DESIGN OF A NAVIGATION UNIT
FOR A NEW UNMANNED AERIAL VEHICLE:
A COMPOSITE OF A CAPSTONE AND RESEARCH PROJECTS

Witold Kinsner1, Stephanie English2, Curtis Einarson2, Bryan Drobot2,
Kelly Riha3, Mohamed Temam Naat4, Rafi Belal Mahabbit1
Barry Prentice1,4, and Dale George4

1Cognitive Systems Laboratory
2Department of Electrical and Computer Engineering
3University of Manitoba
and 4Buoyant Aircraft Systems International (BASI)
Winnipeg, Manitoba, Canada
witold.kinsner@umanitoba.ca

Abstract – Many challenges surround the engineering design process of an unmanned aerial vehicle (UAV). This paper describes a composite of a capstone and two research projects for the design of a navigation subsystem for a UAV. These projects address control, stabilization and autonomy of the vehicle with time-varying, nonlinear dynamics in a challenging environment and variable payloads. The paper provides theoretical and practical solutions, as well as alternatives to current control approaches for UAVs. The results are intended for the design of a UAV that can be used for different types of operations such as connecting isolated remote communities and reinforcing the sovereignty of countries within their borders, without requiring traditional means of protection.

Keywords: Unmanned aerial vehicles, PID controller; integer-order PID (IPID) controller; fractional calculus; fractional-order PID (FPID) controller; Mamdani fuzzy controller.

1. INTRODUCTION

1.1 Motivation

The approach to teaching Design Engineering has been evolving rapidly from the traditional classroom lecturing to a hybrid of (i) theory, formal analysis and synthesis [1] to [2], (ii) multi-tier laboratories [3], (iii) final-year group capstone project, (iv) large complex projects such as the T-Sat1 [4], and (v) applied research projects. The main objective of this approach is to expose university students to the practice of design, in addition to a sound theoretical foundation for analysis and synthesis. The students also learn many other hard professional skills (unfortunately called “soft” skills), including (a) how to work in teams, (b) how to communicate options for solutions, (c) how to develop solutions within budgets, (d) how to incorporate computational intelligence so that the system could adapt to changing conditions autonomously, (e) how to develop designs that are robust, while being environmentally friendly. The inclusion of such long-term objectives into design is mandated by the Canadian Engineering Accreditation Board (CEAB).

This paper describes this new approach through an example which focuses on a capstone project with four undergraduate students, augmented by two applied research projects conducted by graduate students, all supported by industrial partners. The project is to develop subsystems for a new class of peaceful unmanned aerial vehicles (UAVs). UAVs have become useful in many fields. They perform many useful functions, including surveillance, in-flight testing of critical parameters, and delivery of materials. The size of such vehicles ranges from a regular helicopter to drones, to nano-machines flying in constellations (swarms). An example of such a UAV is an electrical quadruple helicopter (quadcopter). It is very stable, flies under remote wireless control from smart phones or tablets, and can land autonomously.

An important motivation for this project is that Canada has many remote and isolated communities in its vast territory. Connecting such locations to the city using terrestrial means such as roads would be very costly. Furthermore, the severe weather conditions (such as extremely low temperatures) would have a devastating impact on the condition of those roads. An alternative to connecting those remote areas would be through the air. Such alternative would represent a longer-term solution for a minimum cost, yet it is more challenging. This can be accomplished through the design of reliable unmanned aerial vehicles. Such vehicles can play a civil role, as well as a military one. For instance, they can reinforce the sovereignty of the country within its borders without requiring a human presence.

A new class of multi-copters is being developed by Buoyant Aircraft Systems International (BASI) in Manitoba for the ultimate intended use to connect remote towns and to perform various tasks in difficult
environments. A navigation controller is an essential part of the quadcopter, and is being developed to very stringent requirements by the capstone team. An alternative form of the controller is also developed through an ongoing research project. Another essential subsystem of the quadcopter relates to data communications with the ground, and is also being developed through a research project.

The paper describes components of the subsystems, their simulation results, lab and field test results, as well as lessons learned from the undergraduate-graduate-industry hybrid approach to design.

1.2 Background

An unmanned aerial vehicle (UAV) is a flying machine (aircraft) without a human pilot onboard. UAVs have been used extensively in remote sensing and mapping, exploration, reconnaissance, search and rescue operations (e.g., [5], [6], [7]). UAVs have different forms (wings, or propellers), sizes, levels of autonomy, endurance, and payloads. A very common propeller-based UAV is the quadcopter, with four independent propellers. A standard hexacopter relies on six propellers placed at the corners for the lift, stabilization, and navigation. The diagonal axis of a hexacopter ranges from 1 to 26 ft. The Orbo hexacopter designed by Basi has a diagonal axis of 8 ft., with a balloon for the lift and high-efficiency motors for the navigation and stabilization. This specific architecture minimizes the stringent requirements on the controller by limiting its main tasks to the navigation and stabilization of the vehicle, which is accomplished by adjusting the speed of the propellers. Each propeller’s speed can be adjusted separately using different types of regulators. A regulator (also known as a controller) is a logic unit used to adjust the output signal with respect to a reference input.

Many challenges surround the engineering design process of a UAV. This paper addresses some design issues related to the control, stabilization and autonomy of the vehicle with time-varying, nonlinear dynamics in a challenging environment, and variable payloads. Many autonomous navigation systems on quadcopters use commercial off-the shelf (COTS) products with preprogrammed standard control algorithms. Examples of such autopilots include (i) The Kestrel autopilot, (ii) the MP2028 series autopilot, and (iii) the Piccolo autopilot system [8]. However, such COTS autopilots do not provide details on the onboard controller’s operation [9]. Furthermore, they offer very basic capabilities (e.g., altitude, speed, turn rate hold), and most of the onboard algorithms suffer from several limitations that impact their performance. Such algorithms are often based on nonlinear formulations [10], [11], evolutionary algorithms [12]-[16], or other optimization techniques [17]. Consequently, their implementation results in time and space complexities that require the usage of a powerful computing unit for real-time execution [18].

Another complication is the tuning of such algorithms that requires an accurate modeling of the system and the environment in which the system operates. However, how can we tune an environment that is almost always ill-defined? A poorly tuned controller results in large overshoots, long time delays, and slow convergence rates (under extreme cases oscillatory behaviors may cause a crash of the vehicle) that results in a power wastage and a longer flight time. Therefore, these controllers cannot cope well with nonlinear, time-varying dynamics, and ill-defined environments. Consequently, today new approaches ought to be developed to provide an autonomous and stable navigation to the future generation of unmanned aerial vehicles.

2. CURRENT CONTROL SCHEMES

2.1 Integer-Order PID Controller

Control engineering practitioners have used the proportional-integral-derivative (PID) controllers for a long time because of their simplicity and good performance. The PID controller is considered the second most important control decision and communication instrument of the 20th century. Throughout the years, the PID control scheme has dominated most of the industry applications. In fact, the use of the proportional-integral (PI) and PID controllers is ubiquitous (they are found in more than 95% of process control applications). This controller consists of three different types of actions: (i) proportional, (ii) derivative, and (iii) integral. Their main effects on the controlled system can be summarized as follows: (i) the proportional action increases the speed of the response, while decreasing the steady-state error and relative stability, (ii) the integral action attempts to eliminate the steady-state error, but, decreases the relative stability; and (iii) the derivative action increases the relative stability and sensitivity to noise. In this paper, an ideal PID controller in a unity feedback block diagram is considered, and is given by

\[ C(s) = k_p + k_i \frac{1}{s} + k_d s \]  

(1)

where \( k_p \), \( k_i \), and \( k_d \) represent tuning parameters for the PID controller, and \( s \) represents Laplace variable. The tuning parameters must be developed either in advance or during the adaptation process. Since the order of \( s \) is integer (representing integration and differentiation), the model is called integer-order PID (IPID). In order to improve the performance of the IPID, one can advance from the integer integration and differentiation to fractional operations, as described next.
2.2 Fractional-Order PID Controller

The fractional-order PID (FPID) controller provides more flexibility as compared to the IPID. As before, we must specify the proportional, integral, and derivative gains. In addition, we must specify the order of the fractional derivative and integral. Such flexibility allows for an optimal trade-off between the advantages and disadvantages of each term (proportional, integral, and derivative); thus leading to more satisfying results. The fractional PID controller can be written as

\[ C(s) = k_p + k_i s^{-\alpha} + k_d s^{\beta} \]  \tag{2}  

where \( \alpha \) and \( \beta \) represent the order of the fractional integral and derivative, respectively. Figure 1 shows a plant controlled by a FPID described by (2), with a unity feedback.

![Fig. 1. Block diagram of a process controller by a FPID controller.](image)

2.3 Mamdani Fuzzy Controller

The IPID and FPID control mechanisms described in the previous sections use crisp mathematics in which each measure has a single, well defined value. Control theoreticians have observed that real environments are neither crisp, nor deterministic. The uncertainty in such environments can be described by either probabilistic measures, or fuzzy measures. There are many probabilistic approaches to control and navigation, including the Kalman filter (KF) [19], the extended Kalman filter (EKF) (e.g., [20] to [23]), and the unscented Kalman filter (UKF) [24]. An alternative to dealing with uncertainty is to consider a measure that is neither crisp, nor probabilistic, but is represented by a membership function (MF). This is particularly important when probability distribution function cannot be obtained.

Fuzzy control (FC) has very attractive characteristics (such as autonomy and modularity), while remaining robust to the uncertainty of the process. Furthermore, FC provides the unique feature of computing with words (CW). These controllers represent one kind of nonlinear dynamic functions of the error and the change of the error signals. The type of nonlinearities depends on the fuzzy MF selected. The decision-making process relies on association and inference operators, as well as CW. Such characteristics are extremely useful in ill-defined environments since they allow expert knowledge to be developed and utilized directly to a problem at hand. Two of the most common fuzzy controllers are: (i) Mamdani and (ii) Takagi-Sugeno (TS) controllers. Mamdani and Assilian were attracted by the capabilities of fuzzy logic and introduced a fuzzy control scheme in 1975 to convert heuristic control rules stated by a human operator into an automatic control scheme. Unlike the Mamdani controller (which employs fuzzy sets as the consequent), the TS fuzzy controllers rely on linear interpolation to calculate the consequent. In this paper, we use a Mamdani-type of controller because of its ability to adjust directly the impact of each consequent, without the need for heuristic linear functions in TS model.

2.3.1 The Fuzzification Process. The controller relies on five Gaussian membership functions to fuzzify the two inputs, which are the error and the change in the error.

2.3.2 The Decision-Making Process. The decision-making process is based on the 16 rules derived to achieve the main objective; that is, minimizing the error and the change in the error. This is accomplished by: (i) Identifying the corresponding antecedents; and (ii) Characterizing the consequents based on the rules of the domain experts.

2.3.3 The Defuzzification Process. This process converts the range of the consequences into the universe of discourse by conducting a scale mapping and identifying the crisp output (correction term for the attitude) based on the center of gravity technique using the following equation

\[ y = \frac{\sum_{i=1}^{k} \mu(x_i) x_i}{\sum_{i=1}^{k} \mu(x_i)} \]  \tag{3}  

where \( \mu(x_i) \) represent the membership function of the \( x_i \) input, and \( i \) is the index of the input.

3. SIMULATION RESULTS

As described in the previous section, various controllers were investigated in order to select the controller for the UAV. The controllers included: proportional-integral-derivative (PID), Kalman filters (KF), fractional PID, and fuzzy. One of the criteria for the selection process was the implementability of a specific
controller using a limited-computing environment, such as a microprocessor or a microcontroller. The PID controllers and Kalman filters could be implemented, but with serious simplifications of the algorithms.

The dynamics governing the flight dynamics of the UAV is non-linear, while both control schemes require linearization. The FPID and variations of the Kalman filters (EKF and UKF) were designed for nonlinear systems, but are very complicated and unimplementable in low-power computers. On the other hand, the fuzzy approach can control a nonlinear system using fuzzy logic, and it is unique because it allows human interpretation to be incorporated in its design. Consequently, the Mamdani fuzzy controller [25] was selected and implemented in MATLAB/Simulink.

The initial step in designing a fuzzy controller is to describe the guidelines of flight control as viewed from the pilot’s perspective. From the guidelines in [27], we determined the inputs to the controller (angle and rate of angle change). Four separate controllers were needed to control the roll, pitch, and yaw axes and height difference. Figure 2 shows the input and output MFs used for the implementation of the fuzzy controller. The desired angle and rate of change rely on three triangular MFs, while the output to the motors uses five MFs. These decisions were influenced by the research in [28] and [30].

Fig. 2. Input and output membership functions. (a) Inputs: angle and rate of angle change, and (b) Output to motors.

Simulations were performed to evaluate the performance of the controllers in open-loop conditions, then with feedback (separately for each controller), and then collectively as a whole system. For the required feedback, we used the transfer functions for the flight dynamics from [31]. The basic understanding of flight dynamics and its application to a UAV can be found in many sources (e.g., [32], [33]).

Open loop simulations were conducted to validate the controller designed. The simulations relied on the surface and rules to visualize the results of the IF/THEN statements. For example, if an input angle is positive, and rate of angle change is also positive, then the output is a large negative.

Closed-loop simulations were implemented to obtain a realistic rate of angle change. Different transfer functions were tested based on the corresponding flight dynamics of the UAV. The controllers were then tuned in order to optimize the performance. Figure 3 shows the step response of the pitch controller as an example. The test resulted in an overshoot of 7.44% and a settling time of 0.42 seconds.

Fig. 3. Step input response for pitch.

Finally, system simulations were done to ensure motors’ controllability. The behavior of the motors depends on the design combinations of all the controllers’ output. Figure 4 shows a block diagram of the entire system including controllers and motors for roll, pitch, yaw, and height.

Fig. 4. Block diagram of system for roll, pitch, yaw, and height.
Once the system simulations were completed, the MATLAB fuzzy controller was ported to an Arduino board. Tests followed to evaluate the robustness and optimality of the system.

4. HARDWARE IMPLEMENTATION

4.1 Flight Controller

The first step in building a flight controller from a hardware perspective was to investigate various COTS components. A study of current flight controllers expanded our background on computing engines suitable for navigation, on-board sensors, and actuators, conditioning circuits for sensors, and many other circuits.

The fundamental requirements for the microcontroller (µc) included: suitability for real time systems (RTS), sufficient number of general-purpose input-output (GPIO) pins, low cost, large memory, and the ability to use various peripherals such as inter-integrated circuit (IC) and universal asynchronous receiver and transmitter (UART). Table 1 compares the µcs (APM2, Arduino Uno, iNemo, NXP and Raspberry Pi), including their clock frequencies, GPIOs, and cost.

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>Clock [MHz]</th>
<th>Core</th>
<th>Flash</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>APM2 [34]</td>
<td>16</td>
<td>Atmega 2560</td>
<td>256 KB</td>
<td>$300</td>
</tr>
<tr>
<td>Arduino Uno [35]</td>
<td>16</td>
<td>Atmega 328</td>
<td>32 KB</td>
<td>$26</td>
</tr>
<tr>
<td>iNemo [36]</td>
<td>72</td>
<td>ARM Cortex M3</td>
<td>512 KB</td>
<td>$250</td>
</tr>
<tr>
<td>NXP LPC 1769 [37]</td>
<td>120</td>
<td>ARM Cortex M3</td>
<td>512 KB</td>
<td>$30</td>
</tr>
<tr>
<td>Raspberry Pi [38]</td>
<td>700</td>
<td>1176JZF-S</td>
<td>Expandable (SD Card)</td>
<td>$50</td>
</tr>
</tbody>
</table>

The iNemo system was initially considered as the best option. Since it was not readily available, the NXP Semiconductors LPC unit was considered. The NXP had the same core, but was cheaper than iNemo. This meant that the required sensors had to be implemented separately. The APM-2 was not considered due to the price point of $300, and the Raspberry Pi was not considered due to the lack of GPIO pins.

4.2 Other Subsystems

In addition to the flight controller, many other modules must be designed and implemented [39]. They include the global positioning system (GPS) to find current location of the UAV, sensors and data handling unit (e.g., a gyroscope, accelerometer, and magnetometer sensors), and a communication module.

Data communication between a UAV and a ground control station (GCS) is fundamental for a successful completion of its operation, and has received considerable attention both in military and civilian domain. The feasibility of employing orthogonal frequency division multiplexing (OFDM) techniques was investigated for transmission between a UAV and a ground control station in [40]. In [41], the authors studied the performance of a distributed transmit beam forming and distributed orthogonal space time block coding (OSTBC) schemes for a UAV to GCS communication link. Control and co-ordination of a network of UAVs have also been investigated in [42], where genetic algorithms were used to create decision trees which allowed UAV collaboration in search missions. The use of UAVs for the extension and improvement of networked communication in a natural disaster scenario has also been considered. For example, a network architecture has been devised with UAVs as relay platform to form equivalent cellular towers in the sky for rapid deployment of wireless networks [43].

5. CONCLUDING REMARKS

This paper describes the design of an optimal controller for an unmanned aerial vehicle. The analysis included several controllers such as IPID, FPID, and Mamdani fuzzy controller. The methodology that followed to design the optimal and robust control scheme was based on research, numerical simulations. The preliminary design resulted in the selection of the Mamdani fuzzy controller for the purpose of stabilizing the spacecraft.

The key novelty presented here is the composite approach to designing complex systems at the final-project undergraduate project, involving a three-prong approach: (i) a capstone project, (ii) two research projects related to the Orbo UAV conducted at the Department of Electrical and Computer Engineering, University of Manitoba, and (iii) interaction with industry professionals developing the Orbo UAV. This makes the project quite unique and different. During these projects, undergraduate students were able to acquire and utilize many engineering skills, such as time management, problem solving, communication skills, and team-working synergy. Interaction with graduate students provided additional understanding of the navigational algorithms used in the research projects, experience from their implementation at a graduate level, and techniques not
taught in the undergraduate classes. Interaction with a very experienced pilot in the field provided another perspective on the relevance of theory and practice. These attributes make the project particularly well-suited to achieving the capstone learning outcomes.

Acknowledgements

We would like to acknowledge partial financial support from the Department of Electrical and Computer Engineering, University of Manitoba. We are also grateful to Buoyant Aircraft Systems International (BASI) for providing access to their UAVs and aerial vehicles.

References


[18] Witold Kinsner, Switching and Automata Theory. Course Notes. Winnipeg MB: Department of


