

# 一种支持自适应程序设计的移动机器人中间件

陈昊 孙辉 许畅 马晓星

(南京大学软件新技术国家重点实验室 南京 210046) (南京大学计算机科学与技术系 南京 210046)

**摘要** 机器人控制中间件是软件中间件设计研究领域的一个热点。为了开发一种支持自适应程序设计的移动机器人中间件平台,首先提出这种中间件平台需要满足的设计要求:跨底层硬件平台、保证服务质量以及具备一定的自适应能力。从 Java 虚拟机的移植、服务质量表现形式的选择、自适应机制及 SCC(Sensor/Compute/Control)设计结构的应用 4 个方面来阐述此中间件平台的设计思路和开发过程。最后,通过真实实验的数据对比,说明此中间件可为移动机器人程序的有效开发提供服务质量保证。

**关键词** 移动机器人中间件,自适应能力,服务质量保证,SCC 结构

**中图法分类号** TP301 **文献标识码** A

## Mobile Robot Middleware Supporting Self-adaptive Programming

CHEN Hao SUN Hui XU Chang MA Xiao-xing

(State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210046, China)

(Department of Computer Science and Technology, Nanjing University, Nanjing 210046, China)

**Abstract** Robot middleware systems facilitate the development of self-adaptive robot applications by providing a set of high-level sensor/actuator APIs that abstract and hide the heterogeneity of different hardware platforms. We presented a middleware infrastructure supporting self-adaptive programming on mobile robot systems. It aims to cope with cross-platform issues and provide guarantee for the service quality in terms of a set of abstracted and quality-guaranteed APIs. We used this middleware infrastructure to support the development of self-adaptive applications on mobile robot systems, so that it could provide a consistent level of QoS guarantee despite of the varying physical differences among different robot car systems. Our experimental evaluation reports promising results that the middleware can effectively support quality-guaranteed self-adaptive programming.

**Keywords** Mobile robot middleware, Self-adaptive capacity, Guarantee of service quality, SCC architecture

## 1 引言

计算机技术、无线通讯、机电技术以及传感器技术的发展使得机器人技术在工业制造、社会服务以及军事等领域中得到了广泛的应用。随之而来的是机器人系统设计的复杂化和功能的多样化,从而使基于机器人系统的软件开发的时间成本、人力成本也随之增加。因此,为了提高系统的重用性,降低软件的开发周期,对机器人中间件的研究成为了近几年的一个热点。

目前,机器人中间件研究的重点在于如何通过中间件来简化机器人程序的开发过程、支持机器人间的通讯和互操作、提供对可用资源的有效利用、提供机器人异构性的抽象、支持与其他系统(无线传感器网络、服务器)的集成、提供常用的机器人应用服务、提供自动的资源发现和服务以及支持嵌入式的设备<sup>[1]</sup>。比如德国乌尔姆大学的 Hans Utzh 等人实现的

Miro(Middleware for Mobile Robot)<sup>[2]</sup>就是一种面向对象的机器人中间件,用以改进移动机器人的软件开发过程并提供移动机器人与企业信息系统之间的交互。技术上,Miro 通过采用 CORBA 标准来实现面向对象的设计,这使得分布式机器人系统的进程间和跨平台的互操作成为可能。同时,Miro 还给用户提供了诸如自定位、绘图以及路径规划等经常需要使用的服务。UPnP 中间件<sup>[3]</sup>是由韩国科学技术研究院开发的一种采用 UPnP(the Universal Plug and Play)结构,用于机器人内外软件交互和普适机器人控制的机器人中间件平台。UPnP 提供了 PC、无线普适设备和智能应用之间的点对点网络连接,并且拥有自动发现和配置机制,从而能够灵活地配置机器人的组件并与周围网络中其他设备和资源进行交互。AWARE<sup>[4]</sup>是由西班牙塞维利亚大学和德国斯图加特大学开发的一种以数据为中心的机器人中间件。AWARE 中间件的目标是为移动机器人和无线传感器网络之间的信息交互提供

到稿日期:2012-01-25 返修日期:2012-03-14 本文受国家自然科学基金(60973044,61100038,61021062),国家高技术研究发展计划(863)(2011AA010103),教育部新世纪优秀人才支持计划(NCET-10-0486)资助。

陈昊(1985-),男,硕士生,主要研究方向为自适应软件系统,E-mail:tonychannju@gmail.com;孙辉(1985-),男,硕士生,主要研究方向为自适应软件系统;许畅(1977-),男,副教授,CCF 会员,主要研究方向为软件工程、软件测试与分析、普适计算上下文管理、物联网、射频识别等;马晓星(1975-),男,教授,博士生导师,CCF 会员,主要研究方向为自适应软件系统、服务计算、软件体系结构与中间件等。

简单的机制。这些信息可以是传感器探测的环境信息,也可以是移动机器人的地理信息。对于任何一种机器人中间件而言,实现上述所有功能并不现实。事实上,从上文列举的这些已取得进展的机器人中间件研究项目来看,它们一般是根据研究目标、实验平台以及应用场景的不同来着力实现其中一个或者几个目标。

本文所介绍的移动机器人中间件主要解决的问题是对于多个移动机器人平台,如何设计一个移动机器人中间件来支持用户进行自适应的程序设计。本文将首先给出这样一个中间件所应满足的要求,然后阐述如何通过一系列技术手段和机制来满足这些设计要求,最后将给出实验结果,并进行分析。

## 2 移动机器人中间件的设计要求

本文认为,一个支持用户进行自适应程序设计的移动机器人中间件需要满足以下这些要求:

跨平台的能力。本文研究的实验移动机器人平台有 3 种(见图 1):嵌入式小车(底盘为履带)、嵌入式小车(底盘为三轮,一个带舵机的主动轮,两个从动轮)以及乐高 NXT 小车。同时,嵌入式小车与乐高 NXT 小车所使用的操作系统和开发语言也有所不同。前者使用的是嵌入式 Linux 系统 ucLinux,开发语言为标准 C 和 C++;而后的操作系统和开发语言并不固定,目前比较流行的是 Lawrie Griffiths 等人 Lego Mindstorms 编写的第三方 Java 虚拟机 leJOS<sup>[5]</sup>作为乐高 NXT 的固件,编程语言为 Java。因此,本文需要针对这样的情况解决语言级的问题并尽量屏蔽移动机器人平台之间的差异。

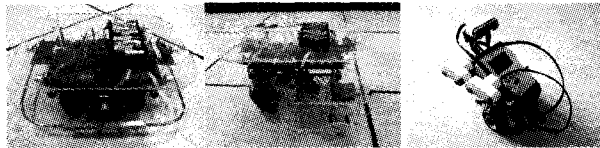


图 1 3 种移动机器人小车的实体图片

服务质量的保证。为了支持用户进行自适应的程序设计,中间件必须为用户提供机器人应用的服务质量保证,否则用户程序的自适应性没有实际的基础。以本文实验所用 3 种移动机器人平台为例,如 3.2 节图 3 所示的通过实验数据模拟的前进轨迹,我们可以发现,在没有一定的质量保证的机制下,移动机器人的运动轨迹会因平台的不同而产生不同的变化,其造成的误差使用户无法编写自适应的程序。因此,移动机器人中间件必须以某种具体形式来体现服务质量,并设计一些机制来为用户保证这样的服务质量。

具有一定的自适应能力。图 2 为一个自适应系统工作的示意图。其工作原理是这样:自适应系统通过 Devices 的 Sensors(Sources)部分从外界环境获取原始信息,然后将其交给 Contexts 部分处理成能够为自适应系统做出正确决策的信息;Controllers 部分则根据这些信息进行对 Devices 的动作执行部分进行控制,从而作用于外界环境,使系统能够感知的外界信息发生改变。我们的中间件平台就处于图 2 的虚线框图部分,即为上层的信息处理(理解)和决策之后动作的执行提供支持。亦即,中间件的作用在于填补用户理解的信息和因之做出的决策与现实的感知和执行部分之间的差

距。因此,中间件必须理解上层的信息和命令,同时做好现实的物理控制,这其实也是一个自适应的过程。所以,中间件必须具有自适应性,并且为上层提供有质量保证的 API。

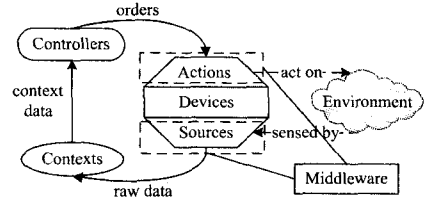


图 2 自适应系统的工作示意图

面向应用的自适应程序设计框架。完整的自适应中间件系统还应为上层应用的自适应逻辑设计提供高抽象级的程序设计框架。我们已经开发了一种基于状态机和规则的自适应程序设计框架,但本文不具体讨论这个框架,有兴趣的读者可参见文献[7]。

## 3 移动机器人中间件的设计思路

### 3.1 Java 虚拟机的移植和平台异构的屏蔽

上文提到,由于移动机器人软硬件系统的多样性,移动机器人中间件必须具备一定的跨平台能力。在我们的研究项目中,嵌入式小车和乐高小车分别使用标准 C/C++ 和 Java 进行开发,而在乐高小车上使用标准 C/C++ 开发目前没有一个较完善的解决方案。因此,必须解决语言级的问题,并且提供屏蔽移动机器人差异的方法。

关于语言级的问题,本文选择的解决办法是将 leJOS 的 Java 虚拟机移植到 ucLinux 系统上。之所以选择这个办法,原因有三。第一,嵌入式小车使用的处理器与乐高 NXT 小车均为 ARM 系列,并且 leJOS 也有 Unix 系统的模拟器版本,这给移植工作带来了一定的便利。第二,Java 语言的特性决定了它更适合作为一个跨平台工作的程序语言。第三,leJOS 除了 Java 虚拟机还包括一系列支持机器人工作的算法和接口,以便功能的继续拓展。

Java 虚拟机的移植工作分两步。首先,Java 虚拟机的 Unix 版本属于废弃版本,需要对照 leJOS 的 0.9.1 版本(Feb 06, 2012)对其进行修改(去掉 0.9 版本没有的 emu\_dump 等程序文件、更改 Unix 版本 JVM 的程序启动顺序、更新关于类和线程的数据结构和操作、修改 Bug),使其在 ucLinux 上能够运转起来。然后,重新设计 Java 虚拟机与底层设备(马达和超声波传感器)的通讯程序,而在 leJOS 的 Java 程序类库中,编写提供用户对嵌入式小车进行底层控制的接口。需要说明的是,这部分的工作只是最低限度地保证了 leJOS 能够在嵌入式小车的 ucLinux 上运行。诸如 System.out.println 方法的重定位和文件管理等支持性的程序的修改,由于对本研究项目的影 响不大,因此暂时没有完成,留待后续工作进行改进,以使 leJOS 更具实用性。

在这些工作的基础上,本文还为 3 种小车提供了进行简易的统一控制的导航程序。用户只要给程序提供移动机器人的底盘参数(如车轮数、车轮直径和车轮距),程序就可以基于一个统一的动力学原理对移动机器人进行控制。这也是本文以后工作的基础,同时也为用户提供了一个更一般的控制方式。

至此,本文在一定程度上使改进的 leJOS 具有跨平台工

作的能力。

### 3.2 服务质量的表现形式——“基本动作”误差

在讨论服务质量的表现形式之前,先阐释为什么对于我们的研究目的来说要关心服务质量这个问题。

首先,需要定义“基本动作”这个概念。某一特定的移动机器人任务  $T$  可以看作由起始点  $S$ 、终止点  $T$  以及中间过程  $P$  构成。其中,起始点  $S$  和终止点  $T$  是由一个或者一组量来表示,比如起始位置和终止位置的地理坐标,这些量来自机器人平台的传感器数据。而中间过程  $P$  的粒度根据具体任务的不同而不同。此处,我们规定  $P$  的最小粒度称为“基本动作”,其包括前进(包括定距离前进和不定距离前进)、后退(包括定距离后退和不定距离后退)、左转某一角度、右转某一角度以及停止。因此,任务  $T$  可以看作是一系列更小粒度的任务  $T_1, T_2, \dots, T_i, \dots, T_{n-1}, T_n$  的有序集合,更进一步地,任务  $T$  也可以看作是一系列“基本动作”的有序集合。

为了更细致地说明问题,下面对“基本动作”的误差问题进行详细阐述。

对于“前进”和“后退”这两个基本动作,可以将其动作效果的误差分解为 3 个维度:角度误差、路线偏差和路线误差(见图 3)。其中  $A$  为移动机器人的初始位置,  $B$  为理想情况下基本动作停止时移动机器人所应到达的位置,而  $C$  为实际情况下移动机器人到达的位置。为了方便实验测量,将  $\angle \alpha$  记为角度误差,  $A$  与  $C$  的边沿中点之间垂直于行驶方向的距离  $b$  记为路线偏差,而  $B$  与  $C$  的前沿中点之间平行于行驶方向的距离  $e$  记为路线误差。

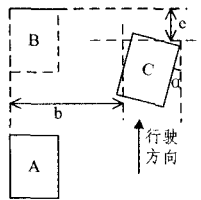


图 3 前进和后退的误差测量示意图

对于“左转”和“右转”这两个基本动作,为了方便实验测量,首先将其简化处理为原地转弯。在实验中,也对移动机器人的实际控制做了这样的处理。因此,转弯的效果误差就转化为角度误差。

本文对 3 种移动机器人平台分别做了“基本动作”的实验,并通过上述方式对“基本动作”的误差进行了测量。图 4、图 5 为通过实验数据模拟的 3 种移动机器人平台做“前进”和“后退”动作的运动轨迹。其中,  $(0,0)$  点为移动机器人的起始位置,“行驶方向”轴上的每一个有数字的刻度为无误差情况下移动机器人所应到达的位置。表 1 为移动机器人平台在“前进”和“后退”动作的所对应角度误差(单位:°)。表 2 为移动机器人做“转弯”动作的角度误差(单位:°)。

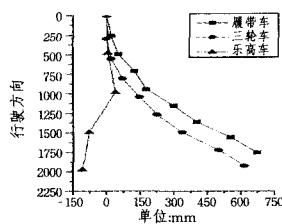
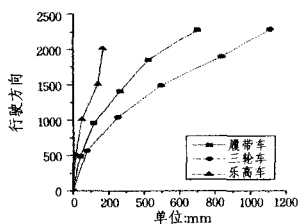


图 4 3 种平台的前进轨迹模拟图 图 5 3 种平台的后退轨迹模拟图

表 1 移动机器人的前进和后退的角度误差(单位:°)

	前进			后退		
	履带	三轮	乐高	履带	三轮	乐高
500mm	7	11	2	10	11	0
1000mm	13	21	2.5	22	21.5	-1
1500mm	20	32	6	31	31	3.5
2000mm	30	40	6	40	39	2

表 2 移动机器人转弯的角度误差(单位:°)

	左转			右转		
	履带	三轮	乐高	履带	三轮	乐高
45	2	-1	-8	-3	4	-13
90	10	7	-13	7	5	-5
135	-5	8	-5	-5	3	-17
180	-10	1	3	-3	-1	-30

通过这些实验数据可以发现,移动机器人的“基本动作”的执行效果随平台的底盘和控制方式的不同而产生不同程度的误差,并且这些误差有的有规律可循,有的却找不到明显的规律。也就是说,“基本动作”在不同的移动机器人平台上的执行效果对于用户来说是不可预测的,并且在运行了较长距离时,误差已经非常大,服务质量完全不能支持自适应程序设计的要求。因此,有必要设计出一个机制来保证移动机器人中间件的服务质量,同时我们也认为中间件服务质量的表现形式可以选择“基本动作”的误差。这样处理的优点在于误差对于用户来说比较直观,并且易于与后文的工作进行对比。

### 3.3 移动机器人中间件的自适应机制

在没有任何调整机制的情况下,移动机器人的动作并不能满足自适应程序所要求的服务质量,因此,有必要设计这样一种机制来保证中间件的服务质量。又由于将“基本动作”的误差选为服务质量的表现形式,因此本节还是从“基本动作”开始阐述移动机器人中间件的自适应调整机制。

对于“前进”、“后退”和“转弯”而言,前两者与后者存在着执行时间和动作误差上的差别。从执行时间上看,“前进”和“后退”的执行时间较“转弯”更长,可以看作是过程性的动作,这就意味着在其动作执行过程中,中间件会有充足的时间对这些动作进行调整。而“转弯”可以看作是瞬时完成的动作,执行过程中就不适合做适当的调整。从动作误差上看,“前进”和“后退”在执行过程中误差一直有增大的趋势,这就方便我们采取一些策略来约束这种变化,从而达到调整动作、缩小误差的目的。而“转弯”的误差没有这样的特点,并不适合采取与前者一样的调整策略。因此,在设计调整机制时,需要分两种情况进行考虑。

对于“前进”和“后退”,我们的调整机制类似于帆船逆风航行的原理(见图 6)。帆船在逆风航行时,水手会将风帆与海风形成一个角度,使帆船在一个与正确航向成一角度的方向向前航行。当帆船航行了一定距离时,再调整风帆,使帆船成一相反角度方向向前航行,从而以“之”字形路线的方式向预定航向航行。从图中可以发现,这种方式也保证了帆船航行的误差在两条虚线以内,不致使其偏离航行路线。

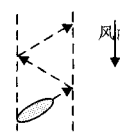


图 6 帆船逆风航行原理图

我们将“前进”和“后退”的调整方式称为“分段自适应调整”。其具体思路为：首先为这种调整机制设置一个分段调整距离，距离的确定与需要保证的动作误差有关。此处，将分段调整距离设置为 500mm。之所以设置这个距离，是因为在上文实验中，小车平台行驶 500mm 的误差已经无法容忍。以前进 1500mm 为例，将此动作分为 3 段来进行，分段数目由分段调整距离和调整过程共同决定。每一段均由两部分工作来完成，先向前行驶 500mm，然后根据移动机器人侧面的超声波传感器探测的数据对机器人进行相应的调整以减小动作误差。如此反复，直至完成此动作。

这里需要说明的问题有两个。首先，对移动机器人的调整需要从两个维度——垂直于路线方向和路线方向进行。一般来说，对垂直于路线方向的误差调整的优先级要高于路线方向，因为只有将垂直于路线方向的误差（至少是角度误差）调整好，才有可能对路线方向的误差进行调整。同时，由于误差调整必须借助于移动机器人的外部传感器，即误差调整必须借助于外界环境的参照物，因此在“基本动作”执行之前，用户需要根据实际的物理环境来设置参照物，从而保证中间件的服务质量。由于调整的维度有两个，并且有优先级，因此用户至少需要设置垂直于路线方向的参照物来保证一定程度的服务质量。对于底盘造成的误差大于传感器误差的情况，设置路线方向的参照物（称为“过程参照物”），就可以对路线方向的误差起到更精确的控制。总起来说，中间件的正常工作需要用户设置参照物，并且参照物设置的维度决定了中间件能够提供的服务质量级别。

其次，上文说到“基本动作”的调整需要设置参照物才能起作用，而参照物的信息必须通过机器人的外部传感器的探测才能获取。所以，外部传感器的类型和性能决定了“基本动作”的调整和中间件的工作条件有一定的环境限制，这一点必须向用户明确。基于本文实验机器人平台的具体情况，移动机器人中间件的物理环境限制如下：对于“前进”和“后退”，垂直于路线方向的参照物必须是与路线方向平行的传感器可探测到的平面，机器人和参照物的距离应在 400mm 到 1200mm 之间；而路线方向的参照物必须是与路线方向垂直的传感器可探测到的平面，机器人和参照物的距离应保持在 4000mm 之内。对于“转弯”，参照上文对垂直路线方向的参照物的限制即可。

上述自适应的调整机制本质而言是一种与物理世界进行交互的 SCC (Sense/Compute/Control) 模型。Damien Cassou 等人在文献[6]中指出，SCC 程序可以组织成一个 4 层结构：(1)位于底层的 Sensors，获取周围环境的信息；(2)然后是 Context Operators，用以处理这些信息；(3)接着是 Control Operators，使用上述处理好的信息进行控制；(4)最高层是 Actuators，用来影响周围环境。如此往复，就构成了一个自适应的控制机制。本文借助上述分层的思想对移动机器人中间件的调整机制进行了分层，以方便中间件程序的实现(见图 7)。调整程序中的 Contextor 组件从超声波传感器获得外部环境信息，并对这些信息进行针对性的容错处理。Judgmentor 组件从 Contextor 组件获得所需要的信息，并且结合用户提供的信息来完成对“是否需要分段处理”和“是否需要

进行调整”的判断。Car Controller 组件则从 Judgmentor 组件外获得这些判断信息，以便对 Executor 组件和 Adjustor 组件发送命令。而 Executor 组件和 Adjustor 组件则结合 Contextor 组件提供的消息来完成对底盘的控制。

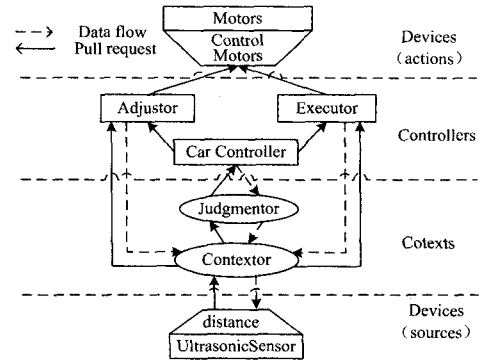


图 7 调整机制的分层结构以及数据流示意图

根据文献[6]提出的交互规约语言，将上述结构形式化为表 3 所列的交互规约，从而实现移动机器人中间件的基本动作。

表 3 自适应调整程序 SCC 结构的交互规约

Level	Modules	Associated interaction contract
Devices(actions)	ControlMotors	$\langle \uparrow(\text{Executor}, \text{Adjustor}), \emptyset, \emptyset \rangle$
Controllers	Executor	$\langle \uparrow(\text{CarController}), \downarrow(\text{Contextor}), \uparrow\text{self} \rangle$
	Adjustor	$\langle \uparrow(\text{CarController}), \downarrow(\text{Contextor}), \uparrow\text{self} \rangle$
	CarController	$\langle \emptyset, \downarrow(\text{Judgmentor}), \uparrow\text{self} \rangle$
Contexts	Judgmentor	$\langle \downarrow\text{self}, \downarrow(\text{Contextor}), ? \emptyset \rangle$
	Contextor	$\langle \downarrow\text{self}, \downarrow(\text{UltrasonicSensors}), \emptyset \rangle$
Devices(sources)	UltrasonicSensors	$\langle \downarrow\text{self}, \emptyset, \emptyset \rangle$

## 4 实验和评估

通过移动机器人中间件编写了“基本动作”的示例程序，并对“基本动作”进行了一定量的实验测试，从而得到了中间件可以提供给用户的“基本工作”的误差保证(见表 4—表 7)。

表 4 3 种小车平台的转弯误差统计数据

	左转 90 度		右转 90 度	
	误差均值	误差范围	误差均值	误差范围
履带车	1.98	-5~6	2.58	-5~5
三轮车	1.94	0~5	1.05	0~3
乐高车	2.35	-5~5	2.75	-4~6

表 5 3 种小车平台的前进误差统计数据(除路线误差)

	左转 90 度		右转 90 度	
	误差均值	误差范围	误差均值	误差范围
履带车	2.25	-7~7	16.14	0~48
三轮车	1.87	-4~3.5	35.23	3~100
乐高车	1.95	0~6	27.32	5~66

表 6 3 种小车平台的后退误差统计数据(除路线误差)

	角度误差(°)		路线偏差(mm)	
	误差均值	误差范围	误差均值	误差范围
履带车	2.61	-5~5	20.05	-13~68
三轮车	3.27	-10~2	39.32	-72~96
乐高车	2.78	-4~6	30.46	-78~40

表7 3种小车平台前进和后退的路线误差统计数据

	前进(无过程参照物)		前进(有过程参照物)	
	误差均值	误差范围	误差均值	误差范围
履带车	51.06	-89~15	34.8	-2~84
三轮车	*	*	117.63	76~142
乐高车	22.4	19~28	14	7~30
	后退(无过程参照物)		后退(有过程参照物)	
	误差均值	误差范围	误差均值	误差范围
履带车	17.33	-55~34	30.86	23~42
三轮车	*	*	134.93	41~194
乐高车	20	-27~-11	13.83	-20~5

注: \* 需要说明的是,三轮车在无过程参照物的情况下,其路线误差随前进(后退)的距离递增。前进动作,1000mm的误差均值为173.8mm,误差范围为167~185(mm);2000mm的误差均值为304.6mm,误差范围为275~327(mm);3000mm的误差均值为406.2mm,误差范围为381~426(mm)。后退动作,1000mm的误差均值为122.8mm,误差范围为110~133(mm);2000mm的误差均值为185.8mm,误差范围为147~208(mm);3000mm的误差均值为417mm,误差范围为367~475mm。

通过与3.2节实验数据的对比可以发现,在移动机器人中间件规定的工作环境下,移动机器人中间件可以保证“基本动作”的角度误差控制在完全可以接受的范围内,也就是说不管做哪个动作,中间件都可以保证移动机器人不会出现偏离行驶方向越来越远的情况。而对于路线偏差维度,三轮车的误差较另外两个移动机器人平台的误差均值大,且误差极值也大,说明三轮车在调整过程中左右(相对于车头方向)振荡比较显著,这与三轮车的底盘有关,但是对比前面的数据,中间件依旧有效地降低了误差,保证了服务的质量。在路线误差方面,履带车和乐高车在有无过程参照物的情况下服务质量的提高并不明显,而三轮车的服务质量提高则要明显得多。这说明,路线误差的改进可能是由底盘的性能和传感器的性能共同决定的。

在所研究的基于规则和状态机的自适应程序设计框架中,用户可借助具有质量保证的中间件进行以“基本动作”为基础的只关心上层行为逻辑的自适应程序的开发。在具体的应用中,我们为用户提供信息类形式的接口与移动机器人中间件进行交互。用户可以通过向中间件发送 RequestMessage 类来获取传感器信息、请求中间件完成所需的“基本动作”,而中间件则返回 ResponseMessage 类获取所需数据和得到此时机器人的底盘工作状态信息,以便决定以后的请求信息。文献[7]的实验表明,通过这样一种体系结构进行自适应程序的开发,上层诸如沿墙避障程序只需要不到200行的代码,而自适应部分的控制代码只有不到10行,并且策略修改灵活,不需要过多地关心底层的复杂实现,维护起来也较为轻松。

## 5 相关工作

Michael Clarke 等人在文献[8]中提出以基于组件模型的方式构造一种自适应性的中间件;Karen Henricksen 等人在文献[9]中提出了一种对传感器逐步分层抽象的、可自适应处理上下文的软件工程框架;而 Stefanos Zachariadis 等人<sup>[10]</sup>则针对移动设备提出了一种具有自适应性的元模型,同时也采用了组件的思想。

Simon Dobson 等人<sup>[11]</sup>在对一系列自适应系统进行研究之后指出,自适应系统存在着以 Collect、Analyze、Decide 和 Act 为主的循环反馈过程。而 Damien Cassou 等人也在文献[6]中提到与之类似的 Sensor/Compute/Control(SCC)自适应程序设计模型,并以此模型为基础提出了对自适应程序结构各层次中和层次间组件之间的数据和控制流的形式化交互规约,同时也开发出名为“DiaSpec”的插件来根据交互规约生成相应的程序框架,从而方便自适应程序的开发和维护。George Edwards 等人<sup>[12]</sup>在机器人系统中应用上述 SCC 和元组件的思想提出了基于自主移动机器人平台的结构驱动的自适应程序设计方法,并在机器人实验场景中对其中故障恢复机制进行了验证。

本文在自适应机制的设计上也采用了类似的 SCC 控制模型,用以提供有服务质量保证的移动机器人中间件。同时,与上述其他领域的自适应中间件或者软件框架类似的是,本文也同样应用了分层的思想,为用户提供具有平台独立性的易于使用的程序接口。

**结束语** 本文的研究目的在于为用户提供一个支持自适应程序设计的移动机器人中间件平台。为了实现此研究目的,本文认为这样的中间件需要满足跨平台、服务质量的保证和一定的自适应能力这3个要求。接着从 Java 虚拟机的移植、“基本动作”误差和自适应机制3个方面阐述了与之对应的解决方案。最后,通过“基本动作”的实验数据对比,说明本文所提出的中间件平台可以为用户提供一定程度的服务质量保证,尤其是在角度误差和路线偏差方面。本文未来的工作将着力于提高调整机制的自适应性,并对此中间件在基于规则的程序设计框架上的应用进行进一步的研究。同时,将对 leJOS 的 JVM 做进一步的改进,使其更具实用价值。

## 参考文献

- [1] Mohamed N, Al-Jaroodi J, Jawhar I. Middleware for Robotics; A Survey [A] // Robotics, Automation, and Mechatronics, 2008 Conference on [C]. Chengdu: IEEE Conference Publications, 2008;736-742
- [2] Utz H, Sablatng S, Enderle S, et al. Miro - Middleware for Mobile Robot Applications [J]. IEEE Transactions on Robotics and Automation, 2002, 18(4): 493-497
- [3] Ahn S, Lim K, Lee J, et al. UPnP Robot Middleware for Ubiquitous Robot Control [A] // The 3rd International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2006) [C]. Korea, Oct. 2006
- [4] Gil P, Maza I, Ollero A, et al. Data Centric Middleware for the Integration of Wireless Sensor Networks and Mobile Robots [A] // Proc. 7th Conference on Mobile Robots and Competitions, ROBOTICA 2007 [C]. Algarve, Portugal, April 2007
- [5] Griffiths L, Shaw A, Glassey R, et al. LeJOS, Java for Lego Mindstorms [EB/OL]. <http://lejos.sourceforge.net/>, 2012-02-06
- [6] Cassou D, Bolland E, Consel C, et al. Leveraging Software Architecture to Guide and Verify the Development of Sense/Compute/Control Applications [A] // Software Engineering (ICSE),

2011 33rd International Conference on [C]. Honolulu HI; IEEE Conference Publication, 2011; 431-440

- [7] Ge Hai-long, Ma Xiao-xing, Xu Chang. Constructing Self-adaptive Software Systems; A Case Study with An Obstacle-Avoiding Mini-Car[J]. Journal of Frontiers of Computer Science and Technology, 2012(6)
- [8] Clarke M, Blair G S, Coulson G, et al. An efficient component model for the construction of adaptive middle-ware [A] // Guerraoui R. Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg (Middle-ware '01)[C]. London, UK; Springer-Verlag, 2001; 160-178
- [9] Henriksen K, Indulska J. A software engineering framework for context-aware pervasive computing[A]//Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communica-

tions (PerCom '04)[C]. Washington, DC, USA; IEEE Computer Society, 2004; 77-86

- [10] Zachariadis S, Mascolo C, Emmerich W. The SATIN component system - a metamodel for engineering adaptable mobile systems [J]. IEEE Transactions on Software Engineering, 2006, 32(11):910-927
- [11] Dobson S, Denazis S, Fernandez A, et al. A survey of autonomic communications[J]. ACM Transactions Autonomous Adaptive Systems (TAAS), 2006, 1(2):223-259
- [12] Edwards G, Garcia J, Tajalli H. Architecture-driven self-adaptation and self-management in robotics systems[A]// Software Engineering for Adaptive and Self-Managing Systems, 2009. SEAMS '09. ICSE Workshop on[C]. Vancouver, Canada; IEEE Conference Publication, 2009; 142-151

(上接第 93 页)

增长幅度都超过了数据包传送率的增长幅度。这是由于网络吞吐量考虑的是单位时间内汇聚节点接收到的数据总字节数,因此该指标与数据包传送率成正比而与平均端到端延时成反比。

**结束语** 三维水声传感器网络用来探测和观察二维水声传感器网络不能充分观察到的现象。为了提高三维水声传感器网络的路由效率,本文提出了基本地理位置路由协议(BGR)和自适应路由协议(SAR)。在基本地理位置路由协议中,每个传感器节点将三维的水下空间划分为 8 个区域。每个节点根据与各自邻居节点之间连线的夹角以及目标节点的地理位置信息来决定转发数据包的候选节点。然后,在基本地理位置路由协议基础上,设计了一种自适应路由协议,其可以较好地适应三维水声传感器网络中部分节点失效的情况。通过仿真实验发现,自适应路由协议可以在数据包传送率、端到端延时和网络吞吐量指标上达到较好的平衡,而基本地理位置路由协议在端到端延时指标上性能最好。

## 参 考 文 献

- [1] Manjula R, Sunilkumar S. Issues in Underwater Acoustic Sensor Networks[J]. International Journal of Computer and Electrical Engineering, 2011, 3(1):101-110
- [2] Akyildiz I, Pompili T. Underwater Acoustic Sensor Networks; Research Challenges[J]. Ad Hoc Networks, 2005, 3(3):257-279
- [3] Partan J, Kurose J, Levine B. A Survey of Practical Issues in Underwater Networks [C]// Proceedings of the 1st ACM International Workshop on Underwater Networks. Los Angeles, USA; IEEE Computer Society, 2006; 17-24
- [4] Jornet J, Stojanovic M, Zorzi M. On Joint Frequency and Power Allocation in a Cross-Layer Protocol for Underwater Acoustic Networks[J]. IEEE Journal of Oceanic engineering, 2010, 35(4):936-947
- [5] Ravelomna V. Extremal Properties of Three dimensional Sensor Networks with Applications[J]. IEEE Transactions in Mobile Computing, 2004, 3(3):246-257
- [6] 李宇, 黄海宁. 基于水声传感器网络的多路径协同多跳通信研究

[J]. 应用声学, 2010, 29(5):324-329

- [7] Teymorian A, Cheng W, Ma L, et al. 3D Underwater Sensor Network Localization[J]. IEEE Transactions on Mobile Computing, 2009, 8(12):1610-1621
- [8] Jiang Z. Underwater Acoustic Networks Issues and Solutions [J]. International Journal of Intelligent Control and Systems, 2008, 13(3):152-161
- [9] Vahdat A, Becker D. Epidemic Routing for Partially Connected Ad Hoc Networks [R]. TR CS-200006. 2000
- [10] Intanagonwiwat C, Govindan R, Estrin D. Directed Diffusion; A Scalable and Robust Communication Paradigm for Sensor Networks [C]// Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. Boston, Massachusetts, USA; ACM, 2000; 56-67
- [11] Xie P, Cui J, Lao L. VBF: Vector-based Forwarding Protocol for Underwater Sensor Networks [C]// Proceedings of Networking 2006. LNCS 3976, Heidelberg; Springer-Verlag, 2006; 1216-1221
- [12] Zhou Z, Cui J, Bagtzoglou A. Scalable Localization with Mobility Prediction for Underwater Sensor Networks [C]// Proceedings of IEEE INFOCOM. Phoenix, Arizona, USA, 2008; 2198-2206
- [13] Xu M, Liu G. Design of a P2P based Collaboration Platform for Underwater Acoustic Sensor Network [C]// Proceedings of 11th IEEE International Symposium on Communications and Information Technologies. Hangzhou, China, 2011; 136-141
- [14] Toussaint G. The Relative Neighborhood Graph of A Finite Planar Set[J]. Pattern Recognition, 1980, 12(4):261-268
- [15] Bose P, Devroye L, Evans W, et al. On the Spanning Ratio of Gabriel Graphs and Beta-Skeletons [C]// Proceedings of the Latin Theoretical Informatics Conference. Berlin, Springer-Verlag, 2002; 479-493
- [16] Supowit K. The Relative Neighborhood Graph with An Application to Minimum Spanning Trees [J]. Journal of Association for Computing Machinery, 1983, 30(3):428-448
- [17] Xie P, Zhou Z, Peng Z, et al. Aqua-Sim: An NS-2 Based Simulator for Underwater Sensor Networks [C]// OCEANS 2009, MTS/IEEE Marine Technology for Our Future, Global and Local Challenges. Biloxi, USA; IEEE Computer Society, 2009; 1-7