Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 35 (2018) 110–119

Transportation Research Procedia

www.elsevier.com/locate/procedia

International Conference on Air Transport – INAIR 2018

# A Mission Coordinator Approach for a Fleet of UAVs in Urban Scenarios

Carlos Perez-Montenegro[a], Matteo Scanavino[b], Nicoletta Bloise[b], Elisa Capello[b,c], Giorgio Guglieri[b], Alessandro Rizzo[a]*

[a]*Politecnico di Torino, Department of Electronics and Telecommunications (DET) , Corso Duca degli Abruzzi 24, Turin 10129, Italy*
[b]*Politecnico di Torino, Department of Mechanical and Aerospace Engineering(DIMEAS) , Corso Duca degli Abruzzi 24, Turin 10129, Italy*
[c]*National Research Council of Italy - Institute of Electronics, Computer and Telecommunication Engineering (CNR-IEIIT) , Corso Duca degli Abruzzi 24, Turin 10129, Italy*

**Abstract**

The use of Unmanned Aerial Vehicles (UAVs) is now common, but although they have been for various applications, there are still a lot of challenges that need to be overcome. One key issue is related to standardizing the use of these vehicles in urban environments and guaranteeing a minimum risk level for the population. To rise to these challenges, autonomous strategies that optimize and coordinate vehicles in cooperative missions and avoid human operators should be developed. The novelty of this paper is the development of an autonomous urban mission coordinator, which is responsible for the high-level logistics of a fleet of heterogeneous vehicles. A multi-variable weighted algorithm based on a tree optimization method is also proposed.



*Keywords:* Smart City; UAVs, Urban Mission, Mission Coordinator

## 1. Introduction

The concept of the smart city is a topical issue that covers many fields of knowledge and aims to improve the quality of life of people as shown in Deakin (2014), Song (2017) and Kar (2017). Multiple technological challenges to obtain information and running applications have emerged. To solve some of these challenges, the use of autonomous vehicles for smart applications have been explored, including opportunities related to the use of UAVs in smart cities (as in Mohammed et al. (2014)). While presenting advantages for monitoring, traffic and crowd management, several issues related to safety, privacy and ethical uses are obstacles their widespread applications in smart cities. Most autonomous vehicles use radio frequency transmission for control purposes, to increase vehicle


10.1016/j.trpro.2018.12.018

operational range and allow execution of missions in wider areas. Mahmoud et al. (2015) have proposed a Cloud Computing integration, where the aerial vehicles become part of a Cloud infrastructure, accessed through an Internet connection. UAVs are suitable platforms for launching many Internet of Things (IoT) applications. For example, in the work of Fotouhi et al. (2017) the controls of a commercial multi-copter performed by an Android program was investigated, to exploit UAV agility, maneuverability and speed providing IoT services.

Many companies have been working on parcel delivery using autonomous vehicles, for example Amazon "Prime Air" (2018) and in DHL "Parcelcopter" (2018). Even though experimental tests have been performed, practical applications in urban scenarios are not allowed, due to risk-based limitations of current regulations. More recently, the UAVs integration into national airspace has become an emerging research field as demonstrate the initiative promoted by the US Department of Transportation (2018), in which ten cities were selected to investigate the effects of this integration. Several techniques for UAV position and attitude control have been developed and tested, a complete review of these technologies are presented in Farid et al (2017). Nevertheless, most of the urban missions still requires several human operators to guarantee low risk level for citizens.

As reported by Yi Wei et al. (2013), the single UAV use will be replaced by swarms and automatic coordination, for this reason in this paper an efficient planning approach of swarm monitoring and dynamic mission is considered. According to Navarro et al. (2012), swarm robotics allows the coordination of large groups of relatively simple robots to perform tasks that are beyond the capabilities of the individual (System of Systems). A multiple UAV scenario has been taken into account because it offers various advantages compared to the single UAV: (i) ability to cover different missions in terms of time and covered distance and (ii) payload characteristics and flexibility. Nowadays, UAVs require many operators to complete a complex or even a simple mission. In our approach, a mission coordinator, which operates autonomously, will help to perform a predefined mission, with a limited number of operators. Moreover, a centralized control of the mission is also proposed. Examples of UAV collaborations are architecture structure building (e.g. wall or rope bridges), light shows or precision farming, such as SAGA project. Varela et al. (2011) proposed a swarm intelligence-based approach for environmental monitoring using UAV teams. The vehicles coordinate themselves for pollution monitoring and source detection, when undesired environmental conditions arise. In Varela et al. (2011) a distributed approach is described, in which different vehicles operate both autonomously and in collaboration with neighbor vehicles.

Regulation constraints are one of the reasons for which UAV fleets are not fully investigated in urban scenarios. New approaches are needed to fly vehicle swarms safely, perform collaborative missions and optimize time, consumption or other relevant factors. The novelty of the proposed research is the design of an autonomous high-level mission coordinator, which optimizes paths to fulfill the predefined cooperative mission. The proposed methodology is based on the construction of a tree with possible connections between docking stations and mission objectives. Each edge of this tree represents an optimized trajectory of a single vehicle and each branch of the tree includes a set of trajectories to accomplish the mission. The main contribution of this work is the definition of a mission coordinator that can integrate low level control and path planning for a single vehicle, to execute a cooperative mission. Some optimization parameters, such as the covered distance and the battery consumption are included to perform an autonomous mission, thus reducing the execution time and the number of operators.

Furthermore, the strategy includes distributed docking stations, with known positions, to manage a fleet of UAVs. Therefore, several feasible and nominal trajectories are pre-computed by the path planner, considering expected obstacles. A dynamic path planner deals with dynamic obstacles. One advantage of this strategy is the achievement of an optimal solution in which the execution time is reduced also in presence of system failures.

The remainder of this paper is organized as follows: Mission definition and classification with a fleet of UAVs are provided in Section 2. Definition of the main scenarios and considerations of the UAV configurations are described in Sections 2.1 and 2.2. Section 3 explains the proposed architecture, highlighting the role of the mission coordinator. The mission coordinator and the proposed method are presented in Section 4. Preliminary simulation results are presented in Section 5. Final discussions and conclusions are summarized in Section 6.

## 2. Mission classification

Several missions with UAVs have been studied in the last decades. In this work, civil and urban applications are covered in more detail, considering a multi-modal system with a fleet of Unmanned and Manned vehicles. The

proposed strategy includes autonomous UAV and manned ground vehicles, the latter working as docking and charging stations for UAVs, while the former performing monitoring or delivery missions. UAV applications in urban environment are wide and four missions can be defined: monitoring, inspection, delivery and intervention.

**Monitoring missions** consist in continuous scanning of a defined area looking for live changes. They are particularly suitable in case of environmental management, such as weather or pollution monitoring, but also for industry purposes, like tracking building progress on construction sites, traffic management or shipping lanes monitoring in port areas.

**Inspection missions** consist in assessment of area to evaluate its current state. UAVs are promising for inspection purposes due to the reduced risk in hazardous environments, cost reduction and inspection time speed up. Some examples are inspections of bridges for cracks or corrosion, power lines and oil/gas pipelines as well as wind turbine.

**Delivery missions** mean collect, transport and deliver goods from a starting to a final point. Employing UAVs should decrease traffic on city roads, reduce air pollution as well as last-mile cost. Delivery mission could be organized differently, such as hub-to-hub, hub to different location and vice-versa where a pick-up point is not a docking station. Consumer deliveries, from warehouse to private homes, are typical uses cases; however, emergency supply (first aid kits or medicines) as well as blood sample and transfusions deliveries are emerging solutions in urban scenarios.

**Intervention missions** consist of interaction with objects or people to improve or support their state. Examples of such missions are obstacle removal from areas where they may cause disruption, fix defect or damage to physical infrastructure and improve the capacity, speed and quality of existing services. Interaction with human activities requires a deep understanding of the surrounding environment.

To define a general smart application problem, the aforementioned UAV applications can be grouped in two categories: (i) coverage missions, in which the vehicles should be able to scan a confined area or take multiple snapshots inside a limited space, and (ii) delivery mission, in which the vehicles should reach a target destination to perform specific actions.

Figure 1 schematically shows the analyzed UAV applications. A Cloud-based approach is presented to fulfill the control and vehicle coordination constraints. In Figure 1, docking stations are represented with the battery symbol; the interconnection between ground and aerial vehicles with a server is represented with the Cloud symbol. Coverage and delivery applications are responsible for complementary mission constraints.

To design and plan autonomous missions, a set of $n$ aerial vehicles and $m$ docking stations (i.e. ground vehicles) in known positions are considered. The position of the aerial $i = 1,\ldots,n$ in the reference frame is represented by $\boldsymbol{x}_i(t)$ whereas the position of the docking station $Dk$ is $\boldsymbol{x}_{dk.i}(t)$, with $j = 1,\ldots,m$.

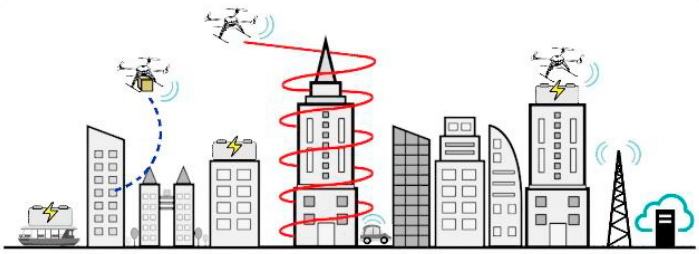

Figure 1 Reference Scenario

### *2.1. Delivery missions*

This type of application is aimed to collect, transport and deliver from a starting point to a final destination. A general theory of the delivery is deeply explained in Otto (2018). Air traffic management and vehicles coordination is mandatory, because dynamic flight paths have to be managed simultaneously. In our case, the focus is exactly on the simultaneous management of different flight paths. The initial position of all vehicles is known and the UAV should reach the pick-up point to collect the package be transported to the target point, as schematically shown in Figure 2.a. The final maneuver time can be optimized or fixed by the operator. If the power level of the vehicle is

too low to accomplish the task, the UAV should reach the closest docking station and a new charged vehicle will take the package and continue the mission.

### *2.2. Coverage missions*

This type of application is aimed to inspect and monitor of areas and structures. A general theory of the coverage is deeply explained in Galceran (2013). In an urban context, coverage includes applications such as traffic and crowd management, environmental monitoring and urban security. Vehicles involved in such missions should be able to demonstrate precise and controlled flight capabilities regardless environmental conditions, while to scan large networks will likely required a good coordination of a large number of heterogeneous vehicles. When performing inspection missions, UAVs are expected to flight in close proximity to gather data on structures. The risk of collision is high and safe guards and emergency protocols are necessary to accomplish the defined mission. Figure 2.b shows a coverage scenario. Once mission requirements are defined by the operator, the closest UAV takes off and executes the mission. In the event of low battery level, a safety landing is imposed and another UAV will complete the task. The number of UAVs involved in the mission is also related to the final maneuver time selected by the operator.

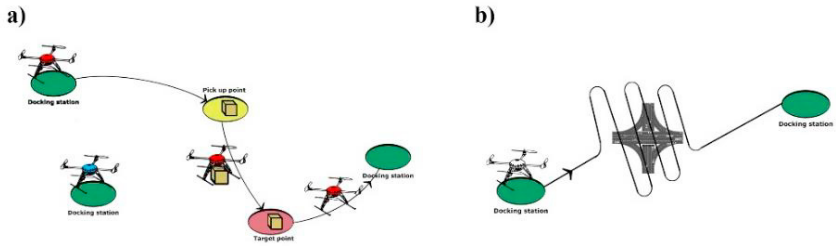


Figure 2 a) Delivery mission, single vehicle b) Coverage scenario

### *2.3. Vehicle Configurations*

Within the framework of this project, two UAV configurations are considered:

- **Fixed wing**: This type of vehicle has advantages such as high speed and operating altitude, as well as good endurance and range performance. However, it requires space and time for the take-off and landing maneuvers. Furthermore, the minimum turn radius should be taken into account to obtain feasible optimal path, bearing in mind an operational constraint.
- **Rotary wing**: This type of vehicle has advantages such as its flexibility and hover capabilities; take-off and landing maneuvers can be performed with little space. The drawbacks are related to endurance and range.

Vehicle kinematics and dynamics are represented by the following equation

$$\boldsymbol{X}(k) = f\big(\boldsymbol{X}(k-1), \boldsymbol{u}(k)\big), \#(1)$$

where $\boldsymbol{u}(k)$ is the command vector and $\boldsymbol{X}(k) = [x(k) \quad v(k) \quad \theta(k) \quad \omega(k)]^T$ is the state vector related to position, speed, attitude and angular rate. More details on the mathematical model for fixed and rotary wings can be found in Etkin (2012). The dynamics of the vehicle therefore defines a set of constraints for each vehicle, such as minimum turn radius and maximum speed. These constraints allow smooth and flyable trajectories.

## 3. Proposed architecture

The Cloud-based framework proposed in this paper is able to manage high computational tasks, such as mission coordination, data processing, map generation, path planning and network control. Few processes are executed onboard, i.e. data acquisition, command execution and onboard control in emergency conditions. The proposed architecture is presented in Figure 3 and it is composed by three main blocks: (i) unmanned aerial vehicle, (ii) ground vehicles and (iii) the Cloud. Note that blue line represents feedback signals for control purposes, green

arrows are sensor data, the red line represents Cloud control commands and finally the dashed black lines are internal block signals.

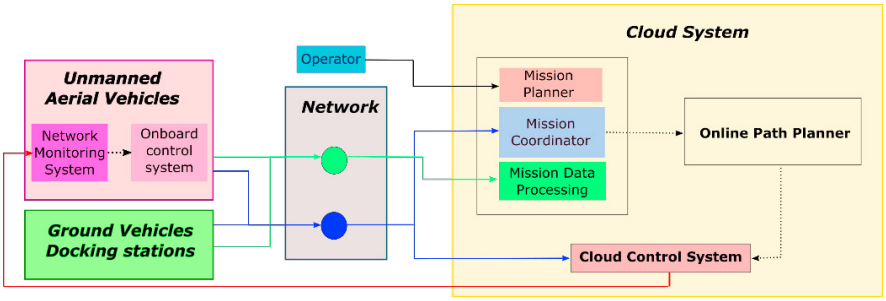


Figure 3 Proposed architecture

The main task of UAVs is to acquire information from onboard sensors and transmits the data to the Cloud to fulfill mission requirements. UAVs receive a set of waypoints and control commands as input from the Cloud. The autopilot and online path planner with the onboard sensors perform collision avoidance and emergency tasks, when the link with the Cloud is poor or missing. Docking stations provide support to UAVs for battery recharge issue. As previously described, positions and states of the docking stations are known and defined offline by operators.

The Cloud performs all the data processing related to the application and requested by the operator and executes mission coordinator and path planning algorithms. Cloud computing allows the execution of high computational algorithms such as optimal control techniques and store data. A communication protocol, as broadband cellular network technology, is needed for the communication with autonomous and ground vehicles. For example, 5G mobile network provides high data rate, low latency communication and large bandwidth. The communication protocol must be able to transmit the state of the vehicles to the server and commands from the server to the vehicles in real-time.

The block related to the smart application consists of three subsystems: mission planner, mission coordinator and mission data processing. The first subsystem defines mission requirements depending on the operator requests. As an example, in a delivery mission the user can set the starting and target points, the payload characteristics to define the suitable and closest vehicles to be involved in the mission. The mission coordinator executes fleet management and vehicle collaboration purposes, while the mission data processing collects and manages the data from UAV sensors. The mission coordinator performs a risk assessment based on available databases and provides dynamic risk maps to the online path planner. The risk maps are generated applying a probabilistic approach and combining several layers related to the population density, sheltering factor, network coverage and obstacle zone databases according to Valavanis (2015).

The path planner block defines the trajectories of autonomous vehicles in a dynamically, based on the maps provided by the previous block. Given the current position of UAVs and the desired target, the path planning seeks for the optimal waypoint sequence.

The control system block includes optimal control strategies to minimize actuator power consumption and to optimize task requirements subjected to urban constraints. As previously mentioned, in this paper the control system is distributed between the UAV and the Cloud, through the onboard autopilot and the On-Cloud control system. The control of UAVs is usually performed by onboard autopilots. Robust and adaptive controllers are implemented, as reported by Capello et al. (2013 and 2012), even if some of recent optimal strategies have high computational cost (Jain 2018). In our strategy, the control is mainly executed in the Cloud, exploiting the advantages of server capabilities. Therefore, advanced algorithms with a high computational cost can be implemented.

## 4. Mission coordinator

Mission coordinator requirements can be summarized in four categories:

1. **Mission Demand:** The mission defines the requirements of position, trajectory and time to execute the smart applications. Depending on the type of mission, requirements include position or time restrictions.
2. **Vehicle Dynamics**: Vehicles cannot execute an arbitrary trajectory, it is necessary to consider the dynamics of each vehicle. For this case, each vehicle configuration defines different restrictions.
3. **Vehicle Resources**: The energy onboard is limited, therefore good endurance capabilities should be mandatory. Relay of the vehicle will be required for some missions. This means that some missions require multiple vehicles.
4. **Traffic management**: Cooperative missions require the use of multiple vehicles, but this leads to additional restrictions, the strategy, for example, must prevent vehicles that are at the same height from having collisions. Moreover, a safe distance must be considered.

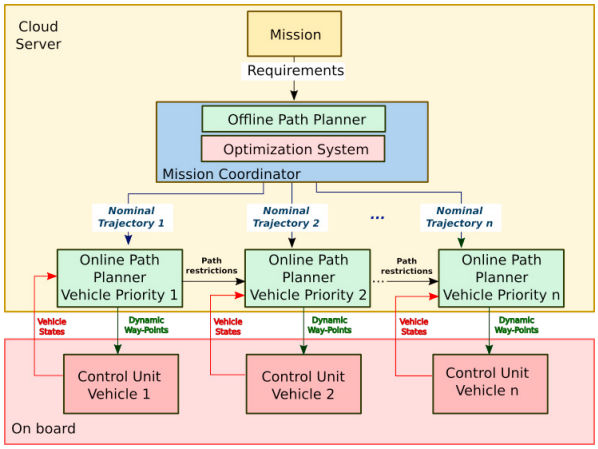


Figure 4 General architecture

The vehicle executes a cooperative action, in which several steps and algorithms are applied at different levels. Figure 4 shows the architecture of the approach used for the vehicle coordination. The mission provides the requirements to the mission coordinator, which provides each vehicle with a nominal trajectory. The mission coordinator minimizes a functional cost, considering the mission and vehicle dynamics constraints. This functional is defined as multivariable system and may include the covered distance, the battery consumption and the risk associated to the defined trajectory. The vehicle path planners are hierarchical. This means that the vehicle priority is assigned when there are intersections between vehicle safety radii to avoid collisions, according to the lowest battery level. The trajectory of high vehicle priority is pre-defined and the other vehicles obtain their trajectories taking into account the trajectories of the other UAVs. The functional cost for the optimization is evaluated in two levels: (i) at the vehicle level (Offline Path planner) a simple trajectory is computed and (ii) at mission level (Optimization System) the optimal solution is selected, taking into account all single trajectories provided by the mission coordinator algorithm (explained in detail in Section 4.1).

To fulfill a mission, different tasks must be executed and each task is performed by one or multiple vehicles. For example, a delivery mission can have several packages and, in this case, the delivery of a single package is a task. To accomplish the mission, all tasks must be executed. As previously described, in our work $m$ docking stations and $n$ available vehicles are considered. The docking station positions $\boldsymbol{x}_{dk,j}(t)$, with $j=1,...,m$ are known. A tree structure is built to accomplish the tasks optimally and to find all possible solutions, considering available vehicles and docking stations. Each node of the tree represents a state of a scenario, edges between nodes have associated vehicle displacements and a branch is a set of nodes and edges that accomplish the mission. All trajectories between docking stations are computed offline and all nominal states are known with their related functional costs.

For the delivery case, let assume that the cargo does not overflight the same docking station twice. The tree is determined considering all docking stations not yet employed and all available vehicles. At the end of this process, all branches bring to the target point and the branch with minimum functional cost represents the optimal trajectory of the cargo. This solution can include multiple vehicles and in a docking station more than one vehicle can stop to

perform cargo transfer suitable to its capacity. Moreover, the system operator has the great potentiality to modify the functional weights, setting the mission priorities. Indeed, each mission has its own features in order of importance, such as in the delivery of first aid resources, time could be the priority. The functional cost $J$ is

$$J = \sum_{j=1}^{i} c_j b_j \ , \#(2)$$

where $\boldsymbol{b} = [b_1, \cdots, b_i]^T$ is the weight vector optimization and $\boldsymbol{c} = [c_1, \cdots, c_i]^T$ is the vector to be optimized and is determined from the offline path planner algorithm based on vehicle model and variables of the environment such as people density, sheltering factor and obstacles.

### *4.1. Tree Construction*

Through the offline path planner and the mission requirements, all possible trajectories between docking stations ($d_i \rightarrow d_j$) are evaluated and stored. The trajectories can be optimized according to some parameters, included those affecting time of day in which the UAV is expected to fly. Algorithm 1 shows the general procedure to generate the tree delineating the optimal trajectory. Nodes are generated according to the number of vehicles and docking stations.

**Algorithm 1** Smart Tree Generation

```
procedure TREEGENERATION(N,M, TRAJECTORIES)
    Add Initial Node
    Add Node
    for j=1:Number of vehicles (n) do
        Add Edge c(Origin → SmartApplication)
    Start Node =Second Node
    End Node =Current Node
    for i=1:Number of docking stations (m) do
        for j=1:Number of vehicles (n) do
            for l=Start Node:End Node do
                if Node is not the Target then
                    for k=1:Number of docking stations (m) do
                        if Node is not in the path then
                            Add Node
                            Add Edge c(d_i → d_j)
                    Add Node
                    Add Edge c(d_i → SmartApplication)
        Start Node = End Node
        End Node = Current Node
```

As an example, Figure 5 shows a scenario with four available docking stations (yellow point), two available vehicles, a start cargo and a target point. Most of the parameters used for the path planning and guidance have a slow dynamics. Fast dynamic obstacles and unexpected events are handled by low level control, not treated here. Regardless of the mission or tasks, vehicles move between docking stations, to uncoupling a multi-vehicle problem in a path generation problem for a single vehicle between docking stations. Calculating offline trajectories and storing them allow to reduce computational load in real time, to anticipate battery consumption and to determine the functional variables $\boldsymbol{c}$.

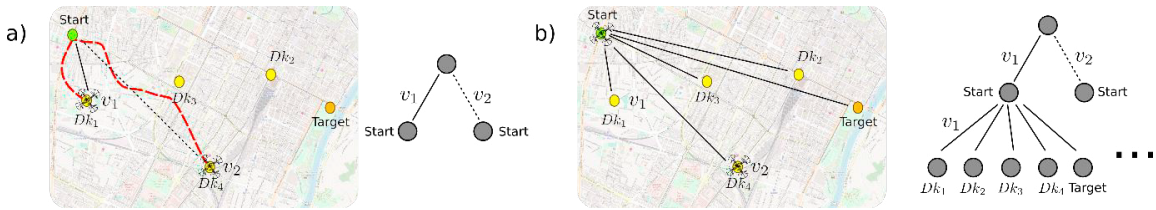


Figure 5 a) Pick-up cargo b) Cargo to docking stations

Figure 5 a) shows the first step of the tree construction for an urban multi-vehicle delivery task. Docking stations are represented with yellow dots and the vehicles are represented with icons. The first two edges represent the vehicle displacement from their current positions to the cargo in order to pick-up the package. There will be many initial edges as the number of the vehicles. In particular in the following figures, the dashed edges represent the displacement of vehicle two. The red line in Figure 5 a) shows the actual trajectory obtained by a single vehicle based on model and requirements. For the coverage case, this first step is not necessary.

The second step is building the tree (as in Figure 5 b)). In this example, vehicle one has reached the cargo point and has collected the package. After this point, vehicle one has several possibilities: it can move to docking stations as intermediate step, in case of low power level, or it can reach directly the target point. The number of edges in this second level is related to the number of docking stations. Note that Figure 6 shows only the expansion for the vehicle one but the arm for vehicle two follows the same algorithm.

The following edges are determined considering the possibility of a vehicle cargo change. The previous rules are used to continue the process until each tree arm reach the target point. Figure 6 shows an example of optimal path with vehicle change in docking station three. In the general case, the chosen arm discards all trajectories in which endurance is not guaranteed. The optimal branch search is performed with classic tree algorithms, as the one implemented in Farach (1995).

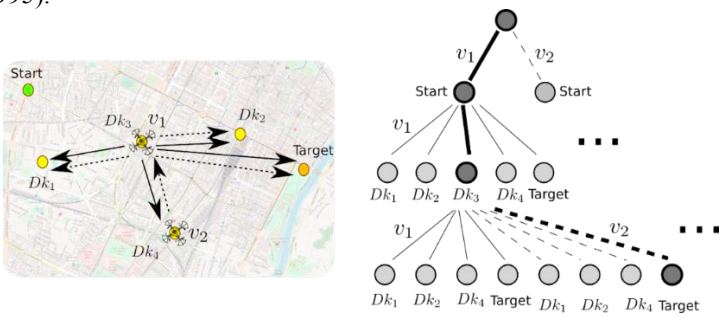


Figure 6 Optimal path example

## 5. Simulation example

The simulation raises the delivery problem shown in Figure 7, where two docking stations and two vehicles are available. The task of mission is to transport the cargo from the pick-up point to the target point.

Following the algorithm 1, the tree obtained from the simulation case is shown in Figure 10 a), where each terminal node represents a possible sequence to complete the task. The tree is generated from all possible trajectories that are shown in Figure 8 and Figure 9 and a polynomial trajectory is assumed to respect dynamic constraints. Using optimization criteria, the optimized covered distance ($c_1 = distance$) by the vehicles and the optimized consumption of the battery ($c_2 = Battery\ Power$) are created. The weights $\boldsymbol{b} = [b_1 \quad b_2]$ give priority to the distance ($b_1 \gg b_2$). The results of vehicle positions are shown in Figure 10 b) and cargo position in Figure 11. The solution expects that vehicle one picks up the cargo, arrives in the docking station two where a cargo transfer occurs and, finally vehicle two reaches the target.

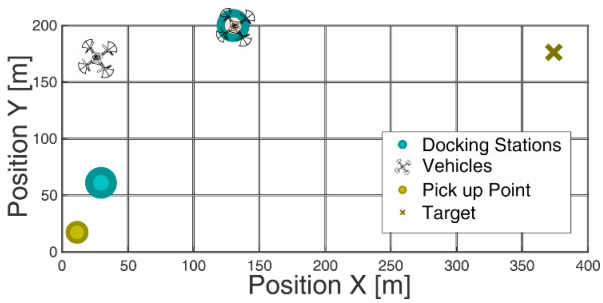


Figure 7 Simulation Scenario

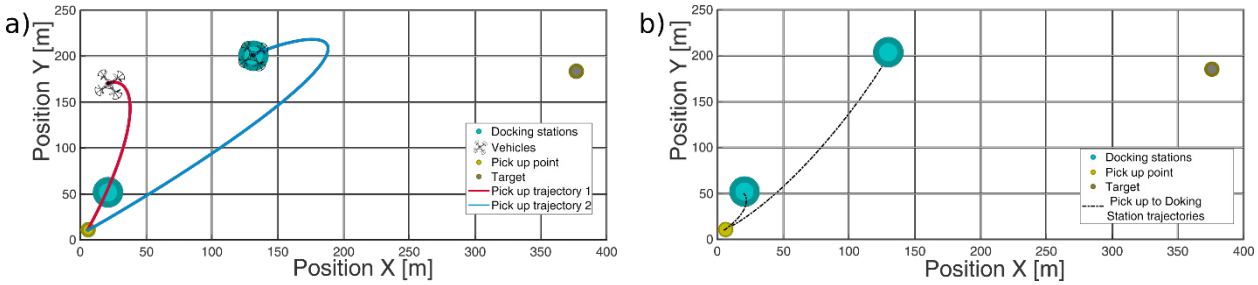


Figure 8 a) Pick up trajectories b) Starting to docking station trajectories

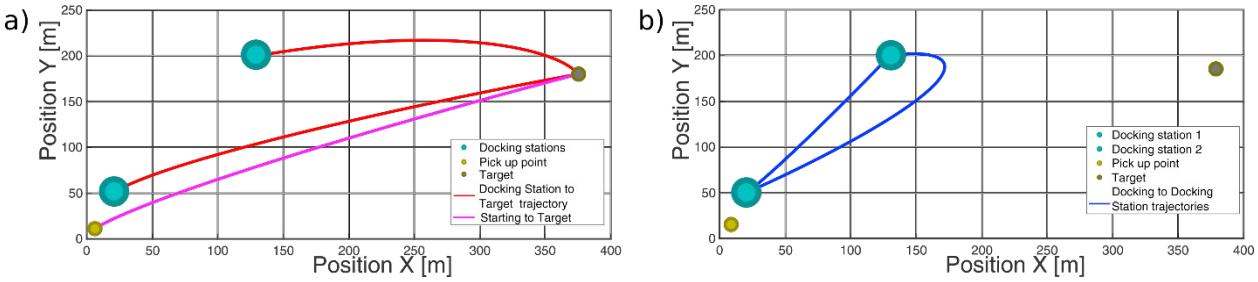


Figure 9 a) Starting to target trajectories b) Docking to docking station trajectories

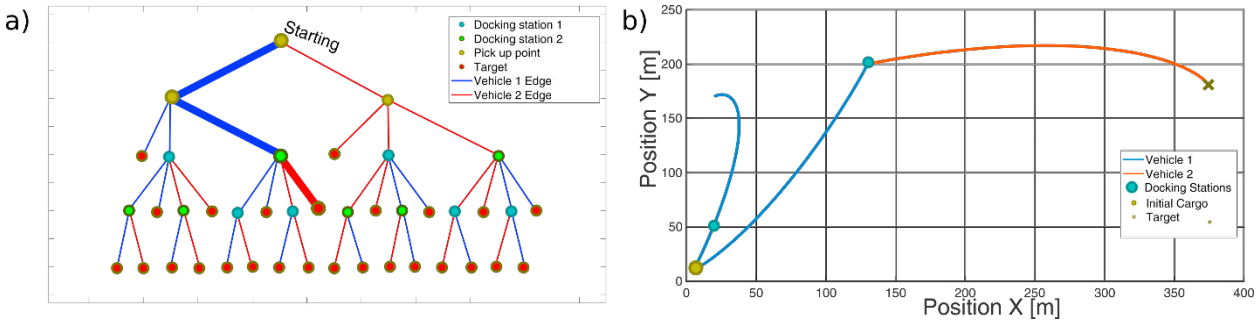


Figure 10 a) Simulation tree b) Vehicle 1 position

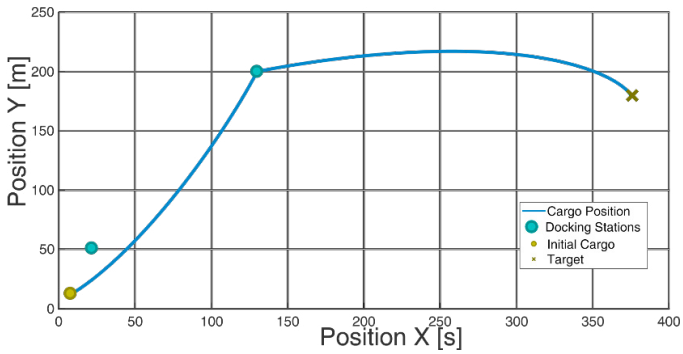


Figure 11 Cargo Position

## 6. Conclusion

In this paper, a high-level methodology for the autonomous coordination of a fleet of UAVs in the field of urban applications has been presented. Coverage and delivery missions have been considered as the main applications in a smart city scenario. Furthermore, a Cloud-Based approach for the management of ground and aerial vehicles has been proposed with a simple simulation example to explain the feasibility study.

Finally, future works will be focused on more simulations with a fleet of heterogeneous vehicles and dynamic docking stations in real urban environments.

## 7. Acknowledgement

This work has been partially supported by a fellowship from TIM, Italy, by Siebel Energy Institute, USA, by Compagnia di San Paolo, Italy and by MIT-Italy MITOR seed grant.

## References

Amazon.com, Inc (2018). Amazon prime air [Website]. Retrieved from https://www.amazon.com/primeair (May, 2018)

Capello, E., Dentis, M., Mascarello, L. N., & Primatesta, S. (2017, October). Regulation analysis and new concept for a cloud-based UAV supervision system in urban environment. In Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2017 Workshop on (pp. 90-95). IEEE.

Deakin, M. (2014). Smart cities: the state-of-the-art and governance challenge. Triple Helix, 1(1):7

DHL Express (2018) DHL Parcelcopter 3.0 [Website]. Retrieved from http://www.dpdhl.com/en/media_relations/specials/parcelcopter.html (May, 2018)

Etkin, B. (2012). Dynamics of atmospheric flight. Courier Corporation.

Farach, M., Kannan, S., Warnow, T., 1995. A robust model for finding optimal evolutionary trees. Algorithmica 13, 155–179.

Farid, G., Hongwei, M., Ali, S. M., & Liwei, Q. (2017). A review on linear and nonlinear control techniques for position and attitude control of a quadrotor. Control and Intelligent Systems, 45(1).

Fotouhi, A., Ding, M., & Hassan, M. (2017, June). Understanding autonomous drone maneuverability for internet of things applications. In A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017 IEEE 18th International Symposium on (pp. 1-6). IEEE.

Galceran, E., Carreras, M., 2013. A survey on coverage path planning for robotics. Rob. Auton. Syst. 61, 1258–1276.

Jain, R.P.K., Aguiar, A.P., Alessandretti, A., Borges de Sousa, J., 2018. Moving Path Following Control of Constrained Underactuated Vehicles: A Nonlinear Model Predictive Control Approach, in: 2018 AIAA. American Institute of Aeronautics and Astronautics, Reston, Virginia.

Mahmoud, S., Mohamed, N., & Al-Jaroodi, J. (2015). Integrating uavs into the cloud using the concept of the web of things. In Journal of Robotics, 2015, 10.

Mohammed, F., Idries, A., Mohamed, N., Al-Jaroodi, J., & Jawhar, I. (2014, May). UAVs for smart cities: Opportunities and challenges. In Unmanned Aircraft Systems (ICUAS), 2014 International Conference on (pp. 267-273). IEEE.

Navarro, I., & Matía, F. (2012). An introduction to swarm robotics. ISRN Robotics, 2013.

Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E., 2018. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. Networks.

Valavanis, K. P. and Vachtsevanos, G. J., editors (2015). Handbook of Unmanned Aerial Vehicles. Springer Netherlands, Dordrecht.

Varela, G., Caamaño, P., Orjales, F., Deibe, Á., López-Peña, F., & Duro, R. J. (2011, October). Swarm intelligence based approach for real time UAV team coordination in search operations. In Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on (pp. 365-370). IEEE.

Wei, Y., Blake, M. B., & Madey, G. R. (2013). An operation-time simulation framework for UAV swarm configuration and mission planning. Procedia Computer Science, 18, 1949-1958.