Autonomy for Mobility on Demand

Z. J. Chong, B. Qin, T. Bandyopadhyay, T. Wongpiromsarn, B. Rebsamen, P. Dai, E. S. Rankin, M. H. Ang Jr.

Abstract We describe the development of our autonomous personal vehicle that attempts to provide mobility on demand service to address the first- and last-mile problem. We discuss the challenges faced for such a system in a campus environment and discuss our approach towards mitigating them. The autonomous vehicle has operated over 30km of autonomous operation in a campus environment interacting with pedestrian and human driven vehicles.

1 Introduction

As the use of private vehicles starts approaching its limits to effectively meet the demand for personal mobility in densely populated cities, mobility-on-demand systems emerge as a more economical and sustainable alternative [3]. These systems rely on the deployment of a fleet of vehicles at different stations that are distributed throughout the city. The customers simply have to walk to a station near their origin, pick up a vehicle, drive it to the station near to their destination and drop it off. Electric ultra-small vehicles or bicycles may be utilized for systems that primarily aim at serving short trips. Such systems can supplement and stimulate the use of public transport by providing a convenient mean for the first- and last-mile transportation (e.g., from home to a transit station and back); thus, improving public transportation accessibility. The feasibility of mobility-on-demand systems that employ traditional bicycles has been demonstrated in many cities [1].

Z. J. Chong, B. Qin, M. H. Ang Jr.
National University of Singapore, e-mail: {chongzj, baoxing.qin, mpeangh}@nus.edu.sg
T. Bandyopadhyay , T. Wongpiromsarn, B. Rebsamen, P. Dai
Singapore-MIT Alliance for Research and Technology, Singapore, e-mail: {tirtha, nok, brice, peilong}@smart.mit.edu
E. S. Rankin
DSO National Laboratories, Singapore, e-mail: erankin@dso.org.sg

One of the main challenges in managing mobility-on-demand systems is in keeping a balanced distribution of the vehicles among different stations to ensure minimal waiting time for the customers at sustainable cost. This problem is critical especially for the cities where some origins and destinations are more popular than others, leading to an unbalanced distribution of the vehicles throughout the city. Hence, most of the existing vehicle sharing systems only offer round-trip service, forcing the customers to return the vehicle only at their origin. In [2], an optimal, real-time rebalancing policy that determines a proper distribution of the vehicles in the anticipation of future demand is proposed. However, a means of transporting the vehicles for the re-balancing trips remains an open problem. In this paper, we propose the use of autonomy to implement the proposed policy and allow efficient operation of mobility-on-demand systems and enable a one-way vehicle sharing option.

Autonomy can play an important role, not only for the re-balancing trips but also for transportation from a pick-up point to a delivery point. This allows the customers to be picked up at their actual origin or dropped off at their actual destination, instead of requiring the customers to walk to or from a station. This problem is closely related to Dynamic one-to-one Pick-up and Delivery problems [4, 5]. We show how this can be accomplished in a fully automatic manner without any human assistance.

Our system aims at providing transportation over a relatively short distance. In particular, the vehicles mainly operate in crowded urban environments that are typically equipped with sensors on the infrastructure including cellular networks, traffic cameras, loop detectors and ERP (Electronic Road Pricing) gantries. The detailed road network and many features of the environment in which the vehicles operate can also be obtained a priori. As opposed to existing autonomous vehicles such as those in the 2007 DARPA Urban Challenge and Google driverless car [6], we take a minimalistic approach and exploit the prior knowledge of the environment features and the availability of the existing infrastructure to ensure that the system is economically feasible.

The rest of the paper is organized as follows. Our mobility-on-demand system is described in Section 2. Section 3 describes our autonomous vehicle, including both the hardware and software components. The operation of the system is demonstrated in Section 4. Finally, Section 5 concludes the paper and discusses future work.

2 Mobility-on-Demand System

The components of our mobility-on-demand system is shown in Figure 1. First, the customers may request or cancel services, specify their pick-up and drop-off locations and view useful service information through a personal electronic device such as a smart phone or through our web interface. Sample

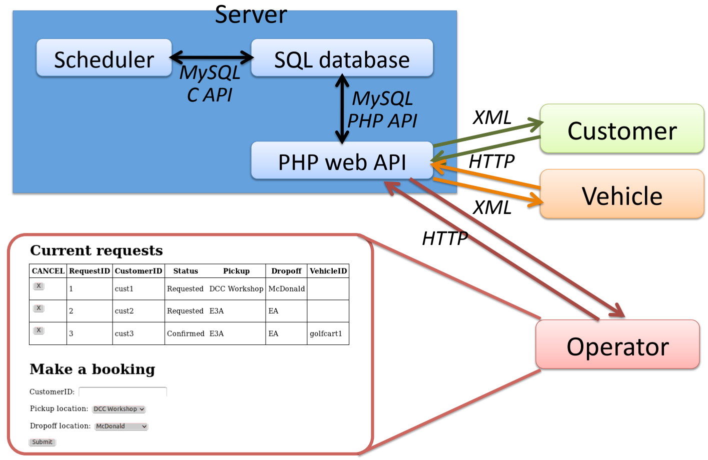
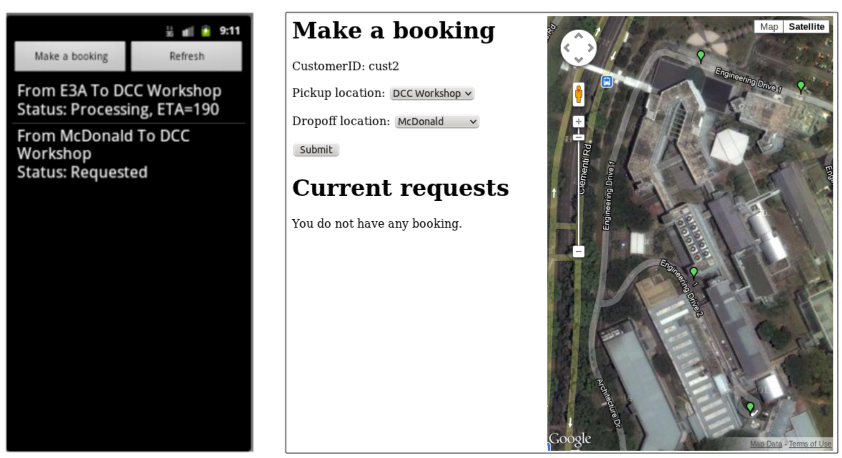


Fig. 1: Mobility-on-demand system

snapshots of our Android mobile phone application and web interface are shown in Figure 2.



(a) Android mobile phone application

(b) Web interface

Fig. 2: Snapshots of service information available

The service requests and cancellation are then added to the SQL database on our server. A scheduler, which is the main component of the server, interacts directly with the database. It determines the order in which the requests will be serviced and assigned each request to a vehicle. Once the assignment is made, the scheduler populates the database with useful service information such as service status, expected waiting time and the vehicle that is assign to each request. This service information will be transfered to the customers through our web API. The server also communicates with each vehicle to obtain its current status and provide the information about its next task (e.g. pick-up and drop-off locations of the next customer the vehicle is supposed to serve).

A vehicle completes each task as follows. First, it has to drive autonomously to a specified pick-up location, come to a complete stop and wait until the customer successfully boards the vehicle. It then goes to the drop-off location. The task is completed when the vehicle reaches the drop-off location, comes to a complete stop and after the customer alights. The vehicle may not start the next task until the current task is completed. More detail on the autonomous operation of our vehicles is provided in the next section. Lastly, a human operator may monitor, add, cancel and modify service requests and access the status of each vehicle through our secure web interface (Figure 2b).

3 Autonomous Personal Transporter

Fig. 3: Autonomous vehicle hardware architecture.

3.1 Hardware Architecture

Our platform is based on a Yamaha G-Max 48 Volt Golf Car G22E. It has a seating capacity of 2 persons with maximum forward speed of 24 km/h. Current setup of the platform is shown in Figure 3 which features the placement of sensors and other hardware. The multifunction frame structure provides flexible sensor configuration which allows changes to the sensors configuration to be done quickly and easily. All hardware are powered by the onboard 6 x 8V US 8VGC deep cycle batteries. Some of the devices which requires AC, i.e. computers and motors get the power supplied from an inverter which draws power directly from the onboard batteries.

Actuators

The golf car has been modified to be able to drive by wire for computer control. An AC motor is connected to the steering column by bevel gear to enable automatic steering. It is designed such that the bevel gear can be disengaged to allow switching back to manual driving. Another AC motor is fitted near the brake pedal to actuate the brake directly. Finally, direct electronic interface into the throttle signal is made to achieve complete control of the vehicle's speed and direction.

The low-level controls, which comprise the controls of steering, throttle and brake, are handled by a realtime system to provide necessary signals as required by different actuators. The 2 AC motors have been configured to receive pulse signals with position controls similar to a stepper motor where the amount of rotation is proportional to the number of pulses. On the other hand, a PWM signal of 3.3 V is used for the throttle to regulate the speed of the vehicle.

Sensors

Both rear wheels of the golf car are mounted with encoders that provide an estimate of the distance traveled. An Inertial Measurement Unit (IMU) MicroStrain 3DM-GX3-25 is mounted at the center of the rear axle to provide attitude and heading of the vehicle. The encoders and IMU are combined to provide odometry information for the vehicle in 6 DOF.

For external sensing, a variety of LIDARs are used. There are 2 SICK LMS 291 mounted in front of the golf car. The SICK LMS 291 provides a single plane range measurement of 180 degree field of view. Both of the LIDARs are connected through USB-COMi-M, which enables high speed connection to the LIDARs, providing measurement rate at 75Hz with 0.25 degree of resolution. The top LIDAR is mounted horizontally to provide measurements of stable building features to allow accurate localization within

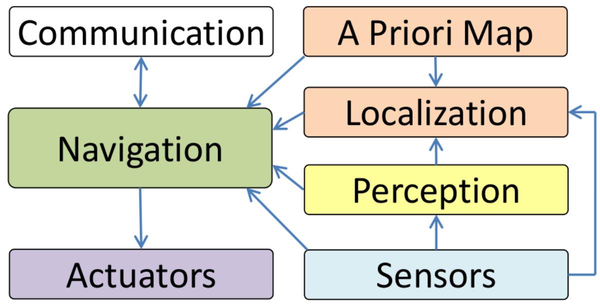


Fig. 4: Overview of the software modules implemented on the autonomous vehicle.

a known environment. The second LIDAR is mounted looking downward and is used to detect the curb lines along the road for navigation purposes.

Additionally, a 4-layer LIDAR, SICK LD-MRS400001 mounted at the waist level provides additional information about the environment. The data returns at the rate of up to 50 Hz with the total operating angle of 110 degree. Just on top of the LIDAR, a USB camera Logitech HD Pro Webcam C910 is placed. The camera is calibrated with the 4-layer LIDAR to provide pedestrian detections. The combination of LIDAR and camera is with the objective of extracting each sensor's different capabilities to achieve a robust detection system. This way, the excellent tracking performance of the LIDARs and the ability of vision to disambiguate different objects can be fully utilized.

Computing

There are 2 regular desktop PCs fitted with Intel i7 quad-core CPUs and interface card. All computers run Ubuntu 10.04 with Robot Operating System (ROS) installed. One of the computers is installed with RealTime Application Interface (RTAI), a real-time extensions for Linux Kernel to provide the low level control to the steering, brake and throttle of the golf car. Modular software architecture has been developed for ease in incorporating additional functionality without modifying the core system, as detailed in the next section.

3.2 Software Architecture

Figure 4 shows a high level view of the software components currently setup in our vehicle. In the following we briefly describe the navigation, localization and perception module.

Navigation module

Since the vehicle navigates on a known road network, all routes from any origin to any destination are generated a-priori as a set of waypoints. The choices of routes are made online depending on the request from the mobility on demand scheduler. The obstacles detected from the sensors are incorporated as a rolling cost map centered on the vehicle. The cost is propagated radially outward with an exponential function. At the low level, speed and steering control are separated. For the speed control, the vehicle considers the following input before planning for next action: the average cost function that is present within a defined area in front of itself and the curvature of the path [7]. The waypoint follower is implemented using a pure pursuit control [9].

Localization

Localization is very important for autonomous navigation. Most of the popular approaches for localization in autonomous navigation outdoors depend heavily on GPS based localization. In fact the DARPA challenge was based on GPS based waypoints as input. However, GPS is not very reliable in urban areas due to satellite blockage and multi-path propagation effect caused by tall buildings. An alternative approach to using GPS is to generate a high fidelity map of the area to be navigated using a high resolution range scanner. This approach is used by Google driverless car where the car, mounted with high fidelity Velodyne 3-D range sensor collect data of the road networks from various runs. Subsequently, when the vehicle travels it matches scans from its range sensors to the collected data and infers its location. However, collecting such a-priori information requires significant investment in terms of cost and manpower.

In line with our vision of lowering the cost of the autonomous vehicle, we use a single 2-D range sensor to detect roadside curb features. We use Adaptive Monte-Carlo Localization scheme to match the detected curb features to a road network map known a-priori. Figure 5 shows the basic components of this algorithm. This algorithm is tested through experiments in the campus environment as seen in Figure 6(a). A snapshot of the result is shown in Figure 6(b), where our vehicle drives from starting point S to goal point

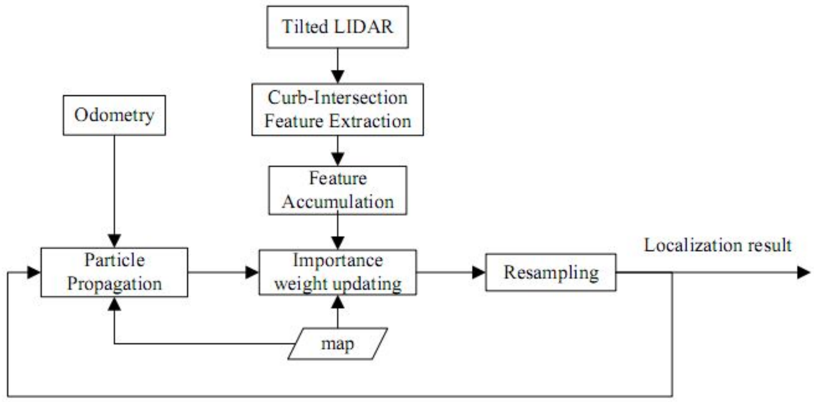
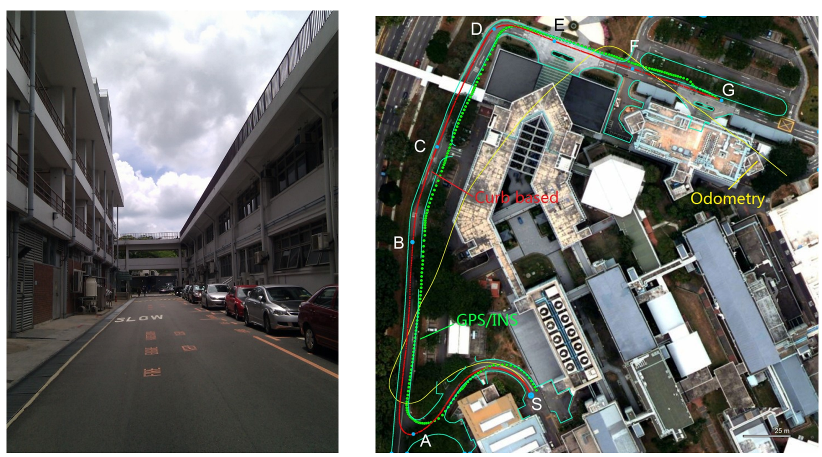


Fig. 5: Localization algorithm flowchart.



(a) A scene from the road segment traversed by the vehicle, typical road segments in Engineering Campus

(b) Comparing different localization approaches. Green shows GPS, Yellow shows IMU odometry and Red shows the estimate of the curb based localization

Fig. 6: Localization experiments

G. We can see that our curb based localization (shown in red) outperforms odometry (shown in yellow) and GPS based localization (shown in green).

The errors in location estimate are plotted in Table 1. It can be seen that position error of our algorithm is usually small, less than 0.6 meter; and the orientation estimation is quite accurate, less than 3 degrees to the ground truth. From Table 1, one can also observe that position errors at some critical points of intersections and turnings (like A, C, D, F) are much smaller than that of the straight road (like B). Fig. 7 shows "estimation variance" vs. "driving distance" in road longitudinal and lateral direction. It can be concluded that, while curb features on straight roads help to estimate the lateral position, the intersection and tightly curved curb features contribute very much to the longitudinal positioning. In our operations, we augmented the curb map with patches of laser map in areas where curb information was not available, e.g. pick up and drop off lobbies. Details are mentioned in [8].

Table 1: Localization error at several marked points

Marked Points	A	B	C	D	E	F	G
Position Error (m)	0.20	0.55	0.06	0.20	0.32	0.06	0.08
Orientation Error (deg)	< 3						

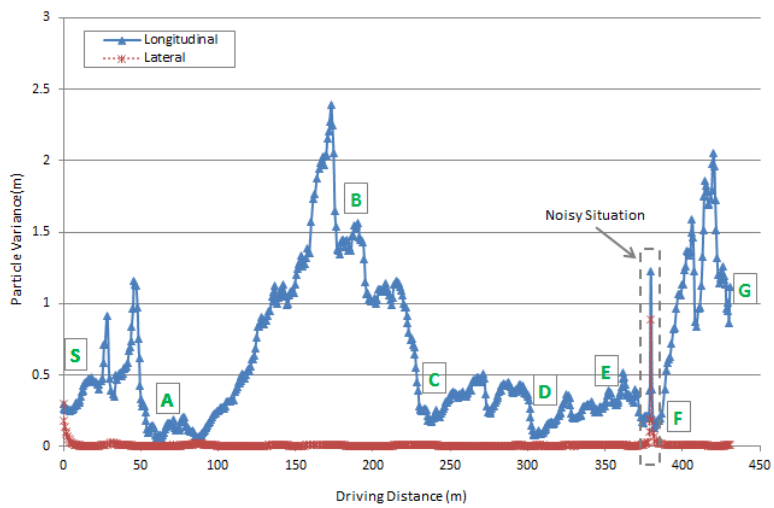


Fig. 7: Position estimation variance

Fig. 8: Vehicle navigating the dynamic environment.

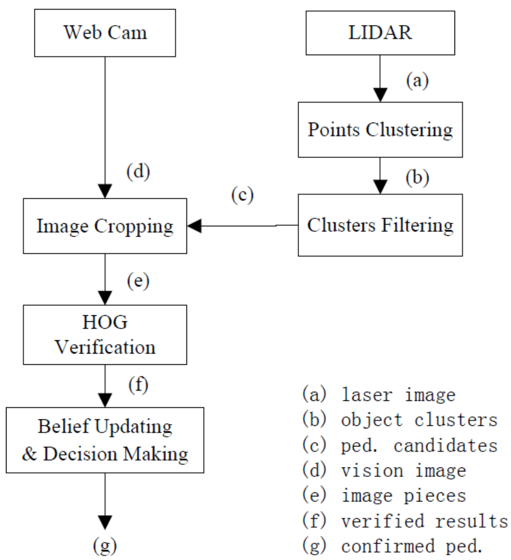
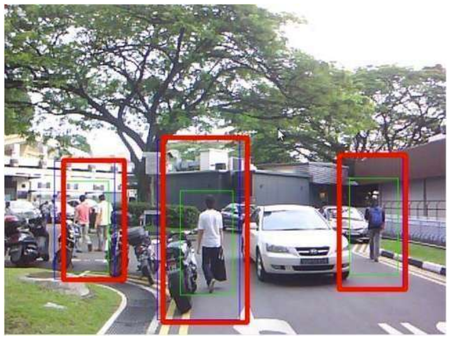
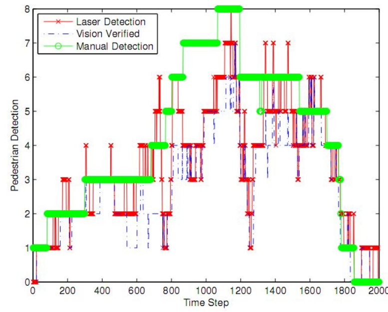


Fig. 9: Pedestrian detection algorithm flowchart.



(a) Vehicle interacting with incoming pedestrians



(b) Detection of the pedestrians

Fig. 10: Pedestrian detection in a dynamic, cluttered environment.

Perception

For autonomous driving, having a good perception of moving objects on the road is extremely important. Figure 8 shows a typical scenario the vehicle has to navigate in. The problem of detecting pedestrians, moving vehicles and static obstacles in cluttered, dynamic and time-varying lighting conditions is extremely complex. In addition, often the on-board sensing is occluded due to the presence of big trucks, buses or other environmental features. While vision systems can detect features more reliably, often ascertaining the distance of the features becomes difficult. On the other hand, while the laser range finders are quite accurate in detecting the distance to the obstacle, they are not well suited to disambiguate similar shaped obstacles like moving pedestrians or a static lamp-post.

We use a the combination of a single laser range finder and a simple web camera calibrated properly to detect pedestrians on the road. Figure 9 shows the basic components of the pedestrian detection algorithm. The laser clusters the sensor information based on the proximity, and the corresponding sub images are sent to a HoG SVM classifier to detect a person. A resulting snapshot of the vehicle while in operation is shown in Figure 10(a), where pedestrians are boxed. Fig.10(b) shows the number of objects tracked by LIDAR, pedestrians verified by webcam, and the ground truth number of pedestrians. In the test, most pedestrians got detected, whether as an individual, or as a group, making safe autonomous driving of our vehicle. Frequency of this detection system is up to 37Hz, limited by scan frequency from LIDAR. Range of effective detection is about 15 meters, limited by resolution of webcam. Details of the algorithm and implementation can be found at [7].

4 Demonstration

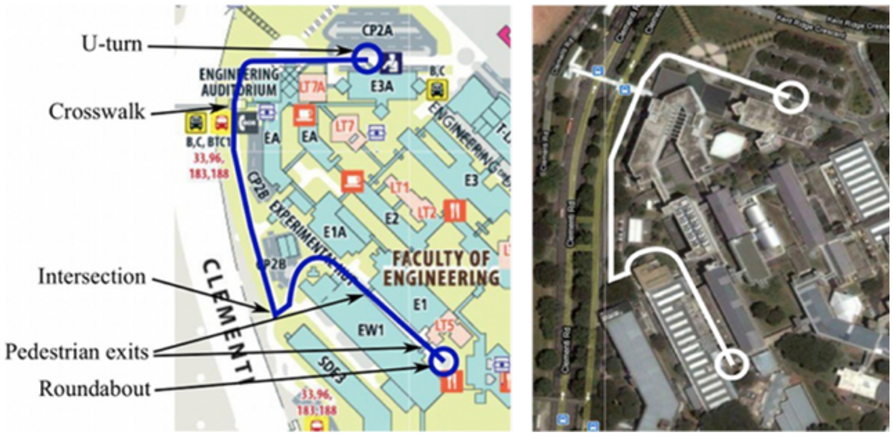


Fig. 11: Route of the autonomous vehicle.

The autonomous vehicle covered over 30km autonomously in the engineering section of National University of Singapore (NUS) campus, the route as shown in Figure 11. The selected route has representative segments of a typical road network, while being on campus the vehicle has to be more conservative in dealing with incoming student pedestrians and other vehicles. There are 4 pickup and drop off stations present in this section of the campus. The customer requests a pickup and drop-off location from either the mobile phone or the web interface shown in Figure 2. It is able to detect pedestrians and other vehicles and safely stop when the pedestrians or the vehicles are within a safety threshold along the vehicle's immediate path, preventing any collision. The videos of the operation are uploaded at (http://bit.ly/xiJDmZ).

5 Conclusion and future work

In this paper we present an autonomous vehicle implementing mobility on demand in a campus environment. Successful operation of the system has been demonstrated where the customers request mobility service and vehicle autonomously picks them up from their desired origin and drops them off at their requested destination.

We are currently incorporating mechanism for the vehicle to interact more meaningfully with other vehicles and pedestrians on the road, to predict their intentions and generate decisions accordingly. We are also including additional personal transport platforms to utilize multiple autonomous vehicles in our mobility on demand setup. We are also looking into incorporating infrastructure sensors to augment the vehicle's perception.

Acknowledgment

The research described in this project was funded in whole or in part by the Singapore National Research Foundation (NRF) through the Singapore-MIT Alliance for Research and Technology (SMART) Center for Future Urban Mobility (FM). The research was partnered with Defence Science Organisation (DSO). We acknowledge the valuable guidance provided by Prof. Emilio Frazzoli, Prof. Daniela Rus and Prof. David Hsu in developing the mobility on demand platform.

References

1. S. A. Shaheen and S. Guzman and H. Zhang, "Bikesharing in Europe, the Americas, and Asia", *Transportation Research Record: Journal of the Transportation Research Board*, pp. 159 - 167, 2010.
2. M. Pavone and S. L. Smith and E. Frazzoli and D. Rus, "Load Balancing for Mobility-on-Demand Systems", *Robotics: Science and Systems*, 2011.
3. W. J. Mitchell and C. E. Borroni-Bird and L. D. Burns, *Reinventing the Automobile: Personal Urban Mobility for the 21st Century.* Cambridge, MA: The MIT Press, 2010.
4. G. Berbeglia and J.-F. Cordeau and G. Laporte, "Dynamic pickup and delivery problems", *European Journal of Operational Research*, vol. 202, no. 1, pp. 8 - 15, 2010
5. M. Pavone and K. Treleaven and E. Frazzoli, "Fundamental Performance Limits and Efficient Polices for Transportation-On-Demand Systems", *IEEE Conf. on Decision and Control*, 2010.
6. J. Markoff, "Google Cars Drive Themselves, in Traffic", The New York Times, October 9, 2010.
7. Z. J. Chong and B. Qin and T. Bandyopadhyay and T. Wongpiromsarn and E. S. Rankin and M. H. Ang Jr. and E. Frazzoli and D. Rus and D. Hsu and K. H. Low, "Autonomous Personal Vehicle for the First- and Last-Mile Transportation Services", *IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, 2011.
8. B. Qin and Z. J. Chong and T. Bandyopadhyay and M. H. Ang Jr. and E. Frazzoli and D. Rus, "Curb-Intersection Feature Based Monte Carlo Localization on Urban Roads", *IEEE International Conference on Robotics and Automation*, 2012.
9. Y. Kuwata and J. Teo and G. Fiore and S. Karaman and E. Frazzoli and J. P. How, "Real-Time Motion Planning With Applications to Autonomous Urban Driving", http://dspace.mit.edu/openaccess-disseminate/1721.1/52527.