

基于快速控制原型的倒立机器人设计与实现

吕强, 张洋, 王国胜, 郭峰

装甲兵工程学院控制工程系, 北京 100072

摘要 介绍了快速控制原型的产生、发展和优势。搭建了倒立机器人快速控制原型硬件平台。针对硬件平台,总结了快速控制原型设计流程,设计了 Matlab/Simulink 模型,实现了自动代码生成,完成了倒立机器人的平衡控制。结果表明,倒立机器人能在 $-2^\circ \sim 2^\circ$ 范围内保持平衡,验证了快速控制原型的优势。在控制系统最初的设计阶段,将 Matlab/Simulink 模型在实物中进行验证,可缩短开发周期,节省经费,提高开发效率。

关键词 快速控制原型;自动代码生成;倒立机器人;Simulink

按照传统的 DSP 开发模式进行开发,需要工程师进行仿真、编程、查错及调试,是一个复杂而又长久的过程。当需要更新算法或出现错误时,需重新进行设计和实现。整个过程较为独立,且开发时间长,效率低。当今市场竞争激烈,产品开发要求在短时间、低成本的前提下完成。快速控制原型技术应运而生。快速控制原型技术将传统开发方式中的各个阶段有机的结合起来,在虚拟、可视的开发平台上进行算法设计、系统仿真,自动代码生成,快速地进行硬件测试,极大地缩短了开发周期,减少代码错误,提高开发效率。

倒立机器人是一种平衡机器人,是自治移动式机器人家族中重要的一类,是一种典型的多变量非线性系统,常常作为科研实验仪器使用。本文搭建了基于 DSP28069 的倒立机器人快速控制原型硬件平台,在 Matlab/Simulink 环境中建立了系统模型并进行了虚拟仿真,利用 Simulink 自动代码生成技术生成二进制代码下载到 DSP 中,进行实物实验,以实现倒立机器人控制系统的快速控制原型设计。

1 快速控制原型技术

1.1 快速控制原型技术简介

快速控制原型(rapid control prototyping, RCP)技术最初产生于制造业,被称为快速原型技术(rapid prototyping, RP),主要思想是:尽可能地在虚拟环境中完成产品设计,从而缩短开发周期、降低开发费用。经过多年发展,西方发达国家已经将该技术广泛应用于家电、航空、军事等领域。中国也在 20 世纪 90 年代引入此技术,目前在家电行业中发展较

快。将控制系统设计和控制算法验证引进 RP 技术中后,RP 技术即被称为快速控制原型技术^[1]。

1.2 快速控制原型技术的优点

传统的系统设计过程是一个串行开发模式,在设计开发过程中,当发现需求与系统设计不符,或者设计过程出现错误时,需要反馈到各阶段重新进行。除此之外,人工输入代码花费了大量的人力和时间,延长了整个开发进度。这种传统的串行开发模式极大限制了系统开发效率。而快速控制原型技术可以很好地改善这些缺陷^[2]。

快速控制原型将传统开发模式中的几个相互独立的阶段有机的结合了起来,其开发过程是一个并行进行的过程,有效实现了不同阶段的无缝集成^[3]。在最初的设计阶段即可在实物中进行各种验证,在开发早期减少或消除可能的错误及缺陷,从而达到节省开支,降低物耗的目的。快速控制原型开发技术利用 Matlab/Simulink 等开发平台建立系统模型,自动生成代码,可以产生高效无误的代码,极大地提高了开发效率。同时,建立的系统模型可以根据各种硬件平台生成相应代码而广泛应用,性价比高、扩展性好、维护方便。

2 倒立机器人建模

2.1 轮式倒立机器人的运动方程

如图 1 所示,建模轮式倒立机器人。设轮式倒立机器人本体姿态倾角从垂直方向顺时针旋转为正。车轮相对于本体转过的角度,顺时针为正。据此规定坐标系,推导轮式倒立机器人运动方程为式(1)和式(2)。

收稿日期:2015-07-02;修回日期:2016-01-28

作者简介:吕强,教授,研究方向为机器人控制技术,电子信箱:rokyou@live.cn

引用格式:吕强,张洋,王国胜,等.基于快速控制原型的倒立机器人设计与实现[J].科技导报,2016,34(18):265-269;doi:10.3981/j.issn.1000-7857.2016.18.036

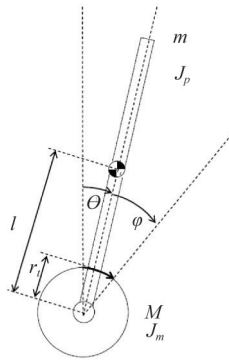


图1 倒立机器人模型

Fig. 1 Model of inverted robot

图1中, m 为本体质量, M 为整体质量, l 为重心到轮轴的距离, J_p 为本体转动惯量, J_m 为电机转子转动惯量, r_i 为车轮半径, θ 为车体相对于车轮转过的角度, φ 为车体的倾斜角。

$$\{(M+m)r_i^2 + mlr_i \cos \theta + J_i + iJ_m\} \ddot{\theta} - mlr_i \sin \theta \cdot \dot{\theta}^2 + \{(M+m)r_i^2 + J_i + i^2 J_m\} \ddot{\varphi} + c\dot{\varphi} = au \quad (1)$$

$$\{(M+m)r_i^2 + 2mlr_i \cos \theta + ml^2 + J_p + J_i + J_m\} \ddot{\theta} - mlr_i \sin \theta \cdot \dot{\theta}^2 - mgl \sin \theta + \{(M+m)r_i^2 + mlr_i \cos \theta + J_i + iJ_m\} \ddot{\varphi} = 0 \quad (2)$$

式中, u 为电机的电流参考值; a 为从 u 到车轮转矩的增益; J_i 为整体转动惯量; c 为轮轴摩擦力; i 为齿轮传动比。

2.2 运动方程的线性化与状态方程

倒立机器人实现直立时, θ 只在 0 附近取值, 所以有 $\sin \theta \approx \theta$, $\cos \theta \approx 1$ 。进一步假设 $\dot{\theta}$ 很小, 可设 $\dot{\theta}^2 \approx 0$, 于是, 运动方程(1)和(2)的线性化方程为

$$\{(M+m)r_i^2 + mlr_i + J_i + iJ_m\} \ddot{\theta} + \{(M+m)r_i^2 + J_i + i^2 J_m\} \ddot{\varphi} + c\dot{\varphi} = au \quad (3)$$

$$\{(M+m)r_i^2 + 2mlr_i + ml^2 + J_p + J_i + J_m\} \ddot{\theta} - mgl\theta + \{(M+m)r_i^2 + mlr_i + J_i + iJ_m\} \ddot{\varphi} = 0 \quad (4)$$

在此, 为了应用现代控制理论, 把上述线性化的运动方程表达成状态空间描述。首先把式(3)和式(4)写成矩阵形式:

$$\alpha \begin{bmatrix} \ddot{\theta} \\ \ddot{\varphi} \end{bmatrix} + \beta \begin{bmatrix} \dot{\theta} \\ \dot{\varphi} \end{bmatrix} + \gamma \begin{bmatrix} \theta \\ \varphi \end{bmatrix} = \delta u \quad (5)$$

式中, 各系数分别为

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}; \alpha_{11} = (M+m)r_i^2 + mlr_i + J_i + iJ_m; \alpha_{12} = (M+m)r_i^2 + J_i + i^2 J_m; \alpha_{21} = (M+m)r_i^2 + 2mlr_i + ml^2 + J_p + J_i + J_m; \alpha_{22} = (M+m)r_i^2 + mlr_i + J_i + iJ_m; \beta = \begin{bmatrix} 0 & c \\ 0 & 0 \end{bmatrix}; \gamma = \begin{bmatrix} 0 & 0 \\ -mgl & 0 \end{bmatrix}; \delta = \begin{bmatrix} a \\ 0 \end{bmatrix}。$$

因此, 可得

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \varphi \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} \\ -\alpha^{-1} \gamma & -\alpha^{-1} \beta \end{bmatrix} \begin{bmatrix} \theta \\ \varphi \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0_{2 \times 2} \\ -\alpha^{-1} \delta \end{bmatrix} u$$

即获得状态空间描述:

$$\dot{x} = Ax + Bu \quad (6)$$

其中,

$$x = \begin{bmatrix} \theta \\ \varphi \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix}, A = \begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} \\ -\alpha^{-1} \gamma & -\alpha^{-1} \beta \end{bmatrix}, B = \begin{bmatrix} 0_{2 \times 2} \\ -\alpha^{-1} \delta \end{bmatrix} \quad (7)$$

3 倒立机器人快速控制原型设计

3.1 倒立机器人硬件设计

倒立机器人概要和实物如图2所示。机器人整体包括本体和两个轮子, 内部从上至下分别为各电路模块、11.1 V 可充电锂电池、两个直流电机和编码器。中间电路板上安装有电源模块、电机驱动模块、DSP 核心板、陀螺加速度计。硬件设计遵循质量分布均衡, 重心尽量偏下, 接线方便的原则。

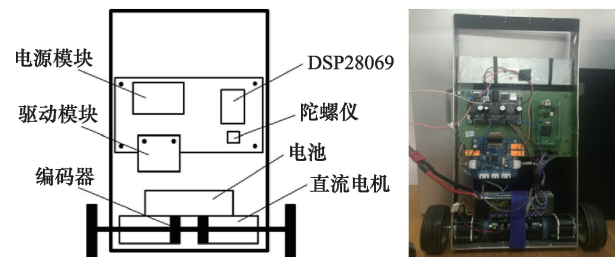


图2 倒立机器人

Fig. 2 Inverted robot

DSP 采用方便高效的德州仪器(Texas Instrument)公司生产的 32 位微处理器 TMS320F28069, 其具有程序编写简易、供电简单、控制可靠等等优点。它的处理器是在以往的基础上增加浮点处理能力的高性能处理器, 工作频率可达 90 MHz, 并且增加了可编程平行加速器, 实现了独立于主 CPU 之外的代码执行^[4]。

使用陀螺加速度计 MPU6050 获取车体倾角和信息。通过模块内处理器读取 MPU6050 的测量数据然后通过串口输出。模块如图 3 所示。模块内部自带电压稳定电路, 兼容 3.3 V/5 V 的嵌入式系统。模块采用数字滤波技术, 能有效降低测量噪声, 提高精度。内部集成姿态解算算法, 配合动态

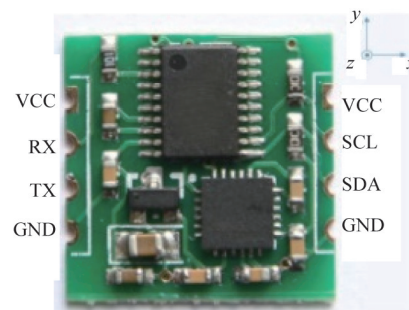


图3 倾角模块

Fig. 3 Dip angle module

卡尔曼滤波算法,能在动态环境下准确输出模块当前姿态,精度0.01°,稳定性高。通信协议如表2和表3所示。倒立机器人仅使用x轴数据即可。计算公式为:

$$\theta = ((RollH \ll 8) | RollL) / 32768 \times 180$$

$$\dot{\theta} = ((RollH \ll 8) | RollL) / 32768 \times 2000$$

表2 角度数据通信协议
Table 2 Angle protocol

数据编号	数据内容	含义
0	0x55	包头
1	0x53	标识角度包
2	RollL	x轴低字节
3	RollH	x轴高字节
4	PitchL	y轴低字节
5	PitchH	y轴高字节
6	YawL	z轴低字节
7	YawH	z轴高字节
8	TL	温度低字节
9	TH	温度高字节
10	Sum	校验和

表3 角速度数据通信协议
Table 3 Angular velocity

数据编号	数据内容	含义
0	0x55	包头
1	0x52	标识角速度包
2	wxL	x轴低字节
3	wxH	x轴高字节
4	wyL	y轴低字节
5	wyH	y轴高字节
6	wzL	z轴低字节
7	wzH	z轴高字节
8	TL	温度低字节
9	TH	温度高字节
10	Sum	校验和

电机驱动使用Neurons智能PID电机驱动模块,如图4所示。模块自带控制器进行PID运算、梯形图控制,由L298N来进行直流电机驱动。通过串口发送8字节命令来控制双路电机的正反转速度,同时通过命令获取编码器速度。

机器人系统结构如图5所示。在系统中,计算机与DSP之间采用USB通信,陀螺模块和电机驱动模块都与DSP之间采用SCI串行通讯。陀螺加速度计通过SCI串口发送数据到DSP。DSP通过SCI串口发送命令到电机驱动模块控制双电

机转速和接收速度信息。在实验过程中,计算机可以实时得到机器人倾角信息。

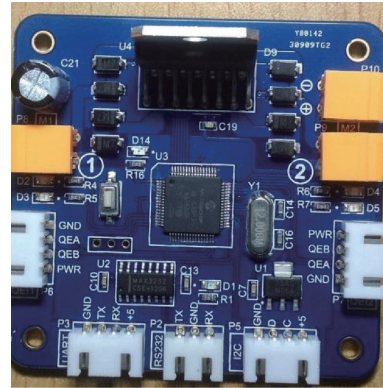


图4 电机驱动模块

Fig. 4 Motor drive module

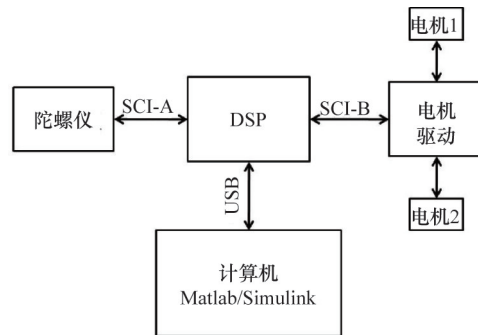


图5 系统结构

Fig. 5 System structure

3.2 快速控制原型设计流程

如图6所示,应用快速控制原型开发的方法设计倒立机器人控制系统的流程是:在Matlab/Simulink中针对设计好的倒立机器人硬件平台进行整体模型建立,完成后对模型进行虚拟仿真,虚拟仿真通过之后,即可自动代码生成,得到对应的C代码。通过CCS(code composer studio)将C源代码下载

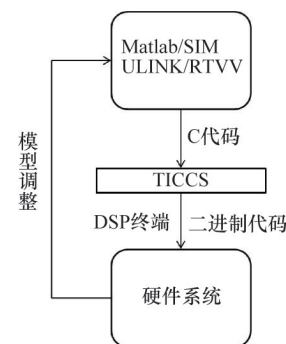


图6 开发流程

Fig. 6 Development process

到 DSP 中运行,得到运行结果^[5-6]。对结果进行分析,进而修正相应模型,再次生成代码,下载程序。控制系统在实物中不断地实验、验证,在仿真环境 Matlab/Simulink 中不断地进行更正、修改的迭代过程,直到得到最优结果。

3.3 模型设计和自动代码生成

传统的 DSP 开发方法需要在 CCS 中利用 C 语言或者汇编语言进行编程,编写大量代码。而快速控制原型开发只需要利用 Matlab/Simulink,建立图形化的模型,再自动生成代码即可。

利用 Matlab/Simulink 软件,需要完全安装软件并激活,之后安装针对 CCS 和 C2000 编译器的附加包。安装完成后,即可进行模型设计。

模型设计以硬件之间信号流向和控制策略为主要根据。本文所用硬件系统,数据通过 SCI 串行口进行传输,模型以串行通讯的配置为主。整体模型如图 7 所示。通过 SCI Receive 接收到的数据,提取其中角度和角速度的高低位,根据上文所列公式进行计算,得到车体倾角和角速度。通过 SCI Receive1 接收到的编码器速度高低位,进行移位和按位与运算,得到车轮速度值,再利用速度值求出车轮角速度。对车轮角速度进行积分,得到车轮的转角。得到 4 个输入量之后,利用状态方程,得到输出量,转换成 32 位整型数据,再分解为 4 个 8 位整型数据,和读取速度的命令同时通过 SCI Transmit2 发送到电机驱动模块。电机根据输出量进行转动,维持机器人直立。

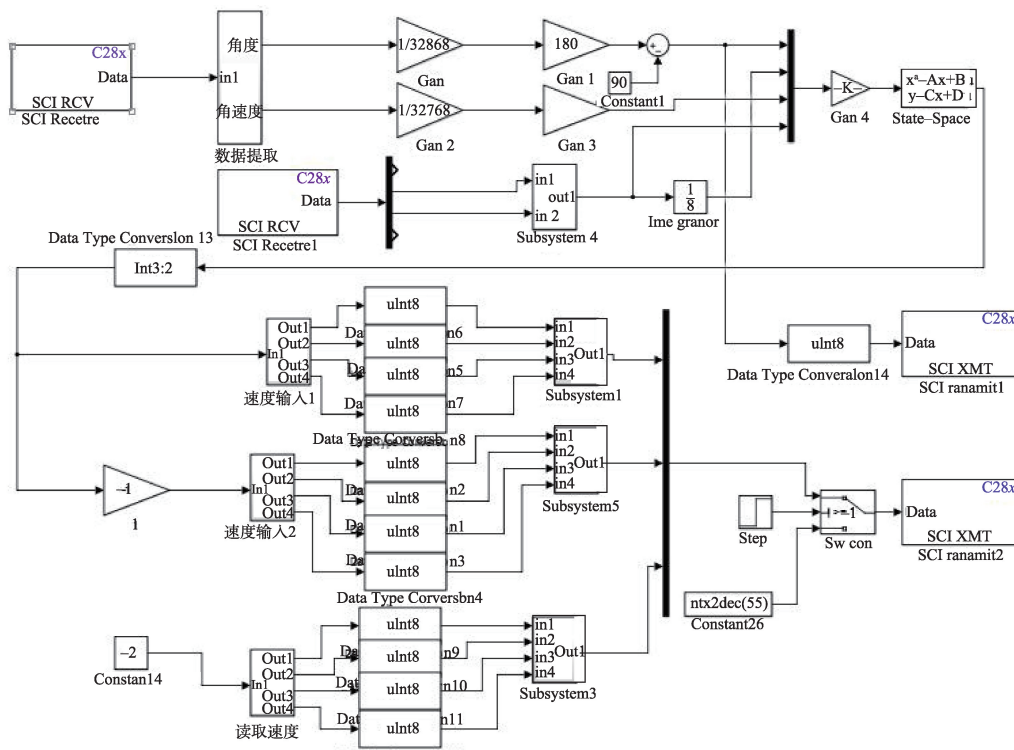


图 7 机器人整体模型

Fig. 7 Robot model

完成模型设计后,需要正确设置仿真参数,才能自动生成针对于 DSP 的可执行代码。在 Simulink 界面下打开设置 Model Configuration Parameters。在 Select 中选中 Solver, Solver options 中的 Type 选择成 Fixed-step 固定步长, Solver 选择为 discrete 离散。只有固定步长才能生成代码。在 Select 中选中 Code Generation, 浏览 System target file 设置为 ert.tlc Embedded Coder。Language 设置成 C 语言,生成的是 C 语言代码。Target hardware 选择 TI Piccolo F2806x, 硬件采用的是 DSP28069。Toolchain 工具链选择 TI CCSv5 (C2000), 系统采用 CCSv5 编译环境。Coder Target 中可以设置具体的引脚使能。

完成模型设计和仿真参数设置后,进行虚拟仿真,根据仿真结果对模型进行更正,直到虚拟仿真通过,即可生成代码。代码生成 .out 文件即 DSP 可执行的二进制代码。

4 实验与分析

将在 Matlab/Simulink 中生成的 .out 文件通过 CCS 下载到 DSP 中,运行倒立机器人。根据机器人运行情况对模型或者参数进行更正,再次生成代码下载到 DSP 中,直到机器人可以稳定平衡。在机器人稳定运行过程中,角度信息不断地通过串口发送到计算机上位端,通过软件串口助手可以实时看到车体倾角的值在 0° 左右变化。实验结果如图 8 所示,数据

采样周期为0.01 s。

从图8可以得出,机器人倾角在 $-2^{\circ}\sim 2^{\circ}$ 之间震荡,震荡频率较为规律,波动较小。实际中机器人能够长时间稳定直立,与数据相符。说明倒立机器人可以在 2° 范围内实现直立。针对倒立机器人的控制系统快速控制原型设计成功实现。

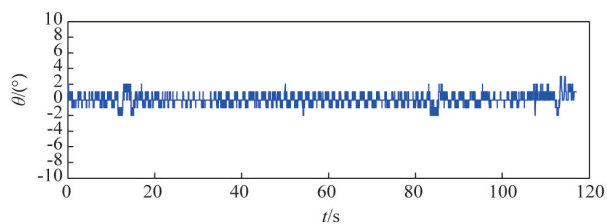


图8 机器人倾角
Fig. 8 Robot dip angle

5 结论

针对基于DSP的倒立机器人控制系统快速控制原型进行了研究,阐述了快速控制原型技术的概念,总结了整体思想,给出了开发方法和流程。介绍了倒立机器人硬件系统平台,设计了针对倒立机器人平衡直立的Simulink模型,生成了C代码,完成了倒立机器人的平衡控制,实现了倒立机器人在 2° 范围内的稳定。通过对模型的虚拟仿真和实物验证,完成了快速控制原型的整个过程。研究表明:快速控制原型(RCP)技术通过在可视化的图形界面进行系统设计,随时在实物中进行实物仿真,大大加快了软硬件的开发速度,提升了开发人员开发效率,提高了DSP开发可靠性和一致性,达

到了节约研究经费,降低成本的目的。快速控制原型技术确实是一种高效、准确、方便的设计方法。

参考文献(References)

- [1] 丁荣军. 快速控制原型技术的发展现状[J]. 机车电传动, 2009(4): 1-3.
Ding Rongjun. Development of rapid control prototyping technology[J]. Electric locomotive, 2009(4): 1-3.
- [2] 李强, 王民钢, 杨尧. 飞行控制系统快速原型设计与实现[J]. 计算机测量与控制, 2009, 17(7): 1305-1307.
Li Qiang, Wang Mingang, Yang Yao. Rapid prototyping design and implementation of flight control system[J]. Computer Measurement and Control, 2009, 17(7): 1305-1307.
- [3] 殷红珍, 唐硕, 黄勇, 等. 组合导航系统快速原型设计[J]. 计算机辅助工程. 2008 17(03): 45-48.
Yin Hongzhen, Tang Shuo, Huang Yong, et al. Rapid prototyping design of integrated navigation system[J]. Computer aided Engineering, 2008 17(03): 45-48.
- [4] 吕红涛, 王国胜, 吕强, 等. 基于TMS320F28069的自平衡机器人控制系统设计[J]. 单片机与嵌入式系统应用, 2013, 12(9): 42-45.
Lü Hongtao, Wang Guosheng, Lü Qiang, et al. Design of self balancing robot control system based on TMS320F28069[J]. MCU and embedded system application, 2013, 12(9): 42-45.
- [5] 洪晓君, 朱磊. 基于MATLAB/RTWEC自动代码生成技术整车控制器快速原型开发[J]. 机电一体化. 2014(8):47-51.
Hong Xiaojun, Zhu Lei. Vehicle controller rapid prototype based on the automatic code generation technology on MATLAB/RTWEC[J]. Mechatronics. 2014(8):47-51.
- [6] 王广玮, 赵津, 张向南, 等. dSPACE快速控制原型技术在无刷直流电动机控制中的应用[J]. 现代制造工程, 2015(1): 20-23.
Wang Guangwei, Zhao Jin, Zhang Xiangnan, et al. Application of dSPACE rapid control prototyping technology in brushless DC motor control[J]. Modern Manufacturing Engineering. 2015(1):20-23.

Inverted robot design and realization with rapid control prototyping

LÜ Qiang, ZHANG Yang, WANG Guosheng, GUO Feng

Department of Control Engineering, Academy of Armored Forces Engineering, Beijing 100072, China

Abstract Rapid control prototyping is introduced in terms of its generation, development and advantage. A hardware platform for inverted robot's rapid control prototyping is established and its design flow is summarized. Then a Matlab/Simulink model is designed and automatic code generation is realized to control the inverted robot. Simulation results show that the inverted robot can keep balance in the range between -2 and 2 degrees, which demonstrates the advantage of the rapid control prototyping. In the initial step of the design procedure the Matlab/Simulink model can be tested in real situation so as to reduce the design cycle, save cost and speed up development efficiency.

Keywords rapid control prototyping; automatic code generation; inverted robot; Simulink

(责任编辑 刘志远)