

2020中国机器人大赛比赛规则

家庭机器人仿真项目

2020中国机器人大赛家庭机器人仿真项目技术委员会

2020年9月15日

目录

一、项目简介.....	3
二、技术委员会.....	4
三、赛项说明.....	5
3.1 参赛要求.....	5
3.2 赛项说明.....	5
3.3 场景描述.....	5
3.4 机器人物理行动.....	7
3.5 自然语言描述简介.....	9
四、比赛场地及器材.....	11
4.1 比赛场地.....	11
4.2 比赛器材.....	11
五、评分标准.....	12

一、项目简介

家庭服务机器人仿真比赛立足于面向室内环境的服务机器人高层功能的探索，主要包括人机交互、自动规划、推理、环境感知和重新规划。为此，将家庭机器人抽象为仿真机器人，并以仿真的室内环境为测试环境，将人机交互抽象为自然语言或命令语言表达的任务描述和人机对话，将机器人感知数据抽象为文件格式的场景描述、观察反馈和执行反馈。

家庭机器人仿真比赛针对自主机器人在家庭环境中的典型应用来设置一系列场景，以测试参赛程序的性能。具体说来，一开始仅被告知部分的环境初始状态和需完成的任务，参赛程序可以通过人机交互（向平台提问）或观察环境的行动以获取缺乏的信息，以便规划出行动序列来完成任务。但在比赛的第二阶段，初始状态信息和人机交互结果有可能是错误的，从而出现规划结果中某些行动无法执行的情况。此时，参赛程序会获得平台反馈的失败结果，并据此进行重新规划。其中，人机交互过程分别用指令语言和自然语言表达，从而构成本赛事的两个不同的比赛项目——指令语言交互项目和自然语言交互项目。

比赛基于一个3D仿真机器人，它有一组固定的原子行动(primitive actions)，包括九种物理行动、人机交互行动和观察行动，对所有问题都不变。本次仿真比赛采用的仿真机器人如图1.1所示，考虑基于其功能的机器人问题求解。此机器人有两个轮子（可以移动），手臂上有一个手爪（一次只能抓取一个物品），和一个盘子（上面只能放一个物品），体现基本的移动、抓取、放下、开门、关门等能力。同时，机器人可以向环境中的用户提问，实现人机交互，也可以观察自身周围环境。在这些功能基础上，测试机器人的人机交互、自动规划、推理、观察、重新规划等能力。家庭机器人仿真比赛要求参赛程序对比赛平台提供的每一个场景，根据其初始状态描述和任务描述，在规定时间内，通过人机交互或观察行动获取必要的环境信息，自动规划出物理行动序列以实现目标，并根据行动反馈，调整规划结果，保证最终完成任务。比赛平台将根据参赛程序在此

过程中的整体性能打分，并根据一个阶段中所有问题的总分决定参赛程序在该阶段的排名。

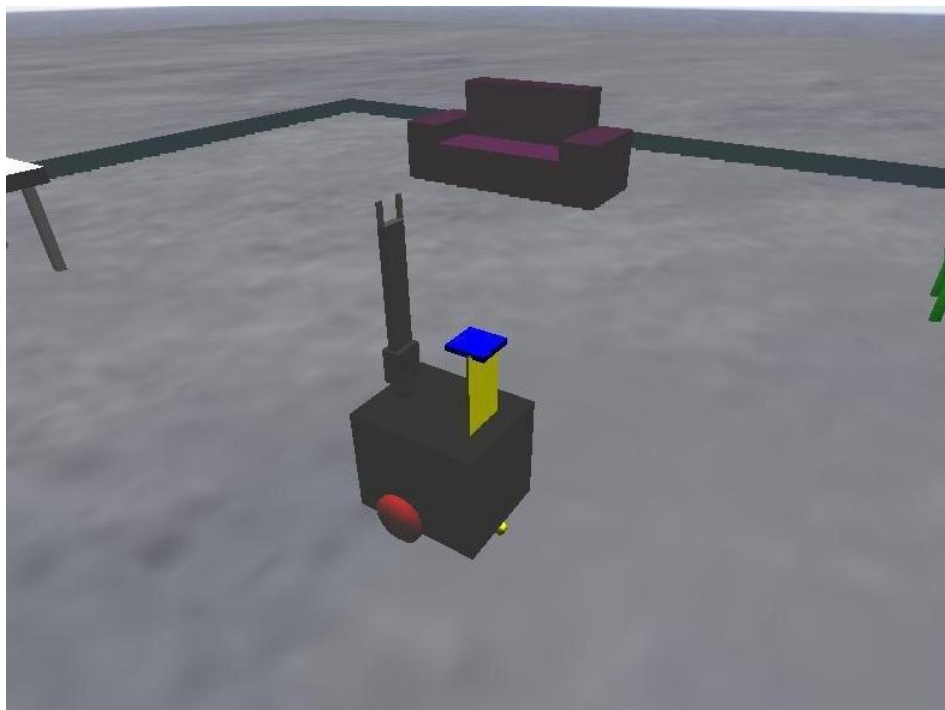


图 1.1: 机器人模型

二、技术委员会

负责人：刘璇 上海大学

成 员：

三、赛项说明

3.1 参赛要求

每支队伍将仿真项目工程文件拷贝到赛场专用竞赛电脑上，由操作手进行统一编译完成，之后在仿真平台上运行。

3.2 赛项说明

比赛过程中，机器人可选的物理行动、人机交互行动和观察行动是固定的。其中，物理行动包括移动、抓取、放下等机器人基本动作，每执行一个物理行动，平台会反馈其执行结果（成功或失败）。人机交互行动用来模拟机器人向用户提问，目前只有一个行动，问某具体物体所在的位置。在比赛的第二阶段中，平台根据一定概率，有三种可能的回答——正确位置，错误位置和不知道，并且对同一个问题，始终只有一种特定的回答。观察行动用来模拟机器人通过传感器对自身位置附近的观察，假设观察行动能够正确的告诉机器人当前所在位置的所有物体（不包括在其他物体里面并且门没有开的“小物体”）。

每个测试都基于一个完整的环境场景。一开始，参赛程序将获得环境的部分初始状态，包括环境中出现的所有物体，环境中所有大物体的位置信息，机器人状态信息，和部分小物体位置信息。其中，小物体位置信息有一定概率会出错。同时，参赛程序也被告知需要完成的多个任务。其中，完成任务所需的一些位置信息，可能事先不知道或有错误，参赛程序可以通过人机交互行动或观察行动获取相关信息，也可以盲目搜索整个环境。

参赛程序每执行一个行动，平台将反馈执行结果。目前，有若干种情况下行动会执行失败，例如，当机器人要抓取物体，而不在当前位置时，此行动执行失败，此时，参赛程序可以通过人机交互行动或去其他位置观察来获取的位置，重新规划，以最终完成所有任务。

3.3 场景描述

场景描述规定当前机器人所面临的环境状态，即当前环境中出现的物体及其属性，包括物体的种类、位置和其他属性，以及机器人的状态等。

此次仿真比赛中，场景描述的内容包括：

- 场景大小：在整个比赛中保持不变。

· 场景中物体描述：用编号表示不同的物体，对每个编号有相应的描述，包括物体的种类、位置和其他属性。这些描述都是可选的，有些物体可能没有完整的描述。其中有一个特殊的物体——人(sort:human)，代表用户。比赛提供的所有场景中，有且仅有一个（以下称为用户），并且在每一个任务的求解和执行过程中其状态不会改变。其他物体的具体描述内容包括：

– sort: 物体的种类，例如：book, cup, bottle, table, chair, microwave等。
– color: 物体的颜色：white, red, green, yellow, blue, black。

– size: 物体的类型：big, small。small和big用来区分物体是否可以被机器人移动，据此物体可以分为两类：

* “大物体”：机器人无法移动的物体，由于没有外部影响，它们的位置和状态在一个给定的问题中不会改变。“大物体”包括“小物体”可以放在上面的家具，如 table, chair等，也包括“小物体”可以放在里面的家具或家电，如 closet, refrigerator, microwave等。

* “小物体”：机器人可以移动的物体，位置可以改变，通常在“大物体”上面或里面。

– location:为整数数值，代表不同的位置编号，并且假设，任意不同的两个位置之间是可达的，即机器人可以从任意位置移动到目标位置。一个位置内，可以同时放置多个物体，但至多只能有一个“大物体”。同时，机器人通过观察行动可以获得当前位置上的所有物体，但不包括在其他物体里面并且门没有开的“小物体”。

– type:目前需要额外说明的类型只有一种container，家庭环境中closet, refrigerator, microwave等都是container。这里假设每个容器都有门可以开关，同时小物体可以放在容器中，也可以从容器中取出。具体对应的状态包括：

*inside:表示“小物体” A（物体编号）在“大物体” B里面。

*opened:表示“大物体”的门处于开的状态。

*closed:表达“大物体”的门处于关的状态。

· 机器人状态：机器人有一个手爪，一个盘子，它们上面是否有东西也决

定机器人的状态。具体的内容包括：

– location: 机器人所在的位置（用 0 表示机器人），与物体位置的表达相同，机器人手爪和盘子上的东西也和机器人有同一位置。

– plate: 机器人盘子的状态，赋值为空(用0表示)或者某个“小物体”的编号。

– hold: 机器人手爪的状态，赋值为空(用0表示)或者某个“小物体”的编号。

比赛平台拥有完整的场景描述，而参赛程序在一开始只能获得部分场景的描述（缺乏一些小物体的位置信息），并且有一定概率出错（小物体位置有可能出错）。参赛程序可以通过人机交互行动、观察行动和行动反馈更新自己的场景描述。

问题求解往往不仅包含有初始状态描述，还需要一些领域知识，例如，一个东西只能有一个位置，机器人盘子中的东西会随着机器人移动。这些知识需要参赛者自己添加。

3.4 机器人物理行动

机器人物理行动，体现机器人的基本行动能力，是仿真平台对机器人底层功能的抽象。物理行动由其执行条件和执行效果来刻画，执行条件为一组对场景的约束，执行效果为对场景的改变。我们假定，机器人只能根据这些物理行动来改变场景状态。

参赛程序需要根据输入的问题，在规定时间内输出规划结果，即原子行动的序列，并根据平台反馈结果不断调整。其中，比赛平台会反馈每个物理行动是否执行成功，当某个物理行动执行失败后，参赛程序需要根据反馈和当前状态重新规划，并将重新规划的结果（行动或行动序列）发给平台执行。平台的评价程序将对最终的执行效果与需要完成任务做对比，并根据任务完成情况给规划结果打分。具体评分原则在后面章节介绍。

本次仿真比赛中，机器人的物理行动为下列九种，在整个比赛中不会改变：

- *move(x)*: 机器人移动到位置X。执行条件为机器人不在此位置（我们假设，地图中标出的其他位置都是可达的）。执行效果为机器人移动到了位置X

(不再在原来的位置)，不受此行动影响的其他状态属性不变（如果机器人手爪拿着东西，或者机器人盘子里面有东西，则这些东西也会随着机器人一起移动）。

- *pickup(A)*:机器人拿起物体A。执行条件为：A在机器人的可取范围内（机器人与A在同一个位置内，并且不在机器人盘子上），A可以拿得动（A为小物体），A不在其他物体（例如refrigerator）内部，机器人手爪没有拿其他东西。执行效果为机器人手爪拿起了物体A（手爪不再空），其他状态属性不变。

- *putdown(A)*:机器人放下物体A。执行条件为机器人手爪正拿着物体A。执行效果为机器人手爪不再拿任何东西，其他状态属性不变（A的位置仍然在机器人当前所在的位置）。

- *toplate(A)*:机器人将物体A放入自己的盘子中。执行条件为机器人手爪拿着物体A，盘子中没有其他物体。执行效果为机器人手爪不再拿任何东西，A在盘子中，其他状态属性不变（A的位置仍然在机器人当前位置，并且以后A会随机器人移动）。

- *fromplate(A)*:机器人将物体A从盘子中拿起。执行条件为A在机器人盘子中，机器人手爪没有拿其他物体。执行效果为A不再在盘子中，机器人手爪拿着A，其他状态属性不变。

- *open(B)*:(必须为container B,下同):机器人打开容器的门。执行条件为机器人和容器B在同一个位置，机器人手爪为空，并且容器的门是关闭的。执行效果为容器的门被打开(*open(B)*)并且不再关闭，其他属性不变。

- *close(B)*:机器人关上容器B。执行条件为机器人和容器在同一个位置，机器人手爪为空，并且容器的门是打开的。执行效果为容器的门关闭不再打开，其他属性不变。

- *putin(A,B)*:机器人将小物体A放到容器B中。执行条件为机器人和容器在同一个位置，机器人手爪正拿着物体A，并且容器的门是打开的。执行效果为，小物体A在容器的内部(*inside(A,B)*)，手爪为空，其他状态属性不变。

- • *takeout(A,B)*:机器人将小物体从中取出来，执行条件为机器人和容器B在

同一个位置,物体A在容器B中,机器人手抓为空, 并且容器的门是打开的。执行效果为, 机器人抓着小物体A,A不再处于容器的内部, 其他状态属性不变。

参赛程序每提交（执行）一个物理行动, 平台将返回此行动是否执行成功。当某行动的执行条件没被满足时, 平台返回执行失败。例如, 执行行动, 但由于错误信息或其他原因, 并不在当前位置, 则此行动执行失败。

3.5 自然语言描述简介

自然语言交互项目中, 用户用词汇受限的英语来表达任务描述。词汇表虽然受限, 但运用这些词汇仍然可以构造出复杂的句子和多种等价的表达。例如: “Give me the red bottle which is near the green bottle.”, “The white cup is on the table.”, “The cup which is white is on the table.”, “There must be a bottle on the desk.”。比赛中自然语言可以出现的词汇如表 3.1 所示。

表 3.1 自然语言词汇表

词汇类型	词汇
代词	me
冠词	a,an,the
助动词	must
动词	do ,be,give,pick,open,close
介词	in,out,on,near,next,to,up,down,from,of,which
形容词	white ,black,red,green,yellow,blue,big,small,each,every,all
副词	there,not
名词	plate,door, 所有物体的种类名称

可以看到表 3.1 中词汇非常有限, 针对应用背景避免了一些复杂情况。代词只有 me, 特指场景中用户(场景中, 人是唯一的), 没有其他代词, 从而避免了自然语言中复杂指代关系的分析。介词中增加 which,从而可以表达常用的定语从句。形容词来自于场景中对物体属性的描述。名词 plate 特指机器人身上的盘子, 用户可以给出任务 “Put the red bottle on the plate.”。一般 “must”, “do not” 用来表达约束。

上面词汇的语法按照英语的一般语法标准, 这里不会特别限制。比赛中, 用这些词汇构成的自然语言语句来表达任务描述。对于任务描述中的目标, 一般用祈使句表达, 例如: “Go to the table.”, “Pick up the bottle which is on the chair.”。其中主动动词含义解释如下 (解释为对物体位置和机器人状态的要求, 其中 A,B 指某个物体):

- give: 一般用法为最终 “give somebody A” 或者 “give A to somebody” 。作为任务，要求物体 A 的位置在 “somebody” 所在的位置，并且它不在机器人的手爪中或盘子上。
- put 要求: 一般用法有三种，第一种 “put A on/near/next to/down to B” ，在物体所在的位置（位置不变），并且它不在机器人的手爪中或盘子上。第二种 “put A down” 或者 “put down A” ，要求物体 A 不再在机器人的手爪中。第三种 “put A in B” ，要求最终物体在容器的内部。
- go: 一般用法为 “go to A” ，要求机器人移动到物体 A 所在的位置。
- pick: 一般用法为 “pickup A” ，要求机器人的手爪握住物体 A。
- take: 一般用法为 “takeout A from B” ，要求从容器 B 中取出物体 A。
- open: 一般用法为 “open B” 或者 “open the door of B” ，要求打开容器 B 的门。
- close: 一般用法为 “close B” 或者 “close the door of B” ，要求关闭容器 B 的门。对于补充信息，用陈述句表达，例如：“The book is on the table.”。一般为 “A is on/near/next to B” 或者 “There is A on/near/next to B” 的形式，它们都解释为物体 A 与 B 在同一个位置，除了为 plate 的情况，此时解释为在机器人的盘子里。又例如，“A can is in the microwave.”，表示一个 ‘can’ 在 ‘microwave’ 中。“The door of the micro wave is opened”，表示 ‘microwave’ 的门是打开的。任务描述中约束用 “Do not...” ， “There must(not)be...” ，或者 “...must(not)be ...” 的方式表达。“Do not” 后面一般接任务目标，例如 “Do not go to ” ， “Do not put on ” 。“There must(not)be” 后面一般接信息描述，例如 “There must(not)be on” 。“...must(not)be...” 一般描述两物体的关系，例如 “The red bottle must(not)be on the plate” 。

四、比赛场地及器材

4.1 比赛场地

仿真的场地需求为：比赛区域长×宽为5米×6米

4.2 比赛器材

(1) 台式计算机2台，配显示器、键盘、鼠标 计算机配置要求如下：

CPU：酷睿i7内存容量：16GB系统位数：64位显卡：1920×1080以上分辨率 操

作系统：Ubuntu18.04LTS 编译器：gcc4.6.5依赖：Boost1.65（或以上）Cmake3.0.0

（或以上）iclingo3.0.5

(2) 投影仪1台

(3) 幕布1套

(4) 桌子10张

(5) 椅子20条

(6) 插线板3个

(7) 网线2根，长度20米

(8) 供书写比赛通知、预告比赛次序、提供成绩公布，信息公布等的白板带书写笔等1套

五、评分标准

比赛要求参赛队伍对比赛平台提供的每一个问题，根据初始的场景描述和任务描述，计算出合理的行动策略，规定由人机交互行动、观察行动和物理行动组成的行动序列以完成任务，并且在执行失败后重新规划。

参赛程序必须在规定时间内结束运行，平台将根据程序运行时间、已完成的任务数、不违反的约束数和执行的行动数，综合计算出参赛程序的分数，并根据所有问题的总分对参赛队伍排名次。

参赛程序完成任务描述的情况，由其完成的目标数目和维护的约束数目决定。一个任务描述可能含有多个目标或约束（不考虑补充信息），自然语言中的一句话，指令表达中的一条指令，就表达一个目标或约束。行动序列完成一个目标或维护一个约束，定义如下：

- 参赛程序的终止状态：正常情况下，从场景描述的初始状态出发，参赛程序一个个的提交行动（包括人机交互行动，观察行动和物理行动）。这些行动有些执行成功，有些执行失败。对于物理行动，若执行成功，则场景状态被相应的改变；

若执行失败，则场景状态保持不变。参赛程序应根据行动的反馈，合理调整行动策略。参赛程序结束运行时或超过规定时间进程中止时的场景状态，作为其终止状态。

- 参赛程序完成一个目标：其终止状态满足此目标的要求。

- 参赛程序维护一个约束：从初始状态到终止状态中间每一步的状态都满足此约束的要求。

- 评分标准如下：

- 完成一个目标，获得 40 分。

- 维护一个约束，获得 20 分。前提是完成至少一个目标，如果一个原子行动序列不能完成任何目标，则不计算其维护的约束分数。

- 执行行动会扣除相应的分数，不论执行是否成功：

- 执行一次 move 行动扣除 4 分。

—执行一次人机交互行动或其他物理行动扣除 2 分。

—执行一个观察行动扣除 1 分

• 从第一个行动开始记分，直到参赛程序结束运行或超时中止。所以，一个参赛程序对于一个问题的基础得分（此处 sgn 为符号函数）：

问题基础得分 = $20 \times \text{维护的约束数目} \times \text{sgn}(\text{完成目标}) + 40 \times \text{完成的目标数目} - 4 \times \text{move 行动个数} - 2 \times \text{人机交互行动} - 2 \times \text{其他物理行动个数} - \text{观察行动个数}$ ，此次比赛在评分中，还考虑参赛程序的效率因素。在规定时间内，每提前 0.1s 增加 1 分（要保证至少完成一个目标）。具体说来，问题的效率得分

（ $\lceil \cdot \rceil$ 为上取整符号）：问题效率得分 = $2 \times \lceil \text{规定时间} - \text{参赛程序结束时间} \rceil \times 10 \times \text{sgn}(\text{完成目标})$ 。参赛程序对于一个问题的最终得分为：问题基础得分 + 问题效率得分。

比赛的每个阶段都会给出 50 道问题，按当前阶段所有问题的总得分进行排名。所有问题的规划结果将以 log 日志的形式被保存下来，赛后各参赛队可在比赛专用电脑上自行拷贝。

```
(when (and (at 0 ?z) (not (= ?z ?y)) (plate ?n))
      (and (not (at ?n ?z)) (at ?n ?y))))
)

(:action pickup
  :parameters (?x - objs)
  :precondition (and (small ?x) (handempty) (samelocs ?x))
  :effect (and (not
                (handempty)
                ) (hold ?x))
)

(:action putdown
  :parameters (?x - objs)
  :precondition (hold ?x)
  :effect
  (and (not
        (hold ?x) (handempty))
  )
)

(:action toplate
  :parameters (?x - objs)
  :precondition (and (hold ?x) (plateempty))
  :effect (and (not
                (hold ?x)) (not
                (plateempty)) (handempty) (plate ?x))
)

(:action fromplate
  :parameters (?x - objs)
  :precondition (and (plate ?x) (handempty))
  :effect
  (and (not (plate ?x))
        (not (handempty)) (plateempty) (hold ?x))
)
```

```
)
(:action open
  :parameters (?x - objs)
  :precondition (and (samelocs ?x)
                     (handempty) (closed ?x))
  :effect
    (and (not closed ?x) (opened ?x) (handempty))
)
(:action close
  :parameters (?x - objs)
  :precondition (and (samelocs ?x)
                     (handempty) (opened ?x))
  :effect
    (and (not opened ?x) (closed ?x) (handempty))
)
(:action putin
  :parameters (?x - objs ?y - objs)
  :precondition (and (samelocs ?x)
                     (samelocs ?y) (opened ?y) (hold ?x))
  :effects
    (and (handempty) (opened ?y) (inside ?x ?y))
)
(:action takeout
  :parameters (?x - objs ?y - objs)
  :precondition (and (samelocs ?x)
                     (samelocs ?y) (opened ?y) (handempty))
  :effects
    (and (hold ?x) (not
            inside ?x ?y
            (opened ?y))
    )
)

(:axiom
:vars (?x
      -objs)

      :context
      (exists (?l -locs)
        (and (at ?x ?l)
```

```

      (at0 ?l)))
:implies (samelocs ?x)
)
)

```

B.1 场景描述 PDDL 描述举例

```

(define (problem HOMESTATE1)
  (:domain HOMEROBOT)
  (:objects white red green yellow blue black - colors human couch
    cupboard chair table teapoy book
  ca
  n
    remotecontrol refrigerator television bottle plant cup sofa
    airconditioner workspace worktable bed desk - sorts)
  (:init (at ROBOT 0) (plateempty) (handempty)
    (sort 1 human) (at 1 1) (big 1)
    (sort 2 couch) (at 2 2) (big 2)
    (sort 3 table) (at 3 3) (big 3)
    (sort 4 cupboard) (at 4 4) (big 4)
    (sort 5 teapoy) (at 5 5) (big 5)
    (sort 6 television) (at 6 6) (big 6)
    (sort 7 worktable) (at 7 7) (big 7)
    (sort 8 refrigerator) (at 8 8) (big 8)
    (sort 9 desk) (at 9 9) (big 9)
    (sort 10 book) (color 10 red) (at 10 7) (small 10)
    (sort 11 can) (color 11 red) (at 11 4) (small 11)
    (sort 12 can) (color 12 green) (at 12 5) (small 12)
    (sort 13 bottle) (color 13 red) (at 13 3) (small 13)
    (sort 14 bottle) (color 14 blue) (at 14 3) (small 14)
    (sort 15 bottle) (color 15 green) (at 15 5) (small 15)
    (sort 16 remotecontrol) (at 16 2) (small 16)
    (sort 17 cup) (color 17 blue) (at 17 9) (small 17))
  (:goal (and (at 13 5) (at 11 5) (plateempty) (handempty))))

```