基于 Agent 的核心计算机操作机制研究

朱志强[1],王建元[2],王 芳[2]
(1.西北工业大学 自动化学院,陕西 西安 710072;
2.飞行自动控制研究所,陕西 西安 710065)

摘 要:无人作战平台(UCAV)核心计算机(Core Computer)是一个并行多处理系统,由多个处理器和功能单元组成,负责平台的智能思维、推理和决策。文中基于智能体(Agent)的理论与方法建立了核心计算机智能体模型和控制结构;定义了不同智能体的功能和操作;对智能体间数据交换、通讯等操作机制进行了探讨。

关键词:核心计算机;智能体;神经网络;操作机制;无人作战平台
中图分类号:TP18 **文献标识码**:A **文章编号**:1673-629X(2007)07-0008-05

Agent-Based Research of Operational Mechanism for Core Computer

ZHU Zhi-qiang[1], WANG Jian-yuan[2], WANG Fang[2]
(1. College of Automation, Northwestern Polytechnical University, Xi'an 710072, China;
2. Flight Automatic Control Research Institute, Xi'an 710065, China)

Abstract: The core computer of Unmanned Combat Aerial Vehicles(UCAV) composed of multiprocessors and functional units is a parallel and intelligent system that can be used to think, reason and make decisions. Based on agent theory and methods, this paper proposes hierarchy and agent model of core computer. The behavior and operations of agents are also abstracted and defined. Moreover, the mechanism of data exchange, communication and cooperation between agents are discussed.

Key words: core computer; agent; neural network; operational mechanism; UCAV

0 引 言

核心计算机(Core Computer)是无人作战平台(UCAV)的核心处理单元,主要完成无人作战平台组织层(MMS)的工作,负责平台组织层任务的接受、分解、策划、推理和决策。核心计算机的工作结果是生成一组控制指令或任务序列传输给相应的功能系统并控制和指挥这些系统完成既定的动作和任务。核心计算机[1]强大的处理能力由其内部多个功能单元共同承担和完成。每一功能单元按照各自的功能特点承担不同的任务。如:DSP 功能单元主要承担信号处理和数值计算;专用硬件单元承担诸如模糊推理、神经元计算等特殊处理功能;综合单元负责常规的控制和运算等[2]。不同功能单元依据各自的操作机制和算法有序地执行。核心计算机可通过通讯链接受地面站或长机的控制指令和任务指令;通过传感器感知外界的变化,并据此做出相应的处理。当通讯链中断后核心计算机将依据先验知识自主地进行规划和推理以完成平台下一阶段的控制和任务。纵观 UCAV 的作战特征可以看到:其核心计算机具有的紧急情况处理(反应性)、中短期任务的策划和规划(推理规划)、外界感知(自发性)、通讯和协同(社会性)等能力都具有鲜明的智能体(Agent)特征。所以,抽象建立核心计算机的智能体模型,利用智能体理论和方法研究和建立核心计算机的操作机制成为研究工作的出发点。在核心计算机智能体的框架下,核心计算机不同的功能单元可看作不同的子智能体,核心计算机本身则构成一个多智能体(Multi-Agent)系统和环境。显然,把观察的视角从核心计算机扩展到 UCAV 再扩展到多 UCAV,智能体的概念和范围也随之扩展。这意味着智能体的理论和方法可更广泛地用于 UCAV 的研究工作中。

1 内部智能体结构

核心计算机智能体控制结构描述了核心计算机内部各智能体间的控制关系以及与无人作战平台其它智能体(子系统或飞行器)间的交连、通讯和控制关系。

收稿日期:2006-10-13
基金项目:航空基金资助项目(04F18002)
作者简介:朱志强(1963-),男,陕西西安人,博士研究生,研究领域为并行计算及容错技术。

无人作战平台核心计算机的功能层可分为六个智能体,由六个不同的处理器或智能单元承担(见图 1),分别负责任务与形势感知、建模、在线规划、通讯与协作、决策与管理、任务控制共六项功能。其中,任务与形势感知智能体负责平台与地面站通讯链和各种任务和状态传感器信息的处理,又称输入信息处理智能体;任务控制智能体又称输出信息处理智能体,它负责把核心计算机的工作结果以命令或任务序列的方式合理、无矛盾冲突地下达到不同的功能子系统;通讯与协作智能体负责与各平台子系统以及其它飞行平台间的信息通讯、数据交换、任务协同等;建模、在线规划、决策与管理三个智能体负责核心计算机的智能思维、推理和决策。

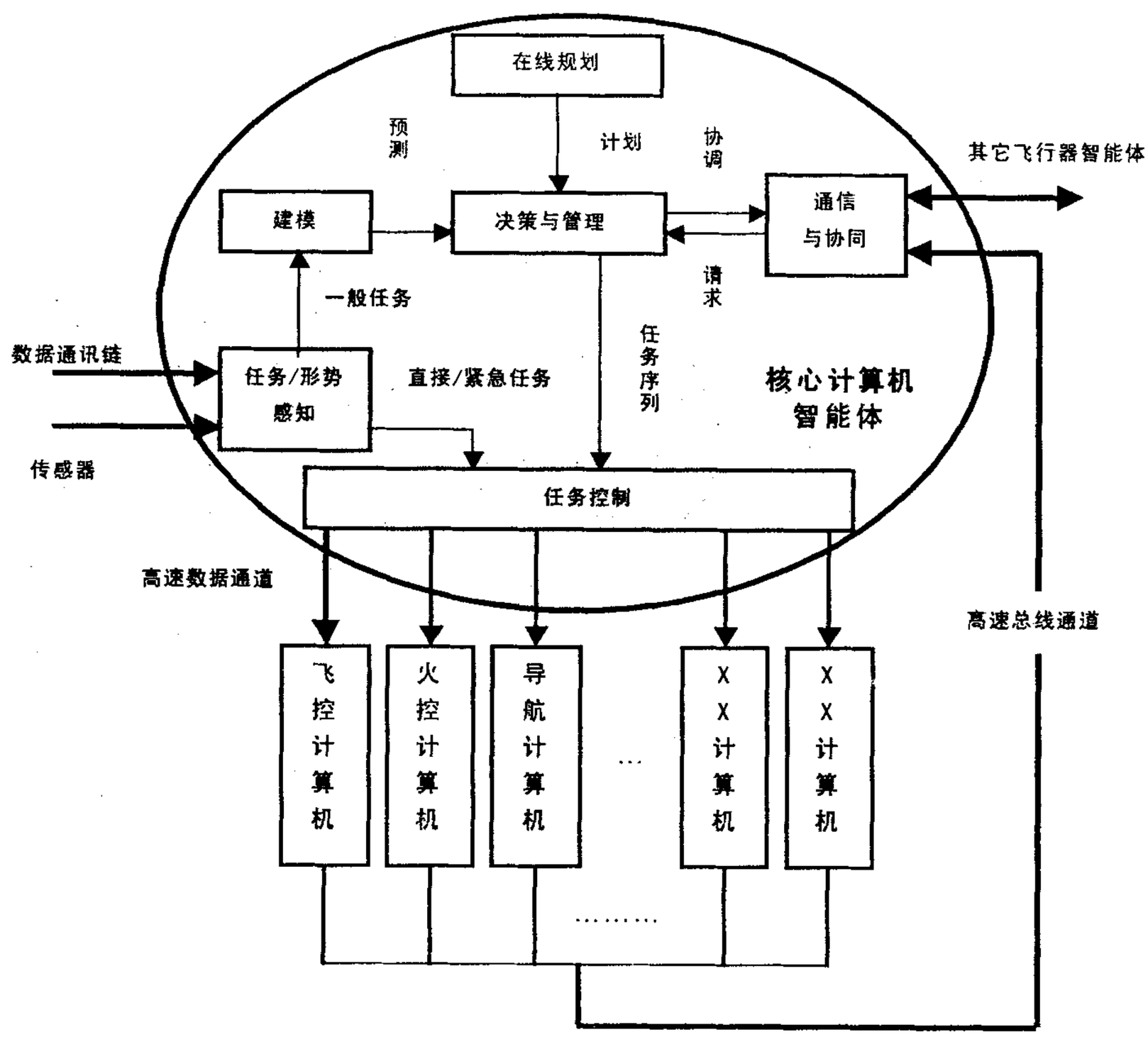

图 1　物理层控制结构图

2　智能体行为机制

2.1　任务与形势感知智能体

任务与形势感知智能体主要负责对任务链指令和各级各类传感器指令进行分析;对紧急类指令进行立即反应和处理;对常识类指令和常规传感器状态通过常识类知识库及一定的推理生成动作指令序列;对一般任务和传感器状态移交到建模与预测智能体进行处理。

2.1.1　任务感知

通过无线通讯链路接受地面站的控制、指挥和任务指令信息,并据此作出分析和响应。接收的控制信息可分为四类:

(1)直接命令信息。以命令方式对飞行器及任务系统进行直接控制,如:平飞、投弹、照相、返航、着陆对准等。

(2) 指令序列信息。指令序列是直接命令信息的批处理形式,如:指令序列 1 = {起飞、*到达目的地*、投弹、返航};指令序列 2 = {*1 号目标侦察*、照相、返航};

注意到指令序列中一部分是直接明确的控制命令,另一部分是一些抽象笼统的指令(斜体),这些抽象的指令可能由多个明确单一的动作组成,也可能其中有些动作具有不确定性(动态性),因此只能通过反馈、感知及推理来确定。另外,通过定义抽象指令集合;进行单指令动作分解;设计不确定动作的推理算法;建立单指令容斥关系模型和判别方法等来界定指令容斥关系可有效避免指令冲突和指令冗余。

(3) 动态数据或程序变更与加载。数据和程序的动态变更主要用于对装订数据进行修改;对数据库、知识库进行更新和维护或对旧的算法程序和控制方式进行升级和优化。

2.1.2　形势感知

根据接受的数据和多传感器信息融合的结果对无人平台的环境态势和威胁进行评估,评估结果分为紧急事件和一般事件。紧急事件指空中格斗、逃逸、躲避威胁、快速还击等紧急情况,一般事件指不需立即反应的事件。为保障飞行器对一些危险和紧急状况作出及时正确的反应,一些关键的传感器信息如:雷达、飞控、敌我识别等将直接纳入智能体刺激性神经反应链中,以便得到及时正确的反应和应急处理,处理方法一般都直接调用应急轨迹库和应急任务库。对一般的形势感知信息经融合处理后传输给建模智能体进行相应的环境数据库进行修正、更新和对象建模[3]。

智能体对外界任务和形势感知的直接响应(刺激响应)可用一组简单的产生式规则来表示,写作 $c_i \rightarrow a_i$,其中 c_i 为布尔逻辑条件表达式,a_i 为动作表达式,它可以是一个动作、任务代码、任务集合,也可以调用另一套产生式系统。整个产生式序列集合为:

$c_1 \rightarrow a_1$

$c_2 \rightarrow a_2$

……

$c_m \rightarrow a_m$

智能体每接收一组任务和形势感知信息，将按以下算法遍历所有产生式系统获得直接(刺激)响应的动作或任务编码。

算法(产生式系统遍历)

输入:所有感知信息$\langle x_1, x_2, x_3, \cdots, x_n \rangle$

输出:动作/任务编码集合 TaskSet〈 〉

```
Begin
    TaskSet = nil;/* 置任务集合为空 */
    c1 = f1(x1, x2,…,xn);……
    cm = fm(x1,x2,…,xn);/* 逻辑条件计算 */
    For i = 1 to m
    {
      IF ci then
      case ai = 任务编码:
      TaskSet = TaskSet + 〈ai 任务码〉;
      case ai = 动作集合:
      While
          执行动作集合;
      until ai = 任务编码;
      case ai = 产生式:
      While
          执行产生式系统;
      until ai = 任务编码;
    }
    Return(TaskSet); /* 返回任务集合 */
End
```

2.2 建模智能体

无人飞机在飞抵某一区域执行任务前需装订或建立该区域的环境模型(数据库)来描述该区域的地理地貌、任务目标、危险点等信息。这些信息不一定很完整和确定，其中相当一部分需在飞行过程中动态进行感知、建立、修正、更改或补充。如:周围其它智能体的位置、特性、意图等需动态感知、动态修正;一些目标信息需确认或更改;雷达、防空火力、恶劣气候等威胁源需建模或对原模型进行修正。该智能体通过感知对原始环境模型进行不断的完善、修正和补充，并根据模型信息对当前情况进行预测，提出当前的行动建议提交给决策模块。该智能体的操作对象是装订到核心计算机中的数据库和相关数字地图。对数字地图的具体操作包括:把目标、威胁源等叠加到数字地图上;对不在网格上的信息点进行插值拟合等。建模智能体的操作原理如图 2 所示。

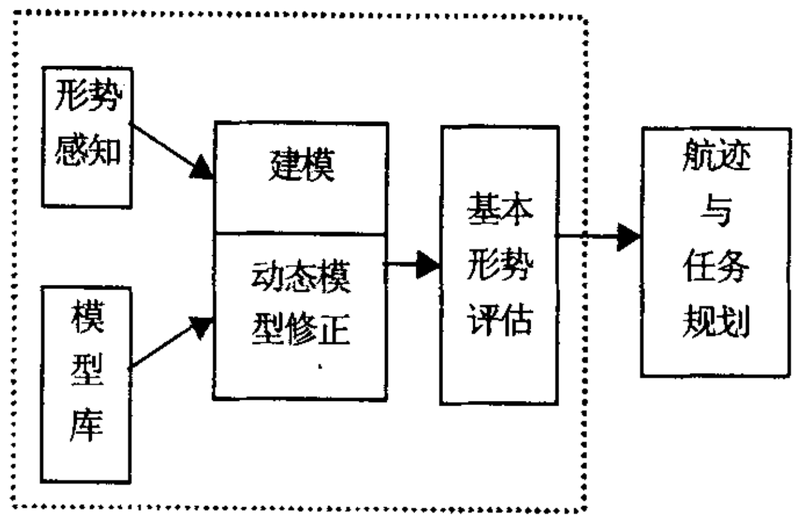

图 2 建模智能体操作机制

2.3 在线规划智能体

该智能体围绕无人机任务目标的实现，根据以往的经验及当前环境模型进行航迹规划和中短期任务规划，规划结果是一组供决策智能体使用的航迹和任务序列。由于环境和战场态势的动态性，这组序列可能未被执行完就被刷新或修正。制定中短期计划的依据有三个:一是按优先级排列的目标集合，它是规划算法的收敛条件;二是环境模型信息，它是指定计划的约束条件;三是经验库，它是一组范例的集合，由条件、规划方案和评价结论组成，可供规划算法借鉴使用。当没有合适的经验可借鉴时，只能通过算法产生新的规划方案。若该新方案得到成功的执行，则可经在线评估后纳入经验库中。该智能体操作原理如图 3 所示。在线规划智能体由规划算法控制，根据多个规划目标要求依次进行规划工作[4]。规划算法首先从经验库中寻找与目标要求相似的规划，并据此推出相应的结果，否则启动模型预测算法进行规划。满足不同规划目标的多个规划结果不断输出至决策与管理智能体供其选用。同时，决策与管理智能体将综合各类信息，并提出新的规划目标返回给规划智能体。

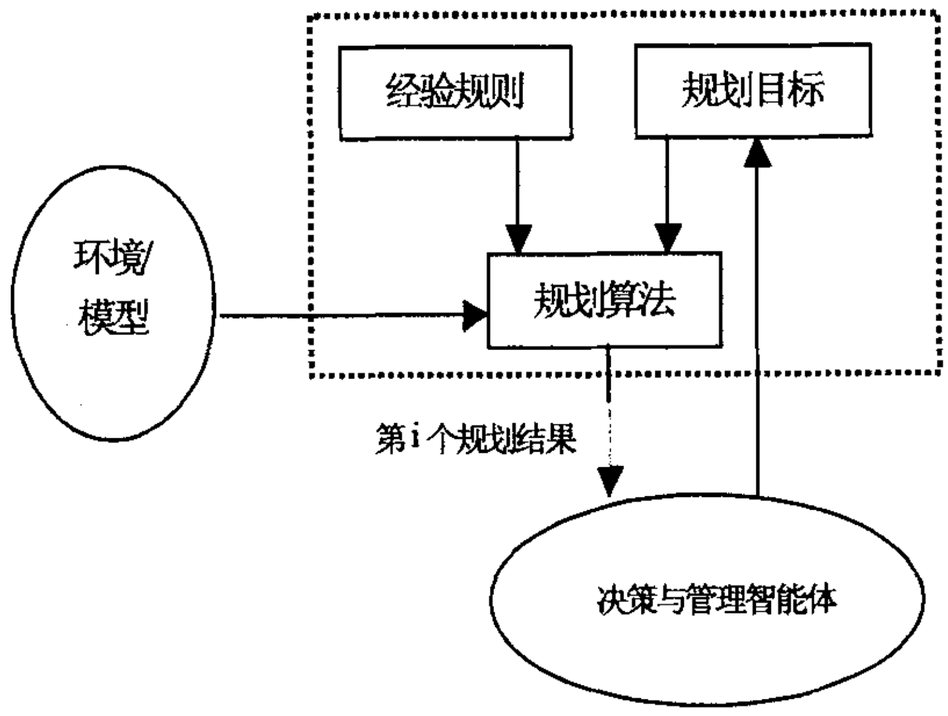

图 3 规划智能体操作机制

2.4 决策与管理智能体

该智能体接受的信息包括:中短期航迹、任务指令序列、预测行动建议、其它智能体的反馈与请求信息。主要任务是按所接受信息的优先级进行处理和发布;

剔除不合理或不现实的计划、建议和要求;消解矛盾和冲突的任务和信息;优化和折中产生最终行动计划的动作序列;制定下一阶段的规划目标,操作原理如图 4 所示。智能体首先对信息缓冲区指令序列的优先级进行判断,然后对优先级最高的序列进行冲突检测和消解,在进行单指令分解后发送至不同的子系统,同时,对输出序列进行评估后制定新的规划目标。

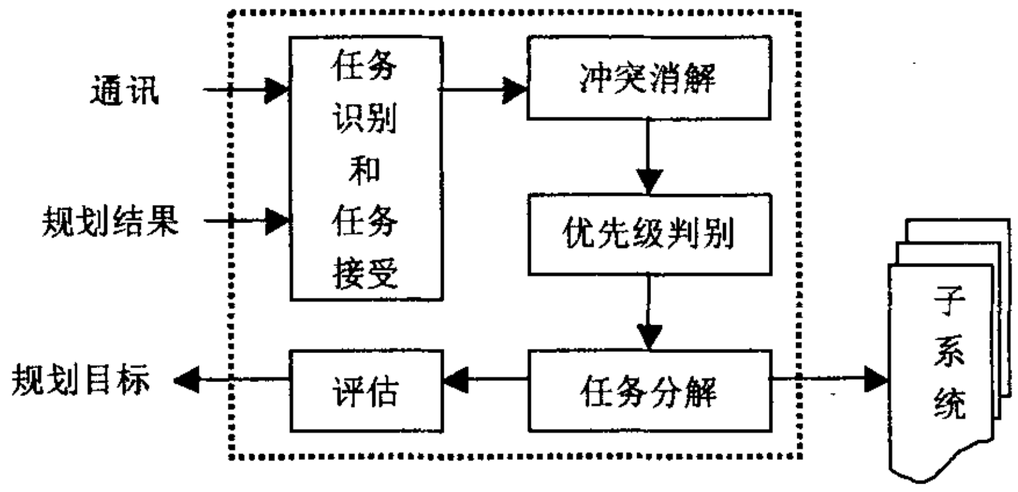

图 4　决策智能体操作机制

2.5　通讯与协同智能体

该智能体是核心计算机与本地(Local)智能体和远程(Remote)智能体进行交流的中心。与该智能体进行通讯的对象主要包括两类:一类是本飞行器中除核心计算机外的其它任务或控制系统的智能体,另一类是其它飞行器(机群作战)智能体。通讯的物理通道有高速总线和无线通讯接收通道。美国 ARPA 知识共享计划(knowledge sharing effort)中提出的 KQML (Knowledge Query and Manipulation Language)语言定义了智能体间信息传递的标准语法和动作表达式。参考 KQML 规范定义并针对航空领域自主无人飞行器的信息通讯内容、范围和特点,笔者参与设计了用于本领域的智能体通讯交互语言 UAVACL(Unmanned Airborne Vehicle Agent Communication Language)。

多智能体间使用无人机智能体交互语言 UAVACL 进行信息交流的主要任务是对发来的信息包进行语法、语义分析和语言理解,把其中的任务请求信息传给决策模块,对状态信息进行记录并传给建模模块供环境模型维护之用,对其它的询问信息以 UAVACL 语言进行应答。根据其"一对多"的特点,可把该智能体作为一个通讯服务器并以网络 Client—Server 的方式进行工作。其它智能体以 Client 的身份在初始启动时通过 UAVACL 语言向服务器(Server)进行登记。这样,在后续应用中服务器将保存各 Client 智能体的动态信息,并可随时与它们进行交流。智能体通讯模型如图 5 所示。

UAVACL 的语法规范可用巴柯斯·诺尔(BNF)范式进行表示。通讯方式目前采用 1553B, Arinc429, RS422 和内存直接操作方式,通讯关键字分别为:

·M1553B: 通讯方式采用 MIL - STD - 1553 总线;

·AR429: 通讯方式采用 ARINC429 总线;

·RS422: 通讯方式采用 RS422 总线;

·MEM: 通讯方式采用直接内存控制方式;

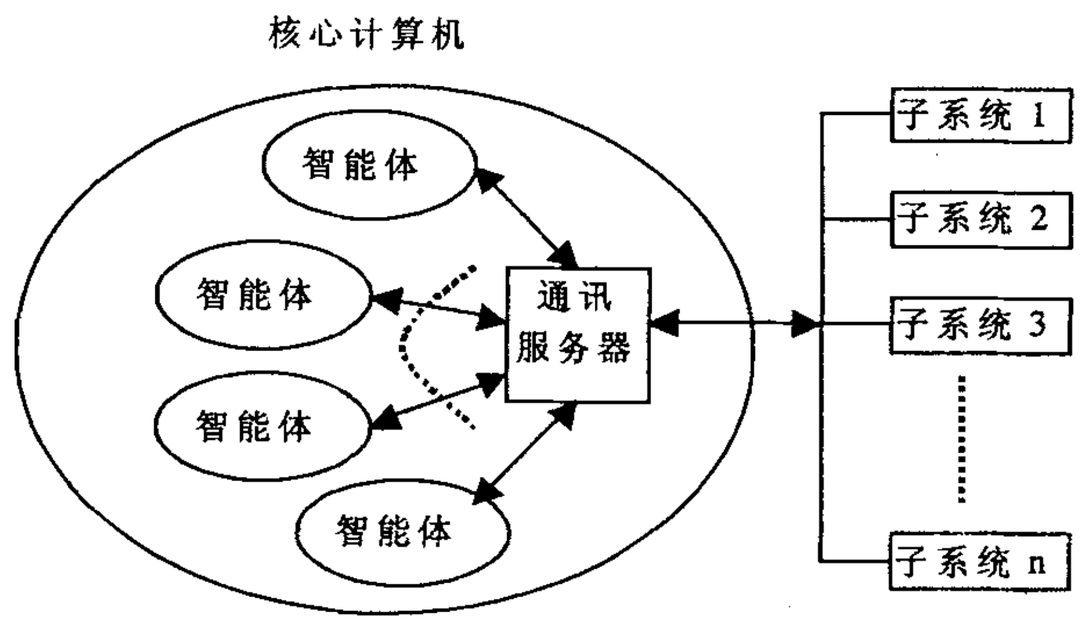

图 5　智能体通讯模型

通讯命令格式如下:

<通讯动作>

:<操作字 1><智能体名称>//<通讯关键字>:

<通讯格式>:<通道号/内存地址>//

:<操作字 2>……

:<操作字 n>……

UAVACL 的部分动作如下:

·Login:登录通讯服务器并留下自己的名称、通讯格式号、通道号或内存地址;

·Logoff:退出通讯服务器并取消有关挡案;

·Ask:询问有关内容;

·Tell:告知有关内容;

UAVACL 的部分操作字如下:

·Sender:发送者;

·Receiver:接受者;

·Package:发送包内容;

·Messgae:文字消息;

·State:工作状态信息;

例:

Login(火控系统通过第 2 个 1553 总线通道以格式 1 方式登录服务器)

:Package "* * * * * * * * * * * * * * * *"

:Sender <火控>

:Receiver <服务器>//M1553:1,2//

Ask(服务器通过第 3 通道 429 总线以格式 2 的方式询问大气机状态)

:State?

:Sender <服务器>

:Receiver <大气机>//Ar429:2,3//

Tell(大气机通过第 4 通道 429 总线以格式 2 的方

式告知服务器其状态)

:State“* * * * * * * * * *”

:Sender <大气机>//Ar429:2,4//

:Receiver <服务器>

2.6 任务控制智能体

该智能体对任务序列进行分解并把任务分配给对应的执行者。对于一些复杂任务,智能体还将对执行者的能力和所需资源进行评估,确定由一个或多个执行者来承担。具体操作分以下二种情况:

(1) 查询任务分配表,若判定该任务由其它子系统承担,则以数据包形式通过高速通道(如:双口RAM/高速总线)把任务派发给相应的子系统。数据包中应包括任务的名称、优先级等,子系统执行结果通过高速通道回馈到核心计算机。

(2) 查询任务分配表,若任务需由多个执行者完成,则根据任务要求的执行时间评估该任务需由几个处理器来承担,若只需一个处理器,则在处理器群中指定一处理器,通过动态加载把任务加载到本地存储器中完成任务执行。若需多个处理器承担任务,则需通过“任务分配和通讯开销”均衡算法把任务分到不同的处理器上动态加载运行。

3 智能体数据交换机制

核心计算机智能体间使用共享内存方式进行数据交换,这块共享内存通常被称为“黑板”。每一智能体都把“黑板”作为自己的私有资源进行数据的存取和操作,使得同一个数据就能被所有智能体共享,从而达到数据交换的目的,当然这里要解决和避免内存操作冲突。显然,若“黑板”机制物理上采用非易失存储器(Non-Volatile)则可作为知识库和数据库的载体使用。核心计算机与外部智能体间的信息交换则通过“邮箱”方式进行。“邮箱”是一种双口 RAM 的机制,核心计算机每一智能体均有自己的“邮箱”。“信件”的分拣由核心计算机的通讯智能体负责。所以,核心计算机智能体可通过查询“黑板”和“邮箱”内容来获取内部和外部的信息,当然也可通过它们来发送信息。

4 工作与展望

在核心计算机硬件架构建立以后,如何建立其高效和强有力的操作机制成为决定核心计算机智能水平和处理能力的关键。智能体的概念较好地表达了核心计算机的智能和并行特征,为核心计算机操作机制研究提供了理论基础,在智能体的理论框架下,文中仅对核心计算机的控制结构和各种功能进行了定义和描述,要建立核心计算机完善的操作机制有大量的开拓性工作要做,其中一些重要的基础性工作包括:

(1)各智能体执行机的设计与验证;

(2)智能体学习与成长机制的建立;

(3)智能体间的任务协同与并行计算技术研究;

(4)智能体动态任务重构技术研究;

(5)数据库、知识库的构造与动态维护。

在完整的操作机制建立后,核心计算机可具备半自主、有限权限自主甚至全自主的运行能力[5]。核心计算机智能和能力提升的途径主要通过神经网络的学习和成长、先进人工智能算法的更新、数据库和知识库等的扩充来完成,是逐步积累和完善的过程。当然,人工智能和智能计算机研究领域的每一点进展和突破都将为核心计算机的研究和发展带来新的曙光。

参考文献:

[1] 陈国良,吴俊敏,章 锋.并行计算机体系结构[M].北京:高等教育出版社,2002.

[2] Jordan H F. Fundamentals of Parallel Processing[M].北京:清华大学出版社,2003.

[3] Meystel A M. Intelligent Systems: Architecture, Design and Control[M].北京:电子工业出版社,2003.

[4] Hamid N, Earl B. Creating an adaptive embedded system by applying multi-agent techniques to reconfigurable hardware [J/OL]. 2004. www.sciencedirect.com.

[5] Culler D E. Parallel Computer Architecture: A Hardware/Software Approach[M].北京:机械工业出版社,2003.

(上接第 3 页)

-7[M]. New York, NY, USA: John Wiley & Sons, Inc, 2002.

[2] Bouthemy P, Gelgon M, Ganansia F. A unified approach to shot change detection and camera motion characterization[J]. IEEE Transaction on Circuits and Systems for Video Technology, 1999, 9: 1030-1044.

[3] Rath G B, Makur A. Iterative least squares and compression based estimations for a four-parameter linear global motion model and global motion compensation[J]. IEEE Transactions on Circuits & Systems for Video Technology, 1999, 9: 1075-1099.

[4] Tan Y P, Kulkarni S R, Ramadge P J. A new method for camera motion parameter estimation[C]// In Proceedings of International Conference Image Processing. [s. l.]: IEEE Computer Society, 1995: 405-408.

[5] Boser B, Guyon I, Vapnik V. A training algorithm for optimal margin classifiers[C]// In: Haussler D, ed. Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory. New York, NY: ACM Press, 1992: 144-152.