

Getting started



自动控制系统仿真

UM 软件入门系列教程

(03)

四川同算科技有限公司 译

2021年3月

本教程介绍使用UM软件与Matlab软件进行联合仿真的两种方法及基本流程。一种是从Matlab/Simulink导出控制系统模型到UM软件,另一种是从UM软件导出机械系统模型到Matlab/Simulink软件。

请读者在学习本课程之前务必先学习《UM软件入门系列教程01:多体系 统动力学仿真》,并熟悉UM软件的基本操作:新建模型,创建几何图形、刚 体、铰和力元。

本教程将通过两个例子讲解机械系统与控制系统联合仿真的方法。第一个 是倒立摆模型,通过控制系统实现稳定倒立;第二个是考虑电机驱动的传动轴 模型,模拟电机从启动到匀速工作的过程,并输出电流、电压、电磁力矩和角 速度等时程曲线。

首先分别在UM和Matlab/Simulink里创建机械系统和控制系统,最后将二 者耦合起来。这种技术可以帮助我们模拟非常复杂的系统,如汽车ABS系统、 磁悬浮列车。基于UM和Matlab实现耦合的方式有两种:第一种是从

Matlab/Simulink输出动态链接库DLL文件到UM模型,称之为Matlab Import; 第二种是从UM输出动力学模型到Matlab/Simulink作为S函数,称之为

$Cosimulation \, {\scriptstyle \circ}$

请读者务必逐页阅读、一步一步操作,有些基本的软件操作在后面不会详 细介绍甚至忽略。

兼容性

UM Control/Matlab Import

Matlab Import是UM Control模块的一个子模块,支持从Matlab/Simulink 导出DLL文件至UM。请先运行UM Input或UM Simulation程序,选择菜单 Help | About,在弹出窗口查看UM Control/Matlab Import一栏是否为"+"标记,若显示为"-",则请重新申请试用或购买正版许可。

目前,**UM Control/Matlab Import**支持**64**位的**Matlab(R2013-R2020)**版本(*UM软件自8.3.3.4版本起停止发布32位安装程序*)。

为了完成编译Matlab/Simulink模型,需要具备Matlab Coder和Simulink Coder两个工具包。读者可以在Matlab命令窗口使用ver命令查看。

另外,还必须安装有至少一个Matlab支持的微软编译器,可在Matlab官方 网站查询:

https://www.mathworks.com/support/requirements/supported-compilers.html https://www.mathworks.com/support/requirements/previous-releases.html 请注意:联合仿真只支持Matlab/Simulink里的连续单元,不支持离散单元。

UM Control/CoSimulation

CoSimulation也是**UM Control**模块的一个子模块,支持从**UM**导出动力学 模型至**Matlab/Simulink。CoSimulation**执行计算时还会用到**UM COM Server** 模块。

目前,**UM Control/CoSimulation**支持**64**位的**Matlab(R2013-R2020)**版本 (*UM软件自8.3.3.4版本起停止发布32位安装程序*)。

版权和商标

本教程仅供读者参考,不同的版本其界面可能有个别不同之处,我们会不 定期进行修订。对于本文档中可能出现的任何错误,我们不承担任何责任或义 务。

版权所有© 2021 Computational Mechanics Ltd.

俄罗斯计算力学有限公司保留所有权利。

联系方式

最新版的UM软件和相应的用户手册下载地址:

http://www.universalmechanism.com/en/pages/index.php?id=3.

若无法访问,请点击: <u>http://www.umlab.ru/en/pages/index.php?id=3</u>.

在使用过程中,读者如有任何报错、疑问和建议,请发送邮件至:

um@universalmechanism.com

UM总部

Computational Mechanics Ltd.

Vostochnaya str. 2-14, Glinischevo, Bryansk region, 241525, Russia

Phone, fax: +7 4832 568637

www.universalmechanism.com www.umlab.ru

UM中国

四川同算科技有限公司

四川省眉山市彭山区蔡山西路2号伟业广场1911室

办公电话: 028-38520556

公司网站: <u>www.tongsuan.cn</u>

电子邮件: <u>um@tongsuan.cn</u>





微信公众号

QQ 交流群

目 录

1.	UM C	CONTROL 模块	1
1.1	MA	TLAB IMPORT 工具	1
1.2	Co	SIMULATION 工具	1
2.	模型简	简介	2
2.1	倒	立摆	2
	2.1.1	模型简介	2
	2.1.2	准备 UM 模型	
2.2	直注	充电机	4
	2.2.1	模型简介	4
	2.2.2	机械系统	6
	2.2.3	电路系统	7
2.3	限台	制条件	7
3.	使用	MATLAB IMPORT 工具	8
3.1	工化	作流程	8
3.2	倒	立摆	9
	3.2.1	从 Matlab/Simulink 输出模型	9
	3.2.2	将 DLL 文件导入 UM	
	3.2.3	运动仿真	
3.3	直注	充电机	24
	3.3.1	Matlab/Simulink 模型	24
	3.3.2	从 Matlab/Simulink 输出模型	26
	3.3.3	将 DLL 文件导入 UM	27
	3.3.4	运动仿真	
4.	使用。	COSIMULATION 工具	32
4.1	工化	作流程	
4.2	倒	立摆	33
	4.2.1	准备 Matlab/Simulink 模型	
	4.2.2	输出 UM 模型	
	4.2.3	连接 UM 模型和 Matlab/Simulink 模型	40
	4.2.4	运动仿真	
4.3	直注	充电机	43
	4.3.1	准备 Matlab/Simulink 模型	43
	4.3.2	输出 UM 模型	44
	4.3.3	连接 UM 模型和 Matlab/Simulink 模型	47
	4.3.4	运动仿真	





1. UM Control 模块

UM 提供了多个工具实现机械系统与控制系统的联合仿真,其中与 Matlab/Simulink 相关的有两个。

1.1 Matlab Import 工具

先在 Matlab/Simulink 中建立控制系统模型,并编译输出动态链接库 (DLL 文件), 然后通过 UM 的外部库向导 Wizard of external libraries 导入 UM, 在 UM 里进行计算。

1.2 CoSimulation 工具

先在 UM 中建立机械系统模型,并通过专用工具 Wizard of export 生成 m 文件,导入 Matlab/Simulink 作为一个 S 函数,在 Matlab 里进行计算。

下面的框图显示了模型传递方向和执行仿真的主程序。实际上,对于两种方法,数据交互都是双向的。







在这一章,我们先介绍倒立摆和直流电机的机械系统模型及其参数。

2.1 倒立摆

2.1.1 模型简介

倒立摆模型如图 2.1 所示。该模型由一个小车 cart 和一个单摆 pendulum 组成。模型参数见表 2.1。机械系统的输出信号为角度ψ,控制系统的输出信号为力 *Force*。随角度变化而变化的控制力使得单摆能保持倒立平衡状态。



图 2.1 倒立摆 表 2.1 模型参数

参数	模型里的参数符号	备注	数值
F	force	作用在小车上的力	
М	mass_cart	小车的质量	0.5 kg
Ι	mass_pend	单摆的质量	0.2 kg
	ix	单摆绕转动轴的转动惯量	0.006 kg*m*m
	1	单摆质心到转动轴的距离	0.3 m
	b	小车与地面的摩擦阻尼系数	0.1 Ns/m
$\overline{\psi}$		单摆竖向摆动的角度	rad

Universal Mechanism 9





2.1.2 准备 UM 模型

虽然 Matlab/Simulink 动态链接库是在仿真阶段才导入 UM Simulation 程序的,但我们必须在建模时就做好准备工作。

如果控制系统通过计算输出力或力矩给机械系统,我们必须先运行 UM Input 程序,在机械系统里先定义这么一个力元,并赋予参数符号。

对于本例的倒立摆模型,我们添加了一个 **T-force** 类型的力元。这个力作用 于小车上,方向为小车局部坐标系的 Y 向,命名为 *force*,如图 2.2 所示。在之 后仿真时,使用外部库向导可以将控制系统计算得到的力传递给 *force*。

Name: Control for	ce 🕂 [4	• 1
Comments/Text att	tribute C	
Body1:	Body2:	
Base0	▼ Cart	-
Reference frame	Base0	-
Reduction point : Ca	art	
<u> </u>	3 <u>C</u>	C
Type of description		
Expression	🔘 File	
Force:		
		P
force		P
		P
Moment:		
		P
		P
		P

图 2.2 用于连接控制系统的力元





2.2 直流电机

2.2.1 模型简介

本例模型机械系统的主要元件是一个转动惯量为 J₀ 的传动轴,如图 2.3 所示。传动轴相对于大地(惯性系)有且仅有一个转动自由度。



图 2.3 模型简图

模型的电路系统在 Matlab/Simulink 里建立, 我们假设电机的转子和传动轴 是刚性连接的。

电机有一个输入信号和三个输出信号。

输入信号:

● 传动轴(电机转子)的角速度

输出信号:

- 电磁驱动力矩
- 电路中的电流
- 电路中的电压

电磁驱动力矩 *M*_a 是在 UM 模型里定义的 T-force 力元,而阻力矩 *M*_r则定义 为传动轴转动铰的阶跃函数,如图 2.4 所示。







图 2.4 用阶跃函数描述的阻力矩 表 2.2 模型参数

参数	模型里的参数符号	备注	数值
M_a	Driving_torque	电磁驱动力矩	
M_c		阻力矩	F(t)
J_o	Іу	轴的转动惯量	0.1 kg*m*m
t_1	t1	图 2.4 中的时刻 t ₁	5 s
t_2	t2	图 2.4 中的时刻 t ₂	8 s
C_{I}	C1	图 2.4 中的阻力矩 C ₁	5 N*m
C_2	C2	图 2.4 中的阻力矩 C ₂	10 N*m





2.2.2 机械系统

如前所述,为了能连接控制系统,我们需要在机械系统里预先创建一个参数 化的力元。

这里,我们打开 UM Input 程序,添加一个 T-force 力元,命名为 Driving torque,如图 2.5 所示。这个 T-force 力元有且仅有一个分量 Ma,表示绕 Y 轴转动的力矩,用于连接控制系统。

转动铰处的阻力矩表示如下:

Mr = c1*heavi(t-t1) + (c2-c1)*heavi(t-t2)

其中, t1, t2, c1 和 c2 是模型参数, 如图 2.4 所示。

请注意, heavi 函数定义如下:

$$heavi(t) = \begin{cases} 1, t > 0 \\ 0, t \le 0 \end{cases}$$

为了在仿真时能观察电流和电压的时程曲线,我们可以添加两个参数符号: I(电流)和 U(电压),如图 2.5 所示。在仿真时,我们将 Matlab/Simulink 模型的 另外两个输出信号分别指定给这两参数。

Name: Driving torque Comments/Text attribu	+	Name: jShaft	+ 🕂 🗊					
Body1:	Body2:	Body1:	Body2:					
Base0 💌	Shaft 🗾 💌	Base0	 Shaft 	-				
Reference frame Reduction point : Shaft	Base0	Type: Kotational	on Joint force	•	Whole list	+ 🛍	–	
<u> </u>		tit List of forces		-	Name	Expression	Value	Comment
Type of description		the clactor forces			r	0.07		
Expression	🔘 File	Resistance			h	0.5		
					iy	0.1		Moment of inertia
Force:	n	Name: Resistance	+ (+)	ti i	Ma	0		
	P			-	I	0		Current, will use to outpu
	P	Comments/Text at	tribute C		U	0		Voltage, will use to outpu
	D				r1	0.025		
	<u> </u>				h1	0.05		
Moment:		a+b Expression		-	ybase	0.67		
	P	Description of force	Imoment		z1	0.1		
Ma	P	Pascal/C expression	: F=F(x.v.t)		c1	-5		
		Examples			t1	5		
	<u>h</u>	-cstiff*(x-x0)-cdiss	*v+ampl*sin(om*t)		c2	-10		
M T= 10 dT=	0.01	F= c1*heavi(t-t1)	+(c2-c1)*heavi(t-t	2) P	t2	8		

图 2.5 驱动力矩、阻力矩和参数列表





2.2.3 电路系统

在本例中,我们将定义一个恒定磁通量的他励直流电机,其参数见表 2.3。

参数	数值
额定功率(kW)	3.6
额定电压(V)	220
额定转速(r.p.m)	3000
最大转速(r.p.m)	4000
效率 (%)	79
15℃下的电阻 (Ohm)	
电枢回路电阻	0.42
附加电极电阻	0.356
励磁回路电阻	129
电枢回路电感(mHn)	4.8

表 2.3 直流申机参数

2.3 限制条件

Matlab/Simulink 与 UM 联合仿真时,存在一些限制条件:可以将 Matlab/Simulink 的输出信号传递给力元相关的符号:如力的作用点、刚度和阻 尼系数。

根据 UM 软件的基本原则,不支持将 Matlab/Simulink 的输出信号传递给刚 体的质量惯量参数、几何参数和较坐标,因为这些参数只能在每次仿真开始前设 置,而不能在仿真过程中随时间不断变化。





3. 使用 Matlab Import 工具

3.1 工作流程

从 Matlab/Simulink 导出控制系统到 UM 进行联合仿真的流程如下:

- 在 Matlab/Simulink 里搭建控制框图;
- 从 Matlab/Simulink 输出动态链接库 (DLL 文件);
- 在 UM Input 程序里创建机械系统模型;
- 在 UM Simulation 程序里加载机械系统模型,利用外部库向导(Wizard of external libraries)导入 Matlab/Simulink DLL 文件,并设置好连接;
- 进行动力学仿真。

UM 将 Matlab/Simulink 模型考虑为一个具有若干个输入和输出信号的黑盒子。输入信号为 UM Simulation 程序里变量向导(Wizard of variables)创建的 变量;输出信号为 UM Input 程序里创建的参数符号。

为了实现力或力矩的主动控制效果,必须在 UM 里创建相关力元,并以参数 化描述,这样才能与 Matlab/Simulink 的输出信号建立连接。





3.2 倒立摆

UM 软件自带的不包含控制系统的倒立摆模型位于本地目录{UM Data} \SAMPLES\TUTORIAL\inv_pend。在开始学习本课程之前,请先确认这个模型 是 否 存 在 。 如 果 没 有 找 到 , 可 以 从 UM 软 件 官 方 网 站 下 载 : http://www.umlab.ru/download/90/inv_pend.zip

我们将直接使用它,这里就不再详细介绍其建模过程,而着重讲解有关连接 Matlab/Simulink 模型的方法。

包含控制系统的 UM 模型位于本地目录 {UM Data}\SAMPLES\ simulink\inv_pend_ctrl。

3.2.1 从 Matlab/Simulink 输出模型

从 Matlab/Simulink 输出控制系统模型意味着将模型编译为动态链接库,用于连接 UM 模型。

备注: 在学习之前,最好先熟悉 Matlab/Simulink 的基本操作,以保证能顺利地 编译成功 DLL 文件。不推荐直接使用软件自带的动态链接库 pendpid_ctrl.dll 文 件,因为它可能是 32 位版本 Matlab/Simulink 编译的,不能用于 64 位 UM 软件 进行仿真。

提示: Matlab/Simulink 里的 Derivative 模块不支持导出使用,可用 Transfer Fnc 模块代替。

Matlab/Simulink 里的基本流程如下:

- 1. 准备一个控制系统模型;
- 2. 复制编译 DLL 所需的文件到模型目录;
- 3. 设置编译选项并执行编译。





准备控制系统模型

首先,必须去掉 Matlab/Simulink 模型中不能被编译的模块/组件:所有的输入/输出组件,以及所有没有源代码的组件。

请注意模型必须有"IN"和"OUT"组件,用于与 UM 模型进行连接。在本例 中,分别有一个"IN"(单摆竖向摆角)接口和"OUT"(作用在小车上的控制力)接口。

Matlab/Simulink 控制系统如图 3.1 所示。



图 3.1 倒立摆模型的控制框图





复制编译所需文件和设置工作目录

 为了将控制模型正确编译成 UM 所需的 DLL 文件,我们需要先从目录 {UM Data}\Simulink 复制相应版本的接口文件到存放 Matlab/Simulink 模型的目录下。

Matlab 8.X / R2013-R2015 32 位

复制{UM Data}\Simulink\R2013_2015\x32 目录下的 rsim.tlc, rsim_vc.tmf 和 um.tls 文件到模型文件夹。

Matlab 8.X / R2013-R2015 64 位

复制{UM Data}\Simulink\R2013_2015\x64 目录下的 rsim.tlc, rsim_vc.tmf 和 um.tls 文件到模型文件夹。

Matlab 9.X / R2016-R2020 64 位

复制{UM Data}\Simulink\R2016_R2020x64 目录下的 rsim.tlc, rsim_vc.tmf 和 um.tls 文件到模型文件夹。

2. 在 Matlab 主页窗口置模型目录为当前工作目录。

MA	TLAB R	2015a										
Н	DME		PLOTS	APP	s							
New Script	New	Open	Find Files	Import Data	Save Workspace	New Variable Open Variable Clear Workspace	Analyze Code	Simulink Library	Layout	 Preferences Set Path Parallel • 	? Help	Community → Request Support + Add-Ons ▼
		FILE			V	ARIABLE	CODE	SIMULINK	EI	NVIRONMENT		RESOURCES





编译 DLL

1. 在 Matlab 命令窗口输入命令: mex -setup (注意 mex 后有一个空格), 然后从列表中选择一个编译器, 或直接输入 mex -setup C++。

```
Command Window
```

>> mex -setup C++

MEX configured to use 'Microsoft Visual C++ 2010' for C++ language compilation.

备注: 我们推荐使用 Microsoft Visual C/C++编译器来编译生成动态链接库。





使用 Matlab 8.X / R2013-R2015 编译 DLL

- 在 Simulink 模型窗口,选择菜单 Code | C/C++ Code | Code Generation Options, 弹出界面如图 3.2。
- 2. 在 Code Generation 页面设置 System target file 为 rsim.tlc。
- 3. 设置 Language 为 C。
- 4. 设置 Make command 为 make_rtw。
- 5. 设置 Template makefile 为 rsim_vc.tmf。
- 6. 在 Solver 页面设置 Type 为 Fixed Step。
- 7. 在 Solver options | Solver 选择合适的求解器,如 ode4(四阶龙格库塔法)。
- 8. 回到 Code Generation 页面,点击 Build,编译生成 DLL 文件。

Configuration Parameters: pendp	oid_cntr/Configuration (A	ctive)		×				
Select:	Target selection							
Solver Data Import/Export	System target file:	rsim.tld		Browse				
Optimization	Language:	С		▼				
 Diagnostics Hardware Implementation 	Description:	Rapid Simulation Ta	arget					
Model Referencing	Build process							
Code Generation	Compiler optimizati	on level: Optim	izations off (faster builds) 🔻					
	TLC options:							
	Makefile configura	tion						
	Generate make	file						
	Make command:	mak	.e_rtw					
	Template makefile: rsim_vc.tmf							
	Code Generation	Advisor						
	Select objective:		Unspecified	~				
	Check model befo	re generating code:	: Off	Check model				
	Generate code o	nly		Build				
	Package code ar	d artifacts		Zip file name:				
				OK Cancel Help Apply				

图 3.2 设置编译选项(Matlab 8.X R2013-R2015)

编译成功后,会出现提示"Creating LIBRARY..\pendpid_cntr.lib and object ..\pendpid_cntr.exp"。最后得到 pendpid_cntr.dll 文件。

备注:为确保正常计算,最好将 Matlab 和 UM 软件安装在同一台计算机。





使用 Matlab 9.X / R2016-R2020 编译 DLL

- 1. 在 Simulink 界面定位到 pendpid_ctrl 模型窗口。
- 选择 MODELLING | Model Settings | Model Settings,或在空白处点击 右键,选择 Model Configuration Parameters 菜单,弹出窗口如图 3.4。



图 3.3 设置编译选项(Matlab 9.X R2016-R2020)

Configuration Parameters: pend	pid_cntr/Configuration (Active) — 🗆 🗙
Q Search	
Solver Data Import/Export Math and Data Types Diagnostics Hardware Implementation Model Referencing Simulation Target Code Generation Optimization Report Comments Identifiers Custom Code Interface RSim Target Coverage	Target selection System target file: rsim.tlc Language: C Description: Rapid Simulation Target Build process Generate code only Package code and artifacts Zip file name: Compiler optimization level: Optimizations off (faster builds) Makefile configuration Generate makefile Template makefile: remplate makefile: remplate makefile: remplate makefile: Select objectives Select objective: Unspecified Check model before generating code: Off
	OK Cancel Help Apply

Universal Mechanism 9





- 3. 定位到 Code Generation 页面。
- 4. 设置 System target file 为 rsim.tlc。
- 5. 设置 Language 为 C。
- 6. 设置 Template makefile 为 rsim_vc.tmf。
- 7. 设置 Make command 为 make_rtw。
- 8. 切换到 Solver 页面。
- 9. 设置 Solver options | Type 为 Fixed Step。
- 10. 在 Solver options | Solver 选择合适的求解器,如 ode4(四阶龙格库塔法)。
- 11. 然后到 Code Generation | Optimization 页面,设置 Default parameter behavior 为 Tunable,如图 3.5。
- 12. 关闭 Configuration Parameters 窗口。

Configuration Parameters: pendpi	d_cntr/Configuration (Active)			-		×
Q Search						
Solver Data Import/Export Math and Data Types Diagnostics Hardware Implementation Model Referencing Simulation Target Code Generation Optimization Report Comments Identifiers Custom Code Interface RSim Target Coverage	Default parameter behavior: Use memcpy for vector ass Loop unrolling threshold: Maximum stack size (bytes): 	Funable signment 5 Inherit from target	Memcpy three	▼ [shold (by	Configur tes): 64	e
		<u>O</u> K	<u>C</u> ancel	<u>H</u> elp	A	oply

图 3.5 Matlab 9.X 的 Tunable 参数

13. 最后从 APPS 里选择 Simulink Coder | Build (Generate code and build model),完成编译,如图 3.6 和图 3.7。

编译成功后,会出现提示"Build process completed successfully"。最后得到 pendpid_cntr.dll 文件,以及源代码(位于 pendpid_cntr_rsim_rtw 目录)。

备注:为确保正常计算,最好将 Matlab 和 UM 软件安装在同一台计算机。







图 3.7 进程查看





3.2.2 将 DLL 文件导入 UM

加载 Matlab/Simulink 动态链接库

- 1. 运行 UM Simulation 程序。
- 2. 加载模型{UM Data}\SAMPLES\TUTORIAL\inv_pend。
- 3. 选择菜单 Tools | External library interface, 弹出外部库向导窗口。



图 3.8 外部库向导

- 4. 点击按钮➡,添加一个外部库。
- 在右上角 Path to external library 处点击按钮 [→],选择在 Matlab 里编
 译的动态链接库文件 pendpid cntr.dll,如图 3.9 所示。
- 6. 在图 3.9 页面左侧, 勾选 Interface0。



图 3.9 添加一个外部库

外部库向导加载动态链接库时,自动识别出控制系统的输入和输出接口,并 列表显示。本例模型只有一个输入和一个输出。

Universal Mechanism 9

Getting Started





重命名

- 1. 在**外部库向导**窗口左侧选中 Interface0, 点右键, 选择菜单 Rename。
- 2. 输入 Control force,并回车。



图 3.10 重命名





定义控制系统的输入信号

- 1. 选择菜单 Tools | Wizard of variables, 打开变量向导。
- 2. 选择角度变量 Angular variables。
- 3. 勾选 Use orientation at zero coordinates。
- 在左侧列表选中 Pendulum,选择 Type of variable 为 Rot.vector,设置 分量 Component 为 X,如图 3.11 所示。
- 5. 点击按钮 [♥] 创建单摆绕 X 轴转动的角度变量。

📑 Wizard of variables							×	
🥩 Variables for gro	oup of bod	ies	T-Forces			🔍 Joint forces		
Coordinates	0	Solver variabl	Solver variables 📑 All forces			id Identifiers		
🛕 Angular variables	🛃 🛃	ear variables	a•b E	xpression	User varia	bles	🕪 Reactions	
🖃 🔳 inv_pend		Selected						
Cart		Pendulum						
Endulum		Use orientation at zero coordinates						
		Type of varia	able	O Ang. v	elocity		ng. acceleration	
		Component	ОY	⊃z	0	V	٥v	
		Resolved in SC of body						
		Base0						
		Relative to body						
		Base0					-	
ang:x(Pendulum)	Ve	ctor of rotation of	of body Pe	endulum relativ	ve to Base0, S	C Base	0, projecti 🐬 👼	
ang:x(Pendulum)								
l								

图 3.11 定义角度变量





6. 将上一步创建的角度变量 ang:x(Pendulum)从变量向导窗口拖到外部库 向导窗口 Input1 处,作为控制系统的输入信号,最后如图 3.12 所 示。

+ Wizard of external libraries			- • ×
External libraries Control force	Path to external library srs\Public\Documents\UM Software L Model name State variables 2	.ab\Universal Mechanism\7\SAMPL d_cntr	ES\TUTORIAL\inv_ 🧭 🦉
	Inputs ✓ Input1 < ang:x(Pendulum)	Outputs ☑ ⊘ Output1> (none)	Parameters Constant_Value = 0 Proportional_Gain = 100 Integral_A = 0 Integral_C = 1 D_Gain = 20 Transfer_Fcn_A = -1000 Transfer_Fcn_C = -100 Transfer_Fcn_D = 1000
OK Apply	Cancel		

图 3.12 定义控制系统的输入变量





定义控制系统的输出信号

- 1. 在外部库向导窗口双击 Output1, 弹出模型参数窗口。
- 2. 从下拉菜单选择参数 force,如图 3.13 所示。
- 3. 点击 OK。

lo] External library -> Model parameters X	
Assign external library output value to the following UM parameter:	
force 👻	
Value of UM parameter when connection is off:	
0	
Assign also	
□ No identifiers of the same name	
OK Cancel	

图 3.13 定义输出变量

4. 在图 3.14 所示界面,点击 Apply (保存设置但不关闭窗口)或 OK (保存设置同时关闭窗口)。

* Wizard of external libraries			
External libraries Control force	Path to external library :rs\Public\Documents\UM Software L Model name pendpic State variables 2	.ab\Universal Mechanism\7\SAMPLI d_cntr	ES\TUTORIAL\inv_ 🧭 🦉
	Inputs ✓ Input1 < ang:x(Pendulum)	Outputs ✓ Output1> force	Parameters Constant_Value = 0 Proportional_Gain = 100 Integral_A = 0 Integral_C = 1 D_Gain = 20 Transfer_Fcn_A = -1000 Transfer_Fcn_C = -100 Transfer_Fcn_D = 1000
OK Apply	Cancel		

图 3.14 完成输入输出设置的界面

至此,倒立摆模型的机械系统和控制系统已创建连接,可以进行数值仿真了。





3.2.3 运动仿真

- 1. 选择菜单 Analysis | Simulation, 弹出仿真控制界面。
- 2. 点击初始条件 Initial conditions。
- 3. 设置 Coordinate/1.2 为 0.3,表示将单摆初始转动 0.3 弧度,如图 3.15 所示。

备注: 坐标 1.1 表示小车的平动自由度, 1.3 和 1.4 则为控制系统的状态变量。

Animation window	Object simula	ation inspector		
a ⊡ ⊡ ⊕ @ & ⊕ ∎ ▶	XVA		Information	Tools
	Solver	Identifiers	Initial condition	ns Object variables
	Coordinates	Constraints or	n initial conditions	
	🖻 🖻		- x=0 ν=0	<u> </u>
1	ψÛ	✓ Coordinate	Velocity Co	omment
	1.1	0	0 jC	art 1c
	1.2	0.3	0 jP	endulum 1a
	1.3	<u> </u>	0 Ac	dd. vars (ODE order 1):1
	1.4	0	0 Ac	dd. vars (ODE order 1):2
	< Message Number of d. Integra	dx= o.f. = 4 tion	0.1 🔟 da= Message	> 0.1 Close

图 3.15 设置初始条件

- 4. 点击 Solver,选择 Park 求解器,设置仿真时间为 0.1s,数据和动画步 长为 0.0002s,容差 Error tolerance 为 1E-7。
- 5. 点击 Integration,开始仿真,待仿真完成后点击 Interrupt。 从动画窗口可以看到,单摆在在控制力作用下能保持稳定。





控制力可视化和变量绘图

- 1. 选择菜单 Tools | Wizard of variables, 打开变量向导。
- 点击 T-Forces,在左侧列表选中 Control force,右侧 Type of variable 选择 Force,设置分量为 V(表示矢量), Acts on 作用于 Body2(小 车)。
- 3. 点击按钮, 创建变量, 并将其拖入动画窗口。
- 4. 点击动画窗口工具栏图标印,模型显示为线框模式。
- 5. 选择菜单 Tools | Graphical window, 打开一个绘图窗口。
- 6. 在变量向导 **T-Force** 页面,设置分量为 **Y**,创建变量,并拖入绘图窗口。
- 7. 再打开一个绘图窗口。
- 8. 点击变量向导 Angular variables 页面。
- 左侧选中 Pendulum,右侧 Type of variable 选择 Rot.vector,设置分量X。
- 10. 创建变量,并拖入到第二个绘图窗口。
- 11. 关闭变量向导。
- 12. 开始仿真。







3.3 直流电机

UM 软件自带的不包含控制系统的直流电机模型位于本地目录{UM Data} \SAMPLES\TUTORIAL\dc_motor。在开始学习本课程之前,请先确认这个模型 是 否 存 在 。 如 果 没 有 找 到 , 可 以 从 UM 软 件 官 方 网 站 下 载 : http://www.umlab.ru/download/90/dc_motor.zip 。

我们将直接使用它,这里就不再详细介绍其建模过程,而着重讲解有关连接 Matlab/Simulink 模型的方法。

包含控制系统的 UM 模型位于本地目录 {UM Data}\SAMPLES\ simulink\dc_motor_fin。

3.3.1 Matlab/Simulink 模型

本例直流电机 Matlab/Simulink 模型有一个输入信号(角速度)和三个输出 信号(电磁力矩、电流和电压),电路控制系统如图 3.16 和图 3.17 所示。



图 3.16 电路系统总图







图 3.17 直流电机子系统

Universal Mechanism 9





3.3.2 从 Matlab/Simulink 输出模型

现在,请参考 3.2.1 章节介绍的方法将"{UM Data} \SAMPLES\TUTORIAL\dc_motor"模型里的 dcmotor1.mdl 文件编译为 dcmotor1.dll。

不推荐直接使用软件自带的动态链接库 dcmotor1.dll 文件,因为它可能是 32 位版本 Matlab/Simulink 编译的,不能用于 64 位 UM 软件进行仿真。







3.3.3 将 DLL 文件导入 UM

加载 UM 模型

- 1. 运行 UM Simulation 程序。
- 2. 加载模型{UM Data}\SAMPLES\TUTORIAL\dc_motor。

加载 Matlab/Simulink 动态链接库

- 1. 选择菜单 Tools | External library Interface..., 弹出外部库向导窗口。
- 2. 点击按钮╋添加一个外部库。
- 3. 在右上角 Path to external library 处点击按钮 3,选择在 Matlab 里编

译好的动态链接库文件 dcmotor1.dll, 如图 3.18 所示。

4. 在图 3.18页面左侧, 勾选 Interface0。

外部库向导加载动态链接库时,自动识别出控制系统的输入和输出接口,并 列表显示。本例模型只有一个输入(轴的角速度)和三个输出(驱动力矩、电流 和电压)。

* Wizard of external libraries			
🛏 🖯 🕂 🖬 🗑	Path to external library	LIM Software Lab\Universal Mecha	
External libraries	Model name di State variables 1 Inputs Input1 < (none)	Cmotor1 Outputs Output1> (none) Output2> (none) Output3> (none)	Parameters cF_Value = 0.5533 Integrator_IC = 0 U_Value = 220 Zero1_Value = 0 R1_Value = 2.5 R1Switch_Threshold = 1.2 Zero2_Value = 0 R2_Value = 2.035 R2Switch_Threshold = 2.5 Ra_Value = 0.776 La_Value = 0.0048
OK Apply	Cancel		

图 3.18 导入 Matlab/Simulink 动态链接库





重命名

- 1. 在外部库向导窗口左侧选中 Interface0, 点右键, 选择菜单 Rename。
- 2. 输入 DC Motor,并回车。

定义控制系统的输入信号

- 1. 选择菜单 Tools | Wizard of variables, 打开变量向导。
- 2. 选择角度变量 Angular variables。
- 在左侧列表选中 Shaft,选择 Type of variable 为 Ang.velocity,设置分量 Component 为 Y,如图 3.19 所示。
- 4. 点击按钮¹创建传动轴轴角速度变量 om:y(Shaft)。

🔄 Wizard of variables						(×
🥩 Variables for gro	up of bodies		T-Forces		R	Joint forces	
Coordinates	🜔 Solver varial	bles	📑 Al	forces		id Identifiers	
🛕 Angular variables	💒 Linear variables	a•b	Expression	User varia	bles	🕀 Reactio	ns
□ dc_motor_fin	Selected						^
Shaft	Shaft						
	Type of vari	able tor	Ang. velo	ocity (🔾 Ang.	acceleration	
	Component O X	● Y	⊖z	01	(1	٥v	
	Resolved in	SC of body					
	Base0					•	
	Relative to b	oody					
	Base0					•	
							~
om:y(Shaft)	Angular velocity of	f body Shaf	t relative to Ba	se0, SC Base0), projec	ction Y	#
om:y(Shaft)							

图 3.19 创建传动轴角速度变量

- 5. 将变量 om:y(Shaft)拖到外部库向导窗口 Input1 处,作为控制系统的输入信号。
- 6. 关闭变量向导。





定义控制系统的输出信号

- 1. 在外部库向导窗口,双击 Output1,弹出模型参数窗口。
- 2. 从下拉菜单选择参数 Ma, 如图 3.20 所示。
- 3. 点击 OK。

External library -> Model parameters X	1
Assign external library output value to the following LIM parameter:	
Assign external library output value to the following on parameter.	4
Ma	
Value of UM parameter when connection is off:	
0	
Assign also	C C
□ No identifiers of the same name	
OK Cancel	J

图 3.20

- 4. 双击 Output2,选择参数 I,点击 OK。
- 5. 双击 Output3,选择参数 U,点击 OK;最终参数设置如图 3.21 所示。
- 6. 点击外部库向导窗口 OK,保存设置并关闭窗口。

↔ Wizard of external libraries			
External libraries DC motor	Path to external library "C:\Users\Public\Documents\L Model name State variables 1	JM Software Lab\Universal Mecha	anism\7\SAMPLES\TU 😹 🚪
	Inputs ✓ Input1 < om:y(Shaft	Outputs ✓ ✓ Output1> Ma ✓ ✓ Output2> I ✓ ✓ Output3> U	Parameters cF_Value = 0.5533 Integrator_IC = 0 U_Value = 220 Zero1_Value = 0 R1_Value = 2.5 R1Switch_Threshold = 1.2 Zero2_Value = 0 R2_Value = 2.035 R2Switch_Threshold = 2.5 Ra_Value = 0.776 La_Value = 0.0048
OK Apply	Cancel		





3.3.4 运动仿真

仿真前的准备

- 1. 选择菜单 Tools | Animation window, 打开一个动画窗口。
- 2. 选择菜单 Tools | Graphic window, 打开一个绘图窗口, 再重复两次, 最终仿真程序桌面布置如图 3.22 所示。



图 3.22 仿真界面窗口布局

- 3. 选择菜单 Tools | Wizard of variables, 打开变量向导。
- 点击变量向导的 Joint force 页面,在左侧列表选中 jShaft,然后设置 类型为 Torque,分量为 Y,选择 Acts on 物体为 body2:Shaft,点击按 钮[♥] 创建变量 jAFy(jShaft)。
- 6. 将变量 Ma 和 jAFy(jShaft)拖入图 3.22 所示左下方的绘图窗口。
- 7. 用同样的方法创建角速度变量(Angular Variable 页面, Type of variable 为 Ang.velocity, Component 为 Y),并拖入右上方的绘图窗 口。
- 8. 用同样的方法创建电流和电压变量(Identifier页面,I和U参数),并 拖入右下方的绘图窗口。
- 9. 关闭变量向导。





运动仿真

- 1. 选择菜单 Analysis | Simulation,打开仿真控制界面。
- 2. 点击 Solver 页面,选择求解器 BDF,仿真时间 10s,动画和数据步长 0.02s,容差 0.001。
- 3. 点击 Integration 开始仿真计算。
- 4. 仿真过程如图 3.23 所示。



图 3.23 仿真过程





4. 使用 CoSimulation 工具

下面我们使用 UM Control/CoSimulation 工具来进行联合仿真,需要将 UM 机械系统模型作为 S 函数导入到 Matlab/Simulink。

4.1 工作流程

从 UM 导出机械系统模型到 Matlab/Simulink 进行联合仿真的流程如下:

- 在 Matlab/Simulink 里搭建控制框图;
- 添加一个 S-function 组件到 Matlab/Simulink 控制系统;
- 在 UM Input 程序里创建机械系统模型;
- 在 UM Simulation 程序里加载机械系统模型,设置好连接,生成 m-file;
- 在 Matlab/Simulink 里进行动力学仿真。

将 UM 模型考虑为一个具有若干个输入和输出信号的黑盒子。输入信号为 UM Input 程序里创建的参数符号,通常与力/力矩关联;输出信号为 UM Simulation 程序里变量向导(Wizard of variables)创建的变量,如速度、加速 度等。





4.2 倒立摆

UM 软件自带的倒立摆模型(已准备好.cosim 和.m 文件)位于本地目录{UM Data} \SAMPLES\cosimulation\inv_pend_cosim。在开始学习本课程之前,请先确认这个模型是否存在。如果没有找到,可以从 UM 软件官方网站下载: www.umlab.ru/download/90/inv pend_cosim.zip 。

我们将直接使用这个模型,这里就不再详细介绍其建模过程,而着重讲解有 关连接 Matlab/Simulink 模型的方法。





4.2.1 准备 Matlab/Simulink 模型

本例用到的 Matlab/Simulink 控制系统模型与图 3.1 的模型非常相似,如图 4.1 所示。不同之处在于,这里用到了一个 S-Function 模块,这个模块可以在 Simulink 的 User-Defined Functions 库找到,缺省名称为 System。

通过 S-Function 可以将 UM 模型和 Simulink 模型连接起来。如图 4.1 所示, S-Function 有一个输入信号(力)和一个输出信号(角度)。



图 4.1 包含 S-Function 的控制系统

图 4.1 对应的 Matlab/Simulink 模型位于本地目录{UM Data}\SAMPLES\ cosimulation\inv_pend_cosim\pendpid_cntr。

接下来,我们先从 UM 输出模型,再到 Simulink 里进行仿真计算。





4.2.2 输出 UM 模型

下面,我们在 UM 模型里定义与 S-Function 相关的输入和输出信号,保存 相关设置,并生成专门的 m 文件。

加载 UM 模型

- 1. 运行 UM Simulation。
- 2. 打开模型{UM Data} \SAMPLES\cosimulation\inv_pend_cosim。

设置初始条件

首先,我们需要将倒立摆初始转动一定角度,使之偏离理想的平衡位置。

- 1. 选择菜单 Analysis | Simulation, 打开仿真控制界面。
- 2. 点击 Initial conditions 页面。
- 3. 设置 Coordinate/1.2 为 0.3, 如图 4.2 所示。



图 4.2 设置初始条件





输出 UM 模型

1. 选择菜单 Tools | Wizard of export, 弹出 UM 输出向导窗口。

🕙 Wizard of export to Matlab/Si	mulink	
🖻 🖪 🕻 🗯 🗏		
+ 🗊	+ 🛍	+ 🗊
Inputs	Outputs	Parameters
Close	·	

图 4.3 UM 输出向导窗口

现在我们需要创建两个变量:一个是作用于小车的控制力变量,作为输入信号;另一个是倒立摆的竖向摆角,作为输出信号。





定义输入信号

- 1. 在图 4.3 界面 Inputs 框点击按钮+,弹出模型参数窗口。
- 2. 从下拉菜单选择 force 参数。
- 3. 点击 OK, 如图 4.4 所示。



图 4.4 定义输入信号





定义输出信号

- 1. 在图 4.3 界面 Outputs 框点击按钮+,添加一个输出信号。
- 2. 打开变量向导。
- 3. 点击 Angular variables 页面。
- 在左侧列表选中 Pendulum,取消选择 Use orientation at zero coordinates,设置 Type of variable 为 Rot.vector, Component 为 X。
- 5. 点击按钮[●], 创建角度变量 ang:x(Pendulum)。

📑 Wizard of variables						×
🥩 Variables for gro	up of bodies		T-Forces		ر 🔍	oint forces
Coordinates	Solver var	iables	II Al	forces		id Identifiers
🛕 Angular variables	🛃 Linear variables	a•b [Expression	User variat	oles	🕪 Reactions
🖃 🔳 inv_pend_cosim	Selected					
Cart	Pendulum					
Pendulum	Use orie	ntation at zer	o coordinates			
	Type of va Rot. ve	ariable ector	🔿 Ang. ve	elocity	Ang	. acceleration
	Componen	t Ο Y	⊖z	0	V	Ov
	-Resolved i	n SC of body				
	Base0					•
	Relative to	body				
	Base0					-
ang:x(Pendulum)	Vector of rotatio	n of body Per	ndulum relative	to Base0, SC E	Base0, pr	ojection : 🗖 률
ang:x(Pendulum)						

图 4.5 变量向导





6. 将上一步创建的变量 ang:x(Pendulum)从变量向导拖入输出向导的 Output1 处,作为输出信号,如图 4.6 所示。

🖆 Wizard of export to Matlab/Si	mulink	
🖙 🖪 🕻 🇯 🗏		
+ 🗊	+ 🛍	+ 🗑
Inputs	Outputs	Parameters
1. force	✓ 1. ang:x(Pendulum) - Vecto	
Close		

图 4.6

至此,UM 模型输出前的准备工作已经全部完成,下面来生成.m 文件和.cosim 文件。

生成 m 文件

- 1. 在图 4.3 界面点击按钮 →, 弹出警告信息, 提示参数列表为空, 点击 Yes, 忽略即可。
- 弹出保存窗口,设置名称 inv_pend_cosim,点击 Save。这样就生成了 inv_pend_cosim.cosim 文件和 inv_pend_cosim.m 文件。在 Matlab/ Simulink 环境中执行仿真时,通过 cosim 文件与 UM 进行数据交互。





4.2.3 连接 UM 模型和 Matlab/Simulink 模型

在前面的操作过程中,生成的 m 文件缺省保存于 UM 模型目录。为了仿真 顺利进行,需要将其复制到 Matlab/Simulink 模型所在目录。只是由于本例的 Matlab/Simulink 模型已经位于 UM 模型目录,因此这里无需操作。

- 1. 运行 Matlab/Simulink。
- 2. 设置当前工作目录为{UM Data}\SAMPLES\cosimulation \inv_pend_cosim。

HOME	PLOTS APPS				
New New Script -	Open Compare Import Save Data Workspace	New Variable Analyz Open Variable De Run ar Clear Workspace De Clear (e Code Internet Commands ▼ Library	Image: Set Path Image: Set Path Image: Parallel	Community Request Support Add-Ons
4 🔶 🔁	FILE VAR	ents 🕨 UM Software Lab 🕨 Uni	versal Mechanism + 9 +	SAMPLES Cosimulatio	n • inv_pend_cosim
3	. 加载图 4.1 所示的	控制系统模型 p	endpid_cntr。	•	
4	. 双击 S-Function,	弹出参数设置窗	「口。		
5	. 设置名称为 inv_p e	end_cosim,即 n	n 文件的名字	z,如图 4.7)	所示。
6	. 点击 OK 。				
	Function Block Para S-Function User-definable blo (Level-1), and For standards. The va passed to the S-fu additional paramet If the S-function building generated function modules' extensions or full 'src.c src1.c'. Parameters S-function name: S-function parameters	meters: S-Function ock. Blocks can tran and must co ariables t, x, u, mction by Simuli ers in the 'S-fu block requires a d code, specify t field. Enter the l pathnames, e.g. inv_pend_cosim ters: s: ''	be written in mform to S-fu and flag are nk. You can mction parame additional sou he filenames filenames on , enter 'src	n C, MATLAB unction e automatical specify eters' field. urce files for in the 'S- nly: do not un srcl', not Edit	ly r se
	0	OK Car	ncel Hel	lp Appl:	у

图 4.7 设置 S-Function 名称





4.2.4 运动仿真

设置仿真参数

1. 选择菜单 Simulation | Configuration Parameters。

🔩 🗸 📻 🧄 👋 📷 😓 Update Diagram Ctrl+D		File Edit View Display Diagram	Simulation	Analysis Code Tools	>>
	endpid_entr	🎭 🖛 🐘 🔤 » 📸	💩 Update	e Diagram	Ctrl+D
Model Configuration Parameters Ctrl+E	endpid_cntr		Model	Configuration Parameters	Ctrl+E

- 2. 点击 Solver 页面。
- 3. 设置 Start time 为 0.0, Stop time 为 0.3, 选择 Type 为 Fixed-step, Solver 为 ode3 (Bogacki-Shampine), 如图 4.8 所示。

elect:	Simulation time		
Solver Data Import/Export	Start time: 0.0	Stop time: 0.3	
 Diagnostics Hardware Implementation 	Solver options		
Model Referencing Simulation Target	Type: Fixed-step	 Solver: ode3 (Bogacki-Shampine) 	•
Code Generation	Fixed-step size (fundamental sample time):	1e-4	
	Tasking and sample time options		
	Periodic sample time constraint:	Unconstrained	•
	Tasking mode for periodic sample times:	Auto	-
	🗏 Automatically handle rate transition for data t	ransfer	
	🗌 🔲 Higher priority value indicates higher task pri	ority	
	"		+
		OK Cancel Help Ap	ply

图 4.8 仿真参数

4. 点击 **OK**。





运动仿真

- 1. 双击打开所有的 Scope 窗口。
- 2. 运行仿真。
- 3. 仿真结果如图 4.9 和图 4.10 所示。



图 4.9 倒立摆竖向摆角



图 4.10 作用在小车上的控制力





4.3 直流电机

UM 软件自带的直流电机模型(已准备好.cosim 和.m 文件)位于本地目录 {UM Data} \SAMPLES\cosimulation\dcmotor_cosim。在开始学习本课程之前, 请先确认这个模型是否存在。如果没有找到,可以从 UM 软件官方网站下载: www.umlab.ru/download/90/dcmotor_cosim.zip 。

4.3.1 准备 Matlab/Simulink 模型

本例用到的 Matlab/Simulink 控制系统模型与图 3.16 的模型非常相似,如 图 4.11 所示。不同之处在于,这里用到了一个 S-Function 模块,这个模块可以 在 Simulink 的 User-Defined Functions 库找到,缺省名称为 System。控制系统 模型位于{UM Data} \SAMPLES\cosimulation\dcmotor_cosim\dcmotor_cntr。



图 4.11





4.3.2 输出 UM 模型

加载 UM 模型

- 1. 运行 UM Simulation。
- 2. 打开模型{UM Data} \SAMPLES\cosimulation\dcmotor_cosim。

输出 UM 模型

1. 选择菜单 Tools | Wizard of export, 弹出输出向导窗口。

现在我们需要创建两个变量:一个是作用于传动轴的控制力矩,作为输入信号;另一个是传动轴的角速度,作为输出信号。

定义输入信号

- 1. 在 Inputs 框点击按钮+,弹出模型参数窗口。
- 2. 从下拉菜单选择 Ma 参数,点击 OK,如图 4.12 所示。

ĺ	linput signal	×
	Assign input signal to the following UM parameter:	
	Ма	-
	Assign also	
	No identifiers of the same name	
	-	
	OK Cancel	_
	图 4.12 定义输入信号	





定义输出信号

示。

- 1. 在 Outputs 框点击按钮+,添加一个输出信号。
- 2. 打开变量向导。
- 3. 点击 Angular variables 页面。
- 4. 在左侧列表选中 Shaft,设置 Type of variable 为 Ang.velocity,

Component 为 Y。点击按钮⁵, 创建变量 om:y(Shaft), 如图 4.13

Coordinates Solver variables All forces Identifiers Variables for group of bodies T-Forces Joint forces Angular variables Image: Solver variables Selected Image: Selected Image: Selected Image: Selected Image: Solver variable Selected Image: Sel	Coordinates Solver variables All forces id Identifiers Variables for group of bodies T-Forces Joint forces Angular variables Image: Coordinates Selected Image: Coordinates Image: Coordinates Image: Coordinates Image: Coordinates Selected Image: Coordinates Selected Image: Coordinates Image: Coordinates Image: Coordinates Selected Shaft Image: Coordinates Image: Coordi	🛃 Linear variables		a+b Expression	User variables	🕀 Reactions	
Variables for group of bodies T-Forces Joint forces Angular variables Image: Selected Selected Image: Select	Image: Second	📫 Coordinates	۲	Solver variables	茸 All forces	id Identifiers	
Selected Shaft Shaft Shaft Type of variable ORot. vector ORot. vector Ang. velocity Component OX OX Y Resolved in SC of body Base0 Relative to body Base0 v(Shaft)	Selected Shaft Shaft Type of variable Ret. vector Ang. velocity Ang. acceleration Component X X Y Resolved in SC of body Base0 Relative to body Base0 v(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje Image: Staft	🥩 Variables for group of b	odies	C T-Forces	🔍 Joint forces	🛕 Angular variable	S
Shaft Shaft Type of variable Rot. vector Ang. velocity Ang. acceleration Component X Y Resolved in SC of body Base0 Relative to body Base0 I (Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje	Shaft Shaft Type of variable Resolved or SC of body Component X Image: Y X Image: Y Image: Y Resolved in SC of body Image: Y Image: Y Relative to body Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Shaft Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Image: Y Ima	dcmotor_cosim		Selected			^
Type of variable Rot. vector Rot. vector Ang. velocity Ang. acceleration Component X Y Resolved in SC of body Base0 Relative to body Base0 Image: Relative to body Image: Relative to body Base0 Image: Relative to body Image: Relative to Base0, SC Base0, proje Image: Relative to Base0, SC Base0,	Type of variable Rot. vector Rot. vector Ang. velocity Ang. acceleration Component X Y Resolved in SC of body Base0 Relative to body Base0 Relative to body Base0 Image: Angular velocity of body Shaft relative to Base0, SC Base0, proje Image: Staft)	Shaft		Shaft]
Component X O Z V V V Resolved in SC of body Base0 Relative to body Base0 :y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje	Component X Y Z IVI V Resolved in SC of body Base0 Image: Component in the second			Type of variable ORot. vector	Ang. velocity	O Ang. acceleration	
Resolved in SC of body Base0 Relative to body Base0 :y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje	Resolved in SC of body Base0 Relative to body Base0 rey(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje			Component O X O Y	Oz (
Base0 Relative to body Base0 :y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje	n:y(Shaft) Base0 Relative to body Base0			Resolved in SC of I	oody		
Relative to body Base0 :y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje	Relative to body Base0			Base0		-	
y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje	n:y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje			Relative to body			
:y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje	n:y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje			Base0		-	
:y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje 🚺 🗗	n:y(Shaft) Angular velocity of body Shaft relative to Base0, SC Base0, proje						~
	:y(Shaft)	n:y(Shaft)	Ar	ngular velocity of body	Shaft relative to Base(), SC Base0, proje 🗖	₩
:v(Shaft)		:v(Shaft)					

- 图 4.13 变量向导
- 5. 将变量 om:y(Shaft)从变量向导拖入输出向导的 Output1 处,作为输出 信号,如图 4.14 所示。



图 4.14





生成 m 文件

- 1. 在点击按钮, 弹出警告信息, 提示参数列表为空, 点击 Yes, 忽略 即可。
- 2. 弹出保存窗口,设置名称 dcmotor_cosim,点击 Save。





4.3.3 连接 UM 模型和 Matlab/Simulink 模型

- 1. 运行 Matlab/Simulink。
- 2. 加载图 4.11 所示的控制系统模型 dcmotor_cosim。
- 3. 双击 S-Function, 弹出参数设置窗口。
- 4. 设置名称为 dcmotor_cosim, 即 m 文件的名字, 如图 4.15 所示。
- 5. 点击 OK。

User-definal (Level-1), a standards. passed to th additional p If the S-fur building ger function mod extensions d 'src.c src1.	ole block. Blocks can be written in C and Fortran and must conform to S-func- The variables t, x, u, and flag are an the S-function by Simulink. You can sp parameters in the 'S-function parameter the block requires additional source merated code, specify the filenames in dules' field. Enter the filenames only or full pathnames, e.g., enter 'src sr c'.	, MATLAB tion ecify rs' field. e files for the 'S- ; do not use cl', not
Parameters	and a second	T dit
5-function :	name: dcmotor_cosim	Edit
S-function ;	parameters:	
S-function :	modules: ''	
_		
0	OK Cancel Help	Apply





4.3.4 运动仿真

运动仿真

- 1. 双击打开所有的 Scope 窗口。
- 2. 运行仿真。
- 3. 仿真结果如图 4.16 和图 4.17 所示。



Universal Mechanism 9