2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

A Turning Radius Prediction Scheme for Sailing Robots under Complex Marine Environment

Weimin Qi, Qinbo Sun and Huihuan Qian†

***Abstract*— This paper presents a strategy for predicting the turning radius of a sailing robot with consideration of aerodynamic and hydrodynamic interferences from the marine environment. The turning radius is initially obtained based on three consecutive designated points during the turning process, which is regarded as the baseline method. Subsequently, on the basis of our constructed turning datasets, a model is trained using Gaussian process regression (GPR) to achieve radius prediction. The feasibility and effectiveness of the proposed scheme have been validated in both simulation and experiments (conducted with OceanVoy as shown in Fig.1). Under experimental circumstances, the Mean Absolute Error (MAE) of the turning radius produced by the trained prediction model is 0.58m. Furthermore, it has been observed that during long-term sailing covering a distance of 1200km, apart from wind speed and robot velocity, the tidal range also has a significant impact on the navigation of sailing robots.**

I. Introduction

The navigation of a sailboat is highly nonlinear, which is attributed to its dependence on unpredictable and variable wind as the sole propulsion force, as well as its underactuated nature. Moreover, disturbances from the marine environment such as waves, currents and tides have a significant impact on the routing of sailing robots that cannot be ignored [1]. When dividing the sailing path into turning segments (TS) and approximately straight-line portions (ASP), the control for turning behavior is more challenging than that of ASP. This is because turning operations, whether tacking or gybing, involve passing through the no-go zone [2], which poses a threat to the safety of the robot and the stability of its control. In addition, environmental disturbances need to be considered more carefully in turning operations because the robot must adjust the rudder and sail at the appropriate moment to complete the turn.

As such, the ability to estimate the turn efficiently is an essential capability that can provide guidance for sailboat navigation and timely warnings for dangerous situations. This estimation method is generally applicable to sailboat robot turning and is particularly relevant in fields where higher performance is required, such as unmanned sailing competitions (e.g., the Microtransat Challenge [3] and World Robotic Sailing Competition (WRSC) [4]) or ocean observation in restricted areas [5].

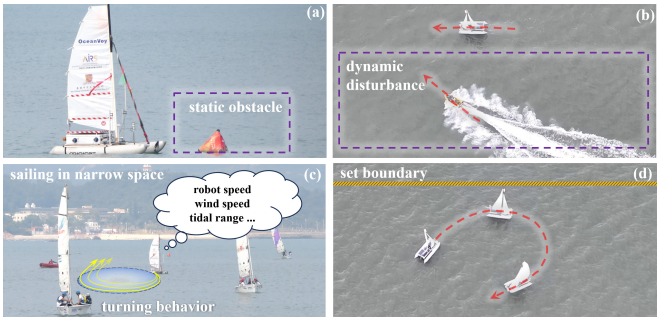


Fig. 1. Four examples of scenarios where OceanVoy (a catamaran with a length of 3.1m) needs to perform turning operations, which illustrate the importance of turning radius for safety-critical navigation in complex marine environments: (a) encountering static obstacles, (b) encountering dynamic wakes from the surrounding vehicle, (c) navigating in crowded conditions, and (d) approaching a preset boundary. In (c), it is clear that the turning radius of a sailboat is impacted by many factors. This paper primarily focuses on the effects of environmental factors and robot states.

Surprisingly, there appears to be little existing literature regarding the turning behavior estimation of sailing robots. Consequently, reference revolves around existing investigations for sailboats to duel with marine environmental disturbances and estimation methods related to our problem. It should be noted that the disturbances mentioned in this study regarding sailboats in the marine environment refer specifically to the influences from the water field, as the wind is regarded as a propulsion force.

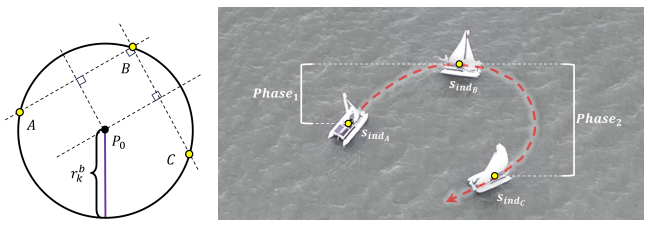
Environmental disturbances are important factors affecting sailboat navigation; however, only a small portion of existing research has taken them into consideration. The influence of ocean currents is significant on sailing routes, as pointed in the literature [6], [7], but they did not incorporate ocean currents in their methods implementation. In [8], the hill-climbing algorithm designed for online optimization implicitly considered the drifts from currents, but provided no further explicit information. A Gaussian process model predictive control (GPMPC) method was proposed to improve the navigation tracking accuracy of sailboats, in which waves with random wave spectrum was generated to learn feature model of the sailing course change [9]. This method has been validated in simulation. The methods mentioned above, however, cannot be directly applied to our problem.

Drawing inspiration from the real-time region estimator described in [10], this paper proposes a radius estimator for autonomous sailboat in the presence of external environ-

This paper is partially supported by Project 2022A1515240063 from Guangdong Basic and Applied Basic Research Foundation, University Stability Support Program from Shenzhen Science and Technology Innovation Commission, and Project AC01202101105 from Shenzhen Institute of Artificial Intelligence and Robotics for Society, China.

All authors are from Shenzhen Institute of Artificial Intelligent and Robotics for Society, The Chinese University of Hong Kong, Shenzhen, Guangdong, China.

† Corresponding author is Huihuan Qian, hhqian@cuhk.edu.cn



(a) (b)

Fig. 2. Calculation of the turning radius in TS. (a) shows the method of radius calculation. A, B and C are three given points to calculate the circle center P_0 and turning radius r_k^b. (b) is an example of a turning process, with key points A, B and C obtained from the state points s_{ind_A}, s_{ind_B} and s_{ind_C} separated by the two phases, $Phase_1$ and $Phase_2$.

Algorithm 1 Baseline Method: $r_k^b = F\left(\mathbf{s}_{ind_A}, \mathbf{s}_{ind_B}, \mathbf{s}_{ind_C}\right)$

Input: $\mathbf{s}_t \in S_k \subseteq \mathbb{S} \subseteq \mathbb{R}^{n_s}$ //State in TS
Output: r_k^b //Calculated turning radius

$P_0 = (P_{0(x)}, P_{0(y)})$
$ind_A = find(s_t^{(x,y)} |$ adjust $\delta_r(t)$ to turn$)$;
$ind_B = find(s_t^{(x,y)} |$ furthest to A and C in path$)$;
$ind_C = find(s_t^{(x,y)} |$ adjust $\delta_r(t)$ to normal$)$;
$A = s_{ind_A}^{(x,y)}, B = s_{ind_B}^{(x,y)}, C = s_{ind_C}^{(x,y)}$
$s_{t_A} = \|P_0, A\|$, $s_{t_B} = \|P_0, B\|$, $s_{t_C} = \|P_0, C\|$;
$[x, y] =$ Solve $(s_{t_A}, s_{t_B}, s_{t_C}, P_0)$;
$P_{0(x)} = x$, $P_{0(x)} = y$
$r_k^b = \|A, P_0\|$

mental interference. We choose the widely used Gaussian process regression (GPR) to train a model that captures the relationship between factors and the turning radius based on our own datasets. This paper's contributions are threefold: (1) formulating the problem of estimating turning radius under environmental disturbances, (2) proposing a GPR-based scheme to address this problem, (3) validating the proposed method through simulation and experiments. To this end, extensive experiments have been conducted in real ocean environment and turning data was collected to facilitate the learning of radius prediction strategies. As shown in Fig.1, the predicted turning radius derived from the proposed method can be directly applied to tasks such as risk warning, collision avoidance, and race scoring in navigation.

The rest of the paper is organized as follows: Section II is about the problem formulation and an overview of methodology, and details of the proposed approach is elaborated in Section III. Results of simulation and experimental validation are presented in Section IV and Section V, respectively. Conclusions given in Section VI.

II. Problem Formulation and Overview

In this section, we formulate the turning radius estimation problem of sailing robots and briefly summarize the framework of our methodology.

As mentioned in Section I, we divide the whole course of sailing into turning segments (TS) and approximately straight-line portions (ASP), which are represented as C_{TS} and C_{ASP}, respectively. As shown in Fig.1, the turning radius in C_{TS} is influenced by many factors (e.g., wind field, water field, static and dynamic obstacles, and human interference) in the complex marine environment. Therefore, our goal is to estimate the turning radius considering these exogenous disturbances.

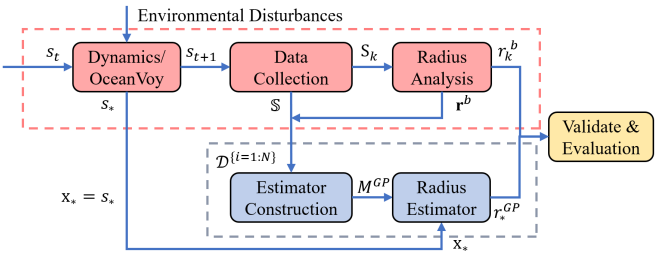


Fig. 3. Architecture of our proposed turning radius estimation approach and illustration of symbols. $\mathbf{s}_t$, $\mathbf{s}_{t+1}$ are sailboat state at time step t and $t+1$, respectively. S_k represents turning segments for radius calculation. $\mathbf{s}_*$ is the input of the proposed radius estimator. $\mathbb{S}$ contains turning segments for training datatset construction. $\mathcal{D}^{\{i=1:N\}}$ is the constructed dataset, where i is the index of C_{TS}. r_k^b and r_*^{GP} represent radius calculated with baseline method and M^{GP} model, respectively.

Set $\mathbf{s}_t \in \mathbb{S} \subseteq \mathbb{R}^{n_s}$ as the system state at times t_i. Let $u_t \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ and $\omega_t \in \mathbb{W} \subseteq \mathbb{R}^{n_\omega}$ denote the control input and exogenous disturbances, respectively. The system dynamic equation governed by locally drift dynamics g is

$$\dot{\mathbf{s}}_t = g(\mathbf{s}_t, t) + u_t + \omega_t \tag{1}$$

Turning radius calculation is related to the equation of a circle as

$$\left(x - P_{0(x)}\right)^2 + \left(y - P_{0(y)}\right)^2 = r^2 \tag{2}$$

where $P_0 = (P_{0(x)}, P_{0(y)})$ is the center coordinate and r is the turning radius.

As shown in Fig.2(a), its simple to derive a circle and the corresponding radius based on three consecutive specified points A, B and C. In Fig.2(b), we apply this as a baseline method to determine the radius corresponding to C_{TS}. In the baseline method, the turning radius r_k^b is calculated with function F by

$$r_k^b = F\left(\mathbf{s}_{ind_A}, \mathbf{s}_{ind_B}, \mathbf{s}_{ind_C}\right) \tag{3}$$

where $\mathbf{s}_{ind_A}$, $\mathbf{s}_{ind_B}$ and $\mathbf{s}_{ind_C}$ are the initial (activating the turning maneuver firstly), middle (changing the heading angle again to finish the turn) and final (finishing the turn successfully) states, respectively. $\mathbf{r}^b$ is composed by r_k^b in different S_k (turning segments for radius calculation).

In Algorithm 1, by assuming the center of the circle as point P_0, the distance equations between center point and the three key state points A, B and C are solved simultaneously. The function Solve$(\cdot)$ is utilized to compute these equations and obtain the estimated center P_0. Finally, the initial turning point A determines the estimated radius r_k^b.

The architecture proposed in Fig.3 includes definitions for turning radius, solvers, modeling and prediction based on the state data of a sailing robot. The red dashed box and red rectangles represent the baseline method, which is utilized

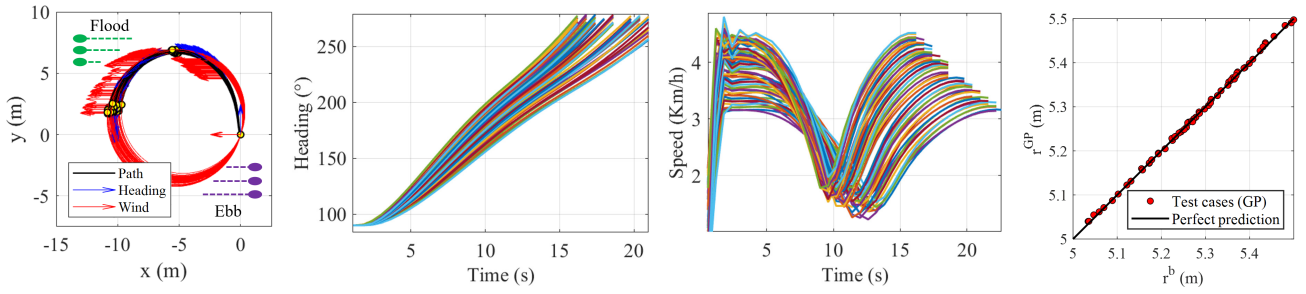


Fig. 4. Simulation implementation. (a) TS in simulation test platform on Matlab. The yellow dots represent the selected three key states. The red curves represent the circles corresponding to turning radii calculated based on the baseline method. Black curves are the paths of the robot. The blue and red arrow show the heading of the robot and wind direction, respectively. The tidal current for flood (green) and ebb (purple) are characterized by their respective directions along the negative and positive x-axis. (b) and (c) show the changing tendency in heading and speed corresponding to (a). Each curve corresponds to a simulated turning result under varied environments and speed. (d) Model between the calculated and predicted turning radii.

to obtain the radius in simulated or real-world turning data. The blue dashed box is an estimator that is built based on the turning data and is used to forecast the turning radius. As indicated in the yellow box, the estimator need to be verified and evaluated using the baseline method.

III. The Turning Radius Prediction Scheme

In this section, we first introduce the data collection methods for both simulation and experiments separately. Then, we used the well-known GPR to determine the feature model that reflects the impact of factors on the radius.

A. Data Collection

In this study, the construction of the dataset $\mathcal{D}$ is based on the collected data $\mathbb{S}$. The datasets used for training and testing are $\mathcal{D}^{train}$ and $\mathcal{D}^{test}$, respectively. For each segment of TS data in $\mathbb{S}$, it is subjected to dimensional reduction analysis and produced a state vector of sailing robot at key state s_{ind_A}. The target variable (radius $\mathbf{r}^b$) of the dataset, is calculated based on each segment of TS data. Therefore, each row of the dataset $\mathcal{D}$ consists of the state s_{ind_A} in C_{TS} and the r_k^b.

We conduct data acquisition for both simulation and experiments. A simulated dataset $\mathcal{D}_{sim}$ is built with a platform in Matlab, including the sailboat physical model, control strategy and environmental disturbances. Experimental dataset $\mathcal{D}_{exp}$ will be collected using the sailing robot OceanVoy [11] as shown in Fig.1 to sail at sea for a sustained period of time. Compared with $\mathcal{D}_{sim}$, $\mathcal{D}_{exp}$ enriched with information from the real marine environment.

The main step for data collecting procedures are described as follows:

- Set the navigation area in the testing platform;
- Design control commands and apply them to a robot;
- Sail repeatedly and store data.

In simulation, we adopt the sailboat model in [12], which exhaustively introduces the relevant parameters and updating strategy of the states. We added the effect of tides to the model as the second-order wave disturbance [13], which is divided into two cases of flood tide and ebb tide. In this work, we define the flood tide as the negative direction of the x-axis and the ebb tide as the positive direction of the x-axis. Tidal range refers to the difference between the highest and lowest tidal levels in a tidal cycle. It is used to reflecting the tidal current and updating the apparent water speed v_{at} and angle ψ_{at}, respectively. The forces generated by sail and rudder are shown as follows

$$\begin{pmatrix} g_s \\ g_r \end{pmatrix} = \begin{pmatrix} p_4 a_{\mathrm{aw}} \sin\left(\delta_s - \psi_{\mathrm{aw}}\right) \\ p_5 v_{\mathrm{at}}^2 \sin\left(\delta_r - \psi_{\mathrm{at}}\right) \end{pmatrix} \tag{4}$$

where p_4 and p_5 are given sail lift and rudder lift, respectively. a_{aw} and ψ_{aw} represent the speed and angle of apparent wind (aw), respectively. δ_s and δ_r are sail angle and rudder

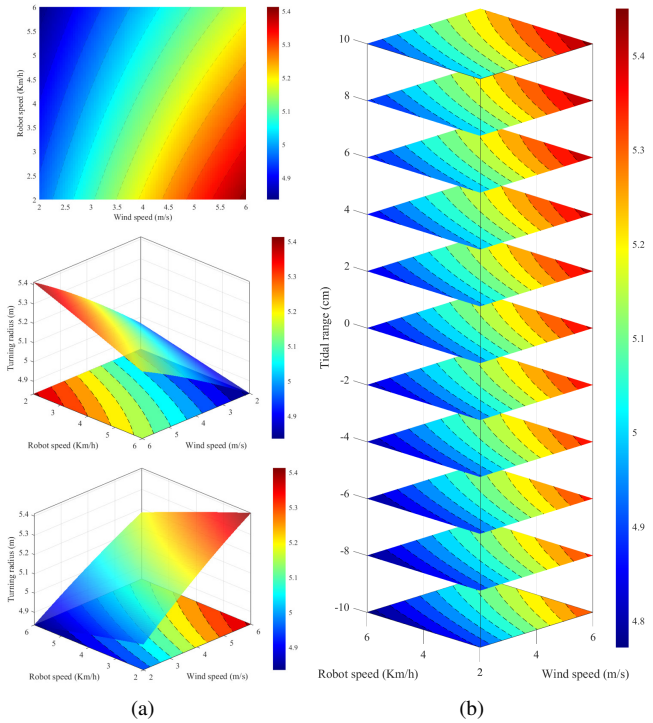
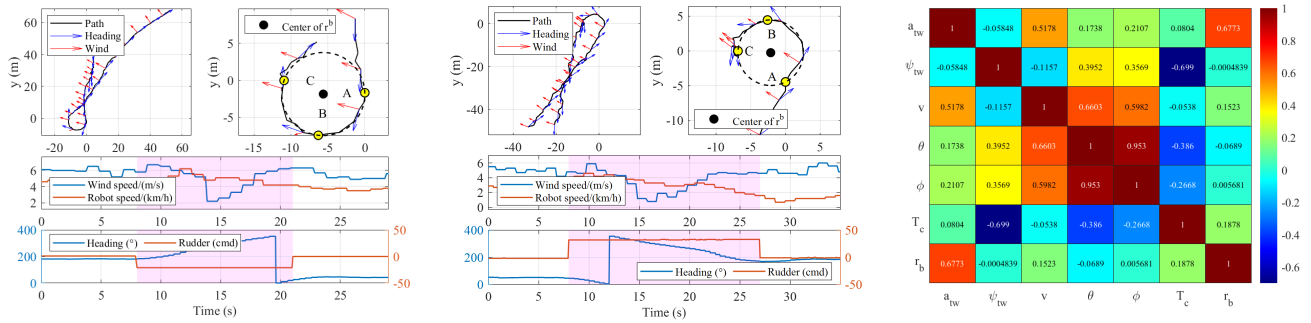


Fig. 5. Simulated results for turning radius predicted by the estimator M_{sim}^{GP}. (a) Predicted turning radius under different simulated wind speeds and robot speeds, with the overhead view, front view, and side view, respectively. (b) The variation of the turning radius with tidal range. The color bars represent the value distribution of predicted turning radius.



(a) (b) (c)

Fig. 6. Experiment implementation. (a) and (b) demonstrate two turning examples of offshore experiment. The top left image illustrates the overview of the TS, with the black line representing the path and the blue and red arrows indicating the heading and wind direction, respectively. The top right image displays the magnified detail of TS and calculates turning radius and its centers using the TS, which are used to construct the real experimental dataset. The two figures below depict the collected wind speed, robot speed, heading and control commands of OceanVoy. The shaded areas represent the states corresponding to A and C. (c) The results of the correlation analysis based on the collected turning dataset via Principal Component Analysis (PCA). a_{tw}, ψ_{tw} , v, θ, ϕ, T_c and r^b are true wind speed, true wind angle, robot velocity, robot heading angle, course angle, tidal range and radius calculated with baseline method, respectively. Among these values, the wind speed, tidal range and robot speed are the main components influencing the turning radius.

angle, respectively. The robot control is achieved using the maximum rudder and sail angles as in [14], [15].

B. Training through GPR

In our constructed training datasets $\mathcal{D}^{train}$, there are input vectors $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ and the corresponding target output $\mathbf{y} = \{y_i\}_{i=1}^{N}$. We expect to approximate a latent function f that represents the correspondence between $\mathbf{X}$ and $\mathbf{y}$. Gaussian process regression (GPR) is a widely used method, and in this study, we refer to its description in [16]. For the first-order system, a zero-mean Gaussian process is applied on f as $y_i = f(x_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$. The prior on the target output is $\mathbf{y} \sim \mathcal{N}(0, \mathbf{K} + \sigma_n^2 \mathbf{I})$. The commonly used squared exponential kernel function is applied for the covariance matrix $\mathbf{K}$.

Therefore, the joint distribution of the training labels and the predicted value of the query point x_* is

$$\begin{bmatrix} \mathbf{y} \\ f(x_*) \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_n^2 \mathbf{I} & \mathbf{k}_* \\ \mathbf{k}_*^\top & k(x_*, x_*) \end{bmatrix}\right) \tag{5}$$

where $[\mathbf{k}_*]_i = k(x_i, x_*)$.

Then the mean and variance of the prediction are

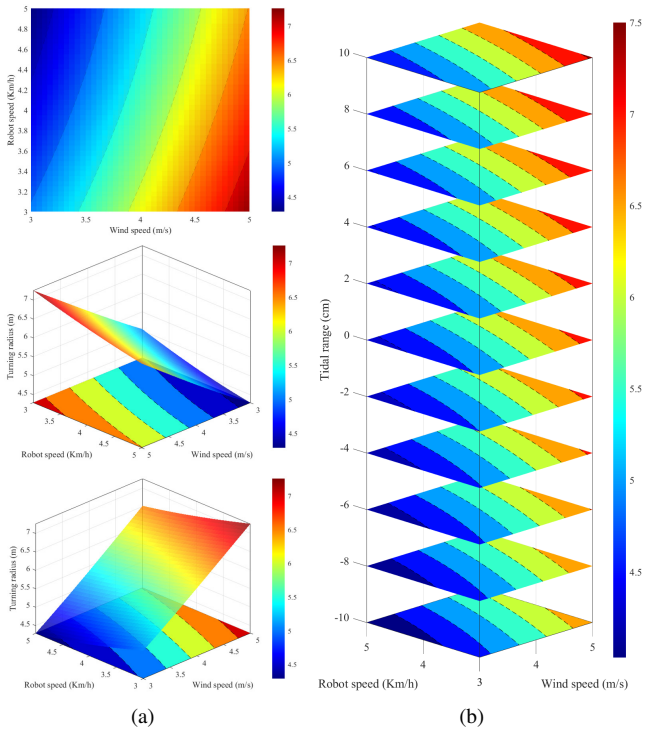
$$\begin{aligned} \mu_{x_*} &= \mathbf{k}_*^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y} \\ \Sigma_{x_*} &= k(x_*, x_*) - \mathbf{k}_*^\top \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}_*. \end{aligned} \tag{6}$$

IV. Simulation Study

In this section, simulation results are presented to show the performance of our turning radius estimation scheme proposed in Section III. All simulations were run on desktops with Intel i7-1165G7@2.80GHz with Matlab R2020b.

A. Simulated data Collection

We implement the data collecting procedure in Section III-A to obtain abundant turning segments. Algorithm 1 is used to obtain the radius in turning paths. Then we built dataset $\mathcal{D}_{sim}^{train}$ for training in Gaussian process model. The dataset will include a variety of turning processes and tidal range calculated from flood and ebb tides.

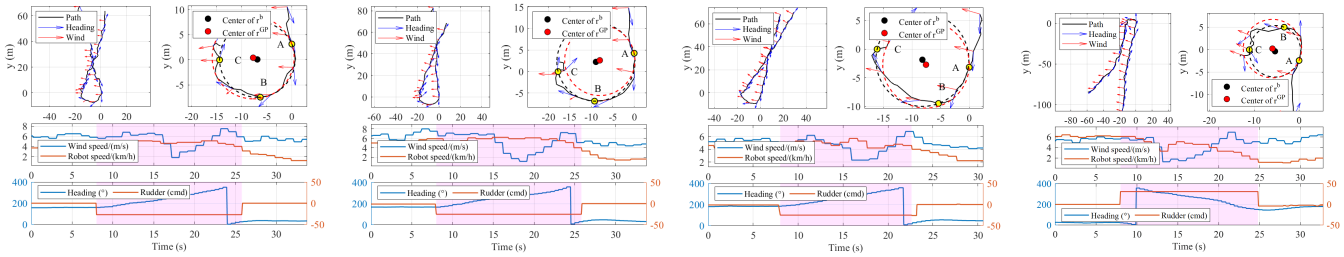


(a) (b)

Fig. 7. Experiment results for turning radius predicted by the estimator M_{exp}^{GP} under different testing scenarios. The color bars show the value distribution of predicted turning radius. (a) Results related to wind speed and robot speed in top view, front view and side view. (b) Results considering tidal range.

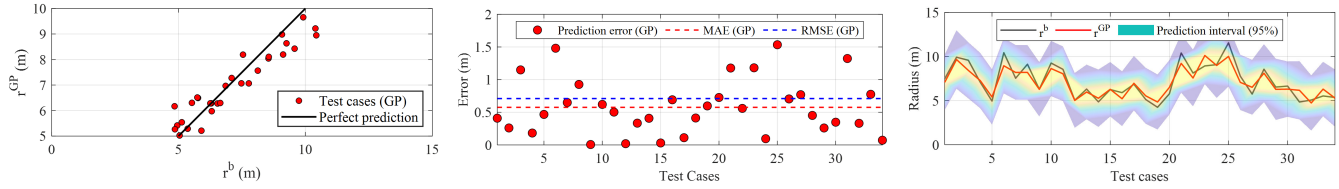
B. Simulation results

The simulated dataset is presented in Fig.4(a), in which the sailing robot starts from point $(0, 0)$. In order to control the influence variables, the wind direction is kept consistent during this data collection. Robot speed, wind speed and tidal range are the factors we focus on in terms of changes in both robot itself and environmental factors. In the offshore bay area, the ebb and flood of tides are very significant



(a) (b) (c) (d)

Fig. 8. Four groups of experimental turning radius prediction results for the cases 7, 8, 22, 27, as shown in Table I. In (a-d), the subplot of the upper right corner shows the predicted radius r^{GP} and center point based on state $\mathbf{s}_{ind_A}$ with a red dashed circle and red dot, respectively. The circle with radius r^b and center point obtained from the TS motion analysis are given by a black dashed circle and dot.



(a) (b) (c)

Fig. 9. Prediction and validation results. (a) The relationship between the calculated and predicted turning radii, and perfect prediction. (b) The error between the estimated and the calculated radii, as well as the MAE and RMSE of the error, which are 0.58 and 0.71, respectively. (c) The predicted turning radius and its prediction interval with 95% confidence.

and generate drift forces on sailboat robots. Therefore, we consider tidal range as a key factor.

Fig.4(b) shows the heading variation trend corresponding to the black courses in Fig.4(a). The bow facing due east is $\theta = 0°$, and during the turning process, there is a similar increasing trend. In Fig.4(c), as stated in [17], there is an obvious stall process during turning. Based on $\mathcal{D}_{sim}$, the learned estimator M_{sim}^{GP} is shown in Fig.4(d), in which prefect prediction has been realized.

Correlation between three main influencing factors (Robot speed, wind speed and tidal range) and turning radius is shown in Fig.5. From the Fig.5(a), the turning radius is influenced by wind speed and robot speed. The increase in wind speed causes the robot to receive an increase in thrust, so the corresponding turning radius is larger.

In Fig.5(b), the flood tide causes the radius to increase with the color warmer than that of in ebb tide. On the contrary, the ebb tide is the opposite results. This is because the scenario is built with wind direction and flood tide as the same direction as shown in Fig.4(a). Similarly, an increase in boat speed corresponds to a larger increase in radius at flood tide than at ebb tide.

V. Experiment

In this section, a series of experiments will be used to explore the key factors that affect turning radius in real sea navigation and to validate the proposed estimation scheme.

A. Set up

For all the experiments, we used a sailing robot named OceanVoy to sail in a real ocean environment. The test area is located near Daya Bay, Shenzhen, China. The dimensions of OceanVoy are $3.1\ \mathrm{m}(L), 1.4\ \mathrm{m}(W), 3.8\ \mathrm{m}(H)$. It is equipped with a weather station, a Geographical Positioning System (GPS) and an inertial measurement unit (IMU). The weather station with inner GPS and IMU can obtain true and relative wind speed and direction. Additionally, tidal data for the testing area was obtained from the local weather platform.

We have achieved a continuous cumulative sailing mileage of more than $1200km$ in real marine environments. It contains a wealth of data (e.g., robot position, heading and wind velocity). We extracted 67 and 34 turning segments C_{TS} from real-world navigation scenarios as the training dataset $\mathcal{D}_{exp}^{train}$ and testing dataset $\mathcal{D}_{exp}^{test}$, respectively. The required data forms the dataset $\mathcal{D}_{exp}$, which contains turning data on different wind fields, tidal range and robot speeds.

B. Experimental Results

Fig.6 shows sample data collected during the experiment and the main factors influencing the turning radius obtained using PCA. From the turning paths in Fig.6(a) and Fig.6(b), it can be observed that they are not as smooth as in the simulation, due to the influence of environmental disturbances in complex marine environments and sensor noise. By comparing the two examples, it is visually apparent that the length of radius is different. The red arrows indicate that the wind direction is essentially the same in both scenarios, but the arrow lengths indicate varying wind speed magnitudes. The purple region corresponds to the wind speed, robot speed, heading angle and rudder control commands in relation to the blue turning path. Similar to the simulation, the experimental results also show the presence of a stall phase during turning.

The correlation coefficient distribution calculated by PCA

TABLE I

DIFFERENT SCENARIOS OF CALCULATED TURNING RADIUS r^b AND PREDICTED r^{GP}

Test Cases	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
BL	7.48	9.91	9.57	7.09	4.95	10.43	7.54	9.12	6.29	9.26	8.55	5.04	6.31	4.86	6.25	5.90	6.86
GP	7.07	9.65	8.43	7.28	5.42	8.95	8.19	8.20	6.28	8.63	8.04	5.02	5.98	5.27	6.28	5.21	6.97
Error (GP)	0.41	0.26	1.15	0.18	0.47	1.48	0.65	0.93	0.01	0.62	0.50	0.02	0.33	0.41	0.03	0.69	0.11

Test Cases	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
BL	5.13	4.25	5.77	10.39	8.12	8.93	9.08	11.54	7.77	5.74	8.56	6.54	6.65	4.85	5.07	5.53	5.36
GP	5.54	4.84	6.49	9.22	7.57	10.11	8.98	10.01	7.07	6.51	8.11	6.28	6.30	6.17	4.74	6.30	5.29
Error (GP)	0.41	0.60	0.73	1.18	0.56	1.18	0.10	1.53	0.70	0.77	0.45	0.26	0.35	1.32	0.33	0.78	0.07

is shown in Fig.6(c), indicating that wind speed, robot speed and tidal range have a significant influence on the turning radius, consistent with the findings in the simulation.

From Fig.7, the relationship between radius and the other three factors is approximately consistent with the simulation. Specifically, a higher wind speed is associated with a larger turning radius. However, due to the influence of the real marine environment, the radius threshold in the experiment is wider. In our testing field, the flood tide movement is corresponding to the negative of x-axis, which the the same as the turning direction. Hereby, from the perspective of tidal influence, the maximum radius of the flood tide is greater than that of the ebb tide, and the minimum radius of the ebb tide is smaller than that of the flood tide.

C. Discussion

Fig.8 presents the predicted results for the OceanVoy's turning radius across different experimental scenarios. Only four turn examples are depicted here, while more predicted results can be found in Table I.

One noticeable point across the scenarios is that the maximum prediction error exceeds $1m$ compared to the baseline. One possible explanation for this phenomenon is the presence of intricate environmental factors, which can lead to larger radius error of turning.

As shown in Fig.9, it can be observed that the predicted radii obtained from the M^{GP} is similar with the baseline method. Further inspection of Fig.9(b) reveals that the mean absolute error (MAE) for GPR is $0.58m$, while the root mean squared error (RMSE) is $0.71m$.

VI. CONCLUSION

In this study, we propose an estimation scheme for the turning radius of a sailing robot based on Gaussian process regression. This scheme takes into account both the robot's own state and the influence from the surrounding environment. Specifically, factors such as the robot's velocity, wind speed and tides are identified as the main influencing factors and are subjected to in-depth analysis. Our method has been validated through both simulation and experiments. In the future, we will investigate the potential implementation of this method within scenarios like sailboat collision risk assessment and obstacle avoidance.

REFERENCES

[1] D. Sidoti, K. R. Pattipati, and Y. Bar-Shalom, "Minimum time sailing boat path algorithm," *IEEE Journal of Oceanic Engineering*, vol. 48, no. 2, pp. 307–322, 2023.

[2] W. Qi, Q. Sun, X. Ji, Y. Liang, Z. Cao, and H. Qian, "A kelvin wake avoidance scheme for autonomous sailing robots based on orientation-restricted dubins path," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 585–11 592, 2021.

[3] "The microtransat challenge," Available on https://test.microtransat.org/content.php?p=2023&top=2023.

[4] "World robotic sailing competition," Available on https://www.roboticsailing.org/.

[5] W. Qi, Q. Sun, Y. Cao, and H. Qian, "Stable station keeping of autonomous sailing robots via the switched systems approach for ocean observation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 404–11 410.

[6] J. Cabrera-Gámez, J. Isern-González, D. Hernández-Sosa, A. C. Domínguez-Brito, and E. Fernández-Perdomo, "Optimization-based weather routing for sailboats," in *Robotic Sailing 2012: Proceedings of the 5th International Robotic Sailing Conference*. Springer, 2013, pp. 23–33.

[7] T. Akiyama, J.-F. Bousquet, K. Roncin, G. Muirhead, and A. Whidden, "An engineering design approach for the development of an autonomous sailboat to cross the atlantic ocean," *Applied Sciences*, vol. 11, no. 17, p. 8046, 2021.

[8] D. Wrede, J. Adam, and J. Jouffroy, "Online optimization of different objectives in robotic sailing: Simulations and experiments," in *2015 IEEE Conference on Control Applications (CCA)*. IEEE, 2015, pp. 876–881.

[9] Y. Dong, N. Wu, J. Qi, X. Chen, and C. Hua, "Predictive course control and guidance of autonomous unmanned sailboat based on efficient sampled gaussian process," *Journal of Marine Science and Engineering*, vol. 9, no. 12, p. 1420, 2021.

[10] J. M. Prendergast, G. A. Formosa, M. J. Fulton, C. R. Heckman, and M. E. Rentschler, "A real-time state dependent region estimator for autonomous endoscope navigation," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 918–934, 2020.

[11] Q. Sun, W. Qi, H. Liu, Z. Sun, T. L. Lam, and H. Qian, "Oceanvoy: A hybrid energy planning system for autonomous sailboat," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2481–2487.

[12] J. Melin, K. Dahl, and M. Waller, "Modeling and control for an autonomous sailboat: a case study," in *World Robotic Sailing championship and International Robotic Sailing Conference*. Springer, 2015, pp. 137–149.

[13] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[14] L. Jaulin and F. Le Bars, "An interval approach for stability analysis: Application to sailboat robotics," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 282–287, 2012.

[15] Q. Sun, W. Qi, X. Ji, and H. Qian, "V-stability based control for energy-saving towards long range sailing," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 328–335, 2021.

[16] Y. Emam, P. Glotfelter, S. Wilson, G. Notomista, and M. Egerstedt, "Data-driven robust barrier functions for safe, long-term operation," *IEEE transactions on robotics*, vol. 38, no. 3, pp. 1671–1685, 2021.

[17] C. Miles and A. Vladimirsky, "Stochastic optimal control of a sailboat," *IEEE Control Systems Letters*, vol. 6, pp. 2048–2053, 2021.