方程的数值解。

关键词

延迟微分方程, Matlab, 传递函数, Simulink环境, 微分方程数值解

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

微分方程是经济学、统计学、工程、科学计算等领域数学建模的基础, 也常常被用来描述动态系统, $x'(t) = f(t, x(t))$ 是微分方程的一般形式, 其全部信号都来自于时刻 t 的当前值。假若微分方程中既有时刻 t 的当前信号, 也有早先时刻的信号值, 这种微分方程就被称作延迟微分方程[1]。

2. 典型延迟微分方程的数值求解

延迟微分方程的一般形式为

$$x'(t) = f[t, x(t), x(t-\xi_1), x(t-\xi_2), \dots, x(t-\xi_n)],$$

式中 ξ_i 表示延迟常数, 要求其为非负数且对应于 $x(t)$ 状态变量。与普通微分方程的差别在于, 此处除了有当前时刻 t 的信号外, 还有先前时刻的信号。因此延迟微分方程(组)的确定除必须有状态向量 x 和常规标量 t 外, 还需设定 \mathbb{Q} 矩阵, \mathbb{Q} 中第 j 列的列向量 $\mathbb{Q}(:, j)$ 对应于延迟时间向量 ξ_j

$$\text{例 1 设 } \begin{cases} 4x(t) = 2y(t) + 3y'(t) + y''(t) \\ x'(t) = 0.2x^3(t-0.5) - x(t-0.5) - 3x(t) - y(t-1) - 1 \end{cases},$$

且当 $t \leq 0$ 时 $x(t) = y(t) = y'(t) = 0$, 试求解方程组。

解: 方程组中包含了 t , $t-1$, $t-0.5$ 时刻的信号值, 故必须设计专门的算法和编写特定的程序来求解该延迟微分方程组, 同时要转换成显式一阶方程组。引进一组状态变量 $x(t) = x_1(t)$, $y(t) = x_2(t)$, $y'(t) = x_3(t)$ 是实现上述转换最便捷高效的方法。原方程组变形为:

$$\begin{cases} x_3'(t) = 4x_1(t) - 2x_2(t) - 3x_3(t) \\ x_2'(t) = x_3(t) \\ x_1'(t) = 0.2x_1^3(t-0.5) - x_1(t-0.5) - 3x_1(t) - x_2(t-1) - 1 \end{cases}$$

再引入时间常数 $\xi_1 = 1$ 对应状态变量 x_2 、 $\xi_2 = 0.5$ 对应变量 x_1 , 用 MATLAB 编写匿名函数描述延迟微分方程, 调用 MATLAB 语句使零向量对应常数初始值, 在 MATLAB 运行程序即可得出该方程组的数值解[2], 如图 1 所示。

3. Simulink 环境简介

假设一个系统能够用微分方程来进行描述, 则使用前文介绍的求解函数就可以直接得出解。但如果微分方程仅仅是复杂系统中的一部分, 输入信号本身由前一个模块传入, 整个系统模型又无法用单独一个方程(组)来描述, 这种情况下前面使用的求解函数则无能为力。

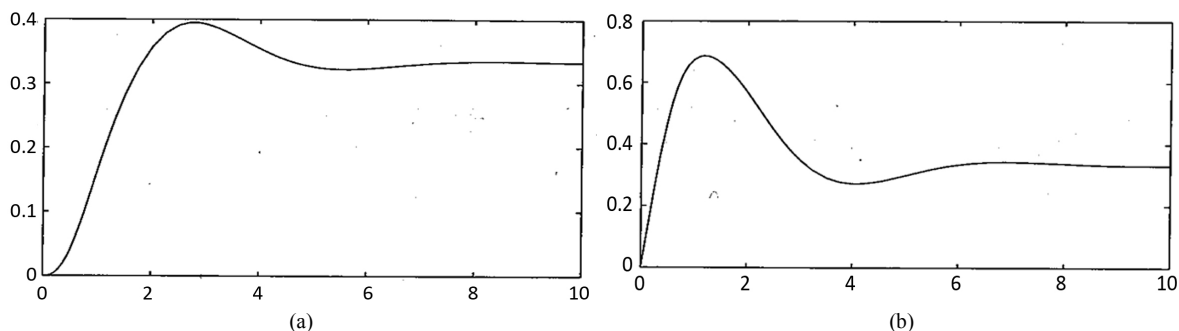


Figure 1. (a) The numerical solution of the delay differential equation (The initial condition is zero); (b) The numerical solution of the delay differential equation (The initial condition is non-zero)

图 1. (a) 延迟微分方程的数值解(初始条件为零时); (b) 延迟微分方程的数值解(初始条件为非零时)

解决这个问题最有效途径就是仿真策略并基于框图实现。Matlab 下的 Simulink 环境就是基于框图仿真策略的理想工具。

Math Works 公司出品的 Simulink, 名字中 Simu 代表仿真, link 表示连接, 合起来的意思就是连接输入输出信号并且用框图来对某个系统进行仿真。求解微分方程(组)只是其众多的强大功能之一。

Simulink 下支持的模块很多, 如信号源输入模块组、自定义函数组等, 如图 2 表示的常用模块组中可以自定义的模块组。

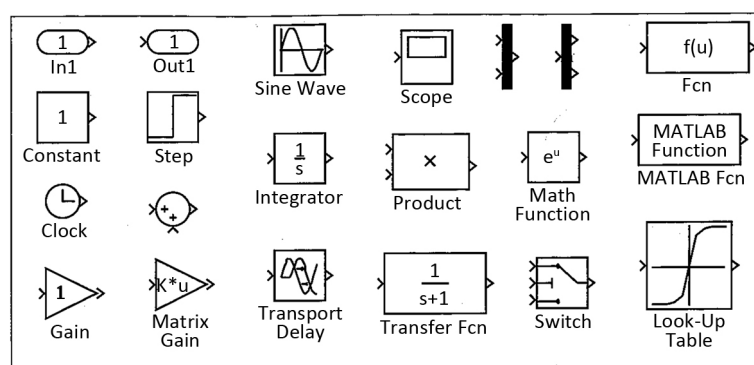


Figure 2. Commonly-used custom module groups

图 2. 常用模块组中可以自定义的模块组

如图中 Transport Delay 表示延迟模块, 本文正是用该模块来实现延迟微分方程的仿真建模与求解。

4. 延迟微分方程的 Simulink 建模

Simulink 的 Continuous 组中提供了几个延迟模块, 如图 3 所示, 用于提取信号的延迟信息。

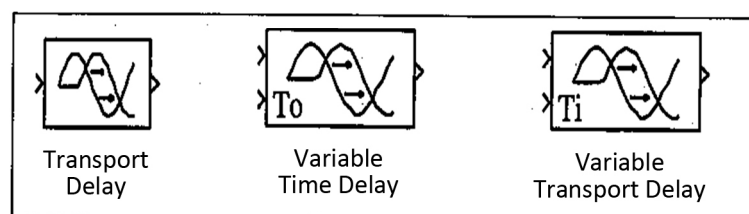


Figure 3. The Various delay modules provided by continuous function groups in Simulink

图 3. Simulink 的 Continuous 组提供的各种延迟模块

例如, Transport Delay 模块描述的是传递函数 e^{-Ls} , L 为延迟时间常数, 若模块的输入信号为 $u(t)$, 则其输出为 $u(t-L)$; 若 Variable Time Delay 模块第二输入信号为 t_0 , 则输出信号为 $u(t-t_0)$, 而 Variable Transport Delay 模块是建立在 Variable Time Delay 模块基础上的可变传输延迟, 第二端口为传输延迟 t_i , 由该值推算出一个等效的时间延迟 t_d , 则模块的输出信号为 $u(t-t_d)$, 其中 $\int_{t-t_d}^t \frac{1}{t_i(\xi)} d\xi = 1$ 。下面将通过例子演示框图构建与仿真方法。

例 2 考虑例 1 中的方程组, 重写如下:

$$\begin{cases} 4x(t) = 2y(t) + 3y'(t) + y''(t) \\ x'(t) = 0.2x^3(t-0.5) - x(t-0.5) - 3x(t) - y(t-1) - 1 \end{cases}$$

已知 $t \leq 0$ 时 $x(t) = y(t) = y'(t) = 0$, 用 Simulink 对该延迟微分方程建模并仿真求解。

解: 将方程式二中的 $-3x(t)$ 移到等式左边得

$$3x(t) + x'(t) = 0.2x^3(t-0.5) - x(t-0.5) - y(t-1) - 1$$

表示在等号右边信号的激励下, $x(t)$ 传输函数模型 $1/(s+3)$ 的信号输出; 另一个方程式表示在 $y(t)$ 信号激励下, $x(t)$ 传递函数模块 $4/(s^2+3s+2)$ 的信号输出。

再在 $x(t)$ 、 $y(t)$ 上接入延迟模块 Transport Delay, 就可以生成这些信号的延迟。如此构建出如图 4 所示的 Simulink 仿真模型[3]。

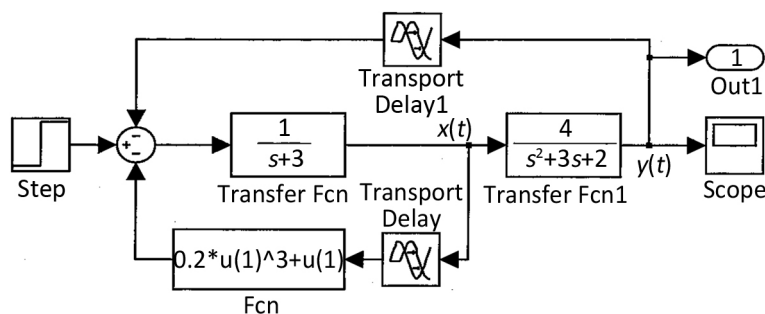


Figure 4. Simulink Modeling of the delay differential equation

图 4. 延迟微分方程的 Simulink 建模

再调用语句

```
>> [t,x] = sim('c7mdde2',[0,10]);
plot(t,x); %绘图显示仿真结果
得到与上文例 1 中图 1(a)、图 1(b)完全相同的曲线。
```

5. 延迟微分方程的 Simulink 模型仿真求解

例 3 已知延迟微分方程

$$x'(t) = M_1 \cdot x \cdot (t-0.15) + M_2 \cdot x' \cdot (t-0.5) + K \cdot u(t),$$

其中, 输入信号 $u(t) \equiv 1$, 且已知矩阵为

$$M_1 = \begin{bmatrix} -13 & 3 & -3 \\ 106 & -116 & 62 \\ 207 & -207 & 113 \end{bmatrix}, M_2 = \begin{bmatrix} 0.02 & 0 & 0 \\ 0 & 0.03 & 0 \\ 0 & 0 & 0.04 \end{bmatrix}, K = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix},$$

试用 Simulink 搭建系统模型, 并求出该方程的数值解。

解: 因为方程中同时包含 $x'(t)$ 和 $x'(t-0.5)$ 项, 所以单纯采用函数直接求解是无能为力的。可以采用 Simulink 仿真模型来求解。在构建框图前, 在 Matlab 中先通过语句来生成已知矩阵,

$$>> M1 = [-13, 3, -3; 106, -116, 62; 207, -207, 113];$$

$$M2 = \text{diag}[0.02, 0.03, 0.04];$$

$$K = [0, 1, 2];$$

因为状态向量 $x(t)$ 已经存在, 只需添加一个积分器就可以构成输入端 $x'(t)$ 、输出端 $x(t)$, 之后给这两端信号接入延迟环节, 设置适当的延迟时间常数, 这样经过一系列处理就构建出如图 5 的 Simulink 框图模型。

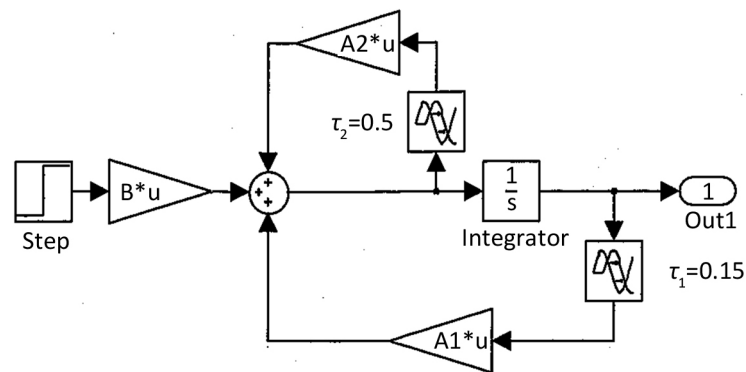


Figure 5. Simulink Modeling with the delay differential equation
图 5. 带有导数延迟的微分方程的 Simulink 模型

再调用程序

```
>> [t,x] = sim('c7mdde3',[0,8]);
```

```
plot(t,x); %将仿真结果绘图
```

得仿真解如图 6 所示[4]。

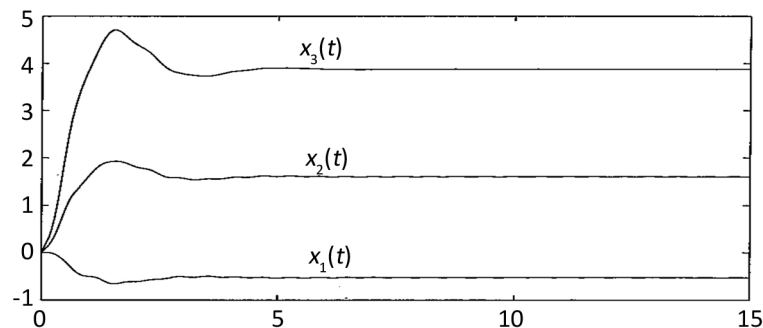


Figure 6. The Simulated experimental result of the delay differential equation
图 6. 延迟微分方程的仿真实验结果

例 4 已知

$$\begin{cases} x_1'(t) = -2x_2(t) - 3x_1(t - 0.2|\sin t|) \\ x_2'(t) = -0.05x_1(t)x_3(t) - 2x_2(t - 0.8) + 2 \\ x_3'(t) = 0.3x_1(t)x_2(t)x_3(t) + \cos[x_1(t)x_2(t)] + 2\sin(0.1t^2) \end{cases}$$

各状态变量的初始条件均为零。试用仿真框图求解方程组。

解：要定义状态变量向量 $x(t)$ ，也需要添加一个向量型积分器，由信号

$$x(t), x_1(t-0.2|\sin t|), x_2(t-0.8), t$$

等构成 Mux 模块的向量输出，得图 7 所示的仿真框图[5]。

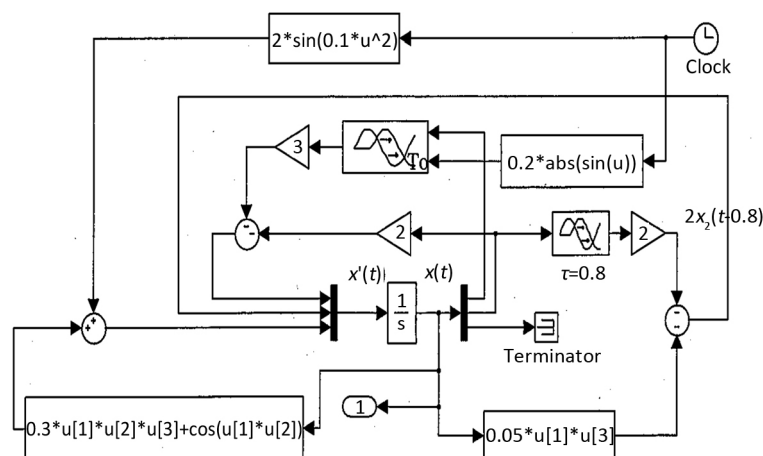


Figure 7. The Simulink model of the variable time delay differential equation
图 7. 变时间延迟微分方程的 Simulink 模型

图 7 中，使用 Variable Time Delay 模块实现变延迟时间， $0.2 \cdot |\sin t|$ 代表第二路输入延迟，求得延迟微分方程的数值解如图 8 所示。

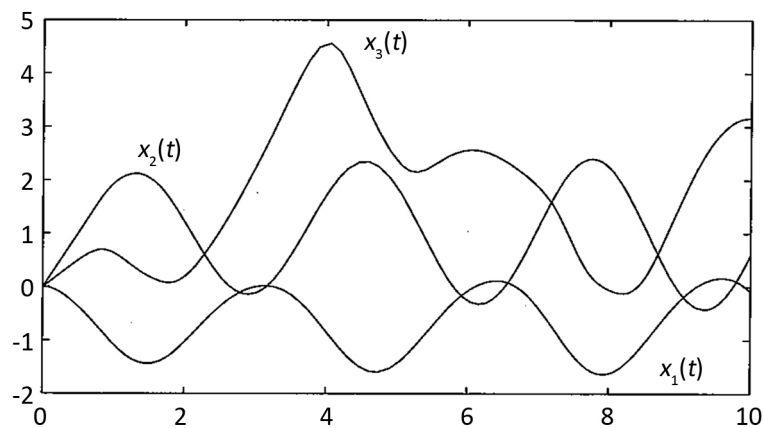


Figure 8. The simulated experimental numerical solution of the variable time delay differential equation
图 8. 变时间延迟微分方程的仿真实验数值解

参考文献

- [1] 薛定宇, 陈阳泉. 基于 MATLAB/Simulink 的系统仿真技术与应用[M]. 第 2 版. 北京: 清华大学出版社, 2011: 7-12.
- [2] 王旭, 王宏, 王文辉. 人工神经网络原理与应用[M]. 沈阳: 东北大学出版社, 2000: 179-185.
- [3] 薛定宇. 分数阶微积分学与分数阶控制[M]. 北京: 科学出版社, 2018: 33-39.
- [4] 邵军力, 张景, 魏长华. 人工智能基础[M]. 北京: 电子工业出版社, 2000: 217-222.
- [5] 薛定宇. 高等应用数学问题的 MATLAB 求解[M]. 第四版. 北京: 清华大学出版社, 2018: 317-326.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2324-7991, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: aam@hanspub.org