Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle

Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle

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

***Abstract*— In this paper, we present the design, modeling, and real-time nonlinear model predictive control (NMPC) of an autonomous robotic boat. The robot is easy to manufacture, highly maneuverable, and capable of accurate trajectory tracking in both indoor and outdoor environments. In particular, a cross type four-thruster configuration is proposed for the robotic boat to produce efficient holonomic motions. The robot prototype is rapidly 3D-printed and then sealed by adhering several layers of fiberglass. To achieve accurate tracking control, we formulate an NMPC strategy for the four-control-input boat with control input constraints, where the nonlinear dynamic model includes a Coriolis and centripetal matrix, the hydrodynamic added mass, and damping. By integrating "GPS" modules and an inertial measurement unit (IMU) into the robot, we demonstrate accurate trajectory tracking of the robotic boat along preplanned paths in both a swimming pool and a natural river. Furthermore, the code generation strategy employed in our paper yields a two order of magnitude improvement in the run time of the NMPC algorithm compared to similar systems. The robot is designed to form the basis for surface swarm robotics testbeds, on which collective algorithms for surface transportation and self-assembly of dynamic floating infrastructures can be assessed.**

I. Introduction

As the demand increases for marine operations such as environment monitoring, hydrology survey, search and rescue, coast defence, and scientific studies, unmanned surface vehicles (USVs) have been drawing more and more attentions in recent years [1]–[9]. Autonomous boats may also be able to play important roles for the future of transportation in many coastal and riverside cities such as Amsterdam, and Venice where the existing infrastructure of roads and bridges is always extremely busy. A fleet of eco-friendly self-driving boats could shift the transport of goods and people to the waterways to get people out of their cars and reduce traffic in the city.

Our recently launched Roboat project seeks to design and develop a fleet of autonomous boats moving throughout the city of Amsterdam. Each water-based unit (a 'roboat', 4 m$\times$2 m) is intended to be used for the transportation of goods and people, and for creating dynamic floating infrastructure, like on-demand bridges and stages, that can be self-assembled or self-disassembled in a matter of hours. For the first step, we would like to develop a fleet of scale (approximately 1:4) roboats that facilitate the examination of waterway autonomy and self-assembly algorithms. To this effect, the scale roboat must be easy to manufacture, highly maneuverable and capable of accurate trajectory tracking to accomplish tasks like docking, latching, and dynamic connecting. Motivated by these functional specifications, this paper addresses the problems of design, modeling, fabrication and trajectory tracking control for such an autonomous boat.

Most of current USVs use kayak-like or catamaran shapes. These streamlined profiles can help reduce USVs' drag from the water. However, the shape irregularity of these USVs makes it difficult to execute holonomic motions and latching actions, which are particularly necessary for the transportation and self-assembly of floating structures in the confined urban water areas such as canals. For this reason, we consider an autonomous boat that has a regular hull shape. By sacrificing some shape efficiency, this novel robotic boat performs elegant holonomic motions and easily integrates a latching system onboard. Moreover, we 3D-printed the boat hull instead of using traditional machining technology to improve the ease of manufacturing the robot. The 3D-printed hull is sealed by adhering several layers of fiberglass.

Trajectory tracking control is one of the essential objectives for autonomous vehicles. There have been a number of studies on the control of ships, boats, and USVs [10], using the sliding mode method, [11], integrator backstepping method [12], [13] and adaptive control [13], [14]. It is worth pointing out that several researchers recently attempted to use the model predictive control (MPC) method for the control of USVs and underwater robots [15]–[20]. Applying model predictive control technology for USVs is a promising choice as the combination of model dynamics and cost function minimization allows for minimal tuning of controller gains. However, few studies have attempted experimental implementation of MPC on real USV platforms. One example is Shahab *et al*. [18] who implement a visual servoing MPC scheme on an underactuated underwater robotic vehicle to stabilize attitude at the desired position, but the dynamics of the robot is not considered. Another example is Bruno *et al*. [17] who demonstrate trajectory tracking control of an underactuated surface craft by implementing a nonlinear model predictive control (NMPC) strategy. However, more than 40 hydrodynamic unknown parameters in the robot

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

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

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

dynamics require identifying and the computation of the NMPC algorithm is time-consuming (0.1 s $\sim$ 0.2 s). To the best of the authors' knowledge, the efficient nonlinear model predictive control problem of USVs with a simple but effective dynamic model has not been solved, which is one of the main contributions of this study. We demonstrate the execution of the accurate NMPC tracking algorithm on an USV in under 1 ms.

This paper is structured as follows. Section II presents the design, fabrication and system implementation of the robot prototype. Section III describes the dynamic model of the robotic boat and the method for identifying the unknown parameters in the model. Section IV formulates the trajectory tracking problem and the NMPC strategy for the over-actuated boat with input and state constraints. Simulations and experimental results are presented in Section V. Section VI concludes this paper.

II. Robot Prototype Design

The scale roboat platform is used for assessing the algorithms of waterway transportation and self-assembly of floating structures, which requires that the robot be easy to manufacture, energy-saving, and highly maneuverable. Following these requirements, the design of our scale robotic boat is as follows.

First, a four-thruster propulsion system is considered for achieving high maneuverability. The "X" shaped actuator configuration is one traditional method for over-actuated boats [21], as shown in Fig. 1(a). However, this configuration

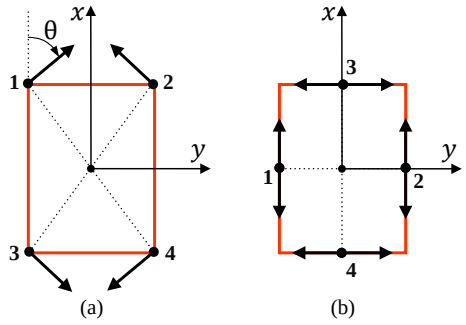


Fig. 1. Comparison of traditional and proposed actuator configuration that allows holonomic motion in the horizontal plane. (a) Traditional "X" shaped configuration; (b) "+" shaped configuration.

is inefficient because the force efficiency index is only 0.5 [22]. We therefore propose a "+" shaped actuator configuration (shown in Fig. 1(b)) whose force efficiency index is 1.0 to achieve efficient propulsion. One additional requirement for this "+" shaped configuration is that the thrusters need to generate both forward and backward forces.

Second, a cube-like shape is designed to facilitate aligning multiple boats. One conceptual floating structure created by the cube-like roboat is rendered in Fig. 2. The preconceived full size of roboat is 4 m $\times$ 2 m. For the first iteration, an approximatively 1:4 scale prototype (0.9 m $\times$ 0.45 m) is realized in Fig. 3(a).

Fig. 2. Concept of creating large floating structures using cube-like roboats.

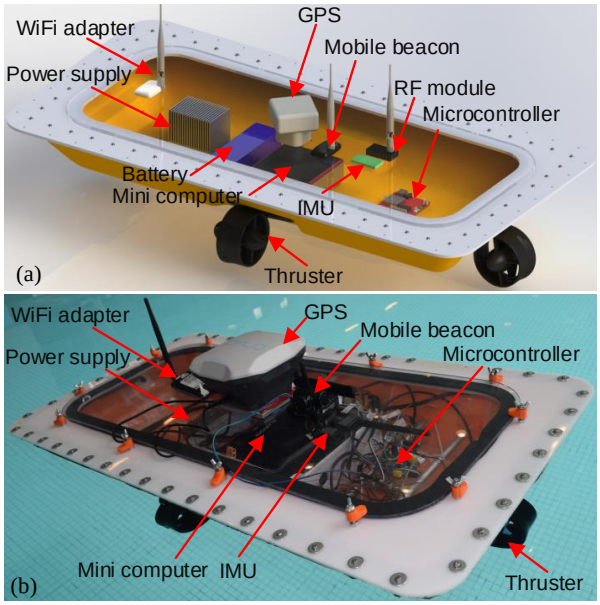
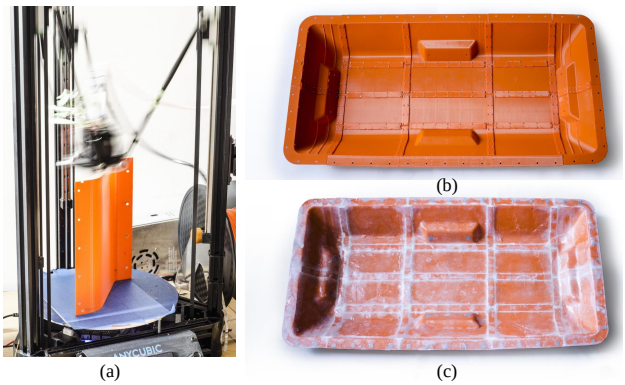


Fig. 3. The developed robotic boat. (a) Model design; (b) robot prototype.

Third, we use 3D-printing to construct the robotic boat since 3D-printing has proved to be a fast, efficient and low-cost manufacturing process. In particular, we used an Anycubic Kossel 3D printer ($\sim$\$300) to machine the hull and sealed it by adhering several layers of fiberglass inside. The manufacturing process is illustrated in Fig. 4. Note that because of the volume limitation of the printer, the boat hull is divided into 16 pieces and then spliced into one using self-tapping screws. It only takes 60 hours to print the whole hull of the boat. The whole fabrication process is easy and fast. Note that a plastic O-ring is used between the cover and robot hull to prevent water from entering. Small bolts are distributed uniformly to fasten the cover.

To guarantee the real-time performance of current optimal control and future obstacle avoidance and path-planning of our autonomous boat, a Gigabyte Mini PC (Intel Core i7-6500U) with 32 GB of memory is adopted as the main controller. Moreover, a 32-bit auxiliary processor, STM32F103, is used for basic locomotion control and multi-sensor data acquisition. Diversified sensors, including a real-time kinematic (RTK) GPS, indoor "GPS", IMU, 3D laser scanner, voltage sensor and a current sensor are installed on the robot. Specif-



(a) (b) (c)

Fig. 4. Manufacturing process of the boat hull. (a) Printing one piece; (b) connecting all printed pieces to form the hull; (c) sealing the hull.

ically, an RTK GPS (Emlid, Reach) is adopted to acquire centimeter-precision position of the robot in outdoor environments. An indoor "GPS" system (Marvelmind robotics) is employed to provide 2 cm position precision of the robot. An IMU (LORD Microstain, 3DM-GX5-25) is positioned parallel to the robot body's principal axes to monitor the yaw, pitch, roll, linear acceleration and angular velocities of the robot. A 3D LiDAR (Velodyne, Puck VLP-16) is installed on the top center of the robot for future obstacle avoidance and SLAM. LiDAR is not used in this paper. One snapshot of the developed robotic boat is exhibited in Fig. 3(b). The robot runs on the Robotic Operating System (ROS), and its detailed specifications are listed in Table I.

TABLE I

TECHNICAL SPECIFICATIONS OF THE PROTOTYPE

Items	**Characteristics**
Dimension (L×W×H)	0.90 m× 0.45 m × 0.15 m
Total mass	∼ 9.2 kg
Drive mode	T100 thrusters (2.36 kgf, forward)
Onboard sensors	RTK GPS, indoor "GPS", IMU, LiDAR
Power supply	11.1 V rechargeable Li-Po battery
Operation time	∼ 2 h
Control mode	Autonomous/Wireless mode
Maximum speed	1.2 BL (Body Length)/s

III. BOAT DYNAMICS

Following the notation developed by Fossen [23], the dynamics of a USV can be generically described by the nonlinear differential equation

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} \tag{1}$$

where $\mathbf{v} = [u \;\; v \;\; r]^T$ denotes the vehicle velocity, which contains the vehicle surge velocity (u), sway velocity (v), and yaw rate (r) in the body fixed frame, $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the positive-definite symmetric added mass and inertia matrix, $\mathbf{C}(\mathbf{v}) \in \mathbb{R}^{3\times3}$ is the skew-symmetric vehicle matrix of Coriolis and centripetal terms, $\mathbf{D}(\mathbf{v})$ is the positive-semi-definite drag matrix-valued function, $\boldsymbol{\tau} \in \mathbb{R}^{3\times1}$ the vector of body-frame forces and moments applied to the vehicle in all three DOFs and $\boldsymbol{\tau} = [\tau_1 \;\; \tau_2 \;\; \tau_3]^T$. Fig. 1 illustrates the two coordinate systems and the thruster forces acting on the vehicle.

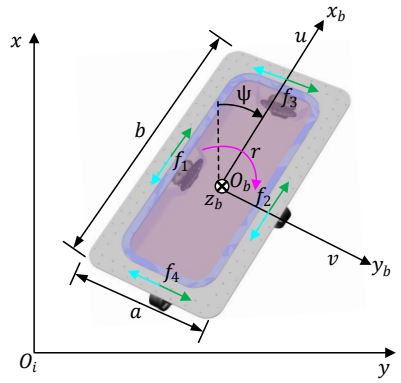


Fig. 5. Oblique view of autonomous boat. We use two coordinate systems: inertial coordinates $\sum_i$, O_i-xy and body-fixed coordinates $\sum_b$, O_b-x_by_b. Green arrows stand for positive force and blue arrows stand for negative force.

We define $\boldsymbol{\eta} = [x \;\; y \;\; \psi]^T$ as the position and orientation of the robot in the inertial frame, relative to the center of mass. The kinematic equation relating velocity components in the inertial frame to those in the body frame is described as

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v} \tag{2}$$

where $\mathbf{R}(\psi)$ is the transformation matrix converting a state vector from body frame to inertial frame

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The decoupled symmetric mass matrix is $\mathbf{M}$, where $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the sum of the vehicle mass and added mass matrix

$$\mathbf{M} = \mathrm{diag}\{m_{11}, m_{22}, m_{33}\} \tag{4}$$

The matrix $\mathbf{C}(\mathbf{v})$ also contains the rigid-body matrix and the added mass matrix. Considering that the origin O_b coincides with the center of mass of the robot, i. e., $x_G = 0$ and $y_G = 0$, $\mathbf{C}(\mathbf{v})$ can be expressed as

$$\mathbf{C}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix} \tag{5}$$

Since our robot is always moving at low speeds, the drag matrix $\mathbf{D}(\mathbf{v})$ is represented by a linear damping term. Moreover, because the platform is designed to be symmetrical with respect to the x_b and y_b axes in the body-fixed frame, the following form of the drag matrix is adopted

$$\mathbf{D}(\mathbf{v}) = \mathrm{diag}\{X_u, Y_v, N_r\} \tag{6}$$

Further, the applied force and moment vector $\boldsymbol{\tau}$ can be written as

$$\boldsymbol{\tau} = \mathbf{B}\mathbf{u} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \frac{a}{2} & -\frac{a}{2} & \frac{b}{2} & -\frac{b}{2} \end{bmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} \tag{7}$$

where $\mathbf{B}$ is the control matrix describing the thruster configuration and $\mathbf{u}$ is the control vector. a is the distance between the transverse propellers and b is the distance between the longitudinal propellers, f_1, f_2, f_3 and f_4 are the forces generated by the corresponding propellers, as shown in Fig. 5. Each propeller is fixed and can generate forward and backward forces. Finally, Eqn. 1 and 2 can be written as

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v} \tag{8}$$

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}\mathbf{B}\mathbf{u} - \mathbf{M}^{-1}(\mathbf{C}(\mathbf{v}) + \mathbf{D}(\mathbf{v}))\mathbf{v} \tag{9}$$

The complete dynamic model of the surface vehicle is reformulated by combining Eqs. (8) and (9), given by

$$\dot{\mathbf{q}} = f(\mathbf{q}, \mathbf{u}) \tag{10}$$

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

Indeed, the formulated dynamic model is a grey-box model with unknown hydrodynamic parameters. The parameters which need to be determined are the mass coefficients m_{11}, m_{22}, and m_{33}, and the drag coefficients X_u, Y_v and N_r. A grey-box identification algorithm [24] is adopted for parameter identification. Essentially, the identification process can be regarded as an optimization problem described below

$$\begin{aligned} &\arg\min_{\lambda} \sum_{t} \varepsilon(t)^T W \varepsilon(t), \\ &s.t.\ \lambda_l \leq \lambda \leq \lambda_u \end{aligned} \tag{11}$$

where $\varepsilon(t)$ donates the deviation between the experimental velocity $U_e(t)$ and the simulated velocity $U_s(t)$ at time t

$$\varepsilon(t) = U_e(t) - U_s(t), \tag{12}$$

λ represents the set of hydrodynamic parameters where $\lambda = \{m_{11},\ m_{22},\ m_{33},\ X_u,\ Y_v,\ N_r\}$. λ_l and λ_u represent the lower and upper bounds of λ, respectively. W represents a diagonal weight matrix exerting weight on each velocity component. The problem in (11) is numerically solved by a nonlinear least square method based on the trust-region-reflective algorithm.

IV. Nonlinear Model Predictive Tracking Control

To be suitable for transportation, docking and multi-boat connecting applications, it is highly desirable that the robotic boat is capable of accurate trajectory tracking as well as energy-efficient. Noticeably, model predictive control can meet the performance objectives such as minimizing the trajectory tracking error while accommodating consideration of control effort. In this section, a trajectory tracking problem is posed and an NMPC scheme is developed based on the dynamic model of the robot.

A. Trajectory Tracking Problem Formulation

A typical trajectory tracking problem is concerned with the design of control laws that force a robot to reach and follow a time parameterized reference. Essentially, it means that the robot is required to track a specified position and orientation at any given time. As $\boldsymbol{\eta}$ denotes the position and orientation of the robot at time t, we define the reference trajectory as

$$\bar{\boldsymbol{\eta}} = [x_d\ y_d\ \psi_d]^T \tag{13}$$

We define $e = [x_e\ y_e\ \psi_e]^T$ as the difference between the planned and actual position and orientation of the robotic boat at time t. The trajectory tracking problem is to design suitable control laws for the four thruster forces, to drive the error of x_e, y_e and ψ_e to zero.

B. Nonlinear Model Predictive Control Design

Nonlinear model predictive control (NMPC) determines the control action by solving a finite-horizon open-loop optimal control problem online at each sampling interval for feedback control of nonlinear systems [25]. A general form of a nonlinear system is considered as

$$\begin{aligned} &\dot{\mathbf{q}}(t) = f(\mathbf{q}(t), \mathbf{u}(t)),\ \mathbf{q}(0) = \mathbf{q}_0, \\ &\text{subject to } \mathbf{u}(t) \in U,\ \mathbf{q}(t) \in Q,\ t \in [0, \infty] \end{aligned} \tag{14}$$

where $\mathbf{q}(t) \in R^N$, $\mathbf{u}(t) \in R^M$, respectively, are the state vector and input vector. The superscripts N and M are the dimensions of the state and input vector, respectively. The set Q and U stand for the feasible states and inputs of the system.

The input applied to the system is given by the solution to the following open-loop optimal control problem with a finite horizon, which is solved at every sampling instant

$$\min_{u(\tau)} = J(q(\tau), u(\tau)) \tag{15}$$

subject to

$$\dot{\mathbf{q}}(\tau) = f(\mathbf{q}(\tau), \mathbf{u}(\tau)),\ \mathbf{q}(0) = \mathbf{q}_0, \tag{16}$$

$$\mathbf{q}(\tau) \in Q, \mathbf{u}(\tau) \in U, \forall \tau \in [t, t+T] \tag{17}$$

where T is the prediction horizon, and $J(q(\tau), u(\tau))$ denotes the objective function, and is described as

$$J(q(t), u(t)) = \int_t^{t+T} F(\mathbf{q}(\tau), \mathbf{u}(\tau)) d\tau + E(\mathbf{q}(t+T)) \tag{18}$$

where F is the cost function defining the desired performance objective and E is the terminal cost. $\mathbf{q}(\cdot)$ is the predicted state vector generated by the input signal $\mathbf{u}(\cdot) : [t, t+T] \to U$ under the initial condition q_0. Note that q_0 is generally the sensor measurements of the robot's current state in an actual system implementation. The input applied to the robot is the sequence of the optimal solutions obtained at every sampling instant δ: $\mathbf{u} = \mathbf{u}^*(\cdot, \mathbf{q}(\delta))$. The nominal closed-loop system is then formulated as

$$\dot{\mathbf{q}}(t) = f(\mathbf{q}(t), \mathbf{u} = \mathbf{u}^*(\cdot, \mathbf{q}(\delta))) \tag{19}$$

To force the robot to follow the planned path, we use the NMPC strategy as described above, and define the following quadratic cost functions as the implementation of (18)

$$F(\mathbf{q}, \mathbf{u}) = \mathbf{e}_q(\tau)^T \tilde{Q} \mathbf{e}_q(\tau) + \mathbf{u}(\tau)^T \tilde{R} \mathbf{u}(\tau) \tag{20a}$$

$$E(\mathbf{q}) = \mathbf{e}_q(t+T)^T \tilde{Q}_N(\mathbf{q}(t+T)) \mathbf{e}_q(t+T) \tag{20b}$$

where $\mathbf{e}_q(\tau) = \mathbf{q}(\tau) - \bar{\mathbf{q}}(\tau)$ where $\bar{\mathbf{q}}(\tau)$ is the reference states of the robot which takes the form $\bar{\mathbf{q}}(\tau) = [x_d(\tau)\ y_d(\tau)\ \psi_d(\tau)\ u_d(\tau)\ v_d(\tau)\ r_d(\tau)]^T$ for the robotic boat. $\tilde{Q}$ and $\tilde{R}$ are the positive definite weight matrices that penalize deviations from the desired values. $\tilde{Q}_N$ is the terminal penalty matrix to improve the stability of the NMPC algorithm. From the literature [26], it can be concluded that the stability of the proposed algorithm can be achieved for relatively long horizons by tuning $\tilde{Q}$, $\tilde{R}$ and T. Based on the built dynamic model of the robot (8)-(10), we implement the NMPC algorithm to efficiently solve the optimal control problem in (15)-(17) in real time, thereby, achieving accurate optimal trajectory tracking for the robotic boat.

V. Experiments and Results

This section contains the results of model identification, simulations, and indoor and outdoor experiments with the developed prototype to validate the efficacy of the trajectory tracking NMPC strategy presented in this paper.

A. Experimental Setup

The experiments for model identification and indoor trajectory tracking were conducted in a 12 m× 6 m swimming pool as shown in Fig. 6(a). The aforementioned indoor

Fig. 6. Indoor and outdoor experimental scenes of the robotic boat. (a) Swimming pool, and (b) Charles River.

"GPS" system contains several stationary ultrasonic beacons, one mobile beacon and one rover. The outdoor tracking experiments were performed in the Charles River, as shown in Fig. 6(b). Since the GPS position is not that stable in the river, we used extended kalman filter (EKF) to obtain a more stable position by integrating GPS position and linear accelerations of the robot. The update rate of all the sensors is set to be 5 Hz.

Both the simulations and the implementation of the NMPC strategy for the robot make use of the ACADO Model Predictive Control Toolkit [27]. The controller is implemented in a mini computer running Ubuntu 16.04 as described above. This computer communicates with an STM32 microcontroller responsible for acquiring data from GPS and IMU, sending this data to the NMPC computer, receiving the result of the controller, and changing the actuation values of each motor on the robotic boat accordingly. Note that the optimal control forces are converted to the actuation values (PWM) for each thruster according to the data chart provided by Bule Robotics. The NMPC algorithm is running on the ROS software framework. The reference trajectory, sampling time, and prediction horizon used in real-time trials are exactly the same as those used in the simulation results.

B. Results of Hydrodynamic Parameter Identification

An accurate dynamic model of robot is crucial for NMPC to achieve accurate trajectory tracking. We therefore identify the unknown parameters in the model experimentally. The identification data was gathered when the robotic boat was commanded to follow a sinusoidal path in swimming pool under closed-loop control. This excitation resulted in coupled motion in the surge, sway, and yaw degrees-of-freedom. The input forces f_1, f_2, f_3, f_4 and the corresponding robot states x, y, ψ, u, v, r were recorded at a rate of 5 Hz in the experiments. The magnitude and frequency of the sinusoidal profile is 0.89 m and 0.032 Hz, respectively. We did 5 experiments to decrease the measurement errors. The duration of each identification experiment is 50 s.

The lower bounds, upper bounds, and initial values of the hydrodynamic parameters λ are listed in Table II. The

TABLE II
Results of Hydrodynamic Parameter Identification

Item	m_{11}	m_{22}	m_{33}	X_u	Y_v	N_r
λ_l	1	1	0.01	0.1	0.1	0.01
λ_u	30	60	10	10	20	5
λ_0	8	8	2	3	8	1
λ^*	12.982	23.318	1.273	6.012	7.112	0.771

Note: the notation λ_0 stands for the initial value of λ, and λ^* represents the identified parameters.

initial values are set according to the rough estimates from [23]. Considering the nonlinearity of the problem in (11), we restrict the range of the parameters to prevent the solution from drifting away. We employed the nonlinear grey-box identification tool in the system identification toolbox of Matlab. Consequently, we obtain the identified hydrodynamic parameters λ listed in Table II.

To validate the identified parameters, we compare the simulated velocities with those measured from experiments, as shown in Fig. 7. Intuitively, the simulated motion data has a good agreement with the measured data. The small discrepancies are probably attributed to the simplification of the mass and drag matrices used in the model. The NRMSE (normalized root mean square error) fitness value between

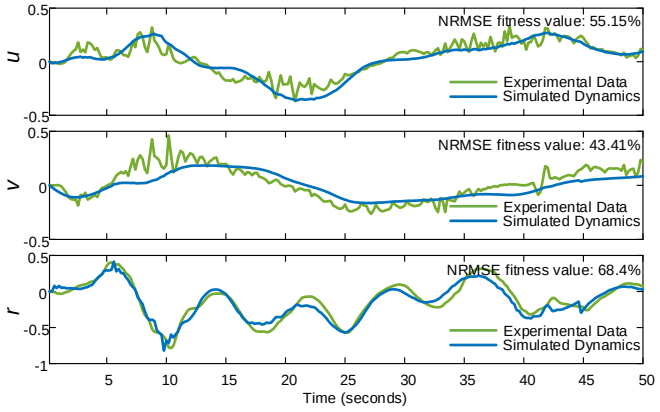


Fig. 7. The comparison between simulated and experimental velocities when the robot performs sine-curve movements.

the simulated and experimental data shown in Fig. 7 is defined as follow (take u for instance)

$$\xi = 1 - \frac{\|u_e - u_s\|}{\|u_e - \bar{u}_e\|} \tag{21}$$

where ξ is the fitness value, u_e and u_s are the experimental and simulated speeds, and $\bar{u}_e$ is the mean value of u_e.

C. Experimental Tracking Results

Based on the above identified dynamic model, we validated the presented NMPC method for the robotic boat in both simulations and experiments. In real applications, we should limit the maximum speed of the robot to prevent damage to people and goods being transported. For example, the boat velocities are always below 10 km/h (2.78 m/s) in Amsterdam canals. Considering the scale between the current boat and the full size boat, we limit the maximum speed of the robot to be 0.69 m/s, corresponding to the maximum force of nearly 2 N. Therefore, we obtain the following constraints on the input force matrix $\mathbf{u}$: $-f_m \leq f_i \leq f_m$, where $i = 1,\ 2,\ 3,\ 4$ and $f_m = 2$ N. The parameters used to implement the NMPC algorithm are listed as follows: sampling interval: $\delta = 0.2$ s; prediction time horizon: $T = 4$ s; weighting matrix: $\tilde{Q} = \text{diag}\{20,20,5,0.001,0.001,0.001\}$, $\tilde{R} = \text{diag}\{1,1,1,1\}$, $\tilde{Q}_n = \text{diag}\{20,20,5,0.001,0.001,0.001\}$.

Two references with a constant speed U_d (0.2 m/s) along the path are adopted to validate the presented control system. One desired trajectory is a sine shaped path. For indoor experiments, the desired sine path oscillates along the x axis where the desired velocity components along the x and y axis are designed as follows

$$\dot{x}_d = \frac{U_d}{\sqrt{1+\cos^2 x_d}},\ \dot{y}_d = \frac{U_d \cos x_d}{\sqrt{1+\cos^2 x_d}} \tag{22}$$

The desired control point position components x_d and y_d are calculated by integration. The desired robot direction is aligned with the tangent of the desired path, as defined below

$$\psi_d = \arctan(\cos(x_d)) \tag{23}$$

Fig. 8 and 9 shows the simulated and experimental tracking results of the NMPC boat with nonzero initial errors in a swimming pool. It can be observed that the NMPC

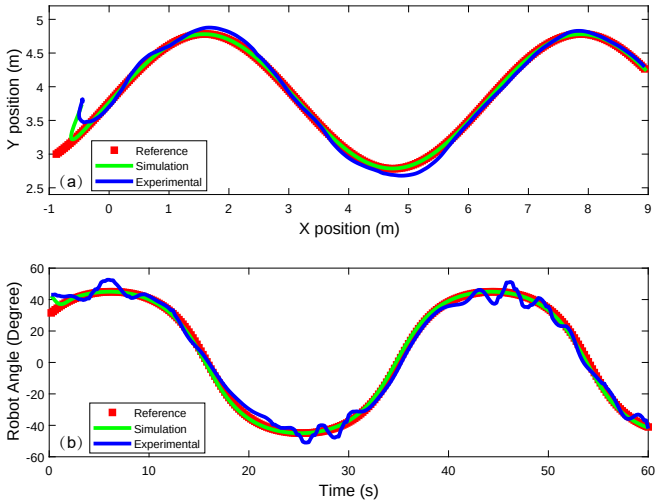


Fig. 8. Performance of indoor sine curve trajectory tracking using NMPC. (a) Position and (b) orientation.

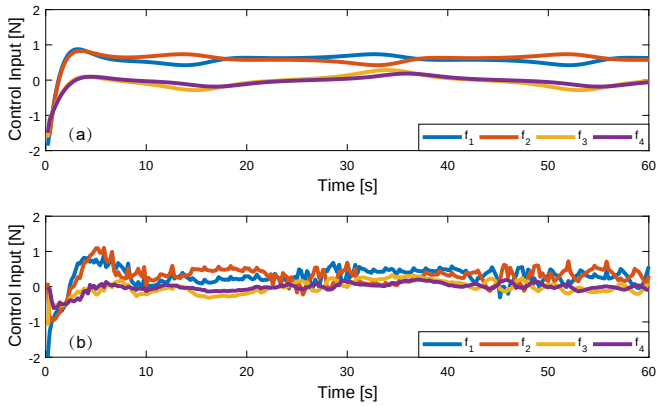


Fig. 9. Optimal control actions of NMPC while tracking a sine trajectory in the swimming pool. (a) Simulation and (b) experimental results.

method could successfully control the robotic boat along the desired sine-shaped trajectory and orientation. The controller is able to track the desired trajectory perfectly in simulations, while sustaining minimal tracking error in the experiments. Moreover, the optimal control actions in the experiments are less smooth than in the simulations. That may be caused by the simplified dynamic model of the robot as well as the measurement noise in the robot state.

We did 10 sine-shaped trajectory tracking experiments to quantify the control performance. The analysis of these experimental results shows that the maximum errors in position and orientation are lower than 0.302 m and 8.2°, respectively. The average error for the position and orientation, respectively, are 0.063 m and 3.3°. At present, we didn't find any trajectory tracking results of USVs that use MPC method in the indoor environment. Nevertheless, our tracking errors are very close to those achieved using other model-based controllers with a fully coupled complicated dynamic model [28].

Moreover, in order to illustrate the NMPC tracking behavior in extreme conditions, including discontinuities in the reference states, rectangle shaped trajectory tracking experiments were conducted. The desired position and orientation for a rectangle path are easy to express. Fig. 10 and 11 shows the simulated and experimental tracking results of the NMPC boat in a swimming pool. Similarly, the NMPC method

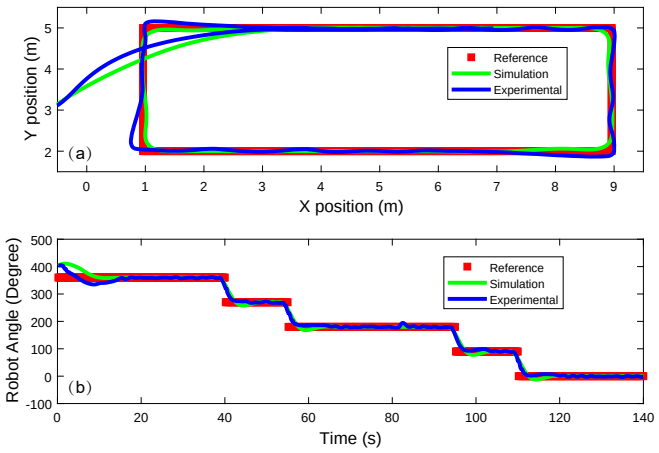


Fig. 10. Performance of indoor rectangle trajectory tracking using NMPC. (a) Position and (b) orientation.

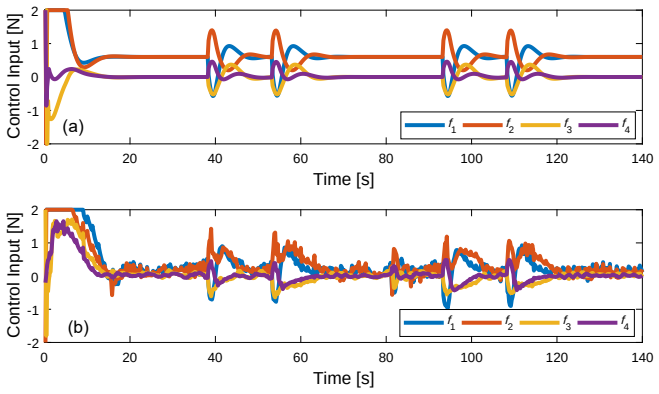


Fig. 11. Optimal control actions of NMPC while tracking rectangle trajectory in the swimming pool. (a) Simulation and (b) experimental results.

could also successfully control the robot along the desired rectangle-curve trajectory and orientation. Moreover, it can be observed that the robot requires more force to follow its four discontinuous references. The good fit between simulations and experiments indicates that the identified dynamic model is sufficient for the formulated NMPC algorithm. We did 5 rectangle-curve tracking experiments to quantify the control performance. The maximum position and direction errors are, respectively, lower than 0.361 m and 18.5°. The average absolute error for the position and angle are, respectively, 0.095 m and 6.3°.

Furthermore, we conducted NMPC tracking experiments in the Charles River where there are significant currents and waves. The reference is a sine shaped path that oscillates along the y axis. Fig. 12 and 13 show the simulated and experimental sine-shaped tracking results of the boat using NMPC. It is clear that the NMPC method still works well in

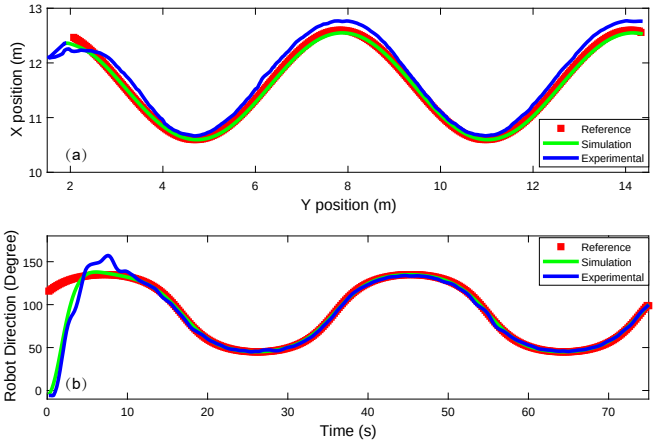


Fig. 12. Performance of the outdoor sine trajectory tracking using NMPC. (a) Position and (b) orientation.

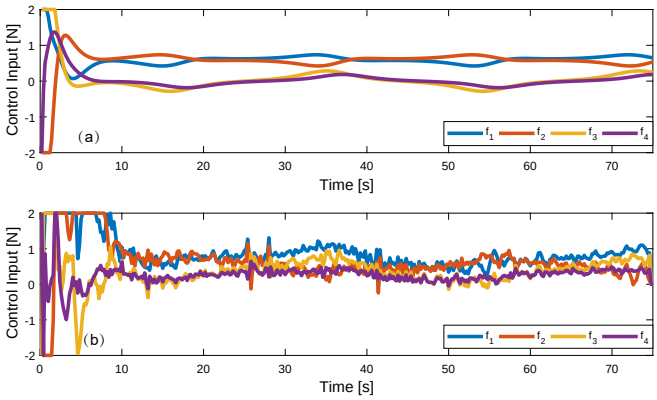


Fig. 13. Optimal control actions of NMPC while tracking sine trajectory in the river. (a) Simulation and (b) experimental results.

natural waters in spite of acquiring slightly larger positional divergences than in indoor environments. A more careful inspection indicates that the control inputs in Fig. 13(b) are larger than those in Fig. 9(b) because the robot takes more effort to counteract the currents and waves around it. Five tracking experiments in the river show that the maximum position and direction errors are lower than 0.32 m and 4.8°, respectively. The average errors are 0.13 m and 1.2°, respectively. Although our NMPC strategy uses a simple dynamic model, its tracking errors are much smaller than those in [29] where the authors employ a complicated dynamic model for their USV. We attribute this to three main reasons. First, our experimentally identified dynamic model may be more accurate than that in [29], which guarantees the accurate state prediction of the NMPC algorithm. Second, the problem reformulation in [29], from a constrained optimization to an unconstrained optimization, may lead to performance reduction in the optimization. Third, the measurement errors may be much smaller here since our robotic boat is able to localize itself with a stable centimeter-level precision.

Finally, we would like to highlight that the implemented NMPC algorithm is extremely efficient because of C/C++ code generation based on ACADO toolkit. In particular, the computation time needed to determine the next control action is always below 1 ms for our algorithm, meaning that the control frequency can reach up to 1000 Hz, which will be very beneficial to high-speed and high-accuracy applications. By contrast, the computation time of the NMPC method in [29] is about 100 ms with a similar computer configuration, a difference of almost two orders of magnitude.

VI. Conclusion and Future Work

This paper presented the design, modeling, and real-time nonlinear model predictive control (NMPC) of an autonomous robotic boat. The novel robotic boat is easy to manufacture, highly maneuverable, and capable of accurate trajectory tracking in both indoor and outdoor environments. To achieve accurate tracking control, we first formulated a simple dynamic model for the robot with six unknown parameters through a nonlinear least square, which ensures the control accuracy of the NMPC strategy to some extent. Finally, we formulated an NMPC strategy for the four-control-input boat with control input constraints. The simulation and experimental results have validated the accuracy and and efficiency of the proposed NMPC method for autonomous surface vehicles, which will be paramount in accomplishing advanced autonomy tasks such as moving obstacle avoidance.

In the future, our work will be extended in the following directions. First, the mass and drag of the robotic boat may change drastically when transporting people and goods. Therefore, incorporating an online model identification system and adaptive controllers will be one of our next steps. Second, wave disturbances always exist in natural waters, which will be addressed in our future controller design.

References

[1] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu, "Experiments with underwater robot localization and tracking," in *2007 IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 4556–4561.

[2] M. Doniec, I. Vasilescu, C. Detweiler, and D. Rus, "Complete SE3 underwater robot control with arbitrary thruster configurations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 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 *MTS/IEEE Proceedings of OCEANS*, 2005, pp. 725–729.

[4] W. Wang and G. Xie, "Online high-precision probabilistic localization of robotic fish using visual and inertial cues," *IEEE Transitions on Industrial Electronics*, vol. 62, no. 2, pp. 1113–1124, 2015.

[5] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[6] A. Dhariwal and G. S. Sukhatme, "Experiments in robotic boat localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 1702–1708.

[7] 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.

[8] L. Wen, T. Wang, G. Wu, and J. Liang, "Quantitative thrust efficiency of a self-propulsive robotic fish: Experimental method and hydrodynamic investigation," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 1027–1038, June 2013.

[9] W. Wang, J. Liu, G. Xie, L. Wen, and J. Zhang, "A bio-inspired electrocommunication system for small underwater robots," *Bioinspiration & Biomimetics*, vol. 12, no. 3, p. 036002, 2017.

[10] H. Ashrafiuon, K. R. Muske, and L. C. McNinch, "Review of nonlinear tracking and setpoint control approaches for autonomous underactuated marine vehicles," in *American Control Conference (ACC), 2010*, 2010, pp. 5203–5211.

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

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

[13] 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 Journal of Oceanic Engineering*, vol. 42, no. 2, pp. 458–476, 2017.

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

[15] Y. Siramdasu and F. Fahimi, "Incorporating input saturation for underactuated surface vessel trajectory tracking control," in *Proceedings of the American Control Conference*, 2012, pp. 6203–6208.

[16] S. K. Sharma and R. Sutton, "An optimised nonlinear model predictive control based autopilot for an uninhabited surface vehicle," *IFAC Proceedings Volumes*, vol. 46, no. 10, pp. 73–78, 2013.

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

[18] S. Heshmati-Alamdari, A. Eqtami, G. C. Karras, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A self-triggered visual servoing model predictive control scheme for under-actuated underwater robotic vehicles," in *IEEE International Conference on Robotics and Automation ICRA*, 2014, pp. 3826–3831.

[19] D. C. Fernandez and G. A. Hollinger, "Model predictive control for underwater robots in?ocean waves," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 88–95, 2017.

[20] H. Li and W. Yan, "Model predictive stabilization of constrained underactuated autonomous underwater vehicles with guaranteed feasibility and stability," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1185–1194, 2017.

[21] D. Nad, N. Miskovic, and F. Mandic, "Navigation, guidance and control of an overactuated marine surface vehicle," *Annual Reviews in Control*, vol. 40, pp. 172–181, 2015.

[22] H. Kharrat, "Optimization of thruster configuration for swimming robots," Ph.D. dissertation, Rice University, 2015.

[23] T. I. Fossen, *Guidance and control of ocean vehicles*. West Sussex PO19 1UD, England: John Wiley & Sons Ltd, 1994.

[24] J. Yu, J. Yuan, Z. Wu, and M. Tan, "Data-driven dynamic modeling for a swimming robotic fish," *IEEE Transaction on Industrial Electronics*, vol. 63, no. 9, pp. 5632–5640, 2016.

[25] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practicea survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[26] L. Grne and J. Pannek, *Nonlinear model predictive control: theory and algorithms*. Springer Publishing Company, Incorporated, 2013.

[27] B. Houska, H. Ferreau, F. Logist, and M. Diehl, "Acado toolkitautomatic control and dynamic optimization," 2012, optimization in Engineering Center (OPTEC), http://acado.github.io/.

[28] S. C. Martin and L. L. Whitcomb, "Nonlinear model-based tracking control of underwater vehicles with three degree-of-freedom fully coupled dynamical plant models: Theory and experimental evaluation," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–11, 2017.

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