Design Analysis, Modelling and Experimental Validation of a Bird-like Flapping-Wing Flying Robot

Aneesh N. Chand*, Michihiro Kawanishi, and Tatsuo Narikiyo
Control Systems Lab, Toyota Technological Institute, Japan

ABSTRACT

The design of a relatively new genre of aerial robots — a full-scale, bird-like flapping wing flying robot — is analyzed, modelled and validated through real experiments in this paper. Quaternions are used instead of Euler angles in the dynamical model to represent the flying robot orientation in 3D space in order to realize smoother rotational manoeuvres using spherical linear interpolation. Effect of the control surfaces on the rotational behaviour of the flying robot is used to deduce the moments induced. The moments are then used to determine the angular accelerations, rates and orientation of the robot body in 3D space. Aerodynamic forces acting are then used to model the translational motion of the robot. For this, we propose practical methods of estimating the lift and propulsion generated by the flapping wings of the flying robot. The robot model is first simulated in a virtual environment to realize basic yawing and pitching manoeuvres and real experiments are conducted subsequently. Simulated motion corroborates with the real sensor data and gives an insight into the type of future controllers that ought to be designed.

1 INTRODUCTION

The last decade has seen concerted research and effort in the field of micro aerial vehicles (MAVs) [1]. In Japan, the need for robust and autonomous MAVs came to light in the aftermath of the Fukushima nuclear disaster. In North America, MAVs are now being touted for use in commercial delivery. The penetration of MAVs in civil applications in Europe in the coming years is expected to be so tremendous that the even the European Commission (EC) has taken a proactive stance and formulated a strategy to deal with it [2].

In the beginning, much of the related research used mainly small-sized fixed and rotary winged aerial robots [3], [4] i.e airplanes and quadropters. Focus then shifted to the realization of flapping-wing MAVs with fundamental design-oriented research done in [5], [6], [7]. Current work on flapping wing MAVs has largely focused on small and micro-sized aerial robots intended for indoor environments e.g the Delfly [8]. Thus, there is much scope to investigate in the realization of large bird-like, flapping wing MAVs capable of strong performance in outdoor flight.

Figure 1: The SlowHawk2 flapping wing flying robot with articulated wings.

The DelFly series of ornithopters [9], [10] are small-sized flapping-wing robots that could autonomously fly using only onboard power. Some models developed are perhaps the only flapping-wing flying robots to have been endowed with autonomous navigation using on-board vision control [11]. The Harvard Microrobotic Fly [12] was a biologically inspired micro aerial vehicle with flapping wings. Driven by a piezoelectric actuator, it could take off vertically while being operated through a tethered wire. Its successor, the RoboBee [13], could perform basic rolling and pitching movements while tethered. Both robots were experimental prototypes and the RoboBee is still under development. The micromechanical flying insect of [14] was a similar project as the Harvard Microrobotic Fly. The design and mathematical model of the robot was based on insect flight aerodynamics while also using non-articulated wings.

This work is discernible from related works in a number of ways. Compared to [9]-[14], this work uses a full-scale, bird-like flapping wing flying robot capable of strong outdoor flight whose flying mechanics and control surfaces are different. While other works have claimed some relevance to flapping wing flying robots, there are major differences in approach: [15] was concerned with reinforcement learning of a fluid system that was a model of flapping flight whereas for the flapping wing robot of [16], motion control equations are not given. Rather, the focus was on investigation of optical flow on flapping wing flying robots. The Smartbird of[17] quite resembles our robot but only design-oriented research was undertaken. Autonomy was not achieved. By full autonomy, we imply a flying robot that can control its movements by itself and without any human control (a manual, remote-controlled operation is not considered autonomous as is often erroneously reported by researchers). Thus, instead of design and development, we are more concerned with au-

*Email: aneesh@toyota-ti.ac.jp

Specification	Value
Power source	11.1V 900mAh Lithium Polymer
Gross weight	430g
Length	78 cm
Wingspan	60cm (1 wing)
Motor	Pulso X2208/22 brushless dc motor
Servos	Hitec HS-56HB Karbonite Micro Servo (Elevator)
	Hitec HS-45 High Speed/Torque Servo (Rudder)
Radio	DMSS 2.4GHz TX module
Payload	160g
ESC	GWS ESC 15A
Sensor	APM 2.6 (3-axis gyro/acc., magnometer, barometer), PIX4FMU

Table 1: Specifications of SlowHawk2 with articulated wings.

tonomous control. Therefore, this work is devoted to modelling and obtaining equations that govern the robot motion such that fully autonomous control can be implemented in the future. Our approach is therefore to use an existing and commercial flapping-wing platform and develop motion control algorithms for it. This subsequently entails a consideration of topics ranging from dynamical modelling and simulation to kinematics formulation and control.

This paper presents the first of our results. It presents the analysis and dynamical modelling of a bird-like flapping wing flying robot. The significant contribution of this paper is (i) the formulation of the dynamical model for a full-scale, outdoor-use flapping wing MAV with articulated wings and (ii) verification of dynamical model using both theory and *real experiments and real sensor data*. We first simulate the flying robot model in MATLAB to realize basic yawing and pitching motions which form the basis for future flight trajectories in 3D spatial space, and then repeat experiments with the real robot. Our intent is not to reverse engineer the flapping wing flying robot but to analyse it and derive a complete generic and scalable kinematics model for the overall genre of articulated-wing ornithopters. Compared to other works, our work focus on using a life-sized flapping wing flying robot with articulated wings intended for outdoor environments.

2 The SlowHawk2 Flapping Wing Flying Robot

2.1 Design Analysis

The flapping-wing MAV under study is the SlowHawk2 flying robot[1] with articulated flapping-wings, shown in Figure 1. The original SlowHawk is available commercially and has been custom modified so that it has articulated wings, thereby being more life-like with increased payload. The full specifications are provided in Table 1.

A single brushless dc motor is the principal actuator responsible for the flapping mechanism. The motor is first connected to a gear train system of three gears $\{G_1, G_2, G_3\}$ for speed reduction. The last train gear G_3 drives a crank arm mechanism. The crank arm is coupled to a connecting rod which converts the motor circular motion into uniform flapping action (up and down flaps) of the two wings. This slider-crank configuration is labelled in the diagram of Figure 2. The brushless motor is used for actuating both wings simultaneously. With such concurrent flapping, one wing can not flap independently of the other. This flapping action contributes to the lift and propulsion of the robot.

[1] SlowHawk2 with Articulated wings remodelled by Kazuhiko Kakuta

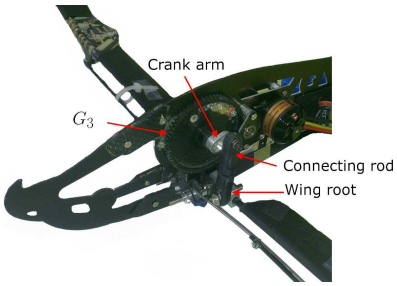


Figure 2: Slider-crank mechanism used to realize wing-flapping motion

The robot frame is made of carbon fiber frames and rods. The wing surface is made of thin, water-resistant vinyl tarpaulin. Two servo motors are used for actuation of the tail as a rudder and elevator. The rudder and elevator deflections control the steering (yaw) and pitching movements of the flying robot. Use of servos implies that the rudder and elevator deflections can be controlled by commanding them to move precise angles using internal feedback when controlled in position mode. Likewise, the instantaneous rudder or elevator position can be determined by reading the current servo position. This feature will be exploited later when controlling the robot autonomously.

A notable feature of the robot is the absence of any control mechanisms for rolling the flying robot. This implies that the robot makes a turn by controlling the tail rudder exclusively, thus without any active banking or rolling. This characteristic will later influence the number of derived equations for the dynamical model.

The payload of weight approximately $158g$ allows us to put various on-board electronics and sensors. Individual modules each weigh less than $50g$ so one or two units can be installed at a time for different measurements. Options for on-board sensors are included in the table of specifications.

2.2 System Modeling

Consider the brushless dc motor responsible for the flapping. Input to this motor is angular velocity in radians per second. Let us denote this as ω_b rad/s. Let η be the total gear ratio of the entire gear train $\{G_1, G_2, G_3\}$ system (found by taking the combined ratios of all individual gears, which is not shown here) The reduced gear speed, of the last gear G_3 in the gear train, is therefore

$$\omega_3 = \eta\ \omega_b \tag{1}$$

According to the well-known physics of a crank mechanism, the connecting rod coupled to the last gear moves up and down uniformly with a velocity given by

$$v = -r\,\omega_3 \left(\sin\theta + \frac{r\sin 2\theta}{2l\cos\beta}\right) = v_{flap} \tag{2}$$

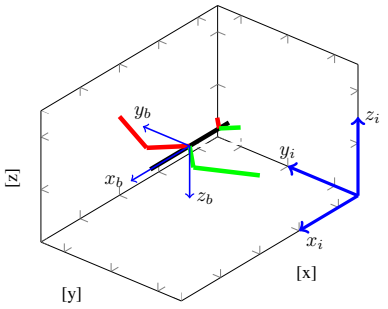


Figure 3: Assignment of reference frames

where θ is the crank angle from the vertical and

$$\cos\beta = \sqrt{(1-(r/l)^2\sin^2\theta)}. \tag{3}$$

where r is the crank length and l is the connecting rod length. Since the connecting rod is coupled to the wing root and actuates it directly, this is the velocity with which the wings flap, i.e the speed of one flap v_{flap}, in terms of the input motor velocity. The frequency of flapping f_{flap} is therefore

$$f_{flap} = \omega_3/2\pi \tag{4}$$

3 Dynamical Model

3.1 Assignment of Axes and coordinate Frames

Two axes frames are assigned to the system as shown in Figure 3. The **body** frame $F_b = \{x_b, y_b, z_b\}$ is assigned relative to the instantaneous position of the flapping wing flying robot by fixing the frame to its body. The frame origin is at the robot c.o.m with the z axis pointing vertically down. The **inertial** reference frame $F_i = \{x_i, y_i, z_i\}$ acts like a global coordinate system and is set with $z = 0$ on the ground plane and pointing upwards while the directions of the x and y axii coincide with that of the body frame.

3.2 Position and Orientation Representation

The pose of the flapping-wing flying robot is specified by a six d.o.f specification. It is given by $(x, y, z, \phi, \theta, \psi)$ where the first triplet (x, y, z) specifies the robot position in the global coordinate system and the second triplet (ϕ, θ, ψ) represents the orientation which is normally done using the x-convention for Euler angles (corresponding to roll, pitch and yaw). If A, B, C are the matrices that specify rotations corresponding to Euler angles (ϕ, θ, ψ), then the orientation R is expressed as

$$R(\phi, \theta, \psi) = CBA \tag{5}$$

To circumvent the problem of the dreaded gimbal lock, the orientation is specified using quaternions instead of Euler angles in this work. Thus, the robot orientation is described as a rotation about an arbitrarily fixed axis using quaternions of the form

$$q = w + xi + yj + zk \tag{6}$$

with $i^2 = j^2 = k^2 = -1$. Here w describes the size of the rotation in $\mathbb{R}^3$ about the arbitrarily fixed and normalized axis defined by the vector x, y, z. Using this system, if the robot is oriented at an angle of θ about the axis defined by x, y, z, then the resulting quaternion is

$$q = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\,xi + \sin\frac{\theta}{2}\,yj + \sin\frac{\theta}{2}\,zk \tag{7}$$

Once a quaternion describing a rotation or orientation in $\mathbb{R}^3$ is obtained using equation (7), the resulting gimbal-lock free, quaternion-derived rotation matrix representing the robot orientation is given by

$$qR = \begin{pmatrix} 1-2y^2-2z^2 & 2xy-2zw & 2xz+2yw \\ 2xy+2zw & 1-2x^2-2z^2 & 2yz-2xw \\ 2xz-2yw & 2yz+2xw & 1-2x^2-2y^2 \end{pmatrix} \tag{8}$$

Then

$$R = qR \tag{9}$$

Using quaternions to represent orientation makes it possible to realize smoother rotations of the flapping wing flying robot. Although regular Euler-derived rotation matrices give the same end result for a rotation, the movement occurs in 'jerks'. This occurs because the final rotation matrix is a sequence of three successive rigid rotations: $R = \phi^\circ \rightarrow \theta^\circ \rightarrow \psi^\circ$. Quaternion-derived rotation matrices, on the hand hand, allow determination and execution of smoother rotations achieved using spherical linear interpolation. Euler angles are therefore only used at the output stage.

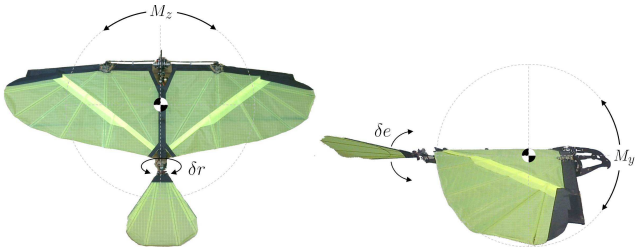
Spherical linear interpolation (*slerp*) [18] finds the quaternion matrix representing the smoothest rotation from orientation q_1 to q_o using the surface of the unit hypersphere for the interpolation. This technique can be used to find the smoothest rotational manoeuvre to change to a certain orientation. For example, external disturbances during flight cause the robot to be disoriented to an orientation q_1. We wish to again smoothly manoeuvre it back into its original orientation q_0. Then we could compute the smoothest manoeuvres using *slerp* according to eq. (10) for $0 \leq t \leq 1$ seconds:

$$\begin{aligned} \cos\Omega &= q_o \bullet q_1 \\ slerp\,(q_0, q_1, t) &= \frac{q_0\sin((1-t)\Omega) + q_1\sin(t\Omega)}{\sin\Omega} \end{aligned} \tag{10}$$

The resulting quaternion is the smoothest, constant-velocity manoeuvre from q_1 to q_0 that occurs in 1 second. To control the manoeuvre speed, the quaternion is simply time-scaled by multiplying with the desired time duration in seconds.

3.3 Dynamical Model

Treating the robot body as rigid, Newtonian mechanics provides the basis of developing motion equations that describe the motion and orientation of the c.o.g of the flapping wing flying robot as it flies. Net forces and net moments acting about the three axes are used to deduce the linear and



(a) (b)

Figure 4: Moments induced by rudder (a) and elevator deflection (b).

angular accelerations which then enable determination of the linear and angular velocities and displacements, in their corresponding frames respectively.

We first consider the rotational component as that yields the robot orientation which is required for the linear modelling. Thus, the moments acting on the flapping wing flying robot are deduced. Short-term moments are induced on the robot due to the movement of each of the control surfaces like tail elevator and rudder. We suppose that when the tail elevator or the rudder moves, the robot pitches or yaws about its center of gravity, respectively as shown in Figure 4. Ignoring damping, short-term moments contributed by the movements of the tail elevator and rudder are

$$\begin{aligned} M_y &= 0.5\,\rho\,V^2\,S_{wing}\,\bar{c}\,(C_{y\delta e}\,\delta e) \\ M_z &= 0.5\,\rho\,V^2\,S_{wing}\,b\,(C_{z\delta r}\,\delta r) \end{aligned} \tag{11}$$

Intuitively, the equations of (11) quantifies the moments M_y and M_z induced in the y and z axes of the F_b frame by the elevator and rudder deflections $\delta e, \delta r$ respectively. The dimensionless coefficients quantify or represent the individual net contribution of the deflecting control surface to the corresponding moment. That is, $C_{y\delta e}$ is the moment contribution of the elevator deflection δe (in degrees) to the moment in the y axis and $C_{z\delta r}$ is the moment contribution of the rudder deflection δr (degrees) to that of in the z axis. $\bar{c}$ is the wing mean chord and b is the wing span. The equations indicate *what* control surface contributes to *which* moment. As one would expect, the elevator deflection δe causes the pitching moment M_y whereas the rudder deflection contributes to the yawing moment M_z. Recall that the flying robot is incapable of banking actively in flight due to the absence of any relevant control surface. This explains the exclusion of any active rolling moments M_x in the equations.

Gravity contributes zero moments. Further, the influence of the flapping wings to moments induced depends on the gyro sensor placement placement. If a gyro sensor is placed on a wing, its moving crank arm mechanism or any other location that moves as a result of the wings flapping, naturally it will register rotational motion. However, if the gyro sensor is placed over the c.o.g, no rotational motion is registered since they only effect of the flapping wings upon the c.o.g is propelling it forward and lifting it up. Thus, only moments M_y and M_z are responsible for overall angular accelerations (and orientation) of the robot body in the F_b axes. The rotational acceleration equations are therefore

$$\begin{aligned} I_{yy}\ddot{\theta} &= M_y \\ I_{zz}\ddot{\psi} &= M_z \end{aligned} \tag{12}$$

where I is the moment of inertia of the robot. The angular accelerations $\ddot{\phi}$, $\ddot{\theta}$ and $\ddot{\psi}$ in the body frame are then easily determined. The angular velocities are

$$\dot{\phi}(t) = 0 \qquad \dot{\theta}(t) = \int_0^t \ddot{\theta}(t)\,\mathrm{d}t \qquad \dot{\psi}(t) = \int_0^t \ddot{\psi}(t)\,\mathrm{d}t \tag{13}$$

Conditions must be imposed in the integration process above to resolve ambiguities due to the dual connotations of acceleration:

1. Zero acceleration implies zero velocity (and not constant velocity)
2. Decreasing acceleration implies magnitude of velocity decreases to zero (and not to a constant value)
3. Negative acceleration implies decreasing velocity in same direction (and not an acceleration in the opposite direction)

Finally the quaternion representing the orientation can be determined from the angular velocities. Given the angular velocity is related to the derivative of the quaternion as

$$\omega = \begin{bmatrix} 0 \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = 2\frac{\mathrm{d}q}{\mathrm{d}t} \otimes \tilde{q} \tag{14}$$

the quaternion representing the orientation can therefore be determined

$$q = \int_0^t \frac{1}{2}\omega\ \mathrm{d}t \otimes \tilde{q} \tag{15}$$

Now consider the free body diagram (f.b.d) of the flapping wing flying robot in steady-state flight in Figure 5. By "steady-state", flight conditions of forward direction at constant velocity, constant altitude and zero pitch and roll are implied. The major forces acting on the flapping wing flying robot are shown: lift L, weight mg, thrust T, drag D and disturbances $F_{disturb}$. According to [19], a whole aircraft can be treated as a single entity when modeling its forces. Therefore, the equal but separate thrusts and lifts generated on either side of the flapping wing flying robot by its flapping wings are treated as one individual lift and thrust vector acting along and on the c.o.m without any significant loss of precision. The combined wing lift is accordingly shown as $L = L_1 + L_2$ acting over the robot c.o.m. Likewise, the total thrust generated by the two wings $T = T_1 + T_2$ is similarly shown acting forward along the x-axis.

Various research groups in aerodynamics have studied the physics and dynamics of flapping-wing flight and computed the thrust and lift generated [20, 21, 22]. However, most of the computations involve wing-specific parameters with specific experimental conditions, assumptions and test-beds. Accordingly, they cannot be unconditionally applied in this work. For example, we refer to the work of [22]. In the analysis of lift generation of a hummingbird wing model, the direction of the lift vector was taken to always point vertically upwards irrespective of the robot orientation. This is attributed to the active torsioning of the wings when flapping whereby the direction of the lift vector varied with the torsioning. However, the wings of the SlowHawk2 do not undergo any torsioning. The lift vector is therefore always remaining perpendicular to the robot body. Furthermore, instead of deriving design models as done in [6] and [7], our objective is to formulate those equations that describe the *motion* of the robot c.o.g as it flies. Consequently, instead of design-oriented direct calculation of thrust and lift, the resultant lift and thrust of the flying robot is deduced. Without any loss of precision, the net thrust T is given by

$$T = \frac{P}{V} \tag{16}$$

by calculating the net power P of the robot using the specifications of the brushless motor and measuring the velocity V of the flying robot in steady flight beforehand. In [22], the same formulation was used to estimate the lift although we give a different approach for lift estimation. It is recognized that the Lift force produced due to the flapping actions will be function of the speed of the flapping wings. Recall that the velocity of a wing flap has been computed. Thus for lift estimation,

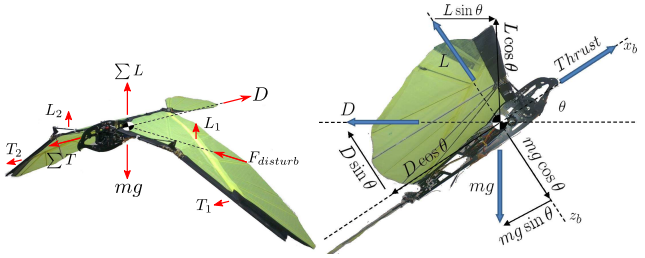
$$L = 0.5\,\rho\,{v_{flap}}^2\,S_{wing}\,C_l \tag{17}$$

Equation (17) is derived from fluid dynamics theory used for calculating forces, in this case lift, acting upon aerial vehicles. Here ρ is the air density, S_{wing} wing surface area and C_l the lift coefficient. v_{flap} is the wing flap velocity calculated in (2) and as long as v_{flap} can be calculated for other types of flapping-wing flying robots, the above expression for lift may be used. The flying robot is to be controlled at constant altitude and velocity so all parameters in this equation remain unvarying. Drag is given by

$$D = 0.5\,\rho\,V^2\,S\,C_d \tag{18}$$

where V is flying robot velocity, S body surface area and C_d the drag coefficient.

Given that a flapping wing flying robot is lighter than other types of MAV's, its motion is consequently more susceptible to disturbances such as wind gusts and Coriolis effect etc. Therefore, all probable disturbance forces are collectively modelled as $F_{disturb}$. Although MATLAB provides some built-in wind models that can be used to model disturbances, these are all designed to military-level specifications.



(a) (b)

Figure 5: Forces acting on flapping wing flying robot when in flight.

Thus, we define a disturbance force ourselves, in the inertial frame, as a vector-valued, quaternion-based force vector as $F_{disturb} = \begin{pmatrix} k \\ q \end{pmatrix}$ where k is the disturbance force magnitude and q the quaternion representing direction relative to the inertial frame. If the robot has some orientation q_0 relative to the inertial frame, the effective direction of $F_{disturb}$ acting on the robot is determined through straightforward quaternion multiplication.

$$F_{disturb}\{q\}|_{\text{effective}} = q_0 \times q_{disturb} \tag{19}$$

Thus, for a flapping wing robot flying oriented with some roll ϕ, pitch θ, and say for example, $F_{disturb}$ effectively acting along the y axis as illustrated in Figure 5, the equations for linear accelerations in each of the axes of the F_i frame are given by

$$\begin{aligned} m\ddot{x} &= \cos\theta\cos\psi\sum_{i=1}^{2} T_i - D\cos\theta\cos\psi - \sin\theta\sum_{i=1}^{2} L_i \\ m\ddot{y} &= F_{disturb} + \sin\psi\sum_{i=1}^{2} T_i - D\sin\psi \\ m\ddot{z} &= \sin\theta\sum_{i=1}^{2} T_i - D\sin\theta + \cos\theta\sum_{i=1}^{2} L_i - mg \end{aligned} \tag{20}$$

The kinematic equations are

$$\begin{aligned} \dot{x} &= V\cos\theta\cos\psi \\ \dot{y} &= V\cos\theta\sin\psi \\ \dot{z} &= V\sin\theta \end{aligned} \tag{21}$$

The motion and orientation of the flapping wing flying robot is therefore described by equations (20), (12), (15) and (21). It is important to state that although acceleration data obtained from on-board sensors will be expected to have white noise and bias, two solutions circumvent this dilemma. First is the use of a high pass filter to remove low frequency noise in the acceleration data which is responsible for integration drift errors. Alternatively, the acceleration data can be neglected altogether since it is the rotational velocity that is used in determining the relative orientation and this can directly be acquired from the gyro sensor. Further, rotational velocity information is primarily intended for executing momentary

manoeuvres in changing flight direction and path only. These momentary corrective manoeuvres occur in the order of milliseconds or an absolute maximum of 1-2 seconds. Therefore, accumulative errors of the gyro will be at a minimum. For position measurement, it is envisaged that absolute positioning systems such as GPS for outdoor use and motion capture systems for indoor use will be employed.

4 Experimental Results

Quaternion SLERP Test. A demonstration of using spherical linear interpolation to determine a smooth correctional manoeuvre is given in the accompanying video[2] (result is given as a video due to the *motion* involved). The robot model is initially shown in its original, steady-state configuration with orientation q_0 and the disturbed orientation, due to external disturbance forces, as q_1. The video contains the the smooth correctional manoeuvre from q_1 to q_0 as determined using quaternion-based spherical linear interpolation from eq. (10).

Model Validation. Equations for the dynamical model were validated by supplying separate input signals for the elevator and rudder deflections that would produce pitching and yawing motions, both in simulation and in real experiments. Motion data generated during simulation was then compared to real sensor data. As a proof-of-concept simulation, constant coefficients in the equations were defined as symbolic expressions since their values are yet unknown. A consequence of this approach is the axes of certain graphs are not quantified discretely, instead they exhibit the general motion profile which suffices to represent the robot behaviour emanating from the control input. For a robot flying in steady-state condition at arbitrary altitude and speed, the commanded input signal applied to the elevator and rudder deflections to realize pitching and yawing was the trapezoidal curve with amplitude A and zero length upper-base specified with the function

$$f(t;A,a,b,c) = \begin{cases} 0 & \text{if } t \leq a \\ A\dfrac{t-a}{b-a} & \text{if } a \leq t \leq b \\ A & \text{if } t = b \\ A\dfrac{c-t}{c-b} & \text{if } b \leq t \leq c \\ 0 & \text{if } t \geq c \end{cases}$$

by the definition of which the control surface deflection commences at $t = a$ steadily, reaches maximum deflection of A° at $t = b$ after which it retracts and returns to zero deflection position by $t = c$. The rate of deflection is therefore A°/s. A visualization of this is given in Figure 6.

Pitching Test. The signal specified for the elevator deflection in this test was $f(t; 10^\circ, 1, 2, 3)$ and for the rudder $f(t; 0^\circ, 1, 2, 3)$ i.e zero deflection.

The complete motion data is shown in Figure 7 for the robot body in the body frame of reference F_b. The angular acceleration data was obtained by applying eq. (11) and eq. (12) using the commanded elevator deflection function $f(x; A, 1, 2, 3)$ as input and adding simulated white Gaussian noise. It is seen that computed motion data subsequently derived from acceleration data correctly characterize the robot pitching motion at every instant of the robot movement apart from minor anomalies that are consequences of the simulated white Gaussian noise. For the first 1 second, the robot is in steady state flight. At the $t = 1$, robot starts pitching while continuing to fly forward. At $t = 2$, the robot halts pitching downwards and starts reverting back to zero pitch position as a consequence of the elevator deflection being gradually reduced. From $t = 3$ onwards, the robot has resumed steady-state condition albeit having lost altitude due to the pitching motion.

Inspecting the visualizations of the simulated rotational motion data in Figure 7, as the robot pitches forwards(downwards) around the y axis, the angular acceleration increases linearly (with minus sign). The corresponding angular velocity increases parabolically while displacement increases exponentially. Velocities around the x and z axii remain unchanged as there is no active rolling or yawing. The velocity curves appears to be free of white noise jitter but this is a superficial effect of integrating acceleration data using the trapezoidal method that has a smoothing effect on the resulting curves.

For the simulated linear motion:

1. z acceleration of the body frame increases linearly in the time interval the robot is pitching forward due to direction of the thrust vector gradually turning towards direction of gravitational pull that exists in the z axis. Same is also true for the y velocity.
2. The linear x acceleration decreases from zero acceleration while the robot was in steady state to a negative value in the time the robot pitches since the acceleration in that direction is reduced momentarily while the robot pitches and rescinds pitching. Consequently, there is also a drop in the x velocity in this interval.
3. y acceleration is unvaried.

The resultant linear velocities and displacements then depict the linear motion of the robot. The change in altitude as a consequence of the pitching motion is clearly seen in the z axis displacement curve in the displacement graph. Finally, the same experiment was repeated by flying the real robot non-autonomously to log sensor data. Comparing the gyro sensor data acquired in Figure 9(a), the Gyro Y data exhibits the same parabolic form as the simulated velocity data in Figure 7 when the robot underwent pitching.

Yawing Test. The specified rudder input command was $f(x; A, 4, 5, 6)$. The simulated results are shown in Figure 8 and the real sensor data is given in Figure 9(b). The consequence of this rudder deflection is the robot yawing to the left and then realigning to be parallel to the original flight direction (see "Y" curve in linear displacement graph). At the mo-

[2] http://aneeshchand.webatu.com/imav2014.html

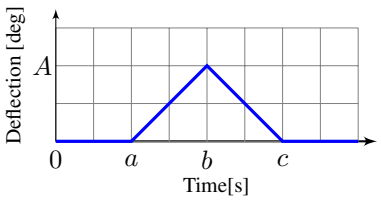


Figure 6: Input signal format for control surface deflection

ment the robot yaws, the real gyro sensor data again exactly replicates the simulated data.

Behaviour of the robot from both theory and experiments indicated a non-linear relationship between control surfaces' deflection and corresponding motion. For instance, for small values of δr or δe, the robot yaw or pitch magnitude is small but as the deflection is increased, the motion changes non-linearly. This was evident from the results: as the angular acceleration increased linearly, the angular rate increased parabolically while the angular displacement was exponential. This indicates that any future controllers designed would strictly be non-linear. Videos in the provided link show one of the simulations and a short demonstration of the real robot.

5 Conclusion

A simplified model describing the motion of a flapping wing flying robot with articulated wings was formalised with flapping averaged dynamics. Quaternions were used to represent the robot orientation and acquire smooth rotations in 3D space. Force-moment equations were then used to derive the dynamical equations that governs the robot motion. Simulations revealed the robot to respond to external stimuli in the form of elevator and pitch input deflections with simulated data qualitatively corroborating with real sensor data. Some constraints were assumed: only the salient active forces and moments were considered in the dynamics. Induced or residual forces and moments from body vibrations and flapping were ignored in the first prototype. Since only short-range navigation in the order of tens or hundreds of meters is considered, Earth's rotation and Coriolis acceleration were neglected. Likewise, only low altitude flying is considered so air-density was assumed to be non-varying. Given that the dynamical and kinematic model for the flapping-wing flying robot has been realized, the immediate next step is to design methods of estimating values of the parameters and coefficients. For reference position measurement, motion capture systems and/or GPS will be utilized.

Finally, given that some of the dynamics will be highly time varying, our future intention is to use adaptive control techniques for autonomous control where the on-line parameter estimation component of the adaptive controller would counter any deficiencies in the model.

Acknowledgements

The authors thank Dr. Kazuhiko Kakuta of Miyagi, Japan, developer of SlowHawk2 for kindly supplying the robot.

References

[1] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang and Daisuke Nakazawa, "Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles", Springer, 2010.

[2] European Commission, "Towards a European strategy for the development of civil applications of Remotely Piloted Aircraft Systems (RPAS)", Brussels, September 2012.

[3] Lorenz Meier, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer and Marc Pollefeys, "PIXHAWK: A micro aerial vehicle design for autonomous ight using onboard computer vision", Autonomous Robots (2012) 33:2139.

[4] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control", In IFAC World Congress, pages 1485-1491, 2011.

[5] W. Bejgerowski, A. Ananthanarayanan, D. Mueller, and S.K. Gupta, "Integrated product and process design for a flapping wing drive-mechanism. ASME Journal of Mechanical Design, 131: 061006, 2009.

[6] Madangopal, R., Khan, Z. A., Agrawal, S. K. "Biologically inspired design of small flapping wing air vehicles using four-bar mechanisms and quasi-steady aerodynamics", Journal of Mechanical Design, Vol. 127, No.4, 2005, 809-816.

[7] Grauer, J., Hubbard, J.E., "A Multibody Model of an Ornithopter, AIAA Paper No. 2009-727.

[8] de Croon, G.C.H.E., M.A. Groen, De Wagter, C., Remes, B.D.W., Ruijsink, R., and van Oudheusden, B.W. "Design, Aerodynamics, and Autonomy of the DelFly" In Bioinspiration and Biomimetics, 2011

[9] Percin, M., Hu, Y., van Oudheusden, B.W., Remes, B., and Scarano, F. "Wing flexibility effects in clap-and-fling" International Journal of Micro Air Vehicles, Volume 3, Number 4, pp. 217-227, 2011.

[10] Tay, W., Hester, B., van Oudheusden, B. "Analysis of biplane flapping flight with tail", at the 30th AIAA Applied Aerodynamics Conference, AIAA-2012-2968, 2012

[11] de Croon, G.C.H.E., de Clerq, K.M.E., Ruijsink, R., Remes, B., and de Wagter, C. "Design, aerodynamics, and vision-based control of the DelFly" In the International Journal of Micro Air Vehicles, Volume 1, Number 2, pp. 71 - 97

[12] Robert J. Wood, "The First Takeoff of a Biologically Inspired At-Scale Robotic Insect", IEEE Transactions on Robotics, Vol .24, No. 2, Apil, 2008. pp. 341-347.

[13] Benjamin M. Finio and Robert J. Wood, "Open-loop roll, pitch and yaw torques for a robotic bee", IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 7-12, 2012, Portugal.

[14] Xinyan Deng, Luca Schenato, Wei Chung Wu and Shankar Sastry, "Flapping Flight for Biomimetic Robotic Insects Part I: System Modeling", IEEE Transactions on Robotics, Vol .22, No. 4, Apil, 2006. pp. 776 - 788.

[15] John W. Roberts, Jun Zhang, and Russ Tedrake, "Motor learning at intermediate reynolds number: Experiments with policy gradient on the flapping flight of a rigid wing", In From Motor to Interaction Learning in Robots. Springer, 2009

[16] Fernando Garcia Bermudez and Ronald Fearing, "Optical flow on a flapping wing robot", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5027 - 5032, St. Louis, 10-15 Oct. 2009

[17] W. Send, Markus Fischer, Kristof Jebens, Rainer Mugrauer, Agalya Nagarathinam and Felix Scharstein. "Artificial Hinged-Wing birdwith Active Torsion and Partially Linear Kinematics", 28th International Congress of the Aeronautical Sciences, Brisbane 2012.

[18] Edward Pervin and Jon A. Webb. "Quaternions in Computer Vision and Robotics". Carnegie-Mellon University, 1992.

[19] David Allerton, "Principles of Flight Simulation", Wiley, pp. 47, 2009

[20] W. Send, "Flapping-Wing Thrust in Compressible Flow" 25th International Congress of Aeronautical Sciences (ICAS), Hamburg (Germany), September 3-8, 2006.

[21] D. Mueller, H.A. Bruck, and S.K. Gupta. "Measurement of thrust and lift forces associated with drag of compliant flapping wing for micro air vehicles using a new test stand design". Experimental Mechanics, 50(6):725735, 2010.

[22] Hiroto Tanaka, Haruka Suzuki, Ikou Kitamura, Masateru Maeda and Hao Liu, "Lift Generation of Hummingbird Wing Models with Flexible Loosened Membranes", IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Nov. 3-8, 2013.

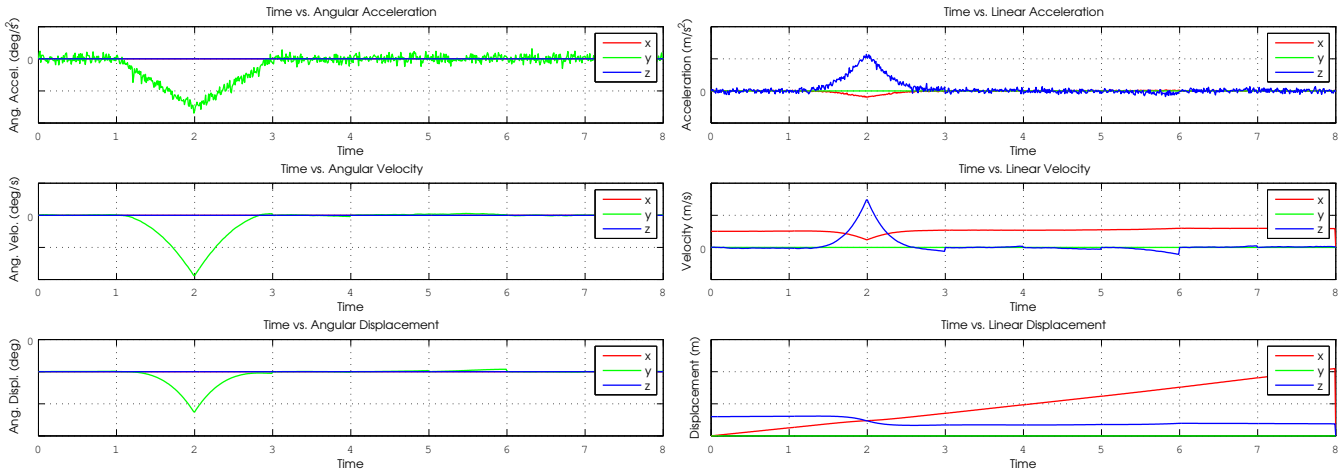


Figure 7: Simulated rotational and linear motion data for pitching movement. The "Z" curve of the linear displacement plot reveals the altitude change as a consequence of the downwards pitching motion

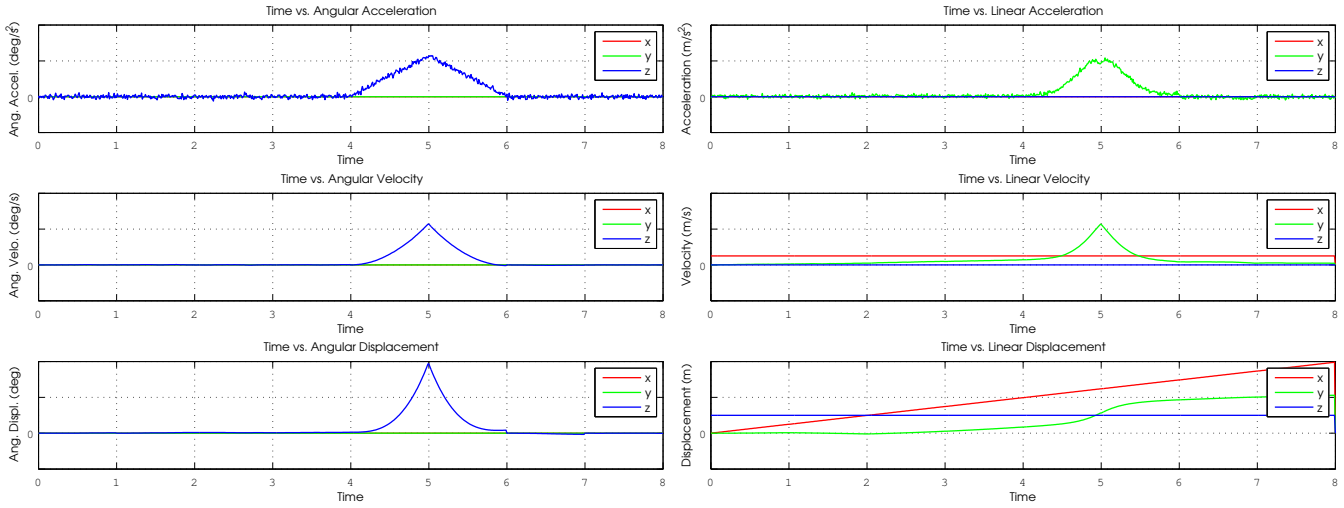
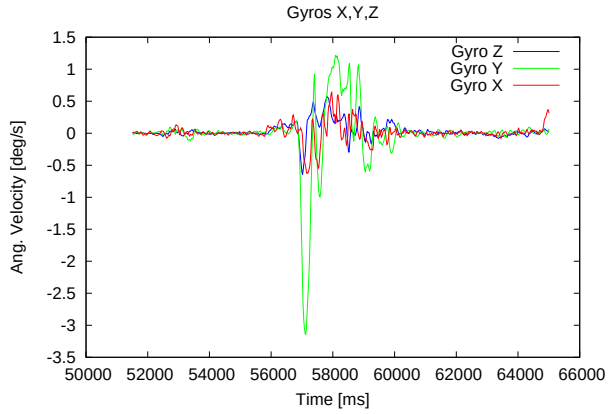
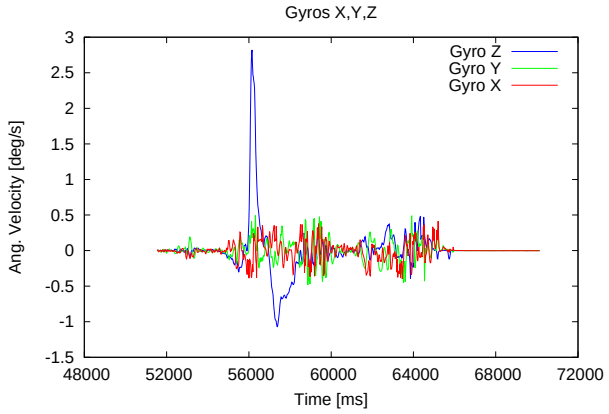


Figure 8: Simulated rotational and linear motion data for yawing movement. The change in flight direction is discernible from the "Y" curve of the linear displacement plot



(a) Real gyro sensor data for pitching motion. Aside from spurious data, it is seen that the Gyro Y data exhibits an exponential curve at approx. $t = 57000$ when the real robot pitched. This matches with simulated data of the angular velocity graph in Figure 7 at $t = 2$



(b) Real gyro sensor data for yawing motion. Again ignoring spurious data that can be easily filtered, note the overall resemblance of the Gyro Z exponential curve at $t = 56000$ when the real robot yawed to simulated data in the angular velocity graph of Figure 8 at $t = 5$

Figure 9: Real gyro sensor data for pitching (a) and yawing (b) motions