

# Kontroler lotu dla bezzałogowych obiektów latających

Przemysław Strzelczyk, Krystyna Macek-Kamińska

Politechnika Opolska, Wydział Elektrotechniki, Automatyki i Informatyki, ul. Sosnkowskiego 31, 45-272 Opole

**Streszczenie:** W artykule przedstawiono autorskie rozwiązanie kontrolera lotu przeznaczonego do sterowania bezzałogowego obiektu latającego. Omówiono schemat blokowy i opisano podstawowe elementy układu sterowania oraz ich zasadę działania. Zaprezentowano wybrane elementy autorskiego systemu operacyjnego czasu rzeczywistego, odpowiedzialne za kontrolę urządzeń układu sterowania oraz komunikację bezprzewodową z układem kontroli lotu.

**Słowa kluczowe:** kontroler lotu, układ sterowania, dron

## 1. Wprowadzenie

Idea bezzałogowych obiektów latających nie jest nowa. Datuje się ją już na połowę 1800 r., kiedy Australijczycy wysłali po raz pierwszy wypełnione bombami balony w kierunku Wenecji. Znane z mediów pojęcie obiektu bezzałogowego jest w ostatnim czasie zastępowane słowem dron (ang. *drone*). Drony pojawiały się już w czasie I i II wojny światowej. Obecnie obiekty bezzałogowe zdominowały przemysł modelarski, filmowy i masowo wkroczyły do przemysłu zbrojeniowego. Odpowiednio zaprojektowany system stabilizacji lotu pozwala na bardzo precyzyjne nagrywanie obrazu nawet z dużych wysokości. Precyzyjne sterowanie obiektem latającym nawet w najbardziej sprzyjających warunkach atmosferycznych jest niemożliwe bez odpowiednio zaprojektowanego kontrolera stabilizacji lotu. Obiekty latające wyposażane są często w kamery termowizyjne, kamery o bardzo dużej rozdzielczości (kamery DARPA mają rozdzielczość do 1,8 Gpx). Dzięki temu możliwe jest fotografowanie szczegółów nawet z dużej odległości od obiektu. Kamery 3D pozwalają budować mapy przestrzenne wewnątrz i na zewnątrz budynków. Wojskowe drony często wyposażane są dodatkowo w różnego rodzaju uzbrojenie. Głównym atutem bezzałogowych obiektów latających jest szeroka gama ich wielkości, zależnie od planowanego zastosowania, oraz eliminacja zagrożenia życia załogi.

Obecnie wiele ośrodków naukowych, zarówno państwowych jak i prywatnych, prowadzi badania dotyczące możliwości wykorzystania obiektów bezzałogowych w różnych dziedzinach życia. Ich układy sterowania coraz częściej wyposażane są w algorytmy sztucznej inteligencji, pozwalające na przykład na naukę podążania po wyznaczonej w przestrzeni trajektorii z umiejętnością samodzielnego omijania przeszkód terenowych [3]. Zaimple-

mentowane mechanizmy przetwarzania obrazu wraz z informacjami pochodzącymi z różnego typu czujników rozmieszczonych w obiekcie, przy zastosowaniu złożonych algorytmów ich interpretacji, umożliwiają analizę otoczenia i szybką reakcję na czynniki zewnętrzne.

Prezentowana realizacja układu kontrolera lotu, bazującego na autorskim systemie operacyjnym czasu rzeczywistego, wymagała opracowania oprogramowania odpowiedzialnego za kontrolę urządzeń peryferyjnych, takich jak czujniki przyspieszenia, żyroskopy, generatory PWM itp., oraz algorytmów przetwarzających pozyskiwane dane. Konieczne było również przygotowanie oprogramowania pozwalającego na komunikację opracowanego układu ze światem zewnętrznym. Pożądane było również zaimplementowanie mechanizmów sprawdzających, czy kontroler nie wykonał nieprzewidzianej operacji i czy program działa prawidłowo. W artykule przedstawiono rozwiązania zastosowane w trakcie realizacji układu stabilizacji i kontroli lotu z wykorzystaniem autorskiego systemu operacyjnego i platformy Raspberry PI.

## 2. Budowa układu kontroli lotu

Schemat blokowy układu kontroli lotu, bez uwzględnienia elementów peryferyjnych takich, jak czujnik GPS (ang. *Global Positioning System*), czujniki ultradźwiękowe, czujniki odległości itp., pokazano na rys. 1. Układ został podzielony na dwa moduły oznaczone symbolami PCB1 oraz PCB2. Jeden z nich kontrolowany jest przez dedykowany do sterowania elementami układu automatyki autorski system operacyjny czasu rzeczywistego, a drugi wykorzystuje powszechnie znany system Raspbian (dystrybucja systemu Linuks, przeznaczona na platformę Raspberry PI) [8]. Podczas budowy drona rozważano różne rozwiązania sprzętowe układu sterowania. Wybrano popularną platformę Raspberry PI. Głównym atutem tego rozwiązania jest bardzo niski koszt oraz niewielkie rozmiary. Zastosowane moduły wyposażone są w elementy takie, jak: złącza USB (ang. *Universal Serial Bus*), złącze HDMI (ang. *High Definition Multimedia Interface*), złącze RJ-45 oraz wyprowadzenia portów procesora. Minikomputer Raspberry PI w wersji 2 oparty jest na układzie Broadcom BCM2835 SoC, składającym się z procesora ARM11 o częstotliwości taktowania 700 MHz, GPU oraz 512 MB pamięci RAM [7]. Moduł zawiera niezbędną liczbę licz-

### Autor korespondujący:

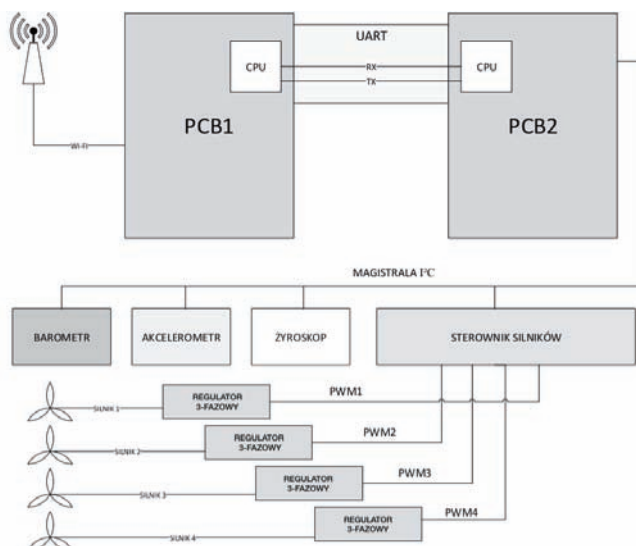
Przemysław Strzelczyk, przemyslawstrzelczyk@gmail.com

### Artykuł recenzowany

nadesłany 6.10.2015 r., przyjęty do druku 3.12.2015 r.



Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0

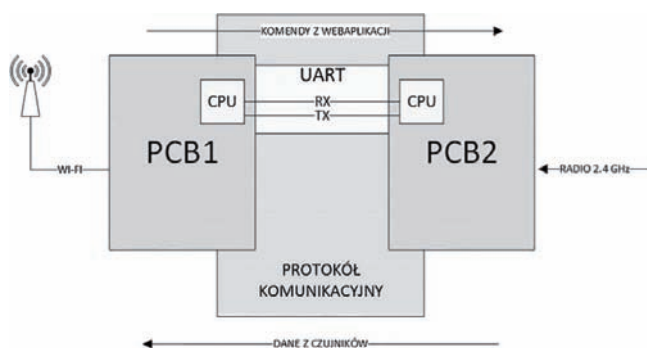


Rys. 1. Schemat budowy układu kontroli lotu  
Fig. 1. Schematic structure of flight control system

ników sprzętowych pozwalających obsłużyć z dużą częstotliwością czujniki takie jak: akcelerometr, żyroskop oraz barometr.

Moduł PCB2 odpowiedzialny jest za kontrolę lotu. Ma zaimplementowane oprogramowanie realizujące filtrowanie danych pomiarowych z czujników. Odczytane wartości są przeliczane na kąt nachylenia obiektu w stosunku do płaszczyzny Ziemi. Dane te następnie przekazywane są do regulatora PID obsługującego daną oś obrotu drona. Regulator odpowiada za dostosowanie sygnału wyjściowego do aktualnych warunków lotu, na które oprócz napędów śmigieł wpływają również warunki zewnętrzne, np. wiatr. Trajektoria lotu zadawana jest zdalnie przez operatora. Dobór nastaw parametrów regulatorów PID w realizowanym projekcie został przeprowadzony zgodnie z metodą Zieglera-Nicholsa. W układzie wymiany informacji (rys. 2) przewidziano możliwość zadania wartości kąta nachylenia w każdej osi obrotu. W opisywanym układzie za komunikację z otoczeniem odpowiada moduł PCB2. Moduł ten wykorzystuje interfejs UART (ang. *Universal Asynchronous Receiver and Transmitter*) do komunikacji z modulem PCB1. Implementacja w module sterującym dodatkowo komunikacji z konsolą operatora powodowała zbyt duże obciążenie modułu, a co za tym idzie, wydłużała czas realizacji algorytmu głównego. Każda komenda przesyłana interfejsem UART ma odpowiednią strukturę odpowiedzi lub ustawień parametrów. Struktura odpowiedzi uzupełniana jest w module PCB2 po przesłaniu zapytania z modułu PCB1. Przykładowo, jeśli występuje konieczność pobrania aktualnych wartości parametrów regulatora PID, to należy najpierw wysłać komendę zawartą w strukturze `struct_cmd_query`, a następnie oczekiwać na odpowiedź o strukturze `struct_get_pid_values`, zawierającej wartości wszystkich parametrów. Podczas projektowania mechanizmu wymiany danych zwrócono szczególną uwagę na różne wyrównania bajtów CPU (układ PCB1 i konsola operatora). Aby uzyskać poprawność i spójność danych zarówno podczas wysyłania jak i odczytu, wymuszono z góry zdefiniowane wyrównanie danych typów zgodnie z zasadą obowiązującą w procesorze modułu PCB1. Wymuszenie następuje po stronie układu PCB1, ponieważ założono, że oprogramowanie zastosowane w tym układzie, ze względu na swoją niskopoziomowość, pozwala na większe uprzywilejowanie i konfigurowalność sprzętu niż oprogramowanie konsoli operatora.

Podczas przesyłania danych sprawdzana jest ich poprawność. W projekcie do sprawdzania poprawności transmisji wykorzystano kontrolę parzystości.

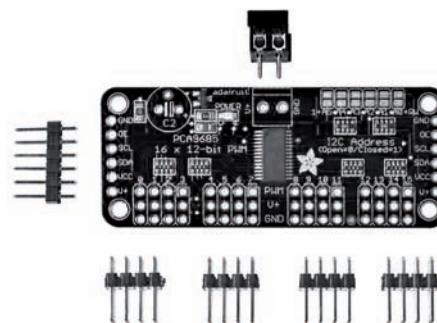


Rys. 2. Układ wymiany informacji między modułami PCB1 i PCB2  
Fig. 2. Information exchange system between PCB1 and PCB2 modules

Końcowym elementem toru komunikacyjnego jest moduł PCB1. Moduł ten wyposażony został w autorski serwer WWW. Serwer WWW udostępnia serwis obsługujący zapytania z aplikacji webowej (aplikacji kontrolera lotu). Aplikacja ta wysyła oraz odbiera wszystkie komunikaty za pomocą technologii AJAX (ang. *Asynchronous JavaScript and XML*) [6]. Po odebraniu zapytania wysyła informacje o odpowiedniej strukturze do modułu PCB2.

### 3. Sterowanie napędami

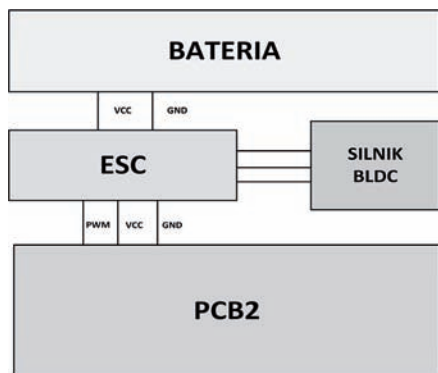
Do płynnej regulacji prędkości obrotowej silników w układzie prototypowym zastosowano modulację szerokości impulsów PWM (ang. *Pulse Width Modulation*). Jest to metoda regulacji sygnału napięciowego lub prądowego o stałej amplitudzie i częstotliwości. Generowanie sygnału PWM może być realizowane programowo lub sprzętowo w mikrokontrolerze, a także za pomocą dedykowanego generatora sprzętowego bazującego na zewnętrznych urządzeniach komunikujących się z procesorem, np. z użyciem interfejsu I<sup>2</sup>C lub SPI.



Rys. 3. Zastosowany w prototypie zewnętrzny generator PWM zarządzany przez interfejs I2C

Fig. 3. External PWM managed by the I2C interface used in the described prototype

W projekcie zastosowano zewnętrzny układ scalony, zawierający 16-kanalowy generator PWM o rozdzielczości 12 bitów (rys. 3). Taka rozdzielczość generatorów PWM jest wystarczająca do ustalania prędkości zastosowanych napędów. Komunikacja z generatorem realizowana jest za pomocą interfejsu I<sup>2</sup>C (ang. *Inter-Integrated Circuit*). Wybrany moduł ma bardzo małą masę (zaledwie 9 g) oraz niewielkie wymiary (62,5 mm × 25,4 mm × 3 mm). Możliwe jest jednoczesne podłączenie do magistrali I<sup>2</sup>C dwóch takich modułów. Sygnały PWM podawane są do regulatorów prędkości silników bezszczotkowych prądu stałego BLDC (ang. *Brushless DC Motor*). Na schemacie blokowym (rys. 4.) moduł ten oznaczono jako ESC (ang. *Electronic Speed Con-*



Rys. 4. Połączenie jednostek napędowych z modułem PCB2  
Fig. 4. The connection of drive units and PCB2 module

troller). Do poprawnej obsługi zewnętrznego generatora PWM została utworzona specjalna klasa umożliwiająca wybór kanału, określenie adresu urządzenia, z którym realizowana jest komunikacja, ustalenie wartości współczynnika wypełnienia sygnału PWM dla każdego z kanałów wyjściowych oraz częstotliwości PWM. W pracy omawianym projekcie częstotliwość pracy generatora została ustalona na 400 Hz.

#### 4. Komunikacja bezprzewodowa z konsolą operatora

Jednym z kluczowych założeń projektu jest możliwość bezprzewodowej komunikacji z obiektem latającym z konsoli operatora. Łączność bezprzewodowa została zrealizowana w sposoby dwóch wariantach.

Pierwszy sposób polega na sterowaniu za pomocą sieci Wi-Fi 2,4 GHz. Możliwość tę wykorzystuje się, jeśli obiekt latający jest w zasięgu takiej sieci. Moduł PCB1 otrzymuje zapytania z aplikacji pełniącej funkcję kontrolera lotu (rys. 5), za pomocą wbudowanego serwera WWW. Po ich przetworzeniu oraz sprawdzeniu poprawności odpowiednie komendy trafiają do modułu PCB2 i zostają wykonane. Aplikacja kontroli lotu pozwala na zamianę parametrów regulatora PID oraz na podgląd wielkości pomiarowych podczas lotu. Wyniki przedstawiane są w formie tekstowej i w postaci wykresów. Operator ma również możliwość podglądu obrazu przekazywanego z kamery zainstalowanej na obiekcie latającym, obsługiwanej przez moduł PCB1 w rozdzielczości HD (ang. *High Definition*). Ze względu na duże wymiary drona, wysoką prędkość obrotową śmigieł oraz relatywnie duży moment obrotowy urządzenie może stanowić zagrożenie dla operatora. W celu ograniczenia ryzyka wypadku wprowadzono trzy tryby lotu – początkujący, średniozaawansowany oraz ekspert. W pierwszych dwóch trybach wprowadzono ograniczenie mocy maksymalnej napędów. Zablokowana została również możliwość edycji parametrów regulatora PID oraz ograniczono dopuszczalne nachylenia obiektu wokół osi X i Y.

Drugi sposób bezprzewodowego sterowania obiektem został uniezależniony od aplikacji kontrolera lotu. Sterowanie obiektem odbywa się za pomocą 9-kanałowej konsoli firmy Hitec. Urządzenie ma wbudowany moduł AFHSS 2,4 GHz. Konsola ma 12-bitową rozdzielczość we wszystkich kanałach. Odbiornik współpracujący z konsolą podłączono do modułu PCB2 za pośrednictwem interfejsu komunikacyjnego UART. Ruchy każdego z drążków sterujących zamieniane są na sygnał o stałej częstotliwości, a wartość nachylenia każdego drążka odpowiada odpowiedniemu współczynnikowi wypełnienia sygnału PWM w module PCB2. Przed rozpoczęciem pracy z aplikacją kontrolera lotu lub z konsolą sterującą, należy odpowiednio wprowadzić hasło operatora w aplikacji kontrolera lotu lub wykonać



Rys. 5. Wygląd aplikacji pełniącej funkcję kontrolera lotu  
Fig. 5. Flight control application

zdefiniowaną kombinację przełączeń w aparaturze sterującej. W ten uniknięty sposób dron jest chroniony przed niepożądanym dostępem.

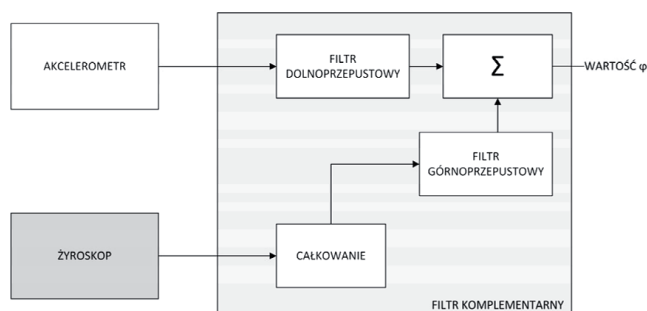
#### 5. Czujniki IMU

Podstawowymi urządzeniami wchodzącymi w skład układu stabilizacji lotu są czujniki IMU (ang. *Inertial Measurement Unit*) – akcelerometr i żyroskop. Na podstawie wyników pomiarów uzyskiwanych z akcelerometru wyznaczane jest położenie przestrzenne obiektu, tj. kąt pochylenia i kąt przekręcenia.

Podczas realizacji projektu zastosowano trzyosiowy akcelerometr ADXL345, który umożliwia jednoczesny pomiar wartości przyspieszeń w trzech osiach. Akcelerometr ten pracuje przy takich samych poziomach napięć sygnałów logicznych, na jakich pracują moduły z mikrokontrolerami. Zastosowany czujnik ma następujące zakresy pomiarowe:  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ ,  $\pm 16g$  oraz rozdzielczość zależną od stopnia czułości, w zakresie 10–13 bitów [1]. Dane wyjściowe z czujnika reprezentowane są w kodzie U2. W związku z tym, operacje dodawania i odejmowania są wykonywane tak samo, jak dla liczb w kodzie binarnym bez znaku, co pozwala ograniczyć liczbę realizowanych rozkazów [3]. Akcelerometr komunikuje się z mikrokontrolerem za pomocą interfejsu I<sup>2</sup>C. W celu ograniczenia poboru energii po stronie procesora odczytującego informacje z akcelerometru dane wyjściowe mogą być przechowywane w 32-poziomowej kolejce FIFO (ang. *First In First Out*). Dzięki zaimplementowanej kolejce FIFO procesor może odczytywać dane z mniejszą częstotliwością i tym sposobem zmniejsza zużycie energii. Zaletą wybranego urządzenia są jego małe wymiary: 3 mm × 5 mm × 1 mm [3]. Każde urządzenie wchodzące w skład systemu kontroli lotu powinno być odpowiednio małe i lekkie, co wpływa na wydłużenie lotu ze względu na mniejsze obciążenie drona.

Żyroskop został wybrany przy uwzględnieniu takich samych kryteriów, jak w przypadku akcelerometru. Urządzenie wykonuje pomiary w trzech osiach. Ma również interfejs I<sup>2</sup>C. Pracuje na tych samych poziomach napięć, jak omówione wcześniej urządzenia. Kolejnym atutem są również jego małe wymiary: 12 mm × 25 mm × 3 mm [4]. Urządzenie zawiera również czujnik temperatury o 8-bitowej rozdzielczości. Wszystkie dane wyjściowe (zmierzone prędkości kątowe) przedstawiane są za pomocą słów 16-bitowych. Moduł ma trzy zakresy pracy: 250/500/2000 dps (ang. *degree per second*). Dodatkowo ma wbudowany rejestr tworzący kolejkę FIFO oraz cechuje się niskim poborem mocy [4]. Obsługę czujników w systemie zapewniają dwie klasy umożliwiające kalibrację przez:

- zebranie określonej liczby próbek i ich uśrednienie oraz określenie poziomu odcięcia szumu,
- pobranie zdefiniowanej liczby próbek oraz wybór urządzenia (w sytuacji gdy zainstalowanych jest kilka urządzeń IMU).



Rys. 6. Budowa filtra komplementarnego  
Fig. 6. Complementary filter scheme

Zastosowanie dwóch różnych typów urządzeń pozwala na korekcję błędów przy wykorzystaniu odpowiednich algorytmów filtrujących. W przypadku żyroskopu dryft zera powoduje systematyczną kumulację błędów odczytu wartości kąta, nawet gdy obiekt znajduje się w spoczynku. Z kolei akcelerometr podatny jest na błąd przesunięcia punktu zero (ang. *zero-g bias level*) [1]. Warto dodać, że poprawa dokładności ustalania pozycji następuje tylko wówczas, gdy obydwa urządzenia charakteryzują inne rodzaje błędów. [2]

## 6. Filtr komplementarny

W celu zapewnienia stabilności obiektu latającego konieczne jest ograniczenie poziomu zakłóceń wielkości pomiarowych. Jest to możliwe przez zaprojektowanie lub zaimplementowanie filtra realizującego fuzję i odpowiednią korekcję danych odczytanych z akcelerometru oraz żyroskopu. Najbardziej rozpowszechnione są filtry Kalmana, Madgwicka i zastosowany w omawianym projekcie filtr komplementarny.

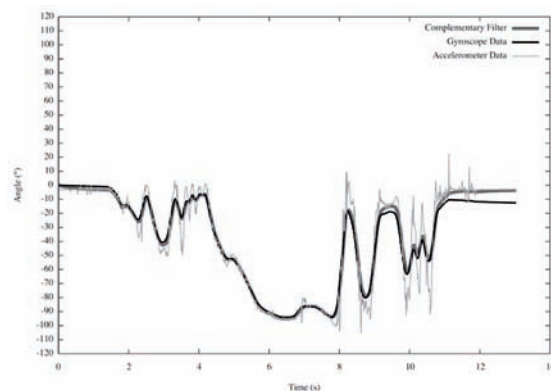
Zastosowanie filtra komplementarnego zostało podyktowane jego dobrą skutecznością przy małej złożoności obliczeniowej, dzięki czemu w znikomym stopniu obciąża mikroprocesor. Działanie filtra komplementarnego polega na połączeniu danych pochodzących z dwóch źródeł, przy jednoczesnym założeniu, że żadne z nich nie udostępnia idealnych odczytów. W przypadku wyznaczania kątów założenie to przedstawiono zależnością (1) [5].

$$angle = 0,98(angle + gyro \cdot dt) + 0,02x\_acc \quad (1)$$

gdzie: *gyro* – aktualna wartość prędkości kątowej odczytana z żyroskopu, *x\_acc* – aktualna wartość kąta obliczonego na podstawie danych z akcelerometru, *dt* – odstęp między kolejnymi próbkami.

Wartości liczbowe 0,98 oraz 0,02 oznaczają procentowy udział w wynikowej wartości zwróconej przez równanie (1) poszczególnych kątów obliczonych na podstawie odczytów z akcelerometru oraz żyroskopu. Filtr komplementarny składa się z dwóch elementów: filtra dolnoprzepustowego oraz filtra górnoprzepustowego.

Filtr dolnoprzepustowy pozwala na korektę szybkozmiennych zakłóceń wprowadzanych przez akcelerometr [5]. Natomiast filtr górnoprzepustowy redukuje błędy wolnozmiennie spowodowane dryftem żyroskopu. W opisywanym projekcie za obsługę filtra odpowiedzialna jest klasa CFilter. Została ona wyposażona w funkcje umożliwiające pobieranie bieżących wartości kątów w każdej z osi obrotu drona. Pobieranie danych i ich filtracja realizowane są w podprogramie obsługi przerwania. W ciele przerwania wykorzystywane są dwie metody klasy CFilter służące do aktualizacji informacji znajdujących się w jej wnętrzu. Generowanie przerwania odpowiedzialnego za filtrację zostało ustawione na taką samą częstotliwość, jak przerwanie



Rys. 7. Wynik działania filtra komplementarnego [5]  
Fig. 7. Complementary filter results [5]

odpowiedzialne za odczyt danych z czujników IMU. W przerwie między przerwaniem procesor przy udziale koprocatora przelicza wartości wszystkich kątów. Na rysunku 7. przedstawiono rezultat działania filtra komplementarnego. Kolorem zielonym oznaczono wartości kąta odczytane na podstawie akcelerometru, a kolorem niebieskim – z żyroskopu. Kolorem czerwonym pokazano sygnał po filtracji. Bez zastosowania algorytmu filtrującego sterowanie oraz stabilizacja układu nie byłyby możliwe.

## 7. Podsumowanie

W artykule przedstawiono projekt autorskiego kontrolera lotu przeznaczonego do sterowania bezzałogowych obiektów latających. Oprogramowanie stworzone na potrzeby niniejszego projektu pozwala operatorowi na zdalną konfigurację parametrów sterowania oraz podgląd wartości odczytanych z czujników. Sterowanie obiektem latającym zostało zrealizowane na dwa sposoby: za pomocą aplikacji pełniącej funkcję kontrolera lotu oraz za pomocą 9-kanalowej konsoli Hitec z wbudowanym modulem AFHSS 2,4 GHz. Wszystkie wymienione założenia projektowe zostały spełnione. Zastosowana architektura oprogramowania pozwala na dalszy rozwój i ewentualne wyspecjalizowanie w żądanym kierunku.

## Bibliografia

1. Analog Devices, *Digital Accelerometer Datasheet ADXL345*, [www.analog.com/static/importedfiles/datasheets/ADXL345.pdf](http://www.analog.com/static/importedfiles/datasheets/ADXL345.pdf), (27.06.2014).
2. Dobliński J., *iMEMS-owe żyro*, <http://ep.com.pl/files/3159.pdf>, (27.06.2014).
3. Hoffmann G.M., Huang H., Waslander S.L., Tomlin C.J., *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*, Navigation and Control Conference, South Carolina, 20–23 August, 2007.
4. *MEMS motion sensor: 3-axis digital output gyroscope*, [www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf), (27.06.2014).
5. Van de Maele P., *Reading a IMU Without Kalman: The Complementary Filter*, [www.pieter-jan.com/node/11](http://www.pieter-jan.com/node/11), (27.06.2014).
6. Batra S., *AJAX – Asynchronous Java Script and XML*, University of Applied Science and Technology Salzburg ITS.
7. Broadcom Europe Ltd., *BCM2835 ARM Peripherals*, [www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf](http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf), (27.06.2014).
8. Raspberry Pi Foundation, *Raspbian Documentation*, [www.raspbian.org/RaspbianDocumentation](http://www.raspbian.org/RaspbianDocumentation), (25.11.2015).

## Flight controller dedicated for unmanned flying objects

**Abstract:** The article presents an original solution of construction the flight controller dedicated for unmanned flying objects. Block diagram has been attached and basic elements of the control system have been described. The principle of individual elements has been also described. Most important elements responsible for the control of the control system devices and wireless communication with the flight control system embedded in authorial RTOS have been presented.

**Keywords:** flight controller, control system, drone

### mgr inż. Przemysław Strzelczyk

przemyslawstrzelczyk@gmail.com

Słuchacz studiów doktoranckich na wydziale Elektrotechniki, Automatyki i Informatyki Politechniki Opolskiej – kierunek Automatyka i Robotyka. Pracuje jako specjalista ds. rozwoju oprogramowania w centrum R&D Nokia Networks.



### dr hab. Inż. Krystyna Macek-Kamińska, prof. PO

k.macek-kaminska@po.opole.pl

Prorektor ds. dydaktyki na Politechnice Opolskiej. Zainteresowania naukowe dotyczą badań symulacyjnych i eksperymentalnych maszyn elektrycznych, metod estymacji parametrów silników indukcyjnych oraz zagadnień automatyzacji napędu elektrycznego.

