

基于行为检测的机器人安全保护方法

戈斌斌 毛新军

(国防科学技术大学计算机学院 长沙 410073)

摘要 机器人是一类典型的信息物理融合系统,由软件控制的机器人行为存在多种形式的安全问题。为了避免这些问题,提出一种基于行为检测的机器人安全保护方法。行为检测是指通过检测机器人执行行为之前和之后的状态,判断机器行为是否安全。该方法针对硬件限制、稳定性和碰撞冲突 3 个方面的行为安全问题,支持机器人执行行为之前对其进行建模和检测,提出碰撞冲突检测算法 ACD。基于 AOP 技术实现该安全保护方法。基于 NAO 机器人开展了若干场景的实验,结果表明该方法可有效检测出危害机器人安全的行为并及时做出相应动作补偿。

关键词 行为检测,机器人安全,AOP

中图法分类号 TP311 文献标识码 A DOI 10.11896/j.issn.1002-137X.2016.8.037

Method for Ensuring Safety of Robot Based on Behavior Detection

GE Bin-bin MAO Xin-jun

(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract Robot is a typical cyber-physical system and its behaviors that are driven and controlled by software are highly related with safety. In order to ensure the safety of robot behaviors, this paper proposed a security protection method based on behavior detection that enables robot to detect its behaviors' safety before their executions. Three kinds of safety issues were considered, including hardware constraints, stability and collision conflict. The corresponding algorithm called ACD algorithm for collision conflict detection was proposed. Implementation framework for the method was proposed on the basis of AOP technology. We made some experiments with NAO robots. The results indicate that the method can effectively detect harmful behaviors and adjust motions in compensation for behaviors.

Keywords Behavior detecting, Robot safety, AOP

1 引言

当前机器人技术发展迅速,在各个领域中应用广泛(如机械加工、医疗、军事、家庭服务等),机器人安全问题也日益凸显^[1]。机器人的自由度较大且具有高速运动的大功率部件,有些机器人需要与人交互^[2],若机器人发生故障,可能会造成严重的危害,甚至威胁到人的生命安全。

案例一:1983年,25岁的美国福特工厂装配线工人 Robert Williams,在密歇根州的福特铸造厂被工业机器人手臂击中身亡。

案例二:美国达芬奇外科机器人在 2001 年到 2013 年之间,手术伤亡事故数量为 245 起,其中 71 起为死亡事故。

从案例一和案例二可知机器人安全事故的形势较为严峻。日本某公司对工业机器人发生安全事故的原因进行调查,其中机器人故障占 85.4%,外界环境干扰导致机器人失控占 11.1%,其他原因(如操作不当)占 3.5%^[3]。机器人故障是导致机器人安全事故发生的主要原因,如何保障机器人的安全,确保其在工作过程中不发生故障是学术界和工业界一个亟待解决的问题。

据相关调查分析,机器人错误的行为导致机器人故障占 66.4%^[3],而机器人的错误行为来自于机器人的控制软件。在机器人控制软件方面,常见的保护措施有两种:1)加强机器人控制软件的测试,降低机器人错误行为发生的概率;2)通过使用机器人模拟平台进行模拟实验,确保控制软件安全之后才将其部署到机器人上(如 NAO 机器人的模拟器 Webots^[16])。第一种方法存在的问题是当软件的规模变大时,操作难度大,而且过度依赖于经验,存在较大的不确定性。第二种方法存在的问题是:首先需要针对不同的机器人开发出不同的机器人模拟平台,增加了机器人开发成本;此外,机器人的开放动态的工作环境,导致模拟难度高、存在的误差大等问题。

针对机器人控制软件中存在的可能会导致机器人故障的缺陷,提出一种基于行为检测的机器人安全保护方法。该方法的基本原理是:在机器人执行可能会危害机器人安全的行为之前,对该行为进行安全检测,如果检测出其会危害机器人的安全,则做出相应的动作补偿或停止该行为并进入保护态。本文基于面向方面编程技术(AOP)^[9]实现该机器人的安全保护方法,通过预编译将安全检测代码织入到机器人控制软

到稿日期:2015-07-07 返修日期:2015-09-25 本文受国家自然科学基金(61379051)资助。

戈斌斌(1991—),男,硕士生,主要研究方向为智能计算,E-mail:20091726@bit.edu.cn;毛新军(1970—),男,博士,教授,CCF 会员,主要研究方向为面向 Agent 的软件工程。

件中,降低了软件代码的耦合性,提高了机器人控制软件的可靠性,实现了保护机器人安全的目的。

本文第2节介绍机器人安全保护的相关工作;第3节阐述机器人行为的建模与检测方法;第4节实现机器人的安全检测方法;第5节开展相关实验并对实验结果进行分析;第6节将提出的机器人安全保护方法与其他方法进行对比;最后对本文进行总结,并展望下一步的研究工作。

2 相关工作

本文的研究工作是为保护机器人的安全及确保机器人在工作过程中不发生故障而提出一种机器人安全保护方法。当前机器人安全保护的研究主要集中在机器人安全性分析与评估、事故分析、安全性验证等方面^[19],如 Dhillon 等提出一种经过容错设计的机器人安全分析方法^[7],Rahimi 等提出基于遗传算法的机器人安全验证方法^[6]。上述方法主要是对机器人进行静态分析,如在机器人工作之前对机器人控制系统与硬件系统进行安全分析,降低了机器人发生故障的概率,但是由于机器人的工作环境是开放的、动态的,部分机器人故障是无法提前预测和分析的。

针对机器人安全的静态分析方法存在的弊端,文献^[4]提出一种“虚拟限位”的机器人安全保护策略。如图1所示,整个安全保护系统是一个实时多任务系统。软限位计算模块通过处理传感器系统检测到的实时数据,计算受限变量(如关节角、电机速度等)。机器人控制系统将结果发送给硬件保护模块,硬件保护模块通过启动机械保护装置保护机器人的安全。文献^[5]等针对工业焊接机器人安全保护问题,提出类似的机器人安全保护方法——“运动参数限制”。通过安全控制器监视机器人的工作状态,计算机器人各关节角度、末端实际位置和姿态,并实时限制机器人关节的运动速度、加速度等参数,从而提高机器人工作的稳定性与安全性。这两种方法都是在机器人工作时通过实时监测机器人的运行参数并根据机器人的运行状态调整或限制运行参数数值的方法,旨在保护机器人的安全。但这两种方法存在以下不足:依赖于机器人硬件,影响机器人正常执行任务的效率,因过度保护而影响任务的正常执行等。本文提出的基于行为检测的机器人安全保护方法可以有效地弥补上述不足。

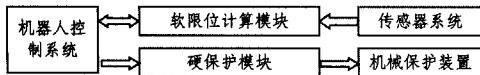


图1 基于“虚拟限位”的安全保护系统

3 机器人行为建模与检测

如果机器人在控制软件中存在不正确或冲突的程序指令,可能会导致机器人发生故障,进入非安全状态。机器人的安全状态是指在特定时间内机器人不会受到损坏或发生故障。机器人的非安全状态主要包括硬件超限、失去平衡、碰撞冲突等。为了有效避免机器人进入非安全状态,本文提出机器人的安全保护方法,如图2所示。在机器人控制软件的功能模块中可能会存在两种行为:安全行为与安全可疑行为。其中安全可疑行为是指对机器人自身安全存在安全隐患的机器人行为,是在机器人执行不正确或冲突的程序指令时产生的行为,如改变机器人某个关节角、提高机器人运动速度等。

从软件的角度,如何进行有效的行为补偿,使机器人可以消除上述安全隐患,是本节需要解决的问题。

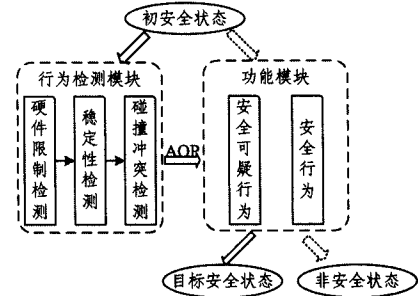


图2 机器人自身安全保护方法

在机器人执行功能模块的指令之前,进行硬件限制检测、稳定性检测、碰撞冲突检测,避免机器人进入不安全状态,从而达到保护机器人的目的。本文以 NAO 机器人^[16]为例,行为检测的过程如图3所示。机器人依次执行摇头(moveHead)、设置关节角度(setAngles)与改变姿势(goToPosture)3个行为,其中摇头是安全行为,设置关节角与改变姿势是安全可疑行为。由于设置关节角可能会导致机器人失去平衡,在其执行之前,通过AOP的代码织入机制^[9]将稳定性检测的代码织入到设置角度之前,完成稳定性检测之后,执行设置角度行为。同理,将碰撞冲突检测的代码织入到改变姿势之前。

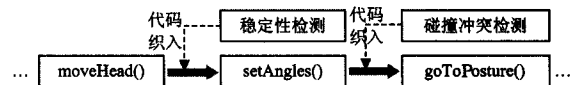


图3 行为检测的过程

3.1 硬件限制检测

机器人硬件限制是指机器人硬件工业技术存在的物理限制^[15],例如机器人的关节角转动范围、电机转速最大值都是存在限制的。虽然一般机器人制造厂家对机器人的硬件会有一些保护措施,但是机器人控制软件的开发人员对机器人的硬件参数可能不熟悉,导致在控制软件中存在超过机器人硬件限制的程序指令。频繁超限会在一定程度上损伤机器人硬件,而且会使机器人控制软件达不到预期的执行效果。为了避免机器人硬件超限,在机器人行为执行之前计算机器人参数的改变值,然后结合机器人硬件的当前参数,判断机器人硬件是否超限。

以机器人的关节角和电机速度为例,根据机器人硬件参数在 t 时刻和经过 Δt 后机器人的参数,讨论硬件超限问题。设机器人在 t 时刻的关节角为 $\theta(t)$,在 $\Delta(t)$ 时间内变化 $\Delta\theta$ 。在 t 时刻电机速度为 $\omega(t)$,在 $\Delta(t)$ 时间内变化 $\Delta\omega$ 。通过不等式(1),来判断机器人关节角和电机速度是否超限。

$$\theta_i^{\min} \leq \theta_i(t) + \Delta\theta_i \leq \theta_i^{\max}, i=1, \dots, n \quad (1)$$

其中, θ_i^{\max} 和 θ_i^{\min} 分别是机器人第 i 个关节角能达到的最大值和最小值, n 为机器人关节角的个数。同理可以得出电机速度的约束条件:

$$\omega_i^{\min} \leq \omega_i(t) + \Delta\omega_i \leq \omega_i^{\max}, i=1, \dots, n \quad (2)$$

其中, ω_i^{\max} 和 ω_i^{\min} 分别是机器人第 i 个电机速度能达到的最大值和最小值, n 为机器人电机的个数。假设在 t 时刻机器人的硬件没有超限,式(1)与式(2)可以确保机器人在 Δt 时间后仍未超限,依此类推可以确保机器人在完成行为的整个过程中均是安全的。在完成上述的约束后,如果存在超限的行为,

则采用最大值,最大程度接近控制软件的需求。由于硬件限制问题本质上是约束问题,本文不单独提出相应算法。

3.2 稳定性检测

在机器人的设计阶段,针对机器人运动过程中的稳定性及控制规律设计机器人的动力学模型^[15]。但是设计出的动力学模型只能确保机器人存在若干稳定的状态,不能确保机器人工作时一直保持稳定状态。机器人的稳定性问题^[15]对机器人自身的安全有较大的威胁,如机器人失去平衡导致摔倒等,造成机器人的硬件损坏。

由于机器人的硬件较为复杂,维持机器人稳定是一个难题。本文的关注点是如何检测出机器人的不稳定状态,在检测出之后,直接停止机器人的任务,使其进入保护状态。

机器人在执行任务时处于运动状态,各个关节、部件的受力情况是时刻变化的,增加了判断机器人稳定性的难度^[17]。目前机器人稳定性判断标准主要有 Zero Moment Point (ZMP)^[10]、Foot Rotation (FRI)^[11]、Center of Pressure (CMP)^[12]等。本文使用 ZMP 对机器人进行稳定性分析,在此之前需要对运动的机器人进行有效的建模。如图 4 所示,在机器人稳定性模型中需要建立两个坐标系:

1) 机器人坐标系,坐标的原点在机器人的髋部,如图 4(a)所示;

2) 脚掌坐标系,坐标原点在踝关节中心,如图 4(b)所示。

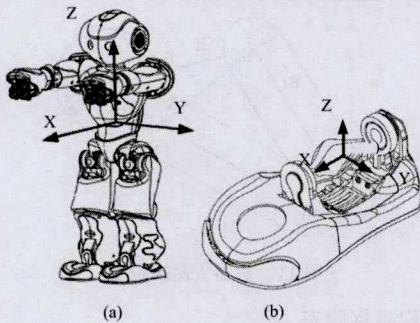


图 4 稳定性模型坐标系

机器人坐标系和脚掌坐标系通过 Denavit-Hartenberg (D-H)模型^[13,14]进行转换。

ZMP 是零力矩点,用于判断机器人稳定性,其定义为:机器人支撑脚掌与地面接触的一点,在该点支撑脚受到的地面反作用力的等效力矩的水平分量为零^[11]。若 ZMP 点位于机器人支撑脚掌与接触地面形成的凸多边形内,则机器人处于稳定状态;否则,机器人处于不稳定状态^[15]。

一般情况下,在机器人的脚掌上存在力学传感器,它可以获取机器人脚掌上的受力和力矩数据。根据上述数据和机器人稳定性模型可以计算 ZMP。由于机器人单脚支撑时更容易失去平衡,因此本节以单脚支撑为例。

如图 5 所示,机器人脚掌坐标系的原点在 O 点,S 点是脚掌力学传感器的位置 $[0 \ 0 \ S_z]$,在此点受到的力 F 和力矩 M 可以从传感器中获取,表示如下:

$$F = [F_x \ F_y \ F_z] \quad (3)$$

$$M = [M_x \ M_y \ M_z] \quad (4)$$

设 P 点为零力矩点 (ZMP),P 点的坐标为 $[P_x \ P_y \ P_z]$,绕 P 点的力矩可以表示为:

$$M_P = \overrightarrow{PS} \otimes F + M \quad (5)$$

根据 ZMP 点的定义,力矩的水平分量为 0,可以得到(设 SQ 的长度为 d):

$$P_x = (M_y + F_x d) / F_z \quad (6)$$

$$P_y = (M_x - F_y d) / F_z \quad (7)$$

所以 P 点在机器人脚掌坐标系内的坐标为: $[(M_y + F_x d) / F_z \ (M_x - F_y d) / F_z \ -h]$,h 为线段 OQ 的长度。

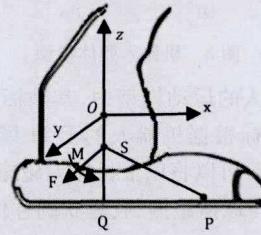


图 5 ZMP 分析

在计算 ZMP 点完成之后,需要判断 ZMP 点与机器人支撑脚掌与接触地面形成的凸多边形的位置关系。如果 P 点在凸多边形外,则机器人将失去稳定性。如果 P 点在凸多边形内,说明机器人目前是稳定的,稳定性的与 P 点到凸多边形的最小距离 d 有关(见图 6),d 越大则机器人的稳定性越强,反之越小,所以当 d 小于某个阈值时,机器人有可能会进入不稳定状态,则停止当前任务,使机器人进入保护状态。

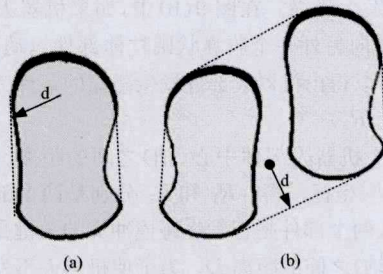


图 6 ZMP 点的最小距离

3.3 碰撞冲突检测

机器人通过改变各个关节的角度,来改变其各个部件的位置,从而完成某项特定的任务。如果在控制软件的实现过程中,只考虑任务,忽略机器人自身的硬件环境,可能会导致机器人硬件之间的碰撞冲突。由图 7 所示,当机器人在执行任务时,需要将手臂由身体前方移动到身体后方(由图 7(a)到图 7(b)),如果只是简单地改变某个关节的角度,会导致碰撞进而损坏机器人硬件。为了避免上述情况的发生,以 NAO 机器人为例建立数学模型,并提出解决问题的方案。

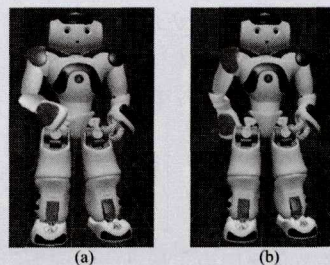


图 7 碰撞冲突场景

3.3.1 碰撞冲突建模

NAO 机器人共有 36 个关节^[16],胳膊和腿部的关节是可

能发生碰撞冲突的主要关节。将胳膊和腿部的关节抽象为胶囊状的圆柱体,如图 8(a)所示;将机器人的头部、胸部和臀部抽象为球体,如图 8(b)所示。

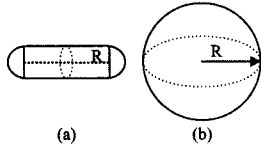


图 8 机器人躯体模型

在 NAO 机器人的运动模型中,其坐标原点规定在胸部中央,其他点的坐标根据机器人的 D-H 模型^[14]可以得出。根据上述抽象模型,可以将以上机器人碰撞的情形总结为两种:胶囊状圆柱体与球体碰撞、胶囊状圆柱体之间碰撞,如图 9 所示。

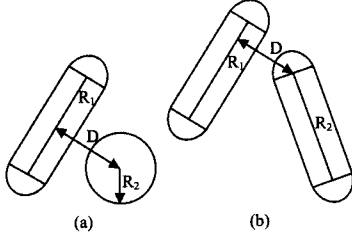


图 9 碰撞冲突模型

在图 9(a)中,如果机器人胶囊状圆柱体部件继续向球体部件运动,则发生碰撞。在图 9(b)中,如果机器人胶囊状圆柱体部件继续向另外一个胶囊状圆柱体部件运动,则发生碰撞。从图 9 可以得出机器人部件发生碰撞的条件为:

$$D - (R_1 + R_2) < 0 \quad (8)$$

其中, D 为两个机器人部件中心(轴)之间的距离。 D 可以通过 D-H 模型^[14]坐标求得。 R_1 和 R_2 分别是两个部件的半径长度。机器人两个部件是否发生碰撞冲突的关键是两个机器人部件中心(轴)之间的距离 D 。为了使机器人不发生碰撞冲突,在运动时必须实时获取机器人部件的位置与速度。下面介绍如何获取机器人部件的运动速度。机器人是通过各个关节的关节角的变化来实现其运动,机器人可以实地地获取每个关节的角度。为了避免机器人部件间的碰撞,需要预测机器人部件在操作空间的位置,并将机器人关节空间映射到操作空间。可以根据机器人的硬件数据和关节空间数据来获取机器人操作空间数据。

如图 10(a)所示,假设在机器人手臂上存在两个关节角 θ_1 和 θ_2 。一般情况下一个机器人关节只能在二维空间中运动,所以可以将手臂抽象成图 10(b)中的模型,可计算出手在操作空间中的坐标 (x, y) :

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (9)$$

$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (10)$$

等式两边同时对时间 t 求导得:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} \begin{bmatrix} \frac{d\theta_1}{dt} \\ \frac{d\theta_2}{dt} \end{bmatrix} \quad (11)$$

机器人的关节角速度为 $\vec{\omega}$, 运动线速度为 \vec{v} :

$$\vec{\omega} = \begin{bmatrix} \frac{d\theta_1}{dt} \\ \frac{d\theta_2}{dt} \end{bmatrix}, \vec{v} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} \quad (12)$$

根据式(11)、式(12)可以得出:

$$\vec{v} = J \vec{\omega} \quad (13)$$

其中, J 是机器人手臂的雅可比(Jacobian)矩阵^[14]。式(13)反映了机器人关节空间到操作空间的映射(线性变换),可以由有两个关节角的机器人关节空间推广到有多个关节角的关节空间,可以由二维机器人操作空间推广到三维操作空间。

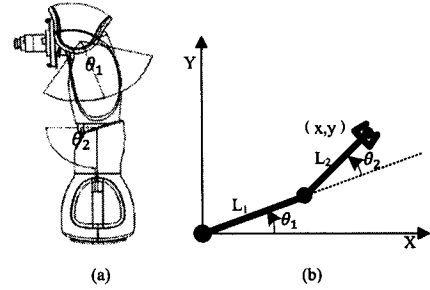


图 10 机器人手臂关节

在机器人执行某个可能导致部件之间碰撞的行为之前,首先对此行为进行预测、判断,如果不会发生碰撞,则执行;如果会发生碰撞,则选择一个与该行为最相近且不会发生碰撞的行为执行,具体的过程包括获取行为的目标状态、计算机器人部件将要运动的距离、判断机器人部件是否会发生碰撞、计算最优解、调整目标行为的状态和执行目标行为。机器人部件发生碰撞的模型如图 11 所示。

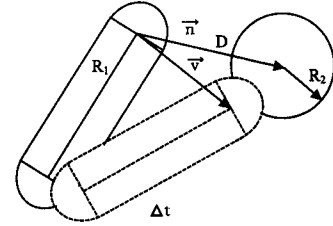


图 11 机器人碰撞模型

(1) 获取目标状态

机器人的任务是由一个行为序列完成的,机器人行为是通过在特定时间内改变其关节角的大小完成。在机器人执行任务的过程中需要不断地保持或更改机器人关节角的速度。机器人的初始状态可以由其各个关节角的角度表示,机器人的工作状态必须由各个关节的角速度构成:

$$\vec{\omega}_{dest} = [\vec{\omega}_1 \quad \vec{\omega}_2 \quad \vec{\omega}_3 \quad \dots]^T \quad (14)$$

根据式(14)可以得出机器人部件在操作空间中的速度:

$$\vec{v}_{dest} = [v_x \quad v_y \quad v_z]^T \quad (15)$$

(2) 计算部件运动距离

机器人部件在 Δt 时间内的运动如图 11 所示,其中 D 是两个部件的初始距离, \vec{n} 为两个部件中心(轴)之间的方向向量,由此可以得出机器人在 Δt 时间内运动的距离:

$$d = (J \vec{\omega}_{dest} \cdot \Delta t) \cdot \vec{n} \quad (16)$$

(3) 判断是否碰撞

首先根据 D-H 模型^[14]中的直角坐标计算机器人运动前部件间的距离 D , 然后根据 D 和部件的运动距离 d 的大小关系来判断是否发生碰撞:

$$d < D - (R_1 + R_2) \quad (17)$$

(4) 计算最优解

如果判断得出会发生碰撞,则需要求出一个与目标状态

最接近且不发生碰撞的最优解。此过程就是在线性约束下求最优解的问题：

$$\min \|\vec{\omega} - \vec{\omega}_{dest}\|^2 \quad (18)$$

约束是：

$$(J\vec{\omega} * \Delta t) \cdot \vec{n} < D - (R_1 + R_2) \quad (19)$$

其中， $\vec{\omega}$ 是在机器人硬件允许的范围内取值。在求出最优解之后，调整机器人目标状态 $\vec{\omega}_{dest}$ ，然后执行该最优解。该最优解是机器人行为检测模块对机器人做出的动作补偿，以确保机器人硬件之间不发生碰撞。

3.3.2 碰撞冲突检测算法

根据上述建模和分析，本文提出碰撞冲突检测的算法。

算法3 Algorithm of Collision Detecting(ACD)

Initialization:

```

1.  $\omega_{dest} = (0, 0, \dots, 0)$ 
2.  $\omega_{compensation} = (0, 0, \dots, 0)$ 
3.  $v_{dest} = (0, 0, 0)$ 
4.  $\theta = (0, 0, 0)$ 
5.  $N = |\omega_{dest}|$ 
6. for i in  $[0 \dots N-1]$ 
7.    $\omega_{dest}[i] = \text{getJointAnglesSpeed}[i]$ 
8. end for
9. for i in  $[0 \dots 2]$ 
10.  for j in  $[0 \dots N-1]$ 
11.     $v_{dest}[i] = v_{dest}[i] + J[j] * \omega_{dest}[i]$ 
12.  end for
13.   $\theta[i] = \text{getDirectionAngle}[i]$ 
14.   $d[i] = v_{dest}[i] * t * \cos\theta[i]$ 
15. end for
16.  $d_{dest} = \text{getDistance}(d[0], d[1], d[2])$ 
17.  $D = \text{DH}()$ 
18. if  $d > D - (R_1 + R_2)$  then
19.    $i = 0$ 
20.    $\Delta\omega_{min} = \text{MAX}$ 
21.    $\Delta\omega = 0$ 
22.   repeat
23.      $\omega = \text{getAngleSpeed}(\text{grad}, i)$ 
24.      $d_{compensation} = \text{getDistance}(\omega_{compensation})$ 
25.     if  $d_{compensation} < D - (R_1 + R_2)$  then
26.        $\Delta\omega = \|\omega - \omega_{dest}\|^2$ 
27.       if  $\Delta\omega < \Delta\omega_{min}$  then
28.          $\Delta\omega_{min} = \Delta\omega$ 
29.          $\omega_{compensation} = \omega$ 
30.       end if
31.     end if
32.      $i++$ 
33.   until  $|\omega| > |\omega_{dest}|$ 
34. end if

```

4 机器人安全保护方法实现

本文采用了 AOP 编程技术实现上述算法。AOP 是通过预编译方式和运行期动态代理实现程序功能的统一维护的一种编程技术^[9]，其普遍应用于程序的日志记录、性能统计、安全控制、事务处理与异常处理等行为中^[8]。将上述行为代码

从业务逻辑代码中分离出来，以在改变这些行为时不影响业务逻辑的代码。

本文采用 AOP 技术实现机器人的安全检测模块的原因有：1)随着人们对机器人需求的不断增高，机器人的任务和控制软件的复杂度都在不断提高，针对已经开发完成的机器人控制软件，直接将机器人安全检测模块的代码加入到机器人控制软件的功能模块中的代价过大，而且可维护性较差；2)通过 AOP 技术，将机器人控制软件的安全检测模块和功能模块分开，实现了松耦合的目的，提高了程序代码的可读性。

如图 12 所示，AOP 技术在智能机器人控制软件的安全检测中的工作原理与三棱镜的散光和聚光原理十分相似。基于 AOP 的机器人控制软件的安全检测主要分为两个阶段：关注点分离和安全检测模块代码织入。

(1) 关注点分离

与三棱镜的散光原理类似，可以将没有安全检测模块的机器人控制软件看作是不同关注点^[9]杂揉在一起的初始系统，AOP 根据功能点的人信息(如方法名)将不同的关注点^[9]分离出来，产生一组连接点^[9]如 walk()方法、moveHands()方法等。机器人执行这些连接点^[9]可能会产生安全可疑行为，如 walk()方法可能会使机器人失去平衡，moveHands()方法可能会使机器人发生碰撞等。

(2) 安全检测模块织入

根据分离出来的关注点和产生的连接点，在 Aspect 中定义切入点^[9]。在整个机器人控制程序预编译时将安全检测模块的代码织入到切入点所在的位置。安全检测模块的代码分为硬件检测、稳定检测、碰撞冲突检测 3 类，分别定义不同的通知(Advice)^[9]，通知的主体是需要织入的安全检测代码。织入之后的机器人控制软件系统具有安全检测能力，不同的功能点会重新融合在一起，构成一个健康的系统。

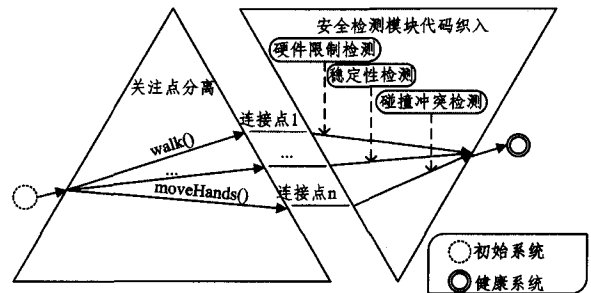


图 12 基于 AOP 的机器人安全检测

5 实验与结果分析

为了测试和验证本文提出的基于行为检测的机器人安全保护方法的有效性，基于 NAO 机器人开展实验，NAO 机器人的基本参数^[16]如表 1 所列。实验方式是使用计算机远程控制机器人。

表 1 NAO 机器人参数

参数类型	参数信息
CPU	ATOM Z530 1.6GHz
内存	1 GB RAM / 2GB flash memory
关节	36 个
传感器	5 类 共 64 个
交互方式	5 种

为了开展实验,本文设计了两个实验场景。

(1)场景一:机器人失去平衡

在此场景中,机器人执行某项任务进入单脚支撑状态,如图 13 所示。在此状态下机器人是稳定的,记为 $State_1$ 。此时机器人需要执行一个任务:机器人左髋关节(LHipYawPitch)处沿着图 13 中指示的方向转动 10(弧度)。在执行完该任务后机器人的状态记为 $State_2$ 。

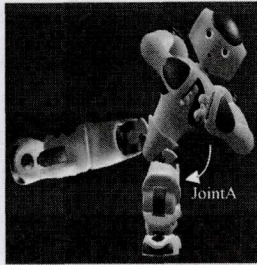


图 13 场景一初始状态

本文采用两种方案执行该任务:1)直接执行功能代码,不做特殊处理;2)将上述的基于 AOP 实现的安全检测算法加入到控制程序中。方案一的效果如图 14 所示,方案二的效果如图 15 所示。

从方案一的效果图中可以看出,机器人执行完任务之后的状态 $State_2$ 是不稳定的,在该状态下机器人失去了平衡,可能会对胳膊、头部等硬件造成损坏。在方案二中(见图 15),机器人在执行该任务之前会对机器人执行完任务之后的状态进行检测,当检测到机器人可能会进入不稳定状态(失去平衡)时,机器人会放弃任务,并进入较为稳定状态(站立)。

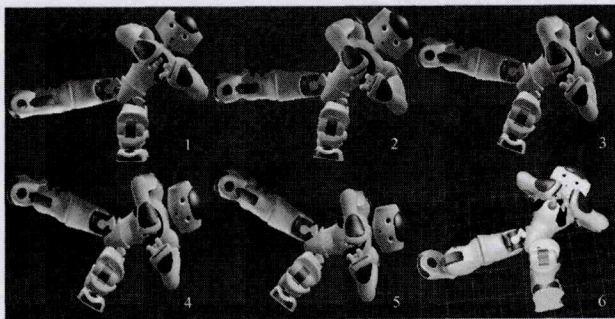


图 14 场景一第一种方案效果图

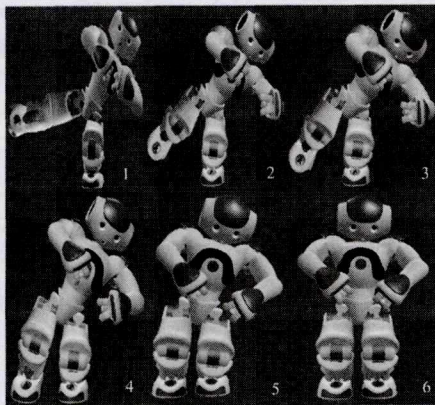


图 15 场景一第二种方案效果图

(2)场景二:机器人肢体冲突

在此场景中机器人处于站立姿势,如图 16(a)所示,记为初始状态 $State_1$ 。此时机器人需要执行一个任务:沿着图 16

(a)中的指示的方向转动 JointB 关节(LShoulderRoll)进入目标状态 $State_2$,如图 16(b)所示。

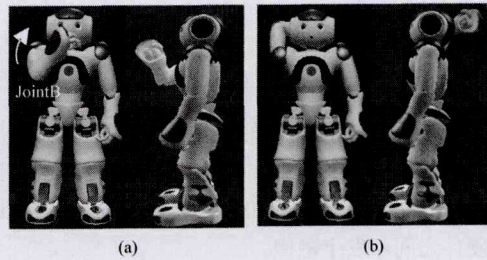


图 16 场景二初始与目标状态

本文采用两种方案执行该任务:1)直接执行功能代码,不做任何特殊处理;2)将 3.3 节中基于 AOP 实现的安全检测算法加入到控制程序中。方案一的效果如图 17(a)所示,方案二的效果如图 17(b)所示。

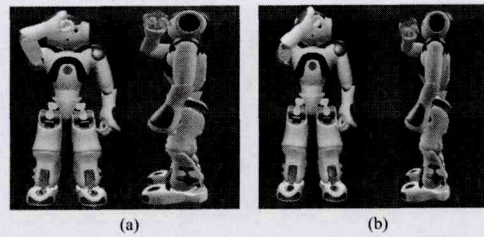


图 17 场景二效果图

此任务的功能代码只是简单地改变关节 JointB 的角度,由 $State_1$ 转变到 $State_2$ 过程中机器人的手臂和头部会发生碰撞。如图 17(a)所示,机器人没有成功达到目标状态 $State_2$,手臂与头部发生碰撞冲突,并持续僵持,这可能会对机器人的手臂或头部造成损坏。在方案二中,在执行此任务之前机器人控制程序会检测是否会发生碰撞,如图 17(b)所示,机器人的手臂和头部没有发生碰撞,而且此时机器人的状态是在不发生碰撞冲突的条件下最大限度地接近目标状态 $State_2$ 。

场景一和场景二分别对本文提出的稳定性检测和碰撞冲突检测算法进行测试,成功地检测出了可能会造成机器人损坏的行为,并及时做出调整,避免了损坏机器人安全行为的发生。由此可以得出基于行为检测的机器人安全保护方法可以有效防止损害机器人行为的发生,实现保护机器人安全的目的。

6 相关工作对比

本文提出的基于行为检测的机器人安全保护方法是针对机器人控制软件中潜在的缺陷或错误可能危害机器人安全的问题,通过在机器人工作时检测机器人的安全可疑行为,实现对机器人的保护。本文从硬件依赖、动作补偿、持续运行与任务分析 4 个方面将本文所提出的基于行为检测方法与现有的机器人安全保护方法如“虚拟限位”^[4]和“运动参数限制”^[5]进行比较分析,结果如表 2 所列。

表 2 机器人安全保护方法对比

机器人安全保护方法	硬件依赖	动作补偿	持续运行	任务分析
虚拟限位	✓	✓	✓	—
运动参数限制	✓	—	✓	—
基于行为检测	—	✓	—	✓

(1)“虚拟限位”^[4]和“运动参数限制”^[5]方法需要专门的机器人硬件保护系统,本文提出的基于行为检测的方法无需额外的硬件,对机器人硬件无依赖性。

(2)“运动参数限制”^[5]方法在检测出运动参数超限时,直接关闭机器人电源,机器人停止任务。“虚拟限位”^[4]和基于行为的检测方法通过给出相应的动作补偿,使机器人脱离危险状态,可以继续执行任务。

(3)“虚拟限位”^[4]和“运动参数限制”^[5]的机器人状态检测和计算模块都需要持续运行,占用机器人计算资源,基于行为检测方法在机器人执行安全可疑行为之前才会运行,不会影响机器人任务的正常执行。

(4)“虚拟限位”^[4]和“运动参数限制”^[5]方法直接操作机器人的关节空间,忽略了机器人任务的特殊性,会导致过度干涉。基于行为检测方法,通过分析任务中的行为,调整机器人关节空间的参数,避免,过度保护而影响机器人任务执行的发生。

通过对比分析可以得出,本文提出的基于行为检测的机器人安全保护方法不仅可以有效地保护机器人安全,而且具有高效性、低干涉性、准确性与低成本等特点。本文尚存在一些不足,如机器人安全可疑行为分析不够全面、当机器人的任务过于复杂时难以进行有效动作补偿等。

结束语 本文提出一种基于行为检测的机器人自身安全保护方法,其主要针对硬件超限、失去平衡和碰撞冲突 3 类可能导致机器人损坏的行为,进行执行前安全检测。基于 AOP 技术实现该安全保护方法。通过实验与结果分析可以得出,对于机器人控制软件存在的安全可疑的行为,该方法可以检测出上述 3 类安全可疑行为,实现保护机器人安全的目的。

下一步工作将对其他类别的机器人安全可疑行为进行建模与检测,并将行为检测方法加入到自主机器人程序开发框架项目中。

参 考 文 献

- [1] Hu Zheng. Review of Robotic Safety Engineering [J]. China Mechanical Engineering, 2004, 15(4): 370-375 (in Chinese)
胡政. 机器人安全性工程研究综述[J]. 中国机械工程, 2004, 15(4): 370-375
- [2] Andrew G. Brooks. Coordinating Human-Robot Communication [D]. Massachusetts: Massachusetts Institute of Technology, 2007
- [3] Zhang Shi-xiang. Analyses and countermeasures on accidents of industrial robots [J]. Industrial Safety & Dust Control, 2002, 28(3): 26-29 (in Chinese)
张世翔. 关于工业机器人的事故分析及其对策[J]. 工业安全与保护, 2002, 28(3): 26-29
- [4] Feng Yuan-jing. Development of the safeguard system of parallel manipulator [J]. Modular Machine Tool & Automatic Manufacturing Technique, 2004(8): 21-24 (in Chinese)
- [5] Liu Sheng-peng, Ye Hua. Design and Implement of Safety Protection System for Industry Welding Robot [C]//Proceedings of 2009 Chinese Intelligent Automation Conference. 2009: 362-365 (in Chinese)
刘升鹏, 叶焯. 工业焊接机器人安全保护系统的设计与实现[C]//2009中国智能自动化会议论文集. 2009: 362-365
- [6] Rahimi M. Framework for Software Safety Verification of Industrial Robot Operations [J]. Computers & Industrial Engineering, 1991, 20(2): 279-287
- [7] Dhillon B S. Availability Analysis of a Robot with Safety System [J]. Microelectronics & Reliability, 1996, 36(2): 169-177
- [8] Sirbi K. Stronger Enforcement of Security Using AOP & Spring AOP [J]. Journal of Computing, 2010, 2(6): 99-105
- [9] Kiczales G. Aspect-Oriented Programming [M]//11th European Conference on Object-Oriented Programming (ECOOP). Springer-Verlag LNCS, 1997: 220-242
- [10] Wieber P B. On the Stability of Walking Systems [C]//The 3rd IARP International Workshop on Humanoid and Human Friendly Robotics. 2002
- [11] Goswami A. Foot Rotation Indicator (FRI) Point: A New Gait Planning Tool to Evaluate Postural Stability of Biped Robots [C]//IEEE International Conference on Robotics and Automation. 1999: 47-52
- [12] Goswami A. Postural Stability of Biped Robots and Foot Rotation Indicator (FRI) Point [J]. International Journal of Robotics Research, 1999, 18(6): 523-533
- [13] Denavit J, Hartenberg R S. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices [J]. Journal of Applied Mechanics, 1955, 22: 215-221
- [14] Shah S V, Saha S K. Denavit-Hartenberg Parameterization of Euler Angles [J]. Journal of Computational and Nonlinear Dynamics, 2012, 7(2): 1-10
- [15] Pu Song-hao, Zhong Qiu-bo. Intelligent Robot [M]. Harbin: Institute of Technology Press, 2012: 71-81 (in Chinese)
朴松昊, 钟秋波. 智能机器人 [M]. 哈尔滨: 哈尔滨大学出版社, 2012: 71-81
- [16] Aldebaran Robotics. Nao Software Documentation [OL]. [2014-07-02]. <https://community.aldebaran.com/doc>
- [17] Mueggler E. Aerial-guided navigation of a ground robot among movable obstacles [C]//2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). 2014: 1-9
- [18] Aloulou A, Boubaker O. A Minimum Jerk-Impedance Controller for planning Stable and Safe Walking Patterns of Biped Robots [J]. Mechanisms & Machine Science, 2015: 385-415
- [19] Navarro S E, Hein B. Capacitive Tactile Proximity Sensing: From Signal Processing to Applications in Manipulation and Safe Human-Robot Interaction [M]//Soft Robotics, 2015: 54-65