2015 International Conference on Process Control (PC)
June 9–12, 2015, Štrbské Pleso, Slovakia

Bicopter stabilization based on IMU sensors

Lukáš Hrečko
Institute of Robotics and Cybernetics, Faculty of Electrical
Engineering and IT, Slovak University of Technology,
Ilkovičova 3, 812 19 Bratislava

Juraj Slačka, Miroslav Halás
Institute of Robotics and Cybernetics, Faculty of Electrical
Engineering and IT, Slovak University of Technology,
Ilkovičova 3, 812 19 Bratislava
[juraj.slacka, miroslav.halas]@stuba.sk

Abstract— **This paper describes the construction of a laboratory model of a bicopter, fully capable of a flight, and shortly discusses all related control problems. Namely, the problem of its stabilization during the flight, the design of the respective control loops and controllers, including signal measurement and processing, are presented in this work.**

Keywords—bicopter; stabilisation; control; IMU sensors

I. Introduction

Due to the complexity of flying vehicles, which includes both construction and mathematical model, dealing with their control problems is usually not a trivial task. Typical control problems one needs to consider are related not only to control the vehicle's flight itself but also to control the process of its take off, landing, and stabilization during the flight. These problems significantly depend on the flying vehicle under consideration, and the corresponding control actions need to be chosen also with respect to the size and type of the vehicle.

In this paper the respective vehicle we consider is the so-called bicopter, which can be described as a two-rotor flying machine without variable tilting propellers attached to it, see Fig. 1. Note that the rotors and the body of the bicopter are not firmly connected, and the whole construction is more pendulum-like. That is, the rotors can rotate around the common axis perpendicular to the body of the bicopter. This makes bicopter one of the least stable aircraft construction. The size of the machine considered here is relatively small (see Fig. 9), which creates high demands for the speed of its control. The problems we focus on here are related mainly to the construction, signal processing, stabilization, and control of the bicopter.

The considered model uses Micro-Electro-Mechanical Systems (MEMS) acceleration and angle speed sensors [1]. Accelerometer and gyroscope are able to set its orientation in space. The output signals of these sensors are rather noisy and considerably drifted in comparison to real the physical quantities they measure [2], which is caused mainly by the vibrations generated by the rotors. Therefore, a certain attention needs to be paid to the process of filtration of these signals. As a suitable filter, meeting the requirements, 1-D Kalman filter [3] was chosen. Note that the output of 1-D Kalman filter is almost a signal without any noise. However, it is considerably drifted, due to the acceleration of the bicopter. To remove the effect of this unwanted drifts, the Mahony filter [4] was implemented. This filter uses quaternions which means that the final algorithm consists of just simple matrix operations, which are, despite of not very powerful microprocessor, rather modest. The output signal of this algorithm are Euler's angles [5] Yaw, Pitch, Roll, angular velocities in all three axes, and acceleration in vertical axis with compensated gravity influence.

After signal processing, control algorithm are considered. As mentioned before, due to the relative small size of the bicopter they have to work rather fast. This includes the whole process of measuring and processing the real variables, and calculating the respective control actions. For those reasons, two microprocessors are used. The first one processes the signal from MEMS sensors, which means that this processor is used as a Digital Signal Processing (DSP) unit. The second one, Multipoint Control Unit (MCU), is used as the main control unit and is more computational effective than the former. It handles the data reading from DSP processor, the main process of the bicopter control, it evaluates statistic data and also communicates with other peripherals.

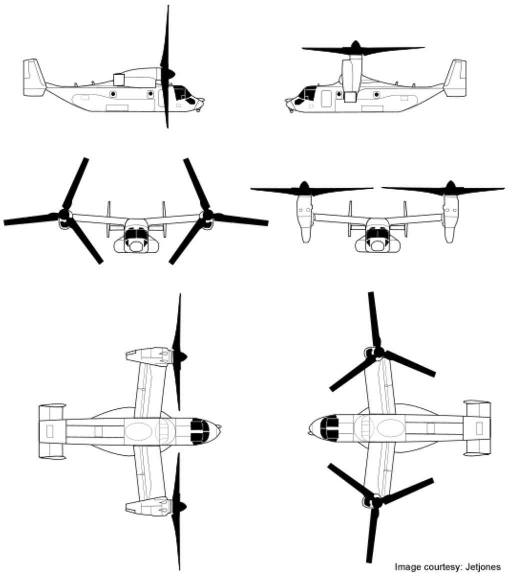

Fig. 1. V22 Osprey – principial scheme of construction

II. Mechanical construction

Mechanical construction of the bicopter body needs to meet certain, rather specific, requirements. The body has to be strong and light-weight at the same time. Ability to overlay the

978-1-4673-6627-4/15/$31.00 ©2015 IEEE

material of the body, to achieve local stiffness, is also required. For those reasons, a composite fiberglass was chosen as the main construction material, which is quite natural choice. Note that such a material is, additionally, dielectric (i.e. it does not conduct the electric current). Hence, it is preferred to for instance carbon fiber, that could be possibly considered too, but it conducts the electric current and can highly influence the direction characteristics of the antennas used to provide necessary communication [6]. As a bonding element, the epoxy bitumen was used.

The key role in the construction of the bicopter plays the so-called center wing, and it is made of fiberglass and carbon fiber. The following two important parts are placed in there: needle bearing and two miniature servomotors MG90, which are used to rotate the arms loaded with the rotors. The arms are inserted into the bearings and are of the pipe-like shape made of aluminum alloy fitted with the steel sliding areas in the place where they touch the bearings. On the bottom side of centroplan there is an Inertial Measurement Unit (IMU), and Printed Circuit Board (PCB). The IMU is placed there because it needs to be as close to the centre of the bicopter as possible.

Next, the center wing is fitted on the top of the main body, and is also made of fiberglass. The construction is very light-weight, stiff, and also flexible. Additionally, in the body of the bicopter, there is a control unit, which consists of developer board STM32VL Discovery, being on the top of the homemade reductive board carrying the power supply and connectors for the peripherals. The respective devices, i.e. brushless DC Motors (BDCM), servos which tilt the rotors, and 433Mhz transceiver Orange OpenLRS which provides the communication with the aircraft, are connected there.

As a power supply a three cell Li-Pol battery with 2650mAh capacity and constant discharge current 70A was chosen. Total weight of aircraft with the battery is then 679g.

Finally, we note that the power of rotors, really necessary to lift and thrust the bicopter, was increased more than twice in order to achieve higher maneuverability and ability to load the bicopter with additional batteries and/or electronics, if needed.

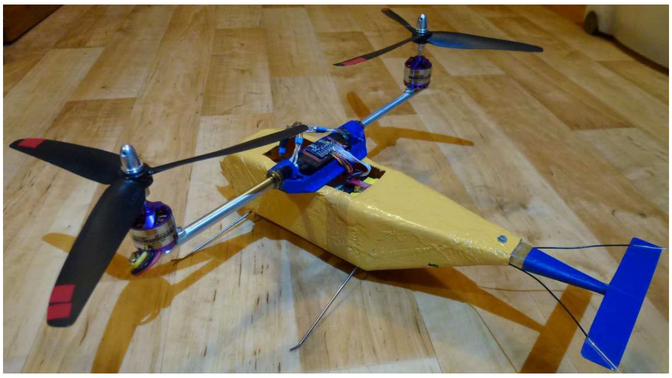

Fig. 2. Photography of mechanical construction. (old photography with two 2.4GHz antennas on tail)

III. Basic control structure

The basic control structure of the bicopter is, in a simplified form, depicted in Fig. 3. The figure shows five groups of blocks (A, B, C, D, E), starting with the remote controller block (A), which is handled by the operator, and followed by the basic control loops using PID controllers (B), one for each of the three axes of rotation (roll, pitch, yaw), and one P controller. The P controller serves mainly to lower the oscillations of the bicopter motion in the vertical direction, which is caused by flying in low altitudes. Note that this is known as a ground effect.

The output of the PID controllers is divided into two parts. The first one controls the main rotors, while the second one controls the servos used to tilt the main rotors. The input to the PID controllers consists of a signal from RC_Controller block, which processes the signal from RC remote controller (A). This signal consists of 4 values: required pitch in X and Y axis, required value of angular speed in Z axis and required power of motors. Subsequently, the angular speed and acceleration, as measured by MEMS sensors (E) and processed in Kalman and Mahony filters (D), represent the second part of the input to the PID controllers. Naturally, the measured variables are influenced by noise, caused by the vibration of the rotors, which is, however, reduced by 78% [7]. Finally, the Mahony filter reduces significantly the influence of unwanted drifts of each sensor.

The signals processed as described above are compared to required values and the final control action is computed by N-PID controllers which compute certain, but not final, control action. The reason is that first it is necessary to consider which output of which controller is needed to control the rotors and which to control the servos. Although, all four controllers are participating in thrust, the influence of controller for rotation around Z axis (yaw) can be ignored. However, tilt of the rotors is influenced just by controllers in pitch and yaw axis. In these summary blocks, the values of the respective controllers are summed based on their priorities. Finally, the output of these blocks is sent to the rotors and servos (blocks D).

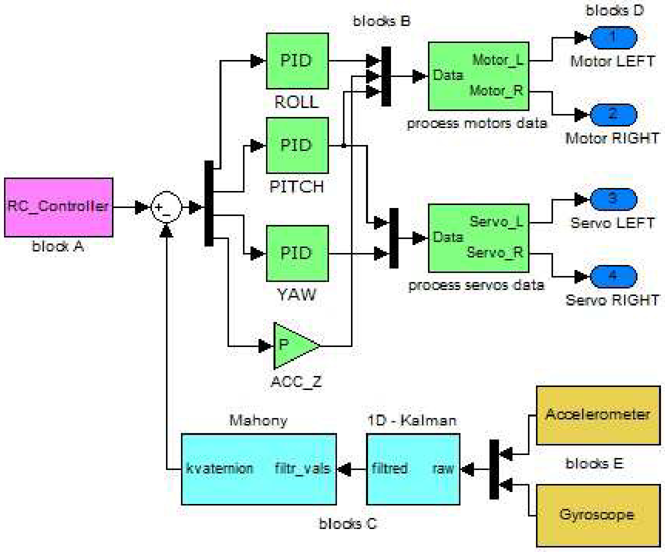

Fig. 3. Basic control loops of the bicopter

IV. Dual microprocessor scheme

The bicopter is equipped with two MCUs. As a DSP processor an Atmel AtMega2560 was chosen. It is a 8-bit microprocessor clocked at 22.1184MHz, which was provided by Atmel company as test sample. It communicates by i^{2}c bus with the InveSense MPU6050 sensor, which provides actual values of angular speed and acceleration in all three axes. In this processor a 1-D Kalman filter algorithm for raw data, together with IMU data fusion Mahony filter and simple communication protocol, were implemented. The DSP processor produces, as its output, the filtered data with the frequency 360Hz approximately.

Before choosing the main processor, several aspects needed to be considered. Among these the most important is the fact that it is required that the processor is computationally effective in working with floating variables. Hence, 32-bit MCUs are more suitable for this job than 8-bit MCUs. Therefore, the range of available MCUs was narrowed down to processors from the STM32F series, which have high computational effectiveness and relatively low price. Though, the power consumption of such processors is not as low as those form the STM32L series. However, this aspect did not play a key role in the development of the bicopter prototype, as the power consumption of whole control electronics is rather negligible when compared to power consumption of the rotors. Next, as a developer board STM32VL Discovery was chosen. The board is manufactured by ST microelectronics and it contains a 32-bit microprocessor STM32F100RB clocked at 24 Mhz. The processor disposes with enough computational effectiveness and a number of peripherals which are needed to realize the whole construction. Two UART buses are used to communicate with DSP processor and a 433MHz OpenLRS module with configurable and servable mobile application on the transmitting side in transmitter. Eight-pulse width modulation (PWM) channels are used, where four of them are used as inputs and the other four as outputs. Three channels are physically connected to the analog accelerometer Freescale MMA7260, which however is not used at the moment. One ADC channel is used to measure the voltage of the battery and one GPIO pin controls signal LED.

The complete communication loop, with the groups of the respective blocks, is shown in Fig. 4.

V. 1-D Kalman filter tuning

In this section, the application of 1-D Kalman filter is discussed. The theory and algorithms of 1-D Kalman filters for microprocessors are well known and described in plenty of works, see e.g. [3]. In general, analogous settings are suggested in these works.

The behavior of the Kalman filter can be tuned by the following two variables:

R – covariance of processing noise

Q – covariance of measuring noise

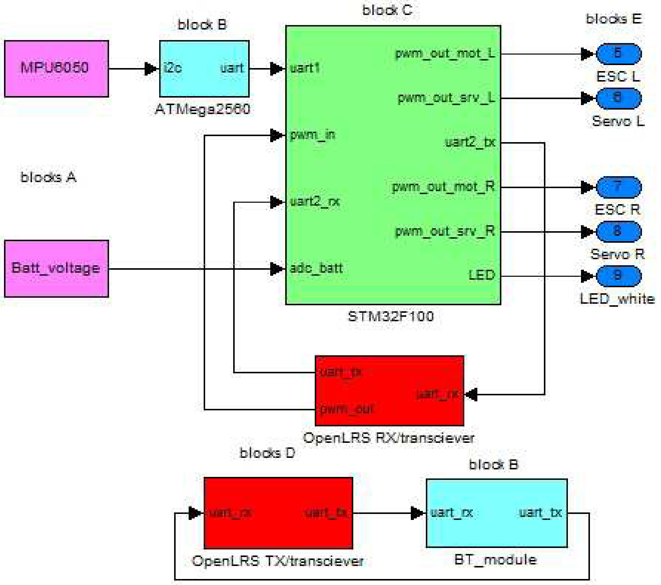

Fig. 4. Stabilization system communication loop

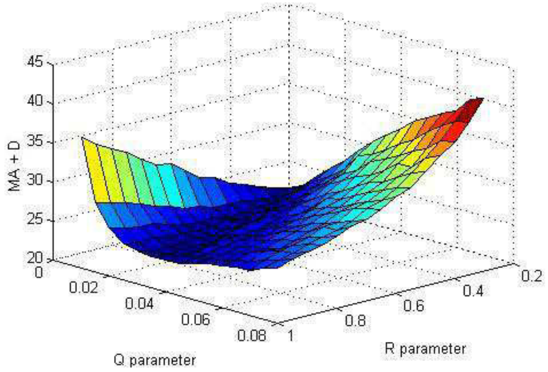

Fig. 5. Output of tuning algorithm for 1-D Kalman filter

To find the optimal values of the two parameters, an algorithm, based on the Mean Absolute (MA) method, was suggested for instance in [7]. Note that the algorithm can easily be implemented in Matlab environment. The MA method itself computes the mean value of the absolute middle error [8]. Then, the sum of the time derivation of the filtered signal rates the overall setting numerically. The outcome of the algorithm is presented graphically in Fig. 5. Note that the lower values actually correspond to better filtration.

The quality of the filtration algorithm can be commented as follows. The MA method itself compares the delay of the filtered signal towards the original signal.

$$MA = \frac{1}{T}\sum_{k=0}^{T-1}\left|\bar{y}_k - h(x_k)\right|$$

However, it cannot just by itself describe the amount of the noise in the respective signal. Due to this reason, it is necessary to add a factor which can describe the noise. The sum of the time derivation of the filtered signal does so.

$$sum(D_{error}) = \sum \frac{d\overline{y}_k}{dt} = \sum_{k=2}^{n} \left| \overline{y}_k - \overline{y}_{k-1} \right|$$

Note that the output of the two methods is dimensionless number, and naturally the lower values correspond to the better result. The final algorithm is therefore sum of the two.

$$(MA + D)_{error} = \frac{1}{T} \sum_{k=0}^{T-1} \left| \overline{y}_k - h(x_k) \right| + \sum \frac{d\overline{y}_k}{dt}$$

To find the optimal parameters, the algorithm ran repeatedly in Matlab environment. More specifically, as the input signal to the algorithm the function sinus from interval <0, 10 *π* >, with the sample period 0.02 *s*, was used. Additionally, the signal was artificially affected by the white Gaussian noise with SNR value = 20 dB. These parameters were selected based on real behavior of machine during actual flight. Sinusoidal signal represents machine vibration around pitch and roll axes during flight. Noise with intensity of 20dB represents vibration caused by motors. Since this can be considered as a random noise for each of the parameters R and Q from the matrix [Qmin:Qmax, Rmin:Rmax], the simulation ran repeatedly several times (e.g. 20 times), and the final values were then given by an average of these simulations.

The output of the algorithm is shown in Fig. 5 where it can be seen that the optimal setting of 1-D Kalman filter is not given by a specific values of R and Q but rather as the ratio between the values R and Q.

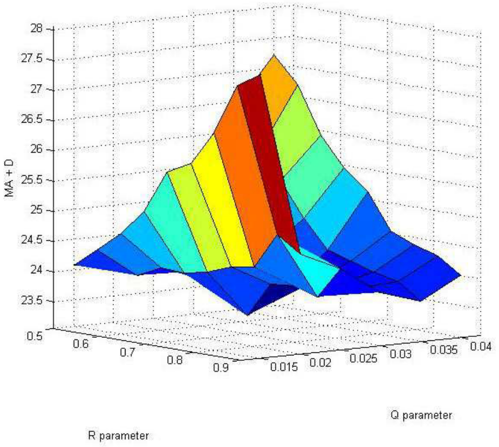

Fig. 6. Local extreme found during the tuning of 1-D Kalman filter

From the simulations one can find local extreme in the point Q = 0.025, R = 0.700. At this point the ratio R/Q reads 28. Note that this ratio is in a good agreement with the values that can be found in many works, see e.g. [9], [10], [11], in which the respective parameters and the ratio are Q = 0.022, R = 0.617, and R/Q = 28.04545.

```
int kalman_filter(int z_measured) {
   static float x_est_last = 0;
   static float P_last = 0;
   float K;
   float P;
   float P_temp;
   float x_temp_est;
   float x_est;
   float Q = 0.025;
   float R = 0.7;

   // prediction
   x_temp_est = x_est_last;
   P_temp = P_last + Q;

   // calculate Kalman gain
   K = P_temp * (1.0/(P_temp + R));

   // correction
   x_est = x_temp_est + K * ((float)z_measured -
   x_temp_est);

   P = (1- K) * P_temp;
   // last value actualization
   P_last = P;
   x_est_last = x_est;

   return (int)x_est;
}
```

Fig. 7. Example of the C code of the 1-D Kalman filter

Finally, the tuned 1-D Kalman filter was used in practice, and the measurement showed that the static noise was reduced by 78% [7].

VI. Control process

During the development of the bicopter the basic control structure, as shown in Fig. 4, underwent various changes and modifications in order to improve the final control process of the real aircraft. It is also supposed that in future additional sensors, e.g. barometer, and controllers will be added. In this section we discuss in more details the process of tuning the parameters of the PID and P controllers shown in Fig. 4, which depicts the basic control loops of the bicopter.

To tune up the parameters of the PID controllers, mainly experimental methods (e.g. a trial-error method) were applied. Basically, these methods were employed during the bicopter flight, and for this purpose a mobile application, providing a real-time communication with the main MCU, was developed. In order to adapt the respective parameters of the PID controllers, the application reads the actual values of the parameters and possibly modifies them in real time.

Since the bicopter time constants are relatively low, which in turns implies that the necessary control actions need to be rather quick, the final setting of the controllers is more PD-like than PID-like. The parameters of the integral actions in the PID controllers are typically 100 times lower than those of the proportional and derivative actions. Nevertheless, the integral actions are present and necessary, as they actually help to balance slight inaccuracies in the rotors power and the stability of the bicopter itself in the steady state.

Finally, note that using the standard PID controllers only, with the parameters tuned up, resulted in the capability of the bicopter to fly satisfactorily. However, quite significant changes in the overall bicopter behavior occured depending on whether the aircraft was flying in low altitudes (where the ground effect affects the bicopter) or not. In conclusion, the setting of the controllers parameters was always a compromise between certain oscillations present either in low or higher altitudes.

VII. N- PID CONTROLLER

In comparison to standrd PID controllers, a use of the so-called N-PID controllers [8] can further improve the bicopter performance during the flight. This is shortly discussed in what follows. The principal structure and the typical use of the N-PID controller is shown in Fig. 7. As can be seen, the block *k*, located before the very PID controller, brings certain nonlinearity to the control process. Though, it slightly influences the P and I part of the controller it affects quite significantly the D part of the controller which can cause undesired effects on the final performance of the bicopter.

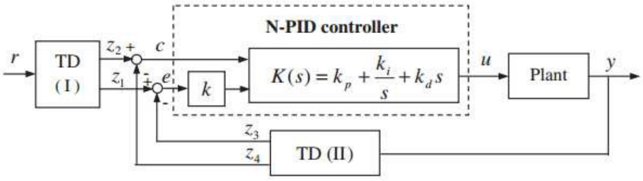

Fig. 8. Design of NE-PID controller from [8]

Therefore, the following modification of the N-PID controller structure was suggested (Fig. 8).

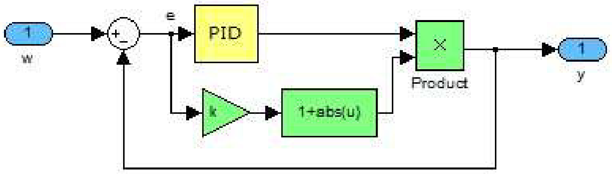

Fig. 9. Modified structure of N-PID controller

The main difference is that in the modified structure of the N-PID controller the block that brings nonlinearity to the control process was placed after the PID controller itself. That is, the N-PID controller output now reads

$$y(t) = PID_{out}(t)(1 + |k.e(t)|)\text{ , for } k > 0 .$$

In conclusions, such a modification of the PID controller acts, in principle, more aggressive in the areas of bigger control error *e*, while on the other hand it reduces to the standard linear PID controller when *e* is relatively small. Due to the constant *k*, which was again tuned up experimentally, the overall control and stabilization process possesses certain robustness and higher stability in the area of nearly zero control error *e*.

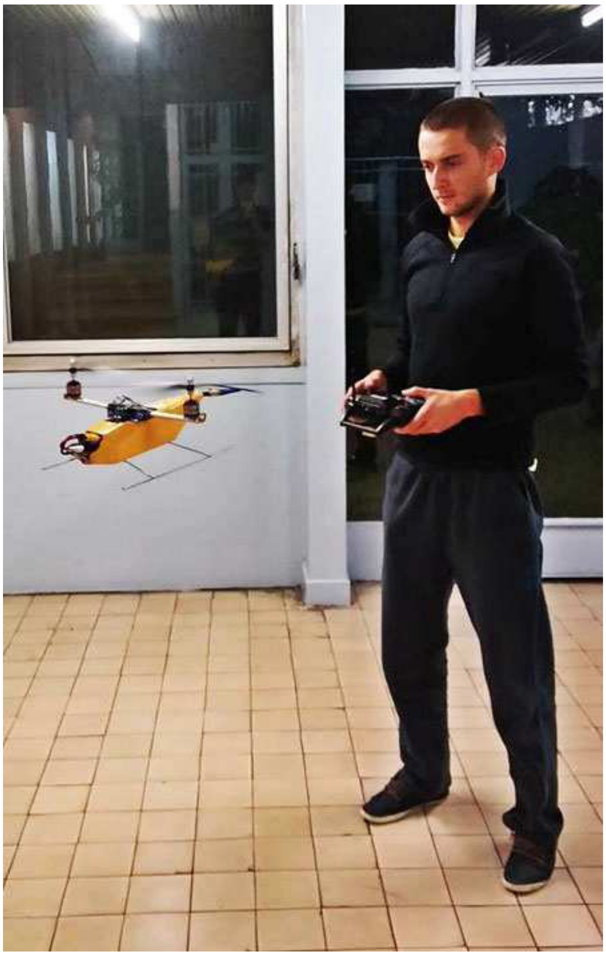

Fig. 10. Demonstration of the stabilization process in practice

VIII. CONCLUSIONS

This paper discussed the process of design, construction, stabilization and control of the homemade bicopter. During this process, more than 50 static tests (approximately 3 hours of a flight time) were performed in a secure room with the aircraft fastened to the ceiling by elastic hinge. Later, the hinge was removed and additional tests were performed outside the secure area. Then, the bicopter was tested more than 5 hours, mainly in closed areas and halls so far. More details about the construction, technology implemented, and some videos can be found on the project webpage [12].

Considering the process of the stabilization of the bicopter discussed in this paper, it is working as expected and the results are promising. The controller of Roll axis works well, however, the certain inaccuracies are still present when controlling Pitch axis. In such a case the control process displays slight angle error during the swift air maneuvers.

Our final comment is related to the use of N-PID controllers which, in comparison to the standard PID controllers, almost completely removed the changes in the bicopter behavior in different altitudes. The overall control process and performance during the flight are more stable, including improved ability to handle unexpected situations and changes of air flow.

ACKNOWLEDGMENT

This work has been supported by the Slovak Grant Agency VEGA, grant No.1/0276/14.

REFERENCES

[1] Kazusuke Maenaka, “ MEMS Inertial Sensors and Their Applications” Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on

[2] Jaw-Kuen Shiau*, Chen-Xuan Huang and Ming-Yu Chang, “ Noise Characteristics of MEMS Gyro’s Null Drift and Temperature Compensation,” Journal of Applied Science and Engineering, Vol. 15, No. 3, pp. 239 246 (2012).

[3] R. E. Kalman, “A new approach to linear filtering and prediction problems,” ASME Trans., J. Basic Eng., ser. D., vol. 82, pp. 35–45, 1960.

[4] Robert Mahony, Tarek Hamel and Jean-Michel Pflimlin, “ Nonlinear Complementary Filters on the Special Orthogonal Group,” IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 53, NO. 5, JUNE 2008.

[5] William Premerlani, Paul Bizard, “ Direction Cosine Matrix IMU: Theory.

[6] Quinton J. Krueger, “Electromagnetic Interference and Radio Frequency Interference Shielding of Carbon-Filled Conductive Resins,” Bachelor of Science, Michigan Technological University, 2001

[7] Lukáš Hrečko, “Spracovanie signálov z MEMS gyroskopov,” bakalársky projekt, Slovak University of Technology, 2014

[8] Y.X. Su, Dong Sun , B.Y. Duan , “ Design of an enhanced nonlinear PID controller,” Mechatronics 15 (2005) 1005–102

[9] https://ideone.com/fork/6ktzh0

[10] http://cboard.cprogramming.com/tech-board/151388-kalman-filter-xy-streams.html

[11] http://wenku.baidu.com/view/a583e254ad02de80d4d840c8.html

[12] Project webpage: http://www.project-kraken.net