Autonomous Multirotor UAV Flight Control System based on GPS and INS data fusion

*Kirill Shilov**
**Moscow Institute of Physics and Technology (MIPT)*
16 Gagarina st, Zhukovsky, Moscow reg., Russia 140180

Abstract
The present paper contains the main ideas and concepts to be guided in the development of multirotor UAV flight control system. Generally, having a compact size, control systems include MEMS sensors (accelerometers, gyroscopes, barometer, magnetometer) which measurement accuracy is not so high. The main navigation sensor commonly applied is GPS. The precision of GPS is several times less than a multirotor size. Multi sensor data fusion methods should be applied for increasing the measurement accuracy. Particularly, the methods and results of barometer and accelerometer fusion for altitude estimation and GPS and INS fusion for position estimation are presented.

1. Introduction

Recently Unmanned Aerial Vehicles (UAVs) being a rapidly developing field have been attracting more and more attention of scientists and costumers. Particular interest is paid to Multirotor UAVs, which are known to be good flying platforms for high-quality aerial photography, videography, monitoring and other terrain explorations due to their ability of low-speed flight, hovering and vertical take off and landing. Described characteristics make them easy to use in space limited conditions. Obviously, the behavior of such vehicles is unstable, therefore, Flight Control System (FCS) responsible for both stabilization and navigation functions is required. Additionally, FCS is able to provide the capability of fully autonomous flight. Rapid development of contemporary electronics makes possible to create low-cost and compact FCS. However, accuracy of implemented measurement units is not high. Multi-sensor data fusion is one of the methods for its increasing. Present paper describes the requirements and general concepts to be guided in the development of FCS as well as results obtained in flight tests and their comparison. Special attention is paid to multi-sensor data fusion methods, which allow increasing flight precision and reliability. Moreover, the description of hardware and software architectures is provided.

2. Hardware

2.1 General concepts and approach

Autonomous control of multirotor UAV can be achieved with the ability of maintaining desired orientation and position in space, therefore, FCS can be subdivided into three global parts:
1. Information acquisition from the sensors (accelerations, angular velocities, GPS position, atmospheric pressure, orientation relatively to the Earth magnetic field, ambient temperature)
2. Obtained information is exposed to verification, filtering, mutual fusion and compensation in order to improve accuracy. Having the information about current state, FCS applies flight control algorithms and calculates control actions required for maintaining the desired orientation
3. Actuators (electronic speed controllers) control the rotation speed of each brushless DC motor.

The first and the second parts are of most interest. The first part assumes the development of the software for microcontroller and sensors inter-communication. This part is often called low-level software. The second part consists of flight control logic algorithms and designed to be hardware independent.

2.2 Common requirements

Common requirements for FCS were identified at the preliminary stage:
- Powerful microcontroller supporing Real Time Operating System
- Inertial Measurement Unit (IMU) for position and orientation determination, including 3-axis gyroscope, 3-axis accelerometer and 3-axis magnetometer
- Barometric pressure sensor for altitude estimation
- GPS receiver for UAV global position estimation
- Wireless communication channel for telemetry transmission and in-flight commands receiving
- MicroSD card support for storing configuration parameters and flight data acquisition
- USB interface for PC connection for firmware uploading and test / debug procedures
- PWM inputs and outputs for standard RC receiver signals handling (required for manual control) and motors control respectively
- Several ADC channels for ultrasonic range finder connection and battery voltage / current monitoring.

2.3 FCS block diagram

Based on the requirements listed above, FCS can be presented in the form of the following block diagram:

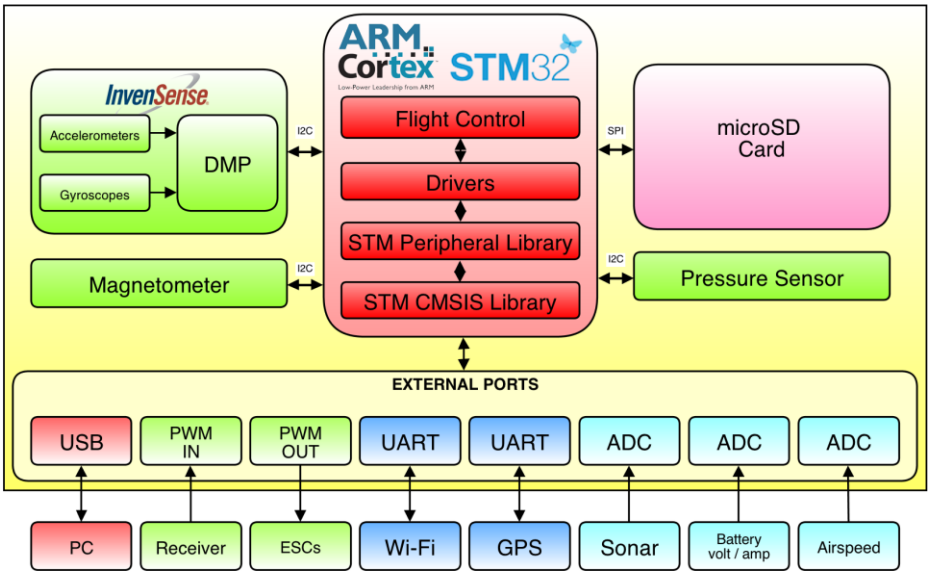

Figure 1: FCS block diagram

The software architecture consists of several levels: from the lowest - STM microcontroller libraries to the highest – flight control logic. CMSIS Library from STM makes possible to create higher software hardware-independent. The second level is STM Library which provides high-level functions for microcontroller peripheral communications. Next level includes drivers for sensors readings and actuators control. The highest level consists of functions responsible for stabilization, navigation, flight control and digital signal filtering where it's necessary. Additionally, the highest level includes GCS interface functions based on MAVLink [1] communication protocol library. The entire software is run on FreeRTOS [2] (Real Time Operating System) developed especially for embedded applications which provides functionality for effective tasks and resources control being crucial for such devices. Every function belongs to its priority group corresponding to its importance for safety guidelines.

2.4 FCS Printed Circuit Board (PCB)

To meet the requirement of small device size the routing is done on four layers PCB with the components placed on both sides. The external layers #1 and #4 contain signal traces. Internal layers #2 and #3 are filled with Ground and Power supply. The size of the PCB is 60 by 40 mm. Schematics and routing were done in Eagle CAD [3].
The routing image, generated 3D render of the PCB and the photo of assembled board are shown below.

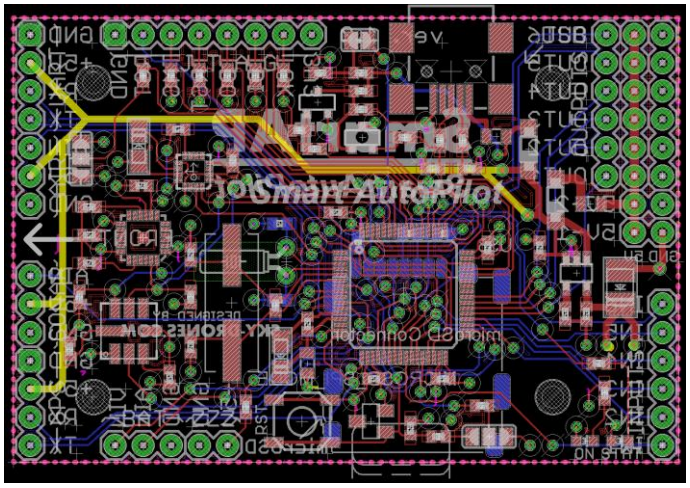

Figure 2: PCB routing

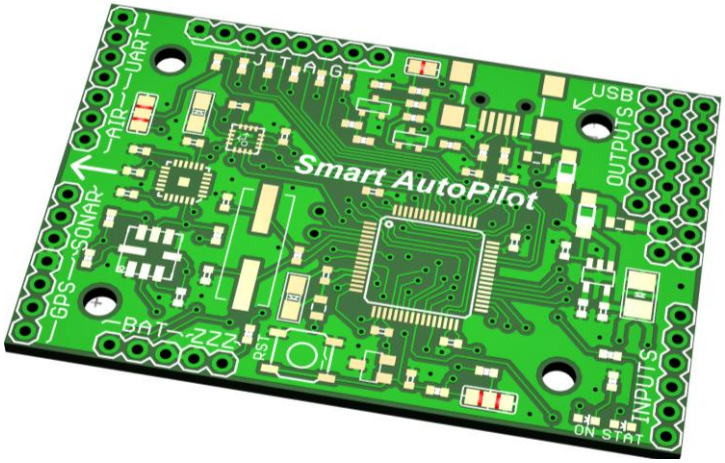

Figure 3: 3D render of the PCB

Figure 4: Assembled FCS hardware

Therefore, hardware platform is prepared for further software development and can be easily integrated in MAV for further flight tests.

3. Sensors communication

3.1 Digital sensors and update rates

For the required sensors were chosen the following ones:

- InvenSense MPU-6050 [4] (accelerometer and gyroscope)
- Honeywell HMC5883 [5] (magnetometer)
- Bosch BMP085 [6] (barometric pressure sensor)

Every sensor has digital I2C (Inter-Integrated Circuit) interface for communication with microcontroller unit (MCU). Maximum operating frequency of I2C bus is 400 kHz.
Firstly, required update rates and corresponding data packets sizes were identified:

Table 1: Sensors update rates and corresponding packets' size

Device	Packet size, B	Update rate, Hz
IMU (Raw data)	10	400
IMU (DMP data)	18	200
Magnetometer	6	50
Barometer	2	50

3.2 IMU data handling

IMU consists of 3-axis gyroscope, 3-axis accelerometer and 3-axis magnetometer. This set of sensors provides sustainable solution for determination of orientation in space. InvenSense MPU-6050 sensor combines 3-axis accelerometer, 3-axis gyro and motion processor DMP (Digital Motion Processing) integrated on the same silicon die in the package of 4.0x4.0x0.9mm. DMP performs calculations (angular rates integration, gyroscope bias correction, extended Kalman filtering) in order to get orientation in space in quaternion representation at the frequency of 200Hz, which significantly helps to reduce main CPU load. However, external magnetometer is still required for Z-gyroscope bias correction.

Communication with the IMU sensor is done in two stages:
• Reading raw data (accelerations, angular velocities) from registers
• Reading processed orientation data from DMP using FIFO (quaternions)

Maximum registers update frequency is 1 kHz. As it was mentioned above, RAW data reading is performed at the frequency of 400 Hz. Theoretically the frequency could be raised, however, this wouldn't be rational and cause extremely high load for the microcontroller. Moreover, the maximum update frequency for motor controllers is 500 Hz (limited by the maximum PWM width of 2 ms). Therefore, the frequency of 400 Hz is optimal for raw accelerometer and gyroscope readings.
DMP calculates the orientation in space at the maximum frequency of 200 Hz. The result is given in quaternion representation, which can be easily converted to Euler angles. Euler angles describe the rotation of a rigid body in three-dimensional space. In this case, roll, pitch and heading are represented with Φ, θ and Ψ angles respectively.

DMP output can be presented in the following form:

$$\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{1}$$

$$|\mathbf{q}|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{2}$$

Quaternions can be associated with the rotation about each axis in the following manner:

$$\begin{aligned} \mathbf{q}_0 &= \cos(\alpha/2) \\ \mathbf{q}_1 &= \sin(\alpha/2)\cos(\beta_x) \\ \mathbf{q}_2 &= \sin(\alpha/2)\cos(\beta_y) \\ \mathbf{q}_3 &= \sin(\alpha/2)\cos(\beta_z) \end{aligned} \tag{3}$$

Where α is the rotation angle around the axis, and cos(βx), cos(βy) and cos(βz) are direction cosines defining the axis of rotation.

Quaternions into Euler angles conversion can be done as follows:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\frac{2(q_0 q_1 + q_2 q_3)}{1-2(q_1^2+q_2^2)} \\ \arcsin(2(q_0 q_2 - q_3 q_1)) \\ \arctan\frac{2(q_0 q_3 + q_1 q_2)}{1-2(q_2^2+q_3^2)} \end{bmatrix} \tag{4}$$

A simple desktop application on Processing [7] was designed to test the IMU. It reads incoming data from the FCS via Virtual COM port and makes the orientation of parallelepiped corresponding to the board rotation:

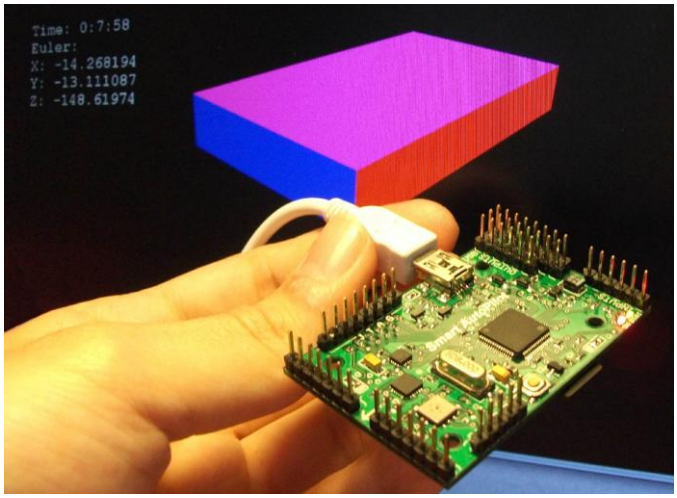

Figure 5: Application for IMU tests

3.3 Pressure sensor data handling

It's known that the relation between atmospheric pressure and altitude can be presented with the following graph:

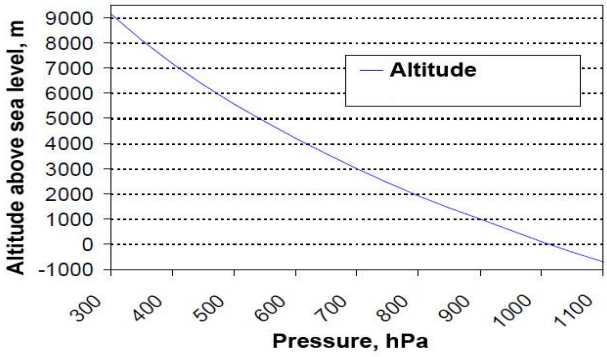

Figure 6: Altitude – Pressure function

This relation can be described with the following equation:

$$H = 44330(1 - (\frac{P}{P_0})^{\frac{1}{5.255}}) \tag{5}$$

H – altitude
P – pressure at the altitude H above sea level
P_0 – pressure at the sea level (101325 Pa)

The reading process is carried out in four stages:
• Sending request for temperature
• Reading temperature
• Sending request for pressure
• Reading pressure

At the initialization stage MCU reads barometer calibration values which are later involved in the calculation of temperature and pressure. Barometer has four modes of operation depending on the required accuracy and current consumption. In this case the highest precision mode is selected (Ultra High Resolution mode). Even though, the accuracy doesn't satisfy the requirements, so multi-sensor data fusion must be applied.

4. Increasing altitude measurement accuracy

4.1 General concepts

Different sensors can be applied for altitude measurement:

- Barometric pressure sensor
- Ultrasonic range finder
- Accelerometer
- GPS

Each of listed sensors has both advantages and disadvantages. The advantage of the first one is that there is no boundary of its applicability, while ultrasonic range finder is able to measure distances only up to 6 meters. Unfortunately, the resolution of barometer is only about 1 meter and the signal is very noisy. Range finder resolution is 1 cm. Accelerometer has outstanding dynamic characteristics, but its measurement accuracy decreases in time. GPS altitude accuracy is no more than 1 meter, moreover GPS can not function indoor. Therefore, pressure sensor and accelerometer are chosen to be the primary sensors.

4.2 Pressure sensor

The spread of readings is about 1.5 meters, moreover the signal is very noisy, therefore, filtering must be applied. Firstly, running average filter [8] was applied. Mathematically, it can be presented in the following way:

$$y_k(n) = \frac{\sum_{m=0}^{k-1} x(n-m)}{k} \tag{6}$$

Where *x(n)* is an input signal for the n-th measurement, *y(n)* is an output signal of n-th measurement and k is the filtering depth. Lower k values correspond low filtration depth and lower latency, higher k values increase smoothness, but also increase latency.
Based on the dynamic model of the multicopter its maximum vertical speed is 5 m/s. After analyzing the behaviour of the averaging filter for different values of k, it was concluded that it's not appropriate due to either weak noise reduction or big latency.
Then it was decided to apply Kalman filter [9]. This filter extrapolates a value and adjusts deviation at each step. The algorithm implemented for filtering can be described in the following way:

$$p = p_0 + q \tag{7}$$

p – estimated deviation, p_0 – deviation at the previous step, *q* – process noise

$$k = \frac{p}{p + r} \tag{8}$$

k – Kalman filter gain, *r* – sensor noise

$$y(n) = y(n) + k(x(n) - y(n)) \tag{9}$$

y(n) – output value, *x(n)* – input value

$$p_1 = (1 - k)p \tag{10}$$

Finally, estimated deviation is updated.

The filter is applied to each measurement and initialized with the initial process noise *q* and noise sensor *r*. Values of the estimated deviation *p* and filtering gain *k* are updated at the each step. Therefore, tuning values for the filter are *q* and *r*. The experimental method was used to select the optimal values of *q* and *r* to achieve the necessary level of filtration depth and lowest possible latency. The result is presented on the graph below.

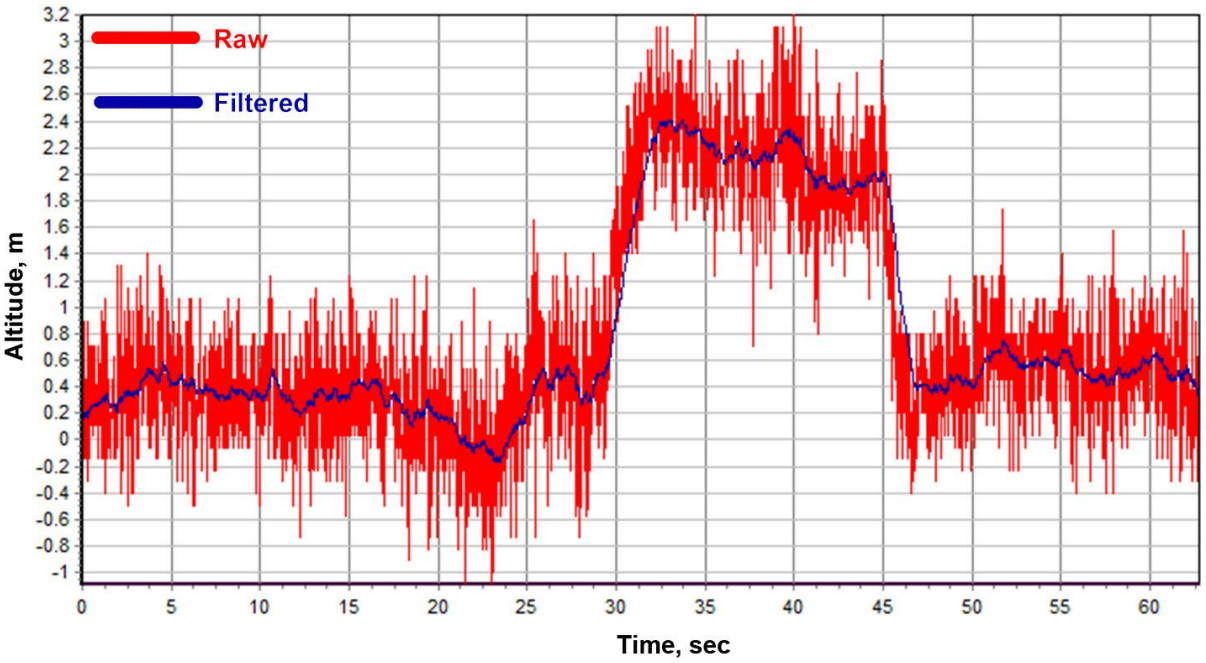

Figure 7: Altitude calculation based on atmospheric pressure with Kalman filter applied

Thus, it is possible to reduce the pressure sensor noise level significantly. However, the result is still not sufficient and can be improved with an accelerometer.

4.3 Accelerometer

It's know that the acceleration is the rate of change of the velocity of an object. At the same time, the velocity is the rate of change of the position of that same object. In other words, the velocity is the derivative of the position and the acceleration is the derivative of the velocity:

$$\vec{a} = \frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \frac{\mathrm{d}^2\vec{s}}{\mathrm{d}t^2} \tag{11}$$

Similarly, knowing the acceleration, velocity and position can be easily obtained:

$$\vec{s} = \int_0^t (\vec{v})dt = \int_0^t (\int_0^t (\vec{a}))dt \tag{12}$$

Graphically, it can be presented as follows (*f(x)* – accelerometer's value):

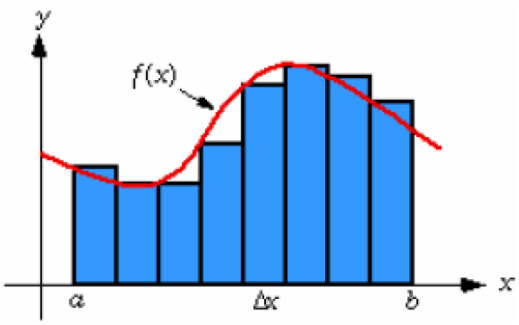

Figure 8: Sampled accelerometer value

After digitization integral can be replaced with the sum:

$$\int_0^t f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i)\Delta x \tag{13}$$

$$\Delta x = \frac{b-a}{n} \tag{14}$$

Due to the sampling process integration error is constantly accumulating. This is shown in the figure below:

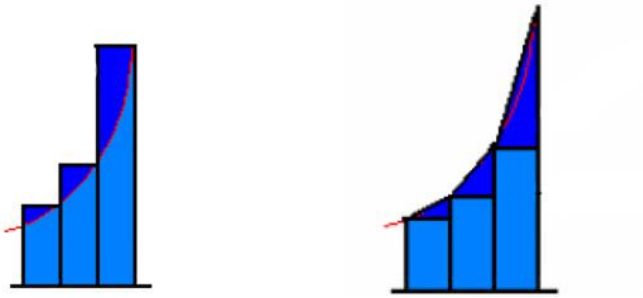

Figure 9: Comparison of "Rectangle" and "Trapezoidal" methods of integration

First-order approximation method (method of trapezoids) is used to reduce the integration error. Area integration S_n at the *n*-th iteration can be presented as follows [10]:

$$S_n = \frac{f(x_n) + f(x_{n-1})}{2}(x_n - x_{n-1}) \tag{15}$$

Summarize some useful ideas which help to move from the theory to practice:

- The signal from an accelerometer contains noise, filter must be applied
- Mechanical noise (e.g. frame vibration) also affects negatively on the sensor
- Accelerometer must be calibrated in the absence of movement to avoid further error accumulation
- Sampling time must be strictly constant

Movement consists of acceleration, constant velocity motion and deceleration. After the movement stops the velocity must have a zero value. In a real world where the area below the positive side of the curve is actually not the same as the area above the negative side, the integration result would never reach zero velocity and therefore would be unstable. It is caused by the fact that MEMS [11] accelerometers have wide noise band, which is approximately ± 0.1 m/s^2 (for the implemented sensor). That's why it is crucial to force the velocity down to zero. Acceleration is being constantly read and compared with zero. If zero acceleration exists during the certain number of samples, velocity is claimed to be zero.

It is clearly seen from the graph that the error is accumulating. This measuring method provides good dynamic characteristics, but the accuracy deteriorates in time. Barometer sensor is intended to compensate that error.

The results of the integration are presented on the graph below:

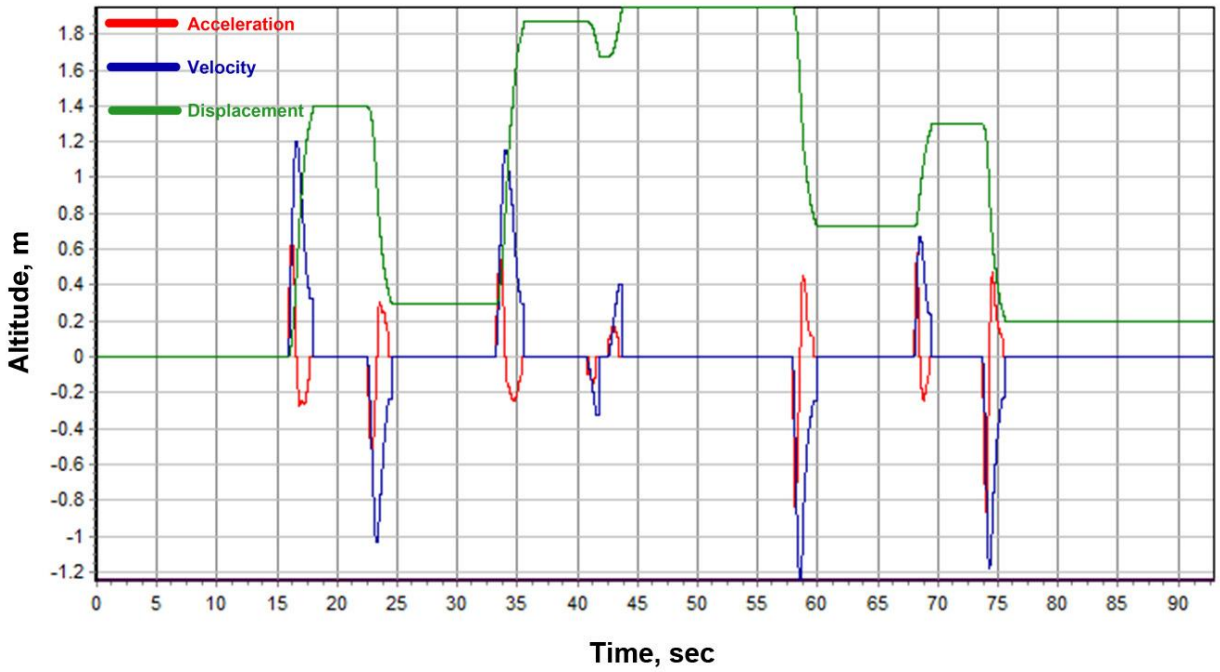

Figure 10: Acceleration – Velocity – Displacement

4.4 Fusion

Considering the basic physical differences in the behaviour of barometer and accelerometer, it can be concluded:

- Barometer
  - Poor dynamic response, big latency
  - The accuracy of measurements remains constant
- Accelerometer
  - Excellent dynamic performance, instant response to the movement
  - Measurement accuracy worsens with time

Based on these ideas the altitude is calculated by accelerometer and barometer simultaneously. The preference is given to the accelerometer, which value is pulled to the barometer value at each iteration. The following idea can be presented as follows:

$$H = H_{accel} - H_{bias} \tag{16}$$

$$H_{bias} = k(H - H_{baro}) \tag{17}$$

H – Altitude (after fusion)
H_{accel} – Altitude calculated with accelerometer readings
H_{baro} – Altitude calculated with barometer readings
H_{bias} – The amount of error which should be contracted at the each step
K – The rate of compensation

Lower K values give more preference to accelerometer, higher – to barometer.
After applying filter and fusion algorithms altitude accuracy is ±15cm without increasing latency in dynamic performance.

Finally, the result of fusion is presented on the graph:

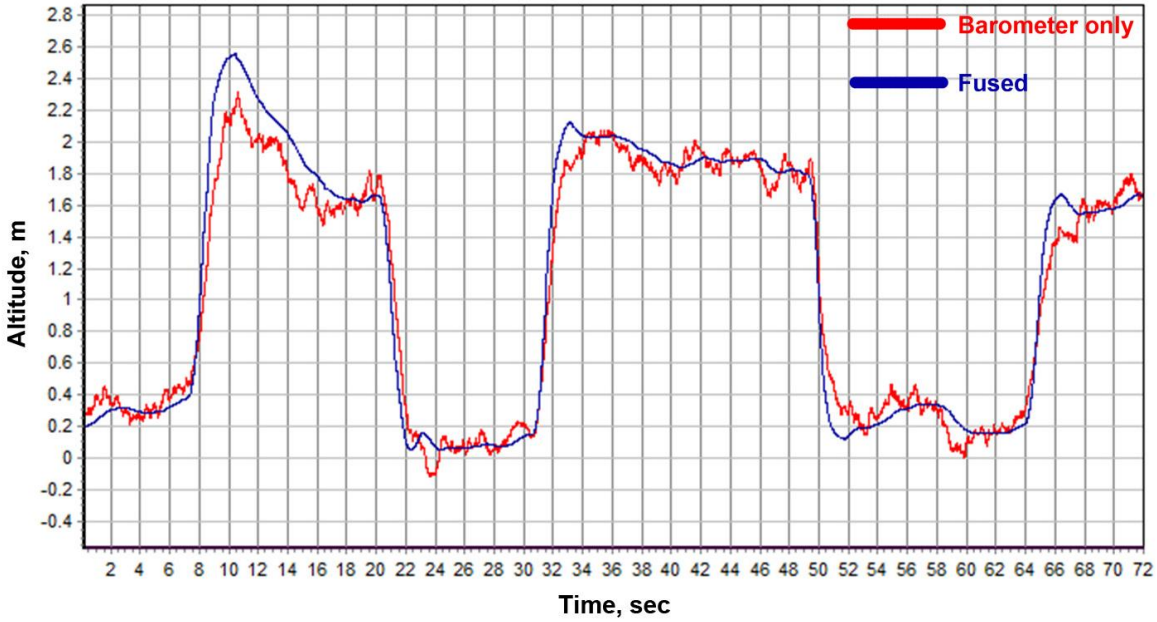

Figure 11: Barometer altitude and fused altitude comparison

5. Increasing position measurement accuracy

5.1 Analazyng raw GPS signal

Firstly, the readings from GPS receiver were analyzed. Ublox LEA6-H [12] was used for the tests. This is 50 channel receiver with 5 Hz update rate. Being located at the same place all the time (for 15 minutes) turned out that the spread of readings is almost about ± 6 meters. The form of the signal is random and can not be filtered, otherwise it would cause a huge latency.

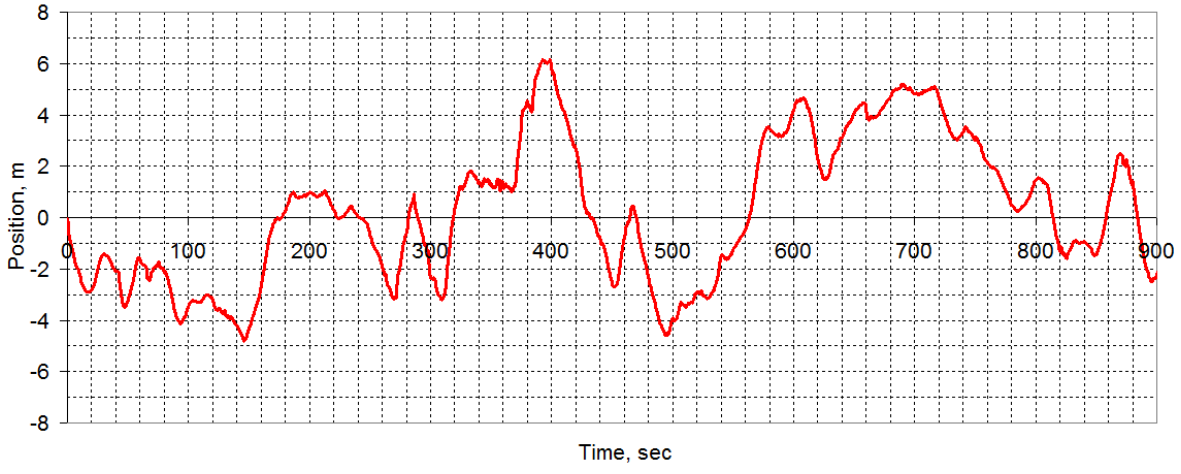

Figure 12: GPS position fluctuation

5.2 GPS and accelerometer fusion

For increasing position measurement accuracy the same approach as for altitude is applied. Accelerometer-based position is chosen to be the primary with compensation by GPS. Detailed description of intergration and fusion methods is given in the previous section and demonstrates the results of increasing altitude measurement accuracy.

One of the differences is that the K factor (the rate of compensation) should be much lower for GPS than for barometer. Such value is chosen to reduce the amplitude of noise, which period time is about 60-100 sec. After fusion the system is also capable to sense short displacements (less than 0.5m) which was impossible with GPS only. Finally, the results are presented on the graphs below.

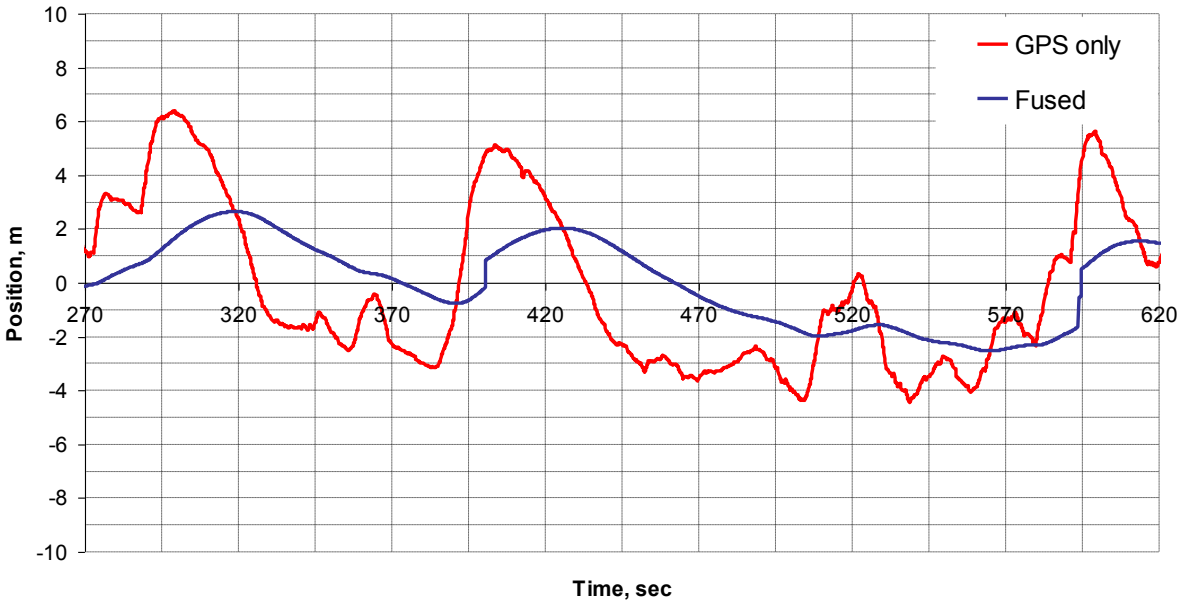

Figure 13: Raw GPS readings and fused position

The graph below shows the capability of sensing of sensing the short displacements.

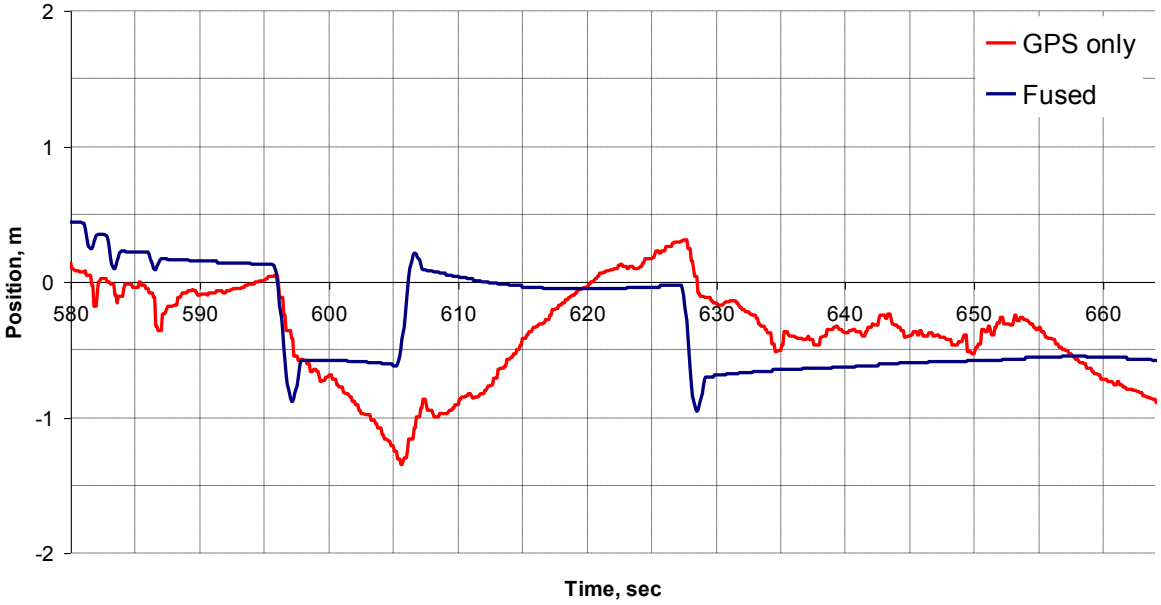

Figure 14: Raw GPS and accelerometer dynamics

Therefore, accelerometer and GPS data fusion method increased position precision at least twice and provided better dynamic characteristics (movement response time is reduced).

6. Conclusion

The general concepts and approaches of multirotor UAV Flight Control System development were presented in the paper. The emphasis of the research was done on increasing UAV positioning precision. It can be achieved with multi-sensor data fusion methods which were considered and described as well as the numerical results presented. Altitude measurement accuracy was increased from ±1m to ±30cm with better dynamic characteristics by the means of accelerometer and barometer data fusion. Position measurement accuracy was increased from ±6m (raw GPS) to ±2m by the means of accelerometer and GPS fusion with improving dynamic characteristics too.

References

[1] MAVLink Micro Air Vehicle Communication Protocol, http://qgroundcontrol.org/mavlink/start
[2] Using the FreeRTOS Real Time Kernel - a Practical Guide - Cortex-M3 Edition, Richard Barry
[3] The essence of Printed Circuit Boards Design with EAGLE, Tetsuya Gokan, 2009: Gijutsu-Hyohron Co., Ltd. ISBN 978-4-7741-4063-6
[4] MPU-6000 and MPU-6050 Product Specification, InvenSense, http://invensense.com/mems/gyro/documents/PS-MPU-6000A.pdf
[5] 3-Axis Digital Compass IC HMC5883L, Honeywell
[6] BMP085 Digital, barometric pressure sensor, Bosch
[7] Processing: A Programming Handbook for Visual Designers and Artists, Casey Reas, Ben Fry, 2007; MIT
[8] Practical Analog and Digital *Filter* Design - DSP-*Book*, Les Thede, Artech House, Inc; 2004
[9] Introduction to Estimation and the Kalman Filter Hugh Durrant-Whyte Australian Centre for Field Robotics The University of Sydney, NSW 2006 Australia, 2001
[10] Implementing Positioning Algorithms Using Accelerometers, Kurt Seifert and Oscar Camacho, Freescale Semiconductor, AN3397
[11] Development of Hight-Perfomance, High-Voulume Consumer MEMS Gyroscopes, InvenSense, Joe Seeger, Martin Lim and Steve Nasiri, http://invensense.com/
[12] *LEA*-6 *u-blox* 6 GPS Module, http://www.u-blox.com