

Modelling and Building of a Monocopter

Christophe van der Walt, Carl Beekhuizen, Patrick Bos, and Roos de Vries

Abstract— Mechanically Simple, Biologically Inspired: Samara or sycamore seeds have been the inspiration for many flying craft over the years. In this report, one such a craft, a monocopter capable of translational motion in the x, y, and z directions, is developed. The development process is described with a theoretical discussion, two models, design specifications, basic programming, signal theory, and testing.

CONTENTS

I	Introduction	1
II	Theory	1
II-A	The model	1
II-B	Simulation	3
II-C	Camber, Angle of Attack and Maximum Lift	3
II-D	Translational Flight	4
III	Design	4
III-A	Goals and Limitations	4
III-B	Design Choices	4
III-B.1	Hardware	4
III-B.2	Electronics	4
III-C	Solidworks Modelling and Visualization	5
IV	Programming and Signal Theory	6
IV-A	Inputs	6
IV-A.1	HMC5883L Compass Sensor	6
IV-A.2	FrSKY D4R-ii Receiver . .	6
IV-B	Outputs	6
IV-C	Translational flight and control mixing	6
V	Experimental Process	8
V-A	Setup and Sub-Experiments	8
V-A.1	Testing the System Outputs	8
V-A.2	Testing the System Inputs .	8
V-A.3	Integrated System and Control	8
V-B	Main Experiment: Vertical Flight . .	9
V-C	Discussion	10
VI	Conclusion	10
References		
Appendix		10
A	Reflection on Project, Group Dynamic	10



Fig. 1. A samara seed [13]

A samara is a seed with a winglike filament which causes it to spin as it falls to the ground. This filament is very light, so that the center of mass remains in the seed itself. The

blade experiences drag from the air. Because of this, the samara begins to spin under the influence of gravity pulling it to the ground. In essence, a samara seed is a single blade extending from a center of mass. When a vertical force is applied, this system begins to spin. The system is stable and behaves in a predictable manner. This simple idea can be recreated in a monocopter.

A monocopter is a flying craft which rotates about its center of mass, pulling a wing through the air in order to generate the required amount of lift. Because a monocopter is much heavier and larger than a samara seed, rotation is achieved through a motor and propeller mounted on an arm away from the center of mass. This motor can be given enough power to overcome the force of gravity and cause the monocopter to spin upward into the air. In effect, a monocopter is a single bladed helicopter.

Much like in helicopters, control of a monocopter is achieved by varying the amount of lift generated by the rotor/wing at different points in its rotation. In helicopters, this is done via a swash plate, whereas in this case, it will be achieved by changing the camber of the wing via a servo, creating a virtual swash plate.

This report describes the development of a monocopter. The development process entailed mathematically understanding the flight, simulating the flight, designing the device, programming system device, testing the device, and evaluating.

II. THEORY

A. The model

Modeling Context and Assumptions:

The monocopter is modeled here as a wing (made with an ag455ct02r-il airfoil Fig. 3), tapered as shown in Fig. 2 and a distributed mass on the inner side of the wing, connected by a spar. It is powered by a motor at a distance r_1 from the center of mass. The center of mass is located along the base of the wing, close to the leading edge.

Forces and torques taken into account here include gravity (\vec{W}), centripetal force (\vec{a}), the gyroscopic precession torque ($\vec{\tau}$), a driving force by the motor (\vec{F}_u), drag (\vec{D}) and lift (\vec{L}). Both lift and drag are assumed to be negligible for all parts of the system barring the wing. The rest of the forces

I. INTRODUCTION

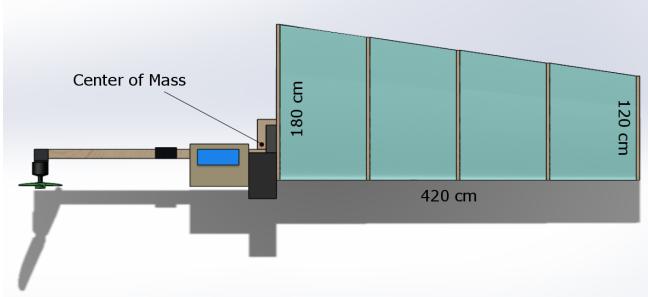


Fig. 2. The designed monocopter, with its centre of mass and wing dimensions

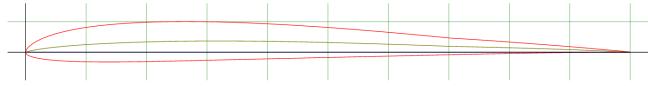


Fig. 3. The ag455ct02r-il airfoil

involved are considered here in their full capacity.

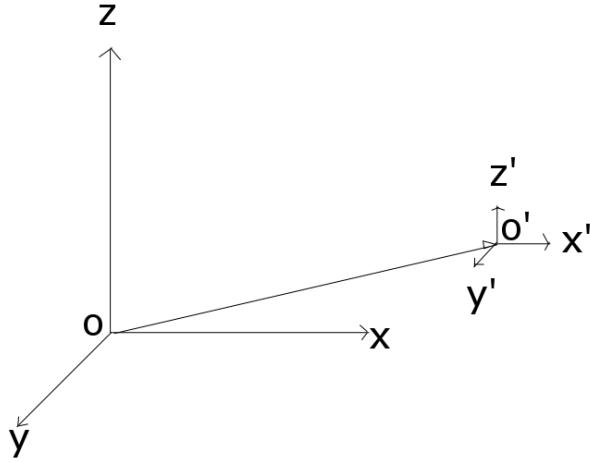


Fig. 4. The two reference frames : O is stationary and O' is the centre of mass

Two reference frames are considered here. The first one is stationary to the ground. The exact orientation of the x and y axes and the origin are unimportant, but the following is to be observed : the x and y axes are horizontal and orthogonal to each other, and the z axis is upward-pointing and orthogonal to the first two. The second reference frame's origin is centred at the centre of mass and rotation. The x' axis points outward towards the tip of the wing, parallel to the rod. The y' axis points in the direction parallel to the instantaneous plane of rotation, orthogonal to the x' axis, against the direction of motion. The z' axis is the axis pointing in the same direction as lift is being generated, orthogonal to the x' and y' axes.

Transition between these reference frames is assured by a 3-dimensional rotation matrix shown in Fig. 5. The make-up of which is described below. Here $a = \theta_1, b = \theta_2, c = \theta_3$,

where the θ s are the components of angular position according to standard nomenclature. This matrix is then multiplied to whatever vector needs to be changed from the fixed reference frame to the one centred at the centre of mass.

See below for a more detailed description of the forces taken into account :

- Gravity :

$$\vec{W} = m\vec{g} \quad (1)$$

, where \vec{g} is earth's gravitational field at its surface

- Centripetal Force :

$$\vec{a} = m\dot{\vec{\theta}} \times \vec{V}_{cm} \quad (2)$$

, where $\vec{\theta}$ is the craft's angular position vector and \vec{V}_{cm} is the craft's translational velocity, both in the first reference frame

- Gyroscopic procession torque :

$$\vec{\tau} = \vec{\theta} \times J\dot{\vec{\theta}} \quad (3)$$

, where J is the first moment of inertia tensor, determined by a Solidworks model.

- Drag :

$$\vec{D} = C_d \frac{\rho}{2} A \left(||\vec{V}_{cm}||^2 L + ||\vec{V}_{cm}|| ||\dot{\vec{\theta}}|| L^2 + ||\dot{\vec{\theta}}||^2 \frac{L^3}{3} \right) \quad (4)$$

, where C_d is the angle of attack dependent drag coefficient, ρ is the density of air, L is the length of the wing, and A is the projective facing area.

- Lift :

$$\vec{L} = C_l \frac{\rho}{2} A \left(||\vec{V}_{cm}||^2 L + ||\vec{V}_{cm}|| ||\dot{\vec{\theta}}|| L^2 + ||\dot{\vec{\theta}}||^2 \frac{L^3}{3} \right) \quad (5)$$

, where C_l is the angle of attack dependent lift coefficient.

The aforementioned aerodynamic forces (lift and drag), were determined from the standard formulas for these, namely

$$\vec{D} = C_d \frac{\rho}{2} A \cdot V^2 \quad (6)$$

$$\vec{L} = C_l \frac{\rho}{2} A \cdot V^2 \quad (7)$$

However, since the wing is rotating in space, different points on the wing are travelling at different velocities. This means that equations (6) and (7) become (assuming the lift and drag coefficients are independent of distance along the wing) :

$$\vec{D} = C_d \frac{\rho}{2} A \int_0^L \left(R(-\vec{\theta}) ||\vec{V}_{cm}|| + r ||\dot{\vec{\theta}}|| \right)^2 dr \quad (8)$$

$$\vec{L} = C_l \frac{\rho}{2} A \int_0^L \left(R(-\vec{\theta}) ||\vec{V}_{cm}|| + r ||\dot{\vec{\theta}}|| \right)^2 dr \quad (9)$$

, where $R(-\vec{\theta})$ is the rotation matrix that transforms the craft's translational velocity into the second reference frame (the one that is stationary to the monocopter). These integrals solve to become equations (4) and (5).

$$R(\theta) = \begin{bmatrix} \cos(b)\cos(c) & -\cos(b)\sin(c) & \sin(b) \\ \sin(a)\sin(b)\cos(c) + \cos(a)\sin(c) & -\sin(a)\sin(b)\sin(c) + \cos(a)\cos(c) & -\sin(a)\cos(b) \\ -\cos(a)\sin(b)\cos(c) + \sin(a)\sin(c) & \cos(a)\sin(b)\sin(c) + \sin(a)\cos(c) & \cos(a)\cos(b) \end{bmatrix}$$

Fig. 5. The 3D rotation matrix

The lift and drag coefficients are assumed independent of radius/distance along the wing, for simplicities sake, mostly, since information about such a radius dependence would only be the result of extensive wind tunnel testing, and would likely only be discretely integrable as well.

The only other assumption made is that the motor is considered to be a point load, so any gyroscopic effects or torques induced by the motor are ignored.

The model itself:

Certain notation conventions are adopted here. Net aerodynamic force is considered for simplicity's sake, and can be written in the reference frame stationary to the craft as $\vec{F}_{aero} = (0, -D, L)^T$. The input force is also considered in the same reference frame as this too makes things easier. Then, $\vec{F}_u = (0, F_u, 0)^T$.

The model, obtained by applying Newton's Second Law, is then :

$$\begin{cases} m\vec{V}_{cm} = R(-\vec{\theta})\vec{F}_{aero} + m\vec{g} - m\dot{\vec{\theta}} \times \vec{V}_{cm} + R(-\vec{\theta})\vec{F}_u \\ J\ddot{\vec{\theta}} = r_1\vec{F}_{aero} - \vec{\theta} \times J\dot{\vec{\theta}} + r_2\vec{F}_u \end{cases} \quad (10)$$

, where r_1 is the distance from the center of mass to the center of pressure (the virtual application point of the aerodynamic forces), and r_2 is the distance from the center of mass to the motor.

B. Simulation

A simulation of the mathematical model was made in Simulink. The block diagram of this is shown in the Appendix B (Fig 16). This can output various system characteristics for inputs of motor force, lift coefficient, and drag coefficient as a function of time. These last two forces can be connected to angle of attack by an experimentally determined function.[12]

The validity of the simulation is quite difficult to prove, since there is no reliable data against which to compare it. However, it was shown that the model is consistent with itself (i.e. no matrix dimension errors or other errors when simply running the program for random inputs).

C. Camber, Angle of Attack and Maximum Lift

Generating maximum lift in terms of the angle of the aileron is not as simple as it may outwardly appear. Both ailerons and fully actuated rotors (as in the case of helicopters) have been thoroughly studied, however, this project

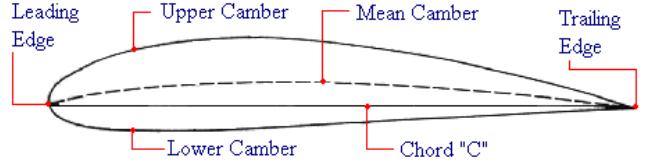


Fig. 6. Diagram depicting the mean camber and aerodynamic chord of a wing.[2]

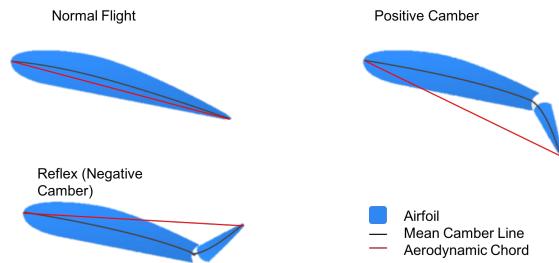


Fig. 7. A wing exhibiting camber and reflex.

is an amalgamation of the two and as such, there is no clear research on the topic. Helicopters achieve varying amounts of lift via controlling the angle of the blade relative to the airflow and thus forcing more or less amounts of air downwards, generating the thrust required for flight. The effect ailerons have on the lift generated by a wing can be described in two manners.

Firstly, the air can be viewed as being deflected by the aileron. In this paradigm if the aileron is deflected downwards, air is forced downwards and thus, by Newton's Third Law, the rear of the wing rises which forces more air downwards. The second interpretation is through the aileron's influence on the effective camber of the wing and the associated changes in the lift coefficient. The mean camber line is the line halfway between the upper and lower wing surfaces. The chord line is a line connecting the leading and trailing edges of a wing. Camber is a measure of how much these lines deviate from one another, as can be seen in Fig. 6. The greater the amount of camber, the greater the amount of lift generated by the wing. The wing can also make use of reflex (negative camber) to produce "negative lift" this is depicted in Fig. 7 Due to the centre of lift being aft of the centre of mass, increased lift produces a torquing arm which in turn pitches the wing downwards and thus less air is pushed downwards and less thrust is produced.

D. Translational Flight

Translational flight in monocopters is achieved in much the same manner as in helicopters. As the wing rotates, it varies the amount of lift it generates, which in turn exerts torques on the system which affects the angle of inclination between the rotating disk and the horizon. In effect, a virtual swashplate (a device which translates a helicopter pilot's control inputs to motion of the blades) is created. One key consideration is the large angular momentum experienced by the craft and the associated gyroscopic procession of all control torques. Thus, the torques generated by the lift of the wing must occur 270° before the desired rotational direction of the wing disk.

III. DESIGN

A. Goals and Limitations

The following list of design goals was developed as a result of limitations inherent for flight or imposed on the situation.

- The device must have components which make it capable of moving freely in the x, y, and z directions.
- The device must be able to communicate with a radio controller. It must be able to translate simple commands into complex actions on board.
- Because the device will be spinning very quickly, every component must have a very fast processing or reaction time. The entire device must respond as quickly as possible.
- Every component in the device must be chosen to be as light as possible.
- To maximize manoeuvrability, every component which is to supply power or torque must be as strong as possible.
- Every component (excluding the wing) must have as little drag as possible.
- Every hardware component must be able to sustain considerable torque during flight, estimated at about 100 g at maximum arm. Every component of the device must be able to survive an impact with the ground.
- Components which can be acquired locally are preferred. To reduce shipping costs, as many components as possible must be ordered from the same source.
- The device must be able to be built and tested within a two week timespan. Every component must arrive within two to three weeks, because this is the time between when the components are ordered and the beginning of the build.
- The combined price of all components in the device may not exceed 100 Euros.

B. Design Choices

As a result of the list of design goals, the following components were deemed necessary.

- A frame to support the wing, motor, and electronics at a desirable position relative to the center of mass
- A wing with an aileron
- A motor to power the propeller

- An ESC to control power to the motor
- A servo to adjust the angle of the aileron
- A battery to power the servo and motor
- A set of accelerometers or a gyroscope to determine the rotational speed and position of the monocopter
- A radio controller to communicate to a radio receiver
- A programmable device to coordinate the electronic components
- A positive mental attitude

1) Hardware:

Frame

The trade-off in choosing a material for the frame is between weight and strength. Many materials, such as carbon fiber and diverse metal alloys were considered; however, these materials were not easily available. Eventually 4mm thick processed poplar wood was chosen as the support of the monocopter because it was readily available, and extremely strong for its weight. The main beam and spars to support the wing were laser cut.

Wing

The wing had to be lightweight, sturdy, and made into the precise shape of the airfoil. To achieve this, a material was needed which could be easily formed. Blue foam was available for use, however, it was predicted that this material would not survive an impact. Clear packaging tape was put over the edges of the foam to protect it. Another downside was that blue foam corrodes when exposed to many forms of glue. Slow drying glues were used to seal the foam to its wooden supports.

Aileron

The aileron needed to be made of a lightweight, sturdy material which would not twist or bend under high pressure from air drag. Plastics and metals were too heavy to use for an entire plate. Eventually, foam board was settled on. To attach the aileron to the foam wing, a few layers of clear packaging tape were used.

Servo

Connection A clear plastic control horn to connect the servo push rod to the aileron was laser cut out of clear plastic. It was glued to the aileron using super glue and hot glue. The push rod was made out of a thin metal rod soldered to two clevises. Each piece was made very precisely to minimize friction as well as shakiness.

Electronics Base

The electronic wiring was placed on a fibreglass veroboard. After all of the wiring was soldered on, the veroboard was cut to as small a size as possible and hot-glued to the main beam, close to the center of mass.

2) Electronics: Each electronic component that was needed was first researched online. A few plausible options were combined into a list with all the important characteristics of the component (power, price, weight, speed, acquiring it etc). From these lists the most suitable components were chosen.

Motor

A DYS SE1407 3600kv motor was

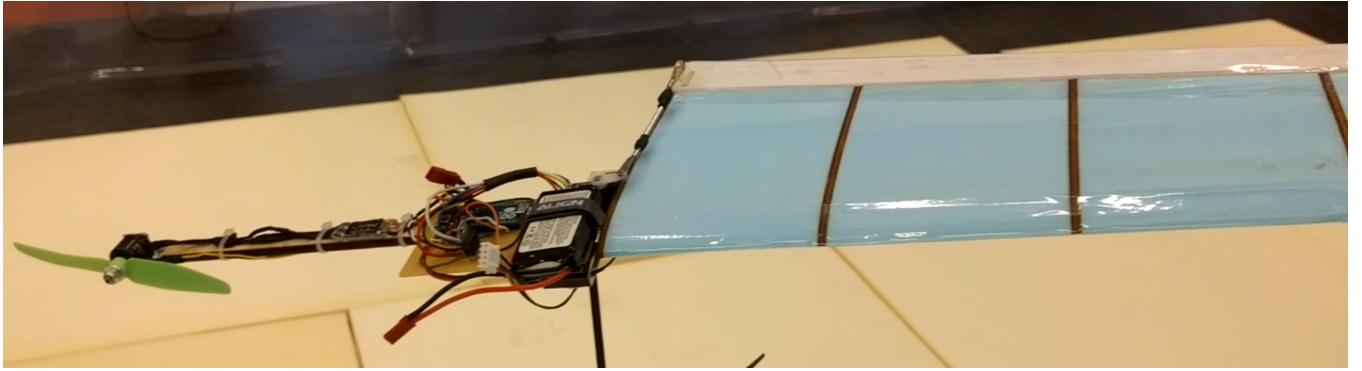


Fig. 8. The completed device assembly

selected. This motor was compact, but had a thrust of 400g, which was considerably more than what was predicted to be the minimum needed thrust (150g). The motor weighed in at 13.95 g, and cost about 19 Euros.

ESC

For an ESC, the Littlebee 20A (BL-Heli) was chosen because it could offer 20 Amps to the motor and weighed the least of any ESC encountered, at just 4 g. The price of the component was 15 Euros.

Servo

The Servo that was chosen was the Savox SH-0262MG. This servo weighed 13.6 grams and cost 21 Euros. It had the quickest reaction time of any servo found, and a metal gear for the arm for extra strength.

Battery

A Sportpower 3S 450mAh battery was selected because this battery could discharge most quickly. The weight was 35 g, and the price was 11 Euros. The battery was one of the heaviest components. It was secured with velcro so that it could easily be removed to be recharged and so that it could be re-positioned to effect the center of mass.

Magnetometer

The HMC5883L magnetometer circuit was chosen for its price: 12 Euros. The weight of this small component was assumed to be negligible (2 g).

Accelerometers

A digital accelerometer (Grove 16G) and an accelerometer and gyroscope (Grove 6DOF) were chosen, each costing 12 Euros. *Note: The accelerometers were not used in the final design.*

Radio and Receiver

One of the project group members already owned a radio and receiver, a FrSKY Tarans 9XD Plus and a FrSKY D4R-ii respectively.

Arduino

One of the project group members already had an Arduino micro, on which programs could be loaded to coordinate all of the other electronic components with each other. This group member knew how to program and correctly wire this Arduino.

All of the electronic components were mounted on the frame and vero board. The motor and servo were placed at

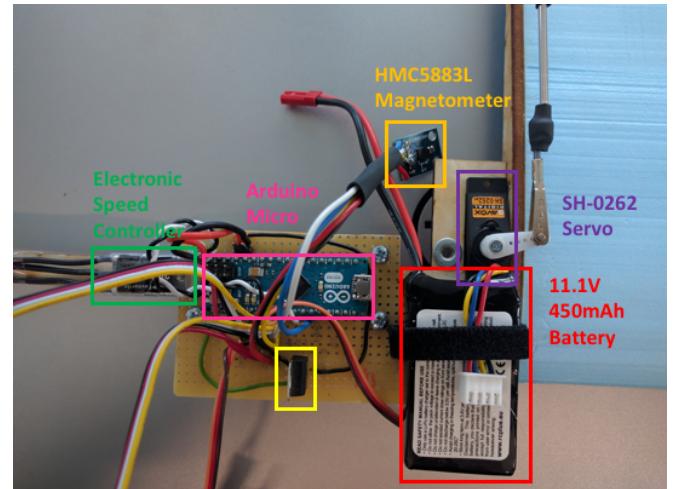


Fig. 9. The basic electronics appearing on the device.

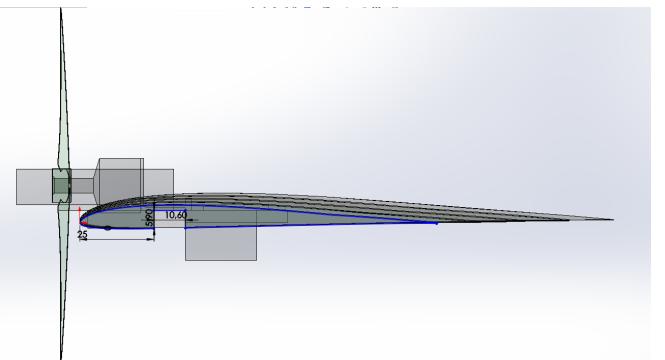


Fig. 10. Solidworks allows parts to be designed clearly and precisely.

desirable positions for the functionality of the device, with small hardware components to hold them in place. The other electronic components were wired and soldered together and to the vero board, as close to the center of mass as possible. Tie wraps were used to safely secure the wires.

C. Solidworks Modelling and Visualization

Some of the parts used in the design, such as the wooden frame of the craft and the control horn, were made using

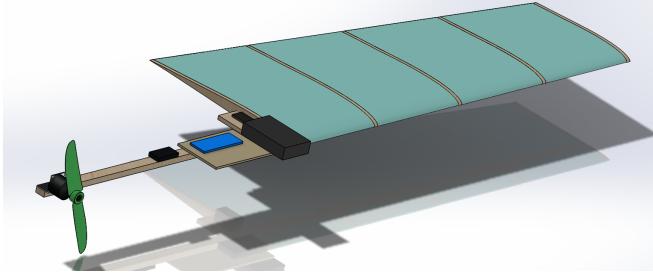


Fig. 11. Solidworks visualization of design, used to estimate moment of inertia tensor.

Solidworks. This allowed for using the laser cutter with which a very high precision was achieved. Solid works was a handy tool to design every component clearly and precisely. Furthermore, it also allowed the group to visualize the product ahead of time, and perform rough calculations about the center of mass and moment of inertia tensor. These characteristics of the system could then be used in the Simulink model.

IV. PROGRAMMING AND SIGNAL THEORY

A. Inputs

The monocopter has two primary inputs, the HMC6883L compass sensor and the FrSKY D4R-ii receiver which transmitted the control inputs.

1) *HMC5883L Compass Sensor:* By setting the correct registers within the sensor, the highest continuous achievable data-rate is 75Hz. While this may be adequate for most applications, due to the high rotational velocity of the monocopter, more was required. The solution hereto was to utilise single measurement mode, but with an interrupt triggering whenever new data becomes available. Whenever the interrupt pin was pulled low by the sensor, the Interrupt Service Routine (ISR) would set a flag and upon the next iteration of the primary control loop, the new compass data would be processed and new data requested. This allows for data to be retrieved at 160-170Hz, which assuming a rotational rate of 8RPS, means new compass data is retrieved every 18 degrees of revolution. [3] The raw data retrieved from the compass sensor is in the form of angles about the X,Y and Z axes. However, in order to be useful, the rotation has to be about the axis of rotation of the craft, thus atan2 is used to resolve the angle.

2) *FrSKY D4R-ii Receiver:* In order to provide control inputs for the monocopter, a Transmitter-Receiver system for use in model aircraft and drones was used. The Transmitter (FrSKY Taranis) transmits the 8 control channels in 2048-bit resolution over a 2.4GHz pseudo-random frequency-hopping spread-spectrum protocol(ACCST). This is then processed by the receiver which outputs a Combined Pulse Position Modulated (CPPM) signal. CPPM is a protocol wherein 8 Pulse-Width-Modulated (PWM) signals are concatenated and followed with a frame buffer signal to indicate the termination of the data packet. The individual PWM signals vary from $1000\mu s$ to $2000\mu s$ and are defined as the time

between two subsequent rises. The frame buffer length is the length required after all the PWM signals have been concatenated such that the packet is $18ms$ in length. An example of such a signal can be seen in Fig. 12.

The decoding of the CPPM signal is done through an interrupt, which, when it detects a rise, records the time since the last rise as the value of the corresponding channel. When the channel length is longer than $2500\mu s$, the pulse is assumed to be the frame buffer and the system returns to recording channel 1. This means that the control information is at worst $18ms$ old which is significantly better than required when considering the human controller providing the input.

B. Outputs

Control of the motor and servo are achieved through PWM signals at 250Hz, with pulse duration varying between $1000\mu s$ and $2000\mu s$. This was achieved through the internal timers within the Arduino. The base clock on an Arduino Micro is $16MHz$ a pre-scaling of 64 ($N = 64$) was used. The formula describing the number up to which the timer must count is as follows:

$$f_{output} = \frac{f_{clock}}{2 \cdot N(1 + \text{Timer}_{\text{Top Value}})} \quad (11)$$

From equation 11, it can be seen that to reach a frequency of 250Hz, the timer must count up to 999. To implement the required PWM signal, two separate implementations were tested. The first, depicted as "Timer 1" in Fig. 13, sets the output high and proceeds to count to a value between 249 and 499 corresponding with the required PWM signal between $100\mu s$ and $2000\mu s$. The output pin is then set low and the counter reset and the counter counts until the sum of the high and itself is 999. Hereafter, the process repeats itself. This is fairly computationally intensive and results in extended ISRs which can interfere with other ISRs resulting in missed and delayed interrupts.

Timer 2 in Fig. 13 counts to a unchanging value of 999 and whenever Timer 2's value crosses the set value OCR1A, the logic level of the output pin is inverted. This system is significantly simpler and relies, to a much larger degree, on the internal functionality of the ATMEGA 32U4 processor. The downside of this implementation is the restriction of the pins available for usage as output. However, as the latter system was ultimately implemented, the system was simply re-soldered to accommodate.

C. Translational flight and control mixing

In order for the craft to fly translationally, the control inputs from the receiver have to be mixed into a function of the angle of rotation of the craft. As the craft is rotating, the outputs need to vary sinusoidally with the angle of the monocopter and be recombined into a single measure which can be outputted to the servo. This output has to be constrained to the pulse duration limits of the servo. The resultant equation can be seen in Fig. 14.

Where t_{servo} , t_{roll} , t_{pitch} and, $t_{collective}$ represent the respective servo output and PWM inputs in milliseconds. The

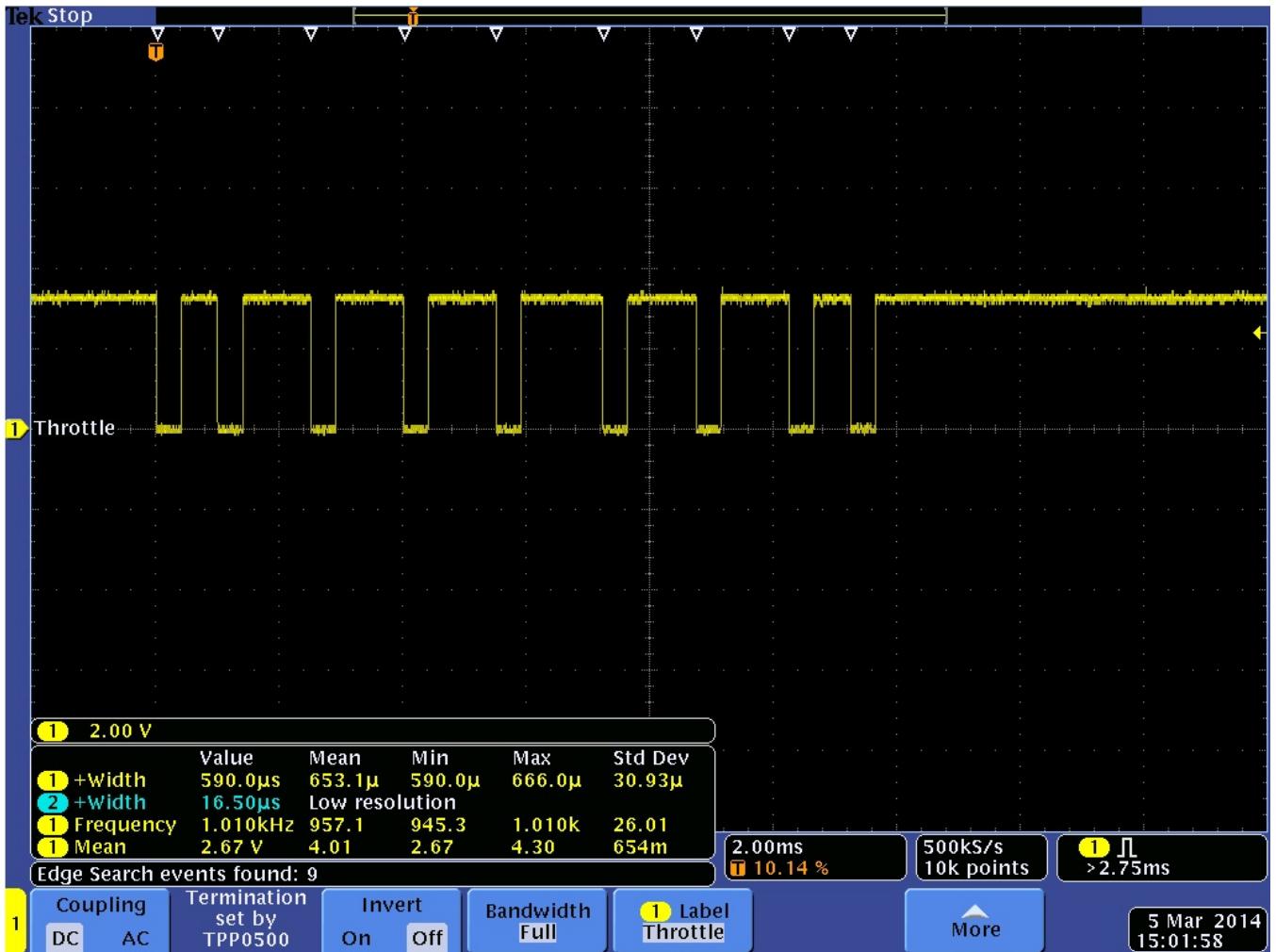


Fig. 12. CPPM Signal Measured on an Oscilloscope

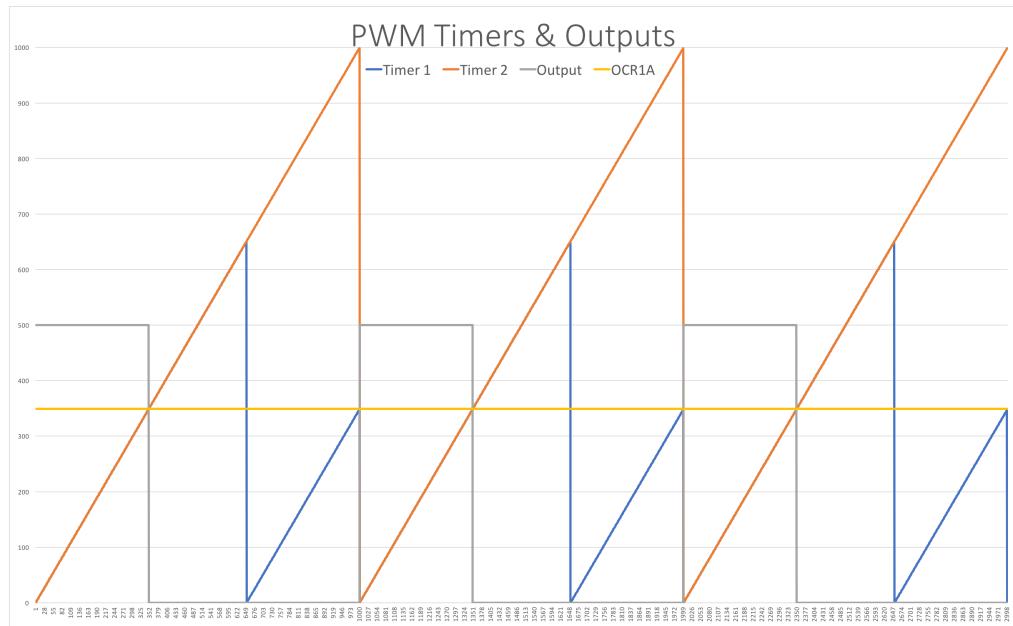


Fig. 13. Plot of how PWM signals are outputted to the motor and servo. (1500 μ s at 250 Hz)

$$t_{servo} = \frac{-\cos(\theta).(t_{roll} - 1500) + \sin(\theta).(t_{pitch} - 1500) + \frac{1}{2}.(t_{collective} - 1500) + 1257}{2.9} + 1500$$

Fig. 14. Length of servo PWM pulse as a function of the control inputs.

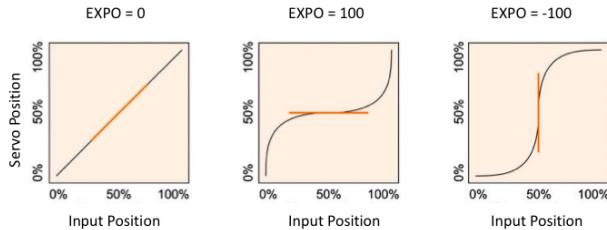


Fig. 15. Effects of various Expos on the relationship between input position and control output.[5]

phase of the above equation, assuming the forward pitch direction to be the origin, is $\frac{2}{3}\pi$. This allows for the effects of gyroscopic precession as discussed above under "Theory". While the above equation linearly maps control inputs to flight movements, the transmitter is programmed to use an exponential mapping between the position of the input gimbals and the control data transmitted. This results in a finer degree of control around the neutral points of control input whilst still preserving full travel as depicted in Fig. 15.

V. EXPERIMENTAL PROCESS

A. Setup and Sub-Experiments

Before the monocopter could be flown, all of the sub-components needed to work accurately, and be correctly integrated together. This was tested and achieved through a series of small experiments. The controller will be hereafter referred to as the "system".

1) Testing the System Outputs: Two elements of the system can be adjusted during flight: the motor and the servomotor. Two sub experiments were executed to test these components. First the motor and servo were hooked up to a variable power source and tested for functionality. Since the servo and ESC motor power regulator each require a PWM (Pulse-Width Modulated) signal to run, a small sub-program involving only these components was developed and loaded onto the Arduino. The Arduino was then tested while still attached to a laptop which sent it signals.

2) Testing the System Inputs: Next, the input of the system, the magnetometer, was tested. The Arduino was again loaded with a simple sub-program to test only this component. The wing was held and slowly rotated around, and the output function was plotted on a computer screen. After some de-bugging, the magnetometer began outputting a realistic signal.

For translational flight, the craft had to be able to "remember" which direction it was facing at a given time. To achieve this, it needed to update its position very quickly and accurately. Another sub-experiment was done to test this function, in which an LED was attached to the craft and set to light whenever the wing was oriented at a small range of angles (variable on the radio controller). The craft was slowly spun and the code debugged. One issue encountered during debugging was a truncation problem adding the change in direction to the initial direction. There was also an issue getting the Arduino to recognize when the magnetometer was beginning transmission.

3) Integrated System and Control: After the inputs and outputs of the system all functioned correctly, all of the components had to be integrated into one system. A program to do this was loaded onto the Arduino, and a laptop as well as the magnetometer sent signals.

For translational flight, the servomotor would need to be readjusted in a sinusoidal pattern during one rotation of the craft. In a sub-experiment to test this, the servo responded to the magnetometer instead of an LED. It was programmed to move the aileron in a sinusoidal fashion in response to the Arduino. The craft was slowly rotated, and the code was tinkered with. The servo was noticed to be higher in one half of the craft's spin cycle, and down in the other half. However, the servo kept twitching erratically at certain orientations. This posed a problem because for translational flight to be successful, a smooth change in the wing lift was needed during one orientation. Since the signal from the magnetometer seemed reasonably clean, the group hypothesized that there must be some problem with the code of the Arduino, or a bad wire somewhere. The twitching issue was never resolved.

In the LED servo test, the phase of the craft (the reference angle) could be adjusted using the radio controller. For translational flight, controlling the phase corresponds to changing the direction of flight in x and y directions in a similar manner.

After the components had been integrated, the receiver and controller were instead used to send signals to the system. The whole system's communication was debugged and tested.

B. Main Experiment: Vertical Flight

After all the components were controllable by the radio, it was time to begin flight testing. Flight Testing was done in a drone testing facility in the Zilverling. Since the craft spun too quickly to be clear to the human eye, slow-motion cameras were used to observe the behaviour of the monocopter during each test flight.

At first, the craft was fitted with a hole presumed to be near the centre of gravity. It was mounted on a thin rod. From afar the motor was powered up. It was observed that the rod began to oscillate when the motor was turned on, likely because the mount was not directly at the centre of gravity. Before the craft could achieve significant lift to rise off the ground, it would catch the air, flip forward and propel itself off of its mount and onto the ground. The battery was repositioned many times to try and change the location of the centre of mass. However, it was of no avail. During one of these tests, the main wing spar snapped in two as the monocopter spun from its mount. The existing launch set-up was evaluated to be unsuccessful, and the frame was reset with epoxy.

The group brainstormed about a way to allow the craft to rotate around its exact centre of gravity, and to avoid manual launches by dropping the craft from significant heights. It was found that many early aircraft were initially outfitted with a pair of ski-like skegs which allowed the craft to slide over the ground. Skegs seemed like a plausible solution to allow the monocopter to freely rotate about and axial centre going through its centre of mass, and keep it nearer to the ground. Two thin wire pieces were mounted, one under the wing near the presumed centre of mass, and one under the motor to keep the propeller from hitting the ground.

More trials were performed adjusting the power and camber of the craft upon launch. Eventually the slow-motion cameras revealed that the wing was beginning to achieve lift. However, the wing would quickly tilt into the air, forcing the propeller into the ground while the wing became vertical. The system would then become unstable and flip over. The wing seemed to be getting too much lift. To solve the problem, the fourth outermost section of the wing was cut off of the craft.

Many more trials were performed adjusting the power and camber of the new, short-winged craft upon launch. The battery and skegs were repositioned to account for a shifted centre of mass. As the motor speed was increased, the craft's orientation began to flatten out instead of becoming unstable. After much trial and error, the monocopter finally lifted its weight off of the skegs and hovered a few centimetres above the ground. Upon powering down the motor, the craft was able to land back on its own skegs.

Trials were performed to try to move the craft up and down in space. Due to the small testing area, the craft tended to approach the walls before getting higher than 1 meter, and would need to be brought down again. Tests never took more than about half a minute, for several reasons. Firstly, the motor began to heat up after a while even though it was only being driven at half power. Secondly, the aluminium motor mount experienced so much torque during flight that it would actually begin to hinge the motor downward, causing the system to behave unstably. The skegs had a similar problem; after each flight test these thick metal wires were stretched away from the centre of mass, having been completely rebent from their initial position.

The movement of the skegs during flight made it pre-

carious to safely land the craft. Furthermore, the metal of the motor mount and skegs began to weaken over trials, causing the problem to worsen each time. Inevitably, the main spar snapped during a bad landing. Before this occurred, however, at least 5 successful flights were completed. From the footage of the high-speed camera, it was observed that the monocopter took about 24 frames to complete a full rotation, with the video shooting at 240 frames per second. This is equivalent to 10 rotations per second. Thus, approximately 120.7g of centripetal acceleration was applied on the outermost components of the monocopter (the motor, skegs and wing-tip). It is no surprise, then that the skegs were being bent outward due to the high forces experienced.

C. Discussion

Some of the problems encountered with the monocopter, such as programming errors and lift off issues, were worked out during experimentation. However, some further discussion arose from the experiments.

One important point to clarify is how the monocopter lifted off the ground. To begin, the aileron was pitched down and the power was slowly increased. As the power increased and the monocopter began to achieve lift, the pitch was brought to neutral. To raise the craft up, the pitch could then be increased again. During operation, the speed of rotation was determined by two counteracting factors; the motor acted to speed up the system, while the change in camber caused by the aileron increased drag and slowed the system down. To keep the monocopter stable, these two factors had to be carefully balanced during flight.

The main goal of the project was to achieve translational flight in the x, y and z directions. Z translation was realized. However, the system was not controllable enough to move predictably in the x and y directions, because of the twitching of the aileron when it was connected to the magnetometer. Perhaps this was because of electric fields generated by accompanying electronics, which could have interfered with the accuracy of the compass readings. Whilst the motor generates the largest magnetic field, it had substantial separation from the compass sensor and research suggests that brushless motors have no effect on compass readings at similar distances. [9] [1] The servo is a more likely candidate for interference as the magnetometer was mounted directly against it and it contains both a motor and MOSFETS which produce large amounts of EM radiation.

Two other obstacles might also make translational flight unrealistic for this system. For one, the magnetometer might not be precise enough to handle the rotational speeds achieved. An experiment which could have been done and was not was to see how accurately the magnetometer could determine the orientation of the craft when it was spinning at higher speeds, using the LED. Another less likely obstacle to translational flight might be that the servo is unable to adjust often enough to control motion. This could best be tested by running a similar experiment.

During development, the project was continuously evaluated and improved. Choices were made which differed from

the initial design decisions. Notably, the two accelerometers were not needed at all, and were removed from the system. Next to this, two skegs were added to the monocopter. These evaluative improvements could be made quickly, but most of the ideas could not be implemented right away.

Luckily, the group is planning to continue with this project. In further designs, the following changes could be made from the current system:

- The wooden frame spar broke twice during crash landings. Next time, it could be made out of carbon fibre instead.
- The skegs and motor mount were not strong enough, even though they were made of aluminium alloys. Next time, these parts could be made out of carbon fibre or stronger metals.
- Furthermore, the skegs should be easily adjustable to account for changes in the centre of mass.
- It might be interesting to make the whole wing twist instead of just the aileron. This would keep the camber of the wing constant, but would allow the pitch to have a larger effect on the craft for smaller changes.
- The wing should definitely be smaller or the craft heavier for future designs. One might experiment with different shapes and aerofoils, maybe even modelling the wing after the wing of a Samara seed. The wing could be made of light carbon fibre, however the blue foam worked remarkably well.
- Some of the electronic components need to be upgraded: the magnetometer, the servo-aileron push rod, and potentially the servo.
- Ideally, components of the device should be easy to swap out for one another without compromising structural integrity. This would make trying different solutions much easier.

VI. CONCLUSION

This project was a partial success on all fronts. A seemingly plausible and self-coherent model and simulation were set up. However, with no realistic input data for it, there is no way of knowing just how accurate it is. Also, the craft did indeed fly, but translational motion was not implemented due to hardware and time constraints. A video of the craft flying can be seen at: <https://youtu.be/JeJ6qmZzpu>. Further development of this project would most definitely call for a better magnetometer and use of sturdier materials.

REFERENCES

- [1] Measurement of magnetometer readings with brushless motor running. 2006.
- [2] Aeronautics Learning Laboratory for Science Technology and Research Network. Aeronautics - Principles of Flight (AIRFOILS) - Level 2, 2008.
- [3] drm0hr. Exceeding the maximum output data rate of the HMC5883L, 2013.
- [4] Kingsley C Fregene and Courtney L Bolden. Dynamics and Control of a Biomimetic Single-Wing Nano Air Vehicle. 2010.
- [5] hexenbesen. dual-rate und expo für fortgeschrittene, 2008.
- [6] JohnRB. FrSky D4R II CPPM Output, 2014.
- [7] Andreas Kellas. The Guided Samara: Design and Development of a Controllable Single-Bladed Autorotating Vehicle. 2007.
- [8] Luppie. Monocopter software/hardware — ModelbouwForum.nl, 2013.
- [9] Michael Shinniok. Bot Thoughts: Magnetometers and Motors, 2011.
- [10] R. KE NORBERG. AUTOROTATION, SELF-STABILITY, AND STRUCTURE OF SINGLE-WINGED FRUITS AND SEEDS (SAMARAS) WITH COMPARATIVE REMARKS ON ANIMAL FLIGHT. *Biological Reviews*, 48(4):561–596, 11 1973.
- [11] Borna Obradovic, Gregory Ho, Rick Barto, Kingsley Fregene, and David Sharp. A Multi-Scale Simulation Methodology for the Samara Monocopter UAV.
- [12] UIUC Airfoli Coordinates Database. AG455ct -02f rot. (ag455ct02r-il).
- [13] University of Washington. Nature Inspired Design, 2011.

APPENDIX

A. Reflection on Project, Group Dynamic

This project was made possible due to the combined efforts of each of the four project members. The group size was ideal in the fact that every member had enough to do without being overwhelmed. One member focussed on simulation, one focussed on programming and analysis, one worked on hardware and design, and another on visualization, modelling and video editing. The group looks forward to continuing the project in future modules.

If the group is to continue the project, the biggest constraint will likely be funding for better components and carbon fibre. A solution to this problem has yet to be found.

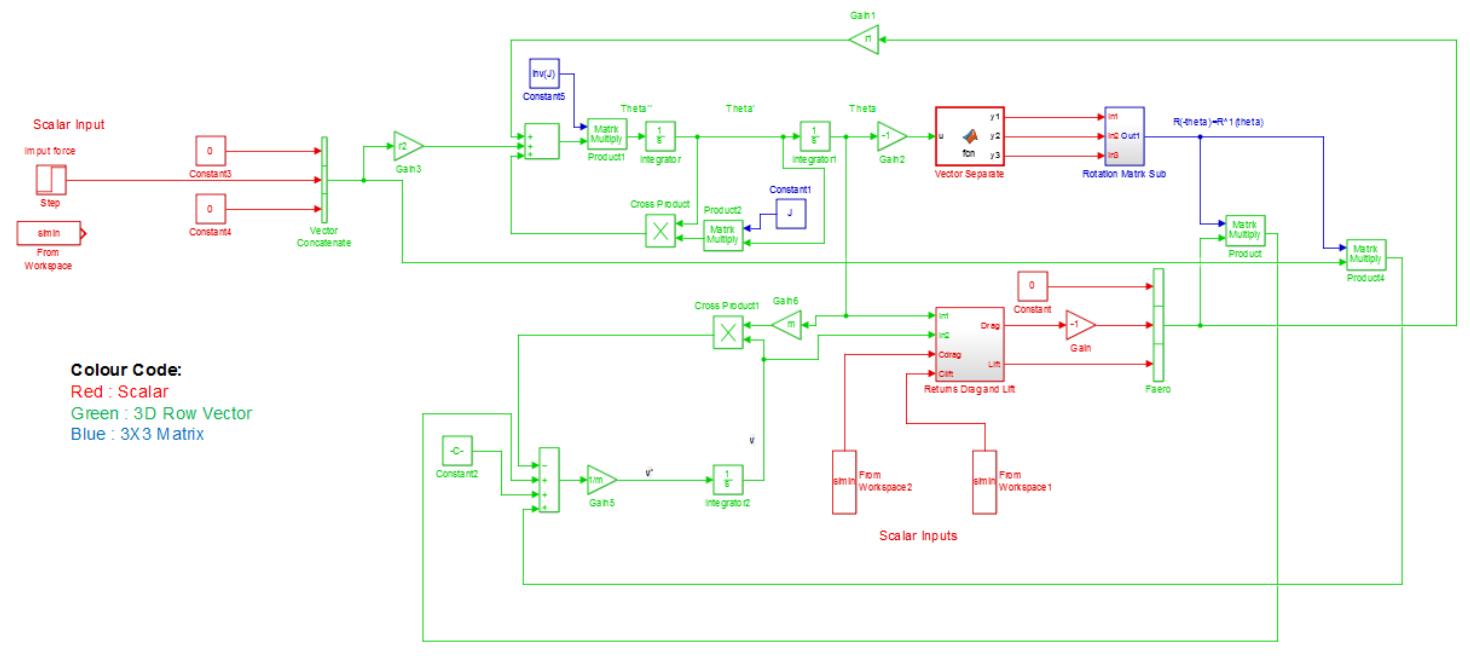


Fig. 16. The main block diagram for the simulation. Refer to colour code.

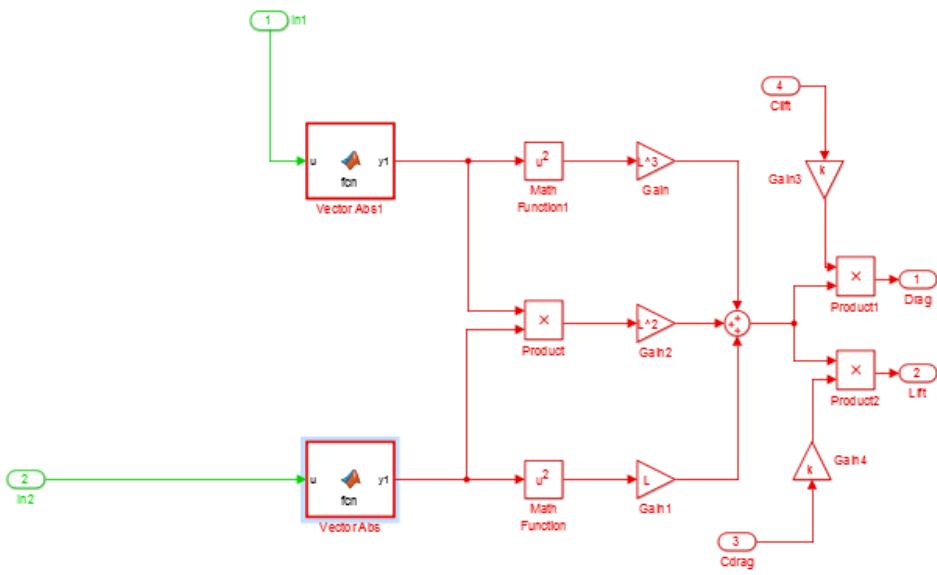


Fig. 17. The lift/drag subsystem of the simulation. Refer to colour code in previous figure.

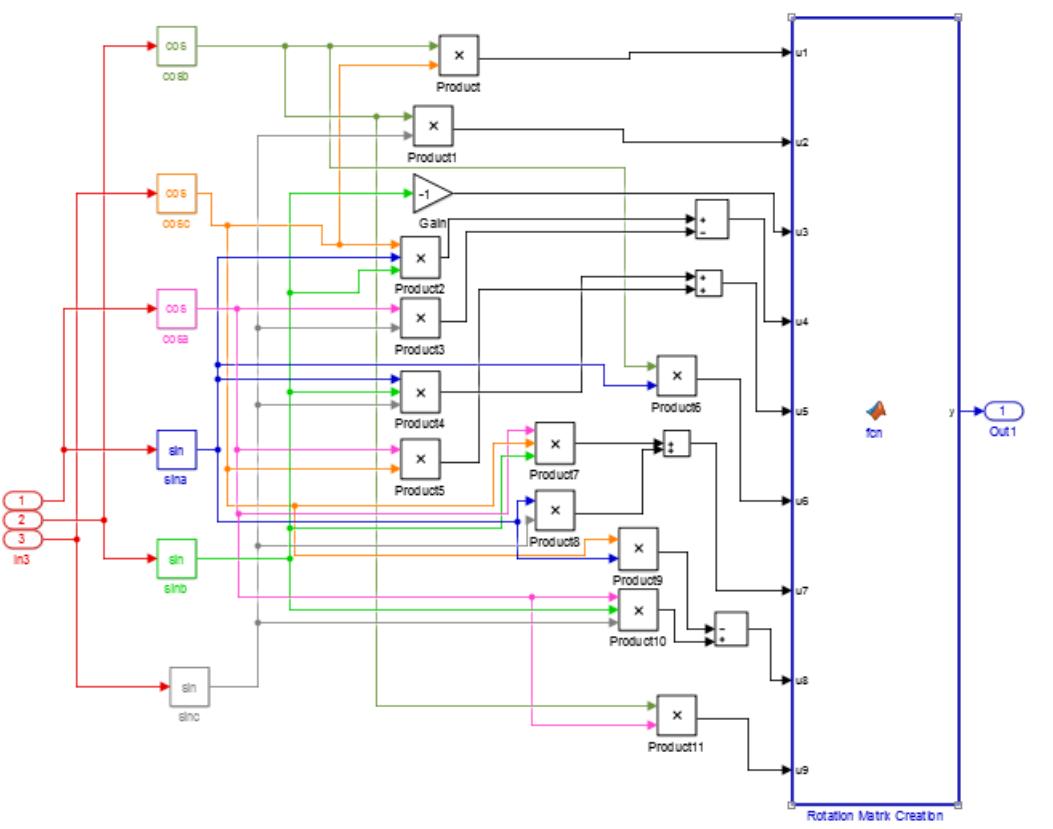


Fig. 18. The rotation matrix subsystem of the simulation. Signal colours are not indicative of dimensionality.

```
function y = fcn(u)
 %#codegen
y=sqrt((u(1)^2)+(u(2)^2)+(u(3)^2))
```

Fig. 19. Matlab code for the simulink block that takes a vector as input and outputs its norm

```
function [y1,y2,y3] = fcn(u)
 %#codegen
y1 = u(1);
y2 = u(2);
y3 = u(3);
```

Fig. 20. Matlab code for simulink block that takes a vector and outputs its coordinates

```
function y = fcn(u1,u2,u3,u4,u5,u6,u7,u8,u9)
 %#codegen
y = [u1 u2 u3; u4 u5 u6; u7 u8 u9];
```

Fig. 21. Matlab Code for simulink block that takes 9 scalars and puts them in a 3X3 matrix