

High Level Design of the Flight Control Software for Small VTOL Unmanned Aerial Vehicle

ÁrVAI László¹

The implementation of an on-board flight control system must be preceded by extensive design activity. The software implementation, test and maintenance effort might be significantly reduced by using carefully designed software. The design procedure itself is also a complex activity, several requirements, constraints must be considered in order to generate an appropriate, well detailed and easily implementable design. The design task is even more complicated when the aircraft has some special features like the capability of vertical take-off and landing. In this case the software must be able to handle the diversity of flight modes where the aircraft behaviour in each mode is unique. This article presents the main aspects of the design procedure.

Introduction

The on-board flight control software of any unmanned aerial vehicle must deal with several very complex tasks. Stability, navigation, communication, payload control are just some of the main tasks and integrating all these in on-board software is directly responsible for the safe, reliable, optimal and smooth operation of the aircraft. The software must accomplish its tasks in real-time with the shortest possible response time, using only limited resources.

The implementation work of an on-board flight control system must be preceded by careful and detailed design activity. The requirements must be collected, defined and analyzed, the constraints must be recognized and after making several decisions and compromises the final high level software design can be obtained. The most important factors in software design include configuration, dimensions and geometry of the aircraft.

Aircraft configuration

The aircraft presented here is a small size, fixed wing, canard configuration airplane with tilted engines. The wingspan is less than two meters, the maximum take-off weight is about one and half kilograms.

The engines are ducted fans, located on both sides of the fuselage, assembled on small rotatable sponsons. The ducted fans are driven by brushless electric motors; the energy comes from a Li-Po battery. The computer generated image of the aircraft is shown in Figure 1.

The aircraft has no gears since STOL² and conventional take-off and landing operations are not desired, because the appropriate implementation of these modes will make the control software even more complicated. However, these features might be added later, because they can be very valuable in the case of an emergency landing, when the engine power is completely lost, then the aircraft still can glide and land conventionally. But in the first version only a small landing skid will be used.

¹ National university of Public Service, Budapest, Hungary

² STOL – Short Take-Off and Landing

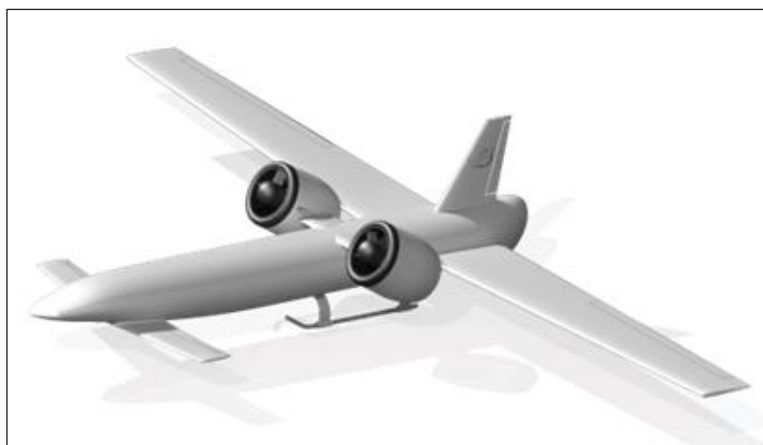


Figure 1. The Cygnus UAV (computer generated image)

Árvai László: High Level Design of the Flight Control Software for Small vTOL Unmanned aerial vehicle position to slightly over vertical position. The picture also shows all conventional control surfaces (ailerons, rudder, and elevators). Because of the lack of conventional landing capability, other surfaces (flaps, speed breaks, etc.) are not required. The control surfaces will be used in horizontal flight mode (when wings produce total lift). In hovering mode (when ducted fans produce the lifting force) vectoring controls will be used.

The only planned payload at the moment is an on-board camera, sending its image wirelessly to the ground station, so obviously the only possible mission for this uAV is an image reconnaissance mission.

Flight profile

The flight of the UAV is composed of several phases. Each phase has different or at least slightly different requirements for the flight control. A typical UAV mission flight profile is shown in Figure 2. The profile picture is based on a flight profile of a fixed wing UAV³ with an adaptation for VTOL uAV.

The UAV follows a predefined flight path, where each waypoint is defined by the position (latitude and longitude), altitude and speed. The waypoint can be defined in the ground station software and uploaded to the on-board flight control electronics via wireless link. After uploading waypoint data the uAV is capable of following its designed path autonomously.

The flight starts with a vertical take-off (cf. 1 in Figure 2) and continues with the transition to horizontal flight (1-2). Even the airframe enables a conventional or short take-off, though it is not planned at the moment. In this mode the main task of the flight control software is maintaining the stability of the aircraft, adjusting the engine RPM to achieve the required lifting force and vertical speed, controlling the transition phase between vertical and horizontal flight until a safe forward flight condition is reached (all lifting force is generated by the wings).

The flight continues with horizontal flight (3-5) where the most important tasks (besides the attitude stability) are the following: regulating the speed, altitude and heading in order to follow the predefined flight path.

3 KurNAZ-CETIN-KAYNAK (2009)

The next phase is either the loiter or hover phase depending on the type of reconnaissance task. The flight control requirements in hovering mode are similar to those in vertical take-off phase and the requirements in loiter mode are the same as in horizontal flight mode. Phases (3-9) might be repeated several times in the case of multiple target areas.

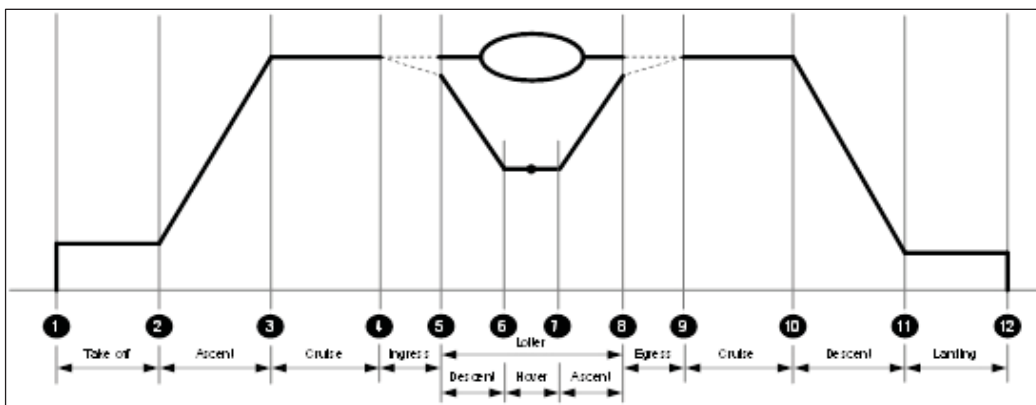


Figure 2. Typical mission flight profile

At the end of a flight similar phases follow each other like the ones at the beginning, but in reverse order. The requirements are basically also the same. There is only one exception, since to obtain completely autonomous landing the clearance of the landing area must be verified and continuously monitored. However, this also makes the flight control system more complicated, therefore this feature will be omitted and even the landing is done automatically, human supervision is required for the sake of safe operation.

In the case of completely autonomous operation, especially in urban regions, collision free path planning is essential.⁴ The cruise/loiter phase of the flight path can be further optimized using soaring/thermals.⁵ Szabolcsi derived flight paths geometry for different flight phases,⁶ and derived requirements for a given flight profile. However, application of the very high level of autonomous operation mentioned earlier in the article is out of the scope of the current research activity. The main goal is to achieve flight stability in VTOL operation and the stable transition from/to horizontal flight autonomously.

Flight control laws

Besides handling each phase of the mission profile, the flight control system must be able to choose the appropriate flight mode (mission goal) as well. There are three modes (or laws) summarized in Table 1.

- 4 NARAYAN (2008)
- 5 ALLEN (2007)
- 6 SZABOLCSI (2010a), SZABOLCSI (2010b)

<i>Mode</i>	<i>Primary control target</i>	<i>Mode selection criteria</i>
Normal law	maintain aircraft stability and follow predefined flight path	This is the default flight mode
Manual law	no stability and flight path control, only fly by wire control	Initiated by the ground crew
Alternate law	maintain aircraft stability, fly to home, or crash land the aircraft	Initiated by the detection of on-board serious failure

Table 1. Flight control laws

Normal law

The normal law is the default flight control law. A manual or alternate flight control law can override this mode but the system can only return to this mode from manual law. In normal flight mode attitude stabilization and navigation are done automatically based on the predefined mission profile. The block schema of this mode is shown in Figure 3.

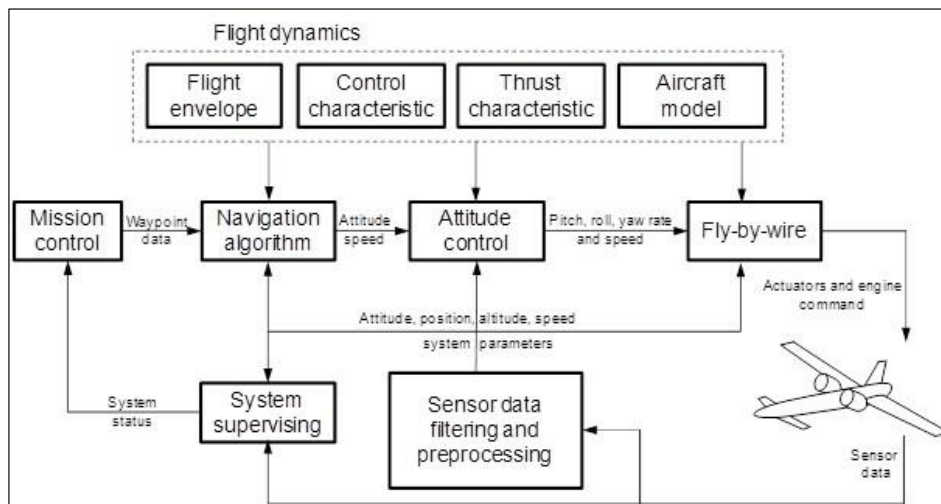


Figure 3. Blockschematics of normal flight control mode

The mission control module defines the actual flight goal based on the preloaded mission flight path and the actual status of the aircraft. It gives the next waypoint information to the navigation algorithm.

The navigation algorithm designs the flight path based on the current position and the next desired waypoint information of the aircraft. It sends attitude and speed information to the attitude control module.

The attitude control stabilizes the aircraft in any flight mode, and from the target attitude and speed information it calculates the necessary pitch, roll, yaw rate (acceleration) and speed information and passes it to the fly-by-wire system.

The fly-by-wire system translates the desired pitch, roll, yaw rate and speed request into the actuator and engine control values.

The feedback loop starts with the sensors located on the aircraft. The sensor data is validated,

scaled, filtered, pre-processed and combined by sensor data filtering and the pre-processing module. The module uses several sensor's data (gyro, acceleration, magnetic field, differential (Pitot tube) and absolute (altitude) air pressure, GPS and system monitoring sensors such as current, voltage, temperature, etc.) in order to generate the attitude, speed, altitude, position and system status information for other modules.

The system supervisor module is continuously monitoring the whole system, analyzing the dataflow between modules. If it detects serious anomalies it analyses the seriousness of the failure and makes its suggestions to the mission control module.

There are additional modules providing information about the flight dynamic of the aircraft. The flight envelope information is used by control modules to stay within the safe aerodynamic conditions. The control and thrust characteristic

is required for the fly-by-wire system to calculate actuator and engine control parameters for the desired attitude and speed changing. The aircraft model is also used by the fly-by-wire system in order to handle various flight conditions (horizontal, hovering, transition), i.e. based on the aircraft behaviour the control system can decide which actuator type (aerodynamic, vectoring) should be used in the current situation.

Manual law

The manual flight control law is initiated by the human operator from the ground station. The human operator is allowed to switch back to normal mode or the system will switch to normal mode when the radio link used for control breaks. In this mode the control of the aircraft can be taken over by a human operator. The flight control system will neither stabilize the aircraft, nor follow the waypoints. However, it helps the human operator with a fly-by-wire mode and flight envelope protection. The block schema of this mode is shown in Figure 4.

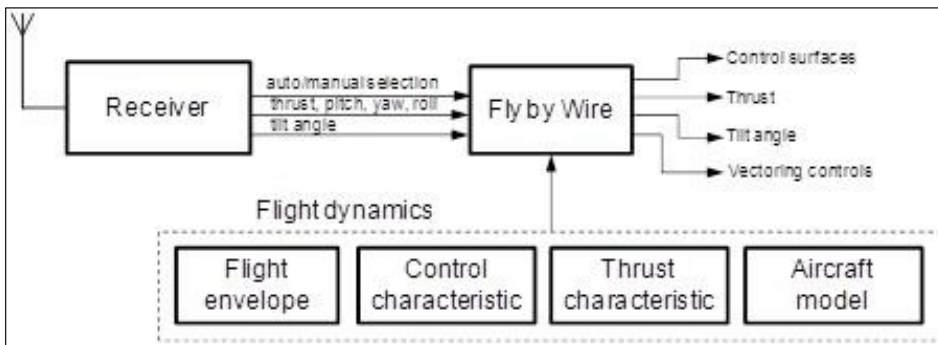


Figure 4. Block schematic of manual flight control mode

The command from a human operator is received via either the telemetry/command communication channel or using a separated 'backup' receiver. Commands issued by the operator are the same as usual aircraft controlling commands, they command the throttle (thrust), pitch, yaw and roll. There is only one special command - because of the VTOL operation- the tilt angle of the engines, i.e. the ratio of the expected lifting force generated by wings or engine thrust. The real tilt angle is determined by the fly-by-wire system.

The command from the ground operator is processed by a fly-by-wire system. There are two important reasons for applying this system. On the one hand, controlling any aircraft from the ground, without

having the real sensation of flight is more difficult, it is relatively easy to break the flight envelope of the aircraft and loose control of it. With an active flight envelope protection this can be avoided.

On the other hand, the VTOL aircraft has different behaviour in different flight modes (horizontal vs. hovering) and these behaviours are blended in the transition stage, however, the operator expects only one controlling method. For example in the normal (horizontal) flight mode the yaw rate is controlled by the aerodynamic force of the vertical stabilizer and rudder. However, in hovering mode, when the airspeed is close to zero all aerodynamic surfaces are useless. In this mode vectoring controls are necessary, i.e. differentially tilting the engines will give the appropriate yaw moment. So the fly-by-wire system needs to convert any rudder (yaw) input to differential tilt angle. Moreover, tilting the engines will reduce the vertical component of the engine thrust (i.e. the lifting force) so it also should be compensated for by increasing the engine power (RPM). So the fly-by-wire system receives only the intention of the operator and decides automatically the best actuator settings for the required operation. To achieve this goal the fly-by-wire system needs to know the actuators/controls characteristics, engine characteristics and the aircraft characteristics, all these should be determined by computer simulation and actual test measurements.

Based on this information the fly-by-wire system can control all aerodynamic surfaces, vectoring controls, engine thrust and engine tilt angle according to the request from the human operator.

Alternate law

Alternate flight control law is initiated by the system when it detects a serious failure. In this mode the aircraft abandons the mission goals and tries to land safely. The flight control system still tries to stabilize the aircraft attitude, but instead of following the flight plan it switches to

'go home' mode, i.e. it goes back to the starting position using the shortest possible route. If the aircraft attitude cannot be controlled because of a failure, or returning to the starting point is not possible because of the lack of energy, it tries to crash land the aircraft in order to minimize the loss.

The alternate law can be overridden by the manual mode, but normal mode will not be enabled until the system is restarted.

Operation system

The flight control system is definitely a real-time system, because response time constraints are very strict, the control must be able to respond to any aerodynamic disturbances quickly in order to maintain stable flight. Thus, the design procedure must start with the assumption of using real-time software architecture. The most plausible starting point of the real-time system design is choosing a real-time embedded operation system. The embedded operation system plays an important role in the complete system, the reliability, response time, complexity and power consumption of the flight control system depends very much on the choice of the embedded operation system. However, in the case of small size uAV, when the computing power and the complexity of the embedded processor is limited, the choice of the operation system is not so straightforward. Some operation systems are able to run on hardware with limited resources.⁷

The procedure of selecting them is also explained there. However, there is another possibility.

7 MARIONI et al. (2006)

Once we have decided to design and build a modular system⁸ (where the module borders are defined among functionality and the module is composed of software and hardware together) we might avoid using real time operation systems. When a module has one or only a few well defined tasks a pre-emptive task scheduler, extended with interrupt driven real time drivers, might do the job as well. The interrupt based drivers give quick responses to any external signal while the microcontroller has only a few well defined tasks. The pre-emptive task schedules give a reasonable response time without the expense (resource) and complexity of the real time operation systems.

The main modules are listed in Table 2.

<i>Module</i>	<i>Software Function</i>
Main board	System supervising, running flight control tasks
IMu	Complete inertial system, filtering and pre-processing sensor signal
GPS	Pre-processing GPS information
Communication	Communication with the base station, on-board data logging
Engine control	Controlling the engine power and monitoring the engine condition

Table 2. Primary modules

The main board as hardware is responsible for the complete interfacing of the modules (communication, power supply) and as a software module it runs the system supervising and primary flight control tasks. It has direct connection to the actuators and it has most of the system monitoring sensors.

The flight control related sensors are located on other modules, the most important one is the IMu module. This module contains the complete inertial system with three axis gyro, acceleration, magnetic sensors and the absolute air pressure sensor (altitude, vertical speed). A differential air pressure sensor (airspeed) can also be connected. It runs the appropriate sensor data processing and filtering algorithms in order to produce the aircraft attitude, altitude, and speed, etc. data.

The other important sensor module is the GPS module. It receives GPS data pre-processes, and parses these and sends them to the main board.

The communication module is responsible for maintaining the wireless link with the base station. The link is used for sending telemetry data and receiving commands. It also logs all telemetry data on-board for off-line analysis and visualization.

The engine control module controls the engine thrust. It drives the brushless electric motors and monitors its RPM and current (power) consumption. The power consumption monitoring is important for the estimation of the remaining energy stored in the battery.

The primary modules together are able to fly the aircraft but adding other modules might be necessary. The auxiliary modules are listed in Table 3.

8 ÁRVAI (2012)

<i>Module</i>	<i>Software Function</i>
Backup communication	Backup communication channel, lower bandwidth than a normal communication channel and mainly used for manual control of the aircraft

AARMS (12)1 (2013) 98

Payload	Controlling the payload, monitoring its functionality and power consumption
---------	---

Table 3. Auxiliary modules

There are several possible auxiliary modules, but only two of them are listed in the table. The backup communication module is mainly used for manual control of the aircraft, usually it has lower bandwidth. However, it also may be the case that it has no telemetry data sending capability or its capability is quite limited, so it can be used only for line-of-sight operation of the aircraft.

The aircraft usually has a payload, in a simple case it might be only a video camera and transmission system, but it also can be complete SIGINT/MASINT equipment. In any case the payload needs power and control. usually the payload has a separate channel for communicating with the ground station.

Software structure

Based on the description given above, the high level software structure can be designed. At this level the software modularity is not shown, because from the designer points of view the module division and the intermodule communication are not so important, however, a complete overview of the necessary functions is required. The complete system overview is given in Figure 5.

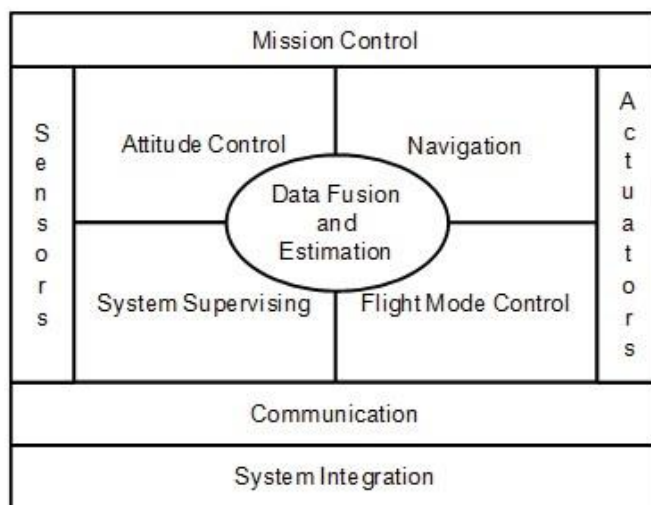


Figure 5. Software structure

At the top of the software structure the mission control system is located. This module has the highest overview of the system and it makes the highest level of decision. It continuously monitors the aircraft condition and determines the current goal for the aircraft and passes this information for all other subsystems. In normal flight control modes it follows the predefined mission profile, so the aircraft visits the predefined waypoints and executes its planned task there. In the case of

an emergency it makes a decision about the safety action to be made, cancelling the mission and bringing the aircraft home or crash landing it in order to minimize the damage.

The sensor subsystem is responsible for the data processing of all sensor data. The processing starts with the data validation. The stability of the communication channel of the sensor (either it is analogue or digital) is continuously monitored, the sensor data compared with the desired range of the sensor. If sensor communication seems to be broken or data seems to be out of range, this piece of information is passed on to the system supervisor module. When the sensor data is valid it is further processed, cleaned, filtered or even combined (data fusion). For example, in the case of inertial sensor data, the gyros, accelerations, magnetic field sensors data or even the GPS data are combined in order to produce Euler angles or Direction Cosine Matrix.⁹

The attitude control module is responsible for active stability, aircraft directional and speed control. It has all the necessary control loops for providing stability about longitudinal, lateral and vertical axis and controlling the desired angular speed about these axes.

The navigation module is responsible for the flight trajectory planning and optimizing, i.e. to reach all waypoints with minimal energy consumption.

The flight mode control basically belongs to the mission control, however, because of its importance it was separated in the picture. It decides which flight control law should be used and on normal law mode it selects the appropriate phase of the flight profile.

The system supervising module is continuously monitoring the system health. In the case of any significant failure it tries to identify the problem and classifies it based on their danger. For recognizing the failure it uses the information from the

Árvai László: High Level Design of the Flight Control Software for Small vTOL Unmanned aerial vehicle sensor driver about the reliability condition of the given sensor but it also compares sensor values with the expected sensor values based on actuator settings and aircraft model. In the case of less serious problems it tries to adapt the system.¹⁰

The actuator control module is responsible for translating desired angular speed about the aircraft axes into the control surface or vectoring control deflection. In manual flight control law this module serves as a flight-by-wire system.

The data fusion and estimator module works strongly together with the system supervising module. It continuously monitors the sensor and actuator data flow. Any significant difference between the measured (sensor) data and the expected behaviour of the aircraft (actuator data) can have only two reasons. One is the sensor/actuator/system failure, which is handled by the system supervising module. The other reason can be some external effect which is not measured directly by sensors. For example, if the aircraft speed vector measured by the inertial system is not coincident with the speed vector measured by the GPS it might show the existence of a crosswind. The direction and speed of the crosswind can be calculated comparing the two kinds of information.¹¹

The communication subsystem is responsible for the telemetry data sent to the base station and it also receives commands from it. The communication must be reliable and secure, both requirements must be handled by high importance when the communication protocol and software are designed. The on-board data logging is also handled by the communication module (however, it could be handled somewhere else), since the communication module has all system data for telemetry data sending.

The system integration part of the software is responsible for handling all on-board devices

9 GEORGE, Michael – SUKKARIEH, Salah: Tightly Coupled INS/GPS with Bias Estimation for uAV Applications. <http://prism2.mem.drexel.edu/~vefa/research/HeliExtLoadStabil/relevantPapers/imu/george.pdf> (01.06.2012.)
PEREZ-D'ARPINO et al. (2011)

10 IPPOLITO – YEH (2007)

11 SIR (2003)

which are not handled by the actuators module. One such task is power management (energy consumption measurement, overload monitoring and handling) and payload control.

Conclusions

Creating on-board flight control software for any UAV is a very complex task. The requirements are diversified and the available resources are very limited and the complete system must be a real time system with a very robust construction. Therefore precise and extensive design work should be done first. In the high level design phase the aircraft purpose, characteristics, typical mission profile must be defined, and special requirements should be collected. Next the control laws should be defined and necessary data flow schema of each mode should be drawn.

The best solution would be to use real-time operation systems, but sometimes the limited resources or the complexity of such a system makes it impractical to use. In such cases a modular design (distributing tasks among several processors) might provide a good solution.

Finally, the high level software structure can be created. The result of high level system design provides all major software subsystem's description, explains their main functionality and defines dataflow between them.

All information created in the high level design procedure will serve as basis for lower level software design.

References

1. KURNAZ, Sefer – CETIN, Omer – KAYNAK, Okyay (2009): "Fuzzy Logic Based Approach to Design of Flight Control and Navigation Tasks for Autonomous unmanned Aerial Vehicles" In: *J Intell Robot Syst*, p. 229-244.
2. NARAYAN, Pritesh (2008): "unmanning uAVs – Addressing Challenges in On-Board Planning and Decision Making" First International Conference on Humans Operating unmanned.
3. ALLEN, Michael J. (2007): "Guidance and Control of an Autonomous Soaring uAV" NASA, Dryden Flight Research Center.
4. SZABOLCSI róbert (2010a): "uAV Flight Path Conceptual Design" In: *Proceedings of the 16th International Conference "The Knowledge-Based Organization – Applied Technical Sciences and Advanced Military Technology"*, pp. 519-524, November 2010.
5. SZABOLCSI róbert (2010b): "Worst Case Flight Scenario for unmanned Aerial Vehicle in D3-Missions" In: *Proceedings of the 12th Mini Conference on Vehicle System Dynamics, Identification and Anomalies*, pp. 351-361, November 2010.
6. MARIONI, Mauro – FACCHINETTI, Tullio – BUTTAZZO, Giorgio – FRANCHINO, Gianluca (2006): "An Embedded real-Time System for Autonomous Flight Control" *ANIPLA*.
7. ÁRVAI László (2012): "Design of the on-board flight control electronics of an unmanned aerial vehicle" In: *Tavaszi szél konferenciakötet*, Budapest. pp. 366-372.
8. GEORGE, Michael – SUKKARIEH, Salah: Tightly Coupled INS/GPS with Bias Estimation for uAV Applications. <http://prism2.mem.drexel.edu/~vefa/research/HeliExtLoadStabil/relevantPapers/imu/george.pdf> (01.06.2012.)
9. PEREZ-D'ARPINO, Claudia – VIGOURoux, Damian – MEDINA-MELENDÉZ, Wilfredis – FERMIÑ, AARMS (2011)

- Leonardo (2011): "Development of a Low Cost Inertial Measurement unit for uAV" *IEEE International Conference on Process Automation, Control and Computing*, pp. 178-183, 2011.
10. IPPOLITO, Corey – yEH, yoo-Hsiu (2007): "Neural Adaptive Flight Control Testing on an un-manned Experimental Aerial Vehicle" *AIAA Infotech@Aerospace*. <https://doi.org/10.2514/6.2007-2827>
 11. Sír, Cristián (2003): *real-time Wind Estimation and Display for Chem/Bio Attack response using uAVData*, Master thesis.