Onboard Time-Optimal Control for Tiny Quadcopters

Westenberger, J.M.; de Wagter, C.; de Croon, G.C.H.E.

Publication date
2021

Document Version
Final published version

Published in
Proceedings of the 12th International Micro Air Vehicle Conference

Citation (APA)
Westenberger, J. M., de Wagter, C., & de Croon, G. C. H. E. (2021). Onboard Time-Optimal Control for Tiny Quadcopters. In J. Martinez-Carranza (Ed.), *Proceedings of the 12th International Micro Air Vehicle Conference* (pp. 93-100). Article IMAV2021-11 http://www.imavs.org/papers/2021/11.pdf

Onboard Time-Optimal Control for Tiny Quadcopters

Jelle Westenberger[1], Christophe De Wagter[1] and Guido C.H.E. de Croon[1] *

ABSTRACT

Time-optimal model predictive control is important for achieving fast racing drones but is computationally intensive and thereby rarely used onboard small quadcopters with limited computational resources. In this work, we simplify the optimal control problem (OCP) of the position loop for several maneuvers by exploiting the fact that the solution resembles a so-called 'bang-bang' in the critical direction, where only the switching time needs to be found. The non-critical direction uses a 'minimum effort' approach. The control parameters are obtained by means of bisection search schemes on an analytical path prediction model. The approach is compared with a classical PID controller and theoretical time-optimal trajectories in simulations. We explain the effects of the OCP simplifications and introduce a method of mitigating one of these effects. Finally, we have implemented the 'bang-bang' controller as a model predictive controller (MPC) onboard a Parrot Bebop and performed indoor flights to compare the controller's performance to a PID controller. We show that the light novel controller outperforms the PID controller in waypoint-to-waypoint flight while requiring only minimal knowledge of the quadcopter's dynamics.

1 INTRODUCTION

Unmanned air vehicles (UAV) are used in an increasing variety of applications [1]. Several applications, such as emergency response or race tasks require the drones to fly as fast as they can. Autonomous drone racing has recently emerged as a discipline to boost the development of fast-flying robots [2–4].

Traditionally the problem of time-optimal control generation is solved offboard as available hardware lacks the computational performance to quickly solve the Optimal Control Problem (OCP) onboard a quadcopter [5]. Fast flight is achieved by tracking these trajectories with high-performance controllers [6].

Recent work demonstrated efficient trajectory optimization for snap and leveraging differential flatness to derive the corresponding control inputs [7, 8]. However, the snap optimization method does not optimize for minimum-time. In fact, the total flight time must be predefined and the dynamical limits of the quadcopter are not taken into account. Including time and dynamic feasibility constraints in the optimization process increases the computational complexity of the problem [9]. On the other hand, [10] defines a sequential quadratic programming problem to simultaneously optimize control inputs for action and perception objectives. Albeit that in this work the reference trajectories are precomputed. [11] has extended on this work and demonstrated a pipeline that is fully embedded and is efficient enough to be implemented as a robust MPC. While the results are great, this comes at a very high computational cost. To address this, optimal control has also been approximated with deep neural nets, which are lighter than the original optimization [12–14]. This approach is powerful but very data intensive. Model predictive control remains very computationally expensive and few onboard implementations exist for very light drones [11, 15]. In this category, classical control remains common [16].

For a lot of trajectories, the time-optimal solution simplifies to a well-timed maximal control deflection. This paper therefore, proposes a light strategy to approximate time-optimal control by computing this timing onboard (See Figure 1).

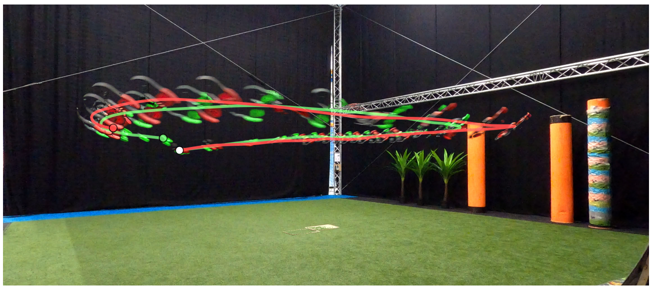

Figure 1: A comparison of a circular flight path between the proposed controller (green) and a classical PID controller (red)

Section 2 shows that the time-optimal position control simplifies to a bang-bang action on the attitude under well-selected conditions. In Section 3 we derive the differential equations that drive the proposed light MPC controller. Simulation results are presented in Section 4. Section 5 augments the model for the latency in attitude. Section 6 shows the re-

*[1]All authors are with the Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands
`jellewestenberger@gmail.com`
`c.dewagter@tudelft.nl`
`g.c.h.e.decroon@tudelft.nl`

sults obtained onboard a Parrot Bebop before Section 7 gives the conclusions.

2 Simplified Time-Optimal Control

We have simplified the OCP by assuming a constant altitude and using the fact that for second-order systems the time-optimal solution consists of a 'bang-bang' motion. For quadcopter position control this translates to a double step in either pitch or roll with maximal amplitude. The OCP is hereby reduced to a problem in which the only parameter to-be optimized is the switching time, which reduces the computational complexity of the problem sufficiently to even allow implementation onboard very small quadcopters. The collective thrust is governed by the constant altitude assumption and is therefore considered to be always equal to $W/\cos\theta\cos\phi$, where W is the weight, θ and ϕ are the pitch and roll angles, respectively. The lateral dynamics can then be further simplified.

2.1 *Proving bang-bang solution for constant angles*

Hehn, Ritz, and D'Andrea [5] show in their work that with Pontryagin's minimum principle it can be proven that the time-optimal solution for a two-dimensional quadcopter trajectory consists of a bang-bang input in thrust and bang-singular-bang in rotational rate. We show that this solution remains valid in our simplified OCP in which we neglect the rotational dynamics, to further reduce the computational expense. Continuing with the Hamiltonian from [5]:

$$H(\mathbf{x},\mathbf{u},\mathbf{p}) = 1+p_1\dot{x}+p_2u_T\sin\theta+p_3\dot{z}+p_4(u_T\cos\theta-1)+p_5u_R \tag{1}$$

where u_T is the thrust input and u_R is the rotation rate input. p_i are the costates. Our model assumes instantaneous attitude changes and no changes in altitude. Therefore we can discard the last three terms of equation 1 and change the state θ to input u_θ. Pontryagin's minimum principle states that the optimal control input $\mathbf{u}^*$ minimizes the Hamiltonian [17].

$$\mathbf{u}^* = \operatorname{argmin} p_2u_T\sin u_\theta \tag{2}$$

Depending on the sign of p_2, u_θ^* is either $\pm 0.5\pi$ or singular when $p_2 = 0$. However, at these pitch angles, it would be impossible to maintain altitude. Therefore, maximum pitch and roll angles are determined based on the thrust-to-weight performance of the quadcopter while reserving a margin of available thrust for additional control.

2.2 *Minimum-effort Approach*

We apply the 'bang-bang' solution to one critical axis. Intuitively, this axis is selected to be the direction with the largest initial position error. Control of the direction perpendicular to this axis is based on a 'minimum effort' approach. The intuition behind this approach is to only spend the minimum required thrust on decreasing the position error in the non-critical dimension such that a maximum available thrust can be spent on the critical dimension. This is achieved by calculating the constant attitude for which the non-critical position target is reached at the same time the critical target is reached.

3 Bang-Bang MPC

Based on the simplified OCP, we have created a controller that calculates the optimal roll and pitch angle from path predictions. We refer to this pipeline as the 'bang-bang' controller.

3.1 *Path Prediction*

For the sake of computational efficiency, we have simplified the dynamics such that the quadcopter's position and velocity can be evaluated analytically. By discarding the rotational and vertical dynamics, and partially decoupling the longitudinal and lateral dynamics we have derived a set 2nd order differential equations to describe the quadcopter's position and velocity.

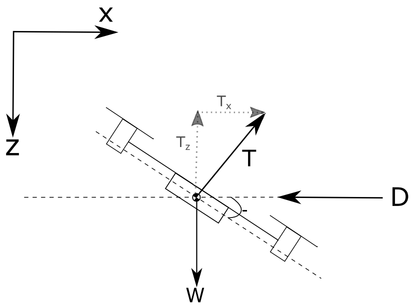

Figure 2: 2-D Quadcopter Dynamics

Based on the aforementioned assumptions and the force diagram depicted in Fig. 2 we state that the pitch angle θ and the thrust are constant and T_z equals the weight W. Furthermore, we assume that drag force D, consists only of flapping drag and is linearly proportional to airspeed $\dot{x}$, which is governed by drag coefficient C_d. So we can write:

$$\ddot{x} = g\tan\theta - \frac{C_d}{m}\dot{x} \tag{3}$$

Where g and m are the gravitational acceleration and mass, respectively.

Equation 3 is a 2nd order, non-homogeneous equation and is easily solved with the characteristic equation and method of undetermined coefficients. This yields:

$$x = c_1 e^{\frac{-C_d}{m}t} + c2 + \frac{W\tan\theta}{C_d}t \tag{4a}$$

$$\dot{x} = c_1\frac{-C_d}{m}e^{\frac{-C_d}{m}t} + \frac{W\tan\theta}{C_d} \tag{4b}$$

Constants c_1 and c_2 are solved with the quadcopter's initial position x_0 and initial velocity $\dot{x}_0$. This procedure can be repeated for the lateral direction, which is the direction out-of-plane in Fig. 2, taking into account the proper Euler angle rotations when deriving the lateral component of the thrust force:

$$y = c_3 e^{\frac{-C_d}{m}t} + c_4 + \frac{W}{\cos\theta}\frac{\tan\phi}{C_d}t \tag{5a}$$

$$\dot{y} = c_3 \frac{-C_d}{m} e^{\frac{-C_d}{m}t} + \frac{W}{\cos\theta}\frac{\tan\phi}{C_d} \tag{5b}$$

c_3 and c_4 are solved with the quadcopter's initial lateral position y_0 and lateral velocity $\dot{y}_0$.

Figure 3 illustrates an example of a single path prediction and the corresponding pitch and roll angles. The bang-bang maneuver of the longitudinal path can be described by evaluating equation 4 for two segments; One segment up to the switching instant (the acceleration phase) and a segment up to the time of arrival (the braking phase). Only the constants and pitch angles change between the two sets. The final velocity and position of the first segments are used as initial conditions for the second segment.
The lateral path can be described by one segment because in the 'minimum effort' approach the roll angle is assumed to be constant for the entire maneuver up to the target.

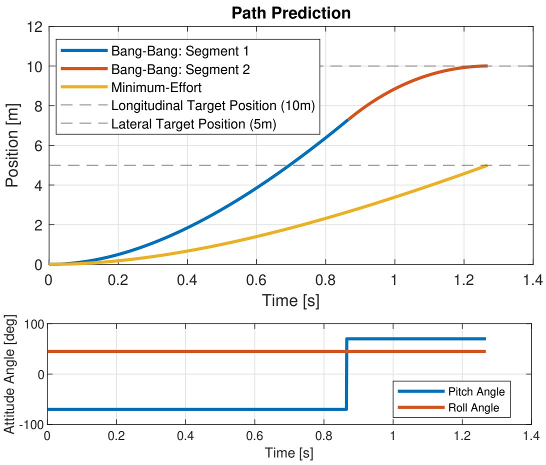

Figure 3: Example of predicted longitudinal and lateral paths. The longitudinal direction takes on a bang-bang motion that consists of two segments with opposing constant pitch angles. The lateral direction takes a minimum-effort approach which consists of a single segments and a constant roll angle.

3.2 Optimizing

In the OCP, the switching time and the non-critical angle are the two parameters that are to be optimized. Thanks to the analytical nature of the path equation a fast iterative bisection scheme can be used to find the optimal switching time and angle.

3.2.1 Solving Switching Time

To solve the switching time a desired velocity at the position target must be given in advance. The bisection scheme then iteratively adapts the switching time to minimize the velocity error at the target position. This procedure is described in Algorithm 1. In addition to an optimized switching time, an estimated time of arrival (ETA) is given as well. This is used in optimizing the non-critical angle.

Algorithm 1

```
$t_0 \leftarrow 0$
$t_1 \leftarrow$ initial guess
$E_t \leftarrow$ error threshold
$y_d \leftarrow$ desired position
while $E > E_T$ do
    $t_s \leftarrow \frac{t_0+t_1}{2}$
    $t_t \leftarrow get\_time\_from\_desired\_speed(v_d)$
    $E \leftarrow get\_position(t_t) - y_d$
    if $E > 0$ then
        $t_1 \leftarrow t_s$
    else
        $t_0 \leftarrow t_s$
    end if
end while
```

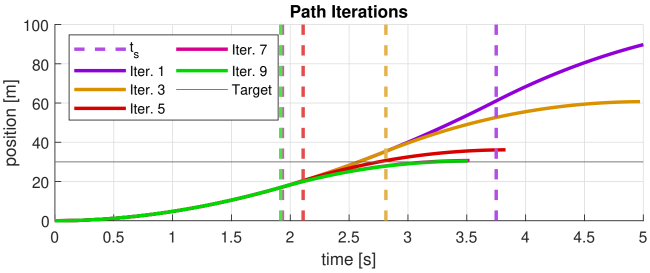

Figure 4: Illustration of the switching time, t_s, optimization process. The goal is to reach the target at 30m with zero rest speed.

3.2.2 Solving Minimum-Effort Angle

Analogously to the critical direction, a bisection scheme is used to iteratively change the angle to minimize the non-critical position error at the estimated time of arrival. The goal for the quadcopter is to reach the critical and non-critical targets simultaneously.
Moreover, during the braking phase of the 'bang-bang' motion, the related angle will also be optimized in this fashion to correct for prediction inaccuracies in this phase.

4 SIMULATIONS

Simulations have been performed to compare the 'bang-bang' controller flight performance to a classical PID controller and to time and snap optimized trajectories, provided

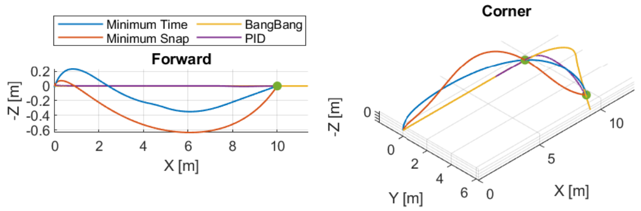

Figure 5: Trajectory comparison in simulation

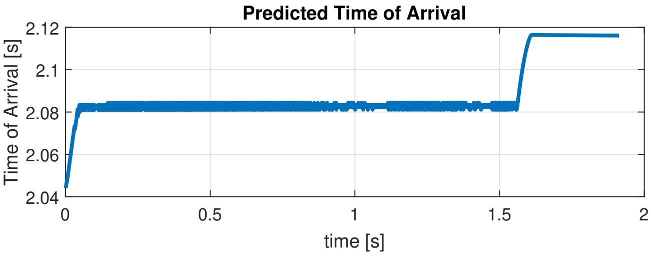

Figure 6: Development of the ETA of a simulated flight, corrected for passed simulation time. For perfect predictions, the time of arrival would be constant.

by the well-known ICLOCS toolbox [18]. This toolbox uses direct collocation to optimize a nonlinear OCP from an initial guess. Two maneuvers, a straight and a cornered trajectory, have been simulated to individually test the longitudinal and lateral flight behavior. The resulting flight times are summarized in table 1.

	Forward	Corner
Bang-Bang	1.52 s	2.53 s
PID	2.51 s	4.16 s
Min. Snap	1.77 s	2.53 s
Min. Time	1.30 s	1.87 s

Table 1: Simulated flight times

It can be seen that the bang-bang controller outperforms the PID controller in all maneuvers and is on par with the snap-optimized solution, but at a fraction of the computational cost.

5 Transition Compensation

It was found in simulations that the instantaneous angle assumption of the path predictor has the largest negative effect on the performance of the ‘bang-bang’ controller. Since a quadcopter cannot achieve infinitely high rotation rates the second part (further called the braking phase) of the bang-bang maneuver will always be initiated too late. As Figure 6 illustrates, path predictions deviate during the rotation from rest to acceleration at 0 s, and during the transition from accelerating to braking around 1.6 s.

In order to mitigate this issue, we have implemented a method that approximates the elapsed time and change of speed and position during the transition. Subsequently, the initial conditions of the braking phase are augmented with these values to improve the path predictions, as figure 7 illustrates. Figure 8 shows the effect of different degrees of compensation for a simulated flight.

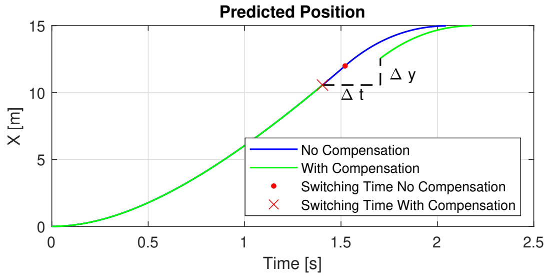

Figure 7: The initial conditions of the braking phase are shifted with $\Delta t, \Delta y$ and Δv to compensate for the rotation dynamics during the transition from accelerating to braking.

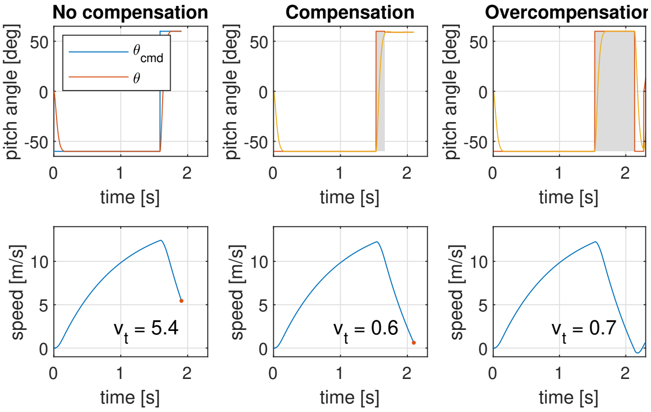

Figure 8: Different degrees of transition compensation are illustrated here. The shaded area indicates the period at which braking is forced. The desired target speed v_t is 0 m/s.

6 Experiments

Flight tests have been performed with a commercial Parrot Bebop quadcopter in which all software was changed. The performance of the ‘bang-bang’ MPC is compared to a traditional PID controller for different types of maneuvers.

6.1 *Experimental Setup*

The ‘bang-bang’ controller has been implemented in the open-source autopilot framework Paparazzi-UAV [19] and is executed onboard a Parrot Bebop quadcopter. The flights were performed in TU Delft’s ’CyberZoo’ indoor flight area outfitted with an Optitrack position and attitude tracking system. The position and heading are sent to the drone via WiFi and the state estimation is executed on board by means of the inertial measurement unit and complementary filters. The 'bang-bang' MPC and PID controller give roll and pitch commands while the inner control loop, based on Incremental Nonlinear Dynamic Inversion (INDI) [20], controls the rotational rates. Figure 9 gives an overview of the control pipeline.

6.1.1 PID Controller

The PID controller is a high-gain cascaded position-velocity controller. That is, the position error will govern the desired speed, which in turn governs the pitch and roll commands. A single set of gain values has been selected that gives the best overall result in all tests. This set is kept constant throughout all flight experiments. Furthermore, we have defined saturation limits for the allowable speed and pitch and roll angles. For a fair comparison, the same limits have also been applied to the 'bang-bang' controller.

6.2 Transition Estimators

As discussed in section 5, we can compensate for the unmodeled transition dynamics by approximating the transition losses. These dynamics are difficult and costly to simulate for a real quadcopter, therefore we have derived a simple linear regression model to approximate transition losses $\Delta t, \Delta y$ and Δv from flight measurements. We assumed that these losses are a function of the speed at the switching time, and of the total angle the quadcopter needs to rotate. We found that the transition losses varied between forward, backward and sideways flight maneuvers. Therefore, three different sets of estimators have been derived, each corresponding with one of these directions. In the control pipeline, one set of estimators is selected based on the direction in which a 'bang-bang' maneuver is planned.

6.3 Motion Primitives Flights

Test flights have been performed to test the bang-bang controller for different motion primitives. That is, six different maneuvers have been established to test the longitudinal and lateral performance in which the quadcopter starts and ends at rest. For each maneuver, a comparison is made between the 'bang-bang' controller, the 'bang-bang' controller with transition compensation, and the high-gain PID controller.

During these maneuvers the heading is kept constant and the critical direction for which the 'bang-bang' maneuver is planned is based on the largest component of the initial position error.

Table 2 lists all maneuvers and their initial and target positions. The controllers are assessed on the time it takes to reach their target, the degree of overshoot, and the velocity error while passing the target.

Maneuver	Initial → Target Position (x,y,z) [m]
Forward	$(-2,0,1.5) \rightarrow (2.5,0,1.5)$
Backward	$(-2,0,1.5) \rightarrow (2.5,0,1.5)$
Sideways	$(0,-2,1.5) \rightarrow (0,2,1.5)$
Forward-Sideways	$(-2,-2,1.5) \rightarrow (2,1,1.5)$
Forward-Up	$(-2,0,1) \rightarrow (2.5,0,2.75)$
Forward-Down	$(-2,0,2.75) \rightarrow (2.5,0,1)$

Table 2: Translated distances for each motion primitive maneuver.

6.3.1 Results

Table 3 summarizes the flight results. It can be seen that the non-compensated bang-bang controller has the largest velocity errors and overshoot. Furthermore, it becomes obvious that the compensation system has a positive effect on the path prediction performance. Unfortunately, the transition loss model is not accurate enough to completely mitigate the transition losses and some degree of overshoot still occurs. In these simple start-stop tests, the PID controller is marginally slower than both bang-bang controllers but has lower overshoot and velocity errors.

6.4 Consecutive Waypoints Flight

To test the proposed controller in a setting that more closely resembles an autonomous drone race, a flight plan with consecutive positional waypoints has been implemented. In this flight plan, the quadcopter is no longer instructed to come to a full stop at each waypoint. The desired speed at each waypoint has been set to 2 m/s as it was found iteratively that this value in combination with a position threshold of 70 cm resulted in smooth trajectories for both the PID and bang-bang controllers (See Figure 1). However, it is expected that the optimal threshold values are controller- and trajectory-dependent.

Because currently no heading changes were incorporated into the 'bang-bang' maneuver planning, the heading is kept constant. The critical direction in which a 'bang-bang' motion is planned is automatically adjusted based on the direction with the largest position error.

6.4.1 Results

Figure 10 shows top views of flights with the two controllers. Both controllers have been assessed on the time it takes to complete one circle and the minimum position error. The results are displayed in figure 11. Here, the 'bang-bang' controller is seen to outperform the PID both in speed and target accuracy. The PID controller is unable to give priority to one direction over the other. Due to the high gain values, roll and pitch angles are quickly saturated even if the position error of one direction is much smaller than the other, which slows down the critical axis. In the various flight runs of the PID controller, large lateral oscillations can be seen. It was also

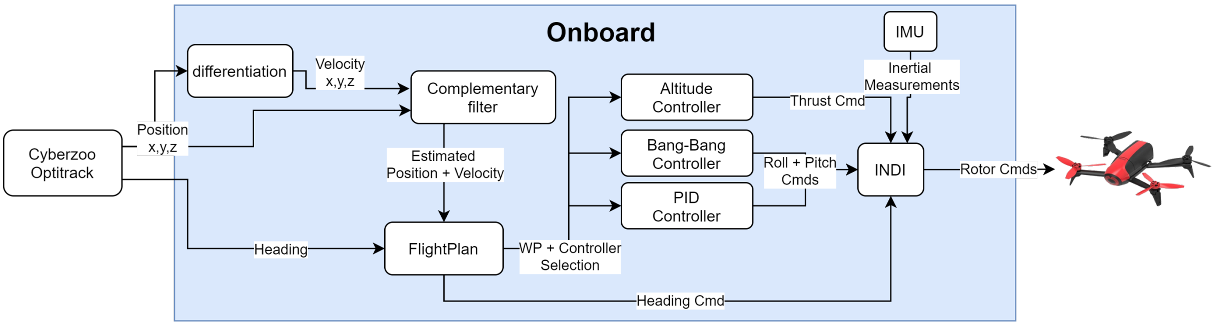

Figure 9: Control Pipeline

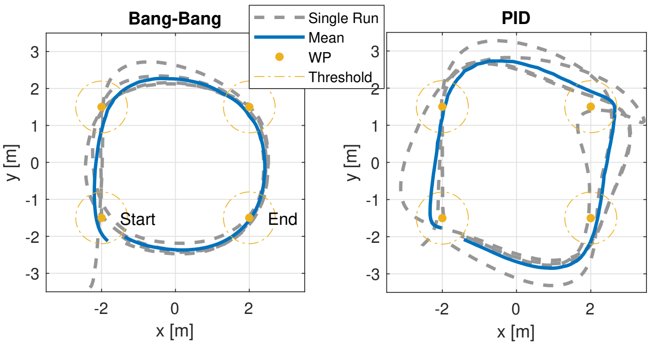

Figure 10: Top-view of consecutive waypoints flights

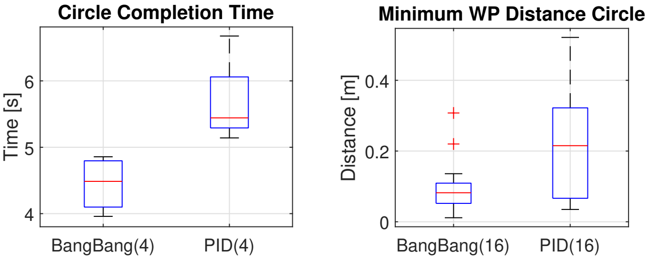

Figure 11: Circle completion time and minimal waypoint distance for consecutive waypoint flights

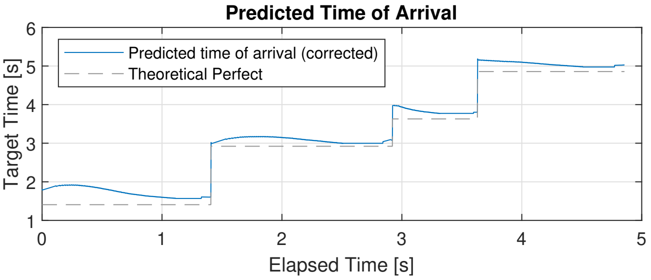

Figure 12: Estimated time of arrival corrected for passed time.

found that the PID controller was more likely to reach unstable situations due to the high simultaneous pitch and roll angles compared to the bang-bang controller.

The predicted times of arrival for a single run are illustrated in Figure 12. From this figure, we can derive the real-time path prediction performance. As expected the time increases during the angular rotations. However, during the acceleration phases, the time is seen to decrease. We think that this is caused by inaccurate aerodynamic drag estimation and by the effect the non-critical angle has on the acceleration in the critical direction.

7 CONCLUSIONS

We have proposed the 'bang-bang' MPC which approaches time-optimal control principles while being computationally efficient enough to run onboard a commercial quadcopter. This is achieved by assuming that the solution consists of a double step control input in attitude angle for one 'critical' direction while the non-critical direction has a constant angle as solution. This simplifies the OCP and drastically reduces the computational complexity. For efficient control parameter optimization, a bisection scheme in combination with an analytical path prediction model is used. We have shown both in simulation and in real-world flights that the 'bang-bang' controller is a feasible option for fast flight. In fact, for the consecutive waypoint flight, the 'bang-bang' controller is shown to be 17.5% faster than a traditional high-gain PID controller on average. The entire control pipeline easily runs at the main control loop frequency of 512Hz on the Bebop and is sufficiently light to run on even smaller and computationally-limited quadcopters. However, a more thorough analysis is needed to quantify the required computational effort.

The bang-bang MPC also shows promise to be an attractive easy-to-implement solution. For the pipeline requires minimal knowledge of the dynamics (only C_d and mass). And despite that the transition compensation in its current state relies on measurement data, future work could mitigate this process with on-line transition loss estimation. Currently, the constant altitude constraint forms the largest deviation from the theoretical time-optimum solution found by ICLOCS. Finally, the pitch, roll, and thrust limits are set conservatively

Controller	**Maneuver**					
	Forward	Backward	Sideways	Forward-Sideways	Forward-Up	Forward-Down
			Mean Time of Arrival [s]			
Bang-Bang	**1.38** (n=4)	**1.41** (n=4)	**1.29** (n=5)	**1.47** (n=4)	**1.25** (n=3)	**1.50** (n=3)
Bang-Bang Comp.	1.42 (n=15)	1.47 (n=12)	1.37 (n=11)	1.53 (n=15)	1.32 (n=7)	1.52 (n=7)
PID	1.48 (n=10)	1.54 (n=8)	1.43 (n=8)	1.51 (n=8)	1.40 (n=5)	1.54 (n=5)
			Mean Overshoot [m]			
Bang-Bang	0.62 (n=4)	0.81 (n=4)	0.53 (n=5)	0.27 (n=4)	1.20 (n=3)	0.77 (n=3)
Bang-Bang Comp.	0.18 (n=15)	0.22 (n=12)	0.06 (n=11)	0.11 (n=15)	0.51 (n=7)	0.20 (n=7)
PID	**0.05** (n=10)	**0.04** (n=8)	**0.04** (n=8)	**0.04** (n=8)	**0.03** (n=5)	**0.14** (n=5)
			Mean Velocity Error [$\frac{m}{s}$]			
Bang-Bang	3.05 (n=4)	3.51 (n=4)	3.06 (n=5)	1.85 (n=4)	3.82 (n=3)	3.38 (n=3).
Bang-Bang Comp.	1.60 (n=15)	1.88 (n=12)	0.69 (n=11)	0.38 (n=15)	2.63 (n=7)	1.63 (n=7)
PID	**0.08** (n=10)	**0.08** (n=8)	**0.14** (n=8)	**0.06** (n=8)	**1.01** (n=5)	**0.20** (n=5)

Table 3: Performance values the different controllers in 6 maneuvers. *n* is the number of runs performed.

and flight performance could be improved if these parameters are made adaptive.

References

[1] M. Hassanalian and A. Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91:99–131, May 2017.

[2] Hyungpil Moon, Yu Sun, Jacky Baltes, and Si Jung Kim. The IROS 2016 Competitions [Competitions]. *IEEE Robotics and Automation Magazine*, 24(1):20–29, 2017.

[3] Hyungpil Moon, Jose Martinez-Carranza, Titus Cieslewski, Matthias Faessler, Davide Falanga, Alessandro Simovic, Davide Scaramuzza, Shuo Li, Michael Ozo, Christophe De Wagter, Guido de Croon, Sunyou Hwang, Sunggoo Jung, Hyunchul Shim, Haeryang Kim, Minhyuk Park, Tsz Chiu Au, and Si Jung Kim. Challenges and implemented technologies used in autonomous drone racing. *Intelligent Service Robotics*, 12(2), 2019.

[4] Philipp Foehn, Dario Brescianini, Elia Kaufmann, Titus Cieslewski, Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. AlphaPilot: Autonomous Drone Racing. In *RSS 2020*, 2020.

[5] Markus Hehn, Robin Ritz, and Raffaello D'Andrea. Performance benchmarking of quadrotor systems using time-optimal control. *Autonomous Robots*, 33(1-2):69–88, 2012.

[6] Ezra Tal and Sertac Karaman. Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness. *Proceedings of the IEEE Conference on Decision and Control*, 2018-Decem:4282–4288, 2019.

[7] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE, 2011.

[8] Matthias Faessler, Antonio Franchi, and Davide Scaramuzza. Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories. *IEEE Robotics and Automation Letters*, 3(2):620–626, 12 2017.

[9] Charles Richter, Adam Bry, and Nicholas Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. *Springer Tracts in Advanced Robotics*, 114(Isrr):649–666, 2016.

[10] Davide Falanga, Philipp Foehn, Peng Lu, and Davide Scaramuzza. PAMPC: Perception-aware model predictive control for quadrotors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2018.

[11] Elia Kaufmann, Mathias Gehrig, Philipp Foehn, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Beauty and the beast: Optimal methods meet learning for drone racing. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2019-May, pages 690–696, 2019.

[12] Antonio Loquercio, Elia Kaufmann, Rene Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scara-

muzza. Deep Drone Racing: From Simulation to Reality With Domain Randomization. *IEEE Transactions on Robotics*, pages 1–14, 2019.

[13] Shuo Li, Ekin Öztürk, Christophe De Wagter, Guido C. H. E. de Croon, and Dario Izzo. Aggressive online control of a quadrotor via deep network representations of optimality principles. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020*, pages 6282–6287, United States, May 2020. IEEE.

[14] Elia Kaufmann, Antonio Loquercio, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Deep Drone Acrobatics. *RSS: Robotics, Science, and Systems*, 2020.

[15] Tomas Baca, Giuseppe Loianno, and Martin Saska. Embedded model predictive control of unmanned micro aerial vehicles. In *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 8 2016.

[16] Shuo Li, Erik Horst, Philipp Duernay, Christophe De Wagter, and Guido C. H. E. Croon. Visual model-predictive localization for computationally efficient autonomous racing of a 72-gram drone. *Journal of Field Robotics*, 37(4):667–692, 5 2020.

[17] Donald E Kirk. *Optimal control theory: An introduction*, volume 1. Dover Publications, 1998.

[18] Yuanbo Nie, Omar Faqir, and Eric C. Kerrigan. ICLOCS2: Try this Optimal Control Problem Solver before you Try the Rest. *2018 UKACC 12th International Conference on Control, CONTROL 2018*, 2(2017):336, 2018.

[19] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard, and Jeremy Tyler. The paparazzi solution. In *MAV 2006, 2nd US-European competition and workshop on micro air vehicles*. Citeseer, 2006.

[20] Ewoud J.J. Smeur, Qiping Chu, and Guido C.H.E. De Croon. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, 2016.