

Mario Kučić

Infobip, Office: Rijeka, Martinkovac 109, 51000 Rijeka, Croatia

E-mail: mario.kucic@infobip.com

Marko Valčić

University of Rijeka, Faculty of Engineering, Vukovarska 58, 51000 Rijeka, Croatia

E-mail: mvalcic@riteh.hr

Stereo Visual Odometry for Indoor Localization of Ship Model

Abstract

Typically, ships are designed for open sea navigation and thus research of autonomous ships is mostly done for that particular area. This paper explores the possibility of using low-cost sensors for localization inside the small navigation area. The localization system is based on the technology used for developing autonomous cars. The main part of the system is visual odometry using stereo cameras fused with Inertial Measurement Unit (IMU) data coupled with Kalman and particle filters to get decimetre level accuracy inside a basin for different surface conditions. The visual odometry uses cropped frames for stereo cameras and Good features to track algorithm for extracting features to get depths for each feature that is used for estimation of ship model movement. Experimental results showed that the proposed system could localize itself within a decimetre accuracy implying that there is a real possibility for ships in using visual odometry for autonomous navigation on narrow waterways, which can have a significant impact on future transportation.

Key words: computer vision, estimation, Kalman filter, particle filter, visual odometry

1. Introduction

Autonomous vehicles could have a large role in the future transportation of people and goods. Trucks that are capable of driving the whole day without stop can reduce the cost of the goods [14]. Personal autonomous cars can reduce city traffic using intelligent transportation systems [18]. Autonomous ships can be used to monitor traffic in the ports and coastal navigation [15]. There is also a potential for boats to transport people in urban cities like Venice and Amsterdam, where are many waterways, which can consequently reduce the traffic on the roads. *Roboat* project is already exploring the possibility of transporting people in Amsterdam, but there are many challenging

problems to be solved [21]. Many of them are related to accurate localization. Most of the autonomous boats today use Inertial Measurement Unit (IMU) and Global Positioning System (GPS) fused with unscented Kalman filter (UKF) or extended Kalman filter (EKF) for meter or decimetre level accuracy of the position [1][3][4][6][15]. Such solutions are designed for open seas or coastal navigation where GPS can provide accurate positioning. The same applies to other Global Navigation Satellite Systems (GNSS) like GLONASS, Galileo, BeiDou, etc.

The problem with localization within a canal is low accuracy of GNSS position due to buildings or trees covering the sky, which can be a difficulty for road vehicles as well [21]. One possible solution is using LiDAR (Light Detection and Ranging) method fused with IMU, camera, and GNSS to get decimetre-level accuracy [23]. This is done in the field of self-driving car research [23], and it is also tried in the channel navigation in Amsterdam [21]. One problem with LiDAR approaches is the price of the system. Some research are indicating that same accuracy can be achieved using only cameras [20]. Using only one camera it is possible to make accurate localization [5][13], but much robust solution is to use multiple cameras, preferably one stereo camera set up to get a robust estimate [20].

Most of the other approaches are using some learning method, which most often needs accurate labels and large amount of data. On the other hand, the proposed system in this paper does not need any kind of learning, i.e. it uses visual odometry for extracting motion from stereo camera setup and fusion with IMU using Kalman filter (KF) and particle filter (PF) in order to get very accurate estimations of ship model movements in a basin.

2. Hardware overview

The ship model used in this research does not have a propulsion system. Instead, it was pulled using a rope, which reduces the complexity and price of the ship model but also reduces the vibrations and realism in conducted experiments. The main microprocessor is the *Raspberry Pi 3B+*, which is not capable of the real-time estimation, but instead, it is used for collecting the data on a USB drive. The model is equipped with two *PlayStation eye* cameras to provide visual odometry for KF and PF localization. An IMU *MPU9250* is installed on the model's principal axes to monitor the magnetic fields in three-axis with a sampling frequency of 100 Hz. Additional camera, *Logitech C920*, is placed directly above the basin. It is connected to PC and used as a reference system in evaluating estimations. The assembled and equipped ship model used in in this research is shown in Figure 1.



Figure 1 – The ship model

3. Computer vision algorithms for estimation of ship model motions

This section describes stereo visual odometry combined with Kalman and particle filters, as well as the algorithm for finding and tracking true position and velocities of the ship model.

3.1 Visual odometry for ship model

The software workflow is given in Figure 2. The main focus is set on the estimating the accurate position and velocities.

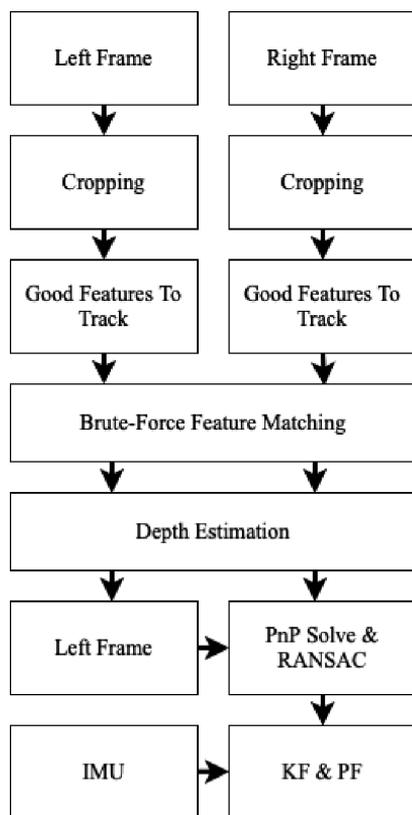


Figure 2 – Software workflow

The model's cameras are calibrated using the Levenberg-Marquardt algorithm [17] implemented in *OpenCV* library [2]. The frames from the cameras are synchronized at 20 Hz with a resolution of 640x480 pixels. To reduce the noise in the further estimation, water is removed from the frames by cropping it to include only upper two-thirds. Extraction of features in the frames is done using *Good Features to Tack* algorithm [19] implemented in *OpenCV*, which gives distribute features across the whole frame. Features are distinguished parts in a frame, most often edges of objects. Extraction is done on the left and right frames in parallel. The result is a set of key-points \mathbf{p} , where each key-point has pair of coordinates (x, y) in the frame. Feature matching is done using brute-force search where each key-point from the left frame is compared with all key-points in the right frame using Hamming distance [2], and vice versa (cross-checking). Only features that have distance smaller than some threshold value are matched points \mathbf{p} . All matched points are in 2D space, depth of each p is estimated according to [16] as

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \frac{x_l - x_r + c_{xl} - c_{xr}}{b} \begin{bmatrix} -c_{xl} \\ -c_{yl} \\ c_{xl} - c_{xr} + f_l \end{bmatrix} \quad (1)$$

where x_l and x_r are the pixel coordinates of a match in left and right frames respectively, b is the physical distance between two cameras, c_{xl} and c_{yl} are the pixels of the real centre of the left camera sensor, respectively, and f_l is the focus of the left camera [2].

Given 3D positions of objects in previous frame and current frame, it is possible to find translation \mathbf{t}_k and rotation \mathbf{R}_k using perspective-n-point (PnP) solver methods [12]. Matrix \mathbf{R}_k is a rotation matrix from which it is possible to extract Euler angles φ , θ , and ψ for roll, pitch and yaw, respectively. Vector \mathbf{t}_k corresponds to velocities in body reference frame as shown in [16].

The PnP is solved using *OpenCV* function *solvePnP*. PnP solver can produce large errors in some cases; therefore, it is used in combination with random sample consensus (RANSAC) method [7] to reduce probability of larger errors. PnP solver returns \mathbf{R}_k and \mathbf{t}_k in-camera reference frame. To track the position of the model it is necessary to convert those measurements into some global, in this case, North-East-Down (NED) reference frame [8]. This can be done by rotating measured vectors by the heading angle, which is defined as an angle between the true North and direction in which the vessel is pointing. The heading angle is given by magnetometer and by tracking rotations between frames. Rotated \mathbf{t}_k provides velocities in the NED, \mathbf{t}_{NED} , which then allows estimating the position of the model. Kalman [22] and particle [10] filters are used to improve the estimation of heading, velocities \mathbf{t}_{NED} , and position in the NED reference frame.

3.1.1 Kalman filter

The Kalman filter (KF) is divided into 2 parts, prediction and update [22]. Prediction uses prior estimated state $\hat{\mathbf{x}}_{k+1}^-$ and state transition matrix Φ in order to predict new state

$$\hat{\mathbf{x}}_{k+1}^- = \Phi \hat{\mathbf{x}}_k. \quad (2)$$

Because the prediction depends on the prior state and the state transition matrix, it is expected to have some uncertainty in the estimation. Uncertainty of the estimation is tracked with the error covariance matrix \mathbf{P} . The \mathbf{P} is updated using process noise covariance matrix \mathbf{Q} as

$$\mathbf{P}_{k+1}^- = \Phi \mathbf{P}_k \Phi^T + \mathbf{Q}. \quad (3)$$

The update step of the KF is done when there are new measurements \mathbf{z}_k available [11]. Based on the \mathbf{P}_{k+1}^- and measurement noise covariance \mathbf{R} , Kalman gain \mathbf{K}_k is

calculated as

$$\mathbf{K}_k = \frac{\mathbf{P}_{k+1}^- \mathbf{H}}{\mathbf{H} \mathbf{P}_{k+1}^- \mathbf{H}^T + \mathbf{R}}. \quad (4)$$

According to [11], \mathbf{K}_k determines how much belief is given to estimation in comparison to a measurement, and \mathbf{H} is adjustment matrix. The update of the state $\hat{\mathbf{x}}_{k+1}^-$ is given with term

$$\hat{\mathbf{x}}_k = \Phi \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_{k+1}^-). \quad (5)$$

The final step is to update the matrix \mathbf{P}_k as

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k+1}^-, \quad (6)$$

where \mathbf{I} is an identity matrix.

3.1.2 Particle Filter

The particle filter (PF) is initialized with N particles with different states sampled from a normal distribution. PF performs prediction and update similarly as KF [11]. The prediction is done for each particle as

$$\hat{\mathbf{x}}_{k+1}^- = \Phi \hat{\mathbf{x}}_k + \omega, \quad (7)$$

where ω is random noise sampled from a normal distribution. According to [11], the first step of update is finding error between measurements \mathbf{z}_k and state $\hat{\mathbf{x}}_{k+1}^-$ for each particle as

$$\epsilon_k = \text{Avg} \left(\frac{1}{\sqrt{2\pi} |\alpha|} \exp \left(\frac{-(\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_{k+1}^-)^2}{2 |\alpha|} \right) \right), \quad (8)$$

where α is the random noise sampled from a normal distribution, and Avg indicates the mean value.

The errors ϵ_k are then normalized to get weight w_k of each particle as

$$w_k = \frac{\epsilon_k}{\sum_{n=0}^{N-1} \epsilon_{kn}}. \quad (9)$$

In the re-sampling, the weight of a particle represents the probability of choosing the particle to be in a new set of particles. The other use of the weights is in estimating the state which is taken from [11] as a weighted average of all particles

$$\bar{\mathbf{x}}_k = \frac{\sum_{n=0}^{N-1} w_{kn} \mathbf{x}_{kn}}{\sum_{n=0}^{N-1} w_{kn}}. \quad (10)$$

3.2 Evaluation algorithm

The setup is very similar to [9] where the author used an optical camera to track markings on the ship model. The same approach is used in this work. The camera is mounted directly above the basin and is perpendicular to the water surface. It records at 20 FPS with a resolution of 1280x720. Each recorded frame is processed independently, which yields a unique position for the red and the green marker that are on the ship model. Extraction of the positions begins with converting blue green red (BGR) format into hue saturated value (HSV), followed by blurring the whole frame using Gaussian blur [2]. Each pixel in the blurred image is then checked if it is in the target range of colours for red or green. Erosion and dilation are used to remove some noise from the processed frame [2]. The final step is finding positions of the largest group of valid pixels in the processed frame by using counter finding method described in [2]. Results are the 2D position of the green and red markers in the frame. Heading of the ship model is given as

$$\psi = \text{atan}\left(\frac{y_{red} - y_{green}}{x_{red} - x_{green}}\right) \quad (11)$$

where x_{red} and y_{red} present the position of the red marker, while x_{green} and y_{green} present the position of the green marker. Function *atan* should be used as *atan2* in order to avoid possible singularities. Position of the ship is given as

$$x = x_{green} + 0.5(x_{red} - x_{green}) \quad (12)$$

and

$$y = y_{green} + 0.5(y_{red} - y_{green}). \quad (13)$$

Velocities of the ship model in NED reference frame, if the camera has y edge of the frame aligned with true North, are

$$\Delta x = x_k - x_{k-1} \quad (14)$$

and

$$\Delta y = y_k - y_{k-1}. \quad (15)$$

where k denotes the time step.

From [8], velocities in the body-reference frame are equal to

$$u = U \cos(\beta) \quad (16)$$

and

$$v = U \sin(\beta), \quad (17)$$

where U is the resultant velocity that can be also expressed as

$$U = \sqrt{\Delta x^2 + \Delta y^2}. \quad (18)$$

Drift angle, i.e. the difference between the heading and course angles, can be determined as

$$\beta = \psi - \text{atan} \left(\frac{y_k - y_{k-1}}{x_k - x_{k-1}} \right), \quad (19)$$

although it should be noted that given angles are derived from a mathematical and computational point of view, while marine navigation uses slightly different references for the angles of 0 rad. Thus, to correct the angles, the following function

$$f(\xi) = \left(\frac{5\pi}{2} - \xi \right) \text{mod } 2\pi \quad (20)$$

was used, where ξ is the value in radians and *mod* is modulo operator.

4. Experiments and results

In this section, performed indoor experiments and associated results to validate the effectiveness of the described algorithm structure are presented. The experiments were conducted in the basin of the Laboratory of Ship Hydromechanics, Department of Naval Architecture and Ocean Engineering, Faculty of Engineering, University of Rijeka, Rijeka, Croatia. The basin has the capability of generating linear waves with constant wavelength and amplitude. The shore, which has the main purpose of breaking the waves without their reflection, is shown in Figure 3 (left). It is installed 3 m from the wave generator that is placed on the opposite side, as also shown in Figure 3 (right).



Figure 3 – The shore (left) and the wave generator (right)

Experiments were conducted with two distinct wave angles of attack γ and with multiple wave amplitudes ζ_a and wavelengths λ , as shown in Table 1. The wave angle of attack γ is defined with respect to the positive part of the x -axis of the body reference frame in counter-clockwise direction.

Table 1 – Angle of attack, amplitude and wavelength for each experiment

Exp.	γ [°]	ζ_a [mm]	λ [mm]
1	-	0	-
2	40	5	210
3	40	7.5	132
4	40	11	59
5	0	5	210
6	0	7.5	132
7	0	11	59

Each experiment is evaluated using KF and PF to estimate positions N and E , velocities u and v in surge and sway, and heading ψ . All obtained estimates were compared with the reference data. Performance indexes used for this comparison were root mean square error (RMSE) and standard deviation (std). The estimates that were obtained using KF are given in Table 2 and the estimates that were obtained using PF are given in Table 3. It should be noted that the values outside parentheses are RMSE values, while those in parentheses present associated standard deviations.

Table 2 – Estimation results based on Kalman filter

Exp.	N [cm]	E [cm]	u [cm/s]	v [cm/s]	ψ [°]
1	1.097 (1.213)	1.608 (1.408)	0.078 (0.103)	0.011 (0.014)	1.387 (1.643)
2	2.232 (2.411)	2.302 (2.193)	0.155 (0.187)	0.044 (0.046)	1.370 (1.611)
3	4.087 (3.671)	4.051 (3.321)	0.269 (0.325)	0.074 (0.073)	1.485 (2.342)
4	6.043 (5.015)	5.116 (4.914)	0.284 (0.333)	0.058 (0.067)	2.170 (3.041)
5	0.761 (1.065)	3.408 (3.870)	0.147 (0.148)	0.021 (0.025)	1.042 (1.318)
6	0.296 (0.293)	3.934 (4.055)	0.110 (0.139)	0.036 (0.042)	1.551 (1.822)
7	0.320 (0.324)	4.340 (4.843)	0.088 (0.091)	0.044 (0.045)	1.531 (1.833)

Table 3 – Estimation results based on particle filter

Exp.	N [cm]	E [cm]	u [cm/s]	v [cm/s]	ψ [°]
1	1.054 (1.125)	1.630 (1.441)	0.072 (0.087)	0.013 (0.017)	1.400 (1.784)
2	2.873 (3.022)	2.881 (2.993)	0.154 (0.185)	0.043 (0.046)	1.399 (1.635)
3	4.189 (3.774)	4.155 (3.391)	0.273 (0.333)	0.076 (0.074)	1.447 (2.039)
4	6.265 (5.180)	5.126 (4.918)	0.255 (0.281)	0.066 (0.074)	2.277 (3.248)
5	0.801 (1.121)	3.393 (3.847)	0.139 (0.142)	0.022 (0.025)	1.013 (1.207)
6	0.304 (0.302)	3.757 (3.834)	0.117 (0.133)	0.038 (0.045)	1.580 (1.801)
7	0.298 (0.329)	4.409 (4.945)	0.093 (0.100)	0.045 (0.046)	1.582 (1.921)

From the obtained results, it can be seen that larger amplitudes and shorter wavelengths cause larger RMSE. In the first experiment, i.e. without waves, accuracy of the ship position is within few centimetres using both KF and PF. In other cases, for example in the fourth experiment, RMSE gets up to 6.265 cm with standard deviation of 5.180 cm. Velocities and heading angle follow the same trend. It is also visible that KF has smaller RMSE values in comparison with PF for almost all experiments. The reason for this could be in the fact that PF was implemented with 1000 particles, and it is possible that with more particles PF would perform better, but with the cost of the processing speed and computational time.

The errors within the positions are directly related with errors in velocities and heading. Therefore, to reduce positioning errors, velocities and heading errors must be reduced. One possible solution related to velocities is using more than one consecutive frame for estimation. This would most likely reduce the high-frequency noise that is observable in visual representation of the results, which can be found in Appendix A.

From these visual results, it is visible that motion of the ship model does not behave linearly. This issue is related to the state transition matrices in KF and PF that do not include nonlinear behaviour. Other issue can be visible in distribution of

residuals. It is obvious that majority of distributions cannot be classified as Gaussian, i.e. majority of residuals cannot be considered normally distributed. Thus, KF and implemented PF have wrong assumptions about the noise in the system. Moreover, KF generally assumes normal distribution of the noise, while PF usually can handle other non-Gaussian distributions, but in this implementation, normal distribution is also assumed.

5. Conclusions

This paper presents an algorithmic structure for localization of a ship model in a basin using only stereo cameras and IMU. The proposed system works well within a controlled and moderate environment, but it has some limitations in general which could be improved. Ship model does not have a propulsion system, which means less vibration of the hull and thus less noise in the system. Therefore, the next iteration of the ship model should include a propulsion system. With included propulsion, the algorithm should also be run in real-time. From the visual odometry perspective, the main limitation is not using more than one consecutive frame for the estimation. From the mathematical model perspective, the state transition matrix should include hydrodynamic properties of the ship model into the motion equations, as well as including the mathematical model of waves. Following those changes, KF is not sufficient for estimation in nonlinear environment, therefore it should be replaced with improved PF or some other variation of KF such as extended Kalman filter. Improvements on the PF side should be on modelling noise distributions so it can handle other distributions as well.

Acknowledgment

This work has been partially supported by the Croatian Science Foundation under the project IP-2018-01-3739 and by the University of Rijeka under the project numbers uniri-tehnic-18-18 and uniri-tehnic-18-266.

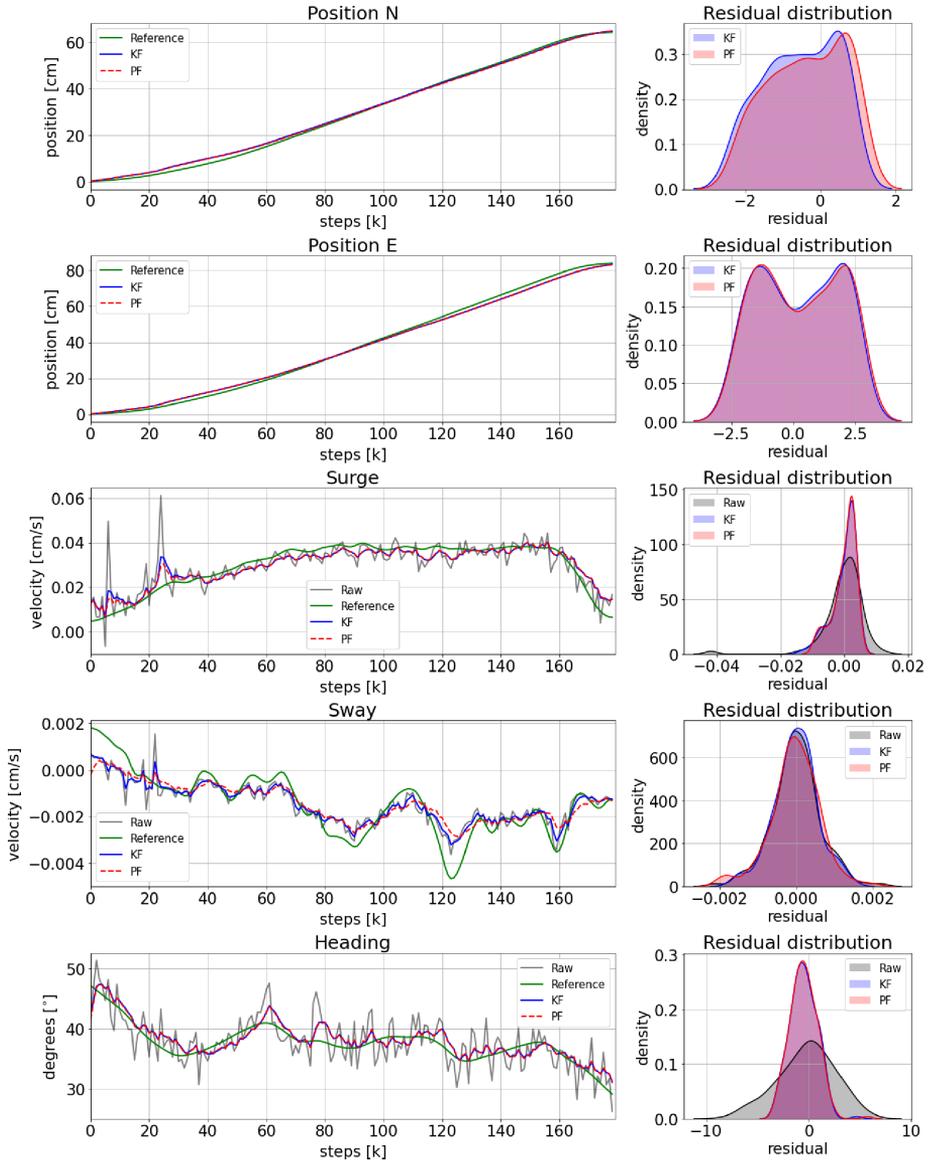
References

1. Azzeri, M., Adnan, F. & Zain, M. (2015) Review of course keeping control system for unmanned surface vehicle. *Jurnal Teknologi*. 74 (5), 11-20.
2. Bradski, G. & Kaehler, A. (2016) *Learning OpenCV 3, Computer Vision in C++ with the OpenCV Library*. Sebastopol, CA, USA, O'Reilly Media.
3. Corke, P., Detweiler, C., Dunbabin, M., Hamilton, M., Rus, D. & Vasilescu, I. (2007) Experiments with underwater robot localization and tracking. In: Proceedings of the 2007 IEEE International Conference on Robotics and Automation, 10-14 April 2007, Roma, Italy. IEEE. pp. 4556-4561.
4. Curcio, J., Leonard, J. & Patrikalakis, A. (2005) Scout-a low cost autonomous surface platform for research in cooperative autonomy. In: Proceedings of OCEANS 2005 MTS/IEEE, 17-23 September 2005, Washington, DC, USA. IEEE. pp. 725-729.

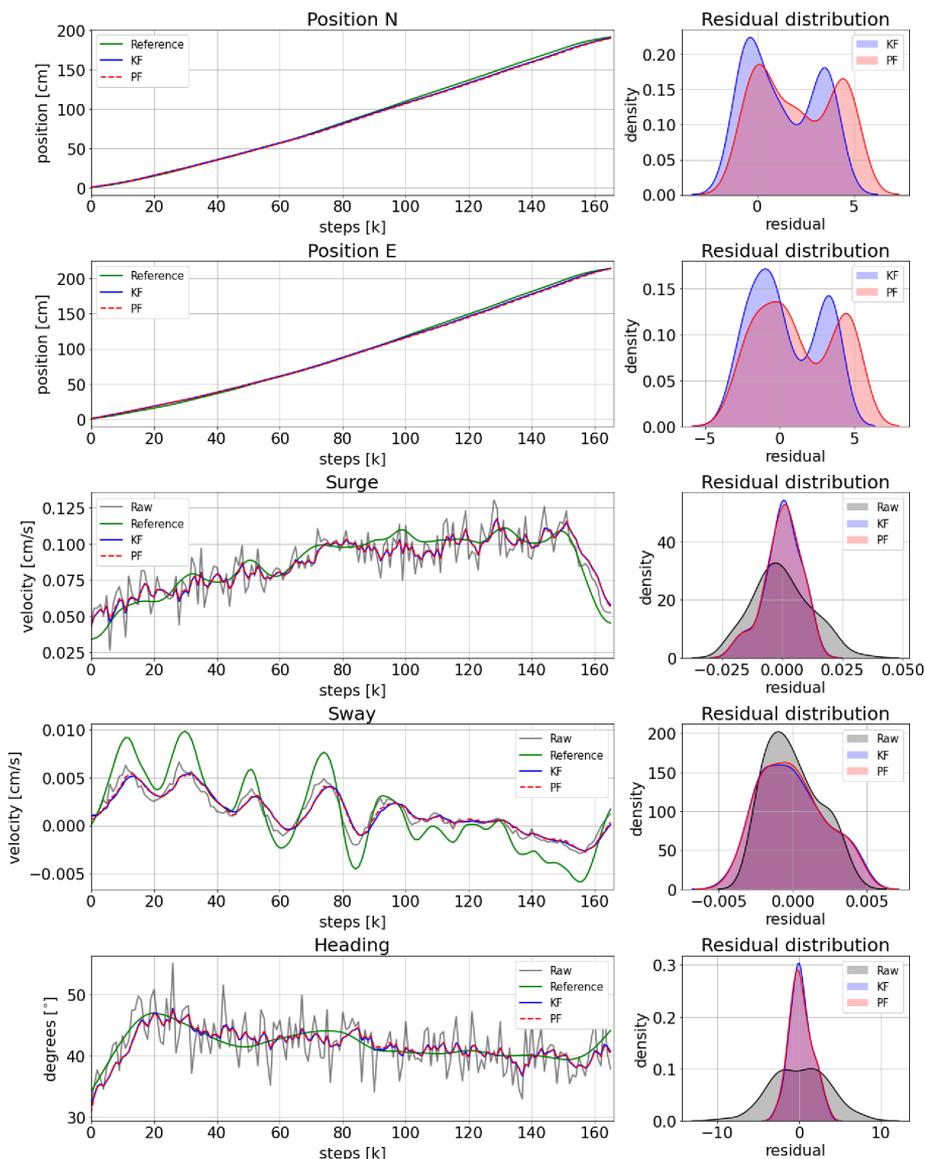
5. Davison, A. J., Reid, I. D., Molton, N. D. & Stasse, O. (2007) MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on pattern analysis and machine intelligence*. 29 (6), 1052-1067.
6. Dhariwal A. & Sukhatme, G. S. (2007) Experiments in robotic boat localization. In: Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 29 October - 2 November 2007, San Diego, CA, USA. IEEE. pp. 1702-1708.
7. Fischler, M. A. & Bolles, R. C. (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. 24 (6), 381-395.
8. Fossen, T. I. (2011) *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, England, John Wiley & Sons Ltd.
9. Koljesnikov, I. (2017) *Position and orientation estimation of radio-controlled vessel*. M.Sc. thesis. University of Rijeka, Faculty of Maritime Studies, Rijeka, Croatia. (in Croatian)
10. Kwok, N. M., Fang, G. & Zhou, W. (2005) Evolutionary particle filter: Resampling from the genetic algorithm perspective. In: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2-6 August 2005, Edmonton, Canada. IEEE. 2935-2940.
11. Labbe, R. R. (2018) *Kalman and Bayesian Filters in Python*. [Online]. Available at: <https://github.com/rllabbe/Kalman-and-Bayesian-Filters-in-Python> [Accessed 8th October 2019].
12. Li, S., Xu, C. & Xie, M. (2012) A Robust O(n) Solution to the Perspective-n-Point Problem. *IEEE Transactions on pattern analysis and machine intelligence*. 34 (7), 1444-1450.
13. Li, Y., Xie, C., Lu, H., Chen, X., Xiao, J. & Zhang, H. (2018) Scale-aware monocular SLAM based on convolutional neural network. In: Proceedings of the 2018 IEEE International Conference on Information and Automation (ICIA), 11-13 August 2018, Wuyishan, China. IEEE. pp. 51-56.
14. Lima, P. F., Trincavelli, M., Nilsson, M., Mårtensson, J. & Wahlberg, B. (2016) Experimental evaluation of economic model predictive control for an autonomous truck. In: Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), 19-22 June 2016, Gothenburg, Sweden. IEEE. pp. 710-715.
15. Liu, Z., Zhang, Y., Yu, X. & Yuan, C. (2016) Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*. 41, 71-93.
16. Meira, G. T. (2016) *Stereo Vision-based Autonomous Vehicle Navigation*. M.Sc. thesis. Worcester Polytechnic Institute (WPI), Worcester, MA, USA.
17. Moré, J. J. (1978) The Levenberg-Marquardt algorithm: Implementation and theory. In: Watson, G. A. (eds) *Numerical Analysis. Lecture Notes in Mathematics*, vol. 630. Berlin, Heidelberg, Germany, Springer, pp. 105-116.
18. Nkoro A. & Vershinin Y. A. (2014) Current and future trends in applications of intelligent transport systems on cars and infrastructure. In: Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 8-11 October 2014, Qingdao, China. IEEE. pp. 514-519.
19. Shi, J. & Tomasi, C. (1994) Good features to track. In: 1994 Proceedings of IEEE conference on computer vision and pattern recognition, 21-23 June 1994, Seattle, WA, USA. IEEE. pp. 593-600.
20. Wang, R., Schworer, M. & Cremers, D. (2017) Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. In: Proceedings of the IEEE International Conference on Computer Vision, 22-29 October 2017, Venice, Italy. IEEE. pp. 3903-3911.
21. Wang, W., Gheneti, B., Mateos, L., Duarte, F., Ratti, C. & Rus, D. (2019) Roboat: An autonomous surface vehicle for urban waterways. In: Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3-8 November 2019, Macau, China. IEEE. pp. 6340-6347.
22. Welch, G. & Bishop, G. (2006) *An Introduction to the Kalman Filter*. [Online]. Available at: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf [Accessed 6th October 2019].
23. Wolcott R. W. & Eustice R. M. (2014) Visual localization within Lidar maps for automated urban driving. In: Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 14-18 September 2014, Chicago, IL, USA. IEEE. pp. 176-183.

APPENDIX A

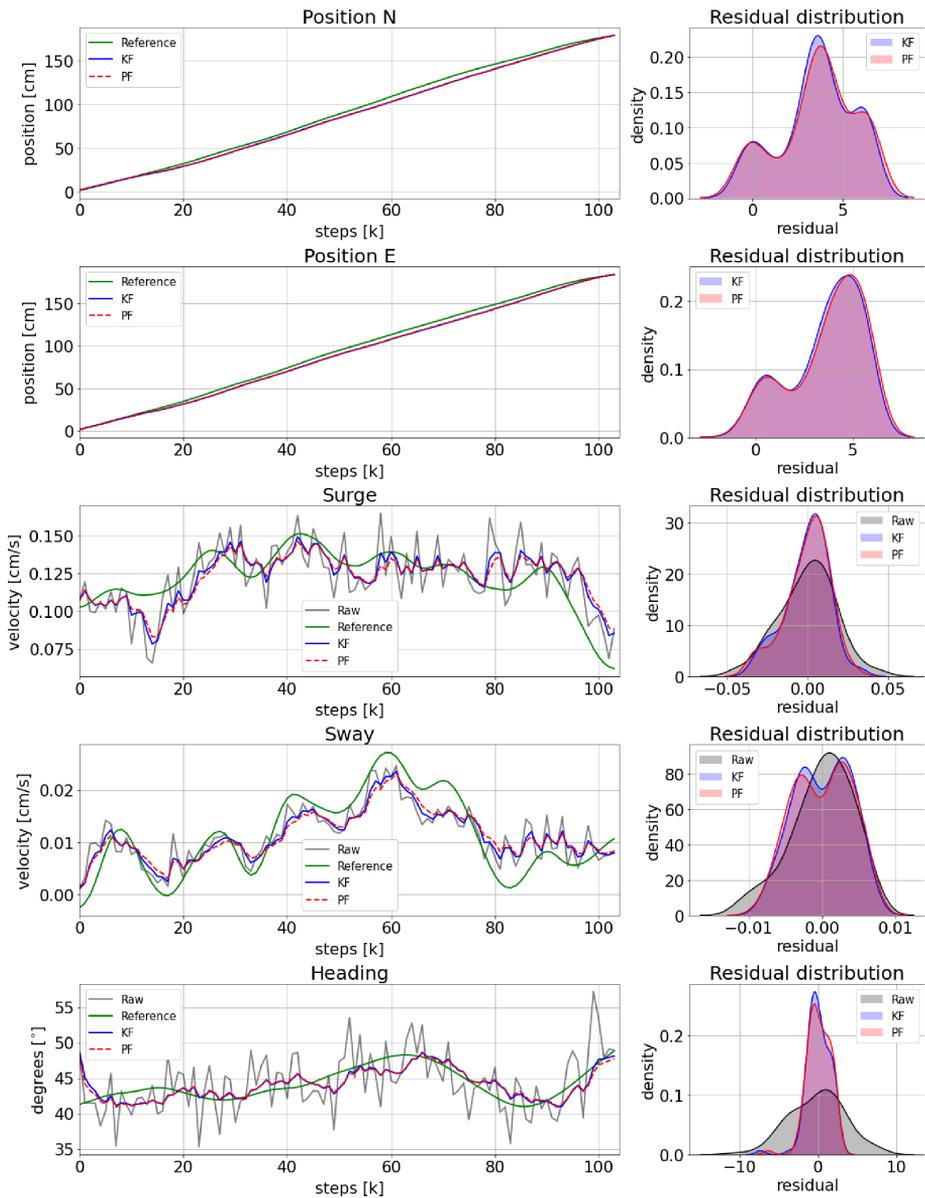
A.1 Results for the experiment #1 with still water conditions ($\zeta_a = 0$ mm)



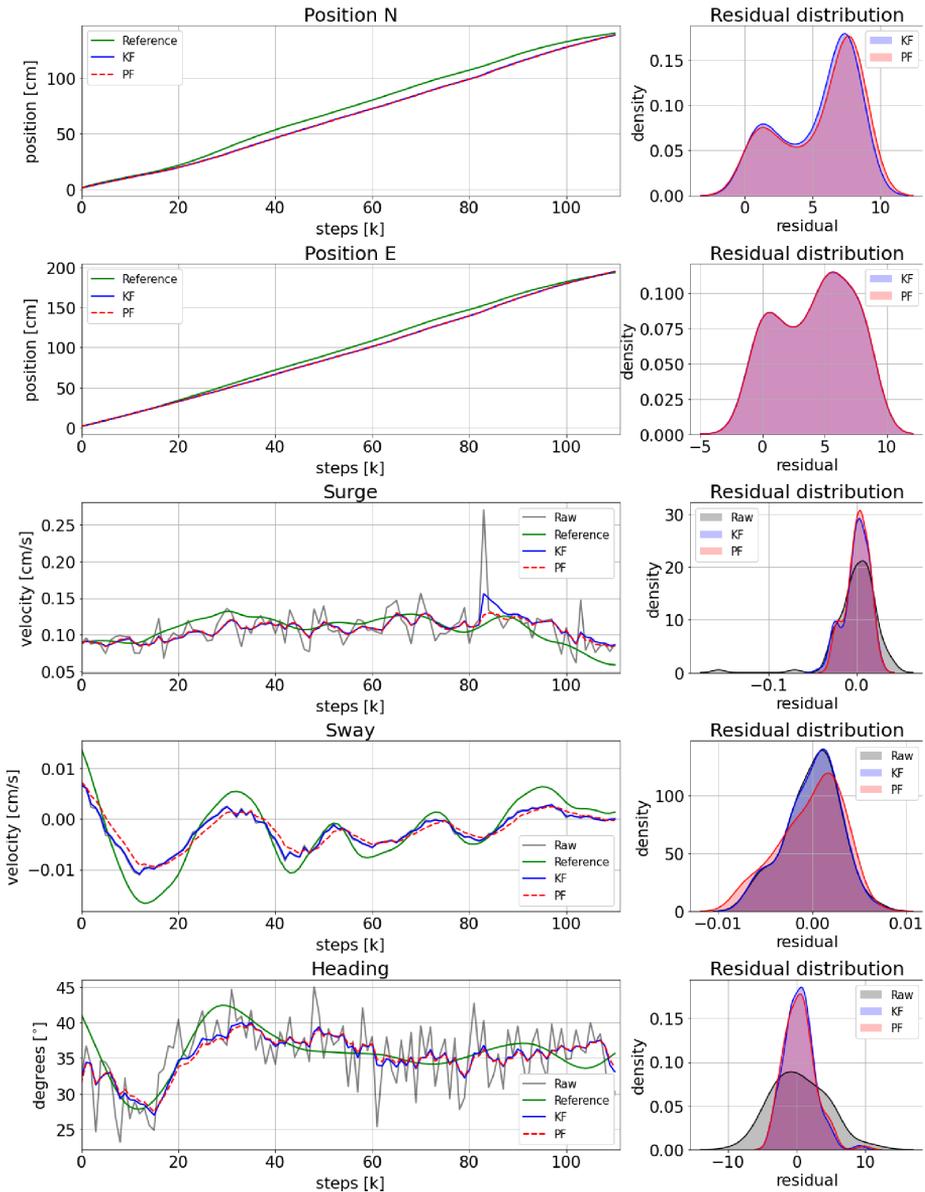
A.2 Results for the experiment #2 with $\gamma = 40^\circ$, $\zeta_a = 5$ mm and $\lambda = 210$ mm



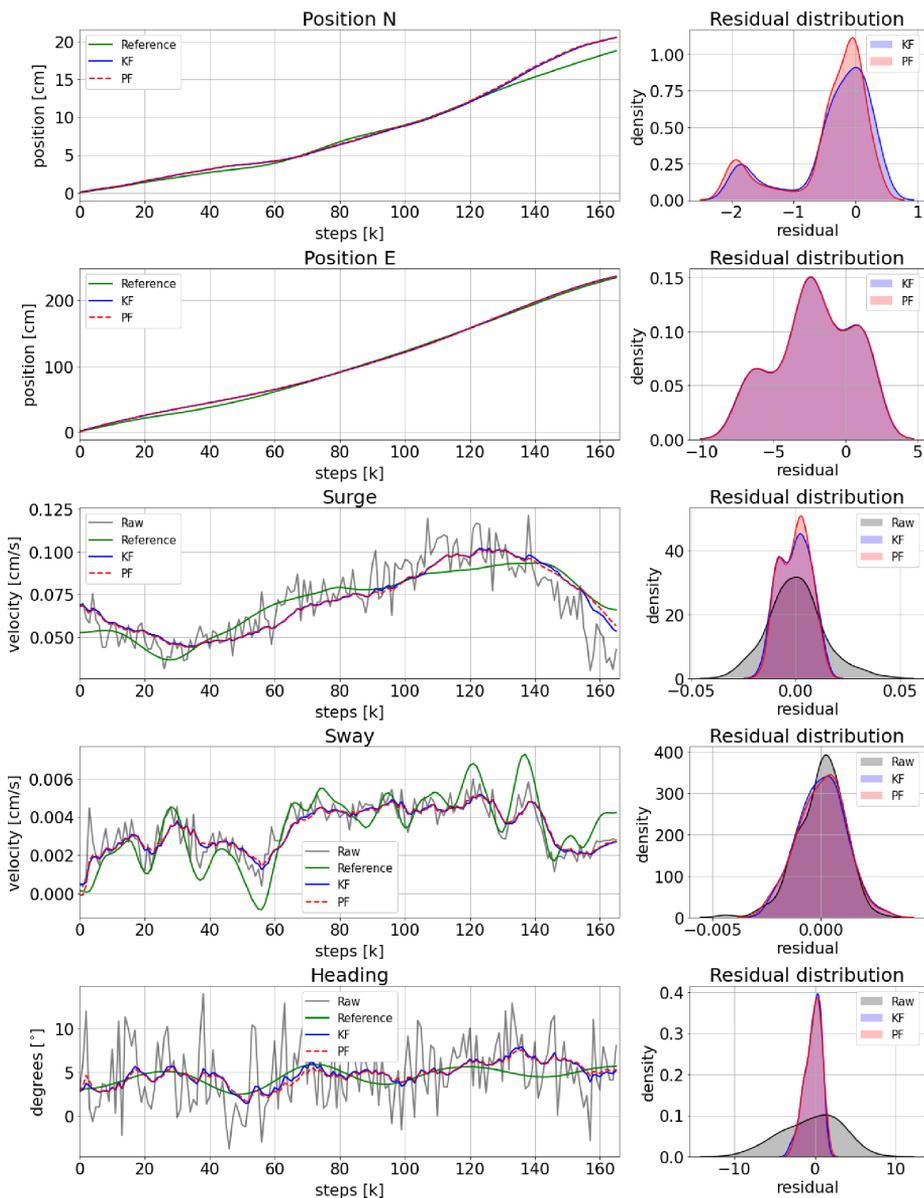
A.3 Results for the experiment #3 with $\gamma = 40^\circ$, $\zeta_a = 7.5$ mm and $\lambda = 132$ mm



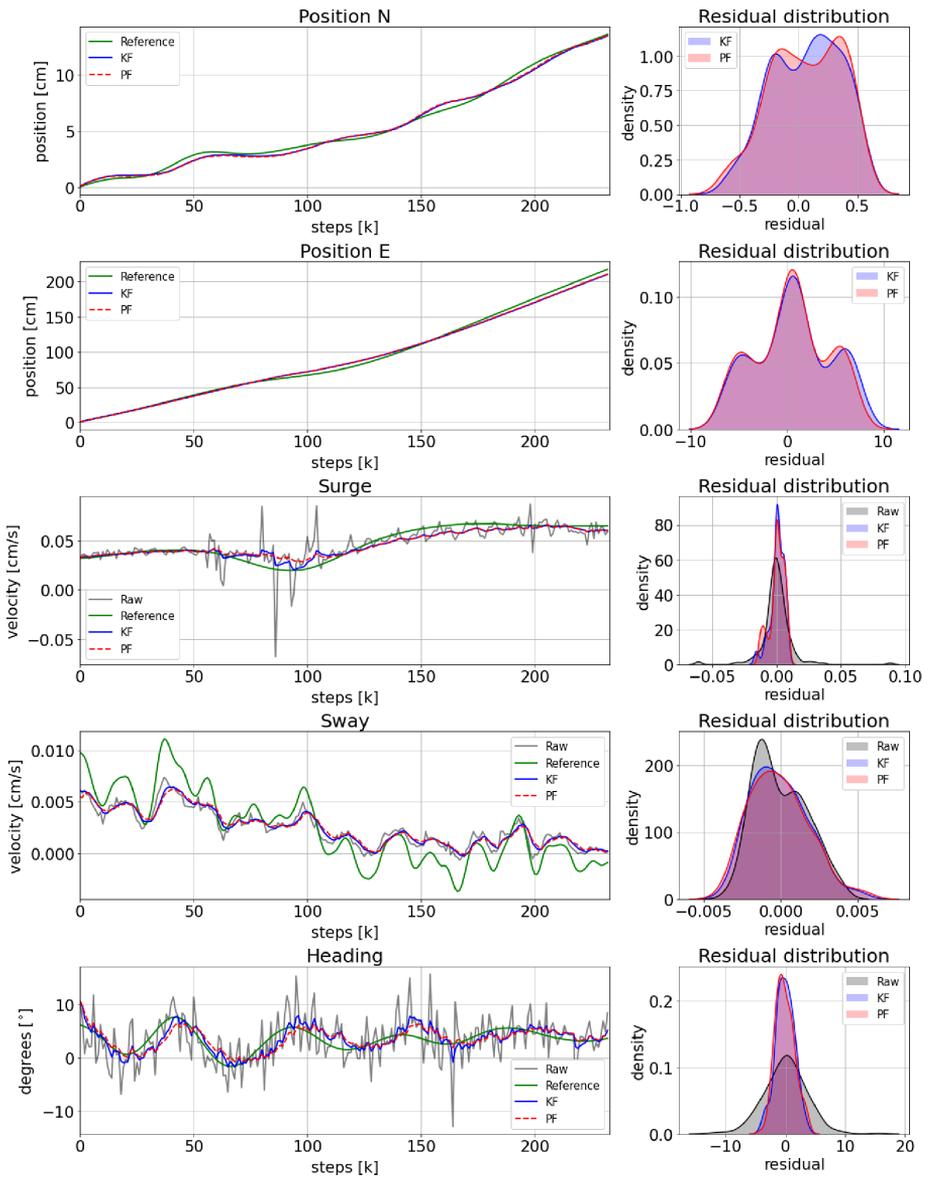
A.4 Results for the experiment #4 with $\gamma = 40^\circ$, $\zeta_a = 11$ mm and $\lambda = 59$ mm



A.5 Results for the experiment #5 with $\gamma = 0^\circ$, $\zeta_a = 5$ mm and $\lambda = 210$ mm



A.6 Results for the experiment #6 with $\gamma = 0^\circ$, $\zeta_a = 7.5$ mm and $\lambda = 132$ mm



A.7 Results for the experiment #7 with $\gamma = 0^\circ$, $\zeta_a = 11$ mm and $\lambda = 59$ mm

