Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-14 (2022) 133–138

Nonlinear Control Method for Backflipping with Miniature Quadcopters ⋆

P. Antal * **T. Péni** * **R. Tóth** *

* *Systems and Control Laboratory, Institute for Computer Science and Control, H-1111 Bp. Kende u. 13-17. (e-mail: antalpeter@sztaki.hu, penitamas@sztaki.hu, tothroland@sztaki.hu).*

Abstract: The paper proposes a nonlinear control method for performing a backflip maneuver with a nano quadcopter. To perform the maneuver, first a feasible reference trajectory is designed that describes the intended state evolution. Then, the designed trajectory is precisely tracked by a nonlinear geometric controller that is able to track even highly challenging reference trajectories. The performance of the proposed method is evaluated and compared to a simple adaptive feedforward control strategy based on simulations and real-world experiments using Bitcraze Crazyflie nano quadcopters.

Keywords: aerial robotics, aerobatics, trajectory planning, geometric control, optimization

1. INTRODUCTION

The aim of this work is to develop and implement trajectory planning and motion control algorithms that allow a nano quadcopter to perform complex maneuvers at high speed. Many common tasks of a miniature quadcopter, such as navigating in a cluttered environment or flying in strong wind require to perform complex, fast maneuvers that push the drones to their physical limits (Loquercio et al., 2021). In these cases, classical flight controllers designed for a linearized dynamical model are no longer applicable and more advanced control methods that exploit the entire operating domain are needed (Lee et al., 2010). These algorithms can be developed based on nonlinear control techniques, or machine learning approaches.

The backflip maneuver has been chosen as an example, because it is a challenging task even for an expert human driver, and it emphasizes the complex nonlinear behaviour of the drone. The complexity and speed of the maneuver is characterized by the fact that it takes less than a second to complete, during which the vehicle is able to make a full turn around one of the horizontal axes.

In the literature, there are several different control strategies to perform the flip maneuver. In El-Badawy and Bakr (2016), energy-based control is applied to overcome the uncontrollability of the quadcopter at singular configurations to follow a circular or clothoidal reference trajectory. In Chen and Pérez-Arancibia (2017), Lyapunov-based controller synthesis is used to execute multi-flip maneuvers with quadcopters. Machine learning approaches are utilized in many cases, for example to imitate the maneuver performed by an expert drone pilot with apprenticeship learning in Abbeel et al. (2010), or design time-optimal trajectories with deep reinforcement learning in Song et al. (2021) and learn acrobatic maneuvers in Kaufmann et al. (2020); Hwangbo et al. (2017).

A simple learning strategy for adaptive feedforward control is proposed in Lupashin et al. (2010), based on the optimization of a parametric motion primitive sequence. As backflipping pushes the actuators of the quadcopter to their physical limits, the application of near-maximal and minimal control inputs are required. This approach builds on the theory of bang-bang control and first-principles motion primitive design to perform and optimize the flip maneuver. The proposed method is easy to implement and it is well suited for generating a feasible trajectory, however, many trials on the real robot are necessary to optimize the parameters of the motion.

The method we propose in this paper is based on geometric control, which is a nonlinear approach for attitude feedback control of rigid bodies in 3D space. In Lee et al. (2010), it is theoretically proven that geometric control is able to stabilize the orientation of a quadcopter in the whole operating domain based on differential geometric considerations and Lyapunov stability. The control law proposed in Lee et al. (2010) has been improved and extended by other researchers and it is the basis of several advanced trajectory design and agile maneuvering control algorithms, e.g. Turpin et al. (2012); Mellinger and Kumar (2011). However, in the literature we have not yet seen the application of geometric control together with systematic trajectory planning for backflipping with quadcopters.

The main contributions of our work are as follows:

(1) We propose an optimization-based trajectory planning method for the backflip maneuver. The designed reference trajectory is tracked by the nonlinear geometric control proposed in Lee et al. (2010).
(2) We compare the proposed method to a feedforward control approach introduced in Lupashin et al. (2010). We evaluate the performance of both methods in simulations and in real experiments, as well.

⋆ This research was supported by the Eötvös Loránd Research Network (grant. number: SA-77/2021) and by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program.

10.1016/j.ifacol.2022.07.595

This paper is structured as follows: firstly, we give an overview of quadrotor modelling and control in Section 2. Section 3 proposes an optimization-based trajectory planning method and introduces geometric control for the precise tracking of the reference trajectory. In Section 4, we compare the performance of the proposed method to an optimization-based feedforward control strategy via numerical simulations. In Section 5, we describe the experimental setup, and evaluate our results based on measurements on the real robot. Finally, the conclusions are summarized in Section 6.

2. QUADCOPTER DYNAMICS

The trajectory planning and motion control algorithms to be introduced later require the mathematical model of the quadcopter. In this section, we present the basic principles of quadcopter modelling and equations of motion, mainly based on Mahony et al. (2012).

Firstly, three main frames are introduced: the inertial frame $\mathcal{F}^i$ interpreted as NED (north-east-down) coordinates, the vehicle frame $\mathcal{F}^v$, and the body frame $\mathcal{F}^b$, which is fixed to the vehicle. The transformation from $\mathcal{F}^i$ to $\mathcal{F}^v$ is a translation, and from $\mathcal{F}^v$ to $\mathcal{F}^b$ a rotation, because the axes of $\mathcal{F}^v$ and $\mathcal{F}^i$ are parallel. In Fig. 1, the three frames are displayed with the Euler angles in the body frame (roll: ϕ, pitch: θ, yaw: ψ), and the direction of the rotor thrusts and angular velocities.

The translational dynamics of the quadcopter are characterized by

$$m\ddot{r} = R_{\mathrm{b}}^{\mathrm{v}} \begin{bmatrix} 0 \\ 0 \\ -F \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \tag{1}$$

where $r = [x, y, z]^\top$ is the position of the quadcopter in the inertial frame, m is the mass of the drone, F is the collective thrust of the propellers, and g is the gravitational acceleration. $R_{\mathrm{b}}^{\mathrm{v}} \in \mathrm{SO}(3)$ is the rotation matrix between the vehicle and body frames, where $\mathrm{SO}(3)$ denotes the three-dimensional special orthogonal group, also called the rotation group.

The rotational dynamics are described by Euler's equations, as

$$\dot{R}_{\mathrm{b}}^{\mathrm{v}} = R_{\mathrm{b}}^{\mathrm{v}} \hat{\omega}^{\mathrm{b}}, \tag{2a}$$

$$\dot{\omega}^{\mathrm{b}} = \left(J^{\mathrm{b}}\right)^{-1} \left(\tau - \omega^{\mathrm{b}} \times J^{\mathrm{b}} \omega^{\mathrm{b}}\right), \tag{2b}$$

where ω^{b} is the angular velocity of the vehicle in the body frame, J^{b} is the inertia matrix, and $\tau = [\tau_{\mathrm{x}}, \tau_{\mathrm{y}}, \tau_{\mathrm{z}}]^\top$ is the vector of torques produced by the propellers. The notion $\hat{\cdot}$ stands for the projection: $\mathbb{R}^3 \to \mathrm{SO}(3)$ ensuring that $\hat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$, where the $\times$ operator corresponds to the vector product of the operands. To simplify the notations, the indication of the coordinate frames is omitted in the rest of the paper and the following notations are introduced: $R = R_{\mathrm{b}}^{\mathrm{v}}, J = J^{\mathrm{b}}, \omega = \omega^{\mathrm{b}}$.

The dynamic model has four inputs, the collective thrust: F in (1), and the torques around the three axes of the body frame: τ in (2). These inputs can be calculated from the individual thrusts of the motors (T_i) as follows:

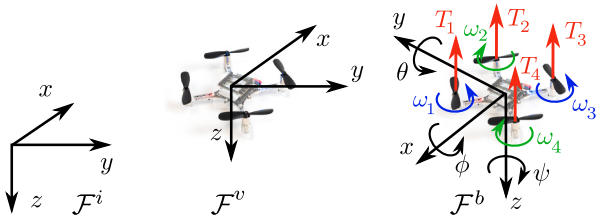

Fig. 1. Inertial, vehicle, and body frames describing the geometric relations of the vehicle and the environment. Thrusts and angular velocities of the rotors are also illustrated.

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -l & -l & l & l \\ l & -l & -l & l \\ \frac{b}{k} & -\frac{b}{k} & \frac{b}{k} & -\frac{b}{k} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}, \tag{3}$$

where l is the distance of two motors along the x axis, b is the drag constant, and k is the thrust constant. Furthermore, the thrust generated by each motor is proportional to the square of the corresponding angular velocity: $T_i = k\omega_i^2$ for $i \in \{1, 2, 3, 4\}$.

3. TRAJECTORY PLANNING AND GEOMETRIC TRACKING CONTROL

We implement backflipping as a 360 degree rotation around the y axis of the quadcopter's body frame, displayed in Fig. 1. The proposed approach for performing the maneuver is based on closed-loop control: first a feasible reference trajectory is designed for the flip, then the trajectory is given to a nonlinear controller that ensures the precise reference tracking. In this section, we first introduce Geometric control as a baseline control algorithm for trajectory tracking. Then the proposed optimization-based trajectory planning method is presented.

3.1 Geometric Tracking Control for Aggressive Maneuvers

The nonlinear geometric tracking control used in this work is based on the one presented in Lee et al. (2010) and Turpin et al. (2012). The control method is able to track reference position $r_{\mathrm{d}}(t) = [x_{\mathrm{d}}(t), y_{\mathrm{d}}(t), z_{\mathrm{d}}(t)]^\top$, and reference attitude $R_{\mathrm{d}}(t) \in \mathrm{SO}(3)$, represented by rotation matrices.

To synthesize the control law, we use (1) and (2) describing the dynamics of the quadcopter. Following the attitude control method proposed in Lee et al. (2010), the force and torque inputs are regulated as

$$F = (-K_{\mathrm{r}} e_{\mathrm{r}} - K_{\mathrm{v}} e_{\mathrm{v}} + mge_3 + m\ddot{r}_{\mathrm{d}}) R e_3, \tag{4a}$$

$$\tau = -K_{\mathrm{R}} e_{\mathrm{R}} - K_{\omega} e_{\omega} + \omega \times J\omega, \tag{4b}$$

with diagonal gain matrices $K_{\mathrm{r}}, K_{\mathrm{v}}, K_{\mathrm{R}}, K_{\omega} \in \mathbb{R}^{3\times3}$, and error terms

$$e_{\mathrm{r}} = r - r_{\mathrm{d}}, \tag{5a}$$

$$e_{\mathrm{v}} = \dot{r} - \dot{r}_{\mathrm{d}}, \tag{5b}$$

$$e_{\mathrm{R}} = \frac{1}{2\sqrt{1 + \mathrm{tr}\left(R_{\mathrm{d}}^\top R\right)}} \left(R_{\mathrm{d}}^\top R - R^\top R_{\mathrm{d}}\right)^\vee, \tag{5c}$$

$$e_{\omega} = \omega - R^\top R_{\mathrm{d}} \omega_{\mathrm{d}}, \tag{5d}$$

where r_d, R_d and ω_d are the position, orientation and angular velocity reference, $\mathrm{tr}(\cdot)$ is the trace operator, and the *vee operator* $(\cdot)^\vee$ is the inverse of the hat operator such that $(\cdot)^\vee : \mathrm{SO}(3) \to \mathbb{R}^3$. With control gains selected from the stable domain, the proposed attitude control approach is proved to be stable in the full space of rotation matrices (excluding exact inversion), as derived in Lee et al. (2010).

3.2 Trajectory Planning for the Flip Maneuver

In this section, we use the geometric tracking controller (4)-(5) to perform the flip maneuver. For this we first design a suitable attitude reference trajectory R_d, and then a position trajectory r_d. The objective of trajectory planning is that the quadcopter should arrive as close to the starting point as possible, while keeping the control inputs within the allowed range during the maneuver.

The attitude reference is specified in unit quaternions: $q_\mathrm{d} = [q_{0\mathrm{d}}, q_{1\mathrm{d}}, q_{2\mathrm{d}}, q_{3\mathrm{d}}]^\top$, where $q_{0\mathrm{d}}$ is the scalar part of the quaternion, and $q_{2\mathrm{d}}$ corresponds to the pitch angle, as $q_{1\mathrm{d}} = q_{3\mathrm{d}} = 0$, because both the roll and yaw angles are zero during the flip. Utilizing that q_d is a unit quaternion, we can express the third element of it as $q_{2\mathrm{d}} = \sqrt{1 - q_{0\mathrm{d}}^2}$, hence it is sufficient to design a trajectory only for q_{0d}.

A 360 degree rotation around the y axis means that the scalar part of the attitude quaternion goes from 1 to -1. In the trajectory design it is important to stay within the $q_{0\mathrm{d}} \in [-1, 1]$ range, because only unit quaternions describe rotation. We have chosen a smooth sigmoid function

$$q_{0\mathrm{d}} = \frac{2}{1 + e^{-v_\mathrm{m}\left(t - \frac{t_\mathrm{m}}{2}\right)}} - 1 \tag{6}$$

to describe the scalar part of the reference attitude, where the parameters are the speed of the maneuver v_m and the execution time t_m. The attitude quaternion reference trajectory is displayed in Fig. 2. Assuming that $\phi \equiv \psi \equiv 0$ during the flip, the conversion to Euler angles yields $\theta = 2\,\mathrm{acos}(q_{0\mathrm{d}})$, where $\theta \in [-\pi, \pi]$. Hence the pitch angle goes smoothly from zero to π, jumps to $-\pi$, and goes smoothly to zero.

Besides of rotation, the maneuver also requires translational motion, because without proper lifting at the beginning of the backflip, the quadcopter would fall to the ground due to gravity. The position reference is designed considering that the rotational and translational equations of the dynamical model are coupled. The translational motion of the flip maneuver is within the $x-z$ plane, therefore $y_\mathrm{d}(t) = 0$. The other two equations of the translational dynamics in (1) are

$$m\ddot{x} = -FR_{13}, \tag{7a}$$
$$m\ddot{z} = -FR_{33} + mg, \tag{7b}$$

where R_{ij} denotes the (i,j)-th entry of the rotation matrix R. However, assuming that the attitude tracking converges fast enough to the reference, we can substitute the reference rotation matrix in (7), resulting in the translational state space representation

$$\dot{\xi} = A\xi + Bu,$$
$$\xi = \begin{bmatrix} x \\ \dot{x} \\ \tilde{z} \\ \dot{\tilde{z}} \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ R_{\mathrm{d},13}/m \\ 0 \\ R_{\mathrm{d},33}/m \end{bmatrix}, \tag{8}$$

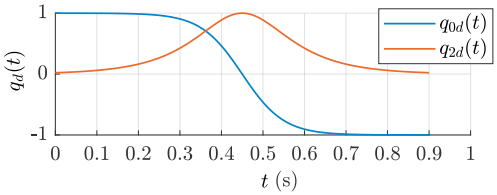

Fig. 2. Attitude quaternion reference trajectory for the backflip maneuver with $v_\mathrm{m} = 20$ 1/s, $t_\mathrm{m} = 0.9$ s.

where ξ is the state vector, $R_{\mathrm{d},ij}$ are the corresponding elements of the reference rotation matrix R_d (converted from the reference quaternion q_d), and A, B are the state space matrices. As the equations are decoupled, the effect of gravity can be added to the z position after a simulation, thus in the equation $\tilde{z}$ denotes the modified state. Notice that (8) is a linear state space representation with the thrust force $u = F$ as the only control input. By discretizing the system, a quadratic programming problem can be formulated over a finite horizon, similarly to model predictive control. We calculate the discrete time state space model using complete, zero-order hold discretization, resulting in the form

$$\xi_{k+1} = A_k\xi_k + B_k u_k,$$
$$A_k = \begin{bmatrix} 1 & T_\mathrm{s} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_\mathrm{s} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_k = \begin{bmatrix} R_{\mathrm{d},13}T_\mathrm{s}^2/(2m) \\ R_{\mathrm{d},13}T_\mathrm{s}/m \\ R_{\mathrm{d},33}T_\mathrm{s}^2/(2m) \\ R_{\mathrm{d},33}T_\mathrm{s}/m \end{bmatrix}, \tag{9}$$

where A_k, B_k are the discrete state space matrices. The input of the model is the collective thrust of the propellers, $u_k = F_k$. For a fixed duration of the maneuver with N discrete time steps, the following quadratic optimization problem is formulated:

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \sum_{k=1}^{N} \left[(\xi_k - \xi_{\mathrm{d},k})^\top Q_k (\xi_k - \xi_{\mathrm{d},k}) + u_k^\top W_k u_k \right] \\ \text{subject to} \quad & \xi_{k+1} = A_k\xi_k + B_k u_k, \\ & \{\xi_k\}_{k=1}^N \in \mathcal{X}, \\ & \{u_k\}_{k=0}^N \in \mathcal{U}, \end{aligned} \tag{10}$$

where $Q_k \in \mathbb{R}^{4\times4}$ and $W_k \in \mathbb{R}$ are weight matrices, and $\mathcal{X}$,$\mathcal{U}$ are the sets of constraints for the states and the control input, respectively. The only objective of the trajectory design is to minimize the final position error of the quadcopter and keep the position within a specified range, therefore the weight matrices are $W_k = 0, Q_k = 0$ for $k = 1 \dots N$, except for the weight of the final state that is $Q_N = \mathrm{diag}(1, 0, 1, 0)$. As all the other weights are zero, it is only required to define a final state position reference $\xi_{\mathrm{d},N}$, the components of which are zero except for the effect of the gravity in $\tilde{z}_{\mathrm{d},N} = 0.5g(T_\mathrm{s}N)^2$.

We specify linear constraints for the states: $x \in [x_-, x_+]$, $z \in [z_-, z_+]$ to model the available space for the maneuver avoiding collisions with other objects or walls. We also define linear constraints for the control input, namely

$$\frac{\|\tau_k\|}{l} \le u_k = F_k \le F_\mathrm{max} - \frac{\|\tau_k\|}{l}, \tag{11}$$

where τ_k is the vector of the three torques around the three body axes, out of which $\tau_x = \tau_z = 0$ normally during the flip, l is the distance of the quadcopter center of mass and

the propellers projected to the $x-z$ plane, and $F_{\max}$ is the maximal collective thrust of the rotors. The torque control input τ_k is calculated from the reference attitude R_{d} based on (4b) assuming that the orientation and angular velocity errors are zero.

The numerical solution of the optimization problem (10) can be obtained easily by using an off-the-shelf QP solver, e.g. by `quadprog` in Matlab. Finally, we fit cubic splines on the discrete points of the optimized reference trajectory, which the quadcopter follows with geometric tracking control, based on the control law (4) and error terms (5).

4. SIMULATIONS

The simulations are based on the dynamic model of a Bitcraze Crazyflie 2.1 miniature quadcopter which we use for demonstrating the experimental results in Section 5, as well. The nonlinear equations of motion are defined in Section 2, and the physical parameters of the drone are shown in Table 1, obtained from Förster (2015).

Our simulation framework is based on an OpenAI Gym environment with PyBullet physics engine, written in Python language (Panerati et al., 2021). All of the simulation code used in this work is available on our GitHub [1], and a video illustrating the simulation results is available at `https://youtu.be/AhqfXZ-CPqM`. In this section and the oncoming sections, we display the measurement results with the z axis pointing upwards (in contrast to the NED convention discussed in Section 2), because the backflip maneuver is more illustrative this way.

The simulation of the flip starts with hovering for about 0.1 s, followed by executing the maneuver, and then switching back to hovering around the initial position. Based on the physical properties of the quadcopter, we have chosen the parameters of the reference pitch trajectory to $v_{\mathrm{m}} = 20$ 1/s, $t_{\mathrm{m}} = 0.9$ s, as illustrated in Fig. 2. The quadratic programming problem (10) is solved under the following constraints:

$$\mathcal{X}: \begin{bmatrix} x_- \\ x_+ \\ z_- \\ z_+ \end{bmatrix} = \begin{bmatrix} -0.6 \\ 0 \\ -0.05 \\ 0.45 \end{bmatrix} \text{ m}; \quad \mathcal{U}: \ F \in [0, 0.64] \text{ N},$$

with sampling time $T_{\mathrm{s}} = 1/480$ s. The maximal collective thrust $F_{\max} = 0.64$ N is from Förster (2015), and the position bounds are chosen such that the trajectory is feasible, and the quadcopter does not get too far from the initial point, for example we can express the available flying space here to avoid collision with walls or other objects. The duration of the flip is $t_{\mathrm{m}} = 0.9$ s, thus the number of simulation steps is $N = t_{\mathrm{m}}/T_{\mathrm{s}} = 432$. The quadratic optimization problem is solved by the `quadprog` Matlab function within milliseconds of computation time.

The controller gains in (4) have been determined based on the implementation of geometric control in the official Crazyflie 2.1 firmware [2]. The numerical values are $K_{\mathrm{r}} = \mathrm{diag}(0.5, 0.5, 1.25)$, $K_{\mathrm{v}} = \mathrm{diag}(0.2, 0.2, 0.8)$, $K_{\mathrm{R}} = 0.08 I_{33}$, and $K_{\omega} = 0.002 I_{33}$, where I_{33} is the 3×3 identity matrix. In Lee et al. (2010), the stable regions of

[1] `https://github.com/AIMotionLab-SZTAKI/aimotion-crazypack`
[2] `https://www.bitcraze.io/documentation/repository/`

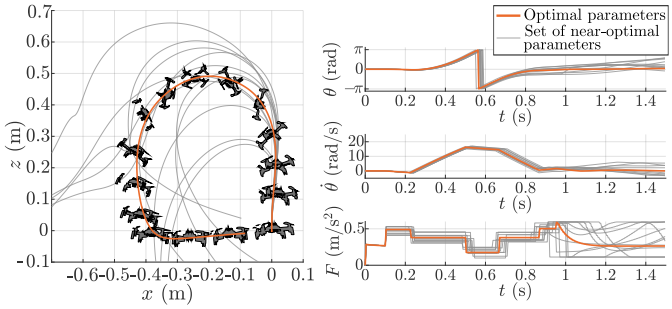

Fig. 3. Backflipping in simulation by feedforward control. The position is displayed on the left, and the pitch angle θ, pitch angular velocity $\dot{\theta}$, and collective thrust F on the right. Orange lines represent the simulation with optimal parameters, and grey lines represent the result of small changes in the parameter set.

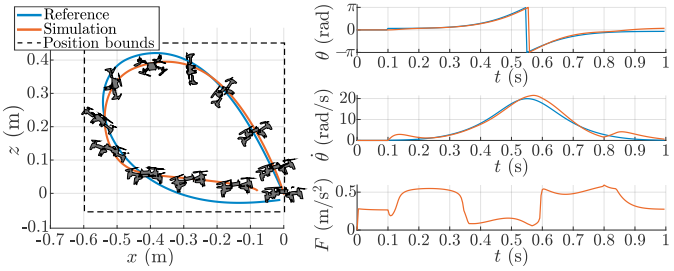

Fig. 4. Backflipping in simulation by geometric control. Position and attitude are depicted on the left, the pitch angle θ, pitch angular velocity $\dot{\theta}$, and collective thrust F on the right.

control gains are given for scalar values. Here, we practically use scalar gains (multiplied by identity matrices), except for an increased gain for the z position and velocity errors to compensate the effect of uncertain gravitational force. However, all elements of the diagonal gain matrices satisfy the stability conditions detailed in Lee et al. (2010).

We compare the results of our method to an adaptive open-loop control strategy for quadcopter backflipping which we have implemented based on Lupashin et al. (2010). The backflip maneuver is performed by optimizing the parameters of a motion primitive sequence, and applying feedforward control. The number of optimization variables is reduced by utilizing bang-bang control, i.e. using near-minimal and near-maximal control inputs. In contrast of the method proposed in Lupashin et al. (2010), we use Bayesian optimization (instead of gradient descent with gradient approximation) to find the parameters of the backflip motion primitive sequence (Shahriari et al., 2016).

Table 1. Physical parameters of a Crazyflie 2.1 quadcopter.

Mass	m	0.028 g
Propeller-to-propeller length	l	92 mm
Diagonal inertia elements	J_{xx}	$1.4 \cdot 10^{-5}$ kgm^2
	J_{yy}	$1.4 \cdot 10^{-5}$ kgm^2
	J_{zz}	$2.17 \cdot 10^{-5}$ kgm^2
Thrust coefficient	k	$2.88 \cdot 10^{-8}$ Ns^2
Drag coefficient	b	$7.24 \cdot 10^{-10}$ Nms^2

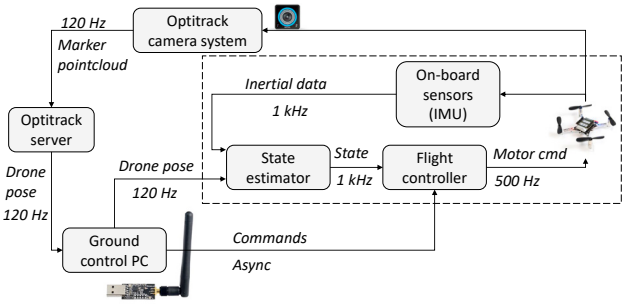

Fig. 5. Block diagram of the experimental setup: indoor quadcopter navigation with internal and external measurement system.

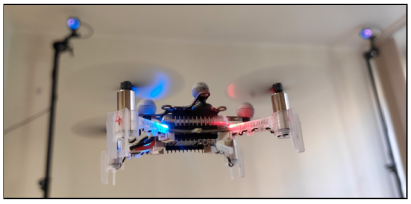

Fig. 6. Experimental setup: OptiTrack Prime 13 infrared cameras and a Crazyflie 2.1 quadcopter with reflective markers.

Simulation results using the feedforward control are displayed in Fig. 3, using the optimal parameter vector and a set of near-optimal parameters, as well. On the left plot, the position of the quadcopter during the flip is shown, with snapshots from the simulation. The end of the optimal maneuver is around the coordinate $(x, z) = (-0.4, 0)$ m with near-zero pitch angle, thus the final state error is only significant in the x position. From that point, a PID controller (Panerati et al., 2021) stabilizes the drone and controls to the origin. On the right, the trajectory of the pitch angle in Euler representation, the angular velocity, and the collective thrust as a control input are shown. The figure illustrates that even small deviations from the optimal parameter set ($< 10\%$) result in significantly decreased performance.

The simulation results of trajectory planning and reference tracking with geometric control are displayed in Fig. 4. The left plot illustrates the reference and simulated pose of the quadcopter during the backflip maneuver, and the right plot contains the trajectory of the pitch angle, angular velocity and collective thrust control input. The trajectory of both the angular velocity and the thrust input are smooth compared to the discontinuous angular acceleration and thrust of the feedforward control. At the discontinuities of the control input, the unmodeled transient behaviour of the actuator dynamics can be significant, therefore the geometric control approach is more robust to such uncertainties than the feedforward method.

5. EXPERIMENTS

The experimental setup consists of the Crazyflie drone, the Optitrack motion capture system (Optitrack image processing server and infrared cameras), and a ground control PC. The block diagram presenting the interconnection of the components is shown in Fig. 5. The quadrotor

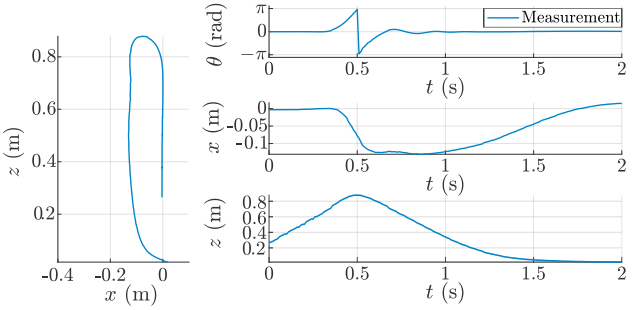

Fig. 7. Backflipping measurement results with feedforward control. The position and pitch angle of the quadcopter are displayed.

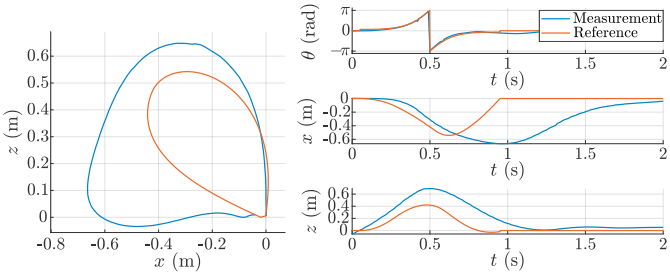

Fig. 8. Backflipping measurement results with geometric control. The trajectories show that the maneuver is performed successfully, and the drone gets back to the initial position at the end.

is equipped with an IMU containing a 3D accelerometer, gyroscope, magnetometer and barometer, and it has two microcontrollers: a STM32F405 for running the flight controller, and a nRF51822 for radio communication and power management. The drone weighs 28 grams, and the propeller-to-propeller distance is 92 millimeters. The quadcopter runs the original Bitcraze firmware, while on the server the Crazyswarm software platform is used to ease the implementation and configuration of high-level control components (Preiss et al., 2017). Optitrack is a high precision motion capture system with submillimeter resolution. We use it to obtain precise pose measurement of the drones in real time.

Firstly, we evaluate the results of performing the backflip with optimization-based feedforward control. Due to the differences of the simulation model and the real quadcopter dynamics, the parameters of the motion primitive sequence used in simulation were re-tuned so that the flip is executed with minimal final state error. This clearly demonstrates the sensitivity of this method to model uncertainties.

The measurement results are displayed in Fig. 7, showing the position and orientation of the quadcopter during the maneuver. It is important to note that an additional lift phase is added to the implementation to gain enough vertical velocity and height, because the quadcopter falls a significant distance in the recovery phase. During the additional lift phase, a PID controller [1] is used to achieve exact vertical lifting and horizontal orientation. Fig. 7 shows that the flip is executed with small final error in the pitch, and also quite small position error. However, it is important to note that the performance of the feedforward

[1] `https://www.bitcraze.io/documentation/repository/`

controller is very sensitive to uncertainties in the dynamics and initial conditions. For example, if the flip maneuver begins when the orientation of the quadcopter is not exactly horizontal, the stability can be lost at the recovery phase. Hence the maneuver is only successful in about 6-7 trials out of 10 using the feedforward controller.

The experimental results of backflipping with geometric control are displayed in Fig. 8. The most important part of reference tracking is the pitch angle θ, because a fast, stable and accurate attitude tracking is required to perform the flip maneuver, and recover successfully. As it is shown on the right plot of the measurement results, the pitch is very close to the reference, and the position also converges with a higher delay. In spite of the imperfect position tracking, the geometric controller is able to perform the backflip maneuver exactly the same way ten out of ten times, which indicates that it is significantly more robust than the feedforward method. Even with smaller changes on the dynamic behaviour of the quadcopter (e.g. changing the inertia by placing a larger reflective marker on the vehicle), the maneuver is performed successfully and the drone remains stable.

6. CONCLUSION

The proposed trajectory planning method proved to be successful, the quadcopter performed the backflip maneuver both in simulation and real-world experiments. An important conclusion is that the feedforward controller is very sensitive to parameter uncertainties and initial conditions, therefore it needs specific tuning for each Crazyflie. Geometric tracking control overcomes this problem, providing a highly robust and consistent performance for the backflipping. Utilizing the robustness of the control approach, the maneuver has been implemented for simultaneous backflipping with three drones, a video of which is available at `https://youtu.be/AhqfXZ-CPqM`.

The proposed trajectory planning method includes model-based optimization, therefore it is straightforward to apply for other types of quadcopters (e.g. medium or large-sized), only the physical parameters need to be adjusted. The introduced geometric tracking control is applied not only for different types of quadcopters in Turpin et al. (2012), but also for other autonomous systems, such as robotic manipulators in Bullo and Lewis (2004). Hence the proposed motion planning and control algorithms could be used in industrial applications, as well.

In our oncoming research work, we intend to use learning methods to perform complex maneuvers with less expert knowledge and extend the capabilities of the miniature quadcopters even more.

REFERENCES

Abbeel, P., Coates, A., and Ng, A.Y. (2010). Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13), 1608–1639.

Bullo, F. and Lewis, A. (2004). *Geometric Control of Mechanical Systems. Modeling, Analysis, and Design for Simple Mechanical Control Systems.* Springer-Verlag.

Chen, Y. and Pérez-Arancibia, N.O. (2017). Lyapunov-based controller synthesis and stability analysis for the execution of high-speed multi-flip quadrotor maneuvers. In *Proc. of the American Control Conference*, 3599–3606.

El-Badawy, A. and Bakr, M. (2016). Quadcopter aggressive maneuvers along singular configurations: An energy-quaternion based approach. *Journal of Control Science and Engineering*, 2016.

Förster, J. (2015). *System Identification of the Crazyflie 2.0 Nano Quadrocopter.* Master's thesis, ETH Zurich, Zurich.

Hwangbo, J., Sa, I., Siegwart, R., and Hutter, M. (2017). Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4), 2096–2103.

Kaufmann, E., Loquercio, A., Ranftl, R., Mueller, M., Koltun, V., and Scaramuzza, D. (2020). Deep drone acrobatics. *Robotics: Science and Systems.*

Lee, T., Leok, M., and McClamroch, N.H. (2010). Geometric tracking control of a quadrotor UAV on SE(3). In *Proc. of the 49th IEEE Conference on Decision and Control*, 5420–5425.

Loquercio, A., Kaufmann, E., Ranftl, R., Müller, M., Koltun, V., and Scaramuzza, D. (2021). Learning high-speed flight in the wild. *Science Robotics*, 6(59), eabg5810.

Lupashin, S., Schöllig, A., Sherback, M., and D'Andrea, R. (2010). A simple learning strategy for high-speed quadrocopter multi-flips. In *Proc. of the IEEE International Conference on Robotics and Automation*, 1642–1648.

Mahony, R., Kumar, V., and Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3), 20–32.

Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *Proc. of the IEEE International Conference on Robotics and Automation*, 2520–2525.

Panerati, J., Zheng, H., Zhou, S., Xu, J., Prorok, A., and Schoellig, A.P. (2021). Learning to fly–a Gym environment with PyBullet Physics for reinforcement learning of multi-agent quadcopter control. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7512–7519.

Preiss, J.A., Hönig, W., Sukhatme, G.S., and Ayanian, N. (2017). Crazyswarm: A large nano-quadcopter swarm. In *Proc. of the International Conference on Robotics and Automation*, 3299–3304. IEEE.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., and de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.

Song, Y., Steinweg, M., Kaufmann, E., and Scaramuzza, D. (2021). Autonomous drone racing with deep reinforcement learning. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1205–1212.

Turpin, M., Michael, N., and Kumar, V.R. (2012). Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33, 143–156.