Part B
Literature review

In this section, an extended literature review is presented which is conducted in the individual sections of the dissertation. Most of the sources are conference papers and journal articles on the basis that they are published recently and thus are more up-to-date compared to books. They reflect the latest developments in each discipline. The vast majority of the sources are less than four years old (dating from 2011 to present). Thorough research is undertaken for every major part of the dissertation in order to get a rounded view of the undergoing research and development that is being carried out.

B.1 Radiation mapping

The radiation anomalies mapping procedure is one of great importance since it can ensure the public safety in cases where disaster happened or in monitoring of nuclear power stations. There are many examples that can show the importance of radiation mapping. In March 2011 The Fukishima Daiichi nuclear power plant was severely stricken by a tsunami causing Units 1 – 4 to contaminate the environment with significant atmospheric release of radioactive material (MEXT, 2015). This case clearly illustrates the requirement for technology that allows rapid and accurate mapping of ground and plume radiation contamination. Moreover, in nuclear power plants there is a constant need for validation of safe operation. This can be achieved by continuous routine surveys to identify contamination releases. Finally, Radon mapping is also important because it can affect the public health as it accumulates inside buildings and basements, greatly increasing the risk of lung cancer.

There are three methods of radiation mapping. Static ground based surveys, which performed by arrays of above ground in situ measuring stations or handheld devices. In situ ground level measurements improve spatial averaging at the cost of vertical profiling. The handheld measurement devices produce an accurate and sensitive representation of how radioisotope concentrations vary spatially or depending on depth. On the other hand, this method is time consuming and captures a limited snapshot of radionuclide distribution (Sanderson et al., 1995). Car-borne radiation surveys conducted using vehicles equipped with detectors, which are calibrated and deployed quickly, and can map 400–1000 km every 24 hours (Mellander, 1995). Using this process, more measurements can be collected compared to the first method but data collection is limited to the road network (hence rural data is often sparse) and to low driving speed (< 30 mph) to maintain an adequate resolution. The third approach, which is the one the project refers to, relates to airborne radiation surveys. This can cover great areas of interest expeditiously. Coverage per unit is 10^6-10^7 times greater than that of ground based and 10^2-10^3 times that of car-borne surveys because of the greater ground clearance, coverage and the lack of obstacles (Schwarz et al., 1995). It can also be considered the safest since the remote nature of the sensing removes pilots from a percentage of the ground based radiation exposure, although this is not true of radiation plume mapping. In this method, the most influential factor is altitude. Air attenuates radiation and consequently as altitude increases, the intensity detected decreases rapidly (fig.B.1).

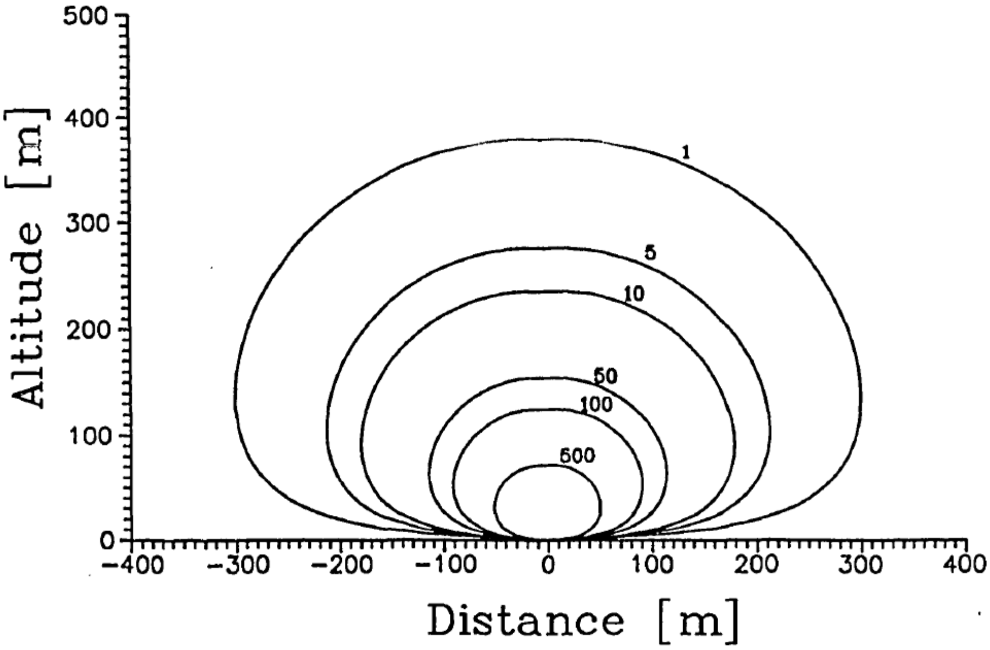

Fig.B.1 Modeled count rate (counts per second) isolines, using a large volume NaI scintillation detector, of a 1.9 GBq (50 mCi) ^{137}Cs source buried at 1 cm depth. (image from Schwarz et at., 1995)

A subcategory of the third approach, which offers the benefits from all three methods, is radiation mapping using UAVs. The utilization of this method seems ideal for the following reasons:

- It can produce high resolution mappings similar to the hand held based surveys since UAVs can hover in low altitudes and follow dense trajectories in given areas.
- UAVs approach the territories of above and avoid local obstacles resulting in a low sampling density. In addition, they have the ability to fly steadily with low velocity in close proximity to nuclear premises and collect accurate measurements.
- UAVs require lower initial expenditure than airborne radiation surveys.
- UAVs ability is not influenced from radioactive debris that could contaminate the UAV like car tyres, which are in constant contact with the road and stick materials from the surface of the road.

One successful approach that managed to implement a UAV based airborne radiation survey comes from Imitec (Imitec Limited, 2014), a University of Bristol spin-out company. They have developed a remote isotropic analysis system, which consist of a lightweight gamma spectrometer and positioning devices, integrated in a UAV (fig.B.2). Using this combination, it is possible to measure meter resolution maps of radiation including over high dose areas and inaccessible locations. It can also provide aerial imaging and observation in conjunction with the radiation measurements. Finally, there is the ability to perform pre-programmed flight paths, which is advantageous for scenarios where routine monitoring is required.

Fig.B.2 The Imitec's Advanced Airborne Radiation Monitoring (AARM) UAV (image from www.imitec.co.uk)

B.2 UAV flight

In every mobile robot application, semi-autonomous or autonomous, the task of mapping, localisation, navigation and collision avoidance is important decision in design. The implementation of each part can be done using properly adapted versions of popular methods and probabilistic algorithms available in the robotics literature. This literature review includes the knowledge of related papers concerning UAVs. In order to be more precise I focused my research on quad or multi copters in order to exploit the strengths and combat the weaknesses in this specific configuration.

The Mendes and Ventura 2012 paper (Mendes and Ventura, 2012) proposes an interesting approach to tackle the problem of assisting tele-operation to safely pilot a UAV; a concept that is closely related to the current project. More precisely, according to the environment in which the UAV flies, driven by the operator, the researchers designed a process that overrides this operator input either by modulation, inhibition, or replacement with a different input. The desired goal is to use sensor data to determine an appropriate assisted tele-operation in order to guarantee, to the best of sensor capabilities, a safe flight.

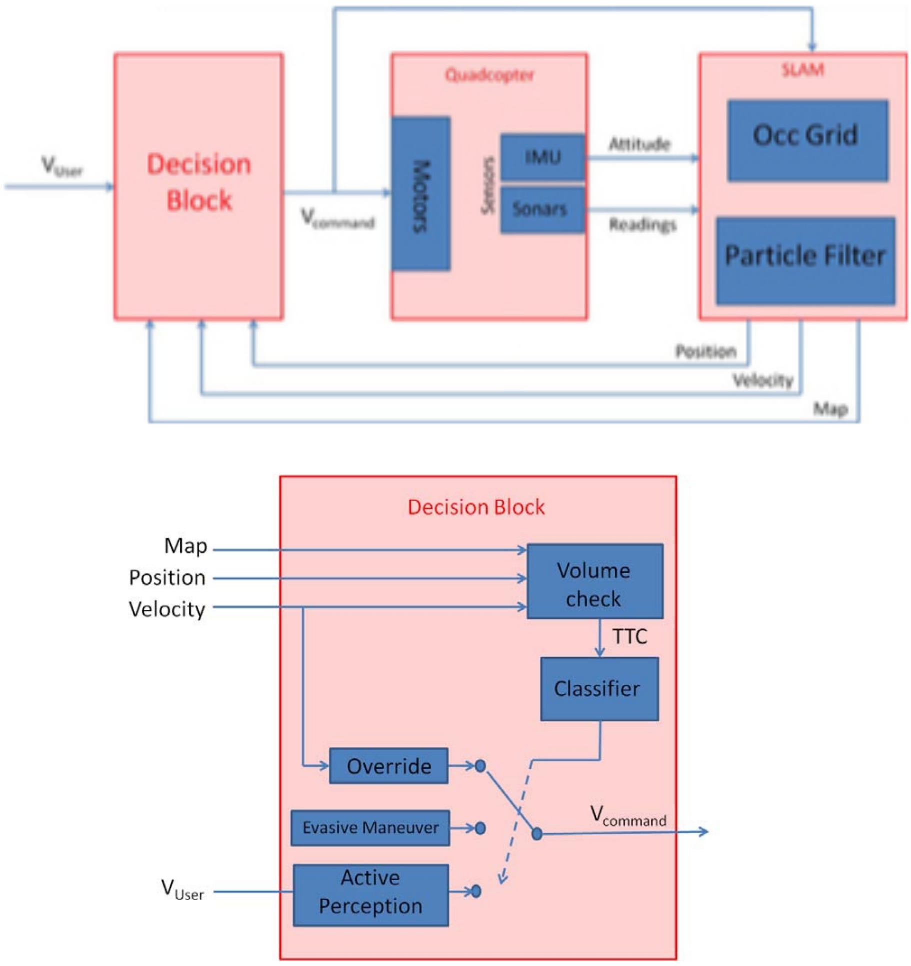

Fig.B.3 Full architecture of the proposed approach and flow chart of the decision block (image from Mendes and Ventura, 2012)

The investigators assume that the UAV operates in unknown, unstructured, GPS-denied, and indoor environments, therefore the main potential threat is collision with obstacles. Thus, distance sensors are used to detect and map these obstacles. The approach is based on the FastSLAM algorithm, together with 3D occupancy grid mapping. Using these techniques they are able to estimate the vehicle's position in the map. At this point a classifier makes a decision based on danger assessment. This classifier overrides the user's inputs if they compromise the quad copter's physical integrity in the near future. Overriding may extend from simple velocity reduction to, in extreme cases, an evasive manoeuvre (fig.B.3).

The approach assumes that the attitude is known using the on-board IMU so the problem is to define the position of the UAV. Using FastSLAM they can estimate both the position and the map of a robot simultaneously. The classifier takes the position estimation and the map as inputs. It categorises each cell of a 3D occupancy grid in one of 3 states Free, Unknown and Occupied. Given the operator's inputs, the classifier computes the desired direction for the vehicle and the distance to the first non-Free cell. If the closest cell is occupied, the inputs are altered accordingly so that the UAV will remain safe and then they are applied to the navigation. If the closest cell is Unknown and if the distance to it is higher than a certain threshold, the algorithm checks whether there is any sonar aligned with the desired direction. Otherwise, the algorithm will autonomously rotate the vehicle and then applies the same speed that the user demanded.

This remarkably interesting approach is based on short-term, rough mapping of the nearby environment. The researchers argued that the real world experiments were successful and thus verified the design. However, it is also mentioned that the four sonars, which were used as proximity sensors, recorded the readings with the quad-rotor's motors set off, for safety issues. Despite the safety benefits in such a procedure the result of the tests could be inaccurate, considering that sonars with their rotors switched on take error prone readings.

In the Chen 2013 paper (Chen et al., 2013) the effort of the research team to develop a UAV system capable of autonomously navigating in a one-level building and producing a 2D map of the space is presented. Moreover, there were specific requirements from the academic partners that needed to be satisfied. More precisely, the system had to be based on an inexpensive, upgradeable, and reconfigurable solution with commercial off-the-shelf (COTS) products.

The hardware of the project consists of an AeroQuad quad copter, a 54-Pin Arduino Mega SDK microcontroller, a Samsung Galaxy SIII Android smartphone, and MaxBotix ultrasonic range sensors.

For the navigation task of the project they decided to use Voronoi Fast Marching motion planner. They argued that this approach offers a thorough description of the exploration process, the ability to integrate with SLAM, and the potential to safely and effectively navigate with a limited field of view. In the SLAM part of the project they evaluated different existing approaches like GraphSLAM, EKFSlam, GridSLAM and FastSLAM. They resolved on choosing the latter, since FastSLAM is computationally efficient and likely to overcome inaccurate data making it the preferred algorithm for this project

Having made these decisions about the algorithms the research team designed the simulation part of the project. In the paper, they presented the results related to the mapping algorithm and the navigation one. They cited that they hadn't managed to construct the localization prototype due to the lack of time and the complexity, thus they couldn't present results of the FastSLAM approach. As expected, they did not present results from any real experiments.

Due to the lack of actual experimentation and the incomplete state of the simulation results this paper could be deemed a premature project Nevertheless, it is interesting to read since it assesses available methods for navigation and SLAM and also describes their experiences from the sensors' behaviour (in the calibration stage).

On the contrary to the previous study, a complete and interesting approach is presented in Hrabar's 2011 paper (Hrabar, 2011). In cases where UAVs performs real applications there is a need to move in cluttered environments, making collisions a real possibility. This paper comes to propose an interesting 3D reactive obstacle avoidance technique suitable for UAVs flight in point-to-point trajectories.

In the introduction the author presents us with common disadvantages in the existing approaches. He refers to techniques that check only current sensor's field of view without keeping memory and also refers to techniques that propose escape points without taking into account the goal point. Ensuingly, he counterpoises an approach where a cylindrical volume in front of the UAV checks for potential collisions and when one is detected an intermediate waypoint (Escape Point) is found in free-space that provides a route past the obstacles that have been mapped (fig.B.4). The Escape Point search proceeds on a series of concentric ellipses, the shape and orientation of which can be adjusted to favour horizontal or vertical avoidance manoeuvres (fig.B.5). Since a waypoint is produced, the technique is suitable for integration with global path planners that plan between waypoints.

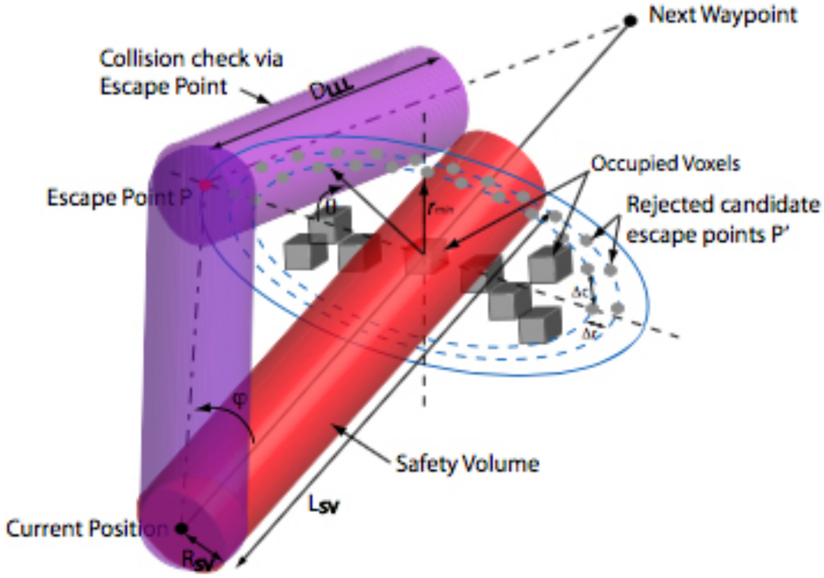

Fig.B.4 The expanding elliptical search for a valid Escape Point (image from Hrabar, 2011)

A voxel-based representation of the 3D environment has been employed to be the basis in which the algorithm is going to be operating. The algorithm maintains an occupancy grid in which both global path planning and reactive avoidance cooperate. In this map-based approach the reactive avoidance mechanism is easily tuneable to favour horizontal or vertical avoidance manoeuvres and generates reactions based on all the previously sensed obstacles, producing an intermediate waypoint which can be used as input for a waypoint-based global path planner.

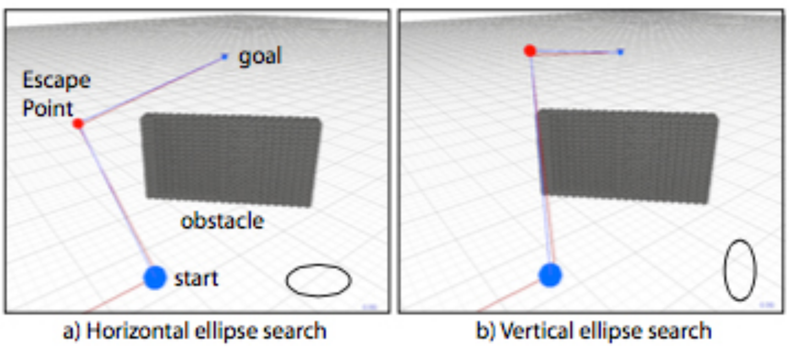

Fig.B.5 With the major axis of the ellipses aligned horizontally, horizontal manoeuvres will be favoured (a), while vertically aligned ellipses will produce vertical manoeuvres (b) (image from Hrabar, 2011)

Special care has been taken to enable the execute of the algorithm in the on-board hardware. The author cites that one of the most demanding tasks of the algorithm is the checking within the occupancy map for occupied voxels, required when testing for potential collisions ahead of the UAV and when evaluating candidate Escape Points. He used a balanced kd-tree (Bkd-tree) structure to represent the occupied voxel centres. Moreover, he compared the Bkd-tree based query times to a traditional brute-force linear search. For smaller voxel populations the techniques show similar performance, however as the voxel count grows the search time increases linearly for the linear search while remaining almost constant for the Bkd-tree search.

After an extensive and detailed explanation of the algorithm, the author presents the experimental results. The UAV was tasked to fly between pairs of waypoints either side of an obstacle, requiring it to detect and avoid the obstacle in order to reach the second waypoint. The results were very promising and the failures that occurred were due to obstacles not being detected and not due to the reactive avoidance algorithm.

The paper is well structured and presents its approach in great detail. Not only the theory of the approach is well defined but also the author presents a wide range of real experiments results. This is highly valued from a researchers standpoint since other papers I have read only include simulations and poor real world experiments, promising improvements in the near future. A possible limitation of this approach is the computational complexity. The UAV was equipped with (1.83GHz Core 2 Duo, 2GB RAM) system rather than popular Arduino based micro controllers (APM or even PX4) that usually be found. However, the progress in the field of SoCs (System on Chip) makes these kinds of tasks easier to be executed. It should be taken into consideration that the experiments were held successfully in 2011 with systems, which were available 4 years ago.

One more paper that presents a flight control system for autonomous UAVs is described in Wang 2014 (Wang et al., 2014). The design is a low-power, small-size autopilot and the research team performed various tests in order to verify its stability and reliability. The hardware used consists mainly of a microprocessor ATmega 2560, GPS, barometer and an MPU unit (gyroscope and accelerometer). In addition, it has a laser distance sensor to achieve a more accurate altitude for automatic take-off and landing. The autopilot system controls a small UAV, having wingspan of 3.8m and powered by a gasoline engine, rather than a multi-copter but the principles of control remain the same.

The design of controller of the autopilot is based on a cascade control method. The outer loop is a trajectory controller; the inner loop is an attitude controller. Among them, the role of the trajectory controller is to calculate the desired attitude and throttle, based on the desired trajectory and current UAV position and height. The attitude controller controls the size of aileron, elevator and rudder, in order to control the attitude of UAV consistent with the attitude calculated by the trajectory controller.

The trajectory controller contains two parts: a lateral controller and a longitudinal controller. Firstly, the lateral controller's role is to reduce, eliminate cross track error, making the UAV flight along the desired trajectory as far as possible. For this part, they used the L1 control method especially suited in the situation where the desired trajectory is a complex curved path. Secondly, the longitudinal controller tries to eliminate the altitude error. In the meantime, the airspeed of UAV remains within a safety range, avoiding under-speed. For this task, the researchers used the Total energy control system (TECS). The principle of this theory is that coordinating the throttle and the desired pitch angle of the UAV controls altitude and airspeed.

Further to the model, the paper includes the experimenting results. In general, the small-scale UAV flight presented an accuracy of trajectory control of about 1.5m and altitude error no more than 0.5m. The paper was coherent but it failed to adequately explain the control methods. However, it presented in great extent the hardware that was used but this did not add something to the novelty of the approach they were trying to offer.

B.3 UAV Stabilisation

One of the requirements for the UAS that I intended to design is the ability to fly low, slow and with augmented precision so as to measure accurate radiation values close to the source. Consequently, there is a need for enhanced stability during the flight. A typical fixed wing aircraft is inherently stable while in flight because the forward motion and airflow over and under the wings create stability. On the contrary, the task of keeping stable a helicopter during hovering requires constant minute corrections to throttle, pitch, roll, and yaw. Turbulences that would not trouble a fixed wing UAV can send a helicopter into a settling with power state or some other unrecoverable situation. The advantage of using multi-copters compared to the helicopters is that the former are simpler platforms since they can control all degrees of freedom by simply redistributing the thrust among the rotors. However, small UAVS flying in a low altitude need to respond very quickly to turbulences so as to avoid crashes.

Fig.B.6 Hovering flight above the landing pad (image from Herissé et al., 2012)

In (Herissé et al., 2012) the writers present an image-based visual control algorithm inspired from biological models like bees. Their purpose was to achieve UAV stabilization relative to the moving platform that maintains a constant offset from a moving reference and secondly automatic vertical landing onto a moving platform (fig.B.6). The image information that is considered is the average optical flow that is obtained from a textured target plane, using additional information delivered by an embedded IMU for de-rotation of the flow. The implementation is based on two nonlinear controllers, a proportional-integral (PI)-type controller designed for hovering flight, while the other is responsible for exploiting the vertical optical for the vertical landing.

In terms of hardware configuration, the system consists of an inertial measurement unit (IMU) and optical-flow measurements yet it additionally includes a digital signal processing (DSP) board, which runs at 150 MIPS and performs the control algorithm of the orientation dynamics and filtering computations. A spherical camera pointing directly down transmits video to a ground station via a wireless link. The data are processed by the ground station and incorporated into the control algorithm where the desired orientation and thrust are generated and sent back to the UAV. A challenging part for the implementation lies in the relatively large time latency between the inertial data and visual features.

The paper is oriented towards the capability of stabilizing the motion of the UAV with respect to a dynamic moving environment. A disadvantage of this approach is that the data cannot be processed on board in the UAV because of the algorithm's complexity. It can be also considered the requirement of contrast textures in order to gain better performance from the optical flow algorithm. On the contrary, it is easily inferred that the use of an optic flow sensor could enhance the stability in challenging situations and environments by giving more flexibility compared to the noise prone acoustic sensors of the range limitations of the infrared ones.

A more focused paper describing the benefits of optical sensors regarding the stability is presented by Fowers, 2007 (Fowers et al., 2007). It approaches the system by developing quad-rotor UAV using a low resolution (640x480) CMOS image sensor and a custom, low-power FPGA platform to perform computationally intensive vision processing tasks on-board, without the need for wireless tethers and computational support on ground stations. Deeper on the algorithms side, it uses Harris feature detection and template matching, giving the UAS the ability to be stable and almost drift-free hover.

The very small payload that the UAS can handle and the limitations in power consumption turns the use of the low power and small size FPGA (fig.B.7) a radical solution for the on-board calculations of the demanding image processing algorithms. The on-board vision system performs the drift-corrections that are not detected by the IMU, by tracking feature movement through consecutive images captured by the camera facing the ground.

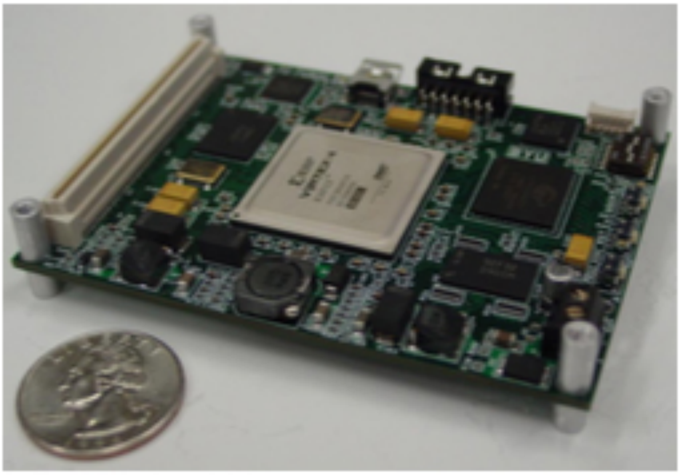

Fig.B.7 FPGA-based helios board (a Xilinx Virtex-4 FPGA) for embedded vision processing (image from Fowers et al., 2007)

Another approach using more advanced configuration is proposed in (Romero, Salazar and Lozano, 2009). This paper proposed an eight-rotor configuration whose translational and rotational displacements are decoupled. Four rotors perform the stabilization of the orientation copter, and the other four are used to drive the lateral displacements. The authors argue that the proposed configuration is useful for image processing since it keeps constant camera orientation. This is particularly important since systems that rely on optic flow for estimating range information need to remove components of optic flow that are induced by rotations of the aircraft, and use only those components that are generated by the translational component of motion. This happens because only the translational components of optic flow provide information on the range to objects in the environment. However, the experiments exhibited that the estimation of translational speed in outdoor environments using optic flow has several limitations, such as the light intensity and the contrast. In indoor environments this strategy seemed to have worked better. Nevertheless, the design of this configuration (fig.B.8) is custom and cannot be applied in commercial Off-The-Shelf (COTS) products without major modifications.

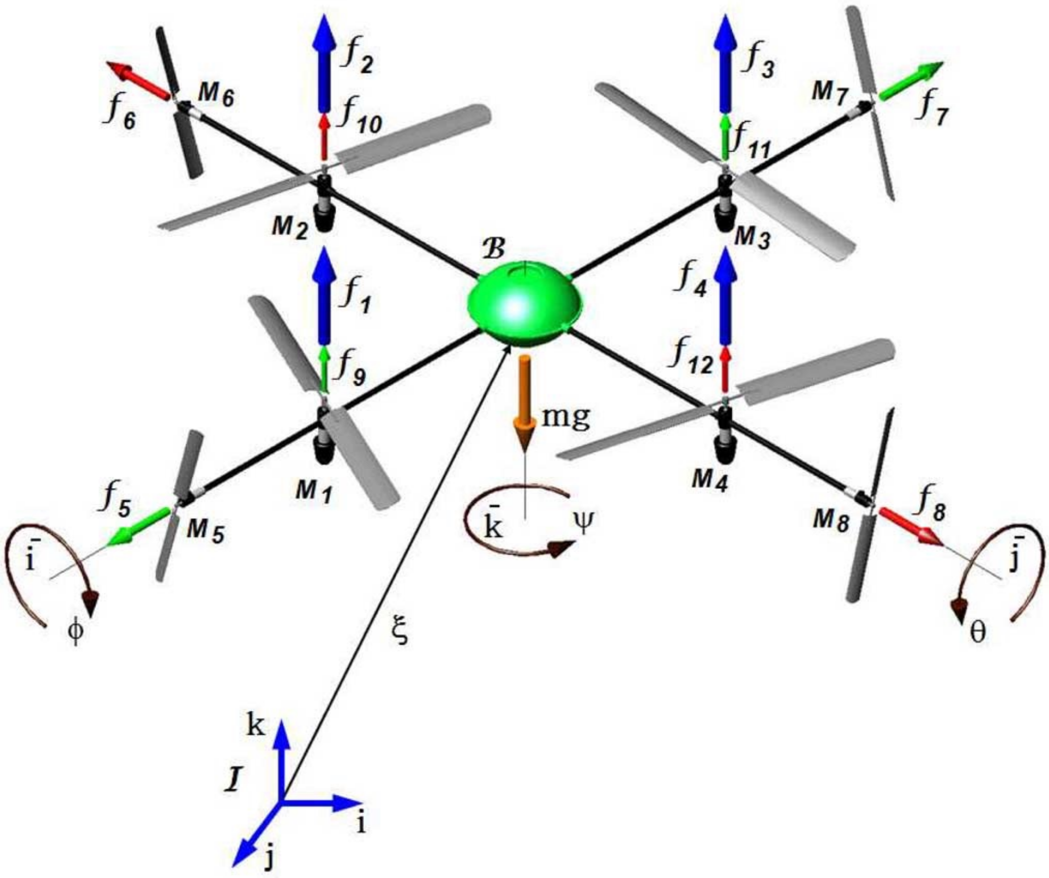

Fig.B.8 Eight-rotor rotorcraft scheme (image from Romero, Salazar and Lozano, 2009)

B.4 UAV sensors

Having reviewed interesting approaches related to UAV collision avoidance and stabilisation I realize that the availability of reliable and flexible sensors will facilitate the task. My project is based on the use of COTS technologies and the research of existing distinguishable hardware is required. In the field of mobile robot navigation there is always the need for an accurate, low power, compact and ideally low cost distance measurement sensor. Pulsedlight (Technology Brief, 2015) has managed to implement a very interesting design (fig.B.9). It measures distances from zero to 40m with +/- 0.025 m accuracy at a reading rate of 100Hz. According to the developers the existing optical sensors solutions was either high accuracy phase-based measurement based on the transmission and reception of a continuously modulated, typically red, pointing and measurement beam or long range low accuracy pulsed measurement using the delay between the transmission and reception of higher power, short duration, infrared pulses. The innovation in this approach is that they implement a device based of the latter, time-of-flight distance measurement method, but using low power and low cost signal digitization hardware unlike the existing solutions, usually military applications. In order to achieve that goal they had to develop a new signal processing approach to optical distance measurement, known as signal correlation technique, which still allowed implementation using a very simple hardware configuration while eliminating the time consuming and processing intensive signal reconstruction of existing approaches. The implementation of this design is released as a single programmable logic chip or System-on-Chip (SoC).

In the stabilisation section, the optical flow measurement seems to play a dominant role in this task. PX4 project, which is an independent, open-source, open-hardware community, has developed a remarkable optical flow measurement device (Honegger et al., 2013). They present low consumption, lightweight and low cost SoC (fig.B.9) that can compute optical flow and compensate for rotation. It is suitable for indoor and outdoor applications having very high light sensitivity. The software implementation and hardware design is open sourced. The system consists of an ARM Cortex M4 microcontroller that performs optical flow processing at 250 frames per second at a subsampled resolution of 64x64 pixels using a CMOS machine vision sensor. An ultrasonic range sensor is also installed to measure the distance towards the scene and to scale optical flow values to metric velocity values. Up until the presentation of this implementation it was a more popular approach to use computer mouse hardware sensors to carry out similar tasks but they suffer from lighting limitations since they require strong lighting to provide accurate measurements. Other systems using more advanced CMOS sensors send the data via wireless links to ground stations for processing or employ dedicated hardware designs implemented in field programmable gate arrays (FPGA) to perform all computations on-board in real-time. The evaluation experiments were conducted in indoor and outdoor environments, with different illuminations. The system is validated with several experiments comparing the presented flow sensor to a standard mouse sensor and GPS and measurements with ground truth using a VICON motion tracking system. The results demonstrated that the sensor operates accurately at high frame rates.

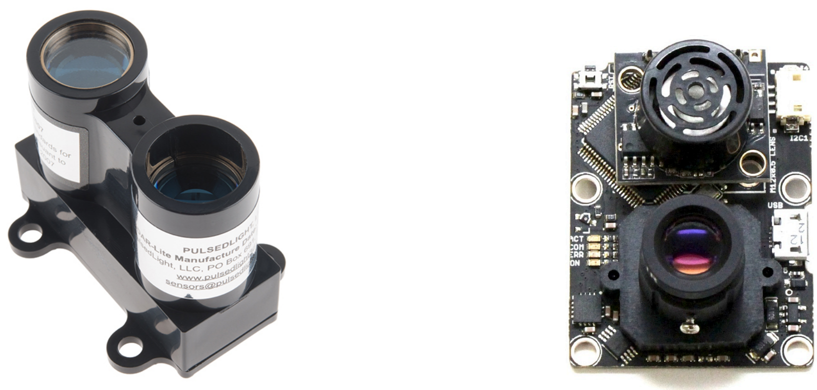

Fig.B.9 PulsedLight's Lidar-Lite laser distance measurement device and PX4Flow optical flow smart camera (images from http://pulsedlight3d.com and https://pixhawk.org/modules/px4flow)

B.5 UAV Simulations

In recent years, computer simulation of robotic concepts has become popular since many tools have been introduced to facilitate the designer's work. The advantages of doing a simulation (Richards, 2015) is that it is cheaper and less risky to experiment using computer models rather than real hardware. Regarding the UAVs, there is a plethora of available simulators each of them targeting a specific task. The problem with these is that they complicate the evaluation of the designed tasks and have steep learning curves. On the other hand, there is always the option to develop a custom simulator only for the given task that the designer requires. The downsides of this approach are the amount of time is spent building things from scratch as well as the absence of some advanced features that the simulators have, such as visualization, readymade environments (indoor and outdoor) and advanced dynamics.

For the specific project I need a simulator that can be used in different environments, to assess the efficiency of the collision avoidance algorithms by employing different proximity sensor technologies as well as to prove the stability of the configuration. For these tasks, I need to be able to plan a path, to build different environments and finite states machines, which add the required intelligence to the UAVs. Most importantly, I need to change the UAV model and add additional sensors in specific parts of the UAV. This set of requirements makes the use of available simulators difficult for my task. I reviewed some of them in order to assess their suitability.

The Robot Operating System (ROS) is a framework for writing robot software. More precisely (Ros.org, 2015), it is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. Gazebo is a robot simulation tool that is used in combination with ROS to test algorithms, robot designs and perform regression testing using realistic scenarios.

The usage of these tools is very popular in the robotics community and it is a standard in many robotics fields. For the quad-rotor UAV systems case there is a published package related to modelling, control and simulation called hector_quadrotor (Meyer et al., 2012). This approach allows simultaneous simulation of diverse aspects such as flight dynamics, on-board sensors like IMUs, external imaging sensors and complex environments. In order to exploit them, the user needs to have experience in C++, preferably, or python. In addition they need to be familiar with working in Linux environment and server – client programming. Like many open-source projects, support is provided mainly from the support forums. There are several tutorials, but in my opinion these cover a small and low-level area of required knowledge. In addition to that, there is a wiki but one needs to be an experience user to take advantage of it. One drawback that I also discovered is the existence of many distributions (distros) and the complexity that this adds to the available third-party libraries and packages. On the bottom line ROS and gazebo is a very powerful (fig.B.10), professional and popular tool yet it needs lot of time to become a user-friendly tool and start producing useful results. It produces a difficult learning curve if the user comes from different background.

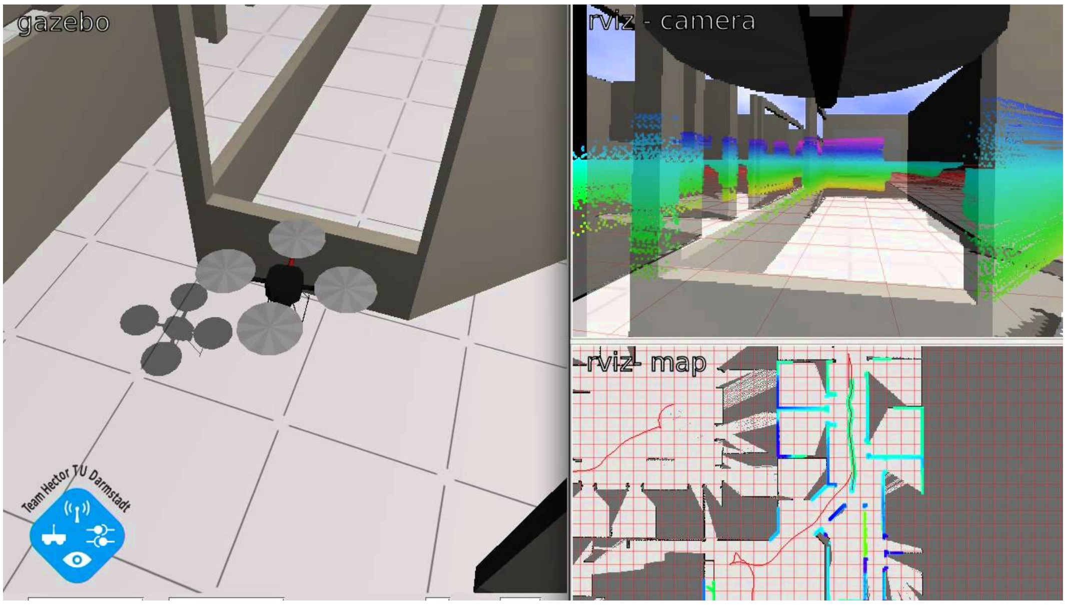

Fig.B.10 Screenshot of an indoor quad-rotor UAV simulation using ROS with rviz and gazebo (image from Meyer et al., 2012)

Very recently (available in Matlab R2015a), complementary to the previous approach, is the release of a new robotics system toolbox trying to resolve some of the weaknesses and bridge some gaps I referred to previously. The system toolbox (Uk.mathworks.com, 2015) provides an interface between MATLAB and Simulink and the ROS that enables the testing and verifications of applications on ROS-enabled robots and Gazebo. It also supports C++ code generation, enabling the generation of a ROS node from a Simulink model and deploy it to a ROS network. It is worth mentioning that if the users need to alter and build custom models they need again to cope with ROS architecture and code. Consequently, I admit that this approach is friendlier than the stiff approach of plain ROS but it needs to have knowledge of this in order to be flexible and creative.

Another package that I tried was a 3D Integrated Simulation Platform for Indoor Quad-rotor Applications provided by (A.R. Al-Omari, A. Jaradat and Jarrah, 2013). It is built entirely in Matlab without the use of Simulink. It offers impressive 3D views and a nice graphical user interface (fig.B.11). On the other hand, it lacks serious documentation and the code is not well commented. There are no instructions on how to edit it, to add sensors or construct new maps. Unfortunately, in the current state it cannot serve its initial purpose, as stated, which was the ability to implement and inspect algorithms in different scenarios.

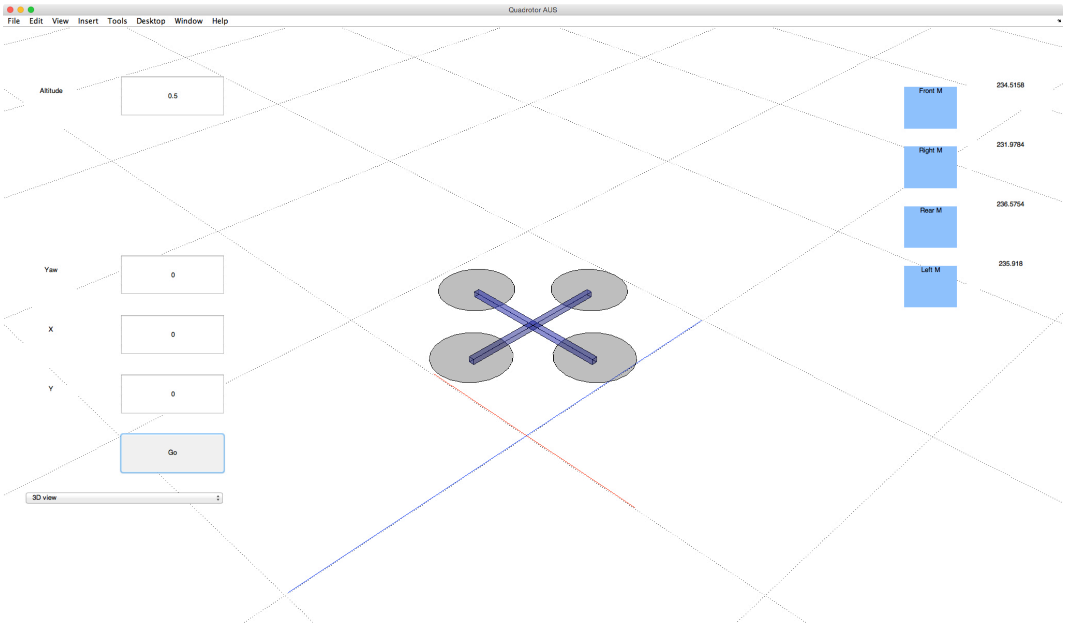

Fig.B.11 Screenshot of the Integrated Simulation Platform for Indoor Quad-rotor Applications (image from A.R. Al-Omari, A. Jaradat and Jarrah, 2013)

A better approach in UAVs simulations is presented in Hartman, 2014 (Hartman et al., 2014). They presented a quite detailed quad-copter dynamic modelling and Simulation package accompanied with extended documentation and instructions. The researchers have built the simulation making extensive use of Matlab and Simulink. Using this tool the designer can define the exact geometry of the dominant parts of the UAV like motors, electronic speed controllers, central hub and arms (fig.B.12) and simulate scenarios with different initial conditions and paths. The documentation does not include instructions on extending the model with various sensors nor does it include the ability to build custom maps with obstacles and to test different behaviours. It is an admirable attempt but, in my opinion, it is intended to different testing scenarios where the designers experiment with the geometry and the configuration of a UAV rather than have a given UAV and test behaviours in different stimuli.

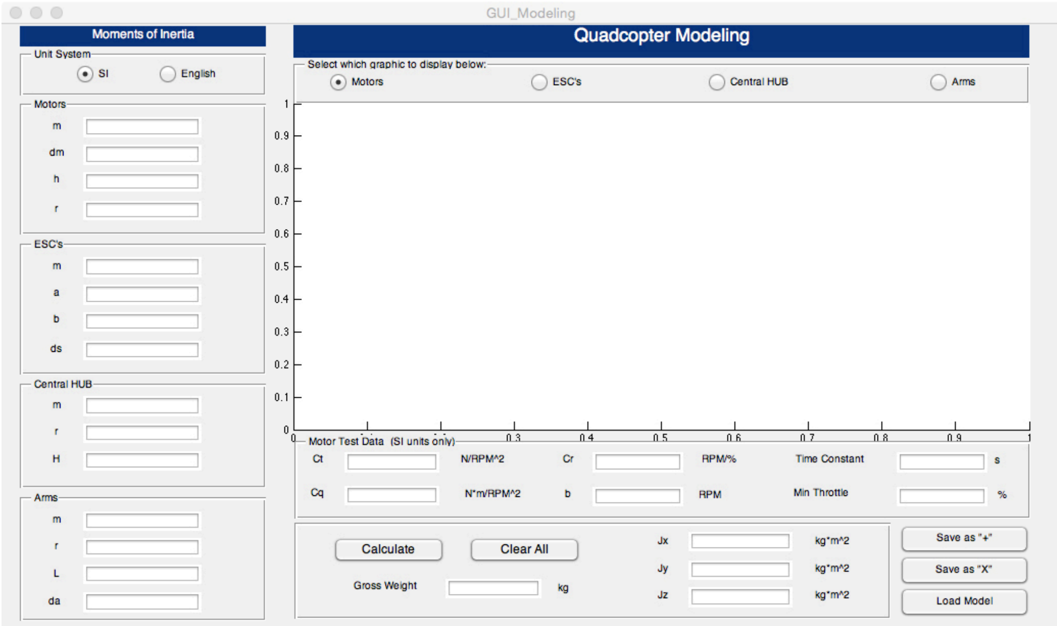

Fig.B.12 Modelling graphical user interface of quad-copter simulation using (image from Hartman et al., 2014)