Agronomy Research **17**(X), xxx–ccc, 2019
https://doi.org/10.15159/AR.19.143

Design and construction of a low-cost remotely piloted aircraft for precision agriculture applications

M.G. Morerira[1], G.A.S. Ferraz[1], B.D.S. Barbosa[1], E.M. Iwasaki, P.F.P Ferraz[1], F.A. Damasceno[2] and G. Rossi[3]

[1]Federal University of Lavras, Department of Agricultural Engineering, University Campus, BR37.200-000, Lavras, Brazil
[2]Federal University of Lavras, Department of Engineering, University Campus, BR37.200-000, Lavras, Brazil
[3]University of Florence, Department of Agriculture, Food, Environment and Forestry (DAGRI), Via San Bonaventura, 13, IT50145 Florence, Italy
[*]Correspondence: gabriel.ferraz@ufla.br

Abstract. This study aimed to construct a low cost RPA capable of recording georeferenced images. For the construction of the prototype of a quadcopter type RPA, only essential materials were used to allow stable flight. A maximum total weight of 2 kg was stipulated, including frame weight, electronic components, motors and cameras. The aircraft was programmed using a low-cost microcontroller widely used in prototyping and automation research. An electronic circuit board is designed to facilitate the connection of the microcontroller with the other components of the design. Specific software was used for flight control. The prototype was built successfully, being able to lift stable and controllable flight. However, we still need to acquire equipment and programming components capable of enabling autonomous images and flights. The final cost of the RPA was on average $ 427.00 on average 50% lower than the values found in the Brazilian ARP market ($ 772.81 to $ 1,288.00)

Key words: arduino, prototype, drone, UAV.

INTRODUCTION

An unmanned aerial vehicle (UAV) is defined as an aircraft without a human pilot aboard (Nonami et al., 2010). The National Civil Aviation Agency (Agência Nacional de Aviação Civil – ANAC) of Brazil classifies a UAV as a function of its operation, defining it as a remotely piloted aircraft (RPA) when controlled by a ground operator (Decea, 2017). This low-cost RPA is able to perform photogrammetric work with digital cameras attached to its frame, and the flight mode may alternate between manual, semi-manual, and automatic (Nex & Remondinho, 2013).

Early in their development, UAVs were primarily used for military purposes, such as the surveillance, reconnaissance and mapping of enemy areas. Currently, UAVs are widely used in several areas such as agriculture, silviculture, archaeology, environment, emergency and ground traffic monitoring services (Nex & Remondinho, 2013). These various applications of UAVs are presently considered a low-cost option for photogrammetric work (Nex & Remondinho, 2013). In addition to remote sensing (RS) work, UAVs have a high potential for terrain mapping, generating digital elevation models (DEM) to assess flood risk areas (Covaney & Roberts, 2017).

Due to the increasing number of technological innovations and to greater applicability of RPA in the civil sector, measures are being adopted to regulate the use of this equipment. No international regulation on the use and operation of RPA is available yet. Several countries have adopted different regulations, thus making it difficult to list the specific objectives of each policy (Nex & Remondinho, 2013).

In recent years, the demand for RPA has increased thanks to technological advances and cost reduction and to the size of the sensors and the benefits they provide, especially low-cost RPA (Segales et al., 2016). These RPA could be operated by farmers themselves to diagnose crop characteristics, such as water stress, and then they can adjust their water management practices as needed (Romero-Trigueros et al., 2016). Hence, UAV technology may bridge the knowledge gap between leaf and canopy, improving the spatiotemporal resolution of data on the vegetative state (Gago et al., 2015).

When using RPA for data collection in agriculture, RS has advantages over satellite and manned aircraft imagery due to the frequency of the image capture and processing, thus meeting the requirements for the rapid monitoring, evaluation and mapping of natural resources on a user-defined spatiotemporal scale (Feng et al., 2015). In addition, the flight height is lower, which can result in images with a higher resolution. This is necessary for managing small crop fields (Huang et al., 2013).

According to Ibrahim et al. (2017), images acquired by RPA have been rather efficiently used in irrigated areas. Furthermore, according to these authors, red, green, and blue (RGB) cameras can be used to identify crop failures and thus plan irrigation. Conversely, the use of near-infrared (NIR) cameras enables farmers to identify irrigation performance factors, such as leaks in the water supply system.

Based on the above, the need for building a low-cost RPA prototype as a tool for precision agriculture applications was identified.

MATERIALS AND METHODS

The first step of this study was to research components available on the market, checking their specifications and performing calculations to decide whether the components were suitable for the project. Initially, we intended to purchase only essential equipment to build an RPA able to maintain a stable and controlled flight. At first, this aircraft would not be able to perform autonomous flights due to the lack of a global positioning system (GPS). The objective will be to develop an aircraft stable enough to record images at the lowest possible cost, thereby establishing the groundwork for developing a more complex and robust system. A maximum total weight of 2 kg, including the weight of the frame, electronic components, motors, and camera, was calculated. This calculated weight made it possible to define which engines should be used to support the aircraft. To support the engines and other components of the RPA, the frame chosen was the generic model F450. This frame is one of the most commonly used in the hobbyist market for recreational drone construction. Weighing approximately 282 grams, with a wingspan of 450 mm, this frame is made of high-strength nylon. The frame also has a printed circuit board (PCB) power distribution board for the engines and landing gear, in flexible plastic. Despite its recreational use, this frame is strong and easily found on the market and is thus a good choice as a development platform.

Turnigy brushless direct current (BLDC) engines were chosen because they can be easily purchased in the Brazilian market. Accordingly, the 1100KV 2836-8 engine was selected. This engine is capable of generating a maximum propulsive efficiency of 1,130 g, according to the manufacturer, providing the RPA with a theoretical propulsive efficiency of 4.5 kg. In practice, part of the propulsion should be reserved to maintain aircraft manoeuvrability. The designed RPA will hover at 50% engine power considering a linear power thrust function, thus reserving 50% of the engine power for ascent and wind resistance manoeuvres. Importantly, the relationship between the thrust generated by the engine-propeller set and the power applied is not linear. However, this consideration suffices for the initial engine choice of the aircraft.

The next step was to choose the battery, Li-Po 4S (14 V), with 5,000 mAh. Despite an approximate weight of 500 g, which accounts for 25% of the total aircraft weight, according to calculations made using the eCalc web tool, the weight-to-load-capacity ratio is appropriate for an acceptable flight time in agricultural applications. The minimum estimated flight time is 10 minutes, when the engines work at full power. BLDC engines require an electronic speed controller (ESC) for coil switching. Four 30 A ESCs (one for each engine), compatible with a 4S battery, were purchased. The propellers selected to build the RPA were plastic ABS propellers sized 10 x 4.5 inches, according to the engine manufacturer's instructions.

The Arduino Nano, an open-source electronic prototyping platform based on the Atmel microcontroller, was chosen as the flight controller and programmed using the manufacturer's open-source integrated development environment (IDE).

An MPU6050, an inertial measurement unit (IMU) capable of communicating through the I2C protocol, was chosen because this unit has 3 accelerometer sensors, along the three Cartesian axes, which are capable of providing acceleration measurements at a resolution ranging from 160 m s^{-2} to 20 m s^{-2}. The unit also has 3 gyro sensors, also along the three Cartesian axes. The resolution of the angular velocity measurement in the MPU6050 is approximately 200º s^{-1}. The direction of the vehicle can be determined using both data to feed a complementary filter.

A TGY-i6 is a 6-channel telemetry 2.4 GHz transmitter. The information transmitted by the TGY-i6 radio control (RC) is converted by the pulse-width modulation (PWM) receiver, and the signal peaks at 5 V, with a standardised pulse width time ranging from 1,000 to 2,000 μs.

An electronic circuit board was designed (Fig. 1) to facilitate the connection between the Arduino and the other components of the project. A voltage divider circuit was used to send a signal proportional to the battery charge to the controller, to allow decision-making when the battery is running low. The controller communicates with the radio receiver and with the ESCs through Arduino I/O digital pins 4 to 11, using 4 pins to receive information from 4 receiver channels and 4 to control the ESCs. The ESC control cable has 3 pins, voltage at the common collector (VCC), ground (GND) and signal pins. Only the Signal and GND pins are used in the connection, and the VCC pin is not necessary. The Wire library of the Arduino IDE implements I2C communication. For this communication, the arduino serial data line (SDA) and serial clock line (SCL) pins are used, which correspond to A4 and A5 in the nano model. The circuit diagram of Fig. 1 shows the circuits used in this project in detail.

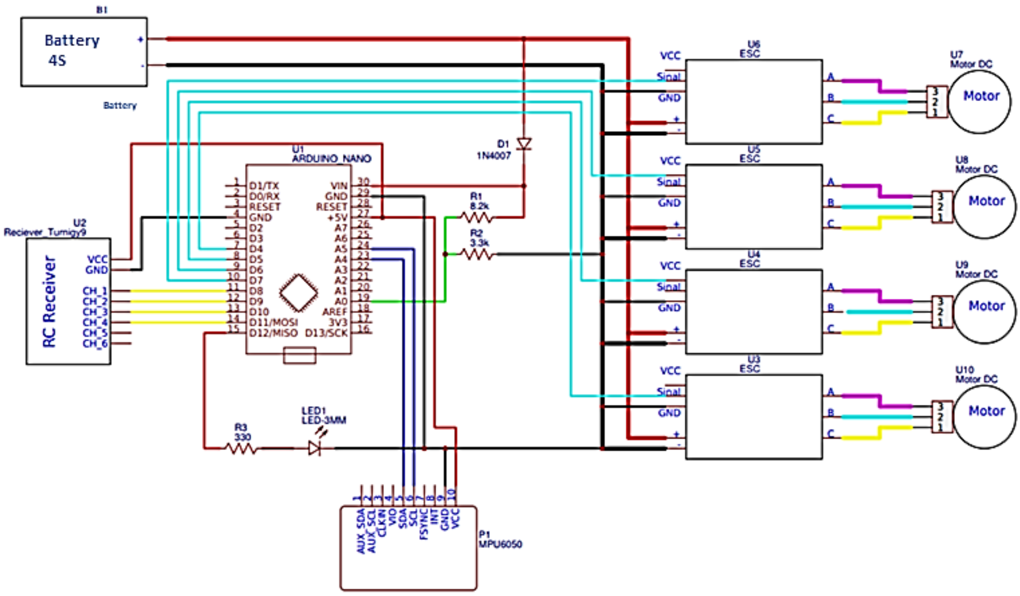

Figure 1. Electronic circuit diagram.

The flight control programme of the RPA was based on YMFC-AL software, created by J. Brokking, which consists of a simple Arduino flight controller, capable of maintaining flight stability from a single IMU. This programme is divided into 3 separate modules due to limited controller memory. Those modules should be manually uploaded to the Arduino memory to set up, test, and fly the aircraft. The configuration programme is used to assess whether all components essential to the RPA are correctly assembled and to calibrate the limits of the radio control signal. To use the radio control, the Arduino Nano is connected to a computer through a USB cable. The propellers of the aircraft should be removed and the battery turned off to avoid inadvertently turning the engines on. In the computer, using the Arduino IDE, the configuration programme of the aircraft should be opened and uploaded to the Arduino. Then, when opening the IDE serial monitor, messages are displayed on the computer screen requesting specific user commands. Those messages ask the user to perform some specific actions using the radio control buttons to identify which channels are being used, the signal amplitude of each channel and the central point. All data are stored in the non-volatile memory (electrically erasable programmable read-only memory – EEPROM) when running the programme. Subsequently, the programme asks the user to rotate the RPA to identify the direction in which the MPU6050 was installed. All configurations saved are used by the other programmes and are not erased when a new code is uploaded to Arduino.

The second programme is used to test the radio communication, the balance of the propellers and the kinetic sensor. To use the sensor, the Arduino should be connected to the computer again and the programme uploaded with the IDE. In the serial monitor, the user can send a letter to the Arduino, requesting one of the aforementioned tests. The radio communication test is activated by sending the letter ‘r’ in the serial monitor. The current position of the controls and the function activated (acceleration, pitching, rolling, yawing) are shown in the monitor. Then, moving the radio controls should change the values shown on the computer screen. The kinetic sensor test is activated by sending the letter ‘a’. The computer screen shows the current orbital inclination of the RPA in degrees. The values of the orbital inclination displayed when the aircraft is levelled should be recorded because they will be subsequently used to calibrate the sensor. The propellers are balanced based on the vibration caused by the rotating propellers. Vibration impairs the stability of the aircraft during flight and therefore should be minimised. To use the aircraft, the propellers should be installed in the aircraft, the remote control turned on and the acceleration control set to the lowest level. With the battery connected to the RPA, a specific engine can be turned on by sending a number from 1 to 4 in the serial monitor. To increase the engine rotation, the RC accelerator should be used. The serial monitor displays the vibration perceived by the IMU. Adhesive tapes were placed to adjust the weight of each side of the propeller to reduce the vibration detected by the sensor. The first two programmes were run with the RPA connected to a computer, and the last programme was used to fly the aircraft. This programme was uploaded by connecting the Arduino to the computer and transferring the programme through the IDE, removing the connection cable after transferring the programme. The programme is run as soon as the battery connector is plugged in. The programme cyclically runs a proportional–integral–derivative (PID) controller for each axis of the aircraft, totalling three PID controllers. The controller of each axis is responsible for controlling the angle of the aircraft along that axis.

The programme consists of three steps that should be cyclically run and a fourth step that is sporadically run. When starting the programme, a configuration sequence runs only once. This sequence assesses whether the configuration programme was run correctly and calibrates the IMU. Data sent by the radio control sets the target aircraft orbital inclination and engine acceleration. Whenever the controller detects a change in the state of communication with the RC, an interrupt is generated, and the desired operation points of the RPA are updated. This operation is the sporadic step that occurs only when the user changes a command in the RC. The first step of the flight algorithm is to collect orbital inclination data and to compare them with the desired point. Then, those differences are used in three PID control algorithms, one for each Cartesian axis. Finally, the result from those operations is interpolated and sent as a control signal to turn the four engines on. The algorithm also includes sequences that limit the amplitude of the signals generated, settings to compensate for battery power loss, and locks in case any faults are detected. To analyse the results of this newly built RPA, a test flight was performed, visually comparing its flight stability with that of a commercial RPA.

RESULTS AND DISCUSSION

The final weight of this newly built RPA was 1.4 kg, at a total cost of $ 412.66. The costs of the individual components are outlined in Table 1.

A GPS module and a telemetry radio are necessary for automatic flight. A GPS 6M sensor is capable of providing global positioning data to the RPA. The telemetry radio enables communication between the ground control station and the RPA. Both components are available on the Brazilian market. The GPS NEO-6M sensor costs approximately $ 25.76, and a telemetry radio transmitter-receiver kit costs approximately $ 38.64. The gimbal should be purchased according to the weight of the camera. A generic 2-axis gimbal for RPA is commercially available for approximately $ 77.28.

The total price of the RPA prototype is $412.66. RPAs with similar capabilities, such as the DJI Spark, Mavic and Phantom IV (DJI, Shenzhen, China) models in the Brazilian market range from $ 772.81 to $ 1,288.00. This difference of price of more than 50% between the prototype and the models available in the market, reintroduces the potential of the equipment under construction in this study. Other items that allow automatic flight modes can be installed in the drone, keeping the price below $ 500.00.

Table 1. Costs of the pieces for assembling the RPA

Parts	Amount	Value ($)	Total ($)
Engine1,100 KV	4	21.08	84.32
ESC 30[a]	4	12.15	48.60
Baterry 4S	1	66.00	66.00
Frame F450	1	24.03	24.03
Baterry charger	1	21.35	21.35
Arduino nano	1	5.54	5.54
Radio control and receiver	1	108.29	108.29
MPU6050	1	3.43	3.43
Board PCB	1	2.64	2.64
Freight	-	32.34	32.34
Other expenses*	-	15.85	15.85
Total	-	-	412.66

* Expenses related to screws and fittings, wires, insulation tape, battery, weld and others.

The project of low cost drone was organized in several stages during one and a half year of development. Most of the time was taken during the research, design, programming and testing.

After arrival of all components, the complete assembly of the low cost RPA took 12 full hours including soldering of custom PCB divided in tree workdays.

Figure 2. Low-cost RPA prototype.

The calculated flight time was 13 minutes. A test performed with motors continuously connected to 50% of power showed that the prototype has an average autonomy of 14 minutes of flight. During the aircraft setup process, an orbital tilt of 2 degrees on the pitch axis was detected in the IMU. This slope has been corrected by changing a section of the flight code reserved for that purpose. During the RPA flight test, the aircraft took off successfully, showing that the engines chosen for this project were of adequate size. By keeping the rolling, throwing and yaw controls in the center position, the aircraft remained level, thus proving that a stability control algorithm with a single PID axis controller can be used.

When moving the controls, the RPA responded to the commands, moving in the desired direction. In the current state, the algorithm run in the flight controller is only responsible for keeping the aircraft level. The pilot should constantly adjust the accelerator to keep the aircraft hovering at a specific altitude and should also compensate for the wind. This aspect of the flight mode makes piloting difficult for beginners, which is undesirable for the purpose of the aircraft.

During the code compilation, we observed that the flight control programme takes up 12104 bytes of memory, corresponding to 40% of the programme memory available in the Arduino Nano. The global variables take up 516 bytes, corresponding to 25% of the available dynamic memory. To avoid future expansion problems, a more powerful controller must be purchased. A promising controller is the Raspberry Pi, a small, inexpensive Unix-based computer (Torres et al., 2016). According to the manufacturer, the simplest version has a single-core 1 GHz processor and 512 MB of RAM.

An altitude hold algorithm should be implemented to avoid the need for adjusting the pilot acceleration. For this purpose, some type of sensor should be used to measure the aircraft altitude. A barometer can be installed to perform this function of measuring the aircraft altitude (Silva, 2013).

The RPA must also hold on the other horizontal axes, known as position hold. For this purpose, the RPA must know its global positioning somehow. Hold algorithms require higher sensor accuracy. Inertial models show a lot of noise, which causes drifting over time (Alexa, Nikolakopoulos & Tzes, 2012). This characteristic prevents its use in defining the local position of the aircraft. To avoid this effect, according to Kim et al. (2003), a GPS must be used to correct this deviation. Sensor fusion techniques can be used to integrate GPS, IMU and barometer data.

Lastly, a telemetry device will be necessary to allow communication between a ground control station and the aircraft. From this station, complex commands may be sent to the RPA, such as the trajectory, altitude and camera operation data. After implementing these changes, already predicted in the project, the aircraft will be able to perform image acquisition tasks for PA applications (Jorge & Inamasu, 2014).

These additional implementations will presumably make it possible to build an aircraft ready for flight at a final cost of $ 679.00. This value is 60% lower than that found on the market for an aircraft capable of performing similar functions. Abade et al. (2016) built a UAV prototype for agricultural monitoring purposes at a final cost of $ 1,242.00

The RPA is presently an optimal base to build a more complex and robust system and makes it possible to easily understand the dynamics involved in the control process, albeit without recording images. For such a purpose, a camera gimbal, a GPS module and a barometer must be installed. The installation of these additional components will overload the Arduino controller. The replacement of this controller by a more powerful one capable of performing operations faster and preferably in parallel should be studied. By introducing these modifications, the RPA may be used to map crops.

CONCLUSION

A low-cost RPA prototype capable of stably lifting off and hovering with potential for future agricultural monitoring applications was built. The final cost of the RPA was $ 412.66. Because it is modular, various other functions can be implemented, and the RPA has a robust base for such purposes.

ACKNOWLEDGEMENTS. The authors thank, the Foundation for Research of the State of Minas Gerais (FAPEMIG), the National Council for Scientific and Technological Development (CNPq), the Coordination for the Improvement of Higher Education Personnel (CAPES), the Federal University of Lavras (UFLA) and University of Firenze (UniFI).

REFERENCES

Abade, A., de Campos, M.D., Porto, L.F., de Farias Coelho, Y., de Moura Sousa, Y. & Nespolo, J.P. 2016. The Optimized Construction of a Drone for Applications in Agriculture and Precision Livestock. *Proceedings of the Regional Computer Science School of the Brazilian Computer Society (SBC) - Regional of Mato Grosso* **7**, pp. 1–10 (in Portuguese).

Alexis, K., Nikolakopoulos, G. & Tzes, A. 2012. Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory & Applications* **6**(1), 1812–1827.

Coveney, S. & Roberts, K. 2017. Lightweight UAV digital elevation models and orthoimagery for environmental applications: data accuracy evaluation and potential for river flood risk modelling. *International journal of remote sensing* **38**(8–10), 3159–3180.

Department of Airspace Control (DECEA). Portaria DECEA nº 415/DGCEA, 9.11.2015. Available in: <http://publicacoes.decea.gov.br/?i=publicacao&id=4262>. Accessed in: 2.10.2017

DJI Store. Available in: <https://store.dji.com/>. Accessed in: 04.17. 2018.

ECalc. The most reliable electric Motor Calculator on the web. Available in: <https://ecalc.ch>. Accessed in: 04.05 2017.

Feng, Q., Liu, J. & Gong, J. 2015. UAV remote sensing for urban vegetation mapping using random forest and texture analysis. *Remote sensing* **7**(1), 1074–1094.

Gago, J., Douthe, C., Coopman, R.E., Gallego, P.P., Ribas-carbo, M., Flexas, J., Escalona, J. & Medrano, H. 2015. UAVs challenge to assess water stress for sustainable agriculture. *Agricultural Water Management* **153**, 9–19. doi: https://doi.org/10.1016/j.agwat.2015.01.020

Huang, Y., Thomson, S.J., Hoffmann, W.C., Lan, Y., & Fritz, B.K. 2013. Development and prospect of unmanned aerial vehicle technologies for agricultural production management. *Int. J. Agric. & Biol. Eng*. **6**(3), 1–10. doi:10.3965/j.ijabe.20130603.001

Ibrahim, J.T., Waskitho, N.T., Wahono, W. & Ma'mun, S.R. 2017. Survey of irrigation area using micro unmanned aerial vehicle (micro-uav) in gumbasa irrigation area. *Agricultural Socio-Economics Journal* **17**(1), 1–5.

Jorge, L.A de C. & Inamasu, R.Y. 2014. Use of unmanned aerial vehicle (UAV) in precision agriculture. In Bernardi, A.D.C., Naime, J.D.M., Resende, A.D., Inamasu, R.Y., Bassoi, L. *Precision Agriculture: Results of a New Look*. Embrapa Instrumentation-Tecnical book (INFOTECA-E), Brasilia, pp. 109–134 (in Portuguese).

Kim, J.H. & Sukkarieh, S., Wishart, S. 2016. Real-Time Navigation, Guidance, and Control of a UAV Using Low-Cost Sensors. In: Yuta, S., Asama, H., Prassler, E., Tsubouchi, T., Thrun, S. (eds) *Field and Service Robotics*. *Springer Tracts in Advanced Robotics*, Springer, Berlin, Heidelberg, pp. 299–309

Nex, F. & Remondino, F. 2014. UAV for 3D mapping applications: a review. *Applied Geomatics* **6**(1), 1–15. doi: https://doi.org/10.1007/s12518-013-0120-x

Nonami, K., Kendoul, F., Suzuki, S., Wang, W. & Nakazawa, D. 2010 Autonomous Flying Robots. *Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Springer. Tokyo New York, 328 pp.

Romero-Trigueros, C., Nortes, P.A., Alarcón, J.J., Hunink, J.E., Parra, M., Contreras, S., ... & Nicolás, E. 2017. Effects of saline reclaimed waters and deficit irrigation on Citrus physiology assessed by UAV remote sensing. *Agricultural Water Management* **183**, 60–69.

Segales, A., Gregor, R., Rodas, J., Gregor, D. & Toledo, S. 2016. Implementation of a low cost UAV for photogrammetry measurement applications. *In Unmanned Aircraft Systems (ICUAS)*, Arlinton, VA. pp. 926–932.

Silva, Leonam Pecly da. 2013. Design and construction of a remotely piloted quadricopter and its attitude control and determination system. Instituto Federal Fluminense, Campos dos Goytacazes, 107 pp.

Torres, A.B.B., Rocha, A.R. & DE Souza, J.N. 2016. Performance analysis of mqtt brokers in low cost system. *In: Proceedings of the XXXVI Congress of the Brazilian Computer Society. Brazilian Society of Computation.* Porto Alegre, Brazil (in Portuguese).