AE5310: Thesis Control and Operations

Improving the Performance of INDI Flight Control for a Quadrotor in the Ceiling Effect

written by

Max Kemmeren 4553209

in partial fulfillment of the requirements for the degree of

Master of Science Aerospace Engineering

at the University of Technology Delft.

supervised by

Ir. Christophe de Wagter
Dr. Ir. Ewoud J. J. Smeur

October 15, 2021

Table of Contents

Part I

Research Paper

Improving the Performance of INDI Flight Control for a Quadrotor in the Ceiling Effect

Max Kemmeren
Delft University of Technology, 2629 HS, the Netherlands

ABSTRACT

As the application areas of Unmanned Aerial Vehicles (UAVs) keep expanding, new flight areas are encountered more often. Small UAVs, named Micro Air Vehicles (MAVs), even fly in areas like sewage pipes. These areas introduce new difficulties such as aerodynamic effects caused by the ground and/or ceiling. In this paper two main contributions are presented that deal with the aerodynamic effects caused by the ceiling: 1) an adaptive model describing the ceiling effect using onboard measurements, which can be altered to describe other aerodynamic effects that occur when flying in constrained spaces, 2) incorporating the adaptive model into an Incremental Nonlinear Dynamic Inversion (INDI) controller. The controller is implemented and tested onto a MAV (Crazyflie). The results have shown stability improvements for close ceiling flight. Moreover the minimal distance the MAV can fly from the ceiling is decreased using the new controller.

1 INTRODUCTION

Within every sector technology advances with time. This is no different for the industry of Unmanned Aerial Vehicles (UAVs). In the year 2014 there was already an expenditure of 700 million dollar on commercial UAVs. The applications for those UAV range from photography and wild life counting, up to surveillance and inspections [1]. Miniaturization and automation of UAVs played a major role in recent years making the UAV smaller and smaller, obtaining the name Micro Air Vehicles (MAVs), which expands the application field. Flying indoors is such an application. Sampedro et al., who developed a MAV to carry out search and rescue operations [2], is one example. Another example is the Airburr drone by Briod et al. [3], also a search and rescue drone which is designed for robustness and is capable of dealing with collisions. Still, the areas explored by those MAVs are relatively large compared to size of the MAV. Smaller areas are also already being explored already, such as sewage pipes, inspected with a caged drone developed by the company Flyability[1].

These smaller areas introduce new challenges related to the airflow and behavior of the MAV. Different aerodynamic effects come into play when flying close to walls, ground and/or ceiling. Several solutions to deal with those effects already exist. A solution for dealing with the ceiling effect was introduced by Jimenez et al. [4], who developed a nonlinear Lyapunov controller incorporating a ceiling effect model. This makes it possible to fly close to the underside of a bridge deck for inspection purposes. Other work from Wei et al. enhanced a PID controller with a model reference controller, this time incorporating a model from the ground effect [5] to fly more stable in ground effect. Both solutions incorporate an effect specific model, which makes it more difficult to also include other aerodynamic effects.

The most used controller type nowadays within MAVs is the Proportional Integral Derivative (PID) controller, but it has difficulty handling highly nonlinear systems and large disturbances [6, 7]. Recently, a new controller was introduced into the MAV industry: the Incremental Nonlinear Dynamic Inversion (INDI) controller. This controller essentially takes over the role of the integral term in the PID and has proven to deal with nonlinearities, disturbances and model uncertainties rather well [8]. Combining the disturbance rejection properties of the INDI controller with the knowledge from the aerodynamic effects led to this work. In this paper two main contributions are presented: 1) an adaptive model describing the ceiling effect using onboard measurements, which can be altered to describe other aerodynamic effects that occur when flying in constrained spaces, 2) incorporating the adaptive model into the INDI controller. These contributions are implemented and demonstrated on a Crazyflie quadcopter. This is a commercially available quadcopter by the company Bitcraze[2], for which the firmware is publicly available on Github[3].

The outline of this paper is as follows: First the ceiling effect is discussed in section 2. Secondly, section 3 deals with the derivation of the altered INDI controller. Then section 4 explains the RBF function and parameter estimation method. Section 5 discusses the implementation of the INDI controller onto the Crazyflie system and explains the experimental setup. Lastly, section 6 elaborates on the results from the conducted flight experiments.

[1] https://www.flyability.com/

[2] https://www.bitcraze.io/

[3] https://github.com/bitcraze

2 CEILING EFFECT

The ceiling effect is a lesser known phenomenon within the aircraft industry due to aircraft normally flying in open air without any constraints above. For MAVs this might not be the case, as they can be deployed inside a confined area. As with a ground plane causing an increase in lift for a propeller and or wing, the ceiling has a similar effect. By obstructing the flow, the induced velocity created by the rotor disk decreases, which relates to an increase in generated thrust [9]. Opposite to the ground effect, which creates an air cushion between the MAV and the ground plane, the ceiling effects causes the MAV to be attracted towards the ceiling, which may lead to a collision.

The ceiling effect is based on the same physical principle as the ground effect. This lead to several researchers investigating if the ground effect models can be used to model the ceiling effect. One of those models is shown in Equation 1 [10]. This relationship shows the ratio of generated thrust close to ground T_g over free air thrust T_∞, with rotor radius R and distance from the ground z_g.

$$\frac{T_g}{T_\infty} = \frac{1}{1 - \frac{R^2}{16 z_g^2}} \qquad \text{with} \quad z_g > 0.25R \tag{1}$$

Results for the experiment conducted by Conyers et al. [9] are shown in Figure 1 and Figure 2. These results show that the ceiling effect matches Equation 1 quite well, for both the single propeller and multi propeller case. It can be noticed that the ceiling effect starts to rise at a closer distance towards the surface compared to the ground effect model for both the single and multi-propeller case. However, the drawback of the relationship in Equation 1 is the singularity when reaching a distance of a quarter rotor radius. This singularity was tackled by He et al. [11] who created an relationship without this singularity as shown in Equation 2.

$$\frac{T_g}{T_\infty} = C_a e^{-\frac{C_b}{R} z_g} + 1 \tag{2}$$

Once more the thrust ratio is expressed in this exponential expression, where C_a and C_b are model parameters that must be determined via experimental flight measurements. This expression will be used during the development of the new INDI controller. Not having a singularity is an advantage due to not having to deal with parameters going to infinity when modeling the aerodynamic effect. Even though Equation 2 was developed for describing the ground effect this is not deemed as a problem. This paper is aimed at developing an adaptive model for the ceiling effect, the theoretical model will only be used during development, and for comparison with the adaptive model in the experiments.

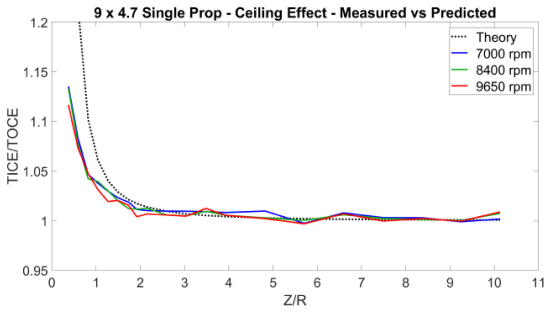


Figure 1: Ceiling effect for single propeller at different RPM, compared to the theory model in Equation 1 [9].

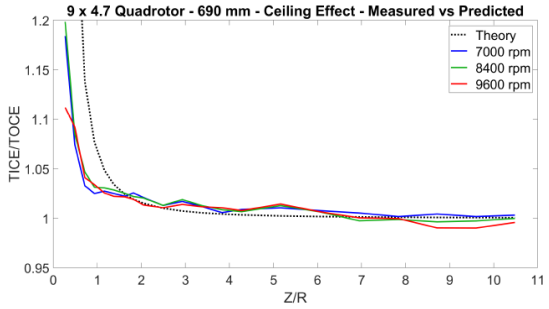


Figure 2: Ceiling effect for quadcopter configuration at different RPM, compared to the theory model in Equation 1 [9].

Besides an increase in induced velocity, Sanchez et al. [12] also observed another phenomenon when flying close to the ceiling. The ceiling effect causes the RPM of the rotors to increase when providing constant power to the motors. This effect leads to an increment in generated thrust.

The ceiling effect is shown to be similar to the ground effect but has some key differences that should be noted and considered within the controller design. First of all, the ceiling effect starts to occur closer to the ceiling plane than the ground effect. Secondly, the effect is not only due to an increase in induced velocity but also affects the rotor RPM.

3 INDI

The controller that will be used for the ceiling effect compensation will be discussed in this section. First of all, a coordinate system is defined, after which the inner attitude control loop and outer position control loop are derived.

3.1 Coordinate System

For the derivation of the controller a coordinate system was defined. Both the world (O) and body fixed (B) coordinate systems are North East Down (NED). To transform between the two coordinate systems an Euler angle transformation is used. The coordinate systems are shown in Figure 3.

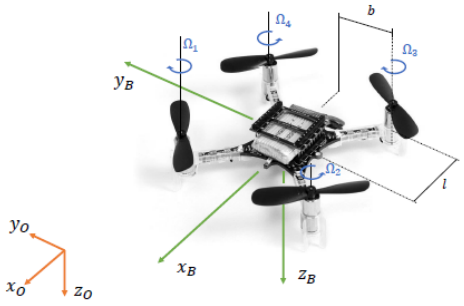


Figure 3: Coordinate system [13].

3.2 *Inner Attitude Control Loop*

To start, the inner attitude loop is derived, which is based on the angular momentum equation shown in Equation 3 [14]. This equation is set in body coordinates, with the vehicle inertia $\boldsymbol{I}_v$, body angular velocity and acceleration $\boldsymbol{\Omega}, \dot{\boldsymbol{\Omega}}$ and total moment $\boldsymbol{M}$.

$$\boldsymbol{I}_v\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \boldsymbol{I}_v\boldsymbol{\Omega} = \boldsymbol{M} \tag{3}$$

The total moment consists of the moment generated by the thrust force offset from the center of gravity ($\boldsymbol{M}_c$), the counter moment due to the spinning of the rotors ($\boldsymbol{M}_r$) and the aerodynamic moment ($\boldsymbol{M}_a$, which depends on the body rate $\boldsymbol{\Omega}$ and the MAVs velocity vector $\boldsymbol{v}$). Therefore the total moment is equal to:

$$\boldsymbol{M} = \boldsymbol{M}_a + \boldsymbol{M}_c - \boldsymbol{M}_r$$

First the moment generated by the rotors due to rotation, gyroscopic moment, is further elaborated on. It is assumed that the rotors are flat in the z-axis, resulting in $I_{r_{xz}} I_{r_{yz}} = 0$. The gyroscopic moment is shown in Equation 4, where the rotor rotational speed is indicated by $\boldsymbol{\omega}$, the rotor inertia by $\boldsymbol{I}_r$ and i indicates the rotor number (quadcopter has four rotors).

$$\begin{aligned}\boldsymbol{M}_r &= \sum_{i=1}^{4} \boldsymbol{M}_{r_i} = \sum_{i=1}^{4} (-1)^{i+1} \begin{bmatrix} I_{r_{zz}}\Omega_y\omega_{z_i} \\ -I_{r_{zz}}\Omega_x\omega_{z_i} \\ I_{r_{zz}}\dot{\omega}_{z_i} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I_{r_{zz}} & -I_{r_{zz}} & I_{r_{zz}} & -I_{r_{zz}} \end{bmatrix} \begin{bmatrix} \dot{\omega}_{z_1} \\ \dot{\omega}_{z_2} \\ \dot{\omega}_{z_3} \\ \dot{\omega}_{z_4} \end{bmatrix} \\ &+ \begin{bmatrix} I_{r_{zz}}\Omega_y & -I_{r_{zz}}\Omega_y & I_{r_{zz}}\Omega_y & -I_{r_{zz}}\Omega_y \\ -I_{r_{zz}}\Omega_x & I_{r_{zz}}\Omega_x & -I_{r_{zz}}\Omega_x & I_{r_{zz}}\Omega_x \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_{z_1} \\ \omega_{z_2} \\ \omega_{z_3} \\ \omega_{z_4} \end{bmatrix}\end{aligned} \tag{4}$$

For the implementation on the crazyflie platform, discussed in section 5 the formula is altered. It is assumed that the second term of Equation 4 is negligible compared to all other generated moments [8]. The motor mixing (transforming controller commands into motor commands) is done outside the control loop. This means that the control loop needs to provide roll, pitch and yaw rate commands. These commands however need to be provided in motor units (subscript m), explained in section 5. Rewriting Equation 4 considering the stated assumption and unit conversion, translates into Equation 5, where $\boldsymbol{G}_2$ is the rotor control effectiveness matrix (a matrix that transforms the motor command in each axis to the actual moment).

$$\boldsymbol{M}_r = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & G_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{p}_m \\ \dot{q}_m \\ \dot{r}_m \end{bmatrix} = G_2\dot{\boldsymbol{\Omega}}_m \tag{5}$$

The thrust force moment also generated by the propellers, due to an offset from the center of gravity, can be written as Equation 6. The thrust force is assumed to rise quadratically with the motor rotational speed. k_1 and k_2 are the force and moment constant of the rotors, which are geometry dependent. b, l are defined in Figure 3, which are the moment arms.

$$\boldsymbol{M}_c = \begin{bmatrix} -bk_1 & bk_1 & bk_1 & -bk_1 \\ lk_1 & lk_1 & -lk_1 & -lk_1 \\ k_2 & -k_2 & k_2 & -k_2 \end{bmatrix} * \boldsymbol{\omega_i}^2 \tag{6}$$

As with the gyroscopic moment, Equation 6 is also altered for implementation. Once more the motor mapping is done after the control loop, thus the moment will be expressed by a control effectiveness matrix ($\boldsymbol{G}_1$) multiplied by the body rate motor commands $\boldsymbol{\Omega}_m$. This is shown in Equation 7.

$$\boldsymbol{M}_c = \begin{bmatrix} G_p & 0 & 0 \\ 0 & G_q & 0 \\ 0 & 0 & G_r \end{bmatrix} \begin{bmatrix} p_m \\ q_m \\ r_m \end{bmatrix} = G_1\boldsymbol{\Omega}_m \tag{7}$$

Using all the information from the expanded moments, Equation 3 can be extended to Equation 8. Subscript v indicates the vehicle, and symbol $\boldsymbol{v}$ the vehicles velocity vector.

$$\begin{aligned}\dot{\boldsymbol{\Omega}} = &- \boldsymbol{I}_v^{-1}(\boldsymbol{\Omega} \times \boldsymbol{I}_v\boldsymbol{\Omega}) + \boldsymbol{I}_v^{-1}\boldsymbol{M}_a(\boldsymbol{\Omega}, \boldsymbol{v}) \\ &+ \boldsymbol{I}_v^{-1}\boldsymbol{M}_c(\boldsymbol{\Omega}_m) - \boldsymbol{I}_v^{-1}\boldsymbol{M}_r(\dot{\boldsymbol{\Omega}}_m)\end{aligned} \tag{8}$$

By grouping terms, Equation 8 can be written as Equation 9.

$$\dot{\boldsymbol{\Omega}} = \boldsymbol{F}(\boldsymbol{\Omega}, \boldsymbol{v}) + \boldsymbol{G}_1\boldsymbol{\Omega}_m + T_s^{-1}\boldsymbol{G}_2\dot{\boldsymbol{\Omega}}_m \tag{9}$$

For which:

$$\boldsymbol{F}(\boldsymbol{\Omega}, \boldsymbol{v}) = \boldsymbol{I}_v^{-1}(\boldsymbol{M}_a - \boldsymbol{\Omega} \times \boldsymbol{I}_v\boldsymbol{\Omega})$$

The T_s^{-1} term accompanying $\boldsymbol{G_2}$ is added to ease future calculations. To derive the INDI controller a first order Taylor

expansion is applied to Equation 9, which results in Equation 10.

$$\begin{aligned}\dot{\boldsymbol{\Omega}}_t = \dot{\boldsymbol{\Omega}}_{\mathbf{0}} &+ \frac{\delta}{\delta \Omega}\boldsymbol{F}(\boldsymbol{\Omega}, \boldsymbol{v_0})|_{\boldsymbol{\Omega}=\boldsymbol{\Omega_0}}(\boldsymbol{\Omega} - \boldsymbol{\Omega_0}) \\ &+ \frac{\delta}{\delta v}\boldsymbol{F}(\boldsymbol{\Omega_0}, \boldsymbol{v})|_{\boldsymbol{v}=\boldsymbol{v_0}}(\boldsymbol{v} - \boldsymbol{v_0}) \\ &+ \frac{\delta}{\delta \Omega_m}\boldsymbol{G_1}(\boldsymbol{\Omega}_m - \boldsymbol{\Omega}_{m_0}) \\ &+ \frac{\delta}{\delta \dot{\Omega}_m}\boldsymbol{G_2}(\dot{\boldsymbol{\Omega}}_m - \dot{\boldsymbol{\Omega}}_{m_0})\end{aligned} \tag{10}$$

The derivative to body rate ($\boldsymbol{\Omega}$) and body velocity ($\boldsymbol{v}$) will be neglected, as those terms will be much smaller than the motor related terms ($\boldsymbol{\Omega}_m$ and $\dot{\boldsymbol{\Omega}}_m$). This can be done based off the principle of time separation [15], which is based on the assumption that the actuators are sufficiently fast and thus have more effect than the change in aerodynamic and precession moments. Isolating the change in motor command $\tilde{\boldsymbol{\Omega}}_m = \boldsymbol{\Omega}_m - \boldsymbol{\Omega}_{m_0}$ leads to Equation 11.

$$\tilde{\boldsymbol{\Omega}}_{\boldsymbol{m}} = \boldsymbol{G_1}^{-1}((\dot{\boldsymbol{\Omega}} - \dot{\boldsymbol{\Omega}}_0) + \boldsymbol{G_2}(\dot{\boldsymbol{\Omega}}_{\boldsymbol{m}} - \dot{\boldsymbol{\Omega}}_{\boldsymbol{m_0}})) \tag{11}$$

As the derivative of the motor command ($\dot{\boldsymbol{\Omega}}_m$) cannot be measured directly, the following substitution is made: $\dot{\boldsymbol{\Omega}}_m = (\boldsymbol{\Omega}_m - \boldsymbol{\Omega}_m z^{-1})T_s^{-1}$. This is a first order discrete filter which can later be implemented on the system. A discrete filter has been chosen because computers work in the discrete time domain (z is the discrete time parameter). Substituting this filter in Equation 11 leads to Equation 12.

$$\tilde{\boldsymbol{\Omega}}_m = (\boldsymbol{G_1} + \boldsymbol{G_2})^{-1}((\boldsymbol{\nu}_{\dot{\boldsymbol{\Omega}}} - \dot{\boldsymbol{\Omega}}_f) + \boldsymbol{G_2} z^{-1}(\boldsymbol{\Omega}_m - \boldsymbol{\Omega}_{\boldsymbol{m_f}})) \tag{12}$$

Lastly $\dot{\boldsymbol{\Omega}}$ is substituted by $\boldsymbol{\nu}_{\dot{\boldsymbol{\Omega}}}$, which is called the virtual input and will be the input provided by a PD controller. To obtain the angular acceleration the gyroscope measurements need to be differentiated. As differentiation amplifies noise, the already noisy gyroscope measurements are filtered to reduce this (indicated by subscript f). This second order filter will introduce a delay into that signal. Within the Taylor expansion all signals need to be from the same point in time, to make all signals synchronous other measurements are filtered equally.

The inner loop control structure is shown in Figure 4. The motor dynamics $A(z)$ has been determined by E. Volodscoi [13]. The INDI controller linearizes the system, and makes it possible to use a standard cascaded PD control loop to generate the virtual input using attitude measurements. The control loop for the virtual input is shown in Figure 5.

3.3 Outer Position Control Loop

To be able to compensate for the ceiling effect the position INDI control loop is altered, for which the original can be found in the work of Smeur et al. [8]. It all starts from the Euler acceleration equation in world frame, which is shown in Equation 13 [14]. With gravitational constant g, mass m, thrust $\boldsymbol{T}_N$ in world frame (depending on the attitude $\boldsymbol{\eta}$), the aerodynamic forces $\boldsymbol{T}_A$ dependent on body velocity $\boldsymbol{v}$ and wind vector $\boldsymbol{\chi}$, and the Radial Basis Function (RBF) model that will be used to describe the ceiling effect ratio (similar to the output of Equation 2). The RBF function will approximate the ratio of extra thrust generated by the ceiling at a certain distance and will thus be multiplied by the generated thrust to get the actual thrust. The developed RBF function structure will be further elaborated on in subsection 5.2.

$$\ddot{\boldsymbol{\xi}} = \boldsymbol{g} + \frac{1}{m}(\boldsymbol{T}_N(\boldsymbol{\eta}, \boldsymbol{T}) \cdot RBF + \boldsymbol{T_A}(\boldsymbol{v}, \boldsymbol{\chi})) \tag{13}$$

$$RBF = w_1 + \sum_{i=2}^{7} w_i e^{-a_i(z_c - c_i)^2}$$

As the thrust is generated along the negative body z axis it needs to be rotated from the body fixed coordinate system to the world fixed coordinate system, using Equation 14 in which ϕ, θ, ψ are the Euler angles in respective rotation order.

$$\begin{aligned}\boldsymbol{T}_N &= {}^N_B\boldsymbol{R} * \boldsymbol{T}_B \\ &= \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ c\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -T_B \end{bmatrix} \\ &= \begin{bmatrix} -T_B(c\psi s\theta c\phi + s\psi s\phi) \\ -T_B(s\psi s\theta c\phi - c\psi s\phi) \\ -T_B(c\theta c\phi) \end{bmatrix}\end{aligned} \tag{14}$$

As within the attitude loop derivation, a first order Taylor expansion is applied to Equation 13, resulting in Equation 15. The aerodynamic force and its derivatives are omitted from the equation for the following reasons. First of all, for the derivatives with respect to $\boldsymbol{v}$ and $\boldsymbol{\chi}$ there are no good estimates available, with wind being unpredictable and thus the best estimate of change is assumed to be zero. Secondly, no aerodynamic model is employed. This however does not mean that the aerodynamic forces are completely omitted as they are measured within the acceleration term $\ddot{\boldsymbol{\xi}}_0$ [8]. Moreover it is assumed that the change in yaw angle (ψ) is small and thus the derivative is neglected.

$$\begin{aligned}\ddot{\boldsymbol{\xi}} = \ddot{\boldsymbol{\xi}}_0 &+ \frac{\delta}{\delta \phi}\frac{1}{m}\boldsymbol{T}_N(\boldsymbol{\eta}_0, \boldsymbol{T}_{B_0})RBF|_{\phi=\phi_0}(\phi - \phi_0) \\ &+ \frac{\delta}{\delta \theta}\frac{1}{m}\boldsymbol{T}_N(\boldsymbol{\eta}_0, \boldsymbol{T}_{B_0})RBF|_{\theta=\theta_0}(\theta - \theta_0) \\ &+ \frac{\delta}{\delta T}\frac{1}{m}\boldsymbol{T}_N(\boldsymbol{\eta}_0, \boldsymbol{T}_B)RBF|_{T_B=T_{B_0}}(T_B - T_{B_0}) \\ &+ \frac{\delta}{\delta z_c}\frac{1}{m}\boldsymbol{T}_N(\boldsymbol{\eta}_0, \boldsymbol{T}_{B_0})RBF|_{z_c=z_{c_0}}(z_c - z_{c_0})\end{aligned} \tag{15}$$

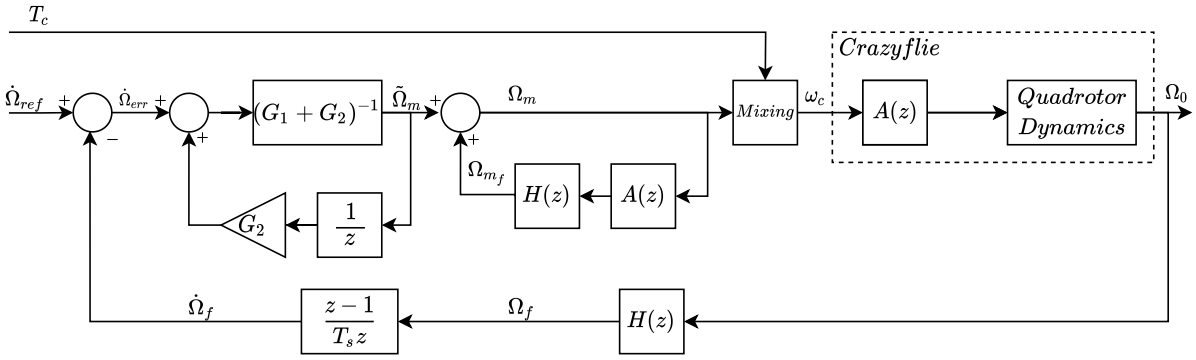


Figure 4: INDI attitude control loop.

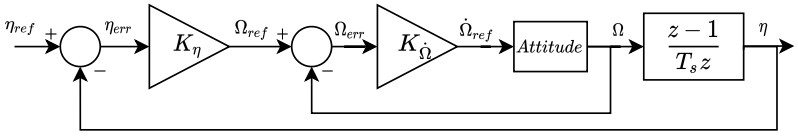


Figure 5: INDI attitude virtual input PD loop.

Applying the derivatives and grouping the terms leads to Equation 16

$$\begin{aligned}\ddot{\boldsymbol{\xi}} = \ddot{\boldsymbol{\xi}}_0 &+ \frac{1}{m}\boldsymbol{G}_4(\boldsymbol{\eta}_0, T_{B_0}, z_{c_0})(\boldsymbol{u}-\boldsymbol{u}_0) \\ &+ \frac{1}{m}\boldsymbol{H}(\boldsymbol{\eta}_0, T_{B_0}, z_0)(z_c - z_{c_0}) \\ \boldsymbol{u} &= \begin{bmatrix}\phi & \theta & T_B\end{bmatrix}\end{aligned} \tag{16}$$

$$\boldsymbol{G}_4(\boldsymbol{\eta}, T_B, z_c) = \begin{bmatrix} (c\phi s\psi - s\phi c\psi s\theta) & (c\phi c\psi c\theta) & s\psi s\phi + c\psi s\theta c\phi \\ (-s\phi s\psi s\theta - c\psi c\phi) & (c\phi s\psi c\theta) & s\psi s\theta c\phi - c\psi s\phi \\ -c\theta s\phi & -s\theta c\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} T_B \\ T_B \\ 0 \end{bmatrix} \cdot RBF$$

$$\boldsymbol{H}(\boldsymbol{\eta}, T_B, z_c) = \begin{bmatrix} s\psi s\phi + c\psi s\theta c\phi \\ s\psi s\theta c\phi - c\psi s\phi \\ c\phi c\theta \end{bmatrix} \cdot T_B \cdot \sum_{i=2}^{7} -2a_i(z_c - c_i)w_i e^{-a_i(z_c-c_i)^2}$$

Next up the controller input is isolated, namely the roll, pitch and thrust increment $\tilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}_0$. This leads to Equation 17. As with the gyroscope the accelerometer measurements are noisy and need filtering. Once more to account for the delay all other signals will be filtered with the same filter such that they are synchronous, indicated with subscript f. At last the wanted acceleration $\ddot{\boldsymbol{\xi}}_0$ is substituted with the virtual input symbol $\boldsymbol{\nu}_{\ddot{\boldsymbol{\xi}}}$.

$$\begin{aligned}\tilde{\boldsymbol{u}}_c = &m\boldsymbol{G}_4^{-1}(\boldsymbol{\eta}_f, T_{B_f}, z_{c_f})((\boldsymbol{\nu}_{\ddot{\boldsymbol{\xi}}} - \ddot{\boldsymbol{\xi}}_f) \\ &- \frac{1}{m}\boldsymbol{H}(\boldsymbol{\eta}_f, T_{B_f}, z_{c_f})(z_c - z_{c_f}))\end{aligned} \tag{17}$$

The adapted position control loop can be seen in Figure 6, with once more a PD controller generating the virtual input shown in Figure 7. When comparing this newly derived INDI loop to the original INDI two main differences arise. The first difference is the extra term in the control effectiveness matrix G_4, namely the RBF function approximating the extra thrust caused by the ceiling effect. This term is always larger than one by definition and thus taking the inverse decreases the final calculated increment, so compensating for the ceiling effect. The second difference is the extra term H. This term will be negative when going closer to the ceiling as the RBF derivative has a negative slope and the displacement $(z_c - z_{c_f})$ will also be negative, once more decreasing the increment. These two extra terms compensate for the increase in thrust caused by the lower induced velocity and the gain in rotor rotational speed for constant motor power (ceiling effect). The gain $K_{\dot{\xi}}$ is elaborated on in section 5.

4 Parameter Estimation Model

During the controller derivation in subsection 3.3, a RBF model was introduced for modeling the ceiling effect. This section discusses the concept of the RBF function and how the model parameters are estimated.

The ceiling effect is dependent on the geometry. To have a ceiling effect approximation model estimated via onboard measurements makes it adaptable to different types of MAV, as long as the required measurements are available. One method for function approximation is making use of the RBF. This function can be described by Equation 18 and has a bell

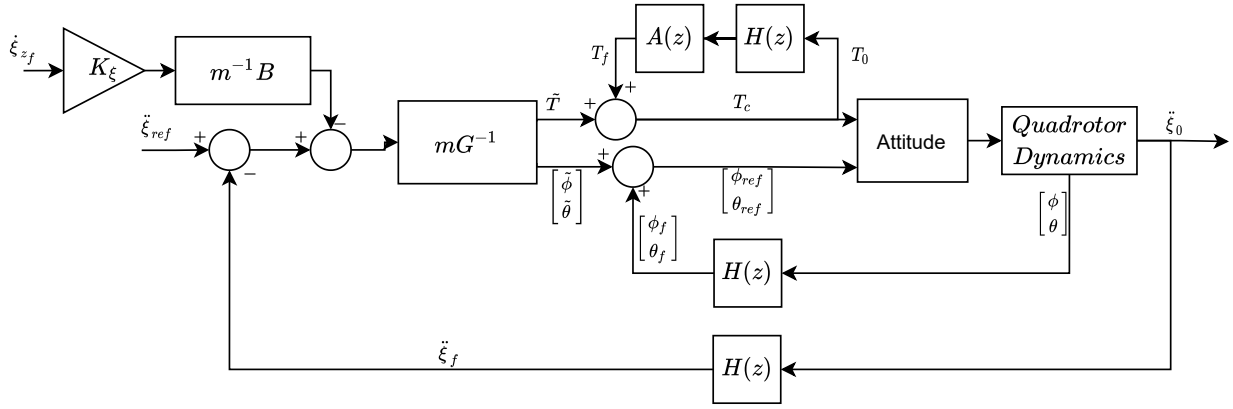


Figure 6: New INDI position control loop.

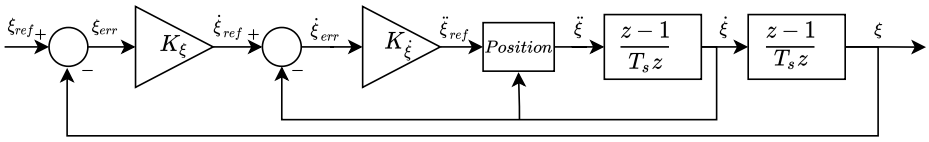


Figure 7: INDI position virtual input PD loop.

shaped curve.

$$f(z_c) = we^{-a(z_c-c)^2} \tag{18}$$

Within the function there are several tuning parameters to change the shape of the bell curve. The weight term w changes the height of the curve, a is for adjusting the width and lastly, c is the placement of the maximum of the bell curve along the horizontal axis. For function approximation a sum of multiple RBF functions may be used as will be seen in subsection 5.2. To simplify the parameter estimation the width and centers of the RBF functions are assumed to be fixed, leaving only the weight term as a variable.

For parameter fitting of the RBF function, Recursive least Squares (RLS) with a forgetting factor (λ) will be used [16]. This method only uses the latest measurements to change the fit, thus having a low computational expense (small regression matrix). The algorithm first calculates a Kalman gain according to Equation 19.

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_k \cdot \boldsymbol{a}_{k+1}^T (\boldsymbol{a}_{k+1} \cdot \boldsymbol{P}_k \cdot \boldsymbol{a}_{k+1}^T + \lambda)^{-1} \tag{19}$$

The regression matrix $\boldsymbol{a}(z_c)$ is a row vector with the number of entries ($p(z_c)$) equal to the number of model parameters that need to be estimated.

$$\boldsymbol{a}(z_c) = \begin{bmatrix} 1 & p_1(z_c) & \dots & p_M(z_c) \end{bmatrix}$$

λ is the forgetting factor ($0 < \lambda \leq 1$), which influences the adaptability of the fit to new incoming data. New data has the most influence on the fit as λ approaches one. Then Equation 20 is used to calculate the model parameters $\hat{\theta}$. $y(z_c)$ is the measurement that will be estimated by the fitting function ($\hat{y}(z_c) = \boldsymbol{a}(z_c)\hat{\boldsymbol{\theta}}$).

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \boldsymbol{K}_{k+1}(y_{k+1} - \boldsymbol{a}_{k+1}\hat{\boldsymbol{\theta}}_k) \tag{20}$$

When the new model parameters are determined, the covariance matrix is updated with Equation 21. The diagonal covariance matrix P indicates how likely a model parameter will change when new data comes in. The smaller this value, the smaller the change is expected to be.

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{K}_{k+1} \cdot \boldsymbol{a}_{k+1} \cdot \boldsymbol{P}_k \tag{21}$$

5 Implementation and Experimental Setup

The controller was implemented on the Crazyflie. As the MAV has certain specifications and ways of dealing with different internal parameters, this section discusses the adaptations made for the controller to fit the system. Moreover the RBF final form is discussed, together with the needed measurements for the parameter estimation. At last the experimental setup is discussed.

5.1 MAV System and Implementation

To be able to perform tests, a fitting experimental platform needed to be selected, having all the necessary measure-

ment capabilities. The MAV onto which the newly developed INDI controller was implemented is the Crazyflie 2.1 from Bitcraze[4], which is also shown in Figure 3. This small modular MAV is lightweight (32 grams) and has interchangeable sensor modules. The INDI controller needs distance measurements towards the ceiling, for which the multiranger module was installed. This deck adds 5 time of flight (ToF) sensors, one for each horizontal direction and one towards the ceiling. Moreover the flowdeck v2.0 was used for relative positioning such that flight tests could be performed. This deck uses an optical flow sensor towards the floor which is able to estimate displacement, as well as a ToF sensor downwards for height measurements.

During the derivation in section 3 it was already stated that the INDI controller was altered to be compatible with the Crazyflie platform. The original INDI loops have been implemented by E. Volodscoi [13]. To be certain that the implementation was correct the code was validated and the control effectiveness of the inner loop was determined once more. This was done using the SD deck module. This deck could log the needed parameters with the 500 Hz attitude loop rate. Using a least squares technique the effectiveness could be recalculated [8]. Moreover a thrust mapping parameter was determined using a similar approach. As the outer INDI loop provides the thrust in newton and the Crazyflie needs it as a 16 bit integer value, zero meaning no power and 65535 meaning full power.

The adapted INDI loop needs the displacement $(z_c - z_{c_f})$ to be able to calculate the commands. For determining this displacement, the MAV's vertical velocity is multiplied by a time step $K_{\dot{\xi}}$. As the system uses actuators which do not have an instantaneous response the expected displacement is delayed. To determine a valid time step gain, the response of the actuators was investigated. For a unit step input the actuators reached 90% of the final value within 0.3 seconds. A value of 90% of the input is assumed to be enough to say that the actuators have responded to the input, thus a value of 0.3 was used in the final controller as $K_{\dot{\xi}}$.

5.2 Ceiling Ratio Estimation and Curve Fitting

First of all the distance towards the ceiling needs to be obtained at the same rate as the altitude loop rate. With the ToF sensor providing measurements at 10 Hz and the altitude loop running at 100 Hz a complementary filter was applied, shown in Algorithm 1. This filter uses the vertical displacement provided by the Kalman filter, to update the ceiling distance measurements in between sensor measurements from the top ToF. Once a new measurement is taken the estimate is corrected using a gain. As the ToF sensors have a minimal distance measurement limit[5], the correction step was only applied up to a distance of five centimeters away from the ceiling. When flying closer the ceiling distance estimate only relies on the displacement measured by the Kalman filter. A summation of the altitude measurement and ceiling distance estimate is used to generate the set points that command the MAV.

Algorithm 1 Ceiling distance approximation.

```
Initiate z_c with ToF measurement
while z_c < 0.5 meters do
    For each ΔT update the ceiling distance with measured
    displacement from the Kalman filter
    if New ToF sensor measurement and z_c > 5 cm then
        Error = measured distance - estimate distance
        New estimate = previous estimate + 0.1 × Error
    end if
end while
```

For the determination of the RBF fitting function, the theoretical ceiling effect curve given by Equation 2 was taken as reference. Given the rotor diameter of 45 millimeter, this implies that the ceiling effect will become noticeable around ten centimeters distance from the ceiling. To approximate the exponential ceiling effect function a summation of six RBF terms and a constant will be used, each with an adaptable weight factor. The weights will be the model parameters for the RLS. This function is described by Equation 22. The constant is added such that the curve can model a value of one without the extra RBF terms, allowing the RBF terms to more easily model the flat part of the curve.

$$\frac{T(z)}{T_\infty}_{est} = w_1 + \sum_{i=2}^{7} w_i e^{-a_i(z-c_i)^2} \tag{22}$$

As the centers and widths within Equation 22 are assumed fixed they need to be determined. This is done by tuning them in such a way that the curve produced by Equation 22 matches closely to the curve produced by Equation 2. With the exponential form, the RBF centers are biased towards zero and spread out for further distances. For the widths a similar approach is taken, a smaller RBF width close to zero for modeling the steep slope, a wider RBF width further away to model a flat slope. The final parameters are the following:

$$\boldsymbol{a}_{(2-7)} = \begin{bmatrix} 15000 & 10000 & 5000 & 2000 & 1250 & 1000 \end{bmatrix}$$

$$\boldsymbol{c}_{(2-7)} = \begin{bmatrix} 0.0 & 0.005 & 0.015 & 0.03 & 0.06 & 0.1 \end{bmatrix}$$

[4] https://store.bitcraze.io/products/crazyflie-2-1

[5] https://www.st.com/resource/en/datasheet/vl53l1x.pdf

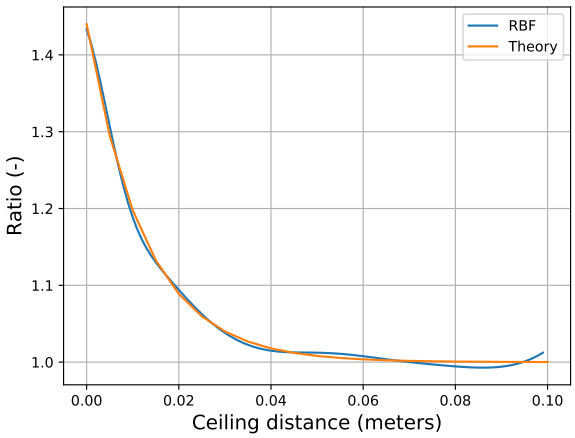


Figure 8: RBF fit onto theoretical model comparison, with model parameters $C_a = 0.44$ and $C_b = -1.8$.

Figure 8 shows a test fit onto the theoretical model. It shows that the chosen RBF function structure is able to model an exponential function. As the real ceiling effect might differ from the theoretical (due to it being the ground effect) the overall shape shall be similar and thus the structure is deemed valid for usage in the INDI controller.

For obtaining the data points, two measurements need to be taken on the Crazyflie. First of all the commanded thrust, which is obtained from the motor command send to each of the four actuators. G. Shi et al. did measurements on the thrust using a force stand and created a relationship that maps the normalized thrust command to thrust force in grams, including a correction for battery drainage [17]. Secondly the actual thrust needs to be measured, which is obtained indirectly via the body z-axis accelerometer onboard. Via Newtons second law, the acceleration measurement can be divided by the mass to get the force. An accelerometer cannot distinguish the body acceleration from the gravitational acceleration. This however was used to its advantage, as during a hover the accelerometer provides a value of one g which translates to a force equal to the body mass. For the thrust ratio this means a value of one, representing flight out of ceiling effect. By dividing the actual thrust measurement from the accelerometer data by the commanded thrust one finds the ratio representing the ceiling effect, used as data to fit the RBF function. Both measurements are taken in the body frame. Once more this data is all filtered similar to the INDI loop, such that all data is synchronous.

5.3 Experiments

To evaluate the newly designed controller, test flights were performed. The flight environment is an open area with a table top used as ceiling, such that the overall altitude of the flight is kept to a limit in case of instability. For comparison each tested controller setup is flown ten times, to gather an average result of the performance. First of all the standard INDI controller is tested to provide a benchmark. The flight conducted is a purely vertical flight were the position is kept constant, to only test the influence of the ceiling effect on altitude stability. The assumption is also made that the roll and pitch angles are small, such that the measured distance of the ToF sensor is the actual distance towards the ceiling. The flight consists of a hover far away from the ceiling, after which the MAV is commanded to a distance of nine, five and three centimeters distance from the ceiling. Each hover altitude is kept constant for a period of around ten seconds, such that the controller has time to command the MAV to the correct altitude and stabilize. The second controller tested is the newly designed INDI controller, with a fixed RBF fit onto the theoretical model (as shown in Figure 8). This is done to be able to compare the real time learned model to the theoretical model. Lastly the new INDI controller with a learned ceiling effect curve is tested. To gather the RBF data, first a learning flight was conducted, which made the MAV hover at different altitudes with increments of 2 centimeters from the ceiling. After the fit was made the model parameters were fixed and the normal test flights were conducted.
Another type of flight was conducted to test the limits of the controllers. For this second experiment the MAV is commanded to climb towards the ceiling with once more increments of two centimeters, with a maximum distance of two centimeters.

6 RESULTS

This section discusses the results collected within the flight experiments. For the first flight experiments the mean and standard deviation of the altitude of the ten flights per controller will be presented. The focus in this paper is the compensation of the ceiling effect, thus only the part of the flight close the ceiling will be shown. This is the part of the flight from approximately 20 to 40 seconds (five and three centimeter distance from the ceiling). This is where the ceiling effect starts rising as seen in Figure 8. The first controller tested was the original INDI controller, for which the result can be seen in Figure 9. The average altitude, commanded altitude and estimated ceiling height are shown. The two climbs in command are the moments where the new altitude command is set, so the first one a climb to five centimeters distance from the ceiling, the second one a climb to a distance of three centimeters. Both times the MAV climbs towards the distance, for which the closer distance shows to be more difficult due to higher oscillations (effect of the ceiling). In both instances the MAV shows to have a steady state error, which is caused by a bias in the accelerometer measurements. The INDI is an incremental controller with similar behavior as an integral gain, meaning convergence to the set command, such that no steady state error exists. However due to the PD controller generating a reference acceleration from the position error,

subtracting the measured acceleration (which has a bias) will provide a virtual input close to zero, meaning the INDI controller will not correct the steady state error.

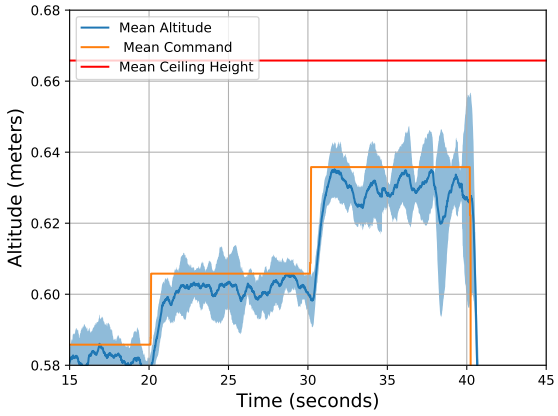


Figure 9: Mean and one standard deviation of altitude, original INDI controller.

Figure 10, shows the results for the new INDI controller where the RBF was fit onto the theoretical model from He et al. One immediately notices that the standard deviation is smaller compared to the original INDI controller. Besides the smaller standard deviation the amplitude of the oscillation is also reduced. Both results show that the two extra terms within the INDI controller counteract the extra generated thrust caused by the ceiling effect. These flights however were using the ground effect model from He et al [11] as model for the ceiling effect, thus even though the model used was not fit using onboard measurements the compensation present already shows improvement. The steady state error is once more present.

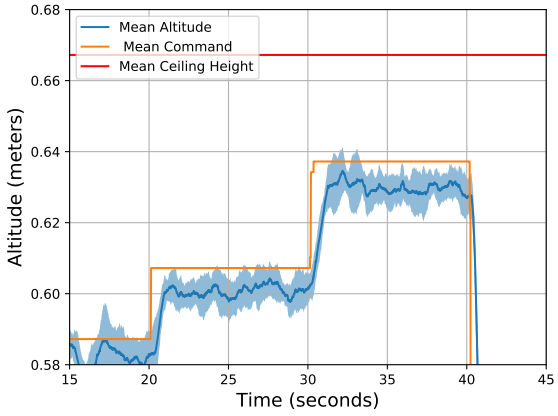


Figure 10: Mean and one standard deviation of altitude, new INDI controller with RBF fit onto theoretical model.

The last controller tested is the new INDI controller where the RBF function is fitted using actual flight data. The result can be seen in Figure 11, where the fitted RLS is shown in Figure 12, in which the theoretical model is also shown for comparison.

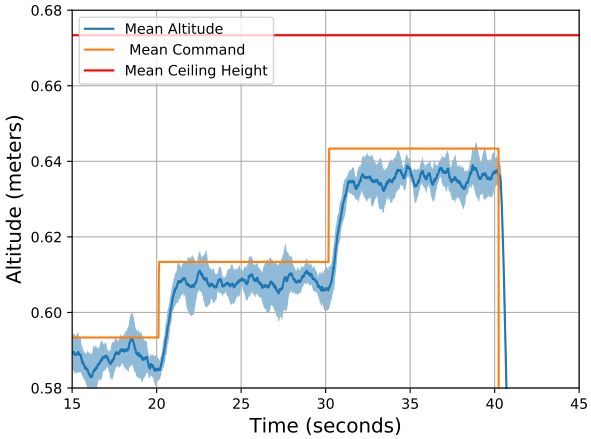


Figure 11: Mean and one standard deviation of altitude, new INDI controller with learned RBF fit.

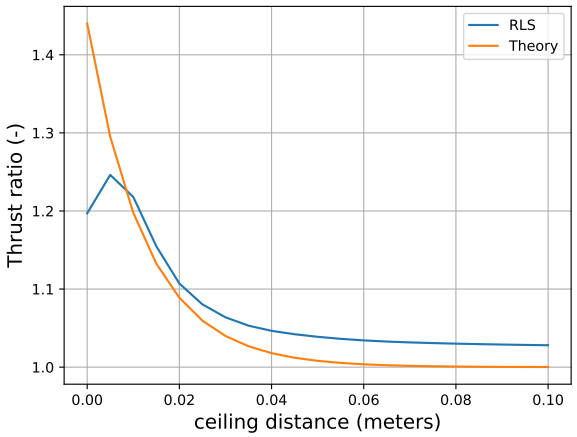


Figure 12: RBF fit using data from the learning flight.

For the flight result it is once more seen that the standard deviation and oscillation is smaller compared to the original INDI controller. Both previous controllers saw a small drop in altitude before settling at the final height. This is not seen in Figure 11 where the MAV climbs directly to its commanded altitude, as there is more compensation the increments will be smaller leading to a slower response. The model fitted using real measurements is seen to have a bias compared to the theoretical model. The bias can have several reasons: First

of all, the propellers and motors could be damaged affecting the performance, which may also cause an error in the thrust model used to calculate the ratio. Secondly, the ceiling effect was shown to be more aggressive than the ground effect model as discussed in section 2, explaining the higher values as well. The drop of around the one centimeter distance is due to not collecting data at that distance.

The results for the second experiment where an even closer ceiling flight was conducted, can be seen in Figure 13. It shows the flight path of a single flight per controller. What can be seen is that the original INDI controller could fly at the distance of four centimeters, but when climbing towards a distance of two centimeters the MAV became unstable, started oscillating and even hit the ceiling as seen at the 50 second mark. This happened for all flights with the original controller. Both the adapted INDI controllers were able to fly stable at the distance of two centimeters, and as seen do not show a large difference in behavior. This has shown that the new INDI controller is better suited to deal with ceiling effect than the original INDI.

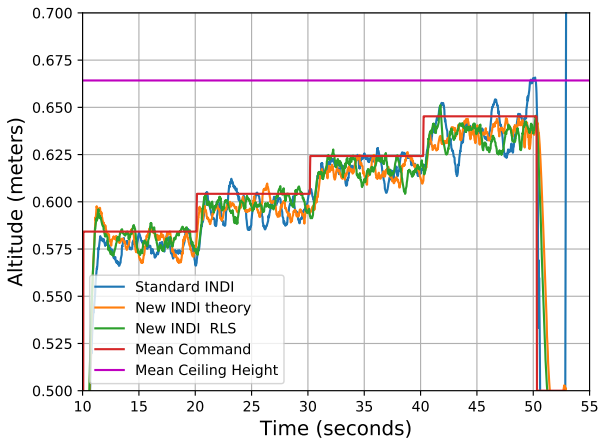


Figure 13: Close ceiling flight experiment, comparison of the three controllers.

7 Conclusion and Future Research

This paper presented an adaptation to the INDI controller to improve flight close towards the ceiling. The inclusion of a ceiling effect model within the dynamical equation changed the structure of the position INDI controller, where the extra terms caused a decrease in the command to act as a compensation for the increase in thrust caused by the ceiling. Flight results have proven that the compensation terms do improve stability, and increased the flight region of the MAV. The RBF model has proven to generate a curve comparable to the expected ceiling effect and has the advantage of being adaptable. With the focus of this paper being the ceiling effect, future research can be done by expanding the RBF model by including the ground effect for example. This can aid the stability when flying in tight spaces where the separation between ground and ceiling is small. Moreover it can be investigated if the time gain combined with the MAVs velocity is the best way of estimating the displacement term ($z_c - z_{c_f}$).

References

[1] B. Rao, A.G. Gopi, and R. Maione. The societal impact of commercial drones. *Technology in Society*, 45:83–90, 2016.

[2] C. Sampedro, A.R. Ramos, H. Bavle, A. Carrio, P. de la Puente, and P. Campoy. A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *Journal of Intelligent & Robotic Systems*, 95:601–627, 2019.

[3] A. Briod, A. Klaptocz, J.C. Zufferey, and D. Floreano. The airburr: A flying robot that can exploit collisions. *International Conference on Complex Medical Engineering*, pages 569–574, 2012.

[4] A.E. Jimenez-Cano, P.J Sanchez-Cuevas, A. Ollero P. Grau, and G. Heredia. Contact-based bridge inspection multirotors: Design, modeling, and control considering the ceiling effect. *IEEE Robotics and Automation Letters*, 4(4):3561–3568, 2019.

[5] P. Wei, S.N. Chan, S. Lee, and Z. Kong. Mitigating ground effect on mini quadcopters with model reference adaptive control. *International Journal of Intelligent Robotics and Applications*, 3:283–297, 2019.

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

[7] E. Fresk and G. Nikolakopoulos. Full quaternion based attitude control for a quadrotor. pages 3864–3869, 07 2013.

[8] E.J.J. Smeur, G. de Croon, and Q.P. Chu. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, 2016.

[9] S.A. Conyers, M.J. Rutherford, and K.P. Valavanis. An empirical evaluation of ceiling effect for small-scale rotorcraft. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 243–249, 2018.

[10] I.C. Cheeseman and W.E. Bennett. The effect of the ground on a helicopter rotor in forward flight. *Aeronautical Research Council Reports and Memoranda*, 1957.

[11] X. He, G. Kou, M. Calaf, and K.K. Leang. In-ground-effect modeling and nonlinear-disturbance observer for multirotor unmanned aerial vehicle control. *Journal of Dynamic Systems, Measurement, and Control*, 141(7), 2019.

[12] P.J. Sanchez-Cuevas, G. Heredia, and A. Ollero. Experimental approach to the aerodynamic effects produced

in multirotors flying close to obstacles. *Third Iberian Robotics Conference, Advances in Intelligent Systems and Computing*, pages 742–752, 2018.

[13] E. Volodscoi. Design, implementation and evaluation of an incremental nonlinear dynamic inversion controller for a nano-quadcopter. Master's thesis, Technische Universität München, Boltzmannstraße 15, 85748 Garching, 4 2020. An optional note.

[14] A. Bedford and W. Fowler. *Engineering Mechanics Dynamics*, pages 507–515. Prentice–Hall, 2008.

[15] P. Simplicio, M. Pavel, E. van Kampen, and Q. Chu. Measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion. *Engineering Practice*, 21(8):1065–107, 2013.

[16] E. Walter, J. Norton, and L. Pronzato. *Identification of Parametric Models: From Experimental Data*. Communications and Control Engineering. Springer, 1997.

[17] G. Shi, X. Shi W. Hönig, Y. Yue, and S.J. Chung. Neural-swarm2: Planning and control of heterogeneous multirotor swarms using learned interactions. *IEEE Transactions on Robotics*, abs/2012.05457, 2020.

Part II

Appendix

A One Dimensional MATLAB Simulation

This section will discuss the one dimensional Matlab simulation, used in testing different situations and concepts. From the literature study it was seen that a feed forward controller had positive effect on the stability near the ceiling, and it never collided the MAV during simulation. For this controller it was seen that the integral gain was omitted, as the feedforward path took over that role. Instead of going further with this feedforward controller there will be looked into the INDI controller. This is an incremental controller, thus also taking over the role of the integral gain. To develop a conceptual design a one dimensional simulation environment was created. This is used to test ideas and provide a benchmark for the actual implementation.

A.1 Derivation

First the structure of the simulation environment and the derivation of the one dimensional INDI controller will be shown. Figure A.1 shows a free body diagram of the simulated environment, the MAV, and the forces acting onto the MAV. The important distances are the total height h_c and distance of MAV body to the ceiling z_c.

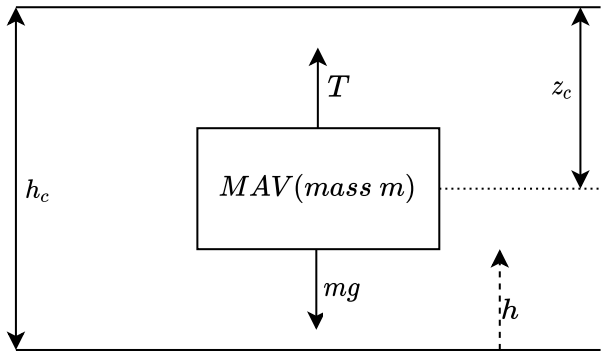


Figure A.1: Free body diagram of the one dimensional simulation setup.

The control parameter in this problem is the thrust, which lets the MAV move up and down. By using Newtons second law Equation A.1 is set up. The forces consist of the thrust T, gravity with m the mass and g the gravitational constant, and a represents the body acceleration in vertical direction.

$$\begin{aligned} F &= m * a \\ T - m * g &= m * a \end{aligned} \tag{A.1}$$

$$\begin{aligned} \dot{a} &= \frac{T_0}{m} - g + \frac{\delta}{\delta T}(\frac{T}{m} - g)|_{T=T_0}(T - T_0) \\ \dot{a} &= a_0 + \frac{1}{m}(T - T_0) \\ \tilde{T} &= m(a - a_0) \end{aligned} \tag{A.2}$$

The INDI controller is derived using a first order Taylor expansion and can be seen in Equation A.2. This shows that the increment in thrust ($\tilde{T} = T - T_0$) can be determined using the acceleration measurement a_0 and the wanted acceleration a. The wanted acceleration will be determined using a PD loop on position. The 0 subscript indicates that those signals need to be from the same point in time.

To be able to simulate the ceiling effect, it was decided to implement the model form He et al. [1] into the dynamical system leading to Equation A.3. In which R is the rotor radius, C_a and C_b (1.8 and 0.44) are model parameters obtained via flight tests, and the distance to the ceiling z_c. Even though this model was developed for the ground effect, the literature study has shown that the ceiling effect is not that different. The advantage of not having the singularity makes it an useful tool for the development of the new controller. Moreover as the final controller will make use of flight data to create a ceiling effect curve this model does not affect the final results.

$$\begin{aligned} F &= m * a \\ T * (1 + C_a e^{\frac{-C_b}{R} z_c}) - m * g &= m * a \end{aligned} \tag{A.3}$$

Once more the INDI controller can be derived, this time with the ceiling effect included. The result is seen in

Equation A.4.

$$\begin{aligned}\dot{a} &= (\frac{T_0}{m} * (1 + C_a e^{\frac{-C_b}{R} z_{c_0}}) - *g) \\ &\quad + \frac{\delta}{\delta T}(\frac{T}{m} * (1 + C_a e^{\frac{-C_b}{R} z_{c_0}}) - g)|_{T=T_0} * (T - T_0) \\ &\quad + \frac{\delta}{\delta z_c}(\frac{T_0}{m} * (1 + C_a e^{\frac{-C_b}{R} z_c}) - g)|_{z_c=z_{c_0}} * (z_c - z_{c_0}) \\ &= a_0 \\ &\quad + (\frac{1}{m} * (1 + C_a e^{\frac{-C_b}{R} z_{c_0}})(T - T_0) \\ &\quad + (\frac{T_0}{m}\frac{(-C_b)}{R} C_a e^{\frac{-C_b}{R} z_{c_0}})(z_c - z_{c_0})\end{aligned} \tag{A.4}$$

Rewriting Equation A.4 such that the increment in thrust is once more isolated results in Equation A.5.

$$\tilde{T} = \frac{m}{1 + C_a e^{\frac{C_b}{R} z_{c_0}}} * ((a - a_0) - (\frac{-C_b}{R}\frac{T_0}{m} C_a e^{\frac{C_b}{R} z_{c_0}}) * (z_c - z_{c_0})) \tag{A.5}$$

What can be seen in this INDI equation is the addition of two extra terms compared to the one in Equation A.2. These two terms will adjust the thrust increment when coming closer to the ceiling. To determine the change in position $(z_c - z_{c_0})$ the velocity will be used with a gain factor acting as a time constant, as time multiplied by velocity is distance. The velocity measurement is used instead of the position measurement for the following reason. When you send a command to the system there will be a delay on the response, because the motors need time to spin up or down. If the position measurements would be used at each time step the position increment would not represent the expected position increment for the set input, due to the delay. The time step gain is one parameter that can be tuned to reach satisfactory performance.

Lastly the RBF INDI controller can be derived in a similar fashion. This means that instead of the theoretical ceiling effect the RBF function model is substituted in. The end result is shown in Equation A.6. The derivation of the RBF function is discussed in Appendix B.

$$\tilde{T} = \frac{m}{RBF} * ((a - a_0) - \frac{1}{m}\sum_{i=2}^{7} -2a_i(z_c - c_i) w_i e^{-a_i(z_c - c_i)^2} * (z_c - z_{c_0})) \tag{A.6}$$

For which:

$$RBF = w_1 + \sum_{i=2}^{7} w_i e^{-a_i(z_c - c_i)^2}$$

The final control loop is shown in Figure A.2. This is based of the RBF derivation, but the other INDI controllers have a similar loop structure. It can be seen that there is a limiter within the loop, this is to control the thrust to not become negative or exceed the maximum thrust. The motors are modeled using a first order transfer function with break frequency at 16 $[\frac{rad}{s}]$, this is similar to the Crazyflie MAV [2], which will be used as the testing platform. The MAV is modeled using Equation A.3, which calculates the acceleration based on the thrust input and provides it as an output.

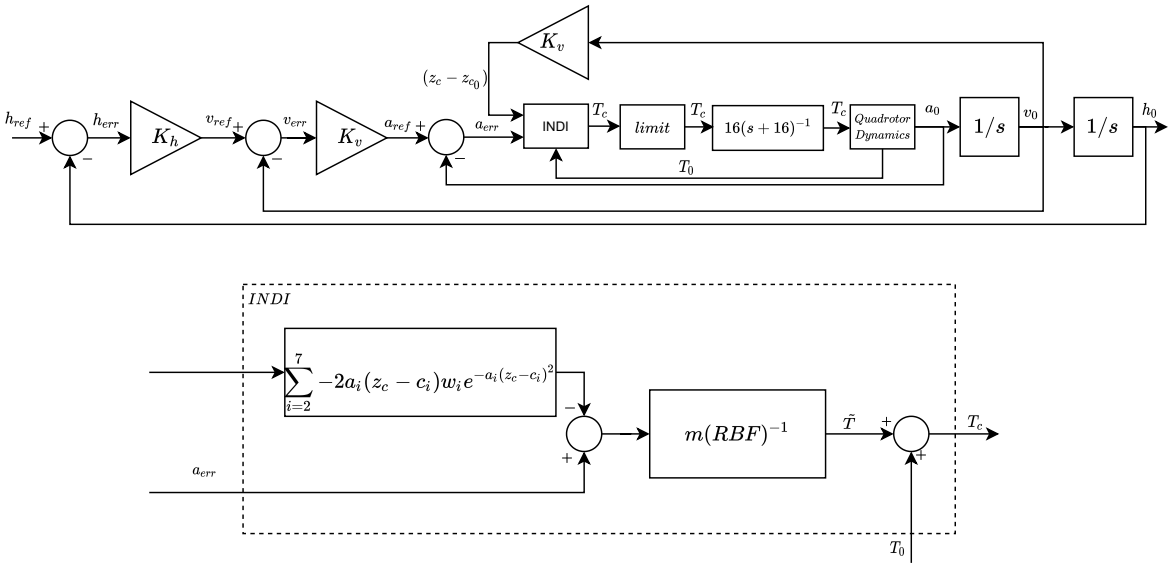


Figure A.2: One dimensional INDI implementation including ceiling effect RBF model.

A.2 Results

With the controller from section A.1 implemented, different results can be analyzed. The simulation step time is set to 0.01 $[s]$ as the Crazyflie outer loop also runs at this rate.

To simulate the ceiling effect within the MAV dynamics, the model of He et al. was used. Figure A.3 shows the result for that relationship using the dimensions of the propellers from the Crazyflie (diameter 45 $[mm]$).

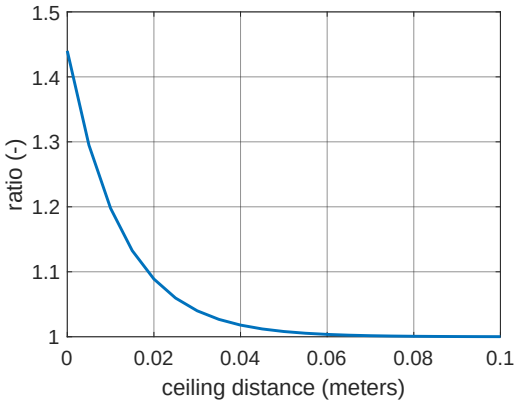


Figure A.3: Simulated Ceiling effect based on model from He et al. [1] with $R = 0.0225$, $C_a = 0.44$, $C_b = -1.8$.

First of all the standard INDI controller is tested in two situations. One where the ceiling effect is omitted from the MAV dynamics, and another where the ceiling effect is implemented. Figure A.4 shows the altitude response of the MAV without the ceiling effect. The flight simulated is take-off to 40 $[cm]$, then hover at 43 $[cm]$ and lastly hover at 48 $[cm]$, where the ceiling is placed at 50 $[cm]$. The standard INDI controller commands the MAV to the correct altitude without overshoot. Figure A.5 shows the result of the same controller but with the ceiling effect present, for which the percentage of extra thrust gained is presented in Figure A.3. As the effect is still minimal at the distances of 10 and 7 $[cm]$ away from the ceiling ($< 5\%$) the controller does not behave differently. When the MAV reaches a distance of 2 $[cm]$ the effect is providing 10% extra thrust which creates an overshoot and oscillation before stabilizing at the correct altitude. This proves that the original INDI controller response is influenced by the ceiling effect, which may lead to stability problems in real life, and thus the adaptation is proposed.

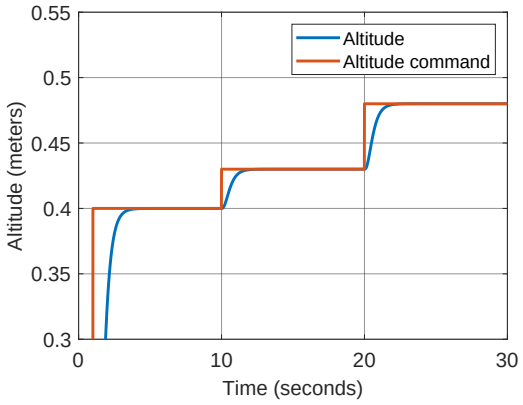


Figure A.4: Original INDI controller, simulation without ceiling effects.

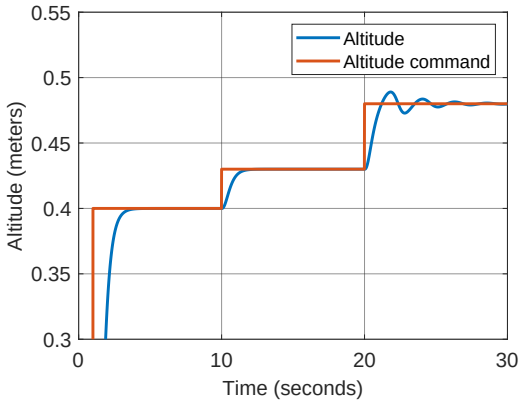


Figure A.5: Original INDI controller, simulation with ceiling effects.

With the possible chance of crashing using the original INDI control loop the newly developed controller is tested for the same flight scenario. The controller is used for distances of 10 $[cm]$ towards the ceiling or closer. This is due to the defined span of the RBF function discussed in Appendix B. First the INDI controller with the theoretical ceiling effect model into the dynamics is simulated (Equation A.5). Different values for the so called time constant K_v were tested to see its effect. The first result seen in Figure A.6 shows that the adapted INDI cannot fully compensate for the ceiling effect. For this simulation the time constant is set equal to the simulation time step.

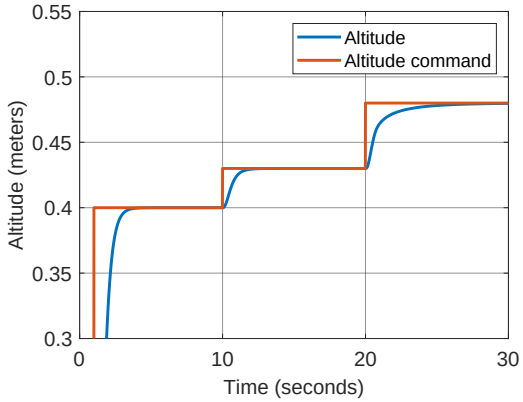


Figure A.6: New INDI with incorporated theoretical model, $K_v = 0.01$.

As the previous time step is deemed to be to small, a new value needs to be set. To find a suitable value for the time constant, the actuator dynamics step response will be investigated. The step response is shown in Figure A.7. A value of 90% of the step response is deemed to be enough to say that the actuator has responded to the given command, this happens after 0.3 $[s]$. The result for $K_v = 0.3$ is seen in Figure A.8. This time it can be seen that the overshoot has disappeared as well as the oscillation. This means that the adapted INDI is able to compensate for the ceiling effect, when using a gain attuned to the actuator dynamics. What can be noticed is that due to the compensation the controller needs more time to reach the final altitude, compared to the response in Figure A.4. This is due to the ceiling effect compensation terms that decrease the thrust command increment.

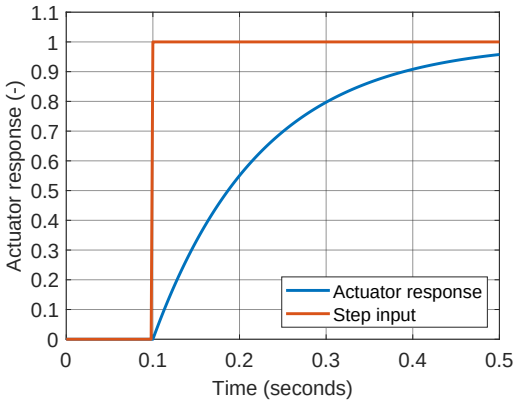


Figure A.7: Actuator step response.

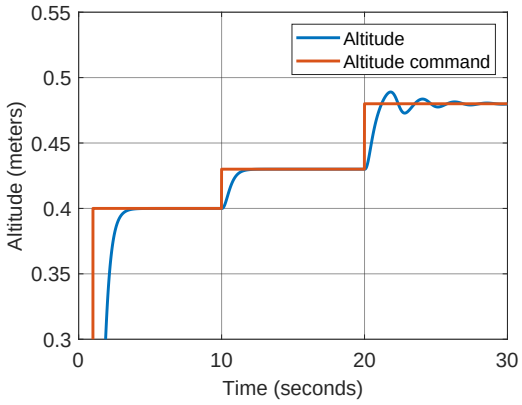


Figure A.8: New INDI controller with incorporated theoretical model, $K_v = 0.3$.

Lastly Figure A.9 shows the response of the controller when the time constant is increased even further, beyond the response time of the actuators. What can be seen is that MAV response becomes slower. As a slow response is not necessarily bad it may be unwanted in certain situation where time is limited. An example of this may be an inspection mission where a certain area needs to be inspected within a certain amount of time, which may not be obtainable due to the slow MAV response. Moreover a slow response can also mean that you are unable to respond quickly enough to an unforeseen event which may lead to a crash. It was was decided to set the $K_v = 0.3$ for the rest of the simulations, based of the response time of the actuators.

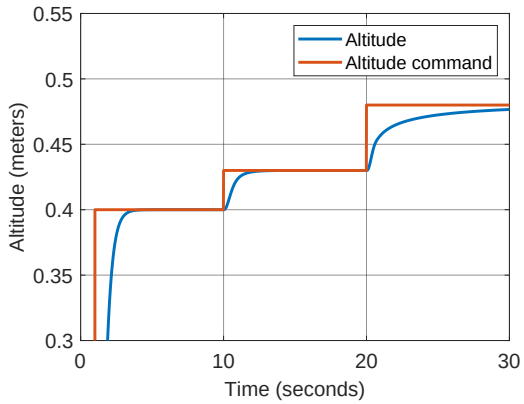


Figure A.9: New INDI controller with incorporated theoretical model, $K_v = 1$.

Previously the ceiling effect within the INDI controller was similar to the one in the MAV dynamics. This will probably not be the case in a real life situation, as modeling errors exist. Thus to improve the robustness of the controller the ceiling effect model was substituted by a RBF model approximation (Equation A.6). To first test the RBF model it was fitted onto the theoretical model. This was done to check if the controller provides a similar response, verifying the implementation. The altitude response seen in Figure A.10 is comparable to the one in Figure A.8. The ceiling effect estimate is shown in Figure A.11, which shows that the RBF fit onto theory almost matches the model perfectly. The value seen in Figure A.11 is only used within the controller when the altitude is within 10 $[cm]$ from the ceiling, the value of 1.4 is thus not used and is caused due to the constant within the RBF model.

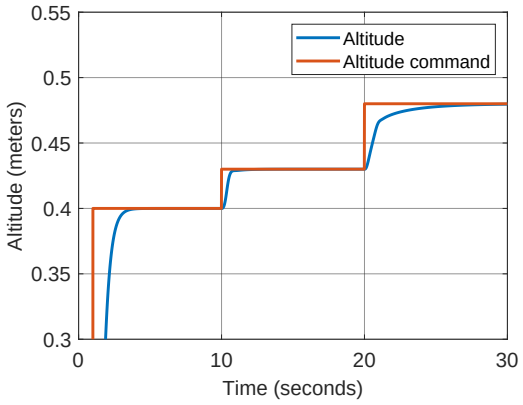


Figure A.10: New INDI controller with incorporated RBF model, theoretical fit, $K_v = 0.3$.

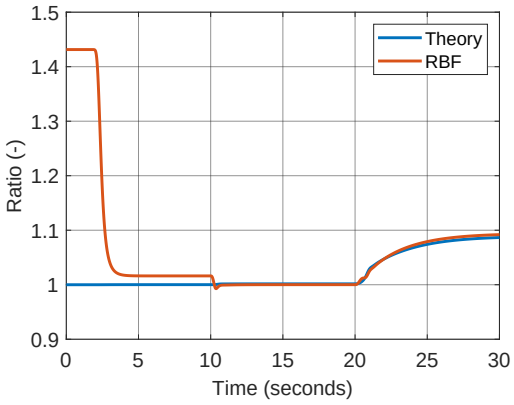


Figure A.11: Ceiling effect theory fit versus theory during flight.

There exist two possible situations for a mismatch between the RBF approximation and the real ceiling effect, being under- or overestimation of the effect. To test the response of the INDI controller in these two scenarios, the RBF was purposely altered to under- or overestimate the ceiling effect implemented within the MAV dynamics. First of all Figure A.12 shows the altitude response of the INDI controller using an underestimated model, shown in Figure A.13. As seen the response became quicker due to less compensation. Even though one might argue that this quicker response is better than the previous one, it also brings it closer to a situation where there is not enough compensation. The overshoot can cause instability due to even getting closer to the ceiling, increasing the ceiling effect, which in the end may lead to a collision with the ceiling.

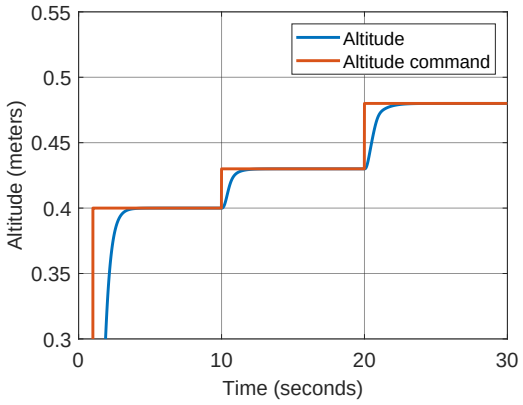


Figure A.12: New INDI controller with incorporated RBF model, underestimation fit, $K_v = 0.3$.

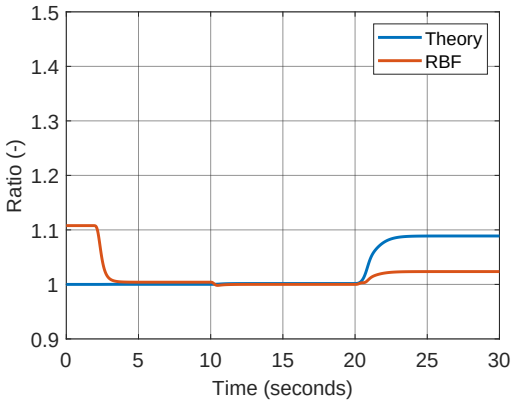


Figure A.13: Ceiling effect underestimation versus theory during flight.

The second situation is the overestimation, shown in Figure A.14 and Figure A.15. This time it can be seen that the response became slower, due to a higher value of the ceiling effect.

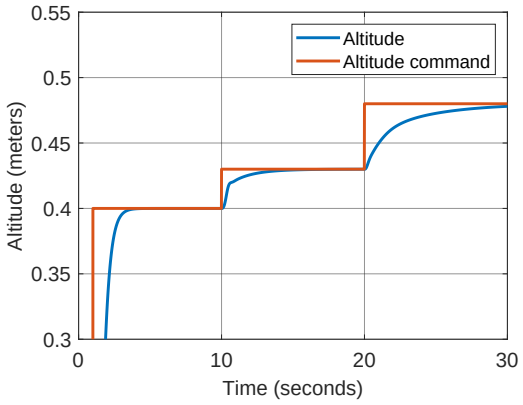


Figure A.14: New INDI controller with incorporated RBF model, overestimation fit, $K_v = 0.3$.

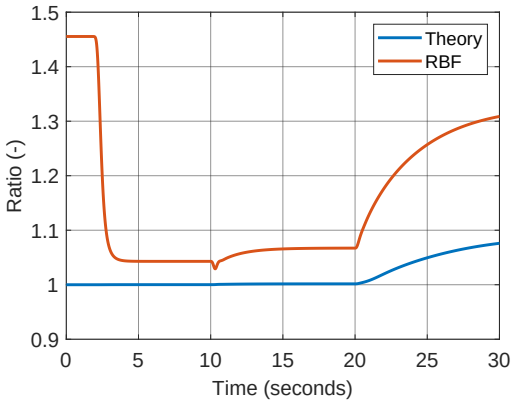


Figure A.15: Ceiling effect overestimation versus theory during flight.

Both situations have shown that the new INDI controller with RBF is able to deal with modeling errors, up to a certain extend. The simulated situations are on the extreme side, the RBF fit is expected to be closer to the theoretical curve than both situations represent. This means that the situations that lay between the ones presented have an acceptable response.

B Radial Basis Function

To create a more adaptable controller it was decided to create a fitting function to model the ceiling effect, that uses onboard measurements from the MAV. A Radial Basis Function or RBF for short, is a bell shaped curve which is described by Equation B.1. The parameters w, a and c can be altered to change the shape and position of the bell curve. w affects the height, a the width and c the position of its peak.

$$we^{-a(z_c-c)^2} \tag{B.1}$$

By taking the sum of RBF functions all different kind of function shapes can be modeled. This makes it an useful tool to make the INDI controller more robust and adaptable to different systems, as a different sized MAV will experience the ceiling effect at other distances, and has other values at those distances. The RBF function could also be adapted to include other aerodynamic effects (e.g. ground effect) in future work.

B.1 Model Setup

For this work the focus was on compensation of the ceiling effect. The ceiling effect starts to become noticeable around a distance of 10 $[cm]$ from the ceiling using the Crazyflie platform (the theoretical curve in Figure A.3 shows the effect starting to rise above 1% from this distance onward). It was thus decided to have the RBF function describe a distance from $0-10$ $[cm]$ from the ceiling. Another factor during the design is the limited computational power of the Crazyflie system, as a larger span of the RBF function may need more RBF elements, increasing the number of parameters and thus the computational load. The RBF function design is based of the form it needs to describe set by the theoretical ceiling effect formula mentioned in Equation 3.4. This function rises exponentially when getting closer to the ceiling. The final RBF function was chosen to consist of a summation of a constant, and six RBF terms as described by Equation B.2.

$$RBF = w_1 + \sum_{i=2}^{7} w_i e^{-a_i(z_c-c_i)^2} \tag{B.2}$$

To determine the placement and width of each of the six RBf terms the function was fitted onto the theoretical ceiling effect. The constant was used to adjust the height of the final curve, as such that the RBF functions only needs to model the curvature. Using trial and error the following values where found.

$$\boldsymbol{a}_i = \begin{bmatrix} 15000.0 & 10000.0 & 5000.0 & 2000.0 & 1250.0 & 1000.0 \end{bmatrix}$$

$$\boldsymbol{c}_i = \begin{bmatrix} 0.0 & 0.05 & 0.015 & 0.3 & 0.6 & 1.0 \end{bmatrix}$$

The six different RBF functions with their corresponding shape are shown in Figure B.1. As can be seen the further away from the ceiling the wider the RBF terms become. A wider RBF term can describe a flat line more easy, vice versa a steeper one is better for fitting the rising part of the curve. The center placement is biased towards the exponentially rising part of the curve such that the steep slope can be modeled better. Using Ordinary Least Squares (OLS) explained in subsection B.2.1 the curve was fitted onto the theoretical model, the result is seen in Figure B.2. 21 data points were used to fit the curve, each spaced half a centimeter apart.

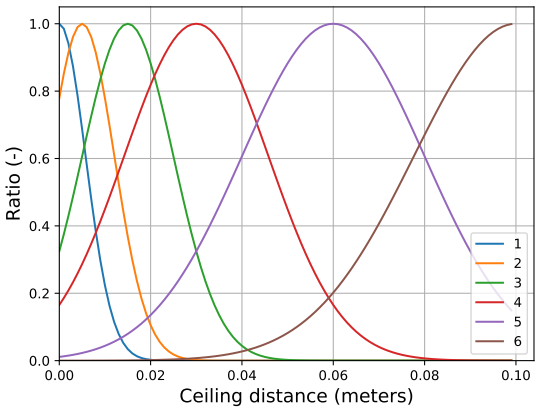


Figure B.1: The six different RBF shaped curves.

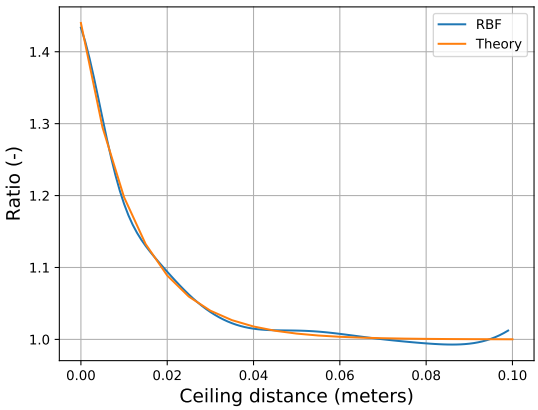


Figure B.2: RBF curve fitted onto the theoretical curve.

B.2 Fitting Method

To be able to determine the last unknown parameters w_i in the RBF function (Equation B.2) a parameter estimation method needs to be chosen. The methods investigated are discussed in this section.

B.2.1 Ordinary Least Squares

The first method which was investigated is OLS [3], which is a relatively simple method to calculate the model parameters for function fitting. Equation B.3 states the estimate is obtained by multiplying the calculated weights $\hat{\boldsymbol{\theta}}$ (model parameters) with the regression matrix $\boldsymbol{A}(z)$.

$$y_{est} = \hat{\boldsymbol{\theta}} \cdot \boldsymbol{A}(z_c) \tag{B.3}$$

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} w_1 & \dots & w_7 \end{bmatrix} \qquad \boldsymbol{A}(z) = \begin{bmatrix} 1 & e^{-a_2(z_c-c_2)^2} & \dots & e^{-a_7(z_c-c_7)^2} \end{bmatrix}$$

To calculate the model parameters one multiplies the inverse of the regression matrix with the data points y that the curve needs to fit, as described in Equation B.4. Instead of taking the normal inverse the pseudo inverse is used. The regression matrix for the RBF chosen has the shape of n by seven, where n is the number of data points used for fitting. When using a number of data points not equal to seven the regression matrix becomes rectangular and thus the inverse must be approximated. With the $m \times n$ matrix where $n = 7$ and more than seven data points ($m > 7$) will be used, the right inverse of the matrix $A(z)$ can be calculated and may be used to approximate the model parameters.

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{A}^T(z_c) \cdot \boldsymbol{A}(z_c))^{-1} \boldsymbol{A}^T(z_c) y \tag{B.4}$$

To create the fit a number of data points needs to be selected along which the function will be fitted. As discussed in section B.1 the distance up until which the fit is created is 10 $[cm]$, where in the last few centimeters the curve changes quiet drastically. To create a good fit each half a centimeter a data point bin was created. This number of data points was also used to create Figure B.2 and thus is deemed valid. This means the regression matrix becomes 21×7 and thus the right inverse exists (matrix has rank 7).

To collect the data, onboard measurements are used which can be noisy. To account for this noise a filter needs to be used to make the fit more robust, as the OLS method does not have any build in mechanism to deal with noise. A moving average approach is investigated. This method is based of weighing new (y_t) and old (y_{t-1}) data points as described by Equation B.5. The total of both weights (v_{t-1} and v_t) will always add up to one.

$$y_t = v_{t-1} y_{t-1} + v_t y_t \tag{B.5}$$

For the measured thrust ratio different weights are tested and evaluated. Four different weights ratios are shown in Figure B.3 up to Figure B.6. Reducing the weight of the new incoming data decreases the noise in the signal. The peaks (between 10-12.5 and 15-17.5 seconds) are reduced most when applying a filter of 0.1 or 0.01 on the incoming data. Especially in Figure B.6 a smooth line is seen where most noise has been removed, but the downside of such an aggressive filter is the delay introduced in the measurement signal.

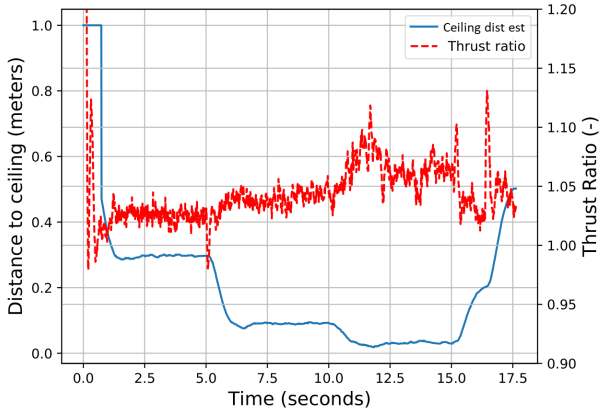


Figure B.3: Measured thrust ratio filtered $w_{t-1} = 0.5 \wedge w_t = 0.5$, with estimated ceiling distance.

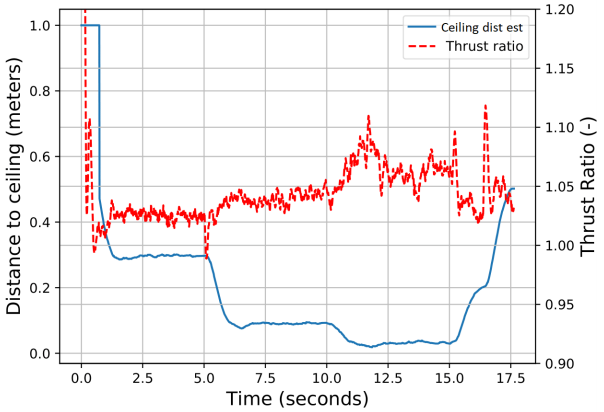


Figure B.4: Measured thrust ratio filtered $w_{t-1} = 0.75 \wedge w_t = 0.25$, with estimated ceiling distance.

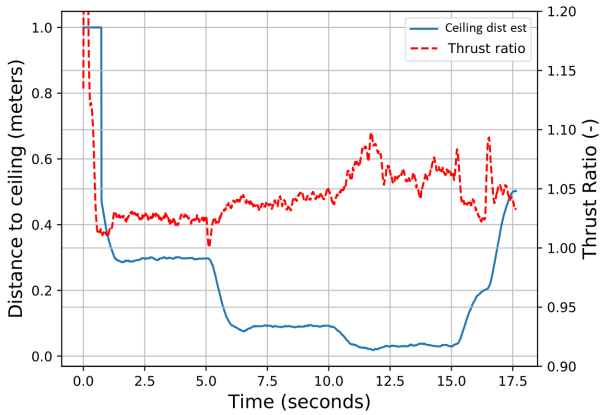


Figure B.5: Measured thrust ratio filtered $w_{t-1} = 0.9 \wedge w_t = 0.1$, with estimated ceiling distance.

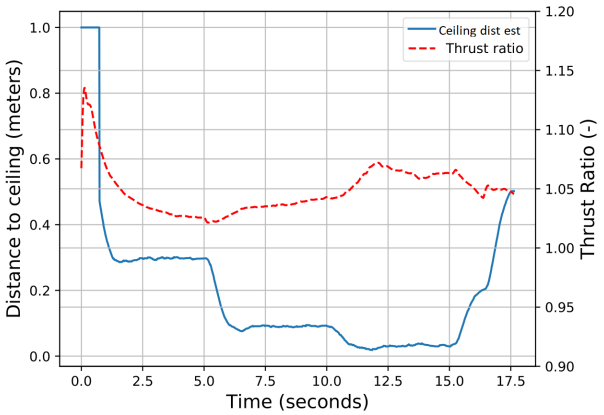


Figure B.6: Measured thrust ratio filtered $w_{t-1} = 0.99 \wedge w_t = 0.01$, with estimated ceiling distance.

To evaluate the performance of the two more aggressive filters (0.1 and 0.01), which are needed to reduce the measurement noise to an acceptable level, a test fit was created using the filtered data. For the 0.1 filter the results can be seen in Figure B.7 and Figure B.8. Where Figure B.7 shows the RBF fitting, with all data points created during flight. Similar colored data points close together mean that they belong to the same data point bin (as there are a total of 21 data points used for fitting there are 21 data point bins). Figure B.8 shows the weights of the fitting process. The RBF weights are initiated at a value of zero, the constant is set to one. This means that the RBF function will provide a value of one when no measurements are taken. The new INDI controller is used when flying within a range of 10 $[cm]$ from the ceiling, for this flight this distance was crossed around a time of 6 $[s]$ (point where the weights change first). What immediately catches the eye, are the two coexisting lines of data points between $4 - 8$ $[cm]$ in Figure B.7, which become even more clear when the most aggressive filter is used, shown in Figure B.9. The final curve is only fitted onto the latest data, which is one of the two lines. The reason for the creation of those two lines is the following. As stated before, a filter introduces delay in a signal, meaning it takes time for the signal to reach its desired value when providing an input. One of the two lines is created during the ascend and the other during the descend. Figure B.5 shows a fast descend from 2 $[cm]$ to 20 $[cm]$ distance. This means that all data points in between $2 - 10$ $[cm]$ will be updated but

only with a few new values each, as the MAV keeps descending. As the ceiling ratio is high at 2 $[cm]$ and the filter adds an delay, the new data values at the lower data points will only differ slightly from the one at 2 $[cm]$, as there was a quick descend. This is an unwanted effect, as the fit does not represent reality, but is misfitted due to the aggressive descend and filter combination. This effect is also seen in Figure B.8 and Figure B.10 where the weights change drastically just before the end of the flight (the descend maneuver).

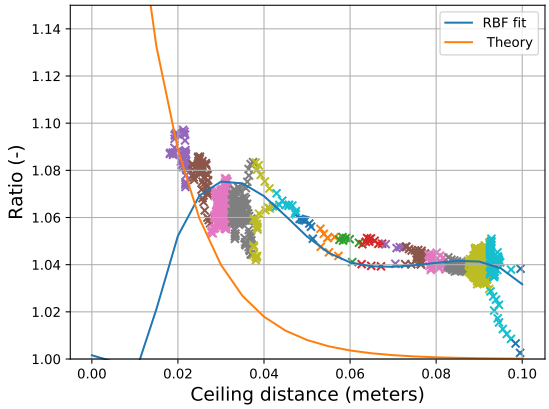


Figure B.7: OLS fit with data filter $w_{t-1} = 0.9 \wedge w_t = 0.1$, compared to theory and with all data points.

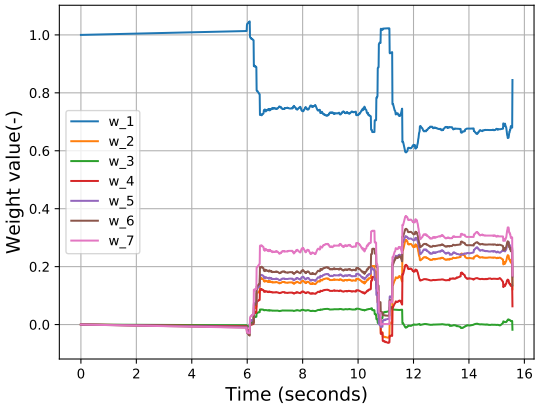


Figure B.8: OLS fitting weights with data filter $w_{t-1} = 0.9 \wedge w_t = 0.1$.

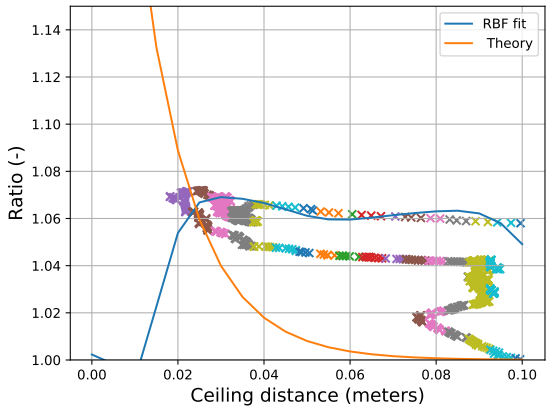


Figure B.9: OLS fit with data filter $w_{t-1} = 0.99 \wedge w_t = 0.01$, compared to theory and with all data points.

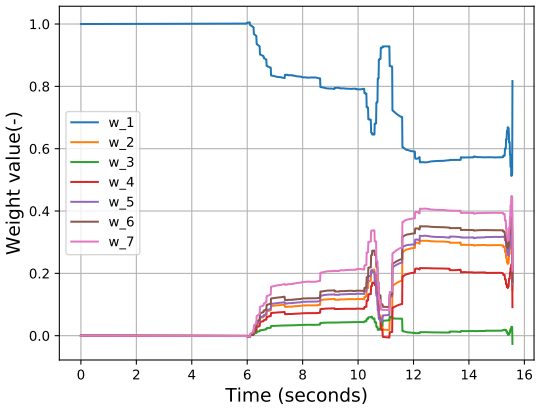


Figure B.10: OLS fitting weights with data filter $w_{t-1} = 0.99 \wedge w_t = 0.01$.

It was thus concluded that the OLS method in combination with the filter is not a valid fitting approach and another fitting method was investigated, which will consider a filter within the method itself.

B.2.2 Recursive Least Squares

A method where a noise filter is build in is Recursive Least Squares (RLS) with a forgetting factor [3]. Here the forgetting factor acts as the filter, telling the algorithm how quick it should adapt to new incoming data. Moreover the method is computationally less expensive as the matrices used are smaller and no expensive inverse is needed. Matrix multiplications cause the inverse to become a scalar. This method calculates the new fit based on a stream of incoming data, and thus in retrospective with OLS it does not need a full data set across the fitting space at each time step.

The model parameter estimation is done with the following formulas. First of all the Kalman gain is calculated according to Equation B.6, in which λ is the forgetting factor, which may have value of $0 < \lambda \leq 1$. New data has the most influence on the fit as λ approaches one. Secondly, using Equation B.7 the model parameters can be calculated using the latest measurement y_{k+1}, the old model parameters $\hat{\boldsymbol{\theta}}_k$, regression matrix $\boldsymbol{A}_{k+1}$, and

the previously calculated Kalman gain $\boldsymbol{K}_{k+1}$. Lastly, using Equation B.8 the covariance matrix $\boldsymbol{P}$ is updated. The covariance matrix will be a seven by seven diagonal matrix. Each diagonal entry within this matrix tells how much each weight is correlated with itself. The smaller this value, the smaller the change is expected to be.

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_k \cdot \boldsymbol{A}_{k+1}^T (\boldsymbol{A}_{k+1} \cdot \boldsymbol{P}_k \cdot \boldsymbol{A}_{k+1}^T + \lambda)^{-1} \tag{B.6}$$

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \boldsymbol{K}_{k+1}(y_{k+1} - \boldsymbol{A}_{k+1}\hat{\boldsymbol{\theta}}_k) \tag{B.7}$$

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{K}_{k+1} \cdot \boldsymbol{A}_{k+1} \cdot \boldsymbol{P}_k \tag{B.8}$$

$$\boldsymbol{A}(z_{c_{k+1}}) = \begin{bmatrix} 1 & p_1(z_{c_{k+1}}) & \dots & p_M(z_{c_{k+1}}) \end{bmatrix} \qquad p_1(z_{c_{k+1}}) = e^{-a_1(z_{c_{k+1}} - c_1)^2}$$

To investigate the influence of the forgetting factor the covariance matrix will be initiated as a diagonal matrix, with entries equal to one. The unfiltered data shown in subsection B.2.1 will be used. Figures B.11 to B.14 show the fitting process for forgetting factor values of 0.01 until 1. The legend indicates how far into the simulated flight the fit was created. Meaning 0 is at the start and 600 is around the end of the flight. From the results it can be seen that a higher forgetting factor leads to a fitting process in which the curve changes more evenly, especially within the area were no data is collected (between 0 and 4 centimeters). The final fit is shown to be similar between all learning parameters.

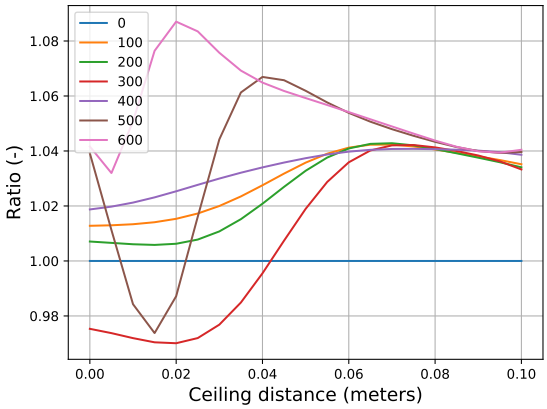


Figure B.11: RLS fitting, forgetting factor 0.01, fitting process.

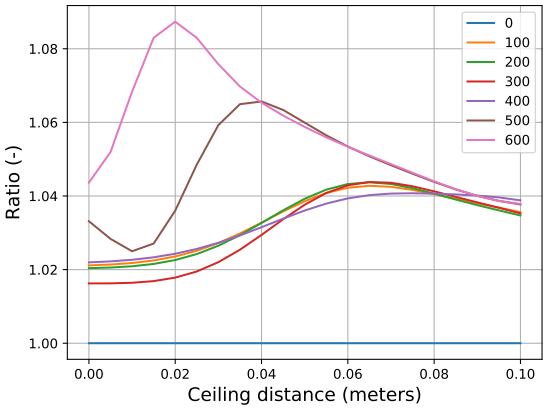


Figure B.12: RLS fitting, forgetting factor 0.1, fitting process.

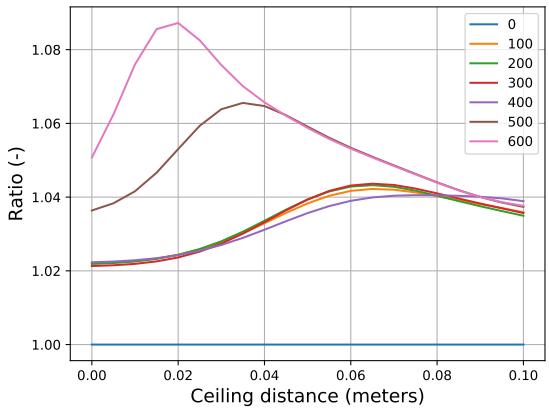


Figure B.13: RLS fitting, forgetting factor 0.5, fitting process.

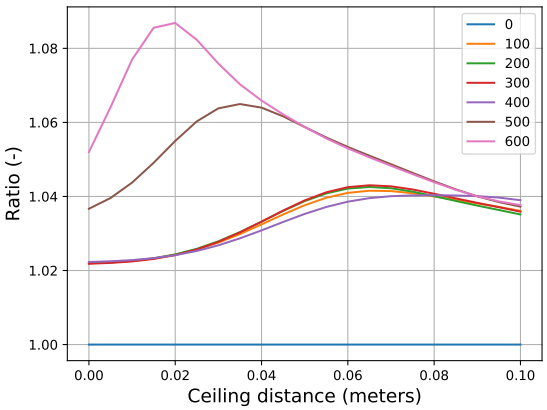


Figure B.14: RLS fitting, forgetting factor 1, fitting process.

Setting the forgetting factor higher than 0.01 does not change the final fit drastically, so to have equal weight between new and old data, the forgetting will be set at 0.5. To investigate how much the covariance matrix initiation affects the fitting process, different starting values are tested. The results are shown in Figure B.15 up to Figure B.17. From these results it can be seen that a lower value for the covariance matrix initiation leads to a fitting process where the curve changes more evenly. Once more the initiation does not seem to greatly impact the final fit.

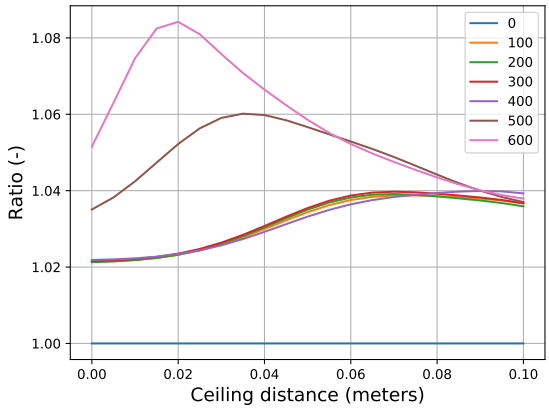


Figure B.15: RLS fitting, forgetting factor 1 and covariance initiation at 0.1, fitting process.

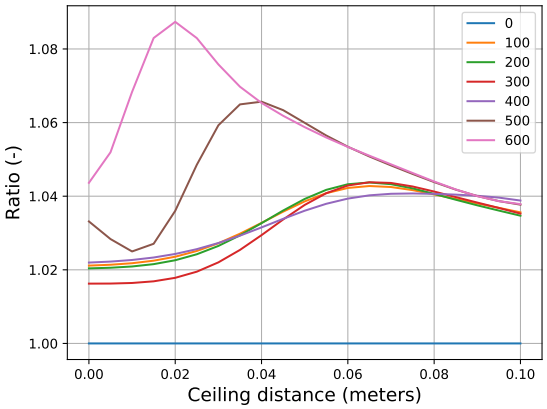


Figure B.16: RLS fitting, forgetting factor 0.5 and covariance initiation at 5, fitting process.

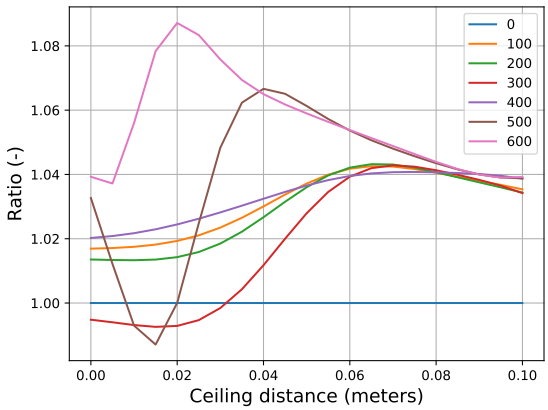


Figure B.17: RLS fitting, forgetting factor 0.5 and covariance initiation at 25, fitting process.

The final values are selected based on the previously shown results, initiation of 1 for the covariance matrix and value of 0.5 for the forgetting factor. Using these values Figure B.18 shows the final fit created by the RLS. Once more all data points collected are shown. The final fit is going through the middle of all those points, meaning this method is less sensitive to the data spread (noise) than OLS. Figure B.19 shows the progress of the weights during the fitting process, and Figure B.20 shows the covariance matrix values of the different weights. What can be seen is that the covariance value of a specific weight only starts to change when the measurements starts to get close to center placement of that RBF term. This shows that new incoming data mostly affects its surrounding area.

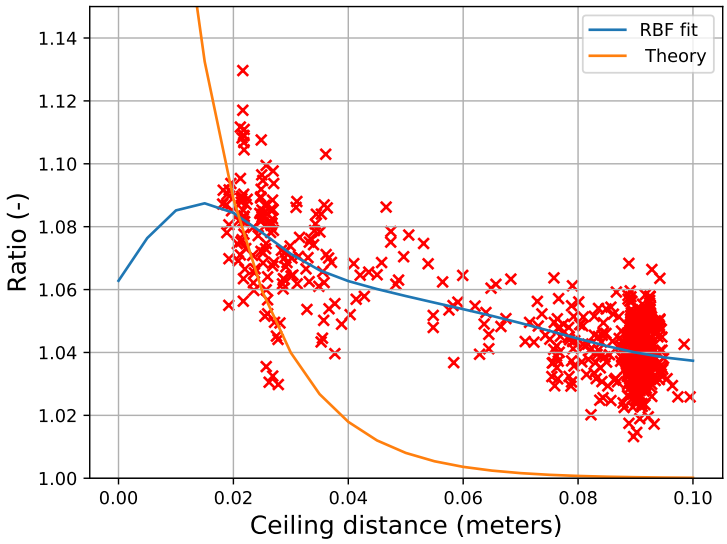


Figure B.18: RLS fitting, forgetting factor 0.5 and covariance initiation at 1, fitting result.

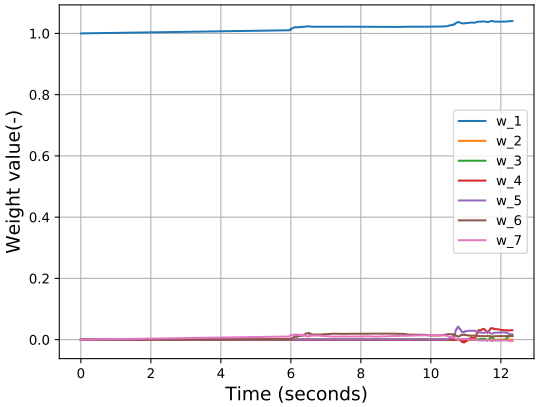


Figure B.19: RLS fitting, forgetting factor 0.5 and covariance initiation at 1, weights.

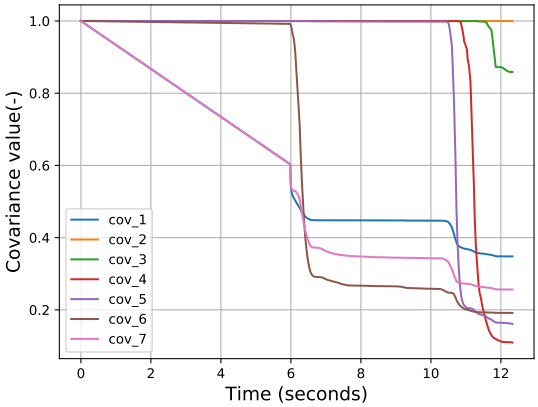


Figure B.20: RLS fitting, forgetting factor 0.5 and covariance initiation at 1, covariance values.

It is concluded that RLS is able to cope with the measurement noise and can estimate the model parameters such that a valid fit is created. Thus the RLS method will be the algorithm used to process the measurements and generate a ceiling effect curve during flight with the new INDI controller.

C Crazyflie Implementation and Results

This section will discuss extra information about the implementation of the new controller and the results that followed, besides everything shown in the scientific paper.

C.1 Crazyflie Protection

For the initial testing a custom bumper was 3D printed to protect the propellers when flying close to the ceiling. This was done as a precaution due to not knowing how the MAV would respond during the first few flights, as there might still have been bugs within the code. During the first flights the MAV became unstable, ending either in a crash, or up side down after a flip. First the new controller got the blame for causing the instability, but after the code was checked and no bugs were found, there was looked further. In the end it became clear the the extra weight of the bumper, besides the Crazyflie with the two expansion decks, made it too heavy. Due to the weight the MAV was flying around the limits of the actuators' capability during hover. The motor mixing does not have any prioritization included for roll and pitch over the thrust. This meant that all the commands provided by the INDI controller could not be fulfilled and the MAV started to move unpredictable. After removing the bumper the unstable behavior also disappeared, which made it clear that the bumper was the cause.

C.2 INDI Implementation

The standard INDI controller was already implemented on the Crazyflie system by E. Volodscoi [2]. The code was checked for mistakes, of which several were found. Most of the mistakes were related to sign convention. Moreover within the implementation the hover thrust was used within the outer loop to calculate the control effectiveness. This has been changed to include the actual thrust using the relationship found in the work from Shi et al. [4]. Lastly the control effectiveness values within the inner attitude loop were recalculated.

C.3 Results

During the first test flights with a finished controller the estimated ceiling height was changing during the flight. This height is used to generate the set points for the MAV and thus need to be correct. To investigate what caused the wrong estimation, data was logged when moving the MAV by hand towards the ceiling. The altitude and ceiling distance can be seen in Figure C.1 and Figure C.2 respectively, with the estimated ceiling height shown in Figure C.3. The MAV was moved with increments of approximately 2 $[cm]$ towards the ceiling. The ceiling height estimate is relatively stable (considering an unsteady hand moving the MAV) up to the 25 $[s]$ mark. Afterwards the ceiling height estimate starts dropping, which is not possible as the ceiling height was not changing, thus it must be caused by wrong onboard measurements and or calculations. The ceiling distance estimate in Figure C.2 is seen to change 4 $[cm]$ at the 30 $[s]$ mark, while the altitude measurement in Figure C.1 only changes approximately 2 $[cm]$, this means there is a mismatch between the two ToF sensors measuring those distances. Within the data sheet of the ToF sensor[1], the minimal measurement distance is stated to be 4 $[cm]$. This implies that the top ToF sensor provided wrong measurements, this is true as during the handheld flight the MAV was never touching the ceiling thus a distance of 0 $[cm]$ is a wrong measurement. The combination of wrong measurements by the top ToF and mismatch in displacement between the two ToF sensors caused the alternating ceiling height.

[1] `https://www.st.com/resource/en/datasheet/vl53l1x.pdf`

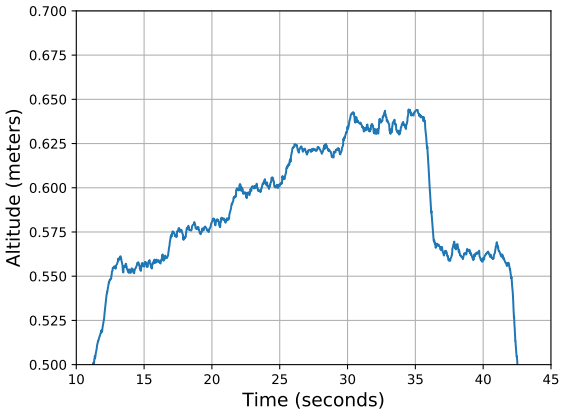


Figure C.1: Altitude for the flight by hand.

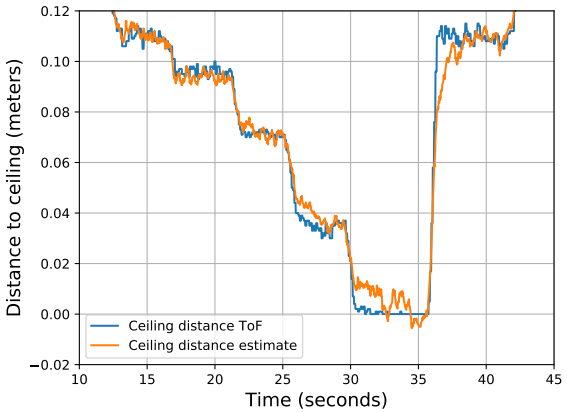


Figure C.2: Ceiling distance for the flight by hand.

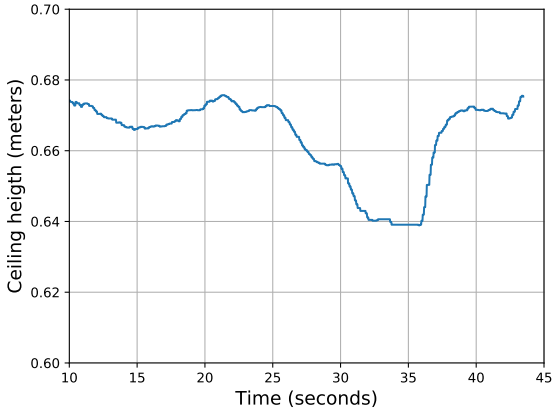


Figure C.3: Ceiling height estimate for the flight by hand.

To solve the issue, the ceiling distance estimate will only be corrected with a ToF measurement when the ceiling distance estimate is larger than 5 $[cm]$, thus staying away from the minimal distance measurement range. This implies that at distances closer than 5 $[cm]$ only the downward measurement is used to adapt the ceiling distance estimate. Figure C.4 until Figure C.6 show results from an actual test flight where the proposed solution was implemented. During the flight the ceiling estimate stayed relatively constant (taking into account sensor noise) as seen in Figure C.6. The ceiling height stays completely constant as soon as the ceiling distance becomes smaller than 5 $[cm]$. Due to removing the correction step the ceiling distance estimate is only based of measurements from the downward facing ToF, this causes the ceiling height estimate to stay constant. When observing the MAV during the test flight the ceiling distance estimate (with implemented solution) was deemed correct. During the final test flights (for generating the paper results) the ceiling height estimate was calculated during hover far from the ceiling (> 20 $[cm]$). This ceiling height was than taken and used during the rest of the flight, making certain that the ceiling height stayed constant.

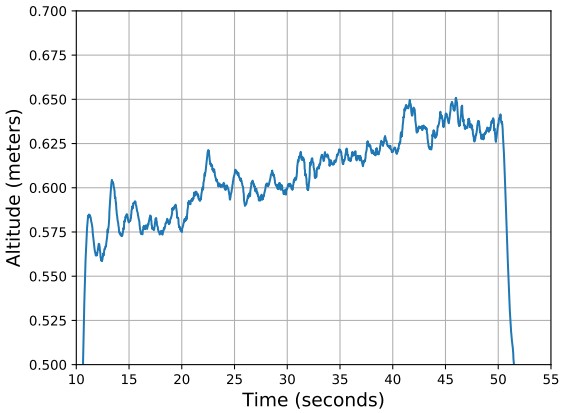


Figure C.4: Altitude of test flight with solution for ceiling distance/height implemented.

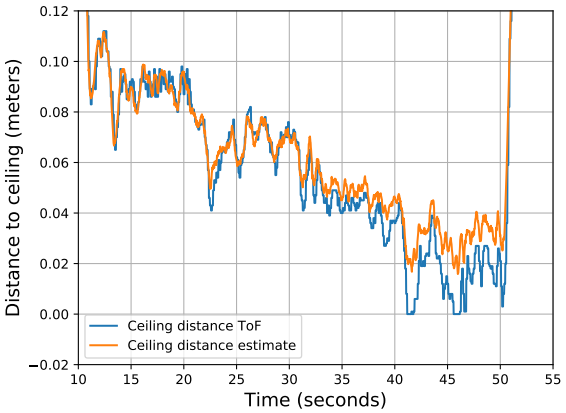


Figure C.5: Ceiling distance for the flight with solution for ceiling distance/height implemented.

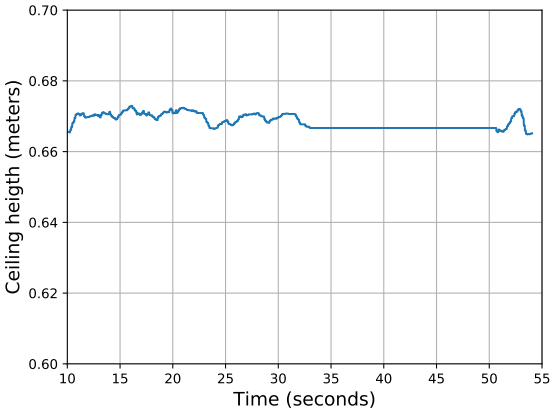


Figure C.6: Ceiling height estimate for the flight with solution for ceiling distance/height implemented.

During the test flights it was assumed that the flight was purely vertical. The x and y position for one of the test flights is shown in Figure C.7 and Figure C.8 (other flights showed similar results). As seen both x and y position are not completely stable and show some movement. This movements did however not show to have a large influence on the controller behavior during the execution of the test flights. For this reason the assumption of pure vertical flight was deemed valid.

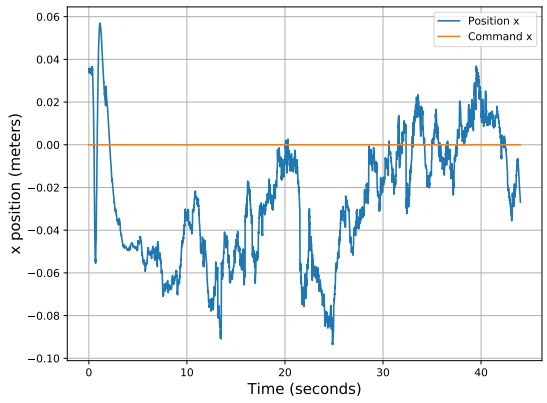


Figure C.7: Test flight x position.

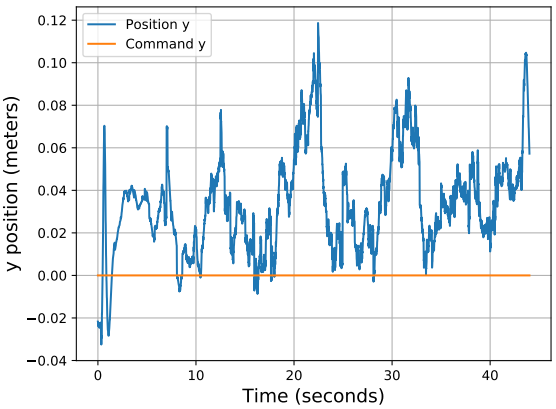


Figure C.8: Test flight y position.

For the new controller a RBF fit needed to be created. The final fit with all the collected data points is shown in Figure C.9. As can be seen the RLS fit shows to be a good fit around the middle of all data points, and seems to have not been altered by the high spikes in the data. The measured data is shown in Figure C.10, with the corresponding ceiling distance shown in Figure C.11. The flight conducted started at 10 cm distance with increments of 2 [cm] with a final distance of 2 [cm].

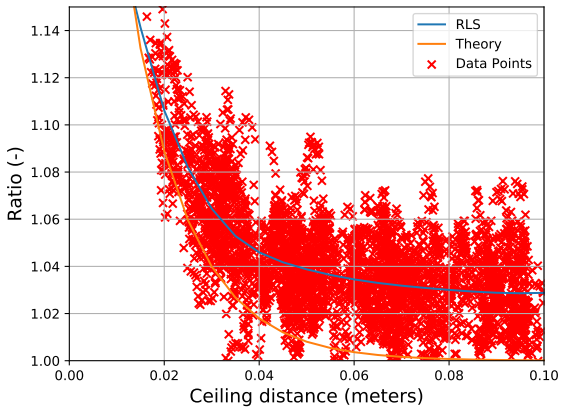


Figure C.9: RBF fit with all data points.

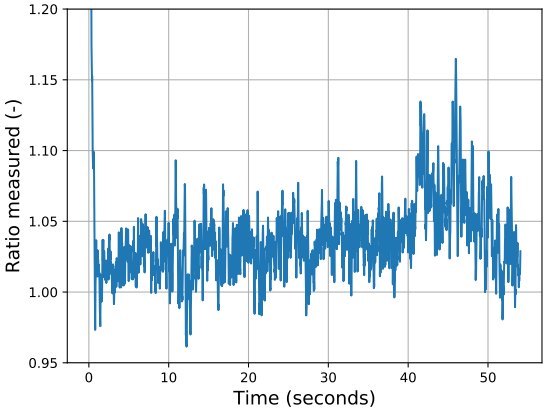


Figure C.10: T ratio measured for the RBF fit flight.

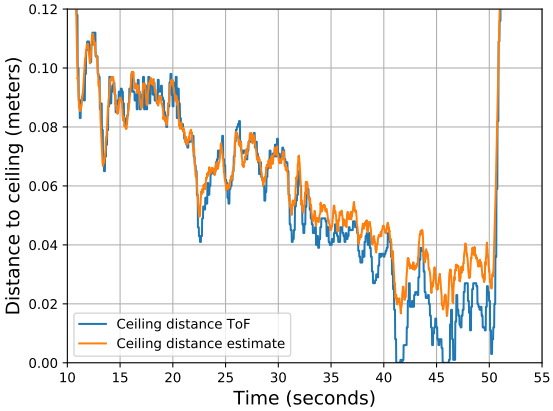


Figure C.11: Ceiling distance for the fitting flight.

Lastly within the results shown in the paper there exists a steady state error. This error is caused by a bias in the accelerometer measurement. A steady 40 $[cm]$ hover flight was conducted with the altitude and acceleration measurements shown in Figure C.12 and Figure C.13. During the periods from $6-9$, $12-16$ seconds the measured and wanted acceleration both shown to be negative. This means an acceleration upwards as the positive z axis is pointing downwards in the defined coordinate system. This is however not seen in the altitude response as the MAV keeps hovering around the same altitude, not constantly moving up. This is caused by the bias which makes the MAV think that its accelerating. The INDI is an incremental controller with similar behavior as an integral gain, meaning convergence to the set command such that no steady state error exists. However due to the PD controller generating a reference acceleration from the position error and subtracting the measured acceleration (which has a bias) will provide a virtual input close to zero, meaning the INDI controller will not correct the steady state error.

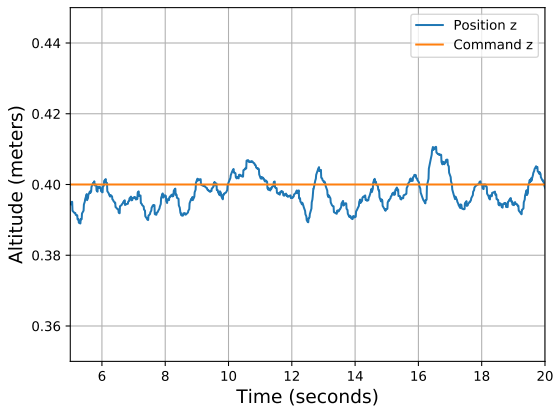


Figure C.12: Bias flight altitude measurements.

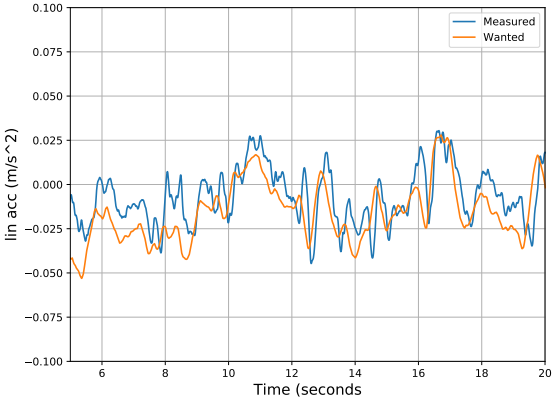


Figure C.13: Bias flight z axis accelerometer measurements.

Part III

Literature Study

1 Introduction

Within every sector technology advances with time. This is no different for the industry of Unmanned Aerial Vehicles (UAVs), or more generally called drones. In the year 2014 there was already an expenditure of 700 million dollar on commercial drones. The applications for those drones range from photography, wild life counting up to surveillance and inspections [5]. Miniaturization and automation of drones played a mayor role in recent years making the drones smaller and smaller, obtaining the name Micro Air Vehicles (MAVs), which expands the application field. Flying indoors is such an application. An example of this is by Sampedro et al. who developed a drone to carry out search and rescue operations [6]. Another example is the Airburr drone by Briod et al. [7], also a search and rescue drone which is designed for robustness and is capable of dealing with collisions. Still the areas explored by those drones are relatively large compared to size of the drone. Smaller areas are also being explored already, such as sewage pipes, explored with a caged drone developed by the company Flyability[1]. Such a caged design makes the drone more robust and keeps it protected from flipping over and being unable to continue its flight. However when a wall needs to be inspected it might not be possible to get close enough with such a caged drone, as such needing a cageless drone. Moreover the Flyability is mostly manually controlled, but communication with a human operator might not always be possible introducing the need of an autonomous controller. This shows the need for innovation.

To make this autonomous flight within small enclosed environments (meaning that the dimensions of the MAV are within ten times the maximum dimensions of the room, e.g. a 10 centimeter MAV within a 50 centimeter sewage pipe) without a cage possible a thesis proposal is made. The aim of the proposed thesis work is the following. The development of an autonomous MAV controller which makes it possible to fly stable and safe within an enclosed area, with the ability of getting close to the environment borders. The aim of this report will be to perform a literature study within the field of indoor autonomous flight, with a focus on disturbances caused by aerodynamic effects and the handling of mild collisions. This will make is possible to setup a research plan for the future work of the thesis.

The structure of the report will be the following. First of all the research plan for the literature study is developed in chapter 2. This will provide a layout for the areas of interest that will aid in obtaining a solution for the thesis objective. Then chapter 3 will elaborate on the literature found for those areas. It will explain what is already done and how this might aid the thesis work. After that chapter 4 will provide the results from two different types of experiments performed. One concerning the aerodynamic flow created by a drone and the other one going into performance of a PID controller. Next up chapter 5 will provide the research plan for the future work for completion of the thesis. Lastly chapter 6 will provide a conclusion.

[1] `https://www.flyability.com/`

2 Research Plan Literature Study

As mentioned within the introduction this thesis aims to make autonomous flight of an MAV within a small enclosed area possible, e.g. a 10 centimeter drone within a 50 centimeter sewage pipe. To be specific the research objective is formulated as follows:

> "To achieve autonomous stable flight of an MAV quadcopter within a small enclosed area by means of developing a controller that is able to cope with the aerodynamic effects, self-induced flow disturbances and mild collisions."

This research objective can be broken down into several topics for the literature study. First of all the 'small enclosed area' will bring several challenges as mentioned. Aerodynamic effects will play a role when flying close to the boundaries (e.g. wall, ceiling and ground). To find out more about these effects, part of the literature study will be focused on finding what effects exist, how they interact with an MAV and if the effects can be modeled. The area wherein the MAV flies will be mostly closed off, the induced flow will start to circulate, also introducing other possible disturbances. To acquire knowledge on this, Computational Fluid Dynamics (CFD) software will be used to visualize the flow. This will be focused on direction, velocity and patterns of the flow.

Secondly the 'autonomous stable flight' implies for the development of a controller, which will be the main deliverable from the thesis work. To decide on the type and what form the controller must have research into existing controllers will be done, to find the advantages and disadvantages. Besides looking into literature a Matlab simulation will be conducted in which a PID controller in both feedback and feedforward structure is tested. This will provide additional information on the challenges that need to be tackled by the controller, and will aid in the understanding of a controller. How the controller will handle disturbances is an important aspect, e.g. do they model the disturbances or are estimations needed based off sensor measurements. This will namely lead to different decisions for the actual hardware. Moreover there will be looked into literature to find what sensors are used for obtaining relative distances to the surroundings. Flying indoors will impose constraints on type of sensors, the sensors itself may pose constraints on the other hardware of the MAV.

To summarize several research topic questions are proposed. It starts with the main question, which is supported by more categorized sub-questions.

How can autonomous, stable and safe flight of an MAV quadcopter within a small enclosed area be achieved?

1. What are the influences of aerodynamic effects on quadcopter stability
   - What type of aerodynamic effects occur within a small area?
   - How do these aerodynamic effects interact with the quadcopter?
   - What mathematical models are derived for the aerodynamic effects?
2. What type of controllers exist within the area of autonomous MAV flight?
   - Are the controllers based on mathematical models?
   - How does the controller handle disturbances?
   - What sort of hardware was used for the development of existing controllers?
3. Which sensors used on an MAV are involved in gaining knowledge of the surroundings?
   - Which type of sensors used on an MAV exist for obtaining distance measurements to area boundaries?
   - What are the constraints on sensors for flying indoor?
   - What are constraints introduced by a sensor on the MAV?

The literature study will mostly focus on answering all the sub-questions, to provide the basis for the development of the controller.

3 Literature

Before a solution to reach the objective can be formulated, there will be looked into literature. The literature will provide information on what is already done by others and how that can be used within this work. The subjects for the literature study have been set out in chapter 2. First of all section 3.1 will explain the different aerodynamic effects encountered when flying indoors and which models describe those effects. Then section 3.4 will provide some general information on autonomous controllers, such that advantages and disadvantages are known before going deeper. section 3.3 will describe methods used in obtaining information on external disturbances, different to using models for the aerodynamic effects. Thereafter section 3.2 shows some controllers used when flying in an area where the aerodynamic effects are encountered, with some methods using aerodynamic models from section 3.1 and others using disturbance estimation from section 3.3. Then section 3.5 dives into methods to handle small collisions. With at last section 3.6 discussing sensors for measuring distances to the surroundings.

3.1 Indoor Aerodynamic Effects

Flying indoors with an MAV brings several challenges, especially when flying in small enclosed areas, where close encounters with the boundaries (e.g. walls) will happen. One of those challenges is the occurrence of several aerodynamic effects.

3.1.1 Ground Effect

One of the first studies done for the ground effect with helicopters is done by Cheeseman and Bennet [8]. The reduction of induced velocity due to the presence of a ground plane causes an increase in thrust compared to a similar situation without the presence of the ground plane. They created a relationship that shows the ratio of extra thrust generated when a single propeller helicopter (main rotor and smaller vertical spinning tail rotor) is in close proximity to the ground plane. This relationship is shown in Equation 3.1.

$$\frac{T_g}{T_\infty} = \frac{1}{1 - \frac{R^2}{16z^2}} \qquad \text{with} \quad z > 0.25R \tag{3.1}$$

Within this relationship T_g is the thrust within ground effect and T_∞ the thrust far enough away from the ground plane where no effect is encountered. R is the propeller radius and z equals the distance from the propeller plane to the ground plane. The singularity sets a minimum height for which the formula holds. Within the thesis project a quadcopter will be used which might not generate the same ground effect and as such Equation 3.1 might not be valid. Conyers et al. [9] performed tests with a quadcopter setup to investigate the validity of the relationship. They created a rigid test stand that allows for different distances of the propeller to a ground plane, by reversing the propellers orientation upwards flow is created and thus a wooden plate above the propeller acts as a ground plane. The experiment validates Equation 3.1 by running single propeller tests, results seen in Figure 3.1. After the single propeller test multiple quadcopter configurations (different propeller separation distances) are tested. Those results can be seen in Figure 3.2. Within the quadcopter experiment a controller makes sure that all four motors spin at the same speed to mimic perfect hover.

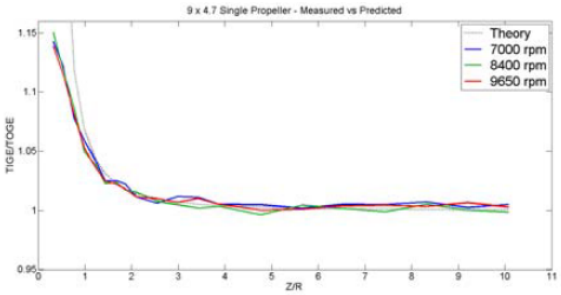


Figure 3.1: Ground effect for single propeller at different RPM [9]

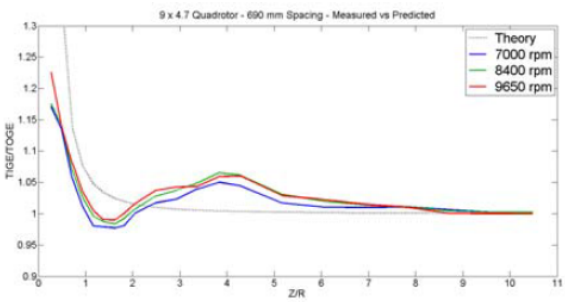


Figure 3.2: Ground effect for quadcopter configuration at different RPM [9]

From the tests it could be concluded that the ground effect for a quadcopter could not be described by Equation 3.1. The effect appears both further from the ground and is stronger in magnitude. An interesting region appears around the two rotor radii away, where the thrust ratio drops (even below 1). At that distance the airflow from different propellers collides and is curled away from the surface, creating a vortex ring that circulates around the spinning rotor. When this happens the loss of thrust cannot be be corrected by increasing power which in aviation is also known as a vortex ring state [9]. From this experiment it also seems that the effect is

independent of RPM, as the lines mostly overlap considering slight measurement errors. He et al. [1] also found the ground effect to be independent of Pulse Width Modulation signal (PWM) provided to the motor, which controls the RPM. Sanchez-Cuevas et al. [10] also investigated the ground effect for a quadrotor MAV with a similar test bench but using a full quadcopter instead of just four separate propellers. The results from their experiments are shown in Figure 3.3

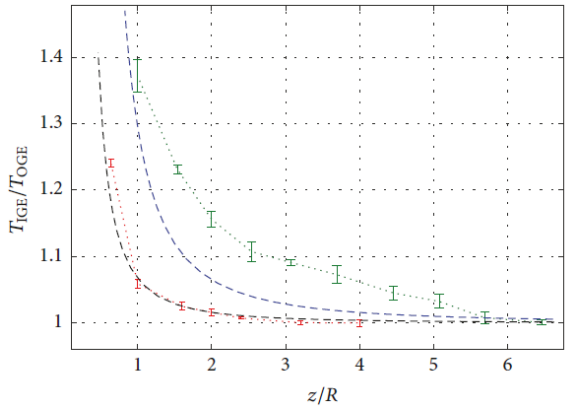


Figure 3.3: Ground effect for a quadcopter. The green line shows the quadrotor results and a single rotor experiment is represented in red. The blue and black lines are models generated from potential flow with the method of images [10].

Once more it can be concluded that the quadrotor effect is stronger. This is caused by the fountain effect: due to collision of flow from different propellers below the body it is curved away from the ground surface and impinges on the drone body forcing it upward. The drop in thrust is not seen in the results from Sanchez compared to Conyers work, this may be explained due to usage of different MAV geometry. The generation of a general model (without intensive model fitting via experiments for each MAV) for the quadcopter ground effect might thus be near impossible if the geometry (besides rotor size) has such a large impact.

3.1.2 Ceiling Effect

Similar to a ground plane a ceiling plane also interacts with the MAV. Like for the ground effect, the ceiling obstructs the flow and decreases the induced velocity created by the rotor plane increasing the thrust [11]. With the ground effect leading to some sort of air cushion between the MAV and the ground pushing it away and as such protecting it against ground collision, the ceiling effect causes the MAV to be sucked towards ceiling leading to contact which might damage the MAV. As the ceiling effect is caused by the same physical principle as the ground effect, Conyers at el. [11] also conducted experiments to see if Equation 3.1 holds for the effect in both single propeller and quadcopter configuration. The experiments were executed with the same test setup, but this time the airflow created by the propellers was directed away from the plate. Figure 3.4 shows the results for single propeller experiments, Figure 3.5 the quadcopter configuration respectively.

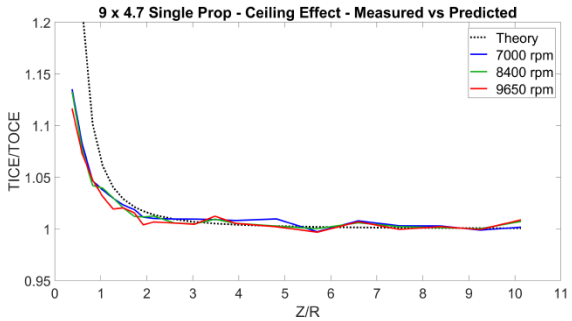


Figure 3.4: Ceiling effect for single propeller at different RPM [11]

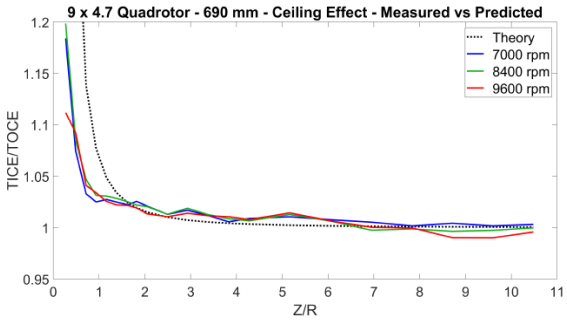


Figure 3.5: Ceiling effect for quadcopter configuration at different RPM [11]

In contrast to the ground effect Equation 3.1 seems to describe the ceiling effect rather well, up until the

singularity where the relationship goes to infinity, but the experiment shows finite results. The results are similar for varying propeller spacing. Sanchez et al. [12] also investigated the ceiling effect but used a single propeller only. Besides the lower induced velocity creating the increase in thrust they observed that the rotor spun faster when getting closer to the ceiling plane, while keeping the input signal to the motor constant. This can be seen in Figure 3.6, where the PWM is practically equal for all experiments but the RPM shows a maximum difference of around 300 RPM.

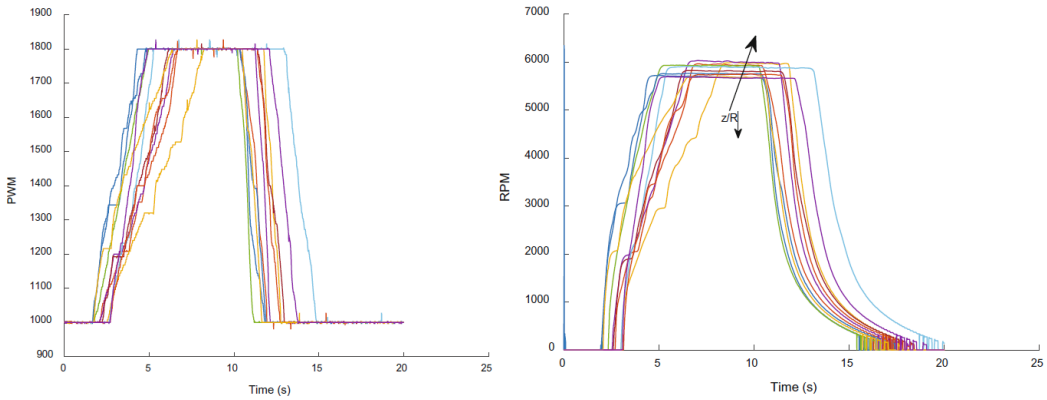


Figure 3.6: PWM signal and RPM for different distances to the ceiling plane [12].

Jimenez-Cano et al. [13] conducted experiments with an MAV having ducts around the propellers (the ducts are not entirely closed). The ducts seem to enhance the ceiling effect which is an interesting find, as flying indoors might require the propellers to be protected against collisions. Due to the stronger effect the relationship by Cheeseman and Bennet is not valid anymore. Without ducts Equation 3.1 seems to describe the ceiling effect for quadcopters, but either experimental measurements or an adaptation to the relationship needs to be made to describe the ceiling effect beyond the singularity. Moreover the ceiling effect starts more drastically and closer to the ceiling when compared to the more subtle growing ground effect. This is an important finding as there will be less time to compensate before a collision with the ceiling might happen, putting a constraint on delay within the future controller.

3.1.3 Wall Effect

When flying inside a room, four of the six environmental boundaries are walls, not considering a possible doorway. Sanchez-Cuevas et al. [12] also used their test stand to see if the wall has an effect on the thrust. Due to the propeller plane being perpendicular to the wall within their test setup, and helicopter theory assuming airflow being almost perpendicular to the rotor plane almost no effect is expected. The results confirm this as a deviation of around 2% in thrust is found compared to the 25% of extra thrust in ground effect respectively. Within actual flight the chances are slim that the rotor plane will be perpendicular to the wall. When the MAV is tilted the wall will act as a ground plane when tilted away and ceiling when tilted towards, this is expected to increase the induced effects when flying close to walls. The wall behaving like a ground or ceiling plane is also supported by findings from Jarding et al. [14]. This tilt causes the effects to be different in magnitude across the quadcopter as the rotors are at different distances from the wall and as such introduce a disturbance moment.

3.1.4 Combination of Effects

When flying indoors, situations will exist where there will be flown close to multiple environmental boundaries, resulting in combinations of the effects. Jardin et al. [14] investigated the In Channel Effect (IChE) which is having both a ground and ceiling plane in close proximity. Two experimental results for a single propeller (Radius 125 mm) are shown in Figure 3.7 and Figure 3.8. In these figures three different scatterplots can be seen, one for the ground effect, one for the ceiling and one for the combined effect. Note the usage of Power Loading (PL) instead of thrust, PL is the ratio between the thrust and power of the rotor.

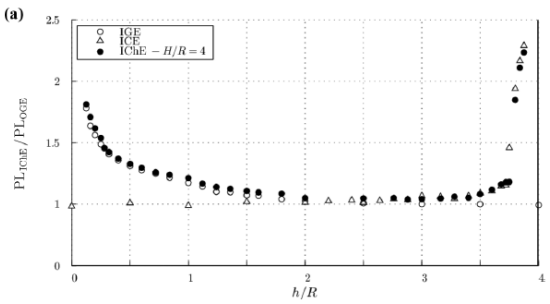


Figure 3.7: IChE for ground-ceiling distance of four rotor radii [14].

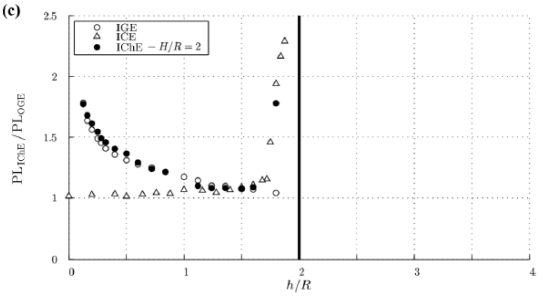


Figure 3.8: IChE for ground-ceiling distance of 2 rotor radii [14].

From these experiments they concluded that it is valid to say that the IChE can be represented as a linear combination of the ground and ceiling effect. Sanchez-Cuevas et al. [10] conducted a single propeller experiment in proximity of a ground plane and wall. From the result in Figure 3.9 a lowering of the thrust is noticed (ratio dropped below one) and as such the ground model is not valid anymore. They suggest that the upward flow between wall and propeller disturbs the flow significant enough in comparison to normal thrust operation decreasing the efficiency, and the fountain effect not being strong enough to compensate. Prothin et al. [15] investigated the aerodynamics of an MAV in a corner close to the ground. Their main finding was that the location of the tip vertex at the wall side created by the spinning rotor moved to be located above the rotor plane instead of below. Even though no concrete numbers where given on thrust performance this finding makes it arguable that a ground effect model developed via experiments with a ground plane only will not be valid anymore. As such describing all effects with exact models might be difficult and complex meaning they will not be able to perfectly match reality.

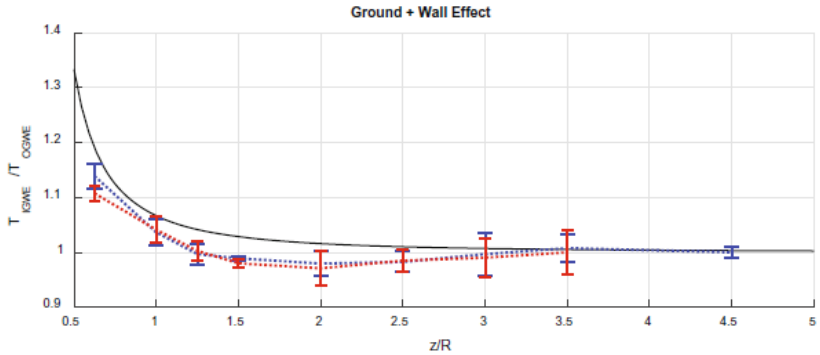


Figure 3.9: Combination of ground and wall plane. Black line is classical ground effect model from Equation 3.1, the blue line is for propeller tip distances of 15 centimeters and the red line for a distance of two centimeters [12].

All the effects influence the amount of thrust generated by the propellers. As the drone will be in motion the propeller plane of the drone will not be parallel with ground and or ceiling, and not perpendicular to the wall. For this reason the effect will be unequal across all four propellers providing a disturbance moment as the propellers are placed at a certain distance from the center of gravity of the MAV. For the ground effect it is a stabilizing moment, as propellers further away provide less thrust compared to those closer to the ground. This stabilizing moment however will make it more difficult for the controller to keep a fixed pitch angle in forward flight when In Ground Effect (IGE). Within ceiling effect the moment is destabilizing as the propeller closer to ceiling has more thrust than a propeller further away. For the wall effect there can be both a stabilizing and destabilizing moment dependent on the flight situation. For flying towards a wall (wall provides some ceiling effect) the moment is stabilizing, the opposite holds for flying away, both have similar reasoning as before.

3.1.5 Models from Literature

Previously the impact of the aerodynamic effects on a quadcopter have been discussed. Now several models developed in literature will be shown. This will provide an overview of what type of models are developed and what challenges may need to be overcome. The models shown are focused on ground effect, similar findings

are there for the ceiling effect. The ceiling effect however could already be modeled relatively accurate by Equation 3.1.

The model from Cheeseman and Bennet [8], provided in Equation 3.1 is one of the most basic found in literature. It is only dependent on the radius of the rotor when looking at drone geometry. For a quadrotor in ground effect it was found that Equation 3.1 underestimated the effect, but for the ceiling effect it does hold. A more simple linear relationship for the ground effect was found by Wei et al. [16]. To determine this equation a test-setup shown in Figure 3.10 was used. This setup is easy to recreate and thus may be recreated when aerodynamic effect experiments are deemed necessary within the thesis work. From the results a linear equation shown in Equation 3.2 was fitted.

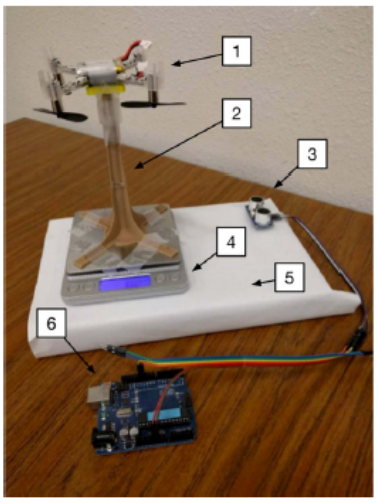


Figure 3.10: Ground Effect measurement experimental setup. It includes (1) a Crazyflie 2.0 quadcopter, (2) a 3D-printed support, (3) an ultrasonic sensor, (4) a digital scale, (5) a flat platform, and (6) a Arduino UNO microcontroller [16]

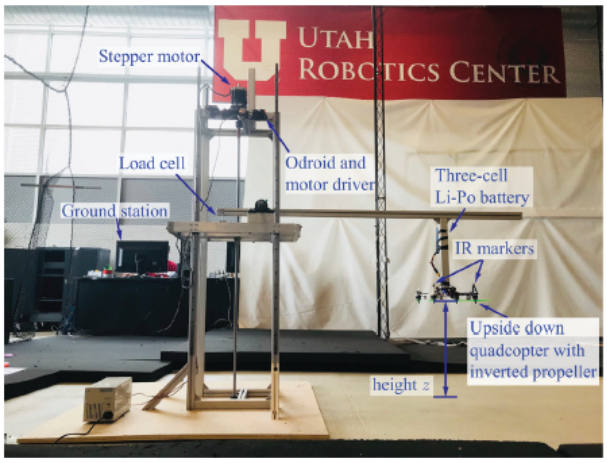


Figure 3.11: Ground Effect measurement experimental setup by He et al. [1]

$$\frac{T}{T_0} = \begin{cases} -0.0373\frac{z}{R} + 1.1651, & \text{if } \frac{z}{R} < 4.42 \\ 1.0000, & \text{otherwise} \end{cases} \tag{3.2}$$

Once again z and R are the relative distance to the ground plane and rotor radius respectively. T is the IGE thrust and T_0 Out of Ground Effect (OGE) thrust. Sanchez et al. [10] developed a more complex model shown in Figure 3.3 in blue, with Equation 3.3 representing the equation.

$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1-(R/4z)^2 - R^2(z/\sqrt{(d^2+4z^2)^3}) - (R^2/2)(z/\sqrt{(2d^2+4z^2)^3}) - 2R^2(z/\sqrt{(b^2+4z^2)^3})K_b} \tag{3.3}$$

T_{IGE} and T_{OGE} being the thrust values for IGE and OGE. R and z the rotor radius and distance to ground plane. d the separation distance between four coplanar rotors. b the separation distance between two opposite rotors and K_b being an empirical body lift coefficient. This relationship does have a singularity and thus shoots to infinity when the distance between rotor plane and ground plane converges to zero. The model is derived from potential flow with the method of images, from which the model from Cheeseman and Bennet was derived as well, thus not solely being based on experimental data.

He et al. [1] also created a model for the ground effect. By conducting experiments with the test stand shown in Figure 3.11 which is a more complex setup compared to the one from Wei et al. With this a model without singularities was created and is shown in Equation 3.4.

$$\frac{T(z)}{T_\infty} = C_a e^{-\frac{C_b}{R}z} + 1 \tag{3.4}$$

$T(z)$ and T_∞, being IGE and OGE thrust. R and z rotor radius and ground distance. C_a and C_b are coefficients that were fit using experimental data using the test-bench shown in Figure 3.11.

More models exist, besides the three mentioned ones. However these three models provide an adequate reflection of the research done in this field, as they show some key differences. The model from Wei et al. (Equation 3.2) is linear, thus when implemented into the nonlinear drone dynamics no extra nonlinear elements are added. Moreover it has no singularity, and thus puts no constraints on the application within the entire flight region. Being linear is computationally less expensive, this is an advantage compared to the other two models. However it is expected to be less accurate, as previous experimental results from other studies clearly show a nonlinear increase in thrust when getting below $\frac{z}{R} = 1$. This underestimation of the effect may lead to too low of a response by a controller when merely based on this model, it may however provide a reference for the effect and be used to compensate some of the effects relatively easy. The other two models are nonlinear, of which the model from Sanchez (Equation 3.3) has a singularity. The singularity may be a problem when a low profile MAV is used. The singularity causes the estimation to shoot to infinity, implying infinite thrust, which will definitely lead to problems. The model from He et al (Equation 3.4) does not include a singularity and provides a finite thrust for all distances to the ground plane. Then the two nonlinear models both include coefficients that need to be determined from data and thus experiments must be conducted. This implies extra work compared to the linear model, which might not be worth it for the following reason. As the thesis will cover flight in small enclosed spaces a combination of effects may be experienced, for which a combination of several models might be needed. Even though some models describe a single effect accurately and Jardin et al. [14] found that the ceiling and ground effect could be added linearly the models may still be too inaccurate. Then having a simpler linear model without the need of conducting experiments and use it to compensate some of the effects and use another technique such as a disturbance observer discussed later in section 3.3 to compensate the rest may be better, simpler and computationally less expensive solution compared to the nonlinear models. Moreover the nonlinear models that need experimental fitting may also need support from a disturbance observer or other tool as they are also not perfect.

A final note needs to be made for the review on aerodynamic effects. All the experiments presented from literature were never conducted within a fully enclosed area, meaning no flow circulation in close proximity to the test stand. As the thesis work will focus on enclosed flight, flow circulation will play a role and affect the aerodynamics causing the thrust variation. For example due to circulations the IChE might not be a simple linear combination of the ceiling and ground effect due to further away walls affecting the total flow. To investigate the flow circulation Computational Fluid Dynamics (CFD) will be used to get a grasp of the intensity of the flow and flow speeds, which will be elaborated on in section 4.1.

3.2 Controller Overview

Before having a look into the work done within flying in enclosed spaces and for example the handling of aerodynamic effects a general overview of different control techniques is given. The overview will both discuss linear and nonlinear control techniques. Some important characteristics to discuss are the following: handling of disturbances caused mostly by the aerodynamic effects. The dependence on exact mathematical dynamic models, due to an MAV quadcopter being highly nonlinear and underactuated obtaining an exact representative model can be difficult. Lastly the computational needs of each control strategy, as the onboard hardware is mostly constrained by weight and power usage, both affecting flight time. The section will close off with a table summarizing the discussed techniques.

3.2.1 Linear Controllers

The most used linear controller is a Proportional-Integral-Derivative controller (PID). It is applicable to Single-Input Single-Output systems (SISO). The simple structure makes it easy to integrate and requires a low computational cost [17]. The basis comes down to controlling the error (difference between reference and actual value) to zero. It can do this via the usage of three gain terms one applied to the error itself, one onto the derivative of the error and the last one onto the integrated error respectively [18]. PID control is non model based (it only uses the error) and thus lacks robustness, if the error measurement is not precise enough the controller might send control signals that are fitted for a completely different situation and thus can worsen the situation. Advantageous of the PID control is ease in creating cascaded loops, creating possibilities for combination with other more robust techniques. As PID is a linear control technique it is tuned around a single equilibrium point, affecting performance in uncertain situations such as disturbance handling. Scheduling different gains can improve the robustness, but introduces more complexity as a higher level controller needs to decide on when to use which gains.

A Linear Quadratic Regulator (LQR) is another linear control technique, but it can deal with both SISO and Multiple-Input Multiple-Output systems (MIMO), and thus a single controller might be able to control the entire system compared to PID where a separate controller needs to be designed for each control parameter. LQR is an optimization based controller and calculates the control inputs via minimization of a predefined cost function [17]. Optimization has higher computational cost, but as it is linear problem no real problem is expected. This technique can make use of noisy conditions and deal with some missing information when controlling the system, thus being more robust and handling disturbances better than PID control [18]. Once more as it is a linear technique, the system needs to be linearized around a certain condition affecting the robustness. Moreover it is a model based technique, meaning parameter uncertainties in the system dynamics can affect the performance.

The aforementioned linear control techniques are used for control of an MAV within literature, but often they are combined with another nonlinear technique. This is done such that the disturbance handling and robustness of the final controller increases. An example is given by He et al. [1] using an outer PID controller but including a nonlinear disturbance observer in the controlled system to improve performance for IGE flight. Thus for dealing with the harsh aerodynamic effects discussed in section 3.1 a sole linear controller will not be the solution, motivating the need of looking into nonlinear controllers.

3.2.2 Nonlinear Controllers

Compared to the linear techniques there exist a wide variety of controllers directly applicable to nonlinear systems. A widely used method is backstepping control, which is a recursive algorithm. The advantage of this technique is the stability guarantee via Lyapunov analysis and the ability to deal with external disturbances. A negative aspect is known as the 'explosion of terms' which is caused by the repetition of differentiation of the virtual inputs. This implies a lack of robustness, and shows the dependence on an exact mathematical model, which as said can be difficult to obtain for an MAV. One aspect of this technique is the need of a strict feedback form, needing full state feedback [19]. The full state feedback means all states must be directly measured or estimated via those measurements, introducing constraints to the system and add computational cost for implementation.

Sliding mode control is another nonlinear control technique. It uses a discontinuous control signal to control the system to slide along a set trajectory. Having a low sensitivity to external disturbances and model uncertainties is a pro. This method is using discrete time, which may create the chattering phenomenon [19]. This is an oscillation around the sliding surface which is situated in between the two control structures that are used to stay on that surface. The switching between the variable control structure introduces some delay in the system, this may be a problem if it becomes to large. As the ceiling effect was seen to be abrupt and aggressive to much of a delay can lead to crashing before the controller can react. Stability can once again be guaranteed via a Lyapunov analysis [20].

A nonlinear optimization controller is the Nonlinear Model Predictive Controller (NMPC). As the word predictive already hints at this controller tries to predict the future up until a set prediction horizon. Once again this method optimizes a cost function that can be subjected to constraints. The optimization method tries to solve a non-linear, non-convex problem which is a challenging task, thus being a computationally intensive method, which may lead to issues in run time during real time application. A disadvantage of this method is the difficulty of handling time varying parameters as exact system knowledge is required for the prediction model, thus also needing full state feedback. This also implies being sensitive to disturbances which are not taken into account within the prediction model. The constraints are a pro of this method, for example a constraint can be set on maximum motor current that may be provided such that controller takes into account the limits of the system [17].

Another strategy that is able to deal with uncertain parameters is the adaptive control method. This method can adjust to an uncertain system to meet specific requirements [17]. This strategy is mostly combined with another control strategy and is then composed of a controller part and the parameter adjustment part. Due to different techniques for adaptation of the parameters, for example least square fitting, the method can vary in computational load. It is however an interesting method for dealing with the aerodynamic effects, as the rotors delivering different amounts of thrust in the different effects, implying different motor parameters during flight time, to which an adaptive method can respond. A downside can be poor tracking performance when the adaptations are not fast enough, either caused by slow real-time implementation or noisy measurements leading to slow convergence of the estimated parameter's. This method also highly relies on sensor measurements to be able to adjust to different conditions. Model dependence relies on the adaptation algorithm chosen [21].

A control strategy that has similarities to human thinking is the so called fuzzy control, which uses fuzzy logic. Unlike a computer that can only categorize something bitwise (0 and 1) fuzzy logic makes it possible to have multi-valued logic. Due to this categorization it indirectly deals with uncertainties. Rule based fuzzy logic is a non model based method that uses a set of predefined rules to categorize inputs and from there determine the corresponding outputs [17]. To create this rules a high level of expert knowledge of the system is needed, which may for example come from experimental data. Robustness also depends on the experts knowledge of possible situations that the controller might need to deal with. Moreover determining stability of the controller mathematically is not possible, thus needing simulations or even test flights which might damage the system due to instability. An advantage is the simplicity of the structure due to human-like thinking making it easy to see what the controller is doing [22].

A method that tries to cancel the nonlinearity's of the system is dynamic inversion. It inverts the dynamics of the system after which it is multiplied with the system. This linearizes the system and as such makes it possible to apply linear control techniques afterwards, which are less computationally intensive. This is an advantage as linear controllers are in general more easy to work with and several tuning techniques exist. The inversion implicates model dependence [17]. Incremental Nonlinear Dynamics Inversion (INDI) is such a method. It has the advantage over normal dynamic inversion that it can handle parametric uncertainties better. However a model of the controls of the system is needed. It is a sensor based technique meaning noisy measurements may degrade performance [23].

The last method to be discussed is feedback linearization. By using a nonlinear feedback method the to be controlled system is linearized. This once more makes it possible to use linear control theory for controlling the system. The linear control solution is converted back to nonlinear domain to control the actual system [20]. Due to linearizing it is sensitive to parameter changes and needs an exact model of the system. It may also need Lie derivatives during linearization which are difficult to determine and sensitive to external disturbances and sensor noise [22].

Finally all the controllers are summarized within Table 3.1. It shows the advantages and disadvantages for each of the methods discussed above.

Table 3.1: Controller overview ([17]-[23])

PID Control		**Linear Quadratic Regulator**	
Advantages • Single Input Single Output (SISO) • Low computational cost • Easy cascaded loop implementation	Disadvantages • Non model based • Lack robustness • Linear control method • Tuning around equilibrium point (hover)	Advantages • SISO and MIMO	Disadvantages • No input constraints • Lack robustness
Backstepping		**Sliding Mode**	
Advantages • Uncertainties handled up until certain level • Stability guaranteed via Lyapunov • Fast Asymptotic convergence	Disadvantage • Strict feedback form • Explosion of terms (large magnitude control term) • Exact mathematical model needed • All states need to be known • Steady state error	Advantages • Robust against parameter variation and uncertainties • Simple Structure • No simplification of dynamics • Stability guaranteed via Lyapunov	Disadvantages • Chattering • Discontinuous control laws
Nonlinear Model Predictive (NMPC)		**Adaptive**	
Advantages • Handles constraints • Optimal control input • Predict future control moves	Disadvantages • Computationally expensive • Accurate state measurements (system knowledge) • Heavily dependent on prediction model	Advantages • Wide operation range • both continuous and discrete tuning	Disadvantages • Limited flexibility of the unknown parameter vector • Complex adaptation laws • Full state knowledge
Fuzzy Logic		**Nonlinear Incremental Dynamic Inversion (INDI)**	
Advantages • Model-free • Indirectly deals with uncertainties • Human-like decisions	Disadvantages • High expertise needed • Need training date • Stability only investigated by experiments	Advantages • Sensor based • Robust • partially model based	Disadvantages • Noisy measurements influence performance • Model of controls
Feedback Linearization			
Advantages • Smooth control signal • Linear control theory can be used	Disadvantages • Needs exact model • Sensitive to parameter changes		

3.3 Disturbance Observers and Estimators

One way of describing disturbances is by obtaining a model for them as was seen in section 3.1. Those models described the aerodynamic effects caused by the boundaries of the flight area, but not all disturbances may be encompassed by those models. Another way for obtaining information on the disturbances is by looking into the difference between a sophisticated dynamic model and measurements taken. This section will discuss several methods for obtaining information on external forces and moments, not using an exact model of the disturbance phenomena.

3.3.1 Models

A nonlinear disturbance observer (NDO) is developed and used by He et al. [1] to estimate the disturbances introduced by the ground effect. The observer will estimate the body disturbance torque along all three body axes, and the body force along the body z-axis. The disturbance forces along the x and y body axis are ignored, as the ground effect only has an influence on thrust which acts along the body z-axis. Besides that can the quadcopter not directly influence forces along the x and y body axis due to being an underactuated system. This design choice may be neglected within the thesis work, as inclusion of the other two body axis may extend the applicability of the nonlinear observer for detection of collisions. The observer needs several measurements namely, the Euler angles, angular velocity and vertical body velocity. These are obtained via the onboard IMU, which has a gyroscope, three linear accelerometers and a pressure senso.

$$\hat{k}_m = \frac{mg}{4\bar{W}^2} \tag{3.5}$$

Moreover the motor parameters are needed ($k_w,\ k_f,\ k_m$), for which an empirical relationship is used shown in Equation 3.5 to estimate k_m after which the other two can be solved, this avoids the need of curve fitting from experimental motor data which is a tedious process. This method will however be an approximation of the motor parameters and thus introduce some uncertainty into the observer. The other parameters in the relationship are mass of the quadcopter m, gravitational constant g and $\bar{W}$ equals the average PWM command signal for OGE hover respectively.

The observer is derived from the standard system dynamics seen in Equation 3.6, which is based of the observer from Chen et al. [24]. The observer has the assumption that the derivative of the disturbance equals zero ($\dot{d} = 0$), however within simulations it is proven that the observer is able to track fast time-varying disturbances. In Equation 3.6 x and u are the state and system input, f the system dynamics and g the system input. Thus the performance for this type of observer is dependent on how accurately the modeled system dynamics represent reality.

$$\ddot{x}(t) = f(\mathbf{x}, t) + g(\mathbf{x}, \mathbf{u}, t) \tag{3.6}$$

Another way of creating a force estimator is shown by Kocer et al. [25], namely using an optimization based approach. As this is an optimization problem it will be computationally more expensive compared to the method of He et al. [1]. The optimization problem is described by Equation 3.7 and Equation 3.8, which uses a discrete time system formulated by Equation 3.9. The solving approach is adopted from work by Kühl et al. [26]

$$\min_{x_k, w_k} \sum_{k=L}^{j} ||y_k - h(x_k, u_k)||_V^2 + \sum_{k=L}^{j-1} ||w_k||_W^2 + ||\hat{x}_L - \bar{x}_L||_{P_L}^2 \tag{3.7a}$$

$$s.t.\ x_{k+1} = f(x_k, u_k) + w_k,\ \forall k \in [0, ..., j-1] \tag{3.7b}$$

$$y_k = h(x_k, u_k) + v_k,\ \forall k \in [0, ..., j] \tag{3.7c}$$

$$x_{min} \leq x_k \leq x_{max} \tag{3.7d}$$

$$arg \min_{x_k, w_k} \sum_{k=-\infty}^{L} ||y_k - h(x_k, u_k)||_V^2 + \sum_{k=-\infty}^{L-1} ||w_k||_W^2 \tag{3.8a}$$

$$s.t.\ x_{k+1} = f(x_k, u_k) + w_k\ \forall k \in [0, ..., j-1] \tag{3.8b}$$

$$y_k = h(x_k, u_k) + v_k\ \forall k \in [0, ..., j] \tag{3.8c}$$

$$x_{k+1} = f(x_k, u_k) + w_k \quad (3.9a) \qquad y_k = h(x_k, u_k) + v_k \quad (3.9b)$$

w_k is process noise, $V \wedge W$ are covariance matrices of the measurement and process noise. x the state update equation with system dynamics f. y_k the output with h being the measurement function. k indicates the time step, the hat indicates an estimate and the bar is the estimation value which is provided by solving Equation 3.8, the weight matrix P_L is also calculated via this equation. It can be solved with linearity assumptions, e.g. Kalman filter. $\bar{x} \wedge P_L$ are then used in Equation 3.7 to solve for the final state estimation $\hat{x}$. The optimization is done over a certain horizon with N measurements ($L = j - N + 1$) such that the problem is constrained to a fixed data size and thus does not become intractable. Looking at the states included in Equation 3.9 the needed measurements are position, velocity, attitude angles and thrust force. These measurements are obtained via an external camera system during the experimentation phase. Once more this method is model based and thus performance depends on accuracy of the dynamic model, however the optimization will be better in dealing with uncertainties as it tries to find the optimal state.

Tomić and Haddadin [27] developed a framework for estimation of the external wrench on a quadcopter which is different opposed to the other methods. Their final method is a combination between momentum and acceleration based techniques. The combination is formed such that directly measured values can be used, which are obtained via onboard measurements. The momentum based technique is used for estimation of the external torques and the acceleration based technique for estimation of the external forces. Both methods are based of the quadcopter dynamics written in Lagrange form, which is comparable to starting point of He et al. [1]. The hybrid technique comes down to Equation 3.10, which is visualized in Figure 3.12.

$$\hat{f}_e = mR_{bi}^T a^b - R_{bi}^T f^b \text{ with } a^b = R_{bi}(\ddot{a} - ge_3) \quad (3.10a)$$

$$\hat{m}_e^b = K_I^m (I\omega - \int_0^t (m + (I\omega) \times \omega - \hat{m}_e^b) ds) \quad (3.10b)$$

$$\hat{m}_e = R_{bi}^T \hat{m}_e^b \quad (3.10c)$$

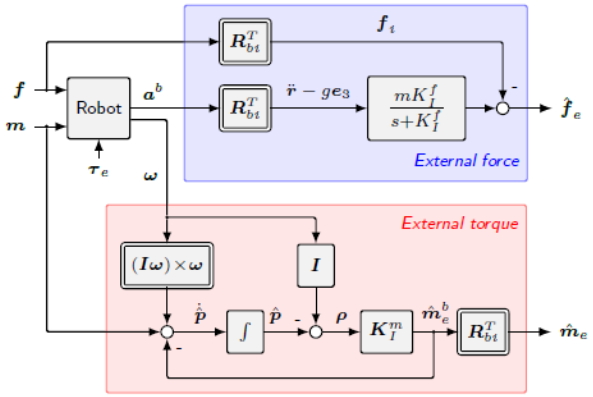


Figure 3.12: External wrench estimator block diagram [27].

$\hat{f}_e$ and $\hat{m}_e$ are the estimated external forces and moments, expressed in the inertial frame. m is the quadcopter mass, I the inertia matrix, f the control forces in body frame, ω the angular velocity and lastly R_{bi} is the rotation matrix from inertial to body frame. Then a^b is the acceleration measured in body frame for which, $\ddot{a}$ equals the linear acceleration in inertial frame, g the gravitational constant and e_3 is the unit vector in inertial z direction. Lastly K is a gain that can be tuned for optimal performance. Within the force estimator the gain acts as a tuning parameter for the first order filter, which is applied to the data from the linear accelerometer to reduce noise. The only two measurements needed within this estimator are the acceleration and angular velocity which can both be obtained via the onboard IMU as was the case with model from He et al. [1]. Moreover the control forces are obtained from a controller, dependent on the controller these forces may need to be converted from PWM to thrust, which implies the need of an accurate actuator model. As with the other techniques the quadcopter model is an important factor for the performance of the estimator.

Lastly Mckinnon and Schoellig [28] introduce anther method for estimation of external forces and torques, their method is based on an Unscented Kalman Filter (UKF). This filter consists of two main steps. First the prediction step, it predicts the state of a system based on measurements, actuator outputs and a dynamic model. The second step is the correction step in which the estimated state is updated to better explain the observed response of the system, namely by incorporating external forces and torques. The discrete dynamical model that is used assumes constant values of the parameters (states) in between time steps. Once more process noise is

taken into account (this models uncertainties, e.g quantization errors). The external force and torque dynamics are modeled as random walk, for which the covariance matrix is a design parameter, in which a low covariance implies slowly changing external forces and torques, and vice versa. Moreover when a purely diagonal covariance matrix is picked it implies that the forces and torques vary independently. The measurements needed for the model are the position and quaternions (quaternions are used instead of Euler angles), which are obtainable via onboard measurements. For the full mathematical explanation of the UKF see the work of McKinnon and Schoellig [28].

3.3.2 Performance

He et al. [1] show no direct performance results for the stand alone nonlinear observer. They extent a PID controller with the nonlinear observer to increase performance, this will be discussed in section 3.4. The nonlinear observer is implemented onboard their Crazyflie platform, thus being computationally lightweight enough for real time performance. Kocer et al. [25] also do not provide performance results of the optimization estimator separately, but also combine it with a controller, this will also be discussed together with the controller in section 3.4.

Figure 3.13 shows results of a force estimation using the method from Tomić and Haddadin [27]. The experiment was conducted with a AscTec Hummingbird quadrotor. The external wrench estimator was running on a remote computer, and the measurements were obtained via a motion capture system, which provides more noisy acceleration and angular velocity data than onboard sensors might provide. As the goal of the thesis work is to implement everything onboard an MAV and thus using onboard sensors, for which the measurements might be less noisy, which then might lead to better results. The results from the experiment shows a relatively smooth estimate with a small delay compared to the ground truth.

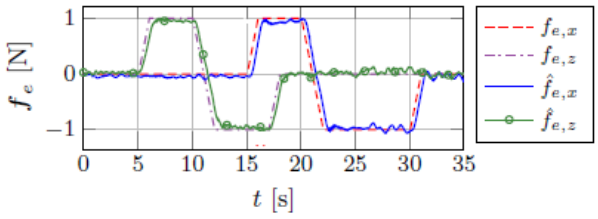


Figure 3.13: Force estimation result by model from Tomić and Haddadin [27].

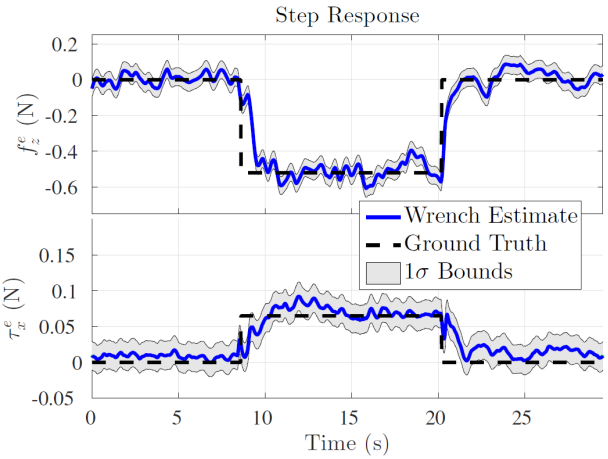


Figure 3.14: External wrench estimator performance when suspending a 53 gram weight under one of the MAV's rotors [28].

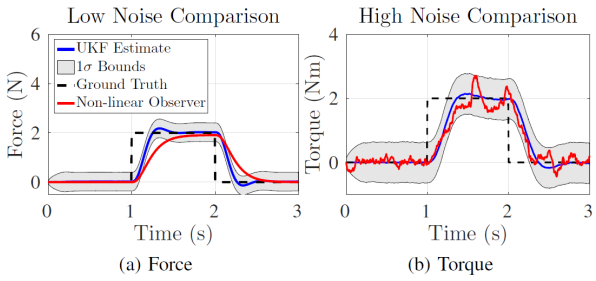


Figure 3.15: External wrench estimator performance comparison to a nonlinear observer [28].

Figure 3.15 and Figure 3.14 are the results from the work of Mckinnon and Schoellig [28]. The experiments were conducted using a Parrot AR. Drone 2.0, with attitude and position measurements being provided by an external camera system. Once again the calculations are not done onboard of the drone. Figure 3.15 is obtained by suspending and removing a 53 gram weight underneath one of the rotors. They reach a rise time of approximately one second and both estimated force and torque show quite some fluctuations, which may cause

oscillatory flight behavior. Then Figure 3.14 shows a comparison of the UKF method to a nonlinear observer, which has similarities to the one from He et al. [1]. The nonlinear observer needed extra low pass filtering to function properly. The UKF shows to have a shorter rise time and especially within the torque estimate have less oscillation within the estimate.

3.3.3 Discussion

A commonality between all the methods is making use of a dynamic model of the MAV, thus all being model based. However due to different approaches one method might be more robust than another. The optimization method, tries to find the best estimate for each time step, thus considering several solutions. This approach is computationally expensive, and thus may not be easy to implement on an MAV due to hardware limits. The other methods explain the difference between the dynamic model and the measurements by setting them equal to the external disturbances, which is thus not a multi solution approach. This is computationally less expensive. Uncertainties in the dynamic model may influence the performance of the estimators, as mismatches between model and reality might lead to higher external disturbance estimates. When choosing to use such a method it is thus important to know the limitations and assumptions on which the dynamical model is based. This will provide insight in what forces/disturbances are not considered and thus will included in the estimate (as that explains the difference between reality and the model).

Another aspect that is important to discuss is delay. From the performances that were evaluated it could be seen that there exists a delay between the actual disturbance and the estimate of that disturbance. This delay occurs due to the time needed for the computations and filtering applied to the measurement data. If the delay is too significant the estimation may be correct, but holds for something that has occurred to far back into the past. This will cause the controller to have a high response time which might cause instability and maybe even a collision. Especially within the ceiling effect the delay is an important aspect, as it occurs close to the ceiling introducing high amounts of extra thrust. This aggressive effect makes it that the system has only a small amount of time to react before it might hit the ceiling. So for the thesis work a small delay is favored.

Lastly some comments about the used measurements within the different estimation techniques. Between the methods the measurements are quite similar, with a commonality being the attitude. Either provided in Euler angles or quaternions, with the latter having an advantage of not including any singularities. However this comes at the cost of an extra normalization step [28]. With most needed measurements being obtainable via the onboard IMU, the position is one which is more difficult, as either a camera system is needed or an integration of acceleration needs to be done, which has a drift and needs a reset every so often. For the thesis work relative distance sensors for obtaining the relative position to obstacles may be used. This means that adaptations to the methods which use position will be needed.

3.4 Controllers Used in Literature

With the main objective of the thesis work being the development of a controller it is important to look into strategies used by others. With the information from the previous sections on aerodynamic effects, general control strategies and modeling disturbances several approaches will be explained and discussed. Within the discussion important aspects such as computational complexity, needed measurements and real time implementation and testing will be highlighted. Those are important for coming to a final design in which aspects of the discussed controllers may be combined.

3.4.1 Models

The aerodynamic model seen in Equation 3.4 and nonlinear observer discussed in section 3.3 are used by He et al. [1] to improve the performance of a PID controller. As a PID controller by itself is not robust enough to handle the ground effect, which is the main focus of his research. The control scheme can be seen in Figure 3.16

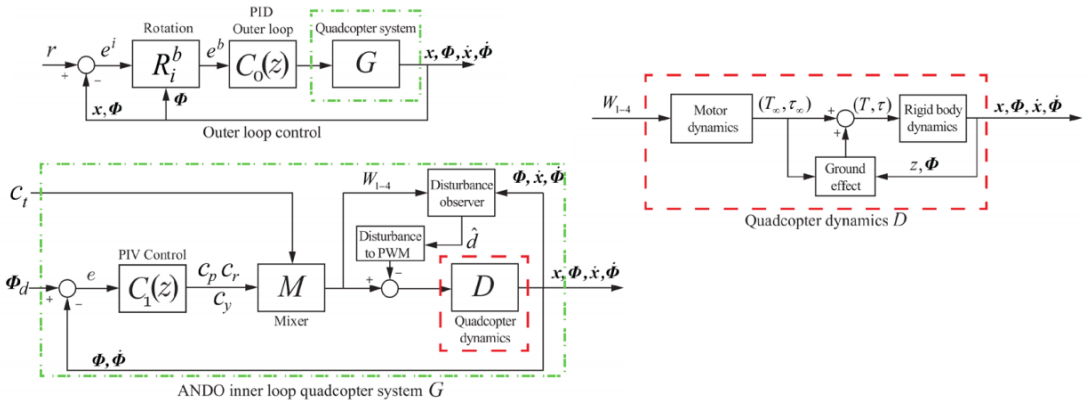


Figure 3.16: PID controller extended with disturbance observer block diagram [1].

Overall the controller is a cascaded PID controller, outer loop position control and inner loop attitude control. The NDO is added within the inner loop, estimates a disturbance force, transforms this into a PWM signal which is subtracted from the PID controller generated signal to counteract the ground effect. Moreover the ground effect is added into the quadcopter dynamics, taking into account the extra thrust generated when close to the ground for simulation purposes. The controller needs measurements of position, z-body axis velocity (nonlinear observer), attitude and angular velocity. For the experiment performed on the Crazyflie platform, the controller from Figure 3.16 is implemented onboard. During testing the IMU provides attitude, angular velocity and z-body axis velocity (fusion of accelerometer and barometer), and an external camera system for obtaining the position.

Sanchez et al. [10] implemented three different types of controllers for investigating partial ground effect flight. This means that not the entire quadcopter is under the effect of a ground plane, thus inducing only extra thrust on some rotors which causes disturbance torques. The three controllers are the following. First of all a standard cascaded PID controller, secondly an estimated torque controller and lastly a feedforward scheme based on rotor height. The PID controller is used as a base reference for performance comparison with the other two controllers. The estimated torque controller makes use of an observer that is derived in a similar fashion as the one used by He et al. The control scheme is seen in Figure 3.17. The second feedforward controller will use distance measurements from each rotor to the ground to prematurely estimate the induced thrust, such that the torque disturbance can be partially compensated, with the control scheme seen in Figure 3.18. Both controllers provide a disturbance force which can be mapped to a motor command and thus being taken into account by the attitude controller (PID).

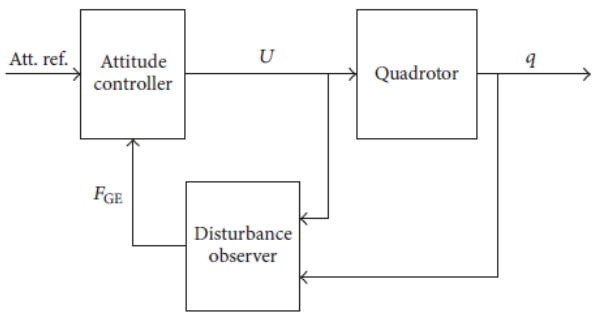


Figure 3.17: Estimated torque controller scheme [10].

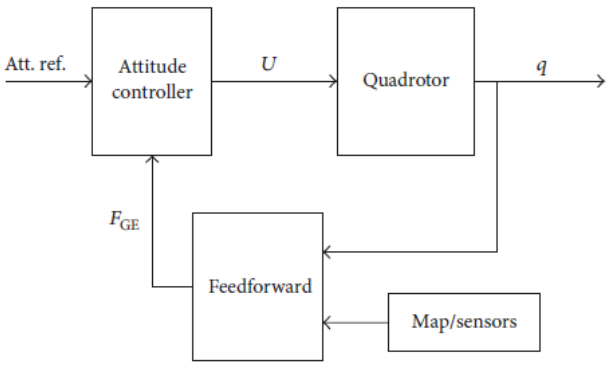


Figure 3.18: Feedforward controller scheme [10].

Experiments were conducted using a PQUAD MAV. It weighs 1200 grams, and has rotors with a radius of 12 centimeter. This is much bigger than an MAV that will be used in the thesis work and will in general have more computational power. Moreover none of the controllers were tested in free flight conditions, only on a test bench where the quadcopter was constrained in both roll and yaw (thus not representative for six degrees of freedom flight). For the measurements no sensors are mentioned, but from the controllers the following can

be concluded. The state from the dynamic model consists of position and attitude. Considering full state feedback these states need to be measured. Moreover the estimated torque controller uses the first and second derivative, thus velocity, linear acceleration, angular velocity and acceleration. The feedforward controller needs the distance to the ground plane which can be measured using a laser sensor or ultrasonic sensor, dependent on their accuracy the performance will be influenced. Tomić and Haddadin [27] created a table seen in Figure 3.19 which shows how several states can be obtained using different sensors. The position may be obtained via an camera system or by double integration of the acceleration.

Sensor	$\dot{r}$	ω	$\ddot{r}$	$\dot{\omega}$
Accelerometer	-	-	●	-
Gyroscope	-	●	-	○
PX4FLOW	△	●	-	○
IMU–odometry fusion	△	●	●	○

Figure 3.19: Measurements provided by sensors which can be fitted onboard an MAV [27]. $\dot{r}$ velocity, ω angular velocity, $\ddot{r}$ linear acceleration and $\dot{\omega}$ angular acceleration. PX4FLOW is an optical flow based technique, thus using a camera. Odemtery also makes use of a camera. The black dot means directly measurable, open dot obtained numerically and the triangle means it can be estimated.

Wei et al. [16] developed a controller using an inner loop full state feedback LQR controller, with an extra feedforward controller. This will stabilize the MAV and track attitude commands. Then the outer loop is a PID controller, with an added Model Reference Adaptive Controller (MRAC) to mitigate the ground effect in z-direction. The MAV dynamics are linearized around hovering equilibrium state, which decouples position and attitude dynamics. The LQR controller uses the Tait-Bryan angles and body angular rates as inputs and then outputs the motor control signals. It stabilizes the vehicle and rejects disturbances. The feedforward inner loop receives attitude and thrust commands and maps this to motor control signals. By adding both outputs of these controllers the final motor commands are generated. The gains and weight matrices used by these techniques were tuned using Matlab. Then the outer loop PID controllers takes inputs position error and yaw angle command, translates this to Tait-Bryan angles and thrust output. Once more tuned via Matlab. Finally the MRAC was added, which is an adaptive control method and thus deals with uncertainties and external disturbances when flying IGE. It adds an extra unknown matched system term to the dynamical system, which is a linear combination of N provided locally Lipschitz-continous basis functions with unknown constant coefficients. For determination of the coefficients a reference model is created. The reference model is obtained via the Matlab System Identification Toolbox, which created a state space system for the vertical dynamics of the MAV when flying OGE (z-position and velocity). This model will recreate the wanted behavior of the system which the adaptive system needs to try and match. The summarized control scheme can be seen in Figure 3.20. Within the experiments the position control was run from a ground station (outer loop) and the attitude control plus state estimator (inner loop) was run onboard the Crazyflie MAV. The position was provided by an external camera system, the Tait-Bryan angles and angular rates by the IMU.

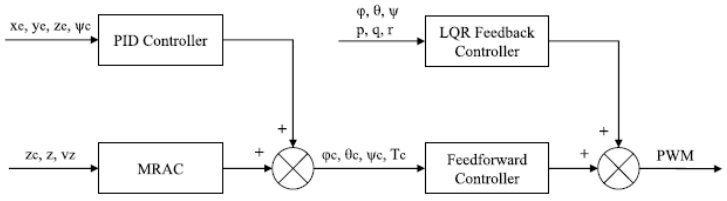


Figure 3.20: Control architecture from Wei et al. [16].

A nonlinear control method is used by Kocer et al. [25], being the NMPC. The control loop is shown in Figure 3.21. A NMPC is used to create desired attitude angles and thrust force, after which a cascaded P and PID controller control the attitude to the desired position.

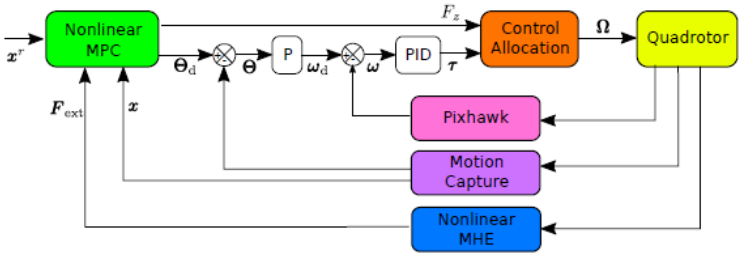


Figure 3.21: The closed loop scheme: a force estimation-based NMPC approach [25].

The NMPC is an optimization function which minimizes the sum of squared state error and input error. Which are based of the reference trajectory and nominal control input. It then outputs the optimal input commands (attitude and thrust force), where the attitude goes into the cascaded PID controller and the thrust is immediately fed into the control allocation matrix. As this control method is combined with the nonlinear force estimator discussed in section 3.3 the same measurements are needed, being position, velocity, attitude angles and thrust force. During experiments the NMPC ran on an external computer. This external computer uses software developed for solving nonlinear optimization problems. As the name already implies the method is highly dependent on the model of the MAV as that is used to find the optimal behavior. An advantage is the ability to include constraints, which make sure that the inputs for the MAV are already bounded and no action is asked of the MAV which it may not be able to do, like command asking more thrust but the motors not being able to generate that.

Smeur et al. [29] developed a controller based on dynamic inversion to counteract gust disturbances (thus not specifically the aerodynamic effects that play a mayor part within the thesis work), but still found applicable as the aerodynamic effect is like a gust. The used inversion technique is the INDI. Both an inner and outer INDI controller are developed, both start by linearizing the quadcopter dynamics via a first order Taylor expansion. The measurements that are used within the controller are filtered to remove noise. As the terms in the controller all need to be from the same instant in time and the filter introduces delay, other terms that are not obtained via measurements need to have the same filter applied. This is important for the correct functioning of the controller. Additionally to the controller a Least Mean Squares (LMS) algorithm that adapts the control effectiveness matrices is used, these matrices contain the effectiveness of each rotor on each axis (roll, pitch, yaw). The inner attitude INDI and outer linear acceleration INDI are than once more enclosed by an PD controller to track position. This is the benefit of the dynamic inversion to which a linear controller can be applied such that linear tuning techniques can be used. The controller needs info on the thrust value which is gathered from experimental setup in which RPM vs Thrust curve is created. This is the part in which the thesis work may differ, as for the aerodynamic effects the thrust is affected, thus by using for example ground effect models this curve can be altered and better estimates of the thrust value for IGE are provided. Other measurements used by the controller are linear accelerations and angular rates.

3.4.2 Performance

The flight test conducted by He et al. is an oscillatory flight which moves into and out of the ground effect. The first two figures (a) (b) in Figure 3.22 show the performance of the PID controller without the NDO. The PID shows good tracking performance when flying completely OGE, but when it needs to get close to the ground it fails, as it cannot compensate the extra thrust provided by the effect. This is expected behavior as a PID controller is tuned for the OGE condition and does not adapt to other situations when the gains are fixed, the integral gain may converge the response but is not fast enough within this experiment. When the NDO is switched on the tracking behavior is improved as seen in (c) of Figure 3.22. When the MAV gets close to the ground the tracking is less smooth but it is able to get to the desired position. This is due to noise from the IMU measurements and due to subtracting the PWM generated by NDO from the PID which is a linear operation, but the thrust to PWM relation has a quadratic shape which is a nonlinear function. The drop when the NDO is switched on is caused by modeling errors (motor parameters, inertia vehicle weight) this introduces an initial error which takes time to correct for.

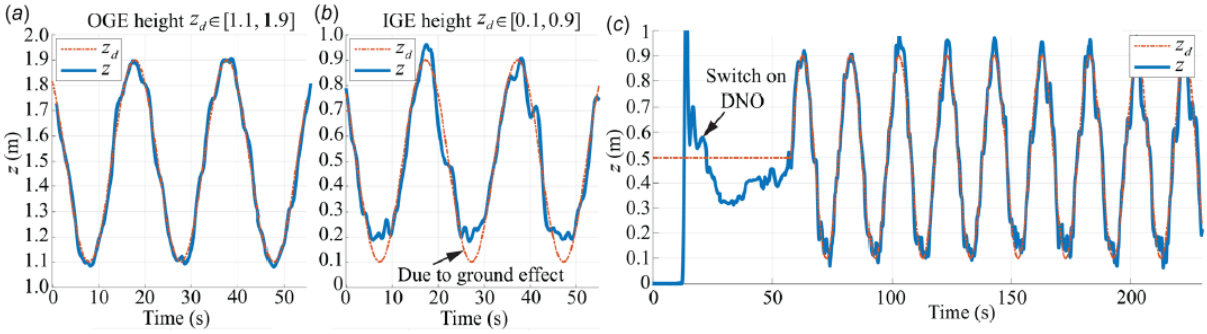


Figure 3.22: (a) and (b) is the tracking performance of PID controller, (c) is the tracking behavior of NDO controller, making use of the Craziefly MAV [1].

As no flight experiments were conducted for the feedforward and torque observer controllers by Sanchez et al [10] results were taken from performed simulations. Within the simulation the ground effect was modeled using Equation 3.3 and a linear relationship for area of a rotor IGE and extra thrust gained was used to take into account partial ground effect. The simulation commanded the MAV to fly over an obstacle at low altitude such that ground effect appears. The results can be seen in Figure 3.23. A standard PID controller is not even able to go over the obstacle with one rotor in ground effect causing an oscillation at the edge of the obstacle. The torque estimator is also not able to reach the reference but got rid of the oscillatory behavior. This is also seen in the pitch angle results, where the initial disturbance is similar to the PID controller, but afterwards it is more or less constant with some small spikes which may be caused by sensor noise. The feedforward controller does reach the reference with almost similar flight result compared to OGE flight (small steady state error exists), both in flight path and pitch angle.

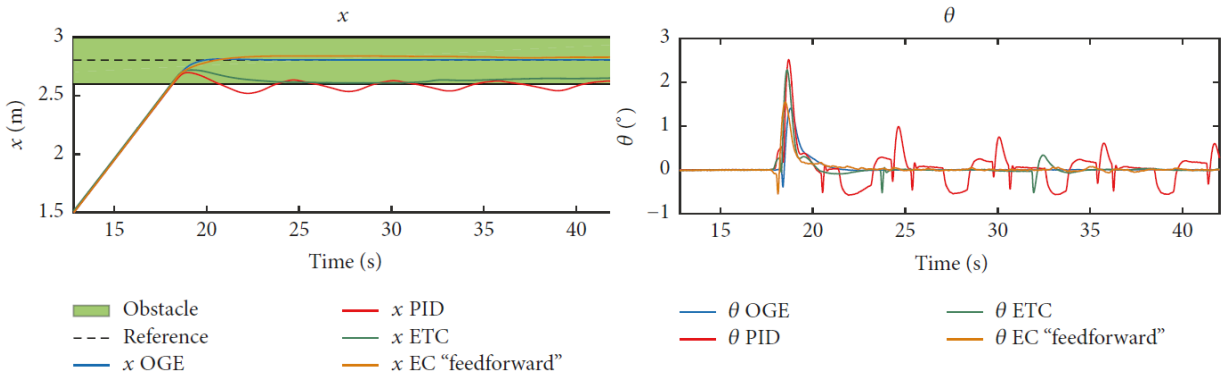


Figure 3.23: Tracking performance and pitch angle comparison for partial IGE flight simulations [10].

With flight experiments conducted by Wei et al. [16] the controller was tested, for which Figure 3.24 shows some results. The experiment conducted was a takeoff and landing procedure. What can be seen is that the PID controller has a very long rise time and does not reach the desired altitude within the recorded time as seen in (a) for the take-off. Similarly within the landing the PID controller slowly descents, thus having difficulty dealing with the extra disturbance thrust. The MRAC controller however is able to perform both a take-off and landing for the set references within approximately ten seconds. The MRAC with linear basis functions has a slower rise time compared to usage of nonlinear RBF basis functions. From all the experiments it was found that the MRAC in general reduces rise time by 80% compared to the PID and the RBF basis function also cause the mean square error and rise time to be 45% and 15% less respectively compared to linear version. For computational cost the RBF needs to update seven parameters compared to only two for the linear version.

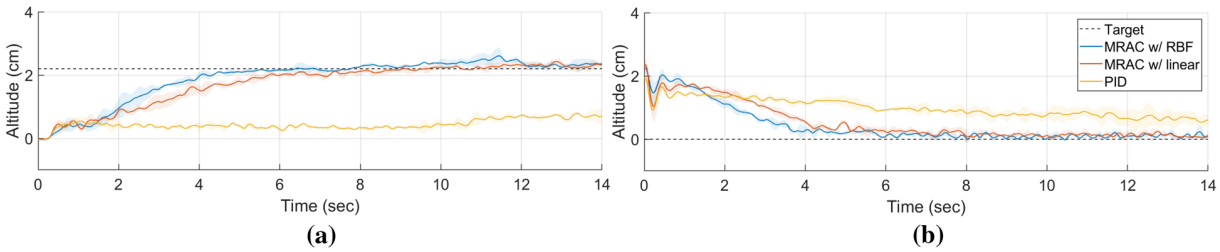
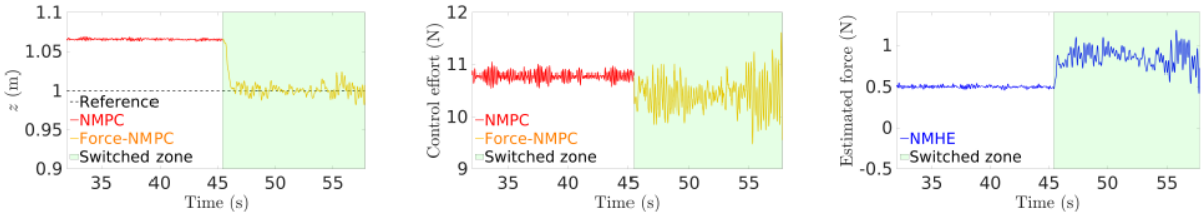


Figure 3.24: Controller performance for take-off (a) and landing (b) flight for a height of $\frac{Z}{R} = 2$ [16].

Kocer et al. also performed flight tests to validate the NMPC controller, aided by an optimization force estimator. The results for a flight with a distance of only $\frac{Z}{R} = 1$ can be seen in Figure 3.25. This shows that without the force estimator the NMPC is not able to fly up to the reference height, but when the estimator is switched on it only takes a split second to reach the reference altitude. Within the results quite some oscillation can be seen, this implies the controller has difficulty to counteract the ceiling effect, which may be caused by the uncertainties in the model used within the optimization. Delay within the estimation used to counteract the effect may also induce oscillation, as the controller is always responding to an effect from the past.



(a) The system performance on vertical axis. (b) Control effort. (c) Force estimation.

Figure 3.25: Close proximity flight using NMPC controller for $\frac{Z}{R} = 1$ [25].

Lastly the performance of the INDI controller by Smeur et al. [29], which was developed to counteract wind disturbances. Figure 3.26 shows outdoor flight test results using an INDI or PID outer-loop controller, both with an INDI inner attitude loop. The flight was a take-off and hover in windy outdoor conditions, using a GPS receiver for obtaining the position, and other measurements retrieved onboard. What can be noticed immediately is that the PID is not able to cope with the disturbance induced by the wind after take-off and shows a high drift for the position when getting of the ground. The INDI also has a small drift but does not show a large change in error after the initial drift, and thus shows higher disturbance handling performance. During one of the experiments however the state estimator did not converge before take-off, causing bias in the accelerometer data which detriments the performance. Moreover some of the INDI tests showed a larger error compared to the one seen in Figure 3.26, this was caused by the acceleration data not corresponding to velocity and position data obtained via the GPS. This was due to errors in GPS measurements.

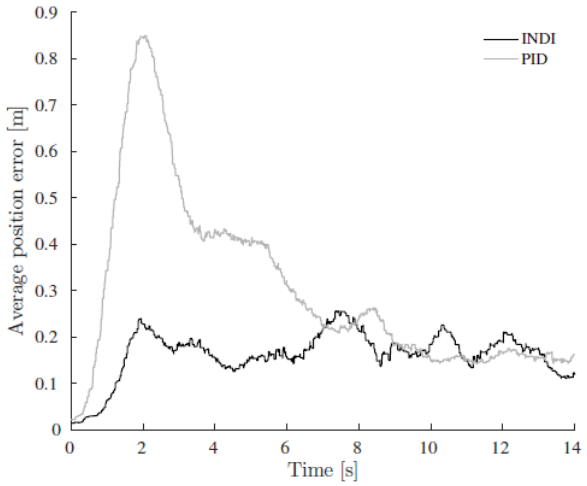


Figure 3.26: Outdoor test flight in windy conditions using a Parrot Bebop [29].

3.4.3 Discussion

Most models shown before are in some way or shape based on a model. One form for a model is making use of an equation which describes an aerodynamic effect. To obtain that model, the authors perform experiments to obtain data for the effect for their specific platform. This data is then fitted to one of the models shown in section 3.1. This will make the controller restricted to that used platform. The other methods are more based on the drone dynamical model, which also depends on geometry parameters but those are mostly already known and thus no experiments are needed, this is an advantage over the aerodynamic model method. The motor parameters which describe conversion of power into thrust might need some experiments if no data is available as was done by Smeur et al. [29].

A PID controller or similar is used widely within the elaborated methods. As that type of controller is simple to implement and easy to tune as was discussed in section 3.2. It is however combined with other methods to increase robustness and effectively counteract disturbances. The shown combinations are with observers, feed forward, adaptive, model predictive and dynamic inversion. With most controllers using the PID controller within the inner loop controlling the attitude of the MAV. The attitude control is a fast dynamical system and thus even though the PID is tuned for one equilibrium point it is able to follow commands well enough in other situations.

Most discussed controllers have shown that they are able to control the MAV within the situation for which they were designed. Several factors are important for the controller that will be designed in the thesis work. First of all delay, which causes a sluggish system, if it is too large. The delay can be created due to several reasons. The computational time of the controller, as for example solving an optimization method done in the NMPC will probably take more time (as it is somewhat of an iterative approach) compared to the feedforward method using an model for the aerodynamic effect which just calculates a result by plugging in values. Moreover with most methods not being run from onboard the drone the real time onboard performance may decrease compared to the shown results. As the MAV will most probably have less computational power compared to offboard computers. Besides computational time, filtering of measurement data will also introduce delay. Dependent on how dependent a method is on measurements the amount of filtering will vary. For example the INDI is a sensor based method and accurate measurements are important, as performance degradation was seen when the GPS and accelerometer provided data which did not match. This means that unbiased, no noise data is wanted which will require more extensive filtering introducing more delay. A method like the LQR optimizer might be less affected by noisy data as it tries to find the optimal situation.

All the controllers need measurements from the MAV platform to gain knowledge of the current situation. Attitude angles and angular rates are used by all methods, likely to be needed in the final design. The authors have shown to use different methods for obtaining those, either using onboard (IMU) or an offboard camera system. For the thesis work it is important to use as much onboard measurements to create a system that is independent of its environment. This will put constraints on the sensors, as mass, power usage are important factors, as they will influence MAV flight time performance. Moreover carrying the sensors onboard might influence there accuracy, for example propellers may introduce vibrations into the MAV body which can lead to extra noise in the measurements. He et al. [1] has shown an onboard IMU is able to provide the attitude angles

and angular rates accurately enough to reach the goal of their work. The position is also needed by all methods to perform a flight, and is always obtained via offboard system within the shown research (except onboard GPS by INDI [29]). As the focus of the thesis work will be on stable autonomous flight and not specifically in path planning and navigation it is likely that the position is obtained from offboard sensors. As obtaining position from for example accelerometers is possible via integration of the data, it is vulnerable to drift and needs a calibration once in while to counteract this, which may be done via an onboard camera and known environmental map. This is however a whole other field of research and beyond the scope of the thesis objective. The thesis work might however use relative position sensors like ultrasonic sensor within the controller to obtain distances from the environment, for usage in aerodynamic disturbance handling.

Finally as a summary for the controllers Table 3.2 provides an overview of the discussed methods. It includes the following information, first of all the type of controller. Secondly if the controller is implemented onboard an actual MAV, as this is the end goal of the thesis work. If the controller is already implemented once, it provides more certainty that the computational cost of the method is not to large to fit on a small MAV platform, increasing the chances of success. Then the main design choices and parameters are shown. This provides an indication of how much of the controller can be adapted to fit the situation better and will be an indication for the amount of work to fit a certain controller to a certain platform. Lastly the measurements needed by the controller are provided, this will provide an indication for the type of sensors that er needed.

Table 3.2: Controllers from literature summary.

Controller	Onboard Implementation	Design Choices/Parameters	Measurements
PID + Nonlinear Observer [1]	Yes: Crazyflie	Aerodynamic model $C_a \wedge C_b$, PID gains, observer gain and matrix, motor parameters,	Position, z-body velocity, attitude angles and rates
Torque disturbance observer [10]	Yes: PQUAD (large MAV)	motor parameters, observer matrix	Position, linear/angular velocity, linear/angular acceleration
Feedforward [10]	Yes: PQUAD (large MAV)	Aerodynamic model K_b, motor parameters	Position, linear/angular velocity, linear/angular acceleration, distance to ground plane
LQR + MRAC + PID [16]	LQR Yes: Crazyflie MRAC No	Aerodynamic model, LQR weight matrices, motor mapping gains, PID gains, MRAC basis function, MRAC reference model	Position, attitude angles, angular rates
NMPC [25]	NMPC No PID Yes: DJI F450	Estimation Horizon (time step), PID gains, Optimization weight matrices, constraints (e.g. max throttle)	Position, velocity, attitude angles and thrust
INDI [29]	Yes: Parrot Bebop	Filter gains, PD gains, Thrust conversion, Online adaptation scheme	RPM, angular rates, linear accelerations

3.5 Collision Detection

When flying in small enclosed areas collisions are tried to be avoided but this may not exclude them completely. It is thus also important to investigate methods that might be able to detect a collision and provide a strategy for recovery.

3.5.1 Models

Besides the development of an external wrench observer Tomić and Haddadin [27] also developed a technique to detect collisions. For the collision detection they extrapolated four phases: detection, classification, isolation and reaction. The first phase is using the external wrench observer force estimation as explained in section 3.3. By setting a threshold on the force, both light and harsh impacts can be detected. As this might be sensitive to noise and modeling errors, a highpass filter is applied. The next phase, namely classification of material for the collision surface is neglected, as it is not within the scope of the thesis objective. Then the third phase is isolation in which the contact location is obtained (contact point on MAV). First of all it is assumed that there is a single point of contact and that the contact force is pointing inwards. The contact force generates a torque on the MAV and has a certain arm on which the contact point must be located. By intersecting the convex hull of the MAV with Equation 3.11 (which describes the torque arm) the point of contact can be found. Using this point a plane can be generated which describes the orientation of the collision surface, being normal to the MAV.

$$r_c = R_{bi}\left(\frac{\hat{f}_e \times \hat{m}_e}{||\hat{f}_e||^2} + \alpha \hat{f}_e\right) \tag{3.11}$$

Within Equation 3.11 $\hat{f}_e \wedge \hat{m}_e$ are estimated external wrench force and moment, R_{bi} rotation matrix from inertial to body frame, α is a free parameter which will be negative for collision in free flight and lastly r_c is the arm of the generated collision moment. Then the last phase, reaction. For the reaction different approaches can be

taken. The responses seeming most useful for the thesis work is either stopping when a collision is detected or performing an equilibrium bounce. The stopping strategy may be useful when flying through small areas with tight margins to the environment, thus implying that backing up after a collision is not an option and might result in another collision. The bounce strategy can be used in areas where the margins are larger. That strategy uses the collision plane and the normal vector. It will create a trajectory along the normal vector away from the collision plane. This new path will then be followed by a controller. A fuzzy logic based strategy is used by Dicker et al. [30]. The entire collision recovery system is seen in Figure 3.27. It comes down to three main phases: detection, characterization and recovery. Within this pipeline, quaternions (q), accelerations (a), angular velocity (ω) and vertical velocity ($\dot{Z}$) are used to find a solution.

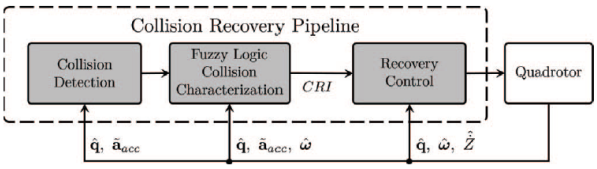


Figure 3.27: Collision recovery pipeline using fuzzy logic [30].

First of all the detection, which uses accelerometer measurements. For now their method only detects horizontal collisions, it may be investigated later within the thesis work if vertical collisions can be detected in a similar manner, when they are needed. To detect a collision the acceleration measurements in x and y direction are rotated from body to inertial frame and corrected for gravity. Then using a threshold on the obtained value will be used to detect a collision. The normal vector to the collision plane is estimated using Equation 3.12.

$$\hat{e}_N = \frac{[\hat{a}_X, \hat{a}_Y]^T}{||[\hat{a}_X, \hat{a}_Y]^T||} \tag{3.12}$$

For which $\hat{e}_N$ is the collision plane normal and $\hat{a}_X \wedge \hat{a}_Y$ are the acceleration measurements. Noise has once again an influence on the collision detection which must thus be taken into account when setting a threshold, they chose one g as the threshold. The characterization phase will determine a Collision Response Intensity (CRI) value which is between $-1 \wedge 1$ and tells the response (flipping away or towards the collision surface) and magnitude. To determine this CRI value a fuzzy logic process is setup. The fuzzy logic process is as follows. The inputs seen in Figure 3.27 are scaled and classified via a membership function, which comes down to a graph consisting of triangular and trapezoidal regions mapping inputs to their degree of membership. Linear segmenting is chosen as this is computational lightweight, thus considering real-time implementation. These graphs are created for their specific platform, as a different platform will respond different to collisions (higher mass MAV might have lower responses for example). Then using a set of rules which are set-up using knowledge of the system and performed experiments [31], the four input degree of membership values are combined and then used to obtain a CRI value. This is once more done using a graph which maps the degree of membership of the combined inputs into a CRI value (linear regions). Then the last phase is the recovery. The CRI value obtained from the fuzzy logic process is mapped into a reference acceleration magnitude, which is then oriented along the wall normal. The mapping is done using a single proportional gain. Then using the reference acceleration orientation and magnitude, correcting it for gravity the desired acceleration is obtained and mapped into body rates which can then be controlled via an attitude controller.

3.5.2 Performance

The performance of the first method by Tomić and Haddadin [27] is shown. First of all the normal vector calculation is tested within a simulation, for which results are shown in Figure 3.28. The MAV is modeled using eight rectangular faces. As can be seen the normal approximation is quiteF accurate except for the horizontal case, which is due to geometric approximation of the MAV. The approximation has the bottom surface already five centimeters below the actual contact point causing a systematic position error. The expected collisions within the domain of the thesis are the straight and vertical ones, thus being less accurate for the horizontal case is of less importance. Secondly a collision flight experiment was conducted in which an MAV (AsTec Hummingbird) collides with a polystyrene block. Only the attitude controller was running onboard, thus the external wrench used for collision control was running on an external computer. For measurements an external camera system was used which obtained position and attitude. Figure 3.29 shows the results for several different control approaches (no reaction, stop, bounce and impedance controller), of which the stop and or bounce are

the interesting ones. Even though the collision detection is highly filtered it is able to detect a collision, for which both the stop and bounce method recover the MAV (PID control), the no reaction procedure ended in a crash.

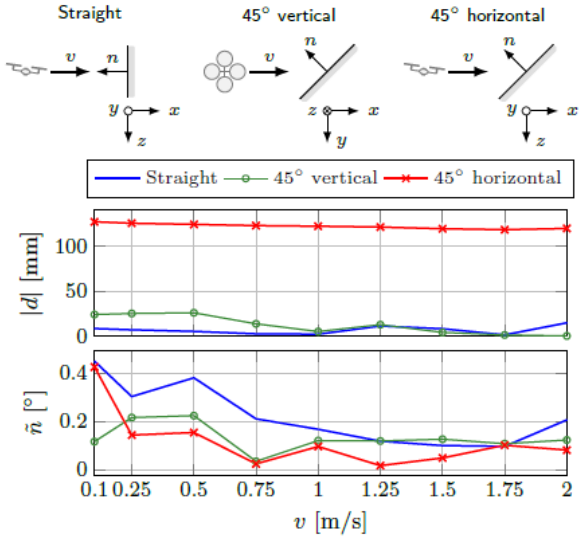


Figure 3.28: Collision normal vector simulation [27]. With d being the absolute distance of the estimated plane position tot he real plane, and $\tilde{n}$ the angular error of the estimated normal.

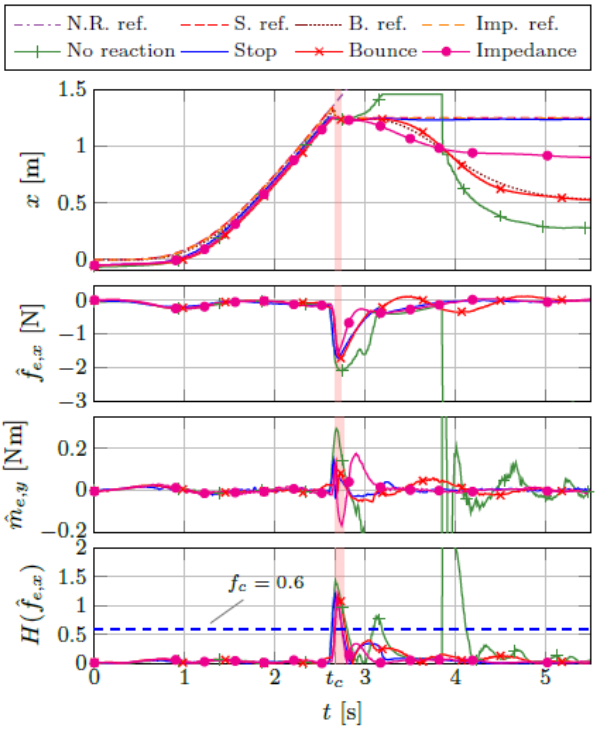


Figure 3.29: Collision reaction experiment using polystyrene block. Showing position, external force estimate, external moment estimate and collision trigger signal (top to bottom). The shaded area shows the instant of the collision [27].

Next up is the performance of the method from Dicker et al. [30]. For validation of the fuzzy logic CRI value a Monte Carlo simulation was run for a 1000 simulations in which an impact orientation was randomized with different attitude angles and velocities. The results are shown in Figure 3.30. This shows that a higher CRI value does correspond to a collision with a higher horizontal angular velocity, thus the CRI does provide an indication of the harshness of a collision. Other experiments for the total recovery system were also conducted. The control strategy for these experiments was based of work from Faessler et al. [32] (aggressive control approach for stabilizing an MAV after being thrown in the air). Measurements were taken from the onboard IMU, for attitude and altitude estimates. An external camera system was used for position and orientation, which were then used to estimate the velocity. The results for 11 trials with an collision inclination of 15 degrees is seen in Figure 3.31. It can be noticed that the recovery pipeline reacts aggressively by inducing an opposing pitch of 40 degrees, after which it is reduced to zero and stabilizes (this is due to adapting aggressive control strategy from Faessler et al.). The pipeline does however stabilize the MAV successfully each time for their experiment.

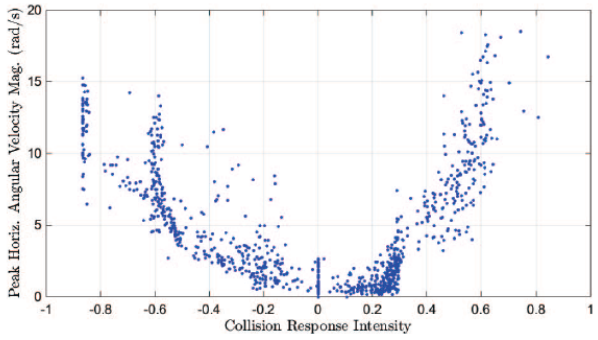


Figure 3.30: Monte Carlo simulation (1000 runs) without recovery control for validation of CRI [30].

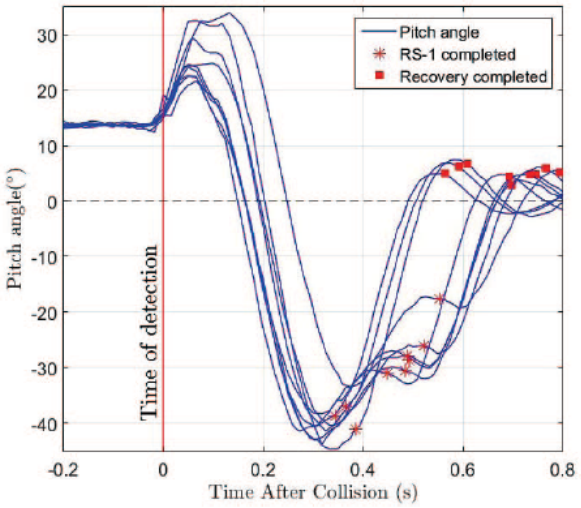


Figure 3.31: Pitch angle results for inclined collisions and recovery from flight experiment [30].

3.5.3 Discussion

Two different methods for detecting a collision have been shown. They follow the same procedure but use a different state for the threshold. One of them used the accelerometer and the other one an estimated external force. This external force can be calculated using an observer from section 3.3, which then indirectly also uses the accelerometer. Dependent on the controller design and thus if such an observer is already implemented both methods may be used, but if no observer is implemented the accelerometer method is preferred. With an observer both methods may even be used simultaneously to create a more robust detection system, this might be investigated later in the thesis work if the observer strategy is chosen.

Then for calculation of the normal plane the method from Dicker er al. [30], using the accelerometer is simpler compared to the one from Tomić and Haddadin [27]. As the information from the accelerometer could directly provide the collision planes normal as shown by Equation 3.12. This is preferred as it is computationally less expensive. However no results of this estimation were shown, compared to simulated results by Tomić and Haddadin in Figure 3.28 which showed promising results.

Lastly the method for recovering after a collision is detected. The fuzzy logic process is quite extensive and difficult to set up. As experience and expert knowledge of the system are necessary to set up the fuzzy logic rules to end up at the CRI value, which were adapted from an extensive research work by Chui [31]. As within this work collisions are tried to be avoided and the main focus of the work is achieving stable flight the stop or bounce strategy from Tomić and Haddadin [27] is preferred, as it is much simpler and faster to achieve. In combination with proximity sensors which can also provide information about where the collision plane is situated relative to the drone a control signal can be determined which drives the MAV away from the wall. Or in a situation where margins are small the MAV can just stop when collisions is detected and stabilize directly. This may however not be possible without moving a little back as contact with a wall may restrict motion in some directions. Moreover a warning signal can be provided by proximity sensors when coming close to a boundary, which may be used to adjust the control policy by restricting motion in one direction.

3.6 Sensors

During the discussion of the observers, controllers and collisions detection techniques two mayor systems parts which contain sensors are mentioned already. The first one being the IMU, which most of the time contains an accelerometer, gyroscope and pressure sensor. The second one being the external camera system used for motion tracking. For the thesis work there will be flown in small enclosed environment, probably unknown areas in which knowledge of the environment will be key. A simple way of gaining such knowledge is using a sensor that can provide relative distances with respect to the environment. This section will discuss several proximity sensors found in literature.

Nakata et al. [33] developed a sensor based on pressure differences. Their work is derived from nature, by looking at a mosquito. A mosquito can detect disturbances introduced by the environment in his self induced

flow, which makes it possible to tell if there is a boundary nearby. A small sensor was developed which is shown in Figure 3.32. The sensor is tested on a 27 gram weighing Crazyflie 2.0 which proved that the sensor is able to detect walls when setting a threshold on the pressure differential. The disadvantage of this type of sensor is the need of pressure tubes at locations where the flow is expected to change most. These tubes need to reach beyond the propellers and thus are vulnerable to collisions which might lead to damage. Other work from Yeo et al. [34] used pressure probes to detect downwash flows from other MAV's nearby, their experimental platform, also shows that the pressure tubes extend out beyond the propellers to be able to get correct measurements, being prone to braking when a collision might occur. The drone dimensions also get larger, which increases the minimum space needed for the drone to fly. Moreover to obtain a distance, data needs to be obtained mapping pressure differential to distance thus needing experiments.

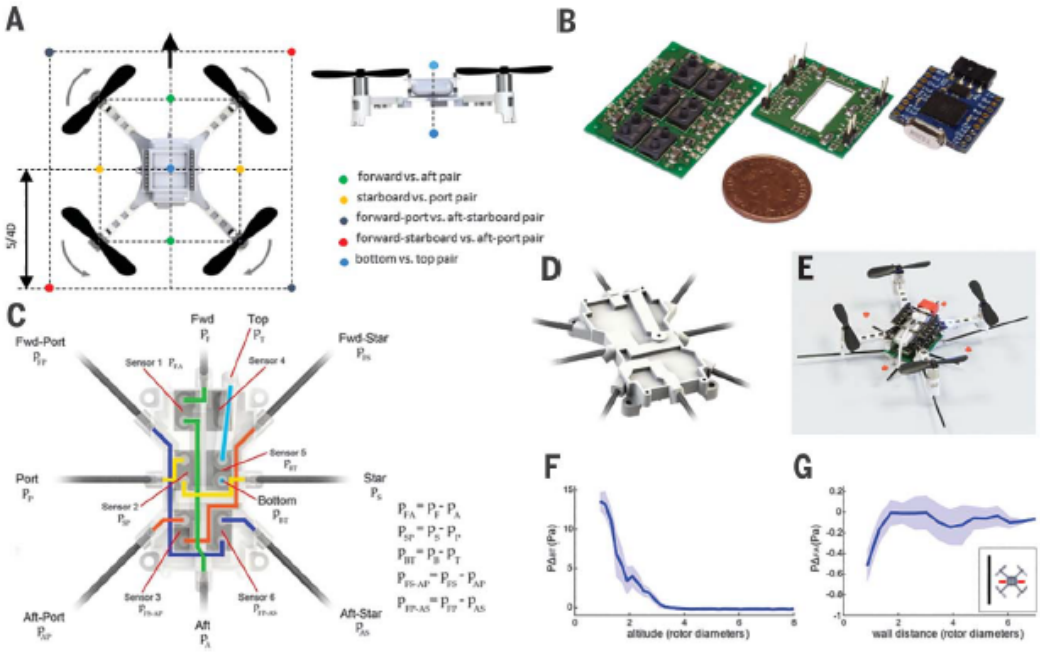


Figure 3.32: (A) shows placement of five paired pressure tubes. (B) shows the sensor components (pressure sensor array, adapted circuit board and microcontroller. (C) Schematic showing internal routing tracks for probes. (D)/(E) show installed sensor module and MAV platform. (F)/(G) show differential pressure results for ground and wall [33].

Lasers based sensors are another viable option for proximity sensing. LiDAR is one sensor option as is used by González-deSantos et al. [35]. The sensor has an accuracy of up to 1 cm and a range of up to 40 meters, but below distances of 5 meters the accuracy goes down to 2.5 centimeters which for flying in small areas may be too rough. Besides the accuracy the sensor is quite large and is not really applicable to the MAV size considered within this thesis work. A better sized laser based proximity sensor is used by Helbling et al. [36] for estimating the altitude of an insect like MAV. The sensor uses infrared light to calculated distance based on time between transmission and reception. The sensor only weighs 20 micrograms which is perfect for usage on a small MAV. Within Figure 3.33 the performance of the sensor can be seen. It shows that the sensor is able to track the altitude accurately for most of the flight, with only a small dip around two seconds is not detected. This might be caused by a part of the MAV blocking the sensor.

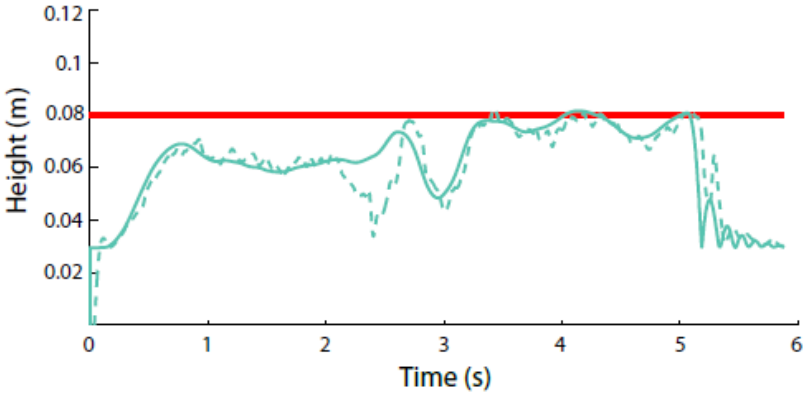


Figure 3.33: Performance of the laser sensor compared to an external camera positioning system. Red line is the wanted hovering altitude of the MAV, dashed line is the altitude by the external camera system and the continuous blue line is the measurement from the laser sensor [36].

Nohara et al. [37] implemented four of those small laser sensors onto a MAV. Three facing outwards in the horizontal plane and one facing downwards. From experiments it was concluded that the sensors provide data without a lot of noise. This may have been caused by the sensors being mounted on a Styrofoam block that encloses the MAV, and thus vibrations introduced by propellers not being of great affect. A disadvantage of such a laser sensor is the need of a reflective surface for the light. A performance study was performed by Adarsh et al. [38] testing an infrared sensor on different materials. The results from this test are shown in Figure 3.34. The standard deviation and correlation coefficient (r) are shown. The correlation describes the accuracy of the sensor, the closer the value is to one the closer the measurement was to the actual distance.

Material	Ultrasonic Sensor		Infrared Sensor		(r) (US-IR)
	Std. Dev	*r*	*Std. Dev*	*r*	
Cardboard	9.3	0.9879	10.6	0.91089	0.88424
Paper Sheet	37	0.2611	20.2	0.97866	0.20663
Sponge	5.8	0.9868	21.6	0.78774	0.72976
Wood	10	0.9999	36.5	-0.3291	-0.3285
Plastic	4.3	0.9995	25.1	0.78681	0.78767
Rubber	4.6	0.9988	58.3	0.90998	0.9216
Tile	11	0.9952	23.8	0.73032	0.73085

Figure 3.34: Performance of infrared and ultrasonic sensor on different materials [38].

It can be seen that the performance of the infrared sensor is relatively sensitive to material compared to an ultrasonic sensor. Wood is a material for which the infrared sensor is not applicable as the rough surface causes the laser to be deflected and not reach the sensor leading to wrong data.

A better option may be a combination with ultrasonic senors. An ultrasonic sensor uses sound wave propagation instead of light to estimate distances. Gageik et al. [39] developed an obstacle detection system using both infrared and ultrasonic sensors. In total 16 infrared and 12 ultrasonic sensors were used to create a redundant system. With two different types of infrared sensors (long and short range) used to enhance accuracy. By applying a data fusion technique the sensors readings are combined to obtain a distance estimate.

Dependent on the controller design and thus needed measurements, different sensors could be considered. Work in literature has shown that small sensors can provide reliable information. During the design one should consider the robustness against different environmental materials as laser sensor has shown to be sensitive to that. Moreover when using multiple ultrasonic sensors, one sensor might catch the sound wave send by another sensor which may lead to wrong measurements. Thus either sensors must be placed far enough part or the controller should be able to adjust which sensors are on and off such that this is avoided.

4 Experiments

Besides an investigation into literature to find out what is already done, two types of experiments have been conducted. First of all section 4.1 will use CFD software to investigate flow circulation. As all experiments within section 3.1 were conducted in relatively open spaces, circulation of the flow which might also create disturbances were not considered. After that section 4.2 will go into a Matlab Simulink simulation to obtain more experience and information on the use of a PID controller.

4.1 CFD

An important aspect not considered within the literature shown in chapter 3 is the recirculation of the airflow within an enclosed area. To investigate this effect Ansys Fluent will be used, which is a computational fluent dynamics simulator. This is a program in which environments can be generated via 3D modeling. Then flow conditions can be set, such as boundary conditions, e.g. telling that certain parts are walls. It will then simulate the flow and provide an indication on speed and direction. This will provide an insight in what type of flow patterns will be created when flying indoors.

4.1.1 Setup

The simulations conducted are representing stationary flight of an actuator disk (representing flow generated by an MAV). The actuator disk used has a diameter of 20 centimeters and generates a flow that will be able to lift a 500 gram MAV. This is representative of the weight and size of Parrot Bepop for example used by Smeur et al. [29]. The flow is generated by setting a momentum source on the disk volume. This will accelerate the flow such that it acts with constant force on the surroundings. The force is equal to the MAV's mass. The simulation will thus represent hover flight conditions. For the disk of diameter 20 centimeters and height of 2.5 centimeters, this equals a momentum source flow of approximately 6250 $kgm^{-2}s^{-2}$.

Table 4.1: CFD mesh parameters.

Mesh Setup	**Walls**	**Disk**
Maximum Size	0.1	0.01
Height	0.1	0.01
Minimum size limit	0.05	0.005

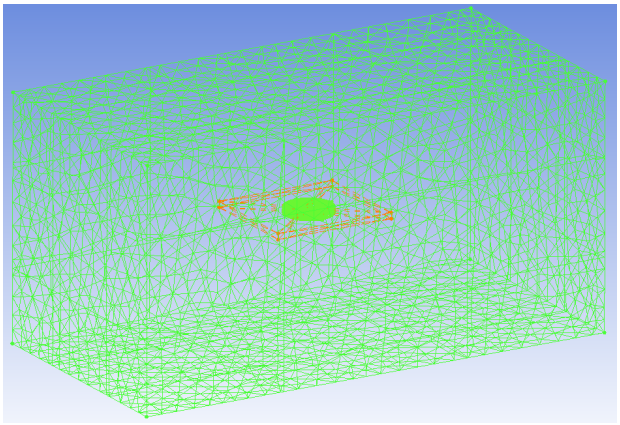


Figure 4.1: CFD simulation environment setup example.

Table 4.1 shows the parameters used to generate the computational mesh. Besides the general mesh set on both the environment and disk area a density mesh is generated for the disk. The parameters for the refined mess are: size 0.2 and ratio 1.5. This means the denser mesh starts at a size five times smaller and every neighboring cell has an increase in mesh size of 1.5. Figure 4.1 shows the setup of the environment for a case where the disk is placed in the middle of the area. The area is a box with dimensions 2 x 1 x 1 meters (l x w x h).

4.1.2 Results

Before some of the results are shown it is important to know the limits of the simulation. First of all the momentum source always generates the same amount of total force onto the flow no matter the location. This means that propeller efficiency is not taken into account. Meaning a more efficient propeller near the ceiling which is a cause of the ceiling effect, generating more thrust is not simulated. Secondly no lateral rotation (circulation caused by rotating propeller) is introduced within the disk, but it is assumed that the overall flow pattern is presumably similar to a real life situation where the rotation does exist. Thirdly as only one disk is used compared to four on a quadcopter MAV no flow interaction between different propellers is simulated, once again simplifying the flow. Lastly no MAV body is created which might disturb the flow. This also means that no forces acting on the MAV can be obtained from the simulation. Even though there are quiet some limits and simplifications on the simulation the general flow-field is obtained which provides more insight in how severe disturbances might be when flying.

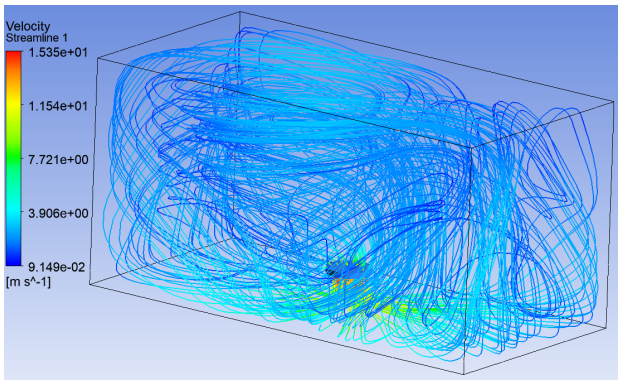


Figure 4.2: CFD simulation middle of the room, 15 centimeter away from ground.

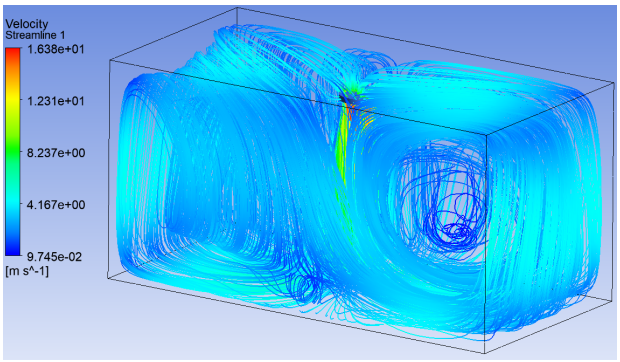


Figure 4.3: CFD simulation middle of the room, 10 centimeters away from ceiling.

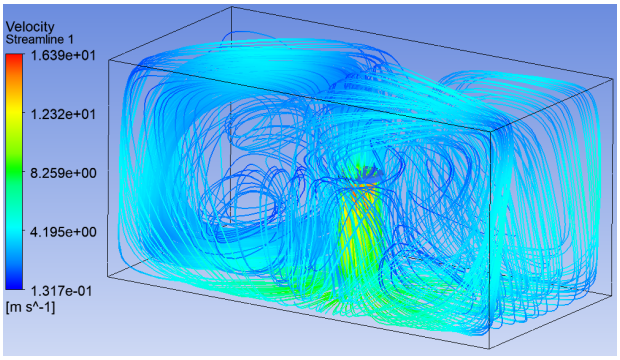


Figure 4.4: CFD simulation middle of the room.

The first case that will be discussed is when the disk is placed in the middle of the room, either located near the ceiling, ground or in between. The results can be seen in Figure 4.2, Figure 4.3 and Figure 4.4. The case with the disk close to the ground does show a different flow pattern, being less symmetric and showing a lateral circulation compared to the other two. The middle and upper case show a more symmetric pattern, with a longitudinal circulation left and right of the MAV. Different from the middle case the upper case also has flow from left to right, instead of down from MAV along the ground to the wall, back up and then via the ceiling towards the disk. Looking to the flow velocity all three cases show similar speeds. The flow directly from the MAV starts around 16 m/s and slows down to around 4 m/s when hitting the ground. Thus the main disturbance flow has a quarter of the velocity of what the MAV produces, which is still quiet significant and is expected to disturb the MAV.

Secondly the MAV was placed close to the wall, once again near the ground, ceiling and in between. Figure 4.5, Figure 4.6 and Figure 4.7 show the results respectively. Completely different flow patterns arise compared to the previous cases. This time the lower and upper case show similar symmetric flow patterns. Two longitudinal flow patterns towards the other side of the room. The flow velocity is once again close to 4 m/s when near the MAV but approximately half way to the other side of the room it decreases to around zero, with the upper case having a lower velocity gradient than the lower one. For the hover in between ground and ceiling Figure 4.7 a more irregular pattern can be seen. The most significant feature is an upwards flow coming from ground along the right front corner and than moving towards the back.

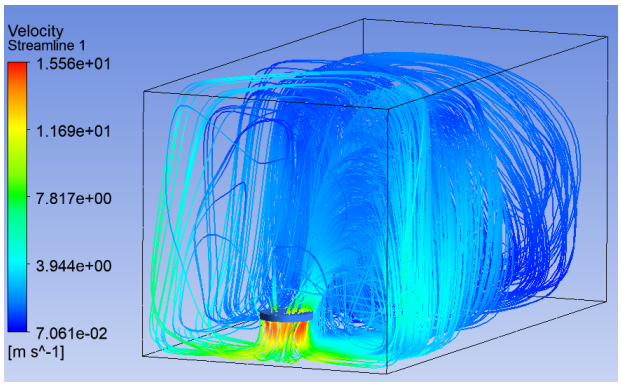


Figure 4.5: CFD simulation near a wall, 10 centimeters away from wall and 15 centimeters from ground.

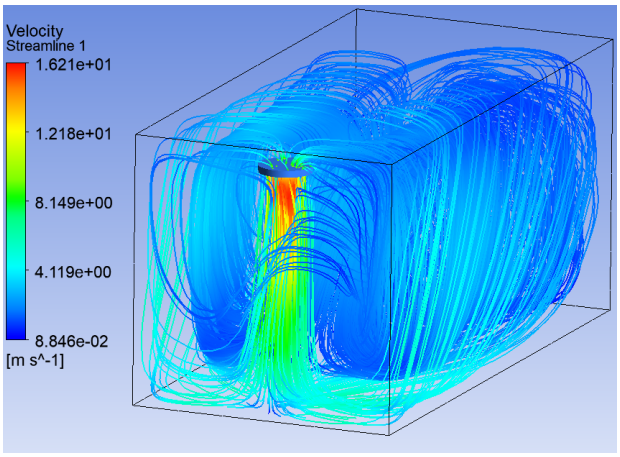


Figure 4.6: CFD simulation near a wall, 10 centimeters away from wall and 10 centimeters from ceiling.

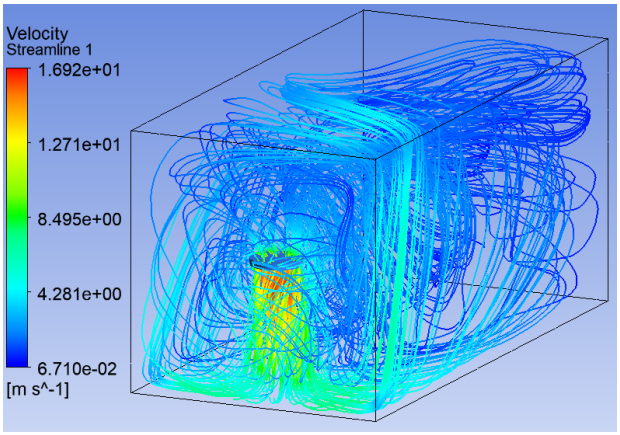


Figure 4.7: CFD simulation near a wall, 10 centimeters away from wall and in the middle between ground and ceiling.

Then flight in a corner of the room was simulated. Once more near ground, ceiling and in between. The simulation results are seen in Figure 4.8, Figure 4.9 and Figure 4.10. This time all three have a comparable pattern that is created, first the flow moves towards the back of the room along the ceiling and then creates a single circulating pattern moving along all four sides back towards the MAV. The magnitude of this circulation seems to depend on the distance from the ground, as the farther the MAV is placed from the ground the stronger the circulation becomes. Once again the average flow velocity of the disturbance is around 4 m/s. Figure 4.9 shows another interesting phenomena, namely the flow seems to be attracted towards the wall after being accelerated downwards. This is also visible in Figure 4.10. This might imply that the entire MAV may be attracted towards the wall.

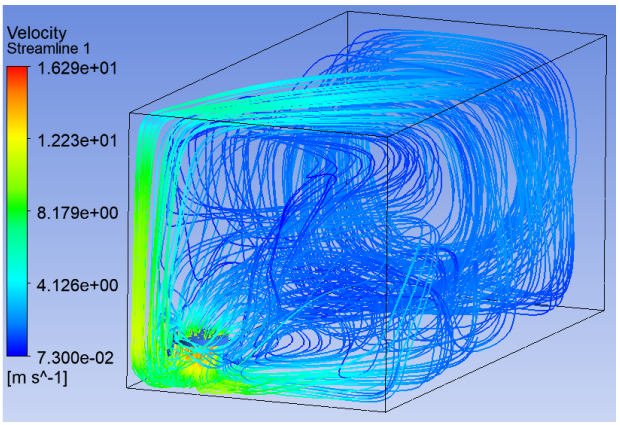


Figure 4.8: CFD simulation in corner of the room, 10 centimeters away from both walls and 15 centimeters away from ground.

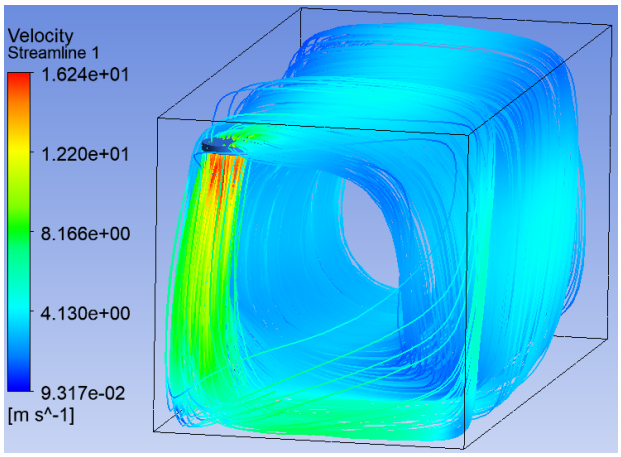


Figure 4.9: CFD simulation in corner of the room, 10 centimeters away from both walls and 10 centimeters away from ceiling.

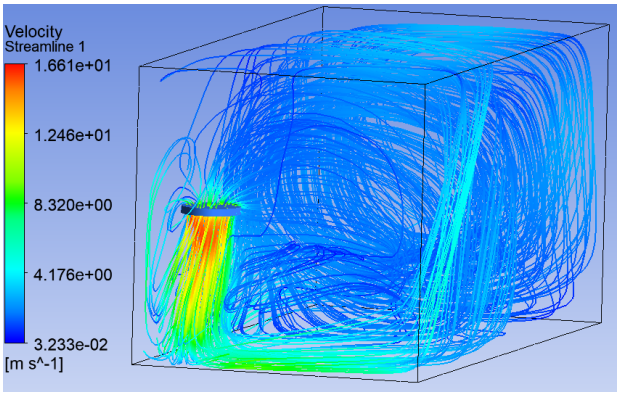


Figure 4.10: CFD simulation in corner of the room, 10 centimeters away from both walls and in the middle between ground and ceiling.

4.1.3 Discussion

Each of the cases has shown different flow patterns that arise. Overall the main velocity of the disturbance flow is around 4 m/s, being a quarter of the velocity magnitude directly under the MAV. From the results it could be seen that each place within the room creates different flow patterns, with some being similar to others, but also quiet some differences. It provided great insight in what can be expected when flying indoors. For example when one has flown close to a corner and needs to move to another space it might be easiest to move along the middle of the room in between ceiling and ground as the main disturbance flow circulates around the walls. This information can be used during flight experiments as it is now known where the largest flow circulation disturbances exist, such that the controller needs to be able to either deal with those or avoid them.

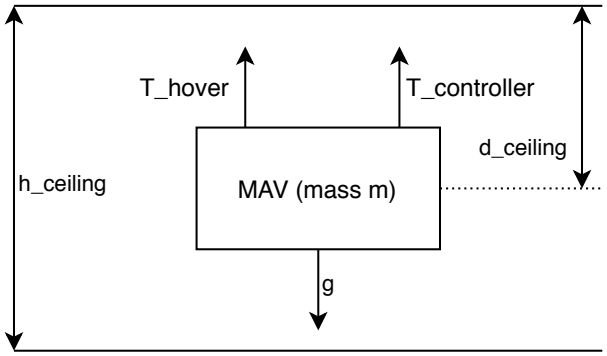
4.2 Matlab

To gain some more insight in the workings of a PID controller a simple Matlab simulation was created using the Simulink toolbox. Within the simulation both continuous time and discrete time controllers were created and tested, such that the effects and difficulties that might arise could be seen. The simulation will be a simple mass that can be accelerated using a thrust force. It will be enclosed by a ground and a ceiling and is constrained to move in one direction (up and down). Furthermore a simple model for the ceiling effect is implemented such that it can be investigated if a PID controller can be tuned for non collision performance.

4.2.1 Setup

Before results can be shown the simulation setup will be explained and the final controller forms will be shown. A situational sketch is provided in Figure 4.11. The main equation determining the acceleration on the MAV

shown in Equation 4.1, which is based on Newtons second law.



$$\begin{aligned} F &= m * a \\ R_{ceiling} * (T_{controller} + T_{hover}) - mg &= m * a \end{aligned} \tag{4.1}$$

Figure 4.11: Situation sketch implemented within Matlab, distances and forces shown.

F are the forces acting on the MAV, which consist of T_{hover} a fixed amount of thrust that equals the weight of the MAV, $T_{controller}$ a changing amount of thrust provided by the controller and the weight of the vehicle acting in opposing direction. Then m is the mass of the MAV, g the gravitational constant and a the acceleration. Lastly $R_{ceiling}$ is a ratio that indicates how much the ceiling effects the thrust value. Within the simulation the ceiling effect model from Cheeseman and Bennet [8] will be used described by Equation 3.1. This is a simple equation which might not exactly describe the effect but can be seen as a well enough estimate. For the purpose of the Matlab simulation this mismatch with reality is not a problem as the dynamics of the MAV are also highly simplified.

Within the simulation two main control strategies will be evaluated. The first one being a position feedback PID controller shown in Figure 4.12. The continuous time simulation diagram is shown and it has the following elements. First of all the MAV starts on the ground and is commanded to a certain height (h_command). Then the altitude error is calculated which is the input for the PID controller. The PID controller provides the extra thrust besides the fixed hover thrust which then goes into the actuator dynamics to create some delay in the system. A corner frequency of 50 radians per second was chosen for those dynamics. Then using Equation 4.1 the acceleration acting on the MAV is calculated considering the ceiling effect after which a double integration is performed to obtain the altitude position, which is then fed back. The output of the PID is bounded to be between $(-10 \wedge 10)$ newton, as a motor also has a maximum and cannot provide infinite thrust. As the MAV simulated is chosen to be 500 grams the PID limits are twice the weight of the drone, thus including hover thrust a maximum acceleration of approximately 3 g can be reached. Moreover the second integrator is also bounded to not output position below zero (ground) and above the ceiling height.

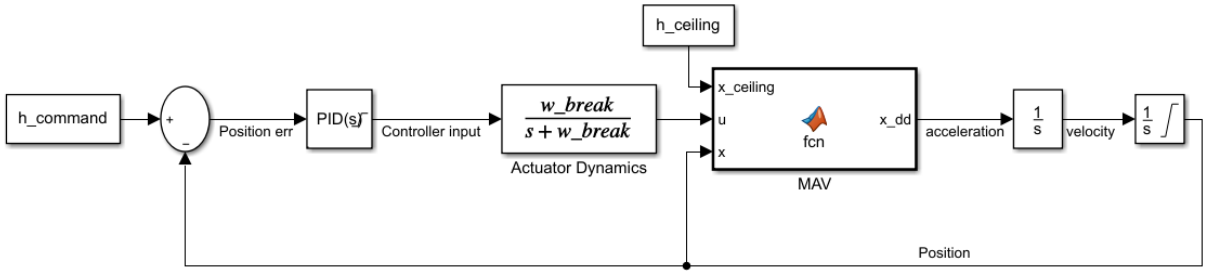


Figure 4.12: PID position feedback controller, shown for continuous time simulation.

The other controller scheme can be seen in Figure 4.13 and as can be seen has an extra feedforward path. This path provides an estimate of the ceiling effect based on Equation 3.1 and the altitude. This estimated thrust is then subtracted from the PID controller output to compensate. This compensations makes it that the PID controller needs less effort to control the system as part of it is done by the feedforward compensator. The information it needs is the thrust at that moment in time and the distance to the ceiling.

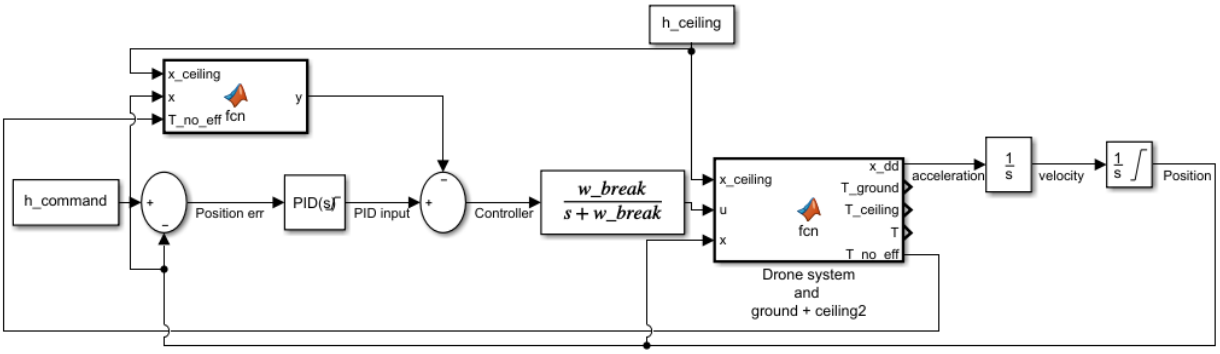


Figure 4.13: PID position feedback controller with an extra feedforward path, shown for continuous time simulation.

4.2.2 Results

Once more before going into the results the limitations are discussed. First of all the simulation is only one dimensional, ignoring the normal six degrees of freedom underactuated drone dynamics. Secondly no sensor noise is considered, thus having a perfect measurement of the altitude. Thirdly the ceiling effect equation used has a singularity and shoots to infinity when getting to zero distance, to avoid this the maximum effect is fixed to a ratio of 1.6, which from the results within section 3.1 seems reasonable. Finally to introduce some delay into the system the actuators are modeled as a first order system, this simulates some spin up/down time for the motors. For the simulation two different ceiling heights were chosen, one being 20 centimeters and another one being 50 centimeters. The MAV is commanded from the ground towards a certain flight altitude after which it needs to hover at the commanded altitude. The simulated MAV is assumed to have rotors with a radius of 5 centimeters, comparable to the Parrot Bebop. The simulated ceiling effects can be seen in Figure 4.14 and Figure 4.15, which shows the maximum being 1.6. z is the distance towards the ceiling (reverse to altitude).

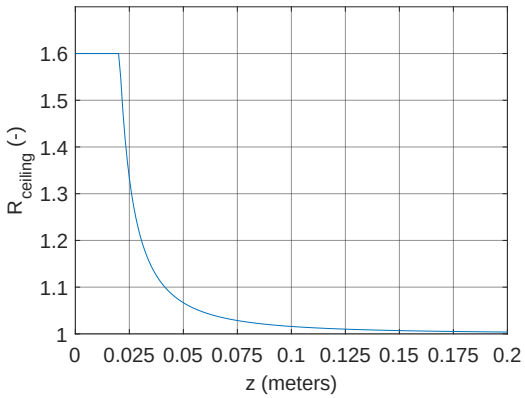


Figure 4.14: Simulated ceiling effect using Equation 3.1, with a ceiling at 20 centimeters.

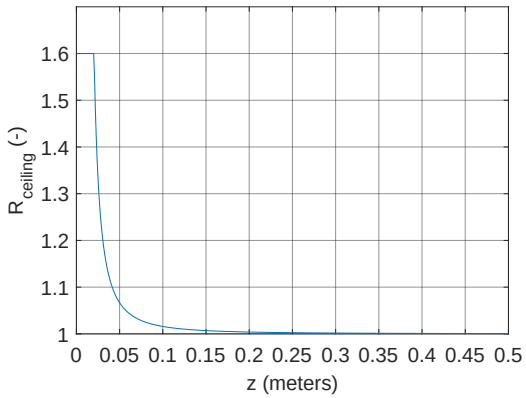


Figure 4.15: Simulated ceiling effect using Equation 3.1, with a ceiling at 50 centimeters.

First off all the simulation will be run in continuous time (this is a feature in Simulink that tries to mimic continuous time, as it is still run on a computer and technically discrete time). The PID gains have been manually tuned for eight cases, four with a ceiling at 50 centimeters and four with a ceiling at 20 centimeters. The gains have been tuned for a quick response with as little overshoot as possible and no steady state error. The feedback controller gains for the eight cases can be seen in Table 4.2 and Table 4.3.

Table 4.2: PID feedback controller, gains for different hovering altitudes, with ceiling at 50 centimeters.

H_command [m]	P	I	D	Collision
0.4	35	5	8	no
0.45	150	15	25	no
0.48	200	15	35	no
0.49	275	15	20	yes

Table 4.3: PID feedback controller, gains for different hovering altitudes, with ceiling at 20 centimeters.

H_command [m]	P	I	D	Collision
0.1	10	2	5	no
0.15	45	5	10	no
0.18	150	15	25	no
0.19	250	20	25	yes

The feedback entering the PID controller is the altitude error with respect to the set hover altitude, being in the order of $0 \leftrightarrow 0.5$. The proportional gains that are needed for stable non colliding performance are high, with in both cases the situation in which an 1 centimeter distance from the ceiling is wanted a collision was unavoidable. These high gains might pose a problem on a real system where noise exists, as the proportional gain will also amplify the noise with that magnitude. A higher P gain is equal to a more aggressive control, faster response, which is needed when getting closer to the ceiling where the effect gets higher in magnitude and thus a higher opposing control reaction is needed. As said the two most extreme cases could not be tuned without a collision. The MAV would hit the ceiling for a certain amount of time, but after a while the integral gain managed to let the MAV decent a little but keeping an error and thus not reaching its final position. As the simulation is highly simplified compared to the actual ceiling effect, the behavior after colliding with the ceiling is expected to differ to much from the simulated case. Between the two ceiling heights a trend of smaller gains in the lower ceiling case is seen. As the drone always starts from the ground it needs to travel less, reaching a lower velocity when braking needs to start requiring a less aggressive controller and thus lower gains.

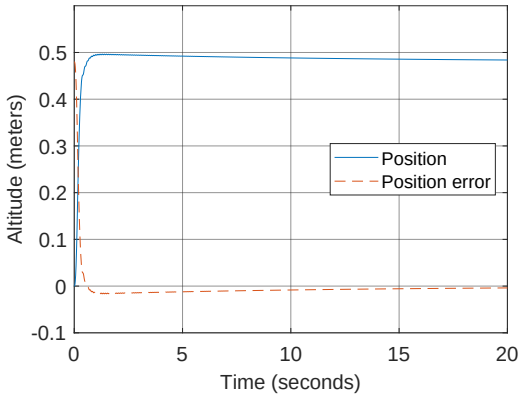


Figure 4.16: Continuous time simulation PID feedback altitude response, for hovering command up to 48 cm with ceiling altitude being 50 cm.

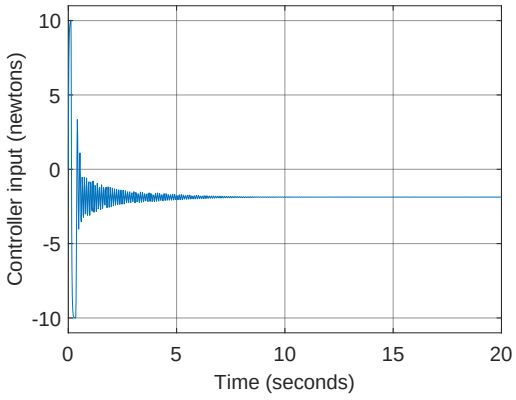


Figure 4.17: Continuous time simulation PID feedback controller input for system, for hovering command up to 48 cm with ceiling altitude being 50 cm.

For both ceiling heights two altitude and controller responses are plotted for the case of hovering two centimeters below the ceiling. This is chosen as this was the most extreme case without a collision, meaning controller needing to handle the highest amount of disturbance. The case of hovering at 48 centimeters can be seen in Figure 4.16 and Figure 4.17. The altitude response shows a little overshoot after which it slowly converges to a zero steady state error. The controller however has a high oscillatory response which damps to one value. This damping is caused by the integral gain which slowly converges the control input to the exact value needed for equilibrium flight at that altitude. This fast oscillatory behavior might however not be possible with a real MAV as the motors need time to adjust. Adjusting the gains to reach a better more stable response was not possible, as adjusting the integral gain let to more overshoot (higher) or even slower convergence (lower), the derivative gain either caused a collision (lower) or even more oscillations in the position response (higher). Adjusting the proportional gain also let to collisions.

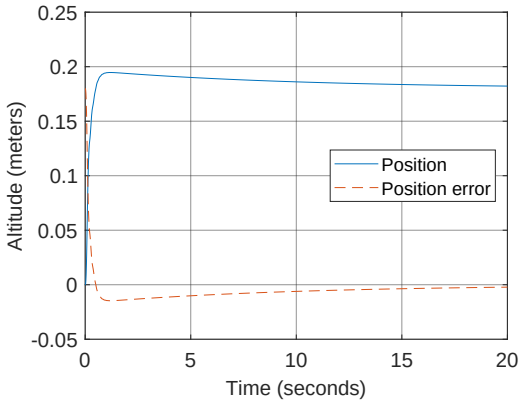


Figure 4.18: Continuous time simulation PID feedback altitude response, for hovering command up to 18 cm with ceiling altitude being 20 cm.

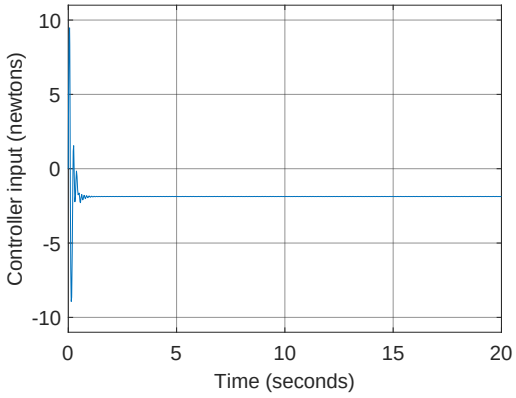


Figure 4.19: Continuous time simulation PID feedback controller input for system, for hovering command up to 18 cm with ceiling altitude being 20 cm.

Figure 4.18 and Figure 4.19 show the results for the ceiling altitude of 20 centimeters and 2 centimeter hover distance. The position response is comparable with little overshoot and slow convergence, but the controller response shows less oscillations. This is probably caused by needing to travel less distance from the ground and the somewhat lower proportional gain. This means reaching a lower velocity before needing to slow down, doing this less aggressive causes less overshoot and thus needing less corrections. This lower velocity implies a lower thrust when reaching the area where the ceiling effect exists thus having a smaller effect, however the final ceiling effect is the same as similar hover distance towards the ceiling is commanded.

The second controller implemented is the feedforward PID controller, for which the tuned gains can be seen in Table 4.4 and Table 4.5.

Table 4.4: PID feedforward controller, gains for different hovering altitudes, with ceiling at 50 centimeters.

H_command [m]	P	I	D	Collision
0.4	0.25	0	1	no
0.45	0.5	0	2	no
0.48	2	0	2	no
0.49	2	0	2	no

Table 4.5: PID feedforward controller, gains for different hovering altitudes, with ceiling at 20 centimeters.

H_command [m]	P	I	D	Collision
0.1	1	0	1	no
0.15	1	0	1.5	no
0.18	1	0	2	no
0.19	1	0	2	no

The gains are significantly lower compared to the pure feedback controller. The feedforward compensator has a perfect model of the ceiling effect, similar to the one used in the dynamics of the system. This means that the calculated ceiling effect which is subtracted from the PID controller command perfectly matches the encountered effect. Thus the PID controller only needs to command the system towards the hover altitude and not consider the ceiling effect at all. The integral gain is even zero, this normally drives the system to have zero steady state error, but the compensator already takes care of this. Moreover this time all cases could be tuned and no contact with the ceiling occurred. Once more a similar trend of having lower gains with the case of lower ceiling is seen. For this controller the responses of the same case as before are shown. Figure 4.20 and Figure 4.21 are the feedforward controller, altitude and control input responses for the 50 centimeter ceiling case.

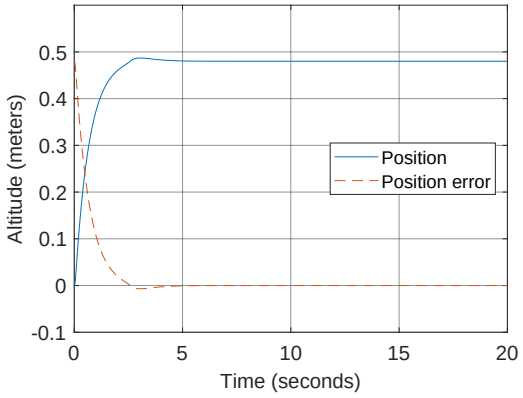


Figure 4.20: Continuous time simulation PID feedforward altitude response, for hovering command up to 48 cm with ceiling altitude being 50 cm.

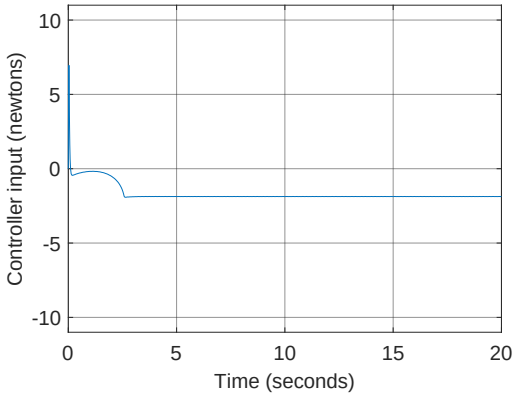


Figure 4.21: Continuous time simulation PID feedforward controller input for system, for hovering command up to 48 cm with ceiling altitude being 50 cm.

For the altitude response once again a small overshoot is seen, this time with a more defined point. The controller input response has no high oscillatory pattern anymore. Within this response the first change almost immediately after the start of the simulation is the point where the ceiling effect starts and thus the compensator does output a value. The second drastic change is located at around the highest point within the altitude response, this is most probably caused by the fixed maximum of the ceiling effect. As around that distance towards the ceiling the ceiling effect does not change anymore as seen in Figure 4.15, thus the compensator output being constant. The responses for the ceiling altitude case of 20 centimeters are shown in Figure 4.22 and Figure 4.23. It shows similar behavior. This time the lower ceiling does not influence the controller behavior as much as was seen before with the 50 centimeter case.

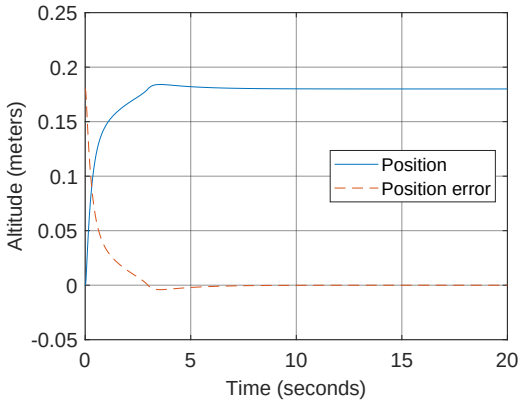


Figure 4.22: Continuous time simulation PID feedforward altitude response, for hovering command up to 18 cm with ceiling altitude being 20 cm.

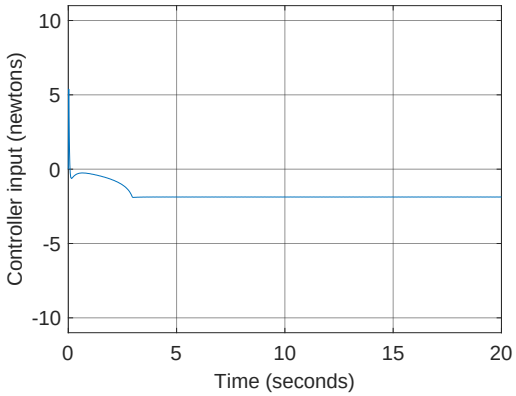


Figure 4.23: Continuous time simulation PID feedforward controller input for system, for hovering command up to 18 cm with ceiling altitude being 20 cm.

All the previous results were obtained by simulating in continuous time, discrete time simulations were also performed. To change the system into discrete time the following adaptations were made. First of all the PID block was set to discrete time. Then each signal that enters the dynamics of the system must be discretized. This means for the feedback controller that within the feedback line for position a zero order hold block (ZOH) was placed, this makes the signal value become constant in between measurement periods (discrete time step). For the feedforward controller a ZOH block was also placed in the total thrust feedback (T_no_eff) and output of the feedforward function.

The results for the discrete time simulations were similar to the ones in continuous time for discrete time steps

below 0.02 seconds (50 Hertz). When increasing the time step to 0.02 seconds or higher the system becomes untunable for avoiding a collision, even for hover distances where the ceiling effect is very low.

4.2.3 Discussion

As discussed in subsection 3.2.1 a PID controller is tuned for one equilibrium point. This is confirmed by the simulation as it was highly dependent on the hover altitude what the gains for optimal performance were. The gains used in hover cases close to the ceiling could also provide a non colliding response for lower hover altitude cases. This could mean for a solution that one only needs to tune for the worst case scenario expected, as it might be able to deal with the less extreme situations using those gains. Within section 3.4 a PID controller with feedforward strategy was shown which already showed promising results. Within the simulation results it could be seen that pure PID control is able to handle the disturbances most of the time but significantly different proportional gains were needed. When the feedforward compensator was added the gains became almost similar for the different conditions. This is because the PID controller only needed to command the system to a certain position and the feedforward path handled the disturbances. The feedforward controller however has a perfect model of the ceiling effect, which in real life would not be the case as there will exist modeling errors. This means that part of the compensation needs to be handled by the PID controller, which as shown had more difficulty with that, thus causing some degradation in performance.

Noise is also an important factor within the performance of a controller which was neglected during these simulations. As noise deviates a signal from its true value the error it carries will be amplified by the PID controller and negatively affect the performance. Especially the performance of the feedforward controller will be affected, as without noise it perfectly compensated for the ceiling effect due to the perfect model, but when there will be noise in the altitude signal it will compensate for the measured position and not the actual position.

Another important aspect shown within the simulation is the performance breakdown that happened during the discrete time simulation. As with all computers and thus electronic systems they work in discrete time, meaning that data is only known at certain moments in time. This discrete time step which influences that real time performance is influenced by two major factors. Namely frequency at which sensors can take measurements and the calculation time needed to obtain the controller results. Other effects such as motor spin up also play a role in real time performance. This is thus an important concept that needs to be taken into account when choosing a solution for the thesis work, as too large of a discrete time step may create an controller that cannot stabilize the system.

5 Final Research Plan

As the literature has been discussed and with some preliminary experiments conducted the main research plan can be developed. To clarify the objective of the thesis work, it is once more stated.

> "To achieve autonomous stable flight of an MAV quadcopter within a small enclosed area by means of developing a controller that is able to cope with the aerodynamic effects, self induced flow disturbances and mild collisions."

A small area is defined as follows. An area for which the dimensions of the drone are within ten times the maximum dimensions of the room (e.g. a 10 centimeter MAV within a 50 centimeter sewage pipe)

Within literature seen in section 3.4 the PID controller is widely used. However from the highly simplified Matlab experiment discussed in section 4.2, several disadvantages were seen. One showing the need for high gains, which could highly affect real time performance, as noise would also be amplified by the high proportional gains. Moreover the integral gain, which makes the controller converge to the commanded hover altitude showed to reduce the error quite slowly, and could not be changed to improve the convergence rate, as oscillation would be introduced within the response. The INDI controller shown by Smeur et al. [29] could be used instead of the PID controller. This incremental method may improve the performance and improve on the convergence rate of the integral gain.

To counteract the disturbances the two main solutions found in literature will be combined. The first solution was using an aerodynamic model, which based on distance to the boundary plane tells how much extra thrust the propellers produce. The other solution was looking at the difference between what is measured and what a mathematical model of the MAV tells, leading to an estimation of the disturbances. The main proposal for the controller will be the following. At the beginning of the flight the controller will estimate disturbances via aerodynamic models, for example the ground effect via Equation 3.4. As these models are not perfect and depend on MAV geometry the controller will be extended with an adaptive part. This adaptive part will be based off the work from Wei et al. [16] which used the MRAC method to create a mathematical model describing the ground effect disturbance via a collection of measurements and comparing it to a base model (without the ground effect disturbances). This adaptive algorithm will improve the disturbance estimation over time. This adaptation will thus try to decrease the model error (for which the model describes the aerodynamic aerodynamic disturbance). As already mentioned it will be combined with an INDI controller instead of a PID. Combining the INDI and MRAC is not a trivial task. A possible solution for combining the two is by including the MRAC, which describes the aerodynamic effects, within the system dynamics. Afterwards the system dynamics can be derived to the form of the INDI controller, resulting in a possible enhanced controller.

The controller may already be able to deal with mild collisions, which will be investigated. If the collision handling is not up to wanted performance an adaptation needs to be made. The adaptation will than be the following. For collision detection the accelerometer based method from Tomić and Haddadin [27] will be used. This will detect collisions based on a threshold. For the response two of the reaction strategies will be used, stop and or move back. As the main focus of the thesis work will be stable and safe flight and collisions are best to be avoided this simpler strategy was chosen over the fuzzy logic one.

The controller will need measurements of distances to its surroundings to be able to fit the disturbance estimation to certain distances from boundaries in the flight environment. This means sensors such as laser or ultrasonic sensors are needed. As with all hardware the sensors have their limitations. Those limitation such as minimum measurement distances need to be taken into account, as those will constrain the flight envelope for the controller.

For the implementation of the controller the Crazyflie[1] platform will be used. This is a small lightweight quadcopter MAV. It is relatively simple to modify with pre-existing expansion decks. Flight tests will be performed, for testing the final controller but also for testing parts of the controller during development. To protect the MAV from damage a form of propeller protection is needed during development phase, it may not be needed for the final work when the controller can guarantee that no collisions will occur. The Cyberzoo at the faculty of Aerospace Engineering at the University of Technology Delft will be used as test site. Within the Cyberzoo the position can be tracked using OptiTrack system. This will be an useful tool when testing parts of the controller, as an method that obtains the position from onboard measurements may not be implemented directly. As the goal is to fly within small enclosed areas, either an existing wingbox will be used or an environment will be created from for example wooden plates. Moreover a single wooden plane can be used to test for ceiling effect only. To command the drone an external laptop will stay in communication via a radio transmitter. This can be for example be used to kill the drone in an emergency or switch between different modes.

[1] `https://store.bitcraze.io/products/crazyflie-2-1`

For evaluation of the performance there will be looked at several aspects. First of all the run time, as too low of a refresh rate may introduce sluggish behavior and thus slow response time, which may lead to a crash. Secondly the closest distance to environmental boundaries at which the MAV will be able to hover. The hover must be stable, this implies no severe oscillations and position changes. Thirdly the ability to perform stable flight, meaning the ability to follow a commanded path. Lastly the ability of handling collisions. The performance analysis is up for change during the development.

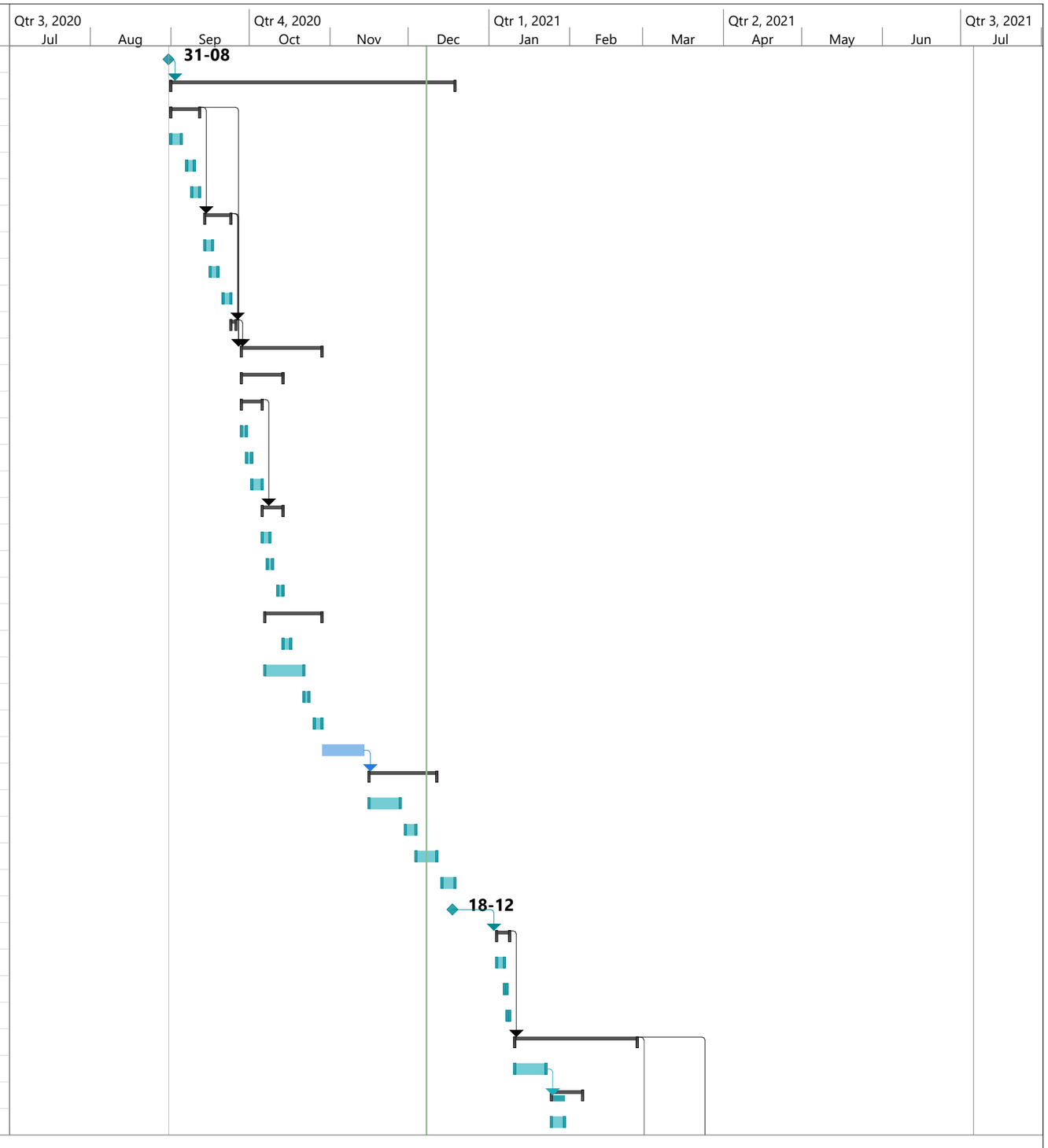
To summarize the solution the following research question arose and will be answered during the thesis work.

Can an autonomous MAV controller, which is a combination of MRAC and INDI, provide stable flight within an enclosed area where the dimensions are a maximum of ten times larger than the MAV dimensions?

1. How will the adaptive MRAC controller part be structured?
    - What initial starting conditions will be used for the MRAC model?
    - Will the basis function for the MRAC be linear or nonlinear?
    - Will the MRAC be updated every time step?
2. How will the MRAC and INDI be integrated together?
    - Where in the INDI control loop will the MRAC be integrated?
3. What is the performance of the final controller when flying in a small enclosed area?
    - What is the performance increase of MRAC+INDI compared to an INDI controller when flying in a small enclosed area?
    - What is the performance increase of the MRAC+INDI controller compared to a PID controller when flying in a small enclosed area?
    - Will the MRAC+INDI controller be able to deal with light collisions?

Finally the research planning is represented in a Gantt Chart. Within the planning there are a few important milestones. These milestones are the following. First of all the literature study presented in this work. Afterwards the development of the controller with at a certain point the midterm review. This review will be used to discuss the progress and see if all problems are tackled or if there are still some points that need extra attention. Before the thesis can be ended a greenlight review will provide permission to finish the work with a final thesis defense.

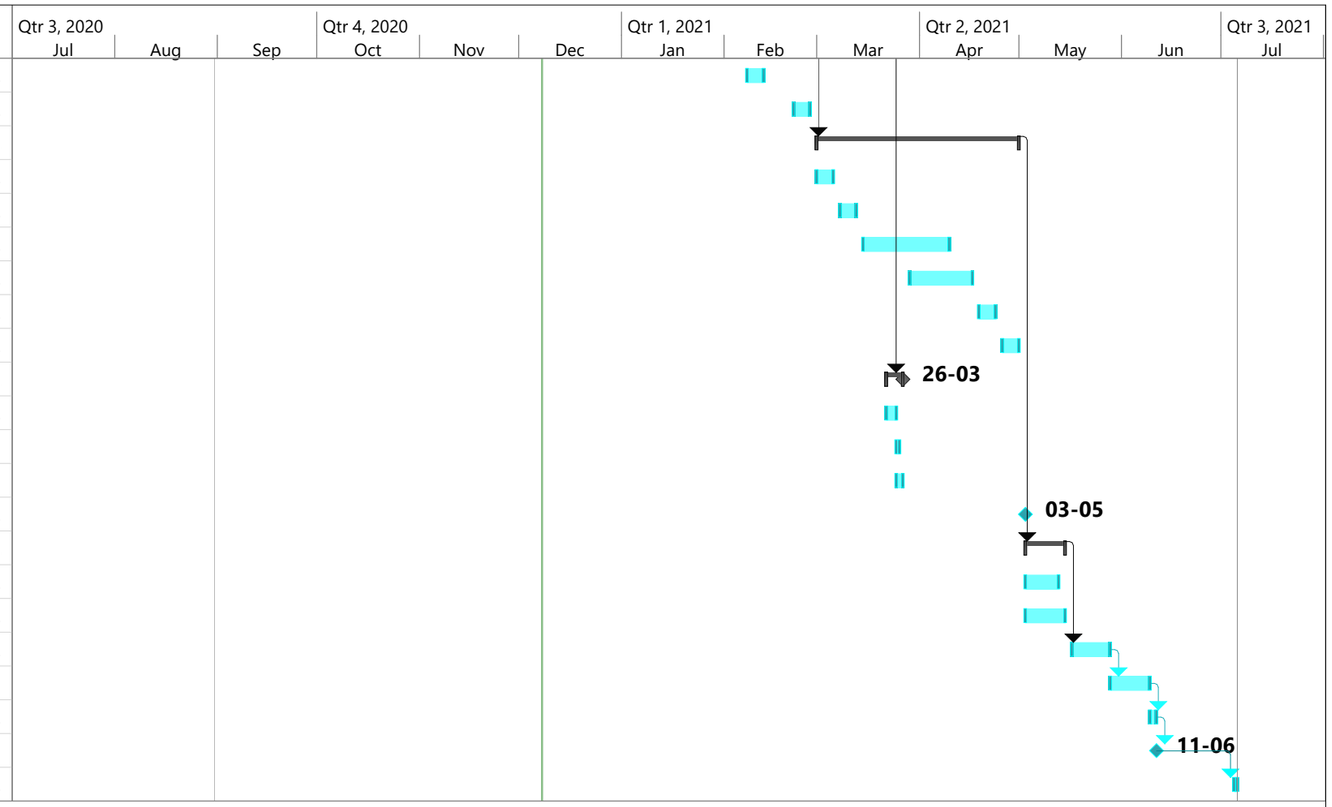
ID	Task Mode	Task Name	Duration	Start
1		1 Kick-off Meeting	0 days	Mon 31-08-20
2		**2 Literature study**	**78.88 days**	**Tue 01-09-20**
3		**2.1 Survey aero effects**	**8.88 days**	**Tue 01-09-20**
4		2.1.1 What effects exist	3.88 days	Tue 01-09-20
5		2.1.2 What models exist	2.88 days	Mon 07-09-20
6		2.1.3 What are combined effect results	2.88 days	Wed 09-09-20
7		**2.2 Survey controllers**	**7.88 days**	**Mon 14-09-20**
8		2.2.1 Which controllers exist	2.88 days	Mon 14-09-20
9		2.2.2 Aerodynamic effect controllers	2.88 days	Wed 16-09-20
10		2.2.3 Collision/Interaction controllers	2.88 days	Mon 21-09-20
11		**2.3 Survey sensors**	**1.88 days**	**Thu 24-09-20**
14		**2.4 Initial simulations and experiments**	**22.88 days**	**Mon 28-09-20**
15		**2.4.1 Matlab**	**11.88 days**	**Mon 28-09-20**
16		**2.4.1.1 ceiling effect**	**5.88 days**	**Mon 28-09-20**
17		2.4.1.1.1 Feedback	1.88 days	Mon 28-09-20
18		2.4.1.1.2 Feedforward	1.88 days	Wed 30-09-20
19		2.4.1.1.3 Include lag filters	1.88 days	Fri 02-10-20
20		**2.4.1.2 Ground + ceiling effect**	**5.88 days**	**Tue 06-10-20**
21		2.4.1.2.1 Feedback	2.88 days	Tue 06-10-20
22		2.4.1.2.2 Feedforward	1.88 days	Thu 08-10-20
23		2.4.1.2.3 Include lag filters	1.88 days	Mon 12-10-20
24		**2.4.2 CFD**	**16 days**	**Tue 06-10-20**
25		2.4.2.1 Create geometries	2.88 days	Wed 14-10-20
26		2.4.2.2 Setup simulations	11 days	Tue 06-10-20
27		2.4.2.3 run simulations	1.88 days	Thu 22-10-20
28		2.4.2.4 analyze simulations	2.88 days	Mon 26-10-20
29		2.5 Write Literature study	11.88 days	Thu 29-10-20
30		**2.6 Developing initial solution**	**19.88 days**	**Mon 16-11-20**
31		2.6.1 INDI and possible adaptations	9.88 days	Mon 16-11-20
32		2.6.2 Final research plan	3.88 days	Mon 30-11-20
33		2.6.3 Preparing Software and Hardware of Crazyflie	5.88 days	Fri 04-12-20
34		2.7 Finalize literature study	4.88 days	Mon 14-12-20
35		2.8 Deadline literature study	0 days	Fri 18-12-20
36		**3 Reflection Literature study**	**4.88 days**	**Mon 04-01-21**
37		3.1 prepare presentation	2.88 days	Mon 04-01-21
38		3.2 Presentation	0.88 days	Thu 07-01-21
39		3.3 discussion with supervisors	0.88 days	Fri 08-01-21
40		**4 Implement INDI controller on Crazyflie**	**34.88 days**	**Mon 11-01-21**
41		4.1 Develop code	9.88 days	Mon 11-01-21
42		**4.2 Test INDI controller with flight tests**	**9.88 days**	**Mon 25-01-21**
43		4.2.1 Develop propeller guards	4.88 days	Mon 25-01-21



Project: Thesis Gannt
Date: Tue 08-12-20

Task	Project Summary	Manual Task	Start-only	Deadline
Split	Inactive Task	Duration-only	Finish-only	Progress
Milestone	Inactive Milestone	Manual Summary Rollup	External Tasks	Manual Progress
Summary	Inactive Summary	Manual Summary	External Milestone	

ID	Task Mode	Task Name	Duration	Start
44		4.3 Evaluate flight test data	4.88 days	Mon 08-02-21
45		4.4 Reporting	4.88 days	Mon 22-02-21
46		**5 Implement MRAC controller**	**44.88 days**	**Mon 01-03-21**
47		5.1 In depth reseach within MRAC solution	4.88 days	Mon 01-03-21
48		5.2 Hardware adaptation (extra sensors)	4.88 days	Mon 08-03-21
49		5.3 Code development	19.88 days	Mon 15-03-21
50		5.4 Flight testing	14.88 days	Mon 29-03-21
51		5.5 Evaluate flight test data	4.88 days	Mon 19-04-21
52		5.6 Reporting	4.88 days	Mon 26-04-21
53		**6 Midterm review**	**4.88 days**	**Mon 22-03-21**
54		6.1 prepare presentation	2.88 days	Mon 22-03-21
55		6.2 do presentation	0.88 days	Thu 25-03-21
56		6.3 discuss progress	1.88 days	Thu 25-03-21
57		7 Greenlight Review	0 days	Mon 03-05-21
58		**8 Overall evaluation of developed controller**	**9.88 days**	**Mon 03-05-21**
59		8.1 Flight testing within complete confined area	7.88 days	Mon 03-05-21
60		8.2 Evaluate flight test data	9.88 days	Mon 03-05-21
61		9 Iterate design / Slack time	9.88 days	Mon 17-05-21
62		10 Reporting	7.88 days	Fri 28-05-21
63		11 finalize code (controller)	1.88 days	Wed 09-06-21
64		12 Final deadline	0 days	Fri 11-06-21
65		13 thesis defence (presentation)	0.88 days	Mon 05-07-21



Project: Thesis Gannt
Date: Tue 08-12-20

Task	Project Summary	Manual Task	Start-only	Deadline	
Split	Inactive Task	Duration-only	Finish-only	Progress	
Milestone	Inactive Milestone	Manual Summary Rollup	External Tasks	Manual Progress	
Summary	Inactive Summary	Manual Summary	External Milestone		

6 Conclusion

To extend the applications of MAV in industry, flying fully autonomously within a small and enclosed area can be made possible with the development of a new controller. Before a solution is proposed a literature study is conducted to gather knowledge from work that has already been done. The aim of this literature study was to investigate what aspects and methods exist to answer the following question.

How can autonomous, stable and safe flight of an MAV quadcopter within a small enclosed environment be achieved?

Several aspects arose during the research from which several conclusion could be drawn. For the aerodynamic effects the ceiling effect was found to be the most aggressive and abrupt. It is also leading to collisions compared to the stabilizing ground effect. The wall effect is not widely investigated yet, but may also lead to problems as flight with a certain pitch angle makes the wall behave like a ground and or ceiling plane. Several aerodynamic models are developed that could be useful for obtaining disturbance information quickly, with the only downside being model fitting as the aerodynamic effect highly depend on geometry of the used platform.

With the main challenge being the handling of disturbances either caused by the aerodynamic effects or self induced flow, there was looked for controllers focused on those disturbances. The PID controller is widely used, but by itself not robust and is thus combined with other techniques to improve the performance. The other techniques came down to either usage of models describing the disturbances or looking into the difference between what a mathematical model provides and what is measured on the real system. Both methods showed successful results for counteracting several aerodynamic effects. Collisions may also cause problems when flying within the enclosed environment. By using information which is already needed for the main controller a collision detection system is relatively easy to setup. The response to a collision will either be stopping or moving away from the collision surface.

Besides looking into literature two kinds of experiments were conducted. A CFD simulation has shown that different flow patterns arise when hovering at different location inside an enclosed area. This showed that the main disturbances in the room have a quarter of the velocity compared to the flow right below the MAV, and is thus expected to disturb the MAV when flown through. From this experiment it could be seen that not only aerodynamic effects will play a main part in disturbance of the drone but that the enclosed environment will also induce extra challenges. The second experiment was the investigation of a PID controller using Matlab Simulink. A highly simplified one dimensional (up, down) flight was simulated using a pure feedback position PID controller and an aerodynamic model feedforward PID controller. It was found that the feedback controller needed high gains to function properly but the most extreme gains are also usable within less extreme conditions, thus one could think that a PID controller tuned for most extreme condition is a solution. However the simulation is not taking into account noise which would also be amplified by the high gains (mainly proportional gain) which may lead to a non functioning controller. The feedforward controller solved the high gains but had the advantage of having a perfect model for the simulated disturbance. Model errors will degrade the performance but it shows that estimation of disturbances is a promising method.

The knowledge from literature and experiments was combined to come up with a solution that will be further developed and tested. Literature had shown two main methods for tackling disturbances both showing promising results. The solution proposed will combine both techniques, it will use aerodynamic models to obtain a first estimate of the disturbances but with time it adapts to try and better describe reality. This has been chosen as the complex flow structures seen from the CFD will cause a mismatch between the developed aerodynamic models from literature and reality and thus adaptations will try to overcome this. The method can be summarized within the main research question, which will be answered by the thesis work. The to be answered question is the following:

Can an autonomous MAV controller, which is a combination of MRAC and INDI, provide stable flight within an enclosed area where the dimensions are a maximum of ten times larger than the MAV dimensions?

Finally the outcome of this thesis will thus provide a new approach into stable indoor flight and will aid in the development of fully autonomous inspection systems. One company that is looking into the use of an autonomous MAV system for indoor flight is Lockheed Martin. They are investigating the replacement of a human, inspecting the inside of fuel tanks in their aircraft by an autonomous MAV.

Note

The research paper does not go into the approach described by the final research question in the literature study. During the research it was found that a combination of MRAC and INDI was not ideal, as both methods were working against each other. It was thus proposed to look into pure INDI control and adjusting it to compensate for the ceiling effect.

Bibliography

[1] X. He, G. Kou, M. Calaf, and K.K. Leang. In-ground-effect modeling and nonlinear-disturbance observer for multirotor unmanned aerial vehicle control. *Journal of Dynamic Systems, Measurement, and Control*, 141(7), 2019. doi:10.1115/1.4043221.

[2] E. Volodscoi. Design, implementation and evaluation of an incremental nonlinear dynamic inversion controller for a nano-quadcopter. Master's thesis, Technische Universität München, Boltzmannstraße 15, 85748 Garching, 4 2020. An optional note.

[3] E. Walter, J. Norton, and L. Pronzato. *Identification of Parametric Models: From Experimental Data*. Communications and Control Engineering. Springer, 1997. URL: https://books.google.nl/books?id=x_1QAAAAMAAJ.

[4] G. Shi, X. Shi W. Hönig, Y. Yue, and S.J. Chung. Neural-swarm2: Planning and control of heterogeneous multirotor swarms using learned interactions. *IEEE Transactions on Robotics*, abs/2012.05457, 2020.

[5] B. Rao, A.G. Gopi, and R. Maione. The societal impact of commercial drones. *Technology in Society*, 45:83–90, 2016. doi:10.1016/j.techsoc.2016.02.009.

[6] C. Sampedro, A.R. Ramos, H. Bavle, A. Carrio, P. de la Puente, and P. Campoy. A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *Journal of Intelligent & Robotic Systems*, 95:601–627, 2019. doi:10.1007/s10846-018-0898-1.

[7] A. Briod, A. Klaptocz, J.C. Zufferey, and D. Floreano. The airburr: A flying robot that can exploit collisions. *International Conference on Complex Medical Engineering*, pages 569–574, 2012. doi:10.1109/ICCME.2012.6275674.

[8] I.C. Cheeseman and W.E. Bennett. The effect of the ground on a helicopter rotor in forward flight. *Aeronautical Research Council Reports and Memoranda*, 1957.

[9] S.A. Conyers, M.J. Rutherford, and K.P. Valavanis. An empirical evaluation of ground effect for small-scale rotorcraft. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1244–1250, 2018. doi:10.1109/ICRA.2018.8461035.

[10] P.J. Sanchez-Cuevas, G. Heredia, and A. Ollero. Characterization of the aerodynamic ground effect and its influence in multirotor control. *International Journal of Aerospace Engineering*, 2017. doi:10.1155/2017/1823056.

[11] S.A. Conyers, M.J. Rutherford, and K.P. Valavanis. An empirical evaluation of ceiling effect for small-scale rotorcraft. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 243–249, 2018. doi:10.1109/ICUAS.2018.8453469.

[12] P.J. Sanchez-Cuevas, G. Heredia, and A. Ollero. Experimental approach to the aerodynamic effects produced in multirotors flying close to obstacles. *Third Iberian Robotics Conference, Advances in Intelligent Systems and Computing*, pages 742–752, 2018. doi:10.1007/978-3-319-70833-1_60.

[13] A.E. Jimenez-Cano, P.J Sanchez-Cuevas, A. Ollero P. Grau, and G. Heredia. Contact-based bridge inspection multirotors: Design, modeling, and control considering the ceiling effect. *IEEE Robotics and Automation Letters*, 4(4):3561–3568, 2019. doi:10.1109/LRA.2019.2928206.

[14] T. Jardin, S. Prothin, and C.G. Magaña. Aerodynamic performance of a hovering microrotor in confined environment. *Journal of The American Helicopter Society*, 62(2):1–7, 2017. doi:10.4050/JAHS.62.022008.

[15] S. Prothin, C.F. Escudero, N. Doué, and T. Jardin. Aerodynamics of mav rotors in ground and corner effect. *International Journal of Micro Air Vehicles*, 11, 2019. doi:10.1177/1756829319861596.

[16] P. Wei, S.N. Chan, S. Lee, and Z. Kong. Mitigating ground effect on mini quadcopters with model reference adaptive control. *International Journal of Intelligent Robotics and Applications*, 3:283–297, 2019. doi:10.1007/s41315-019-00098-z.

[17] M.G. Michailidis, M.J. Rutherford, and K.P. Valavanis. A survey of controller designs for new generation uavs: The challenge of uncertain aerodynamic parameters. *International Journal of Control, Automation and Systems*, 18(4):801–816, 2020. doi:10.1007/s12555-018-0489-8.

[18] H. Nguyen, V.C. Nguyen, T.V. Quyen, and A.M. Le. Control algorithms for uavs: A comprehensive survey. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, 7(23), 2020. doi:10.4108/eai.18-5-2020.164586.

[19] F. Alyoussef and I. Kaya. A review on nonlinear control approaches: Sliding mode control, back-stepping control and feedback linearization control. *International Engineering and Natural Sciences Conference (IENSC)*, pages 608–619, 2019.

[20] J. Kim, S.A. Gadsden, and S.A. Wilkerson. A comprehensive survey of control strategies for autonomous quadrotors. *Canadian Journal of Electrical and Computer Engineering*, 43(1):3–16, 2020. doi:10.1109/CJECE.2019.2920938.

[21] Y. Li and S. Song. A survey of control algorithms for quadrotor unmanned helicopter. *IEEE fifth International Conference on Advanced Computational Intelligence(ICACI)*, pages 365–369, 2012. doi:10.1109/ICACI.2012.6463187.

[22] H. Mo and G. Farid. Nonlinear and adaptive intelligent control techniques for quadrotor uav - a survey. *Asian Journal of Control*, 21(2):989–1008, 2019. doi:10.1002/asjc.1758.

[23] E.J.J. Smeur, G. de Croon, and Q.P. Chu. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, 2016. doi:10.2514/1.G001490.

[24] W.H. Chen, D.J. Ballance, P.J. Gawthorp, and J. O'Reilly. A nonlinear disturbance observer for robotic manipulators. *IEEE Transactions on Industrial Electronics*, 47(4):932–938, 2000. doi:10.1109/41.857974.

[25] B.B. Kocer, M.E. Tiryaki, M. Pratama, T. Tjahjowidodo, and G.G.L. Seet. Aerial robot control in close proximity to ceiling: A force estimation-based nonlinear mpc. *International Conference on Intelligent Robots and Systems*, pages 2813–2819, 2019. doi:10.1109/IROS40897.2019.8967611.

[26] P. Kühl, M. Diehl, T. Kraus, J.P Schlöder, and H.G. Bock. A real-time algorithm for moving horizon state and parameter estimation. *Computer & Chemical Engineering*, 35(1):71–83, 2011. doi:10.1016/j.compchemeng.2010.07.012.

[27] T. Tomić and S. Haddadin. A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots. *International Conference on Intelligent Robots and Systems*, pages 4197–4204, 2014. doi:10.1109/IROS.2014.6943154.

[28] C.D. McKinnon and A.P. Schoellig. Unscented external force and torque estimation for quadrotors. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5651–5657, 2016. doi:10.1109/IROS.2016.7759831.

[29] E.J.J. Smeur, G.C.H.E de Croon, and Q. Chu. Cascaded incremental nonlinear dynamic inversion control for mav disturbance rejection. *Control Engineering Practice*, 73:79–90, 2018. doi:10.1016/j.conengprac.2018.01.003.

[30] G. Dicker, F. Chui, and I. Sharf. Quadrotor collision characterization and recovery control. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5830–5836, 2017. doi:10.1109/ICRA.2017.7989685.

[31] F. Chui. Quadrotor collision dynamics and fuzzy logic characterization. Master's thesis, McGill University, Montreal, Quebec, 12 2016.

[32] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza. Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, 2015. doi:10.1109/ICRA.2015.7139420.

[33] T. Nakata, N. Phillips, P. Simões, I.J. Russell, J.A. Cheney, S.M. Walker, and R.J. Bomphrey. Aerodynamic imaging by mosquitoes inspires a surface detector for autonomous flying vehicles. *Science 368*, 368(6491):634–637, 2020. doi:10.1126/science.aaz9634.

[34] D.W. Yeo, N. Sydney, and D.A. Paley. Onboard flow sensing for downwash detection and avoidance with a small quadrotor helicopter. *AIAA Guidance, Navigation, and Control Conference*, 2015. doi:10.2514/6.2015-1769.

[35] L.M. González-deSantos, J. Martínez-Santos, H. González-Jorge, M. Ribeiro, J.B. de Sousa, and P. Arias. Payload for contact inspection tasks with uav systems. *Sensors*, 19:3752, 2019. doi:10.3390/s19173752.

[36] E.F. Helbling, S.B. Fuller, and R.J. Wood. Altitude estimation and control of an insect-scale robot with an onboard proximity sensor. In *Robotics Research*, pages 57–69. Springer, 2018.

[37] K. Nohara, S. Kawabata, J.H. Lee, S. Okamoto, H. Suzuki, T. Takiguchi, and O.S. Park. Development of drone system embedded with multiple distance sensors for inspection task of social infrastructure. *International MultiConference of Engineers and Computer Scientists*, 2:1–5, 2018.

[38] S. Adarsh, S.M. Kaleemuddin, B. Dinesh, and K.I. Ramachandran. Performance comparison of infrared and ultrasonic sensors for obstacles of different materials in vehicle/ robot navigation applications. *IOP Conference Series: Materials Science and Engineering*, 149(1):012141, 2016. doi:10.1088/1757-899X/149/1/012141.

[39] N. Gagaeik, P. Benz, and S. Montenegro. Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. *IEEE Access*, 3:599–609, 2015. doi:10.1109/ACCESS.2015.2432455.