Iterative learning control for quadrotor landing manoeuvres

Büthker, B.

Award date:
2019

EINDHOVEN UNIVERSITY OF TECHNOLOGY

MASTER MANUFACTURING SYSTEMS ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING
CONTROL SYSTEMS TECHNOLOGY RESEARCH GROUP

Iterative Learning Control for Quadrotor Landing Manoeuvres

MASTER THESIS: CST2019.045

Author:	**Email**	**ID**
B. Büthker	b.buthker@student.tue.nl	0808118

Committee members:

dr. ir. T.A.E. Oomen	t.a.e.oomen@tue.nl
dr. ir. D.G.T. Antunes	d.antunes@tue.nl
dr. ir. R. Toth	r.toth@tue.nl

Supervisors:

dr. ir. D.G.T. Antunes	d.antunes@tue.nl
ir. A.R.P. Andriën	a.r.p.andrien@tue.nl

Eindhoven, Netherlands, May 24, 2019

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

24-05-2019

Name

Bas Büthker

ID-number

0808118

Signature

Submit the signed declaration to the student administration of your department.

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

Iterative Learning Control for Quadrotor Landing Manoeuvres

B. Büthker, A.R.P. Andrien, D.J. Antunes
Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology Research Group
Eindhoven, Netherlands, May 24, 2019

***Abstract*—When a quadrotor is close to the ground, e.g. during landing, a repeatable disturbance occurs. This so called *ground effect* is induced by the downwash, created by the rotors, that cannot move away freely. The aim of this thesis is to reduce the position tracking error resulting from the repeatable ground effect disturbances. We propose the use of a Norm-Optimal Iterative Learning Control (NO-ILC) framework where the feedforward signals coming from the NO-ILC controller are added to the outer-loop of the quadrotor cascaded control loop. The approach is based on a simplified linear model of the quadrotor's dynamic equations. The proposed framework is verified by simulation and implemented on a quadrotor within an experimental setup using an overhead motion-capture system. We show that the NO-ILC controller improves tracking accuracy each iteration and can compensate for the repeatable ground effect.**

***Index Terms*—Iterative Learning Control, Norm-Optimal ILC, quadrotor control, ground effect, trajectory tracking**

I. Introduction

In recent years unmanned aerial vehicles (UAVs), also known as drones, have been receiving increasingly more attention. A specific configuration of UAVs are multi rotor vehicles which can move in the three-dimensional space and have the ability to precisely hover and vertically take-off and land. In particular quadrotors, which are multirotor vehicles with four rotors, have gained much interest from the commercial market as well as for research purposes. For instance Intel autonomously flew 2018 UAVs simultaneously as part of their 50th anniversary [1] and quadrotors are used as spraying mechanisms for plant protection in agriculture [3].

Within the research area of quadrotors it is often the goal to autonomously accomplish difficult tasks, for example balancing an inverted pendulum while flying [4] or building tensile structures with multiple UAVs [5]. In this paper we focus on the task of landing a quadrotor with high speed and accuracy. This is challenging mostly due to what is known as the *ground effect*, which is caused by the so called downwash close to the ground [6]. This downwash is created by the rotors and is necessary during flight to stay airborne. However, when the quadrotor is close to the ground the downwash is unable to move away freely and creates significant disturbances [2], which in general are difficult to model [7], [8] due to turbulent airflows [3].

Motivated by the modelling difficulties, we propose the use of a learning technique in order to autonomously land a quadrotor and reduce the effect of downwash on the position accuracy. In particular our work builds upon the standing assumption that the landing procedure is a repeatable task and the accompanying ground effect disturbance is also repeatable, making iterative learning control (ILC) a suitable candidate. In fact, ILC, as proposed in [9], is meant for systems that perform repeatable tasks. The basic idea is to obtain a control signal for the current cycle based on the error signal of the previous cycle so that the output trajectory converges to the desired reference trajectory [10]. This iterative process can improve tracking accuracy at each cycle, learning the desired feedforward input signal for precise reference tracking and thereby reducing the effect of disturbances that are similar in each iteration. One advantage of ILC over standard feedforward and feedback control techniques is the non-causality of the learning algorithm. In other words, ILC can anticipate and respond to repeated disturbances because the full time sequence of the previous cycle is known and therefore the controller can anticipate on a disturbance yet to come on the current cycle [11]. The non-causal learning algorithm can be implemented in practice due to the repeatability of each cycle.

In [11] different ILC design techniques are presented together with stability requirements to obtain convergence of the tracking error to zero. Some of the well-know ILC techniques are P-type ILC, inverse ILC and Norm-Optimal ILC (NO-ILC) [12]. Different design trade-offs of NO-ILC are investigated during simulation and verified with an experimental setup in [13]. In [14] extended NO-ILC is used to compensate for trial-invariant disturbances while avoiding the amplification of trial-varying disturbances.

ILC has also been applied to improve the tracking performance of UAVs. With the use of motion tracking systems ILC has been successfully simulated and applied to quadrotors for aggressive manoeuvres [15], periodic trajectories [16], [17] as well as real-time trajectory generation [18]. The use of ILC on a quadrotor in a 2D environment is computationally feasible and results in eliminating repeatable disturbances but has limitations with iteration varying systems [19].

In this paper, we use NO-ILC to compensate for the ground effect during landing manoeuvres. The choice of the NO-ILC technique is motivated by the fact that it provides an adequate framework to penalize, in terms of a cost function, deviations from the objectives for the quadrotor, i.e., to track a desired

landing position reference. In particular, the design trade-offs of this framework are discussed. While the model and control of the quadrotor typically consist of two sub-problems, the so-called inner and outer loop control problems, our approach consists of only using ILC for the outer-loop control problem. More specifically, the propellers of the quadrotor generate torques and total thrust which are used to control the thrust vector and its magnitude, respectively; controlling this thrust vector is typically referred to as the inner loop control problem. The thrust vector can be seen as a force, that compensates gravity (and possibly other forces such as the drag force) and controls the position of the quadrotor. This is called the outer loop control problem. For agile quadrotors that have a high ratio between available torque and inertias the time-constants of the inner loop are considerably faster than those of the outer loop. This motivates our approach to view the outer loop as a simple linear model, a double integrator in the three dimensional space given by Newton's law, and therefore suitable for ILC techniques for linear systems. We demonstrate the usefulness of ILC for quadrotor manoeuvres, including landing, both in terms of simulations and experiments using an overhead motion-capture system. In fact, we show that ILC can compensate for unmodeled dynamics and aerodynamic effects both for far and close to ground manoeuvres. At last we discuss further improvements of the ILC framework.

The remainder of this thesis is structured as follows. In Section II the quadrotor model is introduced as well as the ground effect. Furthermore, the cascaded control loop approach is discussed and the problem tackled in the thesis is given. In Section III ILC, lifted ILC, and NO-ILC are formulated and different design parameters are presented. Thereafter, the proposed framework is implemented in simulation in Section IV and validated with experiments in Section V. Section VI discusses the experimental results and finally conclusions are drawn and recommendations are given in Section VII.

II. Problem Formulation and Motivation

In this section, the problem we are interested in is stated by defining the quadrotor model and introducing the repeatable ground effect disturbances. Furthermore, the cascaded control loop framework is described. This section also motivates that ILC is an appropriate candidate to tackle the reference tracking problem induced by the ground effect during landing procedures.

A. Quadrotor Model

The dynamical model of a quadrotor is derived using the Newton-Euler equations as in [20], [21], and [22]. Let $\{\mathcal{I}\}$ denote the world fixed inertial frame which is assumed to follow the north east down (NED) convention and let the body fixed frame at the quadrotor's center of mass be denoted by $\{\mathcal{B}\}$ as shown schematically in Fig. 1. Let $p \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$ represent the position and velocity, respectively, of the center of mass of the quadrotor expressed in frame $\{\mathcal{I}\}$. Furthermore, vector e_3 of frame $\{\mathcal{I}\}$ and b_3 of frame $\{\mathcal{B}\}$ are both aligned with the gravity vector when the quadrotor is hovering.

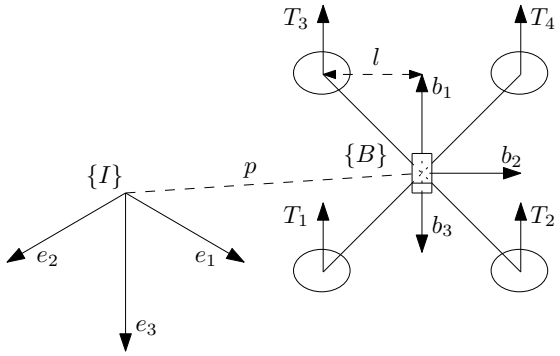


Fig. 1. Schematic representation of the quadrotor with the inertial reference frame $\{\mathcal{I}\}$ in the north-east-down (NED) convention and body-fixed reference frame $\{\mathcal{B}\}$.

Moreover, the rotation matrix of the body-fixed frame $\{\mathcal{B}\}$ w.r.t. the inertial frame $\{\mathcal{I}\}$ is denoted by $R \in \mathcal{SO}(3)$ where $\mathcal{SO}(3) := \{R \in \mathbb{R}^{3\times 3} | R^{\mathrm{T}}R = RR^{\mathrm{T}} = I, \det(R) = 1\}$ is the special orthogonal group of order three.

We assume that there are three forces acting on the rigid body of the quadrotor expressed in frame $\{\mathcal{I}\}$, namely the gravitational force ge_3, with g denoting the gravitational acceleration, the thrust force $-\frac{T}{m}Re_3$, where $m \in \mathbb{R}$ denotes the mass of the quadrotor and $T \in \mathbb{R}$ is the thrust generated by all four rotors, and a velocity dependent drag force $f_{fr}(v)$ that is quadratically related to the quadrotor velocity, e.g. $f_{fr}(v) = -c_D \mathrm{sgn}(v)v^2$, with $c_D = \mathrm{diag}\{c_{D_1},\ c_{D_2},\ c_{D_3}\}$ the friction coefficient. The resulting quadrotor system dynamics are given by:

$$\dot{p} = v \tag{1a}$$
$$\dot{v} = ge_3 - \frac{T}{m}Re_3 + f_{fr}(v) \tag{1b}$$
$$\dot{R} = RS(\omega) \tag{1c}$$
$$\dot{\omega} = J^{-1}(\tau - S(\omega)J\omega) \tag{1d}$$

with $\omega = [\omega_1\ \omega_2\ \omega_3]^{\mathrm{T}} \in \mathbb{R}^3$ the angular velocity expressed in the body fixed frame at the vehicle's center of mass and $S(\omega)$ a skew-symmetric matrix defined as:

$$S(\omega) := \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{2}$$

Furthermore, $J = \mathrm{diag}\{J_1,\ J_2,\ J_3\}$ is the body inertial matrix and $\tau = [\tau_1\ \tau_2\ \tau_3]^{\mathrm{T}}$ are the torques about axis b_1, b_2, and b_3, respectively. The relation between the thrust of each propeller, T_i, and the total thrust T and torque τ is given by:

$$\begin{bmatrix} T \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} T_1 + T_2 + T_3 + T_4 \\ l(T_1 - T_2 + T_3 - T_4) \\ l(-T_1 - T_2 + T_3 + T_4) \\ c(T_1 - T_2 - T_3 + T_4) \end{bmatrix} + \begin{bmatrix} f_{g1} \\ f_{g2} \\ f_{g3} \\ f_{g4} \end{bmatrix} \tag{3}$$

where c is a drag constant, and f_g is a disturbance induced by the ground effect, which is discussed next. Note that we assume the quadrotor is symmetric.

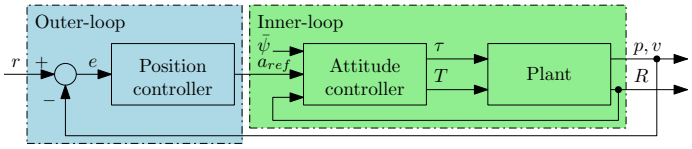


Fig. 2. Cascaded control loop framework consisting of an outer-loop and an inner-loop with control of the position and attitude, respectively.

B. Ground Effect Induced by Downwash

The ground effect of a quadrotor is a hard to model phenomenon [2], [3] where the downwash created by the four rotors of the vehicle cannot move away freely when close to the ground. During landing procedures less thrust is needed close to the ground to keep the same altitude because of a pressure increase beneath the vehicle resulting from the ground effect. This results into repeatable disturbances during landing procedures. Moreover, the ground effect can be different per rotor which can induce a change in attitude.

Suppose that each rotor is subject to a thrust disturbance $d_{ge,i}$ i.e., the actual thrust of rotor i is $T_i = \tilde{T}_i + d_{ge,i}$ where $\tilde{T}_i$ is the thrust in the absence of ground effect. Inspecting (3) we conclude that if the change in rotor thrust due to the ground effect is similar for each rotor, $d_{ge,i} \approx d_{ge}$ for $i = 1, 2, 3, 4$, the main noticeable difference should occur in the total thrust T (which is changed by four times d_{ge}), whereas such a disturbance is cancelled for the torques. This implies that for vertical descend manoeuvres we expect tracking errors mostly in the vertical direction if the thrust vector is approximately aligned with the gravitational vector and for levelled ground profiles, where by symmetry of the quadrotor we expect that the disturbances applied to each propeller, $d_{ge,i}$, should be similar. However, (3) also allows to conclude that, if this is not the case, i.e., if the $d_{ge,i}$ differ significantly due to a ground surface which is not levelled or the quadrotor's manoeuvre is such that the thrust vector is not aligned with the gravitational vector (e.g. oblique descent), then there will be a change in attitude (see also [8]).

C. Cascaded Control Approach

The control of a quadrotor is often tackled by using a cascaded control structure consisting of an outer-loop and an inner-loop (Fig. 2). The outer-loop controls the quadrotor's position to track a position reference, r, whereas the inner-loop controls the thrust and torque to track the acceleration reference, a_{ref}, from the outer-loop and a yaw angle reference $\bar{\psi}$. The thrust and torque are the control inputs for the model. Furthermore, we consider quadrotors with a high thrust to inertia ratio and assume that a stabilizing inner-loop controller with a sufficiently high bandwidth is used, so that we may assume to be able to directly command the attitude of the quadrotor. Moreover, we assume no drag is present. With these assumptions the model in (1) can be rewritten into:

$$\dot{p} = v \tag{4a}$$

$$\dot{v} = \frac{F}{m} + f_d \tag{4b}$$

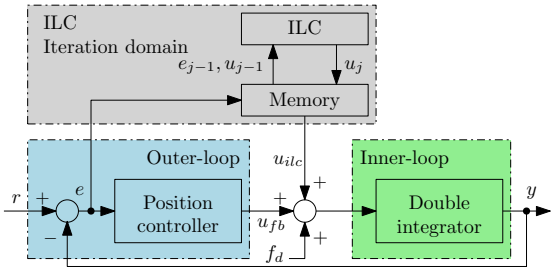


Fig. 3. Cascaded control loop framework with ILC in the iteration domain.

where F is the summation of thrust and gravitational force of (1b) and can be seen as an input signal, and f_d is a term that contains disturbances caused by any remaining inner-loop dynamics as well as the ground effect.

D. Problem Statement

Since the downwash caused by the quadrotor is a hard to model phenomenon, but repeatable disturbance we propose an ILC based controller. This ILC controller aims to reduce the repeatable disturbances induced by the ground effect and results in a faster and autonomous landing procedure for the quadrotor. The problem tackled in this paper can then be stated as follows: Given a position reference for a landing manoeuvre $r(t)$, discretized at sampling period τ, i.e. $\{r(k) = r(k\tau), k \in \{0, 1, \dots\}\}$ and a control law with $u(k) = u_{fb}(k) + u_{ilc}(k)$, where $u_{fb}(k) = \mu(r(k), p(k), v(k))$ is a linear feedback control law and $u_{ilc}(k)$ is a feedforward signal either to be determined or adjusted, find $u_{ilc}(k)$ based on previous error variables $r - p$ by means of an ILC framework.

III. Iterative Learning Control

In Section III-A we introduce an ILC framework and in Section III-B we consider NO-ILC as a special case of ILC, and in Section III-C we discuss different design trade-offs in NO-ILC controller design.

The purpose of ILC is to reduce errors resulting from repetitive disturbances by learning a feedforward signal. This feedforward signal is learned by using non-causal methods in the time domain. By adding a feedforward signal u_{ilc} into the framework of Fig. 2, we obtain the representation as in Fig. 3, where f_d is a disturbance term as in (4b), y is the position output p, and the inner-loop is considered as a double integrator which is discussed next.

A. ILC Framework

1) Double Integrator Model: The model in (4) can be written into a linear time invariant (LTI) state space model as follows

$$\dot{x}(t) = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ I \end{bmatrix} (\underline{u}(t) + f_d(t)) \tag{5a}$$

$$y(t) = \begin{bmatrix} I & 0 \end{bmatrix} x(t) \tag{5b}$$

where $x = [p, v]^{\mathrm{T}}$ is the state, $\underline{u} = \frac{F}{m}$ is the input, y is the output, and t is the time. In order to use the standard ILC

framework we discretize (5) and obtain a discrete-time, linear time invariant system

$$x(k+1) = \underbrace{\begin{bmatrix} I & \tau I \\ 0 & I \end{bmatrix}}_{A} x(k) + \underbrace{\begin{bmatrix} \frac{\tau^2}{2} \\ \tau \end{bmatrix}}_{B} (\underline{u}(k) + f_d(k)) \quad (6a)$$

$$y(k) = \underbrace{\begin{bmatrix} I & 0 \end{bmatrix}}_{C} x(k) \quad (6b)$$

with $k \in \{0, 1, \dots, N\}$ the discrete time index and τ the sample period. The disturbance signal $f_d(k)$ captures the repetitive ground effects and possible disturbances resulting from assuming the double integrator model. Furthermore, the control input $\underline{u}(k)$ is given by:

$$\underline{u}(k) = -Kx(k) + u_{ilc}(k) \quad (7)$$

with a stabilizing position controller K which results in a closed loop system given by:

$$x(k+1) = A_{cl}x(k) + B(u_{ilc}(k) + f_d(k)) \quad (8a)$$

$$y(k) = Cx(k) \quad (8b)$$

with $A_{cl} = A - BK$. In the next Section we introduce the lifted system representation .

2) Lifted System Representation: To exploit the non-causal advantages of the ILC framework the dynamics are translated to the iteration domain, also known as the lifted system representation [11]. First, the input-output solution of (8) is given by

$$y_j(k) = C(qI - A_{cl})^{-1}B(u_j(k) + f_d(k)) + CA_{cl}^k x(0) \quad (9a)$$

$$= H(q)u_j(k) + H(q)f_d(k) + CA_{cl}^k x(0) \quad (9b)$$

$$= H(q)u_j(k) + d(k) \quad (9c)$$

with q the time-shift operator $qx(k) \equiv x(k+1)$, $j \in \{1, 2, \dots\}$ the iteration index, initial condition $x(0)$, and $u_j(k)$ the ILC term. Note that the disturbance term $d(k)$ in (9c) captures the disturbances induced by the ground effect and disturbances caused by inner-loop dynamics f_d as well as the free response of the system to initial condition $x(0)$. Secondly we write (9) in the iteration domain by

$$\underbrace{\begin{bmatrix} y_j(1) \\ y_j(2) \\ \vdots \\ y_j(N) \end{bmatrix}}_{\mathbf{y}_j} = \underbrace{\begin{bmatrix} h_1 & 0 & \cdots & 0 \\ h_2 & h_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_N & h_{N-1} & \cdots & h_1 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} u_j(0) \\ u_j(1) \\ \vdots \\ u_j(N-1) \end{bmatrix}}_{\mathbf{u}_j} + \underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}}_{\mathbf{d}} \quad (10)$$

and

$$\underbrace{\begin{bmatrix} e_j(1) \\ e_j(2) \\ \vdots \\ e_j(N) \end{bmatrix}}_{\mathbf{e}_j} = \underbrace{\begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(N) \end{bmatrix}}_{\mathbf{r}} - \underbrace{\begin{bmatrix} y_j(1) \\ y_j(2) \\ \vdots \\ y_j(N) \end{bmatrix}}_{\mathbf{y}_j} \quad (11)$$

where $\mathbf{y}_j$ is the lifted output, $\mathbf{H}$ is the convolution matrix, $\mathbf{u}_j$ is the lifted input, $\mathbf{r}$ is the desired trajectory, and h_k is the impulse response of the system given by:

$$h_k = \begin{cases} CA_{cl}^{k-1}B, & k \geq 1 \\ 0, & k \leq 0. \end{cases} \quad (12)$$

The lifted system transforms m-input, m-output two-dimensional (time (k) and iteration (j)) into an Nm-input, Nm-output, one-dimensional (iteration) system [12]. Note the output shift in (10) to ignore the inherent delay of the plant.

The most general first-order iteration invariant ILC update law in the iteration domain is

$$\mathbf{u}_{j+1} = \mathbf{L_u}\mathbf{u}_j + \mathbf{L_e}\mathbf{e}_j \quad (13)$$

where $\mathbf{L_u}$, $\mathbf{L_e} \in \mathbb{R}^{Nm \times Nm}$ and $\mathbf{u}_0$ is arbitrary. In order to obtain the closed-loop iteration domain dynamics we substitute (11) into (13) to obtain

$$\mathbf{u}_{j+1} = (\mathbf{L_u} - \mathbf{L_e}\mathbf{H})\mathbf{u}_j + \mathbf{L_e}\mathbf{e}_0 \quad (14)$$

with $\mathbf{e}_0 = \mathbf{r} - \mathbf{d}$. Therefore, (14) is a bounded input bounded output system if and only if

$$|\gamma_i(\mathbf{L_u} - \mathbf{L_e}\mathbf{H})| < 1 \quad (15)$$

for $i = 1, 2, \dots, N$, where $\gamma_i(\cdot)$ is the ith eigenvalue of $(\cdot)$. Furthermore, (13) is monotonic convergent if

$$\|(\mathbf{L_u} - \mathbf{L_e}\mathbf{H})\|_2 < 1 \quad (16)$$

with $\|\cdot\|$ the Euclidean two-norm [12]. Moreover, if (15) holds the asymptotic error, $\mathbf{e}_\infty := \lim_{j \to \infty} \mathbf{e}_j$, is given by

$$\begin{aligned} \mathbf{e}_\infty &= \mathbf{e}_0 - \mathbf{H}\mathbf{u}_\infty \\ &= \left(\mathbf{I} - \mathbf{H}(\mathbf{I} - \mathbf{L_u} + \mathbf{L_e}\mathbf{H})^{-1}\mathbf{L_e}\right)\mathbf{e}_0 \end{aligned} \quad (17)$$

where $\mathbf{u}_\infty$ is obtained by setting $\mathbf{u}_{j+1} = \mathbf{u}_j$ and substituting (17) into (10).

Although we do not explore this in the present thesis, it is worthwhile mentioning that the update law of (13) can also be written in time domain. For the sake of simplicity assume that $\mathbf{L_u} = \mathbf{I}$ and $\mathbf{L_e} = \text{diag}\{\gamma, \dots, \gamma\}, \gamma \in \mathbb{R}^r$ with $\mathbf{L_u}, \mathbf{L_e}$ of appropriate dimensions. This choice for matrices results in the P-type ILC update law in time domain [12] given by:

$$u_{j+1}(k) = u_j(k) + \gamma e_j(k+1). \quad (18)$$

Although both the iteration domain update law of (13) and time-domain update law of (18) are equivalent with this choice of $\mathbf{L_u}$ and $\mathbf{L_e}$ it depends on the practical implementation and construction of $\mathbf{L_u}$ and $\mathbf{L_e}$ which update law is more efficient. The NO-ILC controller introduced next is implemented using the iteration domain update law of (13).

B. Norm-Optimal Iterative Learning Control

The NO-ILC algorithm is designed in the lifted system representation of Section III-A to minimize a quadratic cost [11],

$$\mathcal{J}(j) = \mathbf{e}_{j+1}^{\mathrm{T}}\mathbf{Q}\mathbf{e}_{j+1} + \mathbf{u}_{j+1}^{\mathrm{T}}\mathbf{S}\mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^{\mathrm{T}}\mathbf{R}(\mathbf{u}_{j+1} - \mathbf{u}_j) \tag{19}$$

where $\{\mathbf{Q}, \mathbf{S}, \mathbf{R}\}$ are symmetric positive semi-definite real-valued weighting matrices and $\mathbf{Q}$ and $\mathbf{S}$ are such that $\mathbf{H}^{\mathrm{T}}\mathbf{Q}\mathbf{H}+\mathbf{S}$ is positive definite [12]. Here we focus on the special case when $\{\mathbf{Q}, \mathbf{S}, \mathbf{R}\} \equiv \{q\mathbf{I}, s\mathbf{I}, r\mathbf{I}\}$ with q, s, r real-valued positive scalars. The goal is to compute $\arg\min_{\mathbf{u}_{j+1}} \mathcal{J}(\mathbf{u}_{j+1})$ by substituting $\mathbf{e}_{j+1} = \mathbf{e}_j - \mathbf{H}(\mathbf{u}_{j+1} - \mathbf{u}_j)$, differentiating $\mathcal{J}$ (19) with respect to $\mathbf{u}_{j+1}$, and setting the result to zero [23]. The resulting NO-ILC controller is given by the iteration update law (13) and matrices

$$\mathbf{L_u} = (\mathbf{H}^{\mathrm{T}}\mathbf{Q}\mathbf{H} + \mathbf{S} + \mathbf{R})^{-1}(\mathbf{H}^{\mathrm{T}}\mathbf{Q}\mathbf{H} + \mathbf{R}) \tag{20a}$$

$$\mathbf{L_e} = (\mathbf{H}^{\mathrm{T}}\mathbf{Q}\mathbf{H} + \mathbf{S} + \mathbf{R})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{Q}. \tag{20b}$$

C. Design Trade-offs

From (16), we note that the ILC error convergence speed is determined by $\eta := \|(\mathbf{L_u} - \mathbf{L_e}\mathbf{H})\|_2$. For NO-ILC η can be calculated by substituting (20) into (16) which results into:

$$\eta = \|(\mathbf{H}^{\mathrm{T}}\mathbf{Q}\mathbf{H} + \mathbf{S} + \mathbf{R})^{-1}\mathbf{R}\|_2. \tag{21}$$

From (21) we observe that the convergence speed of the NO-ILC algorithm strongly depends on $\mathbf{R}$ where $\mathbf{R} = 0$ results in deadbeat control and with $\mathbf{R} \to \infty$ the convergence is increasing slower since $\eta \to 1$ By substituting (20) into (17) we obtain the NO-ILC asymptotic error

$$\mathbf{e}_\infty = \left(\mathbf{I} - \mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{Q}\mathbf{H} + \mathbf{S})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{Q}\right)\mathbf{e}_0. \tag{22}$$

Thus, if $\mathbf{H}$ is non-singular, $\mathbf{S} = 0$ leads to zero asymptotic error, $\mathbf{e}_\infty = 0$. Note that $\mathbf{R}$ has no influence on the asymptotic error and therefore the convergence speed is not a function of the asymptotic error.

From (19) we can deduct that $\mathbf{Q}$ corresponds to the weighting of the error. Uniform weighting is acquired when $\mathbf{Q} = \mathbf{I}$. Moreover, a higher value $\mathbf{S}$ leads to improved robustness w.r.t. model uncertainty but leads to a non-zero asymptotic error as in (17). Furthermore, increasing $\mathbf{R}$ results in decreasing asymptotic error fluctuations due to iteration-varying disturbances.

IV. Simulations

A. Circular Trajectory

With the NO-ILC defined in Section III-B we implement the algorithm of (13) and (20) in our cascaded control loop of Fig. 2 which results in the framework of Fig. 3. Because the NO-ILC algorithm operates in the iteration domain a memory block is needed to buffer the error, e, values and unbuffer NO-ILC input vector u_j to output u_{ilc}. The system dynamics of the Avular Curiosity quadrotor [24] are used as in (1) with a drag force constant $C_{D_i} = 0.3$ for $i = 1, 2, 3$ and mass

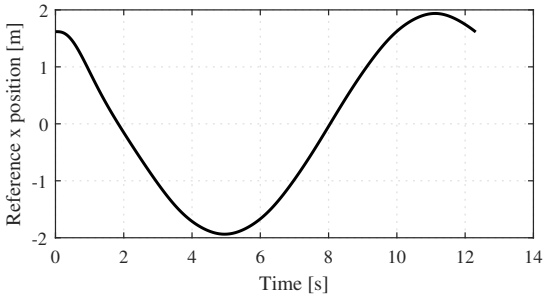


Fig. 4. Reference trajectory of the quadrotor in x-direction.

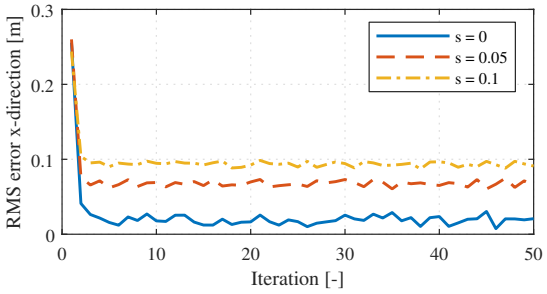


Fig. 5. Quadrotor circular reference trajectory simulation in the x-direction with 50 iterations. Different design parameter s.

$m = 1.034$ kg. Note that the NO-ILC algorithm is based on the double integrator model as described in Section III but in this simulation the full plant dynamics as presented in (1) of the quadrotor are used. The inertia parameters of the Curiosity quadrotor are taken because this quadrotor is used during experiments in Section V. A PD-position controller is used as well as an attitude and altitude controller as in [25]. Furthermore, an input disturbance on the x-axis, f_d, is added which consists of a repetitive disturbance of magnitude 3N for 0.5s at $t = 6.1$s and a stochastic disturbance where each entry is drawn from a uniform distribution with support in the interval $(-0.5, 0.5)$. The closed loop model of Fig. 3 is discretized and sampled at 100Hz. The reference trajectory of the quadrotor is a circle in the xy-plane with a constant z-height of 2m. In Fig. 4 the reference in the x-direction is shown and for visualisation purposes we now focus on this direction. The quadrotor is simulated for 50 iterations with $s = [0\ 0.05\ 0.1]$ whereas $q = 1$ and $r = 0.01$ are kept equal. Fig. 5 shows the RMS position error in the x-direction defined as

$$RMS = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}(x(k)_{ref} - x(k))^2} \tag{23}$$

where $x(k)$ is the position along the x-direction at discrete time k and $x(k)_{ref}$ the reference along the x-direction. Note that (23) holds for all three xyz-axes. Furthermore, we deduct from Fig. 5 that for an increasing value of s the asymptotic error increases, which is equivalent to (17) or (22). Moreover, $\mathbf{e}_\infty \neq 0$ for $s = 0$ because of the iteration variant stochastic

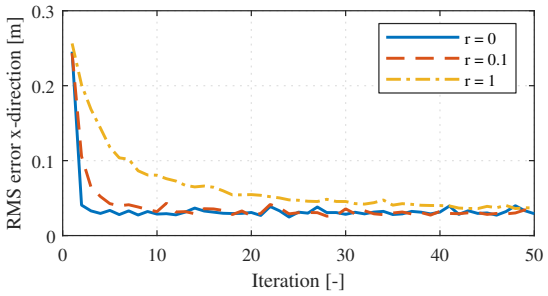


Fig. 6. Quadrotor circular reference trajectory simulation in the x-direction with 50 iterations. Different design parameter r.

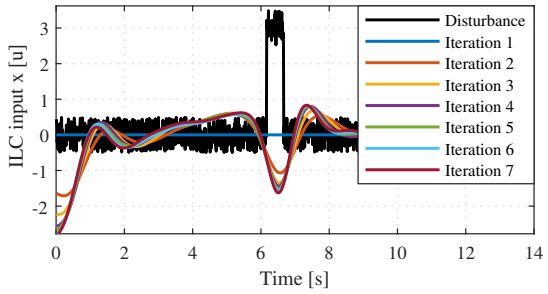


Fig. 7. Quadrotor circular reference trajectory simulation in the x-direction with seven iterations. ILC input with a periodic disturbance and stochastic disturbance. $q = 1$, $s = 0.001$, and $r = 0.01$.

disturbance. Fig. 6 shows simulation results for $r = [0\ 0.1\ 1]$ with $q = 1$ and $s = 0.01$ where we observe that for an increasing value of r the convergence speed, η, decreases which is similar to the conclusion arrived at based on (21). The non-causal behaviour of the NO-ILC algorithm is clearly visible in Fig. 7 where the input, u, of the NO-ILC is shown. Within simulation the NO-ILC algorithm significantly decreases the RMS position error by anticipating on repeatable disturbances.

B. Landing with Ground Effect

The rotors of a quadrotor are usually closer to the ground than rotors of conventional helicopters, i.e. $z > -0.5R$ with R the rotor radius, and z the quadrotor height. Furthermore, it has been claimed in the literature that quadrotors experience a strong ground effect up to $z = -5R$ [8]. With a radius of $R = 0.13$m similar to that of the Curiosity quadrotor simulations are conducted. Within the simulation an increasing ground effect disturbance, $f_g = -10R - 2z$, is applied for $z \in [-5R, 0]$ where white Gaussian noise is added to f_g with a signal to noise ratio of 20dB. A vertical trajectory is provided to the quadrotor, i.e., a trajectory where $x(t)$, $y(t)$, are constant for every t. Fig. 8 shows the z-position reference from -3m until 0m, the z-position error, the z-velocity reference and the ILC input for seven iterations. We observe that the NO-ILC significantly decreases the error induced by the ground effect simulation by almost a factor 100 as can be seen in Fig. 9.

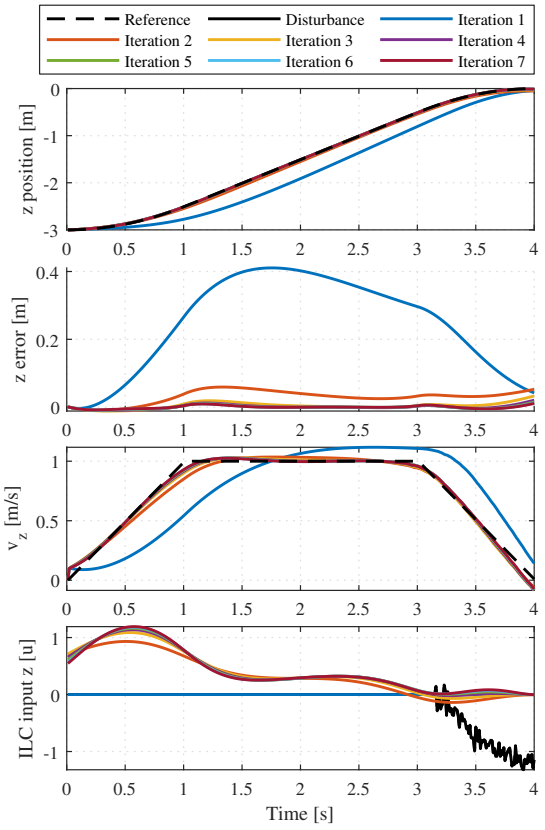


Fig. 8. Quadrotor landing reference trajectory simulation with seven iterations including ground effect $f_g = -10R - 2z$, $z \in [-5R, 0]$. A white Gaussian Noise with a signal to noise ratio of 20 is added to the signal. NO-ILC parameters: $q = 1$, $s = 0.001$, and $r = 0.01$.

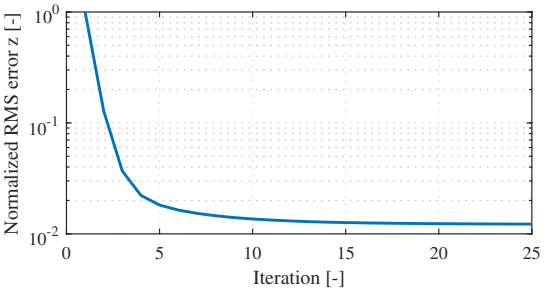


Fig. 9. Normalized RMS error of quadrotor landing reference trajectory simulation with 25 iterations.

V. Experiments

To validate our assumptions made in Section II-C, and simulations with ground effect in the previous Section IV-B, experiments with the Avular Curiosity [24] quadrotor were conducted. In this section we describe the experimental setup and highlight four experiments:

- Horizontal trajectory, one axis (Section V-B)
- Vertical trajectory, one axis (Section V-C1)
- Vertical trajectory, two axes (Section V-C2)
- Vertical trajectory, constant height (Section V-C3)

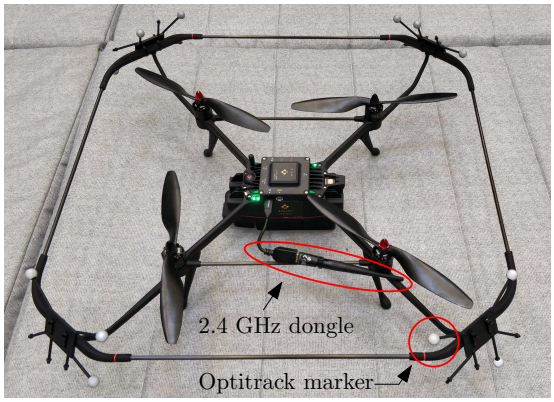


Fig. 10. Avular Curiosity quadrotor including shroud, 12 Optitrack markers, and a 2.4 GHz dongle.

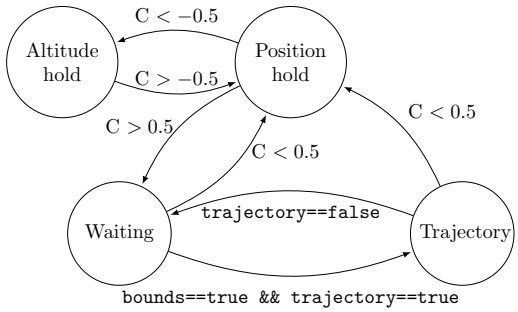


Fig. 11. High-level stateflow representation of the quadrotor with three inputs; the manual controller switch C $\in \{-1, 0, 1\}$, the boolean value `trajectory`, and the boolean value `bounds`.

A. Experimental Setup

The experiments are conducted at the Robotics Soccer Field at the TU/e with the use of the Optitrack Motion Capture System [26]. The system consists of eight Flex 13 cameras and two Opti-hubs to connect the cameras to a computer. Motive software on the computer gives position and attitude of a rigid body created with individual markers. When properly marked the Optitrack systems gives milimeter position precision and degree attitude precision via quaternions on 120Hz. In Fig. 10 the Avular Curiosity quadrotor [24] is shown with 12 markers attached to its shroud. The 2.4 Ghz dongle connects the Raspberry Pi on the quadrotor with the network on which the Motive computer is streaming data. A full schematic representation, software architecture, and more information about the Optitrack Motion Capture System is given in Appendix A.

To conduct the experiments a stateflow diagram, as shown in Fig. 11, is created to make sure that the initial conditions on each iteration are within certain conditions. The stateflow consists of four states (altitude hold, position hold, waiting, and trajectory) and three inputs (the manual controller switch C $\in \{-1, 0, 1\}$, the boolean value `trajectory`, and the boolean value `bounds`). After the take-off procedure the quadrotor is in altitude hold where a constant altitude is maintained. In position hold the position of the quadrotor is fixed in xyz space. Furthermore, when the quadrotor is switched to waiting state from position hold an initial xyz position (p_0) is defined from the current position. Thereafter, the quadrotor waits until predefined conditions on position, translation velocity, and attitude are met:

$$|p_0 - p| < 0.1 \text{ m}, \tag{24a}$$
$$|v| < 0.1 \text{ m/s}, \tag{24b}$$
$$|\phi|, |\theta|, |\psi| < 0.0873 \text{ rad } (5 \text{ deg}), \tag{24c}$$

where ϕ, θ, ψ are the roll, pitch, and yaw angles that define the quadrotors attitude representation in xyz Euler notation. Note that the quadrotor's translation velocity is an estimated value from a Kalmann filter based on Optitrack position and acceleration data from the IMU on the quadrotor. These conditions are introduced to ensure that at each iteration the initial conditions are almost identical. When the conditions of (24) are met, the two boolean values `bounds` and `trajectory` are set to `true` and the quadrotors starts his trajectory. At the final reference point of the trajectory the value of `trajectory` is set to `false`, the feedforward values for the next iteration are calculated, and the quadrotor flies back to his waiting position to potentially start another iteration.

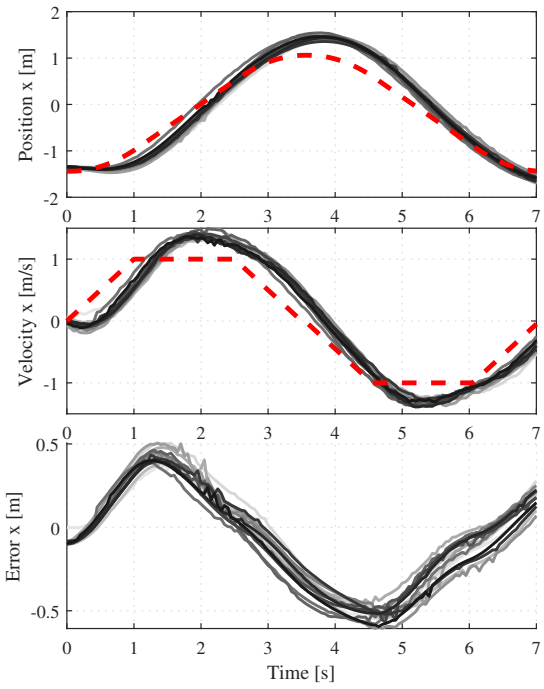
TABLE I
FEEDBACK CONTROLLER GAINS

	x	y	z
P	0.25	0.3	3
I	-	-	-1
D	0.18	0.18	-4

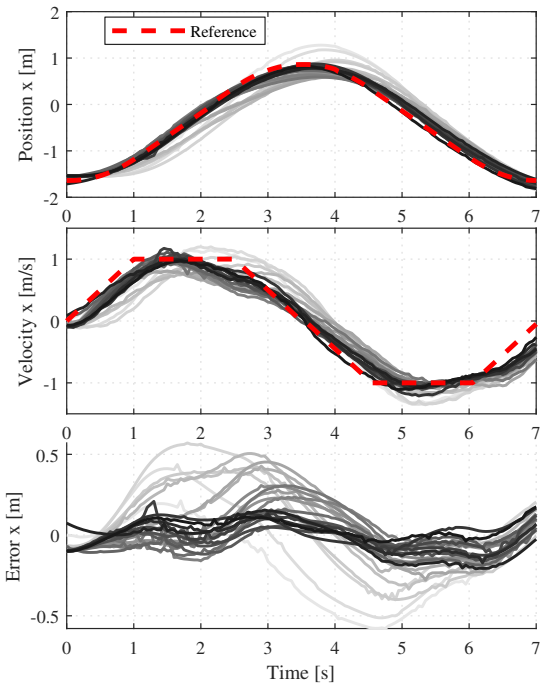
In all experiments the yaw reference heading is set to zero and the inner-loop attitude control is done with PID feedback controllers from the Avular flightstack running on the low-level with a sample frequency of 500Hz. The outer-loop (xyz-direction) control is done with PD feedback controllers in the xy-direction, and PID feedback control in the z-direction and feedforward based on NO-ILC on which the experiments are emphasized. The P(I)D controller gains are shown in Table I. Note that the PD feedback controllers fit the model of the controller considered in Section III, but the PID controller does not, if the integral action is non-zero. This will be discussed further in Section VI.

B. Horizontal Line Trajectory

In this experiment the effectiveness of the NO-ILC controller is shown. The goal is to track a position reference in the x-direction whereas references along y-, and z-direction remain constant. The x-reference the quadrotor has to track is as follows: 2.5m in the positive x-direction and back 2.5m in the negative x-direction in 7s. The initial, halfway, and end velocities are zero and a maximum of 1 m/s is taken. The position and velocity reference is shown in Fig. 12 (dotted red). This experiment is conducted for 15 iterations without



(a) Feedback PD-controller.



(b) Feedback PD-controller and NO-ILC controller with parameters: $q = 0.05$, $s = 2$, and $r = 1$. NO-ILC feedforward values are shown in Appendix C, Fig. 23.

Fig. 12. Experimental results of Section V-B on a back and forth reference in the x-direction with 2.5m in the positive x-direction and back 2.5m in the negative x-direction in 7s. The initial, halfway, and end velocities are zero and a maximum of 1 m/s is taken. From top to bottom: x-position reference and x-position, x-velocity, and x-position error. Iterations are shown from light grey to black for a total of 15 iterations in Fig. 12a and a total of 23 iterations in Fig. 12b.

NO-ILC and only feedback controllers as is shown in Fig. 12a. Hereafter, the experiment is repeated for 23 iterations with a NO-ILC controller added. The parameters of the NO-ILC controller are as follows: $q = 0.05$, $s = 2$, and $r = 1$. Results from the experiment are shown in Fig. 12b where it is clearly visible that the position error decreases after a number of iterations. Furthermore, in comparison with the experiment without the NO-ILC controller, we observe the effectiveness of the non-causality of the NO-ILC because the overshoot and delay decrease significantly. In Fig. 13 the RMS position error, as defined in (23), is shown for the experiment with, and without NO-ILC controller. We can deduct from Fig. 13 that the quadrotor's position error is decreasing over iterations with NO-ILC whereas with only feedback controllers the asymptotic position error stays around 0.3m. In fact, the position RMS error including NO-ILC reduces by about a factor of 3 illustrating the effectiveness of the proposed ILC strategy for this manoeuvre.

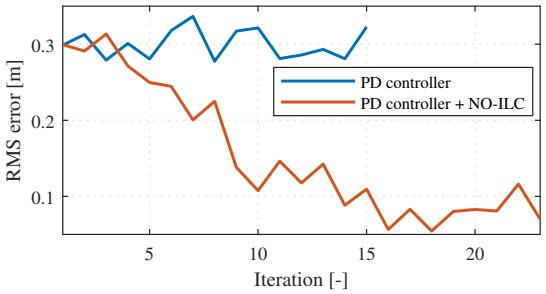


Fig. 13. RMS position error of reference in the x-direction as shown in Fig. 12a and Fig. 12b.

C. Vertical Trajectories

The remaining three experiments are conducted to validate the effectiveness of the proposed NO-ILC algorithm to compensate for the induced ground effect disturbance.

1) One Axis: Within this experiment the x-, and y-position reference are taken to be constant and only the z-position reference is changed. The z-reference consists of a cosine starting at $z = -1.75$m with an amplitude of 0.5m and a period of 8s as can be seen in Fig. 14. At $t = 4$s the position reference of the quadrotor is at $z = -0.25$m. Note that, due to the definition in the Optitrack system of the geometry of quadrotor, $z = -0.15$m when standing on the ground. In Fig. 14 the experimental results are shown for eight iterations with NO-ILC controller parameters: $q = 1$, $s = 0.1$, and $r = 0.05$. Furthermore, the z-position and NO-ILC feedforward values of the total experiment are shown in Appendix C, Fig. 24 and the RMS position error is depicted in Fig. 15. As highlighted in Section IV-B [8], a strong ground effect should take place at $z = -5R$. With a radius of $R = 0.13$m of the Curiosity

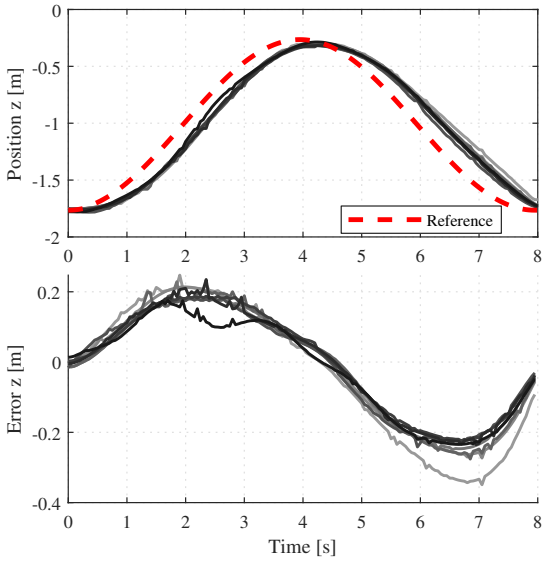


Fig. 14. Experimental results of Section V-C1. Shown are the z-position reference, z-position, and z-position error. Iterations are shown from light grey to black for a total of 8 iterations. Feedback PD-controller and NO-ILC controller with parameters: $q = 1$, $s = 0.1$, and $r = 0.05$. RMS error plot is shown in Fig. 15 and NO-ILC feedforward values are shown in Appendix C, Fig. 24.

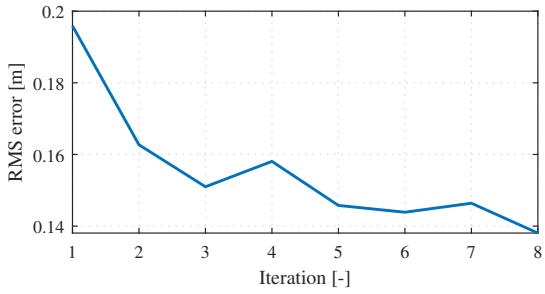


Fig. 15. RMS z-position error of experiment shown in Fig. 14.

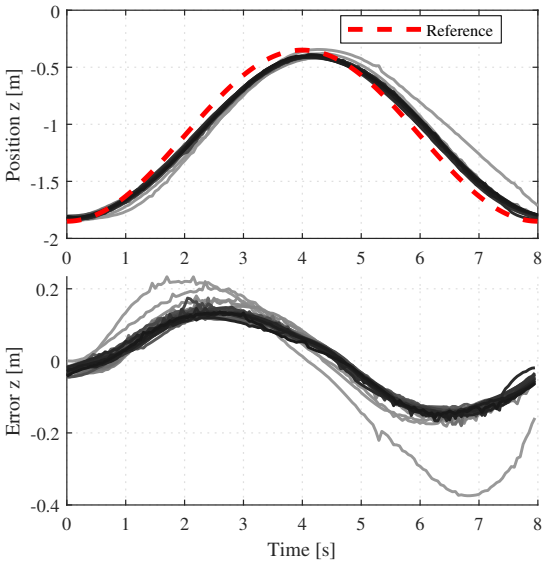


Fig. 16. Experimental results of Section V-C2. Shown are the z-position reference, z-position, and z-position error. Iterations are shown from light grey to black for a total of 31 iterations. Feedback PD-controller and NO-ILC controller in z-direction with parameters: $q = 2$, $s = 0.05$, and $r = 0.01$. RMS error plot is shown in Fig. 17 and NO-ILC feedforward values are shown in Appendix C, Fig. 25.

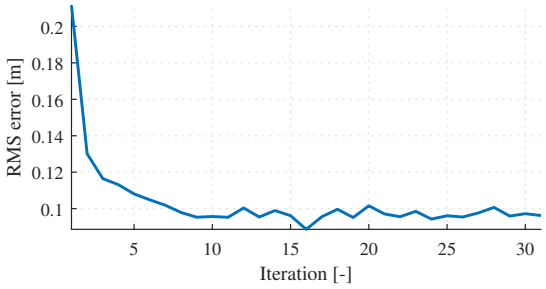


Fig. 17. RMS z-position error of experiment shown in Fig. 16.

quadrotor this indicates that disturbances induced from the ground effect should occur from $z = -0.65$m. However, from the tracking errors plots it is not possible to disclose any consequences of the ground effect in $-0.65 < z < -0.15$ yet, and the reason for this will be discussed in the next subsections. Moreover, we remark that the convergence speed in the RMS position error is lower, as well as the asymptotic RMS position error is higher, than the horizontal experiment in Section V-B, Fig. 13.

2) Two Axes: [1] Because there was no observable ground effect disturbance visible in the previous experiment we now conduct the experiment of Section V-C1 once again but with a reference that spans two axes. In other words, the x-position reference is the same as the z-position reference and the y-position reference is constant. Furthermore, the NO-ILC controller parameters are changed to $q = 2$, $s = 0.05$, and $r = 0.01$ to increase the position error convergence speed and decrease the asymptotic position error. Fig. 16 shows the z-position reference, z-position, and z-position error for 32 iterations. Moreover, the z-position and NO-ILC feedforward values of the total experiment are shown in Appendix C, Fig. 25 and the RMS position error is depicted in Fig. 17. Once again there is no external ground effect visible in Fig. 16 in the region $-0.65\text{m} < z - 0.15\text{m}$. However, in Fig. 17 we observe that the asymptotic position error is decreased and the position error convergence is decreasing faster in comparison to Fig. 15 due to the decreased s and r, respectively.

3) Decreasing Constant Height: The previous two experiments do not allow to disclose ground effect disturbance that can be compensated with ILC. Therefore, the next experiment is conducted where the quadrotor hovers at a constant height. By hovering ($\dot{v} = 0, v = 0$) the force F in (4b) should be equal to zero if there is no additional f_d term which includes the ground effect. Furthermore, if $f_d = 0$, then $T = mg$,

[1] A video of the experiment can be found at: http://tiny.cc/Quad-ILC or https://youtu.be/CEiIBUeQkiw.

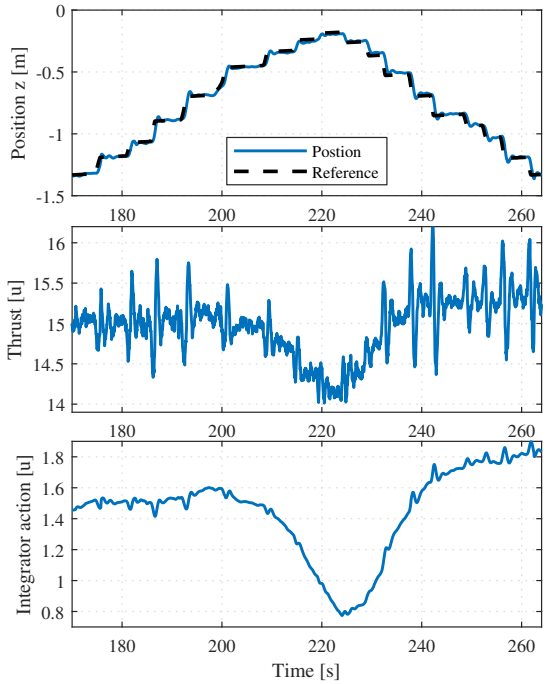


Fig. 18. From top to bottom: z-position reference and z-position, thrust command, and integrator action of the PID position controller in the z-direction.

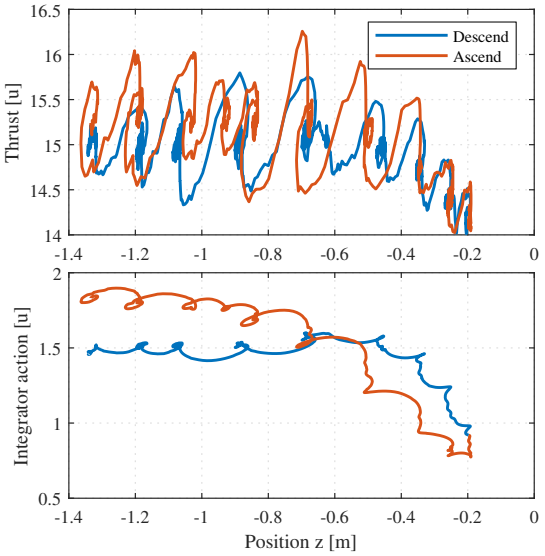


Fig. 19. Top: thrust command. Below: integrator action of the PID position controller in the z-direction. Both vs. the z-position of the quadrotor.

i.e. if there is no ground effect the thrust signal T should be equal at all heights while hovering. However, if there is ground effect the thrust force T should be less when the quadrotor is hovering close to the ground, as explained in Section II-B.

Within the experiment we start at a constant height of $z = -1.3$m and descend ≈ 0.1m every ≈ 5s. After reaching $z = -0.2$m we ascend again to $z = -1.3$m. In Fig. 18 the z-position reference and z-position are shown as well as the thrust command, and integrator action of the PID controller in the z-direction. Between $200\text{s} < t < 235\text{s}$ a decrease in thrust signal is shown which indicates the ground effect. The thrust decrease validates our hypotheses of a lesser thrust signal while the quadrotor is hovering close to the ground. This drop in thrust signal mainly originates from the integral action of the PID position controller. Furthermore, in Fig. 19 the thrust command and integrator action are shown vs. the z-position of the quadrotor where it is clearly visible that between $-0.5\text{m} < z < 0\text{m}$ the thrust command, and the integrator action decrease while descending and increase while ascending.

VI. Discussion of the Experimental Results

The results of Section V-C1 and V-C2 do not highlight the ground effect, which is clearly present as highlighted in the results of Section V-C3. The key to understand this is the integral action: the integral term, which was not modelled in Section III is already a learning mechanism, which adjusts the control input to compensate for the ground effect. Due to time constraints, it was not possible to tune the parameters of the feedback controllers to set this integral action to zero. In fact the integral action plays a crucial role for stabilisation since some parameters such as the motor PWM values that should be given to the quadrotor to compensate for the gravitational force are unknown and are not given by the low level controller used.

The fact that in the z-axis the unmodelled integral action is present, makes it difficult to conclude the effectiveness of ILC for landing manoeuvres from the experimental results. In fact two learning mechanisms are tangled and from Fig. 18 we deduct that the integral action mechanism is actually crucial to compensate for the ground effect. This leads to the fact that the ILC mechanism is unnecessary and only plays a small role. We hypothesise that if either the integral action is not present or for aggressive manoeuvres the ILC algorithm would play an important role.

In turn for the trajectory about the x-axis, where there is no integral action present and the ILC framework in Section III is representative of the dynamics and controller of the system, we do see a factor three improvement in the RMS position error, which we believe is an indication that ILC can be an important tool to compensate for unmodelled dynamics and repeatable disturbances in the context of quadrotors.

As a final note, we mention that to properly take into account the integral action one additional state can be considered in the closed loop equations. In fact, if we write the PID controller as

$$\underline{u}(k) = -Kx(k) + u_{ilc}(k) + x_I(k) \tag{25a}$$

$$x_I(k) = x_I(k-1) + \alpha(Cx(k) - r(k)) \tag{25b}$$

instead of (6), we can still write the closed loop system as (8) by defining a state

$$\bar{x} = \begin{bmatrix} x \\ x_i \end{bmatrix} \tag{26}$$

which results into:

$$\bar{x}(k+1) = A_{cl}\bar{x}(k) + B_{cl}(u_{ilc}(k) + f_d(k)) \quad (27a)$$
$$+F_{cl}r(k) \quad (27b)$$
$$y(k) = C_{cl}\bar{x}(k) \quad (27c)$$

with

$$A_{cl}= \begin{bmatrix} A - BK & B \\ \alpha C & I \end{bmatrix}, B_{cl} = \begin{bmatrix} B \\ 0 \end{bmatrix},$$
$$C_{cl}= \begin{bmatrix} C & 0 \end{bmatrix}, F_{cl} = \begin{bmatrix} 0 \\ \alpha \end{bmatrix}. \quad (28)$$

This could be an approach to take into account the integral action in the ILC computation. However, we suspect that this would not change the conclusions, since the integral action would still be active and even with different matrices, the ILC method would have small influence.

VII. Conclusions and Recommendations

A. Conclusions

In this thesis we propose to use NO-ILC to compensate for repeatable disturbances induced by the ground effect. This NO-ILC controller is based on a double integrator model and adds the 'learned' feedforward signals to the outer-loop (Fig. 3) of the quadrotor cascaded control loop. The effectiveness of the proposed framework where the tunable NO-ILC reduces the ground effect disturbances significantly is verified by simulation. Moreover, an extensive experimental setup is created with an overhead motion-capture system where we apply the proposed NO-ILC framework on a quadrotor. We successfully show that NO-ILC indeed improves tracking accuracy each iteration but is unnecessary to compensate for the observed ground effect disturbances due to a sufficiently fast integral action in the PID feedback z-position controller. However, even without extensive tuning of the NO-ILC parameters a position error reduction of at least a factor two in the z-direction and a factor three in the x-direction is obtained.

B. Recommendations

In future research several improvements can be made to track larger trajectories and reduce the computational load of the ILC algorithm, e.g. as in [27], ([28] Chapter 10). Furthermore, the trajectory reference and offline calculated NO-ILC parameters are now flashed onto the high-level cortex on the quadrotor which are relatively large in terms of memory. To solve this problem one could use the RPi as data storage and send the signals via serial communication. However, timing issues could arise from this implementation. Moreover one could further adapt the NO-ILC parameters (q, r, s) to improve asymptotic error and/or the rate of position error convergence. Moreover, the proposed closed loop model of (27) including I action could be implemented and compared to the current PD based framework.

Finally, the proposed NO-ILC framework requires full state information, most importantly the absolute quadrotor position in space. This absolute position is now given by an overhead motion-capture system. If there is no such system the absolute position depends on the accuracy of the state estimation from the on-board sensors. To implement NO-ILC in an experimental setup without an external motion-capture system the state estimation accuracy needs to be approved.

Acknowledgement

A huge gratitude goes out to Alex who has been very supportive during this whole thesis, especially in the experimental phase. As well as Duarte, thank you for the supervision of my thesis. The meetings with you both were always fruitful and I have learned a lot from your feedback and discussions. Furthermore, I would like to thank everybody in the Robotics lab during my time there. The highly academic conversations during the lunch time is what kept me going in tough times. To my friends from Tachos, E.S.R. Thêta, and all others, thank you for an awesome student time these past years. At last, but not least, I would like to thank my parents, brother, and girlfriend for all the unending support during my study and especially during my thesis.

References

[1] Intel, "https://www.intel.com/content/www/us/en/technology-innovation/aerial-technology-light-show.html."

[2] Y. Zheng, S. Yang, X. Liu, J. Wang, T. Norton, J. Chen, and Y. Tan, "The computational fluid dynamic modeling of downwash flow field for a six-rotor UAV," *Frontiers of Agricultural Science and Engineering*, vol. 5, no. 2, pp. 159–167, 2018. [Online]. Available: http://journal.hep.com.cn/fase/EN/10.15302/J-FASE-2018216

[3] F. Yang, X. Xue, C. Cai, Z. Sun, and Q. Zhou, "Numerical Simulation and Analysis on Spray Drift Movement of Multirotor Plant Protection Unmanned Aerial Vehicle," *Energies*, vol. 11, no. 9, p. 2399, sep 2018. [Online]. Available: http://www.mdpi.com/1996-1073/11/9/2399

[4] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *2011 IEEE International Conference on Robotics and Automation*, no. 2. IEEE, may 2011, pp. 763–770. [Online]. Available: http://ieeexplore.ieee.org/document/5980244/

[5] F. Augugliaro, A. Mirjan, F. Gramazio, M. Kohler, and R. D'Andrea, "Building tensile structures with flying machines," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, nov 2013, pp. 3487–3492. [Online]. Available: http://ieeexplore.ieee.org/document/6696853/

[6] M. B. Vankadari, K. Das, C. Shinde, and S. Kumar, "A Reinforcement Learning Approach for Autonomous Control and Landing of a Quadrotor," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2018, pp. 676–683. [Online]. Available: https://ieeexplore.ieee.org/document/8453468/

[7] D. Yeo, E. Shrestha, D. A. Paley, and E. M. Atkins, "An Empirical Model of Rotorcrafy UAV Downwash for Disturbance Localization and Avoidance," in *AIAA Atmospheric Flight Mechanics Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, jan 2015. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2015-1685

[8] X. He, G. Kou, M. Calaf, and K. K. Leang, "In-Ground-Effect Modeling and Nonlinear Disturbance Observer for Multi-rotor UAV Control," *Journal of Dynamic Systems, Measurement, and Control*, mar 2019.

[9] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems," in *The 23rd IEEE Conference on Decision and Control*. IEEE, dec 1984, pp. 1064–1069. [Online]. Available: http://ieeexplore.ieee.org/document/4048052/

[10] K. L. Moore, *Iterative Learning Control for Deterministic Systems*, ser. Advances in Industrial Control. London: Springer London, 1993. [Online]. Available: http://link.springer.com/10.1007/978-1-4471-1912-8

[11] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, jun 2006. [Online]. Available: http://ieeexplore.ieee.org/document/1636313/

[12] D. Bristow, K. Barton, and A. Alleyne, "Iterative Learning Control," in *The Control Handbook*, 2nd ed. Boca Raton: CRC Press, 2010, pp. 36–1:36–19. [Online]. Available: http://www.crcnetbase.com/doi/abs/10.1201/b10384-42
[13] B. Altn, J. Willems, T. Oomen, and K. Barton, "Iterative Learning Control of Iteration-Varying Systems via Robust Update Laws with Experimental Implementation," *Control Engineering Practice*, vol. 62, no. November 2016, pp. 36–45, may 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0967066117300345
[14] T. Oomen and C. R. Rojas, "Sparse iterative learning control with application to a wafer stage: Achieving performance, resource efficiency, and task flexibility," *Mechatronics*, vol. 47, pp. 134–147, nov 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957415817301289
[15] O. Purwin and R. D'Andrea, "Performing and extending aggressive maneuvers using iterative learning control," *Robotics and Autonomous Systems*, vol. 59, no. 1, pp. 1–11, jan 2011. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2010.09.004http://linkinghub.elsevier.com/retrieve/pii/S0921889010001570
[16] M. Hehn and R. D'Andrea, "An Iterative Learning Scheme for High Performance, Periodic Quadrocopter Trajectories," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, nov 2013, pp. 2445–2451. [Online]. Available: http://www.nt.ntnu.no/users/skoge/prost/proceedings/ecc-2013/data/papers/1133.pdfhttp://ieeexplore.ieee.org/document/6696700/
[17] ——, "A frequency domain iterative learning algorithm for high-performance, periodic quadrocopter maneuvers," *Mechatronics*, vol. 24, no. 8, pp. 954–965, dec 2014. [Online]. Available: http://dx.doi.org/10.1016/j.mechatronics.2014.09.013http://linkinghub.elsevier.com/retrieve/pii/S0957415814001561
[18] ——, "Real-Time Trajectory Generation for Quadrocopters," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 877–892, aug 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7128399/
[19] A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," *Autonomous Robots*, vol. 33, no. 1-2, pp. 103–127, aug 2012. [Online]. Available: http://link.springer.com/10.1007/s10514-012-9283-2
[20] S. van den Eijnden, "Cascade Based Tracking Control of Quadrotors," Master Thesis, Eindhoven University of Technology, 2017.
[21] P. Rooijakkers, "Design and Control of a Quad-Rotor: application to Autonomous Drone Refereeing," Master Thesis, Eindhoven University of Technology, Eindhoven, 2017.
[22] N. Jeurgens, "Identification and Control Implementation of an AR.Drone 2.0," Master Thesis, Eindhoven University of Technology, 2017.
[23] D. A. Bristow and B. Hencey, "A Q, L factorization of Norm-Optimal Iterative Learning Control," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 2380–2384. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4739348
[24] Avular, "https://www.avular.com/curiosity." [Online]. Available: https://www.avular.com/curiosity
[25] D. Kooijman, "Quadrotor Trajectory Tracking using Cascaded Model Predictive Control and Tracking Control in S2 x S1," Master Thesis, Eindhoven University of Technology, 2018.
[26] OptiTrack, "Motion Capture System." [Online]. Available: https://optitrack.com/
[27] J. van Zundert, J. Bolder, S. Koekebakker, and T. Oomen, "Resource Efficient ILC: Enabling Large Tasks on an Industrial Position-Dependent Flatbed Printer," *IFAC-PapersOnLine*, vol. 49, no. 21, pp. 567–574, 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896316322741
[28] J. C. D. van Zundert, "Resource-Aware Motion Control: Feedforward, Learning, and Feedback," Ph.D. dissertation, Eindhoven University of Technology, 2018.
[29] PhaseSpace, "Impulse X2 Motion Caption System." [Online]. Available: http://www.phasespace.com/impulse-motion-capture.html

APPENDIX A
OPTITRACK MOTION CAPTURE SYSTEM

As mentioned in Section V-A the Optitrack Motion Capture Systems [26] consists of eight Flex 13 cameras (Fig. 21: A). These cameras are connected with USB cables to two Opti-hubs (Fig. 21: B). These hubs are connected together with a

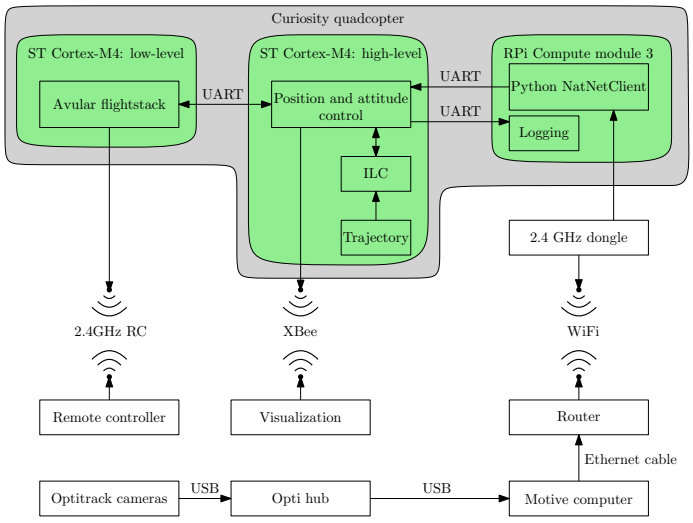


Fig. 20. Schematic representation of the experimental setup with the Optitrack system and the Curiosity quadrotor

synchronization cable and are both individual connected to the computer running Motive software. A schematic representation of the setup is shown in Fig. 20. When the cameras and hubs are connected one can start with the calibration of the system. This calibration is done with the calibration wand including three Optitrack markers on a predefined position (Fig. 21: C). By moving the wand trough the capture volume inside the camera range, Motive software will collect samples per camera. When enough samples are collected the software will triangulate the date and calibrate the system. After the calibration one needs to define the initial location and orientation of the xyz-origin. This is done by setting the ground plane (Fig. 21: D) on the ground and in the center of the capture volume. Motive software will define the x-axis on the short side, the z-axis on the long side, and the y-axis up by definition of the right-handed frame, respectively. The setup is now completed and one can define a rigid body by selecting individual Optitrack markers. The Curiosity quadrotor is defined by twelve markers (Fig. 10). After the rigid body definition Motive software streams xyz position and quaternion attitude on a maximum of 120Hz to the local host or a specific IP-adress. To use the position and attitude data obtained from the Optitrack system on the quadrotor a WiFi network is created with a router. The router is connected to the computer running motive with an Ethernet cable. A Raspberry Pi Compute module 3 (RPi) within the quadrotor is connected with the WiFi network via a 2.4 GHz WiFi dongle (see Fig. 10 and Fig. 20). On the RPi a Python script (NatNetClient) is ran which creates a socket to read the data stream from Motive and unpacks rigid body data packets. When rigid body data is obtained it is sent to the high-level M4 Cortex via serial (UART) communication with the use of custom defined MAVlink messages. See the work of [21] for more information. The high-level (together with the low-level) does the position and attitude control and sends commands to

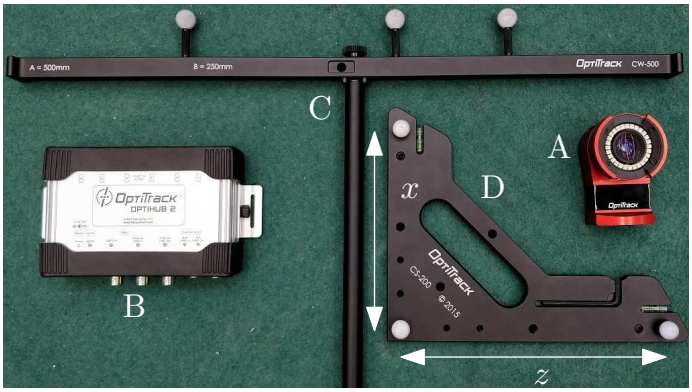


Fig. 21. A: Flex 13 Camera. B: Opti-Hub. C: Calibration Wand on small space setting (= 250 mm). D: Ground Plane including axes definition.

Fig. 22. PhaseSpace hardware. From left to right: three RF transceivers (LED controllers), three active LEDs, 12 Megapixel camera, and server computer.

the four individual motors on the quadrotor.

APPENDIX B
PHASESPACE MOTION CAPTURE SYSTEM

The PhaseSpace Impulse X2 Motion Capture System [29] on the Eindhoven University of Technology is situated in the IPO building in the Virtu/e Lab and consists of eight cameras attached to the wall, two LED controllers, multiple active LEDs and one server computer as shown in Fig. 22. The PhaseSpace cameras record at 960Hz with sub-millimeter precision and are daisy chained with ethernet cables to the server computer. These cameras detect the LED positions and sends them to the server computer that processes the data and calculates actual position and pose. There are API/SDKs available in Python and C++ to process the data on your own client system. The LEDs are connected to the battery powered LED controller/driver unit with strings. Each LED controller supports six strings with a total of 72 LEDs. The LED controller receives 2.4GHz RF timing signals from the server computer and uses this signal to create a modulated pattern of pulses for each active LED marker, which is used to uniquely identify each LED by the cameras. To use the PhaseSpace system with the Avular Curiosity quadrotor [24] a S-Function in MATLAB is written. With this MEX file and the corresponding library file a MATLAB Simulink block is created which outputs position and pose of a rigid body on a specific sample time. The system is tested on a sample frequency of 200Hz and acquires millimeter precision. However, due to the faculty move of the Human Technology Interaction group to a new building and the acquiring of the Optitrack System (see Appendix A) it was decided to terminate the usage of the PhasesSpace system.

APPENDIX C
EXPERIMENTAL RESULTS: FIGURES

Fig. 23, Fig. 24, and Fig. 25 are shown on the next page(s).

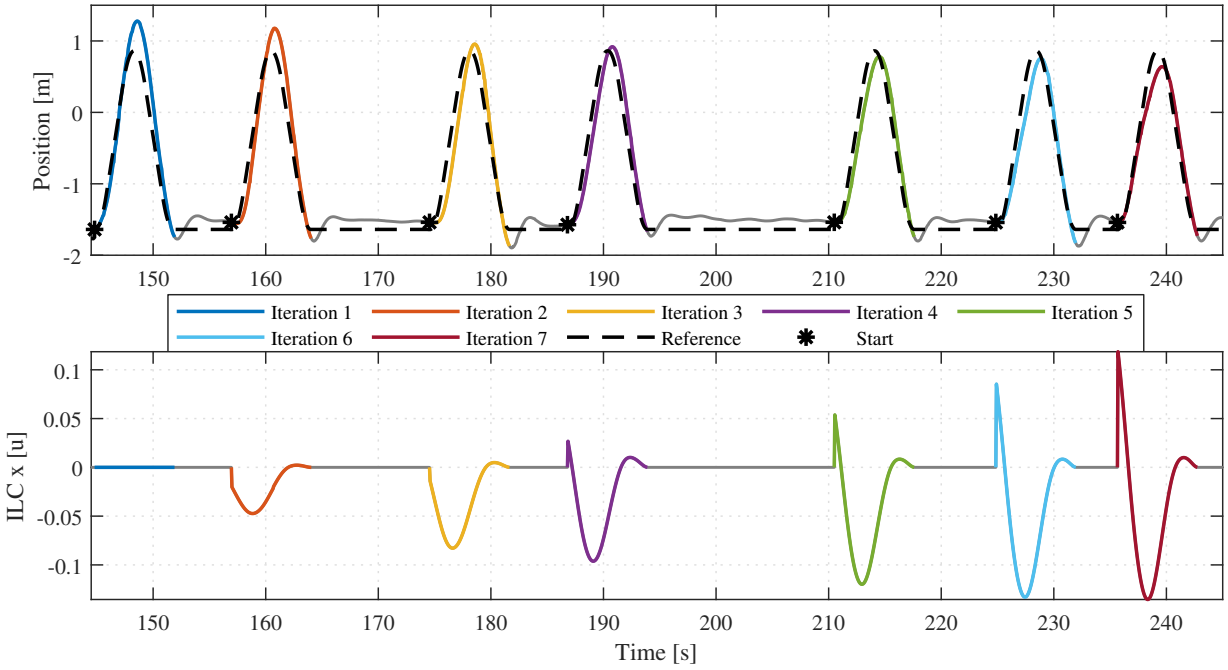


Fig. 23. Experimental results of Section V-B. The x-position and x NO-ILC feedforward values of the first seven iterations from Fig. 12b are shown. The grey lines represent the position and ILC values during the waiting state.

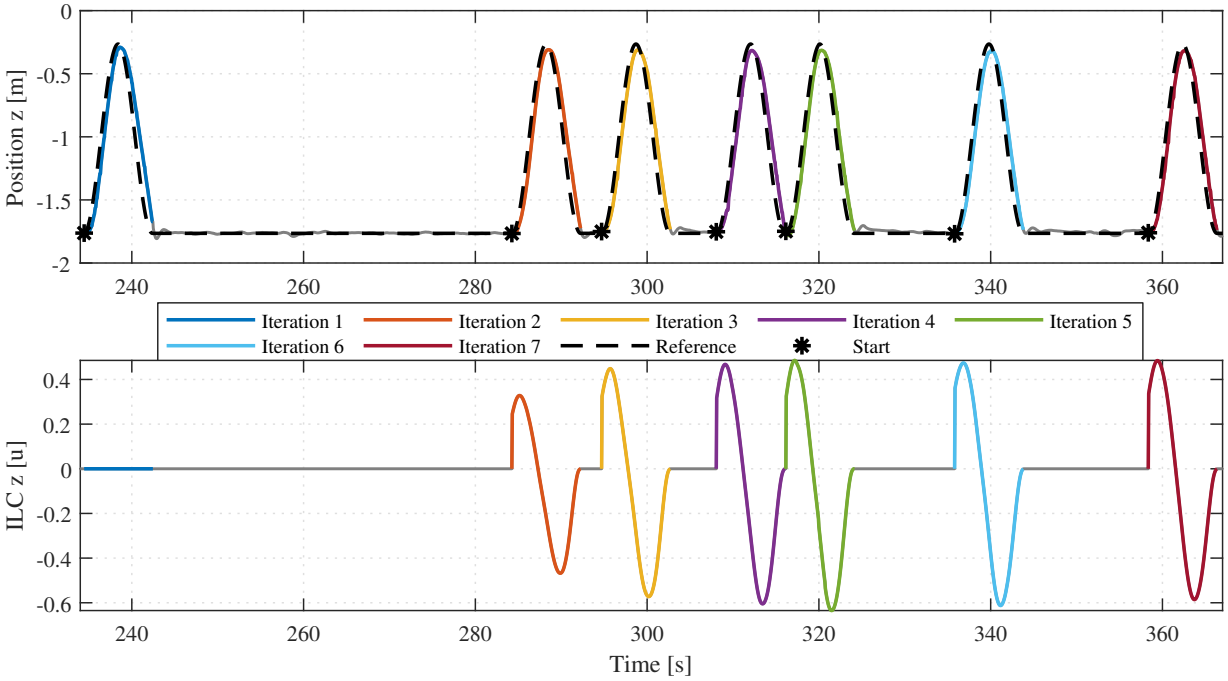


Fig. 24. Experimental results of Section V-C1. The z-position and z NO-ILC feedforward values of the first seven iterations from Fig. 14 are shown. The grey lines represent the position and ILC values during the waiting state.

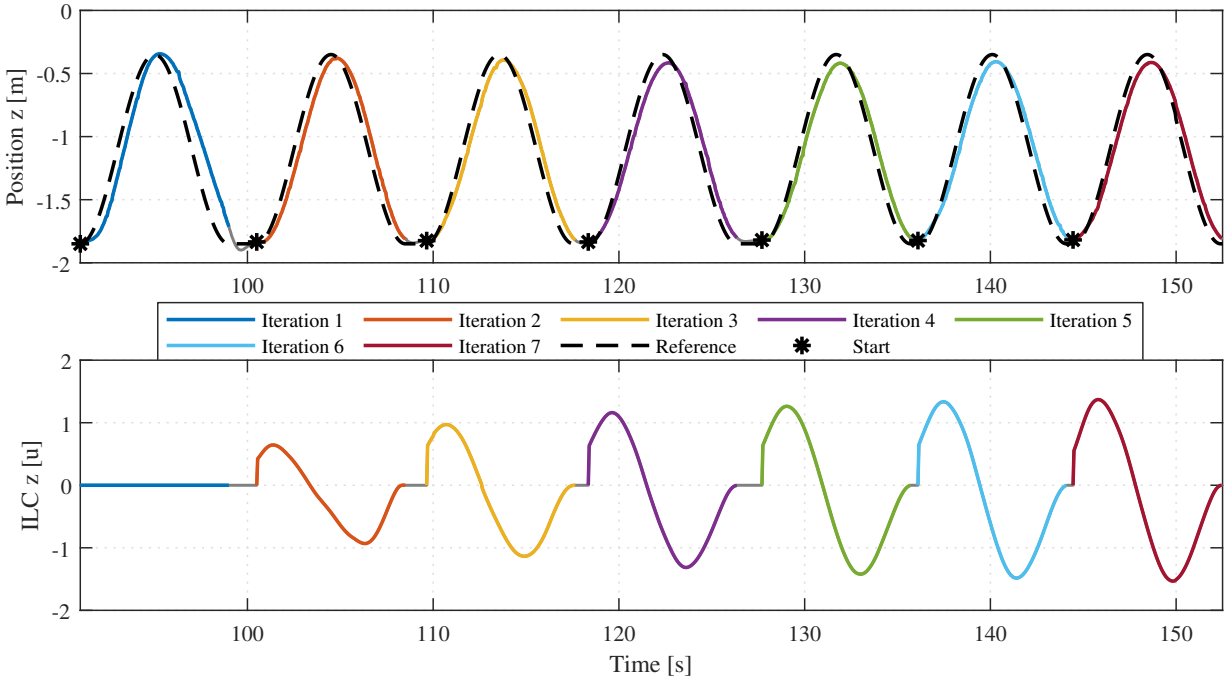


Fig. 25. Experimental results of Section V-C2. The z-position and z NO-ILC feedforward values of the first seven iterations from Fig. 16 are shown. The grey lines represent the position and ILC values during the waiting state.