AirServer: a Mind-Controlled Assistive Quadrotor Drone Aided by an Intelligent Fuzzy PD Controller

Abdel Ilah N. Alshbatat[1a], Liang Dong[2b], Peter J. Vial[3c]

[1]Department of Communications, Electronics and Computer Engineering, Tafila Technical University, Tafila, Jordan
[2]Department of Computer Engineering, Baylor University, Waco, USA
[3]School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Australia
[a]e-mail: a.alshabatat@ttu.edu.jo
[b]e-mail: liang_dong@baylor.edu
[c]e-mail: peter_vial@uow.edu.au

Abstract— Paralysis is the result of a block in the information pathway between the brain and the limbs. Patients losing bodily control in this way are unable to move as they need to and are, therefore, unable to look after their own needs. The goal of this paper is to design a functioning quadrotor drone that will respond to a patient's brain activity and accordingly enables them to have normal daily functions. We have designed an innovative brain computer interface (BCI) system to control the drone using only the power of thought. The drone has been designed and built using commercial components. An Emotiv EPOC headset was used to gather brain activity and communicate it to the computer which uses Emotiv software and a translation program to convert the signal pattern into a command that is able to be read by an OpenPicus FlyPort module installed on the quadrotor drone. Due to the non-linear nature of the quadrotor, an innovative control law was derived using the Fuzzy Proportional Derivative (FPD) technique. A complete simulation was used to tune the controllers in MATLAB Simulink. The controllers were designed and implemented using on-board microcontrollers and an inertial measurement system. The entire system was tested and verified in an actual flight test. The findings indicate the potential of BCI system for controlling quadrotor, and thus enabling paralyzed people to improve their life and maximize communication capabilities and independence.

Keywords— Assistive technology, Brain computer interface, Electroencephalogram, Fuzzy logic controller, PD controller, Quadrotor drone, Unmanned aerial vehicle.

I.   INTRODUCTION

People who cannot use their limbs are, obviously, unable to control objects in their environment. This is often caused by injury to the motor cortex and frequently occurs after a stroke. Previous investigations have reported that although patients often regain some of their motor function after therapy, most remain chronically disabled [1]. Assistive technologies that translate thought into action can help such people to improve their life and maximize communication capabilities and independence [2]-[4]. Recently, there has been much interest in developing a BCI technology to enable disabled people to directly control a drone using their neural signals [5], [6]. The use of this promising technology (AirServer) is more complex and has only recently started to be studied.

AirServer is an intelligent unmanned aerial vehicle (UAV) that automatically responds to a user's brain activity. The drone is smart and autonomous; and it does not need a remote control to fly. It is especially designed for receiving the command from the user's brain by reading electrical signals through the scalp. AirServer requires that the user wears an Electroencephalography (EEG) cap and learns to move a virtual object back, forth, left and right on a computer screen through thinking alone. The BCI system associates those patterns to specific commands and relays them to the AirServer via a Wi-Fi module. An onboard flight controller will receive those commands, and, depending on the type of brain activity detected, guide the drone to the desired location.

A BCI is a communication system which provides a link between the human brain and external devices. This technology allows a person to control electrical objects using their thought only [7]-[11]. The process begins when specific neurons in the motor cortex produce an electrical current which is detected by the electroencephalography (EEG) headset [12], [13]. The signal from the EEG is transmitted to the computer which translates the signal pattern into commands and relays it to the object wirelessly. The primary challenge in translating the signal pattern into commands is to develop software that is able to control the object without having to use any remote control or having any other human interaction with it. Most current BCI systems obtain the relevant information from the brain activity through the EEG. EEG is the recording of electrical signals from the brain along the scalp [14]. These signals are the sum of the synchronous electrical activities of millions of neurons. Since the electrical activity of a single neuron is very weak, EEG signals exhibit a resonance in the range of (1-100Hz). This range is divided into several intervals: Delta (up to 4Hz), Theta (4-8Hz), Alpha or Mu (8-12Hz), Beta (12-30Hz), and Gamma (30-100Hz). Each interval reflects specific activities inside the brain. In addition to these rhythms, positive peaks resulting from infrequent stimulus (P300) [15], slow cortical potentials (SCP), visual evoked potentials (VEP), and steady state VEP (SSVEP) can also be recorded [16]. EEG signal interpretation depends on the position of electrodes on the scalp. The spatial distribution of electrodes on the scalp follows international standards 10-20. The actual distances between adjacent electrodes are either 10% or 20% of the total front-back or right-left distance on the skull. The electrodes are named according to the location on the scalp and given a letter that reflects this position. The letters F, T, C, P and O stand for Frontal, Temporal, Central, Parietal, and Occipital lobes, respectively.

There has been no practical system that allows the drone to be fully controlled using the power of thought. The goal of this paper is to develop an assistive drone that can be steered by the electrical activity of the brain using external electrodes attached to the patient's scalp. An innovative BCI system was developed to control the drone using the power of thought only. A drone was designed and built using commercial components. In this study, we have restricted our work to three cases: arming/disarming the drone, taking off or landing, and allowing the drone during emergency situations to fly towards a pre-defined safe area. The last case required that the drone should be preprogrammed with up-to-date information. For future work, the latter case will be replaced with an on-board sensor to quickly identify a suitable landing area.

The main contributions of the paper are (i) Designing an innovative brain computer interface (BCI) system that will respond to patients' brain activity, and produce a signal commanding the quadrotor to perform some tasks. (ii) Designing a functioning quadrotor drone that will respond to the command received from the BCI system (iii) Designing a Fuzzy Proportional Derivative (FPD) technique that is capable of stabilizing quadrotor and is adaptive to the induced disturbance and uncertainty.

The remainder of this paper is organized as follows. Section II outlines the functional model of the AirServer system. The hardware design and the working principle of the AirServer are described in section III. In section IV, we detail the software design of the AirServer. In Section V, we present simulation and experimental results as well as the performance of the brain commands when the system is subjected to four commands: up, down, left, and right. Finally, we summarize the main results of the work in section VI.

II.   FUNCTIONAL MODEL OF THE AIRSERVER SYSTEM

As shown in Fig. 1, the AirServer combines the use of a brain computer interface and an unmanned aerial vehicle. The unmanned aerial vehicle was built from scratch and has a custom printed circuit board (PCB) attached to its hop. The PCB contains components that are specifically used for receiving, interpreting and processing signals from the BCI system, and transmitting the correct voltages to the flight controller. The Emotiv headset and customized software are the keys to the BCI system. The software developed to connect the BCI system with the drone was written based on the event-driven and client-server mechanisms. A Wi-Fi connection was used and implemented on a FlyPort module. The server utilizes the signals received from the headset and processes them into usable signals for the drone. Arming/disarming the drone is determined by the position of the user's head. For example, the user can arm the drone by tilting his head left. If the user wishes to disarm the drone, he simply tilts his head to the right. The direction in which the drone moves is determined by the thought of the user.
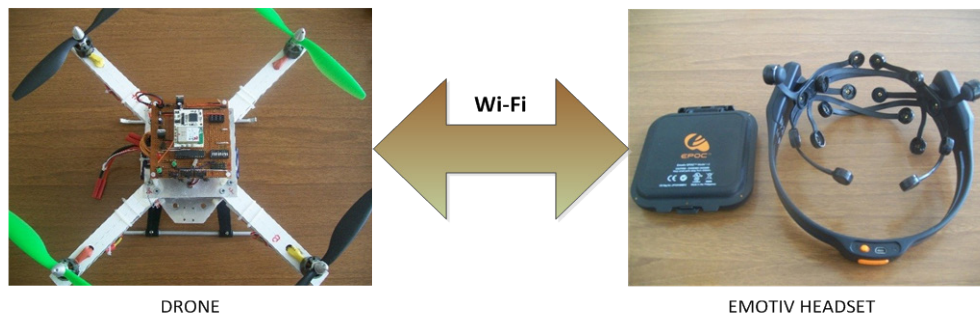

Fig. 1. Structure of the AirServer System

III.   HARDWARE DESIGN OF THE AIRSERVER

A brain computer interface provides a communication link between the human brain and the drone. The system translates thought into action without using muscles, thus enabling disabled people to improve their independence and maximize their communication capabilities [17]. The AirServer system includes five major parts: the Emotiv EPOC neuroheadset, a wireless USB receiver, a computer, a quadrotor drone, and two custom PCBs, one attached to the computer and the other located on the drone. The Emotiv EPOC headset detects and records neuronal electrical activity that reflects the user's intent from different locations on the scalp. That activity is transmitted via Bluetooth to the computer to extract specific signal features. Those features are translated into commands to operate the drone.

A.   Emotiv Neuroheadset

The Emotiv EPOC headset is used to collect EEG data. This data is transmitted wirelessly to a computer; the processed data is used to control the drone. As shown in Fig. 2, the headset contains fourteen saline felt sensors and two reference sensors. The felt sensors are divided into 7 pairs (AF3 and AF4, F3 and F4, F7 and F8, FC5 and FC6, T7 and T8, P7 and P8, O1 and O2) and must be moist all times. Each pair is placed over a lobe of the brain and named by the first letter of the lobe. Table1 summarizes the number given to each sensor, its name and the name of the brain location. Another hardware component comes with the headset: the

Emotiv USB Bluetooth receiver. The USB receiver is used to receive the information from the headset. This information is sent to the computer for processing.
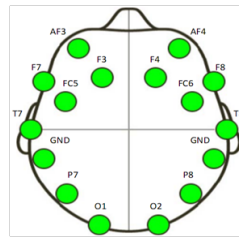

Fig. 2. Location of the sixteen channels by Emotiv EPOC

TABLE 1
NAMES OF EMOTIV SENSORS AND THEIR LOCATIONS ON THE SCALP

Sensor #	Sensor Name	Area of the Brain
1	AF3	Dorsolateral prefrontal cortex
2	F3	Frontal eye fields
3	AF4	Dorsolateral prefrontal cortex
4	F4	Frontal eye fields
5	F8	Anterior prefrontal cortex
6	FC6	Dorsolateral prefrontal cortex
7	T8	Primary gustatory cortex
8	DRL (reference)	Middle temporal gyrus
9	P8	Primary motor cortex
10	O2	Somatosensory association cortex
11	O1	Somatosensory association cortex
12	P7	Primary motor cortex
13	CMS (reference)	Middle temporal gyrus
14	T7	Primary gustatory cortex
15	F7	Anterior prefrontal cortex
16	FC5	Dorsolateral prefrontal cortex

B. Quadrotor Drone

A quadrotor is a small flying robot with four rotors: front, back, right and left. Front and back rotors rotate in a clockwise direction while the other two rotate counter-clockwise. The thrust of each rotor is adjustable and can be used to gain one of the four degrees of freedom: yaw, roll, pitch, and altitude. The main structure of the quadrotor has several components: frame, rotors, propellers, power source (battery), sensors (IMU), microcontroller, electronic speed controller (ESC), and RC transmitter. As shown in Fig. 3a, the frame is the structure that holds all the components together. It consists of three parts: the center hub where electronics are mounted, four arms mounted to the center hub, and four motor mounts connecting the motors to the end of the arms. The 28 cm long aluminum arms with a thickness of 3x1 cm are used for the construction of the arms because of their light weight. The distance between the centers of the two motors on the same arm is 25 cm.

A flight control board from Hobby King Company was used as shown in Fig. 3c. The board is a combination of an Atmel Mega324PA microcontroller, a 3-axis accelerometer and a 3-axis gyroscope. The accelerometer and gyroscope are used to measure acceleration and angular velocity, respectively. Those measurements allow the Atmel Mega324PA microcontroller to calculate the changes in the motor speed. An ultrasonic sensor was installed on the drone in order to measure the distance to the ground. This enables the sensor to keep the quadrotor at a

certain distance from the patient's head. To control the speed of the brushless motors and the quadrotor, the flight controller board is attached to four Turnigy Plush 30 amp speed controllers (ESC). Each ESC is controlled by a pulse-position modulation (PPM) signal and used to generate three different signals and feed them to the rotors. Four DC brushless motors NTM 28-26 1350KV/310W were used. Finally, four 11X45 inch propellers were mounted on the motors, two pieces for a standard rotation and two pieces for a right-hand rotation.

Two lithium polymer batteries were attached to the drone. The first one is 4000mAh, 11.1 volt and used to power the flight control board, the ECS's and the DC motors through the power distribution board shown in Fig. 3b. The second battery is used to power the printed circuit board. A voltage regulator was also used for the PCB in order to properly power the microcontroller and the FlyPort module.
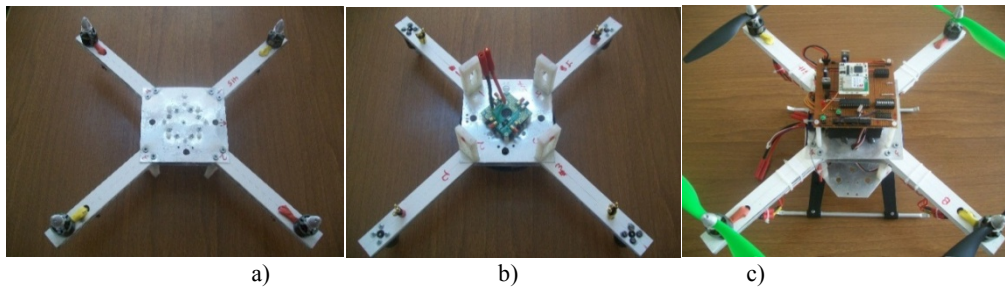

Fig. 3. Structure of the quadrotor: a) 28cm aluminum arms with a 3x1cm, b) power distribution board, c) quadrotor flight controller board

C. Custom Built FlyPort-Based PCB

Two printed circuit boards were created to connect the drone with the BCI system. The first PCB is shown in Fig. 4a. The PCB integrates a FlyPort module and a USB converter. The FlyPort board is a Wi-Fi module based on the microchip technology PIC microcontroller. The board integrates a powerful PIC24FJ256 microcontroller and an 802.11 b/g/n Wi-Fi communication transceiver. It operates on an open source operating system (Free RTOS) and is embedded with a dedicated TCP/IP stack. The FlyPort module also comes with 26 digital pins and a free IDE environment for programming and managing all of its functionalities. The USB converter is based on the FT232RL chip and used to convert a USB connection into 5 volt TX and RX that can be easily connected to the FlyPort module. The converter receives the commands from the computer and feeds them to the FlyPort module. The FlyPort then outputs the commands wirelessly to the module wired to the drone.

Fig. 4b shows the final printed circuit board mounted on the drone with all of the needed components. The PCB integrates a powerful PIC16f876 microcontroller and a FlyPort module. The information that the FlyPort receives from the board attached to the computer is passed to the PIC16f876 microcontroller for processing. The microcontroller then outputs a pulse with modulation signals to the flight control board. To control the motors of the drone, a flight control board located under the PCB is used, as shown in Fig. 3c. That board is based on the Atmel Mega324PA microcontroller and is used to output a set of pulse width modulation signals to the motors. Since each individual motor must be powered by a three-phase signal, an ESC converter is used.
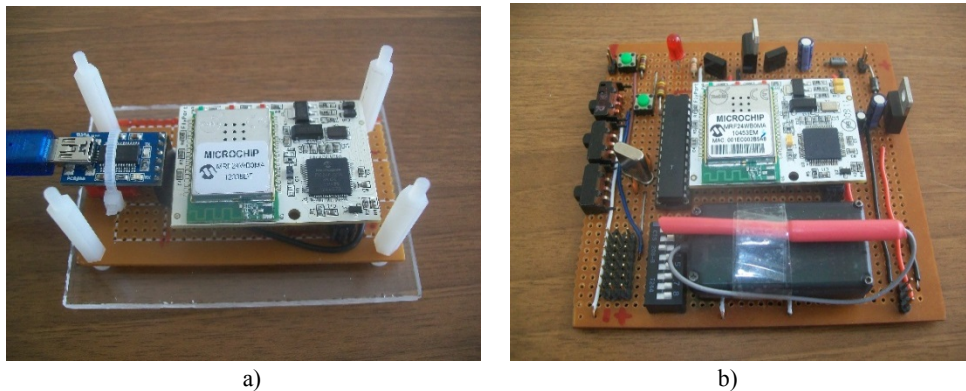
a)   b)

Fig. 4. FlyPort-based PCB a) FlyPort Wi-Fi Module and a USB converter, b) FlyPort Wi-Fi module and a PIC16f876 microcontroller

IV.   SOFTWARE DESIGN OF THE AIRSERVER

In order to establish a connection between the drone and the headset, the software developed in this paper is broken into three parts. The first part resides in a personal computer (PC) and is interfaced to Emotiv software. This software was written using the C# language; and its functions are to receive signals from the Emokey tool distributed by the Emotiv Company, translate them into two characters, and finally send the characters to the PCB shown in Fig. 4a. The second part resides in the FlyPort modules. Two programs were written using C language. The first program is dedicated to the server and resides in the PCB, as shown in Fig. 4a, while the second program is dedicated to the client and resides in the PCB, as shown in Fig. 4b. The software is written based on the event-driven and client-server mechanisms. The client first makes a request to the server, which responds by opening a TCP socket connection. Any events sent by the headset will be processed by Emotiv software and routed to the server which, in turn, sends them to the client that is attached to the drone. The third part was written using C language and loaded onto the PIC16f876 microcontroller that is located in the PCB, as shown in Fig. 4b. The microcontroller implements the fuzzy logic control algorithm and sends the crisp output back to the FlyPort module which, in turn, feeds the flight controller of the drone with PWM signals. Below are the details of the three software components used in this project.

A.   Emotiv Software

Two suites included in the Emotiv SDK were used in this work: Cognitive and Emokey suites. The cognitive suite is grouped with the expressive and affective suites by a tool called the control panel. The EEG raw data from the user scalp is collected, amplified, digitized and transmitted through a Bluetooth module to the computer. The Emotiv software decodes the data, extracts information, and shows EPOC's status. As depicted in Fig. 5, the control panel comes with three tabs dedicated to the cognitive, expressive and affective suites; and it is used to display the contact quality information of each sensor. This also allows users to train and test various thoughts on a virtual 3D cube, and view their emotional state and facial expressions.
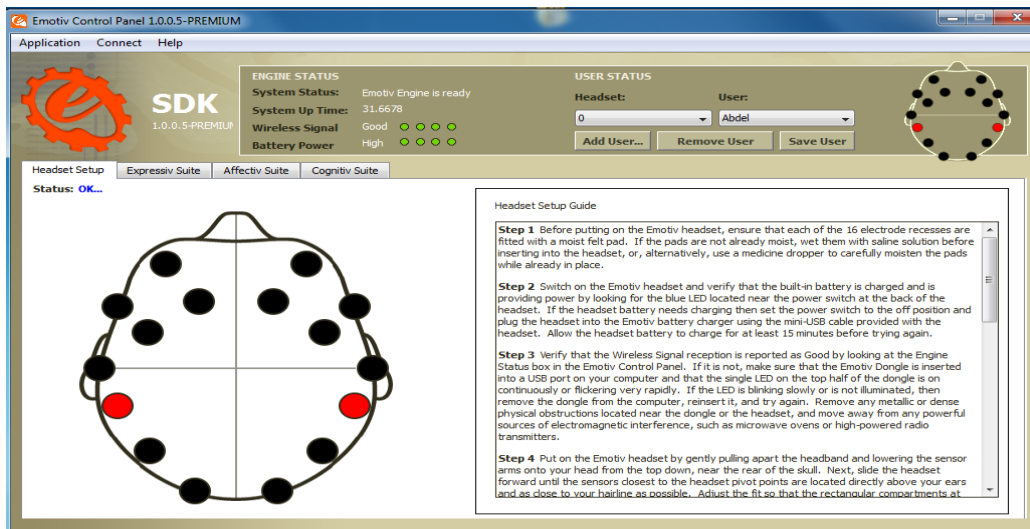
Fig. 5. Screenshot of the control panel showing the location of 16 sensors

The cognitive suite is shown in Fig. 6. The suite detects the user's intentions to perform distinct physical actions on a virtual 3D cube. These intentions are mapped to a pre-trained output. The suite allows the user to choose up to four actions such as push, pull, left, and right movements of the virtual cube. In order to enable detection, the machine must be trained for all the previous actions including the neutral state. Through the use of EmoKey Software, the previous actions can be paired with keystrokes; and the keystrokes can be associated with a certain physical movement. The cognitive signals in this work are used to move the drone up, down, left, and right.
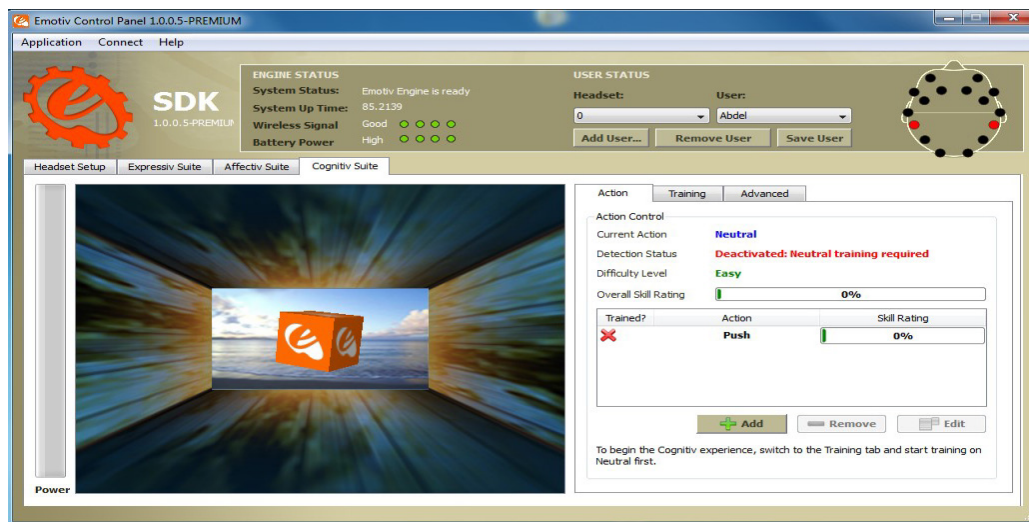

Fig. 6. Screenshot of the cognitive suite

The Emotiv Emokey suite used in this work is shown in Fig. 7. Emokey is used to map the subject's thoughts into keyboard inputs and send them to the input queue of the windows operating system. These inputs are read by a program written for this work and sent as a command through a FlyPort module to control the drone. The program is written using C# language. Its functions are to gain access to the data available at the input queue, translate them into two characters, and send them to USB port, where the FlyPort is connected.
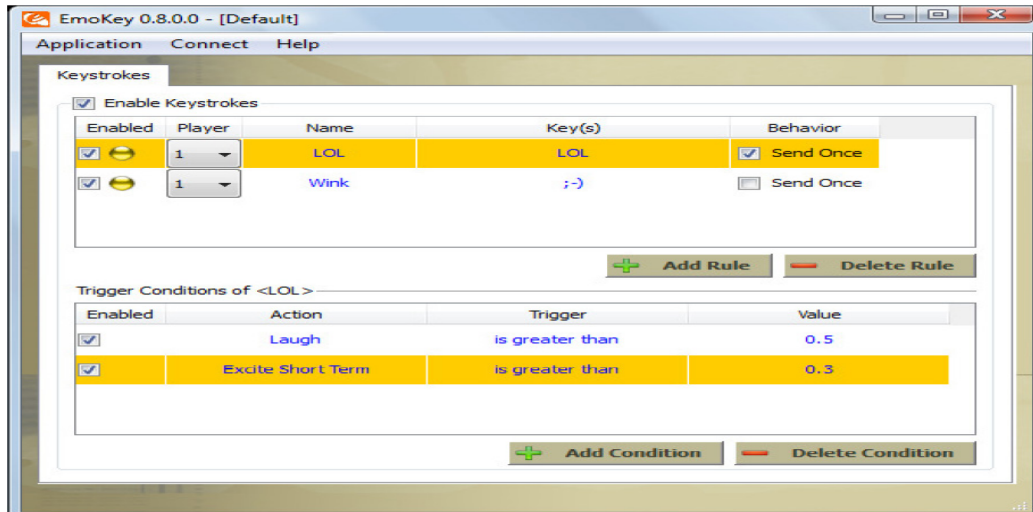
Fig. 7. Emokey mapping for controlling the drone

B. *Event-Driven and Client-Server Mechanisms*

Two blocks of code were written and loaded onto the FlyPort modules using C language. Fig. 8 shows the flow pattern that the two programs follow. The first program is used to gain access to the USB port, read the available data, and send it to the PCB attached to the drone. The software was written based on the Event-Driven and Client-Server mechanisms where each command sent from the headset is considered as an event. Before data transfer between the drone and the computer is initiated, the client first makes a request to the server, which responds accordingly by opening a TCP socket connection. Once the connection has been established, and as soon as the FlyPort module attached to the USB port receives new data from the Emokey, it starts sending them to the drone.

The second program is used to receive the data wirelessly, process them, and send them to the PIC16f876 microcontroller. Ten commands were designed to drive the drone: Arming, Disarming, Taking-off, Landing, Rolling-right, Rolling-left, Pitching-up, Pitching-down, Yawing-right, and Yawing-left. Arming and disarming are determined by the position of the user's head. For example, to arm the drone, the user must tilt his head left. If the user wishes to disarm the drone, he simply tilts his head right. The movement of the drone is determined by the thought of the user. The commands Taking-off, Landing, Rolling-right, Rolling-left, Pitching-up, Pitching-down, Yawing-right, and Yawing-left are triggered by push, pull, right, and left commands generated by the user's head. For example, with double push and double pull commands, the FlyPort module translates those events into a take-off and landing. We assumed that the user accomplished the arming procedure; otherwise the drone would not respond.

As shown in Fig. 9, the FlyPort module embedded with the PCB attached to the drone is programmed to output four PWM signals to the kk2 flight controller board. Commands received from the brain are processed inside this module so as to combine and pass them to the PIC microcontroller. From there, the PIC microcontroller implements the fuzzy logic control algorithm shown in Fig. 10 as a program descriptive language. It also sends the result of the defuzzifier, a numeric value which determines the change in the gains of the PD controller, back to the kk2 flight controller board which, in turn, modifies the duty cycle of the PWM signals that are fed to the DC motors. Once one process is completed, the software waits for the next command to be received from the computer.
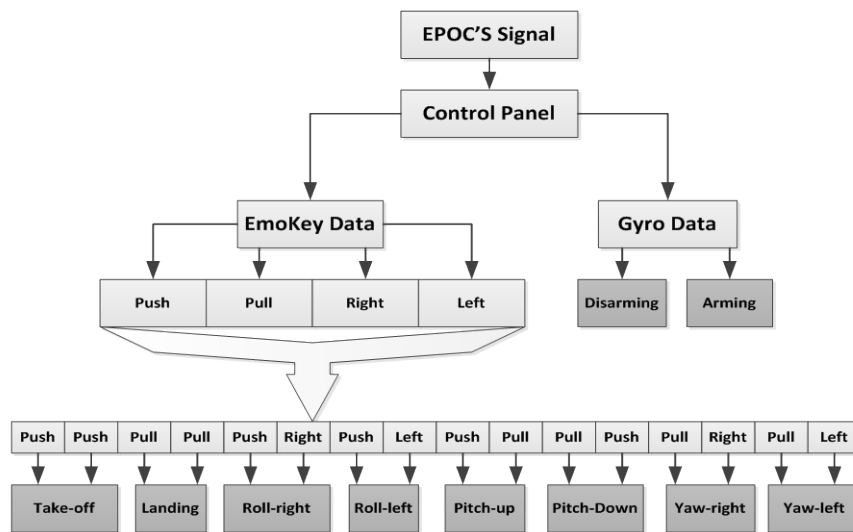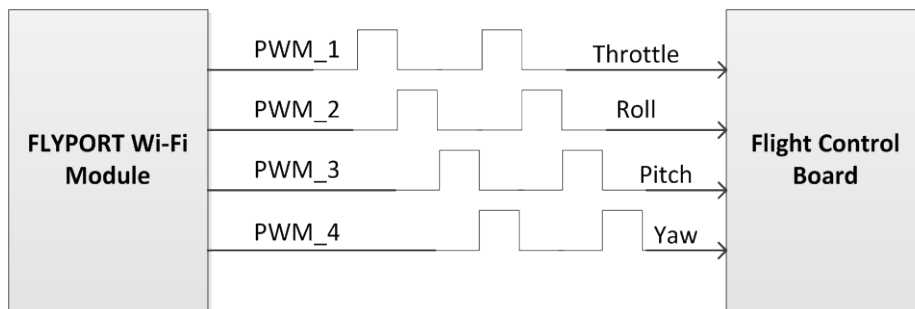
Fig. 8. Event-driven and client-server Architecture

Fig. 9. Communication between FlyPort Module and kk2 flight controller board

Fig. 10. PIC-based fuzzy logic control algorithm

C. Quadrotor Flight Controller Design

As shown in Fig. 11, the quadrotor is a small flying robot with four rotors: front, back, right and left. The thrust generated by the rotors is adjustable and can be used to gain one of the four degrees of freedom: roll, pitch, yaw, and altitude. The PWM signals generated by the microcontrollers are significant in determining how much thrust and torque will be created by the four rotors. In this work, two control loops are used to stabilize the drone, namely the inner and outer loops. The inner loop provides the aircraft with dynamic stability, while the outer loop is used to tune the inner loop, so that the brain commands will not overwhelm the

drone. A flight control board from the Hobby King Company is used in this project to provide the inner loop. The flight control algorithm, which runs on this board, is the proportional derivative (PD) controller. The parameters of the controller will be regulated by the outer loop using the fuzzy logic algorithm.
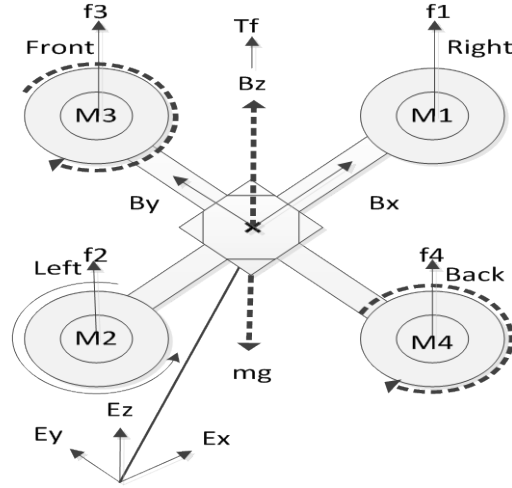

Fig. 11. Quad-rotor schematic including the body and earth frames

C.1. Dynamics of Quadrotor

To model the quadrotor dynamics, two reference frames are defined, namely: the earth inertial frame and the body fixed frame as shown in Fig. 11. The vector describing the linear and the angular positions of the quadrotor is given by:

$$p = [x \quad y \quad z \quad \Phi \quad \theta \quad \psi\ ]^T \tag{1}$$

where Φ, θ , and ψ denote the vehicle's roll, pitch, and yaw angles along the three orthogonal body axes x, y, z, respectively. The vector describing the linear and angular velocities of the quadrotor is given by:

$$v = [u \quad v \quad w \quad p \quad q \quad r\ ]^T \tag{2}$$

where u, v, and w denote the linear velocities of the quadrotor; and p, q, r denote the angular velocities around the roll, pitch, and yaw axes, respectively. Together, (1) and (2) make up the 12-state variables of the quadrotor platform.

The thrust force generated by each motor propeller is given by $f_i = D\, w_i^2$, where D is the thrust factor, and (w_i rad/s) is the angular speed of the rotor (i). The drag torque is given by $\tau_i = Bw_i{}^2$, where B is constant and represents the drag torque factor. Therefore, the net forces acting on the quadrotor airframe with respect to the earth inertial frame is given by:

$$T_f = \sum_{i=1}^{4} fi = D \sum_{i=1}^{4} w_i{}^2 \tag{3}$$

Physically, the PWM signals generated by the microcontrollers are the input variables of the real vehicle. For the purpose of obtaining a transformation model, the following artificial input variables are defined:

$$u_1 = D \sum_{i=1}^{4} w_i{}^2 \tag{4}$$

$$u_2 = Dl(w_3{}^2 - w_4{}^2) \tag{5}$$

$$u_3 = Dl(w_1{}^2 - w_2{}^2) \tag{6}$$

$$u_4 = \frac{B}{D}(f_1 + f_2 - f_3 - f_4) \tag{7}$$

where u_1, u_2, u_3 and u_4 represent the upward force (thrust) generated by the four propellers, rolling torque, pitching torque, and yawing torque, respectively. L is the distance between the motor and the center of gravity. After a certain transformation from one frame to the other, the dynamic behavior of the quadrotor is expressed mathematically as:

$$\ddot{x} = \frac{u_1}{m}(\cos\Phi.\sin\theta.\cos\psi + \sin\Phi.\sin\psi) \tag{8}$$

$$\ddot{y} = \frac{u_1}{m}(\sin\theta.\cos\Phi.\sin\psi - \sin\Phi.\cos\psi) \tag{9}$$

$$\ddot{z} = \frac{u_1}{m}(\cos\Phi.\cos\theta) - g \tag{10}$$

$$\ddot{\Phi} = u_2/jx \tag{11}$$

$$\ddot{\theta} = u_3/jy \tag{12}$$

$$\ddot{\psi} = u_4/jz \tag{13}$$

where $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ represent acceleration components around the x, y, and z axes, respectively. j_i is the rotational inertia around the three axis; m is the total mass of the quadrotor; and finally $\ddot{\Phi}$, $\ddot{\theta}$, and $\ddot{\psi}$ represent the angular acceleration components around the three axes.

C.2.   Fuzzy Logic Flight Controller

Regardless of the controller type, the quadrotor is controlled by varying the angular velocity of the rotors. Two controllers are used in this work to stabilize the drone: the baseline controller and the fuzzy controller. These controllers are used for the inner and the outer loops, respectively. The baseline controller is a proportional derivative (PD) which provides the aircraft with dynamic stability, while the fuzzy controller is used to tune the baseline controller so that the brain commands will not overwhelm the drone. To develop a rule-based fuzzy logic controller, equations (4)-(7) (below) are used. $w_i{}^2$ is replaced with the PWM_i applied to each motor.

$$u_1 = D(PWM_1 + PWM_2 + PWM_3 + PWM_4) \tag{14}$$

$$u_2 = Dl(PWM_3 - PWM_4) \tag{15}$$

$$u_3 = Dl(PWM_1 - PWM_2) \tag{16}$$

$$u_4 = B(PWM_1 + PWM_2 - PWM_3 - PWM_4) \tag{17}$$

As shown in Fig. 12, four Mamdani fuzzy controllers (FLCz, FLCphi, FLCtheta, and FLCpsi), one for each state, are designed to tune the gains of the PD controller Kp, and Kd. All inputs to the controllers are identical and presented as an error E signal and a change of the error signal ΔE, as shown in Equations (18) and (19), respectively. The error signal is the

difference between the desired values (PWM_d) of throttle, roll, pitch, and yaw generated by the brain and the actual values (PWM_a) measured by the IMU system. The change of error signal is the difference between the existing error and the past error. The outputs of the fuzzy controllers are fuzzy proportional gain (FKp) and fuzzy derivative gain (FKd) for each particular state. The goal is to design the Mamdani fuzzy controllers so as to force brain commands (push, pull, right, and left) to converge to their desired values.

$$E(i) = PWM_d(i) - PWM_a(i) \tag{18}$$

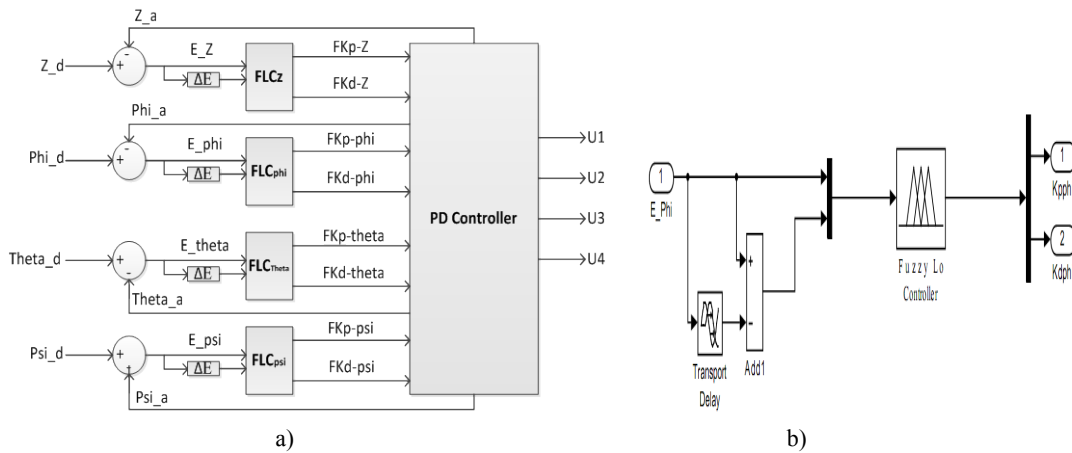$$\Delta E(i) = E(i) - E(i-1) \tag{19}$$

Fig. 12. Basic structure of the controllers, a) fuzzy PD controller, b) FLCphi

To find a relationship between the outputs and the inputs of the fuzzy system, five fuzzy sets are chosen to describe the two inputs (E, ΔE) and the two outputs (Fkp, Fkd). The linguistic variables (NB, NS, Z, PS, PB) describe the two inputs, where NB, NS, Z, PS, and PB denote negative big, negative small, zero, positive small, and positive big, respectively. The values (VS, S, M, B, VB) describe the two outputs, where VS, S, M, B, and VB denote very small, small, medium, big, and very big, respectively.

The membership functions of the inputs and outputs are shown in Fig. 13. The width of the fuzzy sets assigned to input E is from -8 to +8, while the input DE is from -4 to +4. Similarly, the width assigned to the fuzzy sets of output Kp is from 0 to +8, and from 0 to +4 for the output Kd.

As shown in Tables 2 and 3, 25 linguistic rules are used to determine controller gains. The rules can be read as follows: If the error is NB and the change of error is NB, then Kp is M and Kd is M. Hence, the outputs from the system are fuzzy; a center average defuzzifier is used to obtain the crisp outputs and feed them to the PD controller as tuning parameters. Fig. 14 shows the generated surfaces produced by the rules presented above. The overview of the whole system is shown in Fig. 15.
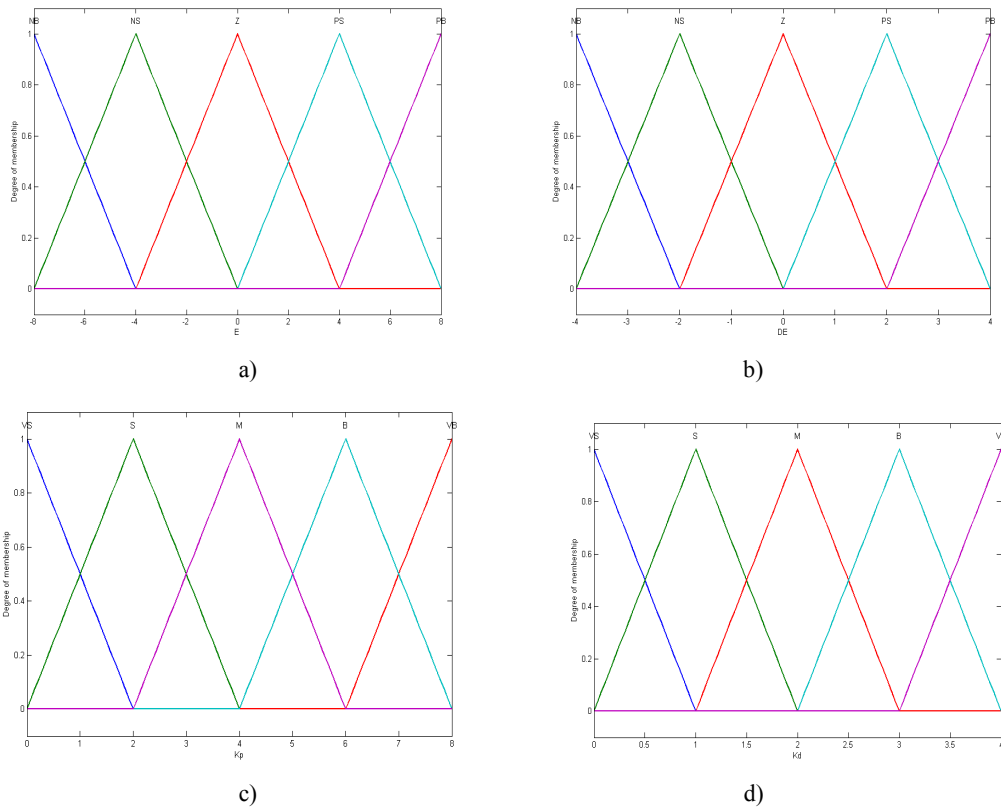
a)

b)

c)

d)

Fig. 13. Membership functions for inputs and outputs. a) E membership function, b) DE membership function, c) Kp membership function, d) Kd membership function

TABLE 2

FUZZY RULES FOR TUNING THE KP GAIN OF THE PD CONTROLLER

		E				
		NB	NS	Z	PS	PB
ΔE	NB	M	S	VS	S	M
	NS	B	M	S	M	B
	Z	VB	B	M	B	VB
	PS	B	M	S	M	B
	PB	M	S	VS	S	M

TABLE 3

FUZZY RULES FOR TUNING THE KD GAIN OF THE PD CONTROLLER

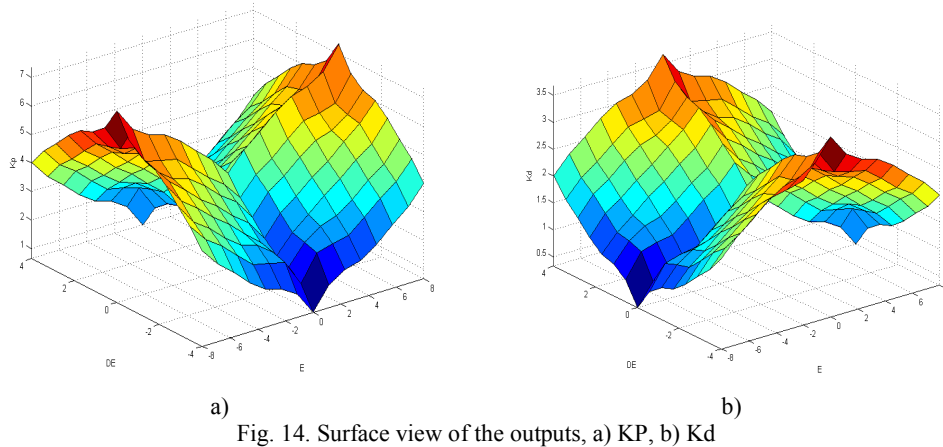|		E				
		NB	NS	Z	PS	PB
ΔE	NB	M	B	VB	B	M
	NS	S	M	B	M	S
	Z	VS	S	M	S	VS
	PS	S	M	B	M	S
	PB	M	B	VB	B	M

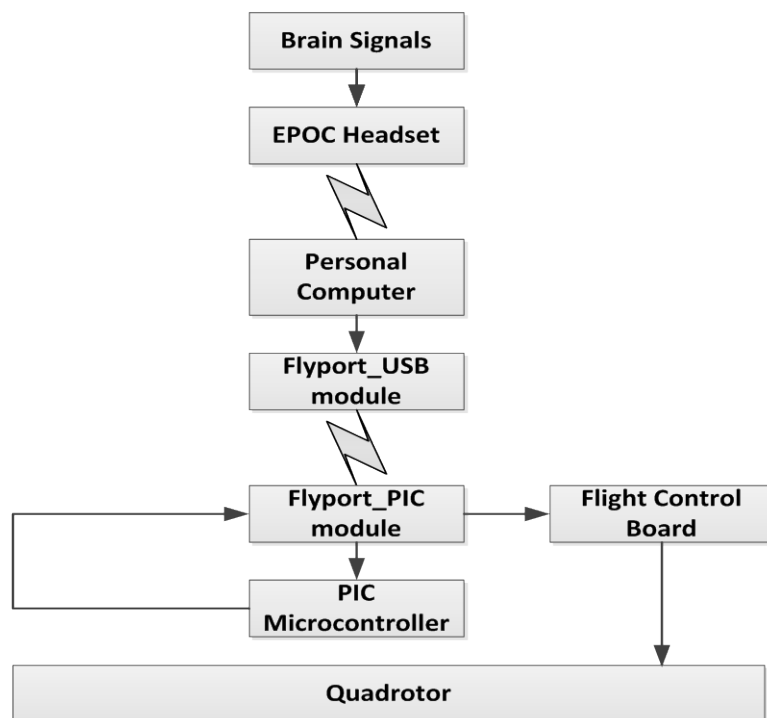Fig. 14. Surface view of the outputs, a) KP, b) Kd

Fig. 15. Block diagram of the AirServer structure

V.   NUMERICAL AND EXPERIMENTAL RESULTS

We performed several tests to validate the previous finding and test how well the brain commands can control the quadrotor. The controllers as well as the quadrotor's dynamic model have been first developed in Matlab Simulink. After attaining satisfactory results, the controllers were implemented on a real quadrotor. The details of the numerical and experimental results are presented in the following subsections.

A. Numerical Results

The quadrotor dynamic as well as the controller's models were simulated on a MATLAB/Simulink. Fig. 16 shows the Simulink model of the Fuzzy PD Controller with unity step input. The inputs to the quadrotor model are u_1, u_2, u_3, and u_4. The outputs from the same model are the three Euler's angles as well as the position of the quadrotor. Those

outputs are used as a feedback input to the FlyPort model_1. The fuzzy controllers (FLC$_z$, FLC$_{phi}$, FLC$_{theta}$, and FLC$_{psi}$) were developed using the MATLAB Fuzzy Logic Toolbox. The outputs from the fuzzy system were used to tune the gains of the PD controller. The gains of the conventional PD controller were initially adjusted using the Ziegler-Nichols method. Based on this method, the following gains were obtained: Kp = 0.0122, Kd =0.0093. The nominal values of the quadrotor parameters used for simulation are presented in Table 4. The step input was used to analyze the response of the attitude and altitude control of the quadrotor. The results for the three Euler angles as well as the altitude after running the system are shown in Fig. 17. It can be observed that the Fuzzy PD controller is able to function correctly and select the most suitable gains in the presence of brain signals.
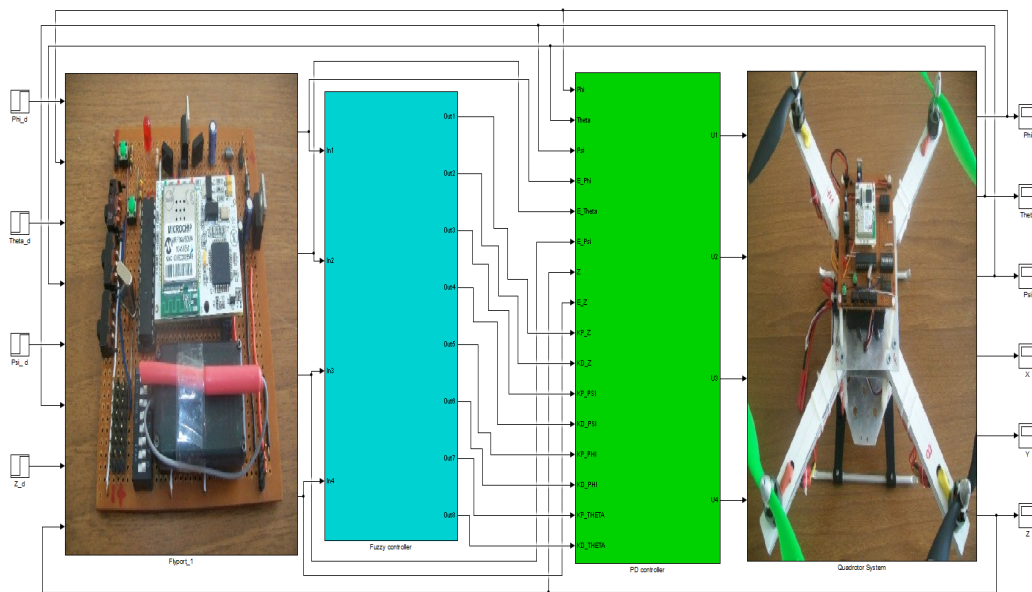

Fig. 16. MATLAB/Simulink block diagram of the quadrotor simulator

TABLE 4
QUADROTOR MODEL PARAMETERS USED IN SIMULATION EXPERIMENTS

| parameters	Value	Unit	Description
L	0.30	m	arm length
g	9.81	m/s^2	gravitational constant
mr	0.228	Kg	Rotors mass
mt	1.619	Kg	total mass of the quad-rotor
jx	0.013	Kg.m^2	moment of inertia around X axes
jy	0.013	Kg.m^2	moment of inertia around Y axes
jz	0.023	Kg.m^2	moment of inertia around Z axes
w	990	Rad/s	rotor angular speed
Dm	0.000092	Kg.m	thrust coefficient
Dt	0.0000071	Kg.m^2	torque coefficient
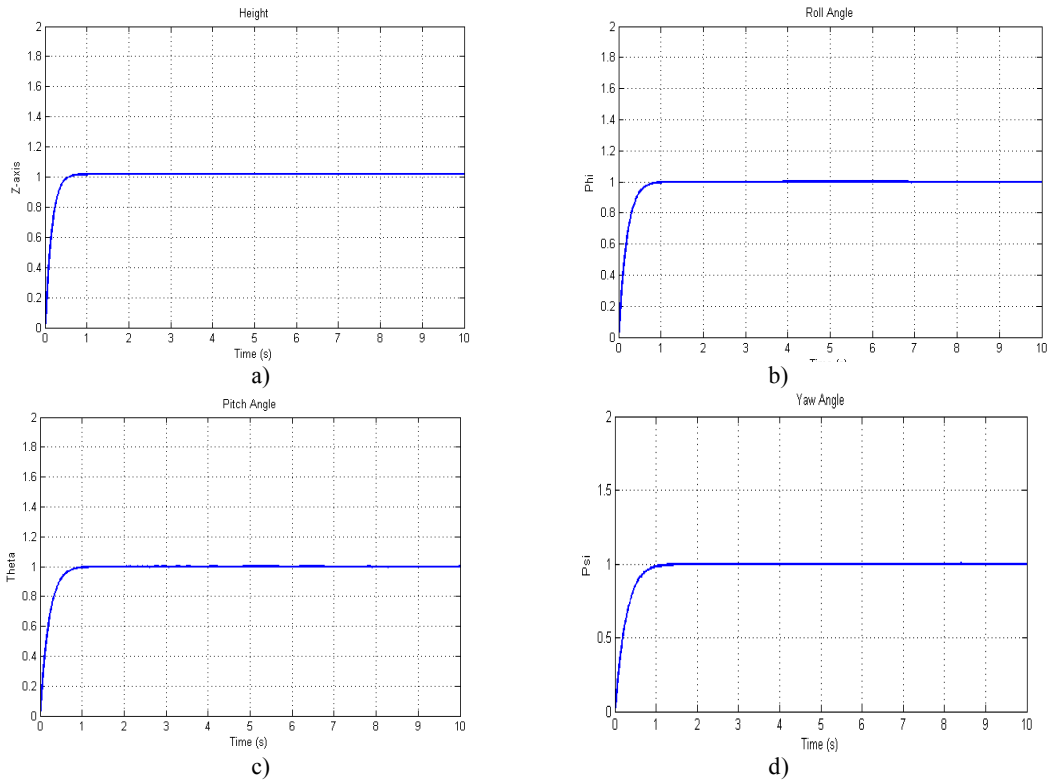
Fig. 17. Simulation results of the fuzzy controller, a) evolution of Z-axis (altitude), b) evolution of roll angle, c) evolution of pitch angle, d) evolution of yaw angle

B. Experimental Results

As shown in Fig. 18, we ran two experiments investigating how the brain signals would control the quadrotor. As an initial step, a healthy participant was seated in front of a computer running all the software required for the experiment and trained to control left, right, pull and push movements of a virtual cube supplied by the Emotiv Company. Training was done in twelve trials; and each lasted 8 seconds for each action. The trial was considered successful if at least one state was in the required direction during the 8 seconds. After 10 minutes of training, a 52% rate of success was achieved in making an action happen when intended. During training trials, we found that the cognitive suite sometimes produced small outputs that were unintended by the user. We solved this problem by considering the state to be unsuccessful if the participant did not see any movement in the correct direction. We also enhanced the previous step by using Emokey filter, at a certain threshold. During the key mapping, we set a condition greater than the threshold of 30% in order to detect the power of the action. In order not to overwhelm the drone with too many commands, we added the further filtering technique of checking for a new state every 50ms. This technique was implemented by an event-driven mechanism. The training process is shown in Fig. 18a.

The second experiment was designed to test how the quadrotor responds to the fuzzy logic controller. As a starting point, the quadrotor was commanded to maintain a fixed hover position and keep the pitch, roll, and yaw angles within the interval (0, 0, 0). This situation was achieved by instructing the participant to only imagine the push command. Due to the presence of disturbances, such as wind disturbance, it was difficult for the participant to maintain the quadrotor at a constant altitude. This problem was solved by adding an ultrasonic sensor to measure the distance to the ground. The ultrasonic sensor was pointed towards the ground and connected with the FlyPort module via a UART interface. The participant was then instructed to do the experiment again by commanding the drone so that it would take off

slowly then proceed until it reached the maximum range of the ultrasonic sensor, Fig. 18b. After reaching the desired altitude, the quadrotor was commanded to pitch-up and then pitch-down.

a)   b)

Fig. 18. Snapshots of the AirServer in use a) Emotiv EPOC headset positioned on the test person scalp, b) result of detecting human thoughts

VI. CONCLUSIONS

The present study shows that it is possible to control the drone with a BCI system. An innovative control law using fuzzy PD was used successfully to stabilize the states of the drone and ensure that the brain waves would not overwhelm the drone with too many commands. A two-input and two-output fuzzy control system was presented. The controller consisted of four fuzzy logic modules designated for the control of the quadrotor height and orientation. The controller was simulated in MATLAB/Simulink and then experimentally tested on our prototype, called AirServer. The source of the control signal was the brain waves recorded from the surface of the scalp using EEG sensors. The cognitive suite received the EEG waves and converted them to commands such as pull, push, right and left. These commands were used to control the AirServer. Simulated results were quite promising and demonstrated the ability of disabled people to steer the drone with the power of thought and use it to look after their own needs. Further study is needed to enhance the movement of the drone.

REFERENCES

[1] S. de Vries, T. Mulder, "Motor imagery and stroke rehabilitation: a critical discussion," *Journal of Rehabilitation Medicine*, vol. 39, no. 1, pp. 5-13, 2007.

[2] T. Hinterberger, J. Mellinger and N. Birbaumer, "The thought translation device: structure of a multimodal brain-computer communication system," *First International IEEE EMBS Conference on Neural Engineering*, pp. 603-606, 2003.

[3] N. Neumann and A. Kübler, "Training locked-in patients: a challenge for the use of brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 169-172, 2003.

[4] N. Birbaumer, A. Kubler, N. Ghanayim, T. Hinterberger, J. Perelmouter, J. Kaiser, I. Iversen, B. Kotchoubey, N. Neumann and H. Flor, "The thought translation device (TTD) for completely

paralysed patients," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 8, no. 2, 2000.

[5] K. LaFleur, K. Cassady, A Doud, K. Shades, E. Rogin and H. Bin, "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain-computer interface," *Journal of Neural Engineering*, vol. 10, no. 4, pp. 1-15, 2013.

[6] L. Jzau-Sheng and J. Zi-Yang, "Implementing remote presence using quadcopter control by a non-invasive BCI device," *Computer Science and Information Technology*, vol. 3, no. 4, pp. 122-126, 2015.

[7] M. Cheng, X. Gao, S. Gao and D. Xu, "Design and implementation of a brain-computer interface with high transfer rates," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 10, pp. 1181-1186, 2002.

[8] P. Muller, R. Gernot and P. Gert, "Control of an electrical prosthesis with an SSVEP-based BCI," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 1, pp. 361-364, 2008.

[9] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G.Pfurtscheller and T. M. Vaughan, "Brain-computer interface for communication and control," *Clinical Neurophysiology*, vol. 133, pp. 767-791, 2002.

[10] A. N. Alshbatat, P. J. Vial, P. Premaratne and L. C. Tran, "EEG-based brain-computer interface for automating home appliances," *Journal of Computers*, vol. 9, no. 9, pp. 2159-2166, 2014.

[11] A. Alshbatat and A. Shubeilat, "Brain-computer interface for controlling a mobile robot utilizing an electroencephalogram signals," *Global Journal on Technology*, vol. 5, pp 124-130, 2014.

[12] W. D. Penny, S. J. Roberts, E. A. Curran and M. J. Stokes, "EEG-based communication: a pattern recognition approach," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 214-215, 2000.

[13] G. Pfurtscheller, C. Neuper, C. Guger, W. Harkam, H. Ramoser, A. Schlogl, B. Obermaier and M. Pregenzer, "Current trends in Graz brain-computer interface (BCI) research," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 216-219, 2000.

[14] G. Schalk and E. Leuthardt, "Brain computer interface using electrocortocographic signals," *IEEE Reviews in Biomedical Engineering*, vol. 4, pp. 140-154, 2011.

[15] H. Zhang, C. Guan and C. Wang, "Asynchronous P300-based brain-computer interfaces: a computational approach with statistical models," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 6, pp. 1754-1763, 2008.

[16] E. Pasqualotto, S. Federici and M. Belardinelli, "Toward functioning and usable brain-computer interfaces (BCIs): a literature review," *Disability and Rehabilitation: Assistive Technology*, vol. 7, no. 2, pp. 89-103, 2012.

[17] J. N. Mak and J. R. Wolpaw, "Clinical applications of brain-computer interfaces: current state and future prospects," *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 187-199, 2009.