Disturbance Estimation and Rejection for High-Precision Multirotor Position Control

Daniel Hentzen[1,2], Thomas Stastny[2], Roland Siegwart[2], Roland Brockers[1]

Abstract—Many multirotor Unmanned Aerial Systems applications have a critical need for precise position control in environments with strong dynamic external disturbances such as wind gusts or ground and wall effects. Moreover, to maximize flight time, small multirotor platforms have to operate within strict constraints on payload and thus computational performance. In this paper, we present the design and experimental comparison of Model Predictive and PID multirotor position controllers augmented with a disturbance estimator to reject strong wind gusts up to 12 m/s and ground effect. For disturbance estimation, we compare Extended and Unscented Kalman filtering. In extensive in- and outdoor flight tests, we evaluate the suitability of the developed control and estimation algorithms to run on a computationally constrained platform. This allows to draw a conclusion on whether potential performance improvements justify the increased computational complexity of MPC for multirotor position control and UKF for disturbance estimation.

Fig. 1. Example of the effect of a strong wind gust on a quadrotor.

SUPPLEMENTARY MATERIAL

Video: `https://youtu.be/-1PvZ5YBIuw`

I. INTRODUCTION

A. Motivation

For a majority of proposed multirotor Unmanned Aerial Systems (UAS) applications, successful deployment relies on the ability of the flight controller to handle unpredictable and potentially adverse conditions inherent to real-world outdoor environments. In particular, the control system must accurately track position references despite the presence of strong and dynamic disturbances, such as wind gusts or ground and wall effects. Examples of relevant applications include outdoor docking maneuvers, for instance for autonomous recharging [1], industrial inspection [2] or aerial package delivery [3]. In the latter application for instance, the vehicle must follow a path and perform landings accurately despite the presence of a static disturbance (due to the varying payload mass) as well as wind gusts and ground effects that occur during flight through urban landscapes.

Recent work has shown that the problem of accurate multirotor position control is well suited to a Model Predictive Control (MPC) formulation, since it is able to compute optimized control inputs while accounting for state and input constraints [4]. However, the capability of multirotor MPC to deal with strong dynamic disturbances remains uncertain.

While previous work has shown that estimates of external disturbances can be computed online and used within the flight control system to improve control performance (see Section I-C), the suitability of this approach with model predictive position controllers is untested in practice and shall thus be studied in this work.

Moreover, we aim to gather conclusive experimental insights on whethe the choice of MPC over classical control formulations for multirotor position control in real-world environments is justified by substantial performance gains, considering MPC's higher complexity and computational cost.

B. Contribution

In this work, we address the above research questions by presenting a first-of-a-kind experimental comparison of two position controllers for a small quadrotor platform: a Nonlinear Model Predictive controller and a conventional PID controller. Both control formulations are augmented with a disturbance estimation module that enables the rejection of dynamic disturbances. For disturbance estimation, we compare Extended and Unscented estimators. In extensive flight tests, we assess the disturbance estimation and rejection performance of the presented algorithms in the presence of a) ground effect and b) wind gusts up to 12 m/s. We also measure the computational footprint of the control and estimation algorithms on a small microprocessor. The disturbance estimator comparison is first performed in a Vicon motion capture environment with near-perfect state estimation (as in existing work on multirotor disturbance estimation), but also repeated for the first time outdoors with degraded GPS-based state estimation. We conclude that the increased complexity

[1]Authors are with the Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Pasadena, CA 91109, USA `hentzend@gmail.com`; `brockers@jpl.nasa.gov`

[2]Authors are with the Swiss Federal Institute of Technology (ETH) Zurich, Autonomous Systems Lab, Leonhardstrasse 21, 8092 Zurich, Switzerland (`firstname.lastname`)`@ethz.ch`
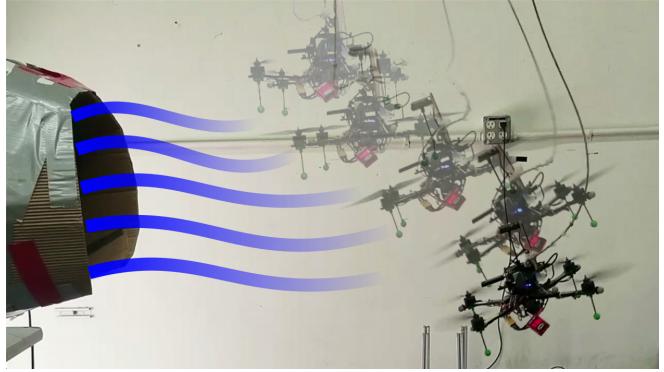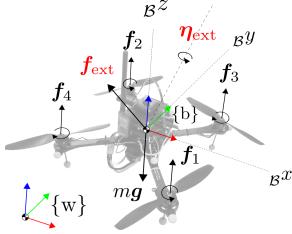
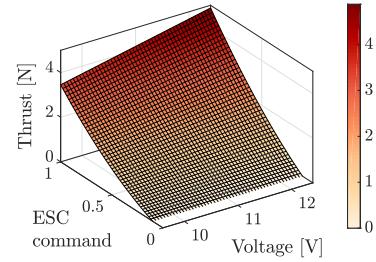Fig. 2.   Quadrotor free-body diagram and coordinate systems.

Fig. 3.   Identified single rotor thrust map for different ESC commands (here shown normalized for a PWM range of 1000 - 2000) and different battery voltage levels.

of MPC is justified by significant improvements in position tracking under strong disturbances compared to PID control. Moreover, we determine that Unscented filtering does not offer compelling performance gains over Extended filtering to justify its higher computational cost.

C. Related Work

Robust control formulations for multirotor applications, such as $\mathcal{H}_\infty$ [5] or Robust MPC [6], have been developed with the aim to provide robust performance guarantees given an unknown but bounded disturbance. However, in unpredictable real-world environments, the disturbance bounds have to be chosen conservatively to guard against worst-case disturbances, resulting in degraded control performance. Alternatively, adaptive control formulations are designed to determine uncertain system parameters online. The authors in [7] present quadrotor controllers based on Model Reference Adaptive Control and show improved reaction to perturbations such as a sudden loss-of-thrust or weight disturbances compared to non-adaptive baseline controllers. In [8], an $\mathcal{L}_1$ adaptive controller is employed to reject disturbances caused by a suspended weight pendulum. A known drawback of adaptive controllers is a lack of robustness due to phenomena such as bursting [7]. This motivates the need for robustness modifications that limit the adaptation performance and thus the rejection of dynamic disturbances.

A promising approach to reject dynamic disturbances is to observe the disturbance online and compute a compensating control input. In [9], a momentum-based estimator of external forces is used in conjunction with an impedance controller. In [10], a Nonlinear Inversion-based position controller uses filtered accelerometer measurements to reconstruct and compensate the acceleration caused by external disturbance, resulting in significant improvement of disturbance rejection. In [11], [12], the external force [11] [12] and torque vectors [11] are estimated using acceleration and velocity measurements. However, the quality of the above disturbance estimation methods degrades with sensor noise, as the proposed observers are deterministic and do not account for measurement or process noise models [13]. It is thus preferable to use stochastic estimators if sufficient computational resources are available. In [14], [13], two disturbance force and torque estimators, based on Unscented Kalman Filtering are presented. Yet it is unclear whether the use of the more computationally costly UKF over an EKF is

justified. In [4], external disturbance forces are estimated by an EKF. However, no experimental results for the disturbance estimator are presented. The disturbance estimates are used within a nonlinear MPC framework to achieve accurate trajectory tracking in the presence of a 11 m/s wind disturbance. In this work, we use the same nonlinear MPC approach as in [4] in a path following formulation, extend it with optional soft constraints and compare its performance to that of an augmented PID controller.

II. SYSTEM MODELING

A. Vehicle Dynamics

We consider a quadrotor controlled by four individual rotor thrusts of magnitude f_i and rotor drag torques of magnitude η_i, yielding a collective thrust T and body torque $\boldsymbol{\eta}_{\text{prop}}$. Additionally, an external disturbance force $\boldsymbol{f}_{\text{ext}}$ and external disturbance torque ${}_{\mathcal{B}}\boldsymbol{\eta}_{\text{ext}}$ act on the vehicle. Figure 2 shows the free-body diagram and coordinate systems.

We can derive the translation dynamics of the quadrotor in world frame using Newton's 2nd law for rigid bodies:

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{1}$$

$$\dot{\boldsymbol{v}} = \frac{1}{m}\big(\mathbf{C}_{\mathcal{B}}^{\mathcal{W}} T e_z^b + \boldsymbol{f}_{\text{ext}}\big) + \boldsymbol{g}, \tag{2}$$

where $\mathbf{C}_{\mathcal{B}}^{\mathcal{W}}$ is the rotation matrix from body to world frame.

The quadrotor attitude dynamics are derived using the Euler equation for rigid bodies and quaternion rate of change kinematics:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \otimes \boldsymbol{q},$$

$$_{\mathcal{B}}\dot{\boldsymbol{\omega}} = J^{-1}\bigg({}_{\mathcal{B}}\boldsymbol{\eta}_{\text{prop}} + {}_{\mathcal{B}}\boldsymbol{\eta}_{\text{ext}} - {}_{\mathcal{B}}\boldsymbol{\omega} \times \big(J\,{}_{\mathcal{B}}\boldsymbol{\omega}\big)\bigg),$$

where $\otimes$ represents a quaternion multiplication.

B. Thrust and Drag Torque Maps

The disturbance estimation and control algorithms presented below require knowledge and control of the output rotor thrusts and drag torques. Due to the lack of both closed-loop motor control and motor RPM measurements on our platform, we identify a map from the commanded ESC commands $u_{\text{ESC},i}$ to rotor thrust and torque magnitudes f_i
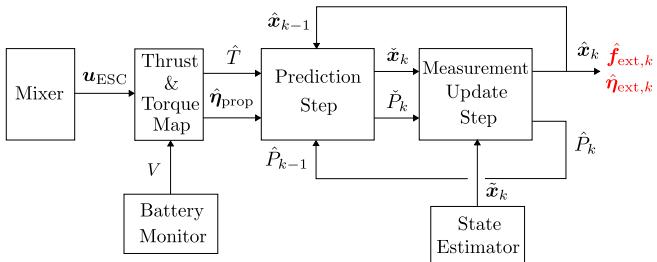
Fig. 4. Recursive filtering architecture for disturbance estimation.

and η_i respectively. We use the same quadratic map between ESC command and output thrust/torque as in [15] and extend it with a linear dependency on battery voltage. Figure 3 shows an example of the identified voltage-dependent thrust map for a single rotor.

III. DISTURBANCE ESTIMATION

To estimate the external force $\boldsymbol{f}_{\text{ext}}$ and torque $\boldsymbol{\eta}_{\text{ext}}$ acting on the quadrotor, we design and implement two different disturbance estimators: an Extended and an Unscented Kalman Filter.

Due to the shared recursive nature, we employ an identical architecture for both filters, illustrated in Figure 4. The individual ESC motor commands $\boldsymbol{u}_{\text{ESC}}$ are converted into an estimated net thrust $\hat{T}$ and body torque $\hat{\boldsymbol{\eta}}_{\text{prop}}$ using our thrust and drag torque calibration (Section II-B). The prediction step then propagates the vehicle dynamics to yield a predicted state mean and covariance. The vehicle state includes the disturbance terms to be estimated. The predicted state mean and covariance are corrected using measurements provided by the onboard state estimator, yielding the estimated disturbance force/torque mean and covariance.

A. Extended Kalman Filter

a) System State: We define the system state $\boldsymbol{x}$ as the vehicle position, velocity, attitude and angular rates in body frame and augment it with the external forces and torques:

$$\boldsymbol{x} := \begin{bmatrix} \boldsymbol{p}^T & \boldsymbol{v}^T & \phi & \theta & \psi & _\mathcal{B}\boldsymbol{\omega}^T & \boldsymbol{f}_{\text{ext}}^T & \boldsymbol{\eta}_{\text{ext}}^T \end{bmatrix}^T.$$

Despite the gimbal-lock singularity at $\theta = \pi/2$, we choose an Euler attitude representation for the EKF design. In contrast to the quaternion representation, the linearization is simplified and does not require quaternion normalization checking [16].

The evolution of the system state $\boldsymbol{x}$ is subject to zero-mean Gaussian process noise $\boldsymbol{v}$ with covariance Q.

b) System Input: The system input $\boldsymbol{u}$ consists of the estimated collective propeller thrust $\hat{T}$ and body torques $_\mathcal{B}\hat{\boldsymbol{\eta}}_{\text{prop}}$:

$$\boldsymbol{u} := \begin{bmatrix} \hat{T} & _\mathcal{B}\hat{\boldsymbol{\eta}}_{\text{prop}}^T \end{bmatrix}^T.$$

c) Measurement Model: The measurement vector $\boldsymbol{z} = \tilde{\boldsymbol{x}}$ is provided by the state estimator:

$$\boldsymbol{z} := \begin{bmatrix} \tilde{\boldsymbol{p}}^T & \tilde{\mathbf{v}}^T & \tilde{\phi} & \tilde{\theta} & \tilde{\psi} & \tilde{\boldsymbol{\omega}}^T \end{bmatrix}^T.$$

The measurements are subject to additive zero-mean Gaussian process noise $\mathbf{w}$ with covariance R.

d) Discrete System Dynamics: To formulate the EKF prediction step for external force and torque estimation, we discretize the quadrotor dynamics presented in Section II-A (with Euler angles replacing quaternions) using a forward Euler scheme. To allow a flexible application of the disturbance estimator, we model the force and torque dynamics as a Gaussian random walk:

$$\boldsymbol{f}_{\text{ext},k} = \boldsymbol{f}_{\text{ext},k-1} + \mathbf{v}_f,$$
$$_\mathcal{B}\boldsymbol{\eta}_{\text{ext},k} = {}_\mathcal{B}\boldsymbol{\eta}_{\text{ext},k-1} + \mathbf{v}_\eta.$$

The process noise value $\mathbf{v}$ is a tuning parameter that allows to control the convergence speed of the estimate, at the cost of a noisier output signal.

e) Prediction and Measurement Update Steps: The prediction and measurement update steps are performed using the standard EKF equations [17].

B. Unscented Kalman Filter

In the UKF, we use the same state as in the EKF, but replace the Euler angle representation with a quaternion representation. Since there is no need to derive a Jacobian matrix, the quaternion representation is easy to handle in the UKF formulation and avoids the gimbal-lock singularity at $\theta = \pi/2$. For the attitude representation, we follow the same multiplicative approach used in [14]. The mean attitude is represented using singularity-free, unit-norm quaternions, whereas the rotational uncertainty is parameterized and passed through the Unscented Transform in the form of Modified Rodrigues Parameters (MRP).

a) State, Input, Measurements: We maintain the same state, input and measurement vectors as in the EKF, but replace Euler angles with quaternions. The process covariance matrix Q, the noise covariance matrix R and the measurement model matrix H follow their EKF counterparts.

b) Discrete System Dynamics: The UKF uses the same discrete translation dynamics as the EKF, while the discrete quaternion-based rotation dynamics are detailed in [18].

c) Prediction and Measurement Update Steps: The prediction and measurement update steps are performed using the standard UKF equations [17], [19]. For switching between the full attitude quaternion and the MRPs we follow the approach outlined in [14].

IV. POSITION CONTROL WITH DISTURBANCE REJECTION

A. Control System Overview

Figure 5 provides a schematic overview of the entire flight control system and how it interfaces with the guidance and disturbance estimation modules described in Sections IV-B and III. The control system is based on a cascaded
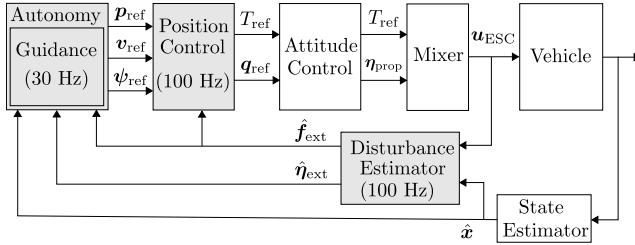
Fig. 5. Guidance and control system architecture. Shaded modules are executed on an *Odroid XU4* companion computer, while white modules are running on a *Pixhawk mini* microcontroller autopilot.

architecture. Given a state estimate as well as position and velocity references issued by the guidance module, the position controller computes a desired acceleration vector. The latter is then transformed into a collective thrust and a reference attitude quaternion, which are passed as references to the low-level attitude controller. Both of the presented position controllers use the force estimates provided by the disturbance estimator to reject the external disturbance acting on the vehicle. While the torque estimates could be used within the attitude controller, we decide to limit the scope of the active disturbance rejection to the position controller, in order to assess performance in combination with the widely used off-the-shelf *PX4* attitude controller.

B. Path Following Guidance

To maximize robustness to large external disturbances, we choose to implement a path following guidance system (as opposed to trajectory tracking), which guides the vehicle to the closest point on the spatially-parameterized reference path while maintaining a pre-defined reference velocity along the path tangent. In the presence of large disturbances, this prevents an accumulation of the position and velocity errors over time and thus excessively aggressive flight maneuvers without the need for computationally expensive motion replanning. Such an approach is well suited for applications that need to achieve a certain level of precision in order to avoid hazards, such as precision landing [1] for instance, where a precise approach and a safe touch-down (for example on a recharging station) is critical.

C. PID Control Formulation

The PID controller computes the desired acceleration components as follows:

$$a_{\text{des},i} = k_{p,i}e_{p,i} + k_d e_{v,i} + k_i \int e_{p,i} - \frac{1}{m}\hat{f}_{\text{ext},i}, \quad (3)$$

where $i \in \{x, y, z\}$. The conventional PID structure is augmented with a feedforward disturbance acceleration term to compensate for the estimated force disturbance $\hat{f}_{\text{ext}}$. The output acceleration vector a_{des} is then transformed into a reference attitude quaternion and collective thrust [20].

D. MPC Control Formulation

a) System Model: We implement a nonlinear MPC based on the full nonlinear system dynamics. In [4] the authors showed that nonlinear MPC offers superior performance to linear MPC for multirotor trajectory tracking, while fast loop times can be achieved for both formulations thanks to recent advances in open-source nonlinear solvers [21].

The translation dynamics are formulated in (1) and (2). The model includes the external disturbance force acting on the vehicle. All unmodeled dynamics (such as aerodynamic drag) are lumped together in this disturbance force. We assume a constant disturbance force over the MPC prediction horizon. As in [4], we identify the attitude dynamics as a first-order system. This allows us to deploy the MPC position controller with arbitrary (open- or closed-source) attitude controllers.

b) Optimal Control Problem: The system state is given by the vehicle position, velocity and Euler angles:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}^T & \boldsymbol{v}^T & \phi & \theta & \psi \end{bmatrix}^T.$$

The control input to be computed consists of the net thrust and a reduced roll-pitch attitude in the form of Euler angles:

$$\boldsymbol{u} = \begin{bmatrix} T_{\text{ref}} & \phi_{\text{ref}} & \theta_{\text{ref}} \end{bmatrix}^T$$

The control inputs serve as the reference to the low-level attitude controller. The yaw reference is directly passed to the attitude controller.

We can now formulate the continuous, nonlinear optimal control problem (OCP) to be solved by the MPC at each iteration:

$$\min_{\boldsymbol{U}} \int_{t=0}^{T} \left[\bar{\boldsymbol{x}}(t)^T Q_x \bar{\boldsymbol{x}}(t) + \bar{\boldsymbol{u}}(t)^T R_u \bar{\boldsymbol{u}}(t) \right] dt$$
$$+ \bar{\boldsymbol{x}}(T)^T P_N \bar{\boldsymbol{x}}(T),$$

subj. to $\quad \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \quad$ (system dynamics)
$$\boldsymbol{u}(t) \in \mathbb{U}, \quad \text{(thrust, roll, pitch input constraints)}$$
$$\boldsymbol{x}(0) = \boldsymbol{x}(t_0), \quad \text{(initial condition)}$$

$$(4)$$

where $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$ denote the state and input error. The references for the states and inputs are provided by the guidance module. The constrained input space is given by

$$\mathbb{U} = \left\{ \boldsymbol{u} \in \mathbb{R}^3 \,\middle|\, \begin{bmatrix} T_{\min} \\ -\phi_{\max} \\ -\theta_{\max} \end{bmatrix} \leq \boldsymbol{u} \leq \begin{bmatrix} T_{\max} \\ \phi_{\max} \\ \theta_{\max} \end{bmatrix} \right\}.$$

c) Soft-Constrained MPC: We implement optional constraints on the cross-error along the x and y axes. To reduce the risk of infeasibility, we use a soft constraint, or slack constraint. The constraint value can be chosen based on the required accuracy or based on the position of potential hazards along the flight path. The soft constraint is given by $|e_{p,xy}(t)| \leq e_{\max} + \epsilon$, where ϵ is the slack control variable, which we add to the vector of control inputs:

$$\boldsymbol{u} = \begin{bmatrix} T & \phi_{\text{ref}} & \theta_{\text{ref}} & \epsilon \end{bmatrix}^T.$$

d) Solver: Solving the nonlinear OCP (4) at a high rate requires an efficient nonlinear solver routine. We formulate the MPC problem and generate a C++ interface using the ACADO toolkit for Matlab [21] and use a qpOASES solver [22] with multiple-shooting discretization technique. The prediction horizon is set to 2 s, the discretization interval to 0.1 s.

V. EXPERIMENTS

A. Implementation

We implement the disturbance estimators and position controllers as C++ libraries wrapped in ROS [23] nodes. All flight tests are performed with a 700 g custom quadrotor based on an *Asctec Hummingbird* frame equipped with a *Pixhawk mini* autopilot connected to an *Odroid XU4* onboard computer (2 GHz ARM processor, 2GB RAM). The state estimation and low-level controllers run on the *Pixhawk* hardware, while the presented estimation and control algorithms run on two of the high-performance cores of the *Odroid XU4*, leaving sufficient resources to run other elements of a typical UAS flight stack on the same processor.

B. Testing Environments

We first test and initially tune the C++ disturbance estimator and controller implementations in a *RotorS*-based [24] simulation. We then fine-tune and validate the disturbance estimator in an indoor *Vicon* environment where we use motion capture inputs for state estimation. Finally, we perform outdoor tests using an off-the-shelf GPS module for GPS-based state estimation. To ensure repeatability and fair comparisons, the control experiments are conducted exclusively in the motion capture environment.

C. Disturbance Estimation

a) Computational Cost: We measure the iteration time of the EKF and UKF over the course of repeated landing maneuvers subject to wind gust disturbance. Both disturbance estimators are running in parallel on separate cores of the *Odroid XU4* companion computer. The EKF iteration time averages at 8 ms, never exceeding 10 ms, whereas the UKF iteration time averages at around 20 ms, a 2.5-fold increase to the EKF. The additional computational cost of the UKF is largely due to the propagation of all sigma points and their recombination into a state mean and covariance at every timestep, involving a costly Cholesky matrix decomposition [14]. On the *Odroid XU4*, the EKF can thus be run at a high update rate of 100 Hz, whereas the UKF can only be run at a maximum of 40 Hz.

b) Transient Performance: To maximize rejection of fast disturbances such as wind gusts, a fast transient estimation is required. At the same time, overshoots and excessive noise in the disturbance estimate signals must be avoided to prevent a degradation of control performance. The tuning of the estimator covariance values is performed in a way that achieves this compromise. The transient performance is evaluated with a disturbance step response. For this purpose we attach a weight ($m = 0.1$ kg) to the upper right foot of
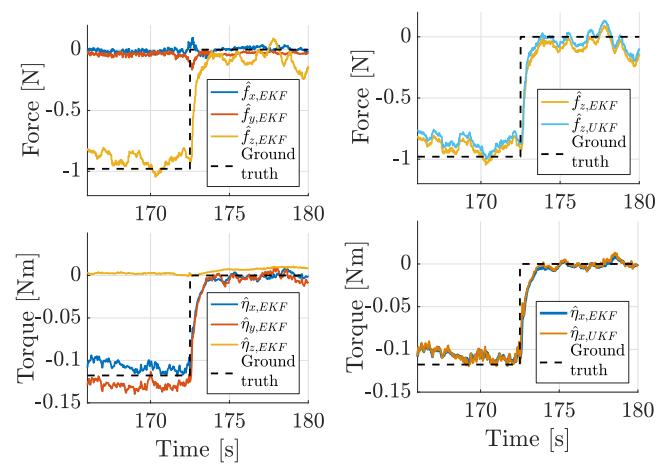

Fig. 6. Step response of the EKF force and torque disturbance estimator, running at 100 Hz.
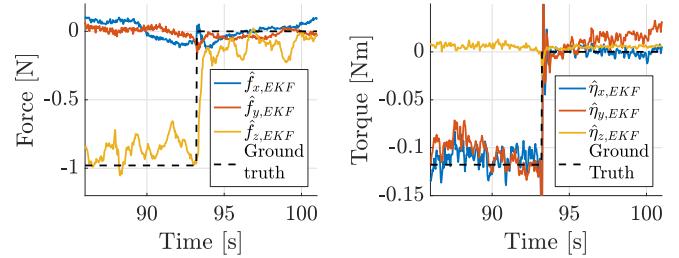
Fig. 7. EKF/UKF comparison of the estimated disturbance force in z-axis and estimated roll torque disturbance.
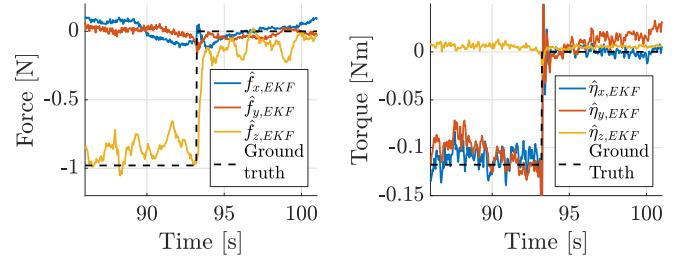

Fig. 8. Step response of the EKF force and torque disturbance estimator outdoors using GPS-based state estimation.

the quadrotor, which we cut off in flight. Figure 6 shows that the EKF force and torque estimates converge to the ground truth values. The force estimate has a t_{90} rise time of 1.1 s, while the torque estimate has a t_{90} rise time of 1.0 s. The slight offset of the roll and pitch torque estimates to the ground truth can be explained by a slightly offset weight positioning.

To evaluate whether the UKF can improve on the EKF in terms of accuracy, convergence speed or noise, we run the UKF in parallel to the EKF at 40 Hz with the same variance settings. Figure 7 shows that the EKF and UKF estimates are nearly identical in terms of the above criteria. We conclude that the UKF does not offer superior performance and choose the EKF for disturbance estimation due to its reduced computational footprint.

To assess the impact of degraded state estimation on the performance of the EKF disturbance estimator, we repeat the experiment outdoors using GPS-based state estimation. Figure 8 shows that the disturbance estimator still provides accurate force and torque disturbance estimates. The force estimate exhibits similar convergence speed and only slightly increased noise compared to the estimate in Figure 6. The torque estimate is considerably noisier, since the state estimator now relies purely on on-board sensing for
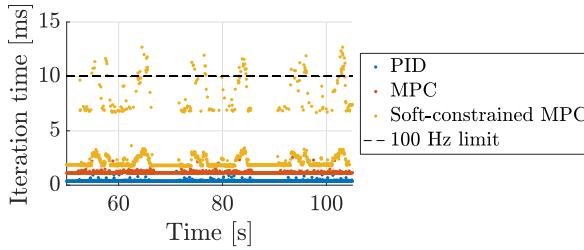
Fig. 9. Comparison of iteration times between the PID, the MPC and the soft-constrained MPC path following controllers during repeated flights through a wind gust of 10 m/s.

attitude estimation, in contrast to the motion-capture-based state estimator used in the indoor experiment.

D. Position Control - Computational Cost

We measure the time it takes to run one control iteration over the course of repeated takeoff and landing maneuvers through a wind gust of around 10 m/s. Figure 9 shows that the PID controller achieves iteration time average of 0.4 ms and maximum iteration times below 2 ms. The nonlinear MPC offers very efficient performance as well, with an iteration time average of 1.3 ms, with all measurements below 4 ms. Thus, the nonlinear MPC can easily be run at a fast update rate of 100 Hz on one core of the *Odroid XU4*.

Computational cost increases significantly when adding an additional soft state constraint to the optimal control problem. In nominal conditions (no significant disturbances present), the average iteration time for the soft-constrained MPC is 2.5 ms. However, when subject to the strong wind gust disturbance, the original constraint is relaxed and the OCP takes significantly longer to solve, occasionally exceeding the 10 ms mark, but always staying below 13 ms. Still, the soft-constrained MPC can be run on the *Odroid XU4* at a reasonably fast update rate of 70 Hz.

E. Ground Effect Disturbance Rejection

a) Experiment Setup: To test the ability of our framework to reject forces induced by ground effect, we perform repeated horizontal flights at a velocity of 0.3 m/s over the edges of a table, in close proximity (8 cm) to the table's surface. The experiment setup is illustrated in Figure 10. As we will show in V-F, there is little difference in the disturbance rejection capability for weak force disturbances between PID and MPC. Hence we perform this experiment using only the PID controller, with and without disturbance estimation and compensation.

b) Results: The top plot in Figure 11 shows that as soon as the vehicle crosses the left edge of the table, the disturbance force z-component converges to a positive value. As the vehicle crosses the opposing table edge, the disturbance estimate converges back to zero. The bottom plot in Figure 11 shows that without disturbance rejection, the vehicle overshoots the position reference by 5 cm to 10 cm as it crosses the left table edge and enters the region affected by ground effect. As the vehicle flies over the
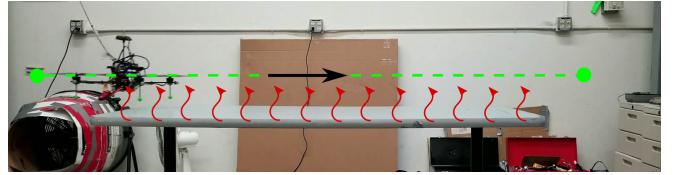

Fig. 10. Ground effect experiment setup. Reference flight path in green. Red arrows represent the zone affected by ground effect.
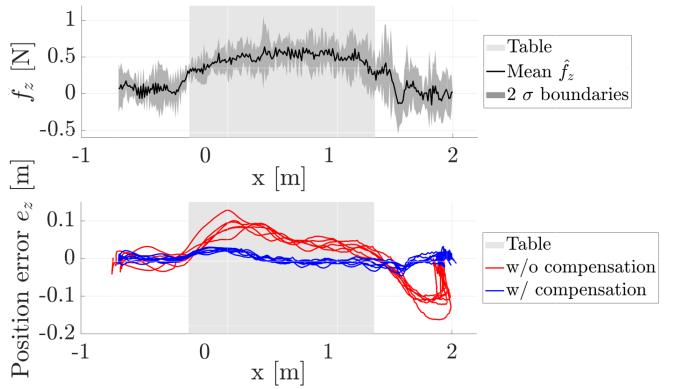

Fig. 11. *Top*: Mean and 2σ bounds of the estimated force disturbance along the z-axis during repeated flights over a table surface (see Figure 10). *Bottom*: Position error along the z-axis for the flights with (in blue) and without (in red) disturbance rejection.

surface, the integral error in the PID controller starts winding up, gradually correcting the error. However, as soon as the vehicle crosses the right edge of the table and exits the region affected by ground effect, the integral component of the controller causes the vehicle to undershoot the reference by around 10 cm. Only when the vehicle is commanded to hover at $x = 2$ m does the integrator wind down, correcting the position error in the process. With disturbance rejection, the disturbance feedforward term is able to compensate for the ground effect force disturbance, keeping the vehicle within 2.5 cm of the position reference at all times. No significant overshoots or undershoots are observed when the vehicle crosses the two table edges.

F. Wind Gust Disturbance Rejection

a) Experiment Setup: Next, we assess the robustness of the disturbance estimation and control system against strong, sudden wind gusts hitting the quadrotor in mid-air. For this purpose, we perform repeated vertical landings crossing a region affected by turbulent wind of different wind velocities produced by a high-powered fan. The wind velocities are measured using a digital anemometer in the center of the wind stream. The experiment setup is illustrated in Figure 1. We compare the disturbance rejection of the different control strategies by measuring the maximum deviations from the position reference. We also analyze the disturbance estimation performance. To enable a fair controller comparison, both the MPC and PID controllers were tuned to achieve an overdamped position step response with minimum rise time.
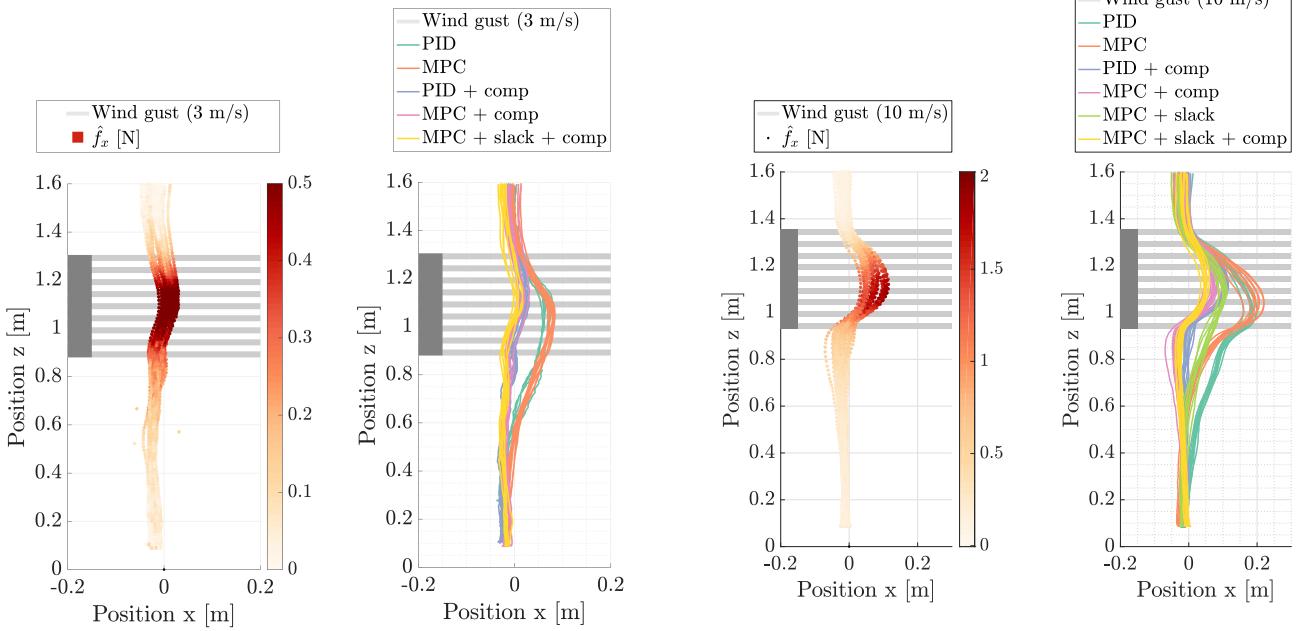
Fig. 12. 3 m/s wind gust experiment. *Left*: Disturbance force estimate in Z-X plane during runs with disturbance rejection. *Right*: Comparison of position error in Z-X plane during landing through strong wind gust for PID and MPC controllers, with and without disturbance rejection.

b) Results: Figure 12 shows the disturbance estimation (left) and rejection performance (right) of the tested controllers for a wind velocity of 3 m/s, corresponding to conditions likely to be encountered outdoors on a slightly windy day. *Without disturbance compensation*, both the PID and the MPC controllers experience an identical deviation from the reference path. *With disturbance compensation*, the wind disturbance is almost completely rejected. No clear difference in performance between the controllers can be observed.

To evaluate the estimation and control frameworks' performance in extreme conditions, we repeat the experiment with increased wind velocities. Figure 13 and 14 show the same type of results for a wind velocity of 10 m/s and 12 m/s respectively, which could be encountered while flying in heavy winds or in narrow urban landscapes. Table I summarizes the position errors and improvements to the baseline PID controller for the experiment with heavy winds. *Without disturbance compensation*, MPC performs similarly to PID but features an improved recovery once the vehicle

TABLE I

Comparison of maximum position error and improvement to baseline PID controller for strong wind gusts.

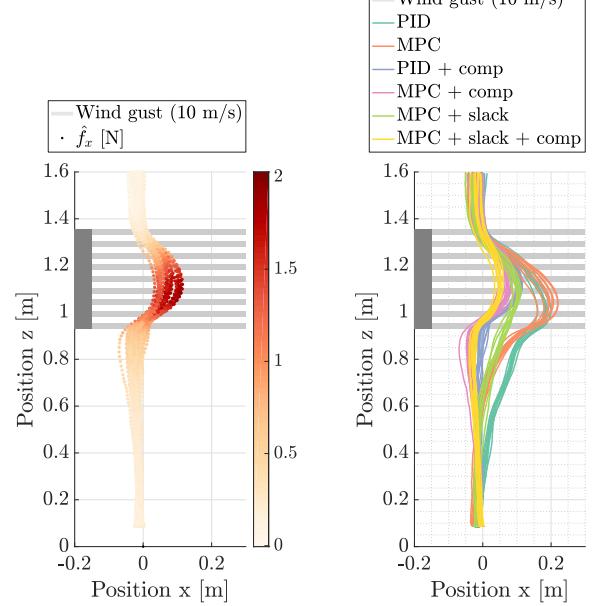| Wind velocity	10 m/s	12 m/s
Controller	Max. horizontal deviation [cm]	
PID (baseline)	17	45
MPC	19 (-12%)	39 (+13%)
MPC + slack	10 (+41%)	22 (+51%)
PID + comp	10 (+41%)	23 (+48%)
MPC + comp	8 (+53%)	17 (+62%)
MPC + slack + comp	5 (+71%)	10 (+78%)

Fig. 13. 10 m/s wind gust experiment. *Left*: Disturbance force estimate in Z-X plane during runs with disturbance rejection. *Right*: Comparison of position error in Z-X plane during landing through strong wind gust for PID and MPC controllers, with and without disturbance rejection.

exits the wind field. Activating a slack constraint of 5 cm on the cross-error further improves the MPC performance, resulting in a 41%-51% improvement to the PID baseline. *With disturbance compensation*, the disturbance rejection for both PID and MPC frameworks is significantly improved. However, the wind gust is not completely rejected. This is caused by the convergence time of the disturbance estimator. While increasing the variance of the additive Gaussian noise on the force estimate $\mathbf{v}_f$ decreases the convergence time, we found that the increased noise on the estimate translates to undesirable high-frequency roll and pitch oscillations. Hence, the requirement of smooth control imposes an upper bound on the magnitude of the additive noise variance, and thus the rejection performance. In both experiments, the best rejection performance is achieved by the slack-constrained MPC with disturbance compensation, featuring a considerable 71%-78% improvement over the baseline controller.

In the 12 m/s experiment, we observe that the MPCs with disturbance compensation control the vehicle to slow down the descent upon entering the wind gust. Due to the reduced penalty on the velocity error compared to the position error, the MPC prioritizes driving the position error to zero over maintaining a constant velocity. In case of very large disturbances, achieving both objectives is indeed not always possible due to motor saturation. This is a desirable property, since the vehicle first returns to its reference path before resuming forward flight.

VI. CONCLUSIONS

In an extensive experimental comparison, we have shown that augmenting MPC multirotor position controllers with a disturbance estimator leads to significantly improved rejection of disturbances such as ground effect and fast wind
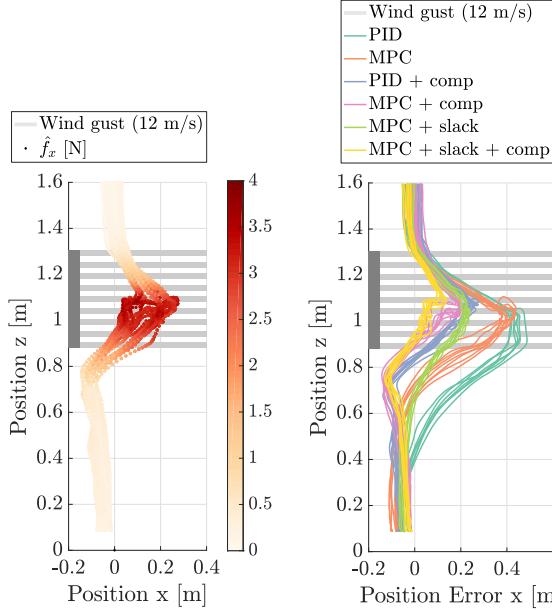
Fig. 14. 12 m/s wind gust experiment. *Left*: Disturbance force estimate in Z-X plane during runs with disturbance rejection. *Right*: Comparison of position error in Z-X plane during landing through strong wind gust for PID and MPC controllers, with and without disturbance rejection.

gusts. We demonstrated that the developed MPC (in particular, the soft-constrained MPC) achieves higher levels of position accuracy in the presence of large external disturbances compared to a PID controller. Combined with its ability to run at a fast rate on a computationally restricted microprocessor, this makes MPC an attractive control strategy for applications with a need for precise position control despite dynamic external disturbances. Nonetheless, the developed PID framework offers much improved disturbance rejection compared to a conventional PID baseline.

For multirotor disturbance estimation, we have shown that an EKF framework is preferable to an Unscented approach due to a 2.5-fold improved computational performance, reduced complexity and nearly identical estimation performance. We also demonstrated that the EKF disturbance estimator is suitable for deployment outdoors with degraded GPS-based state estimation.

An interesting direction for future work would be to combine model-based and data-driven approaches to learn to overcome disturbances, model inaccuracies or delays over time [8], [25]. Moreover, a direct comparison with robust and adaptive strategies would be desirable to benchmark position control accuracy subject to large external disturbances.

ACKNOWLEDGMENT

REFERENCES

[1] C. Brommer, D. Malyuta, D. Hentzen, and R. Brockers, "Long-Duration Autonomy for Small Rotorcraft UAS including Recharging," in *IEEE Intell. Robot. Syst.*, Madrid, Spain, 2018.

[2] S. Omari, P. Gohl, M. Burri, M. Achtelik, and R. Siegwart, "Visual industrial inspection using aerial robots," *Proc. 3rd Int. Conf. Appl. Robot. Power Ind. CARPI 2014*, no. 1, pp. 1–5, 2015.

[3] R. D'Andrea, "Can drones deliver?" *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 647–648, 2014.

[4] 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.

[5] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear H control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.

[6] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Robust Model Predictive Flight Control of Unmanned Rotorcrafts," *J. Intell. Robot. Syst.*, vol. 81, no. 3-4, pp. 443–469, 2016.

[7] M. Achtelik, T. Bierling, J. Wang, L. Höcht, and F. Holzapfel, "Adaptive Control of a Quadcopter in the Presence of large/complete Parameter Uncertainties," in *Infotech@aerosp. 2011*, St. Louis, 2011, pp. 1–15.

[8] K. Pereida, R. R. P. R. Duivenvoorden, and A. P. Schoellig, "High-Precision Trajectory Tracking in Changing Environments Through L 1 Adaptive Feedback and Iterative Learning," *IEEE Int. Conf. Robot. Autom.*, pp. 344–350, 2017.

[9] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, "Impedance Control of VTOL UAVs with a Momentum-based External Generalized Forces Estimator," in *IEEE Int. Conf. Robot. Autom.*, 2014, pp. 2093–2099.

[10] E. J. J. Smeur, G. de Croon, and Q. Chu, "Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection," *Control Eng. Pract.*, vol. 73, pp. 79–90, 2018.

[11] T. Tomic and S. Haddadin, "A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 4197–4204, 2014.

[12] S. J. Lee, S. Kim, K. H. Johansson, and H. J. Kim, "Robust Acceleration Control of a Hexarotor UAV with a Disturbance Observer," in *IEEE Conf. Decis. Control*, 2016, pp. 4166–4171.

[13] C. D. McKinnon and A. P. Schoellig, "Unscented external force and torque estimation for quadrotors," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 5651–5657, 2016.

[14] A. Tagliabue, M. Kamel, S. Verling, R. Siegwart, and J. Nieto, "Collaborative transportation using MAVs via passive force control," in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5766–5773.

[15] M. Faessler, D. Falanga, and D. Scaramuzza, "Thrust Mixing , Saturation , and Body-Rate Control for Accurate Aggressive Quadrotor Flight," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 476–482, 2017.

[16] T. Barfoot, J. R. Forbes, and P. T. Furgale, "Pose Estimation using Linearized Rotations and Quaternion Algebra," *Acta Astronaut.*, vol. 68, no. 1-2, pp. 101–112, 2011.

[17] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. New York, NY, USA: Wiley-Interscience, 2006.

[18] J. L. Crassidis and F. Markley, "Unscented Filtering for Spacecraft Attitude Estimation," *J. Guid. Control. Dyn.*, vol. 26, no. 4, pp. 536–542, 2003.

[19] J. Uhlmann, "Dynamic Map Building and Localization: New Theoretical Foundations," Ph.D. dissertation, University of Oxford, 1995.

[20] D. Brescianini, M. Hehn, and R. D'Andrea, "Nonlinear Quadrocopter Attitude Control," Tech. Rep., 2013.

[21] B. Houska, H. J. Ferreau, and M. Diehl, "{ACADO} {T}oolkit – {A}n {O}pen {S}ource {F}ramework for {A}utomatic {C}ontrol and {D}ynamic {O}ptimization," *Optim. Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[22] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, 2014.

[23] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Work. Open Source Softw.*, 2009.

[24] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham: Springer International Publishing, 2016, pp. 595–625.

[25] V. R. Desaraju and N. Michael, "Leveraging Experience for Computationally Efficient Adaptive Nonlinear Model Predictive Control," *Intl. Conf. Rob. Autom.*, pp. 5314–5320, 2017.