2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Critical subsystem failure mitigation in an indoor UAV testbed

Mark W. Mueller and Raffaello D'Andrea

Abstract— **An autonomous safety mechanism is presented, as implemented in an indoor flying vehicle research testbed. The safety mechanism relies on integration of onboard gyroscope measurements and thrust commands to estimate the vehicle state for short lengths of time. It is used in the case of loss of external control signal or loss of external measurement data, to reduce the likelihood of a vehicle crash, or at least reduce the severity of an unavoidable crash. As UAVs move into ever more mainstream applications with increased public interaction, such safety systems become more critical.**

I. Introduction

The Flying Machine Arena (FMA) is a testbed for aerial vehicle research, with a unique mix of fundamental research and public engagement. Research and demonstrations are done using a fleet of flying vehicles, and the correct functioning of the system relies on a high-precision motion capture system, and a low-latency command radio link. Failure of either of these subsystems renders the vehicles effectively blind, except for their inertial sensors – this paper describes a strategy to reduce the impact of such a failure.

The FMA is a $10 \times 10 \times 10$m space, covered by netting on the sides and padding on the floor. The vehicles used in research are quadrocopters, control of which is split into two layers (onboard the vehicle and offboard on a ground station), which are connected by wireless links. Testbeds similar to the FMA are GRASP at the University of Pennsylvania [1]; Stanford/Berkley Starmac [2] and MIT Raven [3].

The FMA has become well known for regular interactive demonstrations to the public, including dancing to music [4], balancing an inverted pendulum [5] and juggling balls [6]. During 2011, 44 events were held, during which 430 visitors attended demonstrations. Examples of demonstrations are shown in Fig. 1, where the mobile version of the FMA infrastructure is also shown. This mobile infrastructure allows the system to be deployed at temporary sites – the first external demonstration of the FMA was for the Flight Assembled Architecture project at an exhibition in Orléans, France. The fully autonomous construction of a 6 m structure consisting of 1500 foam bricks was done over four days, in front of (and directly above) a large crowd. This mix of research and public engagement places a high demand on the safety features of the system, where it is needed that the system be safe to operate at demonstrations with lay-people, without restricting the performance of the system, and still allowing the system to be flexible enough to allow for the rapid development of new projects.

Traditional safety mechanisms, e.g. safety pilot are not applicable here: the relatively small working space implies little time to react, while flying with a large number of vehicles would necessitate a large number of safety pilots. For example, in [7] the risk reduction strategy requires two safety operators: one to monitor a ground station, and an emergency pilot to take over manual control in the case of an emergency. In this paper, algorithms with low computational cost are developed allowing the vehicle greater autonomy from the ground station, without requiring additional bulky hardware (such as laser scanners, or cameras). It is important to note that the scope of this not as broad as e.g. completely autonomous flight using vision [8] or laser scanners [9].

The paper is organised as follows: Section II describes the system and its components, in III the failure modes are discussed, while the mechanism is discussed in Section IV, with results given in V. A conclusion is given in Section VI.

II. System overview

A. Vehicle dynamics

The vehicles of choice in the FMA are modified Ascending Technologies Hummingbird quadrocopters, as shown in Fig. 1 and 2, each with four alternately rotating propellers. By mixing the thrusts f_i from each propeller i, the vehicle can generate moments about all the body axes, and a total

Fig. 1. Close interaction with the public places high demands on the safety systems of the Flying Machine Arena. At the top, a primary school class attending a demonstration, and at the bottom picture a quadrocopter placing a brick as part of the Flight Assembled Architecture project.

The authors are with the Institute for Dynamic Systems and Control, ETH Zurich, Sonneggstrasse 3, 8092 Zurich, Switzerland. {mullerm, rdandrea}@ethz.ch

978-1-4673-1736-8/12/S31.00 ©2012 IEEE

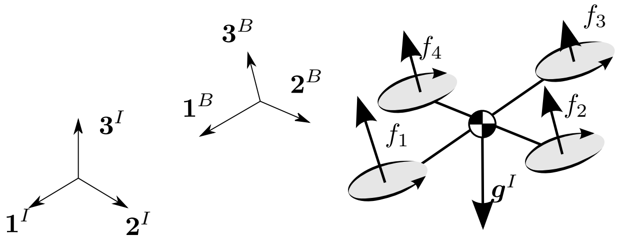

Fig. 2. Dynamics of a quadrocopter, with the triple $\left(\mathbf{1}^I, \mathbf{2}^I, \mathbf{3}^I\right)$ defining the ground-fixed inertial frame, and $\left(\mathbf{1}^B, \mathbf{2}^B, \mathbf{3}^B\right)$ the body-fixed frame, related by the rotation $\boldsymbol{R}^{IB}$ and translation $\boldsymbol{x}^I$ – see also (1).

thrust force f_T. The position, velocity and acceleration of the vehicle in the inertial frame are written as $\boldsymbol{x}^I$, $\dot{\boldsymbol{x}}^I$ and $\ddot{\boldsymbol{x}}^I$, respectively. The rotation of the vehicle body frame w.r.t. the inertial frame is $\boldsymbol{R}^{IB}$, and the body rates (in the body frame) are $\boldsymbol{\omega}^B = (p, q, r)$. The inertia of the vehicle, expressed about its axes, is $\boldsymbol{I}_B^B$ and the mass of the vehicle is m_B.

The rigid body dynamics of the vehicle are given by

$$\ddot{\boldsymbol{x}}^I = \boldsymbol{R}^{IB}\boldsymbol{c}^B + \boldsymbol{g}^I \tag{1}$$

$$\dot{\boldsymbol{R}}^{IB} = \boldsymbol{R}^{IB}\left[\boldsymbol{\omega}^B \times\right] \tag{2}$$

$$\boldsymbol{I}_B^B\dot{\boldsymbol{\omega}}^B = \boldsymbol{I}_B^B\boldsymbol{l}^B - \left[\boldsymbol{\omega}^B \times\right]\boldsymbol{I}_B^B\boldsymbol{\omega}^B, \tag{3}$$

where $\left[\boldsymbol{\omega}^B \times\right]$ is the skew-symmetric 3×3 matrix representation of the cross product of $\boldsymbol{\omega}^B = (p, q, r)$ [10]. The mass-normalised thrust force $\boldsymbol{c}^B$ and moment $\boldsymbol{l}^B$ produced by the propellers are expressed in the body frame as [1]

$$\boldsymbol{l}^B = \begin{bmatrix} l(f_2 - f_4) \\ l(f_3 - f_1) \\ \kappa(f_1 - f_2 + f_3 - f_4) \end{bmatrix} \tag{4}$$

$$\boldsymbol{c}^B = (0, 0, c) \tag{5}$$

$$c = (f_1 + f_2 + f_3 + f_4)/m_B \tag{6}$$

$$\boldsymbol{g}^I = (0, 0, -g)\,. \tag{7}$$

The moment is a function of the thrusts, propeller distance from the body centre l and an experimentally determined constant κ.

The rotation matrix $\boldsymbol{R}^{IB}$ can be be characterised by three Euler angles: here the yaw, pitch, roll sequence is used; rotating consecutively about the inertial z axis by the yaw angle ψ, then about the (new) y axis by the roll angle θ and finally about the (resulting) x axis by the roll angle ϕ, to get the rotated body frame. The rotation matrix is then characterised as $\boldsymbol{R}^{IB}(\psi, \theta, \phi) = \boldsymbol{R}_3(\psi)\boldsymbol{R}_2(\theta)\boldsymbol{R}_1(\phi)$ [10], where

$$\boldsymbol{R}_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{8}$$

$$\boldsymbol{R}_2(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{9}$$

$$\boldsymbol{R}_3(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

B. Communication

There are two communication channels linking the vehicle to the ground station, which will be called the command and the data channel. The command channel is a low-latency, simplex channel, while the data channel is a high-bandwidth, variable latency duplex channel. The output of the high-level controller is transmitted to the vehicle over the command channel, while the data channel is used for non-critical data such as sending offboard state estimates (as described in this paper), and also vehicle feedback and parameter read/write commands from the ground station.

C. Control

Control of the quadrocopter is split into two parts, see Fig. 3: firstly a high-level controller generating desired normalised thrust and desired body rates on the ground station, in order to achieve some high-level goal (e.g. tracking a trajectory to balance a pendulum [5]). This loop runs at approximately 60 Hz, and the commands are sent to the vehicle over the command channel. The high-level controller performs feedback on a state estimate of the vehicle, using pose measurements from the motion capture system.

The second part is the low-level controller, running onboard the vehicle, which controls the motor speeds to achieve the desired body rates and total thrust, using rate gyro measurements in feedback. The onboard controller is run at 800 Hz. For the purposes of this paper, the control of the motor speeds is ignored, and assume that individual motor thrusts are directly controlled.

Each vehicle is marked with a unique configuration of three markers, which are visible to a motion capture system. This system measures the position and attitude of each vehicle, with a precision of approximately 0.1 mm and 0.1°, respectively, at a rate of 200 Hz. These measurements are then used to estimate the state of the vehicle.

Further details of the FMA can be found in [11].

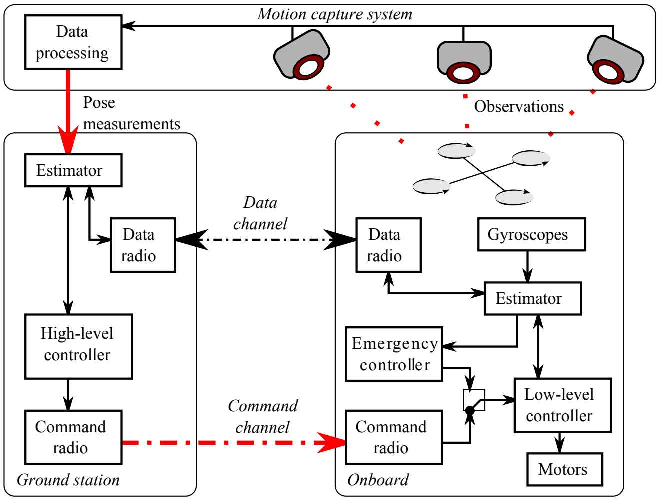

Fig. 3. System layout of the Flying Machine Arena. Critical paths are marked in red, radio connections are marked with dash-dotted lines.

III. Failure modes

We wish to guard against failures of the offboard system, which would result in its inability to control the vehicle normally. Specifically, we are concerned with the red lines of Fig. 3; loss of the motion capture data and a loss of the command channel.

Loss of the motion capture data can be due to loss of data for a specific vehicle (e.g. occlusion), or complete loss of motion capture data (e.g. software crash). Upon losing the motion capture data, the offboard system is rendered blind and can no longer estimate the dynamic state of the vehicle. The second mode of failure is loss of the command channel: in this case the offboard commands can no longer reach the vehicles.

Loss of only the data channel would mean that the failsafe mechanism cannot take over if a critical system were to also fail (as the offboard updates are needed so that the vehicle has a reliable onboard estimate). If a failure of the data channel is detected, the vehicle is landed (using the regular offboard systems) until the data channel is restored.

IV. Fail-safe mechanism

Here a mechanism is developed to delay a vehicle crash in the event of a critical subsystem failure so that the system may recover; or if a crash is inevitable, to mitigate its severity and allow sufficient warning for any people inside the arena. An emergency controller to achieve hover for short periods of time is developed, controlling on a state estimated using inertial sensors and occasional offboard updates.

It is assumed that the vehicle is in near-hover such that the angles ϕ and θ are small and that the yaw angle remains constant, so that dynamics decouple along the axes defined by a yawed reference frame:

$$\boldsymbol{x}^Y = \boldsymbol{R}_3(\psi)^T \boldsymbol{x}^I \tag{11}$$

$$\dot{\boldsymbol{x}}^Y = \boldsymbol{R}_3(\psi)^T \dot{\boldsymbol{x}}^I = (v_x, v_y, v_z) \tag{12}$$

$$\ddot{\boldsymbol{x}}^Y = \boldsymbol{R}_3(\psi)^T \ddot{\boldsymbol{x}}^I = (\dot{v}_x, \dot{v}_y, \dot{v}_z) \tag{13}$$

$$\left(\dot{\phi}, \dot{\theta}, \dot{\psi}\right) \approx (p, q, 0)\,. \tag{14}$$

Note that the yaw angle is typically controlled explicitly to some constant angle by the offboard high level controller.

Because $\boldsymbol{g}^I$ lies along the axis of rotation of $\boldsymbol{R}_3$, and is thus expressed the same in both frames, rewriting (1) yields

$$\ddot{\boldsymbol{x}}^Y \approx \boldsymbol{R}_2(\theta)\boldsymbol{R}_1(\phi)\boldsymbol{c}^B + \boldsymbol{R}_3(\psi)^T \boldsymbol{g}^I \tag{15}$$

$$\begin{bmatrix}\dot{v}_x \\ \dot{v}_y \\ \dot{v}_z\end{bmatrix} \approx \begin{bmatrix}\sin\theta\cos\phi \\ -\sin\phi \\ \cos\theta\cos\phi\end{bmatrix} c + \begin{bmatrix}0 \\ 0 \\ -g\end{bmatrix}. \tag{16}$$

Because the axes are assumed to decouple, only the strategy for estimating and controlling v_x, θ and q are described; the analogue for v_y, ϕ and p being straightforward to derive.

A. Onboard lateral state estimate

Via the data channel, the current off-board estimates of v_x and θ are periodically sent, denoted with $\bar{v}_x(t_n)$ and $\bar{\theta}(t_n)$, respectively, valid at time t_n.

The procedure to estimate the speed v_x of the quadrocopter is then as follows, where $\hat{v}$ denotes the onboard estimate of a variable v. The body rate is estimated directly from the gyroscopes, using a simple predict-correct estimator. Let Δt represent the period of the onboard loop (in reality, $\Delta t = 1.25\,\mathrm{ms}$), $\hat{q}_0$ the estimate of the gyro bias, $\check{q}[k]$ the gyro output at time step k, and I_{xx} is the moment of inertia of the body about the body x axis:

$$\hat{q}^-[k+1] = \hat{q}[k] + \Delta t\, l\, (f_2 - f_4)/I_{xx} \tag{17}$$

$$\hat{q}[k+1] = \lambda\,\hat{q}^-[k+1] + (1-\lambda)\left(\check{q}[k] - \hat{q}_0\right), \tag{18}$$

with λ a filter parameter. The rate estimate is then integrated for the angle estimate, using simple Euler integration. The acceleration is calculated from (16), and integrated for the velocity estimate, where $\hat{\theta}[k|n]$ is the estimate at step k, using an external update number n,

$$\hat{\theta}[k+1|n] = \hat{\theta}[k|n] + \hat{q}[k]\Delta t \tag{19}$$

$$\hat{v}_x[k+1|n] = \hat{v}_x[k|n] + c[k]\,\sin\hat{\theta}[k|n]\Delta t. \tag{20}$$

If a new external estimate is received at step k, the estimate is updated as follows:

$$\theta[k|n+1] = \bar{\theta}[n+1] + \sum_{l=k-N}^{k} \hat{q}[k]\Delta t \tag{21}$$

$$\hat{v}_x[k|n+1] = \bar{v}_x[n+1] + \sum_{l=k-N}^{k} c[k]\sin\hat{\theta}[k]\Delta t, \tag{22}$$

where N is age of the offboard estimate in time steps, which is the delay between the data being captured from which the offboard estimate is formed, and its arrival onboard. This means the last N steps of rate and angle estimates, as well as the collective acceleration command, need to be stored.

Upon loss of the motion capture data, or the data channel, the estimates would be propagated using only the gyroscopes, i.e. (19)-(20). The estimation error is discussed more fully below.

B. Emergency control strategy

Here the control strategy to bring the vehicle to hover is developed, i.e. lateral speeds to zero. Initially, he distinction between true state and the estimate thereof is disregarded, and we neglect the effects of discretization and design the controller in continuous time. The controller is designed by linearising (16) about hover, and it is desired that the vertical acceleration be zero, so that differentiating yields

$$c \approx g,\ \dot{c} \approx 0,\ \ddot{v}_x \approx \dot{\theta}g,\ \sin\theta \approx \theta, \tag{23}$$

where $\dot{\theta} = q^c$ is taken as system input. Choosing a feedback law

$$q^c = -2\zeta\omega_n\theta - \frac{\omega_n^2}{g}v_x \tag{24}$$

results in the differential equation for velocity

$$\ddot{v}_x + 2\zeta\omega_n\dot{v}_x + \omega_n^2 v_x = 0, \tag{25}$$

by which the speed should be driven to zero like a second order system with damping ratio ζ and natural frequency ω_n.

By (3) it is clear that $q^c = q$ is not directly controlled, but only $\dot{q}$, which is controlled through the low-level controller with the proportional feedback law

$$\dot{q} = k_q \left(q^c - q \right), \tag{26}$$

which combined with the above feedback law of (24) introduces a third pole to the system, giving

$$\frac{1}{k_q} \dddot{v}_x + \ddot{v}_x + 2\zeta\omega_n \dot{v}_x + \omega_n^2 v_x = 0. \tag{27}$$

If $k_q \geq 10|\zeta\omega_n|$, the response of the system can be approximated by the dominant roots of the second order system [12]. The values used in the FMA are $k_q = 100\,/\mathrm{s}$, $\zeta = 0.7$ and $\omega_n = 2\,\mathrm{rad/\,s}$, justifying the neglect of the effects of rotational inertia.

In reality we do not have access to the true variables, but only their estimates $\hat{q}$, $\hat{\theta}$ and $\hat{v}_x$. These will have some initial error and be further corrupted by measurement errors, which can will be modelled as a constant bias error $e_q = q[k] - \hat{q}[k]$, and some zero-mean noise whose effect is neglected. We define the error variable $e_v[k] = v_x[k] - \hat{v}_x[k]$, $e_\theta[k] = \theta[k] - \hat{\theta}[k]$, so that

$$e_\theta[k+1] = e_\theta[k] + e_q\, t = e_\theta[0] + e_q \Delta t k \tag{28}$$

$$\begin{aligned} e_v[k+1] &= e_v[k] + g\, e_\theta[k] \Delta t \\ &= e_v[0] + g\, e_\theta[0] \Delta t k + \frac{1}{2} g\, e_q \Delta t^2 k(k+1). \end{aligned} \tag{29}$$

This implies that, although the feedback law acts to drive $\hat{v}_x$ to zero (and therefore $\hat{\theta}$ too), an initial error will cause the true angle θ to settle at a non-zero value, while v_x will grow linearly. A gyro bias will cause θ to linearly diverge, and the speed to grow with k^2. By implication, therefore, the position will grow with k^3. This clearly shows the importance of well calibrated gyros, and unbounded errors mean that this strategy cannot hold the vehicle in a hover indefinitely.

To maintain zero inertial z acceleration,

$$c[k] = \frac{g}{\cos \hat{\theta}[k] \cos \hat{\phi}[k]} \tag{30}$$

is applied, from (16), instead of (23). Here it should be noted $c[k] \geq g$, i.e. the effects of noise on the angle estimate will be visible in the vehicle accelerating upwards. In the implementation, (24) is limited such that $|\theta| < 60°$.

C. Switching logic

The emergency controller is triggered when the vehicle stops receiving external commands via the command channel, or the commands indicate (by an explicit flag) that the external controller has stopped receiving measurements from the motion capture system. When the emergency controller is run for the first time, the vehicle evaluates its state estimate: firstly, the last update needs to be less than 2 s old, and the estimated angles need to be less than 60°, and the speed less than $10\,\mathrm{m/s}$. These values were chosen based on experiments, as limits of a "good" estimate. If the onboard estimate does not satisfy all of these criteria, the vehicle immediately switches off all propellers, and will fall ballistically. The idea here is that it is safer to just fall than to control the vehicle on a bad estimate.

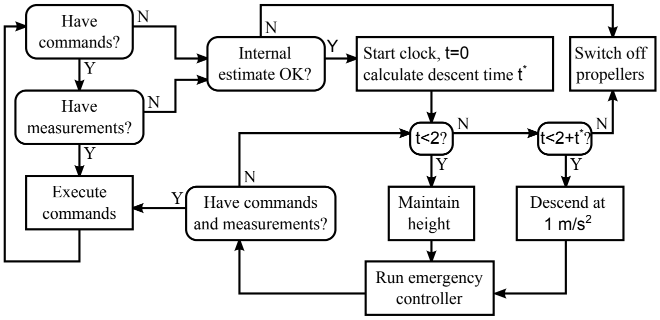

Fig. 4. Emergency controller switching logic.

The vehicle will then fly using the onboard emergency controller for a period of 2 s, after which it will enter a "crash landing" mode, where the total thrust is reduced to $c = g - 1\,\mathrm{m/s^2}$ such that the vehicle will accelerate downwards at $1\,\mathrm{m/s^2}$. The duration of the crash landing manoeuvre is calculated as the time required to descend at $1\,\mathrm{m/s^2}$ from the last measured vehicle height. When the crash landing period is completed, the vehicle switches off its motors and enters an idle mode. At any point, if the cause of the emergency procedure is resolved (i.e. the command channel comes back online, or the motion capture data returns), control is immediately handed back to the offboard controller. This is illustrated in Fig. 4.

V. Experimental results

A. Experimental setup

To test the emergency algorithm, a complete loss of data from the motion capture system is triggered. The vehicle is flown horizontally along the inertial x-axis at constant speed, using feedback on the motion capture system and external commands sent through the command channel, while external state updates are sent to the vehicle at 10 Hz through the data channel. When the vehicle passes through $x = 0$, the motion capture data is cut off, and the vehicle is commanded to fly using only onboard estimates. The external state updates also stop. For two seconds, the vehicle uses the controller of Section IV-B to attempt a level hover, after which it will descend at $1\,\mathrm{m/s^2}$ for the time necessary to descend from its last observed height.

The damping ratio for the emergency controller is set to $\zeta = 0.7$, and the natural frequency to $\omega_n = 2\,\mathrm{rad/\,s}$. The age of the offboard estimate is set to $N = 21$ internal cycles (or about 27 ms) – this is an estimate of the mean total time elapsed between the motion capture frame grab and corresponding update arriving at the vehicle, and is the sum of the motion capture image processing, offboard estimation processing and data channel communication times.

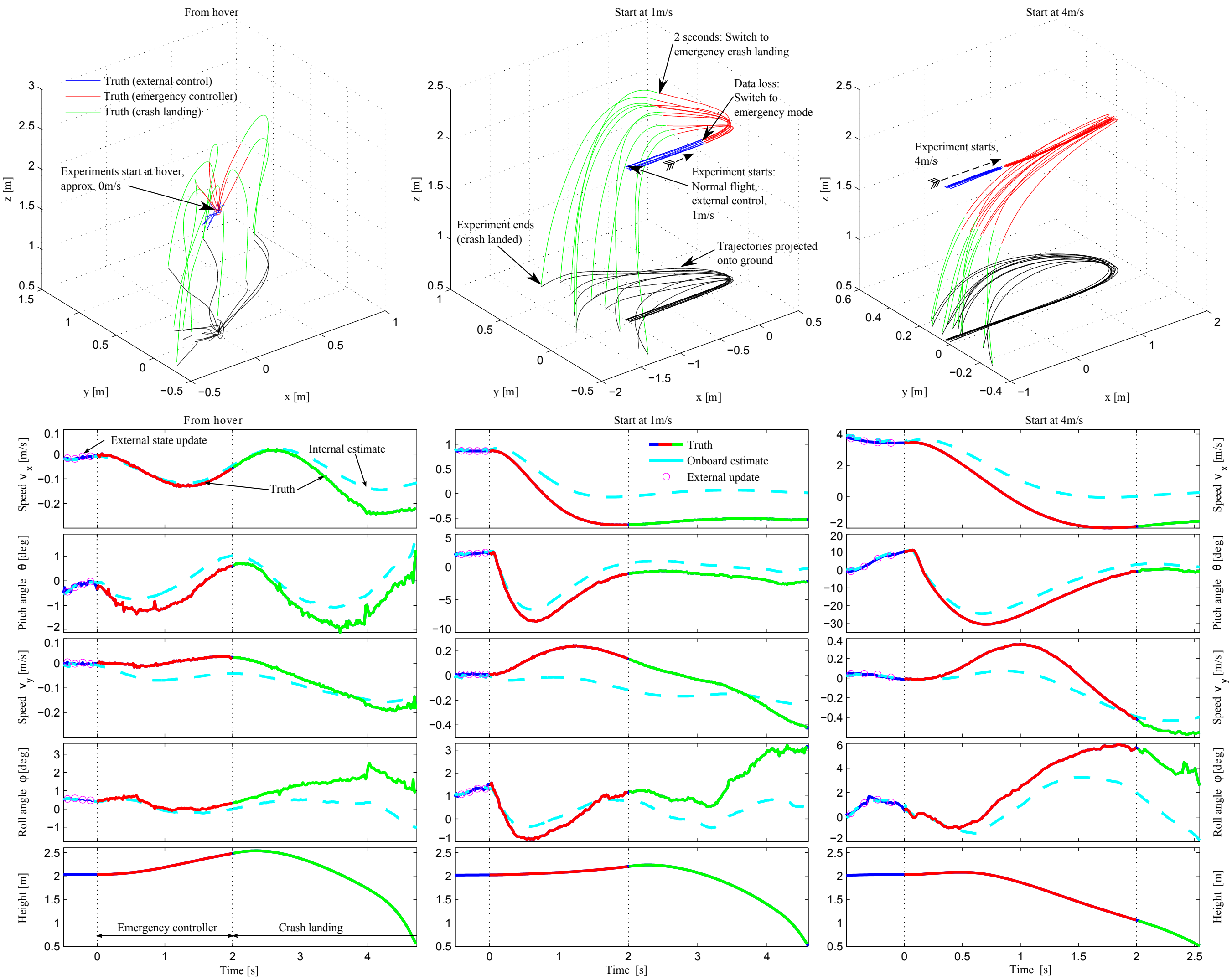

Fig. 5. The top row of plots show vehicle trajectories under emergency onboard control, for three different initial conditions: starting at hover (left), with 1 m/s initial lateral velocity, and 4 m/s (right). Each plot shows ten trajectories, each divided into three stages: external control is marked blue, emergency hover red and the emergency descent crash landing in green. The experiment terminates at a height of 0.5 m. For ease of interpretation, the paths are also projected onto the $x-y$ plane in black. The lower plots show time histories of a typical trajectory for each starting speed, also showing the onboard estimate. Note that the colours of the "truth" line correspond to those of the top plots, and the differing scales for the lower plots. Data from the motion capture system is used as "truth".

B. Calibration

It is clear from Section IV-B that the emergency control strategy relies on a well-calibrated vehicle, i.e. with correctly identified gyroscope biases and the ability to accurately follow motor thrust commands. To this end a static calibration routine has been developed in the FMA, which identifies the gyroscope biases and a thrust mapping factor for each rotor.

The initial guess for the gyroscope biases $\hat{p}_0$ is formed by integrating the gyro output for the first two seconds after the vehicle has been switched on, and taking the mean value. A further calibration is performed when the vehicle is in hover (as determined by the motion capture system), and the gyroscope outputs are again averaged over a similar time period. During the hover periods, the system also calibrates for a static measurement frame misalignment, noting that the thrust vector during hover must point exactly opposite gravity and thus any pitch or roll angles measured are the frame misalignments.

Since direct control of the total thrust is assumed in (23), the propellers must be characterised: due to e.g. motor and propeller wear, the individual motors will not produce exactly the expected forces. To compensate for this, a static thrust correction factor γ_i for each propeller i is calibrated for, which relates the thrust output f_i to the desired motor thrust f_i^c by $f_i = \gamma_i f_i^c$. Combining this with (4) and (6) and setting the moments to zero and total thrust to gravity yields

$$\begin{bmatrix} 0 & l f_2^c & 0 & -l f_4^c \\ -l f_1^c & 0 & l f_3^c & 0 \\ \kappa f_1^c & -\kappa f_2^c & \kappa f_3^c & -\kappa f_4^c \\ \frac{f_1^c}{m} & \frac{f_2^c}{m} & \frac{f_3^c}{m} & \frac{f_4^c}{m} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ g \end{bmatrix}. \qquad (31)$$

Calibration is automatically performed whenever the vehicle is in stationary hover throughout a flight.

C. Results

The trajectory results for a series of experiments are shown in Fig. 5. Over ten runs the average velocity magnitude at the end of 2 s of emergency onboard control is $0.3\,\mathrm{m/s}$ when starting from hover, $0.6\,\mathrm{m/s}$ when starting with a lateral speed of $1\,\mathrm{m/s}$, $1.1\,\mathrm{m/s}$ when starting with at $2\,\mathrm{m/s}$ and $2.0\,\mathrm{m/s}$ when starting with at $4\,\mathrm{m/s}$. When starting at $8\,\mathrm{m/s}$, the vehicle will hit the floor before 2 s, but the velocity magnitude at that time is on average $3.5\,\mathrm{m/s}$, which represents an 80% decrease in vehicle kinetic energy – i.e. even though a crash is not prevented, its severity is hugely decreased. Furthermore, the crash is delayed by some time, allowing warning of any people in the flying space; and crucially if the failure is quickly resolved, the external control can resume control of the vehicle in the air, thereby avoiding a crash altogether.

Notable also is that the vehicle consistently overshoots its velocity target, i.e. the final velocity achieved points in the direction opposite the initial velocity. There are two likely explanations for this error: firstly, the propellers are more efficient during the initial phases of the braking manoeuvre, due to an increased angle of attack [13], meaning that the produced thrust during braking exceeds the desired thrust. Secondly, the vehicle experiences aerodynamic drag for which the emergency controller does not compensate. The effect of the drag can be seen in the history plot when starting at $4\,\mathrm{m/s}$ in Fig. 5, where the vehicle has an initial large pitch angle (ca. $10°$), but is not accelerating, i.e. the lateral components of the thrust are needed to balance out the drag force. The final velocity then represents the total impulse imparted by the drag force.

In Fig. 5 we see that the vehicle consistently accelerates slightly upwards when entering the emergency controller from a hover. This is most likely due to the effects of zero-mean noise on the angle estimate, as noted in Section IV-B, (30). The systematic tendency to turn to the left from speed is most likely due to vehicle mis-calibration, e.g. the left propeller producing less thrust than the opposing propeller.

VI. Conclusion and Outlook

The strategy described in this paper allows for much greater confidence when using the FMA. They allow the system to recover from short-term failures which would otherwise render the vehicles uncontrollable, and reduce the severity of long-term failures. The strategy is implemented in such a way to be transparent to the user, allowing individual users to rapidly implement new controllers and run experiments, without in each case needing to be concerned with additional safety aspects.

This system is currently deployed, and runs in the background whenever a demonstration or experiment is in progress. One notable example of its use was during the architecture project where the mobile infrastructure of the FMA was deployed in Orléans, France (see Fig. 1). During operations, the commercial motion capture software crashed while a vehicle was flying – the emergency controllers kicked in and an alarm sounded. The emergency controller allowed sufficient time for an operator to grab a vehicle, protecting it from what would otherwise have been a fall of approximately 2 m onto a bare concrete floor.

Further improvements are possible to the emergency controller, especially if additional onboard sensors can be incorporated. Use of accelerometers, particularly, could be useful: both in feedback on lateral accelerations, taking advantage of aerodynamic effects to stabilise lateral speeds [14], and in feedback on the body z acceleration, reducing the effects of relative airflow on total produced thrust. Finally, the systematic nature of the errors seen in Fig. 5 suggest that a learning strategy might greatly improve the hovering performance.

VII. Acknowledgements

This work builds on contributions to the FMA by many individuals, and particularly the authors would like to thank Sergei Lupahshin, Markus Hehn, Angela Schöllig and Federico Augugliaro for their discussions and feedback during the implementation of this work; Christoph Wegmüller, Guillaume Ducard and Markus Hehn for their work on vehicle calibration; and Michael Sherback and Thomas Kägi.

References

[1] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *Robotics Automation Magazine, IEEE*, vol. 17, no. 3, pp. 56 –65, September 2010.

[2] G. Hoffmann, D. Rajnarayan, S. Waslander, D. Dostal, J. Jang, and C. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (starmac)," in *Digital Avionics Systems Conference*, 2004.

[3] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *Control Systems Magazine, IEEE*, vol. 28, no. 2, pp. 51–64, April 2008.

[4] A. Schöllig, F. Augugliaro, S. Lupashin, and R. D'Andrea, "Synchronizing the motion of a quadrocopter to music," may. 2010, pp. 3355 –3360.

[5] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011.

[6] M. W. Mueller, S. Lupashin, and R. D'Andrea, "Quadrocopter ball juggling," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[7] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments," in *Robotics: Science and Systems Conference*, June 2008.

[8] S. Weiss, M. Achtelik, M. Chli, and R. Siegwart, "Versatile distributed pose estimation and sensor self-calibration for an autonomous mav," in *IEEE International Conference on Robotics and Automation*, 2012.

[9] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *IEEE International Conference on Robotics and Automation*, 2009.

[10] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics Second Edition*. AIAA, 2007.

[11] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1642 –1648.

[12] R. Dorf and R. Bishop, *Modern Control Systems, Eleventh Edition*. Prentice-Hall, 2008.

[13] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control engineering practice*, vol. 19, pp. 1023–1036, June 2011.

[14] P. Martin and E. Salaün, "The true role of accelerometer feedback in quadrotor control," in *IEEE International Conference on Robotics and Automation*, 2010.