A Cloud-based Vehicle Collision Avoidance Strategy for Unmanned Aircraft System Traffic Management (UTM) in Urban Areas

(Article begins on next page)

A Cloud-based Vehicle Collision Avoidance Strategy for Unmanned Aircraft System Traffic Management (UTM) in Urban Areas

Stefano Primatesta[1], Matteo Scanavino[1], Andrea Lorenzini[2], Francesco Polia[2]
Enrico Stabile[2], Giorgio Guglieri[1], Alessandro Rizzo[3]

***Abstract*—Unmanned Aircraft Systems are increasingly used to monitor and sense our cities and the diffusion of UAS will require a Traffic Management System to coordinate UAS in the low-altitude airspace. In this paper we propose a collision avoidance strategy to be implemented in an Unmanned Aircraft System Traffic Management (UTM). The proposed strategy relies on a Cloud-based architecture that monitors and manages the low-altitude airspace, as well as coordinating the fleet of UAS. The strategy uses a Priority-based Model Predictive Control approach to define the optimal trajectory of the UAS, avoiding obstacles and other UAS with higher priority. The optimal trajectory is shared with other UAS to communicate the own motion track to be avoided by other UAS.**

The suggested method is implemented and tested in simulations with three UAS with conflicting trajectories. Preliminary results positively support the proposed approach.

***Index Terms*—Unmanned Aircraft System, UTM, Collision Avoidance**

I. Introduction

Unmanned Aircraft System (UAS) will play an important role in our daily life. Thanks to their flexibility and low cost, UAS are already used in a wide range of applications [1], both for commercial and personal use.

Even if UAS are getting popular in the last years, they are at the beginning of their popularity. According to [2], the global UAS market generated $25.59 billion in 2018 and they estimate an annual growth of 8.45% during the forecast period 2019-2029.

This trend is caused by the technological evolution of UAS. In fact, the use of unmanned aircraft in Beyond Visual Line-Of-Sight (BVLOS) with increasing of autonomy level will enable the use of UAS in many applications [3].

One of the key technology is the use of the newest mobile networks (e.g. 4G and 5G) to connect the aircraft with a ground segment. Mobile networks offer a secure and reliable connection, long-range, with large bandwidth and with a high quality of service (QoS) [4]. With mobile networks the UAS is connected with Internet opening new opportunities [5]. The UAS can exploit Internet technologies and, in particular, Cloud technologies, such as Cloud Computing and Big Data, opening the Cloud Robotics concept to UAS [6].

Cloud Robotics is a popular concept in robotics in which most of the intelligence resides on the Cloud, while only some essential tasks are executed on-board the robot [7]. Hence, the UAS has unlimited resources on-Cloud and advanced techniques can be used to provide autonomous capabilities.

In the next future UAS will be widely used also in urban areas, introducing many benefits to Smart Cities [8]. Unmanned aircraft are the ideal platform to sense and monitor cities and, for this reason, the term Internet of Drones Things (IoDT) is used in [9]. UAS have potential to collect data as eye-in-the-sky platform and allow fast package and medical deliveries, avoiding traffic congestion, natural and artificial barriers in cities (i.e. rivers, railways). However, cities are a complex and dynamic environment and the integration of UAS in urban areas poses important challenges such as public safety, privacy and cybersecurity [10]. Moreover, with the extensive use of UAS, many aircraft will share the same low-altitude airspace. According to [11], by 2030, it is expected to have 14.2 million of drones in the airspace. For this reason, it is mandatory to define a UAS Traffic Management (UTM) to enable the use of civilian UAS in the low-altitude airspace. At present, there are some UTM projects in development, such as the NASA UTM project [12] in US, and the U-SPACE [13] project in Europe.

In this work we propose a trajectory planning strategy to solve the problem in traffic management between UAS in the same airspace. The proposed solution is based on a Priority-based Distributed Model Predictive Control (MPC) approach. When a potential conflict is detected, the MPC generates an optimal trajectory avoiding other vehicles with higher priority. The proposed strategy relies on a Cloud-based architecture to provide all the functionalities necessary for the UAS Traffic Management. The overall architecture is distributed between the Cloud and UAS, which makes the system easily scalable.

This paper is organized as follows. In Section II, the Cloud-based architecture used to provide the UAS Traffic Management is presented. Then, in Section III, the proposed trajectory planning with the priority-based collision avoidance is presented. Finally, Section IV reports preliminary results.

[1]S. Primatesta, M. Scanavino and G. Guglieri are with the Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy. Corresponding author (e-mail: stefano.primatesta@polito.it)

[2]A. Lorenzini, F. Polia and E. Stabile are with the Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy.

[3]A. Rizzo is with the Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy.

Then, our conclusions are drawn in Section V.

II. Cloud-based Architecture

The proposed strategy relies on a Cloud-based architecture able to manage, coordinate and monitor a fleet of UAS in the same airspace. The proposed architecture is illustrated in Figure 1 and is distributed between the Cloud and UAS. The architecture has three main blocks: *(i)* the Coordinator Manager (CM) manages and monitors the low-altitude airspace; *(ii)* the Navigation Manager (NM) determines the route of each UAS; and, *(iii)* the UAS block consists in both hardware and software on-board the aircraft.

According to the architecture, the Navigation Manager and the UAS blocks are allocated for each UAS representing the so-called Service Space of the UAS in the Cloud-based architecture. On the other hand, a unique Coordinator Manager exists as a central element of the architecture. In the next paragraphs, each block is described in details.

A. Coordinator Manager

The Coordinator Manager is the core of the architecture. It coordinates and monitors all vehicles managed by the Cloud-based architecture and determines which vehicles interact each other.

When a new vehicle is included in the UTM, the CM assigns a priority level to a UAS. The priority can be assigned with different criteria, such as considering the type of mission or the mass of the aircraft. Moreover, we assume that two UAS cannot have the same priority level, in order to handle all trajectory conflict. Once a UAS is included in the UTM, the CM continuously monitors the UAS by tracking its position and computing all Euclidean distances between UAS. If two (or more) vehicles are within a predefined distance d_{op}, the Coordinator Manager notifies the Navigation Manager of the UAS with lower priority to change its trajectory if needed, while the high priority UAS continues on its route.

B. Navigation Manager

The Navigation Manager aims to determine a safe route of the UAS. As already explained, the NM is allocated for each vehicle managed by the UTM and consists in several elements: Map Manager, Risk-aware Path Planning, Trajectory Planning and Connection Diagnostic.

1) Map Manager: The Map Manager determines the map used by the Risk-aware Path Planning to plan a flight mission. The map should cover the operational area of the UAS defined by the mission requirements. In the proposed architecture the Map Manager defines a risk-based map, i.e. a two-dimensional map that quantifies the risk of flying over an urban area. The risk is defined as the *hourly* probability to cause a casualty computed with a probabilistic risk assessment approach, where the risk is defined as a sequence of three conditional events: *(i)* the loss of control of the aircraft with uncontrolled impact on the ground, *(ii)* the impact with at least a person after the crash, and *(iii)* the impacted person suffers fatal injuries. Moreover, the risk-based map defines no flyable areas because of obstacles at the flight altitude and no-fly zones imposed by National aviation authorities. The risk-based map used in this paper is presented in our previous work. Please refer to [14] for detailed information about the generation of the risk-based map.

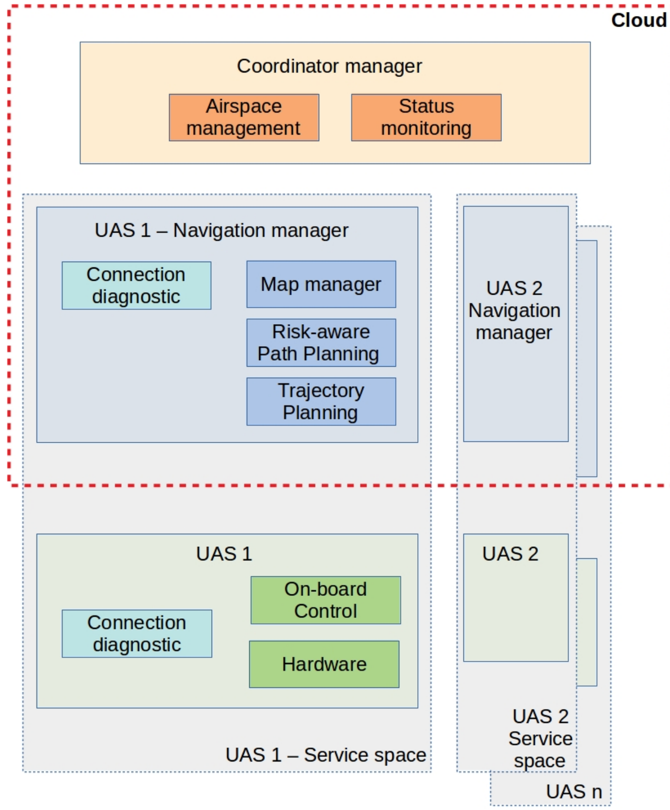

Fig. 1. The Cloud-based architecture for the UAS Traffic Management.

2) Risk-aware Path Planning: The Risk-aware Path planning plans the global flight mission considering the risk-based map. Given the current position of the UAS and a target point defined by the mission requirements, the Risk-aware Path Planning searches for the minimum risk path minimizing the overall risk and the flight time. Practically, the Risk-aware Path Planning solves an optimal path planning problem.

In this work we use the method proposed in our previous work [15], in which the risk-aware path planning algorithm is based on the Optimal Rapidly-exploring Random Tree (RRT*) with the minimization of risk costs defined by the risk-based map and flight time. RRT* is able to compute near-optimal solutions rapidly, even in high-dimensional spaces. Please refer to [15] for more details about the risk-aware path planning.

3) Trajectory Planning: The Trajectory Planning generates the local trajectory to follow the global mission defined by the Risk-aware Path Planning. The Trajectory Planning follows each waypoint sequentially avoiding obstacles and other vehicles. When two (or more) UAS have conflicting trajectories, the Coordinator Manager notifies the Navigation Manager of the UAS with lower priority and, then, the Trajectory Planning avoids UAS with higher priority. In this paper we use a

Priority-based Distributed Model Predictive Control approach detailed in Section III.

C. UAS

The UAS block consists in both hardware and software on-board the vehicle. The main task performed by the UAS is the On-board Control that executes the trajectory provided by the Trajectory Planning. Generally, the On-board Control is performed by a commercial autopilot, such as the PX4 [16] and Ardupilot [17]. Since the UAS is connected with the Cloud using a mobile network, it is essential to evaluate the quality of the connection using the Connection diagnostic block. The Connection diagnostic resides both on-board and on-Cloud and, if a bad quality of connection or a disconnection occurs, the architecture should manage it, guaranteeing a safe operation. One of the safest and simplest solutions is stopping immediately the disconnected UAS that remains in hovering flight until the connection with the Cloud is resumed. Hence, the Coordinator Manager assigns the maximum priority to the disconnected drone to be avoided by all other aircraft managed by the UTM considering a large safety area, due to the uncertainty of the position of the disconnected UAS. In the worst case, if the disconnection persists, an emergency planning is performed.

III. Trajectory planning

The trajectory planning element is the main contribution of this paper. It computes an optimal trajectory to follow the global flight mission and avoiding other vehicles with higher priority notified by the Coordinator Manager.

The proposed solution relies on a Priority-based Distributed Model Predictive Control approach. The approach is priority-based because the Trajectory Planning avoids only other UAS with higher priority, without considering those with lower priority. The Trajectory Planning block is allocated for each vehicle using a distributed logic. The distributed approach brings many advantages in our scenario instead of a centralized or a decentralized one [18]. In fact, a centralized approach is not suitable with many UAS, because the complexity of a centralized controller increases with the number of agents. On the contrary, with a decentralized approach, a controller is provided for each agent, but without any central unit that coordinates the multi-agent system. With a decentralized approach a dedicated trajectory planning element is allocated for each UAS, but the Coordinator Manager plays the role of coordinator advertising UAS when an interaction occurs. As a consequence, the trajectory planning evaluates other UAS only when they are in the proximity, i.e. within a distance d_{op}.

According to the Model Predictive Control approach [19], the planner searches for an optimal trajectory evaluating the future behaviour of the UAS. The future behaviour is optimized over a future horizon according to a receding horizon philosophy. At time t we solve an optimization problem and we obtain the optimal trajectory, but the UAS executes only the first state of the trajectory. The remaining optimal steps of the trajectory are discarded. At time $t+k$ the new optimization problem is solved, considering the updated environments. Moreover, the optimal trajectory over the receding horizon is shared with other UAS with lower priority to have an estimation of the future behaviour of UAS with higher priority and to avoid it.

In our work, the MPC-based approach optimizes a cost function J along the prediction horizon

$$\begin{aligned} J(x(k|k), Z(k|k)) = \sum_{i=0}^{H_p-1} & \lambda_1 \|x_e(k+i|k)\|^2 + \\ & +\lambda_2 \|c_{obs}(x(k+i|k))\|^2 + \\ & +\lambda_3 \|c_{uas}(x(k+i|k), z_m(k+i|k))\|^2, \end{aligned} \tag{1}$$

with H_p is the prediction horizon; $x(k) \in X(k)$ is the state of the UAS at time k and, similarly, $x(k+i|k)$ is the state of the UAS at time $k+i$ estimated at time k; $z_m(k) \in Z(k)$ is the state of the UAS m with $0 \leq m < N$, where N is the number of UAS with higher priority; λ_1, λ_2 and λ_3 are the constant weighting factors.

In the cost function, the first term penalizes $x_e(k+i|k)$, i.e. the distance error between the state $x(k+i|k)$ to the target position, defined by waypoints of the global flight mission that are followed sequentially.

The second term penalizes the *obstacle cost* c_{obs}. Specifically, c_{obs} exponentially increases if the distance between UAS and an obstacle decreases.

The third term penalizes the UAS cost c_{uas}, i.e. a specific cost inversely proportional with the distances with other UAS with higher priority. The UAS cost c_{uas} is defined as follows

$$c_{\text{uas}} = \begin{cases} 1 & \text{if } d \leq d_1 \\ e^{-0.5(d-d_1)} & \text{if } d_1 < d \leq d_2 \\ 0 & \text{if } d > d_2 \end{cases} \tag{2}$$

with d is the actual distance between the UAS and a UAS m with higher priority. d_1 is a safety distance that cannot be violated for safety reason and it is a hard constraint. On the other hand, the safety distance d_2 is a soft constraint and the UAS flies within distance d_2 only if strictly necessary, i.e. if an alternative trajectory with lower cost does not exist. The UAS cost c_{uas} is computed comparing the state $x(k+i|k)$ with the state $z_m(k+i|k)$ at the same time step $k+i$ for each UAS with higher priority.

IV. Preliminary Results

The proposed approach is implement in C++ using the Robot Operating System (ROS) [20] framework. Specifically, each element of the proposed architecture is a ROS node, i.e. an executable ROS-process. In order to test the proposed method, we perform a simulation using ROS, SITL and Gazebo. SITL (Software In The Loop) [21] executes a commercial autopilot on a computer, i.e. PX4 in our simulations. The autopilot controls an aircraft simulated in Gazebo [22], a simulator for robotics application fully compatible with ROS.

The simulation assumes a simplified scenario: *(i)* all vehicles fly at the same fixed altitude; *(ii)* we consider a static

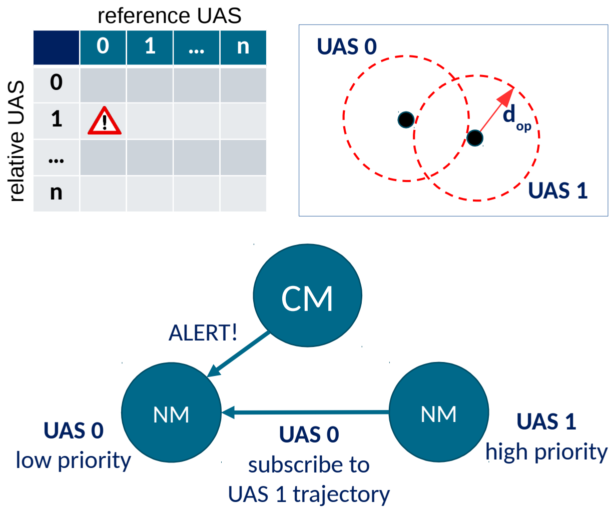

Fig. 2. Graphical representation of the proposed methodology. The Coordinator Manager (CM) monitors all vehicles keeping track of distances between UAS. If UAS 0 and UAS 1 are within a distance d_{op}, the CM alerts the Navigation Manager (NM) of the UAS 0 with lower priority that subscribes to the UAS 1 trajectory to avoid it.

environment, i.e. without considering any unexpected dynamic obstacle; *(iii)* all involved aircraft are of the same model, i.e. the 3DR IRIS, and, as a consequence, the Trajectory Planning is identically tuned and configured for each vehicle; and, *(iv)*, we assume an ideal communication channel without considering latency and packet loss.

The Coordinator Manager subscribes the telemetry data of all UAS registered in the UTM. The registration procedure is very simple in this test: when a UAS is inserted in the simulated environment, the UAS sends a registration request to the Coordinator Manager using a ROS service; if the UAS provides all the required data, the CM responses with a positive response and with a priority number randomly assigned. After the registration procedure, the CM monitors all UAS, computing and keeping track of all the distances between vehicles. If two or more UAS are within the distance $d_{op} = 100$ m, the CM notifies UAS with lower priority to subscribe the estimated trajectory of the UAS with higher priority. In fact, each Trajectory Planning node publishes its trajectory as a ROS topic and, according to the ROS philosophy, topics are broadcasted and are available to all nodes. This logic is graphically represented in Figure 2.

After the registration, the Navigation Manager is allocated by executing the Map Manager, the Risk-aware Path Planning and the Trajectory Planning nodes. Results of the Map Manager and the Risk-aware Path Planning nodes are not reported in this paper, since they are developed in our previous works [14], [15].

The Trajectory Planning computes the optimal trajectory following the global mission and avoiding UAS with higher priority. The UAS is represented using a state $x(k) = [p, v]$, with p is the position and v is the velocity vector state. The optimal trajectory is represented as a sequence of states

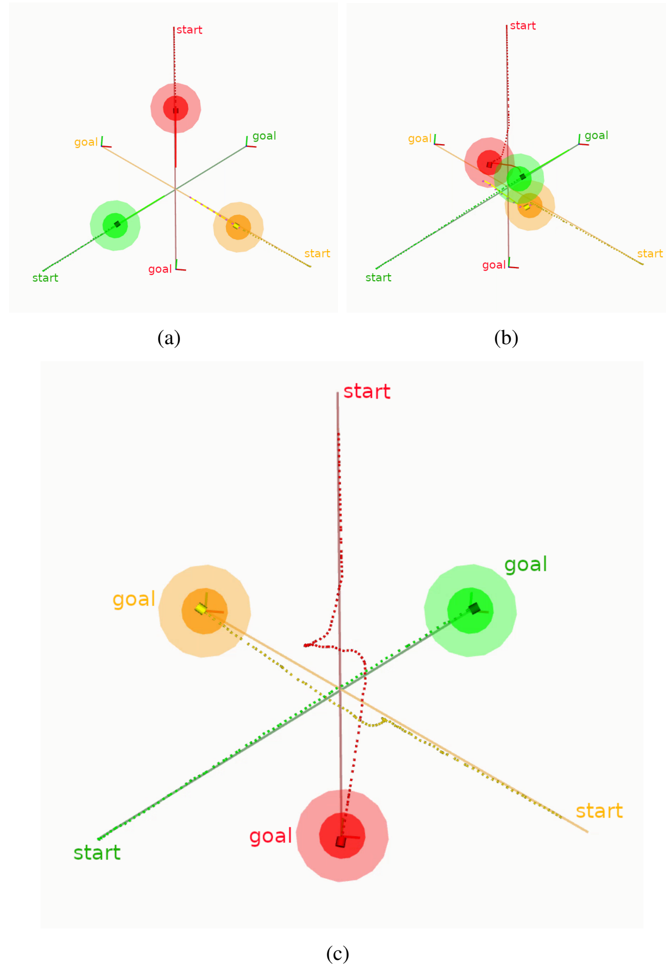

Fig. 3. Preliminary simulation with three UAS: in green the UAS with the highest priority, in yellow with the middle priority and in red with the lowest priority. UAS are represented with a square centred in two concentric circles: the darker circle has radius d_1, while the brighter has radius d_2. In (a), UAS are flying with conflicting trajectories. The Trajectory Planning of each UAS continuously computes a trajectory avoiding other UAS with higher priority in (b) and (c). Dotted lines are the motion of UAS, while continuous straight lines are the reference paths.

over the prediction horizon. Safety distances are defined as $d_1 = 5$ m and $d_2 = 10$ m. The optimization problem is solved using NLOPT [23], an open-source library for nonlinear optimization. The cost function is tuned manually by determining the weighting factors to obtain a goal-oriented and safe trajectory.

As result, the Trajectory Planning computes an optimal trajectory with a frequency of 2 Hz and with a prediction horizon of 15 steps.

Figure 3 illustrates a scenario with three UAS with conflicting trajectories. According with the proposed approach the UAS with the lowest priority (in red) avoids other UAS; the UAS with the middle priority (in yellow) avoids only the high priority UAS; while the UAS with the highest priority (in green) follows the global mission without considering other vehicles. In this scenario UAS compute optimal trajectories and never fly within the safety distances d_1 and d_2 avoiding

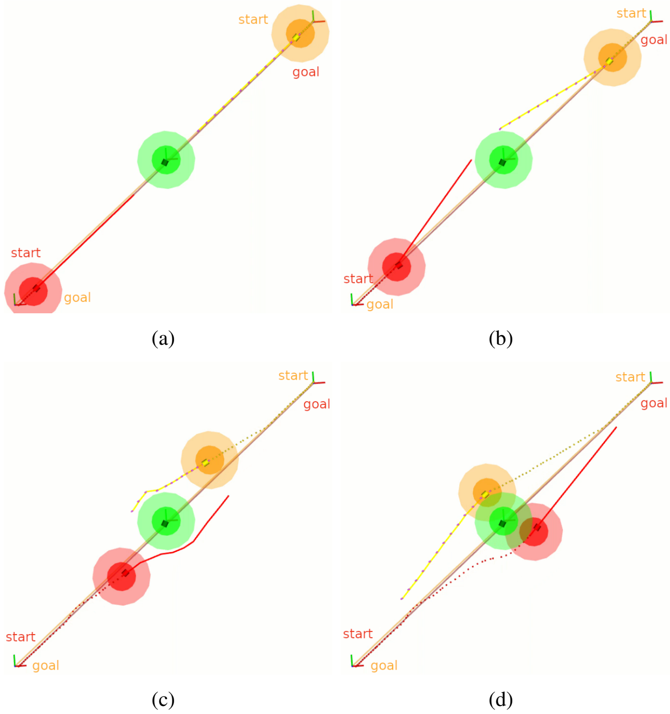

Fig. 4. Preliminary simulation with three UAS: in green the UAS with the highest priority, in yellow with the middle priority and in red with the lowest priority. UAS are represented with a square centred in two concentric circles: the darker circle has radius d_1, while the brighter has radius d_2. In (a), UAS colored in yellow and red are flying with conflicting trajectories, while the green one is hovering on a fixed position. The Trajectory Planning of moving UAS continuously computes a trajectory avoiding other UAS with higher priority in (b), (c) and (d). Dotted lines are the motion of UAS, while continuous straight lines are the reference paths.

collisions with other vehicles.

Another scenario is illustrated in Figure 4 with three UAS, in which two UAS have conflicting trajectory, while another UAS, with the highest priority, is hovering on a fixed position. As in previous scenario, UAS are able to compute optimal trajectories avoiding collisions with other UAS and reaching the desired goal position.

V. Conclusions

In this paper we have proposed a Cloud-based collision avoidance strategy to be implemented in a UAS Traffic Management (UTM). The proposed strategy relies on a Cloud-based Architecture distributed between the Cloud and UAS.

The main contribution of the paper is the Trajectory Planning that uses a Priority-based Distributed Model Predictive Control strategy to compute the optimal trajectory of UAS avoiding obstacles and other UAS with higher priority.

The suggested approach is tested in a simulation environment that shows how the strategy is able to coordinate the fleet of UAS. Future works will include the inclusion of UAS dynamics, as well as the full implementation of the proposed Cloud-based architecture in a more complex scenario, both in a simulation and using real aerial platforms.

Acknowledgment

This work is partially supported by Compagnia di San Paolo and by an Amazon Research Award granted to Dr. A. Rizzo.

References

[1] H. González-Jorge, J. Martínez-Sánchez, M. Bueno *et al.*, "Unmanned aerial systems for civil applications: A review," *Drones*, vol. 1.1, 2017.

[2] BIS Research, "Global unmanned aerial vehicle (uav) market: Focus on vlos and bvlos uavs using satellite communications – analysis and forecast, 2019-2029," BIS Research, Tech. Rep., 2019.

[3] N. Bloise, S. Primatesta, R. Antonini, G. P. Fici, M. Gaspardone, G. Guglieri, and A. Rizzo, "A survey of unmanned aircraft system technologies to enable safe operations in urban areas," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 433–442.

[4] A. Gupta and R. K. Jha, "A survey of 5g network: Architecture and emerging technologies," *IEEE access*, vol. 3, pp. 1206–1232, 2015.

[5] G. Yang, X. Lin, Y. Li, H. Cui, M. Xu, D. Wu, H. Rydén, and S. B. Redhwan, "A telecom perspective on the internet of drones: From lte-advanced to 5g," *preprint*, 2018.

[6] S. Primatesta, E. Capello, R. Antonini, M. Gaspardone, G. Guglieri, and A. Rizzo, "A cloud-based framework for risk-aware intelligent navigation in urban environments," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017, pp. 447–455.

[7] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on automation science and engineering*, vol. 12.2, pp. 398–409, 2015.

[8] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, "Unmanned aerial vehicles applications in future smart cities," *Technological Forecasting and Social Change*, 2018.

[9] A. Nayyar, B.-L. Nguyen, and N. G. Nguyen, "The internet of drone things (iodt): Future envision of smart drones," in *First International Conference on Sustainable Technologies for Computational Intelligence*. Springer, 2020, pp. 563–580.

[10] E. Vattapparamban, İ. Güvenç, A. İ. Yurekli, K. Akkaya, and S. Uluağaç, "Drones for smart cities: Issues in cybersecurity, privacy, and public safety," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2016, pp. 216–221.

[11] BIS Research, "Global uas traffic management (utm) system market: Focus on stakeholders analysis, key technologies enabling utm, and country-wise utm concepts," BIS Research, Tech. Rep., 2019.

[12] NASA, National Aeronautics and Space Administration, "NASA UTM, Unmanned Aircraft System (UAS) Traffic Managment (UTM)," https://utm.arc.nasa.gov/index.shtml, accessed: 2020-01-31.

[13] SESAR Joint Undertaking, "U-space," https://www.sesarju.eu/U-space, accessed: 2020-01-31.

[14] S. Primatesta, A. Rizzo, and A. la Cour-Harbo, "Ground risk map for unmanned aircraft in urban environments," *Journal of Intelligent & Robotic Systems*, pp. 1–21, 2018.

[15] S. Primatesta, L. S. Cuomo, G. Guglieri, and A. Rizzo, "An innovative algorithm to estimate risk optimum path for unmanned aerial vehicles in urban environments," *Transportation research procedia*, vol. 35, pp. 44–53, 2018.

[16] PX4, "PX4 Autopilot," https://px4.io/, accessed: 2020-01-31.

[17] Ardupilot, "Open Source Autopilot," http://ardupilot.org/, accessed: 2020-01-31.

[18] C. E. Luis and A. P. Schoellig, "Trajectory generation for multiagent point-to-point transitions via distributed model predictive control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 375–382, 2019.

[19] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[20] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3.2. Kobe, Japan, 2009, p. 5.

[21] SITL contributors, "SITL guide," http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html, 2017.

[22] N. P. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator." in *IROS*, vol. 4. Citeseer, 2004, pp. 2149–2154.

[23] S. G. Johnson. The nlopt nonlinear-optimization package. [Online]. Available: http://ab-initio.mit.edu/nlopt