2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Roboat: An Autonomous Surface Vehicle for Urban Waterways

Wei Wang, Banti Gheneti, Luis A. Mateos, Fabio Duarte, Carlo Ratti and Daniela Rus

***Abstract*— Unmanned surface vehicles (USVs) are typically designed for open area marine applications. In this paper, we present a new autonomy system (Roboat) for urban waterways which requires robust localization, perception, planning, and control. A novel localization system, based on the extended Kalman filter (EKF), is proposed for USVs, which utilizes LiDAR, camera, and IMU to provide a decimeter-level precision in dynamic GPS-attenuated urban waterways. Area and shape filters are proposed to crop water reflections and street obstacles from a pointcloud. Euclidean clustering and multi-object contour tracking are then introduced to detect and track the static and moving objects reliably in urban waters. An efficient path planner is tailored to calculate optimal trajectories to avoid these static and dynamic obstacles. Lastly, a nonlinear model predictive control (NMPC) scheme with full state integration is formulated for the four-control-input robot to accurately track the trajectory from the planner in rough water. Extensive experiments show that the robot is able to autonomously navigate in both the indoor waterway and the cluttered outdoor waterway in the presence of static and dynamic obstacles, implying that Roboat could have a great impact on the future of transportation in many coastal and riverside cities.**

I. Introduction

Environmental monitoring, search and rescue, hydrology surveying, and national security operations have all led to a strong demand from commercial, scientific, and military communities for the development of innovative unmanned surface vehicles (USVs) [1]–[7]. USVs also have a promising role in the future of transportation for many coastal and riverside cities such as Amsterdam and Venice, where the existing infrastructure of roads and bridges is extremely overburdened. A fleet of eco-friendly self-driving boats could shift the transport of goods and people to waterways to get people out of their cars and reduce traffic in the city.

Our recently launched Roboat project seeks to develop a fleet of autonomous boats for transporting goods and people, and constructing dynamic floating infrastructure, like on-demand bridges and stages in the city of Amsterdam. Fig. 1 illustrates an artistic rendition where the Roboat is transporting people along a canal. In this article, we address urban USV autonomy for the basis of these applications.

Current USVs are usually deployed for coastal and marine applications [7]–[9]. As shown in Table I, these USVs have limited autonomy (especially in perception and planning) which cannot meet the autonomy requirements for applications in narrow and crowded urban environments.

This work was supported by grant from the Amsterdam Institute for Advanced Metropolitan Solutions (AMS) in Netherlands.

W. Wang, B. Gheneti, L. Mateos, F. Duarte, C. Ratti are with the SENSEable City Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. {wweiwang, bgheneti, lamateos, fduarte, ratti}@mit.edu

W. Wang, B. Gheneti, L. Mateos, and D. Rus are with the Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139 USA. {wweiwang, bgheneti, lamateos}@mit.edu, rus@csail.mit.edu

Fig. 1. Autonomous boat for transportation in urban waterways.

TABLE I
AUTONOMY ANALYSIS OF TYPICAL USVS ([7]–[9])

USVs	Localization	Perception	Planning	Control
ARTEMIS	GPS+IMU	N	N	fuzzy
ACES	GPS+IMU	N	N	fuzzy
SCOUT	GPS+IMU	RF	N	PID
AutoCat	GPS+IMU	RF	N	PID
Owls USVs	GPS+IMU	RF	N	cluster-space
S.C. 2000	GPS+IMU	radar	N	PID
Wave Glider	GPS+IMU+acoustic	N	N	Y, unclear
Springer	GPS+IMU+speedlog	N	N	LQG
Blackfish	GPS+IMU	N	N	backstepping
Roaz II	GPS+IMU	radar+camera	C&C	Y, unclear
Minion 2016 [10]	GPS+IMU	LiDAR+camera	Y, unclear	Y, unclear
NaviGator [11]	GPS+IMU	LiDAR+camera	RRT	MRAC

Note: S.C. 2000 stands for Seadoo Challenger 2000, "N" stands for none. "Perception" refers to object detection, and "Planning" here means planing with obstacle avoidance.

First and foremost, to safely navigate in urban waterways, a USV should localize itself with decimeter-level accuracy. Current USVs typically use GPS and IMU (fused by an extended Kalman filter (EKF) or unscented Kalman filter (UKF)) for meter-level precision [7]. These GPS-IMU-based approaches can be unstable in urban waterways, where GPS signals are often severely attenuated. Naeem *et al*. proposed a reliable multi-sensor navigation system which includes GPS, compass, speed log, and a depth sensor to account for sensor failure, but it cannot guarantee high accuracy [12]. To date, there is no feasible solution for accurate urban USV localization. Here, we propose a novel system which depends on LiDAR, camera, and IMU to provide decimeter-level precision in GPS-attenuated dynamic urban waterways.

Second, obstacle detection and tracking is indispensable for an urban USV because it frequently encounters dynamic

objects. Existing studies have mostly considered detection between cooperating vehicles [13]–[15], while other environmental obstacles are commonly ignored in these several studies [10], [16]–[19]. For example, Heidarsson *et al*. [16] employed a profiling sonar in shallow water, but the low resolution of the single beam limits the obstacle detection accuracy [16]. Moreover, vision-based systems have also been used, but are usually designed for imprecisely detecting obstacles at more than 100 meters [17], [19]. In addition, Thompson *et al*. utilized a support vector machine (SVM) classifier on LiDAR pointclouds [10]. However, due to the limitations of their occupancy grid, this approach is unable to classify moving obstacles. Hence, we propose to use LiDAR as the main sensor, and employ the Euclidean clustering and contour tracking algorithms to obtain reliable and accurate state estimation of static and moving obstacles.

Third, reactive obstacle avoidance is usually adopted for traditional USVs since there are few obstacle in open areas. Although some planning algorithms such as artificial potential field algorithms, and the A* search algorithm have been explored for USVs, local replanning for obstacle avoidance is commonly ignored [7], [20]. Therefore, we adapt an efficient path planner for our urban USV to calculate an obstacle-free path, considering the canal boundaries, USV kinematics, and the surrounding stationary and dynamical obstacles.

Fourth, a number of tracking control methods such as the sliding mode method [21], integrator back-stepping method [22], [23] and adaptive control [24] have been proposed for USVs. However, the tracking accuracy (typically larger than 1 m) of these controllers was insufficient for urban USVs navigating in restricted waterways. Hence, we develop an effective dynamic model for our USV and then implement an efficient nonlinear model predictive controller (NMPC) to achieve accurate tracking ability.

In our previous work [25], we designed a quarter-scale robotic boat that was able to track a trajectory based on an NMPC strategy. In this paper, we further develop a complete autonomy system for the Roboat navigating in urban waterways. To the best of the authors' knowledge, this is the first time that an autonomous boat navigates in urban waterways, interacting with static and dynamic obstacles, and current and wave disturbances. In summary, this paper contributes in the following directions:

- a complete autonomy system adapted to the Roboat architecture. Our system navigates, detects and avoids static and dynamic obstacles in a dynamic urban environment;
- robust high-precision state estimation that uses LIDAR, IMU and camera systems in cluttered urban waters;
- robust high-precision LiDAR-based dynamic obstacle detection and tracking on water;
- revised NMPC-EKF control scheme with full state integration for stable and accurate tracking;
- extensive experiments to validate the developed autonomy system in challenging turbulent waterways.

This paper is structured as follows. Section II overviews the Roboat prototype. Section III describes the developed autonomy system for urban USVs, with an emphasis on robust localization, perception, planning and control. Autonomous obstacle avoidance experiments are presented in Section IV. Section V concludes this paper.

II. Hardware Overview

The quarter-scale Roboat platform is shown in Fig. 2, which is an updated version of the USV in [25]. In particular,

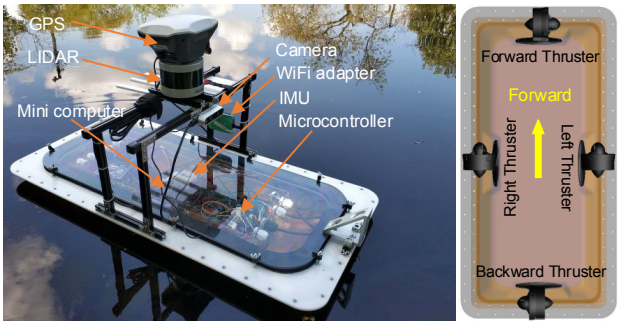

Fig. 2. The developed Roboat prototype. (a) The prototype picture (side view); (b) thruster configuration (bottom view).

the boat has four thrusters with a cross-shaped actuator configuration (as shown in Fig. 2(b)) to achieve maneuverable and efficient propulsion. It has an inverted trapezoidal hull to facilitate boat self-assembly. Instead of 3D printing, we mould the boat hull in this version which produces a stronger and smoother result. We also widen the plastic O-ring between the cover and hull for better waterproofing.

To guarantee the real-time performance, a more powerful Intel NUC (Intel Core i7-8650U) with 32 GB of memory is adopted as the main controller running the Robotic Operating System (ROS). Moreover, a 32-bit auxiliary processor, STM32F103, is used for converting the forces from the main controller to actuator signals. The Roboat also has a diverse set of onboard sensors. A 3D LiDAR (Velodyne, Puck VLP-16) is installed on the top center of the USV for localization and obstacle detection. An IMU (LORD Microstain, 3DM-GX5-25) is positioned parallel to the robot body's principal axes to monitor the robot attitude, linear acceleration and angular velocity. A stereo depth camera (Intel Realsense D435) is installed below the LiDAR to provide visual odometry for EKF localization. GPS is integrated but not used in this paper. The Roboat's specifications are listed in Table II.

TABLE II
Technical Specifications of the Prototype

Items	**Characteristics**
Dimension (L×W×H)	0.90 m× 0.45 m × 0.15 m
Total mass	∼ 15.0 kg
Drive mode	Four T100 thrusters
Onboard sensors	3D LIDAR, IMU, Camera, GPS
Power supply	11.1 V rechargeable Li-Po battery
Operation time	∼ 3.0 hours

III. Autonomy System for Urban Waterways

In this section, we describe an autonomy framework for USVs in urban waterways, as shown in Fig. 3. In particular,

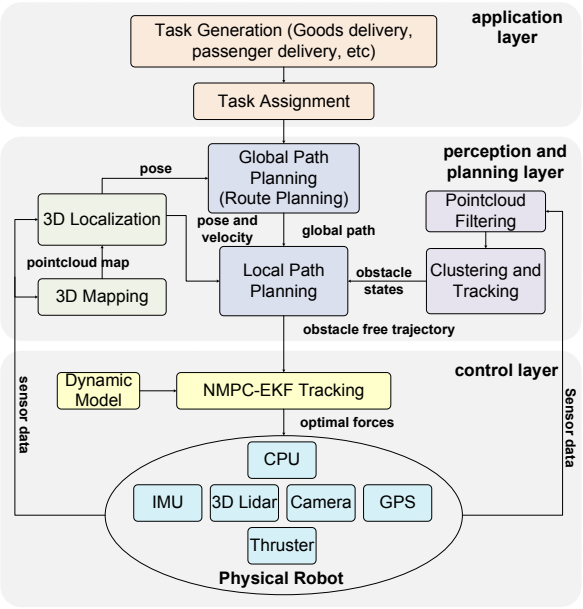

Fig. 3. Autonomy framework of the Roboat. From top to bottom, it contains three layers: Application, Perception and Planning, and Control.

when a task such as passenger delivery is requested, the coordinator will assign this task to one or more Roboats. The Roboat localizes itself, detects surrounding obstacles, plans a global path to the goal, finds a locally optimal obstacle-free path and finally tracks that path to safely finish the delivery, all in an urban waterway. We focus on the perception and planning, and control layers of the autonomy framework.

A. Accurate Localization

As mentioned previously, localization for USVs in urban waterways is more challenging than for traditional USVs on open water and driverless cars on the road. To solve this problem, we propose to collectively use LiDAR, IMU, and camera measurements in an extended Kalman filter (EKF) shown in Fig. 4 to achieve very stable and precise localization in cluttered and dynamic water environments. We select LiDAR as the primary sensor for localization

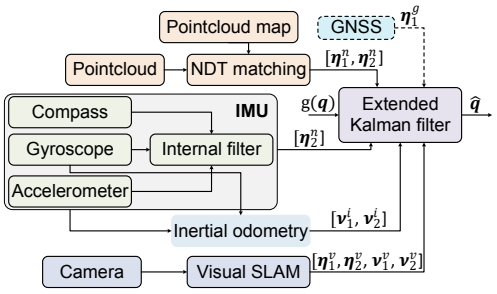

Fig. 4. The proposed localization algorithms for urban waterways. The solid lines stand for the currently employed sensors and algorithms and the dashed line indicate sensors to be integrated in the near future.

because its accuracy has been shown on driverless cars. Moreover, we choose the normal distributions transformation (NDT) matching algorithm [26] because it handles noisy measurements well due to its probabilistic nature. More specifically, we represent the map as a 3D grid and assign a probability distribution to each gridpoint. This way, the NDT matching algorithm fits the detected points to distributions on the map. Note that the 3D LiDAR map is generated offline, also by the NDT algorithm. In the localization experiments, we also use Visual SLAM on an RGB-D image for additional measurements in the EKF due to unstable IMU odometry.

The general motion of a surface vehicle in 6 DOF can be described by following vectors: $\boldsymbol{\eta}_1 = [x\ y\ z]^T$, $\boldsymbol{\eta}_2 = [\phi\ \theta\ \psi]^T$, $\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T, \boldsymbol{\eta}_2^T]^T$, $\boldsymbol{\nu}_1 = [u\ v\ w]^T$, $\boldsymbol{\nu}_2 = [p\ q\ r]^T$, $\boldsymbol{\nu} = [\boldsymbol{\nu}_1^T, \boldsymbol{\nu}_2^T]^T$, and $\boldsymbol{a} = [a_x\ a_y\ a_z]^T$. Here $\boldsymbol{\eta}$ denotes the pose (position and orientation) vector with the coordinates in the earth-fixed frame, $\boldsymbol{\nu}$ denotes the velocity (linear and angular velocity) vector with the coordinates in the body-fixed vector, and $\boldsymbol{a}$ denotes the linear acceleration of the robot.

We designed an EKF observer to efficiently estimate the nonlinear dynamic process $\boldsymbol{q} = [\boldsymbol{\eta}^T, \boldsymbol{\nu}^T, \boldsymbol{a}^T]^T$ which is assumed to be governed by a nonlinear motion equation

$$\boldsymbol{q}_k = g(\boldsymbol{q}_{k-1}) + \boldsymbol{w}_{k-1} \tag{1}$$

with a measurement $\boldsymbol{s}$ that is

$$\boldsymbol{s}_k = h(\boldsymbol{q}_k) + \boldsymbol{v}_k \tag{2}$$

where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ represent the process and measurement noise at time step k: $\boldsymbol{w}_k \sim N(0, \boldsymbol{Q}_k)$ and $\boldsymbol{v}_k \sim N(0, \boldsymbol{R}_k)$ respectively. $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ stand for their covariance matrices. The motion equation $g(\boldsymbol{q}_{k-1})$ takes the following form

$$g(\boldsymbol{q}_{k-1}) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & h\boldsymbol{T}\boldsymbol{I}_3 & \boldsymbol{0}_3 & 0.5h^2\boldsymbol{T}\boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & h\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & h\boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \boldsymbol{a} \end{bmatrix} \tag{3}$$

where h is the sampling interval, $\boldsymbol{T}$ is the transformation matrix converting a state vector from body to inertial frame. The robot state $\boldsymbol{q}$ is then estimated by two sets of equations: time and measurement update equations [27].

We first did ten experiments (each lasting 90 s) in a 12.5 m$\times$ 6.5 m swimming pool randomly moving the Roboat, using the beacon-based localization system (2 cm accuracy) in [25] as the ground truth. The average position error of the NDT localization is 0.1476 m and that of the EKF estimation is 0.1263 m with a 0.202 m standard deviation. The data is not shown here due to space limitations. Moreover, we did five experiments (each lasting 145 s) in a 40 m $\times$ 12 m area in the Charles river, where large currents and waves exist. Fig. 5 and 6 compare the position and velocity estimates between the NDT and the final EKF algorithms. Even though both the NDT and EKF trajectories are smooth, the EKF velocity is significantly more stable than that of the NDT. The average standard deviation of the outdoor position is 0.38 m. Despite not having a ground truth for the outdoor tests, we can infer that the average error of the outdoor

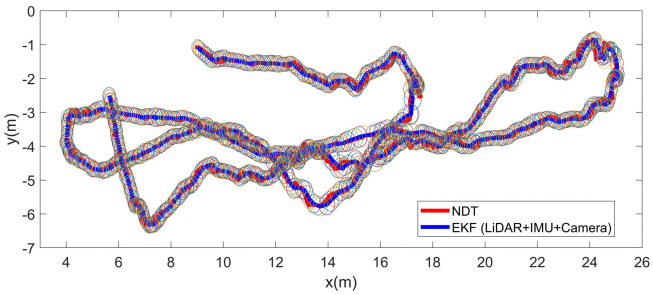

Fig. 5. Position comparison between the NDT and the final EKF. The colored ellipses stand for the covariance of each position.

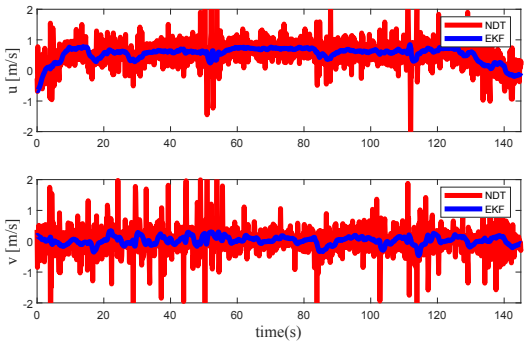

Fig. 6. Velocity comparison between the NDT and the final EKF.

localization is around 0.22 m, by comparing with the indoor data. We also tested the stability of the EKF estimation in these cluttered water environments by continually running it for nearly 2 hours. The estimation only diverged twice when we rotated the robot sharply at extremely high speeds.

B. Obstacle Detection and Tracking

Navigating in urban waterways necessitates that both the object detection and tracking are precise, which is typically not the case for USVs in open waters, as noted in the introduction. On the other hand, driverless cars consider objects like pedestrians and bicycles that may enter their roadways from the outside. For USVs, these obstacles are unlikely to enter the urban waterway. Hence, we filter out canal walls and street obstacles prior to detection and tracking.

1) Filtering: Given an input LiDAR pointcloud P_{in} centered on the boat-fixed frame, we use cropping filters to constrain the detection space within the waterway. This makes the obstacles we detect more relevant and reduces the computational load on the system. We first crop P_{in} to the square area around the boat, $2\Delta \times 2\Delta$ (where Δ is the half edge length of the square) in the $x-y$ plane, and region above the water, with a minimum threshold z_{th} in z-axis, to obtain P_{crop}. A z-axis threshold is preferable to the ground filters used by autonomous cars, because they are not reliable on wet surfaces. This leaves us with points from street obstacles, canal walls, and canal obstacles. Afterwards, we transform P_{crop} from a body-fixed frame to an earth-fixed frame with the transformation matrix $\boldsymbol{T}$, yielding $\tilde{P}_{\text{crop}}$, which is filtered based on the waterway edges to obtain candidate canal obstacle points $\tilde{P}_{\text{out}}$ lying within the boundaries.

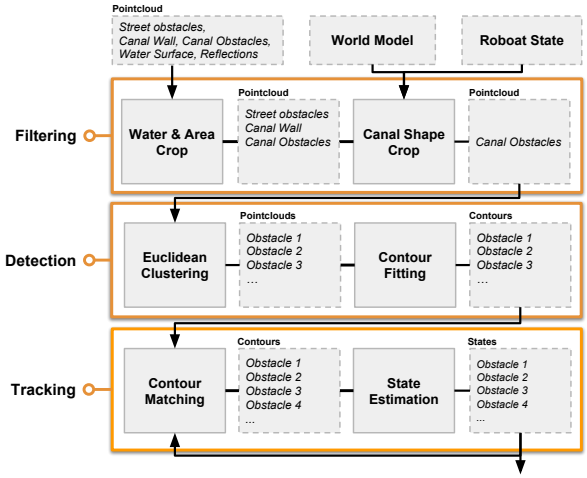

Fig. 7. Perception pipeline: modules in solid lines, data in dashed lines.

2) Detection: To identify obstacles present in $\tilde{P}_{\text{out}}$ we employ the Euclidean clustering implementation in Point Cloud Lib (PCL), a computationally efficient density based clustering algorithm [28], and tune the clustering radius d_{th}, minimum cluster size c_{min}, and inter-cluster merging distance c_{th}. We set d_{th} relatively high because of the low density of VLP16 LiDAR points. We use a relatively large c_{th} to cluster non-convex obstacles such as kayakers as one. We discriminatively set c_{min} to enable clustering small obstacles, while filtering out noise from waves. Considering these factors enables our USV to cluster obstacles of varying sizes, even in choppy conditions. Lastly, the contours of the clusters are represented as convex hulls in the $x-y$ plane [29].

3) Tracking: Our tracker, adapted from that proposed here [29], assigns a unique ID to each contour and Kalman filters the pose associated with this ID. An obstacle in one time step is matched to one from the previous time step if the centroid moves less than ι_{th} and the detected area changes by no more than a factor of κ_{th}. An unobserved obstacle is remembered for up to T_{th} at its last reported location. This allows for robust tracking, even in the most difficult and noisy detection environments, and enables safe planning when passing close to obstacles in urban waterways.

TABLE III
PARAMETER VALUES USED IN OBSTACLE DETECTION AND TRACKING

Condition	Parameters							
	Δ	z_{th}	d_{th}	c_{th}	c_{min}	ι_{th}	κ_{th}	T_{th}
Indoor	5.0	-0.4	0.5	1.5	20	1.0	3.0	5
Outdoor	10.0	-0.4	0.5	1.5	40	1.5	3.0	10

Note: Δ, z_{th}, d_{th}, c_{th} and c_{min} in meters; T_{th} in seconds; ι_{th}, κ_{th} and c_{min} no units.

We carried out experiments along the path in Fig. 8(a) to determine the accuracy of tracked contours at different x distances relative to an obstacle. The accuracy in Fig. 8(b) is measured in terms of the intersection over union (IoU) or Jaccard Index, using the detected contours and the ground truth contour. From Fig. 8(b), we can see that the robot consistently detects the obstacle (IoU > 0), except for the blind zone of the LiDAR when the robot is too

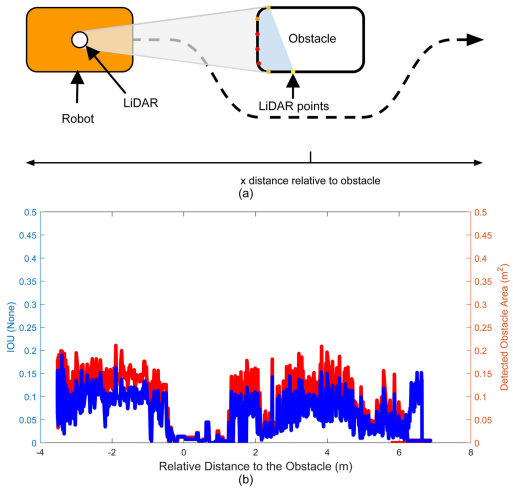

Fig. 8. IoUs changes as the robot moves along the obstacle-free path. (a) Sketch map; (b) IoU.

close to the obstacle. The noisy IoUs indicate that water disturbances sway the robot, reducing sensor visibility a lot across different object surfaces. To alleviate these effects, we will aggregate obstacle pointclouds from different vantage points, and employ a library of candidate shapes as priors for a more consistent IoU. Overall, our solution for obstacle perception is robust because it consistently detects and tracks obstacles in real time. This opens the way for adapting USVs from open water to urban USV applications.

C. Path Planning with Obstacle Avoidance

In this paper, we adapt an efficient open-source planner (open planner) in [30] to surface vehicles in narrow and crowded waterways, because this planner 1) is defined for structured environments, which is ideal for urban waterways such as the Amsterdam canal system; 2) is experimentally verified for its robustness in the presence of large water disturbances; 3) efficiently handles multiple static and dynamic obstacles to calculate the best obstacle-free path. Note that local regulations including COLREGs and additional behavior rules in the canals of Amsterdam are not incorporated in the current planner since developing the challenging autonomy system for the new urban waterway context is the main focus of this article. These regulations will be fit into our current autonomy platform later on.

The architecture of the adapted planner is illustrated in Fig. 9. It mainly includes a global planner and a local planner. The global planner generates a global reference path from a vector (waterway network) map. The local planner generates a set of candidate trajectories, evaluates them according to the obstacle states and chooses the candidate trajectory with the lowest cost, which is always an obstacle-free trajectory. Finally, the behavior generator acting as an orchestrator, switches to a proper predefined behavior state (we define four states for the Roboat, *Forward*, *Swerve*, *Wait*, and *Stop*).

The global planner takes a vector map, a start position and a goal as its input and finds the lowest cost path (shortest path in this paper) using dynamic programming. Urban waterways

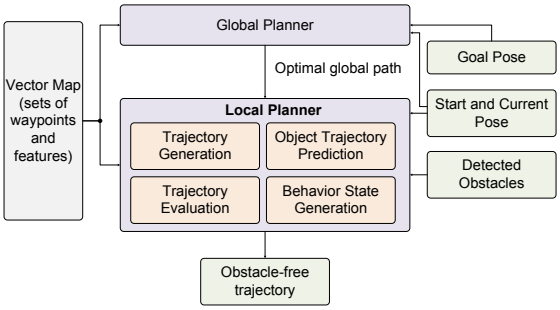

Fig. 9. The planner structure for the urban waterway surface vehicle.

are simpler than roads because no pedestrian crossings and traffic lights are present. Hence, we developed a simplified vector map for our robot which only includes the centerline waypoints and edges of the waterway.

The local planner generates an obstacle-free trajectory for the USV by mainly considering the obstacle states from the perception. Fig. 10 illustrates the basic idea of the local planner. First, the trajectory generator produces a set of

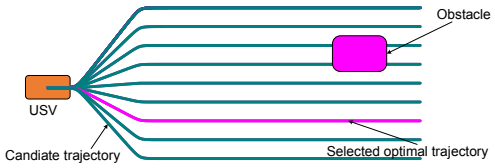

Fig. 10. Working principle of the local planner for surface vehicles.

local candidate trajectories according to the current position and the planning distance. Afterwards, a non-linear iterative conjugate gradient method is used to smooth each trajectory. Note that many parameters, including the motion constraints, smoothing algorithm, planning distance, avoidance distance (from which distance the robot starts avoiding), and the density of candidate trajectories, are adapted for waterway applications.

Further, the trajectory evaluator uses an additive cost function to evaluate each candidate trajectory [30],

$$w_i = g_c(i) + g_t(i) + g_s(i) + g_d(i) \tag{4}$$

w_i stands for the cost function of the i^{th} candidate trajectory, $g_c(i)$ stands for the cost that the i^{th} candidate trajectory deviates from the central one, $g_t(i)$ stands for the cost to go for the i^{th} trajectory, $g_s(i)$ stands for the collision cost between the i^{th} trajectory and the static obstacles, $g_d(i)$ stands for the collision cost between the i^{th} trajectory and the dynamic obstacles. The candidate with the smallest cost is selected as the best obstacle-free trajectory. We are integrating the disturbance (such as water currents) cost into our planner to further improve the planning performance.

D. NMPC Trajectory Tracking

The NMPC controller is responsible for tracking the obstacle-free trajectory from the planner. From now on, we constrain our controller to 2D because our robot operates on the water surface.

1) Boat Dynamics: The dynamics of our robot is described by the following nonlinear differential equation [25]

$$\dot{\boldsymbol{\eta}} = \boldsymbol{T}(\eta)\boldsymbol{\nu} \tag{5}$$

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{u} - \boldsymbol{M}^{-1}(\boldsymbol{C}(\boldsymbol{\nu}) + \boldsymbol{D}(\boldsymbol{\nu}))\boldsymbol{\nu} \tag{6}$$

where $\boldsymbol{\eta} = [x\ y\ \psi]^T$ is the position and orientation of the robot in the inertial frame, $\boldsymbol{\nu} = [u\ v\ r]^T$ denotes the vehicle velocity, which contains the vehicle surge velocity (u), sway velocity (v), and yaw rate (r) in the body fixed frame, $\boldsymbol{T}(\eta)$ is the transformation matrix converting a state vector from body frame to inertial frame, $\boldsymbol{M}$ is the positive-definite symmetric added mass and inertia matrix, $\boldsymbol{C}(\boldsymbol{\nu}) \in \mathbb{R}^{3\times3}$ is the skew-symmetric vehicle matrix of Coriolis and centripetal terms, $\boldsymbol{D}(\boldsymbol{\nu})$ is the positive-semi-definite drag matrix-valued function, $\boldsymbol{B}$ is the control matrix describing the thruster configuration and $\boldsymbol{u} = [f_1\ f_2\ f_3\ f_4]^T$ is the control vector. The unknown hydrodynamic parameters $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ were identified by a nonlinear least squares method based on the trust-region-reflective algorithm. More details of the dynamic model can be found in our previous work [25]. The complete dynamic model of the surface vehicle is reformulated by combining Eqs. (5) and (6), given by

$$\dot{\boldsymbol{q}} = f(\boldsymbol{q}, \boldsymbol{u}) \tag{7}$$

where $\boldsymbol{q} = [x\ y\ \psi\ u\ v\ r]^T$ is the state vector of the robot.

2) NMPC-EKF Control: To further improve the performance in cluttered water conditions, we revise our NMPC controller in the following ways: 1) Integrating the EKF observer into the NMPC to obtain the NMPC-EKF strategy; 2) Controlling the full states $\boldsymbol{q}$ (only three states are effectively controlled in [25]); 3) Increasing the rate of the controller from 5 Hz to 10 Hz. Fig. 11 shows the framework of the NMPC-EKF controller.

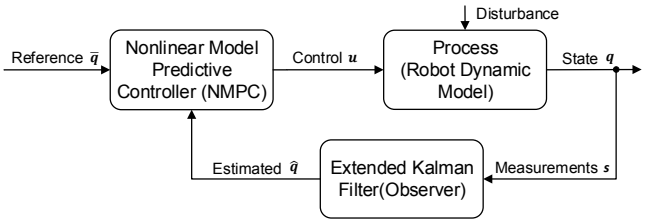

Fig. 11. The NMPC-EKF framework for the trajectory tracking control.

The NMPC determines the control of our robot by iteratively solving the following optimal control problem with a finite time horizon H,

$$\min_{\boldsymbol{u}(\sigma)} = \int_t^{t+H} F(\boldsymbol{q}(\sigma), \boldsymbol{u}(\sigma))d\sigma + E(\boldsymbol{q}(t+H)) \tag{8}$$

subject to

$$\dot{\boldsymbol{q}}(\sigma) = f(\boldsymbol{q}(\sigma), \boldsymbol{u}(\sigma)), \boldsymbol{q}(0) = \boldsymbol{q}_0, \tag{9}$$

$$\boldsymbol{q}(\sigma) \in Q, \boldsymbol{u}(\sigma) \in U, \sigma \in [t, t+H]. \tag{10}$$

where F is the objective cost function and E is the terminal cost. More details of F and E can be found in [25]. In particular, the parameters and constraints of the NMPC-EKF controller are listed as follows: $f_i^{\min} \le u_i \le f_i^{\max}$, where $f_i^{\min} = -6$ N, $f_i^{\max} = 6$ N, and $i = 1,\ 2,\ 3,\ 4$, $H = 4$ s; the positive definite weight matrices $\bar{Q} = \text{diag}\{20, 20, 5, 5, 5, 5\}$, and $\bar{R} = \text{diag}\{1, 1, 1, 1\}$, the terminal penalty matrix $\bar{Q}_n = \text{diag}\{20, 20, 5, 5, 5, 5\}$.

We did five experiments in the swimming pool to compare the performance between the NMPC-EKF in this paper and the NMPC in [25], as shown in Fig. 12 and 13. The

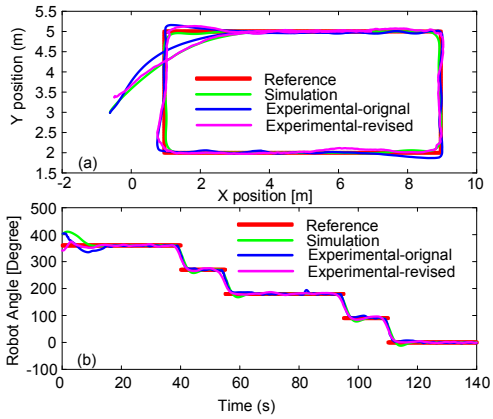

Fig. 12. Rectangle tracking using NMPC. (a) Position and (b) orientation.

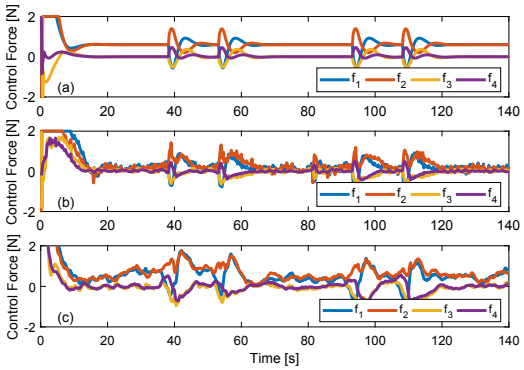

Fig. 13. Control forces from NMPC in the experiments. (a) Simulation, (b) experimental forces in [25], and (c) experimental forces with NMPC-EKF.

average absolute errors for the position and angle decreased from 0.095 m and 6.3° to 0.089 m and 2.3° respectively. Considering the current localization accuracy is around 10 cm (2 cm in [25]), the control performance increased a lot. Moreover, it is obvious that the robot orientation is more stable and accurate. Furthermore, the control forces generated by the NMPC-EKF are smoother than the previous ones. This could be attributed to the integration of velocity control that improves the anti-disturbance performance of the system. As shown in Fig. 14, the Roboat is able to track the linear and angular velocities by the NMPC-EKF strategy.

IV. Experiments and Results

This section presents extensive indoor and outdoor experiments to validate the effectiveness and robustness of the developed autonomy system. Indoor experiments were conducted in a 12.5 m × 6.5 m swimming pool (Fig. 15(a))

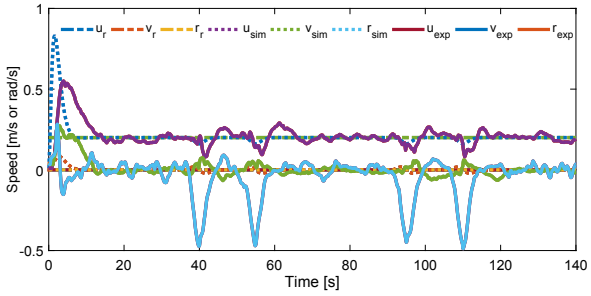

Fig. 14. Speed tracking of the improved NMPC method. u_r, v_r, and r_r are the reference, u_{sim}, v_{sim}, and r_{sim} are the simulated velocities, u_{exp}, v_{exp}, and r_{exp} are the experimental velocities.

to verify the functionality of the autonomy system in a controlled environment. Outdoor experiments were conducted

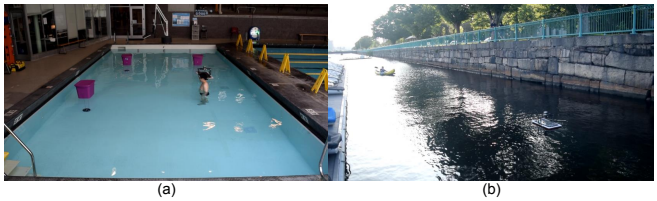

Fig. 15. Experimental scenes. (a) Swimming pool, and (b) Charles River.

in a 40 m $\times$ 12 m waterway area in Charles river (Fig. 15(b)) to verify the robustness of the developed autonomy system while interacting with dynamic obstacles in a natural environment with large currents and wave disturbances.

Fig. 16(a) illustrates the setup for the indoor experiments. Three static obstacles (0.5 m $\times$ 0.3 m) are distributed along the waypoints. During the experiments, one dynamic obstacle (a person) starts to move towards the robot or across its path at a nearly constant speed, near the end of the path. We can see from Fig. 16(b) and (c) that the robot autonomously navigates itself along the man-made urban waterways at the desired speed of 0.3 m/s for 120 s in the presence of static and dynamic obstacles. Note that the pool steps were also identified as an obstacle in the experiments.

We now dissect Fig. 16(c) to explain the autonomy process. At time A, the robot detects the first obstacle on the way and swerves to the optimal obstacle-free trajectory at B. At time C, the robot similarly moves to avoid the static obstacle, staying on course at time D for the next obstacle. At time E the robot spots the moving obstacle. It initially attempts to avoid it to the right but switches to the left at time F when the obstacle's motion increases the risk of its prior plan. Once all obstacles are behind, it returns to the waypoint path. We repeated the experiments 5 times for both situations in Fig. 16(b) and (c). The robot successfully avoided all static and dynamic obstacles in the experiments, validating the integrated autonomy system, as well as the developed localization, perception, planning, and control algorithms.

Fig. 17 shows the robot successfully navigating in very challenging urban waterways at the desired speed of 0.5 m/s for about 80 s. Note that large currents and waves,

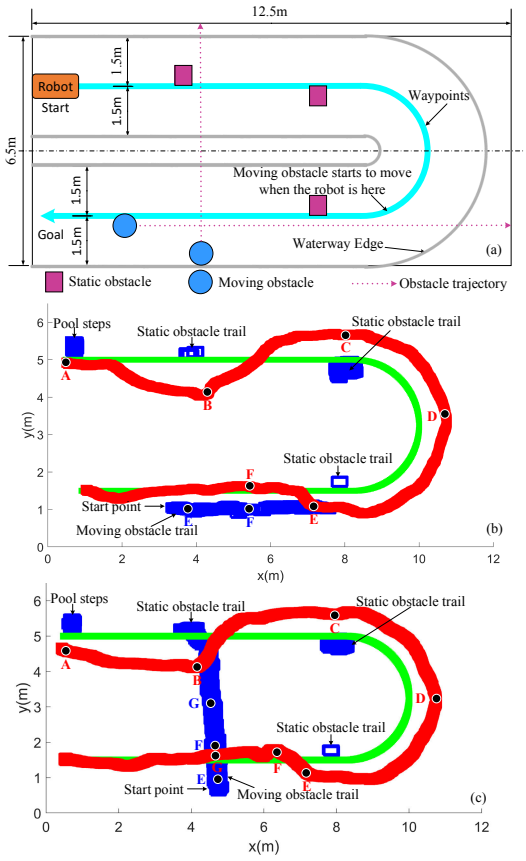

Fig. 16. Indoor autonomy experiments. Red squares are the robot position, green points are the waypoints, and blue squares are the obstacles.

as well as static and dynamic obstacles, are all interacting with the robot in the experiments. We can see that even the static obstacle drifted three meters due to large disturbances. Moreover, the dynamic obstacle moves towards the robot with large lateral oscillations. We carried out five experiments with varying water currents and disturbances in the Charles river over two weeks. Four experiments went exceptionally well, resulting in smooth collision free trips. One resulted in a light collision when, belatedly, the dynamic obstacle aggressively rotated towards the boat due to large disturbances. These extensive experiments demonstrate the effectiveness of our autonomy system for urban waterways.

V. Conclusion and Future Work

In this paper, we have developed an autonomy system for urban waterways which includes robust localization, perception, planning, and control. USVs navigating in urban waterways have some unique challenges compared to traditional USVs in open areas. In this article, we have proposed solutions to these unique challenges in urban USV autonomy, enabling the Roboat to navigate, detect and avoid static and dynamic obstacles in cluttered urban waterways, all in an autonomous manner. At the same time, the current system also has some limitations. For example, the current system cannot handle complex obstacle dynamics. Therefore, our work will be extended in the following directions in the near

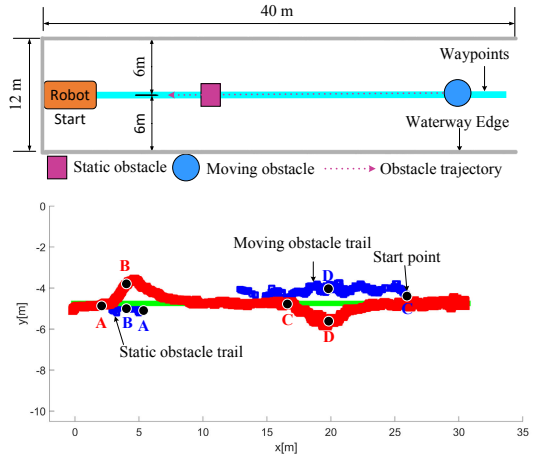

Fig. 17. Outdoor autonomy experiments. Red squares are the robot position, green points are the waypoints, and blue squares are the obstacles.

future. First, we will explore deep reinforcement learning based motion planning algorithms for boat autonomy, to further help the robot handle more complicated scenarios on the water. Second, we will explore active and multimodal perception to improve the Roboat's understanding of its environment for more robust planning performance. Third, we will develop sensor-based estimates for water currents and waves and feed them to the controller. Fourth, we will develop algorithms for multi-robot formation control and self-assembly on the water, enabling the construction of on-demand large-scale infrastructure.

ACKNOWLEDGMENT

The authors would like to thank B. Cai, Y. Zhou, X. Li, W. Tu for help in carrying out the experiments, H. Darweesh and Autoware for software consultation and advice, R. Kelly for writing revision, and P. Leoni for contributions in building the boat hull.

REFERENCES

[1] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu, "Experiments with underwater robot localization and tracking," in *Proc. 2007 IEEE Int. Conf. Robot. Autom.*, 2007, pp. 4556–4561.

[2] M. Doniec, I. Vasilescu, C. Detweiler, and D. Rus, "Complete SE3 underwater robot control with arbitrary thruster configurations," in *Proc. 2010 IEEE Int. Conf. Robot. Autom.*, 2010, pp. 5295–5301.

[3] J. Curcio, J. Leonard, and A. Patrikalakis, "SCOUT - a low cost autonomous surface platform for research in cooperative autonomy," in *Proc. MTS/IEEE Oceans*, 2005, pp. 725–729.

[4] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 131–149, 2014.

[5] A. Dhariwal and G. S. Sukhatme, "Experiments in robotic boat localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, 2007, pp. 1702–1708.

[6] M. N. Azzeria, F. A. Adnanb, and M. Z. M. Zaina, "Review of course keeping control system for unmanned surface vehicle," *Jurnal Teknologi (Sciences & Engineering)*, vol. 5, no. 74, pp. 11–20, 2015.

[7] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: An overview of developments and challenges," *Annu Rev Control*, vol. 41, pp. 71 – 93, 2016.

[8] B. J. Guerreiro, C. Silvestre, R. Cunha, and A. Pascoal, "Trajectory tracking nonlinear model predictive control for autonomous surface craft," *IEEE Trans Control Syst Technol*, vol. 22, no. 6, pp. 2160–2175, 2014.

[9] J. E. Manley, "Unmanned surface vehicles, 15 years of development," in *Proc. MTS/IEEE Oceans*, Sept 2008, pp. 1–4.

[10] D. J. Thompson, "Maritime object detection, tracking, and classification using lidar and Vision-Based sensor fusion," Master's thesis, 2017.

[11] D. Frank, A. Gray, K. Allen, T. Bianchi, K. Cohen, D. Dugger, J. Easterling, M. Griessler, S. Hyman, M. Langford *et al.*, "University of Florida: Team NaviGator AMS," in *RobotX Forum*, 2016.

[12] W. Naeem, R. Sutton, and T. Xu, "An integrated multi-sensor data fusion algorithm and autopilot implementation in an uninhabited surface craft," *OCEAN ENG*, vol. 39, pp. 43–52, 2012.

[13] M. R. Benjamin, J. A. Curcio, J. J. Leonard, and P. M. Newman, "Navigation of unmanned marine vehicles in accordance with the rules of the road," in *Proc. 2006 IEEE Int. Conf. Robot. Autom.*, 2006, pp. 3581–3587.

[14] J. Yu, C. Wang, and G. Xie, "Coordination of multiple robotic fish with applications to underwater robot competition," *IEEE Trans. Ind. Electron*, vol. 63, no. 2, pp. 1280–1288, 2016.

[15] C. Tam and R. Bucknall, "Cooperative path planning algorithm for marine surface vessels," *OCEAN ENG*, vol. 57, pp. 25 – 33, 2013.

[16] H. K. Heidarsson and G. S. Sukhatme, "Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar," in *Proc. 2011 IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 731–736.

[17] D. Frost and J.-R. Tapamo, "Detection and tracking of moving objects in a maritime environment using level set with shape priors," *EURASIP J IMAGE VIDE*, vol. 2013, no. 1, p. 42, Jul. 2013.

[18] H. Wang, X. Mou, W. Mou, S. Yuan, S. Ulun, S. Yang, and B. Shin, "Vision based long range object detection and tracking for unmanned surface vehicle," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2015, pp. 101–105.

[19] M. T. Wolf, C. Assad, Y. Kuwata, A. Howard, H. Aghazarian, D. Zhu, T. Lu, A. Trebi-Ollennu, and T. Huntsberger, "360-degree visual detection and target tracking on an autonomous surface vehicle," *J. Field Robotics*, vol. 27, no. 6, pp. 819–833, Nov. 2010.

[20] S. Campbell, W. Naeem, and G. Irwin, "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres," *Annu Rev Control*, vol. 36, no. 2, pp. 267 – 283, 2012.

[21] H. Ashrafiuon, K. R. Muske, L. C. McNinch, and R. A. Soltan, "Sliding-mode tracking control of surface vessels," *IEEE Trans. Ind. Electron*, vol. 55, no. 11, pp. 4004–4012, 2008.

[22] H. K. Khalil, "Noninear systems," *Prentice-Hall, New Jersey*, vol. 2, no. 5, pp. 5–1, 1996.

[23] W. B. Klinger, I. R. Bertaska, K. D. von Ellenrieder, and M. R. Dhanak, "Control of an unmanned surface vehicle with uncertain displacement and drag," *IEEE J Ocean Eng*, vol. 42, no. 2, pp. 458–476, 2017.

[24] R. Skjetne, Ø. N. Smogeli, and T. I. Fossen, "A nonlinear ship manoeuvering model: Identification and adaptive control with experiments for a model ship," *Model. Ident. Control*, vol. 25, no. 1, p. 3, 2004.

[25] W. Wang, L. Mateos, S. Park, P. Leoni, B. Gheneti, F. Duarte, C. Ratti, and D. Rus, "Design, modeling, and nonlinear model predictive tracking control of a novel autonomous surface vehicle," in *Proc. 2018 IEEE Int. Conf. Robot. Autom*, 2018, pp. 6189–6196.

[26] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, 2003, pp. 2743 – 2748.

[27] G. Bishop, G. Welch *et al.*, "An introduction to the kalman filter," *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-3175, p. 59, 2001.

[28] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. dissertation, Technische Universitaet Muenchen, Germany, 2009.

[29] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, Nov 2015.

[30] H. Darweesh, E. Takeuchi, K. Takeda, Y. Ninomiya, A. Sujiwo, L. Y. Morales, N. Akai, T. Tomizawa, and S. Kato, "Open source integrated planner for autonomous navigation in highly dynamic environments," *J. Robot. Mechatron.*, vol. 29, no. 4, pp. 668–684, 2017.