Fuzzy Logic Control of a Four Rotor Autonomous Aerial Platform

Patrick Mills
NeverLand Software & Systems
E-mail: pmills@ieee.org

Abstract

This paper examines a novel aerial platform utilizing fuzzy logic control with evolutionary tuning. First the new vehicle is introduced, then its dynamics are described. The development process and simulation techniques are then discussed. The fuzzy logic controller derived from the simulations is explained and then the physical vehicle built on the simulation model is examined. Finally, the testing results are presented and future directions are examined.

1 Introduction

Unmanned Aerial Vehicles, UAV, facilitate exploration, reconnaissance, and rescue where human presence is difficult, dangerous, or expensive. Helicopters are the most common vehicle used when maneuverability is a primary goal. However, helicopters suffer from a number of drawbacks, one of the biggest being mechanical complexity. A four rotor aerial platform, FRAP, with flight characteristics similar to a helicopter offers reduced design complexity and therefore increased reliability.

The FRAP is controlled by applying forces and generating moments about the three axes: pitch, roll, and yaw. Forward motion is generated by decreasing thrust to the forward rotor while increasing thrust to the rear rotor. Backward motion is generated in an opposite manner. Right and left motion are similar. In order to balance the FRAP, the forward and rear rotors spin in a clockwise direction while the left and right rotors spin in a counter-clockwise direction. The FRAP may be rotated counter-clockwise by increasing thrust to the forward and rear rotors while decreasing thrust to the right and left rotors. A clockwise rotation is created in a similar manner.

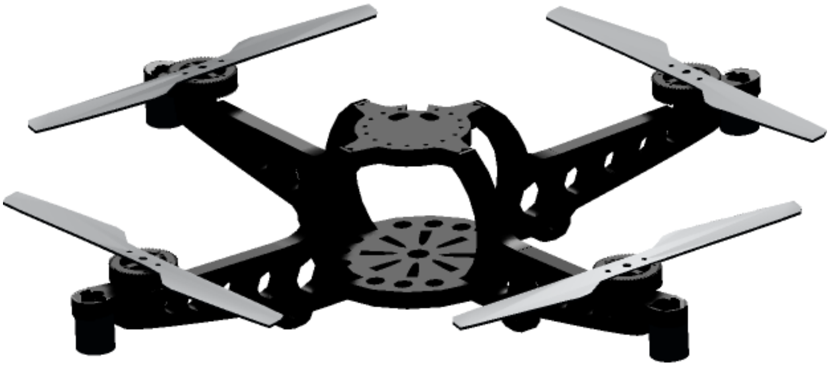

Figure 1: **Four Rotor Aerial Platform**

While a basic helicopter requires a tilting rotor, variable pitch blades, stabilizer fins, and a high speed stabilizer rotor, an FRAP requires only four fixed pitch rotors. Furthermore, control is achieved by varying the velocity of each rotor. This means that complete control of the vehicle can be achieved with four controls: move up / down, move forward / reverse, move left / right, rotate clockwise / counter-clockwise. Although the FRAP's control system is quite simple, the platform's flight dynamics are intrinsically unstable.

In order to develop a reliable control system for the FRAP, it is useful to understand the vehicle's dynamics. This paper develops a dynamics model for the FRAP and through simulation produces a suitable control system. The control system is then implemented in an actual FRAP and tested.

2 FRAP Model

The FRAP model presented here has been simplified to allow rapid modeling. The platform is considered as a rigid body moving in three dimensional space in reaction to forces and moments. Full advantage of the design's symmetry is taken; moments of inertia are calculated using simplified shapes. Following Lee *et al.*, August 1993:

$$x = [\mathrm{P}\ \mathrm{v}\ \mathrm{R}\ {}^{b}\omega]^{T} \in \Re^{3} \times \Re^{3} \times SO(3) \times \Re^{3} \tag{1}$$

Where P and v are the position and velocity vectors of the center of mass in spatial coordinates, R is the rotation matrix of the body axes relative to the spatial axes, and ${}^{b}\omega$ is the body angular velocity vector. Thus:

$$\begin{bmatrix} \dot{P} \\ {}^{p}\dot{v} \\ \dot{R} \\ {}^{b}\dot{\omega} \end{bmatrix} = \begin{bmatrix} {}^{p}v \\ \frac{1}{m} R\,{}^{b}f \\ R\,{}^{b}\hat{\omega} \\ I^{-1}({}^{b}\tau - {}^{b}\omega \times I\,{}^{b}\omega) \end{bmatrix} \tag{2}$$

As shown in Figure 2, the FRAP can be modeled as a center platform, four arms, four motors, and four rotors. We treat each rotor as a source of pure vertical force. Thus:

$$ {}^{b}f = \begin{bmatrix} 0 \\ 0 \\ \sum_{1}^{4} F_{T\,i} \end{bmatrix} + R^{T} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \qquad {}^{b}\tau = \begin{bmatrix} 0 \\ 0 \\ \sum_{1}^{4} Q_{T\,i} \end{bmatrix} + \begin{bmatrix} 0 \\ \sum_{1}^{4} lF_{T\,i} \\ 0 \end{bmatrix} \tag{3}$$

Where F_{Ti} is the Force of Thrust due to Motor i. It should be noted that when opposite rotors are moving at the same velocity, part of their moments sum to zero; when all rotors are moving at the same velocity, all moments sum to zero. Thus in an idealized steady hover, the only forces acting on the body are the four F_T, and the force of gravity.

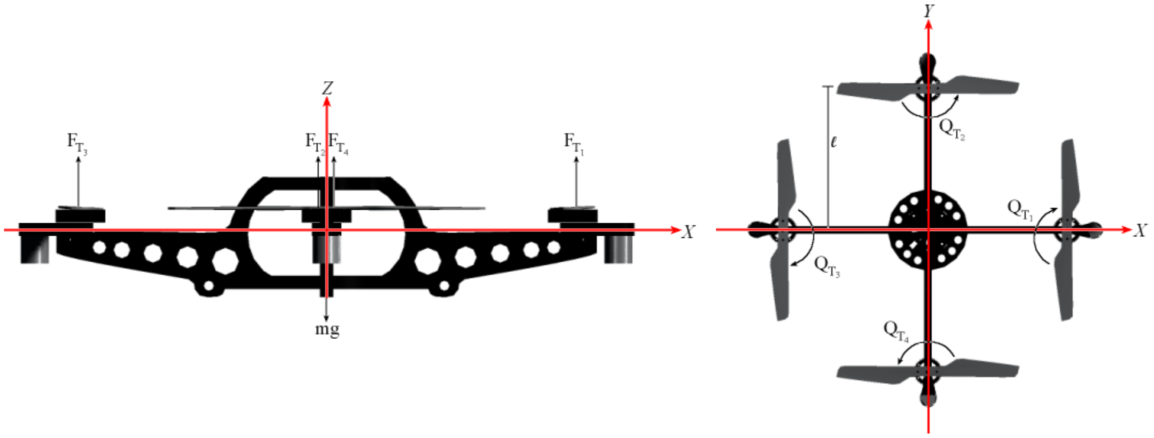

Figure 2: **FRAP free-body diagram**

3 FRAP Simulation

Before any work was done on the physical FRAP, a three dimensional wire frame simulator was created to test the equations derived from the idealized model. The derived equations were further augmented by additional theory necessary to model flight.

Blade element theory was used to model the thrust, T, generated by each rotor. This method also provides a value for the generated torque, Q.

$$\begin{aligned} T &\approx \tfrac{1}{6}\rho(V_f^2+\Omega^2 r^2)Bcr(C_L\cos\phi - C_D\sin\phi) \\ Q &\approx \tfrac{1}{8}\rho(V_f^2+\Omega^2 r^2)Bcr^2(C_L\sin\phi - C_D\cos\phi) \end{aligned} \qquad (4)$$

Where B is the number of blades on the rotor, ρ is the density of Air, V_f is the forward velocity in m/s, Ω is the speed of the rotor in rad/s, c is the chord length of the rotor in m, r is the radius of the rotor in m, C_L is the coefficient of lift, C_D is the coefficient of drag, θ is the blade angle, α is the angle of attack, and $\phi = \theta - \alpha$.

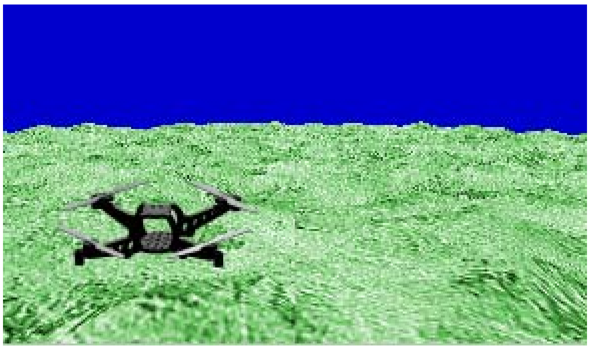

Figure 3: **FRAP Simulation Screen Shot**

When the wire frame simulation proved successful, a fully rendered three dimensional interactive simulator was created. Figure 3 shows the simulator in action. The simulator features a full physics engine which allows calculations such as position, velocity, and acceleration. However, in order to be truly useful, the simulator was enhanced to model features such as motor temperature, current usage, battery power, and even simple air resistance. Impact forces in FRAP / ground collisions were also modeled; future versions will include damage modeling of motor burnout, and rotor and frame failures.

The most important feature of the final simulator was to allow rapid prototyping of various control systems. One of the features incorporated into the simulator is a genetic algorithm generator which dynamically creates and tests various control systems using a set of standard flight maneuvers and a heuristic evaluation function.

The evaluator attempts to find a control system that is capable of maintaining a stable hover and executing low speed movement with the minimal number of sensors. The best control system found by the genetic search algorithm was implemented as an optimized fuzzy logic control system.

4 Fuzzy Logic Control Implementation

In general, a Fuzzy Logic controller is a system where at least some of the variables are represented by fuzzy sets. By representing the state of a variable with a fuzzy set, the system quantizes the variable not as a fixed value, but as a graded membership in a set of values. The graded membership allows the fuzzy system to deal with uncertainty. Fuzzy Logic controllers are implemented as a set of *if-then* rules. More than one rule may be active at a time; moreover, a rule may be off, on, or partially on.

Most fuzzy controllers use knowledge elicited from human experts in the form of rules. The FRAP implementation began as a set of generic rules designed to allow the vehicle to enter a stable hover. The genetic algorithm tweaked the fuzzy sets and provided a more specific solution for our goals. The final simulated control system consisted of three inputs, the Euler angles ϕ (pitch), θ (roll), and ψ (yaw) representing the body orientation to some local reference frame, and four outputs, the velocity of each of the four rotors. The system's main goal is to maintain a stable level hover.

Each cycle, the controller evaluates the three input variables. The controller's goal is to keep the rate of change of each of the input variables at zero. Each input variable and its computed derivative are fed into the fuzzy controller as crisp inputs. The rule set is shown in Figure 4.

		dAcc						
		NS_3	NS_2	NS_1	AZ	PS_1	PS_2	PS_3
Acc	NS_3	PS_3				PS_2	AZ	
	NS_2							
	NS_1			PS_2	PS_1	AZ		
	AZ	PS_2		PS_1	AZ	NS_1	NS_2	
	PS_1			AZ	NS_1	NS_2		
	PS_2	AZ		NS_2	NS_3			
	PS_3							

Figure 4: **Fuzzy Rule Set**

The output of the fuzzy controller is a crisp number which is used to modify the rotational velocity of each rotor as follows:

$$\begin{aligned}
\text{Rotor1} &= \text{Thrust} + \text{Pitch}_{FO} - \text{Yaw}_{FO} \\
\text{Rotor2} &= \text{Thrust} - \text{Roll}_{FO} + \text{Yaw}_{FO} \\
\text{Rotor3} &= \text{Thrust} - \text{Pitch}_{FO} - \text{Yaw}_{FO} \\
\text{Rotor4} &= \text{Thrust} + \text{Roll}_{FO} + \text{Yaw}_{FO}
\end{aligned} \tag{5}$$

These equations are best explained with an example. When the platform is spinning counter-clockwise about the z axis, ψ is between $[0,\pi]$ and increasing, the fuzzy controller wants to increase the rotational velocity of Rotors 2 and 3 while decreasing rotational velocity of Rotors 1 and 4 thus creating an clockwise torque around the z axis.

5 Results

In building the physical FRAP, several changes to the fuzzy logic controller had to be made. First, it was deemed too expensive to implement the hardware necessary to calculate the Euler angles directly, so an alternative was implemented. Three ceramic gyroscopes, which calculate the angular acceleration around a single axis, were mounted at right angles to each other. In this configuration the rate of change of the platform's motion can be measured, but not absolute orientation. The fuzzy logic controller sensor inputs then become the acceleration around each axis.

The controller's main objective is to stabilize the platform. It utilizes the accelerometer readings as error values and attempts to balance the platform so that the acceleration values approach zero. As with the Euler angles, the rate of change of the value is calculated by the controller as a derived input.

In the microprocessor implementation of the fuzzy logic controller, the centroid method of defuzzification was computationally unfeasible. A weighted mean of maxima method was implemented instead.

In testing, the FRAP was unable to achieve a stable hover with the fuzzy logic controller. As the platform gained altitude, it tended to oscillate with increasing amplitude between unstable states. Eventually the oscillations would become so severe that the platform would flip over and crash.

To evaluate the problem, the values of the fuzzy sets were modified and the test maneuvers were performed again. The system performed better on some tests, and worse on others. A classical non-linear control system was tested to determine if a physical defect was preventing the correct functioning of the system. However, this did not succeed either. After several tests, it was determined that the resolution of the sensors was too coarse to detect the fine movements necessary to stabilize the platform. Increased resolution resulted in a platform that could maintain a hover.

Further tests showed that while the system could maintain a hover, it did not transition well to forward flight. In order to fine tune the fuzzy rules, an evolutionary tuner was implemented. The tuner dynamically modifies the fuzzy sets as well as modifying the weights for the mean of maxima defuzzification method. The final controller can be seen in Figure 5.

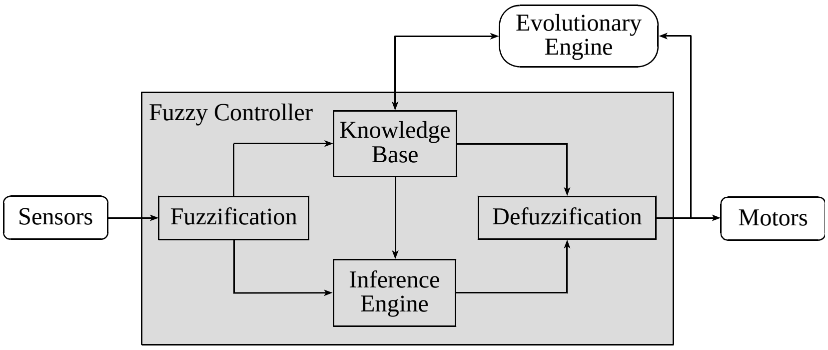

Figure 5: **Final Fuzzy Controller**

Approximately 30 times a second, the controller reads the navigation input and attempts to stabilize the platform in a hover. The success of each output is evaluated during the next cycle and the evolutionary tuner modifies the fuzzy sets as necessary to adjust them towards an optimal result.

Once the hover calculations have been completed, the outputs are perturbed to provide the necessary motion requested by the navigation system.

6 Conclusion

With the evolutionary tuner optimizing the fuzzy sets, the platform quickly achieves a stable hover. Even when an impulse disturbs the platform, it recovers quickly.

At a stable hover four feet above the ground, a three ounce weight suspended from a string was clipped to one of the platforms arms at the midway point. The platform shifted in the direction of the weight, but quickly regained equilibrium as the system adjusted to the new parameters.

During testing, a chunk of a rotor blade was accidentally broken off during a crash. Even though the loss of part of its leading edge created an unbalanced system, the controller was able to compensate and maintain a stable hover.

In most cases, there is enough turbulent air flow in a room to keep the platform in constant motion and therefore keep the controller making minute adjustments. However, as the controller is unable to detect any motion that occurs during zero acceleration around an axis, uncontrolled drift was detected on some occasions. Any perturbation to the system tended to correct the problem. Use of Euler angles would provide a reference value that would eliminate this problem.

7 Future Work

In the next incarnation of the FRAP, a faster processor will be used to allow implementation of the actual Euler angles. A piece of hardware is now available that provides all the necessary information in one package, including a ceramic gyroscope, accelerometer, and a terrestrial magnetometer. These inputs will allow direct computation of Euler angles relating the body frame to an initial local frame as in the final simulated fuzzy logic controller. The addition of a position sensor would allow the elimination of the minimal drift in the current model.

One of the initial design goals which was not met by this version of the FRAP was safety monitoring. The physical device was designed to be able to read its current state and respond to out-of-bounds conditions by shutting down. In practice, this behavior was difficult to achieve. Future versions will attempt to implement this feature successfully; the ultimate goal being a quick controlled landing as soon as any error conditions, such as uncontrolled oscillations or low power, occur.

Another goal is to add a vision system to the FRAP; however, at a little over two pounds, the platform can currently stay aloft for only three to four minutes per battery charge. If more sensors are added, a lighter system must be designed, the rotor efficiency must be increased, or a better battery must be used.

References

[1] T. Alderete. Simulator Aero Model Implementation. NASA Reference Publication 1373, 1995.

[2] R. Heffley and M. Mnich. Minimum-complexity helicopter simulation math model. Technical Report NASA Contractor Report 177476, April 1988.

[3] F. Hoffmann and G. Pfister. Evolutionary design of a fuzzy control rule base for a mobile robot. *International Journal of Approximate Reasoning*, 17(4), 1997.

[4] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, 1995.

[5] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic*. Prentice Hall, 1995.

[6] T. Koo, F. Hoffmann, B. Sinopoli, and S. Sastry. Hybrid control of model helicopter. *Proceedings of IFAC Workshop on Motion Control*, Grenoble France, 1998.

[7] E. Lee, H. Shim, H. Park, and K. Lee. Design of hovering attitude controller for a model helicopter. *IEEE International Society Instrument and Control Engineers*, 1993.

[8] B. McCormick. *Aerodynamics Aeronautics and Flight Mechanics.* John Wiley and Sons, 1995.

[9] R. McFarland. A Standard Kinematic Model for Flight Simulation at NASA - Ames. Technical Report NASA Contractor report 2497, January 1975.

[10] G. Padfield. *Helicopter Flight Dynamics*. AIAA, 1995.

[11] C. Phillips, C. Karr, and G. Walker. Helicopter flight control with fuzzy logic and genetic algorithms. *Engineering Applications of Artificial Intelligence*, 9(2), 1996.

[12] R. Prouty. *Helicopter Performance, Stability, and Control.* Krieger Publishing Co, 1995.

[13] M. Sugeno, I. Hirano, S. Nakamura, and S. Kotsu. Development of an intelligent unmanned helicopter. *IEEE International Conference on Fuzzy Systems*, vol 5, 1995.

[14] D. Shim, T. Koo, F. Hoffmann, and S. Sastry. *A Comprehensive Study of Control Design for an Autonomous Helicopter*. 37th IEEE Conference on Decision and Control, 1998.