

Obstacle Tracking on Unmanned Surface Vehicle Using Kalman Filter

Rusdhianto Effendi Abdul Kadir
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ditto@ee.its.ac.id

Mochammad Sahal
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
sahal@ee.its.ac.id

Yusuf Bilfaqih
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
bilfaqih@ee.its.ac.id

Zulkifli Hidayat
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
zulkifli@ee.its.ac.id

Gaung Jagad
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
gaung.jgd@gmail.com

Abstract—Unmanned Surface Vehicles (USV) are self-driving vehicles that operate on the water surface. In order to be operated autonomously, USV has a guidance system designed for path planning to reach its destination. The ability to detect obstacles in its paths is one of the important factors to plan a new path in order to avoid obstacles and reach its destination optimally. This research designed an obstacle tracking system which integrates USV perception sensors such as camera and Light Detection and Ranging (LiDaR) to gain information of the obstacle's relative position in the surrounding environment to the ship. To improve the relative position estimation of the obstacles to the ship, Kalman filter is applied to reduce the measurements noises. The results of the system design are simulated using MATLAB software so that results can be analyzed to see the performance of the system design. Results obtained using the Kalman filter show 12% noise reduction.

Keywords – filter kalman, obstacle tracking, unmanned surface vehicle.

I. INTRODUCTION

The advancement of unmanned systems can be found everywhere today. Commonly known as drones, this technology is available in lots of variety that could use to help with human needs, especially in military development. The variety of unmanned vehicles consist of ground vehicle, aerial vehicle, surface vehicle, underwater vehicle and even spacecraft. The development of unmanned vehicles is so vast because it needs to combine lots of features to be able to operate autonomously.

Unmanned Surface Vehicle (USV) are one of the drones that operates autonomously on the sea surface, commonly used to take data and patrol the area. USV could be operated manually, guided from distance, or operated autonomously. Because USV does not need crew onboard, the ship size is smaller than most ships with better flexibility to move in hazardous marine environments. These benefits improved personnel safety and enabled USV to take harsh and dangerous tasks [1,2,3]. The usage of USV has proved efficient in hydrographic survey which saves lots of time and power.

USV has a guidance system that enables path planning to reach a destination point. One of the important elements for USV to be able to go through the path is to know the obstacle it is facing. USV sensors have the role to detect obstacles within its line of sight. Detected obstacles will be the reference for the

guidance system to plan a new path which must be taken quickly to avoid collisions. For the USV to be able to make a quick decision it needs an estimation method from the measurement data to estimate the position of the obstacle on its path.

In this research, Kalman Filter is designed to minimize the measurement error to improve its performance. Kalman Filter is known to be reliable in finding the best estimation of states by combining several sensors. Camera and LiDaR are the combined sensors used to measure the relative distance and angle of obstacles to the ship. This paper aims to provide increased reliability in tracking obstacle's movement using the Kalman filter applied to the detection system.

II. METHODS

A. Basic Theory

1) Unmanned Surface Vehicle (USV)

USV is a robot in the form of a ship without a crew that could operate in the lake or sea. This unmanned vehicle is low cost and has high mobility. The NGC (Navigation, Guidance, and Control) system in USV supports the ship to operate autonomously by controlling the ship through a planned path to reach the destination while avoiding all obstacle on its path [4,5,6]. Camera, LiDaR, and radars are commonly used sensors in USV to detect obstacles up to hundreds of meters to the ship. USV is integrated with several sensor and controller variations to support its operation without human help, however on certain conditions, the ship could be controlled manually if needed. USV application is widely used in water environment monitoring. From acquiring the water quality information and mapping the ocean environment.

2) LiDaR

LiDaR (light detection and ranging) is a sensor that works like radar imaging. LiDaR as a laser-based sensor uses coherent light, a light that is composed of a very narrow band of wavelengths. Light in general is composed of various wavelengths with specific color. With a collection of various waveforms, a laser can produce light with a narrow range of wavelengths. This light can be transmitted over long distances as a narrow beam with little distortion compared to light in general [7,8].

By using coherent light, LiDaR is able to measure the distance of itself to an object by receiving reflected laser pulses emitted to objects in its line of sight. The time interval after light is emitted until the reflection is received will give information needed to calculate the distance. LiDaR is proved to be an effective identification tool for its surrounding environment.

3) YOLO

You Only Look Once (YOLO) is an object detection approach using a single convolutional neural network. This convolutional neural network will be run on full images in one evaluation. It will predict multiple bounding boxes simultaneously with class probability of the object detected. Bounding box is a common method used to represent the location of an object detected in an image.

YOLO algorithm will give bounding boxes coordinates relative to the whole image. Each bounding box is represented with 4 vector elements and contains a confidence score for the object predicted. It shows the probability of how accurately the bounding box predicts the object detected. Each bounding has these parameters:

- X left: upper left point of bounding box in x-axis
- Y top: upper left point of bounding box in y-axis
- X right: bottom right point of bounding box in x-axis
- Y bottom: bottom right point of bounding box in y-axis

YOLO divides the image into $S \times S$ grid, where each grid predicts B bounding boxes, confidence score for each box and C class probabilities. [9,10,11]. For every grid with the same single object detected will combine to be region proposals. These region proposals with confidence score above threshold will be represented by a bounding box to show the region of object detected on the image. The simplified process of how YOLO works is shown in Figure 1.

Convolutional neural network is the base element which could detect and classify every object detected in the image. Object detection algorithms need lots of training to be able to distinguish and object from its environment. The training result will be kept in the hidden layer as reference to help the network in classifying the object detected.

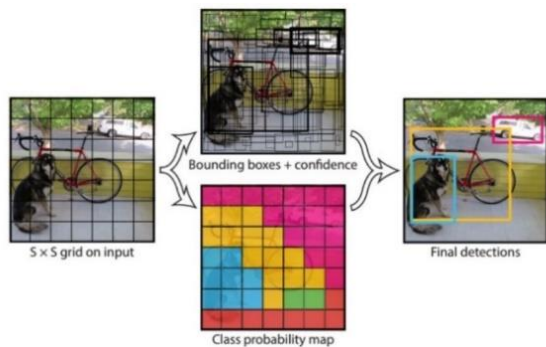


Fig. 1. Simplified illustration of YOLO model [9]

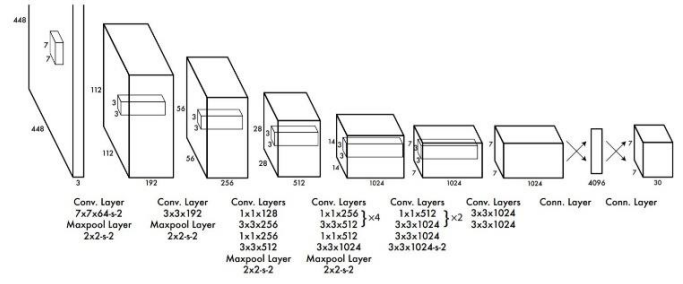


Fig. 2. Convolutional Neural Network in YOLO [9]

Hundreds and thousands of images of an object is needed to train the CNN for the network to have high accuracy of object detection. The object detection Algorithm will only be able to detect objects that have been trained. Untrained objects detected in the image will have a small confidence score and does not show up if it is below the threshold value.

4) Kalman Filter

Kalman Filter is an algorithm that separates a signal from additive noises. With state-space configuration, Kalman filter enables multiple input/multiple output (MIMO) scenarios. The Kalman filter uses least square error which is optimal in minimizing covariance error from its estimation. Kalman Filter needs a system model and measurement model as defined by (1) [12, 13].

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_k \mathbf{x}_k + \mathbf{w}_k$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (1)$$

The first initialization is to assume a temporary estimation notated by $\hat{\mathbf{x}}_k^-$. Then calculate the difference of the estimated state $\hat{\mathbf{x}}_k^-$ and measurement data \mathbf{z}_k to get the optimal estimation by (2).

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (2)$$

Where $\hat{\mathbf{x}}_k$ will be the new and optimal state estimation and \mathbf{K}_k is the Kalman gain acquired from (3). Gain Kalman is needed to get the optimal state estimation. Before going to the next iteration, get the new covariance error matrix by (4) and (5) to have the optimal Kalman gain value for the next iteration.

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (3)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (4)$$

$$\mathbf{P}_{k+1}^- = \boldsymbol{\phi}_k \mathbf{P}_k \boldsymbol{\phi}_k^T + \mathbf{Q}_k \quad (5)$$

B. System Design

1) Image data acquisition

Images that show the USV line of sight are taken by 4 cameras arranged as a sensor that cover 169° angle of view. With the specification each camera has a 46° angle of view. Each camera with its neighboring cameras are set to have a difference of 41° apart from each center. The image size produced by the camera is 640×480 pixels for 4 images, with each image being 320×240 pixels. The structure illustration of the sensor arrangement with camera and LiDaR placed on top of it is shown in Figure 4.

Specifications of images taken from the camera used have 46° angle of view. The sample results of the images taken are shown in Figure 5. Where the camera sequence from right to left is shown from top left to bottom right. The objects that are detected from using the YOLO system are marked with a

number of bounding boxes according to the number of detected objects above the threshold. The sample image output from YOLO is shown in Figure 3.



Fig. 3. Image processing result using YOLO

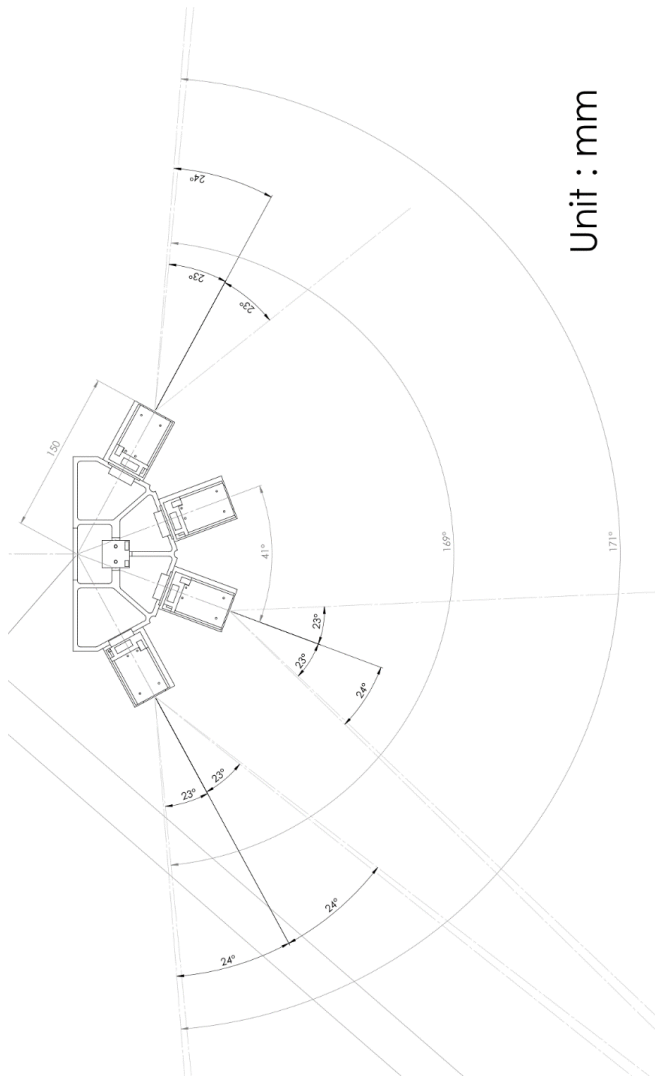


Fig. 4. Camera and LiDaR sensor structure

2) LiDaR Data Reading

The data obtained from LiDaR is processed so that it is obtained in a table with csv format with the distance of object

detected information for each LiDaR divided into 8 beam segments. For each segment it can detect up to 5 obstacles, so it has a table size of 233 columns for 4 LiDaRs.

3) Integrating Camera and LiDaR Data

In the LiDaR data, it is known that the reading of each LiDaR data is divided into 8 segments. So, we divide the area in the image for each segment, so that we can differentiate which object the camera detects is in which segment. Then the minimum distance will be taken in the related segment to be assumed as the relative distance of the object detected to the ship.

By applying this concept, the distance of the object detected by the camera will be known if an object is detected in the LiDaR segment. The coordinates of the detected object in the images are obtained from the bounding boxes. Which is converted into angle data and then used to determine the position of the object to the relevant segment from the LiDaR sensor reading.

It is arranged that the camera sensor structure consists of 4 cameras structured to form a 169° angle of view. With a little intersection at the edges. To get accurate object position results, the angular value for each object detected by each camera will be combined into one, calculated from the center point for all cameras. Suppose the object detected on the 1st camera on the right is illustrated as in Figure 5.

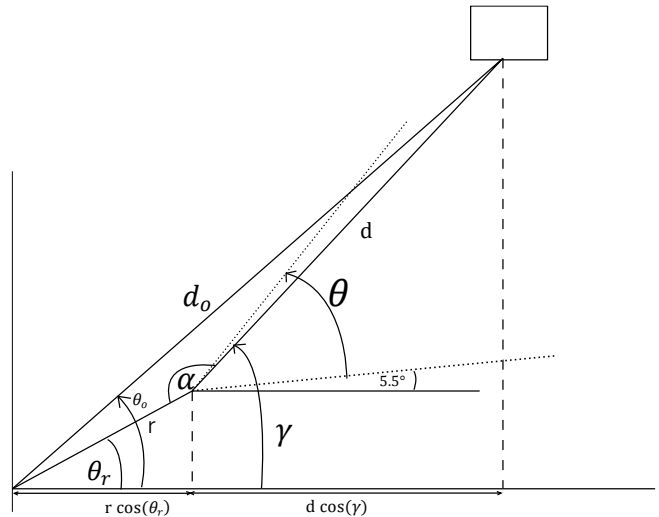


Fig. 5. Distance ang angle transformation illustration

Where d is the distance of the object detected from the camera, d_o is the distance of the object detected to the sensor center point, r is the distance from the sensor center point to the camera's center point, θ is the angle of the object detected the camera, θ_r is the angle from sensor center point to the center point of the first camera. Parameters and variables in the illustration are presented in Table 1.

TABLE I. PARAMETER AND VARIABLE INFORMATION TO CALCULATE DISTANCE AND ANGLE

Symbol	Value	Unit
r	150	mm
θ_r	28.5	deg
θ	46	deg
γ	$\theta + 5.5$	deg
α	$180 - \gamma + \theta_r$	deg

Based on Figure 5, it is acquired the equation to find the distance of the object detected to the center point of sensors by (7). Then to find the angular value of the object detected to the center point of sensors by (9).

$$d_o^2 = r^2 + d^2 - 2 \cdot r \cdot d \cdot \cos(\alpha) \quad (6)$$

$$d_o = \sqrt{r^2 + d^2 - 2 \cdot r \cdot d \cdot \cos(180^\circ - \theta - 5.5^\circ + \theta_r)} \quad (7)$$

$$\cos(\theta_o) = \frac{(r \cdot \cos(\theta_r) + d \cdot \cos(\gamma))}{d_o} \quad (8)$$

$$\theta_o = \cos^{-1} \frac{(r \cdot \cos(\theta_r) + d \cdot \cos(\gamma))}{d_o} \quad (9)$$

The value of γ in (10) and (11) is different for each camera. From right to left, the value of γ are $\theta + 38.5^\circ$, $\theta - 2.5^\circ$, $\theta - 43.5^\circ$, and $\theta - 84.5^\circ$.

4) Obstacle Tracking

The integrated sensors will provide information of obstacle detected position at each timestamp, but it is not known which obstacle detected at the present time is the same object as the obstacle detected at the previous time. So to find out whether the current obstacle is the same object as the previous obstacle, it is necessary to track the object movement based on the object's relative distance and angle displacement to the USV for each object detected at each timestamp. The obstacle's distance displacement is calculated based on the obstacle's distance and angle to the USV by assuming the distance and object angle from the USV at each timestamp is a vector. The displacement distance for each obstacle, can be calculated using the dot product formula in (10)

$$s = \sqrt{d(k)^2 + d(k-1)^2 - 2 \cdot d(k) \cdot d(k-1) \cos(\theta_m)} \quad (10)$$

$$\theta_m = |\theta_o(k) - \theta_o(k-1)| \quad (11)$$

Where s is the displacement of the obstacle's distance, $d(k)$ is the distance of the obstacle relative to the USV at time k , θ_m is the angle formed between the two obstacle distance vectors to the USV, and $\theta_o(k)$ is the angle of obstacle to the USV at time k .

If the distance and angle of of the obstacle displacement is lower than the threshold that has been set, it can be stated that the object at time $t = k$ dan $t = k - 1$ is the same object. The vector illustration is shown in Figure 6.

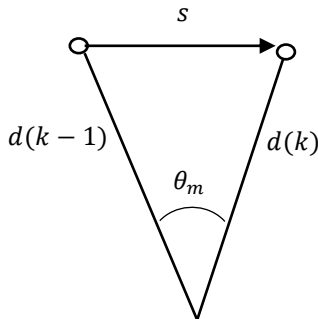


Fig. 6. Obstacle tracking illustration

5) Kalman Filter Design

The concept in using the Kalman filter requires a system model and a measurement model. Therefore, the Kalman Filter design is done by determining the model which is represented

in the form of state-space. The system model and measurement model used in the Kalman Filter application at USV are shown in equation (12).

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \quad (12)$$

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ \Delta t & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ 0 & \Delta t \end{bmatrix} \quad (14)$$

$$\mathbf{x}_k = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} \quad (15)$$

$$\mathbf{u}_k = \begin{bmatrix} a(k) \\ \alpha(k) \end{bmatrix} \quad (16)$$

To run the Kalman Filter, obstacle tracking needs up to three timestamps to have the history information of obstacle distance and angle to the USV from $t = k$ to $t = k - 2$. These measurement data will be used to get initial estimates to obtain velocity, acceleration, angular velocity, and angular acceleration values using equations (17) to (20).

$$x_2(k-1) = \frac{x_1(k-1) - x_1(k-2)}{\Delta t} \quad (17)$$

$$a(k) = \frac{x_1(k) - 2x_1(k-1) + x_1(k-2)}{\Delta t^2} \quad (18)$$

$$x_4(k-1) = \frac{x_3(k-1) - x_3(k-2)}{\Delta t} \quad (19)$$

$$\alpha(k) = \frac{x_3(k) - 2x_3(k-1) + x_3(k-2)}{\Delta t^2} \quad (20)$$

The initial estimated value then can be found at time k . The initial value of the covariance matrix on the Kalman filter will be able to be calculated. So that the Kalman filter algorithm process can be executed properly.

III. RESULTS AND DISCUSSIONS

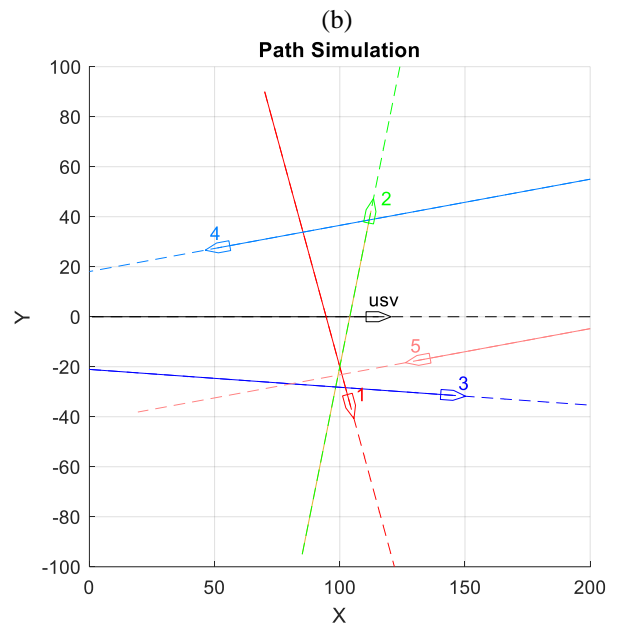
It is decided to take a simulation test because images captured on field test were disturbed by seawater splash on the sensors, resulting in unoptimized obstacle detection because of blurry images. The simulation ran on MATLAB with planned USV and obstacle trajectory. Up to five obstacles are simulated to see the system performance with parameters set beforehand. The obstacles and USV are assumed to have constant velocity in a straight path. Each obstacle velocity is varied to produce different cases. Simulation ran with a variation on covariance \mathbf{Q} and \mathbf{R} value to see the difference in filter performance [14,15]. The 1st simulation ran with covariance \mathbf{Q} value higher than covariance \mathbf{R} . While the 2nd simulation ran with covariance \mathbf{Q} value lower than covariance \mathbf{R} . With covariance \mathbf{Q} and \mathbf{R} used are shown in table 2.

TABLE II. COVARIANCE Q DAN R VALUE

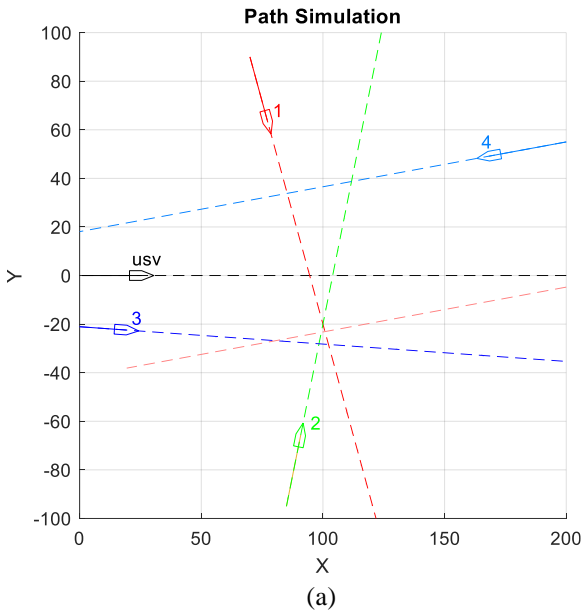
Simulation	Q	R
1st Kalman Filter	$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$
2nd Kalman Filter	$\begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$

In the simulation with five obstacles, some obstacles will be simulated blocking each other. as shown in Figure 8. Plots of the obstacle's distance and angle relative to the ship are shown in from figure 7 to 17.

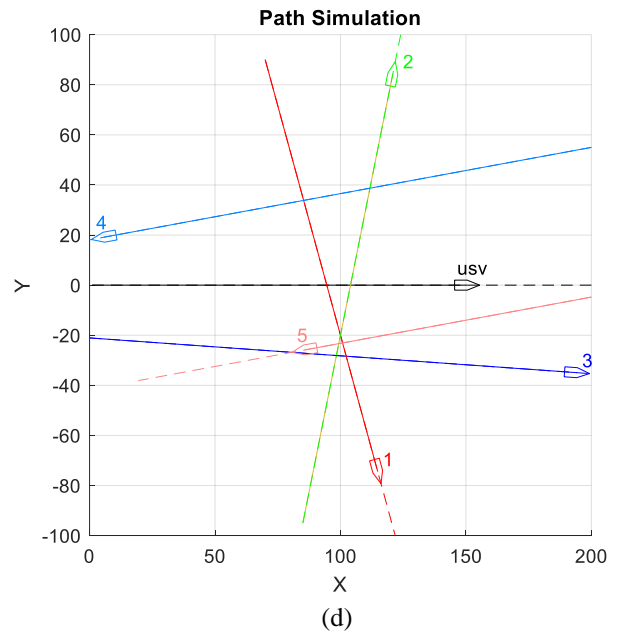
The comparison between Root Mean Square Error (RMSE) of the Kalman filter results and the measurements shows that some of the RMSE of 2nd Kalman filter gives a greater value than the RMSE of the measurement, while the RMSE with 1st Kalman filter gives smaller RMSE value, as shown in table 3.



(b)



(a)



(d)

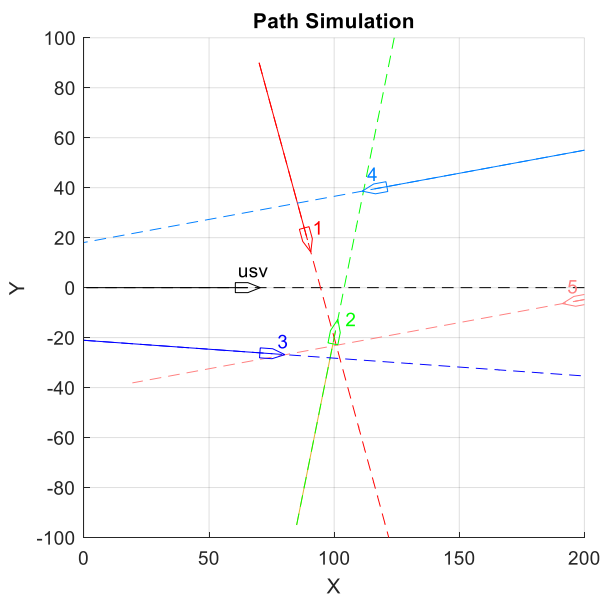


Fig. 7. Path simulation with five obstacles: (a) 5 seconds. (b) 15 seconds. (c) 23 seconds. (d) 30 seconds.

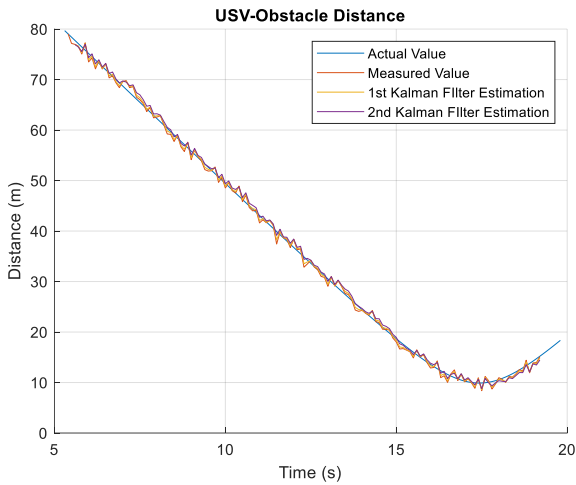


Fig. 8. Relative distance plot of USV to 1st obstacle

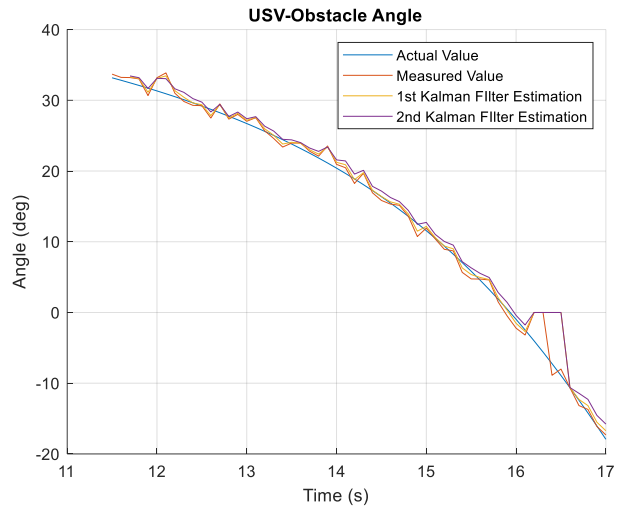


Fig. 11. Relative angle plot of USV to 2nd obstacle

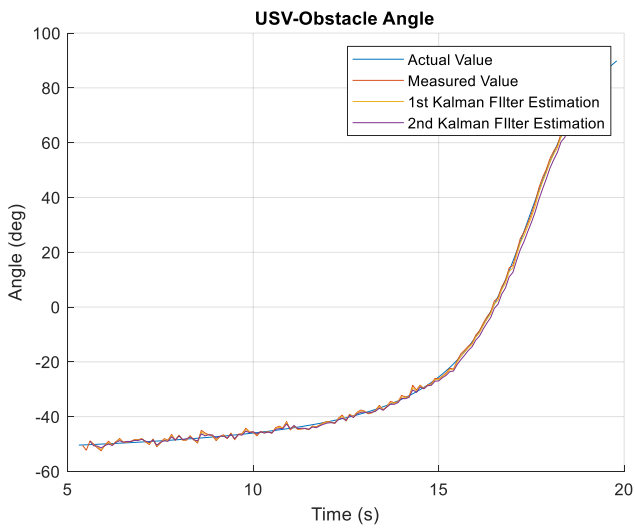


Fig. 9. Relative angle plot of USV to 1st obstacle

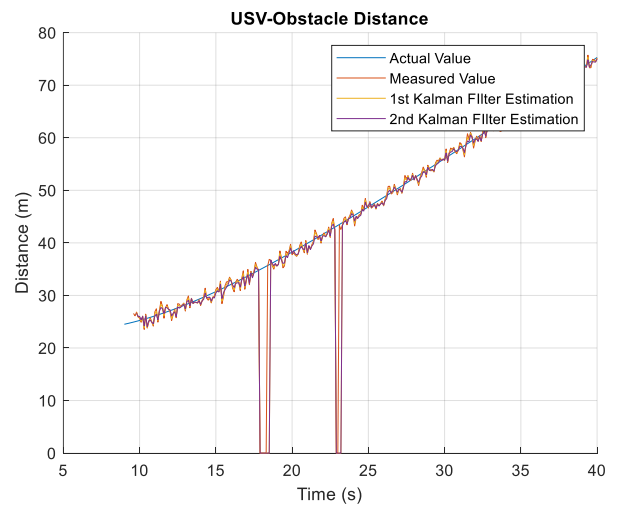


Fig. 12. Relative distance plot of USV to 3rd obstacle

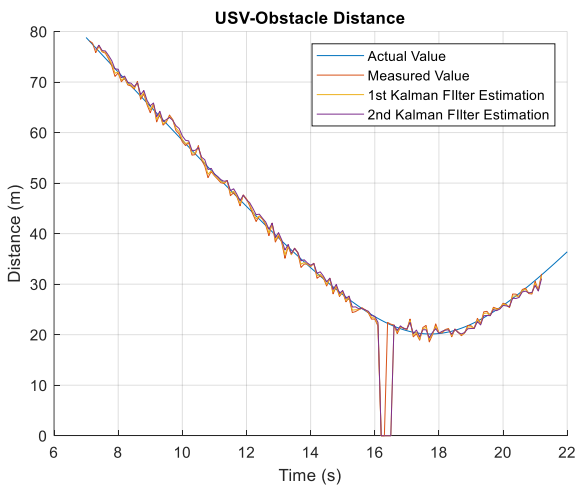


Fig. 10. Relative distance plot of USV to 2nd obstacle

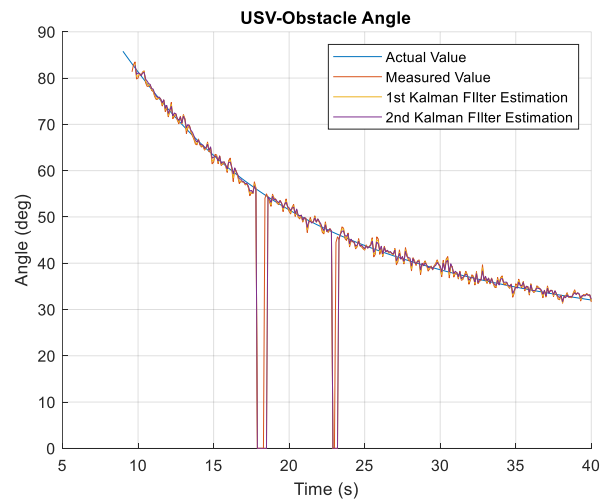


Fig. 13. Relative angle plot of USV to 3rd obstacle

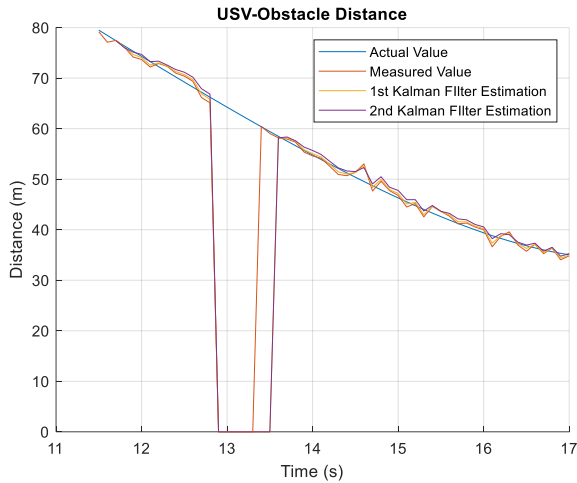


Fig. 14. Relative distance plot of USV to 4th obstacle

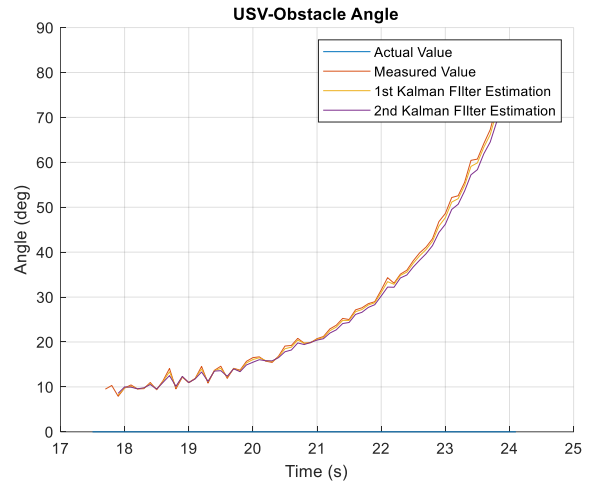


Fig. 17. Relative angle plot of USV to 5th obstacle

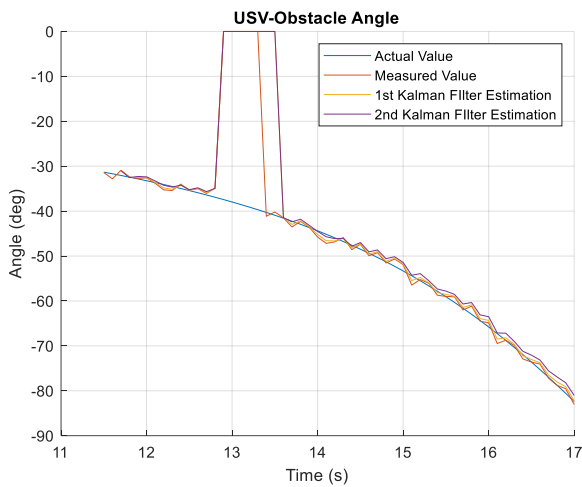


Fig. 15. Relative angle plot of USV to 4th obstacle

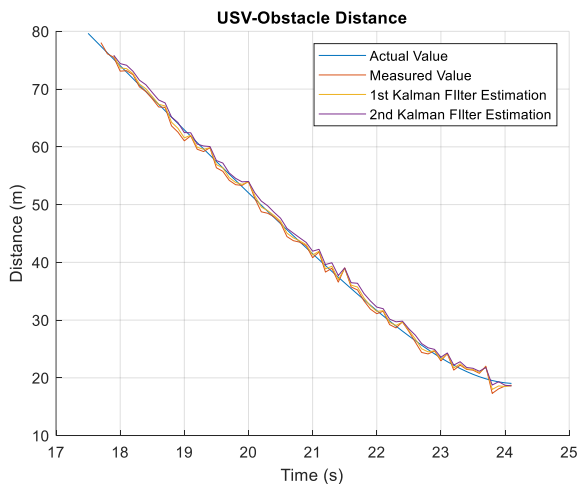


Fig. 16 Relative distance plot of USV to 5th obstacle

TABLE III. FILTER KALMAN RESULTS RMSE COMPARISON

Obstacle		Measurement RMSE	1st Kalman Filter RMSE	2nd Kalman Filter RMSE
1st obstacle	Distance	0.9356	0.8122	0.8290
	Angle	1.0042	0.9652	1.5251
2nd obstacle	Distance	1.0297	0.8885	0.8666
	Angle	0.9833	0.9205	1.2379
3rd obstacle	Distance	1.0984	0.8508	0.6961
	Angle	1.0326	0.8838	0.7340
4th obstacle	Distance	1.0694	0.9730	1.1124
	Angle	0.9904	0.9144	1.2416
5th obstacle	Distance	0.9745	0.8273	0.9908
	Angle	1.0418	0.8868	1.3513

IV. CONCLUSIONS

From the results of the obstacle tracking simulation using Kalman filter, the following conclusions can be taken:

1. The obstacle tracking algorithm with integrated camera sensor data and LiDaR is running well. This is shown by the obstacles that are still detected as the same object even though there are intersections between obstacles.
2. Estimation of relative distance and angle of obstacle detected to ship using the Kalman filter shows a smaller error value, although it is still closer to the measurement value than the actual value.
3. The Kalman filter estimation results give a better estimated value when the covariance Q value is greater than covariance R value. It is shown by the estimation value which is closer to the true value than the measurement value. Because the covariance Q will affect the change in the P covariance, while the relationship between the P covariance and the R covariance will affect the Kalman gain value in each iteration. This can be seen from the comparison of the Kalman RMSE filter.
4. Sensor integration application on the field is still facing problems due to seawater splash obstructing the camera vision, resulting in blurry images processed by YOLO.

REFERENCES

[1] L. Zhixiang, "Unmanned surface vehicles: An overview of developments and challenges," 2016, pp. 71-93.
 [2] Rj. Yan, " Development and missions of unmanned surface vehicle," 2010.

- [3] K. Tanakitkorn, "A review of unmanned surface vehicle development," 2019.
- [4] M. Hossein, "Developing a navigation, guidance and obstacle avoidance algorithm for an Unmanned Surface Vehicle (USV) by algorithms fusion," 2018, pp. 56-65.
- [5] W. Zhan, "Autonomous Visual Perception for Unmanned Surface Vehicle Navigation in an Unknown Environment," 2019.
- [6] M. Caccia, "Basic navigation, guidance and control of an Unmanned Surface Vehicle," 2008.
- [7] C. James B, "Introduction to remote sensing," 2011, pp. 243-244.
- [8] J. Mckean, " Remote Sensing of Channels and Riparian Zones with a Narrow-Beam Aquatic-Terrestrial LIDAR," 2009.
- [9] R. Joseph, "You Only Look Once: Unified, Real-Time Object Detection," 2015.
- [10] R. Joseph, "YOLO9000: Better, Faster, Stronger," 2016
- [11] R. Joseph, "YOLOv3: An Incremental Improvement," 2018.
- [12] G. Welch, G. Bishop, "An introduction to the Kalman filter," 2000.
- [13] B. Robert G. "Introduction to Random Signal Analysis and Kalman Filtering," 2012.
- [14] Y. Singh, R. Mehra. "Relative Study of Measurement Noise Covariance R and Process Noise Covariance Q of the Kalman Filter in Estimation," 2015.
- [15] Q. Ge, T. Shao, Z. Duan, and C. Wen, "Performance analysis of the Kalman filter with mismatched noise covariances," 2016.