Models, Algorithms, and Evaluation for Autonomous Mobility-On-Demand Systems

Rick Zhang, Kevin Spieser, Emilio Frazzoli, and Marco Pavone

***Abstract*— This tutorial paper examines the operational and economic aspects of autonomous mobility-on-demand (AMoD) systems, a rapidly emerging mode of personal transportation wherein robotic, self-driving vehicles transport customers in a given environment. We address AMoD systems along three dimensions: (1) modeling – analytical models capable of capturing the salient dynamic and stochastic features of customer demand, (2) control – coordination algorithms for the vehicles aimed at stability and subsequently throughput maximization, and (3) economic – fleet sizing and financial analyses for case studies of New York City and Singapore. Collectively, the models and algorithms presented in this paper enable a rigorous assessment of the *value* of AMoD systems. In particular, the case study of New York City shows that the current taxi demand in Manhattan can be met with about 8,000 robotic vehicles (roughly 70% of the size of the current taxi fleet), while the case study of Singapore suggests that an AMoD system can meet the personal mobility need of the entire population of Singapore with a number of robotic vehicles that is less than 40% of the current number of passenger vehicles. Directions for future research on AMoD systems are presented and discussed.**

I. Introduction

A. Personal urban mobility in the 21st century

In the past century, *private* automobiles have dramatically changed the paradigm of *personal urban mobility* by enabling *fast* and *anytime* point-to-point travel within cities. However, concerns about the sustainability of this transport model are growing. Arguments against the private car include its high dependence on oil, the production of harmful greenhouse gases, escalating levels of traffic congestion, and an ever-increasing demand for land to build more roads and parking lots [1]. In the US, urban vehicles consume more than half of the oil used by *all* sectors [2], and produce 20% of the total carbon dioxide emissions [3], [4]. Recently, congestion has soared dramatically, a reflection of aging and inadequate infrastructure that cannot keep pace with increasing transportation demand [5]. In 2011, congestion in metropolitan areas increased the collective travel time of Americans living in or commuting through urban areas by 5.5 billion hours (causing a 1% loss of US GDP [6]). This figure is projected to increase by 50% by 2020 [6]. Concurrently, valuable and increasingly rare land is being paved for new roads and parking lots. The problem is even worse at a global scale, due the combined impact of rapid increases in the urban population (projected to reach 5 billion, more than 60% of the world population, by 2030 [7]) and increasing car ownership in developing countries [1]. Owing to these factors, the private automobile is widely recognized as an *unsustainable option for personal mobility in urban areas* [1].

Rick Zhang and Marco Pavone are with the Department of Aeronautics & Astronautics, Stanford University, Stanford, CA 94305 {rickz, pavone}@stanford.edu

Kevin Spieser and Emilio Frazzoli are with the Laboratory for Information and Decision Systems, Department of Aeronautics & Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 {kspieser, frazzoli}@mit.edu

B. The rise of mobility-on-demand

The challenge in finding a sustainable solution to satisfy the demand for urban transportation lies in preserving the benefits of the private automobile, while reducing the dependency on non-renewable resources, minimizing pollution, and avoiding the need drastically expand existing infrastructure, i.e., roads and parking lots. A hint at a solution comes from realizing that most urban vehicles are *over-engineered* and *underutilized*. For example, a typical automobile can attain speeds well over 100 miles per hour, but urban driving is typically confined to much slower speeds (in the 15- to 25-miles per hour range [5], [8]). Furthermore, private automobiles are parked more than 90% of the time [5]. Taking direct aim at these inefficiencies, one of the most promising strategies for personal mobility in urban areas is the concept of *one-way vehicle sharing*. In this system (referred to as Mobility-on-Demand or MoD), small, electric cars are provided at stacks and racks at closely spaced intervals throughout a city [1]. When a person wants to travel, she/he simply walks to the nearest rack, swipes a card to pick up a vehicle, drives it to the rack nearest her/his destination, and drops it off.

In many ways, MoD systems that use electric vehicles directly address the problems of oil dependency (assuming electricity is produced cleanly), pollution, and, via higher utilization rates, parking lot sprawl. Furthermore, they offer greater flexibility as compared to two-way rental systems, in which cars must be returned to the station they were rented from. Finally, they provide *personal*, *anytime* mobility beyond what is offered by traditional taxi systems or alternative one-way ridesharing concepts such as carpooling, vanpooling, and buses. As such, MoD systems have been advocated as a key step toward sustainable *personal* urban mobility in the 21st century [1]. The recent success of Car2Go (a one-way rental company operating over 10,000 two-passenger vehicles in 26 cities worldwide [9]) corroborates this statement (see Figure 1, left).

However, MoD systems have their own limitations and challenges. For example, due to the spatio-temporal nature of urban mobility, certain locations tend to be more popular

Fig. 1. Left figure: A Car2Go vehicle used in a traditional (i.e., non-robotic) MoD system. Right figure: Self-driving vehicle that Google will use in a 100-vehicle AMoD pilot project within the next two years. Image credit: Car2Go and Google.

destinations than others. Consequently, the vehicles in MoD systems inevitably become *unbalanced*: vehicles build up in some parts of a city and become depleted in others. Moreover, MoD systems do *not* directly address the need to reduce congestion. In fact, the need to rebalance vehicles creates additional trip that actually increase the overall mileage driven.

C. Beyond MoD: autonomous mobility-on-demand (AMoD)

Advancement in autonomous driving technology, largely over the last decade, have the potential to address a number of the inherent challenges MoD systems present. For example, robotic vehicles can rebalance themselves (eliminating the rebalancing problem at its core), autonomously monitor and recharge their batteries (at dedicated charging stations), and coordinate their actions at a system-wide level to optimize throughput. Furthermore, robotic vehicles would free passengers from the task of driving, provide a personal mobility option to people unable or unwilling to drive, and offer a number of safety improvements. Recognizing these benefits, a number of companies and car manufacturers are aggressively pursuing "AMoD technology," with initiatives ranging from the design of vehicles specifically tailored to AMoD operations [10], [11], to Google's expected launch of a 100-vehicle AMoD pilot project within the next two years [12] (see Figure 1, right).

Rapid advances in vehicle automation technologies coupled with the increased economic and social interest in MoD systems have fueled heated debates regarding the viability of AMoD systems on these fronts. Assessing the merits of these claims raises a number of important questions. How many robotic vehicles would be needed to achieve a certain quality of service? What would be the cost for their operation? Would AMoD systems decrease congestion? In general, do AMoD systems represent an economically viable, sustainable, and societally-acceptable solution to the future of personal urban mobility?

D. Paper contributions

To answer the above questions, we need to first understand how to *control* AMoD systems, which entails optimally routing, in real time, potentially hundreds of thousands of robotic vehicles. Such routing process must take into account the spatio-temporal variability of mobility demand, together with a number of constraints such as congestion and battery recharging. This represents a networked, heterogeneous, stochastic decision problem with uncertain information, hence complexity is at its heart. Within this context, the contribution of this paper is threefold:

1) We present two spatial queueing-theoretical models for AMoD systems that capture salient dynamic and stochastic features of customer demand. A spatial queueing model entails an exogenous dynamical process that generates "transportation requests" at *spatially localized* queues. Specifically, the first model, referred to as the "distributed" model, transforms the problem of controlling a set of spatially localized queues into one of controlling a single "spatially-averaged" queue and allows the determination of *analytic* scaling laws that can be used to select important system parameters (e.g., fleet size). The second model, referred to as the "lumped" model, exploits the theory of Jackson networks and allows the computation of key performance metrics and the design of system-wide coordination algorithms. For the distributed and lumped models, we provide an inline description of the background necessary to understand the model and how our models can be calibrated using available datasets.
2) We discuss the synthesis of closed-loop control policies within, respectively, the distributed and the lumped models. The policies combine techniques from receding horizon control, combinatorial optimization, and integer programming.
3) Using the control strategies outlined above, we discuss two case studies for the deployment of AMoD systems: one based on taxis in New York City and another targeting all modes of land based transport in Singapore. These case studies suggest that it is much more affordable (and convenient) to access mobility in an AMoD system compared to traditional mobility models based on private vehicle ownership.

The paper concludes with a discussion about future directions for research, with a preliminary discussion about the potential of AMoD systems to *decrease* congestion. The results presented in this paper are primarily based on [13] as well as a number of previous works by the authors and their collaborators, namely [14] for the lumped approach, [15]–[18] for the spatial queueing-theoretical framework and the distributed approach, and [14], [19] for the case studies.

The rest of this paper is structured as follows. Section II presents a general introduction to spatial queueing models for AMoD systems. Sections III and IV present, respectively, the distributed and lumped models of AMoD systems. In each case, the discussion highlights the performance metrics that can be ascertained from relevant model parameters and describes how these parameters may be calibrated from available data. Section V introduces control algorithms for AMoD fleet operations, based on the previously defined models. Section VI leverages analysis and control synthesis tools from Sections II to V to provide an initial *evaluation*

of AMoD systems for two case studies focusing on New York City and Singapore. A financial analysis that reveals the benefits of AMoD systems as well as a human-based approach to rebalancing MoD systems are also provided. Section VII outlines directions for future research, with a particular emphasis on (and some preliminary results for) congestion effects. Finally, Section VIII concludes the paper.

II. Spatial queueing-theoretical models of AMoD systems: an overview

At a high level, an AMoD system can be mathematically modeled as follows. Consider a given environment, where a fleet of self-driving vehicles fulfills transportation requests. Transportation requests arrive according to an *exogenous dynamic process* with associated origin and destination locations within the environment. The transportation request arrival process and the spatial distribution of the origin-destination pairs are modeled as stochastic processes, leading to a probabilistic analysis. Transportation requests queue up within the environment, which gives rise to a network of *spatially localized* queues dynamically served by the self-driving vehicles. Such a network is referred to as a "spatial queueing system." Performance criteria include the availability of vehicles upon the request's arrival and average wait time to receive service. The model is portrayed in Figure 2.

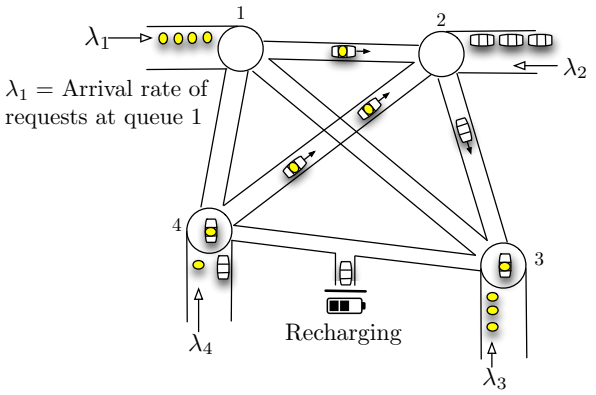


Fig. 2. A spatial queueing model of an AMoD system entails an exogenous dynamical process that generates "transportation requests" (yellow dots) at *spatially localized* queues. Self-driving vehicles (represented by small car icons) travel among such locations according to a given network topology to transport customers.

Controlling a spatial queueing system involves a *joint task allocation and scheduling problem*, whereby vehicle routes should be dynamically designed to allocate vehicles to transportation requests so as to minimize, for example, wait times. In such a dynamic and stochastic setup, we need to design a *closed-loop control policy*, as opposed to open-loop preplanned routes. The problem combines aspects of networked control, queueing theory, combinatorial optimization, and geometric probability (i.e., probabilistic analysis in a geometrical setting). This precludes the direct application of "traditional" queueing theory due to the complexity added by the spatial component (these complexities include, for example, congestion effects on network edges, energy constraints, and statistical couplings induced by the vehicles' motion [18], [20], [21]). It also precludes the direct application of combinatorial static optimization, as the dynamic aspect of the problem implies that the problem instance is *incrementally revealed over time* and static methods can no longer be applied.

Next two sections present two recent, yet promising approaches for modeling AMoD systems within the framework of spatial queueing systems, namely the *distributed approach* (Section III) and the *lumped approach* (Section IV) . Both approaches employ a number of relaxations and approximations to overcome the difficulties in directly applying results from queueing (network) theory to spatial queueing models. A remarkable feature of these approaches is that they generally yield *formal performance bounds* for the control policies and scaling laws for the quality of service in terms of model data, which can provide useful guidelines for selecting system parameters (e.g., number of vehicles). These approaches take their origin from the seminal works on hypercube models for spatial queues [20], on the Dynamic Traveling Repairman problem [21]–[24], and on the Dynamic Traffic Assignment problem [25], [26].

Alternative approaches could be developed by leveraging *worst-case* (as opposed to stochastic) techniques for dynamic vehicle routing, e.g., competitive (online) analysis [27]–[29]. This is an interesting direction for future research.

III. Distributed spatial-queueing model of an AMoD system

A. The model

The key idea behind the distributed approach, discussed in [15]–[18], is that a collection of N stations represents a continuum as $N \to \infty$. In this way, the setup in which demands enter the workspace is similar to the Dynamic Traveling Repairman problem [21]–[24]. In other words, customers can arrive at any point in a given bounded planar environment [16], [17]. Similarly, in a road map, customers can arrive at any point along an edge of the network [16]. In the simplest scenario, a dynamic process generates spatially localized origin-destination requests in a geographical region $Q \subset \mathbb{R}^2$. The process that generates origin-destination requests is modeled as a spatio-temporal Poisson process, namely, (i) the time between consecutive generation instants has an exponential distribution with intensity λ, and (ii) origins and destination are random variables with probability density functions, respectively, φ_{O} and φ_{D}, supported over Q, see Figure 3. Among trips, the origin points are independent and identically distributed (i.i.d.), as are the destination points. Furthermore, the origin and destination points associated with the same trip are independent. Trip requests are serviced by vehicles that travel at a constant speed v and may transport at most one trip demand at a time.

Within the distributed setup, the ultimate objective is to design a routing policy that minimizes the average steady-state time delay between the generation of an origin-destination pair and the time the trip is completed. By removing the

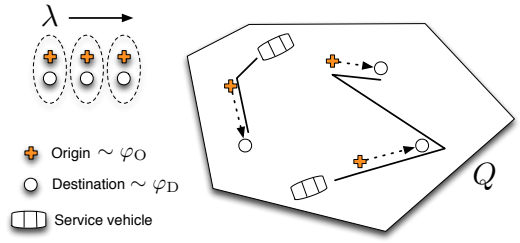


Fig. 3. In a distributed model of AMoD systems, a stochastic process with rate λ generates origin-destination pairs, distributed over a *continuous* domain Q.

constraint that the customers' origin-destination requests are localized at a finite set of points in an environment, one transforms the problem of controlling N different queues into one of controlling a single "spatially-averaged" queue. This considerably simplifies analysis and control, and allows us to derive *analytical* expressions for important design parameters. For example, it is possible to derive stability conditions, as briefly detailed below.

Given a trip arrival process, a fleet of m vehicles is said to *stabilize* the system if there exists a service policy $\pi(m)$ that ensures the expected number of outstanding demands is bounded at all times. Stability implies the fleet, as a whole, must be able to travel at least as fast, on average, as the rate at which trip distance accumulates. Given two points $p_1, p_2 \in Q$, let $d(p_1, p_2)$ denote the shortest distance between p_1 and p_2. Let $d_{\text{av}} = \mathbb{E}\left[d(O_i, O_{i+1}; \pi(m))\right]$ denote the average distance a vehicle must travel between two successive trips under $\pi(m)$. The rate at which work enters the system is then λd_{av}. A fleet of m vehicles, each capable of traveling at speed v, is able to, collectively, cover distance at a rate of mv. Therefore, a necessary condition for system stability is

$$m \geq \frac{\lambda d_{\text{av}}}{v}. \tag{1}$$

A self-driving vehicle successively alternates between two states when completing trips: (i) driving a passenger from an origin O_i to a destination point D_i and (ii) driving (empty) from D_i to the origin point of the next trip O_{i+1}. For $\pi(m)$, let $d_{\text{OD}} = \mathbb{E}\left[d(O_i, D_i)\right]$ denote the average trip distance and $d_e = \mathbb{E}\left[(D_i, O_{i+1})\right]$ the average inter-trip distance. For any reasonable $\pi(m)$, d_{OD} is simply the shortest distance between O and D. Quantifying d_e, however, closely depends on how $\pi(m)$ stipulates the vehicles should behave when transitioning from one job to the next. In [17], rigorous arguments are used to prove $d_e \geq \text{EMD}(\varphi_{\text{O}}, \varphi_{\text{D}})$ for any $\pi(m)$, where $\text{EMD}(\varphi_{\text{O}}, \varphi_{\text{D}})$ is the Earth Mover's Distance (or Wasserstein distance) between φ_{O} and φ_{D}. Intuitively, if φ_{O} and φ_{D} are imagined to be piles of dirt (i.e., earth), each having unit volume, then $\text{EMD}(\varphi_{\text{O}}, \varphi_{\text{D}})$ is the minimum amount of work (dirt $\times$ distance) required to reshape φ_{O} into φ_{D}. See [30] for more on the formal definition.

Noting that $d_{\text{av}} = d_{\text{OD}} + d_e$, the necessary condition on fleet size becomes

$$m \geq \frac{\lambda(d_{\text{OD}} + \text{EMD})}{v}. \tag{2}$$

Rearranging the above equation, we obtain that a necessary (indeed also sufficient [17]) condition for stability is that the load factor

$$\varrho := \lambda \left(\mathbb{E}_{\varphi_{\text{O}}\,\varphi_{\text{D}}}[Y - X] + \text{EMD}(\varphi_{\text{O}}, \varphi_{\text{D}})\right)/(v\,m) \tag{3}$$

is *strictly* less than one.

Equation (3) is useful because it can be used to estimate the minimum fleet size necessary to ensure stability. Although many existing works emphasize the relationship between d_{OD} and minimum fleet size, they often fail to recognize the contribution of $d_e \geq \text{EMD}(\varphi_{\text{O}}, \varphi_{\text{D}})$. This omission is unfortunate, as $\text{EMD}(\varphi_{\text{O}}, \varphi_{\text{D}})$ represents the minimum amount of distance, on average, a vehicle must travel to realign itself with the travel demand, and is a fundamental contributor to system workload. Ignoring $\text{EMD}(\varphi_{\text{O}}, \varphi_{\text{D}})$ is justifiable only in the rare instances where $\varphi_{\text{O}} = \varphi_{\text{D}}$.

By leveraging the distributed model, it is also possible to obtain formal performance bounds (i.e., factors of suboptimality) for receding-horizon control policies, in the *asymptotic* regimes $\varrho \to 1^-$ (heavy-load, system saturated) and $\varrho \to 0^+$ (light-load, system empty of customers) [18], [31].

B. Calibration of the model

1) Data Sources: In order to apply the distributed model to real-world cases, we need to "calibrate" all problem data (e.g., φ_{O}, φ_{D}, v, etc.). In this section, we show how this can be accomplished for the city of Singapore by considering three complementary data sources. We then proceed to describe the methodology for extracting the requisite quantities from the data source. These values will then be used in Section VI to compute the minimum fleet size for a case study of Singapore.

The Household Interview Travel Survey: The Household Interview Travel Survey, or simply HITS, is a comprehensive survey conducted periodically by the Land Transport Authority (LTA) for the purpose of gathering an overview of high-level transportation patterns within Singapore. This work employed the 2008 HITS survey in which 10,840 of the then 1,144,400 households in Singapore were selected to participate in the survey.

The HITS database, which summarizes the survey, is structured as follows. For each household surveyed, h_i, each resident, r_j, reported specific details of each trip, τ_k, taken on a recent weekday of interest. In general, τ_k is comprised of stages $s_1(\tau_k), s_2(\tau_k), \ldots$, with a new stage introduced each time the participant switched their mode of transport during τ_k, e.g., transferred from the subway to bus as part of the same trip. For each τ_k, the resident reported the trip's origin point O_k, destination point D_k, start time t_k^{st}, end time t_k^{end}, and the mode of transport, e.g., car, bus, subway, etc., used

in each substage. For the purposes of our study, each entry in the HITS database is a tuple of the form

$$(h_i, r_j, \tau_k, s_{k,\ell}, O_k, D_k, t_k^{st}, t_k^{end}, mode(s_{k,\ell})). \quad (4)$$

O_k and D_k are affixed postal code values in the HITS database. It should be noted that postal codes are liberally assigned in Singapore and, in many cases, each building is assigned its own postal code. Therefore, O_k and D_k provide sufficient geographic resolution regarding where τ_k begins and ends. To facilitate the analysis, postal codes were converted to their associated longitude and latitude coordinates by consulting an external database. Henceforth, we shall assume O_k and D_k are of the form $O_k = (lon_k^O, lat_k^O)$ and $D_k = (lon_k^D, lat_k^D)$, respectively. A detailed description of the HITS survey may be found in [32].

Taxi Data: Ground truth traffic characteristics were estimated by consulting taxi records collected over the course of a week, in 2012, in Singapore. The data chronicled the movement and activities of approximately sixty percent of all active taxis by listing each vehicle's GPS coordinates, speed, and passenger status, e.g., "passenger-on-board", "vacant", "responding to call", etc. Owing to the high rate at which recordings were taken (approximately every thirty seconds to one minute per vehicle) and the large number of taxis contributing to the database (in excess of 10,000), the fleet, collectively, serves as a distributed, mobile, embedded traffic sensor which was queried to provide an estimate of traffic conditions throughout the city.

Road Network: A graph-based representation of Singapore's road network was used to determine routes robotic vehicles should take when transporting passengers. Edges in this network are categorized based on the type of road they represent, e.g., a major highway or a residential street. This feature enables us view the road network at varying levels of granularity.

2) Methodology for computing λ, EMD, $\varphi_{\rm O}$, $\varphi_{\rm D}$, and v: This section describes the methodology used in [19] to estimate the quantities appearing in (2), i.e., the average demand arrival rate λ, the average trip length $d_{\rm OD}$, the average vehicle speed v, and finally the Earth Mover's Distance EMD measuring the difference between the distributions of origin and destination points.

Computation of λ: Let $\lambda_{\rm HITS}$ represent the average rate at which trips arrive based solely on the HITS survey. The overall arrival rate is then $\lambda = \alpha\lambda_{\rm HITS}$, where $\alpha = 1,144,400/10,840 = 105.57$ is the scaling factor that, inversely, reflects the fraction of the population that took part in the HITS survey. From the HITS data, 56,839 trips were extracted. After eliminating trips for which the GPS coordinates of O_k, D_k, or both were unavailable, 56,673 trips remained. Arrival rates for hour k, $0 \le k \le 23$, were then calculated as $\lambda_k = \alpha\lambda_{{\rm HITS},k}$, where $\lambda_{{\rm HITS},k}$ is the number trips in the HITS database that started in hour k.

Computation of $\varphi_{\rm O}$, $\varphi_{\rm D}$, and EMD*:* As mentioned, EMD is a measure of the driving distance required to reshape one distribution into another. In this regard, it is important to recognize that the HITS data is organized on a trip-by-trip basis and, as such, care must be taken in selecting the time window over which to compute EMD. Namely, for an individual that reports successive trips $(O_1, D_1), (O_2, D_2), \ldots, (O_k, D_k)$ throughout the day, it follows that $D_k = O_1$ and $O_{i+1} = D_i$, $i = 1, \ldots, k-1$. Consequently, when all trips are aggregated over the course of a day, $\varphi_{\rm O} = \varphi_{\rm D}$, which would imply, erroneously, that, having just dropped off a passenger, vehicles are immediately aligned with the transportation demand.

To avoid this problem, EMD was computed on a smaller time scale on par with how quickly trips were completed, namely, every hour, with EMD^k the Earth Mover's Distance associated with pickups and drop-offs in hour $k = 0, \ldots, 23$. Let $T^k := \{\tau : \tau \text{ starts in interval } k+1\}$, $k = 0, 1, \ldots, K$ denote the set of all trips τ that started in hour k. Similarly, let $\varphi_{\rm O}^k$, $\varphi_{\rm D}^k$, and EMD^k denote the origin distribution, destination distribution, and EMD associated with trips in T^k. Further details on the EMD and its calculation can be found in [33]. Here, EMD^k is computed by discretizing Singapore into N regions, $R_1, R_2, \ldots, R_N$. (In this work, for simplicity, we considered a 10-by-10 grid, hence $N = 100$.) Let c_i be the centroid of R_i. The distance between regions R_i and R_j is defined as $d(R_i, R_j) := ||c_i - c_j||$. For trip τ, let $O(\tau)$ and $D(\tau)$ denote τ's origin and destination points, respectively. Finally, let $\mathcal{O}(R_i^k) = \{\tau \in T^k : O(\tau) \in R_i\}$ and, likewise, $\mathcal{D}(R_i^k) = \{\tau \in T^k : D(\tau) \in R_i\}$. Note that $\{|\mathcal{O}(R_i^k)|/|T^k|\}_{i=1}^N$ defines an empirical probability distribution over the discrete regions $R_1, \ldots, R_N$.

Based on the aforementioned discretization of the workspace, EMD^k is given by the solution to the following flow-based optimization problem with decision variables f_{ij}:

$$\begin{aligned} \text{EMD}^k = \min_{f_{ij}} & \sum_{i=1}^N \sum_{j=1}^N f_{ij} d(R_i, R_j) \quad (5) \\ s.t. & \sum_{j=1}^N f_{ij} = \mathcal{O}(R_i^k)/|T^k|, \ i = 1, \ldots, N \\ & \sum_{i=1}^N f_{ji} = \mathcal{D}(R_i^k)/|T^k|, \ i = 1, \ldots, N \\ & f_{ij} \ge 0, \ i, j \in \{1, \ldots, N\}. \end{aligned}$$

Computation of $d_{\rm OD}$: For each O-D pair in the HITS database, we assume the trip takes place on the shortest path (as measured by distance) connecting O and D. Shortest path algorithms, e.g., Dijkstra's algorithm, are computationally efficient, allowing calculations to be run on a detailed roadmap of Singapore. The average trip distance in interval k is $d_{\rm OD}^k$. For an entire day, $d_{\rm OD}$ was 9.4 km.

Computation of v: The GPS traces of taxis in Singapore were used to estimate average traffic speeds throughout the day. To determine how fast, on average, an individual taxi travels during hour k, the total distance traveled by the taxi, with a passenger on board, was divided by the total associated

time for each hour k. These quantities were then averaged over all taxis active in hour k to estimate v^k the average speed of travel during hour $k = 0, \ldots, 23$. Collectively, these parameters will be used in Section VI to find the minimum number of robotic vehicles required for an AMoD system to service all travel demands in Singapore.

IV. Lumped spatial-queueing model of an AMoD system

A. Overview of lumped model

Within the lumped approach [14], customers are assumed to arrive at a set of stations located within a given environment[1], similar to the hypercube model [20]. The arrival process at each station is Poisson with rate λ_i, where $i \in \{1, \ldots, N\}$ and N denotes the number of stations. (Reasonable deviations from the assumption of Poisson arrivals have been found not to substantially alter the predictive accuracy of these models [20].) Upon arrival, a customer at station i selects a destination j according to a probability mass function $\{p_{ij}\}$ (Figure 4). If vehicles are parked at station i, the customer takes a vehicle and is driven to the intended destination, with a travel time modeled as a random variable T_{ij}. However, if the station is empty of vehicles, the customer immediately leaves the system (this is usually referred to as a loss model, which models well systems with impatient customers or systems with high quality of service). The lumped model captures the performance characteristics of an AMoD system by leveraging the rich theory of queueing networks, specifically Jackson networks. The next section reviews several important results and techniques that will allow us to analyze an AMoD system as a *closed Jackson network*.

B. Review of Jackson Networks

A queueing network is a directed graph containing $|\mathcal{N}|$ nodes or queues, where $\mathcal{N}$ is the set of nodes in the network. Discrete agents (usually referred to as customers in the literature) arrive at each node from outside the network according to a stochastic process or move among the nodes. Each agent arriving at a node is serviced by that node before traveling to another node or leaving the network. A network is *closed* if the number of agents within the network remains constant and no agent enters or leaves the network. In contrast, in an open network, agents arrive from outside of the network and eventually leave the network. Though in general, many types of queues can exist in a queueing network (for example, first-come first-serve, last-come first-serve, processor-sharing, etc.), we only consider queues where agents are serviced on a first-come first-serve basis. A Jackson network is a Markovian queueing network where agents move among the nodes according to a stationary routing distribution r_{ij} and the service rate at each node i, $\mu_i(x_i)$, depends only on the number of agents at that node, x_i [34, p.9]. Jackson networks are part of a more general class of queueing networks called BCMP networks [35] that are known to have analytically tractable product-form equilibrium distributions. In equilibrium, the average number of agents moving through a node per unit time (referred to as throughput) satisfies the balance equations

$$\pi_i = \sum_{j \in \mathcal{N}} \pi_j r_{ji} \quad \forall i \in \mathcal{N}. \tag{6}$$

For a closed network, solving (6) only determines the throughputs to a constant factor, hence $\{\pi\}$ is referred to as the relative throughput. The real throughput is determined by a normalization constant, $G(m)$, that arises in the stationary probability distribution of the closed Jackson network, and is dependent on the number of agents, m, in the system. Explicitly, $G(m)$ is given by

$$G(m) = \sum_{x \in \Omega_m} \prod_{j=1}^{|\mathcal{N}|} \pi_j^{x_j} \prod_{n=1}^{x_j} \mu_j(n)^{-1},$$

where $\Omega_m = \{x = (x_1, x_2, ..., x_{|\mathcal{N}|}) : \sum_{i=1}^{|\mathcal{N}|} x_i = m, x_i \in \mathbb{Z}_{\geq 0}\}$ is the state space of the closed Jackson network. Given $G(m)$, the actual throughput of each node is

$$\Lambda_i(m) = \pi_i \frac{G(m-1)}{G(m)}. \tag{7}$$

The probability that at least one agent is waiting at node i is given by dividing the throughput by the service rate at node i

$$A_i(m) = \frac{\pi_i}{\mu_i(1)} \frac{G(m-1)}{G(m)} = \gamma_i \frac{G(m-1)}{G(m)}, \tag{8}$$

where γ_i is referred to as the relative utilization of node i. $A_i(m)$ is referred to as the *availability* of node i [14], [36] because as long as a vehicle (which are the agents in our queueing network) is at a station (a node), it is available for customers to rent. This is an important performance metric for quality of service, and it has been shown that high availability throughout the network corresponds to low customer wait times [14]. Computing the availability using (8) is non-ideal because in general the normalization constant $G(m)$ is expensive to compute. Section IV-D describes a method to calculate the availability and the mean value of the throughput without explicitly computing $G(m)$.

C. Developing a lumped model for AMoD systems

Under the assumptions of Poisson arrivals and exponentially-distributed travel times, an AMoD system is then translated into a closed Jackson network model through an *abstraction procedure* [14], [36], whereby we identify the stations with single-server queues and the roads with infinite-server queues (note that this does not take into account traffic congestion). We let S be the set of single-server station nodes and I be the infinite-server road nodes. The agents in the Jackson network are the vehicles in the AMoD system (the network is closed because there

[1] Alternatively, to model an AMoD system where the vehicles directly pick up the customers, we would decompose a city into N disjoint *regions* $Q_1, Q_2, \ldots, Q_N$. Such regions would replace the notion of stations. When a customer arrives in region Q_i, destined for Q_j, a free vehicle in Q_i is sent to pick up and drop off the customer before parking at the median of Q_j. The two models are then formally *identical* and follow the same mathematical treatment.

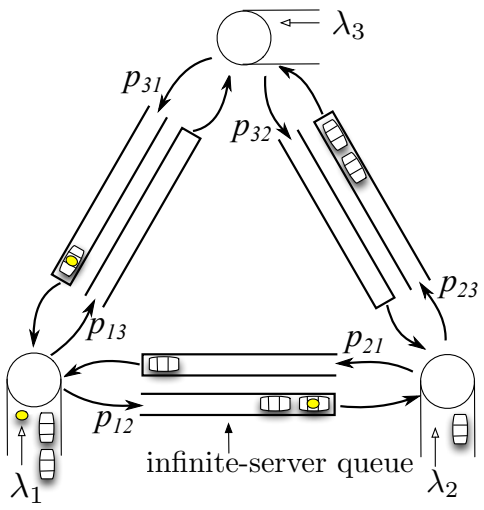


Fig. 4. In the lumped model, an AMoD system is modeled as a Jackson network, where stations are identified with single-server queues and roads are identified with *infinite-server* queues. (Customers are denoted with yellow dots and servicing vehicles are represented by small car icons.) Some vehicles travel without passengers to rebalance the fleet.

is a constant number of vehicles). The Poisson arrival process of customers can be viewed as the exponentially distributed service rate of vehicles at each station node (i.e. $\mu_i(1) = \lambda_i$). The mean service rate at the infinite-server road queue between station i and j is the inverse of the mean travel time, T_{ij}. With this identification, an AMoD system becomes a *closed Jackson network with respect to the vehicles*, which is amenable to *analytical* treatment [14] (Figure 4).

To control the network, the strategy is to autonomously rebalance the vehicles to ensure even vehicle availability (i.e. $A_i(m) = A_j(m)\ \forall i, j$). Remarkably, this condition achieves the dual purpose of maximizing the availabilities across all stations [14], [36]. Rebalancing can be modeled by the addition of *virtual customer streams* [14]. Specifically, we assume that each station i generates "virtual customers" according to a Poisson process with rate ψ_i, independent of the real customer arrival process, and routes these virtual customers to station j with probability α_{ij}. The problem of controlling an AMoD system becomes one of optimizing over the rates $\{\psi_i\}$ and probabilities $\{\alpha_{ij}\}$ with the goal of minimizing the number of rebalancing vehicles on the road. By exploiting the theory of Jackson networks, this can be cast into the following *linear program*

$$\begin{aligned} &\underset{\beta_{ij}}{\textit{minimize}} && \sum_{i,j} T_{ij}\beta_{ij} \\ &\textit{subject to} && \sum_{j\neq i}(\beta_{ij}-\beta_{ji}) = -\lambda_i + \sum_{j\neq i} p_{ji}\lambda_j \\ & && \beta_{ij} \geq 0 \end{aligned} \tag{9}$$

Let $\{\beta^*_{ij}\}_{ij}$ denote an optimal solution to (9). The virtual customer rates are found by making the following substitutions

$$\psi_i = \sum_{j\neq i} \beta^*_{ij}$$

$$\alpha_{ij} = \begin{cases} 0 & \textit{if } i = j \\ \beta^*_{ij}/\psi_i & \textit{if } \psi_i > 0 \\ 1/(N-1) & \textit{otherwise.} \end{cases}$$

The virtual customers process can then be combined with the real customer process to yield a closed Jackson network with total customer arrival rates (both real and virtual)

$$\lambda_i^{\text{tot}} = \lambda_i + \psi_i,$$

and routing probabilities

$$p_{ij}^{\text{tot}} = \alpha_{ij}\frac{\psi_i}{\lambda_i^{\text{tot}}} + p_{ij}\frac{\lambda_i}{\lambda_i^{\text{tot}}}.$$

The rebalancing approach maintains analytic tractability in the Jackson network because the virtual customers are also subject to loss if vehicles are not available, thus rebalancing is "encouraged" rather than enforced. Once the parameters in the Jackson network (λ_i^{tot}, p_{ij}^{tot}, and T_{ij}) have been computed, the performance metric (i.e. vehicle availability $A_i(m)$) can be computed, as discussed in the next section.

D. Computation of availability metric

The vehicle availability $A_i(m)$ can be computed via a technique known as mean value analysis (MVA), which iteratively calculates the mean wait times $W_i(n)$ and mean queue lengths $L_i(n)$ at each node i of the closed Jackson network, where $n = 1, ..., m$ is the number of vehicles over which the algorithm iterates. To use the MVA algorithm, first the relative throughputs π_i must be solved using (6). Due to the special structure of the AMoD model, (6) can be solved just in terms of the single-server station nodes [14] as

$$\pi_i = \sum_{j\in S} \pi_j p_{ji}^{\text{tot}} \quad \forall i \in S. \tag{10}$$

The throughput of the infinite-server node $\{ij\}$ is given by

$$\pi_{ij} = \pi_i p_{ij}^{\text{tot}}. \tag{11}$$

The MVA algorithm proceeds as follows:

Algorithm 1 Mean Value Analysis

function MVA(π_i, π_{ij})
  $W_i(0) \leftarrow 0$, $L_i(0) \leftarrow 0$
  for $n = 1 \rightarrow m$ **do**
    $W_i(n) = (1 + L_i(n-1))/\lambda_i^{\text{tot}}, \quad \forall i \in S$
    $W_{ij}(n) = T_{ij}, \quad \forall \{ij\} \in I$
    $L_i(n) = \frac{n\pi_i W_i(n)}{\sum_{j\in S}\pi_j W_j(n) + \sum_{\{jk\}\in I}\pi_{jk}W_{jk}}, \quad \forall i \in S$
  return $W_i(m)$, $L_i(m)$

The throughput of station i is then given by Little's theorem [37, p.152]

$$\Lambda_i(m) = \frac{L_i(m)}{W_i(m)}, \tag{12}$$

and finally, the availability is given by

$$A_i(m) = \frac{\Lambda_i(m)}{\lambda_i^{\text{tot}}}. \tag{13}$$

E. Calibration of the model

The parameters of the lumped model, N, λ_i, p_{ij}, and T_{ij}, are calibrated in a way similar to the distributed model. For the case study of Singapore, the HITS database was used to calculate the number of stations N, the arrival rates λ_i, and the routing distributions p_{ij}. The taxi database was used to estimate the average speed, which is then used to calculate the mean travel times T_{ij}. For the case study of New York City, a database of taxi trips (including trip origins, destinations, and travel times) was used courtesy of the New York City Taxi & Limousine Commission. This database was used to calibrate all of the model parameters.

The set of trip origin and destinations are clustered into N stations. K-means clustering is used to determine the locations of the stations. N can be chosen based on the average distance a demand is from the closest station. For the New York City case study, $N = 100$ stations in Manhattan meant that a demand is on average less than 300m from the nearest station. Arrivals within each cluster is assigned to the station at the center of the cluster. Once the locations of the stations have been chosen, λ_i and p_{ij} are computed by simply counting the number of arrivals at each cluster and their destinations. If the AMoD system operates in such a way that vehicles provide door-to-door service, the number of "stations" can be a flexible parameter that changes dynamically based on current demand distribution, and the locations of the stations can be inferred using Bayesian nonparametric algorithms such as DP-means [38]. This is an interesting avenue for further research.

F. Comparison of the two approaches

The lumped approach and the distributed approach are *complementary* in a number of ways. Both models provide *formal guarantees* for stability and performance. The former is more realistic (a road topology can be readily mapped into this model) and provides a natural pathway to synthesize control policies. The latter provides significant mathematical simplifications (as we only need to study a spatially-averaged queue) and enables the determination of *analytic* scaling laws that can be used to select system parameters (e.g., fleet sizing). In section VI, we will exploit the interplay between such two approaches to characterize AMoD systems for case studies of New York City and Singapore.

V. Control of AMoD Systems

In this section, we discuss the synthesis of closed-loop control policies within, respectively, the distributed and the lumped models.

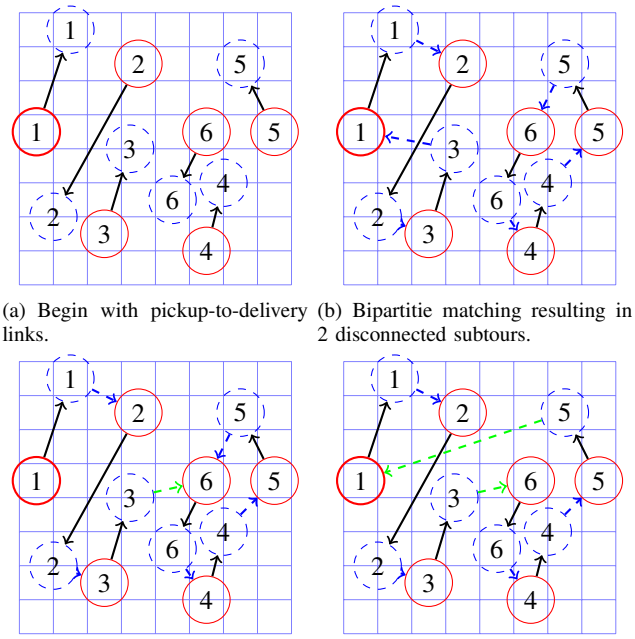
A. Synthesis of closed-loop policies for the distributed model

A simple, yet effective, class of control policy is represented by *gated* policies, whereby *static* instances of the routing problem are repeatedly solved in a receding horizon fashion. Specifically, the strategy is to repeatedly perform the following steps any time *all* servers are idle: (1) solve a static pickup and delivery problem through the outstanding travel demands, (2) splits the tour into m equal length fragments (where m is the number of vehicles), and (3) assigns a fragment to each vehicle. A few comments are in order. First, this approach relies on the capability of solving large instances of pickup-and-delivery problems relatively quickly; we present below an algorithm, referred to as SPLICE [17], which fulfills such requirement. Second, instead of computing a single tour (step 1) and then split it into m fragments (step 2), we could directly compute m tours, one for each vehicle. In the limit where the number of trips n goes to infinity, the two strategies become equivalent. Finally, re-optimization occurs when all vehicles are idle. This mainly serves to maintain tractability in the analysis of the control policy; in reality we would consider an asynchronous implementation whereby new tours are computed as soon as a vehicle becomes idle.

As explained above, the performance of a gated policy hinges upon a computationally efficient and high quality solution to the problem of coordinating the pickup and delivery of n points, for large n, as discussed in [17]. More specifically, let the pickup and delivery locations be $\{x_1, x_2, \ldots, x_n\}$ and $\{y_1, y_2, \ldots, y_n\}$, respectively. For unit capacity service vehicles, x_i must be delivered to y_i immediately after being picked up, $i = 1, \ldots, n$. The challenge in servicing demand therefore lies in finding an effective rule for transitioning from delivery locations to subsequent pickup locations, i.e., from y_i's to x_j's.

The problem of finding a minimal-length tour through the x_i's and y_i's, subject to the aforementioned unit capacity constraint, is referred to as the stacker crane problem (SCP). At its core, the SCP is an instance of a minimal bipartite matching problem that seeks to efficiently link delivery and pickup locations of otherwise unrelated demands. This matching may be solved by a number algorithms, e.g., the Hungarian algorithm, that are constant-factor optimal and run in polynomial time. However, these algorithms generally produce disjoint sub-tours, i.e., two or more cyclic paths through the $\{x_i\}$ and $\{y_i\}$, but no single uninterrupted path.

In the interest of deploying service vehicles and establishing key analytic results, it is preferable to have a single tour through all the $\{x_i\}$ and $\{y_i\}$, i.e., a stacker crane tour. Fortunately, the various sub-tours can be readily rewired by joining or splicing sub-tours to create the desired stacker-crane tour. Moreover, as n increases, the number of disjoint sub-tours grows as $O(\log n)$, and the process of finding a continuous stacker crane tour is dominated by solving the initial the bipartite matching problem. As a reference, [39] provides a constant-factor optimal solution that is $O(n^{2+\epsilon})$ for any $\epsilon > 0$. In [17], the processes by which disjoint sub-tours are fused into a single stacker-crane tour through all $\{x_i\}$ and $\{y_i\}$ is referred to as the SPLICE algorithm. The structure of the SPLICE algorithm is described in Algorithm 2. Using similar scaling arguments, it can be shown that the tours specified by the SPLICE algorithm are asymptotically optimal with respect to distance. Figure 5



(a) Begin with pickup-to-delivery links.

(b) Bipartitie matching resulting in 2 disconnected subtours.

(c) Connect the closest nodes between the subtours.

(d) Complete the cycle to form the stacker crane tour.

Fig. 5. Stages of the SPLICE algorithm, beginning with 6 pickup and delivery locations on the Euclidean plane, and ending with a complete stacker crane tour.

illustrates the general functionality of the SPLICE algorithm on a problem of modest size.

Algorithm 2 SPLICE

INPUT: a set of demands $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $n > 1$

OUTPUT: a stacker crane tour through $\mathcal{S}$.

1: **initialize** $\sigma \leftarrow$ solution to the Euclidean bipartite matching problem between the sets $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_n\}$ computed by using bipartite matching algorithm $\mathcal{M}$.
2: Add the n pickup-to-delivery links $x_i \rightarrow y_i$, $i = 1, \ldots, n$.
3: Add the n matching links $y_i \rightarrow x_{\sigma(i)}$, $i = 1, \ldots, n$.
4: Apply the rewiring heuristic to connect sub-tours.

In [17], the authors show that a gated policy relying on SPLICE is stabilizing whenever the condition for stability (2) is satisfied.

B. Synthesis of closed-loop policies for the lumped model

Synthesis of control policies within the lumped model still relies on receding horizon optimization [40], but at each optimization step one solves the optimization problem introduced in Section IV-C (using the current distribution of vehicles and waiting customers), rather than a static stacker crane problem (as it was done in the distributed model). Specifically, we let $v_i^{\text{own}}(t)$ be the number of vehicles *owned* by station i, defined as

$$v_i^{\text{own}}(t) = v_i(t) + \sum_j v_{ji}(t),$$

where $v_i(t)$ is the number of vehicles currently at station i and $v_{ji}(t)$ is the number of vehicles currently traveling from station j to i. Let $c_i(t)$ be the number of customers at station i. We can define the "excess" number of vehicles at station i to be

$$v_i^{\text{excess}}(t) = v_i^{\text{own}}(t) - c_i(t).$$

These are the vehicles that a station can send to another station to rebalance the system. The total number of excess vehicles is given by $\sum_i v_i^{\text{excess}}(t) = m - \sum_i c_i(t)$. The goal is to distribute these excess vehicles appropriately among the stations. One way to distribute is to divide the vehicles evenly [40], so that the desired number of vehicles for each station, $v_i^d(t)$, is

$$v_i^d(t) = \left\lfloor \frac{m - \sum_j c_j(t)}{n} \right\rfloor.$$

If the customer arrival rates are known or can be estimated, the vehicles can be distributed based on the demand

$$v_i^d(t) = \left\lfloor \frac{\lambda_i(t)(m - \sum_j c_j(t))}{\sum_j \lambda_j(t)} \right\rfloor.$$

Finally, we let n_{ij}^v represent the number of rebalancing vehicles to send from station i to j. The following optimization procedure solves for n_{ij}^v every t_{hor} time steps

$$\begin{aligned} &\underset{n_{ij}^v}{\textit{minimize}} && \sum_{i,j} T_{ij} n_{ij}^v \\ &\textit{subject to} && v_i^{\text{excess}}(t) + \sum_{j \neq i} (n_{ji}^v - n_{ij}^v) \geq v_i^d(t) \\ & && n_{ij}^v \in \mathbb{N}. \end{aligned} \tag{14}$$

It is worth noting that the contraint matrix of this integer linear program is *totally unimodular*, and hence can be solved as a linear program and the solution will necessarily take on integer values [40]. This closed-loop control policy is used to evaluate AMoD systems in Section VI.

Parallel to our discussion in Section IV-F, the control strategies for the distributed model and lumped model are complementary. Both control policies apply receding horizon techniques to optimization problems (combinatorial optimization for the distributed model and linear optimization for the lumped model). The distributed control policy is amenable to analytic treatment and provides performance guarantees, and while the lumped control policy does not provide such guarantees, it is easier to implement and extend in practice (e.g. the inclusion of additional constraints).

VI. Evaluating AMoD systems

Leveraging models and methods from Sections III and IV, this section studies hypothetical deployments of AMoD systems in two major cities, namely New York City and Singapore. Collectively, the results presented in this section provide a preliminary, yet rigorous evaluation of the benefits of AMoD systems based on real-world data. We mention that both case studies do not consider congestion effects – a preliminary discussion about these effects is presented in Section VII. A financial analysis compares the cost of

mobility in an AMoD system to that in a traditional model based on private vehicle ownership. Finally, we describe how the lumped model can be used to model MoD systems where humans perform vehicle rebalancing.

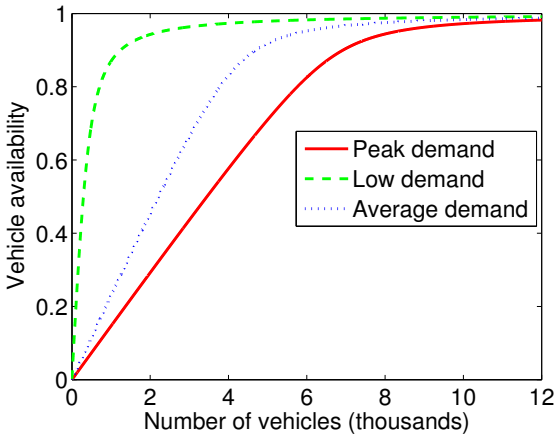
A. Case Study I: AMoD in Manhattan

This case study applies the lumped approach to characterize how many self-driving vehicles in an AMoD system would be required to replace the current fleet of taxis in Manhattan while providing quality service at current customer demand levels [14]. In 2012, over 13,300 taxis in New York City made over 15 million trips a month or 500,000 trips a day, with around 85% of trips within Manhattan. The study uses taxi trip data collected on March 1, 2012 (the data is courtesy of the New York City Taxi & Limousine Commission) consisting of 439,950 trips within Manhattan. First, trip origins and destinations are clustered into $N = 100$ stations, so that a demand is on average less than 300m from the nearest station, or approximately a 3-minute walk. The system parameters such as arrival rates $\{\lambda_i\}$, destination preferences $\{p_{ij}\}$ and travel times $\{T_{ij}\}$ are estimated for each hour of the day using trip data between each pair of stations.

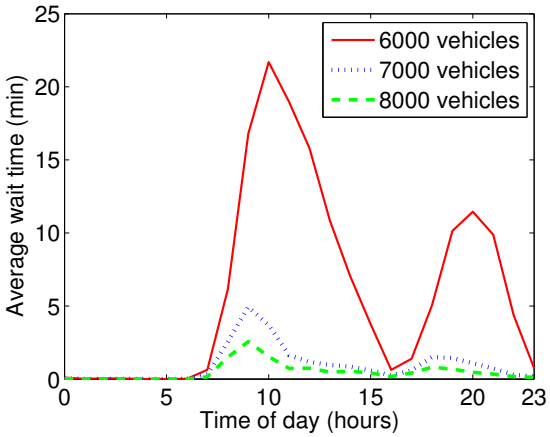
Vehicle availability (i.e., probability of finding a vehicle when walking to a station) is calculated for 3 cases – peak demand (29,485 demands/hour, 7-8pm), low demand (1,982 demands/hour, 4-5am), and average demand (16,930 demands/hour, 4-5pm). For each case, vehicle availability is calculated by solving the linear program discussed in Section IV-C and then applying mean value analysis [36] techniques to recover vehicle availabilities. (The interested reader is referred to [14] for further details). The results are summarized in Figure 6(a).

For high vehicle availability (say, 95%), we would need around 8,000 vehicles (∼70% of the current fleet size operating in Manhattan, which, based on taxi trip data, we approximate as 85% of the total taxi fleet) at peak demand and 6,000 vehicles at average demand. This suggests that an AMoD system with 8,000 vehicles would be able to meet 95% of the taxi demand in Manhattan, assuming 5% of passengers are impatient and are lost when a vehicle is not immediately available. However, in a real system, passengers would wait in line for the next vehicle rather than leave the system, thus it is important to determine how vehicle availability relates to customer waiting times. Customer waiting times are characterized through simulation, using the receding horizon control scheme mentioned in Section V-B. The time-varying system parameters λ_i, p_{ij}, and average speed are piecewise constant, and change each hour based on values estimated from the taxi data. Travel times T_{ij} are based on average speed and Manhattan distance between i and j, and autonomous vehicle rebalancing is performed every 15 minutes. Three sets of simulations are performed for 6,000, 7,000, and 8,000 vehicles, and the resulting average waiting times are shown in Figure 6(b).

Figure 6(b) shows that for a 7,000 vehicle fleet, the peak averaged wait time is less than 5 minutes (9-10am) and for



(a)



(b)

Fig. 6. 6(a): Vehicle availability as a function of system size for 100 stations in Manhattan. Availability is calculated for peak demand (7-8pm), low demand (4-5am), and average demand (4-5pm). 6(b): Average customer wait times over the course of a day, for systems of different sizes.

8,000 vehicles, the average wait time is only 2.5 minutes. The simulation results show that high availability (90-95%) does indeed correspond to low customer waiting time and that an AMoD system with 7,000 to 8,000 vehicles (roughly 70% of the size of the current taxi fleet) can provide adequate service with current taxi demand levels in Manhattan.

B. Case Study II: AMoD in Singapore

This case study discusses an hypothetical deployment of an AMoD systems in Singapore to replace its *entire* transportation infrastructure [19]. The study, which should be interpreted as a thought experiment to investigate the potential benefits of an AMoD solution, addresses three main dimensions (i) minimum fleet size to ensure system stability (i.e., uniform boundedness of the number of outstanding customers), (ii) fleet size to provide acceptable quality of service at current customer demand levels, and (iii) financial estimates to assess economic feasibility.

1) Minimum fleet sizing: In Section III-B, it was argued that an hour was an appropriate time scale over which to compute key parameters of the distributed model. Conse-

quently, we estimate fleet size by modifying the bound in (2) to reflect hourly values, i.e.,

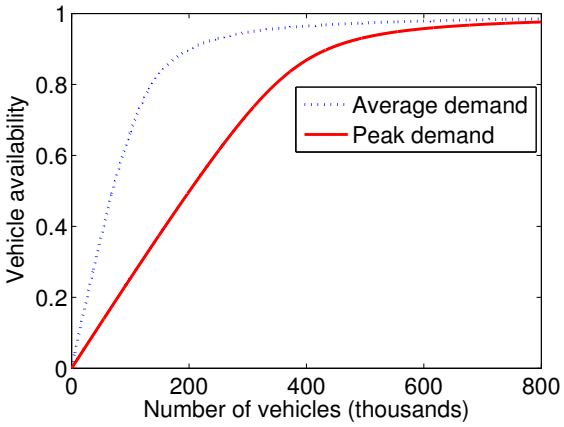
$$m_{min} = \frac{\sum_k \lambda_k (d^k_{\mathrm{OD}} + EMD^k(\varphi^k_{\mathrm{O}}, \varphi^k_{\mathrm{D}}))}{\sum_k v^k}. \qquad (15)$$

Computing the necessary quantities as in subsection III-B.2, equation (15) yields that at least 92,693 self-driving vehicles are required to ensure the transportation demand remains uniformly bounded. To gain an appreciation for the level of vehicle sharing possible in an AMoD system of this size, consider that at 1,144,400 households in Singapore, there would be roughly one shared car for every 12.3 households. Note however, that this should only be seen as a lower bound on the fleet size, since customer wait times would be unacceptably high.

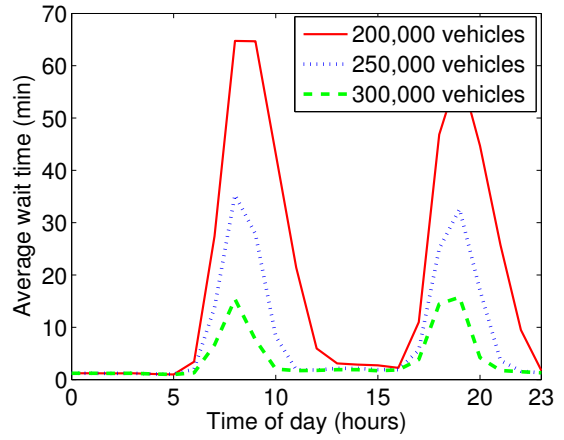
2) Fleet sizing for acceptable quality of service: To ensure acceptable quality of service, we need to increase the fleet size. To characterize such increase, we use the same techniques outlined in Section VI-A, which rely on the lumped approach. Vehicle availability is analyzed in two representative cases. The first is chosen as the 2-3pm bin, since it is the one that is the closest to the "average" traffic condition. The second case considers the 7-8am rush-hour peak. Results are summarized in Figure 7(a). With about 200,000 vehicles availability is about 90% on average, but drops to about 50% at peak times. With 300,000 vehicles in the fleet, availability is about 95% on average and about 72% at peak times. As in Section VI-A, waiting times are characterized through simulation. For 250,000 vehicles, the maximum wait times during peak hours is around 30 minutes, which is comparable with typical congestion delays during rush hour. With 300,000 vehicles, peak wait times are reduced to less than 15 minutes. To put these numbers into perspective, in 2011 there were 779,890 passenger vehicles operating in Singapore [41]. Hence, this case study suggests that an AMoD system can meet the personal mobility needs of the entire population of Singapore with a number of robotic vehicles that is less than 40% of the current number of passenger vehicles.

C. Financial analysis of AMoD systems

This section finally provides a preliminary, yet rigorous economic evaluation of AMoD systems, as first introduced in [19]. Specifically, this section characterizes the total mobility cost (TMC) for users in two competing transportation models. In System 1 (referred to as *traditional system*), users access personal mobility by purchasing (or leasing) a private, human-driven vehicle (PHDV). Conversely, in System 2 (the AMoD system), users access personal mobility by subscribing to a shared AMoD fleet of vehicles. For both systems, the analysis considers not only the explicit costs of access to mobility (referred to as cost of service -COS-), but also hidden costs attributed to the time invested in various mobility-related activities (referred to as cost of time -COT-). A subscript $i = \{1, 2\}$ will denote the system under consideration (e.g., COS_1 denotes the COS for System 1).



(a)



(b)

Fig. 7. 7(a): Performance curve for Singapore with 100 stations, showing the availability of vehicles vs. the size of the system for both average demand (2-3pm) and peak demand (7-8am). 7(b): Average wait times over the course of a day, for systems of different sizes.

Cost of service: The cost of service is defined as the sum of all explicit costs associated with accessing mobility. For example, in System 1, COS_1 reflects the costs to individually purchase, service, park, insure, and fuel a PHDV, which, for the case of Singapore, is estimated for a mid-size car at $\$18,162$/year. For System 2, we need to make an educated guess for the cost incurred with retrofitting production vehicles with the sensors, actuators, and computational power required for automated driving. Based upon the authors' experiences on autonomous vehicles, such cost (assuming some economies of scale for large fleets) is estimated as a one-time fee of $\$15,000$. From the fleet-sizing arguments of Section VI-B.2, one self-driving vehicle in System 2 can effectively serve the role of about 4 PHDVs in System 1, which implies an estimate of 2.5 years for the average lifespan of a self-driving vehicle. Tallying the aforementioned costs on a fleet-wide scale and distributing the sum evenly among the entire Singapore population gives a COS_2 of $12,563/year (see [19] for further details about the cost breakdown). According to COS values, it is more affordable to access mobility in System 2 than System 1.

TABLE I

SUMMARY OF THE FINANCIAL ANALYSIS OF MOBILITY-RELATED COST FOR TRADITIONAL AND AMOD SYSTEMS.

	Cost [USD/km]			Yearly cost [USD/year]		
	COS	COT	**TMC**	COS	COT	**TMC**
Traditional	0.96	0.76	**1.72**	18,162	14,460	**32,622**
AMoD	0.66	0.26	**0.92**	12,563	4,959	**17,522**

Cost of time: To monetize the hidden costs attributed to the time invested in mobility-related activities, the analysis leverages the Value of Travel Time Savings (VTTS) numbers laid out by the Department of Transportation for performing a Cost Benefit Analysis of transportation scenarios in the US [42]. Applying the appropriate VTTS values based on actual driving patterns gives COT_1 = \$14,460/year (which considers an estimated 747 hours/year spent by vehicle owners in Singapore in mobility-related activities, see [19]). To compute COT_2, this analysis prices sitting comfortably in a self-driving vehicle while being able to work, read, or simply relax at 20% of the median wage (as opposed to 50% of the median wage which is the cost of time for driving in free-flowing traffic). Coupling this figure with the facts that a user would spend no time parking, limited time walking to and from the vehicles, and roughly 5 minutes for a requested vehicle to show up (see Section VI-B.2), the end result is a COT_2 equal to \$4,959/year.

Total mobility cost: A summary of the COS, COT, and TMC for the traditional and AMoD systems is provided in Table I (note that the average Singaporean drives 18,997 km each year [19]). Remarkably, combining COS and COT figures, the *TMC for AMoD systems is roughly half of that for a traditional system*. To put this into perspective, these savings represent about one third of GDP per capita. Hence, this analysis reveals it is much more affordable to access mobility in an AMoD system compared to traditional mobility models based on private vehicle ownership.

D. Comparing AMoD to human-driven MoD systems

We can directly compare the potential of AMoD systems to current human-driven MoD systems such as Car2Go [9]. Remarkably, the queueing network approach used in the lumped model can be extended to model human-driven MoD systems. A human-driven MoD system consists of m_v (non-autonomous) vehicles, and rebalancing is performed by a team of m_d hired drivers (called "rebalancers") [43]. As the drivers rebalance the vehicles, however, they themselves become unbalanced. To "rebalance the rebalancers", the strategy is to allow the drivers to drive customers to their destinations similar to a taxi service [43], [44]. If we assume that each driver must have access to a vehicle at all times (otherwise the driver would be stranded at a station), the MoD system can be viewed as a customer-driven carsharing system operating in parallel with a taxi service, which we model using two coupled closed Jackson networks. Specifically, we define System 1 to be the customer-driven carsharing system (with $m_v - m_d$ vehicles) and System 2 to be the taxi system (with m_d vehicles). When a customer arrives at a station, he/she is delegated to one of the two systems. If no vehicles are available in the system to which the customer was delegated, the customer immediately leaves the system. This represents an expanded version of the loss model described in Section IV-A. Virtual customers are generated in the same way to rebalance vehicles, but only in System 2 (the taxi system). Instead, the delegation process of customers will serve to balance System 1 in the absence of autonomous cars. In this case, the parameters of the Jackson network for System 1 become

$$\lambda_i^{(1)} = \lambda_i - \lambda_i^{\text{del}},$$
$$p_{ij}^{(1)} = p_{ij}\frac{\lambda_i}{\lambda_i^{(1)}} - \eta_{ij}\frac{\lambda_i^{\text{del}}}{\lambda_i^{(1)}},$$

where λ_i^{del} is the rate of customers at station i delegated to System 2 and η_{ij} is the routing distribution of the customers delegated to System 2. The parameters for System 2 are

$$\lambda_i^{(2)} = \lambda_i^{\text{del}} + \psi_i,$$
$$p_{ij}^{(2)} = \xi_{ij}\frac{\psi_i}{\lambda_i^{(2)}} + \eta_{ij}\frac{\lambda_i^{\text{del}}}{\lambda_i^{(2)}},$$

where ψ_i is the rate of arrival of virtual customers at station i and ξ_i is the routing distribution of virtual customers.

With this formulation, the optimization variables for rebalancing become λ_i^{del}, η_{ij}, ψ_i, and ξ_{ij}. By direct extension of the AMoD analysis, we can define two decoupled linear programs to solve for these parameters while minimizing the number of rebalancing drivers and vehicles.

$$\begin{aligned} &\underset{\beta_{ij}}{\textit{minimize}} && \sum_{i,j} T_{ij}\beta_{ij} \\ &\textit{subject to} && \sum_{j\neq i}(\beta_{ij}-\beta_{ji}) = \lambda_i - \sum_{j\neq i} p_{ji}\lambda_j \\ & && 0 \le \beta_{ij} \le \lambda_i p_{ij} \end{aligned} \tag{16}$$

$$\begin{aligned} &\underset{\alpha_{ij}}{\textit{minimize}} && \sum_{i,j} T_{ij}\alpha_{ij} \\ &\textit{subject to} && \sum_{j\neq i}(\alpha_{ij}-\alpha_{ji}) = -\lambda_i + \sum_{j\neq i} p_{ji}\lambda_j \\ & && \alpha_{ij} \ge 0 \end{aligned} \tag{17}$$

The rebalancing parameters are given by making the following substitutions [44]:

$$\lambda_i^{\text{del}} = \sum_{j\neq i}\beta_{ij}^*,$$
$$\psi_i = \sum_{j\neq i}\alpha_{ij}^*,$$
$$\eta_{ij} = \begin{cases} 0 & \text{if } i=j, \\ \beta_{ij}^*/\lambda_i^{\text{del}} & \text{if } \lambda_i^{\text{del}}>0, i\neq j, \\ 1/(N-1) & \text{otherwise,} \end{cases}$$
$$\xi_{ij} = \begin{cases} 0 & \text{if } i=j, \\ \alpha_{ij}^*/\psi_i & \text{if } \psi_i>0, i\neq j, \\ 1/(N-1) & \text{otherwise,} \end{cases}$$

where β_{ij}^* is the optimal solution to (16) and α_{ij}^* is the optimal solution to (17). The availabilities of each system can then be calculated using the procedure described in Section IV-D. The overall availability for all *real* passengers is found by consolidating the two systems.

$$A_i^{\text{pass}}(m_v, m_d) = A_i^{(1)}(m_v - m_d)\frac{\lambda_i^{(1)}}{\lambda_i} + A_i^{(2)}(m_d)\frac{\lambda_i^{\text{del}}}{\lambda_i}. \tag{18}$$

Since $\lambda_i^{(1)}$ and λ_i^{del} depend on the station i, the availability metric A_i^{pass} is no longer equal across all stations. Figure 8 shows that the availabilities of the stations no longer overlap with one another, even though availabilities still tend towards one as m_v increases. We can also see that the spread of the availabilities across different stations increase with the vehicle-to-driver ratio (m_v/m_d). This is intuitively clear because fewer drivers means fewer rebalancing trips which leads to a more unbalanced system. The red line in Figure 8 shows the availability for an AMoD system. It is clear that AMoD outperforms human-driven MoD systems because every vehicle is able to rebalance at any time. The human-driven MoD system can still be balanced (i.e. all availabilities are equal) at a particular operating point (i.e. m_v and m_d) if equation (18) is used as a constraint in the optimization. However, the optimization will no longer be a linear program (since MVA is needed to calculate (18)). The optimization can be solved using general nonlinear optimization techniques and MVA can be performed quickly for systems of reasonable size. For much larger systems (i.e. tens or hundreds of thousands of vehicles), an approximate MVA technique exists which transforms the iterative algorithm into a set of nonlinear equations [45]. More details of the nonlinear optimization problem can be found in [44].

As we see from Figure 8, the performance of the human-driven MoD system can be heavily dependent on the vehicle-to-driver ratio, m_v/m_d. The greater the number of rebalancing drivers, the easier it is to balance the system. However, hiring drivers comes with a cost most likely higher than the cost of the vehicle itself. It is therefore possible to conduct a financial comparison between an AMoD system and a human-driven MoD system where the human-driven MoD system is assumed to operate at a vehicle-to-driver ratio that minimizes total cost while maintaining an acceptable level of performance. The total cost, for example, can be

$$c_{\text{total}} = m_v + c_r m_d, \tag{19}$$

where c_r is the cost ratio between a vehicle and a driver. While this financial analysis is left for future research, in [44] it is shown that the optimal vehicle-to-driver ratio that minimizes (19) is between 3 and 5 for a wide range of cost ratios. Depending on the cost of labor for drivers, this analysis may provide significant financial incentives for AMoD systems over human-driven MoD systems.

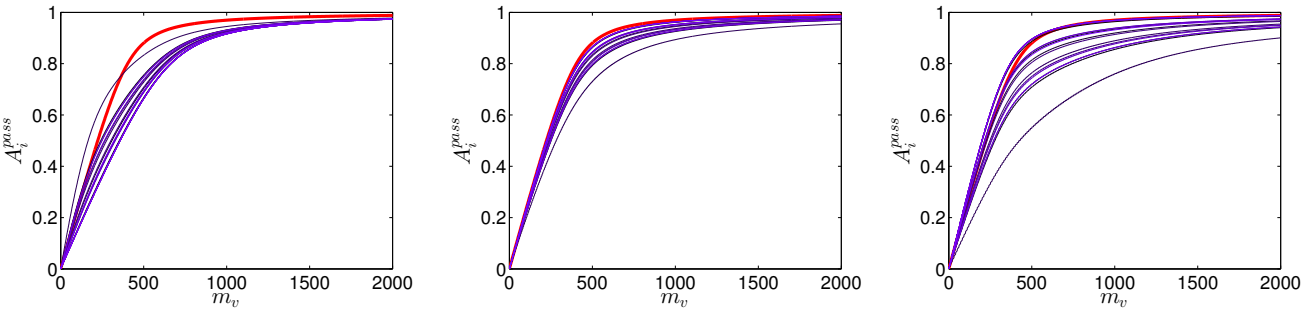
VII. Future research directions

This paper provided an overview of modeling and control techniques for AMoD systems, and a preliminary evaluation of their financial benefits. Future research on this topic should proceed along two main dimensions: efficient control algorithms for increasingly more realistic models and eventually for real-world test beds, and financial analyses for a larger number of deployment options and accounting for positive externalities (e.g., increased safety) in the economic assessment. Such research directions are discussed in some details next, with a particular emphasis on the inclusion of congestion effects and some related preliminary results.

A. Future research on modeling and control

A key direction for future research is the inclusion of congestion effects. In AMoD systems, congestion manifests itself as constraints on the road capacity, which in turn affect travel times throughout the system. To include congestion effects, a promising strategy is to study a modified lumped model whereby the infinite-server road queues are changed to queues with a *finite* number of servers, where the number of servers on each road represents the *capacity* of that road. This approach is used in Figure 9 on a simple 9-station road network, where the aim is to illustrate the impact of autonomously rebalancing vehicles on congestion. Specifically, the stations are placed on a square grid, and joined by 2-way road segments each of which is 0.5 km long. Each road consists of a single lane, with a critical density of 80 vehicles/km. Each vehicle travels at 30 km/h in free flow, which means the travel time along each road segment is 1 minute in free flow. Figure 9 plots the vehicle and road utilization increases due to rebalancing for 500 randomly generated systems (where the arrival rates and routing distributions are randomly generated). The routing algorithm for the rebalancing vehicles is a simple open-loop strategy based on the linear program discussed in Section IV-C. The x-axis shows the ratio of rebalancing vehicles to passenger vehicles on the road, which represents the inherent imbalance in the system. The red data points represent the increase in average road utilization due to rebalancing and the blue data points represent the utilization increase in the most congested road segment due to rebalancing. It is no surprise that the average road utilization rate is a linear function of the number of rebalancing vehicles. However, remarkably, the maximum congestion increases are much lower than the average, and are in most cases zero. This means that while rebalancing generally increases the number of vehicles on the road, *rebalancing vehicles mostly travel along less congested routes and rarely increase the maximum congestion in the system*. This can be seen in the middle figure of Figure 9, where rebalancing clearly increases the number of vehicles on many roads but not on the most congested road segment (from station 6 to station 5).

The simple setup in Figure 9 suggests that AMoD systems would, in general, not lead to an increase in congestion. On the other end, a particularly interesting and intriguing research direction is to devise routing algorithms for AMoD systems that lead to a *decrease* in congestion with current demand levels (or even higher). A promising strategy relies on the idea that if AMoD systems are implemented such



(a) $m_v/m_d = 2$ (b) $m_v/m_d = 5$ (c) $m_v/m_d = 10$

Fig. 8. Overall vehicle availability for passengers for a 20-station human-driven MoD system based on taxi data from Lower Manhattan. The red line shows the availability for an AMoD system with m_v vehicles (or a human-driven MoD system with as many drivers as vehicles). 8(a) shows a vehicle-to-driver ratio of 2, 8(b) shows a vehicle-to-driver ratio of 5, and 8(c) shows a vehicle-to-driver ratio of 10.

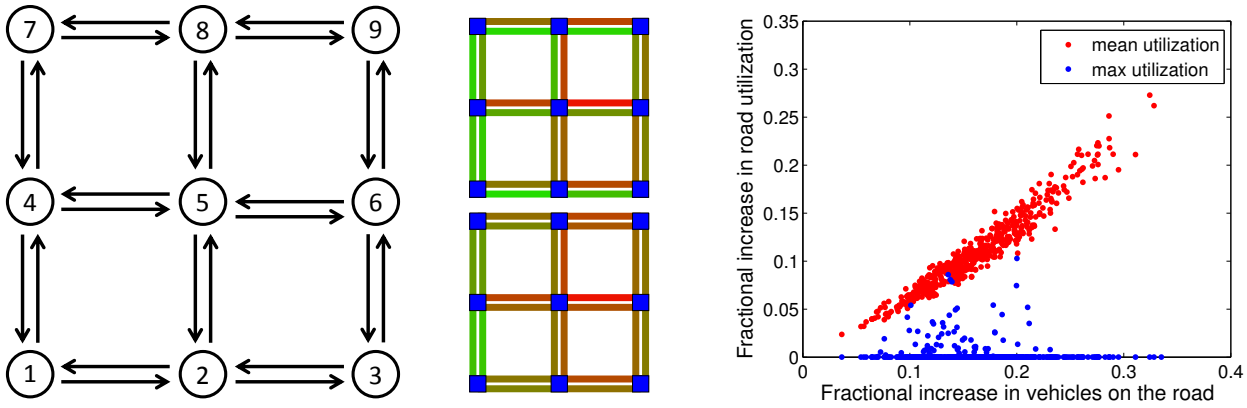


Fig. 9. Left: Layout of the 9-station road network. Each road segment has a capacity of 40 vehicles in each direction. Middle: The top picture shows the 9-station road network without rebalancing. The color on each road segment indicates the level of congestion, where green is no congestion, and red is heavy congestion. The bottom picture is the same road network with rebalancing vehicles. Right: The effects of rebalancing on congestion. The x-axis is the ratio of rebalancing vehicles to passenger vehicles on the road. The y-axis is the fractional increase in road utilization due to rebalancing.

that passengers are given precise pickup times and trips are staggered to avoid too many trips at the same time, congestion may be reduced. Passengers may still spend the same amount of time between requesting a vehicle and arrival at their destination, but the time spent waiting for the vehicle could be used for productive work as opposed to being stuck in traffic. Specifically, for highly congested systems, vehicle departures can be staggered to avoid excessive congestion, and the routing problem is similar to the simultaneous departure and routing problem [46].

Besides congestion, several additional directions are open for future research. As far as modeling is concerned, those include (i) analysis in a time-varying setup (e.g., with periodically time-varying arrival rates), (ii) inclusion of mesoscopic and microscopic effects into the models (e.g., increased throughput due to platooning or automated intersections), and (iii) more complex models for the transportation demands (e.g., time windows or priorities). On the control side, those include (i) inclusion of recharging constraints in the routing process, (ii) control of AMoD systems as part of a *multi-modal* transportation network, which should address synergies between AMoD and alternative transportation modes and interactions with human-driven vehicles, and (iii) deployment of control algorithms on real-world test beds.

B. Future research on AMoD evaluation

The AMoD evaluation presented in Section VI already showed that AMoD systems might hold significant financial benefits. Remarkably, such financial benefits might be even larger when we also account for the positive externalities of an AMoD system, e.g., improved safety, freeing up urban land for other uses, and even creating a new economy based on infotainment systems onboard the autonomous vehicles. Such additional benefits, however, have not been thoroughly characterized yet and require additional analyses. Another research direction involves the evaluation of AMoD systems for more complex deployment options, e.g., as a last-mile solution within a multi-modal transportation system, or with a more sophisticated service structure, e.g., multiple priority classes.

VIII. Conclusions

This paper overviewed recent results regarding the analysis, design, control, and evaluation of autonomous mobility-on-demand systems. Case studies of New York City and Singapore suggest that it would be much more affordable

(and more convenient) to access mobility in an AMoD system compared to traditional mobility models based on private vehicle ownership. More studies are however needed to devise efficient, system-wide coordination algorithms for complex AMoD systems as part of a multi-modal transportation network, and to fully assess the related economic benefits.

ACKNOWLEDGMENTS

The authors acknowledge the collaboration with Kyle Treleaven (MIT) and Daniel Morton (SMART center) on the results presented in this paper. This research was supported in part by the Dr. Cleve B. Moler Stanford Graduate Fellowship.

REFERENCES

[1] W. J. Mitchell, C. E. Borroni-Bird, and L. D. Burns, *Reinventing the Automobile: Personal Urban Mobility for the 21st Century*. Cambridge, MA: The MIT Press, 2010.

[2] U. E. I. Administration, "International Energy Outlook 2013," Tech. Rep., 2013.

[3] U. N. E. Programme, "The Emissions Gap Report 2013 - UNEP," Tech. Rep., 2013.

[4] U. E. P. Agency, "Greenhouse Gas Equivalencies Calculator, online: http://www.epa.gov/cleanenergy/energy-resources/refs.html," Tech. Rep., 2014.

[5] F. H. Administration, "Our Nation's Highways: 2011," Tech. Rep., 2011.

[6] D. Schrank, B. Eisele, and T. Lomax, "TTIs 2012 urban mobility report," 2012, texas A&M Transportation Institute, Texas, USA.

[7] UN, "World urbanization prospects: The 2011 revision population database," United Nations, Tech. Rep., 2011.

[8] A. Santos, N. McGuckin, H. Y. Nakamoto, D. Gray, and S. Liss, "Summary of travel trends: 2009 national household travel survey," Tech. Rep., 2011.

[9] CAR2GO, "CAR2GO Austin. Car Sharing 2.0: Great Idea for a Great City," Tech. Rep., 2011.

[10] J. Motavalli, "G.M. EN-V: Sharpening the focus of future urban mobility," The New York Times, Online: http://wheels.blogs.nytimes.com/2010/03/24/g-m-en-v-sharpening-the-focus-of-future-urban-mobility/, 24 March 2010.

[11] Induct, "Navia - the 100% electric automated transport," Online: http://induct-technology.com/en/products/navia-the-100-electric-automated-transport, 2013.

[12] Google, "Just press go: designing a self-driving vehicle. Online: http://googleblog.blogspot.com/2014/05/just-press-go-designing-self-driving.html," Tech. Rep., 2014.

[13] M. Pavone, "Autonomous mobility-on-demand systems for future urban mobility," in *Springer-Daimler Book on Autonomous Driving*, 2015, in press.

[14] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: a queueing-theoretical perspective," in *Robotics: Science and Systems Conference*, 2014, Best Paper Award Finalist.

[15] M. Pavone, "Dynamic vehicle routing for robotic networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.

[16] K. Treleaven, M. Pavone, and E. Frazzoli, "Models and asymptotically optimal algorithms for pickup and delivery problems on roadmaps," in *Proc. IEEE Conf. on Decision and Control*, 2012, pp. 5691–5698.

[17] ——, "Asymptotically optimal algorithms for one-to-one pickup and delivery problems with applications to transportation systems," *IEEE Trans. on Automatic Control*, vol. 58, no. 9, pp. 2261–2276, 2013.

[18] E. Frazzoli and M. Pavone, "Multi-vehicle routing," in *Springer Encyclopedia of Systems and Control*. Springer, 2014.

[19] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in Singapore," in *Road Vehicle Automation*. Springer, 2014.

[20] R. C. Larson and A. R. Odoni, *Urban operations research*. Prentice-Hall, 1981.

[21] D. J. Bertsimas and G. J. van Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," vol. 39, pp. 601–615, 1991.

[22] D. J. Bertsimas and D. Simchi-Levi, "The new generation of vehicle routing research," vol. 44, pp. 286–304, 1996.

[23] D. J. Bertsimas and G. J. van Ryzin, "Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles," vol. 41, no. 1, pp. 60–76, 1993.

[24] ——, "Stochastic and dynamic vehicle routing with general interarrival and service time distributions," vol. 25, pp. 947–978, 1993.

[25] T. L. Friesz, J. Luque, R. L. Tobin, and B. W. Wie, "Dynamic network traffic assignment considered as a continuous time optimal control problem," *Operations Research*, vol. 37, no. 6, pp. 893–901, 1989.

[26] S. Peeta and A. Ziliaskopoulos, "Foundations of dynamic traffic assignment: The past, the present and the future," *Networks and Spatial Economics*, vol. 1, pp. 233–265, 2001.

[27] E. Feuerstein and L. Stougie, "On-line single-server dial-a-ride problems," *Theoretical Computer Science*, vol. 268, no. 1, pp. 91–105, 2001.

[28] P. Jaillet and M. R. Wagner, "Online routing problems: Value of advanced information and improved competitive ratios," *Transportation Science*, vol. 40, no. 2, pp. 200–210, 2006.

[29] G. Berbeglia, J. F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," vol. 202, no. 1, pp. 8 – 15, 2010.

[30] L. Ruschendorf, "The Wasserstein distance and approximation theorems," *Probability Theory and Related Fields*, vol. 70, pp. 117–129, 1985.

[31] M. Pavone, K. Treleaven, and E. Frazzoli, "Fundamental performance limits and efficient policies for transportation-on-demand systems," in *Proc. IEEE Conf. on Decision and Control*, 2010, pp. 5622–5629.

[32] Singapore Land Transport Authority, "2008 household interview travel survey background information," 2008.

[33] Y. Rubner, C. Tomasi, and L. Guibas, "A metric for distributions with applications to image databases," in *IEEE Conference on Computer Vision*, 1998.

[34] R. Serfozo, *Introduction to stochastic networks*. Springer, 1999, vol. 44.

[35] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," vol. 22, no. 2, pp. 248–260, Apr. 1975.

[36] D. K. George and C. H. Xia, "Fleet-sizing and service availability for a vehicle rental system via closed queueing networks," *European Journal of Operational Research*, vol. 211, no. 1, pp. 198–207, 2011.

[37] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*. Prentice-Hall International, 1992, vol. 2.

[38] B. Kulis and M. I. Jordan, "Revisiting k-means: New algorithms via Bayesian nonparametrics," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, UK, 2012.

[39] P. Agarwal, A. Efrat, and M. Shahir, "Vertical decomposition of shallow levels in 3 dimensional arrangements and its applications," *Proc. of Eleventh Symposium on Computational Geometry*, vol. 211, no. 1, pp. 39–50, 1995.

[40] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 839–854, 2012.

[41] Land Transport Authority, "Singapore land transit statistics in brief," 2012. [Online]. Available: http://www.lta.gov.sg/content/dam/ltaweb/corp/PublicationsResearch/files/FactsandFigures

[42] HEATCO, "Harmonized European approaches for transport costing and project assessment," 2006. [Online]. Available: http://heatco.ier.uni-stuttgart.de

[43] S. L. Smith, M. Pavone, M. Schwager, E. Frazzoli, and D. Rus, "Rebalancing the rebalancers: Optimally routing vehicles and drivers in mobility-on-demand systems," in *Proc. American Control Conf.*, 2013, pp. 2362–2367.

[44] R. Zhang and M. Pavone, "A queueing network approach to the analysis and control of mobility-on-demand systems," in *Proc. American Control Conf.*, 2015, submitted. Available at `http://arxiv.org/abs/1409.6775v2`.

[45] P. Schweitzer, "Approximate analysis of multiclass closed networks of queues," in *Proceedings of International Conference on Stochastic Control and Optimization*, 1979, pp. 25–29.

[46] H. Huang and W. H. K. Lam, "Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues," vol. 36, no. 3, pp. 253–273, 2002.