

MVC 模式的电梯群控仿真系统

钟 灵, 章 云, 王 戈, 杨湛鸿

ZHONG Ling, ZHANG Yun, WANG Ge, YANG Zhan-hong

广东工业大学 自动化学院, 广州 510006

Automation Faculty, Guangdong University of Technology, Guangzhou 510006, China

ZHONG Ling, ZHANG Yun, WANG Ge, et al. Elevator group control simulation system based on MVC. *Computer Engineering and Applications*, 2009, 45(32): 197-199.

Abstract: To evaluate the effectiveness of dispatch algorithms of elevator group control system, a new elevator group simulation system is presented based on the MVC design pattern. Non-stationary Poisson process is utilized to model the passengers' traffic in the system. The extensive experiments show that the system has the expected interaction of passengers and elevators, and simulates the real change under the minimum long dispatch algorithm.

Key words: elevator group control system; non-stationary Poisson process; Model/View/Controller (MVC)

摘 要: 为提供电梯群控调度算法的测试平台, 设计了基于 MVC (Model/View/Controller) 模式的电梯群控仿真系统, 同时引入了非平稳泊松过程来描述各楼层人流交通变化。通过实验证明该系统能够模拟现实的人/梯的交互, 在最小最长调度算法下符合设计预期。

关键词: 电梯群控系统; 非平稳泊松过程; 模式/视图/控制器

DOI: 10.3778/j.issn.1002-8331.2009.32.062 **文章编号:** 1002-8331(2009)32-0197-03 **文献标识码:** A **中图分类号:** TP18

1 引言

电梯作为高层建筑的垂直交通工具在现代社会日益普遍, 其运行效率一直以来都是各电梯厂商研发的重点内容。以多台电梯来统一协调服务楼内乘客的电梯群控系统 (Elevator Group Control System, EGCS) 是目前采用的最常见的多电梯系统方案^[1]。电梯群控系统的调度是一个复杂的、具有非线性和不确定性的多目标随机决策问题^[2]。由于该问题的复杂性, 近年来该问题也渐渐成为各人工智能优化算法效率的测试基准问题之一^[4-8]。在兼顾各个目标指数的情况下, 设计好群控调度算法是电梯群控问题的关键技术。但由于各类群控调度算法难以在真实的场景中测试, 设计和实现具有较好真实性的仿真系统就是此类研究工作的基础^[9]。

介绍了基于 MVC (Model/View/Controller) 设计模式结构的电梯群控仿真系统的设计方法。第 2 章介绍了基于非平稳的泊松过程 (Non-stationary Poisson Process, NSPP) 的建筑内人员流量变化的情况和系统中人/梯的状态转化模型, 第 3 章给出了基于 MVC 设计模式的仿真系统程序结构及 UML 的类图, 第 4 章介绍了经典的群控算法-最小最长候梯算法, 最后通过仿真试验验证了系统的有效性。

2 系统模型

多数仿真系统都采用离散事件驱动 (discrete event-driven)

或时间驱动 (time-driven) 来设计实现。使用事件驱动仿真时, 系统中每个对象单元引发的事件都必须单独处理, 在仿真过程中必须跟踪由这些数据单元引发的所有事件, 因此不能在仿真的速度上和精度上取得平衡; 在电梯仿真系统运行中出现大量人/梯对象的目标事件给抢占等情况, 因此需要相当复杂的仿真事件处理机制。相比之下, 采用时间驱动仿真的原理是将时间离散化, 在时间离散点处理和记录被仿真对象的状态的变化, 这种方法不要对模型进行近似, 因而比较准确和易于实现。鉴于以上所述, 系统采用以时间驱动来仿真。其中在事件推进过程中, 关键的一点就是要处理好人流来往的交通模型和人/梯对象转移的模型。

2.1 人流模型

在目前常见的电梯仿真系统中, 通常使用平稳泊松过程来模拟人员到达或离开大楼的事件^[9]。若 τ_{i+1} 和 τ_i 分别是第 $i+1$ 个和第 i 个人到达/离开的时间, 在该随机过程中 $P(\tau_{i+1}-\tau_i)$ 的概率为:

$$P(\tau_{i+1}-\tau_i) = \begin{cases} 0 & t < 0 \\ 1 - e^{-\lambda t} & t \geq 0 \end{cases} \quad (1)$$

其中, $1/\lambda$ 是人员的到达/离开间隔均值。显然在该模型下 τ_{i+1} 和 τ_i 之间到达分布只与区间长度 t 有关, 与上事件发生时间 τ_i 无关。可由公式 (1) 求得下一个人员的到达/离开时间 τ_{i+1} 为:

$$\tau_{i+1} = \tau_i + t = \tau_i - \frac{\ln(1-p)}{\lambda} \quad (2)$$

基金项目: 国家-广东联合基金重点项目 (the Joint Funds of NSFC-Guangdong of China under Grant No.U0735003); 广东省自然科学基金团队项目 (the Natural Science Foundation of Guangdong Province under Grant No.8351009001000002)。

作者简介: 钟灵 (1978-), 男, 博士研究生, 讲师, 研究内容: 模式识别、优化理论、光学检测系统及算法; 章云 (1963-), 男, 博士生导师, 教授, 研究内容: 智能信号处理、复杂系统控制、模式识别等; 王戈 (1987-), 男, 网络与信息 2006 级本科生; 杨湛鸿 (1988-), 男, 网络与信息 2006 级本科生。

收稿日期: 2009-06-02 **修回日期:** 2009-08-12

由上可知, τ_{i+1} 可分为三个步骤求解: 首先产生(0~1)之间的伪随机数 p , 然后利用公式(2)计算出人员到达/离开时间间隔 t , 最后加上一次同类事件发生时间 τ_i 可求 τ_{i+1} 。

系统将非平稳泊松过程引入交通人流量仿真中^[3], 采用非平稳泊松过程可根据更真实地模拟每天楼内人流变化。典型的可以将每天楼内变化分为、上班高峰模式和层间互访模式。例如在上班高峰模式, 人流是不断增加, 电梯群系统有可能发生满载, 而乘客满意度不断下降的情况。此时, λ 设为随时间变化的函数 $\lambda(t)$:

$$t = \tau_{i+1} - \tau_i = -\frac{\ln(1-p)}{\lambda} \quad (3)$$

2.2 人/梯对象状态

系统是按时间刷新系统状态的。每台电梯每秒刷新状态一次。状态刷新是根据电梯的当前状态, 电梯所在的位置以及厅外召唤和内部目标服务队列综合判断, 决定电梯下一步该转换成哪种状态, 具体状态转移如图 1 所示。

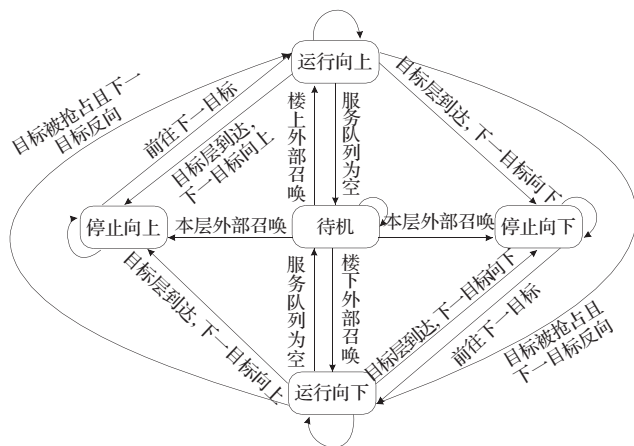


图1 电梯状态转移图

与电梯交互的人共有 4 个状态: 乘梯状态、大楼中停留状态、处于向上等待队列和处于向下等待队列状态。当人要发生楼层转换的时候, 就转到相应的等待队列状态。当有前往目标方向的电梯到来时, 就转入了乘梯状态。在乘梯状态中, 当目标层到达时, 就转入楼中停留状态, 如图 2 所示。

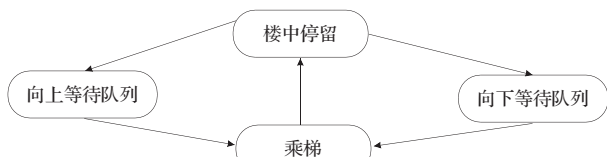


图2 人的状态转移

3 MVC 设计模式

该电梯仿真群控系统是面向对象的设计思想, 并通过 Visual C# 实现了对实际电梯系统运行的仿真。Microsoft Visual C# 2005 是微软开发的 C# 编程集成开发环境, 它是为生成在 .NET Framework 上运行的多种应用程序而设计的。利用 C# 的面向对象编程技术, 可以方便快速地描述电梯系统的行为, 并对电梯群控系统仿真, 程序主界面如图 3 所示。

虽然近年来面向对象的方法已经广泛应用在各类仿真系统的设计和实现中^[8], 但如何设计出具有通用结构和复用性好的仿真软件系统还未有大量文献。该系统采用了模型/视图/控



图3 系统主界面

制器三元组的设计方法。在 MVC 系统中, 模型是对应用对象, 在该系统对应到 Person 类组合成的各层人员队列类 Wait-List 和电梯群类 EleList; 视图对应的是模型在屏幕上的表示, 在该系统对应到主界面 MainForm; 控制器定义用户界面或数据内容输入的响应方式, 该系统中对应时间在时间驱动下, 对数据的控制类 Controller, 它用于将各类对象模型关联后更新到视图上去。

在程序设计上, 系统采用面向对象的设计方法。根据电梯系统的逻辑和面向对象方法, 抽象出的主要对象有: 乘客、电梯、电梯控制器、等待队列。各个类的设计如下: Person(乘客类): 该类的属性包括乘客 ID(在该系统中每个乘客 ID 都是唯一)用来标示乘客、乘客的当前所在层 curFloor、目的层 destFloor、进入某个状态的开始时间 startTime(该属性用来统计乘客在某个状态的时间)。Esystem 类: 该类用来记录系统的基本参数, 其属性有仿真系统的绝对时间 absolutime、系统中电梯数目 eleCount、楼层数目 floorCount 等等。Elevator(电梯类): 该类属性有电梯 ID 号、电梯当前状态 status、电梯目的地链表 destList 和梯内乘客链表 personList(链表是使用 C# 里的泛型类 List<>来实现), 链表的节点的数据类型为 DestNode 和 Person。EleController(电梯控制类): 封装了电梯控制器的调度算法, 改变电梯的状态参数等操作, 只提供一个 NewTimeCome 的函数给外部调用。Wait-List(等待队列类): 用来封装等待队列, 模拟现实中的排队, 先到先服务, 使用 C# 里的泛型类 List<>来实现。具体 UML(United Modeling Language)类图如图 4 所示。

4 群控调度算法

目前已经有大量研究者提出了各类智能的电梯调度算法, 但在实验电梯群系统还主要以最小最长候梯算法和区域候梯算法为主。在群控电梯系统中, 当厅外召唤信号产生时, 最小最长候梯算法预算每台电梯的每个厅外召唤的候梯时间。每台电梯取最大的候梯时间作为该电梯的长候梯时间。然后比较每台电梯的长候梯时间, 选取长候梯时间最小的那台电梯作为当前厅外召唤的服务电梯。

$$k^* = \arg \min \varphi_k \quad (4)$$

$$\varphi_k = \max(TW^n(i_x) + TS^n(K, i_x)) \quad (5)$$

其中, k 为各电梯编号, k^* 为最后选定电梯编号, i_x 为厅外 i 层方向为 x 的呼叫, 而 φ_k 为电梯 k 的最小最长候梯时间, $TW^n(i_x)$ 为当 i_x 被指派某电梯后到第 n 时刻的间隔, $TS^n(K, i_x)$ 为第 n 时刻预计电梯从所处位置运行到楼层 i 所花费时间, 该时间同电梯运行方向、电梯到楼层 i 的相对距离、以及所需的停靠次数等密切相关。显然该算法考虑的是假设将新厅外召唤 j 加入每台电梯后, 选择候梯时间排序最小的电梯。

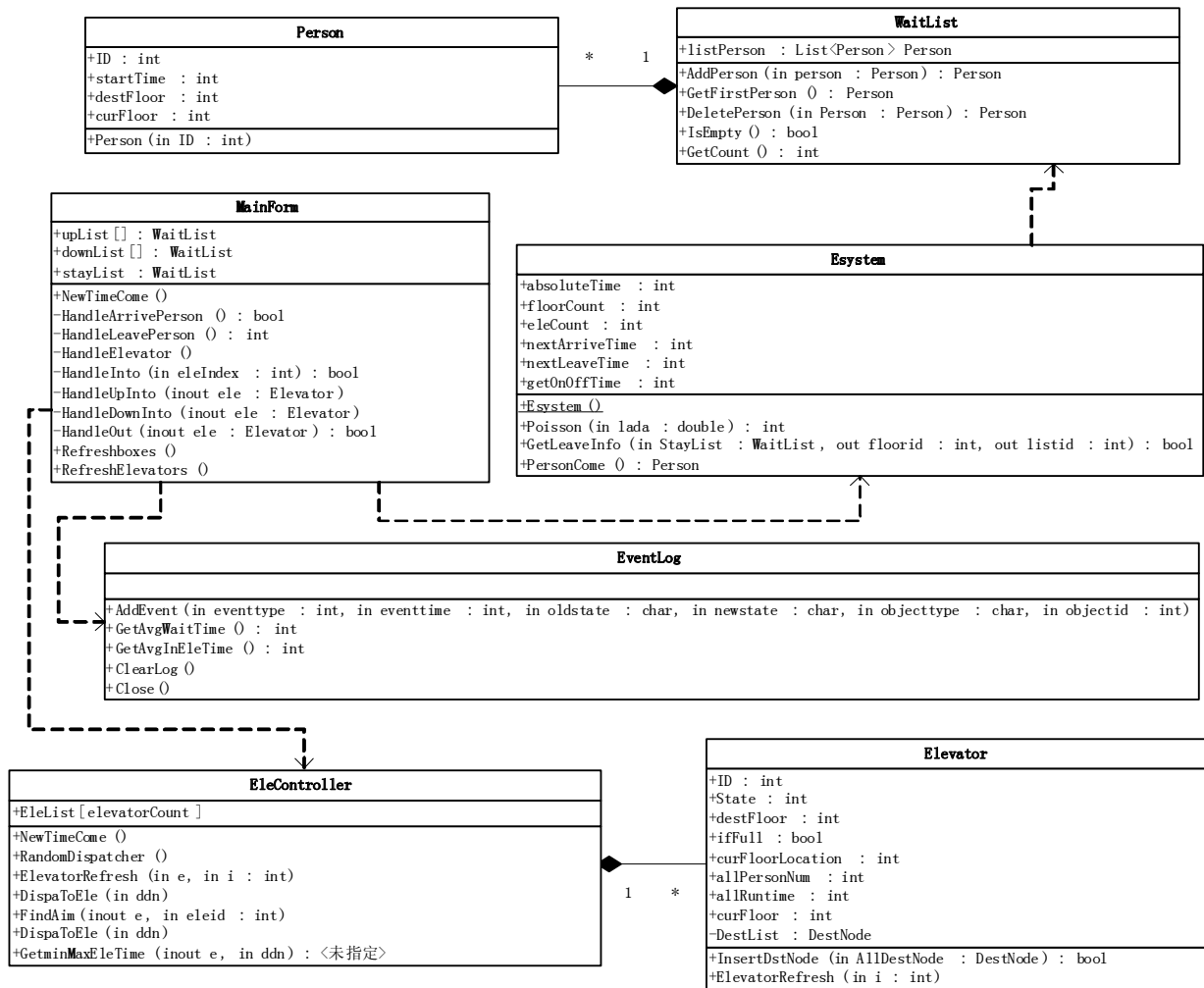


图 4 系统主要类图

5 实验仿真

为评价仿真系统的实际效果,该文通过设计的系列实验给予评价。其中,采用的两个评价指标乘客平均候梯时间和电梯运行效率,其定义如下为:

$$\text{平均候梯时间} = \frac{\text{乘客总等待时间}}{\text{乘客总等待人次}} \quad (6)$$

$$\text{电梯运行效率} = \frac{\text{电梯总服务乘客人次}}{\text{电梯总运行时间}} \quad (7)$$

仿真测试结果均以 10 次均值为结果,电梯群台数最大为 8 台,采用运行 100 s 虚拟仿真时间完成的。仿真系统参数如表 1。

表 1 仿真实验假设参数

楼高最高层	电梯数量	单梯最大载客数	电梯速度/(层·s ⁻¹)	单人上下电梯时间/s
25	8	10	1	1

通过改变不同的到达密度,可以观测出在最小最长候梯算法下的两个评价指标的变化情况。从图 5 可看出,当达到人流密度不断增大时,平均候梯时间也不断增大。当人流期望密度小于 1 人/s 时,可以看出这时候候梯时间的增加速度还比较缓。通过仿真系统的界面观察可知,此时多数从底层出发的电梯都未满载,所以此时的运行效率增加也较快。但当人流密度超过 1 人/s 时,电梯运行效率并未明显增大,而此时平均的候梯

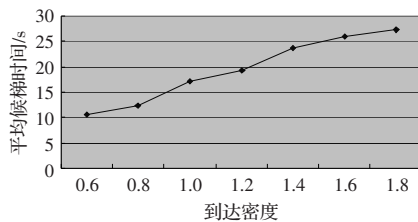


图 5 平均候梯时间与到达密度图

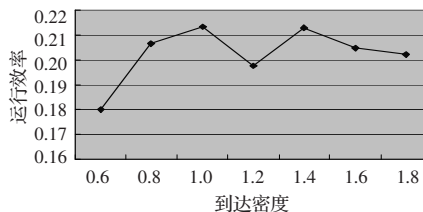


图 6 运行效率与到达密度图

时间快速增加,可预期顾客满意程度会较快下降。可以看出,该系统的仿真结果具有较好的真实性,符合实验预期。

6 结论

该文采用了 MVC 设计模式结构开发出了一套复杂电梯群