2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Trajectory Tracking Nonlinear Model Predictive Control for an Overactuated MAV

Maximilian Brunner, Karen Bodie, Mina Kamel, Michael Pantic, Weixuan Zhang, Juan Nieto, and Roland Siegwart

Abstract— **This work presents a method to control omnidirectional micro aerial vehicles (OMAVs) for the tracking of 6-DoF trajectories in free space. A rigid body model based approach is applied in a receding horizon fashion to generate optimal wrench commands that can be constrained to meet limits given by the mechanical design and actuators of the platform. Allocation of optimal actuator commands is performed in a separate step. A disturbance observer estimates forces and torques that may arise from unmodeled dynamics or external disturbances and fuses them into the optimization to achieve offset-free tracking. Experiments on a fully overactuated MAV show the tracking performance and compare it against a classical PD-based controller.**

I. Introduction

The usage of micro aerial vehicles (MAVs) has steadily increased over the previous years. Traditional multirotors have proven to be a good tool for tasks that do not require physical interaction, such as visual inspection or mapping of environments or buildings. In recent years, the new field of omnidirectional micro aerial vehicles (OMAVs) has emerged. OMAVs exhibit the advantage of being able to exert forces and torques in 6 degrees of freedom (DoF), which is the basis for physical interaction with the environment [1]. Possible interaction tasks include inspection of materials in buildings or other structures, remote spray painting, or remote controlling a floating actuator base at any desired position and orientation in space for tasks such maintenance or repairing [2], [3].

Different approaches for the design of OMAVs have been presented. They can generally be categorized by their degree of overactuation (*fully overactuated* vs. *slightly overactuated*) and their mechanical design (*fixed rotor* vs. *tilt rotor*).

We use the term *fully overactuated* for an OMAV that is able to generate a body *wrench*, i.e. forces and torques, in every direction while hovering in any desired attitude. *Slightly overactuated* OMAVs on the other hand are characterized by their ability to generate forces and torques in a limited range around some principal direction.

Fixed rotor OMAVs exhibit the advantage of being mechanically equally complex as underactuated platforms, while being able to generate a 6-DoF wrench. Omnidirectionality is achieved by having the rotors aligned non-parallel with respect to each other, allowing to generate thrust in more than just one principal direction. In [4] the design and control of a fully overactuated fixed rotor OMAV has been described and tested. Other designs have been evaluated in [5]–[7]. A control framework to tackle the problem of slight overactuation has been described in [8].

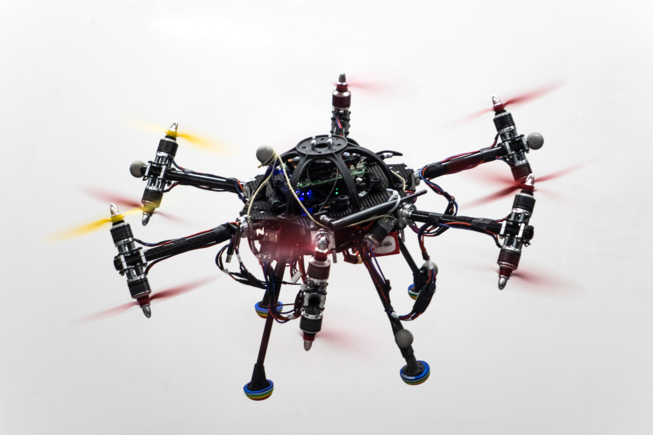

Fig. 1. An overactuated tiltrotor MAV in free flight.

While fixed rotor systems are mechanically not more complex than underactuated systems, they can often not generate a full 6-DoF force envelope or, if they can, are inefficient in doing so. This is because multiple rotors need to be directed in various body directions in order to be able to generate full omnidirectional actuation. This on the other hand leads to an increased weight and to only partial utilization of rotor thrusts.

This is why another hardware design approach has been introduced, using actuated tilt rotors. By rotation of rotor units around one more more additional axes, rotor thrusts can be directed into specific directions. This usually allows for full 6-DoF wrench generation. These platforms tend to be more efficient when compared to fixed rotor systems, as they are able to choose the thrust direction optimally, given a desired total wrench. Tilt rotor OMAV designs often use a traditional quadrotor or hexacopter design and add actuators to each rotor, e.g. in [9], [10], [11], or [12]. While they are able to exert a larger force/torque envelope in a more efficient manner than slightly overactuated platforms, they tend to exhibit more complex dynamics due to more moving parts or interfering airflows. In this work, we are particularly interested in dealing with the complexity that arises from additional actuators and system dynamics.

This research was funded by the NCCR Digital Fabrication through the Swiss National Science Foundation.

Authors are with the Autonomous Systems Lab, ETH Zürich, Leonhardstrasse 21, 8092 Zurich, Switzerland. Email: maximilian.brunner@mavt.ethz.ch

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

A. Related work

Both underactuated and overactuated MAVs are usually controlled by an input-output mapping approach that maps from a desired body wrench to optimal actuator commands. Based on the error between the system's state and a reference, a wrench vector is computed and optimal actuator commands are computed to achieve this wrench (see fig. 2). In the case of a fixed rotor system, this *allocation* of actuator commands computes the optimal rotor speeds. In the case of a tilt rotor system, the allocation optimizes for both rotor speeds as well as optimal rotor tilt angles.

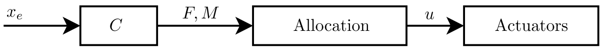

Fig. 2. Control of overactuated MAV.

An OMAV might also exhibit *actuation redundancy*, which results in a null space of multiple solutions to the allocation. In that case, the actuator commands are usually optimized for minimum rotor speeds in order to maximize efficiency of the platform. However, as tilt angle rates are inherently slow due to the relatively high inertias that they are driving and due to gyroscopic effects, it needs to be ensured that the allocation does not result in infeasibly fast changes of tilt angles.

In [9] a control approach has been presented that solves the allocation on a higher differential level, resulting in smooth trajectories for tilt angles and rotor speeds. By exploiting the null space of the allocation, it is then also possible to penalize tilt angle rates or rotor speeds outside of feasible ranges.

In our approach, we keep the generation of wrench commands and the allocation to optimal actuator commands separate. Given a pose tracking task, we design an optimal wrench trajectory that stays within the limits of system constraints. The allocation can then be performed instantaneously, without the need to consider constraints and without using the allocation's null space.

In order to cope with system constraints and with changes in the system model, we study the application of a model based controller, particularly a model predictive controller (MPC). That way we are able to choose constraints and design the shape of wrench trajectories according to the desired behaviour and capabilities of the platform.

MPC has previously been applied on underactuated MAVs, e.g. for fast trajectory optimization [13], [14] or for control under atmospheric disturbances [15]. However, these works design their controllers according to the natural underactuation of their systems, and their models can therefore not be applied on overactuated platforms. In contrast, we want to study the application of MPC on omnidirectional MAVs.

Since the performance of model based controllers depends on the quality of the model, we additionally employ a disturbance observer to estimate unmodeled forces and torques. These might arise from unknown system dynamics or external causes and they are fused into the model formulation.

TABLE I

SYMBOLS AND DEFINITIONS FOR A TILT-ROTOR AERIAL VEHICLE.

Symbol	Definition
$\mathscr{F}_W : \{O_W, x_W, y_W, z_W\}$	inertial frame: origin and primary axes
$\mathscr{F}_B : \{O_B, x_B, y_B, z_B\}$	body-fixed frame
$\mathscr{F}_r : \{O_r, x_r, y_r, z_r\}$	reference frame
m	mass
$\boldsymbol{J}$	inertia matrix
$\boldsymbol{f}$	force vector
$\boldsymbol{\tau}$	torque vector
${}_B\boldsymbol{w}$	total actuation wrench in body frame
${}_W\boldsymbol{g} = [0\ 0\ -g]^\top$	gravity vector, $g = 9.81\ \mathrm{m\,s^{-2}}$
${}_W\boldsymbol{p}$	position in the inertial frame
${}_W\boldsymbol{v}$	linear velocity
$\boldsymbol{q}$	attitude quaternion
${}_B\boldsymbol{\omega}$	angular velocity
$\Delta\boldsymbol{f}$, $\Delta\hat{\boldsymbol{f}}$	true and estimated force disturbance
$\Delta\boldsymbol{\tau}$, $\Delta\hat{\boldsymbol{\tau}}$	true and estimated torque disturbance
α_i	Tilt angle of rotor i
ω_i	rotor speed of rotor i

B. Contributions

We present an MPC for trajectory tracking of overactuated flying platforms that is able to

- adjust to the capabilities of the platform, specifically to constraints on force and torque and their rates, respectively
- include further knowledge about the model, such as specific model dynamics
- implicitly limit actuator control rates
- adapt to disturbances that result from unmodeled dynamics
- run in real-time on an onboard computer

Additionally, we provide experimental validation of the proposed method.

II. SYSTEM MODELING

A. Notation

In general, we denote a vector $\boldsymbol{v}$ expressed in a frame $\mathscr{F}_W$, as ${}_W\boldsymbol{v}$.

We use quaternions to describe the orientation of a rigid body. We write a quaternion $\boldsymbol{q}$ as a 4×1 column vector

$$\boldsymbol{q} = \begin{bmatrix} q_w \\ \boldsymbol{q}_v \end{bmatrix} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}, \quad \boldsymbol{q} \in \mathbb{H} \tag{1}$$

where q_w is referred to as the scalar part and $\boldsymbol{q}_v$ as the vector part [16]. We denote the inverse of a unit quaternion as

$$\boldsymbol{q}^{-1} = \begin{bmatrix} q_w \\ -\boldsymbol{q}_v \end{bmatrix} \tag{2}$$

The rotation of a vector $\boldsymbol{v}$ is described as a function of the rotation quaternion:

$${}_W\boldsymbol{v} = \boldsymbol{q} \otimes {}_B\boldsymbol{v} \otimes \boldsymbol{q}^{-1}, \tag{3}$$

where $\otimes$ denotes the quaternion product.

Table I gives an overview of symbols that are used throughout this paper.

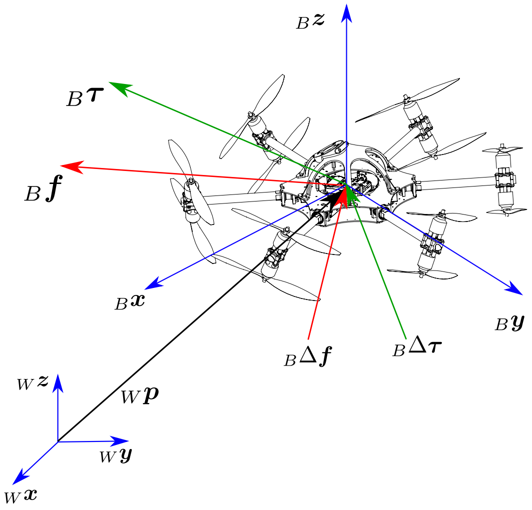

Fig. 3. Illustration of frames and control inputs.

B. System modeling of an OMAV

We assume that an OMAV can be modeled as a rigid body with mass m and inertia $\boldsymbol{J}$ that is affected by the following components:

- the sum of actuator forces and torques, expressed by ${}_B\boldsymbol{f}$ and ${}_B\boldsymbol{\tau}$,
- general force and torque disturbances, expressed by $\Delta\boldsymbol{f}$ and $\Delta\boldsymbol{\tau}$,
- gravity, expressed by $m_B\boldsymbol{g}$.

We describe the state of the system by its position ${}_W\boldsymbol{p} \in \mathbb{R}^3$ in the inertial frame $\mathscr{F}_W$, and its orientation. The orientation of the body fixed frame $\mathscr{F}_B$ w.r.t. $\mathscr{F}_W$ is represented by a quaternion $\boldsymbol{q}$. We can then use the Euler equations of rigid body motion to describe the system dynamics in the body fixed frame $\mathscr{F}_B$:

$$\begin{bmatrix} m\boldsymbol{I}_3 & 0 \\ 0 & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} {}_B\dot{\boldsymbol{v}} \\ {}_B\dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} {}_B\boldsymbol{f} \\ {}_B\boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} -m[\boldsymbol{\omega}]_\times & 0 \\ 0 & [\boldsymbol{J}\boldsymbol{\omega}]_\times \end{bmatrix} \begin{bmatrix} {}_B\boldsymbol{v} \\ {}_B\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} m_B\boldsymbol{g} \\ 0 \end{bmatrix} + \begin{bmatrix} {}_B\Delta\boldsymbol{f} \\ {}_B\Delta\boldsymbol{\tau} \end{bmatrix}, \quad (4)$$

where $[\cdot]_\times \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix of the vector $\cdot \in \mathbb{R}^3$.

The position and attitude dynamics can be written as follows:

$$\begin{aligned} {}_W\dot{\boldsymbol{p}} &= \boldsymbol{q} \otimes {}_B\boldsymbol{v} \otimes \boldsymbol{q}^{-1} \\ \dot{\boldsymbol{q}} &= \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ {}_B\boldsymbol{\omega} \end{bmatrix}. \end{aligned} \quad (5)$$

III. Control Design

The goal of the presented control framework is to track a given 6-DoF trajectory in $\mathbb{R}^3 \times \mathrm{SO}(3)$, while rejecting disturbances that can be caused by unmodeled system dynamics.

We assume that the OMAV can generate any desired wrench within certain known bounds, i.e. $\boldsymbol{w} \in \mathcal{W}$ and $\mathcal{W}$ is known. We further assume that the system is a tiltrotor OMAV, i.e. changes in wrench $\dot{\boldsymbol{w}}$ can not be performed arbitrarily fast due to tiltrotor dynamics. However, the proposed method is not limited to tilt rotor systems but can also be applied on fixed rotor platforms. The allocation of wrench commands is performed in a separate step to find optimal rotor speeds ω_i and tilt angles α_i. That way we aim to generate an optimal wrench trajectory $\boldsymbol{w}(t)$ rather than generating optimal trajectories for the tilt angles and rotor speeds.

The diagram in fig. 4 illustrates the control framework. Wrench generation and allocation are performed in separate steps. In the following, we first describe the allocation and then the generation of optimal wrench commands.

A. Instantaneous allocation of actuator commands

Given a desired body wrench $\boldsymbol{w}$, we want to find an optimal set of rotor speeds ω_i and tilt angles α_i for rotor units $i = 1, \ldots, n$. We follow the approach from [17] to find the combination of commands that minimizes the sum of squared rotor speeds. Based on the geometry of the platform, we describe the relation between the total body wrench $\boldsymbol{w}$ and the forces of each rotor by

$$\boldsymbol{w} = \begin{bmatrix} {}_B\boldsymbol{f} \\ {}_B\boldsymbol{\tau} \end{bmatrix} = \boldsymbol{A}\boldsymbol{X}, \quad \text{with} \quad \boldsymbol{X} = \begin{bmatrix} f_{1,l} \\ f_{1,v} \\ \vdots \\ f_{n,l} \\ f_{n,v} \end{bmatrix} = \begin{bmatrix} \sin(\alpha_1)\omega_1^2 \\ \cos(\alpha_1)\omega_1^2 \\ \vdots \\ \sin(\alpha_n)\omega_n^2 \\ \cos(\alpha_n)\omega_n^2 \end{bmatrix} \quad (6)$$

where $\boldsymbol{X} \in \mathbb{R}^{2n}$ is a vector containing lateral and vertical force components of each rotor, and $\boldsymbol{A} \in \mathbb{R}^{6\times2n}$ is a constant *allocation matrix*. Using the Moore Penrose Pseudoinverse, tilt angles α_i and rotor speeds ω_i can then be derived from any desired wrench command:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{A}^\dagger\boldsymbol{w} \\ \alpha_i &= \arctan\frac{f_{i,l}}{f_{i,v}} \\ \omega_i &= \sqrt{f_{i,l} + f_{i,v}} \end{aligned} \quad (7)$$

This gives us the following relationship between the rate of tilt angles and the rate of wrench changes:

$$\begin{aligned} \dot{\boldsymbol{X}} &= \boldsymbol{A}^\dagger\dot{\boldsymbol{w}} \\ \dot{\alpha}_i &= \frac{1}{1 + \left(\frac{f_{i,l}}{f_{i,v}}\right)^2} \cdot \frac{\dot{f}_{i,l}f_{i,v} - f_{i,l}\dot{f}_{i,v}}{f_{i,v}^2} \end{aligned} \quad (8)$$

From eq. (7) and eq. (8) we see that the tilt angle rates are directly related to the time derivatives of the wrench input. Thus, a limitation of wrench rates will implicitly limit tilt angle rates as well. This can be particularly advantageous, as fast changes in tilt angles often lead to instabilities or dynamics that are difficult to model.

B. MPC formulation

For the MPC formulation we use the rigid body model from eq. (4) and eq. (5). Based on that model, we define the state vector as follows:

$$\boldsymbol{x} = \begin{bmatrix} {}_B\boldsymbol{f}^T & {}_B\boldsymbol{\tau}^T & {}_W\boldsymbol{p}^T & {}_B\boldsymbol{v}^T & \boldsymbol{q}^T & {}_B\boldsymbol{\omega}^T \end{bmatrix}^T \quad (9)$$

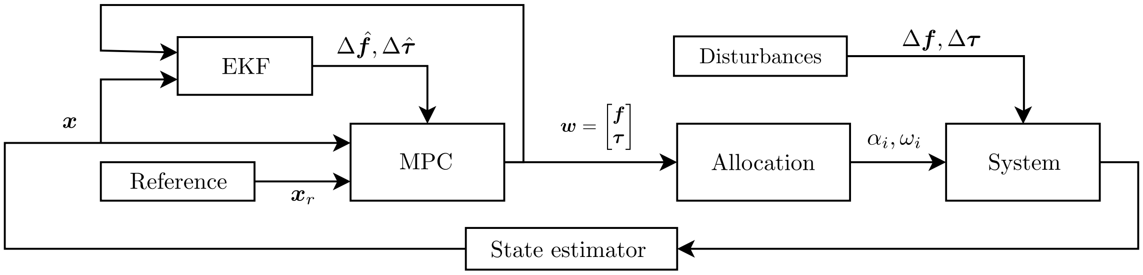
EKF
$\Delta\hat{\boldsymbol{f}}, \Delta\hat{\boldsymbol{\tau}}$
$\boldsymbol{x}$
MPC
Reference
$\boldsymbol{x}_r$
$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix}$
Disturbances
$\Delta\boldsymbol{f}, \Delta\boldsymbol{\tau}$
Allocation
α_i, ω_i
System
State estimator

Fig. 4. Block diagram.

As control inputs, we use the time derivative of the body force and torque, i.e.

$$\boldsymbol{u} = \begin{bmatrix} {}_B\dot{\boldsymbol{f}} \\ {}_B\dot{\boldsymbol{\tau}} \end{bmatrix} = {}_B\dot{\boldsymbol{w}}. \tag{10}$$

Note that the control input $\boldsymbol{u}$ is equivalent to the wrench rate from eq. (8). This allows us to impose cost and constraints on the force and torque rates, which are implicitly coupled with the tilt angle rotation speeds.

The system dynamics $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$ are described by eq. (4) and eq. (5).

The MPC formulation is then described by the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{U}} \sum_{k=0}^{N-1} & \left(\|\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{x}_{r,k})\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_k\|_{\boldsymbol{R}}^2 \right) \\ & + \|\boldsymbol{h}(\boldsymbol{x}_N, \boldsymbol{x}_{r,N})\|_{\boldsymbol{Q}_N}^2 \\ \text{s.t.} \quad & \boldsymbol{x}_k \in \mathcal{X}, \boldsymbol{u}_k \in \mathcal{U} \\ & \dot{\boldsymbol{x}} = f(\boldsymbol{x}) \\ & \boldsymbol{x}_0 = \boldsymbol{x}(t) \\ & \Delta\boldsymbol{f}_k = \Delta\hat{\boldsymbol{f}}(t) \\ & \Delta\boldsymbol{\tau}_k = \Delta\hat{\boldsymbol{\tau}}(t) \end{aligned} \tag{11}$$

where $\boldsymbol{U} = [\boldsymbol{u}_0, \boldsymbol{u}_1, \dots \boldsymbol{u}_{N-1}]$ are the commands that are optimized for. $\mathcal{X}$ and $\mathcal{U}$ denote the feasible sets for states and inputs, respectively. These can be used to limit the body wrench and its rate of change to physically feasible values. The optimization is performed over a finite horizon of N steps. The stage cost at time step k is represented by $\boldsymbol{h}(\boldsymbol{x}_k)$, and the cost for states and for inputs are weighted by the cost matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively. Having a reference trajectory defined by $\boldsymbol{p}_r(t)$, $\boldsymbol{v}_r(t)$, $\boldsymbol{q}_r(t)$, and $\boldsymbol{\omega}_r(t)$, we define the vector of stage cost as follows:

$$\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{x}_{r,k}) = \begin{bmatrix} {}_B\boldsymbol{f}_k \\ {}_B\boldsymbol{\tau}_k \\ {}_W\boldsymbol{p}_k - {}_W\boldsymbol{p}_{r,k} \\ {}_W\boldsymbol{v}_k - {}_W\boldsymbol{v}_{r,k} \\ \boldsymbol{q}_{err,k} \\ {}_B\boldsymbol{\omega}_k - {}_r\boldsymbol{\omega}_{r,k} \end{bmatrix} \tag{12}$$

The quaternion error $\boldsymbol{q}_{err}$ is modeled as the vector part of the quaternion difference between the body and the reference orientation:

$$\boldsymbol{q}_\Delta = \boldsymbol{q} \otimes \boldsymbol{q}_r^{-1} = \begin{bmatrix} q_{w,\Delta} \\ \boldsymbol{q}_{err} \end{bmatrix}, \quad \boldsymbol{q}_{err} \in \mathbb{R}^3 \tag{13}$$

The terminal state cost matrix $\boldsymbol{Q}_N$ is computed by solving the the Continuous Algebraic Ricatti Equation (CARE) to account for the infinite time cost.

We additionally impose constraints on force and torque rates, i.e. on the input $\boldsymbol{u}$:

$$\begin{aligned} -\dot{\boldsymbol{f}}_{max} \leq {}_B\dot{\boldsymbol{f}} \leq \dot{\boldsymbol{f}}_{max} \\ -\dot{\boldsymbol{\tau}}_{max} \leq {}_B\dot{\boldsymbol{\tau}} \leq \dot{\boldsymbol{\tau}}_{max} \end{aligned} \tag{14}$$

and on the forces:

$$-\boldsymbol{f}_{max} \leq {}_B\boldsymbol{f} \leq \boldsymbol{f}_{max} \tag{15}$$

The optimization from eq. (11) is solved at every control iteration and only the force and torque of the first predicted step are applied to the system, i.e. $\boldsymbol{f}_1$, and $\boldsymbol{\tau}_1$. The initial state $\boldsymbol{x}_0$ is updated at each iteration with the most recent state estimates. Disturbance force and torque are also updated at each time step and assumed to be constant throughout a prediction.

C. Disturbance observer

In order to estimate unmodeled system dynamics and external disturbances, we use a disturbance observer based on the same rigid body model as in the MPC formulation. We employ the system dynamics from eq. (4) and eq. (5) in an Extended Kalman Filter (EKF) to estimate disturbance forces and torques $\Delta\boldsymbol{f}$ and $\Delta\boldsymbol{\tau}$. The states of the EKF include all states of the system as well as the disturbances, i.e.

$$\hat{\boldsymbol{x}}_{EKF} = \begin{bmatrix} {}_W\hat{\boldsymbol{p}}^T & {}_B\hat{\boldsymbol{v}}^T & \hat{\boldsymbol{q}}^T & {}_B\hat{\boldsymbol{\omega}}^T & {}_W\Delta\hat{\boldsymbol{f}}^T & {}_B\Delta\hat{\boldsymbol{\tau}}^T \end{bmatrix}^T, \tag{16}$$

and we assume for the dynamics of the disturbances

$$\begin{aligned} {}_W\Delta\dot{\hat{\boldsymbol{f}}} = 0 \\ {}_B\Delta\dot{\hat{\boldsymbol{\tau}}} = 0. \end{aligned} \tag{17}$$

Note that we model the force disturbance to be constant in the world frame, while the torque disturbance is modeled to be constant in the body fixed frame. This is done since we assume that torque disturbances are caused by unmodeled system dynamics, while force disturbances are dominated by external causes (e.g. wind) or slowly varying vertical thrust against gravity (e.g. due to battery drain or thrust inefficiencies in certain flight configurations). Inputs to the model are the commanded forces and torques, and measurements are pose and twist, i.e.

$$\boldsymbol{u}_{EKF} = \begin{bmatrix} {}_B\boldsymbol{f} \\ {}_B\boldsymbol{\tau} \end{bmatrix} \tag{18}$$

$$\boldsymbol{z} = \begin{bmatrix} {}_W\boldsymbol{p}^T & {}_B\boldsymbol{v}^T & \boldsymbol{q}^T & {}_B\boldsymbol{\omega}^T \end{bmatrix}^T \tag{19}$$

IV. Experiments

A. Experimental setup

For experimental validation we use an OMAV that provides full 6-DOF force torque capabilities. It is designed as a dodecacopter with double rotor units that can perform full rotations around their arms' axes (see fig. 3).

State estimation is provided by fusing offboard Vicon estimates with an onboard IMU. The platform is equipped with an Intel NUC as on onboard computer and with a 3DR Pixhawk as flight controller.

The MPC framework is implemented using ROS and ACADO [18]. It employs a multiple shooting method for the discretization of the continuous model formulation and an implicit Runge Kutta method for integration of the discretized system [19]. The solver is executed at a rate of 100 Hz and sends force/torque commands to the flight controller. The flight controller performs the allocation and controls rotor speeds and tilt angles at a rate of 200 Hz, it applies the most recently received wrench command in a zero order hold fashion.

The prediction horizon is set to a length of $N = 20$ steps, with a time discretization of $\Delta t = 0.05\,\text{s}$, which results in a total prediction time of $1\,\text{s}$.

The cost matrices $\mathbf{Q}$ and $\mathbf{R}$ were hand tuned to penalize pose and twist errors as well as fast changes of the input wrench ${}_B\dot{\boldsymbol{w}}$. Too small values of $\boldsymbol{R}$ can lead to aggressive and destabilizing behaviour, while too large values lead to worse pose tracking performance.

B. Experiment design

In order to evaluate the tracking performance of the proposed controller, we design a reference trajectory in the shape of a figure eight with rotations around different axes along the path (see fig. 5). Linear and angular position and velocity setpoints are sampled based on the trajectory at a rate of 100 Hz.

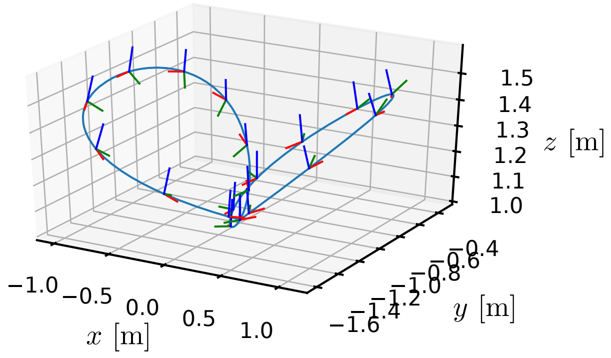

Fig. 5. Illustration of the path and orientations during the standard reference trajectory.

We use this reference trajectory to compare different control approaches with respect to their tracking accuracy.

V. Evaluation

A. Tracking of a 6-DoF trajectory

We evaluate the tracking performance by observing the tracking error separately in 3 directions, i.e. in the world frame x-, y-, and z-axis, and in the Euler angle yaw-pitch-roll convention, denoted by the angles ψ, θ, and ϕ. Figure 6 shows the absolute tracking performance for both position and attitude, using MPC fused with online disturbance estimation. While the controller is generally able to track the reference trajectory, some constant position offset can be observed. This can mainly be explained by a significant difference between real and modeled dynamics. While the disturbance observer partly compensates for this difference, the remaining model difference impedes exact tracking. This can especially be seen when inspecting the position predictions in fig. 7. The deviation of predictions from the eventually executed path indicates the effect of the modeling error.

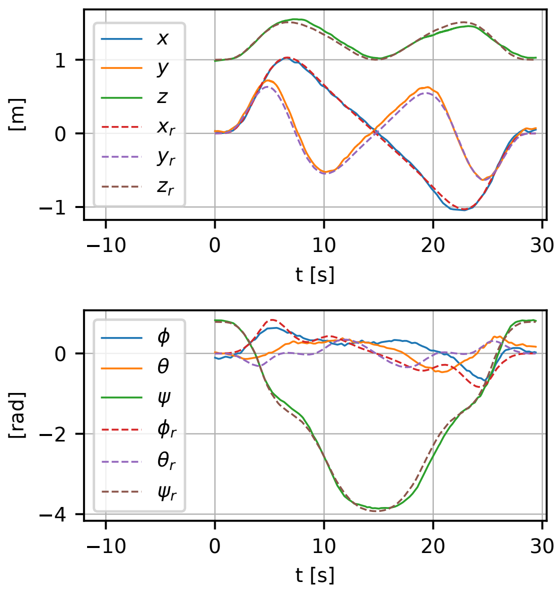

Fig. 6. Position and attitude tracking performance along the standard reference trajectory.

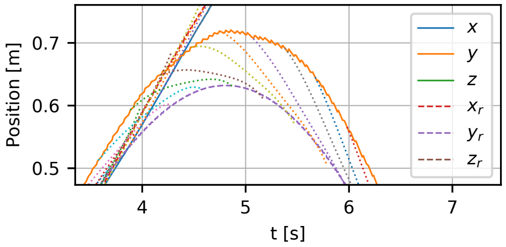

Fig. 7. Detail of the trajectory reference (dashed), predictions (dotted), and the eventually performed trajectory (solid line).

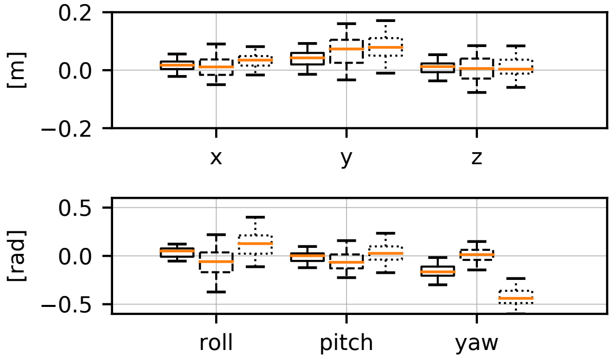

Fig. 8. Comparison of tracking errors of PD (left), MPC with EKF (center), and MPC without EKF fusion for disturbance compensation (right, respectively)

Figure 8 compares the tracking errors of a PD controller, of the MPC with EKF fusion, and with the MPC without EKF fusion. We use boxplots to show the median tracking errors, the upper and lower quartile, as well as outliers within the 1.5 interquartile range.

The plots clearly illustrate the overall good performance of a standard PD controller. While the tracking errors of the MPC with EKF are generally slightly larger than in the PD control, the MPC without EKF compensation exhibits an especially large tracking error in yaw. This can be explained by a constant unmodeled yaw torque that leads to a constant offset in yaw tracking.

B. *Tracking of a pure translation trajectory*

Figures 9 and 10 illustrate the versatility of the chosen approach. In both figures, the same reference trajectory is tracked. In fig. 9, full force omnidirectionality is assumed, while in fig. 10 lateral accelerations are constrained to $a_{max} = 0.2\,\mathrm{m\,s^{-2}}$. This essentially results in the control of a platform with laterally bounded forces, similar to an underactuated MAV. Even though tracking performance is impaired, the controller is still able to track the given reference.

Throughout both types of experiments and for the horizon length of 20 time steps, the computation times of the control loop stayed within the limits of $10\,\mathrm{ms}$ (average $\approx 6\,\mathrm{ms}$).

VI. Conclusion

We presented a model based controller for overactuated flying platforms which generates optimal wrench commands, based on the physical model of the system. Constraints that result from mechanical and/or dynamical limitations can easily be included in the control formulation.

While the disturbance observer helped to improve tracking performance on low angle trajectories, the performance decreased with steeper angles, as the observer was not able to estimate strong disturbances fast enough. We also found that constraining force and torque rates can reduce aggressive tilt angle movements which might otherwise lead to instabilities. While we performed experiments on a fully overactuated platform which allowed for full 6-DoF wrench generation, this method can easily be adapted to slightly overactuated platforms by applying hard constraints on the force and torque envelope.

We hope that identifying model dynamics in more detail will allow to fully exploit the advantages of MPC on overactuated flying platforms. This will be of particular interest when considering future scenarios such as integrating further actuators or for interaction tasks. Model based control could then provide a unified optimal control framework for complex control tasks.

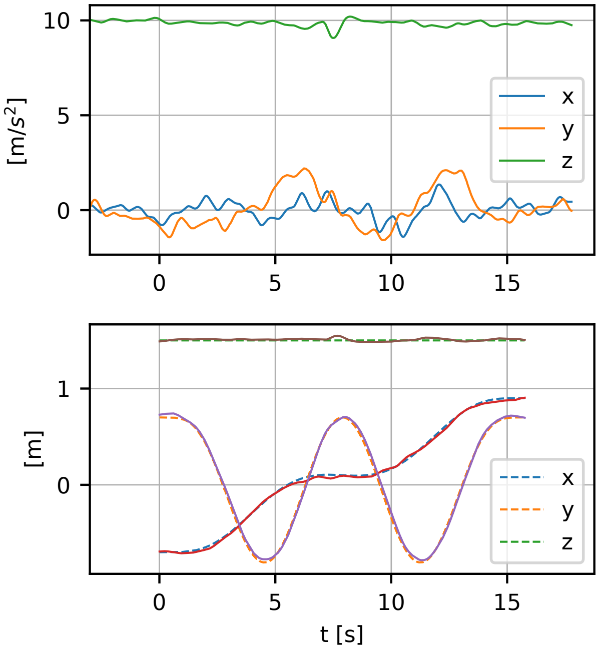

Fig. 9. Tracking of a pure translation trajectory using full omnidirectionality.

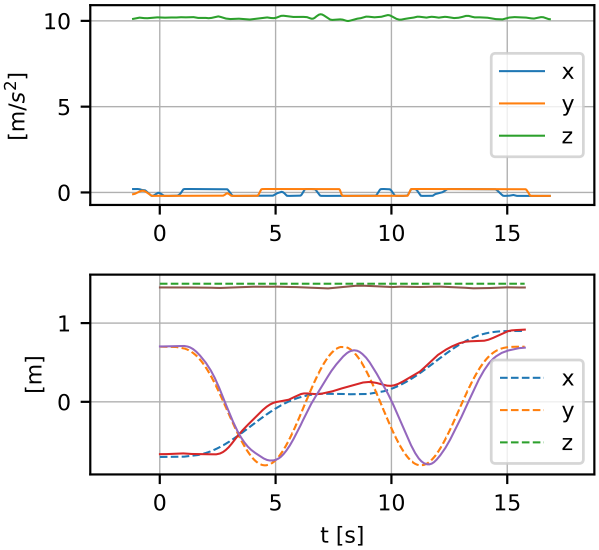

Fig. 10. Acceleration commands and position tracking performance, using constrained lateral accelerations.

REFERENCES

[1] V. Lippiello and F. Ruggiero, "Exploiting redundancy in Cartesian impedance control of UAVs equipped with a robotic arm," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3768–3773, 2012.

[2] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. Ramiro Martinez-De Dios, F. Pierri, J. Cortes, A. Santamaria-Navarro, M. A. T. Soto, R. Balachandran, J. Andrade-Cetto, and A. Rodriguez, "The AEROARMS Project: Aerial Robots with Advanced Manipulation Capabilities for Inspection and Maintenance," *IEEE Robotics and Automation Magazine*, vol. 25, no. 4, pp. 12–23, 2018.

[3] B. Chan, H. Guan, J. Jo, and M. Blumenstein, "Towards UAV-based bridge inspection systems: a review and an application perspective," *Structural Monitoring and Maintenance*, vol. 2, no. 3, pp. 283–300, 2015.

[4] D. Brescianini and R. D'Andrea, "Design , Modeling and Control of an Omni-Directional Aerial Vehicle," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3261–3266, 2016.

[5] S. Park, J. Her, J. Kim, and D. Lee, "Design, modeling and control of omni-directional aerial robot," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 1570–1575, 2016.

[6] S. Park, J. Lee, J. Ahn, M. Kim, J. Her, G.-H. Yang, and D. Lee, "ODAR: Aerial Manipulation Platform Enabling Omni-Directional Wrench Generation," *submitted to IEEE/ASME Transactions on Mechatronics*, vol. PP, no. c, p. 1, 2018.

[7] D. Bicego, N. Staub, V. Arellano, S. Mishra, and A. Franchi, "Towards a Flying Companion Paradigm: the OTHex," pp. 6997–7002, 2018.

[8] A. Franchi, R. Carli, D. Bicego, and M. Ryll, "Full-Pose Tracking Control for Aerial Robotic Systems with Laterally Bounded Input Force," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 534–541, 2018.

[9] M. Ryll, H. H. Bülthoff, and P. R. Giordano, "A Novel Overactuated Quadrotor Unmanned Aerial Vehicle : Modeling , Control , and Experimental Validation," vol. 23, no. 2, pp. 540–556, 2015.

[10] R. Falconi and C. Melchiorri, *Dynamic model and control of an over-actuated quadrotor UAV*. IFAC, 2012, vol. 45, no. 22. [Online]. Available: http://dx.doi.org/10.3182/20120905-3-HR-2030.00031

[11] M. Ryll, D. Bicego, and A. Franchi, "Modeling and Control of FAST-Hex : a Fully Actuated by Synchronized Tilting Hexarotor," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, no. 644271, pp. 1689–1694, 2016.

[12] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto, "An Omnidirectional Aerial Manipulation Platform for Contact-Based Inspection," 2019. [Online]. Available: http://arxiv.org/abs/1905.03502

[13] 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," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 1398–1404, 2016.

[14] M. Kamel, K. Alexis, M. W. Achtelik, and R. Siegwart, "Fast nonlinear model predictive control for multicopter attitude tracking on SO(3)," *2015 IEEE Conference on Control and Applications, CCA 2015 - Proceedings*, no. 3, pp. 1160–1166, 2015.

[15] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances," *Control Engineering Practice*, vol. 19, no. 10, pp. 1195–1207, 10 2011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0967066111001262

[16] J. Solà, "Quaternion kinematics for the error-state Kalman filter," 2017. [Online]. Available: http://arxiv.org/abs/1711.02508

[17] K. Bodie, Z. Taylor, M. Kamel, and R. Siegwart, "Towards Efficient Full Pose Omnidirectionality with Overactuated MAVs."

[18] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, 2011.

[19] R. Quirynen, M. Vukov, and M. Diehl, "Auto generation of implicit integrators for embedded NMPC with microsecond sampling times," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 4, no. PART 1, pp. 175–180, 2012.