

自动化码头水平运输仿真课程设计

吴军 沈立文 赵一桐

2019 年 1 月 18 日

目录

第一阶段 系统分析

- 一、工艺分析
- 二、仿真实体
- 三、系统活动与事件

第二阶段 系统设计与实现

- 一、总体架构
- 二、Simcore 仿真内核
- 三、仿真逻辑控制设计
- 四、交通策略设计
- 五、Unity3d 可视化

第三阶段 仿真数据分析

- 一、初始化数据参数
- 二、简要案例介绍

第一阶段 系统分析

一、生产工艺分析

工艺布局范围覆盖从岸边延伸至堆场边界。主要分为水平运输区和堆场箱区。水平运输区分为岸边装卸作业区、缓冲车道、高速车道。堆场箱区包括缓冲对位区、支架区和箱区。

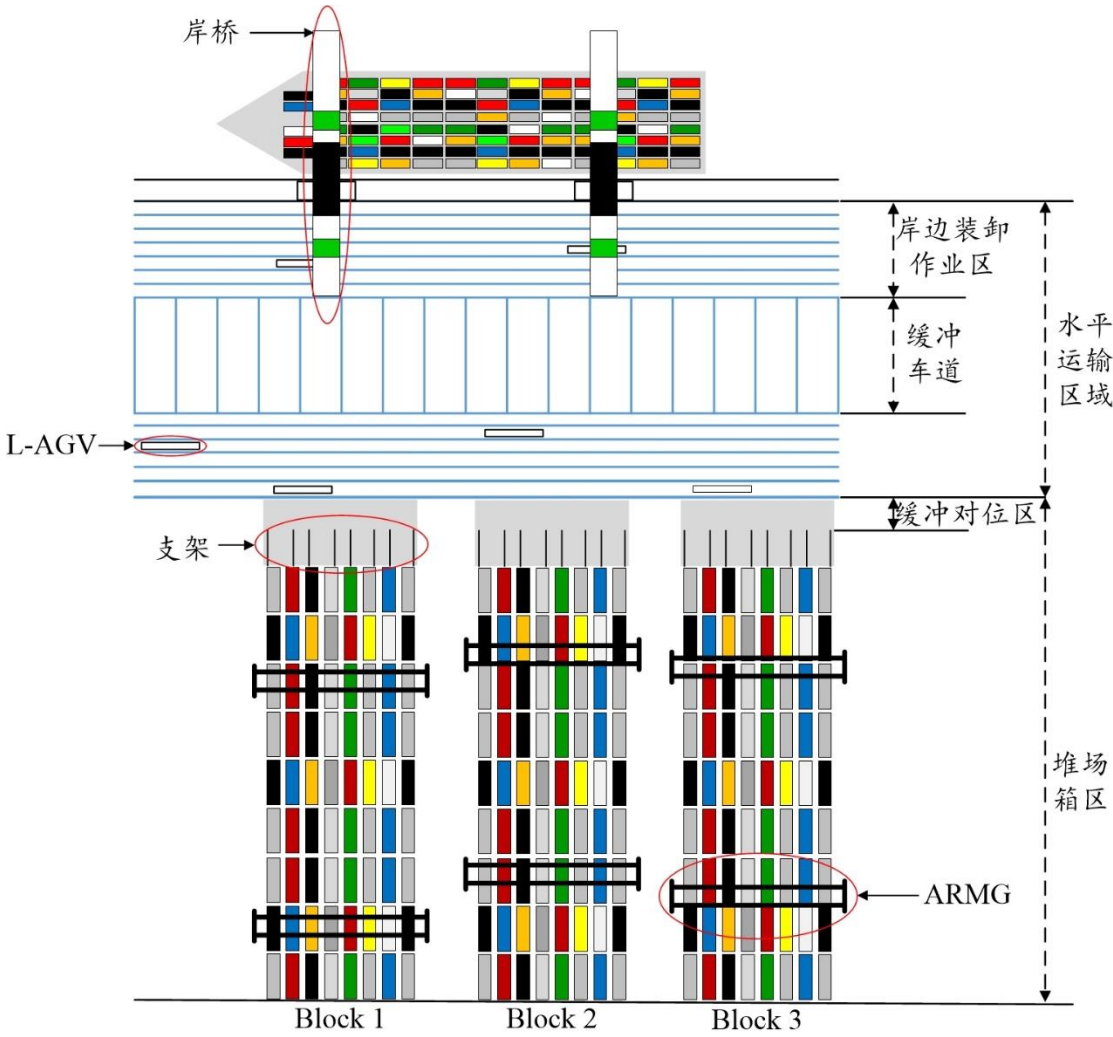


图 1- 1

表格 1- 1 场地布局参数表

布局名称	布局参数	数值设定
岸边车道	岸边车道数量	5 (根)
	岸边车道宽度	4 米
	岸边车道长度	160 米
仿真场地	平面尺寸	160 米 x
	水平运输区域	160 米 x
	堆场箱区	160 米 x
缓冲车道	缓冲车道数量	17 (根)
	缓冲车道长度	16 米
	缓冲车道宽度	4 米
高速车道	高速车道数量	5 (根)
	高速车道长度	160 米
	高速车道宽度	4 米
缓冲对位区	缓冲对位区数量	4
	缓冲对位区尺寸	16 米 x 4 米
支架区	支架区支架数	5
	支架区尺寸	12 米 x 20 米
	单一支架区长度	12 米
	单一支架区宽度	4 米
箱区	箱区长度	
	箱区宽度	
	箱区间距	
	相邻贝位间距	
	箱区场吊间距	

表格 1- 2 设备参数设置

设备名称	设备参数	数值设定	说明
L-AGV	AGV 尺寸	12 米 x2.35 米 x2.4 米	40 英尺标准箱尺寸
	AGV 转向速度	$a_0+a_1*t+a_2*t^2+a_3*t^3+a_4*t^4$	服从四次方多项式函数
	AGV 转向加速度	$a_0+a_1*t+a_2*t^2+a_3*t^3$	服从三次方多项式函数
	AGV 速度		
	AGV 转向加速度		
	AGV 转弯半径		球形插值函数计算所得
Ship	船舶贝位尺寸	10 个贝位, 贝位宽度 16m	
ARMG	场吊运行至目的贝位时间		
	场吊宽度		
	场吊高度		

	小车吊起集装箱时间		
QC	岸桥轨道宽度		
	岸桥小车运行至指定贝位时间		
	岸桥小车吊起集装箱时间		
Holder	支架长度	12 米	同时满足 AGV 和 40ft 集装箱尺寸
	支架宽度	4 米	
	支架高度	4.4 米	

二、仿真实体分析

港口仿真系统涉及到大量的实体，而每一个实体的结构、功能、性能都是借助一系列数据来体现的。港口仿真系统是一个由相互关联、影响的实体依照一定的规则，科学、系统、有机地联系在一起形成的整体。首先，对其实体进行分类。实体的分类方式繁多，有按照实体层次划分、区域划分、物流功能划分、类别划分等。本文按照类别不同进行划分，分为设施类永久实体、临时类实体和运输类永久实体三类。

设施类永久实体是基础性实体，一方面提供集装箱存放地点，限定运输类永久实体的活动范围，如堆场，供运输类永久实体进行装卸搬运等物流活动，岸桥轨道、场吊轨道限制了岸桥和场吊的运行轨迹和作业范围，为运输设备提供了形式规则。另一方面为多种运输实体交接提供间接解耦方式，如支架箱区缓解部分实体利用率低，等待时间长的问题。同时，AGV 自身存在电量消耗的情况，在利用有限场地资源的情况下，利用支架进行充电具有一定优势。

临时类实体主要指船舶和集装箱，与作业密切相关。在港口系统仿真中，集装箱在港区搬运途中坐标位置存在连续变化和顺时变化，是港口设备驱动的来源。

运输类永久实体是实现集装箱空间位置变化的重要实体，如岸桥、场吊、AGV，通过运动形式为自动化集装箱码头提供不同的物流服务。

对于设施类永久实体，以下通过功能分析和属性分析，进行说明。

1. 堆场支架

堆场支架是水平运输和堆场的耦合部分。

主要功能：为 AGV、场吊卸箱提供临时存放方式，节省了设备等待时间，提高了设备利用率。

主要属性：

1) 尺寸：长、宽、高。

2) 位置：坐标。

2. AGV 行驶道路

AGV 行驶道路分为三类，从岸边向堆场分别是装卸车道（岸边车道）、高速车道、缓冲车道。

主要功能：为 AGV 运输设备提供行驶路径及行驶规则。

主要属性：

1) 类别：不同 AGV 行驶道路的功能不同，有装卸和通行之分。

2) 尺寸：长、宽等。

3) 位置：起始点、终点、转弯点坐标等。

4) 通行规则：道路路口通行规则、转弯换道的通行规则等。

3. 场内设备用轨道

场内设备用轨道主要是自动化集装箱码头内部装卸搬运设备的规定行驶轨道，有岸桥轨道和场桥轨道。

主要功能：为装卸设备提供固定行驶轨道。

主要属性：

1) 轨道类别：岸桥轨道、场桥轨道。

2) 轨道尺寸：长度、宽度、跨距等。

3) 轨道位置：起始点、终点坐标等。

4. 集装箱箱区

自动化集装箱码头箱区用于在码头暂时堆存集装箱，本仿真系统中采用堆场垂直于岸线的分布方式，每个箱区配备 2 台自动化场吊。

主要功能：堆放集装箱，为装卸传播作业提供集装箱缓存空间。

主要属性：

1) 箱容量：可堆存最大标准集装箱数量。

2) 箱区名称：箱区的唯一标识。

3) 集装箱箱位：贝、排、层。

“贝”是箱区平行于集装箱长度方向的堆存单位，在本系统中，一个贝位可容纳一个 40ft 标准集装箱位置。

“排”是箱区平行于集装箱宽度方向的堆存单位。

“层”是箱区内集装箱堆放高度表示。

4) 箱区位置：箱区内每个箱位在码头的具体坐标位置。

5) 箱区堆存信息：包括已堆存的各类集装箱数、为堆存的集装箱数量和位置等信息。

6) 箱区计划信息：规划出来的箱区堆存计划信息。

对于临时类实体，以下通过功能分析和属性分析，进行说明。

1. 集装箱

集装箱是指具有一定规格和强度的用以转运货物，可以反复使用的容器。

主要功能：标准装载货物容器，在自动化集装箱码头物流系统中为流经系统的基本货物单元。

主要属性：

1) 箱型：40ft 集装箱

2) 尺寸：标准箱型

3) 堆存位置：堆放或计划堆放位置。

2. 集装箱船舶

集装箱船舶是集装箱海上运输工具。

主要功能：集装箱海上运输工具，从海上运输集装箱进入和离开码头。

主要属性：

- 1) 箱位总容量：船舶的箱位容量是指船舶最大的载箱数量，通常用“TEU”做单位表示，即标准集装箱，箱位容量可表示集装箱船舶的大小。
- 2) 箱位分布信息：箱位分布信息是指集装箱船的每个贝位中的具体箱位。
- 3) 船舶到达规律：集装箱船舶到达的间隔时间分布、装卸箱量分布分布等信息。
- 4) 船舶靠离泊时间、靠泊位置。

对于运输类永久实体，以下通过功能分析和属性分析，进行说明。

1. 岸桥

岸桥，即岸边轨道式龙门起重机，完成集装箱岸边船舶装卸作业。

主要功能：完成岸边集装箱船舶的装卸作业。

主要属性：

- 1) 类型：双小车岸桥、单小车岸桥
- 2) 尺寸参数：起升高度、外伸距、内伸距、轨距
- 3) 设备运行参数：大车、小车和吊具的加速度、速度

2. 场桥

场桥，即堆场内龙门起重机，完成箱区内集装箱装卸搬运作业。

主要功能：完成集装箱在堆场内装卸搬运作业。

主要属性：

- 1) 类型：轨道式场桥
- 2) 尺寸参数：起升高度、轨距
- 3) 设备运行参数：大车、小车和吊具的加速度、速度

3. AGV

AGV 是自动化集装箱码头中常用的水平运输设备

主要功能：完成集装箱在岸边同箱区前端之间的运输作业。

主要属性：

- 1) 设备类别：L-AGV
- 2) 设备尺寸：长、宽、高
- 3) 设备运行参数：速度、加速度、转弯半径

二、系统活动及事件分析

自动化集装箱码头卸船作业主要完成将集装箱从船舶卸至码头箱区的装卸作业。不同的工艺方案，不同集装箱类型所对应的卸船作业不同。卸船进程如下图所示：



图 1- 2 自动化集装箱码头进口箱卸船过程

以上图为例，可以分为 7 个事件和 6 项活动。

事件：

岸桥开始卸箱事件

卸船作业时，岸桥在接收到卸船作业指令时，吊具起升，小车行驶至目标集装箱所在船舶位置，下降吊具装备抓取目标集装箱。该事件开始于岸桥吊具准备起升。

岸桥完成卸箱

自动化集装箱码头使用 AGV 完成岸桥卸箱，当集装箱释放至 AGV 时，岸桥卸箱活动完成。岸桥卸箱开始至岸桥卸箱完成的过程可视为岸桥卸箱活动，该活动主动实体是岸桥，被动实体是集装箱。

集装箱被 AGV 放至支架

AGV 和场吊的交接过程可以通过支架完成间接耦合，活动过程可称为水平运输活动。水平运输活动的主动实体是水平运输设备，被动实体是集装箱。

集装箱被海侧场桥抓取

以集装箱角度看，从集装箱被 AGV 放至支架至集装箱被海侧场桥抓取事件之间的活动过程可称为等待移箱活动。等待移箱活动的主动实体是支架，被动实体是集装箱。

集装箱被卸至堆场过渡箱位

以集装箱角度看，从集装箱被海侧场桥抓取至集装箱被卸至堆场过渡箱位事件之间的活动过程可称为海侧卸箱活动。海侧卸箱活动的主动实体是海侧场吊，被动实体是集装箱。

集装箱被陆侧场桥抓取

对于进口集装箱，最终需要有外集卡到码头后提箱出场，因此需要将集装箱从卸箱是比较靠近还测得位置转移至提箱时比较靠近陆侧的位置。在实际自动化集装箱码头作业过程中，箱区内转移并非一次完成，而是由海侧和陆侧场桥反复翻捣，不断调整集装箱堆存位置来实现的。为了简化这一过程，假设场内移箱是一次完成。当陆侧场桥接收到移箱作业指令时，行驶至目标集装箱所在位置，调整好小车吊具下降抓取目标集装箱，然后起升吊具，开始往箱区陆侧指定贝位行驶。

集装箱被卸至堆场目的箱位

当陆侧场桥带箱行驶至箱区陆侧目标贝位后，调整好小车位置后，吊具开始下降，集装箱被卸至箱区陆侧指定位置。从集装箱被陆侧场桥抓取至集装箱被卸至箱区陆侧指定位置，可视为集装箱箱区内移箱陆侧的活动。当卸船集装箱到达箱区陆侧指定位置后，该集装箱卸船作业可视为完全完成。

装船作业进程

自动化集装箱码头卸船作业主要完成将集装箱从箱区场内至集装箱船舶的装卸搬运作业。本系统采用以下装船进程，事件及活动与卸船过程相似，仅方向相反。



第二阶段 系统设计与实现

一、总体架构

系统分为仿真内核模块、交通规划与控制服务、数据管理后台与 Unity3d 客户端。架构示意简图如下：

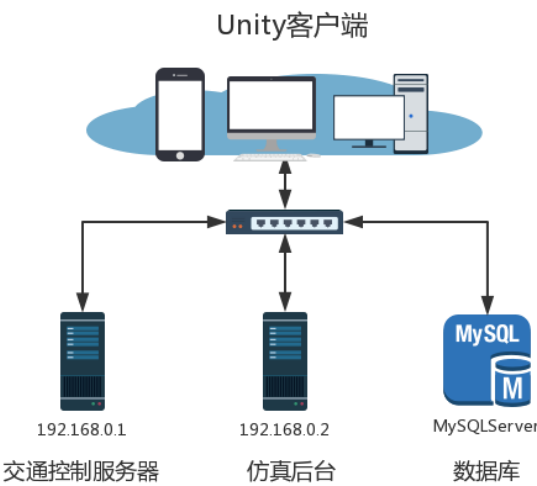


图 2-1 总体架构

仿真内核模块包含离散事件仿真以及仿真初始数据输入与运行数据输出功能，部署于仿真后台服务器，用户仿真实例与交通控制服务器通过 socket 通信实现在仿真过程中进行实时路径规划与控制，交通控制器将计算得到的路径及相关数据发送给 Unity3d 客户端进行实时可视化。

二、Simcore 仿真内核



基于开源离散事件仿真框架 Simpy，使用协程（Python Generator，下文统称协程）方式实现仿真事件推进，内置共享资源（Resource、Container、Store 等）实现协程间的多种同步方式。支持实时仿真钟推进。

- simcore
- simcore.core - 核心仿真组件
- simcore.events – 仿真事件类型

- `simcore.exceptions` – 仿真错误类型
- `simcore.resource` – 协程共享资源同步原语

使用和更改的内容如下 </EasyTerm/server/simcore/>:

class `simcore.core.RtEnvironment`

实时仿真钟运行环境，基于 `simpy.core.RtEnvironment`。

now

当前环境仿真钟。

active_process

当前环境活跃进程（Process）。

schedule(event, priority=1, delay=0)

将`event`事件以优先级`priority`插入未来事件表（未来事件表是`[发生时间, 优先级, 事件编号]`构成的小根堆，每次调用`step`方法弹出堆顶事件，若事件发生时间相同则以优先级降序排序，若优先级也相同则先到事件先处理）。

step()

从未来事件表中弹出下一事件并处理，处理完成后重新激活事件所在协程。

run(until=None)

不断执行`step`方法直到`until`事件发生，如果 `until` 为 `None` 则运行直到所有协程结束为止，与 `simpy` 不同的是可以接受其它进程（操作系统进程）传来的事件，通过 IPC 实现。

class `simcore.events.Timeout(env, delay)`

延时事件，与 `simpy.events.Timeout` 一致，该事件在被生成（`yield`）之后立即插入未来事件表，在经过`delay`秒后触发。

class `simcore.events.Send(env, addr, msg)`

向外部进程发送消息，该事件生成后 `select` 系统调用（Windows 平台下，Unix 为 `epoll`）将监听 IO 事件。

class `simcore.events.Realtime(env, sock)`

配合`Send`事件使用，等待外部进程回应，触发事件。

class `simcore.events.Process(env, generator)`

处理一系列事件组成的生成器`generator`，即用户的仿真活动逻辑，协程在 `yield` 一个事件后挂起，`Process` 的作用是在事件被触发后重新激活被挂起的协程。

`Process` 本身也是一个事件，可以通过 `yield Process()` 生成一组事件。

class `simcore.resources.Store(env, capacity=INF)`

先进先出的资源队列，通过`StorePut`、`StoreGet`事件插入和取出资源，功能类似 Python 线程通信队列 `queue.Queue`

put(env, object)

生成 `StorePut` 事件，插入资源，若容量已满则挂起协程直至有空闲容量。

get(env)

生成 `StoreGet` 事件，取出资源，若无可用资源则挂起协程直至有资源可用。

`class simcore.resources.FilterStore(env, capacity=INF)`

可以通过 filter 函数取出满足特定条件的资源，其余功能与 Store 一致。

`get(env, filter)`

取出 filter 函数返回为 True 的资源，若无满足条件的资源则等待下一次资源插入重新调用。

三、仿真逻辑控制设计

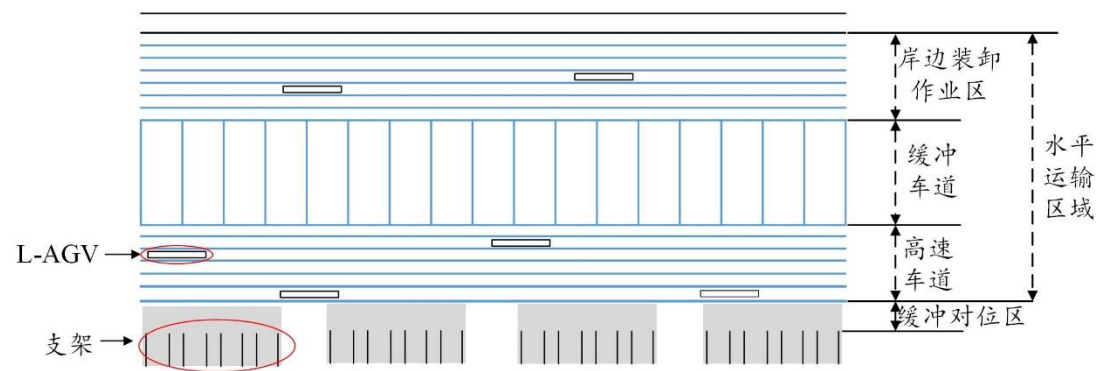


图 2-2 车道布置

1.工艺

主要仿真内容为岸桥与箱区间的水平运输，运输工具为 AGV，海侧采用 5 车道设计，方向均为沿岸线向左，五车道中第 2、4 车道主要用于岸桥与 AGV 的交接，其余 3 条为高速车道。缓冲车道主要用于临时停放暂时因交通原因还不能去岸桥或箱区作业的 AGV 以及空闲的 AGV。岸侧 5 车道方向交错，方便箱区间交换集装箱。

每个箱区配备 5 个支架作为 AGV 与海侧场吊交接的缓冲区，该场景共 4 个箱区，可配置 2 台双小车岸桥。AGV 数量需要小于缓冲车道数量。（具体工艺见系统分析章节）

2.系统输入

客户端通过 csv 文件获取系统输入，示例文件位置 [/EasyTerm/client/data/test_data.csv](#), Flask 后台解析数据后生成仿真实例。

1 用户指定作业设备：

	类型	编号
作业设备	qc	1
作业设备	qc	2
作业设备	armg	1
作业设备	armg	2
作业设备	armg	3
作业设备	armg	4
作业设备	agv	1
作业设备	agv	3
作业设备	agv	5
作业设备	agv	7

图 2-3 指定作业设备

图中 qc 为岸桥、armg 为海侧场吊、agv 为运输车辆，设备信息来自数据库（设备由用户生成，未做用户接口）。

II 用户指定箱任务：

	箱号	类型	优先级	起点设备	编号	终点设备	编号	运输车辆
任务	1	出港	1	armg	1	qc	1	1
任务	2	入港	2	qc	1	armg	1	3
任务	3	岸桥周转	3	qc	2	qc	1	5
任务	4	箱区周转	4	armg	2	armg	1	7

图 2-4 指定箱任务

箱子信息来自数据库（集装箱属性由用户设定，未做用户接口），每个箱任务需要指定对应的类型、优先级、起点设备、终点设备及运输车辆，系统将根据箱任务的优先级为各个设备生成作业序列，例如图 4 中岸桥 1 会先做优先级为 1 的装船任务 1 再做卸船任务 2；同理场吊 1 会先做箱任务 1 再做箱任务 2；AGV3 因为只有一个箱任务 2，它会立即前往箱区 1，而此时若场吊 1 正在执行箱任务 1，则 AGV3 会在支架处等待场吊 1 做完箱任务 1。

3. 仿真逻辑

仿真逻辑位于 `/EasyTerm/server/facilities.py` 文件中，任务基类为 `BaseTask`，定义全局任务编号、类型及作业设备，设备基类为 `BaseFacility`，定义设备位置、任务序列、插入任务、取出任务等基本属性及方法。

设备协程 `process` 伪码中 `yield` 原语后跟一个仿真事件，**yield 事件将会挂起协程，待事件发生重新激活协程。**

```
class QCTask(raw_id, flag, bay)
```

岸桥任务类型。

flag 装卸标志

bay 装卸贝位

lane 装卸车道

transporter=None 运输工具

```
class QC(env, name, position, lanes)
```

岸桥作业逻辑，除基类属性外需指定装卸车道`lanes`，实现简单装卸船逻辑。

shift_store

与 AGV 的共享资源，用于和 AGV 协程的同步。

lanes_queue

各装卸车道排队队列，用于车道选择与设备同步。

select_min_lane(task_id)

选择排队最短的车道。

process()

岸桥装卸逻辑以及和 AGV 的交互伪码如下：

从任务队列取出任务，并选择装卸车道（如果 AGV 未选）

if 卸船任务

yield 等待岸侧小车就绪

yield 等待 AGV 到达

yield 落箱到 AGV

if 装船任务

yield 等待 AGV 到达

yield 从 AGV 提箱
yield 箱子放到船上

class AGVTask(raw_id, flag, bay)

AGV 任务类型

start_facility 起点设备（可以是岸桥或场吊）

end_facility 终点设备（可以是岸桥或场吊）

class AGV(env, name, position, speed)

AGV 作业逻辑，除基类属性外需指定速度`speed`，实现任务接送与水平运输。

speed AGV 行驶速度

transporter=None 运输工具

process()

AGV 运行与交互逻辑伪码如下：

从任务队列取出任务，并选择起点车道/支架（如果起点设备未选，通知支架如果是箱区）

yield 发送去起点设备行驶路径

yield 到达起点

yield 等待起点设备交出箱子

选择终点车道/支架（如果终点设备未选，通知支架如果是箱区）

yield 发送去终点设备的行驶路径

yield 到达终点

yield 等待终点设备提走箱子

if 任务队列为空

yield 返回缓冲车道

class ARMGTask(raw_id, flag, bay)

箱区场吊任务类型。

flag 装卸标志

lane=None 装卸支架

transporter=None 运输工具

class ARMG (env, name, position, lanes)

海侧场吊作业逻辑，除基类属性外需指定装卸支架`lanes`，实现简单出入箱区逻辑。

transporter=None 运输工具

shift_store

与 AGV 的共享资源，用于和 AGV 协程的同步。

lanes_queue

各装卸支架排队队列，用于车道选择与设备同步。

select_min_lane(task_id)

选择排队最短的车道。

process()

场吊运行逻辑以及于 AGV 的交互伪码如下：

从任务队列取出任务，并选择支架（如果 AGV 未选），通知支架

```

if 出堆箱场任务
    yield 等场吊提箱到支架
    yield 等支架空闲
    yield 落箱到支架
if 入堆场箱任务
    yield 等待支架接到箱子
    yield 从支架提箱
    yield 放到箱区内

```

四、交通策略设计

1. 交通控制

仿真逻辑设计中提到所有设备必须按照任务全局优先级顺序作业, 这个策略在没有交通控制的情况下容易在岸边装卸车道与支架处发生 AGV 与岸桥、AGV 与场吊间的死锁, 如下图支架处的死锁情况:

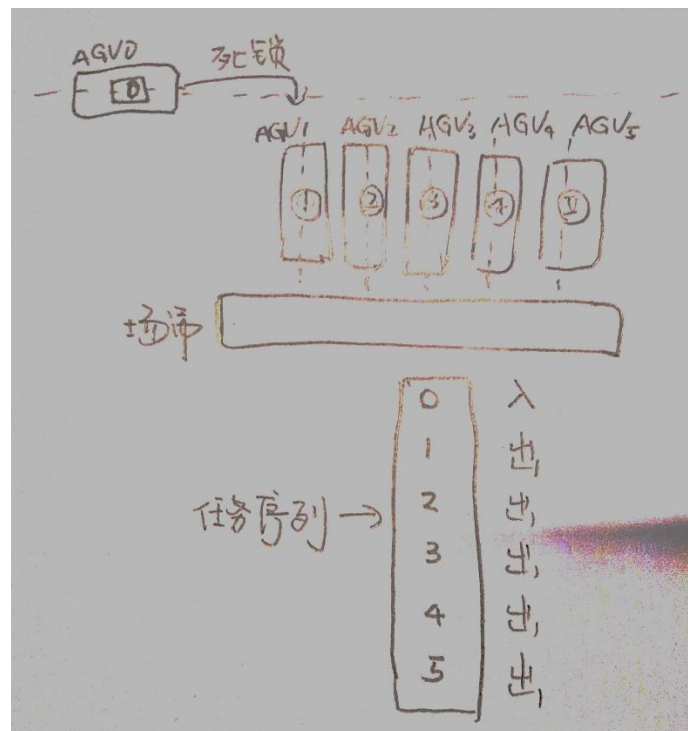


图 2-5 死锁情况

对场吊与 AGV 来说, 支架属于临界资源, 若执行低优先级任务的 AGV 抢先占用了全部支架, 执行高优先级任务的 AGV 则无法作业, 形成死锁。解决方法是限制 AGV 对支架的占用, 即破坏 AGV 对支架的**请求与占用条件**。具体步骤如下:

- (1) 设置车道/支架排队队列共享数据结构, 记录车道/支架占用情况。
- (2) 岸桥/场吊对车道/支架的使用享有绝对优先权, 即车道/支架排队队列中优先级最高任务必定是岸桥/场吊当前任务。
- (3) AGV 需要使用支架时判断当前任务是否已在车道/支架队列中, 若不在则将任务插入队列; **判断当前任务优先级是否大于等于排队队列中的最高优先级, 是则前去作业, 否则等待。**
- (4) 岸桥/场吊完成当前任务后删除车道/支架排队队列中的该任务。

交通控制模块位于 [/EasyTerm/server/traffic.py](#)

2. 路径规划

AGV 路径规划使用 A*算法。

五、Unity3d 可视化

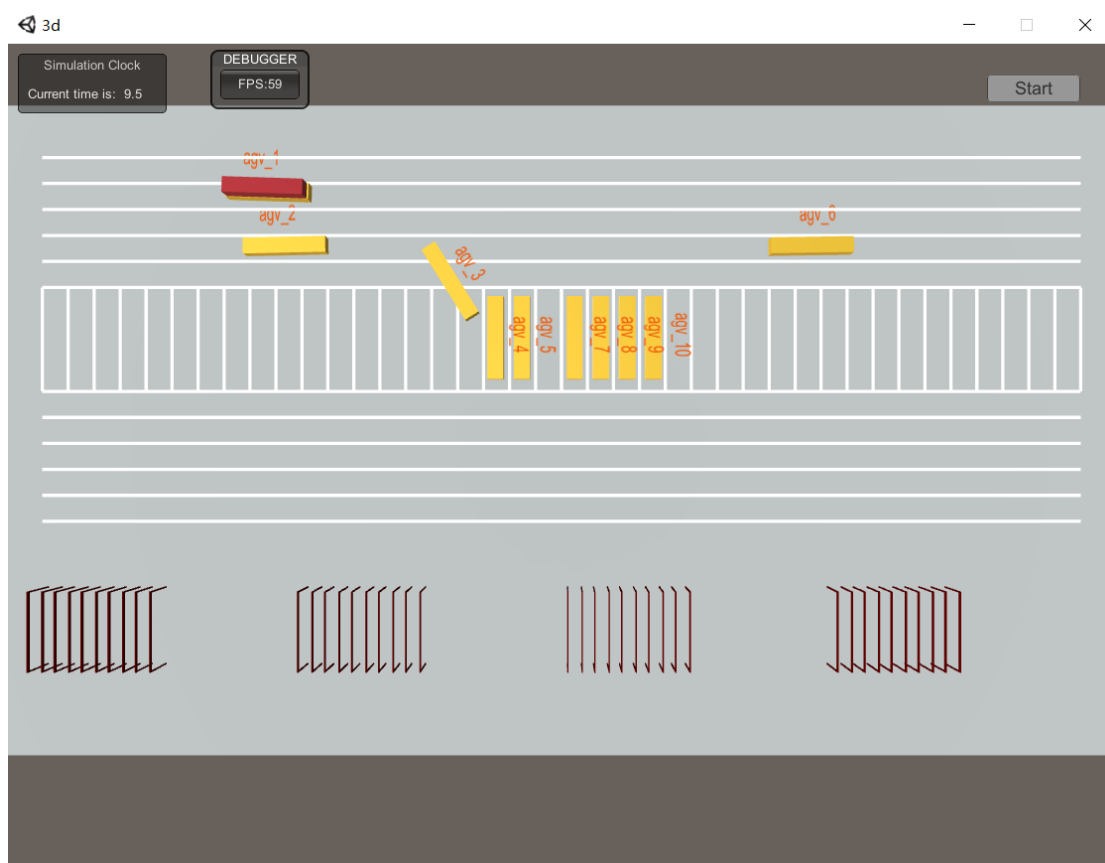


图 2-6 Unity3d 运行场景

使用 Unity3d 中的 iTween 库实现多点路径的行走；通过 socket 套接字与仿真服务器建立连接，用于数据交互如交通指令与仿真钟同步，使用脚本说明如下。

LaneScript 生成道路。

HolderScript 生成支架。

ClockScript 同步后台仿真钟。

StartSim 获取仿真实例数据并请求开始仿真。

TaskManager 仿真场景初始化。

AGVManager AGV 实例管理，TCPSocket 与后台通信。

AGVMovement AGV 运动脚本。

第三阶段 仿真数据分析

一、基本变量和参数

1.输入参数变量

单次仿真，用户输入的参数如下，用户选择的 AGV 数目，用户选择需要作业的集装箱数目，用户选择的优先级序列， 单次仿真的各个任务设备分布。

2.输出变量：

表 3-1

status	本次仿真状态
sim_time	本次仿真用时
qc_x_start	某岸桥开始作业时刻
armg_x_start	某场吊开始作业时刻
agv_xx_start	某 AGV 开始作业时刻
holder_xxx_start	某支架开始作业时刻
armg_x_wait	某场吊等待交接时刻
qc_x_wait	岸桥等待交接时间
agv_x_wait	某 AGV 等待交接时间
holder_xxx_occupied	某支架被集装箱占用时间
armg_x_end	某场吊停止作业时刻
agv_xx_end	某 AGV 停止作业时刻
qc_x_end	某岸桥停止作业时刻
holder_xxx_end	某支架停止作业时刻

3.仿真分析

3.1 仿真分析指标设定

这里，我们选择 AGV 工作时间，AGV 等待时间，岸桥工作时间，岸桥等待时间，场吊工作时间，场吊等待时间，用来衡量本次仿真作业效率的基本指标。

其中：

AGV 工作时间： $agv_xx_work = agv_xx_end - agv_xx_start - agv_xx_wait$

AGV 等待时间： agv_xx_wait

岸桥工作时间： $qc_xx_work = qc_xx_end - qc_xx_start - qc_xx_wait$

岸桥等待时间： qc_xx_wait

场吊工作时间： $armg_xx_work = armg_xx_end - armg_xx_start - armg_xx_wait$

场吊等待时间： $armg_xx_wait$

仿真分析方案，通过控制变量的方式，对 AGV 数量，作业集装箱数量，集装箱作业任务优先级顺序，集装箱作业任务分布情况，进行分析：

表 3-2

Type	Var	NonVar_A	NonVar_B	NonVar_C
1	AGV_Num	Con_Num	Task_Prio	Task_Distr
2	Con_Num	AGV_Num	Task_Prio	Task_Distr
3	Task_Prio	Con_Num	AGV_Num	Task_Distr
4	Task_Distr	AGV_Num	Con_Num	Task_Prio

由于整个系统完整的数据测试需要大量数据的输入，这里以几种样例进行测试，展开一定分析。

3.2 仿真输出情况

3.2.1 简单任务类型：

设备初始参数设置如下：

表 3-3

	类型	编号
作业设备	qc	1
作业设备	qc	2
作业设备	armg	1
作业设备	armg	2
作业设备	armg	3
作业设备	armg	4
作业设备	agv	1
作业设备	agv	3
作业设备	agv	5
作业设备	agv	7

任务初始参数设定：

表 3-4

	箱号	类型	优先级	起点设备	编号	终点设备	编号	运输车辆
任务	1	出港	4	armg	1	qc	1	1
任务	2	入港	3	qc	1	armg	1	3
任务	3	岸桥周转	2	qc	2	qc	1	5
任务	4	箱区周转	1	armg	2	armg	1	7

仿真输出结果:

```
{
  "status": "ok",
  "sim_time": 84.62399999999816,
  "data": {
    "qc_1_start": 0,
    "qc_2_start": 0,
    "armg_1_start": 0,
    "armg_2_start": 0,
    "armg_3_start": 0,
    "armg_3_end": 0,
    "armg_4_start": 0,
    "armg_4_end": 0,
    "agv_1_start": 0,
    "agv_3_start": 0,
    "agv_5_start": 0,
    "agv_7_start": 0,
    "holder_481_start": 0,
    "holder_482_start": 0,
    "holder_483_start": 0,
    "holder_484_start": 0,
    "holder_485_start": 0,
    "holder_491_start": 0,
    "holder_492_start": 0,
    "holder_493_start": 0,
    "holder_494_start": 0,
    "holder_495_start": 0,
    "holder_501_start": 0,
    "holder_502_start": 0,
    "holder_503_start": 0,
    "holder_504_start": 0,
    "holder_505_start": 0,
    "holder_511_start": 0,
    "holder_512_start": 0,
    "holder_513_start": 0,
    "holder_514_start": 0,
    "holder_515_start": 0,
    "armg_2_wait": 0.0,
    "armg_2_end": 6.826358903933011,
    "holder_491_occupied": 10.964647099718047,
    "agv_7_wait": 1.5532941994468032,
    "qc_2_wait": 2.8806736582814256,
    "qc_2_end": 12.138315552653525,
    "agv_5_wait": 1.900631105317757,
```

```
"qc_1_wait": 60.812000000005355,  
"armg_1_wait": 45.585294199453315,  
"holder_481_occupied": 47.385065066381934,  
"agv_5_end": 29.623999999981606,  
"agv_7_end": 30.155999999988126,  
"agv_3_wait": 33.00428899409667,  
"agv_3_end": 57.873999999981606,  
"armg_1_end": 62.48235890392114,  
"holder_482_occupied": 63.25900600364454,  
"agv_1_wait": 52.8033215563153,  
"agv_1_end": 83.59399999998277,  
"qc_1_end": 84.51710148372169,  
"holder_482_end": 84.55900600364338,  
"holder_491_end": 84.56464709971755,  
"holder_481_end": 84.585065066381,  
"holder_483_end": 84.59999999999947,  
"holder_484_end": 84.59999999999947,  
"holder_485_end": 84.59999999999947,  
"holder_492_end": 84.59999999999947,  
"holder_493_end": 84.59999999999947,  
"holder_494_end": 84.59999999999947,  
"holder_495_end": 84.59999999999947,  
"holder_501_end": 84.59999999999947,  
"holder_502_end": 84.59999999999947,  
"holder_503_end": 84.59999999999947,  
"holder_504_end": 84.59999999999947,  
"holder_505_end": 84.59999999999947,  
"holder_511_end": 84.59999999999947,  
"holder_512_end": 84.59999999999947,  
"holder_513_end": 84.59999999999947,  
"holder_514_end": 84.59999999999947,  
"holder_515_end": 84.59999999999947  
}  
}
```

3.2.2 中等难度任务：

test_data_simple.csv
设备初始参数设置如下：

表 3-5

	类型	编号
作业设备	qc	1
作业设备	qc	2
作业设备	armg	1
作业设备	armg	2
作业设备	armg	3
作业设备	armg	4
作业设备	agv	1
作业设备	agv	3
作业设备	agv	5
作业设备	agv	7
作业设备	agv	9

任务初始参数设定：

表 3-6

	箱号	类型	优先级	起点设备	编号	终点设备	编号	运输车辆
任务	201	入港	1	qc	1	armg	3	1
任务	202	入港	2	qc	1	armg	3	3
任务	203	入港	3	qc	1	armg	3	5
任务	204	入港	4	qc	1	armg	3	7
任务	205	入港	5	qc	1	armg	3	9
任务	101	出港	6	armg	2	qc	2	1
任务	102	出港	7	armg	2	qc	2	3
任务	103	出港	8	armg	2	qc	2	5
任务	104	出港	9	armg	2	qc	2	7
任务	105	出港	10	armg	2	qc	2	9

仿真输出结果:

```
{
  "status": "ok",
  "sim_time": 103.56200000000536,
  "data": {
    "qc_1_start": 0,
    "qc_2_start": 0,
    "armg_1_start": 0,
    "armg_1_end": 0,
    "armg_2_start": 0,
    "armg_3_start": 0,
    "armg_4_start": 0,
    "armg_4_end": 0,
    "agv_1_start": 0,
    "agv_3_start": 0,
    "agv_5_start": 0,
    "agv_7_start": 0,
    "agv_9_start": 0,
    "holder_481_start": 0,
    "holder_482_start": 0,
    "holder_483_start": 0,
    "holder_484_start": 0,
    "holder_485_start": 0,
    "holder_491_start": 0,
    "holder_492_start": 0,
    "holder_493_start": 0,
    "holder_494_start": 0,
    "holder_495_start": 0,
    "holder_501_start": 0,
    "holder_502_start": 0,
    "holder_503_start": 0,
    "holder_504_start": 0,
    "holder_505_start": 0,
    "holder_511_start": 0,
    "holder_512_start": 0,
    "holder_513_start": 0,
    "holder_514_start": 0,
    "holder_515_start": 0,
    "armg_2_wait": 0.0,
    "qc_1_wait": 0.0,
    "agv_1_wait": 6.6052516464896485,
    "agv_3_wait": 15.449251646472419,
    "armg_3_wait": 42.25923549863538,
```

```
"holder_501_occupied": 61.967065066388486,  
"agv_5_wait": 14.261251646477774,  
"armg_2_end": 33.98235890392114,  
"holder_502_occupied": 36.19706506638736,  
"agv_7_wait": 13.979251646471269,  
"holder_491_occupied": 38.21464709971804,  
"qc_1_end": 46.13364189436139,  
"agv_9_wait": 12.699251646472433,  
"holder_492_occupied": 40.65664709972224,  
"holder_493_occupied": 41.638647099728786,  
"qc_2_wait": 60.59399999998277,  
"holder_494_occupied": 44.370647099706154,  
"armg_3_end": 67.48007510985751,  
"agv_1_end": 72.52999999999884,  
"holder_495_occupied": 45.350647099704894,  
"agv_3_end": 80.6239999999816,  
"agv_5_end": 89.56200000000536,  
"agv_7_end": 96.3739999999816,  
"qc_2_end": 96.7991014836991,  
"holder_481_end": 96.79999999999878,  
"holder_482_end": 96.79999999999878,  
"holder_483_end": 96.79999999999878,  
"holder_484_end": 96.79999999999878,  
"holder_485_end": 96.79999999999878,  
"holder_503_end": 96.79999999999878,  
"holder_504_end": 96.79999999999878,  
"holder_505_end": 96.79999999999878,  
"holder_511_end": 96.79999999999878,  
"holder_512_end": 96.79999999999878,  
"holder_513_end": 96.79999999999878,  
"holder_514_end": 96.79999999999878,  
"holder_515_end": 96.79999999999878,  
"holder_491_end": 96.81464709971654,  
"holder_493_end": 96.83864709972701,  
"holder_495_end": 96.85064709970364,  
"holder_492_end": 96.8566470997206,  
"holder_501_end": 96.86706506638666,  
"holder_494_end": 96.87064709970436,  
"holder_502_end": 96.89706506638561,  
"agv_9_end": 103.56200000000536  
}  
}
```

3.2.3 困难任务类型：

设备初始参数设置如下：

表 3-7

	类型	编号
作业设备	qc	1
作业设备	qc	2
作业设备	armg	1
作业设备	armg	2
作业设备	armg	3
作业设备	armg	4
作业设备	agv	1
作业设备	agv	2
作业设备	agv	3
作业设备	agv	4
作业设备	agv	5
作业设备	agv	6
作业设备	agv	7
作业设备	agv	8
作业设备	agv	9
作业设备	agv	10

任务初始参数设定：

表 3-8

	箱号	类型	优先级	起点设备	编号	终点设备	编号	运输车辆
任务	201	入港	1	qc	1	armg	3	1
任务	202	入港	2	qc	1	armg	3	2
任务	203	入港	3	qc	1	armg	3	3
任务	204	入港	4	qc	1	armg	3	4
任务	205	入港	5	qc	1	armg	3	5
任务	206	入港	6	qc	2	armg	1	6
任务	207	入港	7	qc	2	armg	1	7
任务	208	入港	8	qc	2	armg	1	8
任务	209	入港	9	qc	2	armg	1	9
任务	210	入港	10	qc	2	armg	1	10
任务	101	出港	11	armg	2	qc	2	1
任务	102	出港	12	armg	2	qc	2	2
任务	103	出港	13	armg	2	qc	2	3
任务	104	出港	14	armg	2	qc	2	4
任务	105	出港	15	armg	2	qc	2	5
任务	106	出港	16	armg	4	qc	1	6
任务	107	出港	17	armg	4	qc	1	7
任务	108	出港	18	armg	4	qc	1	8
任务	109	出港	19	armg	4	qc	1	9
任务	110	出港	20	armg	4	qc	1	10

仿真输出结果:

```
{
  "status": "ok",
  "sim_time": 115.67064709973276,
  "data": {
    "qc_1_start": 0,
    "qc_2_start": 0,
    "armg_1_start": 0,
    "armg_2_start": 0,
    "armg_3_start": 0,
    "armg_4_start": 0,
    "agv_1_start": 0,
    "agv_2_start": 0,
    "agv_3_start": 0,
    "agv_4_start": 0,
    "agv_5_start": 0,
    "agv_6_start": 0,
    "agv_7_start": 0,
    "agv_8_start": 0,
    "agv_9_start": 0,
    "agv_10_start": 0,
    "holder_481_start": 0,
    "holder_482_start": 0,
    "holder_483_start": 0,
    "holder_484_start": 0,
    "holder_485_start": 0,
    "holder_491_start": 0,
    "holder_492_start": 0,
    "holder_493_start": 0,
    "holder_494_start": 0,
    "holder_495_start": 0,
    "holder_501_start": 0,
    "holder_502_start": 0,
    "holder_503_start": 0,
    "holder_504_start": 0,
    "holder_505_start": 0,
    "holder_511_start": 0,
    "holder_512_start": 0,
    "holder_513_start": 0,
    "holder_514_start": 0,
    "holder_515_start": 0,
    "armg_2_wait": 0.0,
    "armg_4_wait": 0.0,
    "qc_1_wait": 33.22000000003027,
```


"agv_1_wait": 6.2932516464551895,
"qc_2_wait": 16.158673658272576,
"agv_6_wait": 3.453925304764553,
"agv_2_wait": 15.54325164645519,
"agv_7_wait": 12.949251646472419,
"armg_3_wait": 43.00523549867822,
"agv_3_wait": 15.105251646489648,
"holder_501_occupied": 45.83506506641097,
"agv_8_wait": 10.729251646471255,
"armg_1_wait": 49.25523549864913,
"armg_2_end": 33.88835890393837,
"armg_4_end": 33.88835890393837,
"holder_481_occupied": 55.89706506638728,
"agv_4_wait": 17.293251646455204,
"holder_502_occupied": 36.75306506640445,
"agv_9_wait": 11.949251646472433,
"holder_491_occupied": 39.55864709972992,
"holder_482_occupied": 43.503065066404474,
"holder_503_occupied": 18.253065066404453,
"agv_5_wait": 14.419251646473597,
"holder_492_occupied": 41.53864709972876,
"agv_10_wait": 11.917251646465914,
"holder_511_occupied": 52.46464709971804,
"holder_483_occupied": 23.23506506641094,
"holder_493_occupied": 43.27064709973531,
"holder_512_occupied": 53.78864709972876,
"holder_494_occupied": 44.126647099723385,
"armg_3_end": 68.16607510987356,
"holder_513_occupied": 57.770647099735314,
"agv_1_end": 72.90600000001723,
"holder_495_occupied": 46.48264709974052,
"armg_1_end": 74.41607510987357,
"agv_6_end": 77.21799999999348,
"holder_514_occupied": 59.81464709971803,
"agv_2_end": 83.71799999999348,
"agv_7_end": 83.90600000001723,
"holder_515_occupied": 62.23264709974052,
"agv_3_end": 90.03200000000652,
"agv_8_end": 95.34400000001187,
"agv_4_end": 96.71799999999348,
"qc_2_end": 98.39310148371098,
"agv_9_end": 103.40600000001723,
"agv_5_end": 103.78200000000652,
"agv_10_end": 113.81200000000536,

```
"qc_1_end": 115.57910148372704,  
"holder_495_end": 115.58264709973825,  
"holder_512_end": 115.58864709972588,  
"holder_481_end": 115.59706506638446,  
"holder_484_end": 115.5999999999977,  
"holder_485_end": 115.5999999999977,  
"holder_504_end": 115.5999999999977,  
"holder_505_end": 115.5999999999977,  
"holder_482_end": 115.60306506640201,  
"holder_514_end": 115.61464709971604,  
"holder_494_end": 115.6266470997205,  
"holder_515_end": 115.63264709973915,  
"holder_501_end": 115.63506506640829,  
"holder_483_end": 115.63506506640832,  
"holder_492_end": 115.63864709972604,  
"holder_502_end": 115.65306506640164,  
"holder_503_end": 115.6530650664019,  
"holder_491_end": 115.65864709972733,  
"holder_511_end": 115.66464709971527,  
"holder_493_end": 115.67064709973243,  
"holder_513_end": 115.67064709973276  
}  
}
```