

RESEARCH

Open Access



GNSS-based navigation systems of autonomous drone for delivering items

Aurello Patrik¹, Gaudi Utama¹, Alexander Agung Santoso Gunawan², Andry Chowanda¹, Jarot S. Suroso³, Rizatus Shofiyanti⁴ and Widodo Budiharto^{1*} 

*Correspondence:

wbudiharto@binus.edu

¹ Science Department,
School of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia

Full list of author information
is available at the end of the
article

Abstract

This paper presents our research on the development of navigation systems of autonomous drone for delivering items that uses a GNSS (Global Navigation Satellite System) and a compass as the main tools in drone. The grand purpose of our research is to deliver important medical aids for patients in emergency situations and implementation in agriculture in Indonesia, as part of the big mission of Society 5.0 and related with big data. In sending process, drone must be able to detect object and reach goal position and go back safely using GPS. We proposed a navigation algorithm for drone including the usage of course-over-ground information to help drone in doing autonomous navigation. In the experiment that we did, the average of positional deviation of landing position between the actual landing position and the desired landing position in the flight tests of flying from start to goal is 1.1125 m and for the tests that use the algorithm which uses course-over-ground, the positional deviation has average of 2.39 m. Navigation using course-over-Ground algorithm is not more reliable than the navigation algorithm with GNSS and Compass at a navigation distance of less than 1 m.

Keywords: Drone, GNSS, Navigation systems, Big data

Introduction

Worldwide media and scientist attention have put Unmanned Aerial Vehicles (UAVs) in the spotlight. UAV or also known as a drone is an unmanned aerial vehicle that has the main functions for intelligence, reconnaissance and surveillance [1]. The recent developments of drone for sprayer pesticide applications and used for delivering items, for example the Amazon Prime Air, where Amazon used an octocopter to deliver items with weighs less than 5 lb or around 2.3 kg [2]. UAVs can be operated more economically than manned helicopters; they are less limited by weather conditions (although this varies by model) and easier to deploy. In recent researches, drone may be built with a purpose to carry a rescue mission with the main module consists an exploration into an area affected by natural disaster and victim identification [3]. Drone also can be used as a tool to survey a building [4], aerial photogrammetry and mapping [5]. Recent advances also pushing drone technology to adopt a newer way of communication, such as an implementation of Mobile Edge Computing (MEC) [6], and Low Power Wide Area Network (LPWAN) [7]. The effectiveness of drone delivery is proven to bring a difference over manned ground transportation [8]. Even the

delivery fleet management may be automated as well in the future [9]. To engage in human activities however, drone needs a capability to do object detection [10]. Deep learning is a fast-growing domain of machine learning, mainly for solving problems in computer vision. It is a class of machine learning algorithms that use a cascade of many layers of nonlinear processing. One of the implementations of deep learning is object localization and detection based on video stream. Object localization and detection are crucial in computer vision. As of now, drone implementation for transportation solution and agriculture have not yet realized in Indonesia. This research will become part of the beginning of drone research in Indonesia, as the drone technology will play a major role in Society 5.0, and the eventual demands of drone fleet traffic control which deals with large amount of data (big data) and came from multiple sources. The proposed system must adopt a technology that suitable in Indonesia, where there is minimum support for mobile network, thus our system should not rely on this technology. In this research, we use Erle Drone [11], and as the heart of our drone system, it uses Erle-Brain 3 hardware autopilot and the APM flight stack. Erle-Brain 3 consists of an Ubuntu Linux based embedded computer with full support for ROS (Robot Operating System) and ROS 2.0 that integrated the sensors, camera, power electronics and abstractions necessary to easily create autonomous vehicles. The paper is organized as follows: We provide the problem statements in introduction section that we answer with this paper in proposed methods. We provide experimental results and discussion. We provide our conclusion in “[Experimental result](#)” section. Our prototype of drone is shown in Fig. 1.

Related works

One of the state of the art researches that have similar approach with this research is the usage of only GPS in drone navigation in the research done by [12]. There’s also another one that uses GPS and onboard sensor in AR. Drone in the research done by [13]. Last but no least, there’s also a research that’s not only uses GPS and onboard sensors, but also uses camera that uses Dense Optical Flow Method in the research done by [14]. In our previous work, we have developed a drone with object detection capabilities to improve delivery system.



Fig. 1 We use drone from Erle Robotics [11] based on the ROS for delivering items. Erle-copter’s flight time is around 20 min with a 5000 mAh battery, and the drone is built to support a payload of up to 1 kg

Navigation system

Geodetic coordinate system

Geodetic coordinate system is a coordinate system which a position defined by 3 numbers, a latitude, longitude, and altitude. A position which defined by geodetic coordinate system is a position on a globe. Latitude is a line that intersects the defined position and a line which parallel to an equator line. Longitude is a line that intersects the defined position and a line which parallel to a prime meridian line. Altitude is a distance between defined position and the ellipsoid Ref. [15] as described in Fig. 2.

Body coordinate system

Body coordinate system is a coordinate system which the origins of 3 axes (x, y, and z) are the center of mass of the vehicle. In a NED (North, East, Down) system, x axis points towards the head of the vehicle, y axis points towards the right side of the vehicle, and z axis points towards the down side of the vehicle [15]. Figure 3 describes the body coordinate system.

Bearing

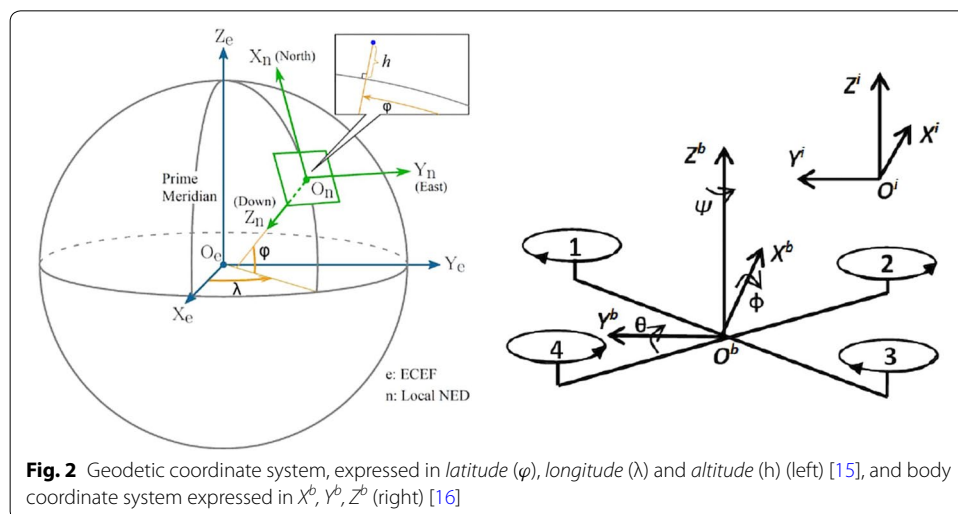
Bearing is an angle between 2 geodetic coordinates. To find bearing a bearing between current position and destination position, the formula is stated below:

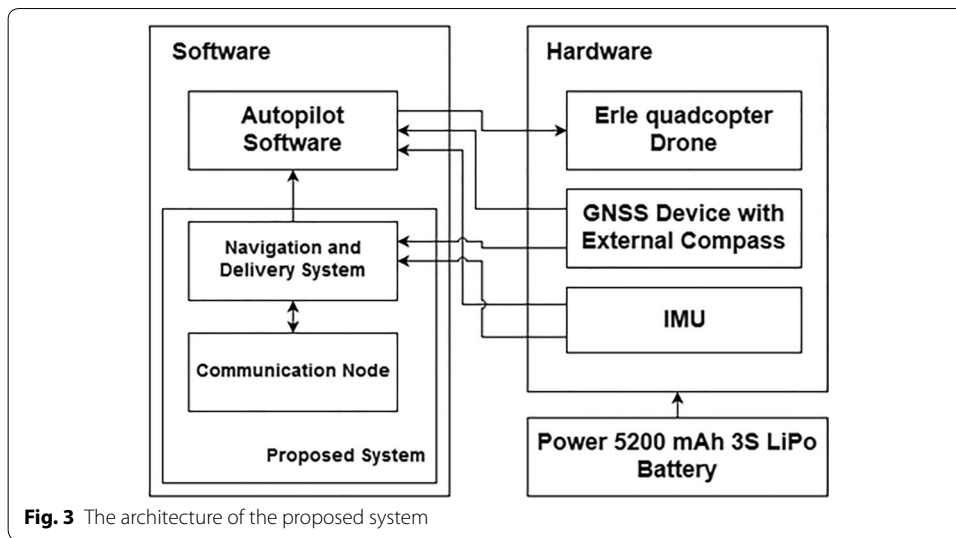
$$d_\lambda = p_\lambda^c - p_\lambda^d \tag{1}$$

$$X = \cos(p_\varphi^d) \sin(d_\lambda) \tag{2}$$

$$Y = \cos(p_\varphi^c) \sin(p_\varphi^d) - \sin(p_\varphi^c) \cos(p_\varphi^d) \cos(d_\lambda) \tag{3}$$

$$B = \text{atan2}(X, Y) \tag{4}$$





where B denotes the bearing, p is a geodetic coordinate with superscript c denotes current geodetic coordinate, and d denotes destination coordinate, and the subscript λ denotes longitude, and subscript φ denotes latitude [17].

Hubeny distance

Hubeny distance is a formula to calculate the distance between a geodetic coordinate in an earth model. The formula is stated below:

$$\bar{p}_\varphi = \frac{\pi(p_\varphi^d + p_\varphi^c)}{180 * 2} \tag{5}$$

$$M = \frac{a(1-s^2)}{\sqrt{(1-s^2 \sin(\bar{p}_\varphi))^2}^3} \tag{6}$$

$$N = \frac{a}{\sqrt{1-s^2 \sin(p_\varphi)^2}} \tag{7}$$

$$d_\varphi = \frac{\pi(p_\varphi^c - p_\varphi^d)}{180} \tag{8}$$

$$d_\lambda = \frac{\pi(p_\lambda^c - p_\lambda^d)}{180} \tag{9}$$

$$d^{c,d} = \sqrt{(Md_\varphi)^2 + (N \cos(\bar{p}_\varphi)d_\lambda)^2} \tag{10}$$

where a denotes the major axis and s denotes eccentricity of the earth model. d with superscript c , d denotes the distance between 2 different geodetic coordinates. A commonly used earth model is WGS84, which has major axis value 6378137, and eccentricity value 0.0818191908426215 [18].

Proposed method

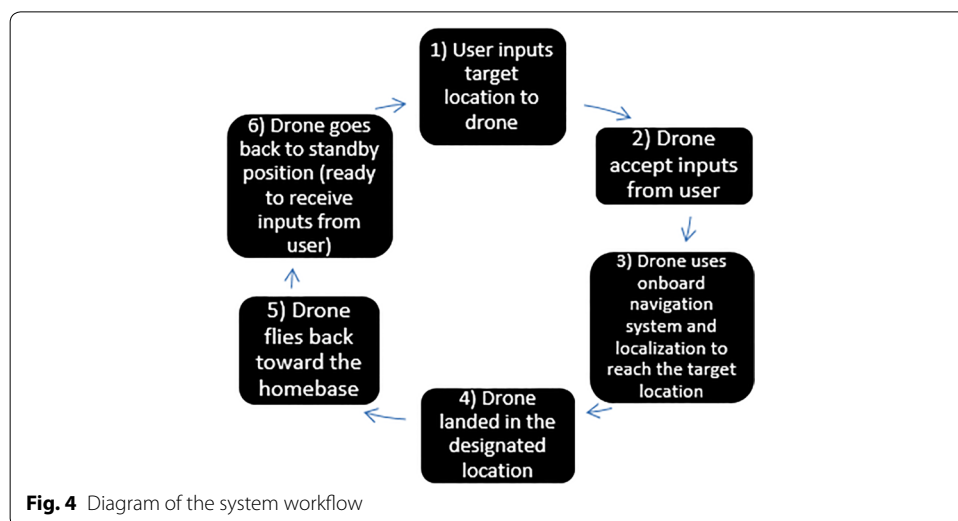
System architecture

In our previous research, we use object detection module that can detect what is in video stream by using combination of MobileNet and the Single Shot Detector (SSD) framework for fast and efficient deep learning-based method to object detection [10]. We use Erle-Brain 3 which consists a Linux based embedded computer and an autopilot shield which we design our system on. Inside the embedded computer, there's a ROS system and Autopilot Software installed. Research on autonomous drone usually using GPS [12], so we use GPS to get information of the position of the drone. To have a smooth navigational performance from autonomous drone, the need of precise positional data is a must. To get a precise positional data, we need to collect a large amount of satellite data which is why we use GNSS and compass that could connect to more variations of satellites. The autopilot shield consists of sensors and other components which essential for flying a drone. Our proposed system acts as an interface between user and both Autopilot Software and sensors as described in Fig. 3.

The proposed system works as follows, there are 6 steps consists of user input the location target for the drone, and then drone accept the location target, then fly to the target location, and then land to the target location, after that drone will fly back to home location, after land, drone goes back to standby position and ready to accept input from user. Figure 4 describes how the system works.

Proposed system on ROS

ROS uses publisher and subscriber pattern as its design, every process that runs on ROS are doing publishing and subscribing towards other processes. These processes are called ROS nodes. As the Autopilot Software provides our system with filtered state of the drone, our system has to subscribe to the pool which provided such data. The pool is called ROS topic. When the proposed system wants to command the drone to do the take-off sequence, the system would request the Autopilot Software to do the take-off



sequence using a ROS service. The relationships between nodes in our proposed system are described in Fig. 5.

Delivering items algorithm

Our proposed method uses delivery switch as a sensor which provides information whether items are loaded into the drone. When user wants to deliver items, user must place the items inside the drone, so the drone will start to fly to the target location. After the drone land on target location, the items must be unloaded from the drone before drone starts to fly to the home location. The algorithm is stated in Fig. 6.

Navigation algorithm

In the proposed system, the system would plan a trajectory from current location to target location using bearing. Then the bearing value will be compared with heading of the drone, as the proposed system will apply the velocity in a body coordinate system. The formula as follow:

$$\theta_d = \theta_w - \Psi \tag{11}$$

$$V_x = V \sin \theta_d \tag{12}$$

$$V_y = V \cos \theta_d \tag{13}$$

which θ denotes angle, subscript d denotes an angle the drone has to go through to approach the target location, and subscript w denotes the angle is a bearing between two

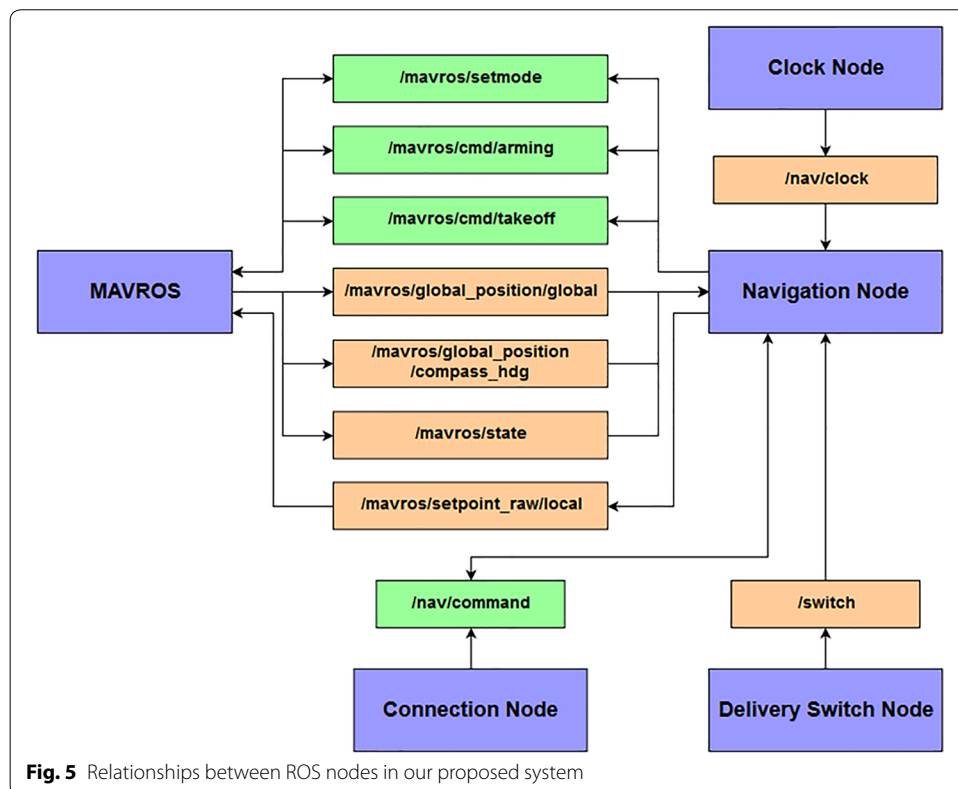
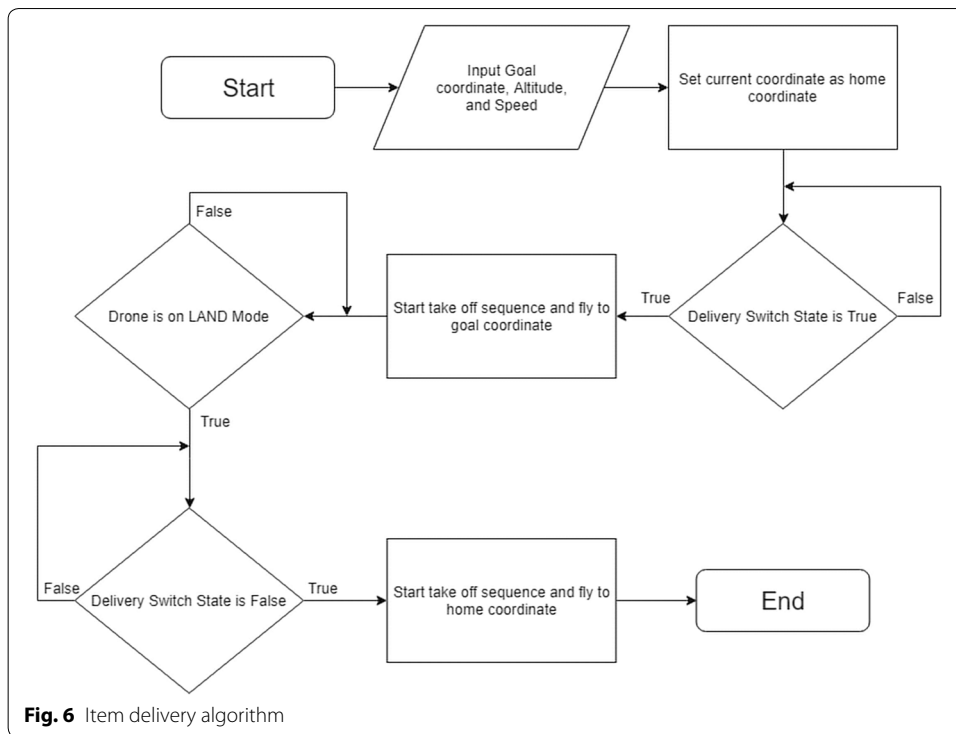


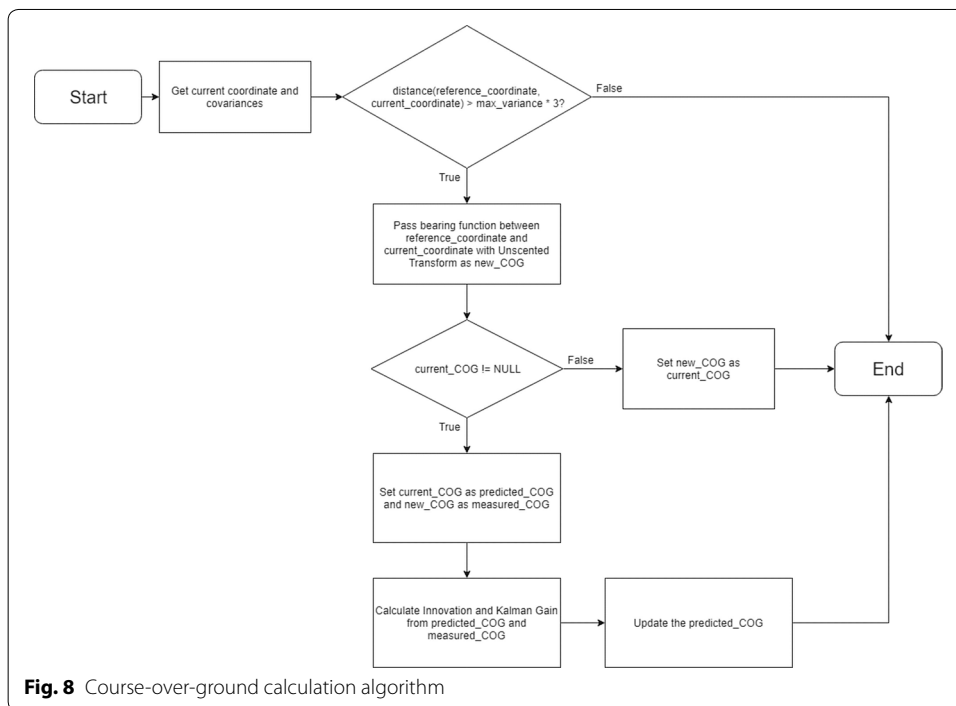
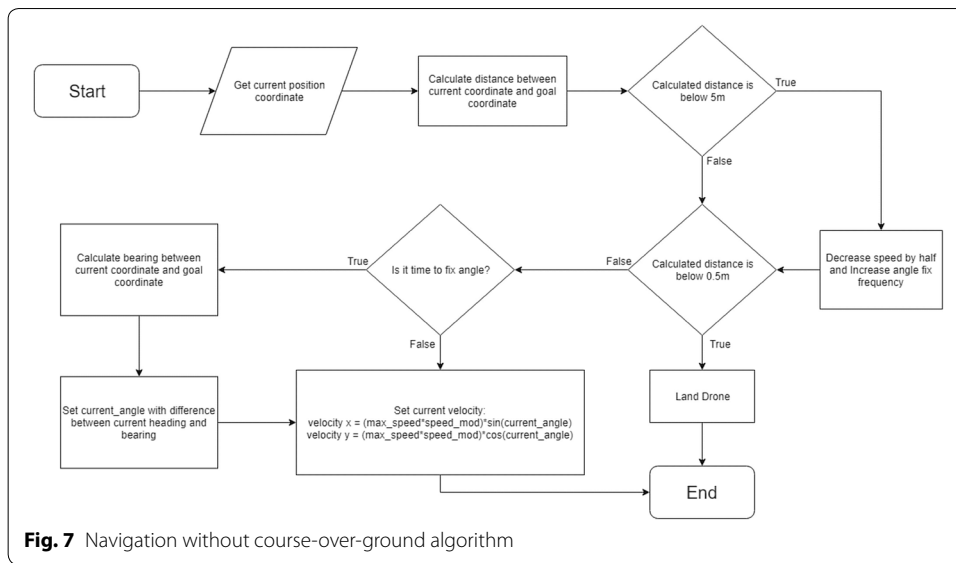
Fig. 5 Relationships between ROS nodes in our proposed system



locations. Ψ denotes heading of the drone. V denotes the velocity the system must apply, subscript x denotes the velocity in x axis, and y denotes the velocity in y axis. When the drone is flying to the target position, every few seconds, our system must recalculate the trajectory between current position and target position. The trajectory recalculation time described in the algorithm with “angle_fix” and “time_to_angle_fix” Booleans to provide system information whether the trajectory recalculation should be done. Also the system will check the distance between current coordinate and target coordinate iteratively. The system will slow down the velocity applied to the drone if the distance to the target is lower than 5 ms. In our proposed system, the drone must land if the distance to the target is lower than 0.5 m. These iterative processes are run every time the information about current position is acquired. The algorithm is stated in Fig. 7.

Navigation algorithm using course-over-ground

Other than using only heading information from IMU sensor, we proposed of using course-over-ground information as another sensorial data that we could use to further refine the drone navigation. course-over-ground is calculated using bearing between 2 recorded geodetic coordinates while drone flying in a specific trajectory created with previous algorithm. To further refine the course-over-ground value, our proposed method uses position covariances provided by Autopilot Software which sent together with current geodetic coordinate of the drone. We use Unscented Transform [19] to calculate bearing between 2 coordinates considering those 2 coordinates have covariances. The problem lies in position covariance provided by the Autopilot Software, which in meters. This causes some disagreement in the convention to define the position of the drone. To fix this, we have to calculate the variance of the position back in degree form.



To do this, we could use the Taylor Approximation of the Hubeny distance to provide us the reverse function from meters to degree. If there are more than one course-over-ground data, we could use Kalman Filter [20] to refine the our measurement. Figure 8 shows the algorithm used to obtain course-over-ground using Unscented Transform.

To use course-over-ground, we could imagine the course-over-ground being the output trajectory of the current trajectory input created from the previous iteration of trajectory calculation. With this in mind, we could replace the need of heading information entirely by comparing the input, output, and the bearing, but in this

system, we calculate the new trajectory together with heading information. To do this, we use Kalman Gain concept [20] to fuse the two calculations into one trajectory input. Before we fuse these calculations, we also use Unscented Transform to pass the trajectory calculation function for both trajectory from heading and trajectory from course-over-ground. The calculation stated below:

$$\theta_{world_t}^{cog}, \sigma_{\theta_{world_t}^{cog}} = UT \left(\begin{bmatrix} p_{\varphi}^d \\ p_{\lambda}^d \\ \theta_{cog_{t-1}} \end{bmatrix}, \begin{bmatrix} \sigma_{p_{\varphi}^d} & 0 & 0 \\ 0 & \sigma_{p_{\lambda}^d} & 0 \\ 0 & 0 & \sigma_{\theta_{cog_{t-1}}} \end{bmatrix}, func \right) \tag{14}$$

$$\theta_{world_t}^{hdg}, \sigma_{\theta_{world_t}^{hdg}} = UT \left(\begin{bmatrix} p_{\varphi}^d \\ p_{\lambda}^d \\ \theta_{hdg_t} \end{bmatrix}, \begin{bmatrix} \sigma_{p_{\varphi}^d} & 0 & 0 \\ 0 & \sigma_{p_{\lambda}^d} & 0 \\ 0 & 0 & \sigma_{\theta_{hdg_t}} \end{bmatrix}, func \right) \tag{15}$$

where $\theta_{world_t}^{cog}$ denotes the difference between course-over-ground and bearing between current position and target position, $\theta_{world_t}^{hdg}$ denotes difference between heading and bearing. Symbol σ denotes variance of its subscript. The UT (v, c, f) function is an Unscented Transform function, which passes vector v with covariance c into function f using Unscented Transform to obtain the result of the function and its covariance. The function func above is a function to find the difference between angle provided as parameter (3rd row of the vector) and the bearing between current position and target position.

After obtaining 2 input trajectories, we fuse those two calculations, below is the formula:

$$\theta_{drone_t}^{cog} = \theta_{drone_{t-1}} + \theta_{world_t}^{cog} \tag{16}$$

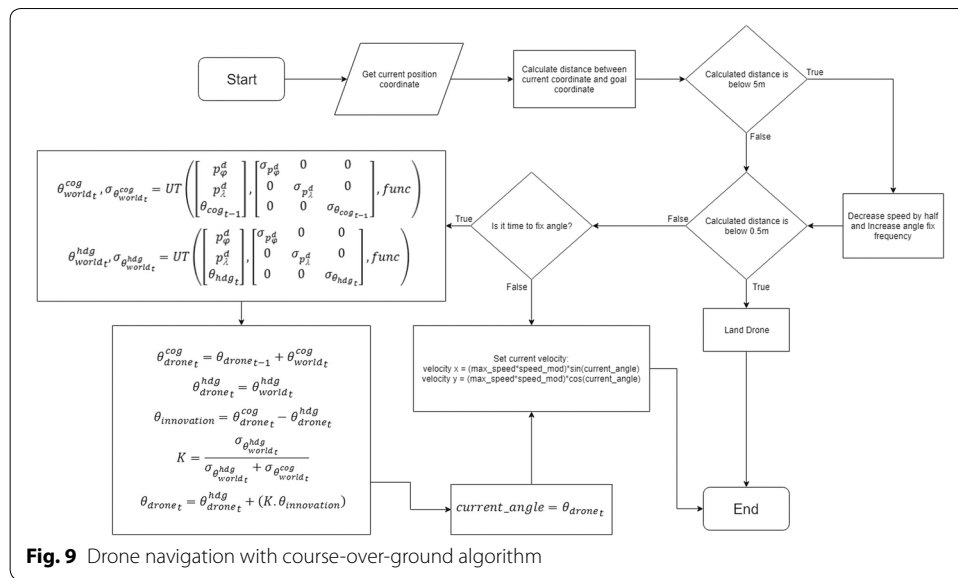
$$\theta_{drone_t}^{hdg} = \theta_{world_t}^{hdg} \tag{17}$$

$$\theta_{innovation} = \theta_{drone_t}^{cog} - \theta_{drone_t}^{hdg} \tag{18}$$

$$K = \frac{\sigma_{\theta_{world_t}^{hdg}}}{\sigma_{\theta_{world_t}^{hdg}} + \sigma_{\theta_{world_t}^{cog}}} \tag{19}$$

$$\theta_{drone_t} = \theta_{drone_t}^{hdg} + (K \cdot \theta_{innovation}) \tag{20}$$

where $\theta_{drone_t}^{cog}$ denotes the difference between the difference between course-over-ground and bearing and the previous input trajectory, $\theta_{drone_t}^{hdg}$ denotes the difference between drone heading and bearing, and θ_{drone_t} denotes the trajectory input for the drone, with subscript t as the time step. Figure 9 shows the full navigation system with course-over-ground.



Experimental result

We’ve done the experiment to test our algorithm in the open field. Our experiment took place in soccer field, Jakarta. Before we begin our experiment, first the drone need to be powered in a level ground to let the Autopilot Software to do a calibration. Also to get a better position fix on the GNSS device. In our experiment, we let the device in the open until the value of Dilution of Precision hit 0.9 or visible satellites count reached minimum 10. After that we execute the flying mission to the goal position using our proposed algorithm. For each algorithm, we run this test 4 times (Fig. 10).

The result of the experiments is shown in Tables 1 and 2.

Figure 11 shows the trajectories created from the recorded filtered position while doing the experiments.

Based on our experiment, the deviation of landing position of the drone with our proposed method are better than previous research, which only use GPS as it’s navigation system. We also develop a mobile application to send the command consisting geodetic coordinate of the goal position, altitude, and speed of the drone. The application will connect to the drone via pre-determined IP and port. Figure 12 shows the interface of the mobile application.

Discussion

The proposed navigation algorithm successfully makes our drone fly from the start position to goal position without problem and have an acceptable deviation, with the mean of deviation in navigation algorithm without course-over-ground 1.1125 m and 2.39 m for the navigation algorithm using course-over-ground. Autonomous navigation system using GNSS module and compass succeeded in finishing a flight mission from start position to goal position nicely and has a relatively small positional deviation. The proposed navigation algorithm still has weakness, such as, this algorithm isn’t good for short distance navigation, so when the drone is nearing the goal position, drone is having a bit of



Fig. 10 The start and goal position of the experiment, taken from Google Maps [21]

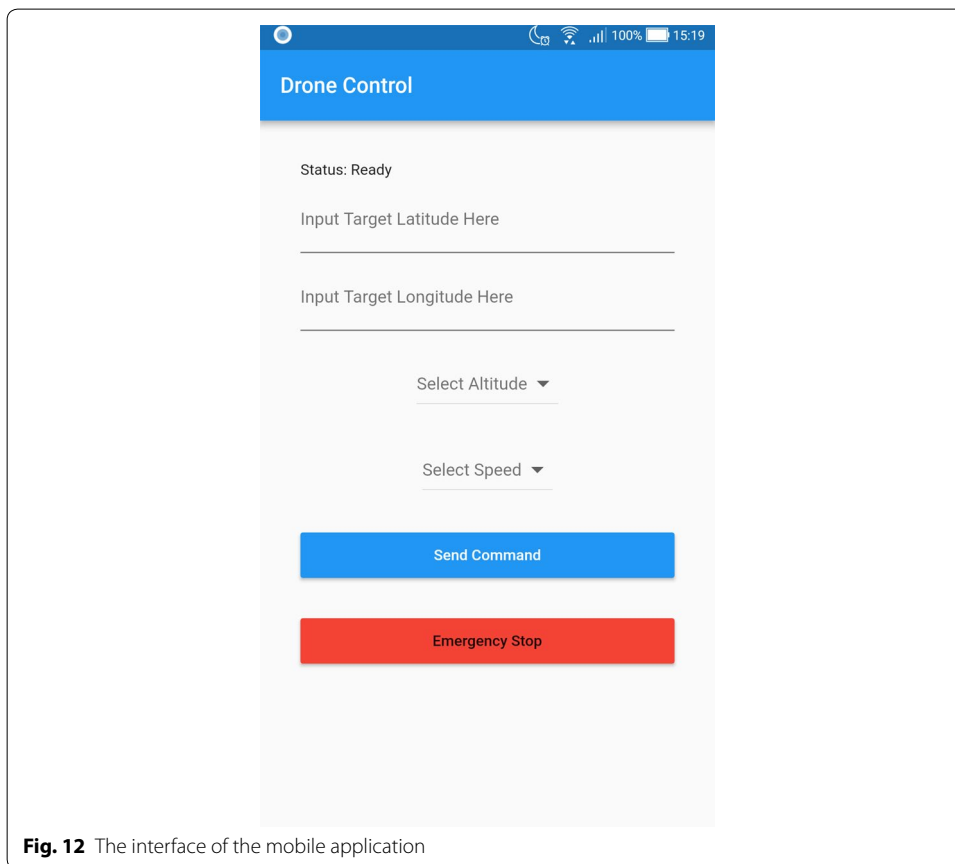
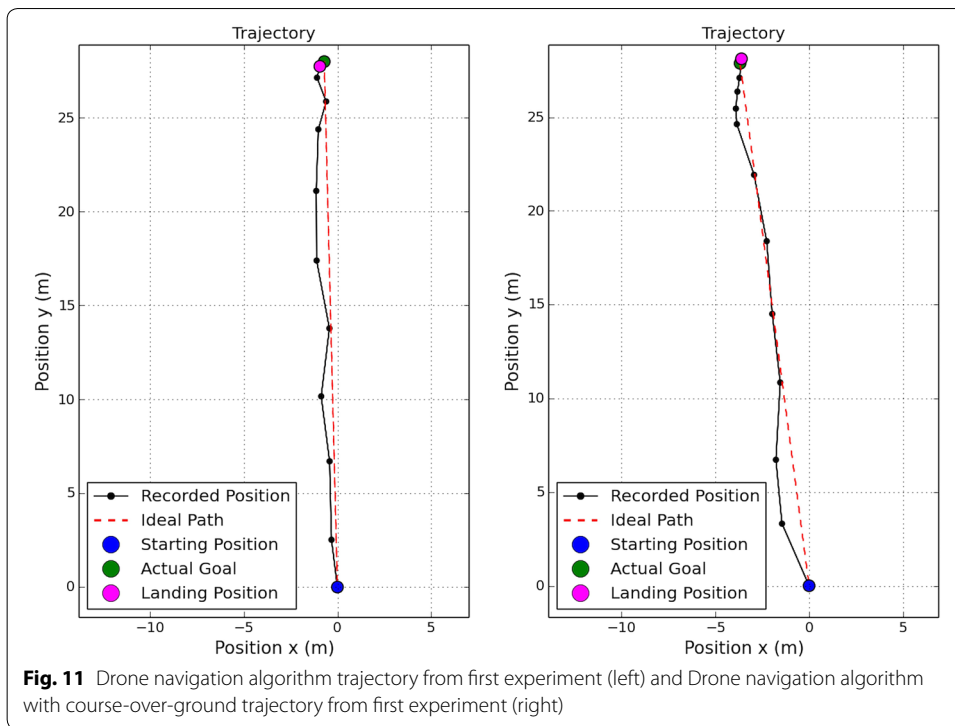
Table 1 Positional deviation from navigation algorithm

Experiment-n	Positional deviation (m)
1	0.44
2	1.04
3	1.15
4	1.82
Average of error	1.1125

Table 2 Positional deviation from navigation algorithm with course-over-ground

Experiment-n	Positional deviation (m)
1	2.5
2	4.46
3	2.2
4	0.4
Average of error	2.39

difficulty to reach the goal position perfectly. One of the factors of this weakness is the mistakes from GNSS module. Other than that, the main weakness of the drone itself is wind speed, as seen in Fig. 11, drone’s trajectory will never match the ideal trajectory



in a windy condition, but the proposed algorithm will ensure the drone will approach and land in the goal position. Navigation using the course-over-ground algorithm is not more reliable than the navigation algorithm with GNSS and Compass at a navigation distance of less than 1 m. When the drone is flying to the target position, every few seconds, our system must recalculate the trajectory between current position and target position.

Conclusion

From the results, we can conclude that the proposed system in this research succeeded in making drone do a simulated items delivery mission with a good navigation, and an acceptable landing position deviation. Comparing both navigation algorithm, we can conclude that navigation without course-over-ground results in better landing position deviation, the reason being the course-over-ground algorithm can't contribute too much in terms of refining the accuracy with course-over-ground calculation at a navigation with distance of less than 1 m. The system has a few features such as altitude and speed settings. Proposed system designed to be able to interact with interactive sensors that supported items delivery, such as items sensor. This system also has an interface in the form of a mobile app that could be used relatively easy. Moreover, the system could run separately from Ground Station, so with only a mobile app and the drone, it could do an items delivery mission. Autonomous drones have potential usage in other fields, by using navigational algorithm from this research as the base for drone flight. Other than that, the navigational algorithm itself could be further enhanced and fixed using a more precise and adaptive algorithm that could navigate from any distance and for any usages. For future work, we need to address some issue regarding larger drone fleet and handle larger amount of navigation data to manage drone traffic. In other side, by using a bigger drone could give opportunity to do researches with a bigger scale experiments, such as delivering items with distance more than 1 km.

Abbreviations

UAVs: Unmanned Aerial Vehicles; GPS: The Global Positioning System; GNSS: Global Navigation Satellite System; ROS: Robot Operating System; SSD: Single Shot Detector.

Acknowledgements

We say thanks to Bina Nusantara University for supporting this research.

Authors' contributions

All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹ Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia. ² Mathematics Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia. ³ Information Systems Department, BINUS Graduate Program-Master of Information Systems Program, Bina Nusantara University, Jakarta 11480, Indonesia. ⁴ Indonesian Agency for Agricultural Research and Development, Ministry of Agriculture, Bogor, West Java, Indonesia.

Received: 28 February 2019 Accepted: 3 June 2019

Published online: 14 June 2019

References

1. Jha AR. Theory, design, and applications of unmanned aerial vehicles. New York: CRC Press; 2016.
2. M. Hickey, Meet Amazon Prime Air, A Delivery-By-Aerial-Drone Project, 2013.
3. Apvrille L, Tanzi T, Dugelay JL. Autonomous drones for assisting rescue services within the context of natural disasters. In: General assembly and scientific symposium (URSI GASS), 2014 XXXIth URSI, 2014, p. 1–4.
4. Entrop AG, Vasenev A. Infrared drones in the construction industry: designing a protocol for building thermography procedures. *Energy Procedia*. 2017;132:63–8.
5. Casella E, et al. Mapping coral reefs using consumer-grade drones and structure from motion photogrammetry techniques. *Coral Reefs*. 2017;36(1):269–75.
6. Motlagh NH, Bagaa M, Taleb T. UAV-based IoT platform: a crowd surveillance use case. *IEEE Commun Mag*. 2017;55(2):128–34.
7. Mekki K, Bajic E, Chaxel F, Meyer F. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, 2018.
8. Claesson A, et al. Time to delivery of an automated external defibrillator using a drone for simulated out-of-hospital cardiac arrests vs emergency medical services. *JAMA*. 2017;317(22):2332–4.
9. Scott J, Scott C. Drone delivery models for healthcare. In: Proc. 50th Hawaii Int. Conf. Syst. Sci., p. 3297–304, 2017.
10. Budiharto W, et al. Fast Object detection for quadcopter drone using Deep Learning, Th. In: 3rd international conference on computer and communication systems (ICCCS), p. 192–5, Japan, 2018.
11. "Erle Copter Drone Kit." <https://erlerobotics.com/blog/product/erle-copter-diy-kit/>. Accessed 15 May 2018.
12. Kan M, Okamoto S, Lee JH. Development of drone capable of autonomous flight using GPS. In: Proceedings of the international multi conference of engineers and computer scientists, vol. 2, 2018.
13. Santana LV, Brandao AS, Sarcinelli-Filho M. Outdoor waypoint navigation with the AR. Drone quadrotor. In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS). Denver: IEEE; 2015. pp. 303–11.
14. Arreola L, de Oca AM, Flores A, Sanchez J, Flores G. Improvement in the UAV position estimation with low-cost GPS, INS and vision-based system: Application to a quadrotor UAV. In: 2018 International Conference on Unmanned Aircraft Systems (ICUAS); 2018, pp. 1248–54.
15. Cai G, Chen BM, Lee TH. Unmanned rotorcraft systems. New York: Springer; 2011.
16. Li P, Garratt M, Lambert A, Pickering M, Mitchell J. Onboard hover control of a quadrotor using template matching and optic flow. In: Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV), 2013, p. 1.
17. "Formula to Find Bearing." <https://www.igismap.com/formula-to-find-bearing-or-heading-angle-between-two-points-latitude-longitude/>. Accessed: 03 Jan 2019.
18. "NGA: DoD World Geodetic System 1984." http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html. Accessed: 03 Jan 2019.
19. Särkkä S. Bayesian filtering and smoothing, vol. 3. Cambridge: Cambridge University Press; 2013.
20. Chui CK, Chen G. Kalman filtering: with real-time applications. New York: Springer; 2017.
21. Google Maps. [Online]. Available: <https://www.google.com/maps>. Accessed 07 June 2019.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
