AI-Based De-confliction and Emergency Landing Algorithm for UAS

Bingze Xia[1], Tong He[1], Iraj Mantegh[2], Wenfang Xie[1]

[1]Department of Mechanical and Industrial Engineering, Concordia University
Montreal, CANADA, H3G 1M8
[2] National Research Council of Canada
Montreal, CANADA, H3T 2B2

([bingze.xia], [tong.he], [wenfang.xie]@concordia.ca), (iraj.mantegh@cnrc-nrc.gc.ca)

ABSTRACT

This paper proposes a new integrated method for Unmanned Aerial Systems (UAS) flying safely in a dynamic 3D urban environment shared by multiple aerial vehicles. This capability is referred to as de-confliction in UAS Traffic Management (UTM). A multi-stage algorithm is designed to reduce the risk of conflict with the other aircraft, combining deep learning for object detection, Markov Decision Process (MDP) and Reinforcement Learning (RL) method. Object detection and MDP are used for local collision avoidance of unexpected intruders; RL methods help the agent take actions with maximum return. In addition, a safe landing process is integrated in case an emergency occurs, such as loss of GPS signal and weather effects. Simulation results in different scenarios show the effectiveness of our algorithm in safe navigation.

Keywords—Unmanned Aircraft Systems (UAS), Artificial Intelligence Methods, De-confliction, Emergency Safe-Landing, UAS Traffic Management, Markov Decision Process, Reinforcement Learning

1.0 INTRODUCTION

With the recent popularization of civilian and commercial Unmanned Aerial Systems (UAS), safe airspace integration and UAS Traffic Management (UTM) [1] is receiving increasing importance. Safe navigation in normal and abnormal situations is the key objective.

Collision avoidance is a crucial enabler for the integration of UAS in controlled airspace and ensures the safety of both the agents and citizens on the ground. Several methods have been studied in the past decades, which can be roughly divided into two groups: non-learning-based [2,3] and learning-based methods. The first group utilizes *a priori* known information of the environment or sensor data to avoid collisions. The second class uses machine learning methods such as reinforcement learning or imitation learning, which solve MDPs by maintaining the memories of past observations and actions. The advantage of learning-based methods, in general, is in their capability to adapt to the environment changes.

On the other hand, for UAS safe navigation, several circumstances may require the UAS to embark on emergency landing, including loss or degradation of GPS or control datalink signal; low battery or propulsion power. Several commercial systems pre-program their UAS auto-pilot to navigate to a pre-set location. However, navigation to these locations, by itself, especially in an urban area, may be challenging due to the presence of other aircraft or objects and cause additional risks.

In this paper, we assume cases with pre-set flight paths from the start point to the destination and several safe landing locations in the vicinity. In case of unexpected encounters of obstacles (like trees or dynamic intruder aircraft) or emergencies during the flight, a multi-stage scheme is thus presented for de-confliction of UAS in airspace and risk reduction through an emergency landing strategy. The method is primarily designed for safe UAS/drone navigation in low altitude urban and rural environments where the Global Navigation Satellite System (GNSS) positioning system may degrade or lose accuracy.

2.0 METHODOLOGY

Our safe navigation algorithm of de-confliction and emergency landing process is executed in four stages. Pseudocodes for specific stages of the algorithm are presented below:

Stage 1: Deep learning model for object detection

At this stage, we deploy on-board Electro-optics (EO) cameras and machine learning to detect and identify intruder aircraft, as well as for detecting the marked landing sites.

Several object detection algorithms like Haar Cascades [4], Convolutional Neural Networks (CNN), and its variants [5] have proven the reliability of deep learning methods by using a custom training dataset of a specific object.

A deep learning model for object detection contains an input layer, several hidden layers (include convolution, pooling, and fully connected layers) and an output layer. YOLO (You Only Look Once) model replaced the fully connected layers with convolutional layers and made it a fully convolutional network. This network successfully minimizes the feature image and realizes real-time multi-object simultaneous detection.

We applied YOLO v4 for our system, which is faster and more accurate than YOLO v3. The implementation of the new architecture in the Backbone (extract the essential features) and the changes in the Neck (detector) have further developed the mAP(mean Average Precision) by 10% and the quantity of FPS (Frame per Second) by 12% [6]. In addition, it has become easier to train this neural network on a single GPU. In training the YOLO model, the DarkNet software package is used to recognize the intruder UAVs and the landing mark.

Stage 2: Markov Decision Process for dynamic obstacle avoidance [7]

The Markov Decision Process is applied here to estimate the probability of the unknown intruders in image view and then to choose the relative actions and policy to make real-time obstacle avoidance. The probability function and the discrete action space is shown below:

$$P_{ss'}^{a} = P[S_{t+1} = s' | S_t = s, A_t = a] \tag{1}$$

$$A = \{a_{inc}, a_{dec}, a_{up}, a_{down}, a_{right}, a_{left}\} \tag{2}$$

Eq. (1) represents the probability that the agent transit to a new state *s'* by taking action *a* from the original state *s*. Eq. (2) shows six potential actions the agent may take to avoid a conflict.

Without any loss of generality, it is assumed that the onboard EO sensor is of a depth camera (RGB-D) that can provide the distance to the intruder in a close range. Other methods, such as stereovision or trained machine learning algorithms [8], can also be applied here.

The pseudocode of the MDP process is presented:

Initialize the RGB-D camera
Compute the relative distance with the intruders
Define the discount factor and the reward rule
Find the optimum policy
While intruder is in the pre-set safety range
Divide the map with cubes
Process the sum of rewards and the probability
Decide the action

Stage 3: Using Deep Reinforcement Learning method to update MDP policy [9,10]

Reinforcement learning can be understood as a recurrent feedback loop of receiving inputs from a sensor to reflect the agent's current states, interact with the environment and use the observed rewards gained for each action it takes as feedback to optimize a policy. To adapt to a highly complex and dynamic environment with uncertainties, we applied the Deep Q Network (DQN) method that helps the MDP update its policy and select the action at each state with the maximum cumulative reward while reaching the destination.

$$R = \begin{cases} -\delta/(d+0.5) & d < \delta \\ \delta + d/5 & d > \delta \\ -10 & Confliction\ occurs \end{cases} \tag{3}$$

A reward function Eq. (3), was designed so that the agent UAS can learn to avoid collisions during navigation. Where *d* is the distance with the obstacle estimated by the camera, and δ is a threshold value of the safe distance (e.g. $\delta = 1.5$). The agent will be rewarded negatively if it is already within the hazardous region or rewarded a positive value if safe; a negative reward of -10 will be given in case a collision happens. After training, the target state-value function could then be represented as Eq. (4) below, and Fig. 1 shows a simplified DQN structure schematic that takes states as system input and actions as output.

$$Q(s_t, a_t) = r_{t+1} + \gamma \max(Q(s_{t+1})) \tag{4}$$

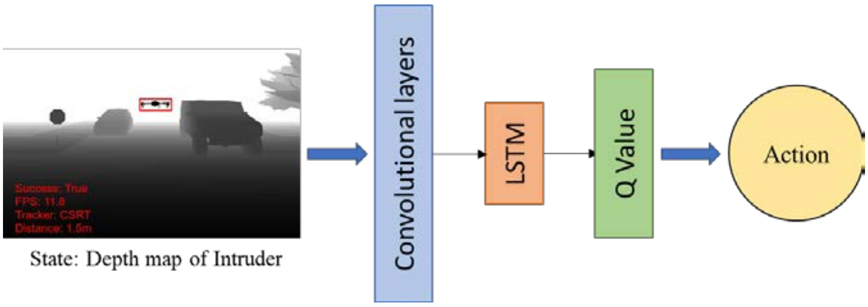

Figure 1. Structure of Deep Q Network

Stage 4: Safe-landing process if an emergency occurs [11]

As a last resort for the safe navigation purpose of our multi-stage de-confliction method, the safe landing process will be activated if the own ship encounters an emergency while executing a task (lose GPS signal or the de-confliction model fails). The agent will not carry on its mission but navigate to the closest pre-set safe landing sites (rally points) in the vicinity. The camera will detect a landing mark on the site, and the agent will land safely on the center of the landing marker using a static vision-based landing algorithm. Meanwhile, the own ship relies on the Inertial Navigation System during the safe landing process. The landing locations were selected close enough to the flight path that ensures the drift won't lead the landing mark out of the onboard camera's detection range. The complete safe emergency landing process pseudocode is presented below:

Compute the distances with each landing site
    Set the shortest-distance site as the target
Compute the bearing angle α
Run Inertial Navigation System to follow the bearing angle
    Return estimated position of the own ship
(Open the RGB-D camera for close-range intruder detection
Run MDP algorithm for short-range de-confliction)
Open the looking-down camera, or rotate the RGB-D camera (with gimbal) for ground view
Run static vision-based landing algorithm
    Descend and disarm

3.0 RESULTS AND DISCUSSION

As illustrated in the simulation results below, Fig. 2(a) shows the own ship accurately detected an intruder UAV by applying the trained YOLO model. Then, Fig. 2(b) demonstrates the application of our collision avoidance process and the own ship successfully ‘jump over’ an intruder aircraft. Also, to test the safe emergency landing algorithm, we manually disabled the GPS signal at Point C when the agent is flying on its pre-set path of the yellow line to the destination point (D), as depicted in Fig. 2(c). The agent then started the safe landing process, changed the flight direction and navigated to the pre-planned landing site (Point R) utilizing the Inertial Navigation System (with gyro drift) with the help of the camera and the collision avoidance algorithm.

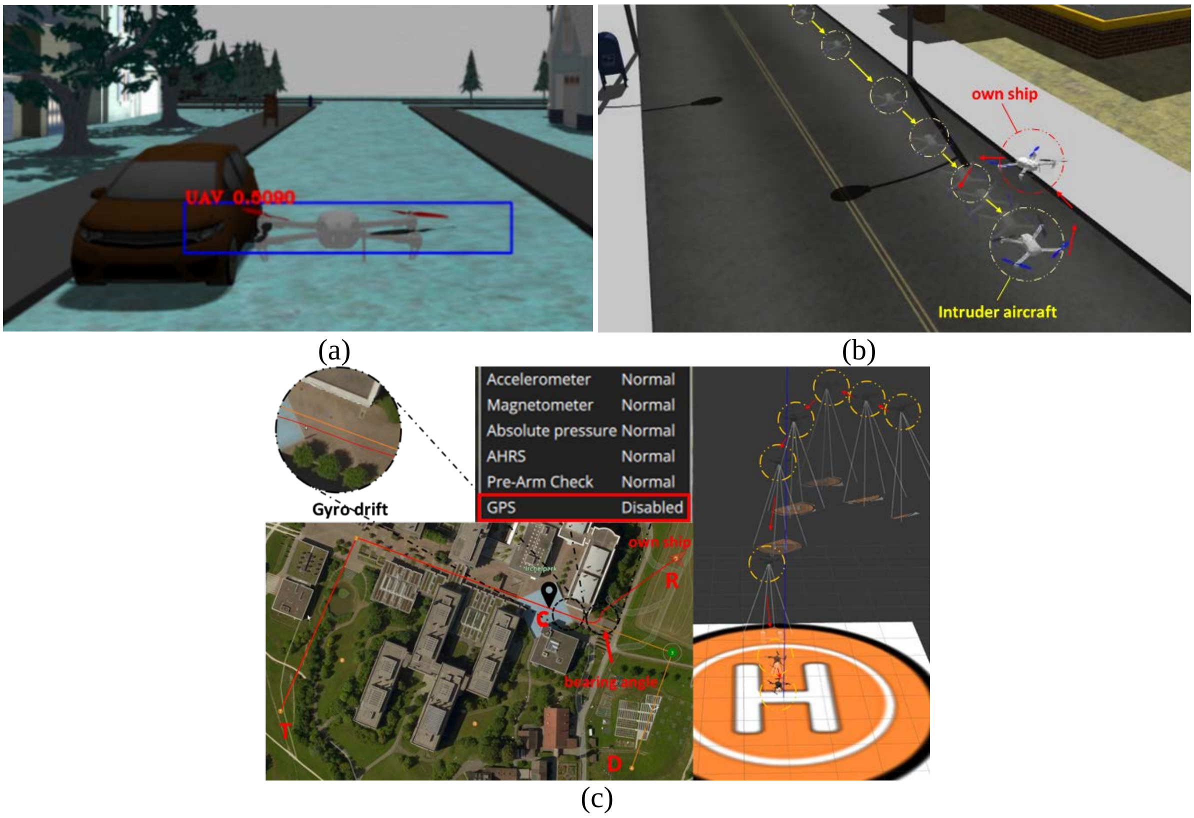

Figure 2. (a). Own ship detects an intruder UAV. (b). The own ship jumps over the intruder aircraft and gets back to its original path to make a collision avoidance. (c). Trajectory (red line) from the taking-off point (T) to the safe landing site (R). Own ship lost its GPS signal at Point C.

In summary, this work presented a new multi-staged algorithm for UAV safe navigation of de-confliction and safe emergency landing in an urban environment. It combines Markov Decision Process, deep learning and reinforcement learning methods to develop a flexible algorithm that can potentially be deployed for Urban Aerial Mobility (UAM) applications. As demonstrated in the simulations, the algorithm can avoid unexpected intruders after training in dynamic airspace shared with other aircraft. For actual cases, this integrated strategy can theoretically be applied to agents with the pre-set flight path, which has avoided the majority of known obstacles but with the uncertainties of unplanned path blocks, based on our simulation results. Scenarios with dense intruder UAVs will be simulated and tested for the robustness of our algorithm in the future and the real flight tests as well.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support from National Research Council Canada's IAM (Integrated Autonomous Mobility) program and the NSERC (Grant No. N00892) for the work reported in this paper.

REFERENCES

[1] NASA UTM roadmap: https://utm.arc.nasa.gov/index.html

[2] W. Xinyu, L. Xiaojuan, G. Yong, S. Jiadong and W. Rui, "Bidirectional Potential Guided RRT* for Motion Planning," in IEEE Access, vol. 7, pp. 95046-95057, 2019.

[3] Jenie Y I, van Kampen E J, de Visser C C, et al. Three-dimensional velocity obstacle method for uncoordinated avoidance maneuvers of unmanned aerial vehicles[J]. Journal of Guidance, Control, and Dynamics, 2016, 39(10): 2312-2323.

[4] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). Ieee.

[5] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2014.81

[6] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (n.d.). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv Vanity. https://www.arxiv-vanity.com/papers/2004.10934/.

[7] T. He, I. Mantegh, L. Chen, C. Vidal and W. Xie, "UAS Flight Path Planning for Dynamic, Multi-Vehicle Environment," 2020 International Conference on Unmanned Aircraft Systems (ICUAS), 2020, pp. 211-219, doi: 10.1109/ICUAS48674.2020.9213911.

[8] Shyam Chanduri, Zeeshan Khan Suri, Igor Vozniak, Christian Müller "CamLessMonoDepth: Monocular Depth Estimation with Unknown Camera Parameters" arXiv - CS - Computer Vision and Pattern Recognition (IF), Pub Date: 2021-10-27, DOI: arxiv-2110.14347

[9] Roghair, J., Niaraki, A., Ko, K., & Jannesari, A. (2021, September). A vision based deep reinforcement learning algorithm for uav obstacle avoidance. In Proceedings of SAI Intelligent Systems Conference (pp. 115-128). Springer, Cham.

[10] Ou, J., Guo, X., Zhu, M., & Lou, W. (2021). Autonomous quadrotor obstacle avoidance based on dueling double deep recurrent Q-learning with monocular vision. Neurocomputing, 441, 300-310.

[11] B. Xia, I. Mantegh and W. Xie, "Integrated Emergency Self-Landing Method for Autonomous UAS in Urban Aerial Mobility," 2021 21st International Conference on Control, Automation and Systems (ICCAS), 2021, pp. 275-282, doi: 10.23919/ICCAS52745.2021.964