

实验三、AGV 系统的建模与仿真

1. 实验目的

通过完成一个简单的轨道小车的循径运动系统的仿真建模，学习 AutoMod 软件 Path mover 子系统，掌握小车调度作业的基本方法，达到独立完成简单轨道搬运系统仿真建模的目的。同时，掌握该类系统仿真建模思想和方法，培养分析该类系统关键参数、问题和效率的能力，提高依据仿真结果正确分辨、判断系统性能的分析能力。

2. 实验内容

(1) 对象系统描述

有一个环形轨道的自动导引小车搬运系统，平面布置示意图如右图所示。

两种货物，到达频率分别为：

normal (800, 20) sec; uniform

(600, 40) sec;

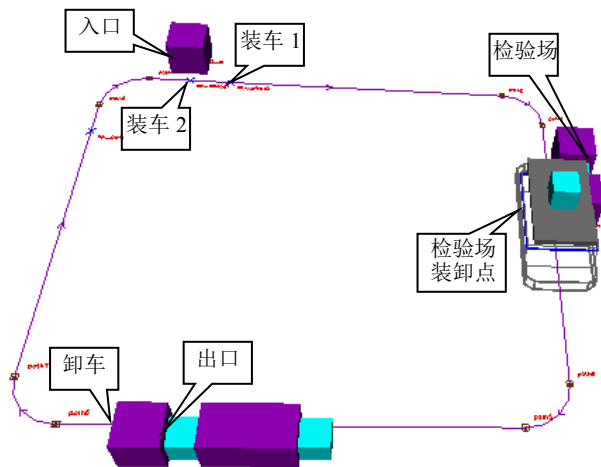
从入口的队列中轮流在两个上载货物的地点装车；

送到检查场前的装卸点，卸载到检查场中检查 uniform (20, 4) min;

检查后仍由检查场的装卸点装车送到出口，卸载到出口队列中；

车辆空闲时就到停车处停车，随时待命，当小车发现入库和检查场有装载任务时就前去装载货物；

该系统为 24 小时连续运行系统。



(2) 实验内容

- ①完成该对象系统的 AutoMod 建模；
- ②对该搬运系统的作业效率和存在的问题进行分析；
- ③提出改进措施，对比小车利用率和系统运行情况。

3. 实验步骤

- (1) 新建一个模型（自动创建一个 process system）；
- (2) 定义小车、队列等实体单元，按平面布置示意图布置相应的实体位置；
- (3) 定义系统参数和实体参数；
- (4) 编辑 source file 文件，编写相应代码，定义系统的流程逻辑；
- (5) 设定运行时间；
- (6) 运行模型；
- (7) 查看仿真动画和仿真输出结果，分析系统瓶颈和系统问题情况；
- (8) 设计改进方案，相应修改仿真模型，仿真运行并比较仿真结果的变化。

作业与思考题

- (1) 系统的瓶颈在哪里？你是如何发现的？
- (2) 哪些参数能有效反映小车搬运能力状况？
- (3) 你解决系统瓶颈问题的思路是什么？请说明理由；
- (4) 你提出的改进方案使该系统产生了哪些变化？
- (5) 谈谈你完成本次实验的心得体会与建议。

自选内容

假设为该系统设计两套小车配置方案，一是1辆容量为2的小车，二是2辆容量为1的小车。试对两个方案进行仿真比较，分析两个方案的系统搬运能力与小车利用率的差别。

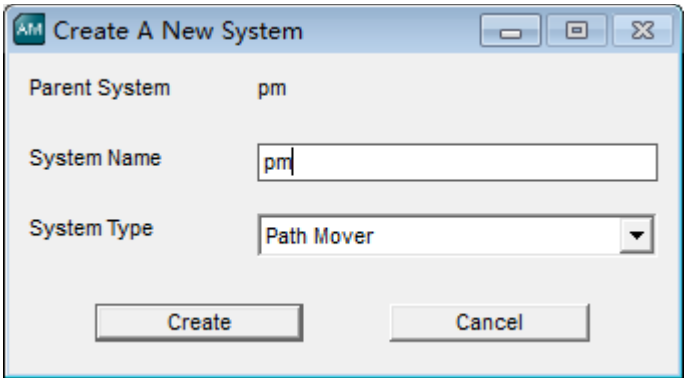
实验三详细解析

本实验学习的重点在于 Load、Resource、Queue 等常见实体的定义及其三维显示的设定以及对 AutoMod 中 pathmover 子系统的简单学习。通过该实验能利用 pathmover 进行简单的循径系统的设计。以下是完成本系统模型所需要的项目。

表 4-1 Pathmover 系统项目设置

Process系统		
项目	名称	备注
Process	P in	工件进入系统，经检验后出库
Loads	L a	工件a
	L b	工件b
Resources	R check	检验场
Queues	Q a	工件a进入系统前等待队列
	Q b	工件b进入系统前等待队列
	Q check	检查场工件a、b等待检查队列
	Q out	工件出库队列

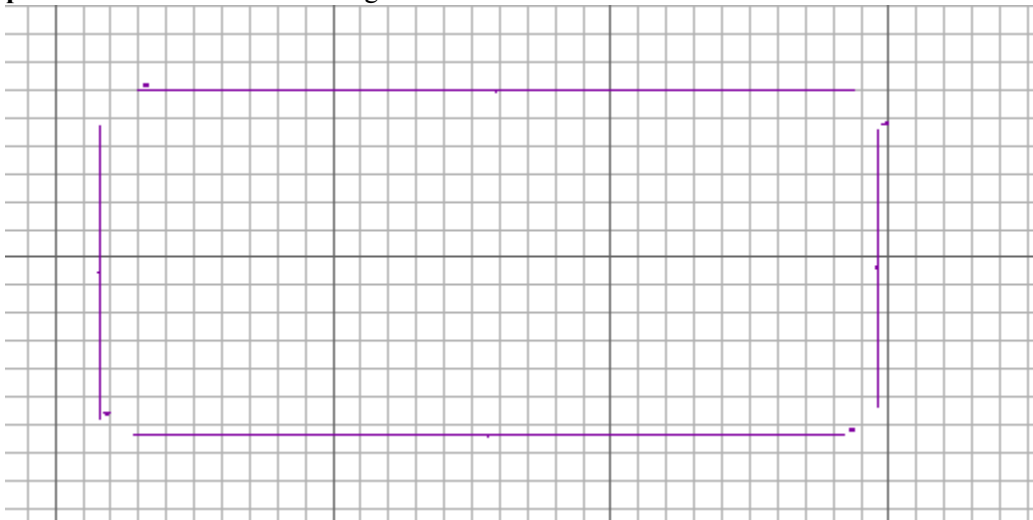
- 1.新建一个模型（pathmover）。修改单元格长度单位为米，并保存和导出模型。
- 2.点击 System 中的 New，新建 Path Mover（pm）子系统。



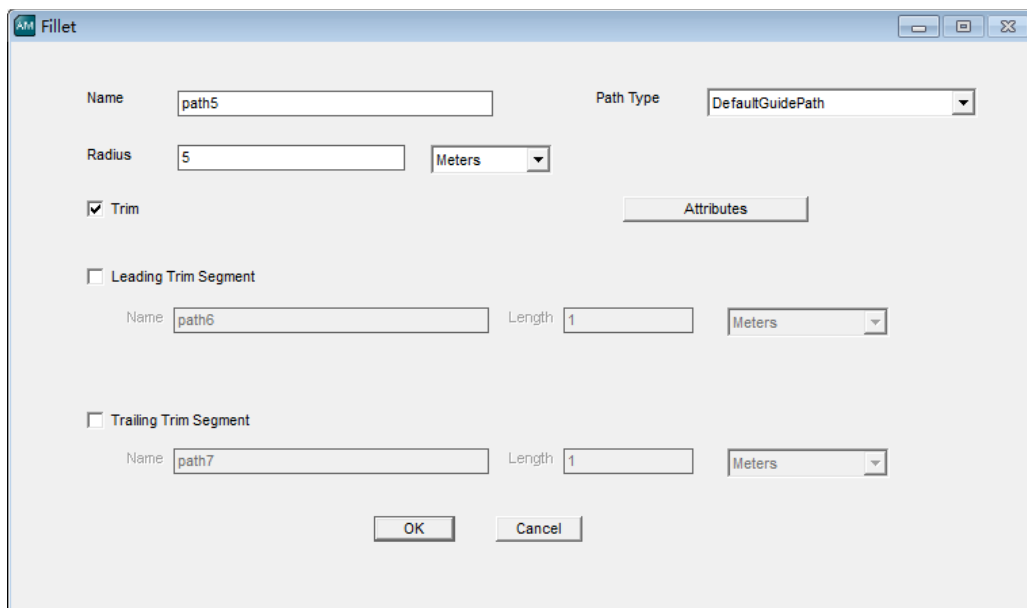
Create A New System 视窗

3.新建路径。

Step1 在出现的窗口中，点击 Sing Line，新建路径如下。



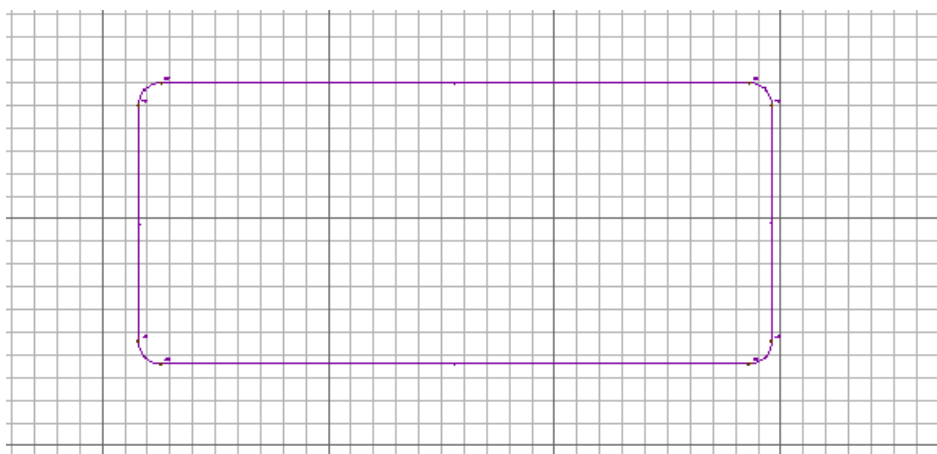
Step2 点击 Pathmover 视窗中的 Fillet，然后选中要连接的两条路径，将其连接起来。依次将所有要连接的路径连接。



Fillet 视窗

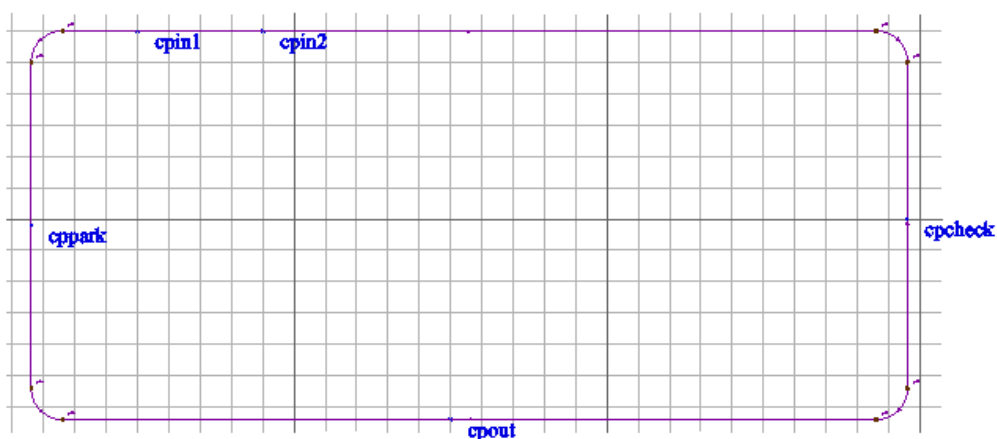
Step3 在上图的 Radius 编辑处，可以设置转角半径。

Step4 所有路径连接完成如下图。



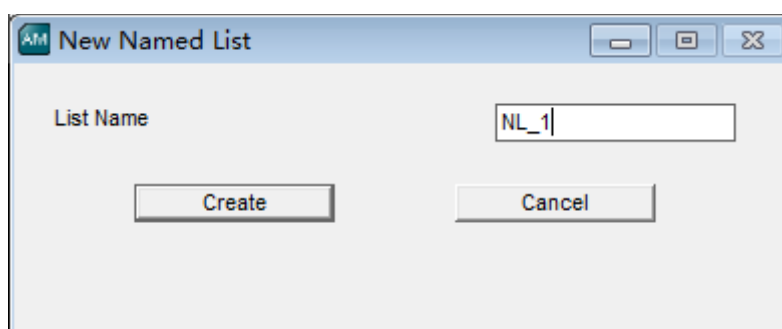
注：每选中两条要连接的路径，需设好转角半径之后，点击 OK 按钮，才能继续选中其他两条要连接的路径。

4.点击 Control Point，新建站点如下。



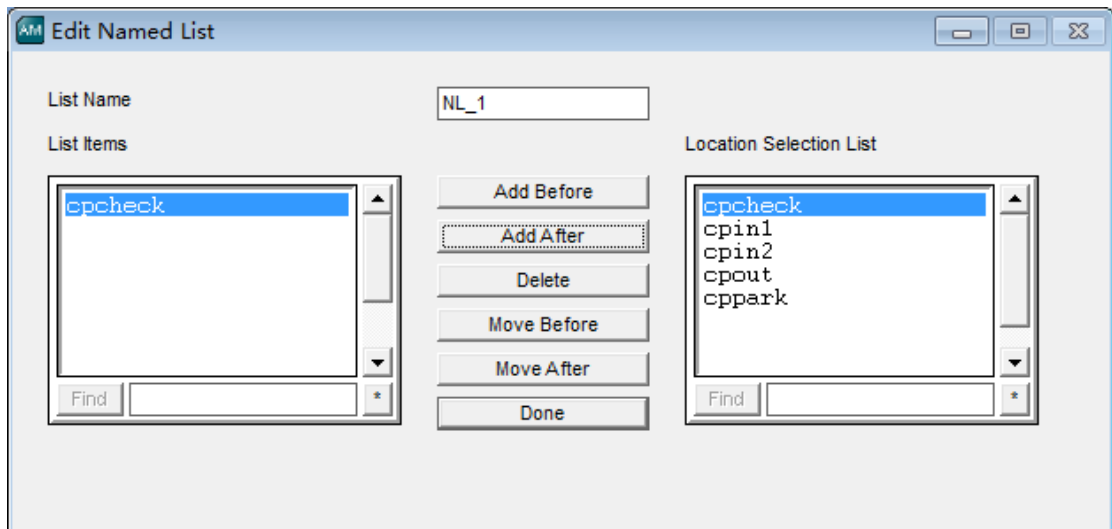
5.新建 Name List。

Step1 点击 Name List，新建一个名称列表。

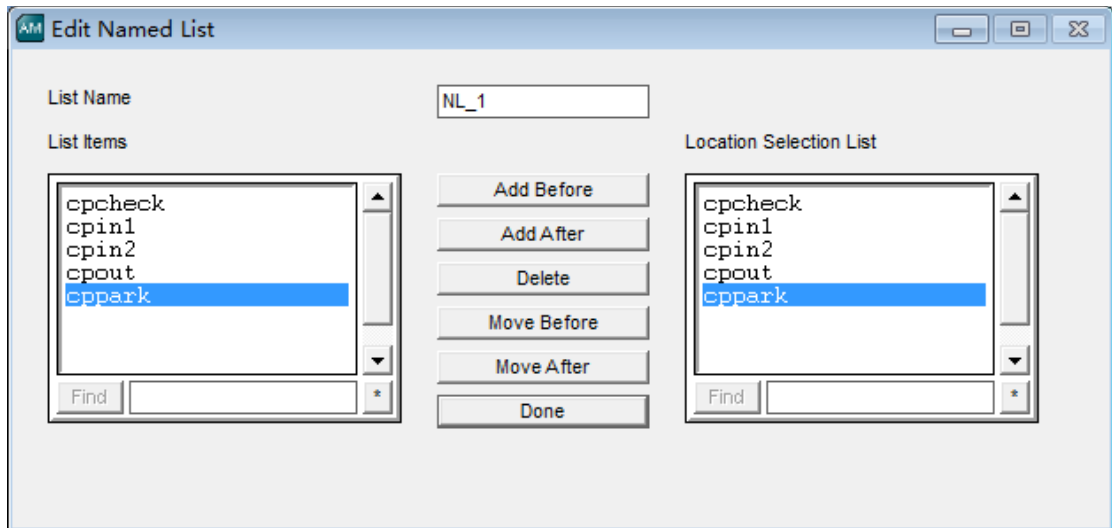


New Named List 视窗

Step2 将所有 Control Point 点都 Add After 到前面的 List Items 中。



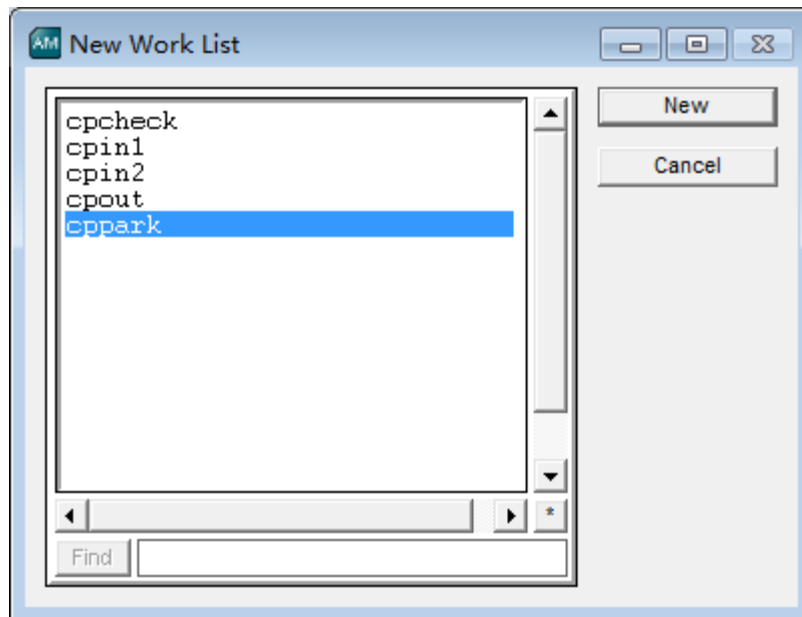
Eidt Named List 编辑视窗



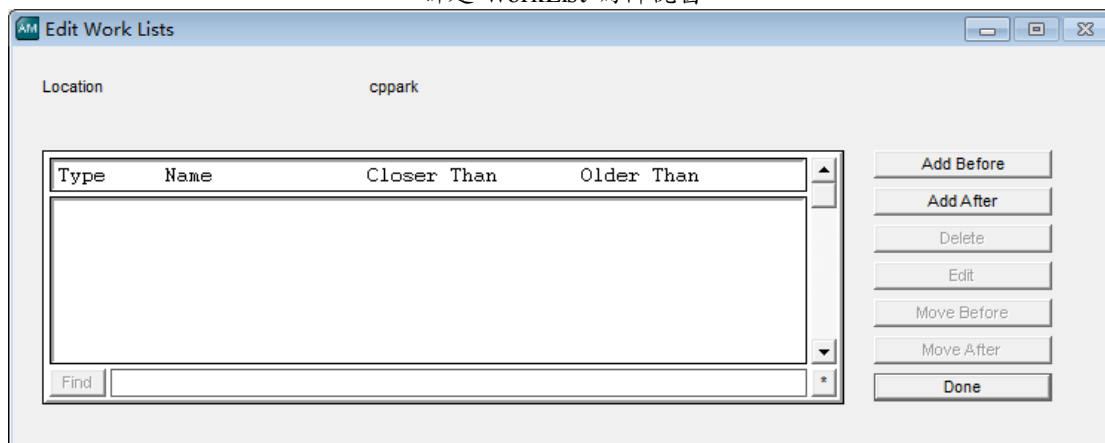
Eidt Named List 编辑视窗

6.新建 Work List 和 Park List。

Step1 新建 Work List， 点击 New

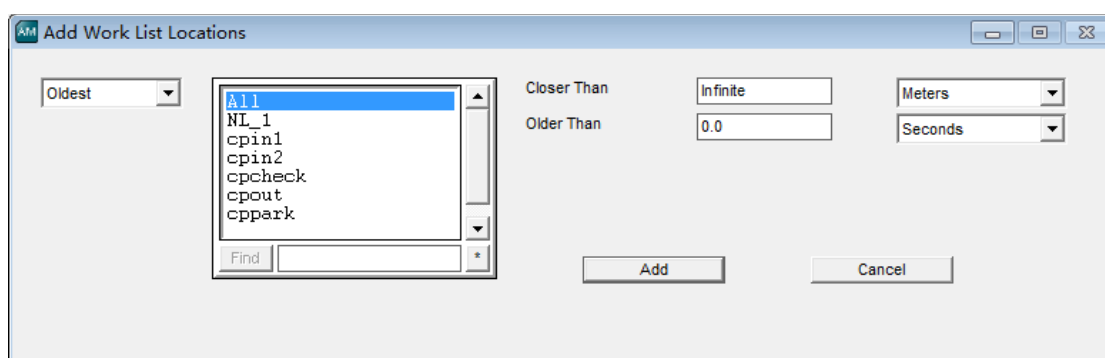


新建 WorkList 编辑视窗

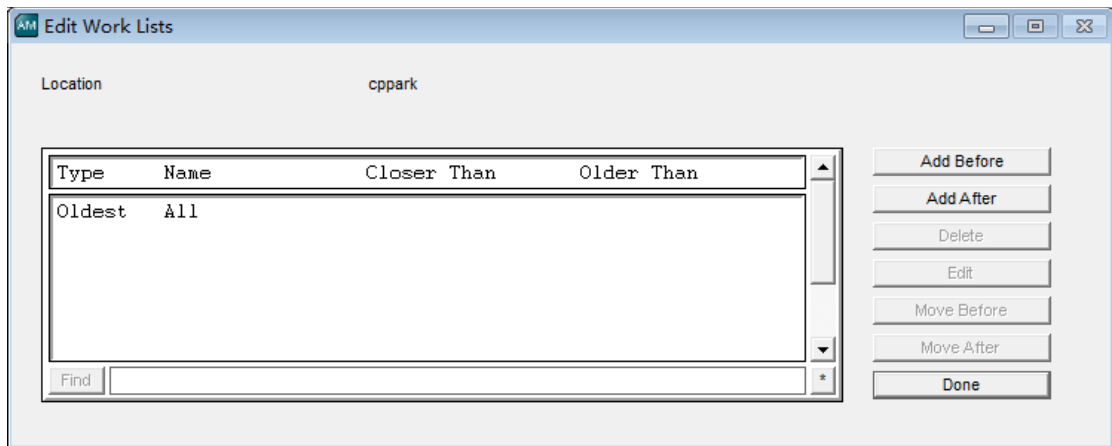


Work Lists 编辑视窗

Step2 点击 Add After，在左边下拉栏选 Oldest，中间选 All，点击 Add。

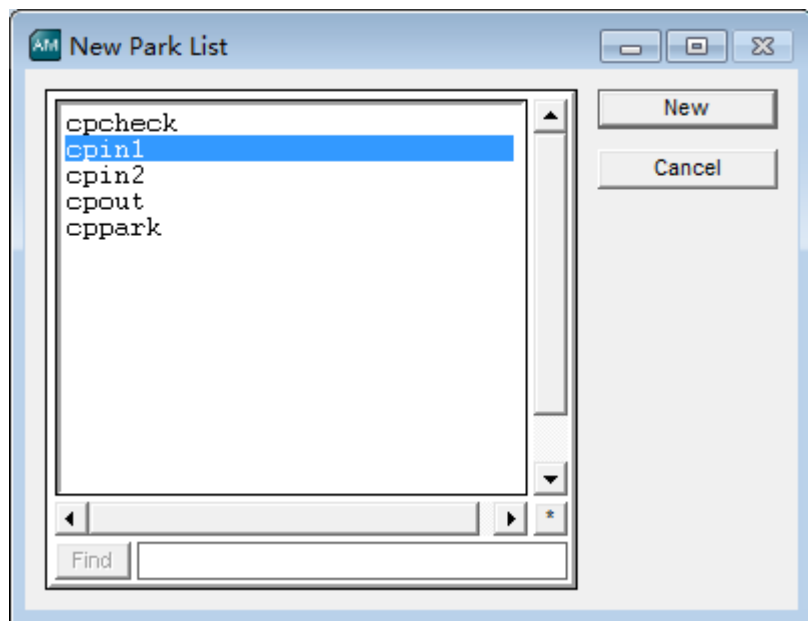


Add Work List Locations 视窗

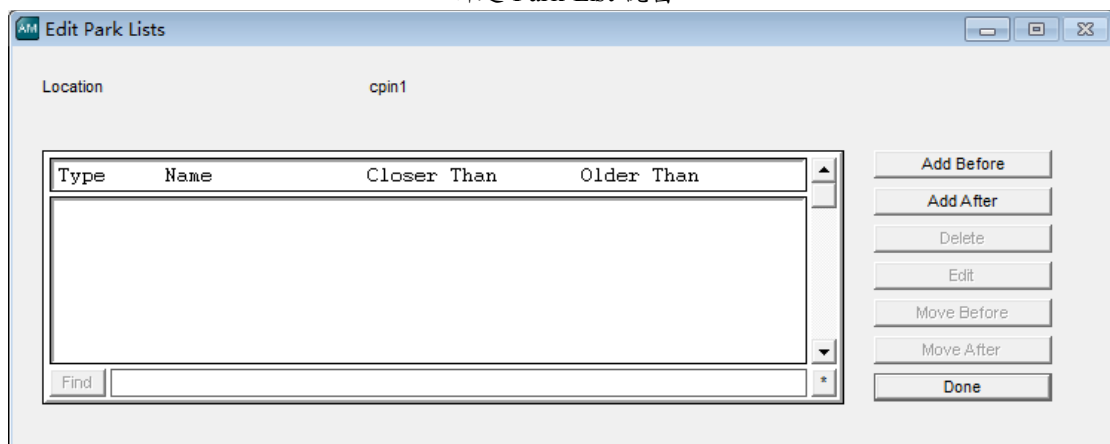


Work Lists 编辑视窗

Step3 新建 Park List，点 New。

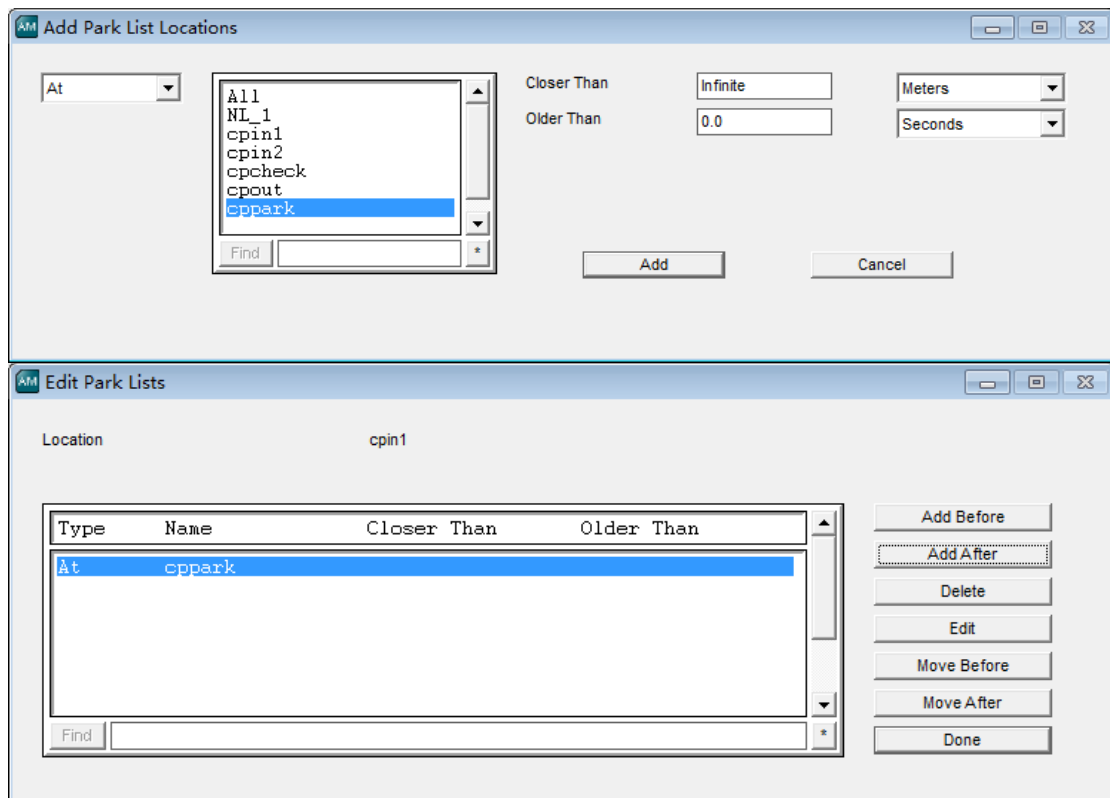


新建 Park List 视窗



Park Lists 编辑视窗

Step4 点击 Add After，左边为默认 At，右边为 cppark。点 Add。



Park Lists 编辑视窗

Step5 同样的，在 cpin2、cpcheck、cpout 处分别建立 Park List（At cppark）。

7.编辑 Segment，点 Done 编辑完成。

Edit A Segment Definition

Segment Type: DefSegment Edit Graphic

Segment Capacity(Loads): 1

Load Pick Up Time: 15 Seconds

Load Set Down Time: 15 Seconds

Connectors:

Front Length: 0.0 Meters Edit Graphic

Rear Length: 0.0 Meters Edit Graphic

Path Attach Points:

☒ Attach graphic to path attach point(s)

☒ Path Attach Point 1 Edit Graphic

☐ Path Attach Point 2 Edit Graphic

Containers:

Find *

New Edit Copy Delete Edit Graphic

Done

Segment Definition 编辑视窗

Edit Segment Graphics

Shape Orientation

Default Transformation: Current Undo Transformation: Current Picture Unit: Meters

1 X Scale

Rotate Translate Scale

X: 0 -38 1

Y: 0 39 1

Z: 0 0 1

☐ Scale All

Shape Definition: Import Inherit

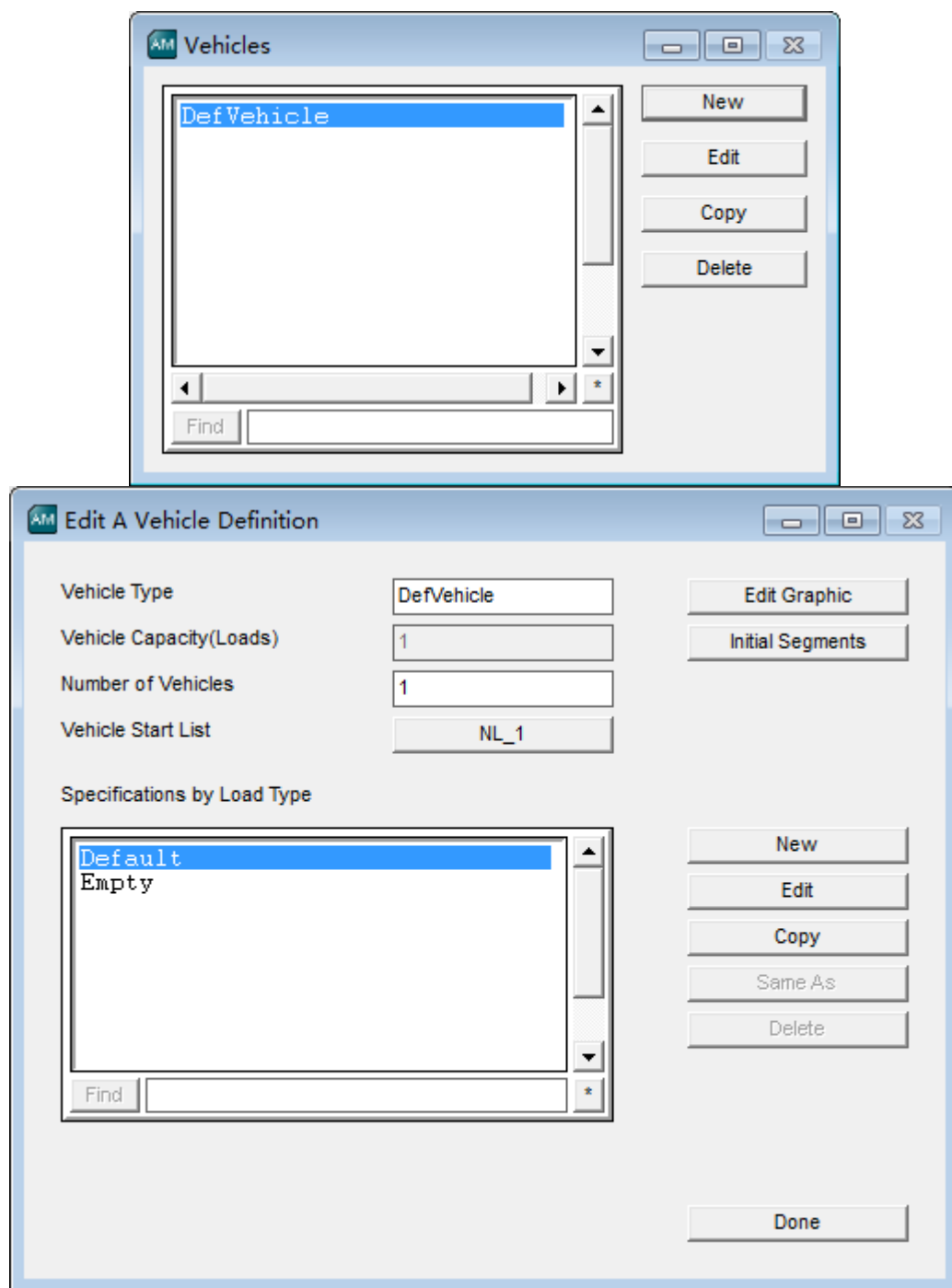
Load Stack: Default No Load Display

Place Move Hide Done

Segment Graphics 编辑视窗

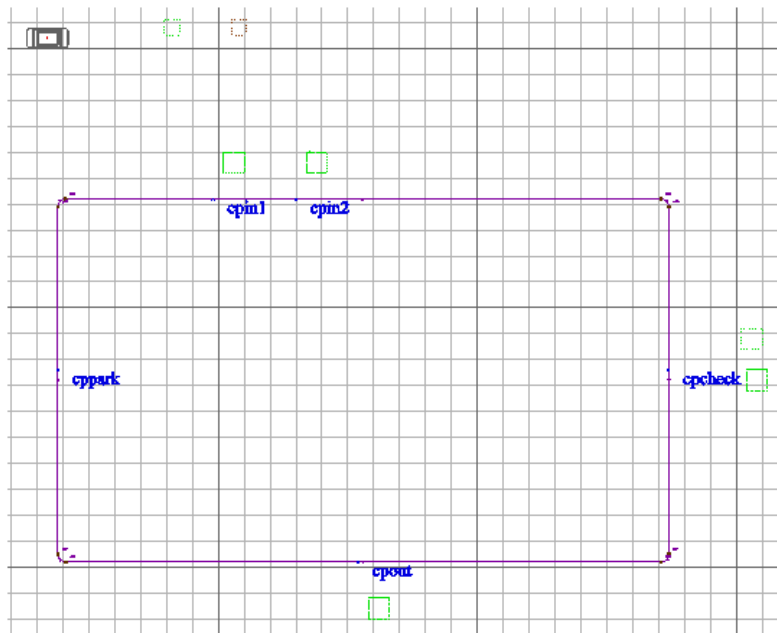
点击 place 按钮，放置车辆。从软件安装路径下的 demos/graphics/cell 下设置小车形式。

8.编辑 Vehicles



Vehicles 编辑视窗

9.按照表中项目设置新建 Loads、Resources、Process 等。



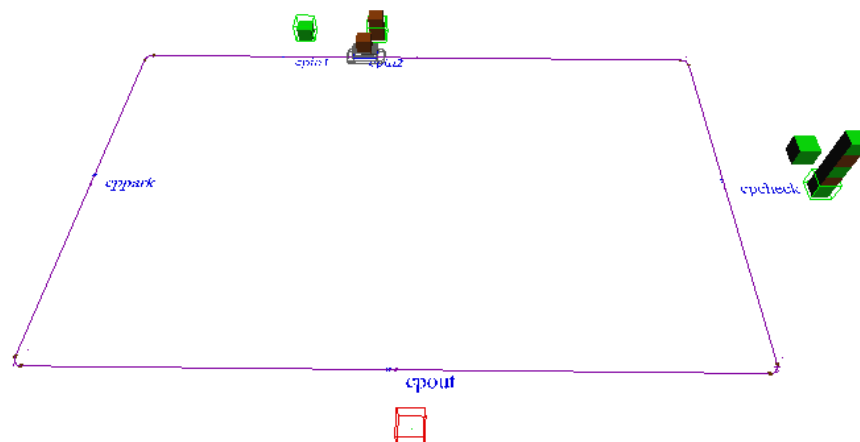
10.设置 Run Control 为 1 snaps，24hrs。

Run Control 编辑视窗

11.编辑 Source File 文件。

```
1begin P_in arriving
2  if load type = L_a then
3    begin
4      move into Q_a
5      move into pm.cpin1
6    end
7  else
8    begin
9      move into Q_b
10     move into pm.cpin2
11   end
12   travel to pm.cpcheck
13   move into Q_check
14   use R_check for uniform 20,4 min
15   move into pm.cpcheck
16   travel to pm.cpout
17   move into Q_out
18   wait for e 5 min
19end
20
```

12.运行模型。



13.查看输出报告。

```

*** AutoMod 12.6 ***
Model pm
Statistics at Absolute Clock = 1:00:00:00.00, Relative Clock = 1:00:00:00.00
CPU time: Absolute: 13.458 sec, Relative: 13.458 sec

Statistics for Process System "pm"

Process Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
P_in          250    195   95.49      --  195   0    --   33002.09

Process Traffic Limit Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
P_in          250    195   95.49 Infinite  195   0    --   33002.09

Queue Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
Space         250     0    0.00 Infinite    1    0    --     0.00
Q_a           108    30   15.66 Infinite   31    0    --  12525.49
Q_b           142    39   20.76 Infinite   40    0    --  12632.09
Q_check       181   126   58.60 Infinite  126    0    --  27973.87
Q_out         55     0    0.21 Infinite    9    0    --    330.55

Resource Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait  State
=====
R_check       71     1    0.99      1    1    0  0.990   1204.87  24875.97  Up  ----

Random Number Streams
Name          Total
=====
stream0       236
stream_L_a_1  146
stream_L_b_1  143
stream_R_check_1  0

Statistics for Path Mover System "pm.pm"

Guide Path Type Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
DefaultGuidePath  1515    1    1.03 Infinite    2    0    --     58.53

Guide Path Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
path1         189     0    0.32 Infinite    1    0    --   144.45
path2         190     0    0.20 Infinite    1    0    --    91.69
path3         190     1    0.30 Infinite    1    0    --   135.45
path4         189     0    0.17 Infinite    1    0    --    79.87
path5         189     0    0.01 Infinite    1    0    --     4.12
path6         189     0    0.01 Infinite    1    0    --     4.12
path7         190     0    0.01 Infinite    1    0    --     4.12
path8         189     0    0.01 Infinite    1    0    --     4.12

Control Point Type Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
DefaultControlPoint  947    1    1.01 Infinite    2    0    --     91.85

Control Point Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
cpin1         189     0    0.17 Infinite    1    0    --    76.49
cpin2         189     0    0.06 Infinite    1    0    --    27.99
cpcheck       190     0    0.29 Infinite    1    0    --   130.72
cpout         190     1    0.25 Infinite    1    0    --   115.37
cppark        189     0    0.24 Infinite    1    0    --   108.33

Vehicle Statistics
Delivering      Retrieving      Going To Park      Parking
Percent      Average Percent      Average Percent      Average Percent
Of Total      Time/      Of Total      Time/      Of Total      Time/      Of Total
Time      Made Trip      Time      Made Trip      Time      Made Trip      Time
=====

```

```

=====
DefVehicle:
0.302 236 110.60 0.239 236 87.68 0.454 190 206.31 0.005
All DefVehicle:
0.302 236 110.60 0.239 236 87.68 0.454 190 206.31 0.005

The Average Capacity Lost Due To Congestion is 0.0 Percent
Of Total Capacity Which is 0.0 Vehicles

Vehicle Segment Statistics
=====
Vehicle          Total      Cur      Average Capacity      Max      Min
=====
DefVehicle:
DefVehicle(1)          1        1        1.00 Infinite          1        0

Current Status Of Vehicles
=====
Vehicle          Status          Time In Current      Destination      Load On or
Cur Status Location      Location      Claimed By Veh
=====
DefVehicle(1)      Move          60.52 cpcheck      cppark      Empty

Current Status Summary
=====
Type          Average Time      Maximum Time      Minimum Time
In Cur Status      In Cur Status      In Cur Status
=====
DefVehicle          60.52          60.52          60.52

```

从报告中，我们可以看出，运行 24 小时内，系统中的 load 共 250 个，其中经队列 Q_a 的有 108 个，经列 Q_b 的有 142 个，经 Q_check 的有 181 个；运行 8 小时后，队列 Q_a 处的 load 有 30 个，队列 Q_b 处的 load 有 39 个，队列 Q_check 处的 load 有 126 个。由此，我们可以看出，队列 Q_check 处堆积了很多 load。说明一辆负载容量为 1 的小车不能满足系统的需求，我们可以更改小车的容量或小车的数量，来改进模型，并进行对比，进而得到最佳方案。