In-flight launch of unmanned aerial vehicles

Niels Nauwynck, Haris Balta, Geert De Cubber, and Hichem Sahli

Abstract—This paper considers the development of a system to enable the in-flight-launch of one aerial system by another. The paper will discuss how an optimal release mechanism was developed, taking into account the aerodynamics of one specific mother and child UAV. Furthermore, it will discuss the PID-based control concept that was introduced in order to autonomously stabilize the child UAV after being released from the mothership UAV. Finally, the paper will show how the concept of a mothership UAV + child UAV combination could be usefully taken into advantage in the context of a search and rescue operation.

Index Terms—Unmanned Aerial Vehicles, Control, Autonomous stabilization, Search and Rescue drones, Heterogeneous systems.

1 Introduction

1.1 Problem statement

As more and more unmanned aerial systems are entering our everyday lives, we also see more and more variety in the systems that are being developed, each towards a different application field. This variety should come as no surprise, at is impossible to create one system that would fit all user needs. Heterogeneous systems, all being used at the same time are therefore the way forward. However, this also leads to new problems in terms of interoperability and the search for optimal collaboration strategies between all these different systems.

In this paper, we focus on the collaborative action between two unmanned aerial systems where one acts as a mothership / carrier / launch platform, capable of launching in-flight a smaller child system that can then be used for close-to-ground search and rescue missions.

The in-flight-launch of one aerial system by another is no easy problem and requires the careful consideration of the aerodynamics and control of the two systems. Indeed, in terms of aerodynamics and flight performance, the mothership and the child UAV impose important forces and constraints on one another that are very different when they are mechanically interlinked and from when they are separated from one another. The autonomous control concept which is implemented for this research experiment on the child UAV needs to be able to cope with these sudden changes in real-time at the moment of release in order to prevent a crash.

1.2 Previous Work

In the field of collaborative Unmanned Aerial Vehicles (UAVs), Lacroix et al. studied already in 2007 the multi-agent decision making process between the different systems in [1]. However, taking these concepts to practical applications and the reality on the field has proven to be a difficult operation, due to the complex nature of operating multiple heterogeneous platforms simultaneously. Serrano et al. have proposed in [2] an interoperability concept that enables the message-passing and collaborative control for multiple heterogeneous UAVs and applied that concept on heterogeneous systems developed within the context of the ICARUS project [3]. They put this interoperability and collaboration concept in practice in [4] in a search-and-rescue use case for the euRathlon challenge [5] where multiple heterogeneous systems (though not all airborne) were validated in a Fukushima-like response simulation scenario. While these operations entailed the use of heterogeneous UAV operations, none of the systems featured an in-flight launch capability.

The in-flight launch of one UAV by a mothership is something which has been considered mostly for military operations. Roberts et al. describe in [6] flight tests to determine the flight envelope and launch system configuration for which a small (maximum gross weight of 80 lbs), unpowered UAV glider could be safely launched from the cargo ramp of a C-130 transport aircraft. Safe separation from a C-130 aircraft was demonstrated, as well as UAV stability for successful wings deployment and fly-out. However, these tests considered a manned aircraft as a mothership and only fixed wing aircraft.

1.3 Hardware and software used

The main aim of this research work is to show the concept of the autonomous in-flight launc stabilization system on commodity hardware multi-copters, as opposed to the heavy military sustems where in-flight launch systems have already been shown. Therefore, we chose to work with modest, low-cost equipment, as presented here.

The platform used for the parent UAV is a DJI Phantom 2. This ready to fly, multi-functional quad-copter is easy to fly, offers precision flight and has stable hovering without too much interaction. Throughout this research work, this system remained a closed system where the only communication was done through the included controller. The DJI Phantom 2 is a consumer product not specifically equipped

- *N. Nauwynck and H. Sahli are with the Department of Electronics and Informatics, Vrije Universiteit Brussel, Brussels, Belgium. E-mail: niels.nauwynck@vub.ac.be*
- *H. Sahli is with the Interuniversity Microelectronics Center (IMEC), Leuven, Belgium. E-mail: hsahli@etrovub.be*
- *H. Balta and G. De Cubber are with the Department of Mechanics of the Royal Military Academy, Brussels, Belgium. E-mail: geert.de.cubber@rma.ac.be*

(a) DJI Phantom 2 (b) Parrot AR Drone 2.0

Fig. 1. DJI Phantom 2 and Parrot AR Drone 2.0 UAVs used as mother-UAV and child-UAV in this research work.

(a) Release mechanism on the child UAV (b) Release mechanism on the mother UAV

Fig. 2. Release mechanism on the child UAV.

to carry any load but did offer the requirements for the proof of concept. By removing the pre-installed camera, the total mass of the parent UAV is 1093 g. Figure 1a displays the Phantom 2 without the camera attached.

The platform used for the child UAV is the Parrot AR Drone 2.0. This UAV is mostly conceived as a toy which makes it quite popular and affordable. This UAV has a starting mass of 501 gram. By sacrificing security and durability we are able to reduce the weight with 58 g. This however meant that no protection hull was present during crashes, bringing the lowest mass to 443 g. Figure 1b displays the Parrot AR Drone 2.0 without the protective hull.

The Parrot AR Drone 2.0 is used frequently in research since it is programmable in a ROS [7] interface, making use of WiFi communication for input and output. A ROS-driver is provided to create a communication channel with the UAV. This communication driver offers a great deal of functionalities that were used for the in-flight launch software, such as:

- 3-dimensional rotation values from the X, Y & Z axis;
- magnetometer readings in three-dimensional space;
- pressure from the barometer;
- linear velocity in three-dimensional space;
- linear acceleration in three-dimensional space;
- estimated altitude;
- motor pulse width modulation values;
- forward and downward facing camera stream;
- movability through yaw, pitch and roll.

2 Design of the release mechanism

As the child UAV still has a task to complete after being launched, as much weight as possible should be left on the parent UAV. This meant that a design was made where the actual launch mechanism was hanging on the parent UAV.

A major issue in the design process of developing a release mechanism on the child UAV was to prevent any unwanted rotations due to wind etc, which would cause system instability. Therefore, a child-UAV release mechanism was designed, consisting of a base plate and a locking mechanism, terminating in an O-ring where a hook can be attached. Once the design was fully made it was 3D printed. The design turned out to be 44 g. Adding the 44g to the 443g of the child UAV made sure that the child UAV now had a total mass of 487 g. Note that it is technically not possible for a DJI Phantom 2 to support such a payload, therefore it is required for the child UAV to help with lifting its own mass pre-release by spinning its rotors. Figure 2a shows the result of this design: a lightweight, stern and rotation resistant component capable of carrying the child UAV.

As discussed above, the child UAV can be carried through an o-ring. This was specifically done to create an easy to use launch mechanism on the parent UAV. The major difficulty on the parent-side was to include a mechanism that can increase or decrease the distance between the parent and child UAV. Indeed, due to turbulence effects under the mother aircraft, it is required to release the child UAV at a reasonable distance from the mother UAV, sufficiently away from the turbulence zone. This so-called ”downwash” area can be modeled or experimentally measured [8]. In our case, as we lacked the input of the necessary modelling parameters, an experimental study was required. We therefore needed to experiment with difference release altitudes (measured between the mother and child UAV) in order to study these effects. Therefore, a winch system was developed, consisting of a PCB-controlled servo motor. Once 3D printed, the base plate extension creates a functional winch system as seen in Figure 2b. The parent UAV now has the possibility to lower the UAV to any desired launch height from a remote site. The final design of the parent UAV release mechanism has a mass of 245g, bringing the total mass of the parent UAV to 1338g.

3 Autonomous stabilization

In order to be platform independent a new PID controller is created that takes over the default hovering function embedded in the used devices, taking into account the constant turbulence by the parent UAV. Since we wanted a platform-independent solution, we did not rely for this on the built-in stabilization method that also makes use of the downward facing camera. For the creation of the PID controller, a custom package was created that subscribed to the navigation data and odometry. In return, it could publish to the necessary yaw, pitch and roll values, calculated as control commands to stabilize the UAV.

In the implementation the maximum reference speed of the UAV is limited to 0.6 which prevents it from performing jerky movements. The velocity error is calculated by the difference of the navigation commands of yaw, pitch and roll and the incoming odometry values. This value is assigned to the proportional gain. The integral gain is calculated with the previous integral gain and the proportional gain. By using the proportional gain we are able to determine the integral gain seen on Figure 3, based on a set limit, the current situation of the error (new_err) and the previous integral gain (i_term). Lastly, the derivate gain is calculated by filtering the incoming odometry data.

```
def FilterVelocity(self, velocity):
    result = 0.0


    self.m_input_buffer[0] = velocity

    for i in range(0, 30):
        result += self.m_input_buffer[i] * self.m_coeffs[i]

    for x in range(0, 30):
        self.m_input_buffer[x] = self.m_input_buffer[x - 1];

    return result

def ITermIncrease(self, i_term, new_err, cap):
    result = 0.0
    if new_err < 0 and i_term > 0:
        result = max(0.0, i_term + 2.5 * new_err)
    elif new_err > 0 and i_term < 0:
        result = min(0.0, i_term + 2.5 * new_err)
    else:
        result = i_term + new_err

    if i_term > cap:
        result = cap
    if i_term < -cap:
        result = -cap

    return result
```

Fig. 3. Calculation of the PID controller values

After a successful series of static tests (manually pushing the UAV from its stable position, using the proposed PID-based stabilization method to prevent a crash), in-flight launch tests were performed, with a different separation distance between the both UAVs: 140cm, 100cm and 60cm.

(a) UAVs before launch (b) UAVs after launch

Fig. 4. In-flight launch on the child UAV by the mother UAV.

Using a **140cm launch distance** (https://youtu.be/hvxIr1gvgtc), the PID controller does not need to change the yaw, pitch or roll values. Its only task is increasing the power on all four motors to counteracting the descent. This is a fairly easy task and the release goes therefore quite smooth.

Using a **100cm launch distance** (https://youtu.be/-HsyfGzBpow), the behaviour is in most cases similar to the previous case (140cm). However, sometimes we observe that the child UAV needs to compensate pre-release already for the extra downward forces induced by the downwash of the mother UAV. The result is that the PID controller acquires the correct height by lifting its own weight, not relying on the strength of the parent UAV. Once the child UAV is released, it does not need to adjust anymore to any turbulence anymore, just like in the previous experiments and the release goes smooth.

Using a **60cm launch distance** (https://youtu.be/3Xvp1fMt6tg), the PID controller is no longer capable of recovering the turbulence induced by the rotors of the parent UAV and the child UAV always crashes upon release. In all of the four runs made, the release was never possible because the parent UAV created turbulence on the child UAV. This turbulence interfered with the spinning propellers of the child UAV which made it move all over the place. Because of the moment of the child UAV, the parent UAV also started to wiggle which only increased the movement on the child UAV, repeating this pattern until a crash occurred. Obviously, this means that here we have reached the limits of what was possible with the given platforms and the proposed control and stabilization paradigm.

4 Validation of the concept in a search & rescue use case

In order to present a meaningful use case for the validation of the proposed system, the field of search and rescue was chosen. This specific domain was not chosen by accident, as the specific requirements of the search and rescue workers [9] often demand for multiple heterogeneous robotic tools to be deployed. Indeed, large fixed wing systems are required to have a permanent eye in the sky and to create a map of the area, whereas rotorcraft are generally more suited for outdoor victim search or dropping rescue kits, whereas small rotorcraft are excellent for indoor victim search. In this context, we envision a search and rescue operation where a large UAV launches a smaller one at a specific site, such that this small UAV can go and search for victims.

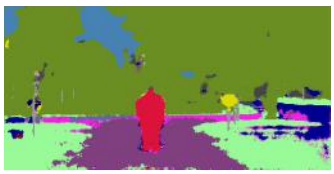
A necessary requirement for using a UAV for victim search is the capability to detect human survivors in a totally unstructured environment. For scene analysis, using the on-board camera, the UAV has to detect and classify the objects seen by the camera. For this purpose a a deep neural network is used to achieve semantic segmentation, assigning a class label to every pixel. A deep neural network is another form of an artificial neural network which has shown spectacular accuracy on datasets with large feature and solution space. Since deeper networks often have more vanishing gradient problems and exploding gradient problems, they are harder to train than other networks.

For this application, we will use the on the ENet semantic segmentation algorithm [10], which uses a deep neural network architecture to provide real-time semantic segmentation for self-driving vehicles. By requiring 75 times less FLOPs and 79 less parameters it functions eighteen times faster than existing models by early down-sampling, nonlinear operations, changing the decoder size, regularization and much more.

To train from a dataset a modified version of Caffe [11] was used which supported all the necessary layers for ENet. This requires a training and testing set where first the encoder is trained with pre-labeled objects from the data set [12]. After about 75 000 iterations, we noticed convergence with a minimum of 80% training accuracy. After finishing training the decoder, the encoder was further trained to obtain also a 80% training accuracy.

After launching the child UAV from the parent UAV, the ENET semantic segmentation algorithm was activated on the images of the Parrot AR Drone 2.0 front-facing camera, which has a resolution of 1280x720 at 30 fps. The first test, shown on Figure 5, shows an example of how the output on a small access road to a building to mimic the idea of a small road in open country side.

(a) Visual image frame from the child UAV

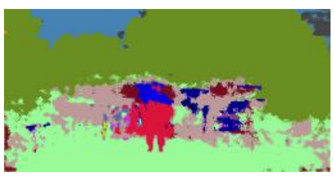


(b) ENET segmentation of the image frame (red=victim)

Fig. 5. ENet's semantic segmentation input image of a lost person on small road in open country side.

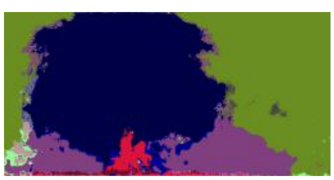
The second experiment set can be seen on Figure 6 and displays the detection possibilities in front of tunnels and shows that while inside a dark tunnel, person detection becomes less obvious.

(a) Visual image frame from the child UAV



(b) ENET segmentation of the image frame (red=victim)

(c) Visual image frame from the child UAV



(d) ENET segmentation of the image frame (red=victim)

Fig. 6. ENet's semantic segmentation output image of a lost person in front and inside of a tunnel.

5 Conclusion

Within this paper, an in-flight launch concept has been proposed for a child rotorcraft UAV by a parent rotorcraft UAV. The solution developed not only in theory, but also in practice, by the design of a release mechanism and a control concept in order to stabilize the child UAV after the launch procedure. The system was extensively validated by multiple launch experiments, evaluating the limits of the control concept. Furthermore, a practical use case was elaborated where this concept could be put into practice: search and rescue. Therefore, a deep neural network was implemented in order to perform a semantic segmentation of the video data of the child UAV (after being released in a disaster area by the parent UAV), enabling autonomous victim search operations.

It must be stressed that the objective of this research work was to provide a proof of concept, using cheap hardware. Future work will thus mainly focus on porting this concept to more performing hardware platforms, such that real use cases can be performed.

References

[1] S. Lacroix, R. Alami, T. Lemaire, G. Hattenberger, and J. Gancet, "Decision making in multi-UAVs systems: Architecture and algorithms," in *Multiple Unmanned Aerial Vehicles*, ser. Tracts on Advanced Robotics, A. Ollero and I. Maza, Eds. Springer, 2007.

[2] D. S. López, G. Moreno, J. Cordero, J. Sanchez, S. Govindaraj, M. M. Marques, V. Lobo, S. Fioravanti, A. Grati, K. Rudin *et al.*, "Interoperability in a heterogeneous team of search and rescue robots," in *Search and Rescue Robotics-From Theory to Practice*. InTech, 2017.

[3] G. De Cubber, D. Serrano, K. Berns, K. Chintamani, R. Sabino, S. Ourevitch, D. Doroftei, C. Armbrust, T. Flamma, and Y. Baudoin, "Search and rescue robots developed by the european icarus project," in *7th Int. Workshop on Robotics for Risky Environments*, 2013.

[4] M. M. Marques, R. Parreira, V. Lobo, A. Martins, A. Matos, N. Cruz, J. M. Almeida, J. C. Alves, E. Silva, J. Bdkowski, K. Majek, M. Peka, P. Musialik, H. Ferreira, A. Dias, B. Ferreira, G. Amaral, A. Figueiredo, R. Almeida, F. Silva, D. Serrano, G. Moreno, G. D. Cubber, H. Balta, and H. Beglerovi, "Use of multi-domain robots in search and rescue operations contributions of the icarus team to the eurathlon 2015 challenge," in *OCEANS 2016 - Shanghai*, April 2016, pp. 1–7.

[5] A. F. T. Winfield, M. P. Franco, B. Brüggemann, A. Castro, M. C. Limon, G. Ferri, F. Ferreira, X. Liu, Y. R. Petillot, J. Röning, F. E. Schneider, E. Stengler, D. Sosa, and A. Viguria, "eurathlon 2015: A multi-domain multi-robot grand challenge for search and rescue robots," in *TAROS*, ser. Lecture Notes in Computer Science, vol. 9716. Springer, 2016, pp. 351–363.

[6] D. W. Roberts and A. D. Judy, "Separation Flight Tests of a Small Unmanned Air Vehicle from a C-130 Transport Aircraft," *NATO STO Meeting*, vol. MP-SCI-162-19, 2012. [Online]. Available: {https://www.sto.nato.int//publications//STOMeetingProceedings//RTO-MP-SCI-162//MP-SCI-162-19.pdf}

[7] M. Quigley, K. Conley, B. P. 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*, 2009.

[8] M. Sjholm, N. Angelou, P. Hansen, K. H. Hansen, T. Mikkelsen, S. Haga, J. A. Silgjerd, and N. Starsmore, "Two-dimensional rotorcraft downwash flow field measurements by lidar-based wind scanners with agile beam steering," *Journal of Atmospheric and Oceanic Technology*, vol. 31, no. 4, pp. 930–937, 2014. [Online]. Available: https://doi.org/10.1175/JTECH-D-13-00010.1

[9] D. Doroftei, A. Matos, and G. De Cubber, "Designing search and rescue robots towards realistic user requirements," in *Advanced Concepts on Mechanical Engineering (ACME)*, 2014.

[10] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," *CoRR*, vol. abs/1606.02147, 2016. [Online]. Available: http://arxiv.org/abs/1606.02147

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," *ArXiv e-prints*, Jun. 2014.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.