Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Cascaded Nonlinear MPC for Realtime Quadrotor Position Tracking ⋆

Jonas Schlagenhauf * **Peter Hofmeier** **
Thilo Bronnenmeyer ** **Reinhart Paelinck** ** **Moritz Diehl** *

* *Systems, Control and Optimization Laboratory, Department of Microsystems Engineering (IMTEK), University of Freiburg, Georges-Köhler-Allee 102, 79110 Freiburg, Germany. (e-mail: jonas.schlagenhauf@imtek.de, diehl@imtek.de).*
** *Kiteswarms GmbH, 79427 Eschbach, Germany. (email: peter@kiteswarms.com, thilo@kiteswarms.com, reinhart@kiteswarms.com).*

Abstract: In this paper we present a cascaded control approach using nonlinear model-predictive controllers for both stages. Using a quadrotor platform as an exemplary target plant with fast nonlinear dynamics, a realtime capable design is proposed that does not require the plant dynamics to exhibit clearly separable time constants as in classical cascaded control. In contrast to similar work we employ NMPC for the inner control loop instead of classical control approaches such as PID control, allowing to keep predictive properties as well as explicit constraint handling. We demonstrate via Monte-Carlo simulations that our design is able to achieve a significantly better position tracking performance of the quadrotor while being equally computationally expensive compared to monolithic position tracking NMPC.

Keywords: model predictive control, cascaded control, realtime control, modeling for control optimization, quadrotor control

1. INTRODUCTION

Modern control systems are increasingly integrated into applications that require accurate realtime control of a multitude of coupled variables under a wide variety of constraints. Advances in realtime optimal control promise highly-dynamic closed-loop behavior of complex systems, that were infeasible to control only a few decades back. Common examples are autonomous driving, power system control and aerospace applications.

While the sophistication of control approaches increases steadily, actual implementations suffer from increasingly unmaintainable complexity and thus development cost. Advanced control methods have worsened this trend rather than the opposite. Expensive computation, interdependent effects and couplings make it hard for control engineers to design a monolithic realtime controller for many complex use cases.

A solution to this was already applied as early as the 1960s in the form of cascading control: designing two separate controllers for specific aspects of the plant dynamics that achieve the overall control objective together. An inner control loop regulates fast-changing aspects of the plant whereas an outer control loop treats the inner loop as part of its plant dynamics, generating reference values for the inner loop. Cascading control shares the benefits with other distributed control approaches: modular structure, better controller design due to a tighter design focus for each module, shorter transmission delays, distributed computation loads and scalability.

Formulating a cascaded control structure is however a non-trivial task. While the underlying plant dynamics usually set the context for the appropriate structure, many design details are left to the control engineer, an aspect that is only amplified with the increased design freedom of modern control theory. Compared to a single-stage controller where the inputs and outputs are dictated by physical circumstances of the plant, the interface between the stages of cascaded controllers can be chosen much more freely. If the abstraction of the subsystems however neglects important cross-couplings, the local control performance can be heavily degraded by the actions of the other control loop. Optimal control approaches such as (nonlinear) model predictive control (NMPC) lend themselves to be applied in cascaded control systems due to their flexibility in formulation of inputs and outputs and explicit constraint handling compared to classic control methods. It is however a non-trivial task to properly translate explicit constraints and correct predictive behavior into a cascade structure, where extra care has to be taken to ensure consistency in constraint satisfaction and prediction by possibly conflicting behavior between the stages.

In this work we propose a cascaded control structure using nonlinear model predictive control for both stages, tackling the outlined problems of cascade structure design and constraint handling, which, to the best of the authors knowledge, has not been proposed before. Using a faster-sampling inner cascade makes it possible to outperform a comparable monolithic controller at the same or lower total computational cost. We challenge the assumption

⋆ This research has been funded by the company Kiteswarms in the KITE project (ZVK2017022302) with the University of Freiburg.

made for classical cascades having clearly separable time constants and introduce a resampling approach better suited for predictive cascades.
Using a quadrotor platform as an example target plant, we demonstrate via simulation that this approach produces a cascaded control structure able to perform accurate reference tracking for a highly dynamic, unstable system with higher tolerances against unmodeled disturbances compared to a monolithic NMPC implementation while keeping similar, realtime capable computation times. Using unit quaternion representations of rotations we allow arbitrary orientations without encountering singularities. The presented design is realized as a realtime-capable implementation exploiting on-the-fly linearization of the quadrotor rotation dynamics for speedups of the inner control loop.

2. RELATED WORK

Similar work has emerged predominantly in context where large-scale plants occur.
Ulbig et al. (2011) proposed a linear MPC cascade structure for managing different time scales of a power system where a monolithic optimal control approach would be infeasible.
A different kind of hierarchical structure was designed and realized by Touretzky and Baldea (2014). Here a slow, long-term scheduler was used to provide set points to an inner loop MPC with economic costs, applied to a building heating system.
In contrast, Vermillion et al. (2011) proposed a hierarchical MPC structure for a plant where the subsystems do not possess a strong time scale separation. On basis of this approach stability proofs were derived using a tank stirring problem as a benchmark problem, using the inner MPC to control the actuator dynamics.
Luchini et al. (2017) employed a similar strategy by using a mixed-integer-MPC formulation for interfacing the actuators of an ice box, having a slow, long-term outer loop MPC for generating cooling reference values.
Barcelli et al. (2011) proposed a decentralized hierarchy of local stabilizing controllers without explicit constraint handling, relying on the supervising MPC controllers to provide feasible reference values.
A cascaded NMPC approach for a simplified quadrotor platform was presented by Liu et al. (2014). A two-rotor, partially fixed, 3DoF setup was used as a target plant, performing real-world experiments while keeping the model dynamics complexity low.
Aiming for full quadrotor dynamics, Chen and Wang (2013) designed a dual MPC cascade with the goal of providing convenient set points for the inner control loop to employ linear control methods for faster control rates.
Nguyen et al. (2017) developed a control structure similar to the approach presented in this paper. Using a two-stage cascade (called "layers") consisting of a trajectory-tracking MPC and a torque-controlling MPC combined with a feedback linearization through CTC. They validate the trajectory tracking performance of their design in simulation in nominal and wind-disturbed conditions. The possibility of sampling the inner casade with a faster rate was however not addressed.
Neunert et al. (2016) presented a sequential-linear-quadratic (SLQ) MPC formulation to perform fast trajectory tracking in realtime. Two target plants, a quadrotor and a balancing robot, are used to demonstrate the stabilization and trajectory tracking performance. While the proposed approach relies on a lower cascade to apply the computed trajectories to the plant, the cascaded nature of the system is not specifically addressed or exploited. The formulation also does not treat plant constraints explicitly.
To improve the realtime capabilities of their quadrotor NMPC formulation, Zanelli et al. (2018) proposed a new RTI scheme variant and demonstrated the results by deploying the design on an embedded platform and performing real-world flight experiments.

3. QUADROTOR DYNAMICS

We use a conventional symmetric quadrotor platform as our target plant. Figure 1 shows the coordinate system definitions for the reference systems and actuators. Off-

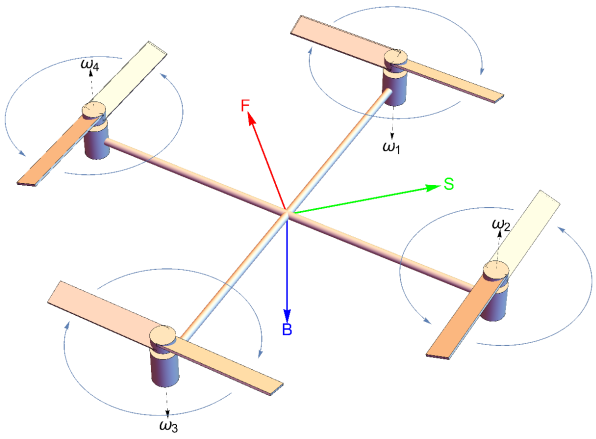

Fig. 1. Quadrotor coordinate system and motor axes conventions.

the-shelf motor drivers allow to set the propeller rotation speed, where we assume sufficiently fast timing characteristics of the driver and the motor compared to the quadrotor dynamics to neglect it in the quadrotor model.

3.1 Drive Map

Having a symmetric motor arrangement and orthogonal motor axes we can formulate a map from motor speeds $[\omega_0\ \omega_1\ \omega_2\ \omega_3]^\top$ to upward thrust F_{thrust} and 3-axes torque $\tau = [\tau_{\text{roll}}\ \tau_{\text{pitch}}\ \tau_{\text{yaw}}]^\top$ in the quadrotor frame as

$$\begin{bmatrix} F_{\text{thrust}} \\ \tau \end{bmatrix} = \underbrace{\begin{bmatrix} -a & -a & -a & -a \\ -b & -b & +b & +b \\ +b & -b & -b & +b \\ -c & +c & -c & +c \end{bmatrix}}_{M} \begin{bmatrix} \omega_0^2 \\ \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \end{bmatrix} \tag{1}$$

where we call M the *drive map*. The model parameters a, b, c are determined by the propeller lift and drag coefficients as well as the quadrotor arm length (see Table 1 for the actual values).

For this quadrotor configuration, the inverse M^{-1} is defined for positive model parameter values, providing us also with the inverse transformation

$$M^{-1}\left[F_{\text{thrust}}\ \tau_{\text{roll}}\ \tau_{\text{pitch}}\ \tau_{\text{yaw}}\right]^\top = \left[\omega_0^2\ \omega_1^2\ \omega_2^2\ \omega_3^2\right]^\top. \quad (2)$$

3.2 Rigid-body Dynamics

The drive map allows to set up simplified quadrotor dynamics as a rigid body in 3D with additional external forces and torques. The time-continuous model ODE then becomes

$$\frac{d}{dt}\begin{bmatrix} p \\ v \\ q \\ \Omega \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m} q \mathbb{F} \bar{q} + g \\ \frac{1}{2} q \begin{bmatrix} 0 \\ \Omega \end{bmatrix} \\ I^{-1}\tau \end{bmatrix} \quad (3)$$

where $p \in \mathbb{R}^3$ is the position, $v \in \mathbb{R}^3$ the velocity, $q \in SO(3)$ the unit quaternion representing orientation of the quadrotor frame w.r.t. the world frame and $\Omega \in \mathbb{R}^3$ the angular velocity of the quadrotor. We denote $\mathbb{F} = [0, 0, 0, -F_{\text{thrust}}]^\top$ the upward thrust in negative z-direction of the quadrotor frame, interpreted as an imaginary quaternion. Together with the torques $\tau \in \mathbb{R}^3$ they constitute the virtual control inputs to the plant. The gravitational acceleration $g \in \mathbb{R}$, the mass $m \in \mathbb{R}$ and the inertia tensor $I \in \mathbb{R}^3$ are model parameters that are empirically determined.

A natural plant separation becomes apparent based on the above model equations: rotation and translation dynamics are only coupled via the orientation of the forward thrust. Moreover, this coupling shows a hierarchy in the subsystems, since we first need to adjust the quadrotor orientation using the torque controls before the position can be changed via the upward thrust controls.

We therefore conveniently define the inner cascade as the attitude control loop and the outer cascade as the position control loop.

4. CASCADE STRUCTURE

Using the insight gained from the model analysis, we construct the cascade structure with two stages. The inner control loop (*attitude stage*) tracks the quadrotor's rotational dynamics by applying torques to the plant (Section 4.3). The outer control loop (*position stage*) tracks the quadrotor position and yaw by computing a feasible trajectory using forward thrust and torques, feeding reference orientations to the inner loop (Section 4.1). The position stage has full model knowledge to be able to generate feasible reference trajectories for the attitude stage.

In addition we use the control output of the position as control reference for the attitude stage. While the upward thrust does not enter the rotation dynamics of the attitude stage, this makes it possible to negotiate thrust vs. torque to obey actuator limits inside the attitude stage.

Since the sampling rates and number of prediction steps of position and attitude stage differ, a resampling step has to be performed when handing reference values down to the inner controller (Section 4.2). Using a forward simulation of the reference values on the time grid of the attitude stage the sample times of both stages are decoupled.

In the last step the attitude stage applies its computed upward thrust and torques to the plant.

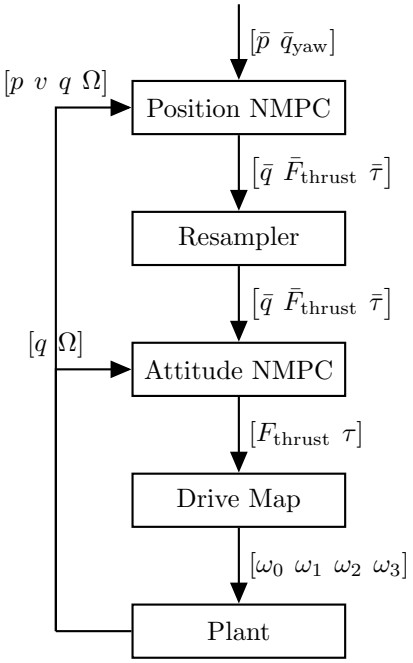

Fig. 2. Structure of the cascaded control system.

In this setup, we use the simulated internal plant state as full state feedback to the controllers. In a real-world setup this would be achieved by a suitable state estimator.

Note: For the sake of a concise formulation, mathematical symbols in the next two sections may be named identically. There is however no overlap except where denoted otherwise.

4.1 Position Stage

The outer control loop tracks the 3D position and yaw of the quadrotor. To ensure that we produce feasible reference values for the inner control loop the full quadrotor dynamics (equation 3) with states and controls

$$x = \begin{bmatrix} p \\ v \\ q \\ \Omega \end{bmatrix},\ u = \begin{bmatrix} F_{\text{thrust}} \\ \tau \end{bmatrix} \quad (4)$$

are utilized.

For every invocation of the position stage the optimal trajectory is computed as the result of the OCP

$$\min_{\substack{x_0,\dots,x_{N_\text{p}} \\ u_0,\dots,u_{N_\text{p}-1}}} \quad \sum_{k=0}^{N_\text{p}-1} l(x_k, \bar{x}_k, u_k, \bar{u}_k) + l_{N_\text{p}}(x_{N_\text{p}}) \quad (5a)$$

$$\text{s.t.} \quad x_0 = \bar{x}_0, \quad (5b)$$

$$x_{k+1} = f_\text{p}(x_k, u_k, h_\text{p}) \quad \forall k = 0, \dots, N_\text{p} - 1, \quad (5c)$$

$$0 \le M^{-1} u_k \le \omega_{\max}^2 \forall k = 0, \dots, N_\text{p} - 1 \quad (5d)$$

given the current state estimate $\bar{x}_0$ and a state-control reference trajectory $(\bar{x}_0, \dots, \bar{x}_{N_\text{p}}, \bar{u}_0, \dots, \bar{u}_{N_\text{p}-1})$.

Here, $f_\text{p}(x_k, u_k, h_p)$ is the result of discretizing Equation 3 with a fourth-order Runge-Kutta (RK4) integrator for a step size h_p, ensuring consistency between trajectory states and model prediction. Constraint (Equation 5b)

fixes the initial state x_0 to the current estimated state $\bar{x}_0$ as obtained from the estimator. To take the motor limits into account, we also constrain the total thrust and torques via the inverse motor map to the maximally possible motor values $\omega_{\max}$ (eq 5d).
The cost terms penalize the tracking error $x_k - \bar{x}_k$ and $u_k - \bar{u}_k$. Extra steps have to be taken for rotation differences represented as unit quaternions. We define the difference between two unit quaternions as

$$q_{\text{err}} = q_{\text{a}} - q_{\text{b}} := \text{vec}(q_{\text{a}}^{-1} q_{\text{b}}) \tag{6}$$

where q^{-1} denotes the conjugate quaternion and vec() returning the quaternion's imaginary part. This approximates the actual angular distance between the quaternions with sufficient accuracy without introducing nonlinearities. The costs function can then be formulated straightforward as

$$\begin{aligned} l(x_k, \bar{x}_k, u_k, \bar{u}_k) = &\begin{bmatrix} p_k - \bar{p}_k \\ q_k - \bar{q}_k \\ \Omega_k - \bar{\Omega}_k \end{bmatrix}^\top Q_{\text{p}} \begin{bmatrix} p_k - \bar{p}_k \\ q_k - \bar{q}_k \\ \Omega_k - \bar{\Omega}_k \end{bmatrix} \\ &+ \begin{bmatrix} F_k - \bar{F}_k \\ \tau_k - \bar{\tau}_k \end{bmatrix}^\top R_{\text{p}} \begin{bmatrix} F_k - \bar{F}_k \\ \tau_k - \bar{\tau}_k \end{bmatrix} \\ l_{N_{\text{p}}}(x_{N_{\text{p}}}) = &\Omega_{N_{\text{p}}}^\top Q_{\text{p,N}} \Omega_{N_{\text{p}}} \end{aligned} \tag{7}$$

with weight matrices $Q_{\text{p}} \in \mathbb{R}^{10\times10}$, $R_{\text{p}} \in \mathbb{R}^{4\times4}$ and $Q_{\text{p,N}} \in \mathbb{R}^{3\times3}$.

4.2 Resampling

Solving the above optimization problem yields a state-control-trajectory of length N_{p} sampled with step size h_p. To use the predicted orientation, angular velocity and controls as reference values for the inner control loop the trajectory has to be resampled onto the time grid of the inner control loop. Classical control approaches usually assume that the time scales of both stages are so far apart that the inner loop dynamics appear constant to the outer loop. There, zero-order-hold (ZOH) would suffice to obtain a sample. In this scenario and due to the predictive nature of the controllers assuming a constant reference between position stage samples would lead to large errors.
We therefore use the system model to simulate the reference trajectory on the time grid of the inner control loop instead, i.e. for every state-control-reference pair $\bar{w}_{\text{p},k} = (\bar{x}_{\text{p},k}, \bar{u}_{\text{p},k})$ obtained from the position stage we extrapolate new reference values $\bar{w}_{\text{p},k,j}$ via

$$\bar{w}_{\text{p},k,j} = \begin{cases} \bar{w}_{\text{p,k}} & j = 0 \\ f_{\text{p}}(\bar{w}_{\text{p},k,j-1}, h_{\text{a}}) & j \leq \left\lceil \frac{N_{\text{p}} \cdot h_{\text{p}}}{h_{\text{a}}} \right\rceil \end{cases} \tag{8}$$

which then are passed to the attitude stage (here, h_{a} is the prediction horizon length of the attitude stage).

4.3 Attitude Stage

The inner control loop tracks the reference orientation of the quadrotor that is handed down from the outer control loop. We therefore only consider the rotation dynamics of the plant:

$$\frac{dx}{dt} = \frac{d}{dt}\begin{bmatrix} q \\ \Omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} q \begin{bmatrix} 0 \\ \Omega \end{bmatrix} \\ I^{-1}\tau \end{bmatrix}, \; u = \begin{bmatrix} F_{\text{thrust}} \\ \tau \end{bmatrix}. \tag{9}$$

Since the torque controls compete with the total thrust for the resulting motor speeds, the thrust value computed by the position controller enters the inner control loop as a control reference. This allows the attitude stage to deviate from that reference thrust value if it would violate actuator constraints.
The attitude tracking problem is then formulated as

$$\min_{\substack{x_0,\dots,x_{N_{\text{a}}} \\ u_0,\dots,u_{N_{\text{a}}-1}}} \quad \sum_{k=0}^{N_{\text{a}}-1} l(x_k, \bar{x}_k, u_k, \bar{u}_k) \tag{10a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_0, \tag{10b}$$

$$x_{k+1} = f_{\text{a}}(x_k, u_k, h_{\text{a}}) \forall k = 0, \dots, N_{\text{a}} - 1, \tag{10c}$$

$$0 \leq M^{-1} u_k \leq \omega_{\max}^2 \qquad \forall k = 0, \dots, N_{\text{a}} - 1. \tag{10d}$$

Equivalent to the position OCP we constrain the initial value to the estimate obtained from the state estimator (Equation 10b) and limit the controls to physically feasible values. We however define $f_{\text{a}}(x_k, u_k, h_{\text{a}})$ as the resulting function of discretizing equation 9 via RK4 for a step size h_a. We define the cost function as

$$\begin{aligned} l(x_k, \bar{x}_k, u_k, \bar{u}_k) = &\left[q_k - \bar{q}_k\right]^\top Q_a \left[q_k - \bar{q}_k\right] \\ &+ \begin{bmatrix} F_k - \bar{F}_k \\ \tau_k - \bar{\tau}_k \end{bmatrix}^\top R_a \begin{bmatrix} F_k - \bar{F}_k \\ \tau_k - \bar{\tau}_k \end{bmatrix}. \end{aligned} \tag{11}$$

with weight matrices $Q_{\text{a}} \in \mathbb{R}^{4\times4}$ and $R_{\text{a}} \in \mathbb{R}^{4\times4}$. The above nonlinear program (NLP) is converted into a sequential quadratic program (SQP) by linearizing cost and constraint functions in every iteration of the inner control loop around the current system state. It is then solved using the realtime iteration (RTI) scheme ((Diehl et al., 2005), (Gros et al., 2016)). From the resulting state-control-trajectory the first element is then taken and applied to the plant.

5. IMPLEMENTATION

The presented control structure was implemented in Python3.6. CasADi (Andersson et al., 2018) was used for symbolic expressions and solver interfaces. The position OCP was solved using IPOPT (Wächter and Biegler, 2006) / MUMPS. The QPs arising from the linearized attitude control problem were solved with qpOASES (Ferreau et al., 2014).
A list of all default parameters and values used for the simulation can be found in Table 1. Deviations from these parameters for specific simulation scenarios are declared in the next section. The model parameters represent an existing physical quadrotor platform and were identified beforehand.

6. SIMULATION

To analyze the proposed cascaded structure qualitatively and quantitatively, we perform closed-loop simulation scenarios based on the dynamics described in Equation 3, that highlight the relevant performance criteria of a tracking control system. As a comparison reference the position stage without inner cascade is used, representing a single-stage monolithic NMPC approach.
A step-response scenario and a more complex rectangle trajectory tracking scenario are chosen to compare both

Table 1. Controller and model parameters used for simulation scenarios.

Parameter		Value
Position Stage		
Prediction Steps	N_p	30
Sampling Time	h_p	300 ms
State weight	Q_p	diag([1, 1, 100, 0, 0, 0, 1, 0.05, 0.05, 0.05])
Terminal weight	$Q_{\mathrm{p,N}}$	$\mathrm{diag}([1,1,1]) \cdot 10^{-1}$
Control weight	R_p	$\mathrm{diag}([1,1,1,1]) \cdot 10^{-5}$
Attitude Stage		
Prediction steps	N_a	5
Sampling time	h_a	2 ms
State weight	Q_a	$\mathrm{diag}([0,5,5,5])$
Control weight	R_a	$\mathrm{diag}([1,1,1,1]) \cdot 10^{-3}$
Model parameters		
Motor map param.s	a	$1.086 \times 10^{-5}\,\mathrm{N\,rad^{-2}\,s^{-2}}$
	b	$1.738 \times 10^{-6}\,\mathrm{N\,m\,rad^{-2}\,s^{-2}}$
	c	$1.515 \times 10^{-7}\,\mathrm{N\,m\,rad^{-2}\,s^{-2}}$
Max. motor speed	ω_max	$1000\,\mathrm{rad\,s^{-1}}$
Mass	m	1.5 kg
Inerta tensor	I	$\mathrm{diag}([0.014, 0.014, 0.024])\,\mathrm{kg\,m^2}$

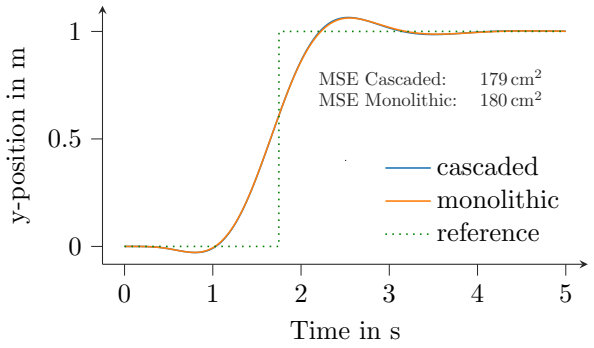

Fig. 3. Comparison between cascaded and monolithic controller step responses to verify identical behavior. The reference trajectory is represented by a dotted line.

candidates. Artificial external torque disturbances are further introduced to investigate and compare the control behavior under non-ideal conditions.
As error metric we compute the mean squared error (MSE) between two vectors a and b of length n as

$$\mathrm{MSE}(a,b) = \frac{1}{n}\sum_i (a_i - b_i)^2. \tag{12}$$

6.1 Step-response

We generate a reference trajectory that contains a one meter step along the y-position axis. To verify that both candidates exhibit the same optimal tracking behavior, we first simulate five seconds of closed-loop behavior with identical parameters in a disturbance-free scenario. As Figure 3 shows, the step-response behavior of the cascaded controller and the reference controller are almost identical with MSEs of 179 cm^2 and 180 cm^2 respectively.
The difference between both trajectories $p_\mathrm{centr} - p_\mathrm{casc}$ is shown in Figure 4. The difference does not exceed 1.5 cm / 1.5 % of the step height, which we determine to be small

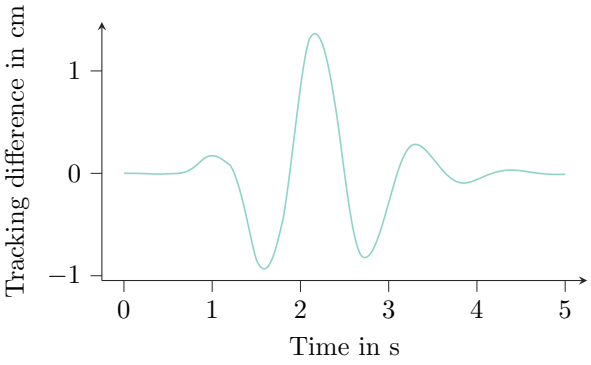

Fig. 4. Difference between cascaded and monolithic controller trajectories.

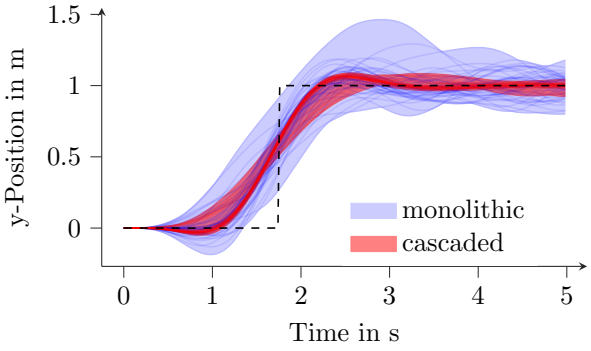

Fig. 5. Comparison of step tracking performance under random disturbances, showing 50 trajectories for each controller as well as the total covered range per time step displayed by colored areas.

enough to be disregarded in further simulations.
To highlight the performance difference of both candidates in non-ideal conditions, angular acceleration disturbances v sampled from a normal distribution

$$v \sim \mathcal{N}_3\left(\begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}\sigma & 0 & 0\\ 0 & \sigma & 0\\ 0 & 0 & \sigma\end{bmatrix}\right) \tag{13}$$

are added to the rotation dynamics of the simulation:

$$\frac{d}{dt}\Omega = I^{-1}\tau + v. \tag{14}$$

A Monte-Carlo simulation with 50 samples is performed for each controller where we choose $\sigma = 0.001\,\mathrm{rad\,s^{-1}}$. Figure 5 shows all resulting trajectories. A general qualitative difference in disturbance rejection performance can be seen, highlighting the expected reduction in disturbance rejection by the attitude stage compared to the single-stage controller.
Analyzing the torque control values as generated by the position stage and the attitude stage in Figure 6 gives more insight into the difference in control action by the respective control approaches.
An in-depth look into the behavior of both controllers for different disturbance levels is achieved by performing the above scenario for a range of external disturbance values. For the sake of clarity only the MSE and its distribution is shown in figure 7. As expected both controllers perform worse with increasing disturbances. For the presented disturbance values, the monolithic controller creates a worst-case median MSE of 380 cm^2, the cascaded approach

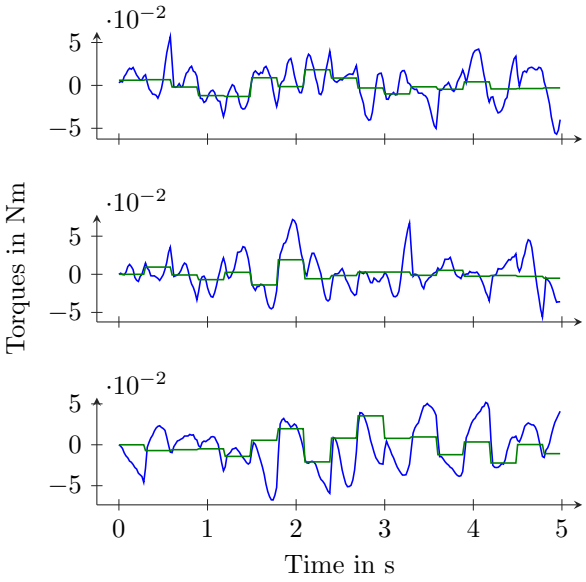

Fig. 6. Roll, pitch and yaw torque control values of position (green) and attitude (blue) stage for an exemplary step reference. The position stage torques serve as a reference value for the attitude stage torques.

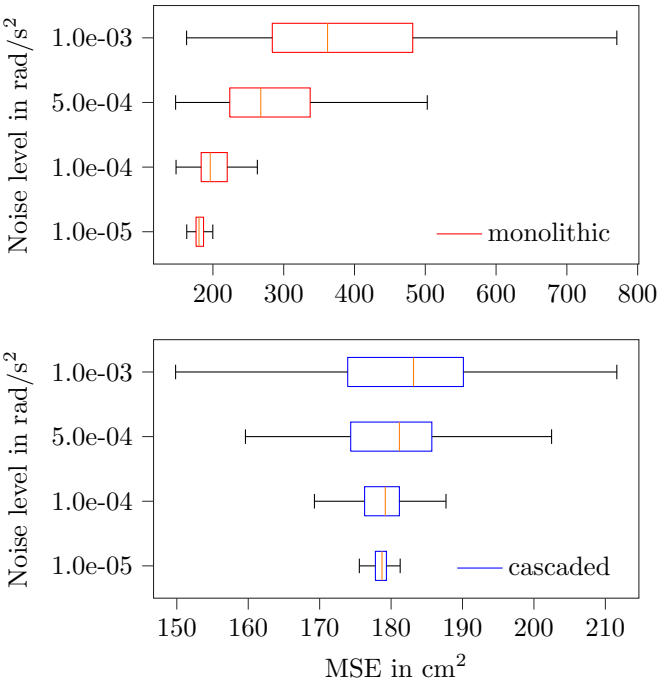

Fig. 7. Comparison of cascaded and monolithic controller for increasing angular acceleration disturbance values. The performance is represented as distributions of MSE values.

reaches a median MSE of 185 cm^2 in the same scenario.

To explore the importance of the chosen sampling times, a similar evaluation is performed by fixing the disturbance values to $1 \times 10^{-4}\,\mathrm{rad\,s^{-2}}$ and observing the step response behavior for a range of sampling rates of the position stage. Figure 8 shows the results of this simulation. Since shorter sampling times increase the possibilities of each controller to counteract unmodeled disturbances, a reduction in tracking errors can be observed with decreasing sampling time for both candidates. Since the attitude stage allows to compensate disturbances and return to the reference values in between position stage samples, we can see a distinct performance gain for the cascaded controller. We further investigate the computation time of both controllers in Table 2. Due to the RTI approach of the attitude stage, computation of the attitude control loop generally takes less than a millisecond. Since the monolithic controller is equivalent to the position stage of the cascaded controller, computation times are also almost identical. The high maximum computation times are the result of the first iteration, in which the OCP is initialized far from the converged solution.

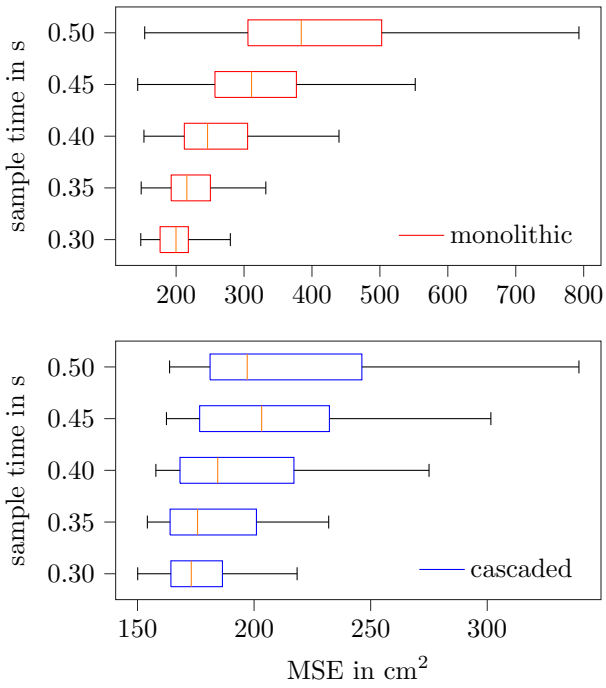

Fig. 8. Comparison of cascaded and monolithic controller for decreasing position stage sampling times. The performance is represented as distributions of MSE values.

Table 2. Computation times of one control loop iteration in milliseconds, based on 200 disturbance-free step-tracking simulations.

	median	Q1	Q3	min	max
Monolithic	32.959	29.812	40.220	28.251	570.317
Casc.-Pos.	32.555	28.614	39.924	28.058	452.575
Casc.-Att.	0.801	0.793	0.823	0.775	8.295

6.2 Rectangle trajectory

To demonstrate the control behavior of the cascaded system in a more practical scenario, we additionally perform reference tracking of a rectangle in 3D space. Starting from hovering, the quadrotor follows a 4-by-4-meter square shape in the Y-Z plane, returning to the starting point. Figure 9 shows the resulting trajectories of the cascaded controller and the reference.

A stable tracking performance is achieved, reaching a MSE of 251.01 cm^2.

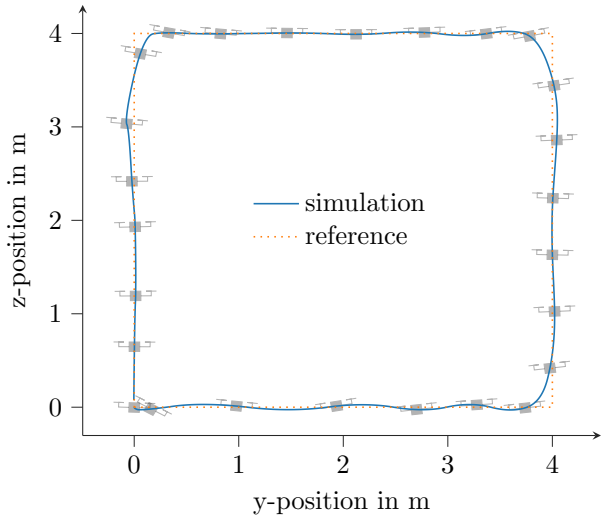

Fig. 9. Rectangle shape trajectory tracking simulation

7. CONCLUSION & OUTLOOK

In this paper we presented a new control approach applying nonlinear model-predictive control to a cascaded control structure. The process of creating this control structure was demonstrated on basis of a quadrotor model where the assumption of clearly separable time constants cannot be made using a position stage and an attitude stage, linked by a model-based resampling approach.
Based on Monte-Carlo-simulations we can show a clear benefit of this approach over a single-stage, monolithic nonlinear model-predictive controller. As the simulation results show, the disturbance rejection of the inner control loop leads to a drastically reduced tracking error without increasing the overall computation time. This also allows to increase the sampling time of the position stage, allowing longer prediction horizons or reduced computational cost.
Further simulation analysis and real-world experiments are required to fully explore the performance difference compared to a monolithic approach. Continuing this work a range of additional improvements are now viable such as a spatial distribution of the cascade stages and generalization to hierarchical control systems with multiple inner controllers.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2018). CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*.

Barcelli, D., Bemporad, A., and Ripaccioli, G. (2011). Decentralized hierarchical multi-rate control of constrained linear systems. *IFAC Proceedings Volumes*, 44(1), 277–283.

Chen, X. and Wang, L. (2013). Cascaded model predictive control of a quadrotor uav. In *2013 Australian Control Conference*, 354–359. IEEE.

Diehl, M., Bock, H.G., and Schlöder, J.P. (2005). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5), 1714–1736. doi:10.1137/S0363012902400713. URL `http://epubs.siam.org/sicon/resource/1/sjcodc/v43/i5/p1714\_s1`.

Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., and Diehl, M. (2014). qpOASES: a parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4), 327–363.

Gros, S., Zanon, M., Quirynen, R., Bemporad, A., and Diehl, M. (2016). From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of Control*.

Liu, H., Xi, J., and Zhong, Y. (2014). Robust hierarchical control of a laboratory helicopter. *Journal of The Franklin Institute*, 351(1), 259–276.

Luchini, E., Schirrer, A., and Kozek, M. (2017). A hierarchical mpc for multi-objective mixed-integer optimisation applied to redundant refrigeration circuits. *IFAC-PapersOnLine*, 50(1), 9058–9064.

Neunert, M., de Crousaz, C., Furrer, F., Kamel, M., Farshidian, F., Siegwart, R., and Buchli, J. (2016). Fast nonlinear model predictive control for unified trajectory optimization and tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1398–1404. doi:10.1109/ICRA.2016.7487274.

Nguyen, N.T., Prodan, I., and Lefevre, L. (2017). Multi-layer optimization-based control design for quadcopter trajectory tracking. In *2017 25th Mediterranean Conference on Control and Automation (MED)*, 601–606. IEEE.

Touretzky, C.R. and Baldea, M. (2014). Integrating scheduling and control for economic mpc of buildings with energy storage. *Journal of Process Control*, 24(8), 1292–1300.

Ulbig, A., Arnold, M., Chatzivasileiadis, S., and Andersson, G. (2011). Framework for multiple time-scale cascaded mpc application in power systems. *IFAC Proceedings Volumes*, 44(1), 10472–10480.

Vermillion, C., Menezes, A., and Kolmanovsky, I. (2011). Stable hierarchical model predictive control using an inner loop reference model. *IFAC Proceedings Volumes*, 44(1), 9278–9283.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.

Zanelli, A., Horn, G., Frison, G., and Diehl, M. (2018). Nonlinear model predictive control of a human-sized quadrotor. In *Proceedings of the European Control Conference (ECC)*, 1542–1547.