Modeling, Identification and Control, Vol. 37, No. 2, 2016, pp. 77–97, ISSN 1890–1328

Motion- and Communication-Planning of Unmanned Aerial Vehicles in Delay Tolerant Network using Mixed-Integer Linear Programming

E. I. Grøtli[1] T. A. Johansen[2]

[1] *Department of Applied Cybernetics, SINTEF ICT, PO 4760 Sluppen, N-7465 Trondheim, Norway.*
E-mail: EstenIngar.Grotli@sintef.no

[2] *Department of Engineering Cybernetics, Norwegian University of Science and Technology, Center for Autonomous Marine Operations and Systems, N-7491 Trondheim, Norway.*
E-mail: Tor.Arne.Johansen@itk.ntnu.no

Abstract

Large amounts of data are typically generated in applications such as surveillance of power lines and railways, inspection of gas pipes, and security surveillance. In the latter application it is a necessity that the data is transmitted to the control centre “on-the-fly” for analysis. Also missions related to other applications would greatly benefit from near real-time analysis and operator interaction based on captured data. This is the motivation behind this paper on coarse offline motion- and communication-planning for cooperating Unmanned Aerial Vehicles (UAVs). A Mixed-Integer Linear Programming (MILP) problem is defined in order to solve the surveillance mission. To efficiently transmit the data back to the base station the vehicles are allowed to store data for later transmission and transmit via other vehicles, in addition to direct transmission.

The paths obtained by solving the optimization problem are analyzed using a realistic radio propagation path loss simulator. If the radio propagation path loss exceeds the maximum design criterion the optimization problem is solved again with a stricter communication constraint, and the procedure is continued in an iterative manner until the criterion is met. The proposed algorithm is supported by simulations showing the resulting paths and communication topologies for different choices of delay tolerance.

Keywords: Planning; UAV; Surveillance, Delay Tolerant Network

1 Introduction

Among the first to consider multi-vehicle path planning using MILP, were Schouwenaars et al. (2001), Earl and D’Andrea (2002) and Richards and How (2002). Trajectory generation for autonomous vehicles was considered in Schouwenaars et al. (2001), while a robotic ball game was the application of Earl and D’Andrea (2002). Since then MILP has been used extensively for path planning problems, in particular for UAVs, both for single and multi-vehicle systems, see Richards and How (2002), Ma and Miller (2006), Shengxiang and Hailong (2008), Kuwata and How (2011), Grøtli and Johansen (2012b) and Grøtli and Johansen (2012c). In Schouwenaars et al. (2006) connectivity constrained trajectory planning for autonomous helicopters through cluttered environments was studied. Line-of-sight connectivity between a leader helicopter and base station was maintained by coordinating the motion of intermediate helicopters. Other motion planning techniques for improving network or communication properties, which do not necessarily involve solving an optimization prob-

doi:10.4173/mic.2016.2.1

lem, have been treated in Spanos and Murray (2005), Dixon and Frew (2005) and Dixon and Frew (2007) among many others.

In on of our previous papers, Grøtli and Johansen (2012b), we described how paths for multiple UAVs can be planned to create a communication chain between two base stations, also known as *chaining*, Dixon and Frew (2007). In Grøtli and Johansen (2012c) paths were planned for a multi-task mission similar to that presented in this paper, and where the mission objective only could be met if one of the vehicles were used as a relay node. By *relaying*, we mean that additional nodes are used to receive a transmission from a source and retransmit it to a destination, Dixon and Frew (2007), Frew and Brown (2008). Task assignment for multiple UAVs has been considered in several earlier papers, see for instance Kim et al. (2007), Alighanbari et al. (2003) and Kingston and Schumacher (2005), but the focus of these papers has mainly been on scheduling or selection of straight line paths.

The results from Grøtli and Johansen (2012c) are generalized in this paper in order to allow also for ferrying in a Delay Tolerant Network (DTN). The first notion of message ferrying was developed by Zhao and Ammar (2003). *Ferrying* means that a mobile node physically stores and carries data from one location to another Frew et al. (2006), Frew and Brown (2008). This has the advantage of extended range and possibly increased total data transmission rate compared to direct communication, see Figure 1. Henkel and Brown (2006) studied route design for aerial data ferrying nodes. The route designs were evaluated analytically for relative comparison based on node velocity, data rate, and buffer size. In this paper ferrying of data is incorporated into the optimization problem. This makes the optimization problem much more complex, as it is necessary for instance to keep track of which node stores the data, how much data is stored on each node and how long data has been in the network. The reason for this is physical constraints on the storage capacity of each node and desired limitation on the time taken from when the data is collected by the UAV until it is received at the base station. On the upside, ferrying means increased flexibility with respect to coordination between vehicles compared to relaying, and it also means that the surveillance range of a mission can be extended beyond the line-of-sight distance of a chain of relaying vehicles.

As in Grøtli and Johansen (2012b) and Grøtli and Johansen (2012c) we analyze the planned paths in a radio propagation simulator to get a more accurate prediction of the radio path loss between the vehicles, and between the vehicles and the base station. In the MILP formulation the ability to communicate at a certain data rate depends on the distance between the nodes. For this distance to give a realistic picture of the communication properties, the paths found by solving the MILP problem are analyzed at every time step using the radio path loss simulator SPLAT! Maglicane (2010 (Accessed August 18, 2010), Wright (2011 (Accessed June 29, 2011). SPLAT! uses digital elevation data to calculate field strength and path loss based on the Longley-Rice Irregular Terrain Model Longley and Rice (1968). If the path loss estimate calculated by SPLAT! is too high to maintain communication at the desired rate, the communication constraints of the MILP problem are tightened. This means that the maximal distance where communication is assumed feasible is reduced by a certain value. The process is then repeated until paths are found in which communication can be maintained at a predefined criterion during the servicing of the tasks. We emphasize that we have proposed an offline algorithm, which provides coarse motion- and communication-planning for the UAVs. In Beard and McLain (2012) they discriminate between two different approaches to motion planning: "*deliberative* motion planning, where explicit paths and trajectories are computed based on global world knowledge, and *reactive* motion planning, which uses behavioral methods to react to local sensor information". Our method falls in the first category as it is designed for preplanning of trajectories, which then can serve as an input towards reactive planning methods. For instance the online re-planning methods described in Grancharova et al. (2012, 2014) are well suited for improving robustness towards inaccuracies in the model and uncertainties that become apparent in real-time. Many other methods have been proposed in order to solve the motion planning problem. For an extensive overview over path- and motion-planning techniques the interested reader is referred to LaValle (2006) or Tsourdos et al. (2010). In our opinion MILP is well suited for complex missions as outlined here. Particular benefits of MILP are that logics, approximations of nonlinear functions, and non-convexity can easily be handled, Richards and How (2005); Bemporad and Morari (1999); Williams (1999). Another important important property is the ability to define hard constraints. Other path- and motion-planning methods do not easily handle hard inequality constraints. Instead, dissatisfied constraints are penalized in the cost function. This makes these methods hard to tune, because of the many (often competing) objectives that must be weighted appropriately. Also, many optimization methods are not easily modified to allow for coordination between multiple vehicles. Examples of popular methods used for path- or motion planning include rapidly exploring random trees, LaValle and Kuffner

(2001); Kuwata et al. (2009), particle swarm optimization Kennedy and Eberhart (1995); Ho et al. (2013); Saska et al. (2006), $A^\star$ and $D^\star$ Hart et al. (1968); Stentz (1994); Likachev et al. (2005) or variants of these. The performance of these methods relies heavily on the choice of a good heuristic potential. Also the application to the specific mission in this paper is challenging because of the above-mentioned reasons. Their primary strengths are within real-time local path planning, a category where many methods are shown to outperform MILP, Thunberg et al. (2008). Finally, the fact that the MILP provides a globally optimal solution makes it useful for benchmarking other optimization methods, *i.e* how far away is the solution of other sub-optimal methods.

2 Contribution

The main benefit of the proposed approach compared to earlier works, lies in the flexibility obtained by allowing for direct transmission, relaying and ferrying of data in the same framework. To our knowledge, we present a novel approach to an optimization problem by combining motion- and data-transmission planning while incorporating important constraints on vehicle motion (visiting of waypoints, vehicle speed, height above ground) and communication parameters (bandwidth, buffer size, path loss, delay, etc.). Although there are several authors who consider a combined path- and communication-planning problem, the ability to communicate is typically approximated as a function of the relative distance between vehicles. Even though this is also true for the MILP problem proposed in this paper, we emphasize that all paths will eventually have to satisfy a radio propagation path loss requirement in the simulator SPLAT! as described above.

In Grøtli and Johansen (2012b) we dealt with motion planning in order to create a relay chain between two stationary ground- or surface nodes. In comparison, in this paper we have presented a solution to a surveillance mission planning problem. In order to achieve the main objective of surveillance of all tasks as efficiently as possible, the means of communication (direct, relaying, ferrying) during the mission is chosen based on important communication parameters such as buffer size, delay, and bandwidth. We point out that this paper is an extended version of our conference paper, Grøtli and Johansen (2012a). The latter paper lacked a detailed description of many of the constraints of the MILP optimization problem due to space constraints. These are now included. In addition, we have extended the simulations section to cover both when the UAVs have the possibility to store data for later transmission, and when the transmission has to be immediate.

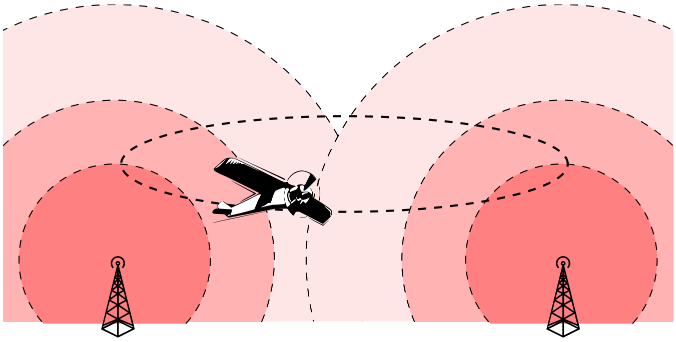


Figure 1: This figure illustrates how *delay tolerant networking* can be used not only to extend the surveillance range, but also to increase the total data transmission rate. In the dark red areas the UAV can communicate with the base stations at a high transmission rate, whereas the transmission rate is poor in the light red areas. By bringing the UAV into range of the dark red areas, the total data transmission rate can be substantially increased even when taking into account the UAV’s flight time. To quote Tanenbaum (2003): “Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway”. The process of storing data on a mobile node and physically bringing it from one location to another is called *ferrying*.

3 MILP Problem formulation

The mission objective is to perform data acquisition along sequences of waypoints by UAVs, and real-time transmission of sensor data back to the base station. The waypoints are not necessarily within communication range for direct transmission, and we will therefore allow for one or more UAVs to function as relay links. We will assume that the tasks consist of segments (e.g. road, railway or power line segments), which can be described by one or more waypoints. The interest in real-time transmission of sensor data while observing these segments, is to allow for a human operator to intervene if something irregular is found in the sensor data.

3.1 Notation

Abbreviations:

- MILP - Mixed Integer Linear Programming
- ECEF - Earth Centered, Earth Fixed
- ENU - East-North-Up

- TIN - Triangulated Irregular Network

Constants:

- n_p - total number of vehicles
- N - optimization horizon
- Δt - sample time
- $\underline{V}_p$, $\overline{V}_p$ - minimum and maximum velocities, respectively, of vehicle p
- D^{vel} - discretization level of velocity vector approximation
- M^{vel}_{pkl} - big-M constant related to approximation constraints for velocity of vehicle p, where $k \in \{1, \ldots, D^{\text{vel}}\}$, and $l \in \{1, \ldots, D^{\text{vel}}/2\}$
- $\boldsymbol{r}_p \in \mathbb{R}^3_{\geq 0}$ - a nonnegative weighting vector for the acceleration term of the cost function
- n_t - number of tasks
- $\underline{x}, \underline{y}, \underline{z}$ and $\overline{x}, \overline{y}, \overline{z}$ - lower and upper bounds, respectively, on the state vector in the east, north and up directions
- d_x, d_y and d_z - safety distances between vehicles in the east, north and up directions, respectively
- M^{col}_1, M^{col}_2, M^{col}_3 - big-M constants related to anticollision constraints
- M^{finish} - big-M constant related to related to constraints on time steps elapsed before the vehicle returns to landing site
- d^{TIN} - safety distance between the UAVs and the ground
- T^{TIN} - number of non-overlapping triangles used to represent the TIN
- m^{TIN} - number of vertices used to represented the TIN
- $P_l = (x^{\text{TIN}}_l, y^{\text{TIN}}_l, h^{\text{TIN}}_l)^\top$ - vertex $l \in \{1, \ldots, m^{\text{TIN}}\}$ in the TIN
- $t^{\text{separation}}$ - number of time steps separating UAVs arrival at the final waypoint
- $n_{\mathcal{W}_t}$ - number of waypoints of task $t \in \mathcal{T}^{n_t}_1$
- d^{wp} - distance from a waypoint to the side of a cube, which within a UAV is considered to have visited the waypoint
- $M^{\text{wp}}_{pw1}, \ldots, M^{\text{wp}}_{pw6}$ - big-M constants related to vehicle p flying through waypoint w
- c^{sensor} - the rate at which sensor data is gathered for later (or immediate) transmission back to the base station
- $\overline{h}_p$ - maximum storage capacity on node p
- t^{delay} - maximum number of time steps before collected data should be forwarded
- $C^{\max}$ is the maximum data rate between any two nodes
- $C^{\text{max\_out}}$, $C^{\text{max\_in}}$ - maximum data rate in to, respectively out from, any node
- D^{con} - discretization level of connectivity vector approximation
- R^{con}_{qpi} - connectivity distance between node q and node p at some time instance i

Index sets:

- $\mathcal{P}^b_a = \{a, a+1, \ldots, b\} \quad a, b \in \mathbb{Z}$ - set of vehicle/node indices. We consider n_p vehicles, each equipped with a communication node. The set of indices $\mathcal{P}^{n_p}_1$ refer, depending on the situation, *either* to the physical vehicles *or* the communication nodes onboard these vehicles. In addition we consider one communication node at the base station, which is given the index $\{n_p + 1\}$, such that the index set of all communication nodes is $\mathcal{P}^{n_p+1}_1 = \mathcal{P}^{n_p}_1 \cup \{n_p + 1\}$
- $\mathcal{I}^b_a = \{a, a+1, \ldots, b\} \quad a, b \in \mathbb{Z}$, - sample time index
- $\mathcal{D}_l$ - set of indices to triangles that have a common vertex P_l, where $l \in \{1, \ldots, m_l\}$
- $\mathcal{T}^b_a = \{a, a+1, \ldots, b\} \quad a, b \in \mathbb{Z}$ - set of task indices
- $\mathcal{W}_t$ - indices of the waypoints belonging to task t, where $t \in \mathcal{T}^{n_t}_1$. Every waypoint belongs to one and only one task, that is $\mathcal{W}_r \cap \mathcal{W}_s = \emptyset \ \forall r, s \in \mathcal{T}^{n_t}_1$ and $r \neq s$
- $\mathcal{T}^+$ - tasks with more than one waypoint
- $\mathcal{W}^{\text{first}}_t, \mathcal{W}^{\text{last}}_t$ - index to the first, respectively, the last waypoints of task $t \in \mathcal{T}^+$

Optimization variables:

- $\boldsymbol{p}_{pi} := (p_{1pi}, p_{2pi}, p_{3pi}) := (x_{pi}, y_{pi}, z_{pi})^\top$ - position vector of vehicle p and time step i along the axes of a local East-North-Up (ENU) coordinate reference frame, see Figure 3
- $\boldsymbol{v}_{pi} := (v_{1pi}, v_{2pi}, v_{3pi})^\top$ - velocity of vehicle p at time step i along the axes of a local ENU coordinate reference frame

- V_{pi} is an approximation of the magnitude of the velocity vector $\boldsymbol{v}_{pi}$
- b^{vel}_{pikl} - binary variable. When *true*, the projection of the velocity vector $\boldsymbol{v}_{pi}$, $p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_1^N$, onto the unit vector ξ_{kl}, $k \in \{1, \dots, D^{\text{vel}}/2\}$, $l \in \{1, \dots, D^{\text{vel}}\}$ is greater than V_p/α^{vel}
- b^{wp}_{piw} - binary variable. When *true*, vehicle $p \in \mathcal{P}_1^{n_p}$ flies through waypoint $w \in \mathcal{W}_1 \cup \ldots \cup \mathcal{W}_{n_t}$ at time step $i \in \mathcal{I}_1^N$
- b^{col}_{pqil} - binary variable. When *false*, the relative distance between vehicle $p \in \mathcal{P}_1^{n_p-1}$ and vehicle $q \in \mathcal{P}_p^{n_p}$, in at least one of the direction of the ENU frame is greater than the minimum separation distance
- $\boldsymbol{w}^{\text{acc}}_{pi} := (w^{\text{acc}}_{1pi}, w^{\text{acc}}_{2pi}, w^{\text{acc}}_{3pi})^\top$ - vector of slack variables, used to penalize the acceleration in each direction of the ENU frame
- $\lambda^{\text{TIN}}_{pil} \in [0, 1]$ - barycentric area coordinates
- $b^{\text{TIN}}_{pit} \in \{0, 1\}$ - binary variable which by constraints is forced to be true if and only if vehicle $p \in \mathcal{P}_1^{n_p}$ flies over triangle $t \in T^{\text{TIN}}$ at time step $i \in \mathcal{I}_1^N$
- h_{pi} - height of the terrain immediately below vehicle $p \in \mathcal{P}_1^{n_p}$ at time step $i \in \mathcal{I}_1^N$
- θ_w - time steps elapsed before waypoint $w \in \mathcal{W}_1 \cup \ldots \cup \mathcal{W}_{n_t-1}$ is visited
- θ^{finish}_p - time steps elapsed before vehicle $p \in \mathcal{P}_1^{n_p}$ returns to the landing site
- η^{finish} - time steps elapsed before the last vehicle returns to the base station
- b^{task}_{pt} - binary variable which is *true* if and only if task $t \in \mathcal{T}_1^{n_t-1}$ is served by vehicle $p \in \mathcal{P}_1^{n_p}$
- b^{sensor}_{pi} - binary variable which is *true* if and only if vehicle $p \in \mathcal{P}_1^{n_p}$ is serving a task at time step $i \in \mathcal{I}_1^N$
- $\lambda'_{pit} \in [0, 1]$ - auxiliary variable which is 1 at the time step $i \in \mathcal{I}_1^N$ at which vehicle $p \in \mathcal{P}_1^{n_p}$ is servicing task $t \in \mathcal{T}_1^{n_t-1}$
- m_{pisj} - the amount of the message with source node $s \in \mathcal{P}_1^{n_p}$, created at time step $j \in \mathcal{I}_1^N$, which is stored on node $p \in \mathcal{P}_1^{n_p+1}$ at time step $i \in \mathcal{I}_j^N$
- c_{pqisj} - the transmission rate from node $p \in \mathcal{P}_1^{n_p+1}$ to node $q \in \mathcal{P}_1^{n_p+1}$ at time step $i \in \mathcal{I}_j^N$ of the message with source node $s \in \mathcal{P}_1^{n_p}$ created at time step $j \in \mathcal{I}_1^N$
- $\tilde{b}^{\text{con}}_{pqi}$ - binary variable which is *true* if and only if node $q \in \mathcal{P}_1^{n_p+1}$ is within communication distance of node $p \in \mathcal{P}_1^{n_p}$ at time step $i \in \mathcal{I}_1^N$
- $\boldsymbol{\chi}_{pqi} := (x_{pi} - x_{qi}, y_{pi} - y_{qi}, z_{pi} - z_{qi})^\top$ - vector of relative distance between node $p \in \mathcal{P}_1^{n_p+1}$ and node $q \in \mathcal{P}_1^{n_p+1}$ along each of the ENU coordinate directions at time step $i \in \mathcal{I}_1^N$

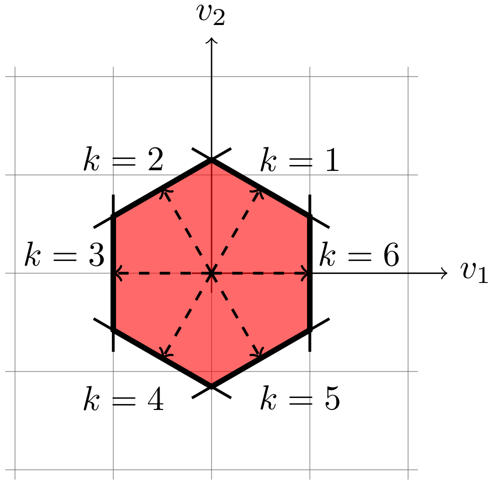


Figure 2: The figure illustrates the constraints $\boldsymbol{v}_{pi}^\top \boldsymbol{\xi}_{kl} \leq 1$ in the horizontal plane, for some particular $p \in \mathcal{P}_1^{n_p}$, some particular $i \in \mathcal{I}_0^{N-1}$, $\forall k \in \{1, \dots, D^{\text{vel}} = 6\}$, and $l = 1$. The dashed arrows illustrate the unit vectors, $\boldsymbol{\xi}_{k1}$, $k \in \{1, \dots, D^{\text{vel}} = 6\}$.

3.2 Vehicle model

In the planning problem we assume that the p^{th} UAV is described by the discrete time model

$$\boldsymbol{p}_{p(i+1)} = \boldsymbol{p}_{pi} + \Delta t \boldsymbol{v}_{pi}\,, \tag{1}$$

$\forall p \in \mathcal{P}_1^{n_p}, \forall i \in \mathcal{I}_0^{N-1}$, where n_p is the number of UAVs, Δt is the sample time, and $\boldsymbol{p}_{pi}$ and $\boldsymbol{v}_{pi}$ are vectors with elements being positions and velocities along the orthogonal axes of a local East-North-Up (ENU) coordinate reference frame. The simplicity of the model makes it suitable for the optimization problem described in this paper, where we focus on *coarse* path planning. The relationship between the ENU and the Earth-Centered Earth-Fixed (ECEF) frames is explained in Figure 3. The ECEF frame, is a coordinate system where its origin is at the center of Earth, and with axes X, Y, Z, rotating with the angular velocity of Earth. The ENU coordinate frame is a local

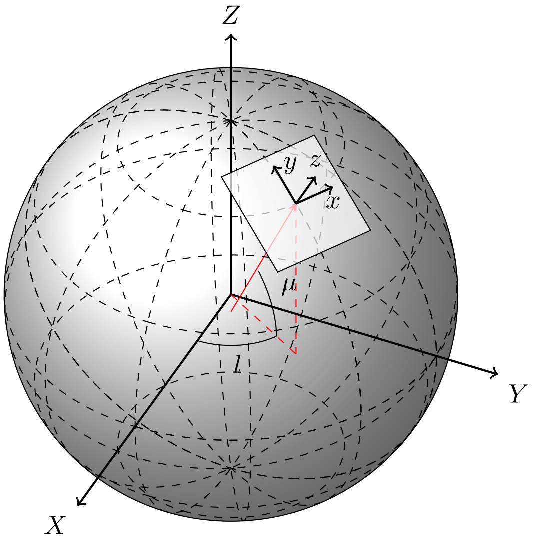


Figure 3: The Earth-Centered Earth-Fixed (ECEF) frame with axes X, Y, Z, and the East-North-Up (ENU) with axes x, y, z. The origin of ENU frame is typically represented in terms of the reference ellipsoidal parameters longitude l, and geodetic latitude μ.

geodetic coordinate system whose tangent plane is fitted to the geodetic reference ellipsoid at some convenient point for local measurements. The x axis points towards East, the y axis points towards North and the z axis completes the right-handed orthogonal frame by pointing away from the Earth perpendicular to the reference ellipsoid. The origin of the ENU frame is typically represented in terms of the reference ellipsoidal parameters longitude l, and geodetic latitude μ.

3.3 Velocity constraints

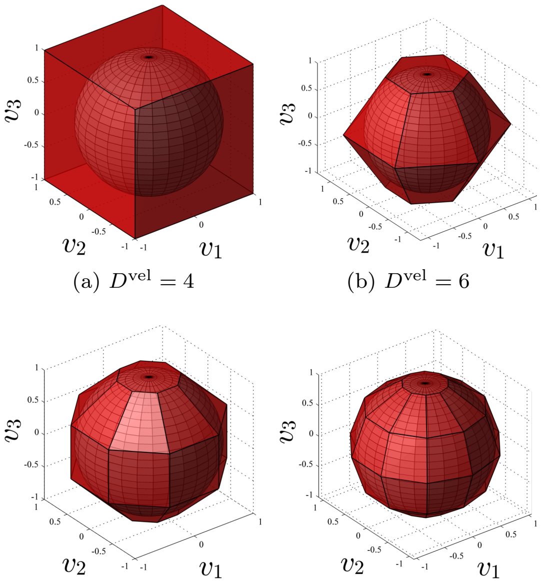
We approximate V_{pi} in a similar manner as in Chaudhry et al. (2004), here in the three-dimensional case as in Grøtli and Johansen (2012b), by introducing the constraints:

$$\boldsymbol{v}_{pi}^{\top}\boldsymbol{\xi}_{kl} \leq V_{pi}\,, \tag{2}$$

$$\alpha^{\mathrm{vel}}\boldsymbol{v}_{pi}^{\top}\boldsymbol{\xi}_{kl} \geq V_{pi} - M_{pkl}^{\mathrm{vel}}(1 - b_{pikl}^{\mathrm{vel}})\,, \tag{3}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_0^{N-1}$, $k \in \{1,\dots,D^{\mathrm{vel}}\}$, $l \in \{1,\dots,D^{\mathrm{vel}}/2\}$, and

$$\sum_{k=1}^{D^{\mathrm{vel}}}\sum_{l=1}^{D^{\mathrm{vel}}/2} b_{pikl}^{\mathrm{vel}} = 1\,, \tag{4}$$



(a) $D^{\mathrm{vel}} = 4$

(b) $D^{\mathrm{vel}} = 6$

(c) $D^{\mathrm{vel}} = 8$

(d) $D^{\mathrm{vel}} = 10$

Figure 4: The accuracy of the approximation in (2) depends on the discretization level D^{vel}. Figures 4a ($D^{\mathrm{vel}} = 4$), 4b ($D^{\mathrm{vel}} = 6$), 4c ($D^{\mathrm{vel}} = 8$), 4d ($D^{\mathrm{vel}} = 10$) show that by including all constraints $\forall l \in \{1,\dots,D^{\mathrm{vel}}/2\}$ the velocity vector is restricted to be within a polytope approximating the unit sphere in 3D space.

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_0^{N-1}$, where b_{pikl}^{vel} are binary optimization variables, the unit vector

$$\boldsymbol{\xi}_{kl} := \begin{bmatrix} \cos(\theta_k)\sin(\phi_l) \\ \sin(\theta_k)\sin(\phi_l) \\ \cos(\phi_l) \end{bmatrix}\,, \tag{5}$$

with $\theta_k := 2\pi k/D^{\mathrm{vel}}$, $\phi_l := 2\pi l/D^{\mathrm{vel}}$, $k \in \{1,\dots,D^{\mathrm{vel}}/2\}$, $l \in \{1,\dots,D^{\mathrm{vel}}\}$ and the discretization level D^{vel} is some constant even integer greater or equal to four.

By (2), V_{pi} is constrained to be larger than the scalar projection of $\boldsymbol{v}_{pi}$ onto any of the unit vectors $\boldsymbol{\xi}_{kl}$. To illustrate this constraint, consider the simpler constraint $\boldsymbol{v}_{pi}^{\top}\boldsymbol{\xi}_{kl} \leq 1$ in the horizontal plane in Figure 2, for some particular $p \in \mathcal{P}_1^{n_p}$, some particular $i \in \mathcal{I}_0^{N-1}$, $\forall k \in \{1,\dots,D^{\mathrm{vel}} = 6\}$, and $l = 1$. The dashed arrows illustrate the unit vectors, $\boldsymbol{\xi}_{k1}$, $k \in \{1,\dots,D^{\mathrm{vel}} = 6\}$. These constraints ensure that the projection of the velocity $\boldsymbol{v}_{pi}$ onto the unit vectors $\boldsymbol{\xi}_{k1}$ (the scalar product of $\boldsymbol{v}_{pi}$ and $\boldsymbol{\xi}_{k1}$) are less or equal to 1. Hence, the velocity vector should be within the red area in the horizontal plane, which is a polygonal approximation to the unit circle. The constant D^{vel} is required to be an even integer greater or equal to four in order to keep symmetry about both axes in the horizontal plane. In the constraints in (2) the right-hand side is substituted with the speed V_{pi}, which in turn is constrained by $V_{pi} \leq \overline{V}_p$, a constant maximum allowed speed for vehicle p.

In (3) V_p is required to be less than $\alpha^{\text{vel}} \boldsymbol{v}_{pi}^\top \boldsymbol{\xi}_{kl}$, *or* the corresponding binary variable b_{pikl}^{vel} must be zero. To prevent (3) to be trivially satisfied with all b_{pikl}^{vel} being zero, we add the additional constraints (4). The accuracy of the approximation depends of course on the discretization level D^{vel} (see Figure 4), but also on α^{vel}, a constant slightly greater than one. The closer to one α^{vel} is, the better is the approximation; however, taking it too close may have a negative impact on the computation time of the MILP problem, Chaudhry et al. (2004). The constants M_{pkl}^{vel} should be chosen sufficiently large.

Remark 1 *Here, and in the rest of the document, we mean by* sufficiently large *(or* sufficiently small*) in this context that the constant should be chosen large (small) enough to maintain the original logical implication the constraint is meant to realize. Consider for instance the constraint $f(x) \leq 0$, where $f : \mathbb{R}^n \to \mathbb{R}$ is linear, and $x \in \mathcal{X}$ for a given bounded set $\mathcal{X}$. Then a sufficiently large (small) constant can be chosen as $M = \max_{x \in \mathcal{X}} f(x)$ ($m = \min_{x \in \mathcal{X}} f(x)$), Bemporad and Morari (1999). Although M (m) in theory could be taken to be arbitrarily large (small), this is not recommended for computational efficiency, Williams (1999).*

In YALMIP Löfberg (2008), a MATLAB toolbox for implementing optimization problems, logic implications can be expressed instead of big-M formulations such as (3), and YALMIP will automatically derive big-M coefficients by analyzing the constraints on the other variables involved in the expression. As the speed of the vehicles is approximated by (2), (3) and (4), we simply use that

$$\underline{V}_p(1 - b_{piw}^{\text{wp}}) \leq V_{pi} \leq \overline{V}_p(1 - b_{piw}^{\text{wp}}) \,, \tag{6}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_0^{N-1}$, $w \in \mathcal{W}_{n_t}$, where $\underline{V}_p$ and $\overline{V}_p$ are the minimum and maximum velocities, respectively, of vehicle p. If the binary variable b_{piw}^{wp} is *true*, this means that vehicle p is visiting waypoint w at time step i. More on this implication and visiting of waypoints is postponed and considered in Section 3.8. Equation (6) will constrain the velocity of the vehicles to zero when they have arrived at the waypoint $w \in \mathcal{W}_{n_t}$, which is the final waypoint of the mission.

3.4 Acceleration cost

To avoid fluctuations in the speed, we introduce the following cost function similar to the one proposed in Schouwenaars et al. (2001),

$$J^{\text{acc}} = \sum_{p \in \mathcal{P}_1^{n_p}} \sum_{i \in \mathcal{I}_0^{N-2}} \boldsymbol{r}_p^\top \boldsymbol{w}_{pi}^{\text{acc}} \,, \tag{7}$$

with the additional constraints

$$\begin{aligned} (v_{jpk} - v_{jpi}) &\leq w_{jpi}^{\text{acc}} \,, & (8)\\ -(v_{jpk} - v_{jpi}) &\leq w_{jpi}^{\text{acc}} \,, & (9) \end{aligned}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_0^{N-2}$, $k = i + 1$, $j \in \{1, 2, 3\}$ where $\boldsymbol{w}_{pi}^{\text{acc}} := (w_{1pi}^{\text{acc}}, w_{2pi}^{\text{acc}}, w_{3pi}^{\text{acc}})^\top$ and $\boldsymbol{r}_p \in \mathbb{R}_{\geq 0}^3$ is a nonnegative weighting vector. The motivation behind (7) is to penalize the absolute value of acceleration in each direction of the ENU frame. Also, to avoid a piecewise linear cost function, we have introduced slack variables w_{jpi}^{acc}.

3.5 Position constraints

There are typically restrictions on where UAVs are allowed to fly. This may for instance be air space used for other air traffic, air space over a certain altitude, or air space over populated areas. In addition the operator might want to avoid flying into regions with bad weather, outside the area where the operator is able to communicate with UAVs or in case of military applications: areas with enemies and enemy radars. If the region the UAVs are required to stay within is convex (e.g. a rectangular box), the constraints may simply be written

$$\begin{aligned} \underline{x} \leq & x_{pi} \leq \overline{x} \,, & (10)\\ \underline{y} \leq & y_{pi} \leq \overline{y} \,, & (11)\\ \underline{z} \leq & z_{pi} \leq \overline{z} \,, & (12) \end{aligned}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_1^N$, where $\underline{x}, \underline{y}, \underline{z}$ and $\overline{x}, \overline{y}, \overline{z}$ are the constant lower and upper bounds, respectively, on the state vector in the east, north and up directions. More generally, unions of convex sets can be implemented.

3.6 Anti-collision constraints

To avoid collision between vehicles we will implement the method of Schouwenaars et al. (2001). Let the position of vehicle p and vehicle q at time step i be given by (x_{pi}, y_{pi}, z_{pi}) and (x_{qi}, y_{qi}, z_{qi}), respectively. The constraints on their relative position are then given as

$$d_x - M_1^{\text{col}} b_{pqi1}^{\text{col}} \le x_{pi} - x_{qi} \le M_1^{\text{col}} b_{pqi2}^{\text{col}} - d_x \,, \tag{13}$$

$$d_y - M_2^{\text{col}} b_{pqi3}^{\text{col}} \le y_{pi} - y_{qi} \le M_2^{\text{col}} b_{pqi4}^{\text{col}} - d_y \,, \tag{14}$$

$$d_z - M_3^{\text{col}} b_{pqi5}^{\text{col}} \le z_{pi} - z_{qi} \le M_3^{\text{col}} b_{pqi6}^{\text{col}} - d_z \,, \tag{15}$$

$$\sum_{l=1}^{6} b_{pqil}^{\text{col}} \le 5 \tag{16}$$

$\forall p \in \mathcal{P}_1^{n_p-1}$, $q \in \mathcal{P}_{p+1}^{n_p}$, $i \in \mathcal{I}_1^N$, where d_x, d_y and d_z are the safety distances in the east, north and up directions, respectively. These safety distances represents the separation required to still maintain the ability to perform avoidance maneuvers. The binary variables b_{pqil}^{col}, ensure that there is a minimum separation distance between the vehicles in at least one of the directions of the ENU frame. The constant M_1^{col} should be taken sufficiently large, see Remark 1, for instance $M_1^{\text{col}} > \overline{x} - \underline{x} + d_x$ with $\overline{x}, \underline{x}$ as in (10), and correspondingly for M_2^{col} and M_3^{col}.

3.7 Anti-grounding constraints

As in Ma and Miller (2006) we will represent the terrain as a triangulated irregular network (TIN). Terrain avoidance constraints in MILP form are given in Shengxiang and Hailong (2008), and will be used here. T^{TIN} non-overlapping triangles with m^{TIN} vertices $P_l(x_l^{\text{TIN}}, y_l^{\text{TIN}}, h_l^{\text{TIN}})$ are used to represent the piecewise affine terrain surface. The point strictly below vehicle p at time step i is given by (x_{pi}, y_{pi}, h_{pi}), and satisfy

$$x_{pi} = \sum_{l=1}^{m^{\text{TIN}}} \lambda_{pil}^{\text{TIN}} x_l^{\text{TIN}} \,, \tag{17}$$

$$y_{pi} = \sum_{l=1}^{m^{\text{TIN}}} \lambda_{pil}^{\text{TIN}} y_l^{\text{TIN}} \,, \tag{18}$$

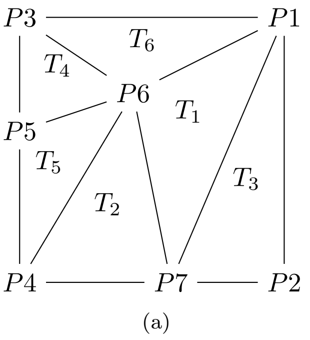
$$h_{pi} = \sum_{l=1}^{m^{\text{TIN}}} \lambda_{pil}^{\text{TIN}} h_l^{\text{TIN}} \,, \tag{19}$$

$$0 \le \lambda_{pil}^{\text{TIN}} \le 1 \quad \forall l = \{1, \dots, m^{\text{TIN}}\} \tag{20}$$

$$\sum_{l=1}^{m^{\text{TIN}}} \lambda_{pil}^{\text{TIN}} = 1 \,, \tag{21}$$

$$\sum_{t=1}^{T^{\text{TIN}}} b_{pit}^{\text{TIN}} = 1 \,, \tag{22}$$

$$\lambda_{pil}^{\text{TIN}} \le \sum_{t \in \mathcal{D}_l} b_{pit}^{\text{TIN}} \quad \forall l = \{1, \dots, m^{\text{TIN}}\} \,, \tag{23}$$



$\mathcal{D}_1 = \{1, 3, 6\}$
$\mathcal{D}_2 = \{3\}$
$\mathcal{D}_3 = \{4, 6\}$
$\mathcal{D}_4 = \{5, 2\}$
$\mathcal{D}_5 = \{4, 5\}$
$\mathcal{D}_6 = \{6, 1, 2, 4, 5\}$
$\mathcal{D}_7 = \{2, 1, 3\}$

(b)

Figure 5: In 5a a top view of a TIN with vertices P_l, $l \in \{1, \dots, 7\}$ is presented, and 5b shows the corresponding sets $\mathcal{D}_l$ of indices to triangles which have P_l as a common vertex.

$\forall p \in \mathcal{P}_1^{n_p}, i \in \mathcal{I}_1^N$. $\mathcal{D}_l$ is the set of indices of triangles that have a common vertex P_l. An example of the enumeration of triangles, vertices and the corresponding sets D_l is shown in Figure 5. Equations (17)-(21) describe the position strictly below the vehicle in terms of its barycentric coordinates, as illustrated in Figure 6. By (22), the binary variables b_{pit}^{TIN} are forced to be *true* if and only if vehicle p flies over triangle t at time step i. By (23), the variables λ_{pil} corresponding to vertices which are not adjacent to the particular triangle are set to zero. Finally, the terrain avoidance constraint can be expressed as

$$z_{pi} \ge h_{pi} + d^{\text{TIN}} \,, \tag{24}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_1^N$, where d^{TIN} is the minimum vertical distance from the UAVs to the ground. The TIN is generated from the elevation data by incremental Delaunay triangulation. This reduces the complexity of the problem, since only a subset of the available data is used in the MILP formulation.

3.8 Task assignment

We will assume that there are n_t tasks, and that each task $t \in \mathcal{T}_1^{n_t}$ is comprised of a set of waypoint indices. Let $\mathcal{W}_t$ denote the set of the indices of the waypoints which belong to task $t \in \mathcal{T}_1^{n_t}$. A special meaning is given to the final task $\mathcal{W}_{n_t}$. It contains only one waypoint, which is located above the designated landing site for the UAVs. Take-off and landing is not considered in this paper, but assumed to be handled separately. We require that $\mathcal{W}_r \bigcap \mathcal{W}_s = \emptyset$, $\forall r, s \in \mathcal{T}_1^{n_t}$, $r \neq s$, i.e. that each waypoint belongs to one and only one task. A waypoint characterized by the ENU coordinates $(x_{1w}^{\text{wp}}, y_{2w}^{\text{wp}}, z_{3w}^{\text{wp}})$ is considered to be visited if a UAV is flying through a cube containing the waypoint. More precisely, we assume each waypoint to be a cube with sides of length $2d^{\text{wp}}$, and require that Richards

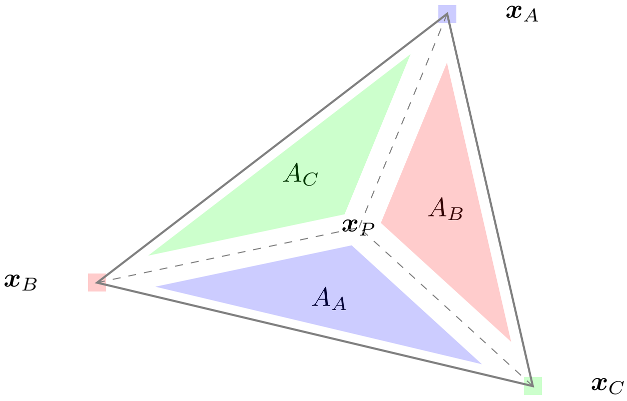


Figure 6: Let $\boldsymbol{x}_A$, $\boldsymbol{x}_B$ and $\boldsymbol{x}_C$ be the position vectors of the blue, red and green vertices of the triangle, respectively. The position of any point $\boldsymbol{x}_P$ on the triangle is given by $\boldsymbol{x}_P = \lambda_A \boldsymbol{x}_A + \lambda_B \boldsymbol{x}_B + \lambda_C \boldsymbol{x}_C$, where $\lambda_A = \frac{A_A}{A}$, $\lambda_A = \frac{A_B}{A}$, and $\lambda_C = \frac{A_C}{A}$. A_A, A_B, A_C are the areas of the blue, red and green subtriangles, respectively, and A is the total area of the triangle such that $A = A_A + A_B + A_C$. Hence, $\lambda_A + \lambda_B + \lambda_C = 1$.

and How (2002)

$$x_{pi} - x_w^{\text{wp}} - d^{\text{wp}} \le M_{pw1}^{\text{wp}}(1 - b_{piw}^{\text{wp}}) \,, \tag{25}$$
$$-x_{pi} + x_w^{\text{wp}} - d^{\text{wp}} \le M_{pw2}^{\text{wp}}(1 - b_{piw}^{\text{wp}}) \,, \tag{26}$$
$$y_{pi} - y_w^{\text{wp}} - d^{\text{wp}} \le M_{pw3}^{\text{wp}}(1 - b_{piw}^{\text{wp}}) \,, \tag{27}$$
$$-y_{pi} + y_w^{\text{wp}} - d^{\text{wp}} \le M_{pw4}^{\text{wp}}(1 - b_{piw}^{\text{wp}}) \,, \tag{28}$$
$$z_{pi} - z_w^{\text{wp}} - d^{\text{wp}} \le M_{pw5}^{\text{wp}}(1 - b_{piw}^{\text{wp}}) \,, \tag{29}$$
$$-z_{pi} + z_w^{\text{wp}} - d^{\text{wp}} \le M_{pw6}^{\text{wp}}(1 - b_{piw}^{\text{wp}}) \,, \tag{30}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_1^N$, $w \in \mathcal{W}_1 \cup \ldots \cup \mathcal{W}_{n_t}$, where $M_{pw1}^{\text{wp}} \ldots M_{pw6}^{\text{wp}}$ are chosen sufficiently large. This way, the binary variable b_{piw}^{wp} is *true* then vehicle p flies through waypoint w at time step i. We will require that each waypoint of tasks $\mathcal{W}_t, \forall t \in \mathcal{T}_1^{n_t-1}$, is visited once and once only. Mathematically this is formulated by the equality constraint

$$\sum_{p \in \mathcal{P}_1^{n_p}} \sum_{i \in \mathcal{I}_1^N} b_{piw}^{\text{wp}} = 1 \,, \tag{31}$$

$\forall w \in \mathcal{W}_1 \cup \ldots \cup \mathcal{W}_{n_t-1}$. By assigning all vehicles the final task, all vehicles will return to the landing site before the end of the mission. This assignment is ensured by the constraints

$$\sum_{i \in \mathcal{I}_1^N} b_{piw}^{\text{wp}} \ge 1 \,, \tag{32}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $w \in \mathcal{W}_{n_t}$. We also add the constraints

$$b_{p(i+1)\mathcal{W}_{n_t}}^{\text{wp}} \ge b_{piw}^{\text{wp}} \,, \tag{33}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_1^{N-1}$, $w \in \mathcal{W}_{n_t}$, which means that once b_{piw}^{wp} has become *true* for some time step i, it will remain *true* for the rest of the horizon. Together with the implication presented in (6) and (25), this also means that vehicle p will remain at the final waypoint (landing site) once it has arrived there. Given that b_{piw} is *true*, the implication in (25) constrains the position of vehicle p to the final waypoint, whereas Equation (6) constrains the velocity to zero. These constraints indirectly force all tasks $\mathcal{T}_1^{n_t-1}$ to be executed before the the last vehicle returns to the landing site. Since each waypoint $w \in \mathcal{W}_1 \cup \ldots \cup \mathcal{W}_{n_t-1}$, is visited only once, the time steps elapsed before a waypoint is visited are given by

$$\theta_w = \sum_{p \in \mathcal{P}_1^{n_p}} \sum_{i \in \mathcal{I}_1^N} i b_{piw}^{\text{wp}} \,, \tag{34}$$

$\forall w \in \mathcal{W}_1 \cup \ldots \cup \mathcal{W}_{n_t-1}$, where θ_w is a variable we have introduced in our optimization problem. Furthermore, we require that the waypoints within the same task are visited in a specific order. Let $\mathcal{T}^+$ represent those tasks with more than one waypoint. Then, $\mathcal{W}_t \backslash \mathcal{W}_t^{\text{Last}}$, $t \in \mathcal{T}^+$ contains the indices of all the waypoints of task t, except the last one. Then the visiting order can be achieved by requiring that

$$\theta_{w+1} > \theta_w \,, \tag{35}$$

$\forall w \in \mathcal{W}_t \backslash \mathcal{W}_t^{\text{Last}}$, $t \in \mathcal{T}^+$. The number of time steps elapsed before vehicle p returns to the landing site is given by θ_p^{finish}, if we use the constraints

$$\theta_p^{\text{finish}} \le M^{\text{finish}}(1 - b_{piw}^{\text{wp}}) + i b_{piw}^{\text{wp}} \,, \tag{36}$$
$$\theta_p^{\text{finish}} \ge (i+1)(1 - b_{piw}^{\text{wp}}) \,, \tag{37}$$

$\forall p \in \mathcal{P}_1^{n_p}$, $i \in \mathcal{I}_0^N$, $w \in \mathcal{W}_{n_t}$ where M^{finish} is a constant chosen sufficiently large, see Remark 1, for instance as $M^{\text{finish}} := N$. Recall that for each vehicle p, b_{piw}^{wp} may be true for many consecutive time steps i, so we cannot use the same approach as in (34) to find the time elapsed before vehicle p arrives at waypoint $w \in \mathcal{W}_{n_t}$. Instead with the upper- and lower-bounds on θ_p^{finish} given by (36) and (37), respectively, θ_p^{finish} will represent the exact time step of arrival at the final waypoint. Since we want to minimize the overall mission time - the time elapsed until the last vehicle arrives at the final waypoint - we introduce the variable η^{finish} and require that

$$\eta^{\text{finish}} \ge \theta_p^{\text{finish}} \,, \tag{38}$$

$\forall p \in \mathcal{P}_1^{n_p}$, and set our objective to minimize the cost function

$$J^{\text{finish}} = \gamma^{\text{finish}} \eta^{\text{finish}} \,, \tag{39}$$

where γ^{finish} is a positive scalar. Equation (38) can be satisfied for any sufficiently large η^{finish}. However, by minimizing η^{finish} in the cost function, we achieve the desired effect, which is to minimize the overall mission time. We do not want vehicles to arrive at the final waypoint simultaneously, as this may cause the UAVs to collide. Therefore, we also require a temporal separation between the arrival at the final waypoint, that is,

$$\theta_p^{\text{finish}} \geq \theta_q^{\text{finish}} + t^{\text{separation}} \tag{40}$$

$\forall p \in \mathcal{P}_1^{n_p - 1}, q \in \mathcal{P}_{p+1}^{n_p}$, where $t^{\text{separation}} \in \mathbb{N}$ is the number of time steps separating the UAVs at the arrival of the final waypoint. Thus far, there is nothing restricting multiple vehicles each accomplishing parts of a task. As this may be undesirable, for instance because we want video to be recorded continuously between waypoints of a task, we introduce an additional binary variable b_{pt}^{task} which is *true* if and only if task t is served by vehicle p. This is achieved by imposing the constraints

$$-\sum_{w \in \mathcal{W}_t} \sum_{i \in \mathcal{I}_1^N} b_{piw}^{\text{wp}} \leq -n_{\mathcal{W}_t} b_{pt}^{\text{task}} \tag{41}$$

$\forall p \in \mathcal{P}_1^{n_p}, \forall t \in \mathcal{T}_1^{n_t - 1}$ and

$$\sum_{p \in \mathcal{P}_1^{n_p}} b_{pt}^{\text{task}} = 1 \,, \tag{42}$$

$\forall t \in \mathcal{T}_1^{n_t - 1}$, where $n_{\mathcal{W}_t}$ is the number of waypoints of task t. Still, there is a possibility that a UAV will switch back and forth between different tasks. This behavior is allowed, but as it is shown in the simulations, it is more beneficial to accomplish one task at the time. This is due to the demanding communication constraints during the accomplishment of a task, which we will impose in Section 3.10.

3.9 Data gathering

We introduce the binary variable b_{pi}^{sensor} which is *true* if and only if vehicle p is serving a task at time step i. This implication can be achieved by using the auxiliary variable $\lambda_{pit}^{'} \in [0, 1]$ and imposing the constraints

$$\lambda_{pit}^{'} \leq \sum_{k=1}^{i} \sum_{w \in \mathcal{W}_t} b_{pkw}^{\text{wp}} \,, \quad \lambda_{pit}^{'} \leq \sum_{k=i}^{N} \sum_{w \in \mathcal{W}_t} b_{pkw}^{\text{wp}} \,, \tag{43}$$

$\forall p \in \mathcal{P}_1^{n_p}, \, i \in \mathcal{I}_1^N, \, t \in \mathcal{T}_1^{n_t - 1}$,

$$\sum_{i=1}^{N} \lambda_{pit}^{'} = \sum_{i=1}^{N} i b_{piw_1}^{\text{wp}} - \sum_{i=1}^{N} (i+1) b_{piw_2}^{\text{wp}} \,, \tag{44}$$

$\forall p \in \mathcal{P}_1^{n_p}, \, w_1 \in \mathcal{W}_t^{\text{First}}, \, w_2 \in \mathcal{W}_t^{\text{Last}}, \, t \in \mathcal{T}_1^{n_t - 1}$, and finally

$$\sum_{t \in \mathcal{T}_1^{n_t}} \lambda_{pit}^{'} = b_{pi}^{\text{sensor}} \,, \tag{45}$$

$\forall p \in \mathcal{P}_1^{n_p}, \, i \in \mathcal{I}_1^N$. $\mathcal{W}_t^{\text{First}}$ and $\mathcal{W}_t^{\text{Last}}$ represent the set with index to the first and the last waypoint, respectively, of task t. The inequalities in (43) force $\lambda_{pit}^{'}$ to be less or equal to one for every time step i during the servicing of task t, and zero otherwise. Equation (44) makes $\sum_{i \in \mathcal{I}_1^N} \lambda_{pit}^{'}$, for a specific vehicle p and a specific task t, to be equal to the number of time steps elapsed from when the vehicle visited the first waypoint of the task and until it visited the last (and where we have assumed that the servicing of the task ends at the end of the step). Together, (43) and (44) constraint $\lambda_{pit}^{'}$ to be one at the time steps i at which vehicle p is servicing task t and zero otherwise. Finally, (45) will give the variables b_{pi}^{sensor} the desired property.

3.10 Data flow for delayed transmission

In the following we will sometimes commonly refer to the base station (where the antenna for the operator user interface is located) and the UAVs as *nodes*. The communication network will be modeled as a buffered flow network, a directed graph where each edge represents a limited transmission capacity, but where the nodes have the ability to store data. We assume that while vehicle p is servicing a task - that is for those time steps i the binary variable b_{pi}^{sensor} is *true* - the rate at which sensor data is gathered for later (or immediate) transmission back to base station, is given by c^{sensor}. In our setup, the base station is the only sink, whereas the UAVs act as sources during servicing of a task. We assume that the bandwidth required for transmission is substantially larger during the execution of a task, and ignore the possible need for communication during transit between tasks. Immediate transmission of data back to the base station requires that the vehicles and therefore also the tasks lie within direct communication distance of the base station, or at least the communication distance of a *chain* of multiple UAVs. As this limits the surveillance area, we also allow for *ferrying*. This may increase the surveillance area. It could potentially also increase the data transfer rate, as the vehicles can ferry the data into a range where high bandwidth transmission is possible. The drawback is added complexity, and of course the additional delay between sensing and receiving of data. In our task assignment scenario, we require that the collected data is forwarded within t^{delay} time steps. This allows a human operator to analyze the data, and possibly command the UAVs to service the task again if the data shows something of interest.

We will assume that each vehicle could possibly service a task at any time instant over the horizon N, and that the servicing of a task would mean that the vehicle has collected some piece of data to be transmitted back to the base station. Such piece of data will from now on be referred to as a *message*, and a new message is created every time step a vehicle is servicing a task. These messages can be divided into even smaller pieces and each piece could possibly be routed differently back to the base station. It will therefore be important to label when each piece is collected in order to constrain the total time elapsed before the whole message is received by the base station. Motivated by Jain et al. (2004), we introduce the optimization variables m_{pisj} to represent the amount of the message with source node s created at time step j which is stored at node p at time step i. Similarly, c_{pqisj} is the transmission rate from node p to node q at time step i of the message with source node s created at time step j. These variables can only take on nonnegative values, and we introduce the constraints

$$m_{pisj} \geq 0\,, \tag{46}$$

$\forall p \in \mathcal{P}_1^{n_p+1}, j \in \mathcal{I}_1^N, s \in \mathcal{P}_1^{n_p}, i \in \mathcal{I}_j^N$,

$$c_{pqisj} \geq 0\,, \tag{47}$$

$\forall p, q \in \mathcal{P}_1^{n_p+1}, j \in \mathcal{I}_1^N, s \in \mathcal{P}_1^{n_p}, i \in \mathcal{I}_j^N$. The subscripts p and q can take on the value $n_p + 1$, which refers to the base station. Notice that the subscript i of the the variables m_{pisj} and c_{pqisj} start at j in (46) and (47) since no message or transmission of message can exist before the message is created. The flow equations relating the introduced variables are given by

$$m_{pisj} = \Delta t \left(c^{\text{sensor}} b_{pj}^{\text{sensor}} - \sum_{q \in \mathcal{P}_1^{n_p+1} \setminus \{p\}} c_{pqisj} \right), \tag{48}$$

$\forall s \in \mathcal{P}_1^{n_p}, j \in \mathcal{I}_1^N, i = j, p = s$, and

$$m_{pisj} = m_{p(i-1)sj} + \Delta t \left(\sum_{q \in \mathcal{P}_1^{n_p+1} \setminus \{p\}} (c_{qpisj} - c_{pqisj}) \right), \tag{49}$$

$\forall p, s \in \mathcal{P}_1^{n_p}, j \in \mathcal{I}_1^N, i \in \mathcal{I}_{j+1}^N$, and where $c_{p(n_p+1)isj}$ denotes the data rate at which the message with source s created at time step j is transmitted from vehicle p to the base station (denoted by subscript $n_p + 1$) at time step i. Equations (48) and (49) can be thought of as representing the conservation of data. In particular, if b_{pj}^{sensor} is *true* vehicle p is servicing a task at time step j and the amount of data gathered (message size) is given by $\Delta t c^{\text{sensor}}$. The amount of data immediately transmitted to other vehicles is given by the second term on the right-hand side of (48), and the left-hand side represents the amount that will be stored on vehicle p for the next time step. After the data has been gathered the amount of a specific message stored at node p will remain unchanged, unless parts of this message are transmitted to or received from other nodes, see (49). We assume that each vehicle is equipped with a buffer or a hard drive, and that the buffer can store a limited amount of data. The data, which is the sum of all messages on the node, should be less than the buffer size, $\overline{h}_p$, that is

$$\sum_{s \in \mathcal{P}_1^{n_p}} \sum_{j \in \mathcal{I}_1^i} m_{pisj} \leq \overline{h}_p \tag{50}$$

$\forall p \in \mathcal{P}_1^{n_p}, \forall i \in \mathcal{I}_1^N$. If a task is not serviced by a vehicle at a specific time instant, then the message size will be zero for the whole horizon. We achieve this with the constraints

$$m_{pisj} \leq b_{pj}^{\text{sensor}} M^{\text{msg}}\,, \tag{51}$$

$\forall p, s \in \mathcal{P}_1^{n_p}, j \in \mathcal{I}_1^N, i \in \mathcal{I}_j^N$, where the constant $M^{\text{msg}} := \Delta t c^{\text{sensor}}$ is the maximum message size. The delay criterion is ensured by the constraints

$$\sum_{i \in \mathcal{I}^N_{\min\{j+t^{\text{delay}},N\}}} \sum_{p \in \mathcal{P}_1^{n_p}} m_{pisj} = 0 \tag{52}$$

$\forall s \in \mathcal{P}_1^{n_p}, j \in \mathcal{I}_1^N$. These constraints mean that no part of a message will remain on any node $p \in \mathcal{P}_1^{n_p}$, t^{delay} time steps after it was created on a node $s \in \mathcal{P}_1^{n_p}$. Furthermore, it also enforces that no data is left on any of the vehicle nodes at the end of the optimization horizon. If instantaneous transmission is required, the constant t^{delay} should be set to 0. Since the base station is only receiving data, we require that

$$c_{(n_p+1)qisj} \leq 0\,, \tag{53}$$

$\forall q, s \in \mathcal{P}_1^{n_p}, j \in \mathcal{I}_1^N, i \in \mathcal{I}_j^N$. To reflect the fact that transmission is only possible when the different nodes are within each others' communication range, we also add to our optimization problem the constraints

$$c_{pqisj} \leq C^{\max} \tilde{b}_{pqi}^{\text{con}} \tag{54}$$

$\forall p, s \in \mathcal{P}_1^{n_p}, \forall q \in \mathcal{P}_1^{n_p+1}, j \in \mathcal{I}_1^N, i \in \mathcal{I}_j^N$, where the constant $C^{\max}$ is the maximum data rate and $\tilde{b}_{pqi}^{\text{con}}$ is a binary variable which is *true* if and only if vehicle q is within communication distance of vehicle p at time step i. The constraints required to give $\tilde{b}_{pqi}^{\text{con}}$ this property, are introduced in the section to follow. We also want to bound the collective incoming and outgoing data rate at each time step, and require that

$$\sum_{q\in\mathcal{P}_1^{n_p+1}\setminus\{p\}} \sum_{s\in\mathcal{P}_1^{n_p}} \sum_{j\in\mathcal{I}_1^i} c_{pqisj} \le C^{\mathrm{max\_out}}\,, \tag{55}$$

$$\sum_{q\in\mathcal{P}_1^{n_p}\setminus\{p\}} \sum_{s\in\mathcal{P}_1^{n_p}} \sum_{j\in\mathcal{I}_1^i} c_{qpisj} \le C^{\mathrm{max\_in}}\,, \tag{56}$$

$\forall p \in \mathcal{P}_1^{n_p}, i \in \mathcal{I}_1^N$. The constants $C^{\mathrm{max\_out}}$ and $C^{\mathrm{max\_in}}$ may be different from $C^{\max}$ in Equation (54), and they may be different for each UAV although we consider them all to be equal for all UAVs in the simulations in Section 4. Depending on the communication equipment, each node may for instance be communicating with multiple other nodes over different bands simultaneously.

Remark 2 *Notice that we have chosen to consider the raw flow between nodes in the above formulation. The advantage of considering the raw flow instead of the net flow, is that the raw flow can correctly model asymmetric flow in a duplex communication network. Raw network flows are characterized by nonnegative upper and lower capacity constraints (as in (47) and (53)-(56)), and conservation of flow (as in (48) and (49)).*

Remark 3 *In the above formulation we have not penalized the total amount of data sent in the network. This means that information can transmitted back and forth between nodes, or in loops when there are three or more communication nodes. This behavior does not influence the feasibility of the MILP problem, but the planned path may unnecessarily be considered infeasible in SPLAT! with respect to the radio path loss design criterion introduced in the next section. This behavior can for instance be avoided by penalizing in the objective function the amount of data transmitted in the network. In our case the problem will not be dominant since we penalize the total time taken from data is gathered and until it is received by the base station.*

3.11 Connectivity constraints

When formulating the MILP, we will assume that the ability of node p to successfully transmit data at a specified rate to node q, at some time instance i, depends on whether the relative distance between the two nodes is below a certain threshold. This threshold would typically depend on the antenna gains of the receiver and transmitter node, surrounding terrain, data rate, radio frequency band, etc. We stress that R_{pqi}^{con} is not necessarily equal to R_{qpi}^{con}, that is, the threshold depends on the direction of communication. Instead of requiring that node q is within a sphere of radius R_{pqi}^{con} of node p, we require that node q is within a polyhedron that approximates the sphere. The approximation is formed by taking the inner product of the vector $\boldsymbol{\chi}_{pqi} := (x_{pi} - x_{qi}, y_{pi} - y_{qi}, z_{pi} - z_{qi})^\top$ and $\boldsymbol{\xi}_{kl}$, where $\boldsymbol{\xi}_{kl}$ was defined in (5) with $\theta_k := 2\pi k / D^{\mathrm{con}}$ and $\phi_l := 2\pi l / D^{\mathrm{con}}$ and D^{con} is some constant even integer greater or equal to four. As already pointed out, we introduce binary indicator variables $\tilde{b}_{pqi}^{\mathrm{con}}$ such that

$$\tilde{b}_{pqi}^{\mathrm{con}} = 1 \iff \boldsymbol{\chi}_{pqi}^\top \boldsymbol{\xi}_{kl} - R_{pqi}^{\mathrm{con}} \le 0\,, \tag{57}$$

$\forall p, q \in \mathcal{P}_1^{n_p+1}$, $i \in \mathcal{I}_1^N$, $k \in \{1, \ldots, D^{\mathrm{con}}/2\}$, $l \in \{1, \ldots, D^{\mathrm{con}}\}$, that is, the indicator variable $\tilde{b}_{pqi}^{\mathrm{con}}$ is *true* if and only if, node p can directly transmit to vehicle q, where $p, q = n_p{+}1$, denote the base station. The logical statement in (57) can be achieved by the introduction of a number of additional optimization variables, but out of brevity we will simply refer to Bemporad and Morari (1999) for this procedure.

Initially R_{pqi}^{con} should be chosen large enough to give a feasible solution to the MILP problem. The solution will then be analyzed in SPLAT!, meaning that the radio propagation path loss between all nodes will be calculated at every time step. We denote the maximum allowed path loss between vehicle p and q to maintain the desired bandwidth during transmission by $L_{pq}^{\max}$. If, for any time step i, the path loss calculated by SPLAT! is below $L_{pq}^{\max}$ while at the same time $c_{pqi} \neq 0$, then R_{pqi}^{con} will be reduced and we solve the MILP problem again.

4 Simulations

4.1 Mission set-up

We will consider three different scenarios all including two UAVs. Scenarios 1 and 2 show simulations for the proposed method when delay tolerant communication is not allowed and allowed, respectively. Scenario 0 does not include the rather complex communication constraints we have proposed in this paper, and serves as a justification for why this is necessary in order to satisfy the communication constraints. The tasks and their corresponding waypoints for all scenarios are listed in Table 1. The initial positions of the vehicles with respect to the local ENU frame are given in Table 2, whereas the base station is located at $(3750, 1450, 175)^\top$. We assume that a human operator will bring the UAVs up to the initial position, and down from the final waypoint, as this is a common approach in practice.

Table 3 shows the parameters used in the MILP problem. The default solver parameters of the IBM CPLEX Optimizer 12.6 were used, except for `epgap` which was set to 1.0×10^{-2}. The algorithm was run on a HP EliteBook 8540w, with Intel Core i7 CPU Q720 @1.6 GHz, 16 GB RAM and a Windows 7, 64-bit operating system. Furthermore, we used MATLAB

Table 1: Tasks, and coordinates of their corresponding waypoints of Scenarios 1 and 2

Task	Waypoint coordinates
$\mathcal{W}_1$	{(4100,2150,200),(4000,2250,200),(3900,2350,200)}
$\mathcal{W}_2$	{(2800,1850,300),(2900,1950,300),(3000,2050,300)}
$\mathcal{W}_3$	{(3500,1000,250)}
$\mathcal{W}_4$	{(4000,1200,250),(4000,1300,250),(4000,1400,250),(4000,1500,250)}
$\mathcal{W}_5$	{(3000,1450,250)}

Table 2: Initial position of UAVs

UAV	Initial position
1	$(3000, 1350, 225)^\top$
2	$(3000, 1350, 300)^\top$

version R2014a and YALMIP Löfberg (2008) version R20140221. The Windows version of the radio propagation path loss simulator SPLAT! was provided by Wright (2011 (Accessed June 29, 2011), and the parameters used are shown in Table 4. SPLAT! is mainly intended for ground-based antennas. However, it is possible to adjust the antenna height, and we have therefore set the antenna height equal to the altitude of the vehicles when analyzing the radio path loss. The point-to-point radio path loss is calculated using the Irregular Terrain Model (ITM), Longley and Rice (1968), with 60% clearance of the first Fresnel zone. It should be noticed, however, that there are some constraints to the parameters used in the ITM. For instance should the antenna heights be less than 1 km above ground level, and the relative distance should be between 1 km and 1000 km. We take a conservative approach and use the maximum of the path loss calculated using ITM and the free space model. Due to the additional constraint of $d^{\text{TIN}} = 100\,\text{m}$ safety distance to the ground, the attenuation due to terrain shielding is likely to be small for short relative distances, and the free space path loss should be a good approximation to the actual path loss.

We set the maximum allowed path loss $L^{\max}_{pq} = L^{\max} = 98$ for any $p, q \in \mathcal{P}^3_1$, $p \neq q$, although it *can* be different between different nodes, and does not even have to be symmetric between a pair of nodes. We initially set $R^{\text{con}}_{pqi} = 750\ \forall p, q \in \mathcal{P}^3_1$, $p \neq q$, $i \in \mathcal{I}^N_1$. How to choose the initial value of R^{con}_{pqi} can for instance be guided by using the *area prediction mode* in SPLAT!, which gives a prediction of the regional coverage from the base station. If the calculated path loss at time step i is below $L^{\max}$ while at the same time $c_{pqi} \neq 0$ *or* $c_{qpi} \neq 0$, then $R^{\text{con}}_{qpi} = R^{\text{con}}_{pqi} = R^{\text{con}}_{pqi} - r^{\text{con}}$, so as to maintain the symmetry of R^{con} in this particular example. We use $r^{\text{con}} = 150$. The horizon is 160 s for both Scenarios 1 and 2. With a discretization step of 5 s, we find that $N = 32$.

Table 3: MILP parameters

Parameter	Value	Unit	Parameter	Value	Unit
D^{con}	8	-	$\overline{x}$	4200	m
D^{vel}	8	-	$\underline{x}$	2700	m
d^{TIN}	100	m	$\overline{y}$	2600	m
c^{sensor}	2	Mbits s^{-1}	$\underline{y}$	500	m
$C^{\max}$	4	Mbits s^{-1}	$\overline{z}$	550	m
$t^{\text{separation}}$	5	s	$\underline{z}$	0	m
$\overline{V}_1$, $\overline{V}_2$	28	m s^{-1}	$\underline{V}_1, \underline{V}_2$	6	m s^{-1}
$\boldsymbol{r}^{\text{acc}}_1$, $\boldsymbol{r}^{\text{acc}}_2$	$(0.1, 0.1, 0.2)^\top$	-	γ^{finish}	100	-
d_x, d_y, d_z	50	m	d^{wp}	10	m
$\overline{h}$	100	Mbits			

Table 4: SPLAT! parameters

Parameter	Value
Earth Dielectric Constant (Relative permittivity)	15
Earth Conductivity (Siemens per meter)	0.005
Atmospheric Bending Constant (N-units)	301
Frequency in MHz (20 MHz to 20 GHz)	2400
Radio Climate (5 = Continental Temperate)	6
Polarization (0 = Horizontal, 1 = Vertical)	0
Fraction of situations (50% of locations)	0.5
Fraction of time (50% of the time)	0.5

4.2 Scenario 0: Two UAVs, without communication constraints

An interesting question is whether or not the solution obtained by iterating between the optimization problem and the simulator SPLAT! can justify the increased complexity. We have therefore solved the pure task assignment problem without taking into account any of the communication constraints, in order to be able to compare the performance. The cost to be minimized is given by

$$J = J^{\text{acc}} + J^{\text{finish}}\,, \tag{58}$$

where J^{acc} and J^{finish} are given in (7) and (39), subject to the constraints (1)-(4), (6), (8)-(38), (40)-(42). Consider therefore Figure 7 which shows a top view of the planned path, and Figure 8 which shows the calculated path losses. During the servicing of a task we have assumed that the UAV will send sensed information on the route with the least maximum path loss, either direct or by relaying through the other UAV. Even by chosing the best possible data transmission route, we can see that the path loss calculated using SPLAT! (solid black) violates the maximum path loss of 98 dB (dotted black) during communication (shaded areas). This shows that the introduced iterative interplay between optimization and simulation is necessary in order to enforce the design requirements with respect to path loss. By comparing Table 1, Figure 8 and Figure 7, it can be noted that UAV 1 communicates directly with the base station during the servicing of task $\mathcal{W}_2$. UAV 2 also communicates directly with the base station during the servicing of task $\mathcal{W}_3$. UAV 2 then services task $\mathcal{W}_4$, first by relaying information to base station via UAV 1, then by direct communication. Finally, task $\mathcal{W}_1$ is serviced by UAV 1, by relaying information to the base station. The UAVs fly in almost straight lines between waypoints. This is partly because there are no communication path loss constraints enforcing synchronization of their motion during relaying. Further penalizing of the acceleration constraints could have resulted in planned trajectories of more accuracy with respect to vehicle dynamics.

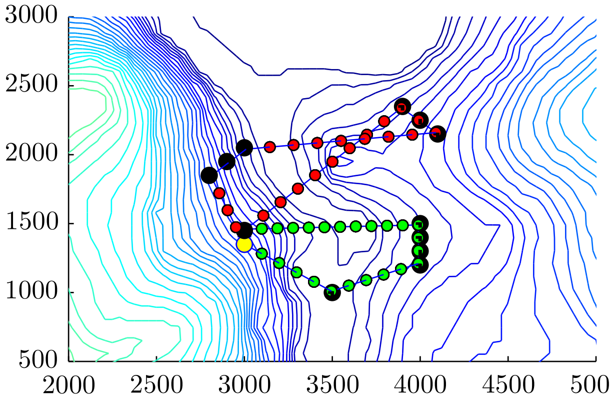


Figure 7: Scenario 0: Top view of the planned path. The black circles represent the waypoints, and the yellow circle is the initial location of the UAVs where also the base station is located. The red circles represent UAV 1, and the green circles represent UAV 2. The contours in the background represent the height of the terrain.

4.3 Scenario 1: Two UAVs, direct or relaying

In this scenario ferrying is not allowed, that is, $t^{\text{delay}} = 0$ seconds. The cost to be minimized is given by

$$J = J^{\text{acc}} + J^{\text{finish}}\,, \tag{59}$$

where J^{acc} and J^{finish} are given in (7) and (39), subject to the constraints (1)-(4), (6), (8)-(38), (40)-(57).

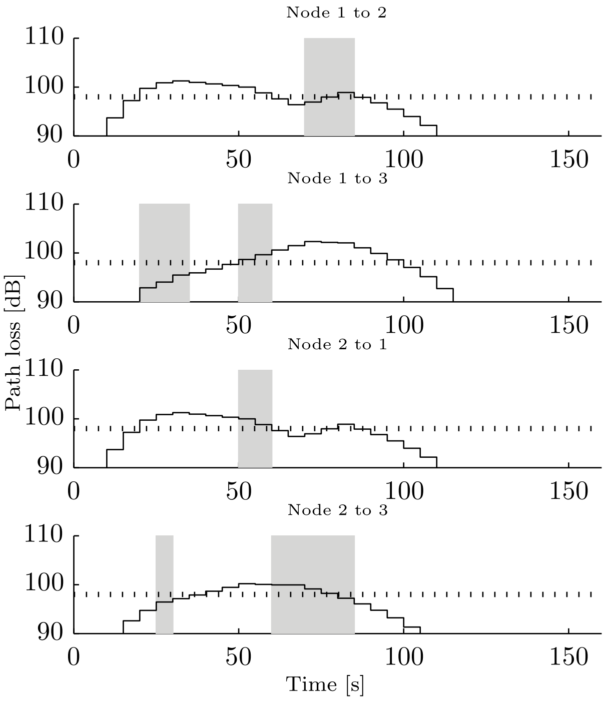
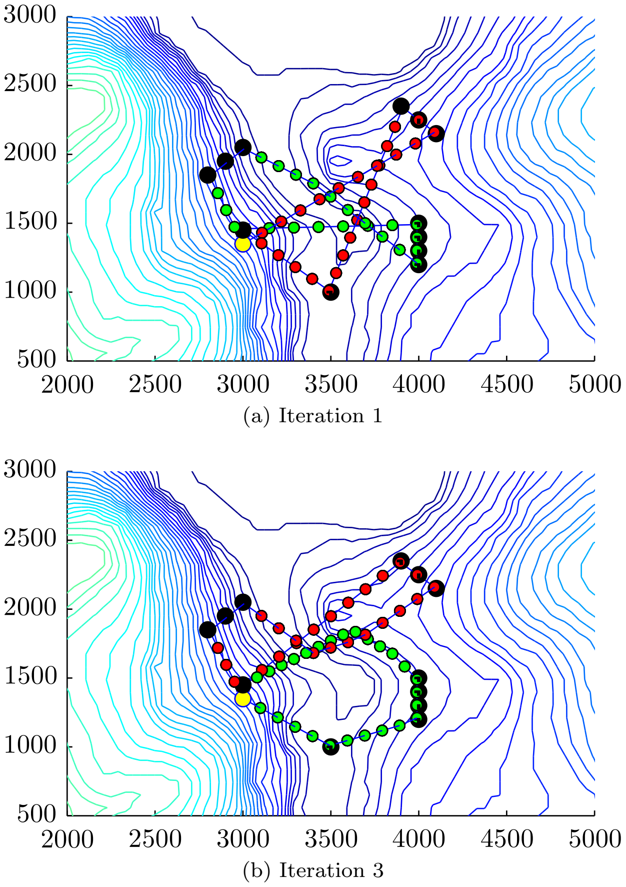


Figure 8: Scenario 0: The calculated path loss using SPLAT! (solid black), maximum allowed path loss for communication 98 dB (dashed black) and time steps when high bandwidth communication is required (shaded grey area). Notice that, the communication path loss requirement of 98 dB is violated at several instants.

Figure 9 shows a top view of the planned path for different iterations. The black circles represent the waypoints, and the yellow circle is the initial location of the UAVs, which East-North location coincides with the location of the base station. In particular, take notice that the trajectories planned at the first iteration are rather straight lines, while after the last iteration they are optimized in order to satisfy the communication path loss constraints during relaying. Figure 10 shows the calculated path loss using SPLAT! (solid black), maximum allowed path loss for communication 98 dB (dashed black), communication data rate (solid blue) and time steps when high bandwidth communication is required (shaded grey area). Comparing this figure with Table 1 and Figure 9b we see that UAV 1 and UAV 2 serve tasks $\mathcal{W}_2$ and $\mathcal{W}_3$, respectively, and transmit the sensor data directly to the base station. Then UAV 2 serves task $\mathcal{W}_4$ by relaying data by UAV 1 to the base station, before UAV 1 uses UAV2 as a relay node when servicing task $\mathcal{W}_1$. Figure 11 which shows the relative distance between the nodes at the final iteration. Also here the shaded areas represents time steps the nodes are communicating. The relative distance (solid lines) should preferably be below the MILP maximum relative distance (dashed lines) at these time steps. This may not always be the case, since we used an over-approximation of the Euclidean norm of the relative between the nodes in (57). Notice that the dashed lines are no longer constantly equal to 750, but have been reduced at some time steps during the iterations. Figure 12 shows the altitude of the vehicles at the final iteration.



(a) Iteration 1

(b) Iteration 3

Figure 9: Scenario 1: Top view of the planned path at (a) 1st iteration, and (b) 3rd and final iteration. See also caption text for Figure 7.

4.4 Scenario 2: Two UAVs, direct, relaying or ferrying

In this scenario ferrying is allowed, and we set $t^{\text{delay}} = 25$ seconds. The cost to be minimized is given by

$$J = J^{\text{acc}} + J^{\text{finish}}, \tag{60}$$

where J^{acc} and J^{finish} are given in (7) and (39), subject to the constraints (1)-(4), (6), (8)-(38), (40)-(57).

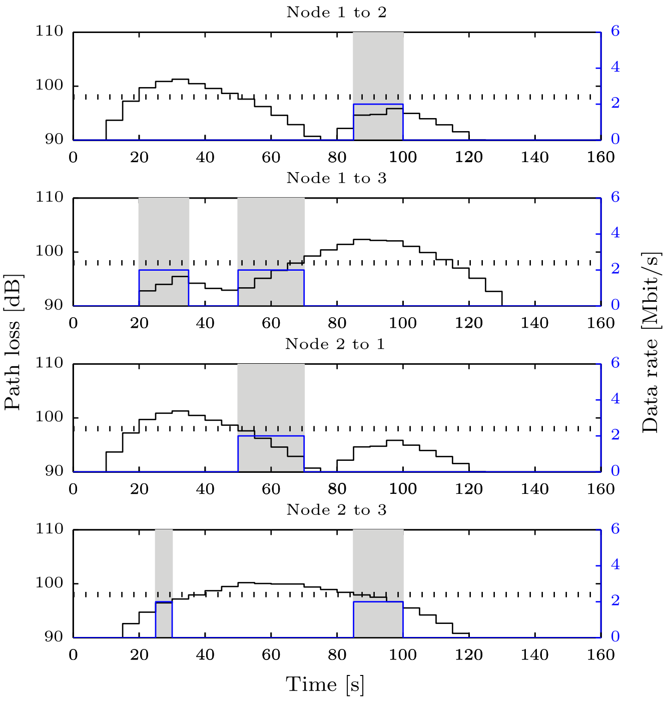


Figure 10: Scenario 1: The calculated path loss using SPLAT! (solid black), maximum allowed path loss for communication 98 dB (dashed black), communication data rate (solid blue) and time steps when high bandwidth communication is required (shaded grey area) to accomplish the task at the 3rd and final iteration. Notice that the communication path loss requirements of maximum 98 dB between all nodes are now satisfied.

Figures 13 to 16 correspond to Figures 9 to 11 in Scenario 1. In addition, the amount of data stored at each node is shown in Figure 17. For easier comparison between buffered and transmitted data, we have used the unit bit for stored data. Comparing Figure 9 with 13 it can be noticed that the UAVs can fly in slightly straighter paths between the tasks when delayed networking is allowed. This means that the total mission time is also shorter. The UAVs will still not fly in straight lines for two reasons: 1) We are minimizing the total time of the mission, and not the time spent by the individual UAVs. Hence, if one UAV will have to take a longer route than the other, the UAV with the shortest route will not be hurried to the final waypoint, but may rather serve as a relay vehicle. This may, or may not be a desirable feature depending on whether it is most important with short flight time or getting the sensor data transferred back to the base station as soon as possible. 2) For better comparison,

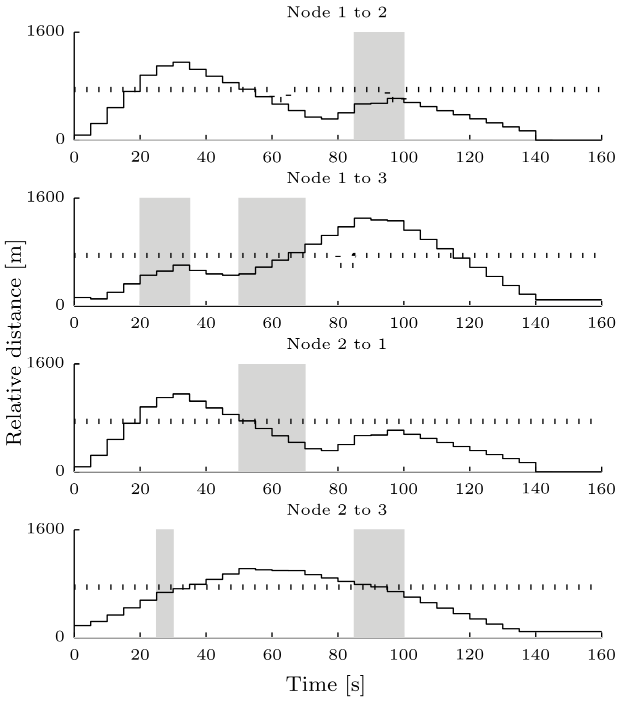


Figure 11: Scenario 1: The relative distance between nodes (solid black), and the MILP relative distance constraint for communication (dashed black) at the 3rd and final iteration. The shaded grey areas are time steps when high bandwidth communication is required, and therefore represents time instances when the solid line should preferably be at or below the dashed line. The reason why this is not the case here, is due to the fact that the solid line in the figure represents the actual relative distance, where as in the MILP problem we have used an over-approximation of the relative distance. The violation is irrelevant for the objective of our algorithm, which is to ensure that the communication path loss requirements are satisfied.

we are using the same initial conditions as in the previous simulation scenario. This means that the maximal distance between the UAVs is bounded by R^{con} which was set to 750 meters. As a result, seen from Figure 15 the UAVs will typically relay sensor information when possible, and delay the transmission only when necessary. Taking a close look at Figure 15 and comparing it with Table 1, Figure 13b and Figure 17 we see that first UAV 1 and UAV 2 serve tasks $\mathcal{W}_2$ and $\mathcal{W}_3$, respectively, and transmit the sensor data directly to the base station. Then UAV 2 starts collecting data at the waypoints of task $\mathcal{W}_4$, even if it is not possible to communicate with the base station neither directly nor by relaying data through UAV 1. Instead it will have to store the data onboard until it gets into communication range of UAV 1. It will then start relaying both the sensed and the stored data through UAV 1, hence the increased data transmission rate seen in Figure 15. Finally, task $\mathcal{W}_1$ is then served by UAV 1, using UAV 2 as a relay node. From Figure 15 alone, there is no apparent reason why the transmission should be interrupted since the communication path loss criteria is satisfied between all nodes. Taking a close look at Figure 11 reveals that the relative distance constraint between UAV 1 and UAV 2 has been tightened at the time instant, and the transmission will therefore need to be interrupted. The sensed data is temporarily stored on UAV 1, as seen from Figure 17.

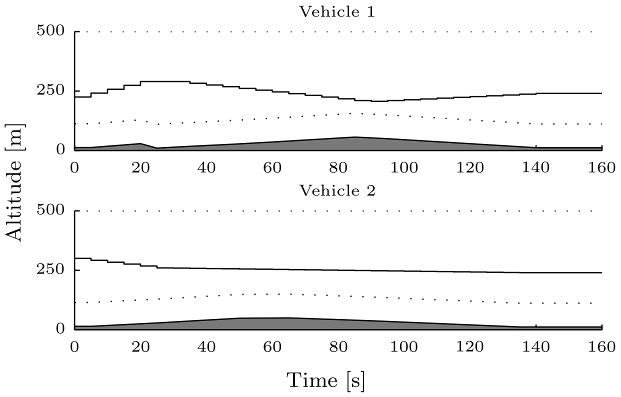
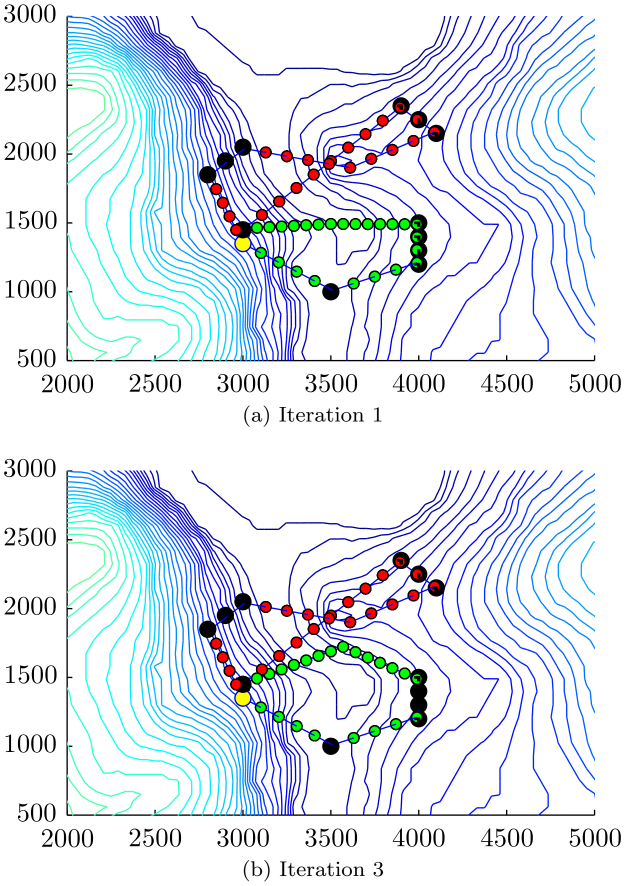


Figure 12: Scenario 1: Altitude of the vehicles at the final iteration. The shaded region is the terrain height, and the dashed lines are the upper and lower bounds on the altitude of the vehicles. The actual altitude of the vehicles is depicted with the continuous line during the mission, and even after they reach the final waypoint. When the final waypoint is reached they are expected to be landed manually by a pilot.

4.5 Discussion

As pointed out in the introduction the intention of this paper is to provide a coarse motion- and communication-plan for a complex surveillance mission. In addition, an online re-planner might be used to account for model uncertainties and effects that become apparent in real-time. We will now discuss some of the uncertainties that should be captured by such a re-planner. Here, a simple point mass model has been used for the UAV, whereas the maneuverability of the physical UAVs will highly depend on its design parameters such as weight, engine power and aerody-



(a) Iteration 1

(b) Iteration 3

Figure 13: Scenario 2: Top view of the planned path at (a) 1st iteration, and (b) 3rd and final iteration. See also caption text for Figure 7.

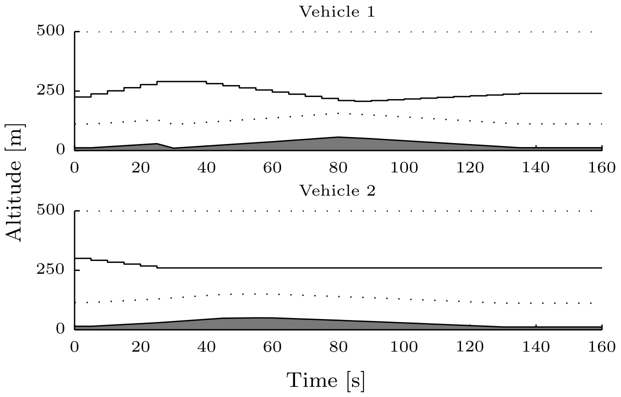


Figure 14: Scenario 2: Altitude of the vehicles at the final iteration. The shaded region is the terrain height, and the dashed lines are the upper and lower bounds on the altitude of the vehicles. The actual altitude of the vehicles is depicted with the solid line. When arriving at the final waypoint, the UAV will be brought down by remote control.

namics including lift- and drag forces. We have also ignored some of the environmental effects, and in particularly wind will affect the lift and drag of the UAVs. In addition the communication properties can be heavily attenuated by rain at certain frequencies, and although SPLAT! is a relatively advanced radio propagation model there will be local effects that are not captured. The sampling based nature of the problem, and in particular when analyzing the radio propagation path loss in the simulator, means that the results may be quite different between samples. Furthermore, the Doppler effect is not taken into account, which may have some effect for high velocity flights. A general recommendation would be to use conservative constraints in the offline path planner, in particular for the vehicle speed and the radio propagation constraints. This means that there would be some robustness to changes in radio propagation path loss between samples, and that vehicle speed can be decreased or increased in order to compensate for wind-effects and lack of ability to perform planned maneuvers. Finally, a minor drawback of the method was illustrated in Section 4.4 where the relative distance constraints of the MILP problem may cause interruption in the communication even when the actual communication path loss is below the design criterion. In some cases this could be remedied by only tightening the relative distance constraint by a little amount at each iteration. This will not work in general as the order in which the tasks are served may change from iteration to iteration and therefore the necessity of transmission at a certain time instant will also change.

The motion- and communication-planning algorithm described in this paper is only intended for offline use, and computational complexity may therefore be of less importance than for real-time algorithms. It is however important to realize that low computation time is desirable also for offline methods. An example is the inspections of critical infrastructure with a known failure, where it is important to locate the failure as soon as possible for it to be dealt with. Reducing the computational complexity has not been emphasized in this work, but in this section we will discuss the many means available for decreasing the computation time. For instance, the code could be made more efficient by estimating bounds on the optimization variables in advance such that tighter constraints could be provided to the solver; bounds on the costs of the optimization problem could be estimated and added as constraints; the required accuracy of the optimal solution could be reduced by adjusting the solver parameters (it should be related to the accuracy of the model pro-

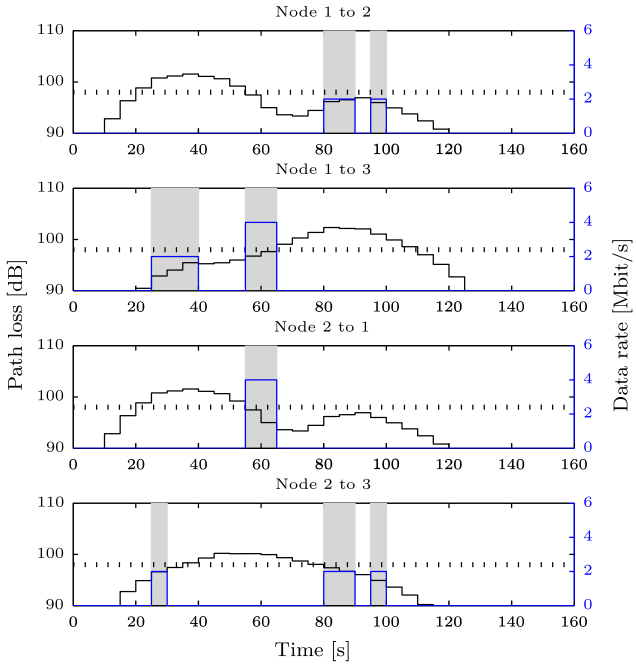


Figure 15: Scenario 2: The calculated path loss using SPLAT! (solid blue), maximum allowed path loss for communication 98 dB (dashed blue), communication data rate (solid blue) and time steps when high bandwidth communication is required (shaded grey area) to accomplish the task at the 3rd and final iteration.

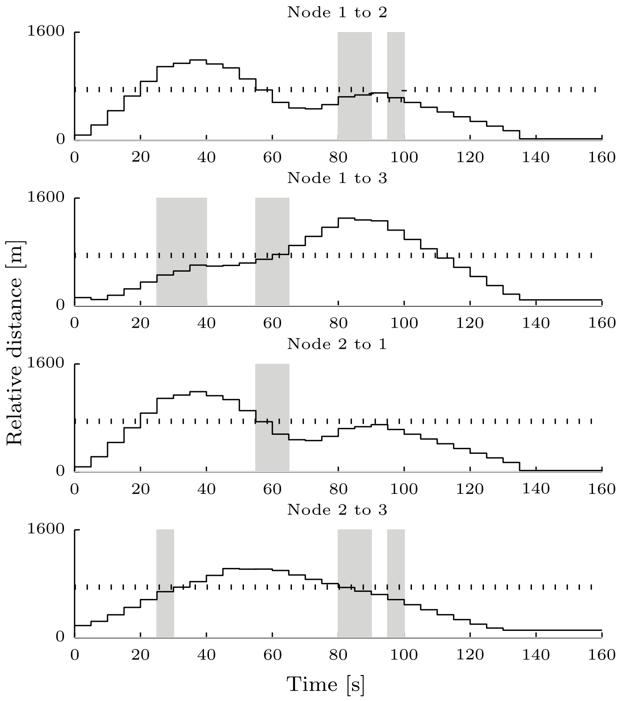


Figure 16: Scenario 2: The relative distance between nodes (solid black), and the MILP relative distance constraint for communication (dashed black) at the 3rd and final iteration. See also caption text for Figure 11.

vided); a non-optimal solution to the problem could be provided at start-up, Vitus et al. (2008); Kuwata and How (2011); the computation could be (decomposed and) run in parallel over several computers, Gunnerud and Foss (2010); Kuwata and How (2011); the computational complexity could be reduced by reducing the number of binary variables, Vielma et al. (2010); Prodan et al. (2012). In general it takes longer to find a feasible solution when we allow for ferrying due to the additional variables we have introduced. However, ferrying means that it is possible to find a feasible solution with lower maximum allowed path loss and/or for tasks placed further away from the base station. Also, it does not require coordination between vehicles, and can therefore be considered a more robust alternative to relaying.

One of the benefits of using MILP is that it gives a global optimal solution, and is therefore suitable for the benchmarking of other, possibly sub-optimal methods. One such method is Particle Swarm Optimization (PSO), Kennedy and Eberhart (1995), which has successfully been applied for optimal path planning for UAVs in *e.g* Sujit and Beard (2009), Ho et al. (2013) and Ho et al. (2015). However, it turned out that PSO is not very well suited for an implementation of an optimization problem similar to that of Section 3. By just requiring the UAVs to visit all the waypoints (and disregard any communication constraints), the resulting paths using PSO were comparable to the one achieved with MILP, but with computation time exceeding the computation time of MILP with communication constraints. The dimension of the search space was in this case $n_p \times N$. By including the communication constraints in the PSO the dimension of the search space would increase to $n_p^2 \times (n_p+1)^2 \times N^3$ which leads to very high computational complexity. The key problems of using PSO for this scenario were identified as: a high-dimensional search space; PSO lacks the ability to implement hard inequality constraints, which resulted in a large number of cost function terms that were difficult to weigh in case of competing objectives (*e.g* minimizing the mission time, while constraining the velocity of the UAVs); PSO lacks the ability to implement hard equality constraints, which resulted in a problem of conserving the data flow in and out of network nodes.

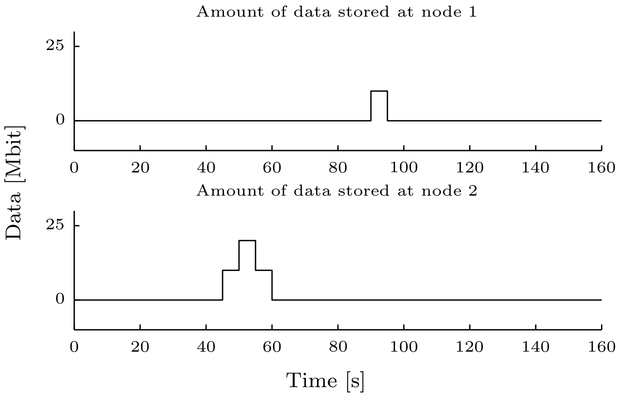


Figure 17: Scenario 2: The amount of data stored at the nodes at the 3rd and final iteration. Notice that the increments are typically of 10 Mbits, which is equivalent to $c^{\text{sensor}}\Delta t$, i.e. the rate at which data is collected multiplied with the sample time.

5 Conclusion

In this paper we have addressed the problem of task assignment and path- and communication-planning for multiple UAVs. In the MILP formulation a general communication topology is allowed such that sensor data can be transferred efficiently back to the base station by direct transmission, ferrying or relaying. Tolerating delays introduces flexibility into the motion- and communication-planning problem, and means that the range between tasks and base station can be extended, and/or that the maximum allowed path loss can be reduced. We have improved the accuracy of the communication properties by using a path loss simulator to analyze the paths generated from the optimization problem.

Acknowledgment

This work was supported by the Research Council of Norway through the Center of Excellence on Autonomous Marine Operations and Systems.

References

Alighanbari, M., Kuwata, Y., and How, J. Coordination and control of multiple UAVs with timing constraints and loitering. In *Proc. of the American Control Conference*, volume 6. pages 5311 – 5316 vol.6, 2003. doi:10.1109/ACC.2003.1242572.

Beard, R. W. and McLain, T. W. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

Bemporad, A. and Morari, M. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 1999. 35:407–427. doi:10.1016/S0005-1098(98)00178-2.

Chaudhry, A., Misovec, K., and D’Andrea, R. Low observability path planning for an unmanned air vehicle using mixed integer linear programming. In *Proc. of the 43rd IEEE Conference on Decision and Control*. pages 3823–3829, 2004. doi:10.1109/CDC.2004.1429334.

Dixon, C. and Frew, E. Electronic leashing of an unmanned aircraft to a radio source. In *Proc. of the 44th IEEE Conference on Decision and Control and European Control Conference*. pages 3560–3565, 2005. doi:10.1109/CDC.2005.1582714.

Dixon, C. and Frew, E. W. Decentralized extremum-seeking control of nonholonomic vehicles to form a communication chain. In *Advances in Cooperative Control and Optimization*, pages 311–322. Springer Berlin / Heidelberg, 2007. doi:10.1007/978-3-540-74356-$9_1$9.

Earl, M. G. and D’Andrea, R. Modeling and control of a multi-agent system using mixed integer linear programming. In *Proc. of the IEEE Conference on Decision and Control*. pages 107 – 111, 2002. doi:10.1109/CDC.2002.1184476.

Frew, E. W. and Brown, T. X. Networking issues for small unmanned aircraft systems. *Journal of Intelligent and Robotic Systems*, 2008. 54:21–37. doi:10.1007/978-1-4020-9137-7_3.

Frew, E. W., Brown, T. X., Dixon, C., and Henkel, D. Establishment and maintenance of a delay tolerant network through decentralized mobility control. In *Proc. of the IEEE International Conference on Networking, Sensing and Control*. pages 584–589, 2006. doi:10.1109/ICNSC.2006.1673211.

Grancharova, A., Grøtli, E. I., Ho, D.-T., and Johansen, T. A. UAVs trajectory planning by distributed MPC under radio communication path loss constraints. *Journal of Intelligent and Robotic Systems*, 2014. doi:10.1007/s10846-014-0090-1.

Grancharova, A., Grøtli, E. I., and Johansen, T. A. Distributed path planning for a UAV communication chain by dual decomposition. In *Proc. of the IFAC Workshop on Multivehicle Systems*. pages 43–48, 2012. doi:10.3182/20121003-3-SF-4024.00001.

Grøtli, E. I. and Johansen, T. A. Path- and data transmission planning for cooperating UAVs in delay tolerant network. In *Proc. of the 3rd International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles: Architectures, Protocols and Applications*. pages 1568–1573, 2012a. doi:10.1109/GLOCOMW.2012.6477819.

Grøtli, E. I. and Johansen, T. A. Path planning for UAVs under communication constraints using SPLAT! and MILP. *Journal of Intelligent and Robotic Systems*, 2012b. 65:265–282. doi:10.1007/s10846-011-9619-8.

Grøtli, E. I. and Johansen, T. A. Task assignment for cooperating UAVs under radio propagation path loss constraints. In *Proc. of the American Control Conference*. pages 3278–3283, 2012c. doi:10.1109/ACC.2012.6315041.

Gunnerud, V. and Foss, B. Oil production optimization - a piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering*, 2010. 34:1803–1812. doi:10.1016/j.compchemeng.2009.10.019.

Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 1968. 4(2):100–107. doi:10.1109/TSSC.1968.300136.

Henkel, D. and Brown, T. X. On controlled node mobility in delay-tolerant networks of unmanned aerial vehicles. In *Proc. of International Symposium on Advanced Radio Technologies*. 2006.

Ho, D.-T., Grøtli, E. I., Sujit, P. B., Johansen, T. A., and de Sousa, J. B. Performance evaluation of cooperative relay and particle swarm optimization path planning for uav and wireless sensor network. In *Proc. of the 4th International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles: Architectures, Protocols and Applications*. pages 1403 – 1408, 2013. doi:10.1109/GLOCOMW.2013.6825191.

Ho, D.-T., Grøtli, E. I., Sujit, P. B., Johansen, T. A., and de Sousa, J. B. Optimization of wireless sensor network and UAV data acquisition. *Journal of Intelligent and Robotic Systems*, 2015. doi:10.1007/s10846-015-0175-5.

Jain, S., Fall, K., and Patra, R. Routing in a delay tolerant network. In *Proc. of the conference on Applications, technologies, architectures, and protocols for computer communications*. pages 145–158, 2004.

Kennedy, J. and Eberhart, R. Particle swarm optimization. In *Proc. of the IEEE International Conference on Neural Networks*. pages 1942–1948, 1995. doi:10.1109/ICNN.1995.488968.

Kim, Y., Gu, D.-W., and Postlethwaite, I. Real-time optimal mission scheduling and flight path selection. *IEEE Transactions on Automatic Control*, 2007. 52(6):1119 –1123. doi:10.1109/TAC.2007.899048.

Kingston, D. B. and Schumacher, C. J. Time-dependent cooperative assignment. In *Proc. of the American Control Conference*. pages 4084–4089, 2005. doi:10.1109/ACC.2005.1470617.

Kuwata, Y. and How, J. P. Cooperative distributed robust trajectory optimization using receding horizon MILP. *IEEE Transaction on Control Systems Technology*, 2011. 19(2):423–431. doi:10.1109/TCST.2010.2045501.

Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., and How, J. P. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 2009. 17(5):1105–1118. doi:10.1109/TCST.2008.2012116.

LaValle, S. M. *Planning Algorithms*. Cambridge University Press, 2006.

LaValle, S. M. and Kuffner, J. J. J. Randomized kinodynamic planning. *International Journal of Robotics Research*, 2001. 20(378):378–400. doi:10.1177/02783640122067453.

Likachev, M., Ferguson, D., Gordon, G., Stentz, A., and Thrun, S. Anytime dynamic A*: An anytime, replanning algorithm. In *Proc. of the 15th International Conference on Automated Planning and Scheduling*. pages 262–271, 2005.

Löfberg, J. Modeling and solving uncertain optimization problems in YALMIP. In *Proc. of IFAC World Congress*. pages 1337–1341, 2008. doi:10.3182/20080706-5-KR-1001.00229.

Longley, A. G. and Rice, P. L. Prediction of tropospheric radio transmission loss over irregular terrain: A computer method. Technical report, U. S. Goverment, 1968.

Ma, C. S. and Miller, R. H. MILP optimal path planning for real-time applications. In *Proc. of the American Control Conference*. pages 4945–4950, 2006. doi:10.1109/ACC.2006.1657504.

Maglicane, J. SPLAT! An RF signal propagation, loss and terrain analysis tool. 2010 (Accessed August 18, 2010). URL http://www.qsl.net/kd2bd/splat.html.

Prodan, I., Stoican, F., Olaru, S., and Niculescu, S.-I. Enhancements on the hyperplane arrangements in mixed-integer programming techniques. *Journal of Optimization Theory and Applications*, 2012. 154(2):549–572. doi:10.1007/s10957-012-0022-9.

Reinl, C. and von Stryk, O. Optimal control of multi-vehicle-systems under communication constraints using mixed-integer linear programming. In *Proc. of the International Conference on Robot Communication and Coordination*. 2007.

Richards, A. and How, J. Mixed-integer programming for control. In *Proc. of the American Control Conference*. pages 2676–2683, 2005. doi:10.1109/ACC.2005.1470372.

Richards, A. and How, J. P. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proc. of the American Control Conference*. pages 1936–1941, 2002. doi:10.1109/ACC.2002.1023918.

Saska, M., Macas, M., Preucil, L., and Lhotska, L. Robot path planning using particle swarm optimization of Ferguson splines. In *Proc. of the IEEE Conference on Emerging Technologies and Factory Automation*. pages 833 – 839, 2006. doi:10.1109/ETFA.2006.355416.

Schouwenaars, T., De Moor, B., Feron, E., and How, J. Mixed integer programming for multi-vehicle path planning. In *Proc. of the European Control Conference*. 2001.

Schouwenaars, T., Stubbs, A., Paduano, J., and Feron, E. Multivehicle path planning for nonline-of-sight communication. *Journal of Field Robotics*, 2006. 23:269–290. doi:10.1002/rob.20119.

Shengxiang, Z. and Hailong, P. Real-time optimal trajectory planning with terrain avoidance using MILP. In *Proc. of the International Symposium on Systems and Control in Aerospace and Astronautics*. pages 1 – 5, 2008. doi:10.1109/ISSCAA.2008.4776318.

Spanos, D. P. and Murray, R. M. Motion planning with wireless network constraints. In *Proc. of the American Control Conference*. pages 87 – 92, 2005. doi:10.1109/ACC.2005.1469913.

Stentz, A. Optimal and efficient path planning for partially-known environments. In *Proc. of the International Conference on Robotics and Automation*. pages 3310–3317, 1994. doi:10.1109/ROBOT.1994.351061.

Sujit, P. B. and Beard, R. Multiple UAV path planning using anytime algorithms. In *Proc. of the American Control Conference*. pages 2978–2982, 2009. doi:10.1109/ACC.2009.5160222.

Tanenbaum, A. S. *Computer Networks*. Prentice Hall PTR, 4 edition, 2003.

Thunberg, J., Anisi, D. A., and Ögren, P. A comparative study of task assignment and path planning methods for multi-UGV missions. In *Proc. of the International Conference on Cooperative Control and Optimization*. 2008.

Tsourdos, A., White, B., and Shanmugavel, M. *Cooperative Path Planning of Unmanned Aerial Vehicles*. Wiley, 2010.

Vielma, J. P., Ahmed, S., and Nemhauser, G. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research*, 2010. 58:303–315. doi:10.1287/opre.1090.0721.

Vitus, M. P., Pradeep, V., Hoffmann, G. M., Waslander, S. L., and Tomlin, C. J. Tunnel MILP: Path planning with sequential convex polytopes. In *Proc of the AIAA Guidance, Navigation, and Control Conference*. 2008.

Williams, H. P. *Model Building in Mathematical Programming*. John Wiley and Sons, 4 edition, 1999. ISBN: 978-0471997887.

Wright, A. SPLAT! 1.3.1 for Windows. 2011 (Accessed June 29, 2011). URL http://www.ve3ncq.ca/wordpress/?page_id=62.

Zhao, W. and Ammar, M. Message ferrying: Proactive routing in highly-partioned wireless ad hoc networks. In *Proc. of the International Workshop on Future Trends of Distributed Computing Systems*. pages 308–314, 2003. doi:10.1109/FTDCS.2003.1204352.