UAV Cooperative Perception for Target Detection and Tracking in Maritime Environment

Guilherme Amaral, Hugo Silva, Flávio Lopes, João Pedro Ribeiro,
Sara Freitas, Carlos Almeida, Alfredo Martins, José Almeida, Eduardo Silva

INESC TEC Institute for Systems and Computer Engineering of Porto
ISEP - School of Engineering, Porto Polytechnic Institute, Porto, Portugal
Email: gasilva, hugo.m.silva, fmlopes, jmpr, scf, carlos.almeida, aom, jose.m.almeida, eduardo.silva @inesctec.pt

***Abstract*—This paper addresses the topic of target detection and tracking using a team of UAVs for maritime border surveillance. We present a novel method on how to integrate the perception into the control loop using two distinct teams of UAVs that are cooperatively tracking the same target. We demonstrate and evaluate the effectiveness of our approach in a simulation environment.**

***Index Terms*—Target Detection, Target Tracking, Cooperative Perception.**

I. Introduction

The use of Unmanned Aerial Systems (UAS) for maritime surveillance is becoming evermore frequent. It will not be long until the majority of the aerial maritime surveillance flights are performed by unmanned platforms. The European Union (EU) is facing increasing challenges in patrolling its maritime borders and novel solutions are being develop to replace the regular manned patrol flights (MPA). One of this solutions is being develop within the SUNNY (*Smart UNmanned aerial vehicle sensor Network for detection of border crossing and irregular entrY*) Project [1].

The SUNNY project aims to develop and integrate airborne sensors to cover large areas of maritime border. To do so, it uses a two level Unmanned Aerial Vehicle (UAV) sensor network. The first-level UAVs (L1) fly at high altitude in order to provide wide area coverage, while the second-level UAVs (L2) fly at low altitude and provide a more focused attention on the target. The main idea of such concept is that the first-level UAVs, are used for patrolling large border areas in order to detect suspicious targets and provide global situation awareness. Fed with the information collected by the first-level UAVs, the second level UAVs, will be deploy to provide a more focused surveillance capability by tracking the targets and collecting further evidence for a more accurate target recognition and threat evaluation.

The second-level UAVs have the capability to detect, track and identify suspicious targets, and can perform this task using two different approaches: First, the UAVs can be equipped with similar sensors, which implies autonomous coordination between them, this approach is used to optimise area coverage; Second, using heterogeneous sensors perform coordinate sensing and exploit the data collected by different sensors carried in each UAV to improve the coverage effectiveness.

Despite the intended approach or type of sensor used, automatic target detection and tracking in maritime scenario with UAV is a challenging problem due to the environment nature. Some features such as sunlight reflection over the sea surface, strong winds, or the possibility of occlusions due to wave-crest combined with target and UAV motion, poses difficulties for autonomous detecting and tracking operations.

Concerning vision based target detection and tracking, the water movement is one of the factors that cause failures in algorithms based on the background subtraction [2]. To avoid such concerns its common to use thermal cameras in maritime scenario. Since this type of sensor is more robust to changes in the lighting conditions, and does not capture the sunlight reflection over the sea surface or over the vehicle and also even not complety is also more immune to the influence of white foam [3], [4]. However, these cameras present disadvantages such as the low number of features that can be extracted, when compared to Electro-Optical cameras and also a higher energy consumption.

One way of overcoming such limitations is to use a team of UAVs, behaving in a coordinated fashion. Multiple vehicles offer the advantage of large number of observations at various angles resulting in increased accuracy for tracking and mapping. In the literature, depending on the probability distribution, some approaches are considered such as Bayes Filters [5], Particle Filters [6], or Kalman/Information Filters [7], [8].

In formation control literature, two different types of formation control structures are referenced: Hierarchical and Non-hierarchical. The former comprehends formations wherein the control tasks are not equal for all agents. A common case is the leader-follower configuration [9], [10], [11], [12], [13], [14], [15], [16], in which the leader has the goal of navigating or tracking something and the followers have the goal of follow the leader. In non-hierarchical formations [17], [18] the goal is common for all agents, i.e., to follow a target. The control distance between the formation is also subject of intense research. A formation is said to be symmetric, if exists a joint effort of both agents A and B simultaneously and they try to actively keep the distance between them [11], [12], [13]. A formation will be considered as asymmetric, if only one of the agents in each neighbour agent pair actively maintains its position relatively to its neighbour [9], [10], [14], [15], [16],

978-1-5090-5278-3/17/$31.00 ©2017 IEEE

[17], [18].

In this paper we propose a cooperative system for target detection and tracking using UAVs in maritime environment, using L1 and L2 concept. Simulation results based on the Modular OpenRobots Simulation Engine (MORSE) [19] and the Robot Operating System (ROS) [20], are presented. The aerial vehicles are equipped with optical sensors for surveying, detecting and tracking the target. The L1 UAV vehicle is also equipped with a gimbal unit in order to track the target for long periods of time, and exchange the target position with the L2 UAVs.

The paper outline is as follows: in the next section, we will briefly describe our simulation environment. In section III, we describe our ROS based system architecture, followed by the results in section IV. Finally, we present some conclusions and discuss our future work in the remaining section.

II. Simulation Environment

The chosen simulator was MORSE. It is a powerful simulator that runs on the Blender Game Engine and interfaces a 3D environment with external robotics software. It is develop in Python language, and holds sensors that are fully supported by the middleware and can provide data as their real world counterparts allowing also the choice of different levels of realism and abstraction. MORSE proved to be very versatile, modular, flexible and reusable [21]. The integration between ROS and MORSE allows us to test the possibility of real-time vehicles behaviour visualization. Considering our maritime environment as scenario, some cooperative tracking approaches simulations were performed, as well as UAV formation/coordination strategies. Regarding the simulation platforms we considered one rubber boat (Figure 1a) and two UAVs, one for being our first level UAV and another for simulating a second level UAV. The first level UAV, which operates higher, comprehends a fixed wing UAV (ANTEX) (Figure 1c). The L2 is composed by one rotary wing UAV - a generic quadrotor (Figure 1b).

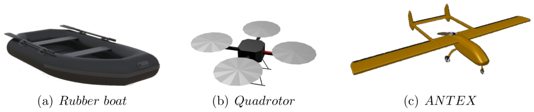

(a) *Rubber boat* (b) *Quadrotor* (c) *ANTEX*

Fig. 1. Used vehicles 3D models

The vehicles and the cameras are illustrated in Figure 2. The XYZ axis corresponds to the red, green and cyan colours respectively. The ANTEX (A), the Rubber boat (B) and the Quadrotor (Q) reference frames are concordant with the World (W) reference frame - ENU (*East North Up*) coordinates. The downward cameras applied to the aerial vehicles, also share the same reference frame orientation: Ac for the ANTEX camera and Qc for the Quadrotor camera. The Pan&Tilt Unit reference frame is marked as Ag.

A. Image Processing

For each received frame, image is converted to OpenCV image format, in BGR colour map and then processed. As first step, for image segmentation purpose, colour map space is changed to HSV. Based on an adaptive threshold method, a binary image is created with highlighted white target shape in a black background. Resorting to OpenCV library functions, image blobs are extracted and by filtering results, using density area information target contours are extracted. The blob centre of mass is considered the observed target coordinates, related to image reference frame (${}^{I}P_{obs}$).

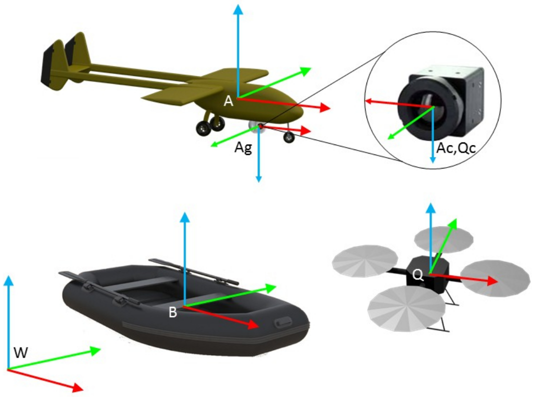


Fig. 2. Simulation reference frames

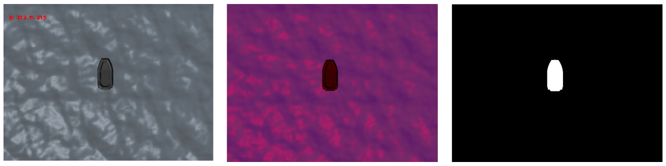


Fig. 3. Example of BGR camera image (with some annotations), and the respective HSV and binary image

In order to obtain the vehicle coordinates in the world reference frame, based on the camera/image processing measurements, a pinhole camera geometry model is considered (see Figure 4). The conversion between the image reference frame and the camera reference frame is achieved using the camera intrinsic parameters, as defined in equation 1:

$$C = \begin{bmatrix} 0 & f_x & O_x \\ -f_y & 0 & O_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where O_x and O_y defines the optical centre ($o.c.$) and the parameters fx and f_y defines the focal distance (f).

The reference frames follow the right-hand rule procedure. The camera reference frame has its origin at the camera centre and is equal to the gimbal reference frame which has its origin in the intersection of its Pan&Tilt axis (perpendicular among them). The inertial reference frame has its x axis pointed to the UAV front and the y axis pointed towards the left wing. The UAV reference frame has its origins coincident with the vehicle centre of mass and respects the ENU reference frame with x axis pointed to the front of the UAV.

Based on this, we can relate all determined points in the image reference frame to the world reference frame, as defined

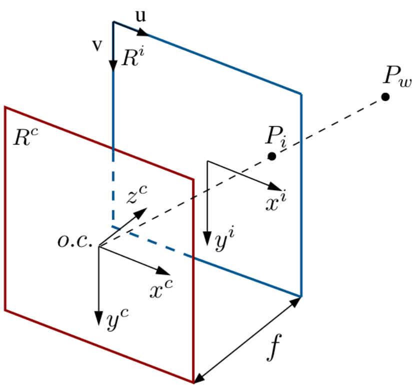


Fig. 4. Pin-hole camera geometry schematic

in equation 2.

$$ {}^{W}P_{obs}(t) = {}^{W}_{B}R \cdot {}^{B}_{G}R \cdot {}^{G}_{C}R \cdot {}^{C}_{I}R \cdot {}^{I}P_{obs}(t) \tag{2} $$

Considering a point ${}^{I}P_{obs}(t)$ related to the image reference frame, at instant t. The transformation of its reference frame for the camera reference frame is achieved by the multiplication of the coordinates by the ${}^{C}_{I}R$ matrix. The previous outcome (${}^{C}P_{obs}(t)$) multiplied by the ${}^{G}_{C}R$ matrix will result in the considered point expressed in the gimbal reference frame (${}^{G}P_{obs}(t)$).

The target coordinates, related to the image reference frame, are moved to the camera reference frame using the camera intrinsic parameters. An homogeneous vector is thus defined as shown in equation 3.

$$ {}^{C}P(t) = \begin{bmatrix} \frac{{}^{I}P_x(t) - O_x}{f_x} \\ \frac{{}^{I}P_y(t) - O_y}{f_y} \\ 1 \\ 1 \end{bmatrix} \tag{3} $$

The matrix that relates the target coordinates in the camera reference frame with the gimbal reference frame is shown in equation 4.

$$ {}^{G}_{C}R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

Being the points expressed in the gimbal reference frame, the gimbal Pan&Tilt angles should be considered. The gimbal matrix G, presented in equation 5, affects the target coordinates by the gimbal angles. The *pan* and *tilt* angles are represented as α and φ respectively. The $c()$ stands the cossine function and $s()$ stands the sine function.

$$ G = \begin{bmatrix} c_\alpha c_\varphi & -s_\alpha & c_\varphi & 0 \\ c_\varphi s_\alpha & c_\alpha & s_\alpha s_\varphi & 0 \\ -s_\varphi & 0 & c_\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5} $$

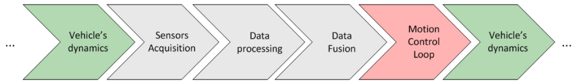


Fig. 5. Sensing Network Flow

The transformation of point coordinates expressed in the gimbal reference frame to body reference frame is achieved through matrix ${}^{B}_{G}R$ according to equation 6.

$$ {}^{B}_{G}R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6} $$

Finally, the UAV attitude is considered.

$$ {}^{W}_{B}R = \begin{bmatrix} c_\theta c_\psi & c_\psi s_\phi s_\theta - c_\phi s_\psi & s_\phi s_\psi + c_\psi s_\theta c_\phi & 0 \\ c_\theta s_\psi & c_\psi c_\phi + s_\psi s_\phi s_\theta & c_\phi s_\psi s_\theta - c_\psi s_\phi & 0 \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7} $$

B. Vehicles Control

The vehicles motion is controlled through inputs references imposed by the control ROS nodes. For the fixed-wing vehicles, the MORSE offers a position control based on the definition of a linear and angular speeds. Linear speed is imposed in x, y, or z axis and angular speeds around roll (ϕ), pitch(θ) or yaw(ψ) angles.

For the rotary wing vehicle, we set a control ROS node with private configuration parameters, such as: vehicle role for formation control algorithm or position control gains. Independent from the desired formation geometry, all aerial vehicles have, at least, one vehicle to follow.

III. System Architecture

Our system architecture, is distributed in software blocks developed under ROS nodes. This solution not only allows a high level of code organization but supports scalability. A new node instance can be launched whenever necessary, i.e. L1 UAV motion node is common for all evolved agents, changing only some parameters. Figure 5 illustrates the flow (from left to right) of the system architecture. Grey blocks are related to the perception layer, the red block represents the motion control node and the green block is the vehicle's dynamics.

Each sensor has a processing node that reads raw data, process it and share useful information with other agents. In some cases a data fusion node is required to combine heterogeneous data and/or from heterogeneous vehicles. The dissemination is done through the publication of ROS topics.

Assuming that all vehicles are equipped with an autopilot with waypoints navigation feature, we need to develop a motion control node (one per vehicle). This node generates the desired vehicle position along the mission. The actual position and attitude are two fundamental inputs that need to be subscribed. The publication of a ROS topic with the 3D position is expected from the control loop process. To avoid

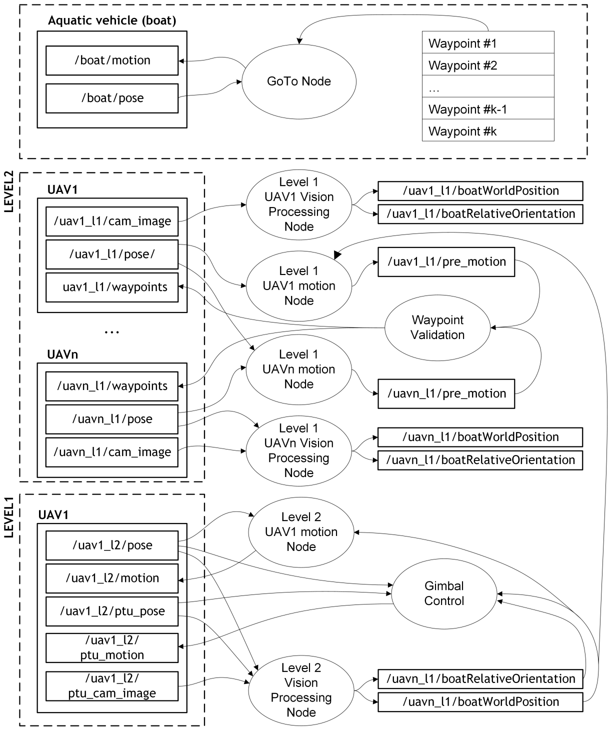


Fig. 6. ROS nodes architecture

collisions, an obstacle avoidance node is considered and has as inputs all generated waypoints. The collision avoidance node identifies dangerous paths or not normal actions, and performs path changes ensuring the system viability. Independently of modifications, the accepted waypoints or new ones that are computed on the fly are published in the respective vehicle waypoint topic affecting the vehicle dynamics.

Figure 6 displays the interconnections between ROS nodes, as well as, the respective publish/subscribed topics.

Each vehicle has two topics associated: The pose topic, which is used by the simulator framework to publish the vehicle attitude and position and the motion topic that is used by the other nodes to publish the desired control input references of the vehicle.

The aquatic vehicle has both available topics connected to a *GoTo* node. It has as input a circular FIFO buffer of waypoints to be followed and the subscribed boat pose. The implemented machine state generates linear velocities as output and publish it to the boat's motion topic.

Both first-level and second-level UAVs, for each optical sensor, have an image processing node. It has as input the respectively UAV pose and the image sensor output. The output of those nodes is the boat position, related to world reference frame, as well as the boat bearing. For the case of the L1 UAV, which carries a gimbal unit, the gimbal pose is also subscribed and used as node input since it must be considered in the boat position determination process. Once the position of the boat has been determinate, it is used as input for the second-level UAVs motion control nodes.

The gimbal control node runs on a non-finite machine state that drives the gimbal behavior depending the target detection. If a target is not detected in the sensor image output, a scan manoeuvre is performed sweeping a pre-defined interval of the pan angle. Otherwise, the node generates desired pan and tilt angles in order to track the target.

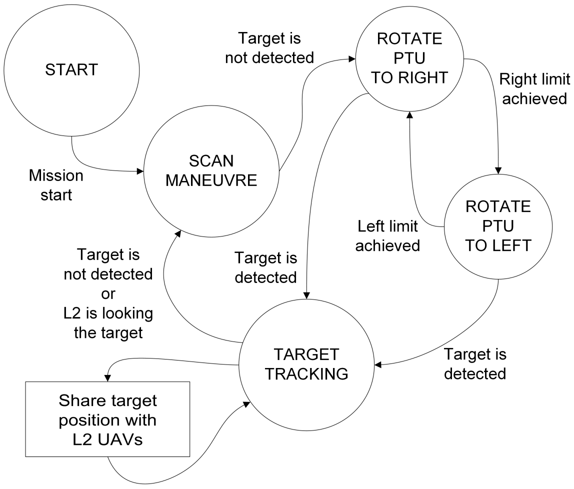


Fig. 7. Patrolling algorithm machine state

The L2 UAV, has a motion node responsible to guide the surveillance manoeuvre, as well as the UAV path wherever a target is detected. This node has as input the L2 UAV pose and the boat position determined by the image processing node. The UAV path is defined based on the node output (velocities references) which is published in the L2 UAV motion topic.

IV. Results

In the present simulation, while L1 UAV is patrolling a pre-defined zone, L2 UAV is waiting for instructions. The patrolling manoeuvre consists in the following of an (almost) oval path, whilst an image scan is performed using a Pan&Tilt Unit (PTU). The target detection event triggers the PTU control algorithm to maintain the target in the centre of the image (by using the target world position provided by the image processing node), see Figure 9.

Whenever the target is not detected the scan mode is applied. The PTU scans the zone delimited by the pre-defined angular limits. If a target is detected, the target tracking process starts. The target position, determined by image processing node, is shared with L2 UAV. A question pop-up window is launched in the simulation control graphical user interface for target human validation. While answer is not submitted, L1 UAV performs loitering manoeuvre to avoid losing the target from its field of view. If target is confirmed by the human, L2 UAV starts to follow the target. Only when visual contact with the target is established, L1 UAV leaves the loitering pattern and continues the search task of patrolling the surveillance area. When L2 UAV achieves a stationary flight above the target (e.g. idle position), then it starts the loitering manoeuvre. If target starts moving, it is followed by L2 UAV that backs to loitering manoeuvre if target is in idle position. Therefore validating the mixed-initiative target evidence.

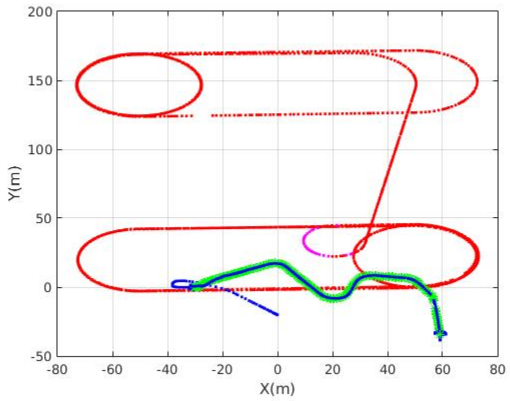


Fig. 8. Example of 2D vehicles coordinates in simulation scenario

Figure 8 shows the vehicles position along the simulation. Red points represent the absolute L1 UAV position. When target is observed these points are in magenta colour. Target position is displayed in green and L2 UAV in blue colour.

Initially, L1 UAV has performed a scan manoeuvre following circular and oval paths. As it is shown in Figure 9, at time instant 125 s, the UAV has completed a full turn without observing the target. When it passes nearer, the PTU was pointed towards an angle that did not allow the target to appear in the camera field of view. Only at time instant 225 s the first visual contact was occurred. L1 UAV has started to perform a loitering manoeuvre and an alert signal was launched in the simulation control GUI. The user has confirmed the target few moments before time instant 231 s, and the L2 vehicle has started to follow the target while L1 UAV continues its scan manoeuvre.

V. Conclusions and Future Work

A simulation of SUNNY concept scenario was successfully performed. A fix wing UAV, acting as member of L1, has detected a target ”boat” in the water during a patrol turn and started to perform the loitering maneuver. Meanwhile, an end-user received an alert and started to analyse the received event. After having confirmed the target as threat, the L2 UAV was launched and start tracking the boat. The L2 also started the loitering maneuver when the stationary flight above the target was reached. As soon as L2 UAV assumed the target tracking, the L1 UAV continues its wide surveillance mission.

In future work, we plan to test this sequence of events using real UAVs in flight with L1/L2 altitudes, performing coordinative perception of the same target using heterogeneous sensors (radar and visible/infrared image information) in the detection and tracking of a target.

Acknowledgments

This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, by the SUNNY Project funded by the European Commission under the FP7 programme Grant Agreement number: 313243, and by National Funds through the FCT - Fundacao para a Ciencia e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

[1] S. Project, “SUNNY Project webpage.” [Online]. Available: http://www.sunnyproject.eu/

[2] D. Frost and J.-R. Tapamo, “Detection and tracking of moving objects in a maritime environment using level set with shape priors,” *Journal on Image and Video Processing*, pp. 2013–2042, 2013.

[3] L. O. Alvares, A. M. Loy, C. Renso, and V. Bogorny, “An algorithm to identify avoidance behavior in moving object trajectories,” *Journal of the Brazilian Computer Society*, vol. 17, no. 3, pp. 193–203, 2011.

[4] F. Robert-Inacio, A. Raybaud, and É. Clément, “Multispectral Target Detection and Tracking for Seaport Video Surveillance,” *Journal on Image and Video Computing*, no. December, pp. 169–174, 2007.

[5] E. M. Wong, F. Bourgault, and T. Furukawa, “Multi-vehicle Bayesian search for multiple lost targets,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, no. April 2005, pp. 3169–3174, 2005.

[6] L.-l. Ong, L.-l. Ong, B. Upcroft, B. Upcroft, T. Bailey, T. Bailey, M. Ridley, M. Ridley, S. Sukkarieh, S. Sukkarieh, H. Durrant-whyte, and H. Durrant-whyte, “A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles,” *October*, pp. 4539–4544, 2006.

[7] D. Y. Kim and M. Jeon, “Data fusion of radar and image measurements for multi-object tracking via Kalman filtering,” *Information Sciences*, vol. 278, pp. 641–652, 2014.

[8] F. Morbidi and G. L. Mariottini, “On active target tracking and cooperative localization for multiple aerial vehicles,” *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2229–2234, 2011.

[9] A. Sutton, B. Fidan, and D. V. D. Walle, “Hierarchical UAV formation control for cooperative surveillance,” *International Federation of Automatic Control*, no. October, pp. 12 087–12 092, 2008.

[10] J. How and E. King, “Flight Demonstrations of Cooperative Control for UAV Teams,” *Architecture*, vol. 1, no. September, pp. 505–513, 2004.

[11] J. Moon and R. Sattigeri, “Adaptive guidance and control for autonomous formation flight,” *American Helicopter Society 63rd Annual Forum*, 2007.

[12] G. Vasarhelyi, C. Viragh, G. Somorjai, N. Tarcai, T. Szorenyi, R. Nepusz, and T. Vicsek, “Outdoor flocking and formation flight with autonomous aerial robots,” in *Proceedings of IROS 2004, the IEEE International Conference on Intelligent Robots and Systems*, 2014.

[13] Y. Gu, G. Campa, B. Seanor, S. Gururajan, and M. R. Napolitano, “Autonomous Formation Flight - Design and Experiments,” *Aerial Vehicles*, pp. 233–256, 2009.

[14] Z. X. Liu, X. Yu, C. Yuan, and Y. M. Zhang, “Leader-Follower Formation Control of Unmanned Aerial Vehicles with Fault Tolerant and Collision Avoidance Capabilities,” in *Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 1025–1030.

[15] Z. Hou and I. Fantoni, “Distributed Leader-Follower Formation Control for Multiple Quadrotors with Weighted Topology,” in *System of Systems Engineering Conference (SoSE), 10th*, 2015, pp. 256–261.

[16] H. Ribeiro, “Formation Control for Unmanned Aerial Vehicles,” 2012.

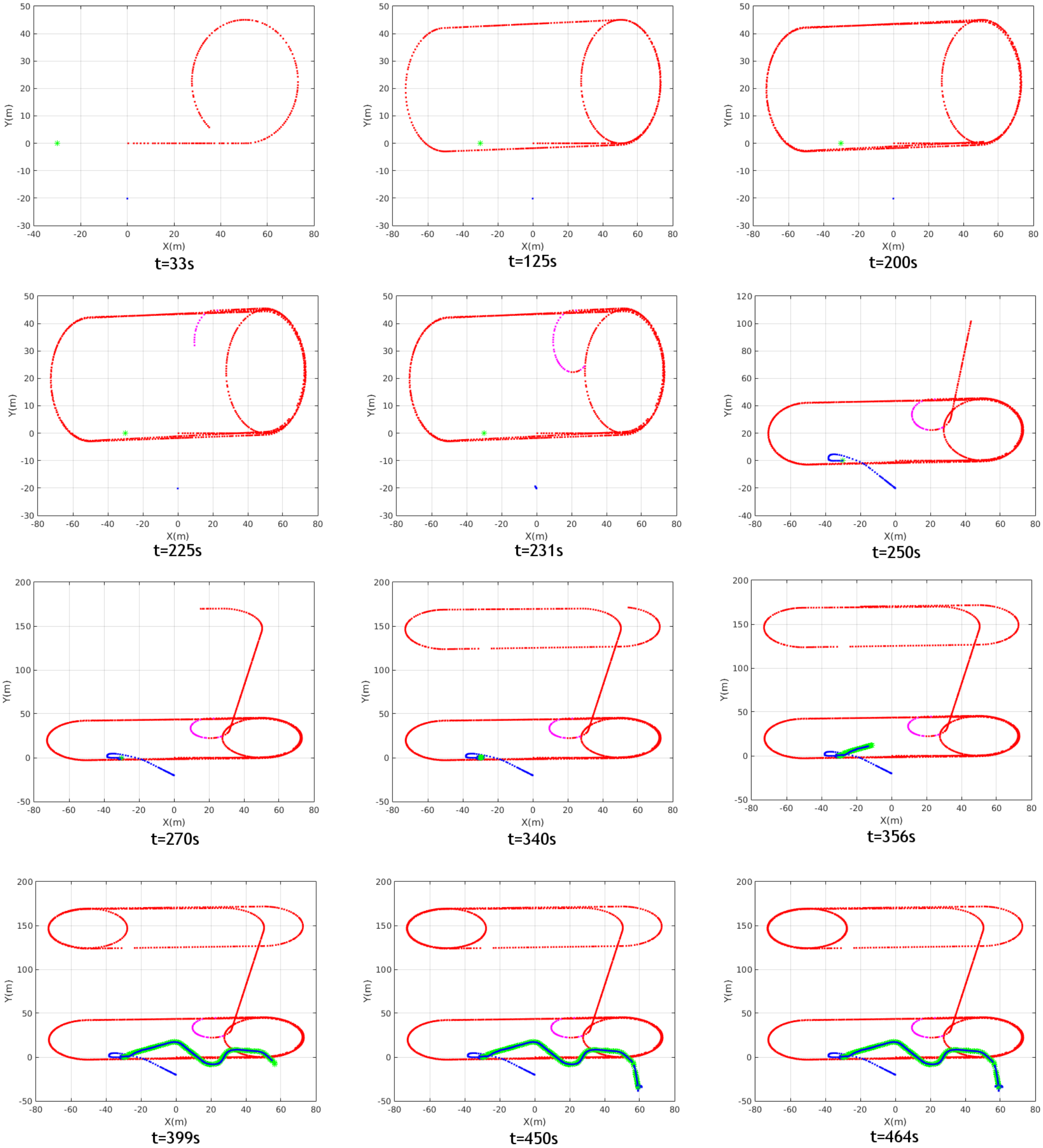
[17] D. van der Walle, B. Fidan, A. Sutton, Changbin Yu, and B. D. Anderson, “Non-hierarchical UAV formation control for surveillance tasks,” *2008 American Control Conference*, pp. 777–782, 2008.

[18] G. Hattenberger, S. Lacroix, and R. Alami, “Formation flight: Evaluation of autonomous configuration control algorithms,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 2628–2633, 2007.

[19] “MORSE Project Webpage.” [Online]. Available: https://www.openrobots.org/morse/doc/stable/morse.html

[20] “ROS Webpage.” [Online]. Available: www.ros.org

[21] G. Echeverria, S. Lemaignan, and A. Degroote, “Simulating complex robotic scenarios with MORSE,” in *The International Conference on Robotics and Automation*. Ieee, may 2011, pp. 46–51.



t=33s

t=125s

t=200s

t=225s

t=231s

t=250s

t=270s

t=340s

t=356s

t=399s

t=450s

t=464s

Fig. 9. Simulation instants review