2015 IEEE Conference on Control Applications (CCA)
Part of 2015 IEEE Multi-Conference on Systems and Control
September 21-23, 2015. Sydney, Australia

Fast Nonlinear Model Predictive Control for Multicopter Attitude Tracking on SO(3)

Mina Kamel, Kostas Alexis, Markus Achtelik and Roland Siegwart

Abstract— **Exploiting the highly dynamic flight envelope of a multirotor Micro Aerial Vehicle (MAV) is a particularly challenging task that requires special treatment of its attitude control loop. In this paper, we propose a fast nonlinear model predictive control approach based on a geometric formulation of the error to track the MAV attitude on the SO(3) special orthogonal group. The proposed controller is combined with an optimal position tracking control strategy in a cascaded fashion. The overall framework is implemented on–board a hexacopter and verified in both high fidelity simulations as well as extensive experimental studies. As shown, the resulting closed–loop system exploits the dynamics of the platform, is able to track aggressive trajectories, recover from arbitrary attitude configurations and also handle a propeller failure.**

I. INTRODUCTION

The field of aerial robotics has recently seen rapid growth. Prerequisite technologies have developed to the point that we are not far from the day when utilization of aerial robots is prevalent in our society. With an application range that includes infrastructure inspection [1–3], , mapping [4], disaster relief [5], crop monitoring [6] and more, unmanned aerial vehicles (UAVs) already provide added value to several critical and financially significant applications, and are widely acknowledged for their potential to achieve a large impact in terms of development and growth. Examples of compelling existing use cases include the mapping of the Colorado flood area in 2003, the 3D reconstruction of the "Christ the Redeemer" statue in Brazil, the Matterhorn mountain reconstruction and the live offshore flare inspection that took place in the North Sea [7].

Despite these notable achievements and the outstanding efforts of the community, aerial robotics have still not yet shown their full potential. Widely used systems like multicopters present very agile dynamics, yet the majority of the on–board controllers only employs a rather limited part of the flight envelope. The typical response is that of a hovering and rather slowly maneuvering platform and not that of a system that fully exploits its dynamic envelope. In addition, despite the fact that such aerial vehicles often implement mechanical redundancy that could ensure safe flight in case of an actuator failure, the vast majority of the control loops are not designed to account for these scenarios. Few exceptions such as the work in [8] indicate the underlying potential in such aerial robotic configurations.

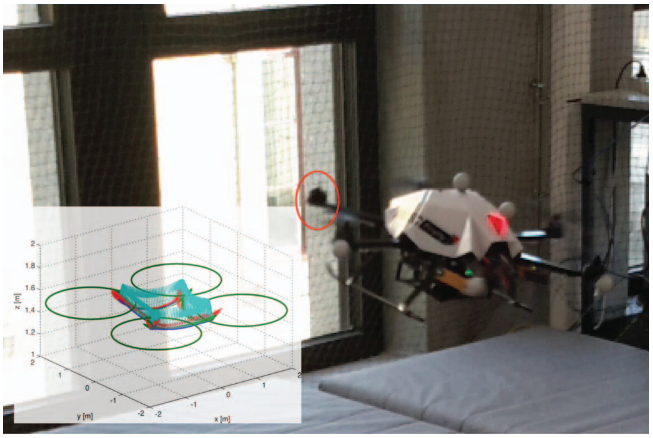


Fig. 1: Instant of the experimental evaluation studies of the proposed nonlinear model predictive controller. During this experiment, the controller ensures agile position tracking despite the fact that one propeller is intentionally removed from the AscTec Firefly micro aerial vehicle.

In [9–11] almost global asymptotically stable multicopter attitude controllers based on unit quaternion are presented. While in [12, 13] attitude controllers based on rotation matrix representation are proposed. However, non of these controllers takes into account the vehice's physical limitations, neither the problem of control allocation is considered in details.

This work proposes a new control framework for the attitude control of multirotor Micro Aerial Vehicles (MAVs) that aims to optimally exploit the full flight envelope and control allocation properties of the system, present superior disturbance rejection characteristics and for the case of vehicle configurations that present levels of actuator redundancy ensure stable flight even subject to propeller loss. To achieve this goal we tackle the problem of attitude control from a geometric point of view in a Nonlinear Model Predictive Control (NMPC) framework. The controller operates on the SO(3) Special Orthogonal group and the optimization problem aims to compute the best possible control action that provides high performance tracking characteristics by exploiting the nonlinear dynamics of the system. The control problem is solved in a receding horizon fashion and therefore it also respects the system's physical limitations as modeled by state and input constraints. Encoding and accounting for all possible control allocation options that the six propellers of the system provide, the proposed NMPC strategy ensures stable flight even during the loss of one of the propellers. The controller was evaluated regarding its

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No.644128, AEROWORKS.

All authors are with the Autonomous Systems Lab at ETH Zurich, Leonhardstrasse 21, 8092, Zurich, Switzerland. mina.kamel@mavt.ethz.ch

978-1-4799-7787-1/15/$31.00 ©2015 IEEE

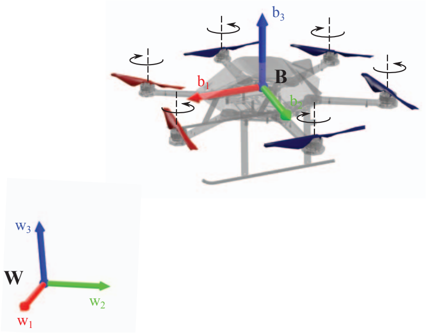


Fig. 2: Illustration of the employed Firefly hexacopter from Ascending Technologies with the attached BFF $\mathbb{B}$ and the inertial frame $\mathbb{W}$

performance and the aforementioned properties using both simulation and extensive experimental studies and Figure 1 depicts an instant of these studies. It was implemented on–board an AscTec Firefly hexacopter using state–of–the–art nonlinear optimization techniques. A video with example responses during propeller loss may be found at: `https://youtu.be/cocvUrPfyfo`.

The remainder of this paper is organized as follows: In Section II the model of the considered multicopter is presented followed by the detailed description of the NMPC strategy in Section III. Subsequently, an overview of the implementation methods is provided in Section IV. Extensive simulation and experimental studies are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Hexacopter Model

In this section we present the hexacopter model used in the controller formulation. This multirotor MAV is considered as a single 6-Degrees of Freedom (DoF) rigid body with Body–Fixed Frame (BFF) $\mathbb{B}$ attached as shown in Figure 2. The vehicle configuration is described by the position of the center of mass $\mathbf{p} \in \mathbb{R}^3$ in the inertial frame $\mathbb{W}$ and the orientation of the BFF with respect to the inertial frame. The orientation of the vehicle is represented by a rotation matrix $\mathbf{R} \in SO(3) = \left\{\mathbb{R}^{3\times3} | \mathbf{R}^T\mathbf{R} = \mathbf{I}, det\left(\mathbf{R}\right) = 1\right\}$ expressed on the Special Orthogonal manifold. In the following, we will denote the columns of the matrix $\mathbf{R}$ with the vectors $\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}$, i.e. :

$$\mathbf{R} = \left[\begin{array}{ccc} \mathbf{b_1} & \mathbf{b_2} & \mathbf{b_3} \end{array}\right]$$

In our model we assume that the axis of each propeller is parallel to the third axis of the BFF $\mathbf{b_3}$. Each propeller generates thrust along its axis that is proportional to the square of the propeller rotation speed. Furthermore, as the motor dynamics are considerably faster compared to those of the vehicle body, they are neglected.

The generated thrust and moment from the $i-th$ propeller is given by:

$$f_i = k_n n_i^2, \tag{1a}$$

$$M_i = (-1)^{i-1} k_m f_i, \tag{1b}$$

where f_i is the generated thrust by the $i-th$ propeller, k_n is a positive constant, n_i the propeller rotation speed, M_i the generated moment and k_m is a positive constant. Note that the sign of each moment M_i depends on the rotation direction of the $i-th$ propeller.

Let $\mathbf{v}$ be the velocity of the center of mass expressed in the inertial frame $\mathbb{W}$, g the acceleration of gravity, T the total thrust generated by the propellers, m the total mass, ω the angular velocity of the vehicle expressed in the BFF and $\mathbf{J}$ the inertia matrix of the vehicle with respect to the BFF. The system dynamics can be described by the following equations [14, 15]:

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{2a}$$

$$\dot{\mathbf{v}} = \mathbf{R}\begin{bmatrix} 0 \\ 0 \\ T/m \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \tag{2b}$$

$$\dot{\mathbf{R}} = \mathbf{R}\lfloor\boldsymbol{\omega}\times\rfloor, \tag{2c}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathcal{A}\begin{bmatrix} f_1 \\ \vdots \\ f_6 \end{bmatrix}, \tag{2d}$$

where the operator $\lfloor\times\rfloor$ conducts to the skew symmetric calculation, $\mathcal{A}$ is an appropriate matrix that maps the propellers thrust into moments around each axis of the body frame. For the considered hexacopter configuration, this matrix takes the form [16]:

$$\mathcal{A} = \begin{bmatrix} ls_{30} & l & ls_{30} & -ls_{30} & -l & ls_{30} \\ -lc_{60} & 0 & lc_{60} & lc_{60} & 0 & -lc_{60} \\ -k_m & k_m & -k_m & k_m & -k_m & k_m \end{bmatrix} \tag{3}$$

where l is the vehicle's arm length, s_x stands for $\sin x$ and c_x for $\cos x$, while x is expressed in degrees.

III. Controller Formulation

In this section we present a two-loop controller strategy with the outer loop aimed to control the vehicle position and the inner loop based on NMPC aimed to control the vehicle attitude on the SO(3). In Subsection III-A, we present the controller structure. Within Subsection III-B, we discuss the outer loop trajectory tracking controller in detail. Finally, in Subsection III-C we present an attitude controller based on NMPC which corresponds to the main contribution of this work.

A. Controller Structure

A block diagram representation of the hierarchical controller proposed in this paper is shown in Figure 3. An LQR controller with integral action is employed in the outer loop to track a trajectory defined by the desired position $\mathbf{p_d(t)}$, velocity $\mathbf{v_d(t)}$ and acceleration $\mathbf{a_d(t)}$. The outer loop controller will generate a desired thrust vector $\mathbf{T_d}$. To fully define the attitude of the vehicle, beside the thrust vector, the desired heading angle of the vehicle ψ_d is required. The inner attitude controller loop is based on a NMPC that computes the best possible individual motor

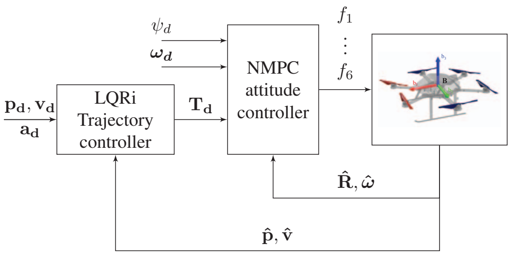

ψ_d
$\boldsymbol{\omega}_\mathbf{d}$
$\mathbf{p_d}, \mathbf{v_d}$
$\mathbf{a_d}$
LQRi Trajectory controller
$\mathbf{T_d}$
NMPC attitude controller
f_1
$\vdots$
f_6
$\hat{\mathbf{R}}, \hat{\boldsymbol{\omega}}$
$\hat{\mathbf{p}}, \hat{\mathbf{v}}$

Fig. 3: Controller structure

thrust to track the desired attitude respecting the physical constraints of the vehicle.

The estimated state of the vehicle provided to the outer loop consists of the position $\hat{\mathbf{p}}$ and velocity $\hat{\mathbf{v}}$, while the inner loop takes the estimated attitude in rotation matrix representation $\hat{\mathbf{R}}$ and the angular velocity $\hat{\boldsymbol{\omega}}$. These states are estimated using a state-of-the-art multi sensor fusion based on an Extended Kalman Filter as thoroughly described in our previous work in [17].

B. Outer Loop

For the outer loop design, the typically employed method of decoupling the three axes is considered. For simplicity we discuss here the controller for one axis. Subsequently, the linear 2^{nd} order system used to compute the LQR gain for a single axis is given by:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot u, \tag{4}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}, u = T \tag{5}$$

Using standard LQR techniques we obtain the following control action:

$$u = -\mathbf{K}_{lqr} \left(\mathbf{x}_{ref} - \hat{\mathbf{x}} \right) + \mathbf{K}_i \int_0^t \left(\mathbf{x}_{ref} - \hat{\mathbf{x}} \right) dt + u_{ff}, \tag{6}$$

where $\mathbf{K}_{lqr}$ is the LQR gain, $\mathbf{K}_i$ is the integral action constant and u_{ff} is a feed forward term from the trajectory generation which is the desired acceleration (e.g. as in [16]). The feed forward term improves the trajectory tracking when aggressive maneuvers are required.

C. Inner Loop

The inner loop consists of a NMPC for the vehicle attitude control. Following a similar approach as in [15] we construct the desired rotation matrix $\mathbf{R}_d$ as described below. Firstly, the desired third body frame axis is given by:

$$\mathbf{b}_{3d} = \frac{\mathbf{T}_d}{\|\mathbf{T}_d\|}, \ \mathbf{T}_d \neq \mathbf{0}.$$

The first body frame is defined initially in the xy plane, then it is projected on the plane perpendicular to $\mathbf{b}_{3d}$. Hence $\mathbf{b}_{1d}$ is given by:

$$\mathbf{b}_{1d} = \begin{bmatrix} \cos\psi_d & \sin\psi_d & 0 \end{bmatrix}^T.$$

The second body frame is perpendicular to the plane that contains $\mathbf{b}_{1d}, \mathbf{b}_{3d}$, hence

$$\mathbf{b}_{2d} = \frac{\mathbf{b}_{3d} \times \mathbf{b}_{1d}}{\|\mathbf{b}_{3d} \times \mathbf{b}_{1d}\|}.$$

Finally, the desired rotation matrix $\mathbf{R}_d$ is given by:

$$\mathbf{R}_d = \begin{bmatrix} \mathbf{b}_{2d} \times \mathbf{b}_{3d} & \mathbf{b}_{2d} & \mathbf{b}_{3d} \end{bmatrix}.$$

Note that this construction of $\mathbf{R}_d$ assumes that $\mathbf{b}_{3d}$ and $\mathbf{b}_{1d}$ are not parallel.

1) Error on SO(3): The attitude and angular velocity evolve on the tangent bundle of the SO(3) manifold, thus the tracking error should be carefully chosen. Given a tracking command $\mathbf{R}_d, \boldsymbol{\omega}_d$ and current attitude and angular velocity we define the attitude error as follows [15]:

$$\mathbf{e}_R = \frac{1}{2} \left(\mathbf{R}_d^T \mathbf{R} - \mathbf{R}^T \mathbf{R}_d \right)^\vee. \tag{7}$$

where the *vee* map $^\vee$: SO(3) $\rightarrow \mathbb{R}^3$ is the inverse of the hat map. Furthermore, the angular velocity error is defined as:

$$\mathbf{e}_\omega = \boldsymbol{\omega} - \mathbf{R}^T \mathbf{R}_d \boldsymbol{\omega}_d. \tag{8}$$

The attitude error function defined in (7) has the advantage of covering almost all SO(3), and it also avoids singularities associated with local coordinates of SO(3). However for all rotation matrices that can be expressed by $e^{\{\pi \lfloor \mathbf{v} \times \rfloor\}}$ for all $\mathbf{v} \in \left\{ \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix} \in \mathbb{R}^3 | v_1^2 + v_2^2 + v_3^2 = 1 \right\}$, the error function has a critical point, i.e. the error vector is zero. As the $2 - norm$ of the attitude error appears in the cost function, these critical points become attraction points of the system causing instability. We will show next how appropriate terminal constraint in the optimization problem will solve this problem.

2) System Constraints: Among the most important properties of predictive control is its inherent capacity to incorporate and account for the physical constraints of the system. The first obvious physical limitation is the limited thrust that each propeller can provide. This can be written as:

$$0 \leq f_i \leq f_{\max} \quad \text{for} \quad i = 1, \ldots, 6 \tag{9}$$

where $f_{\max} > 0$ is the maximum thrust values that can be generated by each propeller (the minimum thrust $f_{\min}$ is not set to zero due to a technical limitation of the employed electronic speed controller interfacing commands).

Apart from the input constraints, additional restrictions that have to be taken into account come at the form of state constraints due to sensor limitations. More specifically, the on–board gyro sensors get saturated above certain angular rates which subsequently does not allow the correct estimation of the motion of the vehicle. Therefore, angular rate constraints are imposed to constraint the dynamic flight envelope of the vehicle to the subset within which the sensor updates are valid:

$$\begin{aligned} -\omega_{1.\max} &\leq \omega_1 \leq \omega_{1,\max}, \\ -\omega_{2,\max} &\leq \omega_2 \leq \omega_{2,\max}, \\ -\omega_{3,\max} &\leq \omega_3 \leq \omega_{3,\max}, \end{aligned} \tag{10}$$

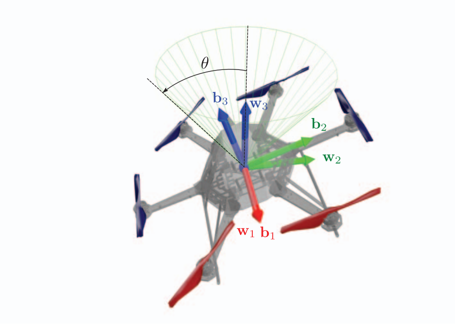


Fig. 4: Terminal constraints on body frame third axis.

where $\omega_{1,\max}, \omega_{2,\max}$ and $\omega_{3,\max}$ are the maximum angular rates around the $\mathbf{b}_1, \mathbf{b}_2$. and $\mathbf{b}_3$ axes respectively.

3) Terminal Constraint: Apart from the encoded state and input constraints, which encode physical limitations of the system, a terminal constraint that is required for stability of the vehicle is also introduced. The imposed terminal constraint requires the body third axis $\mathbf{b}_3$ to be inside a cone around the world third axis $\mathbf{w}_3$. A graphical illustration of this constraint is shown in Figure 4 where θ is the half angle of the cone. This constraint will avoid the critical points of the error function when the vehicle is for instance in an upside down configuration. Such a constraint can be represented as follows:

$$\mathbf{b}_3^T(t + T_h) \cdot \mathbf{w}_3 \geq \cos\theta \tag{11}$$

where t is the current time and T_h is the prediction horizon.

4) NMPC Formulation: For the NMPC optimization problem we define the following state vector:

$$\mathbf{X} = \left[\mathbf{b}_1^T, \mathbf{b}_2^T, \mathbf{b}_3^T, \boldsymbol{\omega}^T\right]^T, \tag{12}$$

and the following control input:

$$\mathbf{U} = [f_1, \ldots, f_6]^T. \tag{13}$$

The time evolution of the system state vector (12) is described by Equations (2c) and (2d) which can be described by a nonlinear function f as follows:

$$\dot{\mathbf{X}} = \mathfrak{f}(\mathbf{X}, \mathbf{U}). \tag{14}$$

The NMPC scheme repeatedly solves the following optimal control problem (OCP):

$$\begin{aligned} \min_{\mathbf{X},\mathbf{U}} & \int_t^{t+T_h} (h_R + h_\omega + h_T + h_U)\, dt \\ \text{s.t. } & \mathbf{X}(t) = \hat{\mathbf{X}}(t), \\ & \text{Equations (14) \& (9) \& (10)}, \\ & \text{Equation (11)} \end{aligned} \tag{15}$$

where

$$h_R = \mathbf{e}_R^T \mathbf{Q}_R \mathbf{e}_R$$

is the penalty on the attitude error,

$$h_\omega = \mathbf{e}_\omega^T \mathbf{Q}_\omega \mathbf{e}_\omega,$$

is the penalty on the angular velocity error,

$$h_T = Q_T \left(\sum_{i=1}^{6} f_i - \|\mathbf{T}_d\|_2\right)^2,$$

is the penalty on desired thrust magnitude and

$$h_U = (\mathbf{U} - \mathbf{U}_{ss})^T \mathbf{R}_U (\mathbf{U} - \mathbf{U}_{ss}),$$

is the penalty on the control action, with $\mathbf{U}_{ss}$ the steady state motors thrust given by

$$\mathbf{U}_{ss} = \frac{\|\mathbf{T}_d\|_2}{6} \mathbf{1}_{6\times 1}.$$

and $\mathbf{Q}_R \succeq \mathbf{0}$, $\mathbf{Q}_\omega \succeq \mathbf{0}$ are positive semi-definite matrices that represent the penalty matrices of the attitude error and angular velocity error respectively. $Q_T \geq 0$ is non-negative scalar that penalizes the difference between generated thrust and desired thrust and $\mathbf{R}_U \succ \mathbf{0}$ is positive definite matrix that penalizes the control action.

5) Propeller Failure: The presented controller can handle actuator failure such as a propeller loss or motor failure by exploiting the full control allocation capabilities of the platform. From the geometric structure of the hexacopter, it is intuitively understood that if one propeller is lost, it makes sense to reduce the thrust produced by the opposite propeller and fly with four propellers. However, in this case the propeller configuration is not symmetric anymore, and authority on heading angle is lost to recover altitude and position stability and tracking properties.

Without loss of generality, let's assume that a failure occurred in the first propeller. When the failure is detected, we change the steady state control action as follows:

$$\mathbf{U}_{ss} = \frac{\|\mathbf{T}_d\|_2}{4} \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}^T.$$

Beside setting the damaged propeller and its opposite thrust to zero, it is necessary to fly with free heading for the aformentioned reason. This can be achieved simply by setting desired heading to current vehicle heading.

IV. Real-time Implementation

Computing solutions of (12) repeatedly over time is a challenging task. A particular family of techniques to solve the aforementioned optimization problem that gained attention recently are the so–called *direct methods* [18], which is nowadays the most widely used technique to address optimal control problems.

In this section we discuss real time implementation of the NMPC controller using the ACADO toolkit [19].

A. Optimal Control Problem Solution

A direct multiple shooting technique is used to solve the optimal control problem (15). In this approach the system dynamics are discretized over a time grid $t_0, \ldots, t_N$ within the time intervals $[t_k, t_{k+1}]$. The inequality constraints and control action are discretized over the same time grid. A boundary value problem (BVP) is solved for each interval and additional continuity constraints are imposed. Due to the

nature of the system dynamics and the imposed constraints, the optimization problem becomes a nonlinear program (NLP). This NLP is solved using Sequential Quadratic Programming (SQP) technique where the Quadratic Programs (QP) are solved by active set method using the qpOASES solver [20].
Note that, in case of infeasibility of the underlying QP, $\ell1$ penalized slack variables are introduced to relax all constraints.
The controller is implemented in a receding horizon fashion where only the first computed control action is applied to the system, and the rest of the predicted state and control trajectory is used as initial guess for the OCP to solve in the next iteration.

B. Attitude Dynamics Integration

A problem with the discrete integration of the attitude dynamics is that classic integration techniques don't guarantee that the predicted $\mathbf{R}_{k+1}$ is on SO(3) if $\mathbf{R}_k$ is on SO(3). Our approach to tackle this problem is to adopt a geometric discrete integration technique that preserves the SO(3) group structure. In particular, the Crouch-Grossman method is used to integrate the attitude dynamics. This method uses the exponential map which is computationally expensive, thus an approximation up to finite terms is used. More details about this method can be found in [21] and [22].

C. Real-time Iteration

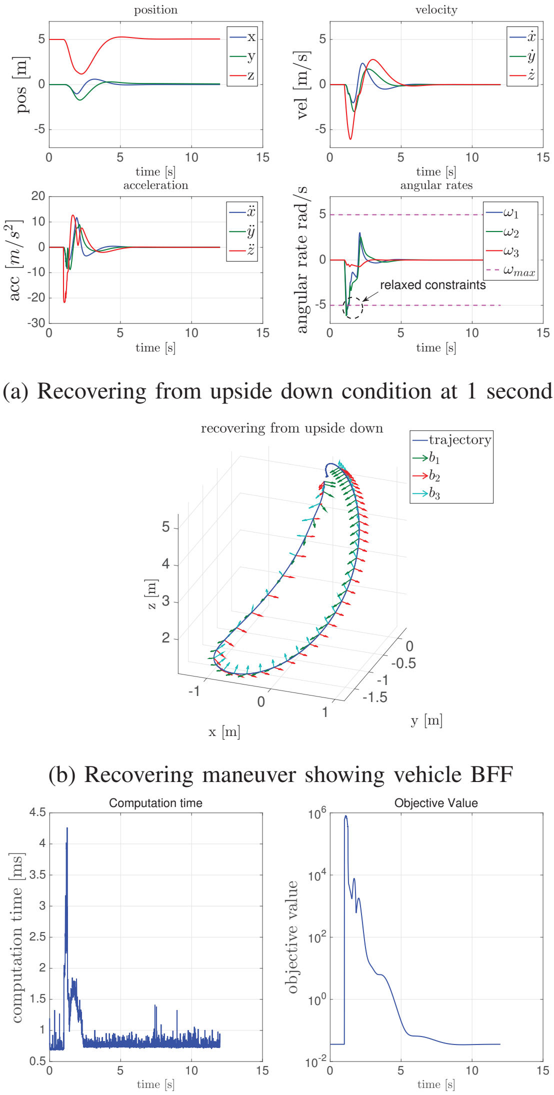
A *real time iteration* [23] scheme is used based on Gauss-Newton iteration. This scheme approximates the on-line solution of the optimization problem and iteratively improves the solution during the runtime of the process, improving the speed of execution of the optimization algorithm.

V. Evaluation Studies

In this section we show simulation and experimental results that prove the efficiency of the proposed controller. In subsection V-A we show simulation results where the ability to recover from extreme condition is tested. Furthermore, results of trajectory tracking performance are also presented. Subsequently, within subsection V-B we present experimental results of waypoint navigation using the AscTec Firefly MAV and we also demonstrate the NMPC capacity to exploit the control allocation capabilities of this platform by showing position tracking subject to the loss of one propeller.

For all the results presented below, the controller is running at $200\,\mathrm{Hz}$ with a prediction horizon set to $T_h = 50\,\mathrm{ms}$. The selection of such a short–time prediction horizon is driven by two facts: the first reason is to reduce the optimization complexity to achieve high execution rate, while the second reason is related with the uncertainty in the model equations such as parametric inaccuracies e.g. not perfect knowledge of the inertia tensor (possibly due to varying payloads) or the motor constants and unmodeled aerodynamics effects.

The considered parameters of the AscTec Firefly hexacopter used in our simulation and experiments are: $\mathbf{J} = diag\{\ 0.0348\quad 0.0459\quad 0.0978\ \}\,\mathrm{kgm}^2$, $m = 1.54\,\mathrm{kg}$, $k_n = 6.7\cdot 10^{-6}\,\mathrm{N\,s/rad}$, $k_m = 0.0365\,\mathrm{m}$ and $f_{max} = 6.13\,\mathrm{N}$.



(a) Recovering from upside down condition at 1 second.

(b) Recovering maneuver showing vehicle BFF

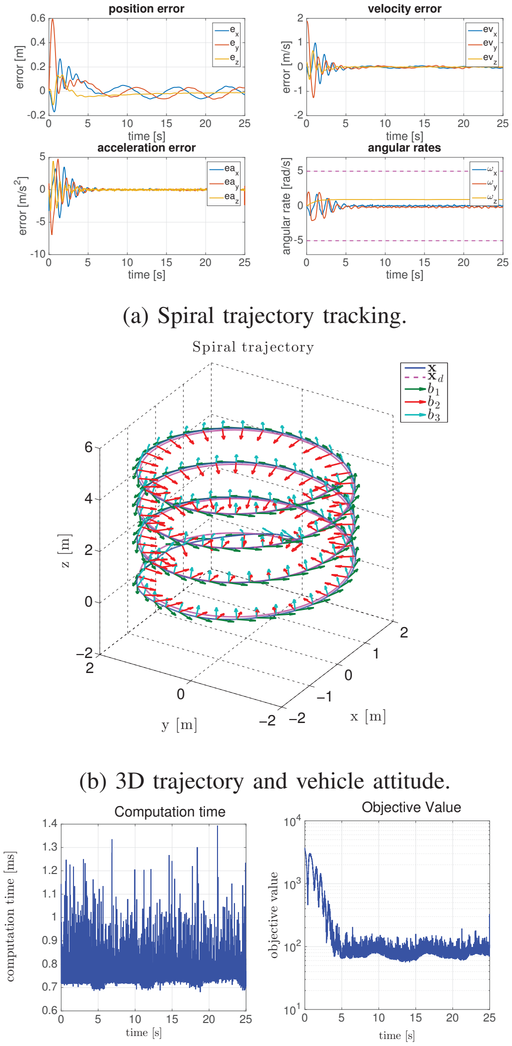
(c) Objective function value and computation time of NMPC attitude controller.

Fig. 5: Recovering from extreme condition simulation result

A. Simulation Results

We first show simulation results where the vehicle recovers from an attitude error of exactly 180°. In this situation the attitude error as expressed in equation (7) is zero as discussed in section III-C.1, yet the controller reacts and stabilizes the system thanks to the imposed terminal constraint (11). This indeed further motivates the utilization of control approaches capable of incorporating such constraints. Next, we present a spiral trajectory tracking example while keeping the vehicle heading aligned with the velocity vector.

1) Recovering from Upside Down: In this simulation, the vehicle is hovering at $5\,\mathrm{m}$ and a strong disturbance



(a) Spiral trajectory tracking.

(b) 3D trajectory and vehicle attitude.

(c) Objective function value and computation time of NMPC attitude controller.

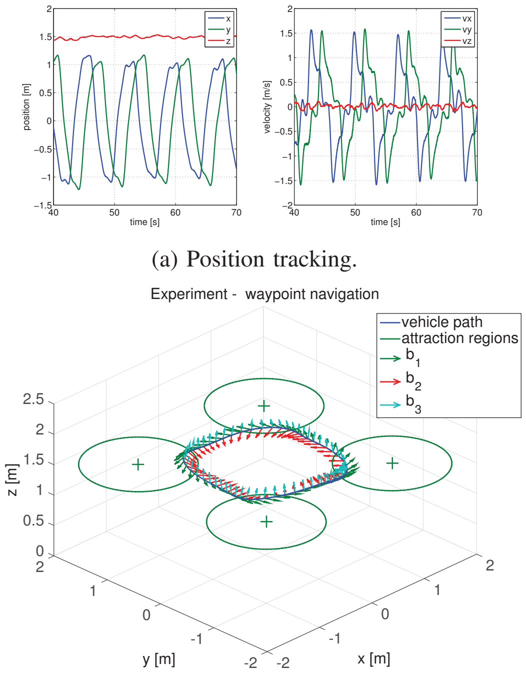
Fig. 6: Spiral trajectory tracking simulation result

disturbance is introduced at time $1\,\mathrm{s}$, causing the vehicle to go upside down. Figure 6a depicts that the vehicle loses approximately $1.8\,\mathrm{m}$ at altitude before it completely recovers its attitude configuration. A solver infeasibility occurs when the system is largely disturbed, however constrains are automatically relaxed using slack variables as discussed in Section IV.In Figure 6b we show the trajectory of the multicopter during recovery maneuver. In Figure 6c we show that the computation time on an intel 2.8 GHz Intel Core i7 is below $1\,\mathrm{ms}$ except when the vehicle is trying to recover from extreme condition where the computation time goes to $4.3\,\mathrm{ms}$.

2) Spiral Trajectory Tracking: In this simulation, the trajectory tracking capabilities of the proposed controller are evaluated by commanding it to track a challenging spiral trajectory defined by the following expression:

$$\mathbf{p}(t) = \begin{bmatrix} 2\cos 0.943t & 2\sin 0.943t & 0.2t \end{bmatrix}$$

and its derivatives. The results of the simulation are illustrated in Figure 6. As shown, high performance and accuracy is achieved.



(a) Position tracking.

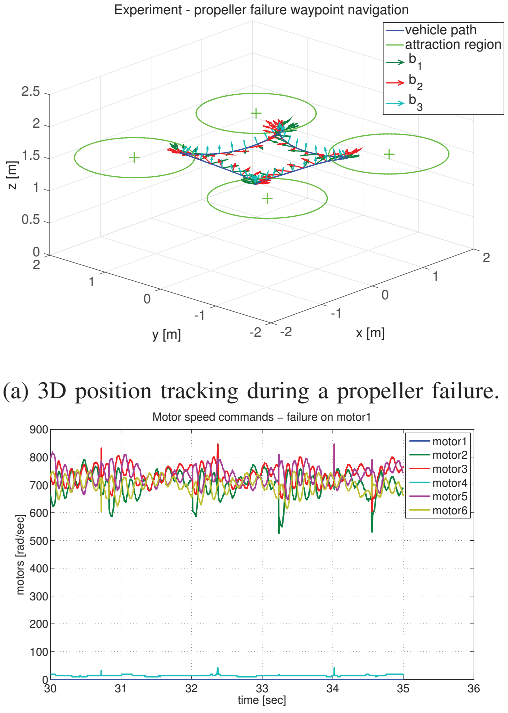
(b) 3D position tracking and vehicle attitude.

Fig. 7: Position tracking experimental results

B. Experimental Results

In this subsection we present the results of two experimental studies conducted using an AscTec Firefly MAV. The first experiment shows position tracking of four (4) points that form a square. An acceptance radius is introduced around each point such that when the vehicle is inside the corresponding sphere, the next point is passed to the controller as position reference. The second experiment is aimed to demonstrate that the controller ability to handle propeller failure. One propeller is intentionally removed and the same trajectory of the first experiment is repeated.

The Firefly hexacopter is equipped with a *MasterMind* embedded 1.86 GHz Core2Duo processor on-board computer from Ascending Technologies. The controller is implemented to run on-board this computer and direct motors commands are set to the motors through the Firefly low level electronics. The on-board computer is running the Robot Operating System (ROS) [24], and our controller is integrated into that. The controller computes individual propeller thrust values $f_i,\ i = 1\ldots 6$ by inverting (1a) we obtain each individual motor speed in rad/s which is mapped to proper motor command as described in [16]. All the aforementioned experiments were performed at the facilities of the Autonomous Systems Lab at ETH Zurich in a flying space equipped with a Vicon motion capture system, which is used as pose measurement in our state estimation framework.



(a) 3D position tracking during a propeller failure.

(b) Motors command during a propeller failure.

Fig. 8: Position tracking during a propeller failure experimental results

Figure 7 shows the position tracking results. The heading reference is aligned with the velocity vector as shown in Figure 7b. Overall high maneuverability and tracking performance is achieved.

Figure 8 shows the same experiment with an intentionally removed propeller. As shown, the tracking capabilities of the closed-loop system are retained therefore allowing safe navigation despite the propeller loss. During this experiment the MAV flies with free heading by setting heading reference ψ_d to current heading. A video containing such experimental responses is available at `https://youtu.be/cocvUrPfyfo`.

VI. CONCLUSIONS

A nonlinear model predictive control strategy based on a geometric integration to tract the attitude of a multirotor MAV that was further combined with optimal position control was presented in this paper. The proposed approach aims to exploit the complete nonlinear flight envelope and the control allocation capabilities of the platform. It was implemented on–board a hexacopter and thoroughly evaluated in both simulations and experimental studies. As shown, highly agile maneuvering and accurate tracking is achieved while the system is capable of recovering from inverted attitude configurations and retain its position tracking capabilities even subject to the loss of one of its propellers.

REFERENCES

[1] N. Metni and T. Hamel, "A UAV for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, vol. 17, no. 1, pp. 3–10, November 2007.

[2] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2015, (accepted).

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

[4] P. Oettershagen, T. J. Stastny, T. A. Mantel, A. S. Melzer, K. Rudin, G. Agamennoni, K. Alexis, R. Siegwart, "Long-endurance sensing and mapping using a hand-launchable solar-powered uav," in *Field and Service Robotics, 10th Conference on*, June 2015, (accepted).

[5] P. Rudol and P. Doherty, "Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery," in *Aerospace Conference, 2008 IEEE*, 2008, pp. 1–8.

[6] E. R. Hunt, W. D. Hively, S. J. Fujikawa, D. S. Linden, C. S. T. Daughtry, and G. W. McCarty, "Acquisition of nir-green-blue digital photographs from unmanned aircraft for crop monitoring," *Remote Sensing*, vol. 2, no. 1, pp. 290–305, 2010.

[7] Cyberhawk, 2015, http://www.thecyberhawk.com/.

[8] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," ser. 2014 IEEE International Conference on Robotics and Automation (ICRA). Piscataway, N.J.: IEEE, 2014, pp. 45–52.

[9] H. M. Brescianini, Dario and R. D'Andrea, "Nonlinear Quadrocopter Attitude Control," idgenössische Technische Hochschule Zürich, Departement Maschinenbau und Verfahrenstechnik, Tech. Rep., 2013.

[10] J. Dvorak, M. De Lellis, and Z. Hurak, "Advanced control of quadrotor using eigenaxis rotation." in *CCA*, 2011, pp. 153–158.

[11] A. Tayebi, "Unit quaternion-based output feedback for the attitude tracking problem," *Automatic Control, IEEE Transactions on*, vol. 53, no. 6, pp. 1516–1520, 2008.

[12] N. Chaturvedi, A. K. Sanyal, N. H. McClamroch *et al.*, "Rigid-body attitude control," *Control Systems, IEEE*, vol. 31, no. 3, pp. 30–51, 2011.

[13] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.

[14] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*. IEEE, 2007, pp. 153–158.

[15] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, Dec 2010, pp. 5420–5425.

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

[17] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3923–3929.

[18] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems." PROCEEDINGS OF THE IFAC WORLD CONGRESS, 1984.

[19] B. Houska, H. Ferreau, M. Vukov, and R. Quirynen, "ACADO Toolkit User's Manual," http://www.acadotoolkit.org, 2009–2013.

[20] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[21] J. Park and W.-K. Chung, "Geometric integration on euclidean group with application to articulated multibody systems," *Robotics, IEEE Transactions on*, vol. 21, no. 5, pp. 850–863, Oct 2005.

[22] A. Marthinsen and B. Owren, "A note on the construction of crouch-grossman methods," 1998.

[23] B. Houska, H. Ferreau, and M. Diehl, "An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.

[24] "Robot operating system." [Online]. Available: http://www.ros.org