计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210300050

基于吸收态马尔可夫链的智能无人车系统实时性能分析

吴培培[1]　吴兆贤[1]　唐文兵[2]
1 浙江理工大学信息学院　杭州 310018
2 华东师范大学软件工程学院　上海 200062

摘　要　随着人工智能技术的进步和人机物融合系统的发展,智能无人车系统成为了新一代人工智能研究的前沿。智能无人车系统根据车辆和环境数据进行实时决策以控制无人车运行,因而该系统具有较高的实时性能要求,对系统实时性的分析是保障系统安全可靠的方法之一。为了对智能无人车系统的实时性能进行分析,以智能无人车变道系统为例,首先使用 MARTE 模型对智能无人车变道系统进行建模,在系统设计早期就引入性能需求参数;然后,通过模型转换将 MARTE 模型转化为吸收态马尔可夫链;最后,利用吸收态马尔可夫链的相关理论和公式综合估算了智能无人车系统的实时性能指标,并针对影响整个系统实时性的关键模块进行了分析。实验结果表明,文中所提模型和分析方法可以较好地分析智能无人车系统的实时性能。分析发现系统中智能模块的准确率与响应时间相互制约,在不同的运行场景下需要找到二者之间的平衡点以获得更优的实时性能。

关键词:智能无人车系统;MARTE 模型;吸收态马尔可夫链;实时性能分析
中图法分类号　TP242.6

Real-time Performance Analysis of Intelligent Unmanned Vehicle System Based on Absorbing Markov Chain

WU Pei-pei[1], WU Zhao-xian[1] and TANG Wen-bing[2]
1 School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China
2 Software Engineering Institute, East China Normal University, Shanghai 200062, China

Abstract　With the advancements of artificial intelligence technology and the development of human-cyber-physical systems, intelligent unmanned vehicle systems are becoming the forefront of the new generation of artificial intelligence research. The intelligent unmanned vehicle system performs real-time decision based on vehicle and environmental data to control the unmanned vehicle. Therefore, the intelligent unmanned vehicle system has high real-time performance requirements. Analysis of the real-time performance of the system is one of the methods to ensure the safety and reliability of this kind of system. In order to analyze the real-time performance of the intelligent unmanned vehicle system, this paper takes the intelligent unmanned vehicle lane changing system as a scenario. First, the MARTE model is used to model the intelligent unmanned vehicle lane changing system, and the performance requirements parameters are added in the early system design. Then, through model transformation, the MARTE model is transformed into an absorption Markov chain. Finally, the relevant theories and formulas of the absorption Markov chain are used to comprehensively estimate the real-time performance indicators of the intelligent unmanned vehicle system, and analyze the key modules that affect the real-time performance of the entire system. The experimental results show that the model and analysis method proposed in the article can better analyze the real-time performance of the intelligent unmanned vehicle system. The analysis found that the accuracy and response time of the intelligent modules in the system restrict each other, and it is necessary to find a balance between the two in different operating scenarios to obtain better real-time performance.

Keywords　Intelligent unmanned vehicle system, MARTE model, Absorbing Markov chain, Real-time performance analysis

1　引言

智能无人系统是人工智能的重要应用之一,是集计算机、软件、机械、控制以及材料等多学科技术于一体的复杂系统。随着深度学习等人工智能技术的进步和人机物融合的发展,以智能无人车、服务机器人为代表的智能无人系统已被应用到无人驾驶、管道巡检、交通运输、智能测控等领域[1]。其中,智能无人车系统是解决交通安全、道路拥堵、环境污染等问题的重要技术途径之一。

智能无人车系统往往运行在动态不确定环境中,无人车

基金项目:国家自然科学基金项目(61210004,61170015)
This work was supported by the National Natural Science Foundation of China(61210004,61170015).
通信作者:吴培培(peipeiwu@126.com)

需要根据环境和自身传感器信息进行实时反应，如减速、加速、变道等。准确及时地检测出行车周围的各种障碍物，并进行及时的反应，是保证智能无人车系统安全性的关键因素。因此，实时性是保证无人车系统正确、安全、稳定、可靠、可信的基本要求，对智能无人车系统进行实时性能分析很有必要。但是面对多样化的需求环境和不断丰富的软硬件资源，市场上出现了应用于不同场景的智能无人车系统，并且智能模块的加入更是给无人车系统带来了许多不确定因素，这在一定程度上给系统的实时性能分析带来了新的困难。

传统的基于测量的性能分析方法通常在开发的最后阶段才考虑性能需求，即所谓的“过后调整”策略[2]，它已经不能满足智能无人车系统对实时性等方面的要求。因此，可以在开发过程的早期，利用模型引入对系统性能需求的考虑，使得性能信息“内建”于系统，以便尽早发现并解决潜在的性能问题，提高开发效率并降低成本[3-4]，此方法为智能无人车系统的开发、测试等方面提供了有效手段。

UML(Unified Modeling Language)作为面向对象建模语言，可以从多个角度描述系统的结构和行为特征。SysML 常用于各种复杂系统的分析、设计以及验证等。然而，UML 与 SysML 都缺少用于对开发系统进行时间约束的建模结构[5]。MARTE(Modeling and Analysis for Real-Time Embedded systems)规范[6]作为 UML 的扩展，为实时嵌入式系统提供了非功能需求方面的有效建模与分析机制。如文献[7]提出了利用 MARTE 模型对自主机器人系统的非功能需求进行建模与分析；而且 MARTE 中的时间包是建模实时行为的主要规范，其定义了时间访问、时间使用以及其他必要的时间建模元素与方法。如文献[8]针对复杂系统的实时性问题，提出了将 MARTE 的时钟语义信息加入到 UML 模型元素中用以描述时间约束信息。但 MARTE 缺乏精确的语义，难以直接对其进行性能分析，因此可将 MARTE 模型转换为不同的分析模型进行性能分析。

目前，基于模型的系统性能分析方法[9-10]已有一些研究工作。部分基于排队网络(Queuing Network，QN)的方法在 UML 模型中使用自定义的性能标记来描述系统行为，从中直接导出 QN 模型[11-13]进行性能分析。但 QN 模型的描述能力还有不足，也难以集成到系统开发过程中[2]。随机 Petri 网(Stochastic Petri Net，SPN)是用于描述离散事件动态系统的形式化工具，在对系统进行分析时，还可以被扩展为广义随机 Petri 网[14-15]。随机进程代数(Stochastic Process Algebra，SPA)是另一种强有力的形式化工具，其表达式可以比 SPN 更好地体现模型结构层次关系。文献[16]提出了使用性能评估进程代数(Performance Evaluation Process Algebra，PEPA)网来建立性能模型，以及将 PEPA 网转换为 PEPA 描述。PEPA 网结合了 SPN 和 SPA 的优点，能够对移动性和状态改变进行更好的描述。但 PEPA 对复杂系统的描述较为繁琐，并且模型复杂时可能会引起状态空间爆炸[17]。马尔可夫理论在智能无人车系统研究领域应用十分广泛，在智能无人车完成路况识别、道路检测和跟踪以及车辆避障等各项任务时，都可以利用马尔可夫模型进行辅助设计[18]。通常马尔可夫过程也被作为一种数学模型应用于许多实际系统的建模和分析中[19-21]。

本文利用吸收态马尔可夫链(Absorbing Markov Chain，AMC)对智能无人车系统的实时性能进行分析。文中以基于 LSTM 的智能无人车变道系统为例，首先通过 UML 模型对该系统进行建模，并利用 MARTE 规范添加相关实时性能信息，然后将其转换为 AMC，最后基于 AMC 计算节点状态访问次数期望值和系统各模块的响应时间，以此分析系统的实时性能。此方法考虑到了智能无人车系统执行过程中的不确定性，将系统的整个执行过程细分为若干节点，通过对单个节点状态时间变化的分析，找到影响系统实时性的关键模块。实验结果表明，本文所提方法可以有效地分析智能无人车系统的实时性能，并指出智能模块对系统的实时性能影响较大；在不同运行场景下，可选择训练程度适中的智能模块，平衡准确率与响应时间，以此改善智能无人车系统的实时性能。

本文第 2 节介绍基于 LSTM 的智能无人车变道系统架构，并利用 MARTE 模型对其建模；第 3 节介绍 AMC 的基本性质和公式，并描述了由 MARTE 模型到 AMC 的模型转换；第 4 节给出了基于 AMC 的实时性能分析方法；第 5 节给出实验结果与分析；最后总结全文并展望未来。

2 基于 MARTE 的智能无人车系统建模

本节首先介绍智能无人车变道系统的架构，然后利用 MARTE 对其进行建模。

2.1 基于 LSTM 的智能无人车变道系统

智能无人车变道系统主要包括环境感知、决策规划和控制执行等模块，其中决策控制模块是系统的核心，要根据自身车辆状态和周围动态的环境信息自主地做出实时决策，控制车辆实现变道行为。图 1 给出了智能无人车变道系统的架构。

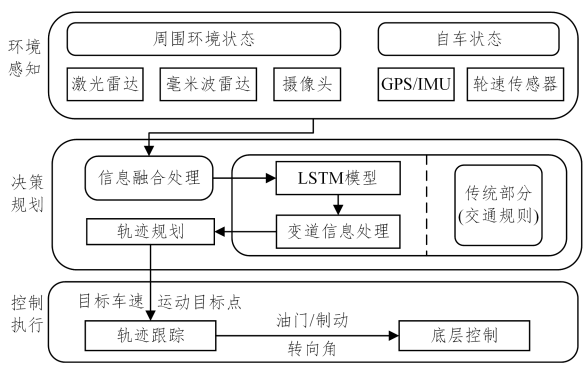


图 1 智能无人车变道系统的架构

Fig. 1 Architecture of intelligent unmanned vehicle lane changing system

智能无人车变道系统处理流程为：首先，感知模块融合多个传感器获取周围车辆信息以及自身车辆状态信息，为决策模块提供周围环境依据；接着，数据流向行为决策模块，如基于长短期记忆(Long Short-Term Memory，LSTM)网络的模型[22]，进行决策并输出变道行为(如向左变道、继续跟随、向右变道)的预测值，将预测值传送给变道信息处理模块进行处

理；然后，轨迹规划模块接收变道指令后规划出合适的路径、车速等信息；最后，底层控制模块接收到轨迹跟踪的详细信息后，对车辆的姿态进行控制，使其循迹行驶，保障无人车系统产生安全、合理的驾驶变道行为。

2.2 基于 MARTE 的智能无人车变道系统建模

MARTE 模型由 UML 和 MARTE 规范组成。其中，UML 是一种可以从多角度分析系统结构和行为特征的通用标准化语言；MARTE 规范是用于实时嵌入式系统开发建模的 UML 扩展。在系统非功能属性方面，使用 MARTE 模型能够更好地开展智能无人车系统在基于模型实时性能分析方面的工作。本文将 MARTE 规范以构造型的方式添加到 UML 元素上，用于描述智能无人车系统的实时性能参数。因此，本文采用的基于 MARTE 的智能无人车系统性能分析模型主要由部署图、活动图以及描述系统性能信息的 MARTE 构造型《PaStep》组成，分别从静态和动态的角度描述智能无人车系统，实现对智能无人车系统较为全面的建模。以图 1 所示的智能无人车变道系统为例，下面介绍本文所用 MARTE 模型的主要构成。

图 2(a)为系统的部署图，描述了智能无人车系统在运行时模块结构的构成，包括 Perception，Behavior Decision，Planning，Control 4 个模块节点，并通过 Data Transmission 节点进行数据传输。图 2(b)是包含智能无人车系统各种活动节点的活动图，其描述了智能无人车系统功能的动态行为，以及活动之间的执行逻辑顺序。本文中的活动图主要包括初始节点(Initial Node)、结束节点(Activity Final Node)、活动节点(Action Node)、分叉节点(Fork Node)和汇合节点(Join Node)、概率选择节点(Probabilistic Decision Node)。为了更好地分析系统实时性能，本文为每个活动节点添加了附属构造型《PaStep》：={*Host*，*execTime*}，其中属性 Host 表示活动节点由部署图中相应的模块来执行，属性 execTime 表示节点的执行时间。例如，节点 s_6 所对应构造型为：《PaStep》s_6 ={*Host*=Behavior Decision，*execTime*=1.44 s}，表示该节点的执行时间为 1.44s，并由部署图中的 Behavior Decision 模块执行。

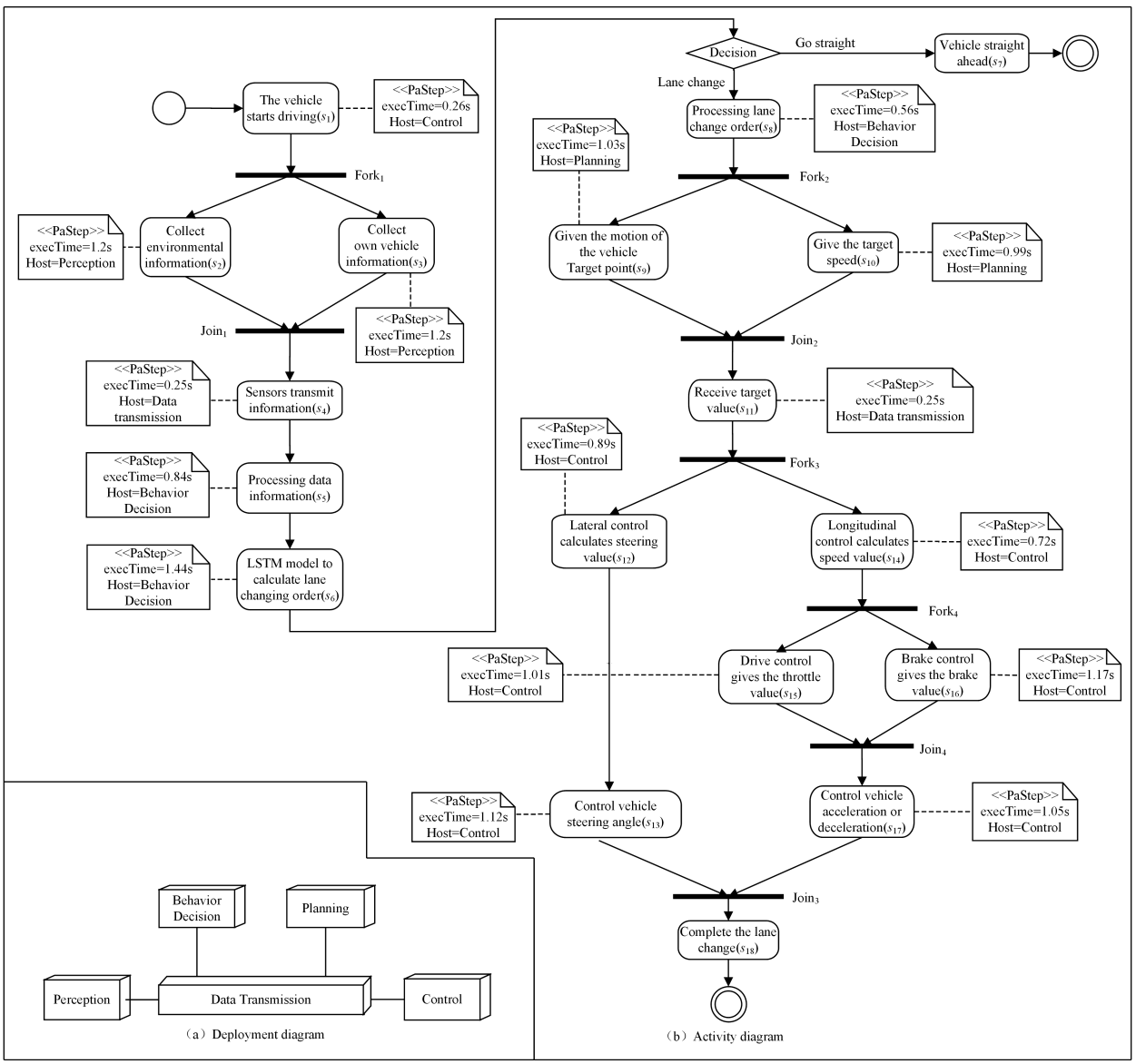


图 2 MARTE 模型

Fig. 2 MARTE model

3 模型转换

基于 MARTE 的智能无人车变道系统性能模型示例中，活动图描述了智能无人车系统完成变道的过程。一方面，马尔可夫链[23](Markov Chain，MC)的无后效性符合图中状态转移仅与相邻状态有关的特点；另一方面，图中存在至少一种终止节点状态，这样的特征与 AMC 的吸收态相符。因此，使用 AMC 可以更好地对智能无人车系统进行实时性能分析。基于以上分析，本节先介绍 AMC 的相关定义及公式，再将 MARTE 模型转换为 AMC。

3.1 AMC 基本性质及公式

智能无人车系统执行过程中状态节点访问次数的期望值表示了该运行场景下系统对该状态的使用程度，可以为系统进行实时性能分析提供良好的基础。为了得到状态节点访问次数的期望值，首先给出 AMC 的相关定义。

定义 1(马尔可夫链 MC) 对于一个离散的包含有限状态的随机序列集合 $X=\{x_1,x_2,\cdots,x_n\}$，如果每个状态值仅与前一个相邻的状态值有关，而与再之前的状态无关，则称为马尔可夫链。即对任意整数 i 有：

$$p(x_i|x_{i-1},x_{i-2},\cdots,x_1)=p(x_i|x_{i-1}) \tag{1}$$

定义 2(状态转移概率矩阵 $\boldsymbol{P}$) 将 MC 中的状态转移概率用矩阵 $\boldsymbol{P}$ 表示，其中 $p_{i,j}$ 表示状态 $x_i\rightarrow x_j$ 的概率，若状态 $x_i\rightarrow x_j$ 不可达，令 $p_{i,j}=0$。矩阵 $\boldsymbol{P}$ 满足：

$$\sum_{j=1}^{n}p_{i,j}=1,\forall\ 1\leqslant i\leqslant n \tag{2}$$

其中，$0\leqslant p_{i,j}\leqslant 1(1\leqslant i,j\leqslant n)$。

定义 3(初始状态) 智能无人车系统执行过程的出发状态，该状态节点只有流出边。

定义 4(吸收状态) 智能无人车系统执行过程的目标状态，该状态节点只有流入边。

定义 5(过渡状态) 状态序列中，除去吸收状态的其他状态称为过渡状态。

定义 6(吸收态马尔可夫链 AMC[23]) 含有吸收状态的马尔可夫链称为吸收态马尔可夫链，对应状态转移概率矩阵可表示为：

$$\boldsymbol{P}=\begin{pmatrix}\boldsymbol{Q} & \boldsymbol{C}\\ 0 & \mathbf{I}\end{pmatrix} \tag{3}$$

其中，$\boldsymbol{Q}$ 是$(n-m)\times(n-m)$的矩阵，表示过渡状态间的转移概率；$\mathbf{I}$ 是 $m\times m$ 的单位矩阵；0 是 $m\times(n-m)$的零矩阵；$\boldsymbol{C}$ 是$(n-m)\times m$ 的非零矩阵；n 表示该 AMC 的状态个数，m 表示 n 个状态中吸收状态的个数。

接下来给出节点状态访问次数期望值的计算流程。设 $p_{i,j}$ 表示智能无人车系统执行过程从状态节点 i 转移到 j 的概率，令 $p_{i,j}^{m}$ 表示该执行过程从状态 i 出发，经过 m 步转移后，到达状态 j 的概率，则：

$$\boldsymbol{P}^{m}=(p_{i,j}^{m})=\begin{pmatrix}\boldsymbol{Q}^{m} & \sum_{k=0}^{m-1}\boldsymbol{Q}^{k}\boldsymbol{C}\\ 0 & \mathbf{I}\end{pmatrix} \tag{4}$$

由于过渡状态最终都要进入吸收状态，即经过 $m(m\rightarrow\infty)$ 步后，过渡状态的期望转移概率为 0，满足：

$$\lim_{m\rightarrow\infty}\boldsymbol{P}^{m}=\begin{pmatrix}0 & \sum_{k=0}^{m-1}\boldsymbol{Q}^{k}\boldsymbol{C}\\ 0 & \mathbf{I}\end{pmatrix} \tag{5}$$

智能无人车系统执行过程从过渡状态 i 出发，在吸收前访问过渡状态 j 的平均转移次数用 $N_{i,j}$ 表示，其中 $1\leqslant i,j\leqslant n$，设 $\boldsymbol{N}$ 为 $n\times n$ 的矩阵，将 $N_{i,j}$ 赋值为矩阵 $\boldsymbol{N}$ 中位置(i,j)的元素，称 $\boldsymbol{N}$ 为平均转移次数矩阵，则：

$$\boldsymbol{N}=(I-\boldsymbol{Q})^{-1} \tag{6}$$

假设该执行过程从状态 1 开始，$K_{1,3}$ 表示从状态 1 开始到此过程进入吸收状态之前访问状态 3 的次数。而平均转移次数矩阵 $\boldsymbol{N}$ 的第(1,3)元素 $N_{1,3}$ 表示从状态 1 开始进入吸收状态之前访问状态 3 的次数的期望值，可以得到：

$$k_3=E[K_{1,3}]=N_{1,3} \tag{7}$$

其中，k_3 表示 $K_{1,3}$ 的期望值。因此，$k_j(j=1,2,3\cdots)$可表示为智能无人车系统执行过程中状态节点访问次数的期望值。

3.2 MARTE 模型到 AMC 的转换

由于 MARTE 模型中的活动节点分别被部署到部署图中的模块上，无需对部署图单独做转换，因此 MARTE 模型到 AMC 的转换就相当于活动图到 AMC 的转换。

将活动图中的初始节点(Init Node)、分叉节点(Fork Node)和汇合节点(Join Node)的后继工作流分别转换为模型 $Init\_AMC$，$Fork\_AMC$ 和 $Join\_AMC$，并且模型之间的通信通过全局变量 $f_{p,q}$ 和 $j_{p,q}(p,q=1,2,\cdots)$实现。其中，活动图中的各个节点都映射成为 AMC 中的相应状态。概率选择节点的后继工作流中的概率选择特征与 AMC 中动作的概率迁移特征相同，因此可将其转换为状态 $P\_decision$。活动结束节点表示活动图的执行结束，因此将活动结束节点转换为状态 End。

活动图到 AMC 的映射规则参考文献[24]，图 2(b)中的初始节点转换为图 3(a)的模型 $Init_$AMC，其初始状态为 $ADStart$；图 2(b)中的分叉节点 $Fork_1$ 转换为图 3(a)中的状态 $Fork_1$、图 3(b)的模型 $Fork_1\_AMC_1$ 和图 3(c)模型 $Fork_1\_AMC_2$。状态 $Fork_1$ 后继的迁移将全局变量 f_{11} 和 f_{12} 赋值为 true(T)之后，到达状态 $Over$。模型 $Fork_1\_AMC_1$ 和 $Fork_1\_AMC_2$ 的初始状态分别为 $FStart_1$ 和 $FStart_2$，并且其后继迁移当全局变量 f_{11} 和 f_{12} 全部为 true 时执行，同时将 f_{11} 和 f_{12} 全部赋值为 false(F)。同理可以得到，图 2(b)中的分叉节点 $Fork_2$，$Fork_3$，$Fork_4$ 分别转换为图 3 中(d)－(k)的模型 $Fork_2\_AMC_1$、模型 $Fork_2\_AMC_2$、模型 $Fork_3\_AMC_1$、模型 $Fork_3\_AMC_2$、模型 $Fork_4\_AMC_1$ 以及模型 $Fork_4\_AMC_2$；图 2(b)中的汇合节点 $Join_1$ 转换为图 3 中(d)的模型 $Join_1\_AMC$、图 3(b)中的状态 ToJ_{11} 和图 3(c)中的状态 ToJ_{12}，并且状态 ToJ_{11} 和 ToJ_{12} 的后继迁移将全局变量 j_{11} 和 j_{12} 赋值为 true 之后，到达状态 $Over$。模型 $Join_1\_AMC$ 的初始状态为 $JStart$，其中迁移当全局变量 j_{11} 和 j_{12} 为 true 时执行，同时将 j_{11} 和 j_{12} 赋值为 false。同理可得到图 2(b)中的汇合节点在图 3 中的转换关系。

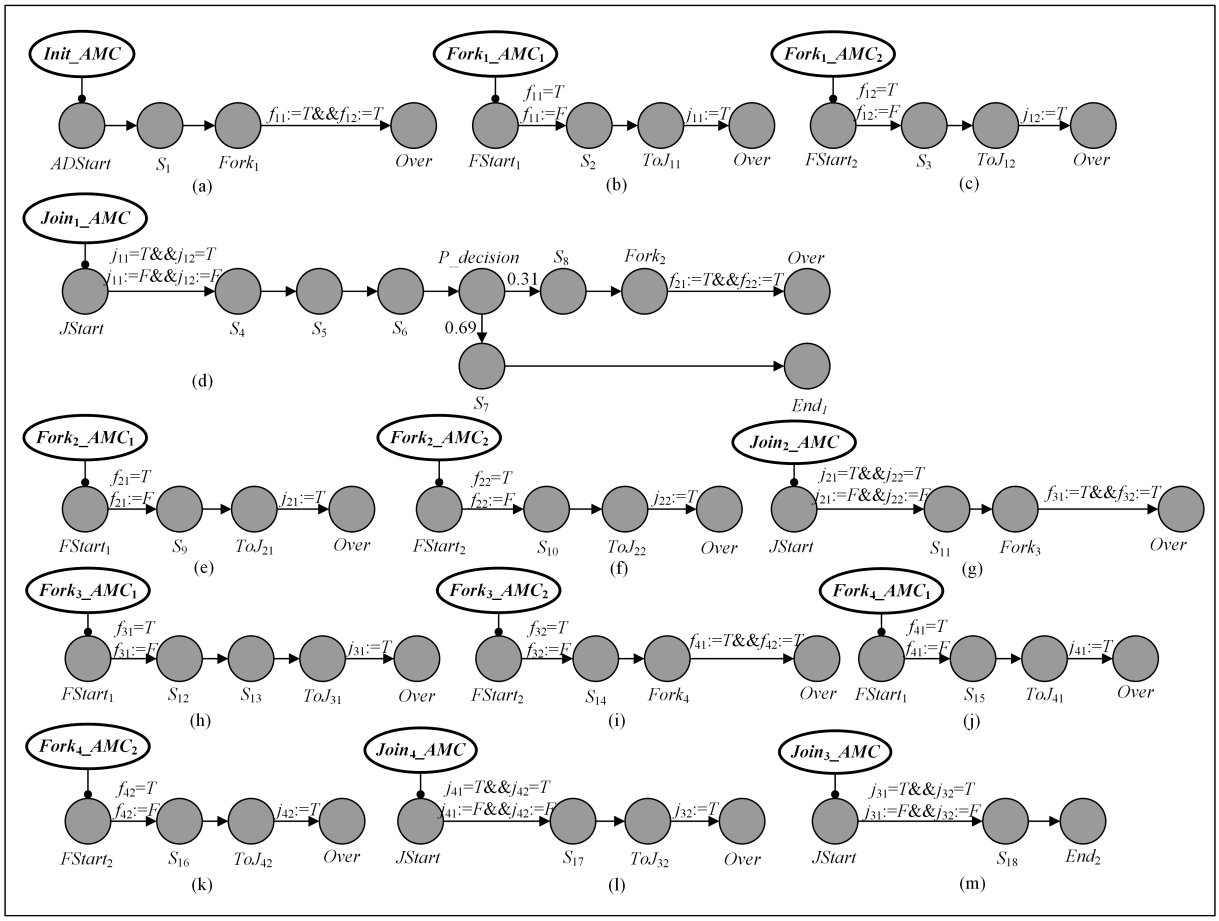


图 3　由图 2(b)转换的 AMC

Fig. 3　AMC transformed from Fig. 2(b)

4　实时性能分析

本文主要分析智能无人车系统的实时性能。实时是一种用物理时钟衡量的时间概念，其要求系统在所指定的时间段内完成计算、处理、执行等任务。在智能无人车系统中，其可以被定义为对环境中发生事件的响应时间，即事件发生、智能无人车系统检测、决策和做出响应之间所耗费的时间[25]，通常情况下耗费时间越少则表明该系统的实时性性能越优。

实时性能分析的目标是在运行场景下，根据节点状态的执行时间求得系统的响应时间。对于运行场景 s，假设 $t_{i,s}$ 表示节点状态在运行时的执行时间。节点状态 i 在运行场景 s 下的响应时间 $T_{i,s}$ 为：

$$T_{i,s}=N_{i,s}t_{i,s} \tag{8}$$

其中，$N_{i,s}$ 表示运行场景下，系统完成某项任务中节点状态 i 的执行次数。$N_{i,s}$ 和 $T_{i,s}$ 都是随机变量，而使用泰勒展式可以求解出由随机变量组成的函数的数学期望，因此执行一次过程下节点状态 i 的期望值 $E[T_{i,s}]$ 可以表示为：

$$E[T_{i,s}]=E[t_{i,s}^{N_{i,s}}]=k_{i,s}t_{i,s} \tag{9}$$

其中，$k_{i,s}$ 表示运行场景 s 执行一次过程中节点状态 i 执行次数的期望值，可由式(7)求出。

运行场景 s 下整个系统(包含 n 个节点状态)的响应时间 T_s 可以由上述推导的节点状态的响应时间得到：

$$T_s=\sum_{i=1}^{n}T_{i,s} \tag{10}$$

由式(9)可以得到运行场景 s 下整个系统的响应时间的期望值 $E[T_s]$：

$$E[T_s]=\sum_{i=1}^{n}k_{i,s}t_{i,s} \tag{11}$$

例如，系统执行过程中过渡状态为 a_1，a_2，a_3，吸收状态为 a_4。其中 a_1，a_2，a_3 的节点状态访问期望值为 0.35，0.58，0.46，节点状态的执行时间为 0.84 s，0.67 s，0.91 s，那么该运行场景下系统的响应时间期望值由式(11)计算得 1.1012 s。

5　实验

本节对上述的 AMC 图进行分析，在此，仅考虑智能无人车系统成功完成变道过程，即节点状态 $ADStart \rightarrow End_2$。

通过使用 Matlab 软件计算第 3.1 节所述公式可得平均转移次数矩阵 $\boldsymbol{N}$，从矩阵 $\boldsymbol{N}$ 中可得初始状态 $ADStart$ 到达吸收状态 End_2 前经过各状态节点访问次数的期望值 k_j，结果如表 1 所列，表中还包括图 2 中引入的各节点状态的执行时间。这里需要指出的是，本文所采用的数据只是一个举例来验证算法，该类数据可通过仿真或者专家意见等方式来收集。

另外，依据第 4 节所述的分析方法，可以得到表 2 中系统各模块响应时间的相关指标。表 1 和表 2 数据表明，在智能无人车系统完成变道的过程中，环境感知模块中的收集周围车辆信息状态 s_2、收集自车信息状态 s_3 以及行为决策模块中计算变道信息的 LSTM 神经网络模型[22]，对系统变道执行过程中响应时间的影响最大，经对比可得整个系统中智能模块

(LSTM Model)所耗费的时间多于系统中其他传统模块的耗时。文中所提模型可对智能无人车系统进行实时性能分析，并根据此数据可以针对性地采取相关措施，以改善系统的实时性能。

表 1 状态数据

Table 1 Statedata

System Module		State	k_j	T_i/s
Perception		s_2	1.00	1.20
		s_3	1.00	1.20
Behavior Decision (LSTM Model)		s_5	1.00	0.84
		s_6	1.00	1.44
		s_8	0.31	0.56
Planning		s_9	0.31	1.03
		s_{10}	0.31	0.99
Control	Trajectory Tracking	s_{12}	0.31	0.89
		s_{14}	0.31	0.72
		s_{15}	0.31	1.01
		s_{16}	0.31	1.17
	Motion Control	s_1	1.00	0.26
		s_{13}	0.31	1.12
		s_{17}	0.31	1.05
DataTransmission		s_4	1.00	0.25
		s_{11}	0.31	0.25

表 2 响应时间

Table 2 Response time

System Module	Response Time/s
Perception	2.4000
Behavior Decision	2.4536
Planning	0.6262
TrajectoryTracking	1.1749
Motion Control	0.9327
DataTransmission	0.3275
System	7.9149

从上述分析结果来看，若想提升智能无人车系统的实时性性能，需要减少系统中智能模块的耗时。为此，本文还分析了智能模块的训练程度对其准确率、响应时间的影响。实验结果如表 3 所列。

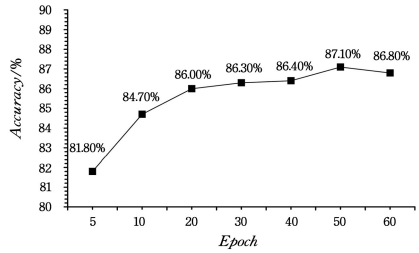
表 3 训练程度对准确率与响应时间的影响

Table 3 Influence ofepoch on accuracy and response Time

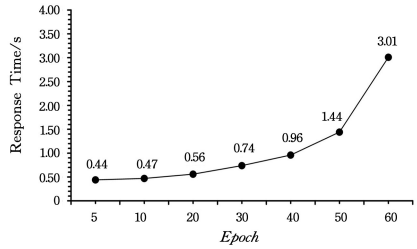
Epoch	*Accuracy*/%	Response Time/s
5	81.8	0.44
10	84.7	0.47
20	86.0	0.56
30	86.3	0.74
40	86.4	0.96
50	87.1	1.44
60	86.8	3.01

图 4 更直观地显示出了智能模块的训练程度与准确率的变化关系，以及训练程度与响应时间的变化关系。

由表 3 和图 4 分析可得，智能模块的训练程度对系统的准确率和响应时间都有影响，随着训练程度增大，智能模块的准确率达到一定程度后便不会再有太大改善，而响应时间是一直在增大的，并且增加幅度越来越大。因此，可以根据具体运行场景有针对性地应用训练程度不同的智能模块，从而改善系统的实时性能。



(a) Epoch and Accuracy



(b) Epoch and Response Time

图 4 训练程度、准确率与响应时间之间的关系

Fig. 4 Relationship between epoch, accuracy and response time

结束语 为了对智能无人车系统的实时性能问题进行分析，本文采用了基于 MARTE 的智能无人车系统的实时性能分析模型，在系统设计早期就引入性能参数，并提出了 AMC 的分析方法求得系统的实时性能指标，从而对该系统进行实时性能分析，最后针对影响系统实时性的关键模块进行分析，提出了改进措施。以基于 LSTM 的智能无人车变道系统为例，验证了该方法的可行性，并且实验表明，此方法能够指出系统的实时性能瓶颈，分析出智能模块是影响系统实时性能的关键模块，在实际应用场景中可根据不同的需求选择训练程度适中的智能模块，进而改善整体系统的实时性能，具有一定的指导意义。

由于智能无人车系统实时性能问题在真实场景中要比本文讨论的情况复杂很多，系统往往受到系统资源和应用场景等多种相关因素制约，在运行时可能会存在故障问题，因此接下来的研究工作主要包括两个方面：考虑更多的性能指标，如吞吐量、利用率等，从而可以根据这些性能指标进一步分析系统的性能；通过仿真实验与本文方法对比，验证了所提模型的有效性，并在后续工作中进行理论分析。

参 考 文 献

[1] ZHANG T, LI Q, ZHANG C S, et al. Current trends in the development of intelligent unmanned autonomous systems[J]. Frontiers of Information Technology&Electronic Engineering, 2017, 18(1): 68-85.

[2] BALSAMO S, MARCO A D, INVERARDI P, et al. Model-based performance prediction in software development: a survey [J]. IEEE Transactions on Software Engineering, 2004, 30(5): 295-310.

[3] DENARO G, POLINI A, EMMERICH W. Early performance testing of distributed software applications[C]// Proceedings of the 4th International Workshop on Software and Performance. 2004: 94-103.

[4] WOODSIDE M,PETRIU D C,MERSEGUER J,et al. Transformation challenges:from software models to performance models [J]. Software & Systems Modeling,2014,13(4):1529-1552.

[5] QUADRI I,BAGNATO A,BROSSE E,et al. Modeling methodologies for Cyber-Physical Systems:Research field study on inherent and future challenges [J]. Ada User Journal, 2015, 36(4):246-253.

[6] FAUGERE M,BOURBEAU T,DE SIMONE R,et al. Marte: Also an uml profile for modeling aadl applications[C]// 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007). IEEE,2007:359-364.

[7] BRUGALI D. Modeling and Analysis of safety requirements in robot navigation with an extension of UML MARTE[C]// 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR). IEEE,2018:439-444.

[8] XIA H,JIAO J,DONG J. Extend UML based timeliness modeling approach for complex system[C]// 2018 International Conference on Mathematics,Modeling,Simulation and Statistics Application (MMSSA 2018). Atlantis Press,2019:1-6.

[9] BECKER S,KOZIOLEK H,REUSSNER R. Model-Based performance prediction with the palladio component model[C]// Proceedings of the 6th International Workshop on Software and Performance. ACM,2007:54-65.

[10] PETRIU D C,ALHAJ M,TAWHID R. Software performance modeling[C]// International School on Formal Methods for the Design of Computer, Communication and Software Systems. Berlin:Springer,2012:219-262.

[11] BALSAMO S,PERSONE VDN,INVERARDI P. A review on queueing network models with finite capacity queues for software architectures performance prediction[J]. Performance Evaluation,2003,51(2/3/4):269-288.

[12] BALSAMO S,MARZOLLA M. Performance evaluation of UML software architectures with multiclass queueing network models[C]// Proceedings of the 5th international workshop on Software and performance. 2005:37-42.

[13] ARCAINI P,INVERSO O,TRUBIANIC. Automated model-based performance analysis of software product lines under uncertainty[J]. Information and Software Technology,2020,127:106371.

[14] HU X,JIAO L,CHAI Y S. Transforming UML to GSPN for Performance Analysis[J]. Computer Science,2016,43(11):49-54.

[15] SHAILESH T,NAYAK A,PRASAD D. An UML Based Performance Evaluation of Real-Time Systems Using Timed Petri Net[J]. Computers,2020,9(4):94.

[16] GILMORE S,HILLSTON J,KLOUL L,et al. Software performance modelling using PEPA nets[J]. ACM SIGSOFT Software Engineering Notes,2004,29(1):13-23.

[17] LV H,WANG H,ZHAO Q,et al. Modeling and Analysis of Self-Reflection Based on Continuous State-Space Approximation of PEPA[C]// 2009 Eighth IEEE International Conference on Dependable,Autonomic and Secure Computing. IEEE,2009:84-89.

[18] SHIN J,SUNWOO M. Vehicle speed prediction using a Markov chain with speed constraints[J]. IEEE Transactions on Intelligent Transportation Systems,2018,20(9):3201-3211.

[19] PEREZ J F,SILVA D F,GOEZ J C,et al. Algorithm 972: jMarkov:An integrated framework for Markov chain modeling [J]. ACM Transactions on Mathematical Software (TOMS), 2017,43(3):1-22.

[20] SHEPERO M,MUNKHAMMAR J. Spatial Markov chain model for electric vehicle charging in cities using geographical information system (GIS) data[J]. Applied energy,2018,231: 1089-1099.

[21] CHEN M,WANG T,OTA K,et al. Intelligent resource allocation management for vehicles network: An A3C learning approach[J]. Computer Communications,2020,151:485-494.

[22] SU S,MUELLING K,DOLAN J,et al. Learning vehicle surrounding-aware lane-changing behavior from observed trajectories[C]// 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE,2018:1412-1417.

[23] BERTSEKAS D P,TSITSIKLIS J N. Introduction to probability [M]. Belmont,MA:Athena Scientific,2002:339-380.

[24] CHAI Y S,ZHU X Y,YAN R J,et al. MARTE Models Based System Reliability Prediction [J]. Computer Science, 2015, 42(12):82-86.

[25] BAUMS A,GORDYUSHIN A. Response time of a mobile robot [J]. Automatic Control and Computer Sciences, 2013, 47(6): 352-358.

WU Pei-pei, born in 1994, postgraduate. Her main research interests include intelligent information processing and intelligent software performance evaluation.