Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Co-operative Collision Avoidance for Unmanned Aerial Vehicles using both Centralised and Decoupled Approaches

L. Palmer J. A. A. Engelbrecht

Electrical and Electronic Engineering Department, Stellenbosch University, Stellenbosch, South Africa, (18334504@sun.ac.za, jengelbr@sun.ac.za)

Abstract: This paper presents the design of co-operative collision avoidance algorithms for Unmanned Aerial Vehicles (UAVs) using vertical avoidance manoeuvres. The co-operative collision avoidance problem is formulated as an optimal control problem and solved using an A^* search algorithm. Two different approaches are developed and compared: a centralised approach where the collision avoidance trajectories for all UAVs are planned simultaneously, and a decoupled approach where the individual collision avoidance trajectories for each UAV is planned sequentially, with the planning sequence determined by a UAV priority order. The UAVs co-operate by sharing their state and intent information with one another and with a central node, if present. The co-operative collision avoidance algorithms are verified and evaluated using illustrative simulations. These simulations support the expected behaviour of the algorithms. The centralised approach finds the most optimal solution to the problem while the solution found by the decoupled approach depends on the priority allocation of the UAVs. The decoupled approach can find either the most optimal or a sub-optimal solution to the problem, with the priority allocation occasionally resulting in the decoupled approach being unable to find a solution. This suggests that the centralised approach will, on average, find solutions more often and find more optimal solutions than the decoupled approach.

Keywords: Co-operative control, Optimal Control, Path Planning, Collision Avoidance, Obstacle Avoidance.

1. INTRODUCTION

Interest in autonomous vehicles has increased dramatically in recent years and this has led to many new developments in the field of automation. The automation of Unmanned Aerial Vehicles (UAVs) has vastly increased their uses and UAVs are now commonly used in many fields, ranging from the film industry to agricultural applications to military operations. This boom in the use of UAVs has created new challenges for automation to overcome, such as the need for UAVs to fly in crowded airspaces.

One of the key enabling technologies for the integration of UAVs into commercial airspace, is automatic collision avoidance, or the ability to “sense and avoid”. Traditionally, research on automatic collision avoidance for UAVs has focused on using on-board sensors such as vision-based sensors, lidar, and radar to predict and avoid collisions. We propose a co-operative framework where all UAVs communicate their positions, velocities, and intended flight paths, and therefore act as collision prediction “sensors” for one another. Additionally, when a collision is predicted, then all UAVs co-operate to plan and execute collision avoidance trajectories.

This paper presents two different co-operative collision avoidance algorithms: a centralised approach where the collision avoidance trajectories for all UAVs are planned simultaneously, and a decoupled approach where the individual collision avoidance trajectories for each UAV is planned sequentially, with the planning sequence determined by a UAV priority order. Both algorithms attempt to find optimal solutions using vertical avoidance manoeuvres that avoid collisions, but minimise the deviation of the UAVs from their original trajectories and minimise the collision avoidance effort. The two algorithms are evaluated and compared using illustrative simulations.

The remainder of this article is organised into five sections. Section 2 discusses previous work done in the field. Section 3 discusses the co-operative collision avoidance framework. Section 4 covers co-operative collision prediction. Section 5 covers co-operative path planning and Section 6 describes the results obtained.

2. PREVIOUS WORK

Collision avoidance for manned aircraft is currently performed using a combination of ATC (air traffic control), TCAS (Traffic Collision Avoidance System), EGPWS (Enhanced Ground Proximity Warning System), and ADS-B (Automatic Dependent Surveillance - Broadcast). TCAS is a rules-based collision avoidance system that uses only vertical maneuvers (climb and descend) to avoid aircraft to aircraft collisions. TCAS does not control the aircraft directly but advises resolution actions for the human pilot to follow (FAA, 2011). EGPWS is a rules-based ground

Copyright lies with the authors

collision avoidance system that uses only vertical maneuvers to avoid aircraft collisions with terrain (Bateman, 1999). ADS-B is a surveillance technology in which an aircraft determines its position via satellite navigation and periodically broadcasts it, enabling it to be tracked. The ADS-B information can be received by air traffic control ground stations, and can also be received by other aircraft to provide situational awareness and to allow self-separation (FAA, 2015).

A large number of different approaches and technologies have been developed to provide automatic collision prediction and avoidance (sense and avoid) for UAVs. A survey of collision avoidance approaches for UAVs was performed by Albaker and Rahim (2009), and an overview of sense and avoid technologies applicable to unmanned aircraft systems was provided by Yu and Zhang (2015).

One approach is to integrate UAVs with the existing TCAS and ADS-B infrastructure. Asmat et al. (2007) proposed an unmanned aerial collision avoidance system (UCAS) designed to interact with the Traffic Alert Collision Avoidance System (TCAS) implemented on manned aircraft. Portilla et al. (2007) investigated the feasibility of a sense and avoid system for UAVs that is compatible with manned TCAS II-equipped aircraft. Stark et al. (2013) studied the use of ADS-B for small Unmanned Aerial Systems. Orefice et al. (2015) performed real-time validation of an ADS-B based aircraft conflict detection system.

Another approach is for individual UAVs to perform non-cooperative collision prediction and avoidance using their onboard sensors. A multitude of vision-based collision avoidance approaches have been proposed using monocular vision (Choi et al. (2013), Chiu et al. (2014), Saha et al. (2014), Ma et al. (2015), Al-Kaff et al. (2016), and Potena et al. (2019)) and stereo vision (Park and Kim (2012) and Yu et al. (2018)). Other approaches use onboard radar, lidar, or multi-sensor systems. Ben et al. (2017) developed a radar-based detect and avoid system. Fasano et al. (2008) presented a multi-sensor, non-cooperative collision avoidance system. Farinella and Bhandari (2016) proposed system that uses scanning lidar and ADS-B.

Some approaches also incorporate varying levels of coordatination and co-operation. The FACES algorithm developed by Alliot et al. (2000) uses token passing to manage UAVs replanning by prioritising UAVs based on their transponder number. UAVs pass tokens amongst those that they are in conflict with and the UAVs with the highest transponder number replan first. Many researchers have studied trajectory generation and control for teams of UAVs performing co-operative missions (Beard and McLain (2003)) or multiple UAVs flying in formation (Wang et al. (2007), Eskandarpour and Majd (2014), Kuriki and Namerikawa (2015), Seo et al. (2017), Jin et al. (2018)). These algorithms usually include an element of collision avoidance. Often the purpose is not to avoid collisions with other co-operative UAVs, but rather to co-operatively avoid collisions with static obstacles and non-cooperative dynamic obstacles (Boivin et al. (2008), Kothari et al. (2009), Cichella et al. (2014)).

Only rarely is co-operative collision avoidance performed for independent UAVs executing independent missions. Alejo et al. (2009) proposed a co-operative collision avoidance method for multiple UAVs and other non-cooperative aircraft. However, their approach was to modify the velocities of the UAVs, and they did not employ vertical avoidance manuevers. Lao and Tang (2017) presented a co-operative multi-UAV collision avoidance system based on distributed dynamic optimization and causal analysis. However, their approach assumed a segregated airspace, and did not include terrain avoidance. Fabra et al. (2019) developed a co-operative collision avoidance approach for independent multirotor UAVs following independent missions. However, they used a rules-based approach and did not employ path planning for collision avoidance.

In summary, our literature review reveals that co-operative, path-planning-based collision avoidance for independent UAVs following independent flight plans in en-route airspace has not been adequately covered by previous research. In addition, co-operative collision avoidance for UAVs has not been solved using both centralised and decoupled approaches. We therefore present a co-operative, path-planning-based collision avoidance system that enables co-operatve UAVs to avoid collisions with one another, with static terrain, and with non-cooperative dynamic obstacles. We will formulate the co-operative collision avoidance task as an optimal control problem, and we will solve it using both centralised and decoupled approaches.

3. COLLISION AVOIDANCE FRAMEWORK

Our proposed collision avoidance framework consists of three modules, a modelling module, a collision prediction module, and a path planning module, that work together to perform collision avoidance. The collision prediction module executes continuously to calculate the probability of future collisions occurring in the short-term prediction horizon. If the probability of a future collision exceeds a certain threshold, then the path planning module is activated to plan new collision-free paths. The collision prediction and path planning modules both use the modelling module to propagate the vehicles and dynamic obstacles forward in time, to calculate the probability of future collisions between vehicles, and future collisions with static or dynamic obstacles.

4. CO-OPERATIVE COLLISION PREDICTION

The co-operative collision prediction module assumes that all co-operative UAVs publish and communicate their path plans for the next N time steps into the future. It also assumes that a map of all local static obstacles is available, and that all dynamic obstacles (e.g. uncooperative UAVs) are being tracked so that their predicted trajectories for the next N time steps are available. At every time step, the co-operative collision prediction algorithm simulates the motion of the co-operative UAVs and the dynamic obstacles forward in time, and checks for predicted collisions of co-operative UAVs with one another, of co-operative UAVs with static obstacles, and of co-operative UAVs with dynamic obstacles. The co-operative UAVs are propagated along their published path plans, and the dynamic obstacles are propagated along their predicted trajectories, for N time steps into the future. If any collisions are predicted, then the co-operative path planning algorithm is activated to re-plan the state trajectories of all co-operative UAVs, and to produce and publish new collision-free path plans for the next N time steps.

5. CO-OPERATIVE PATH PLANNING

We propose two multi-robot path planning algorithms to perform co-operative collision avoidance for unmanned aerial vehicles. We first formulate the co-operative collision avoidance task as an optimal control problem. We then solve the optimal control problem using two path planning algorithms that both apply the A^* search algorithm, one that follows a centralised approach, and another that follows a decoupled approach.

5.1 The Optimal Control Problem

The co-operative path planning task is formulated as an optimal control problem with the objective of finding the optimal state trajectories and the optimal actions that minimise some cost function. This must be done while avoiding UAV collisions with one another and with static and dynamic obstacles and while also obeying the differential constraints of each individual UAV.

Problem Formulation: Given the reference flight paths $\mathbf{x}_{\text{ref}}(t)$ for all of the co-operative UAVs, the time-independent conflict region $\mathcal{C}_{\text{obs}}$ representing the static terrain, and the time-varying conflict region $\mathcal{C}_{\text{obs,dyn}}(\text{t})$ representing the predicted trajectories of the uncooperative UAVs, the objective is to determine the optimal collision-free vehicle state trajectories $\mathbf{x}^*(t)$ and the optimal vehicle actions $\mathbf{u}^*(t)$ that will avoid all collisions, while minimising the deviation of the executed flight paths from the original reference flight paths, and while minimising the avoidance effort.

System dynamics: The system dynamics are described by the equations of motion of all of the co-operative UAVs

$$\dot{\mathbf{x}}(t) = \mathbf{v}(t), \tag{1}$$

where the state vector $\mathbf{x}$ is an array containing the positions of all of the co-operative UAVs, and the input vector $\mathbf{v}$ is an array containing the velocities of all of the co-operative UAVs.

System state: The state vector $\mathbf{x}$ is defined as

$$\mathbf{x}(t) = [\mathbf{x}_1(t)\ \mathbf{x}_2(t)\ \ldots\ \mathbf{x}_n(t)]^T, \tag{2}$$

where $\mathbf{x}_1$ to $\mathbf{x}_n$ are the positions of the n co-operative UAVs.

Control input: The control input vector $\mathbf{v}$ is defined as

$$\mathbf{v}(t) = [\mathbf{v}_1(t)\ \mathbf{v}_2(t)\ \ldots\ \mathbf{v}_n(t)]^T, \tag{3}$$

where $\mathbf{v}_1$ to $\mathbf{v}_n$ are the velocity actions of the n co-operative UAVs.

State constraints: The state constraints are specified by defining the set of admissible states where none of the co-operative UAVs are in conflict with one another, with the static obstacle region, or with the dynamic obstacle region.

$$\mathbf{x}_i(t) \notin \mathcal{C}_{\text{sep}}(\mathbf{x}_j(t)),\ i \neq j \tag{4}$$

$$\mathbf{x}_i(t) \notin \mathcal{C}_{\text{obs}} \tag{5}$$

$$\mathbf{x}_i(t) \notin \mathcal{C}_{\text{obs,dyn}}(t). \tag{6}$$

Equations 4, 5 and 6 describe these constraints for all i from 1 to n, where $\mathbf{x}_i$ is the position of the i'th UAV, $\mathcal{C}_{\text{sep}}(\mathbf{x}_j(t))$ is the conflict region around the j'th UAV, $\mathcal{C}_{\text{obs}}$ is the static obstacle region, and $\mathcal{C}_{\text{obs,dyn}}(t)$ is the dynamic obstacle region.

Input constraints: The control input constraints are specified by defining the set of admissible horizontal and vertical velocity ranges for each of the unmanned UAVs. The velocity action for each UAV can be decomposed into a horizontal (forward) velocity and a vertical (climb rate) velocity as

$$\mathbf{v}_i(t) = [v_{i,\text{hor}}\ v_{i,\text{vert}}]^T, \tag{7}$$

where $v_{i,\text{hor}}$ is the horizontal (forward) velocity and $v_{i,\text{vert}}$ is the vertical velocity (climb rate) of the i'th UAV. The input constraints are then specified as

$$v_{i,\text{hor, min}} \leq v_{i,\text{hor}}(t) \leq v_{i,\text{hor, max}} \tag{8}$$

$$v_{i,\text{vert, min}} \leq v_{i,\text{vert}}(t) \leq v_{i,\text{vert, max}}, \tag{9}$$

where $v_{i,\text{hor, min}}$ and $v_{i,\text{hor, max}}$ are the minimum and maximum forward velocities, and $v_{i,\text{vert, min}}$ and $v_{i,\text{vert, max}}$ are the minimum and maximum climb rates for the i'th UAV. Since different UAVs can be given different input constraints, this formulation allows for a heterogeneous set of co-operative UAVs.

Terminal state constraints: The co-operative collision avoidance algorithm must provide collision-free planned trajectories for a minimum planning time interval into the future. This requirement for a minimum "planning horizon" is translated into the following terminal state constraint

$$t_f \geq t_i + T_{\text{plan}}, \tag{10}$$

where the t_i is the initial time, t_f is the final time, and T_{plan} is the minimum planning time interval into the future. Note that no further constraints are placed on the "final" positions of the UAVs at the end of the planning time interval, and that the UAVs are not required to have returned to their reference paths by the "final" time. This is to provide for possibility that the "final" reference positions (or goal states) of one or more UAVs may be occupied by static or dynamic obstacles at the end of the planning time interval.

Hierarchical multi-objective cost function: A novel hierarchical multi-objective cost function is designed to represent the primary objective to minimise the deviation of the UAVs from their reference trajectories, and the secondary objective to minimise the avoidance effort.

The primary cost function $J_{\mathbf{x}}$ is the deviation of the UAVs from their reference trajectories, defined as

$$J_{\mathbf{x}} = \int_{t_i}^{t_f} |\mathbf{x}(t) - \mathbf{x}_{\text{ref}}(t)|\, dt, \tag{11}$$

where $\mathbf{x}(t)$ are the planned trajectories and $\mathbf{x}_{\text{ref}}(t)$ are the reference trajectories.

The secondary cost function $J_{\mathbf{u}}$ is the avoidance effort, defined as

$$J_{\mathbf{u}} = \int_{t_i}^{t_f} g_{\mathbf{u}}(\mathbf{v}(t))dt, \tag{12}$$

where $g_{\mathbf{u}}(\mathbf{v}(t))$ is an arbitrary function representing the transition cost of input action $\mathbf{v}(t)$.

The hierarchical cost function is minimised as follows: First, the primary cost function $J_{\mathbf{x}}$ is minimised without considering the secondary cost function. Next, if more than one solution minimises the primary cost function, then the solution that also minimise the secondary cost function $J_{\mathbf{u}}$ is selected from among the solutions that already minimise the primary cost function. (The secondary cost function is not activated frequently, since steep vertical maneuvers often also result in a smaller path deviation area than normal maneuvers.)

5.2 The A^* Solution: Centralised and Decoupled

The centralised and decoupled path planning approaches both use the general A^* algorithm, shown in algorithm 1, described in LaValle (2016). The A^* algorithm will always explore the cheapest node first to guarantee that the first path found it the cheapest. To implement this, the insert function (line 10 of algorithm 1) will insert nodes into the open queue such that the nodes in the queue are arranged in increasing order of cost. The A^* algorithm also makes use of a heuristic in its cost function. The heuristic estimates the cost to reach the goal from the current node, and this is added to the cost to reach that node to produce the total cost for the node that is used by the insert function.

Algorithm 1 Generic A^* Algorithm

1: $Q_{open}.insert(x_I)$
2: **while** Q_{open} not empty **and** size(Q_{closed}) $< \beta$ **do**
3: $\quad x \leftarrow Q_{open}.getFirst()$
4: $\quad$ **if** $x == x_G$ **then**
5: $\qquad$ **return** SUCCESS
6: $\quad$ **else**
7: $\qquad$ **for all** $u \in U$ **do**
8: $\qquad\quad x' \leftarrow f(x, u)$
9: $\qquad\quad$ **if** x' valid **then**
10: $\qquad\qquad Q_{open}.insert(x')$
11: $\qquad Q_{closed}.insert(x)$
12: **return** FAILURE

Algorithm execution: Centralised approach: In the centralised planning approach, the composite trajectory for all UAVs is planned simultaneously using the A^* algorithm. A single centralised search graph is created for all UAVs. Each node in the graph contains the composite state, the composite action, the composite trajectory deviation cost, the composite action cost, and the time index for all UAVs.

The composite state transition equation for all UAVs is used to generate the child nodes from a given parent node for all of the composite actions in the action space. The composite state of each child node is checked for collisions to verify that the new state is admissible before it is added to the search graph. The individual UAVs' states within the composite state are checked against one another to verify that there are no collisions between co-operative UAVs, and the individual UAVs' states are also checked against the static obstacle and dynamic obstacle maps to verify that there are no collisions with static or dynamic obstacles.

The nodes in the open queue are first ordered from lowest to highest composite trajectory deviation cost, then ordered from lowest to highest composite action cost, and finally ordered from highest to lowest time index k. Nodes with the same composite trajectory deviation cost and composite action cost are therefore ordered so that nodes that are closer to the final time index are higher in the open queue and will be expanded first, to encourage the A^* algorithm to reach the goal state within the minimum number of iterations.

The A^* algorithm terminates when the goal set is reached, when the open queue is empty, or when the algorithm exceeds a specified maximum number of iterations. If the algorithm does not reach the goal state, then either no solution exists, or no solution could be found within the specified maximum number of iterations.

Algorithm Execution: Decoupled Approach: In the decoupled approach, the individual trajectories for the UAVs are planned sequentially using the A^* algorithm and the planning sequence is determined by the priority order of the UAVs. UAVs with higher priority ignore lower priority UAVs when planning their new trajectories, while UAVs with lower priority treat higher priority UAVs as dynamic obstacles that must be avoided.

A decoupled search tree is created for each individual UAV. Each node in the graph contains the individual state, the individual action, the individual trajectory deviation cost, the individual action cost, and the time index of the UAV. The individual state transition equation for the UAV is used to generate the child nodes from a given parent node using the individual actions from the action space. The individual state of each child node is checked for collisions to verify that the new state is admissible before it is added to the search graph.

The individual state of the UAV is checked against the state trajectories of all higher priority UAVs to check for collisions with other co-operative UAVs, and the individual UAV's state is also checked against the static obstacle and dynamic obstacle maps to verify that there are no collisions with static or dynamic obstacles. The nodes in the open queue are first ordered from lowest to highest individual trajectory deviation cost, then ordered from lowest to highest individual action cost, and finally ordered from highest to lowest time index k. Nodes with the same individual trajectory deviation cost and individual action cost are therefore ordered so that nodes that are closer to the final time index are higher in the open queue and will be expanded first, to encourage the A^* algorithm to reach the goal state within the minimum number of iterations.

Once an individual UAV has completed its planning, it publishes its new state trajectory, and becomes a dynamic obstacle for all of the UAVs lower in the priority order. The next UAV in the priority order then proceeds to plan its new state trajectory, using the published state trajectories of all the previous UAVs. This cycle is repeated until all of the UAVs have performed their individual planning. The A^* algorithm terminates when the goal set is reached, when the open queue is empty, or when the algorithm exceeds a predefined maximum number of iterations. If for any of the individual UAVs the A^* algorithm does not successfully reach its goal state, then a solution does not exist for that specific priority order. However, a solution may still exist if a different priority order is used. The priority order should then be modified, and the sequential planning should be repeated using the new priority order.

Individual state transition equation: The following state transition equation for an individual UAV is used to generate the child nodes from a given parent node

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \mathbf{v}_i(k)\Delta t. \tag{13}$$

$\mathbf{x}_i(k)$ and $\mathbf{v}_i(k)$ are the UAV position and UAV climb rate action at the parent node, $\mathbf{x}_i(k+1)$ is the UAV position at the child node, and Δt is the sampling period of the discrete time step.

Individual action space: The following discrete action space for individual UAVs, consisting of a finite set of available velocity actions, is used to generate the child nodes from a given parent node, in the decoupled approach.

$$\mathbf{v}_i(k) \in U_i \tag{14}$$

$$U_i \in \begin{Bmatrix} \mathbf{v}_{\text{i,steep climb}} \\ \mathbf{v}_{\text{i,climb}} \\ \mathbf{v}_{\text{i,level}} \\ \mathbf{v}_{\text{i,dive}} \\ \mathbf{v}_{\text{i,steep dive}} \end{Bmatrix}. \tag{15}$$

The available actions for each UAV are chosen as "maintain level flight", "climb", "dive", "steep climb", and "steep dive". (This research assumes that the UAVs only use vertical actions for collision avoidance and as such horizontal actions or speed changes are not considered.) The velocity actions assume that the UAV is travelling horizontally at its cruise speed $v_{i,\text{cruise}}$, and that it only adjusts its flight path angle γ. The velocity action for an individual UAV can therefore be represented by the following horizontal (forward) velocity and vertical (climb rate) velocity actions.

$$\mathbf{v}_i(k) = \left[v_{i,\text{cruise}} \;\; v_{i,\text{cruise}} \tan\gamma(k)\right]^T . \tag{16}$$

Individual cost to come: The individual cost to come for each UAV is calculated incrementally as nodes are created and added to the search tree. The individual cost to come of a child node is the sum of the individual cost to come of the parent node and the incremental cost to transition from the parent node to the child node.

Trajectory deviation cost to come: The individual cost to come for a UAV due to its deviation from its reference trajectory is calculated with

$$G_{\mathbf{x}_i}(k+1) = G_{\mathbf{x}_i}(k) + \Delta G_{\mathbf{x}_i}(k, k+1), \tag{17}$$

where $G_{\mathbf{x}_i}(k+1)$ is the total cost to come of the child state, $G_{\mathbf{x}_i}(k)$ is the total cost to come of the parent state, and $\Delta G_{\mathbf{x}_i}$ is the incremental state transition cost. The incremental state transition cost is the area of the surface that is formed between the UAV's reference trajectory and its actual trajectory over the given time step.

Avoidance effort cost to come: The individual cost to come for a UAV due to its avoidance effort is calculated with

$$G_{\mathbf{u}_i}(k+1) = G_{\mathbf{u}_i}(k) + \Delta G_{\mathbf{u}_i}(\mathbf{v}_i(k)), \tag{18}$$

where $G_{\mathbf{u}_i}(k+1)$ is the total cost to come of the child state, $G_{\mathbf{u}_i}(k)$ is the total cost to come of the parent state, and $\Delta G_{\mathbf{u}_i}(\mathbf{v}_i(k))$ is the incremental action cost. The incremental cost $\Delta G_{\mathbf{u}_i}(\mathbf{v}_i(k))$ of an individual UAV's avoidance effort is arbitrarily defined as

$$\Delta G_{\mathbf{u}_i}(\mathbf{v}_i(k)) = \begin{cases} 0, & \text{for } \mathbf{v}_i(k) = \mathbf{v}_{\text{level}} \\ 1, & \text{for } \mathbf{v}_i(k) = \mathbf{v}_{\text{climb}} \\ 1, & \text{for } \mathbf{v}_i(k) = \mathbf{v}_{\text{dive}} \\ 3, & \text{for } \mathbf{v}_i(k) = \mathbf{v}_{\text{steep climb}} \\ 3, & \text{for } \mathbf{v}_i(k) = \mathbf{v}_{\text{steep dive}} \end{cases} . \tag{19}$$

The action costs have been chosen so that a steep climb action is more expensive than two consecutive normal climb actions, and a steep dive action is more expensive than two consecutive normal dive actions.

Individual cost to go heuristic: The individual cost to go for an individual UAV is calculated using a heuristic function that estimates the cost of the cheapest path from the child node to a goal node.

Trajectory deviation: The cost to go for the trajectory deviation is estimated to be the area of the triangle that is formed when the UAV returns to the reference altitude from its current altitude using a "steep climb" or "steep dive" action.

$$H_{\mathbf{x}_i}(k+1) = \frac{1}{2}|h_i(k+1)| \left|\frac{h_i(k+1)}{\tan\gamma_{steep}}\right|. \tag{20}$$

The "steep climb" or "steep dive" action will produce the smallest future trajectory deviation and therefore represents the lowest cost to go. The lowest cost to go is selected for the trajectory deviation to ensure that the cost to go is underestimated.

Avoidance effort: The cost to go for the avoidance effort is selected to be zero, to make sure that the cost to go for avoidance effort is underestimated.

$$H_{\mathbf{u}_i}(k+1) = 0. \tag{21}$$

Individual total path cost: The total path cost for an individual UAV is the sum of its individual cost to come and its individual cost to go.

$$J_{\mathbf{x}_i}(k+1) = G_{\mathbf{x}_i}(k+1) + H_{\mathbf{x}_i}(k+1) \quad (22)$$

$$J_{\mathbf{u}_i}(k+1) = G_{\mathbf{u}_i}(k+1) + H_{\mathbf{u}_i}(k+1). \quad (23)$$

$J_{\mathbf{x}_i}$ is the total trajectory deviation path cost, $G_{\mathbf{x}_i}$ is the trajectory deviation cost to come, and $H_{\mathbf{x}_i}$ is the trajectory deviation cost to go for an individual UAV. $J_{\mathbf{u}_i}$ is the total avoidance effort path cost, $G_{\mathbf{u}_i}$ is the avoidance effort cost to come, and $H_{\mathbf{u}_i}$ is the avoidance effort cost to go for an individual UAV.

Composite state transition equation: The following composite state transition equation for all UAVs is used to generate the child nodes from a given parent node, in the centralised approach:

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{v}(k)\Delta t, \quad (24)$$

with

$$\mathbf{x}(k) = [\mathbf{v}_1(k)\ \mathbf{k}_2(t)\ \ldots\ \mathbf{v}_n(k)]^T \quad (25)$$

$$\mathbf{v}(t) = [\mathbf{v}_1(k)\ \mathbf{v}_2(k)\ \ldots\ \mathbf{v}_n(k)]^T. \quad (26)$$

$\mathbf{x}(k)$ and $\mathbf{v}(k)$ contain all UAV positions and all UAV climb rate actions at the parent node, $\mathbf{x}(k+1)$ contains all UAV positions at the child node, and Δt is the sampling period of the discrete time step.

Composite action space: The following composite action space for all UAVs, consisting of all possible combinations of individual UAV actions, is used to generate the child nodes from a given parent node, in the centralised approach:

$$\mathbf{v}(k) \in U \quad (27)$$

$$U = U_1 \times U_2 \times \ldots \times U_i \times \cdots \times U_n. \quad (28)$$

The composite action space U for all of the UAVs is the Cartesian product of the individual action spaces U_1 to U_n for the individual UAVs.

Composite costs: The composite cost to come, composite cost to go, and composite total path cost for all UAVs, is the sum of the individual costs to come, individual costs to go, and individual total path costs of the individual UAVs.

$$G_{\mathbf{x}}(k) = \sum_{i=1}^{n} G_{\mathbf{x}_i}(k) \quad (29)$$

$$G_{\mathbf{u}}(k) = \sum_{i=1}^{n} G_{\mathbf{u}_i}(k) \quad (30)$$

$$H_{\mathbf{x}}(k) = \sum_{i=1}^{n} H_{\mathbf{x}_i}(k+1) \quad (31)$$

$$H_{\mathbf{u}}(k) = \sum_{i=1}^{n} H_{\mathbf{u}_i}(k+1) \quad (32)$$

$$J_{\mathbf{x}}(k) = \sum_{i=1}^{n} J_{\mathbf{x}_i}(k) \quad (33)$$

$$J_{\mathbf{u}}(k) = \sum_{i=1}^{n} J_{\mathbf{u}_i}(k). \quad (34)$$

6. SIMULATION AND RESULTS

The centralised and decoupled collision resolution strategies were tested through implementing the simultaneous co-operative collision avoidance algorithm in a simulation environment. The simulation environment implements the specific formulation of the optimal control problem, the centralised and decoupled approaches, and approximates terrain as rectangular obstacles. This simulation environment was used to perform illustrative simulations on the algorithm for different numbers of UAVs replanning their routes simultaneously, namely all together (centralised approach) or individually (decoupled approach). To do this, a simulation environment was developed and results obtained and discussed.

Figure 1 shows an illustrative case of collision and obstacle avoidance for the centralised approach. The bottom UAV flying left-to-right (red) must climb to avoid the obstacle which causes the UAV above it (blue) to have to climb too. The UAV flying right-to-left (green) cannot pass between the blue and red UAVs and as such must avoid them. This solution is chosen as it has a lower global cost than causing the red and blue UAVs to detour.

Figure 2 shows an illustrative case of collision and obstacle avoidance for the decoupled approach. This solution appears different to the centralised solution shown in figure 1 as the green UAV finds a different route, but the resultant cost is the same. The decoupled approach merely chooses to climb instead of dive and as these actions have the same cost the solutions are considered equivalent. The A^* algorithm does not differentiate between actions with identical costs and, as such, if there are equal-cost options available, which one is chosen depends only on which one the A^* algorithm finds first.

The solution found by the decoupled approach is dependent on the priority order of the UAVs. Figure 3 shows the same scenario as in figure 2 but with a different priority order. In this scenario, the green UAV has a higher priority than the red or blue UAVs and such replans first. It chooses to go straight, causing both the red and blue UAVs to have to detour around it. As two UAVs are performing avoidance manoeuvres, the resultant cost is higher than the solution shown in figure 2 and, as such, the solution found for this priority scheme is less optimal.

A further consequence of the choice of the priority allocation of the UAVs is illustrated in figure 4. This is the same scenario as the previous examples, but the red UAV has the lowest priority and the blue UAV has the highest priority. As a result, the blue UAV replans first and chooses to go straight. The green UAV then chooses to go as straight as possible, but avoids the blue UAV. The red UAV replans last. As the blue UAV has chosen to fly straight, the red UAV is unable to avoid the terrain obstacles. As a result, this scenario is unsolvable for this priority scheme.

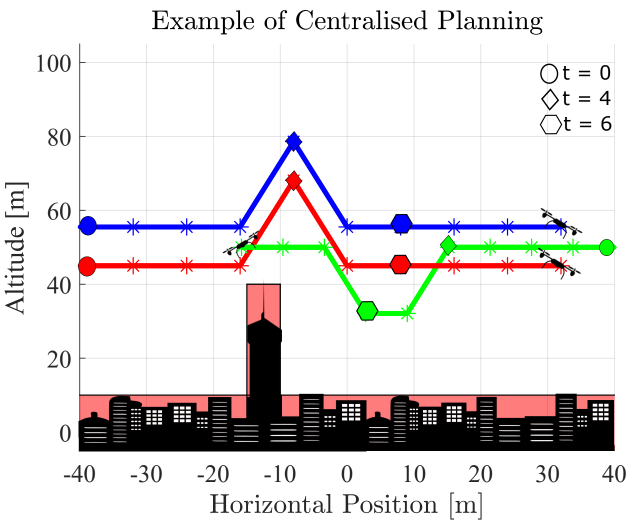

Fig. 1. Example of Solution for Centralised Strategy

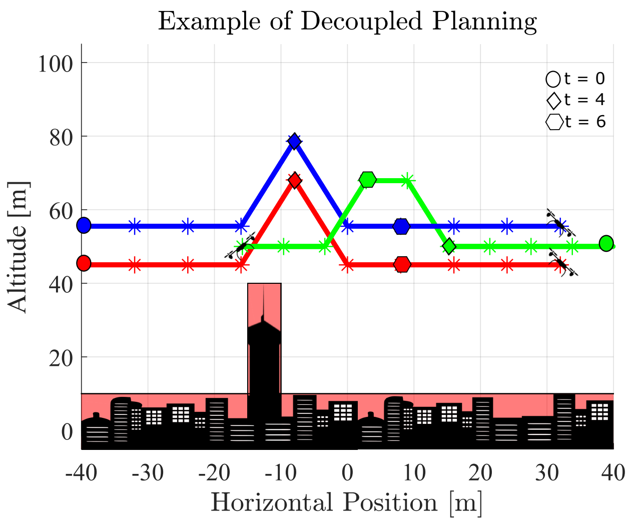

Fig. 2. Example of Optimal Solution for Decoupled Strategy

The centralised approach find the most optimal solution to the collision avoidance scenario. The solution found by the decoupled approach is dependent on the priority order of the UAVs. The priority chosen can cause the decoupled approach to find either the most optimal approach or a sub-optimal approach. The priority allocation can also result in the decoupled approach being unable to find a solution to the scenario. From this it can be assumed that, due to the decoupled approach being dependent on the priority allocation to find a solution, the centralised solution will find solutions more often than the decoupled approach as it is guaranteed to find a solution if one exists. The reliance of the decoupled approach on the priority allocation also means that the centralised approach will, on average, find cheaper solutions than the decoupled approach as the centralised approach will always find the most optimal solution and the decoupled approach cannot make the same claim.

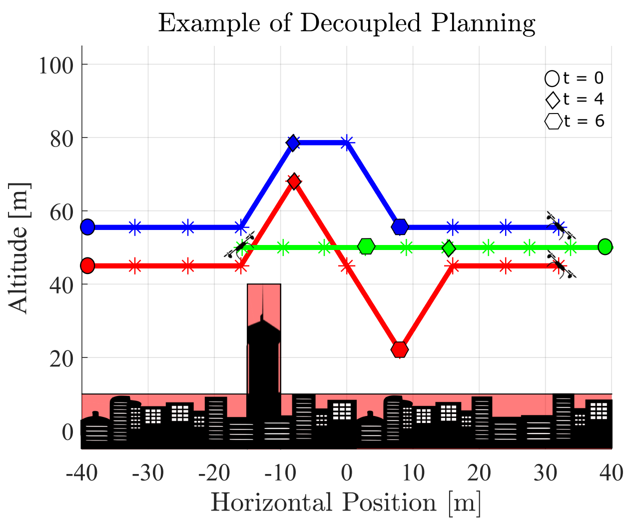

Fig. 3. Example of Suboptimal Solution for Decoupled Strategy

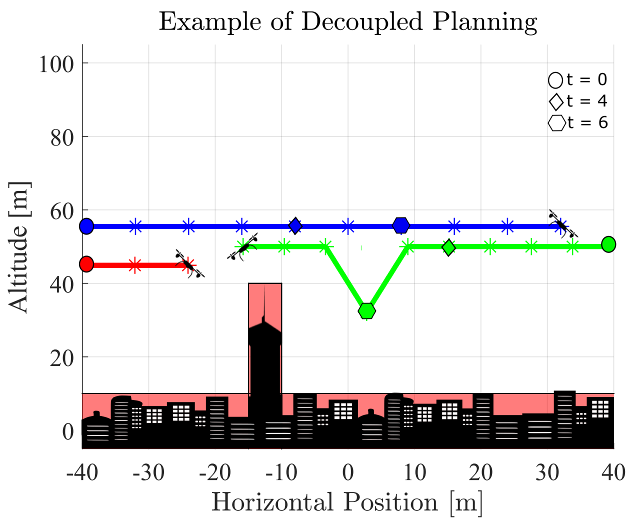

Fig. 4. Example of Unsolvable Solution for Decoupled Strategy

REFERENCES

Al-Kaff, A., Meng, Q., Martin, D., De La Escalera, A., and Armingol, J.M. (2016). Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2016-Augus(Iv), 92–97.

Albaker, B.M. and Rahim, N.A. (2009). A survey of collision avoidance approaches for unmanned aerial vehicles. In *2009 International Conference for Technical Postgraduates (TECHPOS)*, 1–7.

Alejo, D., Conde, R., Cabana, J.A., and Ollero, A. (2009). Multi-UAV collision avoidance with separation assurance under uncertainties. In *IEEE 2009 International Conference on Mechatronics, ICM 2009*.

Alliot, J.M., Durand, N., and Granger, G. (2000). Faces: A free flight autonomous and coordinated embarked solver. *Air Traffic Control Quarterly*, 8(2), 130.

Asmat, J., Rhodes, B., Umansky, J., Villavicencio, C., Yunas, A., Donohue, G., and Lacher, A. (2007). UAS safety: Unmanned Aerial Collision Avoidance System

(UCAS). In *Proceedings of the 2006 IEEE Systems and Information Engineering Design Symposium, SIEDS'06*, 43–49.

Bateman, C. (1999). The introduction of enhanced ground-proximity warning systems(egpws) into civil aviation operations around the world. *Flight safety: Management, measurement and margins*, 259–273.

Beard, R.W. and McLain, T.W. (2003). Multiple UAV Cooperative Search under Collision Avoidance and Limited Range Communication Constraints. Technical report.

Ben, C., Keller, M., Voget, N., and Moormann, D. (2017). Evaluation of a radar based three-dimensional detect and avoid system for small unmanned aerial systems. In *17th AIAA Aviation Technology, Integration, and Operations Conference, 2017*. American Institute of Aeronautics and Astronautics Inc, AIAA.

Boivin, E., Desbiens, A., and Gagnon, E. (2008). UAV collision avoidance using cooperative predictive control. *2008 Mediterranean Conference on Control and Automation - Conference Proceedings, MED'08*, 682–688.

Chiu, C.C., Teng, Y.J., Huang, K.L., and Lai, C.C. (2014). Vision-based automatic obstacle avoidance for unmanned aerial vehicles. *Information (Japan)*, 17(2), 797–805.

Choi, H., Kim, Y., and Hwang, I. (2013). Reactive collision avoidance of unmanned aerial vehicles using a single vision sensor. *Journal of Guidance, Control, and Dynamics*, 36(4), 1234–1240.

Cichella, V., Choe, R., Mehdi, S.B., Xargay, E., Hovakimyan, N., Dobrokhodov, V., and Kaminer, I. (2014). Trajectory generation and collision avoidance for safe operation of cooperating UAVs. In *AIAA Guidance, Navigation, and Control Conference*.

Eskandarpour, A. and Majd, V.J. (2014). Cooperative formation control of quadrotors with obstacle avoidance and self collisions based on a hierarchical MPC approach. *2014 2nd RSI/ISM International Conference on Robotics and Mechatronics, ICRoM 2014*, 351–356.

FAA (2011). Introduction to tcas ii version 7.1.

FAA (2015). Automatic dependent surveillance-broadcast (ads-b) out performance requirements to support air traffic control (atc) service; technical amendment.

Fabra, F., Zamora, W., Sangüesa, J., Calafate, C.T., Cano, J.C., and Manzoni, P. (2019). A distributed approach for collision avoidance between multirotor UAVs following planned missions. *Sensors (Switzerland)*, 19(10).

Farinella, J. and Bhandari, S. (2016). UAV collision avoidance using a predictive rapidly-exploring random tree (RRT). *AIAA Infotech @ Aerospace Conference*, (January), 1–17.

Fasano, G., Accardo, D., Moccia, A., Carbone, G., Ciniglio, U., Corraro, F., and Luongo, S. (2008). Multi-sensor-based fully autonomous non-cooperative collision avoidance system for unmanned air vehicles. *Journal of Aerospace Computing, Information and Communication*, 5(10), 338–360.

Jin, P., Yu, J., Ai, X., and Shen, Y. (2018). Distributed optimal formation control with collision avoidance. *Proceedings of 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference, ITOEC 2018*, (Itoec), 615–619.

Kothari, M., Postlethwaite, I., and Gu, D.W. (2009). Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees. *Proceedings of the IEEE Conference on Decision and Control*, 3069–3074.

Kuriki, Y. and Namerikawa, T. (2015). Formation control with collision avoidance for a multi-UAV system using decentralized MPC and consensus-based control. *2015 European Control Conference, ECC 2015*.

Lao, M. and Tang, J. (2017). Cooperative multi-UAV collision avoidance based on distributed dynamic optimization and causal analysis. *Applied Sciences (Switzerland)*, 7(1).

LaValle, S.M. (2016). *Planning Algorithms*. Cambridge University Press.

Ma, Z., Hu, T., Shen, L., Kong, W., and Zhao, B. (2015). A detection and relative direction estimation method for UAV in sense-and-avoid. *2015 IEEE International Conference on Information and Automation, ICIA 2015 - In conjunction with 2015 IEEE International Conference on Automation and Logistics*, (August), 2677–2682.

Orefice, M., Di Vito, V., and Torrano, G. (2015). Sense and Avoid: Systems and Methods. *Encyclopedia of Aerospace Engineering*, (October), 1–9.

Park, J. and Kim, Y. (2012). Stereo vision based collision avoidance of quadrotor UAV. *International Conference on Control, Automation and Systems*, 173–178.

Portilla, E., Fung, A., Chen, W.Z., Shakernia, O., and Molnar, T. (2007). Sense and Avoid (SAA) & Traffic Alert and Collision Avoidance System (TCAS) integration for Unmanned Aerial Systems (UAS). In *Collection of Technical Papers - 2007 AIAA InfoTech at Aerospace Conference*, volume 3, 2796–2807.

Potena, C., Nardi, D., and Pretto, A. (2019). Joint Vision-Based Navigation, Control and Obstacle Avoidance for UAVs in Dynamic Environments.

Saha, S., Natraj, A., and Waharte, S. (2014). A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment. *Proceeding - ICARES 2014: 2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology*, 189–195.

Seo, J., Kim, Y., Kim, S., and Tsourdos, A. (2017). Collision Avoidance Strategies for Unmanned Aerial Vehicles in Formation Flight. *IEEE Transactions on Aerospace and Electronic Systems*.

Stark, B., Stevenson, B., and Chen, Y. (2013). Ads-b for small unmanned aerial systems: Case study and regulatory practices. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 152–159.

Wang, X., Yadav, V., and Balakrishnan, S.N. (2007). Cooperative UAV formation flying with obstacle/collision avoidance. *IEEE Transactions on Control Systems Technology*, 15(4), 672–679.

Yu, X. and Zhang, Y. (2015). Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Progress in Aerospace Sciences*, 74, 152–166.

Yu, Y., Tingting, W., Long, C., and Weiwei, Z. (2018). Stereo vision based obstacle avoidance strategy for quadcopter UAV. *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, 490–494.