Autonomous Robots (2020) 44:859–875
https://doi.org/10.1007/s10514-020-09902-3

Autonomous ballistic airdrop of objects from a small fixed-wing unmanned aerial vehicle

Siri Gulaker Mathisen[1] · Frederik Stendahl Leira[1] · Håkon Hagen Helgesen[1] · Kristoffer Gryte[1] · Tor Arne Johansen[1]

Received: 27 August 2018 / Accepted: 18 January 2020 / Published online: 30 January 2020
© The Author(s) 2020

Abstract
Autonomous airdrop is a useful basic operation for a fixed-wing unmanned aerial system. Being able to deliver an object to a known target position extends operational range without risking human lives, but is still limited to known delivery locations. If the fixed-wing unmanned aerial vehicle delivering the object could also recognize its target, the system would take one step further in the direction of autonomy. This paper presents a closed-loop autonomous delivery system that uses machine vision to identify a target marked with a distinct colour, calculates the geographical coordinates of the target location and plans a path to a release point, where it delivers the object. Experimental results present a visual target estimator with a mean error distance of 3.4 m and objects delivered with a mean error distance of 5.5 m.

Keywords Real-time machine vision · Autonomous UAV · Target recognition · Path planning · Guidance and control · Target identification

1 Introduction

There are numerous situations where delivery of a small object from an unmanned flying vehicle could be useful. Fixed-wing aircraft are capable of significantly larger range and speed compared to hovering aircraft. Placement of sensors on top of an iceberg is one area of application that requires long range. A possible success scenario could be a fixed-wing unmanned aerial vehicle (UAV) scanning a marine area looking for icebergs, then deploying sensors on the iceberg. Using a UAV instead of manned aircrafts for this purpose would minimize the risk for human lives at the same time as keeping the costs down: Manned aircrafts demand another dimension for the amount of fuel used, hiring a plane or a helicopter is costly and flying low in the Arctic is quite risky. The UAV could be launched from and later return to a ship, further prolonging the operational range. Another application is cargo transport in rural areas, or emergency equipment in search and rescue operations.

Precision drops have been researched for both civil and military use: Williams and Trivailo (2006) discusses a technique for cable guided delivery from a fixed-wing manned aircraft, Wright et al. (2005), Benney et al. (2005), Tavan (2006) and Joshua and Eaton (2013) describe a military precision airdrop system from fixed-wing aircrafts, using a steerable parachute to guide large cargo to the ground. Klein and Rogers (2015) presents simulation results where a probabilistic mission planner for unguided drops with parachutes is used. Gerlach et al. (2016) uses airdrops with parachutes while optimizing the travel distance required to recover the dropped objects to a base station. VanderMey et al. (2015) wants to improve the precision of drops with parachutes by offering a model-based framework. All of these studies are performed on, or intended for, larger, manned aircraft releasing its load from a high altitude. McGill et al. (2011) describes flight tests for deployment of global positioning system (GPS) sensors on icebergs, and the research describes use of manually guided, small UAVs. UAVs would bring an extensive advantage over manned aircraft when operating off-shore, keeping the human in the loop safely on-board a

This work was supported by the Research Council of Norway through the Centres of Excellence funding scheme, Project Number 223254. It has also been suppoorted by NFR/MAROFF with Project Number 269480.

✉ Siri Gulaker Mathisen
siri.mathisen@sintef.no

[1] Department of Engineering Cybernetics, Centre for Autonomous Marine Operations and Systems (NTNU AMOS), NTNU, Norwegian University of Science and Technology, Trondheim, Norway

ship or another platform. If the UAV is small as well, the costs would be reduced further in case of loss of the vehicle.

To determine where to release the object, the UAV could be provided with the geographical coordinates of a pre-detected iceberg, then autonomously fly there and perform its mission. However, as icebergs move, though slowly, the observation of the iceberg would have to be very recent or the velocity estimated. Even though it is possible to detect icebergs using a satellite (Enderlin et al. 2018; Shi et al. 2017; Schaffer et al. 2017; Sulak et al. 2017; Zakharov et al. 2017; Moctezuma-Flores and Parmiggiani 2017; Kucheikoa et al. 2016), the information latency may be hours or days, and various conditions like surrounding sea ice can make this difficult and can cause false detections (Zakharov et al. 2017). A solution could be to use the same UAV to detect the icebergs (Leira et al. 2017) and to deliver the object.

This paper presents research on a closed-loop autonomous solution for a UAV to deliver a small object on a given target. The UAV scans an area looking for specific visual features and uses machine vision with real-time on-board data processing to identify a target on the ground. It then calculates a release point and flies autonomously to this point, where it releases the object to fall down ballistically on the target. A visual spectrum camera is used to collect images from the area where the target is expected to be located. Real-time image processing based on colour segmentation is used to detect the target and find the pixel position in the image. Georeferencing (Leira et al. 2015; Helgesen et al. 2017) is used to transform pixel coordinates from a single image into more informative Earth-fixed coordinates. Because georeferenced coordinates based on a single observation (image) can be uncertain, the mean georeferenced position, reconstructed from a batch of images, is used as the position of the target. These coordinates are sent to the drop algorithm which is responsible for calculating the release point and execute the drop.

The main contribution in this paper is a closed-loop method for autonomous air drops. Not only does the system deliver objects to the target, it is also able to autonomously locate the target using on-board sensors and processing. Experimental results validate the method, and all processing is performed in real-time on-board the UAV. The solution builds on the publications in Mathisen et al. (2017) and Leira et al. (2015), where this paper's novel contribution is the combination of a target detection system, a target position estimation system, an airdrop planning system and a guidance and control system. The airdrop planning system is simpler and more precise than the one described in Mathisen et al. (2017): The airdrop planning system presented in this present paper does not optimize its path frequently, like the one in Mathisen et al. (2017), but it contains a more finely tuned ballistic model and presents more reliable results where the ability to deliver is higher. Moreover, the detection and georeferencing systems are implemented on the on-board computer and used in real-time, which was not the case in Leira et al. (2015). Therefore, this paper also shows that target detection through image-processing and georeferencing can be conducted in real-time on a small embedded computer with limited processing capacity.

2 Closed-loop autonomous delivery system

The system consists of three separate subsystems, the target recognition and position estimation subsystem, the airdrop planning subsystem, and the guidance and control subsystem, as illustrated in Fig. 1. The subsystems are executed on the on-board computer, performing all their processing in real-time.

The target recognition system is responsible for estimating Earth-fixed coordinates for the target on the ground. The input to this subsystem is visual spectrum images and target characteristics. An automatic search pattern is used to capture images and the detection algorithm searches for the target signature in the images. If the target is detected, Earth-fixed coordinates are calculated with georeferencing. When a sufficient amount of observations and a reliable estimate of the target coordinates are gathered, the subsystem sends these to the airdrop planning system.

The airdrop planning subsystem receives a target position estimate as input. It calculates a path from the UAV's current position through a truncated Dubins path (Beard and McLain 2012) to an ideal release point, where the UAV should release the object it carries. The output of this path planning algorithm is a series of reference points that are ordered in pairs, creating reference lines for the UAV to follow. These reference lines are sent to the guidance and control subsystem.

In the guidance and control subsystem, a line of sight (LOS) guidance algorithm (Beard and McLain 2012) calculates the desired course the UAV should follow, and thus guides the UAV. In the longitudinal plane, an unpublished LOS-based controller (Nevstad 2016) based on You et al. (2012) keeps the UAV's height stable.

The navigation system is provided by the autopilot, which receives set-points from the guidance and control subsystem and performs low-level control of the UAV. The autopilot delivers wind estimates to the airdrop planning subsystem and guidance and control subsystem.

If something should happen and the UAV does not get close enough to the release point, it experiences a missed target and has to start over and recalculate the release point. The interface between the first two subsystems is the coordinates of the target, which the target recognition and position estimation subsystem has estimated, and which the airdrop planning subsystem uses as input to its path planning algorithm. The interface between the airdrop planning subsystem

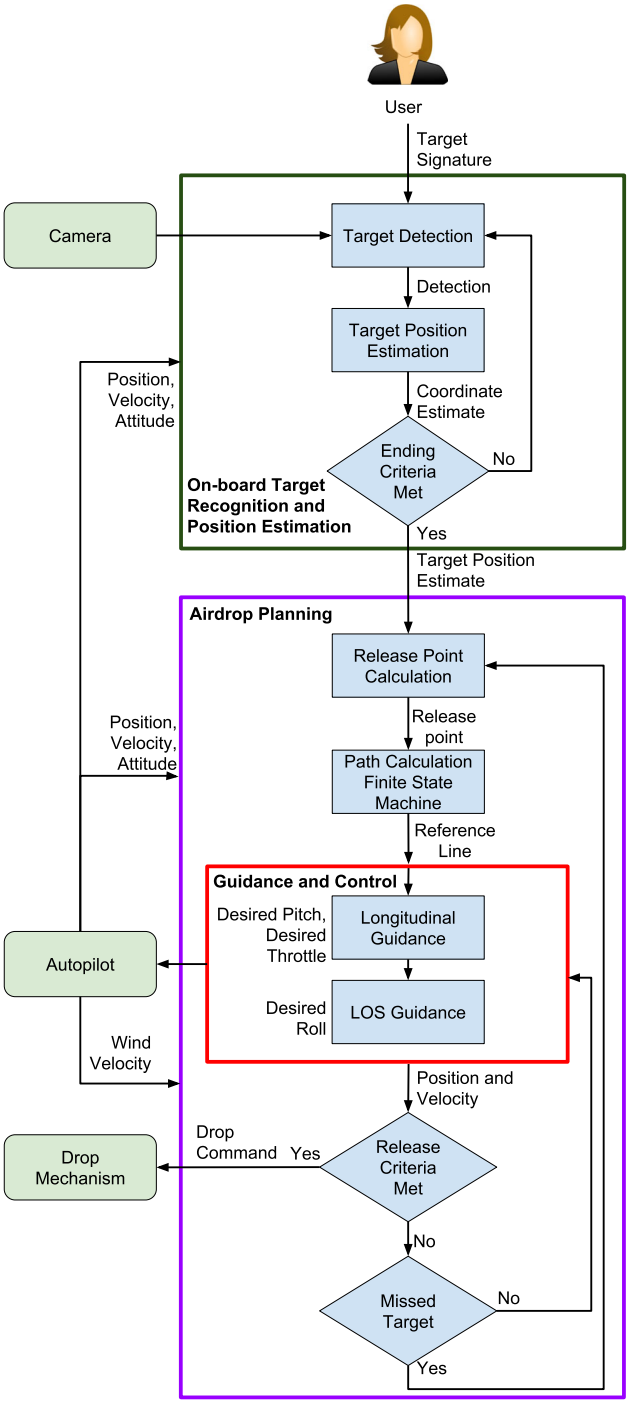

Fig. 1 Closed-loop on-board autonomous delivery system

and the guidance and control subsystem is the set of reference points. Moreover, the airdrop planning subsystem receives position and velocity estimates as feedback from the guidance and control system. This modular structure is beneficial when it comes to improving the system, as each subsystem can be developed separately as long as the interface is maintained.

Fig. 2 Raw image with the blue tarpaulin visible in the middle of the image

2.1 On-board target recognition and position estimation

This section describes the target detection algorithm and georeferencing. As a proof of concept, the target is chosen to be the blue tarpaulin displayed in Fig. 2. The target has a distinct colour which is easy to separate from the background in the visual spectrum. Therefore, the target detection algorithm utilizes colour segmentation, which is a simple and effective strategy in this particular case. However, it should be noted that any computer vision algorithm that segments an image into target and non-target regions could easily be integrated with the overall system described in this paper.

A pixel in a visual spectrum image is normally represented as a 3D-vector with three colour values, namely R (red), G (green) and B (blue). However, it is possible to represent pixels in different ways, and colour segmentation is usually easier to conduct in the so called HSV (hue, saturation, value) colour space. The HSV colour space separates colour information from lighting and reduces the impact from variations in lighting conditions. Thresholding techniques can, therefore, be conducted in a more reliable manner.

The main goal with colour segmentation is to create a binary image, which is separated into foreground (target) and background (no target). This is achieved through thresholding. After a new image is sent from the camera to the the on-board computer, the following steps are performed to search for targets:

1. Transform the colour space of the image from RGB to HSV.
2. Use thresholding to detect blue regions in the image that matches the target signature. In this work the detection is based on colour, but any visual characteristic(s) of the target could be used. This gives you a binary image where detected regions are white and other regions are black. Figure 3 shows the resulting image after this operation.
3. Search for contours in the binary image to find pixels that are connected and creates a region.

Fig. 3 The binary image after thresholding. The tarpaulin is easy to distinguish from the background

4. Calculate the area of all detected regions and only keep the detections with similar size as the target. Since the size of the target decreases with the altitude of the UAV, it can be necessary to adjust the size in real-time if the altitude varies much. This step is used to filter out noise and regions that only consists of a few pixels. Moreover, regions inside larger regions will also be removed with this approach. Various extensions with other features (e.g filter regions by shape) could also be used.
5. Calculate the pixel centre of the detection. If the centre is located a sufficient distance from the image border, it will be kept as a valid detection. Because of a large field of view, the horizon will be visible during turns with large roll angles, and detections too close to the border are discarded because they can be caused by the horizon or sky.

Figure 4 shows the resulting image where the image in Fig. 2 is processed through the steps above. The white circle is the detected centre of the tarpaulin and the red regions show the contours detected in the image. It is desirable to avoid false detections. Thus, it is necessary to use ranges for the thresholding operation that are strict and avoids contours with a colour that is somewhat similar. That is also the reason for why the entire tarpaulin is not marked in Fig. 4 since the blue colour is weaker near the boundary of the tarpaulin. Nevertheless, this is not restricting the accuracy of the detected position in the image. This is because the position is calculated as the centre of the detection so it is only necessary to detect a region that is centered around the true centre and not necessary to find the whole boundary of the tarpaulin. However, it is important to find the majority of the tarpaulin so it is not filtered out by the area requirement. In addition, missing the true centre with a few pixels may not be critical as it corresponds to a small distance in the Earth-fixed coordinate frame.

When a valid and accepted detection is located in an image, the next step is to use georeferencing to find Earth-fixed coordinates. Georeferencing is the transformation of image pixels into an Earth-fixed north-east-down (NED) reference system.

Fig. 4 The resulting processed image with the detection marked. The red areas mark the detected contours and the white circle marks the center of detection that is used for georeferencing. Notice that multiple red contours are displayed in the image, and that contours within contours are removed after this image is generated so that only the largest contour remains (Color figure online)

Given the pixel coordinates of the centroid of an object (u, v) and by assuming that the object is located at the ground, the object's position (North and East coordinate) in the NE frame can be found by the following equation (Leira et al. 2015)

$$\frac{1}{\lambda}\begin{bmatrix} N^{obj} \\ E^{obj} \\ 1 \end{bmatrix} = \boldsymbol{G}_{NE}^{-1}\boldsymbol{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{1}$$

where $\boldsymbol{G}_{NE}$ is the extrinsic camera parameter matrix (camera attitude and position) given by

$$\boldsymbol{G}_{NE} = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & -\boldsymbol{R}_{ned}^{cam}\boldsymbol{p}_{cam} \end{bmatrix} \tag{2}$$

$\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are the first and second column vector of $\boldsymbol{R}_{ned}^{cam}$, which is the rotation matrix for the rotation from the NED frame to the UAV's camera frame. $\boldsymbol{p}_{cam}$ is the position of the camera center given in NED frame coordinates. λ is a scaling factor required to isolate the normalized homogeneous coordinates of the North-East position of the object, $[N^{obj}, E^{obj}, 1]^T$. $\boldsymbol{K}$ is the intrinsic camera parameters, i.e the matrix transforming the normalized homogeneous pixel coordinates $[u, v, 1]$ from image coordinates (pixels) to camera coordinates (world units). $\boldsymbol{K}$ can be estimated by standard camera calibration algorithms (Helgesen et al. 2019).

Georeferencing will not be described in further detail since it is described more closely in several other articles (Leira et al. 2015; Helgesen et al. 2017; Rehak et al. 2013; Hemerly 2014). However, a few important considerations and challenges need to be addressed. Georeferencing utilizes a nonlinear transformation to find Earth-fixed coordinates. The transformation depends on the attitude and position of the camera, which again depends on the pose of the UAV.

Therefore, errors between the estimates in the navigation system and the true pose of the UAV will lead to errors in the georeferenced position. Moreover, mounting misalignment errors between the camera and the navigation system can also produce errors in the georeferenced position. The error in the georeferenced position is proportional to the altitude when errors in roll, pitch and yaw are considered. Therefore, it is important to have a reliable navigation system that is synchronized in time with the camera and calibrated to be careful when the camera and navigation system are mounted. Georeferencing also depends on the intrinsic matrix of the camera, so it is necessary to perform a camera calibration. To reconstruct 3D-coordinates in the Earth-fixed frame from a 2D-image, an assumption called the flat-earth assumption is used. With the flat-earth assumption, it is assumed that all pixels are located on the same height (form a planar surface) so that the coordinate indicating the height above the ground is known through the altitude of the UAV. This is described more closely in the aforementioned articles.

The georeferenced position from a single image can be somewhat inaccurate. Moreover, false detections are problematic if only a few measurements are used. Therefore, it is desirable to combine several measurements of the position to get a filtered estimate instead of a single measurement. It is possible to use a Kalman filter to estimate the position with a suitable motion model, and this corresponds to tracking the target in time. In this paper, the mean georeferenced position is used since this is a proof of concept and false measurements are not expected to be a large issue with the tailor-made detection algorithm. Moreover, the target is known to be stationary, but that may not be the case for other objects such as icebergs. Nevertheless, if false detections or a moving target can be expected, it is better to use a tracking filter to remove noisy measurements and outliers that fit poorly with the rest of the data. When the required amount of measurements are received, the mean position of the target is sent to the airdrop planning subsystem.

2.2 Ballistic model

To calculate the ballistic path of the falling object released from the UAV, the use of Newton's 2nd law, assuming only air resistance and gravity acting on the object, leads to the ballistic equations:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ -\frac{C_D \rho A}{2m} (v_x - w_x) V_r \\ -\frac{C_D \rho A}{2m} (v_y - w_y) V_r \\ g - \frac{C_D \rho A}{2m} (v_z - w_z) V_r \end{bmatrix} \tag{3}$$

$$V_r = \sqrt{(v_x - w_x)^2 + (v_y - w_y)^2 + (v_z - w_z)^2} \tag{4}$$

The vector $\mathbf{x}$ contains the position and velocity in the inertial reference frame, where x points in the north direction, y points in the east direction and z points in the down direction. V_r is the speed of the vehicle in the inertial frame relative to the air and w_x, w_y, w_z are the wind velocity components in the inertial frame. The constants C_D, m, A, ρ and g are the drag coefficient, mass and area of the released object, the air density and the standard acceleration due to gravity.

2.3 Airdrop planning

Once the on-board computer knows the coordinates of the target, it calculates a release position for the UAV. To calculate the release position, the ballistic equations for a falling object (3) are simulated from a start height equal to the release altitude, with a given release airspeed and a release direction opposite to the estimated wind direction. If there is no estimated wind, the release direction is set to be one that gives the shortest flight path for the UAV. Throughout the simulated drop, the wind is updated according to the wind profile power law (5) described in Touma (1977), where p is 1/7.

$$\begin{aligned} \frac{w_1}{w_2} &= \left(\frac{Z_1}{Z_2}\right)^p, \\ w_1 &= \text{wind at height } Z_1, \\ w_2 &= \text{wind at height } Z_2, \\ p &= \text{exponential parameter} \end{aligned} \tag{5}$$

The computation of the geographical coordinates of the release point is given in Algorithm 1:

Algorithm 1 Compute Release Point

```
releasePoint.lat = target.lat
releasePoint.lon = target.lon
z = release height
x = y = 0
while z > 0 do
    Update wind estimate
    Update airspeed V_r
    Simulate equation (3) 1 timestep
end while
Displace releasePoint with (-x,-y) meters in (north,east) direction
```

The ballistic equation (3) is simulated from the release height to zero heigh or ground level. Then the release coordinates are equal to the target coordinates, shifted $-$ x m in the north direction and $-$ y m in the east direction.

Once the release pose has been calculated, the on-board computer calculates a path to the release point, approaching it against the wind direction. This path is based on a Dubins path, which gives the optimal path from one directed point to another, using a selected turn radius as radius for the curves in the path (Beard and McLain 2012). The first step to create

the path is to construct a line through the release point in the same direction as the wind. At a given distance d, which would depend on the terrain and dynamics of the UAV, away from the release point at the same altitude, a point p is placed, see Fig. 5. Tangent to p, a circle with radius r is constructed. Its center s will be the first reference sent to the guidance and control subsystem, with the UAV's current position as starting reference for the line. The UAV is supposed to fly towards the point s, and upon approaching it, loiter around it in a clockwise direction. Once it reaches the point p, it will recalculate the release point and draw a new line through the release point, using the same direction for the line as originally intended. The reason for recalculating the release point is to get the current ground velocity, instead of calculating it based on the air velocity of the vehicle and the estimated wind velocity, as the ground velocity is dependent on the direction of the vehicle. From point p to the release point, the direction of the UAV is fixed.

The system is not using new updates of the wind velocity at this point. The reason for this is to handle small wind speeds as well, since the wind direction can vary a lot when there is little wind. When there is stronger wind, a small change in the wind will not affect the wind vector significantly, so the system is robust to changes in the wind. In turbulent wind conditions, the wind in point p can be subject to a sudden gust and turn out to be very different from the wind in the release point, and an update in point p would not give much improvement. A solution could have been to update the release point continuously, as suggested in Mathisen et al. (2017), but more focus here was given on creating a robust system and eliminating other sources of error, see Sect. 5.

The reference points for the line through the new release point are sent to the guidance and control subsystem, which follows the line using desired roll, speed and climb rate commands for the autopilot. When the UAV is within a given distance from the release point and starts moving away from the release point, a command is sent to the drop mechanism to make it release the object. When the object is released, the plan is ended. If the UAV starts moving away from the release point without being within the release proximity circle, the algorithm would be disrupted and a new release point would have to be calculated to restart the algorithm. This situation is named a *missed target* situation in Fig. 1.

2.4 Guidance and control

The guidance module will steer the UAV along the reference line against the wind, while keeping a constant, stable height. The advantage of using a LOS guidance law, as opposed to a more simplistic pure pursuit towards the release point, is that not only is the end destination of the flight decided, but also the route the UAV will take towards it. As the release point

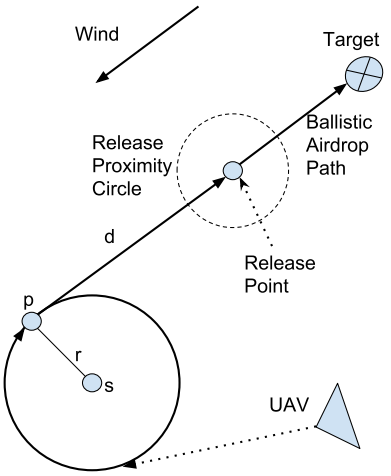

Fig. 5 The truncated Dubins path towards the release point

is dependent on the direction of the release velocity vector, this is imperative for the algorithm.

The implementation of the guidance law is based on Park et al. (2007), which finds a commanded lateral acceleration that brings the UAV closer to the desired path.

$$a_{s_{\text{cmd}}} = 2\frac{V_g^2}{L_1}\sin\eta, \tag{6}$$

where $V_g = \|v_x, v_y, v_z\|$ is the ground speed, L_1 is the tunable length of a vector, starting from the UAV, which intersects the desired path in front of the UAV, while η is the angle between the this vector and the velocity vector of the UAV, see Park et al. (2007, fig.1). The desired roll angle can then be found from:

$$\phi_d = \cos\theta \arctan\left(\frac{a_{s_{\text{cmd}}}}{g}\right), \tag{7}$$

where θ is the pitch angle.

Due to the sensitivity in the drop location to lateral motions of the UAV, the length d of the approach to the release point should be large enough for the transients in roll and cross-track error to be negligible. Further, the tuning of the lateral control should not be too aggressive. A small cross-track velocity at the time of release, applied to reduce a small cross-track error, could affect the drop target error more than the small cross-track error itself. Thus, it can be beneficial to choose a relatively large L_1.

The longitudinal controller is an cascaded LOS-based controller based on You et al. (2012). It consists of a height controller, which provides a desired climb rate $\dot{h}_d$ according to LOS theory and calculates a desired pitch and desired throttle based on the desired climb rate and desired speed.

The desired pitch θ_d is found from Nevstad (2016):

$$\theta_d = -c_1 (\gamma - \gamma_d) + \gamma_d + \alpha_{trim}, \tag{8}$$

where c_1 is a constant, γ is the flight path angle and γ_d is the desired flight path angle. α_{trim} is the trim angle of attack. The desired flight path angle or (negative) climb rate is found from Nevstad (2016):

$$\gamma_d = \gamma_p + \arctan\left(\frac{K_{ph} e_v + K_{ih} \int e_v}{\Delta}\right), \tag{9}$$

where γ_p is UAV's the rotation around the y-axis, K_{ph} and K_{ih} are the proportional and integral gains and Δ is the lookahead distance. The speed controller is found from Nevstad (2016):

$$\delta_{thr} = K_{p,v_a} \tilde{V}_a + K_{i,v_a} \int_0^t \tilde{V}_a (\tau)\, d\tau + \delta_{htr,trim} + K_{p,e_v} e_v (t), \tag{10}$$

where δ_{thr} is the desired throttle, K_{p,v_a} and K_{i,v_a} are proportional and integral gains, $\tilde{V}_a = V_a - V_{a,desired}$ and the last term is a proportional gain to minimum the vertical-track error e_v. In this paper, the UAV keeps a constant height and speed. For further insight into the longitudinal controller, see Nevstad (2016).

The input to the guidance and control system is a reference line from one waypoint to another. The autopilot performs the low-level control of roll, pitch and throttle using the Arduplane controllers, and provides estimated wind velocity along with position, velocity and attitude of the UAV.

3 Flight test

The subsystems of the closed-loop autonomous delivery system were tested in the spring of 2018. Separate tests of the target recognition and position estimation subsystem, the airdrop planning system and the guidance and control subsystem were carried out, before the whole system was tested. All tests were performed in wind no stronger than 10 m/s.

3.1 Unmanned aerial system architecture

The unmanned aerial system (UAS) that these experiments were performed on uses the X8 UAV from Skywalker Technology. This is a flying wing with a pusher propeller, where an electric motor has been used. The payload consists of a Pixhawk autopilot (Pixhawk 2018) with Ardupilot Flightstack (ArduPilot 2018) as firmware, a beaglebone black

Fig. 6 The UAV used in the case study, placed in its launcher

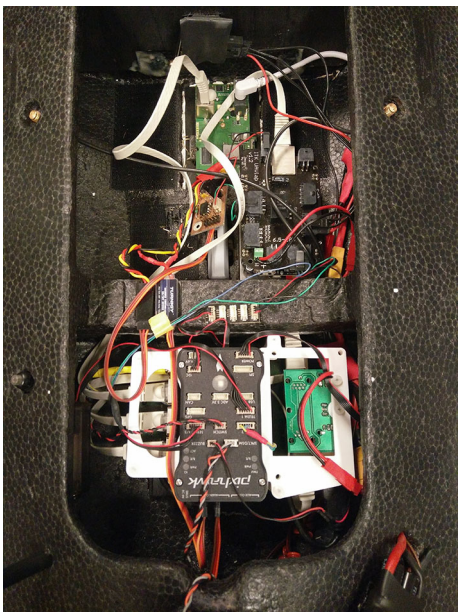

Fig. 7 The payload inside the UAV's fuselage

(beagleboard.org 2018) as payload computer where the custom airdrop system code is implemented on an AM335x, 1GHz ARM Cortex-A8 processor, an Axis M1025 network visual spectrum Camera mounted rigidly underneath the UAV fuselage, pointing directly downwards, with a maximum resolution of 1920 × 1080 pixels and a field of view of 94 degrees, a Ubiquity Rocket M5 5GHz communication link and an EFLA405 Servoless Payload Release, controlled by a PWM signal and holding the object to be released. To navigate, the system uses an external ublox NEO6 GPS (U-blox 2017). Ardupilot delivers wind estimates (ArduPilot 2018). The system architecture is described more thoroughly in Zolich et al. (2015), and the UAV and payload used in the case study are shown in Figs. 6 and 7.

The software on the payload computer uses the LSTS toolchain (LSTS 2015), which consists of the modular and open source embedded software DUNE, the inter-module communication protocol (IMC), the light-weight linux based operating system GLUED and ground station and control center Neptus. DUNE is the software used to implement the airdrop system. It consists of several modules performing

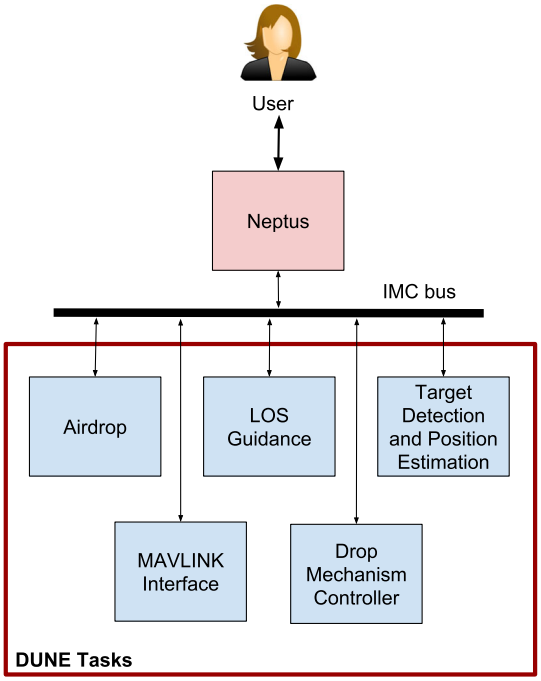

Fig. 8 The organization of the DUNE tasks used in this mission

their own task, like guidance, target detection or commanding the drop mechanism. These tasks communicate with each other using IMC messages, which are placed on a bus and available for any task that has subscribed to this particular message. A working set of these tasks are included in each mission by specifying them in a DUNE configuration file. When the software in the payload computer is started, the only tasks that are active are those decided in the configuration file. The user can plan the mission and observe the UAV's path and actions from the control centre Neptus. Neptus also communicates with IMC messages, and can access the same messages as the DUNE tasks. When the mission is over, all IMC messages and log data can be downloaded through Neptus' mission review and analysis centre. In the payload used in this mission, the DUNE tasks are organized as shown in Fig. 8. Control commands and navigation data are exchanged between DUNE and Ardupilot using the MAVLINK protocol. The LSTS toolchain is published in Pinto et al. (2012), Pinto et al. (2013). The image processing uses Open Source Computer Vision Library (OpenCV) (OpenCV 2018).

3.2 Adaptation to the UAS architecture

To adapt the closed-loop autonomous delivery system to the UAS architecture presented above, some tuning and modification were needed. The object to be released from the UAV is attached on the underside of the UAV's fuselage, see Fig. 9. The UAV used in these experiments has a propeller on the hind part, in which the released object could get entangled.

Fig. 9 The object to be released attached to the underside of the UAV's fuselage

Therefore, this could compromise the path of the released object and potentially destroy the propellers. To avoid this problem, the motor is turned off a short time before releasing the object, causing the propellers to fold. As this causes the UAV to lose speed, especially in strong wind, the release point is based on a ground speed that is lower than the UAV's speed in the point p, Fig. 5. Empirical tuning showed that a speed reduction compensation of 2.0 m/s was sufficient to produce a mean velocity error in the release point approximately at zero. This is elaborated in Sect. 5.2.

3.3 Dropped object

The object to be released in the case studies is a small disc carved out of PVC, see Fig. 10. The disc is approximated as a sphere during the fall, since it will spin around and there is no better way to approximate the surface area during a drop. The best guess for the surface area will therefore be the largest surface area of the disc, given by πr^2. The mass, air density and acceleration due to gravity are known. The drag coefficient C_D was identified through empirical tuning of the ballistic equation, and the tuning process is given in Table 1. The wind varied from around 8 m/s in the beginning of the testing to around 6 m/s at the end of the testing, at one point as low as 4 m/s, keeping the same direction. The model used for the ballistic equations in (3) is simple and may ignore other forces that can act on the object, assuming that they are negligible. Therefore, the along-track error in a drop result is attempted compensated for by adjusting the drag coefficient assumed in the model (3). If the calculations use a drag coefficient that is larger than the object's actual drag coefficient, then the calculated along-track drop path of the object will be shorter than the actual along-track drop path

Fig. 10 The object released from the UAV during the test cases

of the object. If the calculations use a drag coefficient that is smaller than the object's actual drag coefficient, then the calculated along-track drop path of the object will be longer than the actual along-track drop path of the object. Based on this, a drop where the object landed ahead of the target would call for a decrease of the drag coefficient estimate, while a drop not reaching the target would call for an increase of the drag coefficient estimate. The numerical parameters of the ballistic equations in (3) are given in Table 2.

3.4 Test procedure

This section describes the test procedure for the different subsystems.

3.4.1 Airdrop subsystem test and tuning

To test the airdrop subsystem only, the same procedure was followed both for tuning of the system and for repeating the tests with the accepted parameters. The UAV was launched manually and set to an automatic loiter controlled by the autopilot. Then the target coordinates were sent manually to the payload computer from the ground station, and the operation started. The UAV performed its manoeuvre as described in Sect. 2.3, before the pilot once again took control over the UAV and landed it manually.

3.4.2 Target recognition and position estimation

It was desirable to tune the target recognition and position estimation system in real-time. Therefore, images were sent to the ground station during flight so that different parameters could be adjusted online. This was important so that the tarpaulin could be detected effectively and limit the number of false detections. Moreover, the mean georeferenced position was also sent to the ground station so that the accuracy of the mean georeferenced position could be monitored. The raw image was downscaled to a resolution of 640×360 pixels to reduce the computational load and the following values for the thresholding operation were chosen during testing:

- Hue (has a range from 0 to 180)—values in the range from 100 to 110 kept as a possible target. This is where the blue colour is located in Hue.
- Saturation (has a range from 0 to 1)—values from 0.75 to 1 kept as a possible target. The saturation value models various shades of colour and a large value gives a bright colour with less gray shades.
- Value (has a range from 0 to 1)—values from 0.75 to 1 kept as a possible target. The value component gives the amount of black and white and a large value represents a larger amount of white.

Note that these parameters are tailored for detecting the tarpaulin robustly. The large saturation and value components are chosen since the tarpaulin has a bright blue colour without gray shades. The parameters need adjustment if targets with different properties are searched for.

If a pixel has values corresponding to the ones above and fulfills the intervals for all the channels (H, S and V), the corresponding pixel in the binary image is kept as one belonging to a target. Pixels outside of these intervals are not considered as a target and marked as background in the binary image. When searching for contours in the binary image, the largest contour with an area above 50 pixels is kept as possible detection. Moreover, if the detected centre position of the contour is more than 60 pixels from the horizontal image borders and 30 pixels from the vertical image borders, the detection is validated and sent to the georeferencing algorithm. The position acquired with georeferencing is used to update the mean position of the target.

3.4.3 Full system test

To test the entire system, a similar procedure as described above was followed: The target used for these tests was a 3 m $\times$ 5 m blue tarp, laid out on the ground, see Figs. 2 and 11. First, the UAV was launched manually, then the autopilot assumed control of the UAV and controlled it in a given search pattern. In this search pattern, the target recognition task detected and estimated the coordinates of the target continuously, until the ending criterion was met. 10 full system tests were conducted initially where the ending criterion was a manual verification of the estimated target position. Thus, the operator needed to approve the estimated target position before it was sent to the precision drop system. Typically 70–115 images with detections were used to get an accuracy corresponding to a few meters (see Fig. 15). The manual verification was used as a safety layer to ensure that the accuracy of the estimated target position was acceptable. Note that the estimated target position could not be changed or adjusted during the verification, but that the operator could prevent the precision drop module from starting in case of poor accuracy. To further illustrate that this can be done autonomously with-

Table 1 Drag coefficient tuning results

Drag coefficient	Error (m)	Comment	Wind speed (m/s)
0.9	22	Too long	8
0.9	14	Too long	8
0.75	21	Too long	9
0.75	12	Too long	7
0.6	17	Too long	8.5
0.6	13	Too long	8
0.4	6	Too long	6
0.4	10	Too long	8
0.3	1	Too long	8
0.3	9	Too short	10
0.3	20	Too long	8
0.3	2	Too long	8
0.3	22	Too long	9
0.3	3	Too long	6
0.25	11	Too long	6
0.25	6	Too short	7
0.25	3	Too long	7
0.25	10	Too long, skew	4
0.25	5	On line	6
0.25	13	Too long, skew	6
0.25	2	On line	6.6
0.25	8	Too long	8
0.25	3	Too long, skew	6
0.25	12	Too short	5.5

Table 2 Numerical values for ballistic equation

Property	Value
m	0.312 kg
ρ	1.246 kg/m^3
A	0.011304 m^2
C_D	0.25
g	9.81 m/s^2

Fig. 11 The tarp used as target in the full system test, laid out on the ground

out human interaction, a final test was executed. The ending criterion in this test was simply a check of the total number of detections and the estimated target position was automatically approved after 75 images with detections of the tarp. After 75 target detections, the mean georeferenced position was sent to the precision drop system, which then started automatically. Note that a more advanced strategy such as investigating the derivative of the mean position to find out if convergence is reached faster can be used together with a requirement for the minimum amount of detections. After the ending criterion was met, the target position coordinate estimates were sent to the airdrop planning system, and the UAV performed its maneuver as described in Sect. 2.3.

4 Results

The results of the case studies described in Sect. 3 are given here. All airdrop tests were performed at an altitude of 50 m, and the target recognition subsystem operated on an altitude of 100 m. The reason for using different altitudes was to operate each subsystem with optimal conditions, to optimize the results in this proof of concept research. For the target

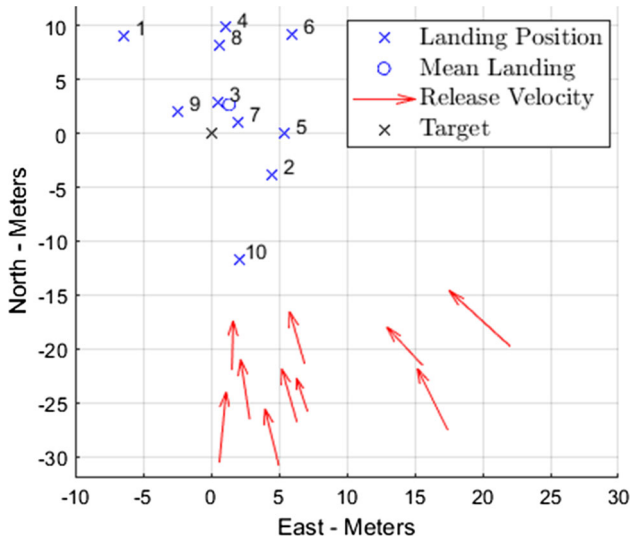

Fig. 12 Landing position from precision drop tests. The number on the crosses correspond to the numbers in Table 3

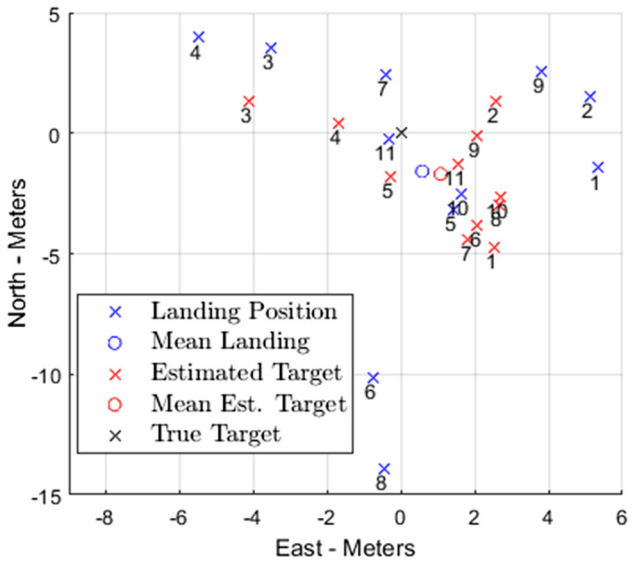

Fig. 13 Landing positions and target estimates from the full system test. The numbers correspond to the numbers in Table 4

recognition subsystem, the ground coverage is proportional to the altitude. However, increasing the ground coverage also increases the coverage of a single pixel. Therefore, an altitude of 100 m was decided to give a reasonable trade-off between ground coverage and detail level. The uncertainty of a drop increases with the altitude. Therefore, an altitude of 50 m was decided to be high enough to maintain safety, though still as low as possible. The commanded airspeed was 18 m/s.

4.1 Drop tests with known target

The results from the 10 airdrops performed without the target recognition system, i.e. with known target position, are shown in Fig. 12. These drops have a mean absolute error of 7.19 m with a standard deviation of 3.70 m, and the mean value of all drops is placed 3.01 m away from the target with a standard deviation of 7.81 m. In Table 3, the Target Drop Error, Release Point Error and Release Velocity Error of each drop are presented. Target Drop Error is the distance from the object's landing position to the target, the Release Point Error is the distance from the calculated release point to the position where the UAV actually released the object, and the Release Velocity Error is the velocity that the UAV had while releasing the object, subtracting the velocity that it was supposed to have while releasing the object. During these tests, the estimated wind varied between 4 and 8 m/s.

4.2 Drop tests with closed-loop autonomous delivery system

The full system with target recognition and drop was tested with 11 drops, and the results are shown in Fig. 13. These drops have a mean absolute error of 5.51 m relative to the true target position, with a standard deviation of 3.75 m, and 4.41 m relative to their corresponding estimated target, with a standard deviation of 3.05 m. The estimated targets have a mean error of 3.37 m relative to the real target with a standard deviation of 1.31 m. The mean value of all drops is 1.68 m away from the real target, with a standard deviation of 6.66 m, and 0.51 m away from their corresponding estimated target, with a standard deviation of 5.51 m. The mean value of all the target estimates is 2.01 m away from the real target, with a standard deviation of 3.13 m. In Table 4, the results are presented. The column Real Target Drop Error [m] is the delivered object position error relative to the real target, the column Estimated Target Drop Error [m] is the delivered object position error relative to the estimated target position and the column Target Estimation Error [m] is the error of the estimated target position (georeferencing) relative to the real target position. We see that coincidentally, the best hit relative to the real target was drop 11, which was the drop without human in the loop. During these tests, the wind varied between 1 and 5 m/s, where the 6 last drops had a different wind direction than the first 5, which means that the UAV approached the release point from a different angle (as it always releases its object against the wind).

The results of the target detection and position estimation system are displayed in Figs. 14 and 15. Each test corresponds to one curve in Fig. 15 and the error in the end is also stated as the *Target Estimation Error* in Table 4. Notice that for almost every test, convergence is reached after approximately 30-40 measurements so it may not be necessary with a larger amount of measurements. Notice also that one test is standing out from the rest because of a couple of false detections that moves the mean estimated position significantly.

Table 3 Precision drop system tests

#	Target drop error (m)	Release point error (m)	Release velocity error (v_x, v_y) (m/s)	Wind velocity at release point (w_x, w_y) (m/s)
1	11.06	2.13	(3.07, −1.48)	−4.39, 3.81
2	5.82	1.76	(−0.53, 0.55)	−6.97, 3.80
3	2.95	2.62	(0.21, 0.28)	−6.94, 1.34
4	9.91	1.71	(1.57, 1.21)	−4.51, 2.46
5	5.44	1.68	(1.02, 0.25)	−8.14, 0.97
6	10.99	1.19	(1.61, 0.96)	−5.65, −0.29
7	2.27	1.74	(1.64, −0.24)	−6.54, 0.69
8	8.27	3.21	(2.01, 0.03)	−7.29, 1.60
9	3.25	2.40	(−0.43, −0.22)	−6.17, 0.74
10	11.92	2.39	(−3.52, 0.66)	−6.14, 0.73

This is actually the first test and the thresholding values were changed to the ones listed in Sect. 3.4.2 after this test. These detections are caused by the blue horizon being detected during sharp turns and could be filtered out with a more advanced detection framework or by a tracking filter. If a tracking filter was used, consistency analysis could have revealed that the false measurements could not possibly be originated from the tarpaulin. However, after the adjustment of the thresholding ranges, no more false detections were experienced throughout the the 10 other tests.

The location of georeferenced points in Fig. 14 is interesting since they are distributed in a symmetric way around the true position (the origin), but very few detections are very close to the origin. This indicates that a systematic error is present in the navigation system which is canceled by observing the tarpaulin from different orientations and positions. A bias in the height may give this kind of distribution and is not unlikely considering the fact that the flat-earth assumption is used together with a single-frequency GPS receiver. In addition, since the mean georeferenced position is quite accurate when observing the target from different poses, it is not considered to be of great importance to compensate for the bias explicitly.

The tracking performance of the LOS Guidance system is shown in Fig. 16. Each test corresponds to one curve, which shows the UAV's distance from the tracked line of sight. As all of the lines tend to converge to zero, the tracking in the release point is fairly accurate.

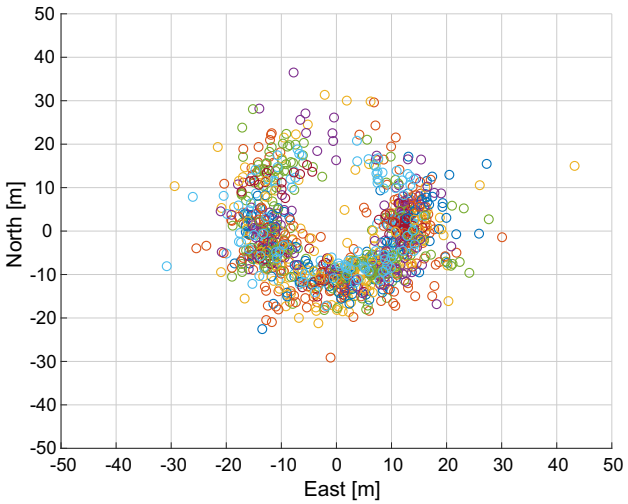

Fig. 14 Distribution of the georeferenced positions for all 11 closed-loop tests. The true position of the target is for illustration purposes in the origin of NED

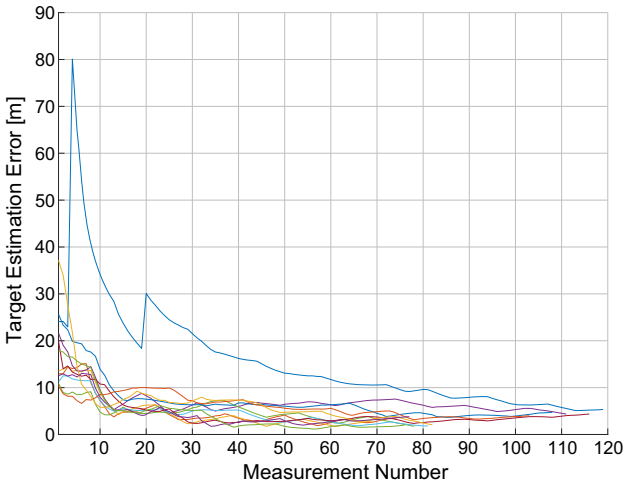

Fig. 15 Error in the mean georeferenced position as a function of the number of measurements in all closed-loop tests

5 Error analysis

The standard deviations in the positions of all delivered objects are 7.81 m for the tests without target identification and 5.51 m relative to the estimated target for the tests with closed-loop autonomous delivery system. These standard

Table 4 Closed-loop autonomous delivery system tests

#	Real target drop error (m)	Estimated target drop error (m)	Target estimation error (m)	Wind velocity at release point (w_x, w_y) (m/s)
1	5.53	4.32	5.34	−1.51, 1.10
2	5.32	2.56	2.88	0.13, 2.58
3	5.00	2.29	4.35	−0.97, 3.12
4	6.76	5.19	1.75	−1.02, 3.15
5	3.49	2.21	1.81	−1.60, −2.84
6	10.19	6.94	4.36	−3.27, −2.04
7	2.44	7.15	4.75	−1.75, −0.09
8	13.90	11.36	3.99	−3.56, −0.74
9	4.61	3.20	2.06	−2.96, 0.52
10	2.97	1.10	3.80	−2.04, −0.95
11	0.43	2.15	2.01	0.29, 2.06

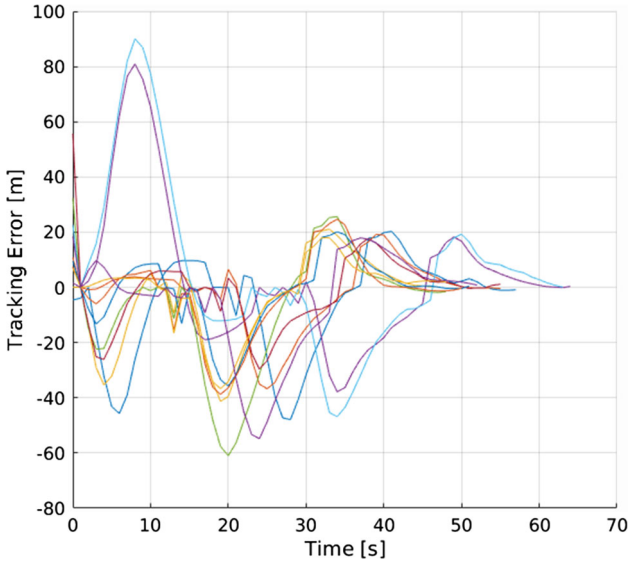

Fig. 16 Tracking performance of the LOS Guidance system as a function of the number of measurements in all closed-loop tests

deviations indicate that there remain errors and uncertainty to the system that is not controlled. It is possible to organize the possible sources of error into sources of error affecting the on-board target recognition and position estimation subsystem, sources of error affecting the airdrop planning subsystem, sources of error affecting the guidance and control subsystem, and sources of error affecting all three subsystems. In turn, the sources of error affecting the airdrop system can be divided into along-track errors and cross-track errors. The cross-track errors are the ones caused by oscillations in the guidance and control subsystem.

5.1 Sources of error affecting the on-board target recognition and position estimation subsystem

The main error source for the target recognition and position estimation system is related to the georeferencing algorithm. Attitude errors in the navigation system will have a large influence on the georeferenced position. Therefore, time synchronization between the camera and the navigation system is imperative and it is important that the navigation system is calibrated to be aligned with the true body axes of the UAV. It is also important to minimize the misalignment errors between the camera and the naviagation system so careful mounting is necessary. Bias in the inertial sensors in the navigation system can also be an issue. The error in the georeferenced position grows linearly with the altitude for navigation errors in the attitude. Thus, it is important to limit attitude errors as much as possible. If a constant bias is present in the system, it is desirable to design a search-pattern where the target is approached from different directions. This can contribute to cancel out the error. Approaching the target from the same position and with the same orientation will in general not remove the bias and will then give a constant error in the georeferenced position.

It can also be errors in the detection algorithm so that the detected pixel position is a few pixels away from the true centre. With an altitude of 100 m and image resolution of 640×360 pixels, a single pixel covers approximately 23 cm on the ground. Missing the centre with a few pixel is, therefore, a small error compared to a situation where e.g. the roll estimate is a couple degrees from the true roll angle. False detections can be an issue, as experienced in the first test, but was not encountered during the other tests because the detection algorithm was tuned to be strict and only find regions with a definite blue colour. As a consequence, the tarpaulin was not detected in several images where it was visible because of the small acceptable range in the thresholding operation. Nevertheless, missing the tarpaulin in some images was considered to be a safer option than getting false detections that could demolish the mean georeferenced position.

The dominating source of error affecting the on-board target recognition and position estimation subsystem is misalignment in the orientation of the camera with respect to NED (between the true orientation and the one given by the navigation system). This can be caused by estimation errors in the attitude given by the navigation system or that a constant bias is introduced during mounting of the camera and navigation system. Since the georeferencing error is proportional with the altitude for errors in the orientation, care should be taken during mounting. The error is limited by approaching the target from different directions since a constant bias will be mitigated by a varying path. The georeferencing error turns out to be a relatively small error, based on the estimates shown in Fig. 13, and the remaining error is most likely dominated by errors in the orientation of the camera.

5.2 Sources of error affecting the airdrop planning subsystem

One source of error that would effect the airdrop system is an incorrect wind estimator. However, Mathisen et al. (2017) shows that the target error due to 1 m/s increase in wind will be approximately 0.1 m. The same paper shows that the target error due to 1 m error in altitude is about 0.4 m, and that the target error due to 1 m/s error in the ground speed is about 3.4 m.

The main source of error affecting the along-track error of the precision drop system seems to be the release velocity error. It arises when the UAV does not have the ideal velocity at the release point, and is caused by a combination of the UAV's longitudinal speed controller not functioning perfectly, and the UAV losing speed when the motor is turned off. As the lost speed due to no motor power is expected, it is attempted compensated for, which is the main reason behind recalculating the release point in point p, Fig. 5. The UAV is commanded to fly with an airspeed of 18 m/s on the LOS through the release point, but the final release point is calculated based on the ground speed that the UAV has in point p, Fig. 5. From this ground speed, 2 m/s is subtracted and the ballistic path is calculated using this as the initial ground speed. With an approximately constant wind speed, the ground speed resulting from a commanded constant airspeed should be approximately constant as well. Still, there are some variations, which are presented as the release velocity error in Table 3. The mean value of the difference between the calculated release ground speed for the UAV and the actual release ground speed is $(0.26, -0.58)$ m/s, which means that a speed reduction constant of 2 m/s is close to correct. The size of the speed reduction constant will be dependent on the time without the motor turned on, before releasing the object, and the strength of the wind. When the UAV has a longer gliding time, the speed reduction will be larger, and for a shorter time, the speed reduction will be smaller. A stronger head wind will also give a larger speed reduction. A measure to reduce the mean release speed error could be to model this relationship instead of using a constant time with the motor turned off, and a constant speed reduction.

Comparing Table 3 and Fig. 12, a weak correlation between the release velocity error and target error can be indicated. We see that drop 2 and 10, which were too short, had release velocities that were smaller than they should have been, and drop 1 and 8, which were too long, had release velocities larger than they should have been. Drop number 3, 7 and 9 have the smallest target errors, and drop 3 and 9 have the smallest release velocity errors as well. Following the same train of thought, according to Fig. 12, drops 1, 3, 4, 6, 7, 8, 9 should have a release velocity where the x-component should be positive and drops 2 and 10 should have a release velocity where the x-component should be negative. Except for drop 9, this is true. Looking on the east-component of the release velocity error, drops 1 and 9 should have a negative y-component whereas drops 2, 5, 6 and 10 should have a positive y-component. This is true in all cases. For the rest of the drops, the effect of the different release velocity error components are not as pronounced.

Another possible source of error affecting the along-track error could be model mismatch. As mentioned in Sect. 3.2, the ballistic equations in (3) are simple, and it is possible that some effects are omitted. However, it is likely that these effects are small, and that the model mismatch comes from an incorrect drag coefficient. Even though the tuning led to a drag coefficient of 0.25, which was found sufficent, it is not perfect. Mathisen et al. (2017) shows that the effect of a faulty drag coefficient is between 0.1 and 0.4 m, and that the effect increases with increased speed.

The main source of along-track error affecting the airdrop planning subsystem is the release velocity error, which varies with the wind speed, but which is not modelled thoroughly. The relative importance of this source of error is quite large, and this error could profitably be minimized by work in the relationship between head wind and release speed reduction.

5.3 Sources of error affecting the guidance and control subsystem

Errors in the guidance and control subsystem will appear when the UAV is not able to follow a line correctly. This could be caused by the algorithm not being robust enough against the wind disturbances, because the algorithm is poorly tuned in general or against this particular UAV. The lookahead distance Δ mentioned in Sect. 2.4 was tuned in the initial airdrop planning subsystem tuning, to compensate for a more oscillating system.

The cross-track error of the dropped object is most likely caused by oscillations while trying to follow the LOS through the release point. Inspection of the path of the UAV during this part of the flight showed that it was not entirely straight, and moreover, that the roll of the UAV was not equal to zero. Tuning of the guidance control has attempted to avoid these errors, so they are minimal, but still present. The cross-track error of the LOS guidance can also be seen in the cross-track component of the release point error, see Table 3.

This is measured by the tracking performance, shown in Fig. 16. As we can see from the figure, the tracking error tends to zero in the release point. This is assumed to be a relatively small source of error.

5.4 General sources of error

As both the target recognition system and the precision drop system are dependent on an accurate positioning system, an error here will cause an error in both the georeferenced position and in the drop precision. An example of this is the GPS mean sea level (MSL) height, which changed every time the autopilot was rebooted. However, this altitude error was assumed constant throughout the test. The circular error probability (CEP) velocity and horizontal position accuracies of the autopilot's GPS receiver are 0.1 m/s and 2.5 m respectively (U-blox 2017), which means that the horizontal position inaccuracy will be dominating according to the error analysis in Mathisen et al. (2017). The RMS accuracy for the timepulse signal of the GPS is 20 ns (U-blox 2017), which is negligible, but there is also a time delay from when the GPS solution is valid to it gets to the computer. This is attempted compensated for by calculating the position of the UAV 0.1 s ahead, based on the current values for velocity and position.

These sources of error are not assumed to be relatively important, as they are constant and will presumably disappear when the system is tuned.

6 Conclusion

The experimental results show that the closed-loop autonomous delivery system can deliver objects from a small fixed-wing UAV with an expected target error distance of less than 6 m from 50 m altitude and 18 m/s speed, based on a machine vision based target estimate with an expected error of less than 4 m. The system has been proven to work in wind velocities ranging from 1 to 8 m/s. There are some known sources of error, where the release velocity error is the relatively most important ones. However, they have been attempted reduced during tuning. The intention of this research is to be used as a proof of concept, without focusing on eliminating all possible sources of error.

Acknowledgements Open Access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital). The authors are grateful for the flawless support from the excellent UAV operators Lars Semb and Pål Kvaløy.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

ArduPilot. (2018). Retrieved August 30, 2018 from www.ardupilot.org.

beagleboard.org. (2018). BeagleBone Black

Beard, R. W., & McLain, T. W. (2012). *Small unmanned aircraft: Theory and practice*. Princeton: Princeton University Press.

Benney, R., Barber, J., McGrath, J., McHugh, J., Noetscher, G., & Tavan, S. (2005). The new military applications of precision airdrop systems. In *Infotech at aerospace*.

Enderlin, E., Carrigan, C., Kochtitzky, W., Cuadros, A., Moon, T., & Hamiltona, G. (2018). Greenland iceberg melt variability from high-resolution satellite observations. *Cryosphere*, *12*(2), 565–575. https://doi.org/10.5194/tc-12-565-2018.

Gerlach, A. R., Manyam, S. G., & Doman, D. B. (2016). Precision airdrop transition altitude optimization via the one-in-a-set traveling salesman problem. In *American control conference (ACC)*.

Helgesen, H. H., Leira, F. S., Fossen, T. I., & Johansen, T. A. (2017). Tracking of ocean surface objects from unmanned aerial vehicles with a pan/tilt unit using a thermal camera. *Journal of Intelligent & Robotic Systems*,. https://doi.org/10.1007/s10846-017-0722-3.

Helgesen, H. H., Leira, F. S., Bryne, T. H., Albrektsen, S. M., & Johansen, T. A. (2019). Real-time georeferencing of thermal images using small fixed-wing uavs in maritime environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, *154*, 84–97. https://doi.org/10.1016/j.isprsjprs.2019.05.009.

Hemerly, E. M. (2014). Automatic georeferencing of images acquired by UAV's. *International Journal of Automation and Computing*, *11*(4), 347–352.

Joshua, M., & Eaton, A. N. (2013). Point of impact: Delivering mission essential supplies to the warfighter through the joint precision airdrop system. In *Systems conference (SysCon)*.

Klein, B., & Rogers, J. D. (2015). A probabilistic approach to unguided airdrop. In *Aerodynamic decelerator systems technology conferences*.

Kucheikoa, A. A., Ivanovb, A. Y., Davydovc, A. A., & Antonyuka, A. Y. (2016). Iceberg drifting and distribution in the vilkitsky strait studied by detailed satellite radar and optical images. *Izvestiya, Atmospheric and Oceanic Physics*, *52*(9), 1031–1040.

Leira, F. S., Johansen, T. A., & Fossen, T. I. (2017). A UAV ice tracking framework for autonomous sea ice management. In *International conference on unmanned aircraft systems (ICUAS)*. IEEE. https://doi.org/10.1109/ICUAS.2017.7991435

Leira, F. S., Trnka, K., Fossen, T. I., & Johansen, T. A. (2015). A ligthweight thermal camera payload with georeferencing capabilities

for small fixed-wing UAVs. In *The international conference on unmanned aircraft systems* (pp. 485–494).
LSTS. (2015). *LSTS Toolchain*. Retrieved June 25, 2015 from http://lsts.fe.up.pt/toolchain.
Mathisen, S. H., Grindheim V., & Johansen, T. A. (2017). Approach methods for autonomous precision aerial drop from a small unmanned aerial vehicle. In *IFAC-PapersOnline*. https://doi.org/10.1016/j.ifacol.2017.08.624.
McGill, P., Reisenbichler, K., Etchemendy, S., Dawe, T., & Hobson, B. (2011). Aerial surveys and tagging of free-drifting icebergs using an unmanned aerial vehicle (UAV). *Deep Sea Research Part II: Topical Studies in Oceanography*, *58*, 1318–1326.
Moctezuma-Flores, M., & Parmiggiani, F. (2017). Tracking of the iceberg created by the Nansen Ice Shelf collapse. *International Journal of Remote Sensing*, *38*(5), 1224–1234. https://doi.org/10.1080/01431161.2016.1275054.
Nevstad, S. O. (2016). *Autonomous landing of fixed-wing UAV in net suspended by multi-rotor UAVs*. Master's thesis, NTNU. Retrieved June 15, 2017 from http://hdl.handle.net/11250/2408392
OpenCV. (2018). Retrieved June 27, 2018 from www.opencv.org.
Park, S., Deyst, J., & How, J. (2007). Performance and Lyapunov stability of a nonlinear path-following guidance method. *Journal of Guidance, Control, and Dynamics*, *10*(2514/1), 28957.
Pinto, J., Calado, P., Braga, J., Dias, P., Martins, R., Marques, E., & Sousa, J. (2012). Implementation of a control architecture for networked vehicle systems. In *3rd IFAC workshop on navigation, guidance and control of underwater vehicles*.
Pinto, J., Dias, P. S., Martins, R., Fortuna, J., Marques, E., & Sousa, J. (2013). The LSTS toolchain for networked vehicle systems. In *OCEANS-Bergen, 2013 MTS/IEEE*.
Pixhawk. (2018). Retrieved June 27, 2018 from www.pixhawk.org.
Rehak, M., Mabillard, R., & Skaloud, J. (2013). A micro-UAV with the capability of direct georeferencing. *The International Archives of the Photogrammetry, Remote Sensing Spatial Information Sciences*, *40*, 317–323.
Schaffer, N., Copland, L., & Zdanowicz, C. (2017). Ice velocity changes on penny ice cap, baffin island, since the 1950s. *Journal of Glaciology*, *63*(240), 716–730. https://doi.org/10.1017/jog.2017.40.
Shi, L., Wang, Q., Zeng, T., & Liang, C. (2017). Chinese GF-1 and GF-3 satellites observed the giant iceberg calving off the Larsen C ice shelf. *Acta Oceanologica Sinica*, *36*, 115.
Sulak, D. J., Sutherland, D. A., Enderlin, E. M., Stearns, L. A., & Hamilton, G. S. (2017). Iceberg properties and distributions in three greenlandic fjords using satellite imagery. *Annals of Glaciology*, *58*(74), 92–106. https://doi.org/10.1017/aog.2017.5.
Tavan, S. (2006). Status and context of high altitude precision aerial delivery systems. In *AIAA guidance, navigation, and control conference and exhibit keystone*.
Touma, J. S. (1977). Dependence of the wind profile power law on stability for various locations. *Journal of the Air Pollution Control Association*, *27*(9), 863–866. https://doi.org/10.1080/00022470.1977.10470503.
U-blox. (2017). Neo-6 u-blox 6 GPS modules data sheet. https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf.
VanderMey, J. T., Doman, D. B., & Gerlach, A. R. (2015). Release point determination and dispersion reduction for ballistic airdrops. *Journal of Guidance, Control, and Dynamics*, *38*(11), 2227–2235.
Williams, P., & Trivailo, P. (2006). Cable-supported sliding payload deployment from a circling fixed-wing aircraft. *Journal of Aircraft*, *43*(5), 1567–1570.
Wright, R., Benney, R., & McHugh, J. (2005). Precision airdrop system. In *18th AIAA aerodynamic decelerator systems technology conference and seminar*.
You, D. I., Jung, Y. D., Cho, S. W., Shin, H. M., Lee, S. H., & Shim, D. H. (2012). A guidance and control law design for precision automatic take-off and landing of fixed-wing UAVs. In *AIAA guidance, navigation, and control conference*. https://doi.org/10.2514/6.2012-4674.
Zakharov, I., Power, D., Howell, M., & Warren, S. (2017). Improved detection of icebergs in sea ice with radarsat-2 polarimetric data. In *International geoscience and remote sensing symposium (IGARSS)* (pp. 2294–2297). IEEE. https://doi.org/10.1109/IGARSS.2017.8127448.
Zolich, A. P., Johansen, T. A., Cisek, K. P., & Klausen, K. (2015). *Unmanned aerial system architecture for maritime missions. design and hardware description*. Workshop on research, education and development of unmanned aerial systems (RED_ UAS).

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

Siri Gulaker Mathisen Received her M.Sc. degree in Engineering Cybernetics in 2014 at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. She is currently working on her Ph.D. degree at the Department of Engineering Cybernetics, NTNU, specializing in non-linear mission planning and guidance for autonomous UAVs. From 2014, she has been working on precision drops from small UAVs since 2014, in parallel with work on non-linear model predictive control (MPC) for guidance of a small UAV in a deep stall. Her main research interests include non-linear model predictive control and deep stall of UAVs.

Frederik Stendahl Leira (M.Sc., Ph.D.) received his M.Sc. degree (specializing in robotics and computer vision) and his Ph.D. degree (in the topic Object Detection and Tracking with UAVs) in 2013 and 2017, both in Engineering Cybernetics at the Norwegian University of Science and Technology. He is currently a Post-Doctoral Fellow within the Center of Excellence on Autonomous Marine Operations and Systems (AMOS), where he continuous to do research on target detection and tracking utilizing UAVs, computer vision and model predictive control for path planning.

Håkon Hagen Helgesen Received the M.Sc. degree in 2015 in Engineering Cybernetics from the Norwegian University of Science and Technology (NTNU), Trondheim. In 2015, he started as a Ph.D. candidate at the NTNU Centre for Autonomous Marine Operations and Systems and at the Department of Engineering Cybernetics. He has been working with topics related to navigation, machine vision and target tracking for unmanned aerial vehicles. His current Ph.D. topic is detection and tracking of floating objects at sea.

Kristoffer Gryte Received his M.Sc. in Engineering Cybernetics in 2015 at the Norwegian University of Science and Technology. His master was within guidance, navigation and control of vehicles, where the master thesis focused on modeling and landing of a fixed wing Unmanned Aerial Vehicle. He is currently continuing this work into a Ph.D. related to UAV technology, focusing on autonomous take-off and landing of fixed-wing UAVs. This work is supervised by Professor Thor Inge Fossen and Professor Tor Arne Johansen.

Tor Arne Johansen Received the M.Sc. degree in 1989 and the Ph.D. degree in 1994, in both electrical and computer engineering, from the Norwegian University of Science and Technology, Trondheim, Norway. From 1995 to 1997, he worked at SINTEF as a researcher before he was appointed Associated Professor at the Norwegian University of Science and Technology in Trondheim in 1997 and Professor in 2001. He has published several hundred articles in the areas of control, estimation and optimization with applications in the marine, aerospace, automotive, biomedical and process industries. In 2002, Johansen co-founded the company Marine Cybernetics AS where he was Vice President until 2008. Prof. Johansen received the 2006 Arch T. Colwell Merit Award of the SAE, and is currently a principal researcher within the Center of Excellence on Autonomous Marine Operations and Systems (NTNU-AMOS) and director of the Unmanned Aerial Vehicle Laboratory at NTNU. In 2017, he co-founded the spin-off companies Scout Drone Inspection AS and UBIQ Aerospace AS. He is currently the leader of the SmallSat Lab at NTNU.