

DOI: 10.13973/j.cnki.robot.2017.0089

一种机器人云平台服务构建与调度新方法

周凤余, 尹磊, 宋锐, 田天, 陈宏兴

(山东大学控制科学与工程学院, 山东 济南 250061)

摘要: 针对服务机器人服务扩展和服务质量提升等问题, 提出了一种机器人云平台服务构建与调度新方法。首先, 详细设计了服务机器人云平台整体架构及接口层开发框架; 其次, 采用基于历史数据并动态更新的服务质量评估算法实现了接口层对云平台服务的质量评价与调度; 然后, 基于 Apache CXF 实现了云平台服务的开发; 最后, 选择人脸特征提取和文件上传等典型服务实例对提出的方法进行了性能测试, 并将本文算法与目前常用算法进行了对比实验。实验验证了本文提出的机器人云平台服务构建与调度方法的有效性和先进性。该方法为云机器人的开发者提供了高效的服务开发方法, 消除了异构机器人调用服务问题并提高了服务代码的复用性。

关键词: 机器人云平台; 服务质量; 服务调度; 云计算; 异构性

中图分类号: TP24

文献标识码: A

文章编号: 1002-0446(2017)-01-0089-10

A Novel Building and Scheduling Method of Cloud Platform Services for Robot

ZHOU Fengyu, YIN Lei, SONG Rui, TIAN Tian, CHEN Hongxing

(School of Control Science and Engineering, Shandong University, Jinan 250061, China)

Abstract: In view of services extension and service quality improvement, a novel building and scheduling method of cloud platform services for robot is presented. Firstly, an architecture of service robot cloud platform and a framework for interface layer development are designed in detail. Secondly, evaluation and scheduling of cloud platform service are realized by the interface layer with a service quality evaluation algorithm based on historical data and dynamically updated data. Thirdly, the service of cloud platform is developed based on Apache CXF. Finally, the performance of the proposed method is tested through some typical service instances such as face feature extraction service and file upload service. The proposed algorithm is compared with some common algorithms through experiments. The experimental results verify the reliability and advancement of the service processing method in service building and scheduling. The proposed method is an efficient service development method for cloud robot developers to eliminate the heterogeneity problem and reuse the procedure.

Keywords: robot cloud platform; quality of service; service scheduling; cloud computing; heterogeneity

1 引言 (Introduction)

服务机器人工作在与人共处、高度动态的复杂环境中, 仅依靠自身配备的传感器和计算能力, 无法从根本上解决机器人自主性差、智能水平低及成本高等一系列问题, 阻碍了其推广应用。2010年, 卡耐基·梅隆大学的 Kuffner 教授^[1-3]提出了“云机器人”的概念, 将机器人学和云计算相结合, 把机器人的数据处理、规划、决策、协作等复杂计算功能卸载到云端, 机器人本体只需配备简单的网络设备和基本的传感器就可以完成复杂的服务任务, 为彻底解决机器人“物美”与“价廉”的矛盾提供了新思路。Arumugam^[4]等人开展了 DAVinCi 项目, 该项目利用云平台作为机器人的大脑, 实现

了快速 SLAM (同步定位与地图创建) 算法。2011年, 欧洲科学家启动 RoboEarth 项目^[5], 机器人通过云平台进行信息共享和相互学习。Doriya 等人^[6]提出 Robot-cloud 云计算框架, 为异构机器人提供服务。2013年, Kehoe 等人^[7-8]提出一种基于云的物体抓取系统框架, 实现了云机器人在家居环境下的物体识别和抓取。2015年, 基于 RoboEarth, Mohanarajah 等^[9]提出了 Raptuya 云机器人平台, Raptuya 不仅通过 RoboEarth 为机器人提供资源, 而且优化了机器人任务管理、数据结构及通信协议。在国内, 2011年, 清华大学 Du 等人^[10]提出了 RCC (robot cloud center) 框架和 RaaS (robot as a service, 机器人即服务) 概念。2012年, 北京理工大学的赵连翔等人^[11]提出机器人云操作平台,

实现了机器人的遥操作。

云机器人的优势在于云平台可以为机器人提供计算密集型服务,但上述文献大都采用点对点的服务提供方式,机器人与云平台的接口不统一,导致云平台需要重复构建服务,降低了云平台服务的效率和质量。2015年,陈宏兴等人^[12]设计了一种服务机器人云计算平台 SOA (面向服务的体系架构) 接口层模型,由于 SOA 把不同应用封装成了统一服务,屏蔽了系统间的差异性,因此在云平台中引入 SOA 解决了平台与机器人间的强耦合问题,它的优劣直接影响到服务的复杂度、耦合度和运行效率^[13-14]。Cai^[15]等人构建了多机器人多层 SOA 架构并实现了 SLAM 实例服务。Insaurralde^[16]采用 SOA 服务模型解决了多无人机的服务调用问题,但他们均未考虑服务提供者的能力提升问题。

针对上述问题,并基于服务机器人的使用特点,本文提出了一套适用于医院、养老院及家庭等

实际场景的基于 SOA 接口层的机器人云平台服务质量评估、服务调度以及云计算服务开发新方法,目的是依靠云平台的分布式架构为大量异构服务机器人提供高效的计算密集型服务。通过大量实验对所提方法的可行性和有效性进行了验证。

2 服务机器人云平台总体架构 (Architecture of the cloud platform for service robot)

本文搭建的服务机器人云平台总体架构如图 1 所示,分为 4 层:物理资源层,云平台基础层,服务实现层和 SOA 接口层。

物理资源层属于 IaaS (infrastructure as a service, 基础设施即服务) 层次服务,将基础硬件虚拟化为平台可用的硬件资源,为云平台各服务实例提供运行环境支持。云平台底层使用 OpenStack 进行硬件的定制和灵活管理。

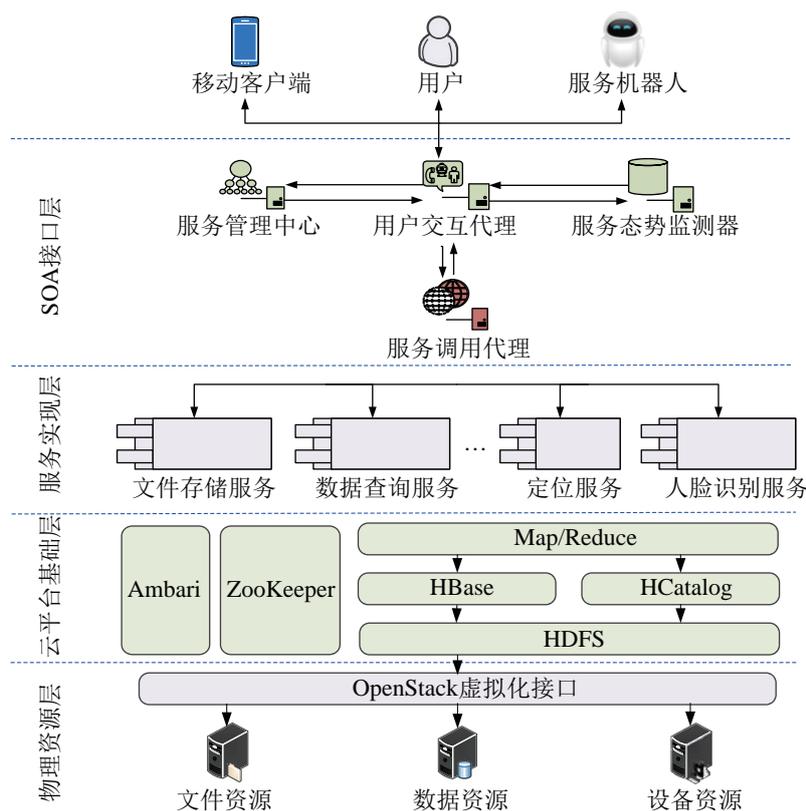


图 1 服务机器人云平台架构图

Fig.1 Architecture of the cloud platform for service robot

云平台基础层属于 PaaS (platform as a service, 平台即服务) 层次的云计算服务,负责数据存储管理,为服务提供统一的访问接口。基础层基于 Apache Hadoop 项目搭建,其核心为分布式文件系统 HDFS 和并行计算模型 Map/Reduce^[17-18]。基础

层的部署基于集群配置工具 Ambari 实现,使用分布式数据库 HBase 和数据管理工具 HCatalog 处理数据,并使用分布式应用程序协调服务 Zookeeper 确保数据的强一致性。

服务实现层进行服务的封装,由平台开发者更

新、管理和维护。平台在 SOA 接口层提供服务的配置和管理接口, 服务态势信息由 SOA 接口层监测和记录。

SOA 接口层是平台与用户交互的中间件, 主要实现用户交互、服务管理、服务调用和服务态势监测等功能。

3 SOA 接口层的交互过程 (Interactive process of SOA interface)

服务机器人云平台是机器人综合应用和开发平台, SOA 接口层是机器人和云平台之间的桥梁和纽带, 它的引入使得云平台可方便地为异构服务机器人提供通用服务。SOA 接口层以 Apache CXF 框架为基础, 该框架规定了服务的设计依据、标准和通信协议, 并对消息进行编码和解码。机器人、服务接口层与云计算平台之间的交互过程见图 2。

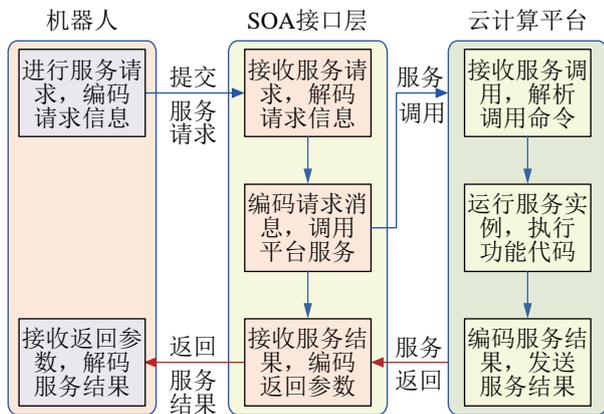


图 2 SOA 接口层与平台的交互过程

Fig.2 Interactive process between SOA interface layer and platform

当服务机器人申请服务时, Apache CXF 框架对服务请求信息进行编码并传输至 SOA 接口层, 完成服务请求的提交。

SOA 接口层接收到服务申请后对其进行编码, 生成服务调用指令, 在调用云平台服务的同时进入等待接收服务结果线程。

云计算平台收到接口层服务调用要求后, 解析调用指令的机器人 ID、服务名称等参数, 使用 Apache CXF 运行服务实例, 执行功能代码并将结果编码后返回给 SOA 接口层。

SOA 接口层将接收到的服务结果编码后传送给机器人, 完成整个的服务调用过程。

4 机器人云平台服务质量评估与调度 (Quality estimation and scheduling of cloud platform services for robot)

服务机器人云平台的质量评估和调度过程是通

过 SOA 接口层实现的。根据接口层的服务设计、机器人云平台的调用方法及服务态势检测机制, 本文采用服务质量 (QoS) 评估方法实现了云平台服务调度。

4.1 服务设计与信息监测

(1) 接口层服务设计

SOA 接口层将云平台的各功能封装成服务, 服务机器人通过调用接口层服务使用云平台提供的各项功能。

接口层提供了服务的管理和调用方法。服务管理方法包括服务的注册、查询和修改等, 为机器人提供云平台所有服务的管理接口。

机器人通过 SOA 接口层发布的中间层服务 CallService 来调用云平台的功能性服务, 该服务为机器人提供云平台各功能的调用方法。机器人云平台封装并发布相应的算法及功能性服务, 在机器人执行任务的过程中通过调用 CallService 向云平台请求服务, 由接口层将服务申请映射为云平台可识别的操作指令, 通过运行云平台功能代码完成云计算任务。

(2) 云平台服务调用

用户交互代理是云平台 and 用户间的媒介。用户向用户交互代理提交服务申请后, 首先进行身份校验, 然后通过服务发现和评估, 进行服务最优路径的选择, 服务调用后向用户返回服务结果。用户交互代理结构如图 3 所示。

用户交互代理使用输入输出队列的双队列方式完成申请和结果的处理, 使用服务申请队列暂存用户的服务申请, 降低平台并发请求负载; 使用调用结果队列暂存返回结果, 并允许用户以同步和异步两种方式获取服务调用的结果。

(3) 服务信息监测

服务态势监测器主要是对机器人服务调用过程中的数据进行收集和处理。服务调用代理在云平台创建服务代理后, 在动态数据区中添加服务实例, 开辟数据区, 追踪服务实例的态势信息。服务实例运行时, 服务调用代理使用更新器完成态势信息的更新, 调用结束后使用合并器将动态态势信息合并至 QoS 信息中, 并使用删除器清理数据区。使用评估器获取 QoS 信息, 并完成服务的评估工作, 服务态势监测器的结构如图 4 所示。

4.2 服务调度管理设计

云平台服务调度管理主要是对服务池中的功能性服务进行质量评估、调度, 以提高多机器人服务调用效率。

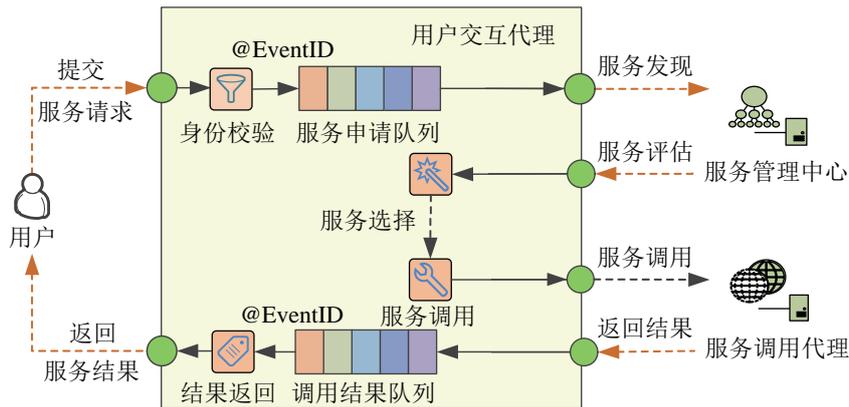


图3 用户交互代理的结构图

Fig.3 Structure of the interaction agent for user

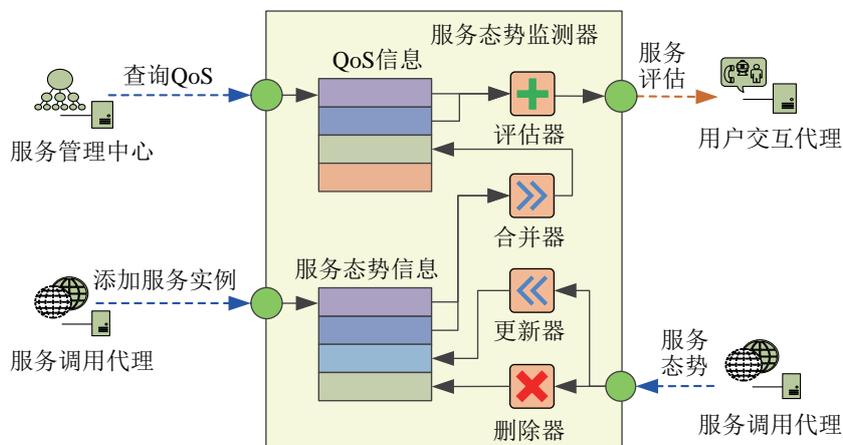


图4 服务态势监测器的结构图

Fig.4 Structure of the service state monitor

4.2.1 服务评估

根据服务机器人云平台的特点，考虑指标的监测难度和监测成本，本文的服务评价机制关注 QoS 的时间性和可靠性两类指标^[9]。

时间类指标选用调用周期 q_{echo} 。调用周期指标描述从服务代理调用实例至此线程结束运行的时间间隔，时间类指标可综合评价节点对服务的执行效率。

可靠性指标选用成功率 $u_{success}$ 。描述服务提供者提供的被调用服务次数中成功调用的几率，从频率角度描述该服务的可用性。可靠性指标对 QoS 的影响较大。

在 QoS 属性中，可靠性为正向属性，调用周期为负向属性。假设属性数据为 q_i ($i = 1, 2, \dots, n$, n 为数据个数)，对属性数据 q_i 进行归一化处理去除属性的量纲， q_i 的值越大越好，归一化公式为

$$q_i = \begin{cases} \frac{q_i - \min q_i}{\max q_i - \min q_i}, & \max q_i - \min q_i \neq 0 \\ 1, & \max q_i - \min q_i = 0 \end{cases} \quad (1)$$

$$q_i = \begin{cases} \frac{\max q_i - q_i}{\max q_i - \min q_i}, & \max q_i - \min q_i \neq 0 \\ 1, & \max q_i - \min q_i = 0 \end{cases} \quad (2)$$

式 (1) 为正向属性的归一化方法，式 (2) 为负向属性的归一化方法。评估 QoS 时应当综合时间性指标和可靠性指标，时间指标使用加权平均方式，可用性指标使用相乘方式，可得到 QoS 的定义函数：

$$Q_i = q_{echo} \times w_{echo} + u_{success} \times w_{reliability} \quad (3)$$

式中， w_{echo} 、 $w_{reliability}$ 为 QoS 聚合属性值的权重值， $w_{(*)} \in (0, 1)$ 且 $w_{echo} + w_{reliability} = 1$ ； Q_i 越大，服务质量越好。本文取 $w_{echo} = 0.4$ 、 $w_{reliability} = 0.6$ 。QoS 的评估由服务态势监测器完成，所需指标保存在动态信息区中，每次服务调用完成后由 QoS 更新器进行动态更新。QoS 评估器以服务提供者作为粒度对服务按照式 (3) QoS 函数进行服务质量评估，并将评估结果返回至服务选择模块。

4.2.2 服务调度

云平台是机器人的服务提供者，在云平台的

服务池中有多种服务等待调用. 云服务器 ECS (elastic compute service) 的硬件配置及网络状况等导致服务质量参差不齐, 为了提高机器人的服务调用效率, 本文提出如图 5 所示的调度方法.

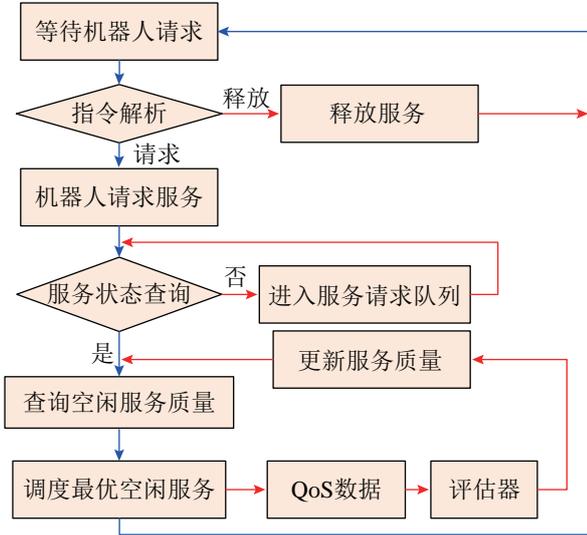


图 5 服务调度过程

Fig.5 The process of service scheduling

(1) 开发人员采用 Apache CXF 在云平台服务池开发功能性服务并定义相关接口参数.

(2) 服务管理中心等待机器人请求, 收到机器人请求后通过指令解析, 如果指令为释放服务, 则调度器根据指令参数释放相应服务并继续等待机器人指令. 如果为请求服务指令, 则进入第(3)步.

(3) 机器人的服务调用请求需经过身份验证发送至服务调度管理中心, 管理中心收到调用请求后查询服务池中功能性服务的空闲状态, 如果服务池中不存在机器人需要的空闲服务, 机器人请求进入服务请求队列等待空闲服务, 如果服务池中存在空闲服务则进入服务选择阶段.

(4) 服务调度管理中心采用 QoS 评估器对云服务池内的空闲服务进行排名, 将最优服务提供者的 IP 地址及接口参数发送给机器人, 机器人在调用服务的过程中, 服务态势检测器将实时地收集 QoS 信息并发送给服务评估器, 对新一轮的 QoS 进行评估并产生新的排名, 等待进入下一次服务调度过程.

5 云计算服务开发 (The development of cloud computing services)

在云平台中, 服务以服务定义和服务实例两种方式存在. 云平台用户可对服务定义进行更新和修改. 服务实例在平台中用于完成特定功能的线程.

SOA 接口层对服务的检测是基于服务的反馈方

式. 接口层将服务的定义和态势信息实时更新至云平台.

服务实例在设计中需要满足的共性要求称为云平台的服务实例模板, 如图 6 所示.

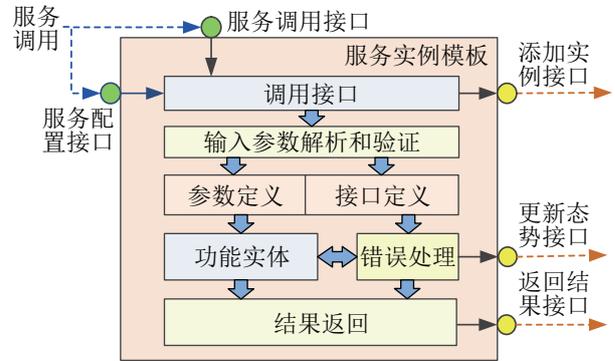


图 6 云平台的服务实例模板

Fig.6 Service instance template of the cloud platform

服务实例模板描述了服务接口和服务实例的基本结构. 服务开发者按照 SOA 服务模板设计与接口层配套使用的服务. 服务实例在运行中向服务态势监测器反馈服务的实时态势. 添加实例接口用于在实例创建后向服务态势监测器中添加记录区, 更新态势接口用于反馈服务实例中功能实体的执行结果和错误情况, 反馈接口用于向服务机器人返回服务的运行情况.

云平台基于 Apache CXF 框架进行服务的开发, 平台开发者在云平台中部署、注册并发布云计算服务. 云平台采用面向服务的模型封装系统功能, 屏蔽了异构机器人之间的平台差异, 实现了跨平台的服务调用及响应.

6 实验 (Experiment)

6.1 实验条件

实验用机器人照片如图 7 所示, 分别是研究团队自主开发的陪护机器人、护士机器人和智能陪伴机器人. 陪护机器人配置 X86 架构联想 IdeaCentre Q180 迷你主机, Windows 7 操作系统; 护士机器人配置联想扬天 V370 A2410 笔记本电脑, Ubuntu 系统; 智能陪伴机器人配置 ARM 架构树莓派 II, Linux 操作系统. 所有机器人均配置索尼 IMX219 高清摄像头和 WiFi 模块; 云平台硬件采用 3 台阿里云 ECS 服务器 (4 核 CPU, 8G 内存, 10 M 带宽) 构成, 系统软件均采用 Java 语言编写, 开发工具采用 Eclipse-4.4, 接口层采用 Apache CXF 网络服务框架, 分布式服务器集群基础框架采用 Hadoop-0.20.2 版本.



(a) 陪护机器人 (b) 护士机器人 (c) 智能陪伴机器人

图7 机器人实验平台

Fig.7 Experimental platform of the robot

6.2 服务发布及调用

云平台使用 Apache CXF 开发的接口层发布 CallService 服务，该服务为中转服务，为调用其他功能服务的中间件。所有的异构服务机器人均可以根据云平台服务器 IP 地址查看 CallService 的网络服务描述信息，图 8 给出的是陪护机器人查询的网络服务描述信息，其他机器人与之相同。

```
<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://test/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap12="http://schemas.xmlsoap.org/soap/12/"
name="ServiceRequestHandlerService" targetNamespace="http://test/">
<wsdl:types>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema"
xmlns:xmime="http://www.w3.org/2005/05/xmime" xmlns:tns="http://test/"
elementFormDefault="unqualified" targetNamespace="http://test/" version="1.0">
<xs:import namespace="http://www.w3.org/2005/05/xmime"/>
<xs:element name="CallService" type="tns:CallService"/>
<xs:element name="CallServiceResponse" type="tns:CallServiceResponse"/>
<xs:element name="CallService" type="tns:CallService"/>
<xs:element name="CallServiceResponse" type="tns:CallServiceResponse"/>
<xs:complexType name="CallService">
<xs:complexType name="CallServiceResponse">
<xs:complexType name="CallService">
<xs:complexType name="myFile">
<xs:complexType name="CallServiceResponse">
</xs:schema>
</wsdl:types>
<wsdl:message name="CallServiceResponse">
<wsdl:message name="CallServiceResponse">
<wsdl:message name="CallService">
<wsdl:message name="CallService">
<wsdl:part element="tns:CallService" name="parameters"/>
</wsdl:message>
<wsdl:portType name="CallService">
<wsdl:binding name="ServiceRequestHandlerServiceSoapBinding" type="tns:CallService">
<wsdl:service name="ServiceRequestHandlerService">
<wsdl:port binding="tns:ServiceRequestHandlerServiceSoapBinding" name="ServiceRequestHandlerPort">
<soap:address location="http://192.168.200.132:8080/CallService"/>
</wsdl:port>
```

图8 CallService 的网络服务描述信息

Fig.8 Web service information of CallService

```
<wsdl:definitions name="FeatureServiceImplService" targetNamespace="http://test/">
<wsdl:types>
<xs:schema elementFormDefault="unqualified" targetNamespace="http://test/" version="1.0">
<xs:import namespace="http://www.w3.org/2005/05/xmime"/>
<xs:element name="Feature" type="tns:Feature"/>
<xs:element name="FeatureResponse" type="tns:FeatureResponse"/>
<xs:complexType name="Feature">
<xs:sequence>
<xs:element minOccurs="0" name="arg0" type="tns:myFile"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="myFile">
<xs:sequence>
<xs:element minOccurs="0" name="mDataHandler" type="xs:base64Binary"
xmime:expectedContentTypes="application/octet-stream"/>
<xs:element minOccurs="0" name="file" type="xs:base64Binary"
xmime:expectedContentTypes="application/octet-stream"/>
<xs:element minOccurs="0" name="fileExtension" type="xs:string"/>
<xs:element minOccurs="0" name="fileName" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="FeatureResponse">
<xs:sequence>
<xs:element name="return" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

图9 FeatureService 的网络服务描述信息

Fig.9 Web service information of FeatureService

下面以人脸特征提取服务为实例验证系统的灵活性。云平台封装基于 OpenCV 的人脸特征提取程序为 FeatureService 服务，发布并在接口层注册。

图 9 给出的是护士机器人平台查询的网络服务描述信息，其他机器人与之相同。

云平台下的所有机器人只要配置 Apache CXF 环境，编写如图 10 所示的服务调用代码，调用 CallService 服务，传入机器人 ID、服务名称 FeatureService、操作文件属性等参数即可实现对云平台服务的调用。

```
/*设置文件属性*/
MyFile fileWrapper = new MyFile();
String filePath = "E:\\test\\set.xml";
DataSource source = new FileDataSource(new File(filePath));
fileWrapper.setFile(new DataHandler(source));
/*新建服务代理对象*/
JaxWsProxyFactoryBean factory = new JaxWsProxyFactoryBean();
factory.setServiceClass(CallService.class);
/*设置MTOM属性*/
Map<String, Object> props = new HashMap<String, Object>();
props.put("mtom-enabled", Boolean.TRUE);
factory.setProperties(props);
factory.setAddress("http://192.168.200.132:8080/CallService");
/*调用服务*/
CallService serviceproxy = factory.create(CallService.class);
JSONArray jsonArray = serviceproxy.CallService("{\"ID\":\"101\", \"service_name\":\"FeatureService\"}");
fileWrapper;
```

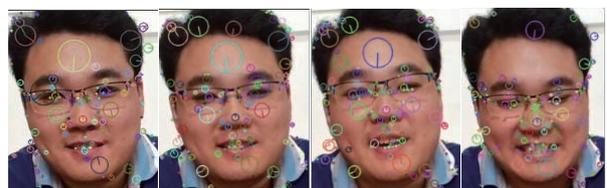
图10 机器人服务调用代码

Fig.10 Codes for calling robot service

云平台采用 Apache CXF 封装图像 SIFT（比例不变特征变换）特征算法并发布为 FeatureService 服务。机器人调用接口层发布的 CallService 服务，传入机器人 ID、服务名称 FeatureService 等参数，实现对机器人采集到的图像特征点数据的实时提取。图 11(a) 为机器人采集到的实时图片，图 11(b) 为机器人调用服务后对图像特征点的提取，最终机器人获取包含实时图片全部特征点信息的 JSON 数据。



(a) 机器人采集的人脸样本



(b) 服务器端对人脸的特征点提取

图11 特征点提取服务实验

Fig.11 Experiment of feature extraction service

由图 8 ~ 10 可以看出，云平台发布的服务不因系统的异构而不同。服务调用代码采取统一的调用方式，且代码复杂度较低，只需要配置 CXF 环境，调用函数传入参数即可实现对云平台发布的复

杂服务的调用, 因此机器人本体无需安装任何复杂的 SDK 开发包及库文件, 大大降低了程序的开发难度, 提高了复杂服务代码的复用率并消除了机器人平台的异构性, 实现了跨平台服务. 由图 12 可知, 机器人调用人脸识别服务的时间周期均值仅为 100 ms, 完全满足机器人对人脸特征提取的实时性要求.

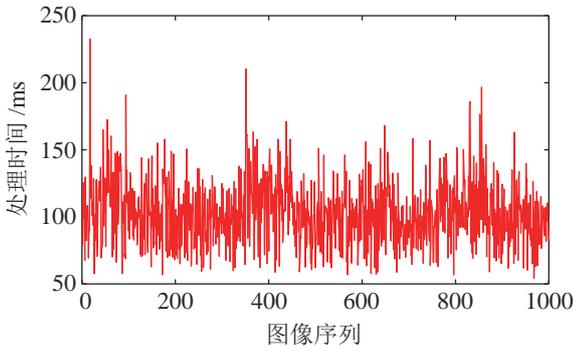


图 12 特征点提取服务调用周期
Fig.12 Time cycle of calling the feature point extraction service

6.3 服务调度

为了便于说明服务调度流程, 该实验采用了 3 台阿里云 ECS 服务器, 发布 3 个相同服务, 动用 2 台异构机器人. 在大规模应用时, 服务器数目、发布的服务数量均需随机器人数量增加而动态调整, 否则会造成机器人的服务请求长久堵塞于服务请求队列中, 导致服务调用周期过长, 降低系统的实时性. 但无论规模有多大, 服务调度过程都是相同的.

本文的服务调度实验以云平台发布的文件上传服务为实例, 护士机器人和陪伴机器人为服务请求平台. 云平台 SOA 接口层发布 CallService 服务, 云平台服务器在云平台服务池内发布 3 个 Upload-FileService 服务, 以下简称为服务 A、B 和 C. 图 13 为机器人实际调用服务时采集的 3 个服务的调用周期和可靠率, 均作为服务质量评估函数的历史数据 (服务被调用时自动更新), 从图 13(a)、(c)、(e) 可知服务 A、B、C 的调用周期分别为 200 ms、300 ms 和 400 ms, 对应的服务调用成功率分别为 0.75、0.60 和 0.50, 如图 13(b)、(d)、(f) 所示. 图 13(g) 为通过服务质量评估函数式 (3) 计算的 Q_i 值 (Q_i 越大, 服务质量越好), 从图中可知服务 A 的 Q_i 值为 0.8, 质量要优于 B 和 C 的 0.7 和 0.56. 接口层对服务质量排名为服务 A > 服务 B > 服务 C.

启动护士机器人通过接口层服务 CallService 调用云平台的服务, 服务调度器根据服务池中的服务

排名调度服务 A 为护士机器人使用, 服务质量评估函数根据实时调用数据计算服务质量权值 Q_i 并在云平台服务池中更新服务 A 的质量排名. 从图 14(a) 可以看出, 服务 A 的调用周期为 340 ms, 比图 13(a) 服务 A 调用周期 250 ms 有所增加, 与网络的稳定度有关, 对整体的服务调度过程没有影响. 图 14(b) 为 QoS 评估函数式 (3) 计算获得的 QoS 权值 Q_i , 均值为 0.9. 此时, 服务 B 还未被调用, 服务 B 的历史数据仍采用图 13(g) 中的数据.

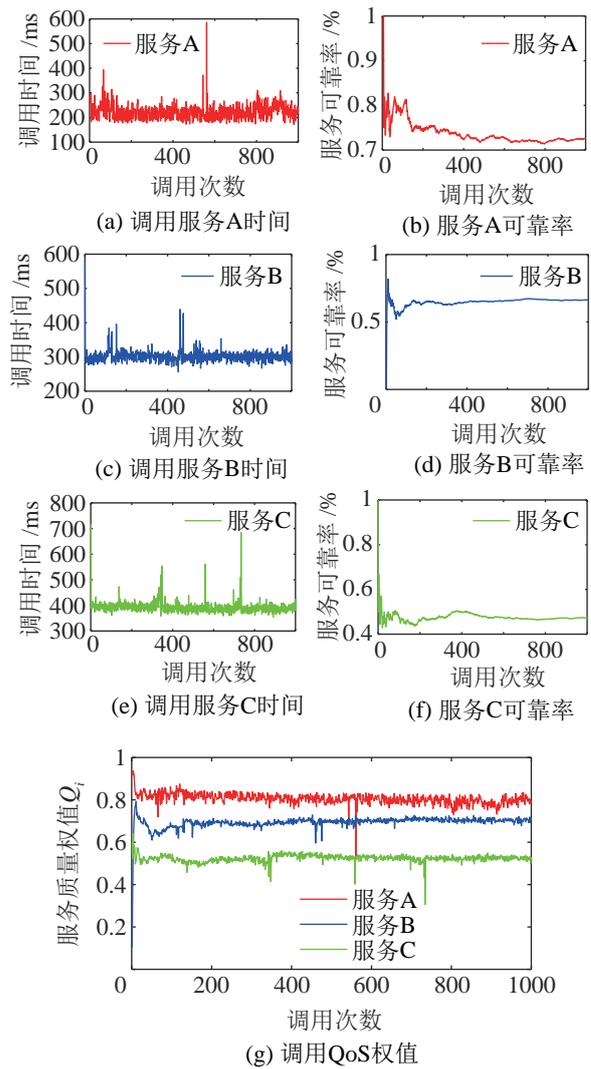


图 13 云平台发布服务的质量

Fig.13 The quality of the service published by cloud platform

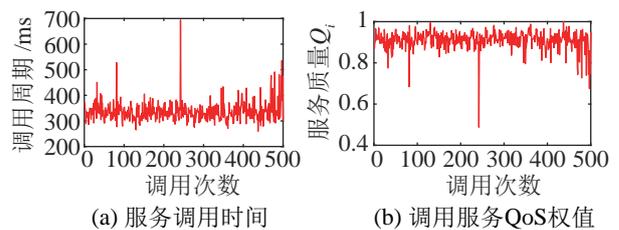


图 14 护士机器人服务调度

Fig.14 Service scheduling for the nurse robot

在护士机器人调用服务 A 的过程中，启动陪伴机器人加入系统调用云平台服务，服务调度器根据服务池中的服务排名为陪伴机器人分配服务 B，图 15(a) 表明陪伴机器人一开始调用的是服务 B，接口层根据陪伴机器人的实时调用数据更新服务 B 的 QoS 权值 Q_i 和服务池的质量排名。

当 2 台机器人均进入正常调用后，关闭护士机器人退出服务调度过程，此时服务 A 处于无调用状态，服务调度器根据服务 QoS 排名调度服务 A 为陪伴机器人使用，具体调度过程如图 16 所示。可以看出，陪伴机器人的调用周期从第 304 次的 500 ms 突然降低至 300 ms，即为服务切换过程，通过图 15(b) 得出，服务质量也从 0.8 提升到 0.9，完成服务调度过程。

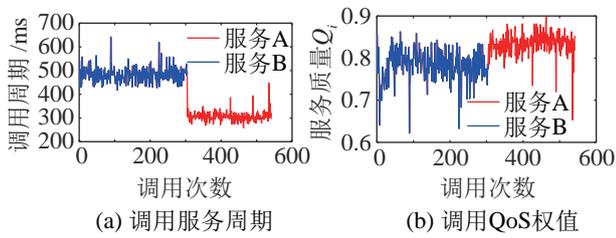


图 15 两台机器人服务调度
Fig.15 Service scheduling for two robots

6.4 对比实验及性能分析

对比实验采用 CloudSim 云计算仿真软件，构建 10 个云服务中心和 500 个机器人用户。将本文算法与目前常用的 max-min 算法^[20]、min-min 算法^[21]和 ICA (immune clonal algorithm) 算法^[22]进行了对比实验，并分别在可用性、负载均衡离差和有效时间 3 个指标上进行了对比分析。

(1) 可用性测试实验

可用性指标 u_s 反映了机器人调用服务成功的概率^[23]。任务 T_j ($j = 1, 2, \dots, m$, m 为机器人个数) 选择服务 s_i ($i = 1, 2, \dots, n$, n 为服务数量) 后， u_s 计算公式为

$$u_s = \sqrt{\frac{1}{m} \sum_{j=1}^m \sum_{i=1}^n \xi_{ij}}, \xi_{ij} = \begin{cases} 1, & s_i \text{ 对应的 } T_j \text{ 成功} \\ 0, & \text{其他} \end{cases} \quad (4)$$

由图 16 可以发现，随着机器人数量的增加，算法的可用性均在降低。当机器人数量小于 100 台时，4 种算法的可用性差别并不大；当机器人数量为 100 ~ 200 台时，ICA 算法的可用性略低；当机器人数量达到 500 台时，本文提出的 QoS 算法的可用性分别比 min-min、max-min 算法高出 14% 和 12%，比 ICA 算法高出 7%，并且算法的可用性始

终保持在 87% 以上，波动相对较小，验证了本文提出的服务模型能有效提高服务的质量和稳定性。

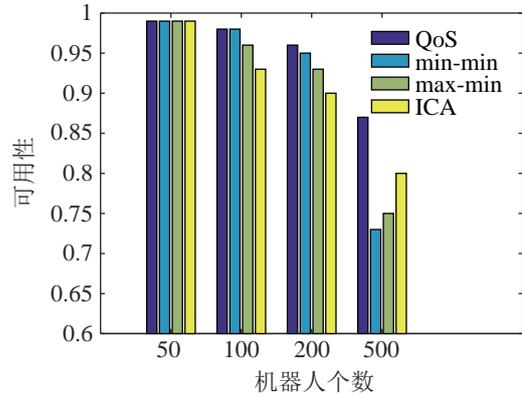


图 16 可用性测试结果
Fig.16 Results of the availability performance

i

(2) 负载均衡离差测试实验

负载均衡离差 σ 主要反映了云平台服务器的负载均衡情况，即为云平台的资源公平利用率，避免造成服务器资源的浪费。负载均衡离差 σ 满足：

$$\sigma = \sqrt{\frac{1}{n-1} \cdot \sum_{i=0}^n (d_i - \bar{d})^2}, i \in \{1, 2, 3, \dots, n\} \quad (5)$$

式中，

$$d_i = \frac{c_i}{\sum_{i=0}^n c_i} / \frac{l_j}{\sum_{j=0}^m l_j}, \quad (6)$$

$$i \in \{1, 2, 3, \dots, n\}; j \in \{1, 2, 3, \dots, m\}$$

其中， c_i 为服务 s_i 的费用， l_j 为任务 T_j 的计算量。

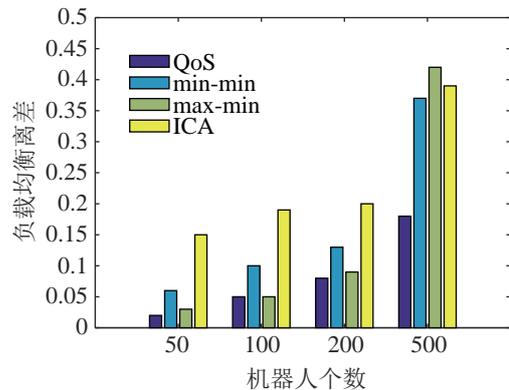


图 17 负载均衡离差测试结果
Fig.17 Results of the load balancing deviation

从图 17 可知，随着机器人数量的增加，负载均衡离差逐渐增大，但本文提出的 QoS 算法负载均衡离差维持在 0.02 ~ 0.20 之间，均低于其它 3 种算法，特别当机器人数量达到 500 台时，负载均衡

离差比 min-min 算法、max-min 算法和 ICA 算法分别低 19%、24% 和 21%，云平台资源利用率相对较高。

(3) 有效时间测试实验

有效时间 \bar{v}_i 反映了机器人云平台任务执行和服务调度的时间有效利用率，无效利用时间主要消耗在等待任务执行。有效时间 \bar{v}_i 满足

$$\bar{v}_i = \frac{1}{m} \cdot \sum_{j=0}^m \frac{l_j/e_j}{\Delta t_j}, j \in \{1, 2, \dots, m\}; i \in \{1, 2, \dots, n\} \quad (7)$$

式中, Δt_j 为任务 T_j 的实际提交时间和最终完成时间差, e_i 为任务 T_j 所分配的服务 s_i 的实际处理能力。

如图 18 所示, 随着机器人数量的不断增加, 机器人的有效时间 \bar{v}_i 呈现不断下降的趋势, 当服务数量低于 50 时, 算法的有效时间基本都接近 99%, 但当数量增加到 500 台时, 本文提出的 QoS 算法有效时间比 min-min 和 max-min 算法高出近 20%, 比 ICA 算法高出 10%, 机器人调用服务的等待时间相对较小, 云平台服务的实时性较好。

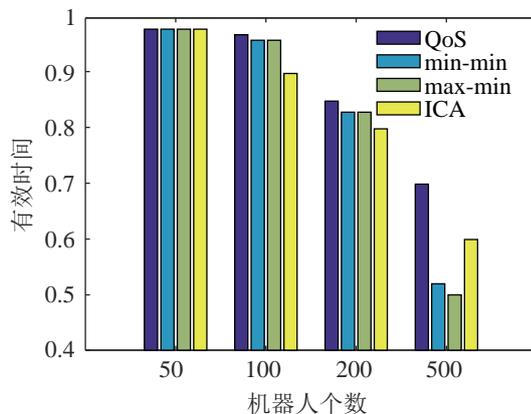


图 18 有效时间测试结果

Fig.18 Results of the effective time test

7 结论 (Conclusion)

随着云机器人技术的发展, 服务机器人云平台的研究也更加深入。本文提出了一种云机器人服务模式构建方法, 详细地设计了接口层配套的服务模板, 并实现了对人脸特征提取、文件上传等具体服务的跨平台调用, 提升了云平台的可扩展性和对异构机器人的兼容性; 采用 QoS 评价机制对云平台服务池内的服务进行质量评估并为机器人提供最优服务, 实现了接口层对云平台服务的调度和管理, 并通过与常用算法的对比验证了本文提出的机器人云平台服务构建和调度方法在服务质量、资源利用率及实时性等方面的优越性。

参考文献 (References)

- [1] Kuffner J J, LaValle S M. Space-filling trees: A new perspective on incremental search for motion planning[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2011: 2199-2206.
- [2] Sugiura K, Shiga Y, Kawai H, et al. A cloud robotics approach towards dialogue-oriented robot speech[J]. Advanced Robotics, 2015, 29(7): 449-456.
- [3] Wu H Y, Lou L, Chen C C, et al. Cloud-based networked visual servo control[J]. IEEE Transactions on Industrial Electronics, 2013, 60(2): 554-566.
- [4] Arumugam R, Enti V R, Liu B B, et al. DAVinCi: A cloud computing framework for service robots[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2010: 3084-3089.
- [5] Tenorth M, Perzylo A C, Lafrenz R, et al. Representation and exchange of knowledge about actions, objects, and environments in the ROBOEARTH framework[J]. IEEE Transactions on Automation Science and Engineering, 2013, 10(3): 643-651.
- [6] Doriya R, Chakraborty P, Nandi G C. 'Robot-Cloud': A framework to assist heterogeneous low cost robots[C]//International Conference on Communication, Information & Computing Technology. Piscataway, USA: IEEE, 2012.
- [7] Kehoe B, Matsukawa A, Candido S, et al. Cloud-based robot grasping with the google object recognition engine[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2013: 4263-4270.
- [8] Kehoe B, Warriar D, Patil S, et al. Cloud-based grasp analysis and planning for toleranced parts using parallelized Monte Carlo sampling[J]. IEEE Transactions on Automation Science and Engineering, 2015, 12(2): 455-470.
- [9] Mohanarajah G, Hunziker D, D'Andrea R, et al. Rapyuta: A cloud robotics platform[J]. IEEE Transactions on Automation Science and Engineering, 2015, 12(2): 481-493.
- [10] Du Z H, Yang W Q, Chen Y N, et al. Design of a robot cloud center[C]//10th International Symposium on Autonomous Decentralized Systems. Piscataway, USA: IEEE, 2011: 269-275.
- [11] 赵连翔, 王全玉, 贾金苗, 等. 机器人云操作平台的实现研究[J]. 华中科技大学学报: 自然科学版, 2012(S1): 161-164.
- [12] Zhao L X, Wang Q Y, Jia J M, et al. Research on realizing robot cloud-operation platform[J]. Journal of Huazhong University of Science and Technology: Nature Science, 2012(S1): 161-164.
- [12] 陈宏兴, 周风余, 田天, 等. 服务机器人云计算平台 SOA 接口层模型设计[J]. 山东大学学报: 工学版, 2015, 45(4): 31-39.
- [12] Chen H X, Zhou F Y, Tian T, et al. Design of SOA interface model in service robot cloud computing platform[J]. Journal of Shandong University: Engineering Science, 2015, 45(4): 31-39.
- [13] Lee J, Kim J, Lee B, et al. Utilizing semantic Web 2.0 for self-reconfiguration of SOA based agent applications in intelligent service robots[C]//IEEE International Conference on Computer and Information Technology. Piscataway, USA: IEEE, 2008: 784-789.
- [14] Yang T H, Lee W P. A service-oriented framework for the development of home robots[J]. International Journal of Advanced Robotic Systems, 2013, 10: No.122.

- [15] Cai Y F, Tang Z M, Ding Y H, et al. Theory and application of multi-robot service-oriented architecture[J]. IEEE/CAA Journal of Automatica Sinica, 2016, 3(1): 15-25.
- [16] Insaurralde C C. Service-oriented agent architecture for unmanned air vehicles[C]//IEEE/AIAA 33rd Digital Avionics Systems Conference. Piscataway, USA: IEEE, 2014: 8B1-1-8B1-14.
- [17] Wu Y W, Ye F, Chen K, et al. Modeling of distributed file systems for practical performance analysis[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(1): 156-166.
- [18] Zhang F, Cao J W, Khan S U, et al. A task-level adaptive MapReduce framework for real-time streaming data in healthcare applications[J]. Future Generation Computer Systems, 2015, 43-44(C): 149-160.
- [19] Wang Y. A formal model of QoS-aware web service orchestration engine[J]. IEEE Transactions on Network and Service Management, 2016, 13(1): 113-125.
- [20] Panda S K, Agrawal P M, Khilar P M, et al. Skewness-based min-min max-min heuristic for grid task scheduling[C]//4th International Conference on Advanced Computing & Communication Technologies. Piscataway, USA: IEEE, 2014: 282-289.
- [21] Chen H, Wang F, Helian N, et al. User-priority guided min-min scheduling algorithm for load balancing in cloud computing[C]//2013 National Conference on Parallel Computing Technologies. Piscataway, USA: IEEE, 2013.
- [22] Yang H H, Zhou X, Wang Y, et al. A new adaptive immune clonal algorithm for underwater acoustic target sample selection[C]//IEEE International Conference of IEEE Region 10. Piscataway, USA: IEEE, 2013.
- [23] 孙大为, 常桂然, 李凤云, 等. 一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法 [J]. 电子学报, 2011, 39(8): 1824-1831.
Sun D W, Chang G R, Li F Y, et al. Optimizing multi-dimensional QoS cloud resource scheduling by immune clonal with preference[J]. Acta Electronica Sinica, 2011, 39(8): 1824-1831.

作者简介:

周凤余 (1969-), 男, 博士, 教授. 研究领域: 智能机器人, 智能控制理论和方法.

尹 磊 (1986-), 男, 博士生. 研究领域: 云机器人, 计算机软件编程.

(上接第 88 页)

- [19] Wu Q C, Wang X S, Du F P. Analytical inverse kinematic resolution of a redundant exoskeleton for upper-limb rehabilitation[J]. International Journal of Humanoid Robotics, 2015, 13(3): doi: 10.1142/S0219843615500425.
- [20] Arakelian V H, Smith M R. Complete shaking force and shaking moment balancing of linkages[J]. Mechanism and Machine Theory, 1999, 34(8): 1141-1153.
- [21] Veer S, Sujatha S. Approximate spring balancing of linkages to reduce actuator requirements[J]. Mechanism and Machine Theory, 2015, 86: 108-124.
- [22] Craig J J. Introduction to robotics: Mechanics and control[M]. Upper Saddle River, USA: Pearson Prentice Hall, 2005.
- [23] Winter D A. Biomechanics and motor control of human movement[M]. Hoboken, USA: John Wiley & Sons, 2009.
- [24] 李庆玲. 基于 sEMG 信号的外骨骼式机器人上肢康复系统研究 [D]. 哈尔滨: 哈尔滨工业大学, 2009.
Li Q L. Study on sEMG based exoskeletal robot for upper limbs rehabilitation[D]. Harbin: Harbin Institute of Technology, 2009.

作者简介:

吴青聪 (1988-), 男, 博士, 讲师. 研究领域: 机器人动力学及其控制, 康复机器人, 先进医疗器械等.

王兴松 (1965-), 男, 博士, 教授, 博士生导师. 研究领域: 机器人动力学及其控制, 生物机械电子, 先进医疗器械等.

吴洪涛 (1962-), 男, 博士后, 教授, 博士生导师. 研究领域: 机器人学及智能机器人, 机械多体系统, 并联运动学机器人等.