Nonlinear MPC with Motor Failure Identification and Recovery for Safe and Aggressive Multicopter Flight

Dimos Tzoumanikas, Qingyue Yan, and Stefan Leutenegger

***Abstract*— Safe and precise reference tracking is a crucial characteristic of Micro Aerial Vehicles (MAVs) that have to operate under the influence of external disturbances in cluttered environments. In this paper, we present a Nonlinear Model Predictive Control (NMPC) that exploits the fully physics based non-linear dynamics of the system. We furthermore show how the moment and thrust control inputs can be transformed into feasible actuator commands. In order to guarantee safe operation despite potential loss of a motor under which we show our system keeps operating safely, we developed an Extended Kalman Filter (EKF) based motor failure identification algorithm. We verify the effectiveness of the developed pipeline in flight experiments with and without motor failures.**

I. Introduction

During the past years, the use of MAVs in applications such as environmental monitoring, aerial filming, surveillance and search and rescue has dramatically increased. However, most commonly used control algorithms lack the necessary robustness needed for the critical applications stated above and they often struggle when aggressive maneuvers are required. To some extent, these problems can be eliminated when the Micro Aerial Vehicle (MAV) model is taken into account in the control design. Model based control approaches such as Model Predictive Controller (MPC) have become more popular in robotics thanks to increasing computational capabilities and improved algorithmic efficiency. The design and implementation of such algorithms on real robots has become significantly easier due to the open source availability of optimisation and control toolboxes such as [1], [2], [3]. At the same time, robust performance under mechanical failures (such as motor failures), can only be achieved when the failure can be correctly identified and appropriately handled by the control design. In our paper we address the problem of aggressive, precise and fault tolerant MAV navigation. Our contributions are as follows:

- The design of a quaternion based non-linear model predictive controller with body torques and collective thrust as the control inputs.
- The design of a new control allocation algorithm that maps the desired control inputs into feasible thrust commands for each motor. We consider the general case where the motors can generate both positive and negative thrust. Our algorithm takes into account the different motor coefficients for normal and inverted rotation eliminating the need for symmetrical propellers. We show how this can be applied in a motor failure scenario.
- The design of an EKF which monitors the health of each individual actuator (motor/propeller) which we use for fast identification of an actuator failure.
- Seamless integration of the failure detection scheme with NMPC and allocation under actuator failure: in particular, our hexacopter platform maintains full controllability (position and yaw) after the loss of one actuator.

This work has been supported by the EPSRC grant Aerial ABM EP/N018494/1 and Imperial College London.

The authors are with the Smart Robotics Lab, Department of Computing, Imperial College London, UK. {dt214, qy916, s.leutenegger}@ic.ac.uk

Video link: https://youtu.be/cAQeSZ3tIqY

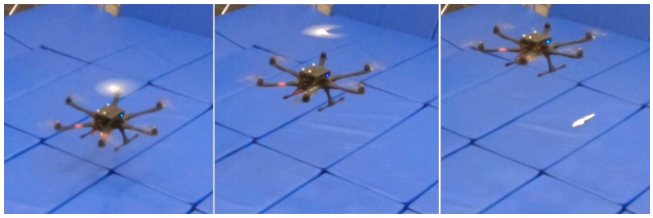

Fig. 1. Our MAV experiencing a propeller loss. The fault is identified online and the MAV can still control its position and orientation.

A. Related Work

Regarding MPC in MAVs, the most common approach is that of a cascade connection between a position and an attitude controller. In the simplest form, a linear model can be used for the translational dynamics, resulting in an optimisation problem whose solution can be solved online [4], [5] or pre-computed in the form of lookup tables [6], [7]. The use of a non-linear model for the translational dynamics such as the one presented in [8] presents performance improvements especially when tracking of aggressive trajectories is required. The approach of the cascaded position-attitude controllers has become popular due to its ease of use, since most of the available platforms come with a pre-tuned attitude controller. However, it relies on the assumption that the attitude dynamics can be controlled independently, requiring bandwidth separation between the successive loops, i.e. slow control of attitude. The aforementioned works furthermore use Euler angles for the vehicle orientation which prohibits the operation close to gimbal lock. Analogously to the position-attitude approach the authors of [9] and [10] propose a quaternion based position controller which uses the angular rates as control inputs. These were assumed to be tracked perfectly by a separate angular rate controller.

The benefits of using the true non-linear model of the

MAV has been successfully illustrated in [11] where an attitude NMPC was employed to stabilise a hexacopter with a motor failure. Additionally, the authors of [12] proposed an Sequential Linear Quadratic (SLQ) MPC algorithm able to run onboard an MAV and capable of following full state trajectory commands. Similarly, in [13] simulation results of an SLQ controller stabilizing a quadrotor with slung load and a quadrotor with motor failure were presented.

For the control allocation – that is, the mapping of the control inputs to actuator commands – the most commonly used method employs the pseudo-inverse of the allocation matrix (e.g. [14], [15]). In this case the actuator commands can be obtained through a simple matrix by vector multiplication. However, the main drawback is the fact that it can produce actuator commands that are not feasible. Control allocation techniques that respect the actuator limits, such as the ones presented in [16], [17], result in better trajectory tracking. This is partially due to the prioritisation of the roll/pitch moments and collective thrust over the yaw moment which does not directly contribute to position tracking. Another interesting approach is the one presented in [18], where the minimum energy solution is obtained by solving an optimisation problem. The authors exploit the structure of the allocation matrix nullspace in order to transform the original optimisation problem into a computationally cheaper one. Their method can be used on platforms equipped with bidirectional capable motors but requires the use of symmetrical propellers. When non symmetric propellers are used, the resulting allocation matrix is not constant but depends on the direction of rotation of each motor.

As far as fault identification and fault tolerant control are concerned, the authors of [19] were among the first to show stable position control (despite losing yaw control) with a quadrotor despite the loss of a single or two opposing propellers. The authors of [20] stabilised a hexacopter experiencing a motor failure. However, unlike us, they used an unconventional hexacopter motor layout which enables orientation control despite the loss of up to two motors. None of the above methods includes online fault estimation. This is done in [21] where the residuals between the measured and predicted orientation and angular rate are used as criteria for detecting motor failures. Another example is the work presented in [22], where faults are identified based on the measured motor speed and electrical current. Compared to these approaches, our method achieves up to three times faster fault detection without relying on additional sensors apart from the onboard Inertia Measurement Unit (IMU).

II. Notation and Definitions

Vectors are denoted as e.g. $\mathbf{v}$, when required with coordinate frame $\underrightarrow{\mathcal{F}}_A$ representation as ${}_A\mathbf{v}$. A rotation matrix $\mathbf{C}_{AB}$ changes the coordinates of a vector from $\underrightarrow{\mathcal{F}}_B$ to $\underrightarrow{\mathcal{F}}_A$ as ${}_A\mathbf{v} = \mathbf{C}_{AB}\,{}_B\mathbf{v}$. We use quaternions analogously i.e. $\mathbf{q}_{AB}$. We further denote the position of a point P relative to the origin of $\underrightarrow{\mathcal{F}}_A$ as ${}_P\mathbf{r}_A$. The motion of the MAV, with body frame $\underrightarrow{\mathcal{F}}_B$ (x: forward, y: left, z: upward), is referenced relative to the World-frame $\underrightarrow{\mathcal{F}}_W$ (z-axis upward).

III. System Overview

The software pipeline presented in this paper consists of the following different blocks: (i) the NMPC which receives full state trajectory estimates and commands and outputs body torques and collective thrust as the control inputs; (ii) the Control allocation block which transforms the control inputs to actuator commands and finally (iii) the failure detection algorithm which estimates the health status of each motor and notifies the control allocation block in the case of a failure. An overview of the system is given in Figure 2.

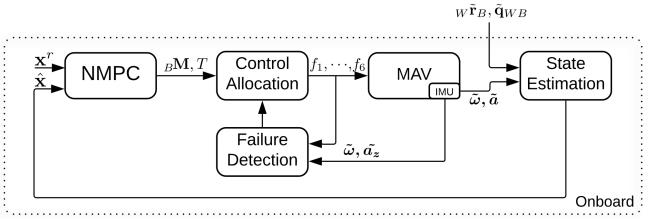

Fig. 2. Overview of the various software components that run onboard the MAV. The control loop runs at 100Hz while the failure detection EKF at 400Hz.

IV. Model Based Control

A. MAV Dynamics

The Newton-Euler equations are used to model the MAV dynamics. We ignore less significant phenomena such as the effect of the aerodynamic friction and the gyroscopic moments due to the rotation of the propellers (but our model could be extended accordingly with ease). The MAV dynamics can then be written in the following form:

$$ {}_W\dot{\mathbf{r}}_B = {}_W\mathbf{v}_B\,, \tag{1a}$$

$$ \dot{\mathbf{q}}_{WB} = \frac{1}{2}\mathbf{\Omega}({}_B\boldsymbol{\omega})\mathbf{q}_{WB}\,, \tag{1b}$$

$$ {}_W\dot{\mathbf{v}}_B = \frac{1}{m}\mathbf{C}_{WB}\,{}_B\mathbf{T} + {}_W\mathbf{g}, \tag{1c}$$

$$ {}_B\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}({}_B\mathbf{M} - {}_B\boldsymbol{\omega} \times \mathbf{J}\,{}_B\boldsymbol{\omega}), \tag{1d}$$

$$ \mathbf{\Omega} = \begin{bmatrix} {}_B\boldsymbol{\omega}^{\times} & {}_B\boldsymbol{\omega} \\ -{}_B\boldsymbol{\omega}^{\top} & 0 \end{bmatrix}, \tag{1e}$$

where $[\,]^{\times}$ stands for the skew symmetric operator, ${}_W\mathbf{g}$ for the gravitational acceleration, m for the MAV mass, and $\mathbf{J}$ for the inertia tensor. The thrust vector ${}_B\mathbf{T} := [0, 0, T]^{\top}$ acting on the MAV Centre of Mass (CoM) solely depends on the collective thrust T generated by the motors. This together with the moments ${}_B\mathbf{M}$ are considered as the control input $\mathbf{u} := [{}_B\mathbf{M}, T]^{\top} \in \mathbb{R}^4$. The control state $\mathbf{x} := [{}_W\mathbf{r}_B, \mathbf{q}_{WB}, {}_W\mathbf{v}_B, {}_B\boldsymbol{\omega}]^{\top} \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^6$, consists of the MAV position, orientation, linear and angular velocities respectively. The motor dynamics are considered significantly faster than the MAV body dynamics and are thus neglected. The generated thrust and moments from the i^{th} motor are given by:

$$ f_i = k_T\omega_i^2, \tag{2a}$$

$$ M_i = (-1)^{i+1}k_M f_i. \tag{2b}$$

Unlike approaches such as [16] where the relationship between the motor command and the achieved motor thrust was

approximated as a quadratic polynomial, we first estimated the motor and moment coefficients defined in (2) and we later identified the relationship between the motor command and the achieved angular velocity. Figure 3 shows the results obtained from the experimental identification of the thrust and moment coefficients k_T and k_M using a load cell. Since we use non symmetrical propellers which are optimised for rotation in one direction, we identified two sets of coefficients k_T and k_M one for normal rotation and another one for inverted. Regarding the motor command to angular velocity relationship, we experimentally determined the dependency on input battery voltage which does not remain constant during flight. The identification results are illustrated in Figure 4. The obtained quadratic polynomials for different voltage levels were stored in lookup tables and were used online depending on the measured battery voltage [1].

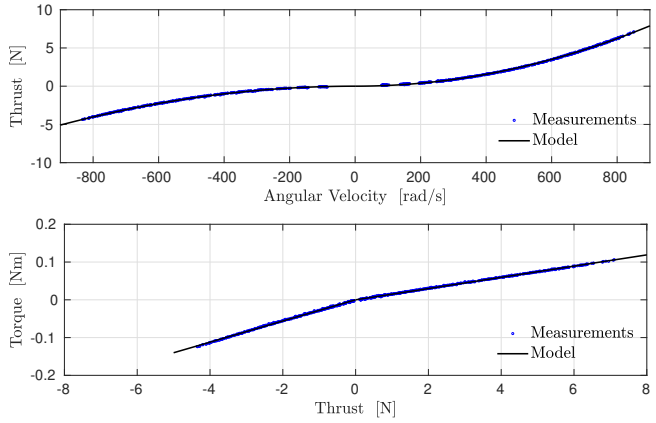

Fig. 3. Experimental identification of the thrust and moment coefficients using a load cell. Each dot corresponds to the mean of 100 measurements and the solid lines correspond to the fitted model as defined in (2). The use of non symmetrical propellers results in different curves for normal and inverted rotation. The least-squares fit error for the thrust and moment model for normal motor rotation is 5×10^{-2} N and 8.5×10^{-4} Nm respectively. The same quantities for inverted rotation are 4.4×10^{-2} N and 1.3×10^{-3} Nm.

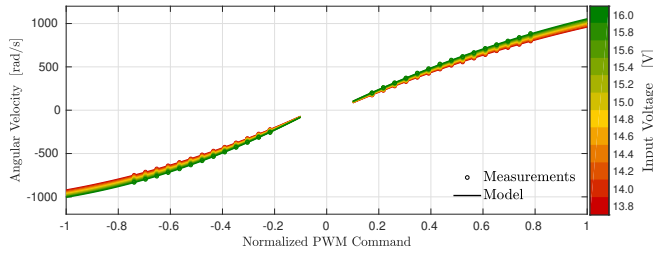

Fig. 4. Experimental identification of the relationship between the normalised PWM command and the achieved angular velocity. Dots correspond to averaged measurements and solid lines to the fitted quadratic polynomials. Different colours correspond to different voltage levels applied on the ESCs (here only plotting 12 different cases for visualization purposes).

[1] An easier and more accurate way of handling this problem is by using motors which are equipped with encoders and perform closed loop angular velocity control using the encoder information. However, the commercially available hardware, see `http://www.iq-control.com/` is mainly designed for small racing drones and not ones like ours which carries significant payload.

B. Nonlinear MPC (NMPC)

For the control formulation, we define the following time-varying error functions for the position, linear and angular velocity, orientation and control input respectively:

$$\mathbf{e}_r = {}_W\mathbf{r}_B - {}_W\mathbf{r}_B^r, \tag{3a}$$
$$\mathbf{e}_v = {}_W\mathbf{v}_B - {}_W\mathbf{v}_B^r, \tag{3b}$$
$$\mathbf{e}_\omega = {}_B\boldsymbol{\omega} - \mathbf{C}_{BB^r}\, {}_{B^r}\boldsymbol{\omega}^r, \tag{3c}$$
$$\mathbf{e}_q = [\mathbf{q}_{WB}^{-1} \otimes \mathbf{q}_{WB}^r]_{1:3}, \tag{3d}$$
$$\mathbf{e}_u = \mathbf{u} - \mathbf{u}^r. \tag{3e}$$

Apart from the orientation error which is obtained through quaternion multiplication, the rest corresponds to the Euclidean difference between the actual and desired (here denoted with the superscript r) quantity.

We compute the optimal control input $\mathbf{u}^*$ sequence online by solving the following optimisation problem:

$$\mathbf{u}^* = \underset{\mathbf{u}_0,\dots,\mathbf{u}_{N_f-1}}{\operatorname{argmin}} \Big\{ \Phi(\mathbf{x}_{N_f}, \mathbf{x}_{N_f}^r) + \sum_{n=0}^{N_f-1} L(\mathbf{x}_n, \mathbf{x}_n^r, \mathbf{u}_n) \Big\}, \tag{4a}$$
$$\text{s.t.} : \mathbf{x}_{n+1} = \mathbf{f}_d(\mathbf{x}_n, \mathbf{u}_n), \tag{4b}$$
$$\mathbf{x}_0 = \hat{\mathbf{x}}, \tag{4c}$$
$$u_{\text{lb}} \leq u_i \leq u_{\text{ub}}, \quad i = 1, \dots, N, , \tag{4d}$$

where N_f is the number of time steps, $\hat{\mathbf{x}}$ a known initial state, $\mathbf{f}_d$ the discrete-time version of the MAV dynamics given in (1) and u_{lb}, u_{ub} lower and upper bounds for the inputs u_i. We use quadratic costs for the final and intermediate terms defined as:

$$\Phi(\mathbf{x}_{N_f}, \mathbf{x}_{N_f}^r) = \mathbf{e}_r^\top \mathbf{Q}_r \mathbf{e}_r + \mathbf{e}_v^\top \mathbf{Q}_v \mathbf{e}_v + \mathbf{e}_q^\top \mathbf{Q}_q \mathbf{e}_q + \mathbf{e}_\omega^\top \mathbf{Q}_\omega \mathbf{e}_\omega,$$
$$L(\mathbf{x}, \mathbf{x}^r, \mathbf{u}) = \mathbf{e}_r^\top \mathbf{Q}_r \mathbf{e}_r + \mathbf{e}_v^\top \mathbf{Q}_v \mathbf{e}_v + \mathbf{e}_q^\top \mathbf{Q}_q \mathbf{e}_q + \mathbf{e}_\omega^\top \mathbf{Q}_\omega \mathbf{e}_\omega + \mathbf{e}_u^\top \mathbf{Q}_u \mathbf{e}_u,$$

with $\mathbf{Q} \succcurlyeq 0$ gain matrices of appropriate dimensions which are considered tuning parameters. In our implementation we use a 10 ms discretisation step and a constant time horizon T_f = 2.0 s. For the online computation of the optimal input we use the CT toolbox [2] and the Gauss-Newton Multiple Shooting (GNMS) algorithm (outlined in [23]) which result in an average computation time of 5.2 ms with standard deviation of 0.6 ms. At each GNMS iteration dynamically feasible state and input increments $\delta\mathbf{x}$, $\delta\mathbf{u}$ around the state and input trajectories $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{N_f}\}$, $\bar{\mathbf{U}} = \{\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1, \cdots, \bar{\mathbf{u}}_{N_f-1}\}$ are computed. We obtain the dynamics of the minimal state perturbation $\delta\mathbf{x} := [\delta\mathbf{r}, \delta\boldsymbol{\theta}, \delta\mathbf{v}, \delta\boldsymbol{\omega}]^\top \in \mathbb{R}^{12}$ around the state $\bar{\mathbf{x}}$ by introducing the local quaternion perturbation $\mathbf{q} = \bar{\mathbf{q}} \otimes \boldsymbol{\delta}\mathbf{q}$ with $\boldsymbol{\delta}\mathbf{q} := \left[\operatorname{sinc}\left\|\frac{\delta\boldsymbol{\theta}}{2}\right\| \frac{\delta\boldsymbol{\theta}}{2}, \cos\left\|\frac{\delta\boldsymbol{\theta}}{2}\right\|\right]^\top$. The dynamics for the rotation vector $\delta\boldsymbol{\theta} \in \mathbb{R}^3$ are given by: $\dot{\delta\boldsymbol{\theta}} = {}_B\boldsymbol{\omega} - \frac{1}{2}\, {}_B\boldsymbol{\omega}^\times \delta\boldsymbol{\theta}$, while the rotation matrix $\mathbf{C}_{WB}$ can be approximated as: $\mathbf{C}_{WB} \approx \bar{\mathbf{C}}_{WB}(\mathbf{I} + \delta\boldsymbol{\theta}^\times)$. After each iteration the state trajectory is updated as: $\mathbf{x} = [{}_w\bar{\mathbf{r}}_B + \delta\mathbf{r}, \bar{\mathbf{q}}_{WB} \otimes \delta\mathbf{q}, {}_W\bar{\mathbf{v}}_B + \boldsymbol{\delta}\mathbf{v}, {}_B\bar{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}]^\top$.

C. Control allocation

As stated earlier, the control allocation problem involves mapping the control inputs $\mathbf{u}^*$ to feasible actuator commands $\mathbf{f}$. We tackle this by solving the following Quadratic Program (QP):

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}} \Big(\left\| \mathbf{A}\mathbf{f} - \mathbf{u}^* \right\|_{\mathbf{W}}^2 + \lambda \left\| \mathbf{f} \right\|_2^2 \Big) \tag{6a}$$

$$\text{s.t.} : f_{\min} \leq f_i \leq f_{\max}, \quad i = 1, \ldots, N, \tag{6b}$$

where N is the number of motors. The allocation matrix $\mathbf{A} \in \mathbb{R}^{4\times N}$, which we will present later, depends on the MAV geometry and its motor coefficients, whereas $f_{\min}, f_{\max}$ correspond to the minimum and maximum attainable thrust. In order to prioritise the roll/pitch moments and the collective thrust over the yaw moment, we use the weighting matrix $\mathbf{W} \in \mathbb{R}^{4\times 4}$. The scalar $\lambda \in \mathbb{R}^+$ is used such that solutions with smaller norm are preferred. When a feasible control input is commanded, solution of (6) coincides with the one obtained by using the pseudo-inverse of $\mathbf{A}$, namely $\mathbf{f} = \mathbf{A}^\dagger \mathbf{u}^*$.

Since we are interested in solving the control allocation problem for the general case where the motors can produce both positive and negative thrust, we introduce the vector $\mathbf{d} = [d_1, d_2, \cdots, d_N]$ with $d_i \in \{0, 1\}, \forall i = 1, \ldots N$. We thus use the binary variables d_i to indicate whether the i^{th} motor is spinning in its intended normal direction–corresponding to positive thrust ($d_i = 0$)–or otherwise in the inverse ($d_i = 1$).

The original optimisation problem (6) is transformed to:

$$\mathbf{f}^*, \mathbf{d}^* = \underset{\mathbf{f},\mathbf{d}}{\operatorname{argmin}} \Big(\left\| \mathbf{A}(\mathbf{d})\mathbf{f} - \mathbf{u}^* \right\|_{\mathbf{W}}^2 + \lambda \left\| \mathbf{f} \right\|_2^2 \Big) \tag{7a}$$

$$\text{s.t.} : f_{\min}^+(1 - d_i) + f_{\min}^- d_i \leq f_i \leq f_{\max}^+(1 - d_i) + f_{\max}^- d_i, \tag{7b}$$

where the superscript $+$ or $-$ in $f_{\min}^+$, $f_{\min}^-$, $f_{\max}^+$, $f_{\max}^-$ has been used to indicate normal and inverted rotation respectively. The vector $\mathbf{d}$ which encodes the direction of rotation is now an optimization variable and the allocation matrix $\mathbf{A}$ is a function of d. For the case of the hexacopter used in our experiments, $\mathbf{A}(\mathbf{d})$ takes the following form:

$$\mathbf{A}(\mathbf{d}) = \begin{bmatrix} ls_{30} & l & ls_{30} & -ls_{30} & -l & -ls_{30} \\ -lc_{30} & 0 & lc_{30} & lc_{30} & 0 & -lc_{30} \\ k_M(d_1) & -k_M(d_2) & k_M(d_3) & -k_M(d_4) & k_M(d_5) & -k_M(d_6) \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{8}$$

where l stands for the MAV arm length, $s_{30} = \sin(30^o)$, $c_{30} = \cos(30^o)$, $k_M(d_i) = (1 - d_i)k_M^+ + d_i k_M^-$ and k_M^+, k_M^- denote the normal and inverted moment coefficients identified in Section IV-A.

The resulting optimisation is a mixed integer programming problem. However, since the possible values of $\mathbf{d}$ are finite (72 in the case of a hexacopter), we can solve a single QP for each single value of $\mathbf{d}$. The global optimum $\mathbf{f}^*$ of the optimisation problem defined in (7) corresponds to the solution of the QP with the minimum cost. From a practical perspective solving 72 QPs instead of a single one does not affect significantly the overall control computation time as this is dominated by the computation of the optimal input $\mathbf{u}^*$ as described in the previous section. This is owing to the small number of optimisation variables in a single QP tailored to the solver, CVXGEN [3]. In our implementation, solving 72 QPs, storing the results in a vector and finally sorting it in ascending order consistently takes less than 0.4 ms. We acknowledge, however, that our method is more resource demanding compared to methods using the pseudoinverse which can be easily implemented on a microcontroller.

It was experimentally found that reverting the direction of rotation during flight is particularly impractical. This is because the motor dynamics are significantly slower when a direction change is commanded. As our control model does not capture this behaviour, we can prevent unnecessary direction change commands by augmenting the optimisation (7) similarly to [18] with the $\mathbf{f} \in \mathcal{F}_{\text{hyst}}$ constraint, where $\mathcal{F}_{\text{hyst}}$ is the set of rotor thrusts that does not require a per motor direction change when this has already happened during the past time interval t_{hyst}. The solution satisfying this constraint can be found with a single iteration over the vector of 72 possibilities. The threshold t_{hyst} can be iteratively decreased until a good (e.g. $\left\| \mathbf{A}\mathbf{f} - \mathbf{u}^* \right\|_{\mathbf{W}}^2 < \epsilon$) solution is found.

V. Fault Identification

Our goal is to online estimate whether one or more motors have failed (consequently limiting maximum thrust and moments). To do so, we introduce the health variable $h_i \in \mathbb{R}$ for each individual motor i and assume that the effective force generated from the i^{th} motor is $f_i^e = L(h_i)f_i$, where $L(h) = \frac{1.05}{1+e^{-h}}$ is the logistic function shown in Figure 5 and f_i corresponds to the respective motor thrust. Intuitively, we expect that $L(h_i) = 1$ for a healthy motor and $L(h_i) \to 0$ for a stopped one. We implemented an EKF that estimates the set of health variables h_i online (and thus the motor thrust f_i) for each individual motor. The effective body torques and collective thrust are now given by:

$$\begin{bmatrix} {}_B\mathbf{M} \\ T \end{bmatrix} = \mathbf{A} \begin{bmatrix} L(h_1)f_1 & \cdots & L(h_6)f_6 \end{bmatrix}^\top, \tag{9}$$

where $\mathbf{A}$ is the allocation matrix defined in (8), which depends on the moment coefficients and the motor direction of rotation.

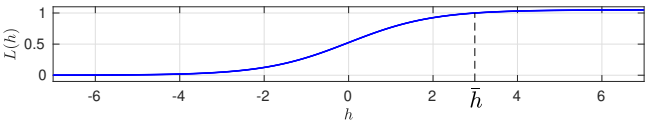

Fig. 5. The logistic function used for the EKF. Notice, that it is appropriately scaled such that $\bar{h} = L^{-1}(1)$ has finite value.

In our EKF, we use the following prediction and observation models:

$${}_B\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}({}_B\mathbf{M} - {}_B\boldsymbol{\omega} \times \mathbf{J}\,{}_B\boldsymbol{\omega}) + \boldsymbol{w}_\omega, \tag{10a}$$

$$\dot{h}_i = \frac{1}{\tau_h}(\bar{h} - h_i) + w_h, \tag{10b}$$

$$\dot{f}_i = \frac{1}{\tau_f}(f_i^r - f_i + w_f), \tag{10c}$$

$$\mathbf{z} := \begin{bmatrix} {}_B\boldsymbol{\omega} + \boldsymbol{v}_{\tilde{\omega}} \\ T + v_T \end{bmatrix}. \tag{10d}$$

The noises $\boldsymbol{w}_\omega$, w_h, and w_f are Gaussian white noise processes with densities σ_ω, σ_h, and σ_f, respectively; the measurement $\mathbf{z}$ is assumed to be corrupted by $\boldsymbol{v}_{\tilde{\omega}} \sim \mathcal{N}(0, \sigma^2_{\tilde{\omega}}\mathbf{I})$ and $v_T \sim \mathcal{N}(0, \sigma^2_{\tilde{T}})$. Furthermore, f_i^r stands for the per-motor reference thrust as given by the control allocation and τ_f for the time constant characterising the first order motor thrust dynamics. The measurements ${}_B\tilde{\boldsymbol{\omega}}$ and $\tilde{T}$ required for the EKF update are obtained using the onboard IMU. For ${}_B\tilde{\boldsymbol{\omega}}$ we use the bias corrected gyro measurements and for measured collective thrust $\tilde{T}$ we use $\tilde{T} \approx ma_z$ with m denoting the known mass of the MAV and a_z the accelerometer measurement along the z axis. Notice that our observation model for the collective thrust does not account for the ${}_B\boldsymbol{\omega} \times {}_B\mathbf{v}$ term which appears in the Body frame expressed linear acceleration dynamics. The values for the noise parameters and model constants are given in Table I.

TABLE I
EKF PARAMETERS

σ_ω	3.16 rad/(s$\sqrt{\text{hz}}$)	σ_h	0.31 $\sqrt{\text{hz}}^{-1}$	σ_f	0.94 N/$\sqrt{\text{hz}}$	$\sigma_{\tilde{\omega}}$	0.01 rad/s
$\sigma_{\tilde{T}}$	0.1 N	τ_h	0.3 s	τ_f	0.01 s	$\bar{h}$	2.99

In order to avoid false positives due to e.g. inaccurate model, we use the estimated value of h_i and its estimated uncertainty. We thus consider a motor failed when $L(h_i + 3\sigma_i) < 0.5$ (with σ_i denoting the health state standard deviation obtained as a marginal from the state covariance matrix). When the above inequality is true we update the control allocation algorithm by setting $f_{\text{min}} = f_{\text{max}} = 0$ for the failed motor and enabling the bidirectional mode for the opposite.

VI. EXPERIMENTS

We showcase the capabilities of our algorithms in two different scenarios, namely response to step commands and autonomous detection and recovery after a motor failure. For the experiments presented we used a custom-built hexacopter using off-the-shelf components. It consists of a 550 mm wide frame, a Pixhawk flight controller flashed with a modified version of the PX4 firmware and an Intel NUC-7567U onboard computer. We used 960KV motors coupled with the carbon reinforced Aeronaut CAMcarbon 9.5×4.5 propellers and the DYS ARIA bidirectional capable ESCs. A Vicon motion capture system was responsible for providing external position and orientation measurements while all the other components run onboard the MAV.

A. Aggressive step commands

In order to verify the tracking capabilities of the designed NMPC, Figure 6 shows the MAV response for a 2 m step in x and z and a 180^o step in yaw. The NMPC generated dynamically feasible trajectories which can steer the MAV in any orientation and achieve large linear accelerations (given the physical limitations of the platform) without overshooting. We conducted the same experiment twice using low and high orientation gains and observed that, in the latter case, the NMPC reduces the yaw error faster by simultaneously performing a half flip in roll and pitch.

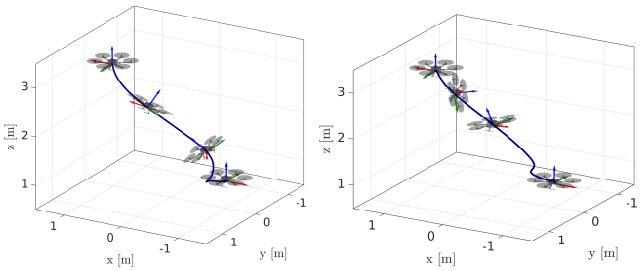

Fig. 6. Our MAV executing a 2 m step in x and z and a 180^o step in yaw with low (left) and high (right) orientation gains. Peak acceleration exceeds 15m/s^2 while peak orientation exceeds 120^o in pitch.

B. Fault detection and recovery

We tested the failure detection and autonomous recovery in two different scenarios where one motor was switched off (i) while hovering and another (ii) while the MAV was following setpoint commands. The results regarding the position and yaw tracking along with the online estimated health status of each motor, are shown in Figures 7 and 8. The injected motor failure was correctly identified with a maximum delay of 0.18 s. In both scenarios, the MAV was able to recover with a maximum height loss of 0.6 m. Position and yaw references were still tractable however the 5-motor asymmetric configuration resulted in slower tracking response. Regarding the health status variables of the functioning motors, these always remain close to 1. It can be seen that, in the setpoint experiment, there exist some short-in-duration deviations from 1. These spikes correspond to time instants when large angular accelerations were executed. We consider the main reason for this behaviour to be the mismatch between the EKF prediction model (which does not take into account less significant phenomena, such as gyroscopic moments) and the real one. In any case the estimated upper bound was always greater than 0.8 and thus unable to trigger a false positive.

VII. CONCLUSION AND FUTURE WORK

This paper presented a series of algorithms that can be used for aggressive and fault tolerant multicopter navigation. We experimentally verified their performance using a hexacopter although the same approach can be implemented on any other multirotor with minor modifications. Control performance can be further improved by using a more accurate system model, as the current one does not take into account effects such as the motor dynamics, rotor drag and gyroscopic moments. The fault detection EKF can seamlessly be implemented as an algorithmic update on any MAV as it only requires inertial measurements. It can be accordingly extended with speed or current measurements in order to prevent false positives due to large external disturbances. The disadvantage of our approach is that it requires carefully identifying physical parameters such as the motor coefficients and the inertia tensor. By online estimating these as shown in [24], we can make the controller adaptive to model changes and eliminate the need for tedious accurate offline identification.

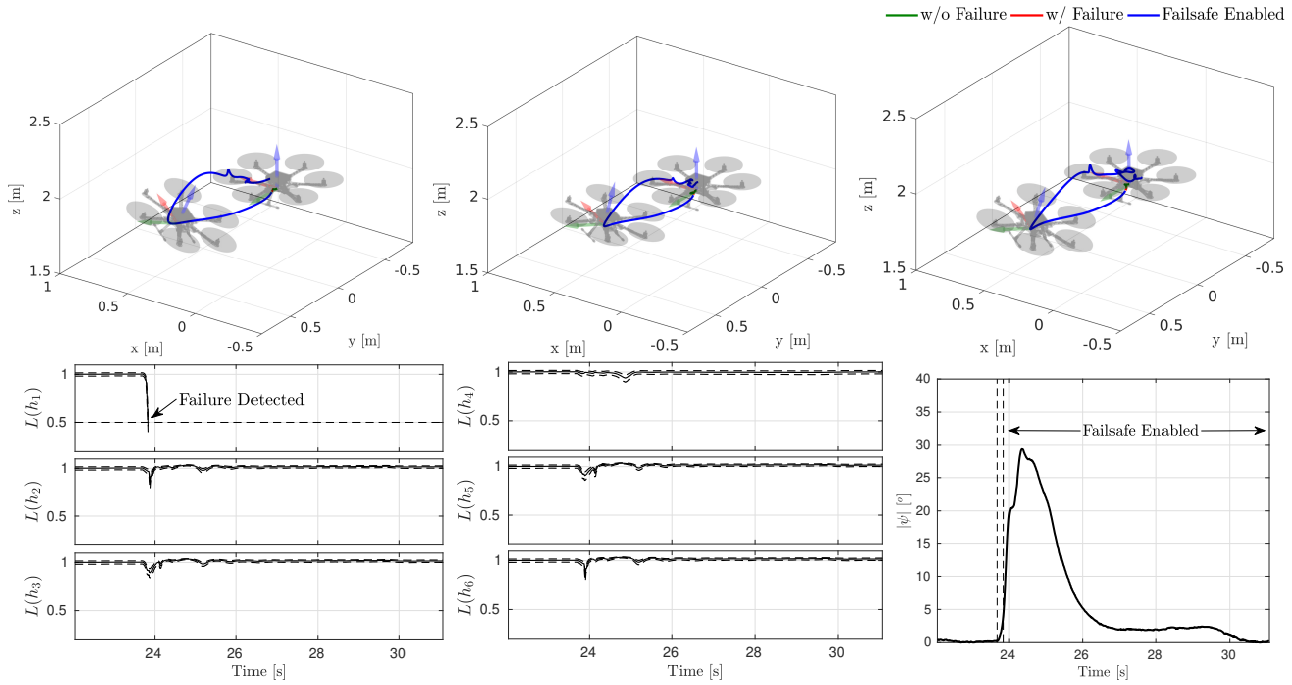

Fig. 7. Top row: Three different experiments with autonomous fault identification and recovery during hover. In all the experiments the failure was identified and the fail-safe was triggered within 0.18 s after the manual deactivation of Motor 1. The MAV was able to recover with a maximum height loss of 0.40 m. Bottom row: The online estimates of the health status $L(h_i)$ and their corresponding 3σ confidence bounds and the absolute yaw error for the first experiment. Notice how the upper bound estimate $L(h_1 + 3\sigma_1)$ for Motor 1 drops below the 0.5 threshold after the motor deactivation at t = 23.68s. Once the fail-safe is triggered at t = 23.85s, control of yaw (bottom right) is maintained and the error converges to zero.

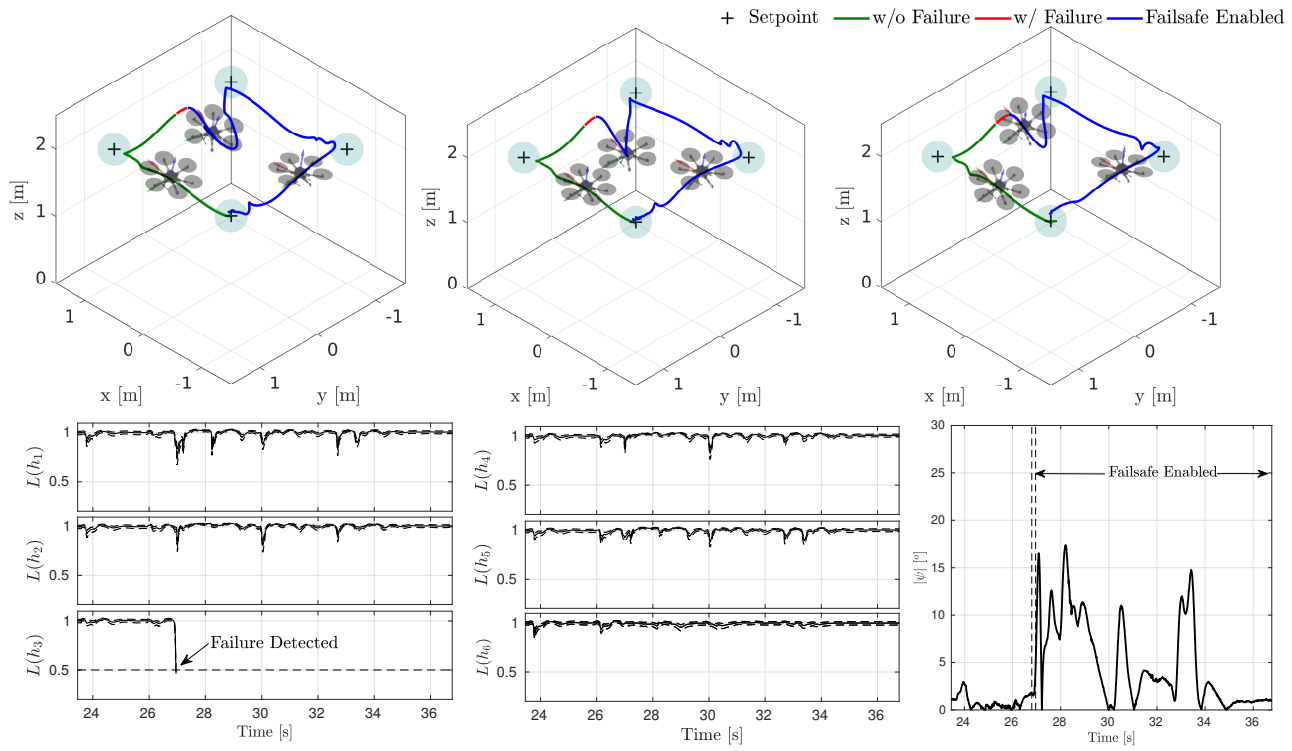

Fig. 8. Top row: Three different experiments with autonomous fault identification and recovery while following setpoint commands. In all the experiments the failure was identified and the fail-safe was triggered within 0.18 s after the manual deactivation of Motor 3. The MAV was able to recover with a maximum height loss of 0.60 m. Bottom row: The online estimates of the health status $L(h_i)$ and their corresponding 3σ confidence bounds and the absolute yaw error for the first experiment. The upper bound estimate $L(h_1 + 3\sigma_1)$ for Motor 3 drops below the 0.5 threshold after the motor deactivation at t = 26.80 s. Once the fail-safe is triggered at t = 26.96 s, control of yaw (bottom right) is maintained and the error converges to zero.

REFERENCES

[1] B. Houska, H. Ferreau, and M. Diehl, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[2] M. Giftthaler, M. Neunert, M. Stäuble, and J. Buchli, "The Control Toolbox - an open-source C++ library for robotics, optimal and model predictive control," in *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, May 2018, pp. 123–129.

[3] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[4] D. Tzoumanikas, W. Li, M. Grimm, K. Zhang, M. Kovac, and S. Leutenegger, "Fully autonomous micro air vehicle flight and landing on a moving target using visual-inertial estimation and model-predictive control," *Journal of Field Robotics*, vol. 36, no. 1, pp. 49–77, 2019.

[5] I. Sa, M. Kamel, R. Khanna, M. Popović, J. Nieto, and R. Siegwart, "Dynamic system identification, and control for a cost-effective and open-source multi-rotor mav," in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds. Cham: Springer International Publishing, 2018, pp. 605–620.

[6] G. Darivianakis, K. Alexis, M. Burri, and R. Siegwart, "Hybrid predictive control for aerial robotic physical interaction towards inspection operations," in *IEEE International Conference on Robotics and Automation*, May 2014, pp. 53–58.

[7] C. Papachristos, K. Alexis, and A. Tzes, "Dual–authority thrust–vectoring of a tri–tiltrotor employing model predictive control," *Journal of Intelligent & Robotic Systems*, vol. 81, no. 3, pp. 471–504, Mar 2016.

[8] M. Kamel, M. Burri, and R. Siegwart, "Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463 – 3469, 2017, 20th IFAC World Congress.

[9] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1–8.

[10] P. Foehn and D. Scaramuzza, "Onboard State Dependent LQR for Agile Quadrotors," in *IEEE International Conference on Robotics and Automation*, May 2018, pp. 6566–6572.

[11] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart, "Fast nonlinear model predictive control for multicopter attitude tracking on SO(3)," in *IEEE Conference on Control Applications*, Sep. 2015, pp. 1160–1166.

[12] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *IEEE International Conference on Robotics and Automation*, May 2016, pp. 1398–1404.

[13] C. de Crousaz, F. Farshidian, M. Neunert, and J. Buchli, "Unified motion control for dynamic quadrotor maneuvers demonstrated on slung load and rotor failure tasks," in *IEEE International Conference on Robotics and Automation*, May 2015, pp. 2223–2229.

[14] M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Inversion based direct position control and trajectory following for micro aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 2933–2939.

[15] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *IEEE Conference on Decision and Control*, Dec 2010, pp. 5420–5425.

[16] M. Faessler, D. Falanga, and D. Scaramuzza, "Thrust Mixing, Saturation, and Body-Rate Control for Accurate Aggressive Quadrotor Flight," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 476–482, April 2017.

[17] D. Brescianini and R. D'Andrea, "Tilt-prioritized quadrocopter attitude control," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2018.

[18] D. Brescianini and R. DAndrea, "An omni-directional multirotor vehicle," *Mechatronics*, vol. 55, pp. 76 – 93, 2018.

[19] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *IEEE International Conference on Robotics and Automation*, May 2014, pp. 45–52.

[20] T. Schneider, G. Ducard, R. Konrad, and S. Pascal, "Fault-tolerant Control Allocation for Multirotor Helicopters Using Parametric Programming," in *International Micro Air Vehicle Conference and Flight Competition*, Braunschweig, Germany, Jul. 2012.

[21] M. Saied, B. Lussier, I. Fantoni, C. Francis, H. Shraim, and G. Sanahuja, "Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor," in *IEEE International Conference on Robotics and Automation*, May 2015, pp. 5266–5271.

[22] M. Saied, B. Lussier, I. Fantoni, H. Shraim, and C. Francis, "Fault diagnosis and fault-tolerant control of an octorotor uav using motors speeds measurements," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5263 – 5268, 2017, 20th IFAC World Congress.

[23] M. Giftthaler, M. Neunert, M. Stäuble, J. Buchli, and M. Diehl, "A Family of Iterative Gauss-Newton Shooting Methods for Nonlinear Optimal Control," *CoRR*, vol. abs/1711.11006, 2017.

[24] M. Burri, M. Bloesch, Z. Taylor, R. Siegwart, and J. Nieto, "A framework for maximum likelihood parameter identification applied on MAVs," *Journal of Field Robotics*, vol. 35, no. 1, pp. 5–22, 2018.