The Next Generation Design of Autonomous MAV Flight Control System SmartAP

Kirill Shilov*
Department of Aeromechanics and Flight Engineering
Moscow Institute of Physics and Technology
16 Gagarina st, Zhukovsky, Moscow reg., Russia 140180

ABSTRACT

The present article contains the most recent and significant results of the author's work in the field of autonomous flight control system for multirotors. The work is based on author's custom designed autopilot called SmartAP. By the moment, the system has two generations - version 1.0 released in 2012 and version 2.0 released recently in 2014. The changes affected both hardware and software. The new architecture is capable of running more complex algorithms helping to increase navigation precision and enhancing autonomy level. The developed system is capable of fully autonomous flight including take off and landing.

1 INTRODUCTION

Aerial monitoring has always been important part of surveillance and terrain exploration. Until recently, this task could have been solved only with piloted aircrafts, however, due to the latest progress in electronics, becomes possible creating of small Unmanned Aerial Vehicles (UAV) or Micro Air Vehicles (MAV). The capability of carrying payload (photo / video camera) and accomplishing preprogrammed missions autonomously helps to reduce exploitation costs significantly and doesn't require piloting skills from operator. Special interest is paid to multirotor UAVs which are capable of vertical take off and landing and hovering flight. These features makes them easy to operate in indoor environment and space-limited conditions. Obviously, the behavior of such vehicles is unstable and onboard Flight Control System (FCS) responsible for stabilization (the ability to stay in the air) and navigation (the ability to follow the desired trajectory) is required.
Research and development in the area of autonomous multirotor flight control systems are of interest. This paper contains the information on the difference in hardware design between the first and the second generation of the author's custom developed system as well as the description of the software architecture, navigation and control algorithms and provides the results achieved during autonomous flight tests.

*Email address: aviaks.kirill@gmail.com

Figure 1: Quadrotor powered by SmartAP autopilot performs the mission at IMAV 2013, France

2 THE HARDWARE

One of the main requirements for the next generation hardware were compact size and maximum compatibility with the previous version. The routing was done on four-layer PCB, the components placed on both sides.
The size of the board is 60x40 mm, weight is 16 grams. The top side of the board is shown in the Figure 2.

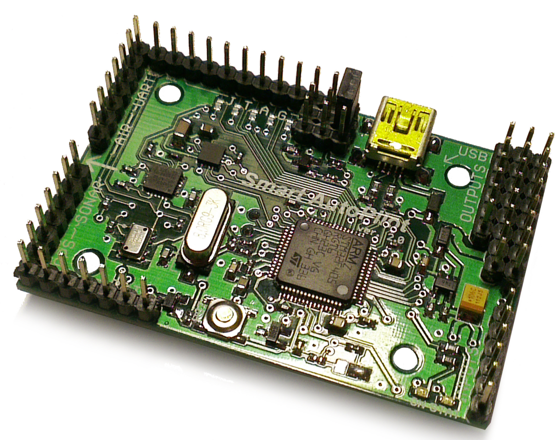

Figure 2: SmartAP v2.0 Hardware

This following section explains the general differences in the hardware design between two generations of the board:

2.1 *Microcontroller*

One of the key features of the second generation hardware is more powerful microcontroller. The first generation of the hardware has STM32F103 microcontroller, while the next is based on STM32F405 [1]. It has not only higher operating frequency (168 MHz vs 72 MHz) and more flash memory (1 MB vs 256 kB), but also Floating Point Unit (FPU) module which helps to reduce CPU load significantly during math operations, therefore, allowing to perform more complex attitude estimation algorithms and adaptive control loops.

2.2 *Pressure sensor*

Pressure sensor (barometer) plays very important role in MAV's altitude estimation and defines the precision of altitude hold mode. The previous generation of the board has Bosch BMP085 [2] barometer. The best accuracy of the measurements with this sensor was around $\pm$ 80 cm. After studying and comparing various pressure sensors test results it was concluded that MS5611-01BA03 [3] barometer from Measurement Specialties has the best performance available at the moment. The typical measurement results for two sensors are shown in the Figure 3 and 4 below:

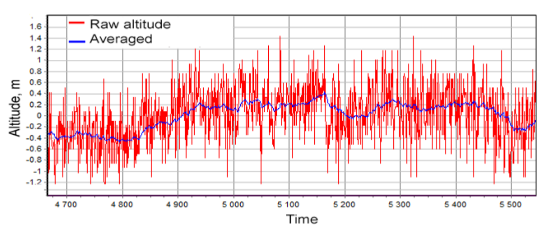

Figure 3: BMP085 pressure measurement results

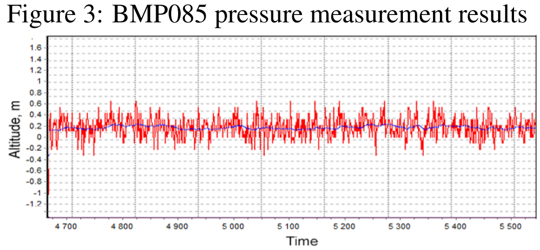

Figure 4: MS5611 pressure measurement results

2.3 *Signaling*

Sound and light signaling devices can be used to inform a user about the state of MAV (arm / disarm, calibration procedure, error and etc.). Therefore, two external digital output ports triggered through transistor for electromagnetic sounder and bright LED (maximum current is 100 mA for each channel) were added.

2.4 *Battery voltage monitoring*

One of ADC input ports connected through voltage divider allows measuring battery voltage directly in the range up to 30 V with 12 bit resolution.

2.5 *External ports*

Several external power supply ports (3.3 V and 5 V) as well as I2C line were added for further sensors on breakout board tests and external magnetometer connection [4].

3 SOFTWARE ARCHITECTURE

Autonomous control of multirotor MAV can be achieved with the ability of maintaining desired orientation and position in space, therefore, FCS is subdivided into three parts:

1. Information acquisition from the sensors (accelerations, angular rates, GPS position, atmospheric pressure and etc.)
2. Obtained information is exposed to verification, filtering and fusion in order to improve accuracy
3. Navigation and control algorithms execution.

The software architecture is based on Real Time Operating System FreeRTOS [5] running on STM32 ARM Cortex M4 microcontroller and presented in the form of a block diagram in the Figure 5:

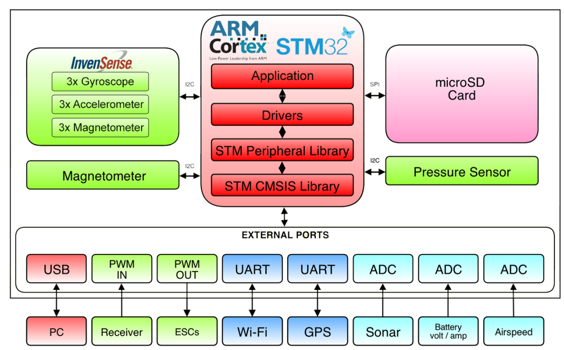

Figure 5: Block diagram architecture of the system

The software is divided into four subgroups. STM CMSIS Library and STM Peripheral Library form hardware abstraction level. The highest level - Application, includes attitude estimation and control algorithms as well as autopilot service functions. Therefore, the transition to another hardware can be done easily. The greatest computational time is taken by IMU attitude estimation algorithm and sensors' data acquisition task. Nevertheless, IDLE time is 71%. More details concerning task and corresponding CPU load are presented in the Table 1.

Task name	CPU Load, %
IDLE	71
IMU attitude estimation	16
Sensors' data acquisition	8
Control algorithms	3
Other	2

Table 1: Task name and corresponding CPU load

4 Inertial Measurement Unit

The previous generation of SmartAP was based on InvenSense Digital Motion Processor (DMP), meaning that the sensor itself calculated attitude solution presented in the form of quaternions [6]. This method contributes to a significant reduction of main CPU load, however, it provides only 6 DoF solution since MPU-6050 [7] doesn't have integrated magnetometer and standalone Z-gyroscope bias correction is impossible.

Therefore, it was decided to refuse from DMP and read raw accelerometer, gyroscope and magnetometer data instead. Attitude estimation uses a quaternion representation, allowing accelerometer and magnetometer data to be used in an analytically derived and optimized gradient-descent algorithm to compute the direction of the gyroscope measurement error as a quaternion derivative.

This approach helps reducing measurement error related with gyroscope bias and also makes the solution more sustainable to vibrations. Attitude estimation with the described algorithm proved to be better than DMP solution in real tests. Screen of application for real-time IMU comparison tests is shown in the Figure 6.

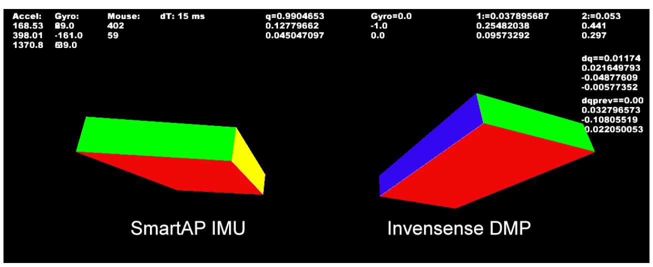

Figure 6: Application for real-time comparison of internal and DMP attitude solutions (left parallelepiped - own solution, right parallelepiped - DMP solution)

Yaw control of MAV was tested during the flight and showed good results for transition process when reference heading was changed as well as the absence of bias. The results are shown in the Figure 7.

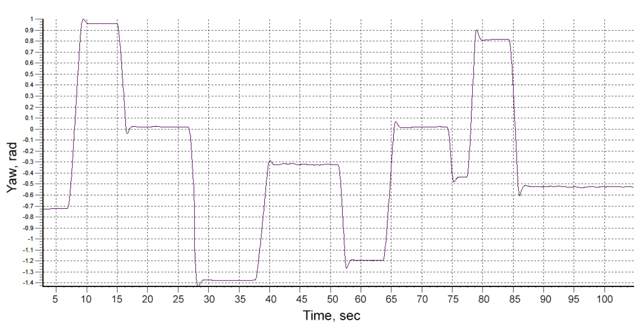

Figure 7: Measured yaw angle

5 Control

The control algorithms are based on PID controller [8], which output signal $u(t)$ can be presented as follows:

$$u(t) = P + I + D = K_p e(t) + K_i \int_o^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

where $e(t)$ is an error and K_p, K_i and K_d are proportional, integral and derivative gains respectively. This method can be applied for angles stabilization, as shown in the Figure 8 below:

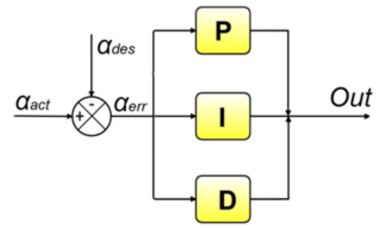

Figure 8: PID control block diagram

where α_{act} is measured angle and α_{des} is a reference angle. However, this approach turned out to be quite difficult for tuning and satisfied process behavior couldn't been achieved. Typical response curve for PID loop angle control looks as shown in the Figure 9:

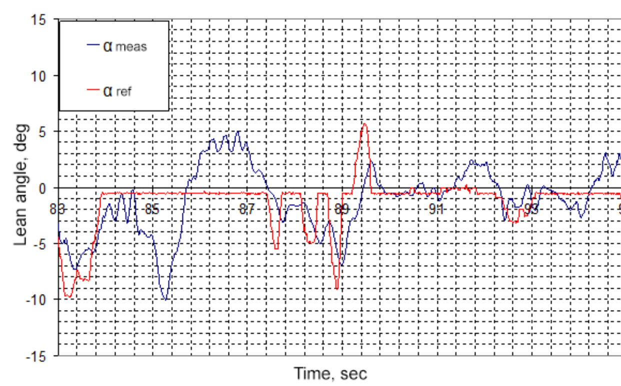

Figure 9: Response curve for PID loop

The same idea, but with some modifications gave much better results. It includes two PI controllers, one is used to stabilize the angular rate of rotation, while the second is used to stabilize the angle itself. This method turned out to be much easier for tuning and gave better results of vehicle's stabilization. Block diagram of the described method is presented in the Figure 10.

And typical response curve is shown in the Figure 11.

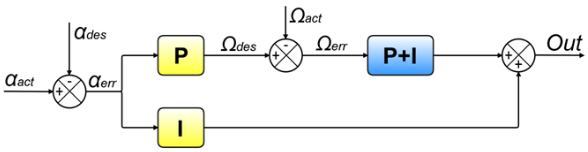

Figure 10: PI-PI control block diagram

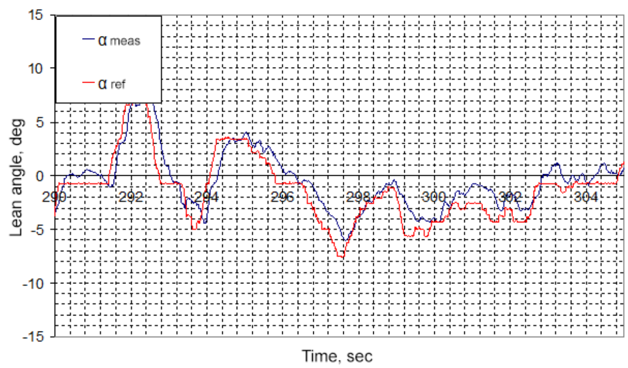

Figure 11: Response curve for PI-PI loop

6 Autonomous Take off and Landing

Take off and landing modes are ones of the most important for autonomous flight and had been studied and tested carefully.

6.1 Take off mode

After activating takeoff mode the motors start spinning and MAV tries to achieve the predefined altitude of 5 meters. Upon reaching this altitude it goes to "Loiter" mode.
Altitude can be measured with barometer and ultrasonic range finder (sonar). Sonar gives better resolution (1 cm), however, its measurement results are very noisy at low distances. Pressure sensor has much less resolution, but measurements are not affected by noise. One of the difficulties is determination of average throttle value required for hovering. This value is fixed when lift-off occurs.
The typical take off pattern is shown in the Figure 12. Both barometer and sonar altitudes are presented, where sonar noise can be clearly seen during lift-off.
After achieving the altitude of five meters, accuracy of altitude hold is no worse than $\pm$ 50 cm.

6.2 Landing mode

After activating landing mode MAV maintains the vertical speed of -0.5 m/s. Touch down with the ground is determined by Z-accelerometer readings, which in turn signalize that the motors can be turned off. The typical pattern of the landing is shown in the Figure 13.

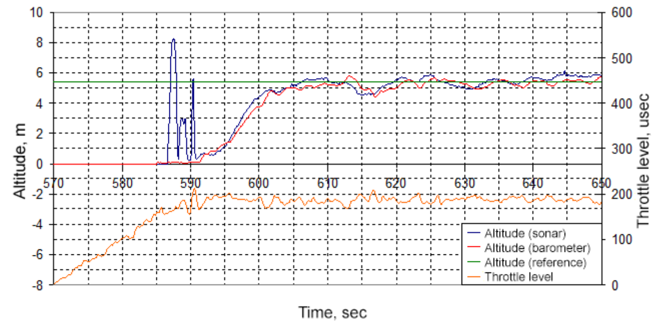

Figure 12: Flight parameters during autonomous take off

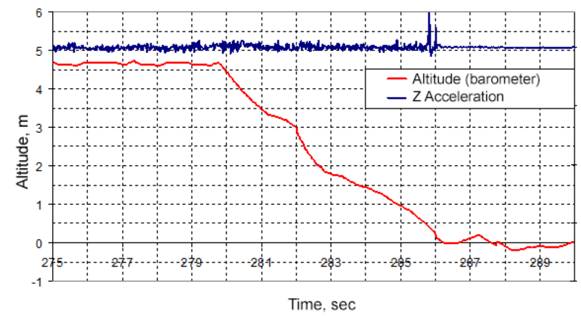

Figure 13: Flight parameters during autonomous landing

The vehicle is holding the altitude of 4.7 m. At 286 sec it is staring descending with the vertical speed of 0.7 m/s. Peak in acceleration readings can be clearly seen meaning the vehicle touched the ground. The system shuts down motors and disarms.

7 Autonomous Position Hold

Similar PI-PI cascade is used for position control. Firstly, the reference velocity for current position error is calculated. Reference velocity is compared to the measured one and is converted to lean angles in north and east directions required to achieve the reference position. Navigation algorithm converts north / east lean angles to roll and pitch angles. The accuracy of position hold is about $\pm$ 0.5 m. The typical trajectory of the vehicle in position hold mode is presented in the Figure 14.

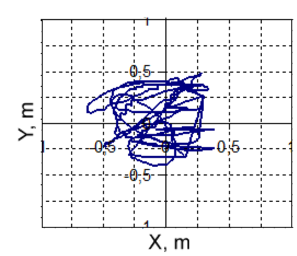

Figure 14: MAV Trajectory in position hold mode

8 AUTONOMOUS WAYPOINT FLIGHT

Autonomous waypoint flight mode is generally based on the position hold mode. Upon reaching every new desired point, the next point of waypoints list is set as the desired until the last waypoint is reached. After that, home point is set as the desired.
Reference trajectory (including four waypoint and home position) and vehicle's trajectory are shown in the Figure 15:

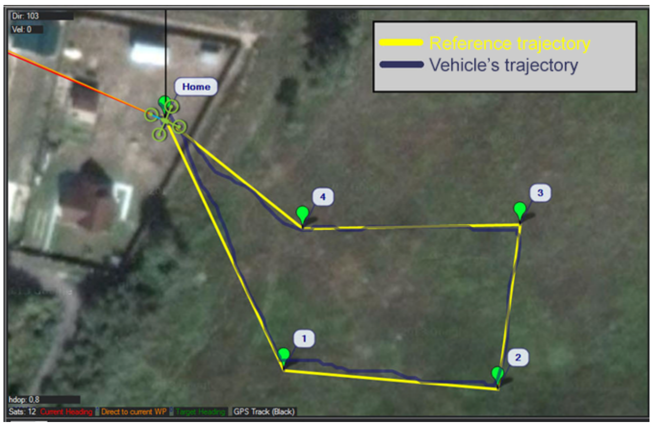

Figure 15: MAV trajectory in waypoint flight

9 GROUND CONTROL STATION

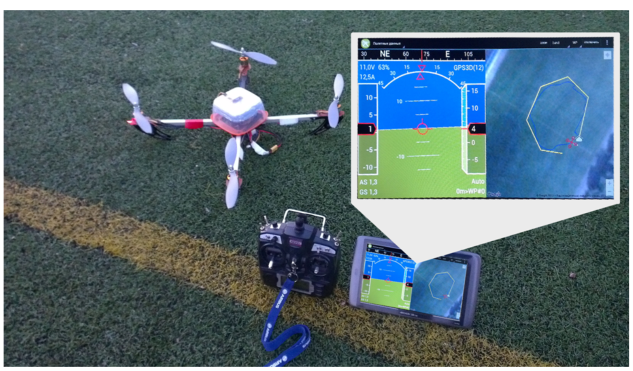

Figure 16: Android-based Ground Control Station

Communication between MAV and operator can be done with the help of the Ground Control Station (GCS) through the radio link. The autopilot supports MAVLink protocol [9], meaning that it is compatible with any GCS supporting MAVLink too.
GCS provides high-level mission control and doesn't require piloting skills from operator, which makes the system very easy to use.
One of supported ground stations is Droidplanner (for Android-based devices) is shown in the Figure 16.

10 CONCLUSION

The general aspects regarding mutirotor MAV flight control system design for autonomous flights were considered in the present article. Brief overview of the hardware and software architectures was given as well as the differences between two generations of the board. Two attitude estimation methods had been compared. Two types of PID controllers had been also tested and compared. Autonomous take off and landing mode based on barometer and ultrasonic range finder altitude estimation had been analyzed. Some results on autonomous waypoint flight presented as well. Current research topics are positioning accuracy improvement by the means of multisensor data fusion and Kalman filtering and model-based adaptive control algorithms.
More information can be found on project's website [10] .

11 REFERENCING

[1] High-density performance line ARM-based 32-bit MCU, USB, CAN, 11 timers, 3 ADCs, 13 communication interfaces, STMicroelectronics, http://st.com/
[2] BMP085, Digital, barometric pressure sensor, BOSCH, http://www.bosch-sensortec.com
[3] MS5611-01BA03 Barometric Pressure Sensor, with stainless steel cap. DA5611-01BA03_011 www.meas-spec.com. Oct. 26, 2012. 000056111624 ECN1742
[4] 3-Axis Digital Compass IC HMC5883, Honeywell, http://honeywell.com/
[5] FreeRTOS - market leading real time operating system (or RTOS) from Real Time Engineers Ltd., http://www.freertos.org/
[6] Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, J. B. Kuipers, 2002
[7] MPU-6000 and MPU-6050 Product Specification Revision 3.2, Invensense, http://invensense.com/
[8] Introduction to PID Controllers - Theory, Tuning and Application to Frontier Areas, Edited by Rames C. Panda, ISBN 978-953-307-927-1, 258 pages, Publisher: InTech, Chapters published
[9] MAVLink Micro Air Vehicle Communication Protocol - http://qgroundcontrol.org/mavlink/start
[10] http://sky-drones.com/