doi: 10.3969/j.issn.1006-1576.2011.04.031

自主 Robot 系统软件

赵永滨，骆云志
（中国兵器工业第 58 研究所 军品部，四川 绵阳 621000）

摘要：针对自主 Robot 软件设计的复杂性，提出模块化设计思路。介绍系统体系结构设计，设计感知系统、规划与决策系统的具体方案。结果表明，该设计能降低系统软件实现的复杂度、提高代码的复用率、节省系统开销。

关键词：自主 Robot; 系统软件; 并行设计
中图分类号：TP311.52 **文献标志码**：A

Autonomous Robot System Software

Zhao Yongbin, Luo Yunzhi
(Dept. of Armament Products, No. 58 Research Institute of China Ordnance Industries, Mianyang 621000, China)

Abstract: Aiming at the complexity of autonomous robot software design, the module design method is proposed. The system architecture design is introduced, and the detailed scheme of perception system and plan and decision system are designed. The experiments verify that the complexity of the system software realized is reduced, and the reusing rate of the code is improved, and the system cost is reduced.

Keywords: autonomous robot; system software; parallel design

0 引言

自主 Robot 是指：在室外环境中，Robot 按照预先给定的目标任务，结合环境感知信息做出路径规划，在移动过程中不断感知、判断周围的环境信息，自主地作出各种决策，随时调整自身的移动状态并执行相应的动作和操作。自主 Robot 集自动控制、人工智能、视觉计算、程序设计、组合导航、信息融合等众多技术于一体，是计算机科学、模式识别和智能控制技术高度发展的产物。20 世纪 80 年代以来，在新材料技术、新能源技术、信息和自动化技术等高新技术群的共同推动下，世界各主要国家纷纷加大了在军用自主 Robot 方面的研发力度。作为无人地面移动平台的重要组成部分，自主 Robot 可用来执行侦察、巡逻、物资运输、攻击、搜索、营救、探索、监视、科学数据采集等任务，能显著降低人员伤亡率，具有广泛的应用前景。因此，笔者对系统级自主 Robot 软件进行设计。

1 系统体系结构设计

Robot 的系统体系结构模块包括：感知（感觉和信息提取）、定位和地图生成（知识数据库、环境模型和局部地图）、规划与决策和路径规划（位置、全局地图和任务命令）以及动力控制（路径执行和避开障碍物）。

自主 Robot 的系统软件体系结构如图 1。整个系统可以看作以下信息的循环：

1) 感知系统从传感器接受数据，并通过对数据的分析和理解，构建一个精确的外部世界状态和 Robot 状态。

2) 规划与决策系统使用外部世界状态和 Robot 状态来执行其行为。

3) 规划与决策系统发送命令到 Robot 控制接口，在 Robot 中把那些信号转化成物理行为。

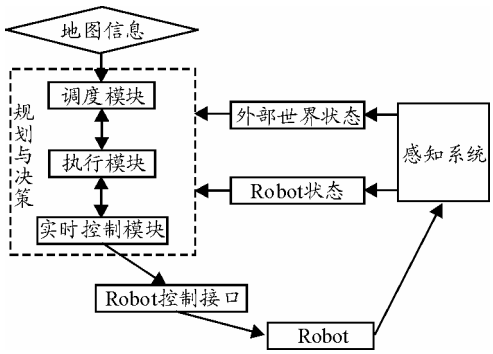

图 1 Robot 的系统软件体系结构图

2 软件设计

2.1 感知系统

感知是对由各种外部传感器提供的数据的分析。这些外部传感器包括：摄像头、激光扫描仪、声纳、GPS/INS、指南针等。感知系统主要由数据融合处理、跟踪和分类 3 个模块组成。

2.1.1 数据融合处理

数据融合是将来自不同传感器的数据进行融

收稿日期：2010-12-10；修回日期：2011-01-25
作者简介：赵永滨（1984—），男，四川人，硕士，从事数字信号处理研究。

合。融合后的结果会给出数据表达区域一个正确概括，为后续步骤提供准确的评估。

从环境感知的测量数据中提取特征。每一个特征都属于一个目标，一个特征包含的测量数据仅来自一个目标，一个目标可能有多个特征。

2.1.2 目标跟踪

目标跟踪是测量数据处理的核心。它把特征作为输入，产生“目标”，目标是虚拟的实体，被不断测量追踪。一个测量目标关联一个真实的目标。由于不断地测量追踪目标，每个目标的速度、加速度等动态参数就能被计算出来。越长的目标，越容易被跟踪，这些参数也越稳定。

运用重心跟踪算法实现目标跟踪。重心跟踪算法是基于跟踪目标重心的移动。这个算法是鲁棒、简单和可靠的，且需要的处理量少。但如果改变相关特征物的外形，速度计算不会精确。

利用跟踪方式计算目标的位置、相对速度和绝对速度。如果可以得到 Robot 的移动信息，那么就可以计算目标的绝对速度。为了达到好的效果，Robot 的运动必须高度精确。

2.1.3 分类

在目标跟踪期间，测量数据已经被转化成高级物体信息。在分类阶段，将进一步处理目标参数位置和速度。

分类算法使用多特征提取器和加权系统把所有目标分成 TRUCK、CAR、BORDER、LANE、SMALL 或 BIG。SMALL 和 BIG 两类收集不满足任何其他类的所有目标。例如：一个 BIG 目标分类的例子就是墙，而树或点属于 SMALL 物体。

2.2 规划与决策系统

规划与决策系统主要由调度器、执行器和实时控制器 3 个子系统组成。这些模块能在主或从模式下并行运行。上层模块负责全部的任务命令，而下层模块负责立即任务。这种设计把工程分解成更多的可管理的小任务块，优点是上层的模块能控制下层的模块。

2.2.1 调度器

调度器是最上层的模块，负责与整个任务相关的作业。它收到预先定义好的全局地图文件，计算出到达目的地的最优路径。

调度模块还负责将全局地图分解成更短的片段，叫路段（segment）。路段是由简单的 2 个点组成。通常是一条直线，交给执行器作进一步的处理。因为有不同类型的路段，所以调度模块必须从地图中推断出正确的路段类型。调度模块由 4 个主要部分组成，如图 2。

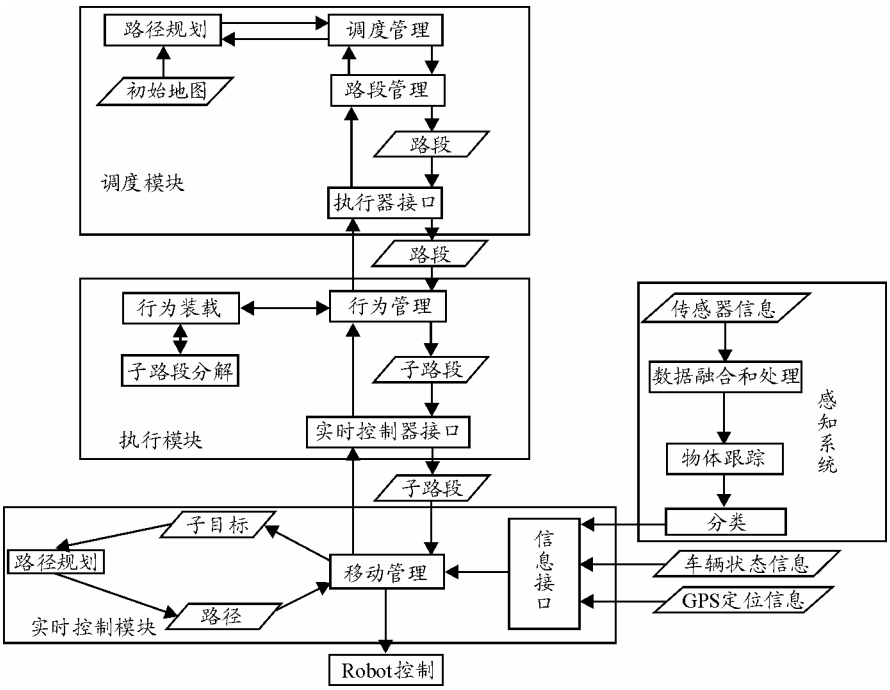

图 2 Robot 的系统状态流程图

调度管理子模块负责跟踪路径规划。它要确认路段是否被跟踪，当有新路段信号到达时，发布新的路段。

一旦调度管理子模块决定发布新的路段，就发

送一个信息到路段管理子模块。路段管理子模块负责推断路段的类型。表 1 给出了主要的路段类型的简单描述。

一旦正确的路段对象被创建，它就通过执行器接口发送到执行器。执行器接口是在执行器和调度器之间翻译信息的一个简单的封装接口，执行器接口定义如表 2。

表 1　路段类型表

路段类型	描述
Straight-line	直线移动
Right-turn	右转弯
Left-turn	左转弯

表 2　执行器接口表

信息	方向	描述
New-segment	调度器 → 执行器	调度器向执行器发送一个新的路段对象
Segment-complete	调度器 ← 执行器	Robot 通过一个路段，执行器等待执行下一个路段
Segment-impassible	调度器 ← 执行器	Robot 不能通过该路段

2.2.2　执行器

执行器是中间模块，负责导航 Robot 通过一个路段。它从调度器接受路段，并计算出一条穿过路段的最优路径。

执行器将把路段分解成更小的段，即子路段（sub-segment）。子路段由目标点和到实时控制器需进一步处理的方向组成。执行器必须充分地分解路段，让实时控制器不需要推断任何关于它的详细细节。基于调度器发送来的路段的类型，执行器必须执行一个正确的行为。最简单的情况是 straight-line。在这种情况下，执行器必须朝着终点在一条直线中创建子路段。比较复杂的路段是 U-turn 路段。在这种情况下，路段仍然是 2 个点，执行器必须推断出子路段的恰当的序列，这样 Robot 会运动到指定的终点。执行器会接收到来自实时控制器的信号，指示出它已经完成或不能完成分配的路径。在不能完成的情况下，执行器必须推断出，在当前的短路段中，是否能通过障碍物（例如，如果在当前的道路中有一个障碍物，它决定是否能从旁边通过障碍物）。如果能通过，执行器必须下载一个行为来通过障碍物。如果不能通过，执行器必须发送信号给调度器，调度器将在受阻位置重新计算一条完整的任务路径。否则，当穿过指定路段时，执行器就发送一个信号给调度器，等待进一步的命令。执行器由 4 个主要的部分组成，如图 2。

如上规定，不同类型的行为绑定了不同类型的路段。这些行为被装载到行为管理子模块，行为管理子模块扮演着行为关联的角色。每个行为处理不同的路段。例如，straight-line 行为相对 U-turn 行为来说是不同的。因此，很多在行为管理子模块中的模块担当工具的作用，来帮助当前的执行行为。

子路段分解子模块专用于行为类型。但它更常用的是从已经确定了行为的父亲路段建立子路段。

执行模块是 3 个部分中最复杂的模块。因此，抽象行为的设计允许其它操作并行运行。

一旦行为管理子模块决定发布一个新的子路段，它就发送子路段对象到实时控制器接口。实时控制器接口将依次发送信息到实时控制器，实时控制器接口定义如表 3。

表 3　实时控制器接口表

信息	方向	描述
New sub-segment	执行器 → 实时控制器	发送新的子路段对象
Sub-segment complete	执行器 ← 实时控制器	Robot 通过一个子路段，实时控制器等待执行下一个子路段
Sub-segment impassable	执行器 ← 实时控制器	Robot 不能通过该路段
Emergency stop	执行器 ← 实时控制器	意外障碍物被探测到，Robot 紧急停止

2.2.3　实时控制器

实时控制器是系统中最底层的处理器。它直接和 Robot 的控制系统相连，是最终的控制判别器。故实时控制系统的主要职责是确保 Robot 的安全。

实时控制器从执行器接受子路段。子路段被定义为 Robot 的目标矢量和目标方向。此外，实时控制器负责当子路段已经被通过时或当它探测到一个障碍物不能被避开时，通知执行器。

实时控制器必须连续监视感知输入。如果一个障碍物被探测到，实时控制器必须让 Robot 停止，并通知执行器。而且，它必须确保 Robot 是在正确道路的中心。

实时控制器可以在任何时候从执行器接受新的子路段。当一个新的子路段被接受后，实时控制器必须规划出一条平稳的路径，按正确的方向到达目的地。规划出的路径必须使 Robot 处于正确的道路中，且能避开其它的一些障碍物。

实时控制器不需要推断任何关于子路段的信息。所有的推导都在执行器和调度器预先计算出来了。所以，实时控制器需要具备 2 种能力：1) 如何导航到目的地；2) 如何处理障碍物。实时控制器把

资源用于确保安全、成功地通过子路段。实时控制器的主要组成如图2。

表4 实时控制器控制参数

控制输出	描述
Set velocity	设置机器人的速度
Set wheel turn	设置机器人的方向

运动管理子模块负责跟踪给出的子路段的进展。它连续监视并及时感知周围（由道路感知接口产生），确保Robot处于正确的道路中，使它不撞击障碍物。当一个障碍物被探测到，或当它感知到Robot不在道路中心时，它将使用路径规划器建立一个更新的路径。如果规划器不能建立一条路径，移动管理子模块将发一个信号给执行器，告知不能通过子路段。实时控制器发送信号给Robot控制器，实时控制器的主要控制输出定义如表4。Robot控制器还有一个和仿真器相连的通讯接口实现静态或动态仿真调试。

3 结束语

该设计采用模块化并行运行模式，提高了系统的运行效率。同时，将Robot行为抽象化，实现地图的软件分割，大大减小了系统运行的复杂度，具有一定的实际使用价值。

参考文献：

[1] Thomas C.Henderson. Robust Autonomous Vehicles. university of Utah.2007
[2] Martin Dittmer.Team-LUX.2007
[3] 丁广帅, 雷运洪. 基于机器视觉区域处理技术的水中机器人定位方法[J]. 兵工自动化, 2010, 29(11): 74-78.

****

（上接第92页）

AEC429A_SetTriggerDepth(hCard, i, len-1);

2) 开放中断

AEC429A_SetIntMaskReg(hCard, TRUE);

3) 设置中断触发事件

使用标准API函数CreateEvent[4]，中断事件句柄为hevent。程序代码为：

```
HANDLE hevent=NULL;
hevent=::CreateEvent(NULL, FALSE, FALSE,
NULL);
AEC429A_SetEvent(hCard, hevent));
```

4) 创建接收线程

使用标准API函数CreateThread，接收线程函数为ThreadFunc。程序代码为：

```
hThread=::CreateThread(NULL, 0, ThreadFunc,
(LPVOID)ghWnd, 0, 0);
```

5) 在线程中判断是否产生中断

标准API函数WaitForSingleObject。

```
WaitForSingleObject(hevent, INFINITE);
```

6) 接收数据

使用函数AEC429A_RxRead。

7) 数据分析

根据接收到数据的30～31位，判断发送器硬件状态是否正常，若正常，再根据1～8位即Label识别信息含义[5]，作出相应的数据处理，同时更新显示内容；若硬件异常则数据无效，给出异常提示。

2.3 发送数据过程

发送数据采用非定时发送方式，当按下命令按钮时，即发送对应相机的控制字、状态字等模拟数据。先向发送缓冲区 dwTxBuf[j]中写数据，再用AEC429A_TxFIFOSTR判断发送FIFO状态，若为不满状态，则使用AEC429A_TxWrite发送数据。

2.4 编程应注意的问题

在编程时应注意采用模块化设计，将完成某一功能的若干语句放在一个自定义函数内，以增强程序的可读性，方便调用，简化程序，便于调试；在调用板卡打开、复位、关闭、设置触发深度等函数时要对返回值判断，若调用成功，继续进行，若失败给出提示退出程序；另外，在程序退出前应使用AEC429A_Close (hCard)关闭板卡，释放板卡资源。

3 结论

经过部队一年多的使用证明：该软件运行可靠，操作简单，完全可以同时检测多台（种）航空相机，大大提高了检测效率。同时，由于通讯程序采用模块化、通用性设计，开发其它控制系统或检查仪时，只要ARINC429总线接口卡的型号不变，就无需更改通讯程序。

参考文献：

[1] 王勇, 于宏坤. 机载计算机系统[M]. 北京: 国防工业出版社, 2008: 94-97.
[2] 范秀英, 范鹏飞, 谷峰, 等. 基于ARINC429总线接口卡的航空相机综合控制系统[J]. 兵工自动化, 2010, 29(1): 72-73.
[3] AEC429-PCI/CPCI 用户手册[Z]. 北京: 北京神州飞航科技有限责任公司, 2008.
[4] 朱友芹. 新编Windows API参考大全[M]. 北京: 电子工业出版社, 2001: 745-746.
[5] ARINC Protocol Tutorial.ARINC429 Tutorial[Z/OL]. Condor Engineering, 2002. http:\www.condoreng.com