

RAM

ROBOTICS
AND
MECHATRONICS

LINEAR CONTROLLER DESIGN FOR AN OMNIDIRECTIONAL MORPHING UAV AT DIFFERENT OPERATING POINTS

A. (Aaron) Saini

BSC ASSIGNMENT

Committee:

prof. dr. ir. A. Franchi
Y.A.L.A. Aboudorra, MSc
dr. W. Kuijper

July, 2023

031RaM2023
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract—Unmanned Aerial Vehicles (UAVs) are an interesting tool for executing a wide variety of missions. This thesis studies the design of a linear controller for the OmniMorph, a novel omnidirectional morphing UAV equipped with eight tiltable, bidirectional rotors. By tilting the rotors the platform can change its configuration mid-flight in order to adapt to the requirements of a mission. A model of the platform is used to study the capabilities and limitations of the system, as well as providing a foundation for designing the controller. The controller will adapt to the tilt angle of the rotors, enabling it to fully exploit the capabilities at each tilt angle. To adapt to the operating point, two gain scheduling methods are proposed. Firstly, a simple method with limited performance is presented, followed by a more sophisticated optimization algorithm that offers enhanced performance. These techniques are initially developed for a simplified 1D case, after which the extension of these techniques to the OmniMorph is studied. The results for the 1D case show that the gain scheduling methods successfully enable the controller to fully exploit the system’s capabilities at each operating point. Although the techniques have not yet been fully adapted to accommodate the OmniMorph, the partial results show promising progress.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are widely studied in literature, with focus ranging from classical designs such as the coplanar, colinear quadrotor, to novel configurations that overcome the limitations that some simpler designs have. A great limitation of the traditional quadrotor is that it is underactuated, meaning that it cannot independently control all its degrees of freedom.

A possible solution to this limitation is to tilt the rotors such that each rotor produces its thrust in a different direction. In this way, the platform can apply thrust along more directions than just along the platform’s vertical axis, allowing the platform to become fully actuated. The ability to independently control each degree of freedom allows the platform to carry out a much greater variety of tasks. An interesting application for fully actuated UAVs is aerial manipulation, which is physical interaction between the platform and the environment [1]. There are a number of platforms proposed in literature that tilt the rotors under a fixed angle such as the omnidirectional platform in [2] and the ODAR in [3]. However, with the tilt angle fixed, the platform is bound to a single configuration, which may not always be optimal for the task at hand. By actuating the tilt angle, such as the FAST-Hex in [4], it becomes possible to switch between different configurations mid-flight.

This thesis will examine the OmniMorph [5], a novel omnidirectional platform concept that can actuate the tilt angle of its rotors. This platform consists of eight tiltable, bidirectional rotors placed on the vertices of a cube. Each rotor can tilt about an axis that is an extension of an edge of the aforementioned cube. The tilt mechanism of each of the rotors is controlled by the same actuator so all rotors will always have an equal tilt angle α . A render of the platform can be seen in figure 1.

The maximum forces and torques that the platform can apply in a certain direction change depending on α . For example, when all rotors are aligned, the platform cannot apply any lateral force. In this configuration the platform is equivalent to two stacked quadrotors. However, when α

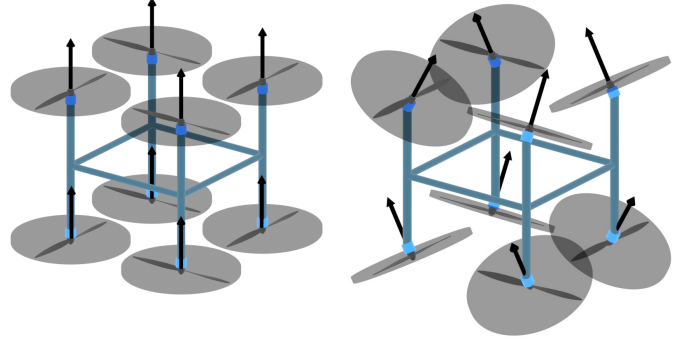


Fig. 1. The OmniMorph platform at $\alpha = 0^\circ$ (left) and at $\alpha = 30^\circ$ (right).

increases it becomes possible to apply more lateral force in exchange for a decreased maximum vertical force. In order to fully exploit the feasible force and torque at each tilt angle, the controller will automatically adapt to α by means of a technique formally known as gain scheduling. The aim of this thesis is to design a linear controller for this platform that automatically adapts to the tilt angle of the rotors, such that the platform can be utilized to its full potential.

The remainder of this paper is structured as follows. Section II will present a model of the OmniMorph and will analyze the properties, capabilities and limitations of the platform. Section III will propose a controller with fixed gains. Section IV will first analyze the limitations imposed by fixed gains for a simplified 1D case, after which two methods will be proposed to fully exploit the system. After this, the extension of these results to accommodate the OmniMorph is studied. At the moment, this adaptation is only partial due to the time constraints imposed on the project. However, suggestions to complete this adaptation in future work are presented.

II. MODEL

A. Wrench analysis

The model will use two reference frames: the world frame W at the world origin and the body-fixed frame B attached to the center of mass of the UAV. Both frames follow the east-north-up (ENU) convention. The position and rotation of B with respect to W are given by ${}^W\mathbf{p}$ and ${}^W\mathbf{R}_B$, respectively.

Each rotor exerts a force $\mathbf{f}_{r,i}$ to the body of the UAV along the rotation axis of the propeller, normal to the rotor plane. This normal is given by \mathbf{n}_i . The total force is the sum of all the individual thrusts:

$${}^B\mathbf{f}_{total} = \sum_{i=1}^8 \mathbf{f}_{r,i} \mathbf{n}_i = \sum_{i=1}^N c_f u_i \mathbf{n}_i \quad (1)$$

Where c_f is the thrust coefficient and u_i is the input of the i^{th} rotor, defined as $u_i = \omega_i |\omega_i|$, where ω_i is the spinning rate of the i^{th} rotor.

There will also be a moment about the center of mass of the UAV. The total moment is given by:

$${}^B\boldsymbol{\tau}_{total} = \sum_{i=1}^8 f_{r,i}(\mathbf{p}_i \times \mathbf{n}_i) + k_i \frac{c_\tau}{c_f} f_{r,i} \mathbf{n}_i \quad (2)$$

$$= \sum_{i=1}^8 c_f u_i (\mathbf{p}_i \times \mathbf{n}_i) + k_i c_\tau u_i \mathbf{n}_i$$

The moment consists of two parts, the thrust moment and the drag moment. The thrust moment is induced because the thrust is generated at a certain offset \mathbf{p}_i from the center of mass. The drag moment is caused by the aerodynamic drag of the propellers, which induces a moment to the body in the direction opposite to the spinning direction. The drag moment is about the rotor normal and is scaled by the drag coefficient c_τ . The direction of the drag moment of a certain rotor depends on its spinning direction, which is accounted for by $k_i = \pm 1$.

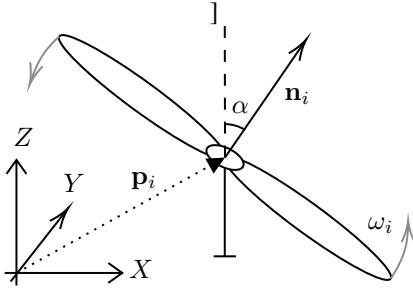


Fig. 2. Schematic overview of a rotor showing the rotor position \mathbf{p}_i , the rotor normal \mathbf{n}_i and the tilt angle α .

B. Allocation

The rotational velocities of the rotors are the inputs of the system. The mapping from the actuator inputs $\mathbf{u} = [\omega_1|\omega_1|, \dots, \omega_8|\omega_8|]^T = [u_1, u_2, \dots, u_8]^T$ to the total forces and moments (wrench \mathbf{w}) can conveniently be described with the wrench map $\mathbf{F} \in \mathbb{R}^{6 \times 8}$:

$${}^B\mathbf{w} = \mathbf{F}\mathbf{u} \quad (3)$$

Where ${}^B\mathbf{w} = [f_x, f_y, f_z, \tau_x, \tau_y, \tau_z]^T = [{}^B\mathbf{f}_{total}, {}^B\boldsymbol{\tau}_{total}]^T$ is the wrench applied to the body expressed in the body frame. Equation (4) shows the full definition of the wrench map substituted in (3).

$$\begin{bmatrix} {}^B\mathbf{f}_{total} \\ {}^B\boldsymbol{\tau}_{total} \end{bmatrix} = \begin{bmatrix} c_f \mathbf{N} \\ c_f(\mathbf{P} \times \mathbf{N}) + c_\tau \mathbf{N}\mathbf{K} \end{bmatrix} \mathbf{u} \quad (4)$$

Where $\mathbf{K} = \text{diag}(k_1, k_2, \dots, k_8)$ accounts for the spinning directions of the rotors, $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_8]$ collects all rotor normals, and $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_8]$ collects all rotor offsets from the center of mass. The full definitions of \mathbf{N} and \mathbf{P} are given by (5) and (6), respectively. For the sake of compact notation, $\sin(\alpha)$ and $\cos(\alpha)$ are abbreviated to s and c , respectively. The distance from the center of mass (COM) to each rotor is denoted by D .

$$\mathbf{N} = \begin{bmatrix} -s & 0 & s & 0 & s & 0 & -s & 0 \\ 0 & -s & 0 & s & 0 & s & 0 & -s \\ c & c & c & c & c & c & c & c \end{bmatrix} \quad (5)$$

$$\mathbf{P} = \frac{D}{\sqrt{3}} \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix} \quad (6)$$

C. Body dynamics

The dynamics of the platform can be described using the Newton-Euler equations. The translational dynamics are described by (7), the rotational dynamics are described by (8).

$$m {}^W\ddot{\mathbf{p}} = {}^W\mathbf{R}_B {}^B\mathbf{f}_{total} - mg {}^W\hat{\mathbf{z}} \quad (7)$$

$$\mathbf{J} {}^B\dot{\boldsymbol{\omega}} = {}^B\boldsymbol{\tau}_{total} - {}^B\boldsymbol{\omega} \times \mathbf{J} {}^B\boldsymbol{\omega} \quad (8)$$

In the Newton-Euler equations, m denotes the mass of the platform, \mathbf{J} denotes the rotational inertia of the platform and ${}^B\boldsymbol{\omega}$ denotes the angular velocity vector expressed in the body frame. The total force and torque on the platform are given by ${}^W\mathbf{f}_{total}$ and ${}^W\boldsymbol{\tau}_{total}$ respectively, as defined in (1) and (2). The gravitational force on the platform is described by the product of the mass m , the gravitational acceleration g and the unit vector ${}^W\hat{\mathbf{z}}$ expressed in the world frame.

The attitude kinematics are given by (9). This equation relates the change of the attitude over time to the angular velocity. Here $s(\cdot)$ denotes the mapping from a 3-dimensional vector to the associated skew-symmetric matrix.

$${}^W\dot{\mathbf{R}}_B = s({}^B\boldsymbol{\omega}) {}^W\mathbf{R}_B \quad (9)$$

D. Feasible wrench

The force that each rotor can generate is limited according to $f_{i,min} \leq f_{r,i} \leq f_{i,max}$. Depending on the tilt angle α , these actuator limits map differently to certain limits on the wrench that the platform can generate. First, the set of feasible force with zero moment is considered. An intuitive way to look at it is as follows: when the tilt angle is small, most force can be applied vertically, but almost no force can be applied laterally. When α increases, the amount of lateral force that can be applied increases at the expense of decreased feasible vertical force. This can be seen in figure 3, where the convex hulls of the feasible force sets are visualized for different values of α .

In order to calculate the set of feasible force, for each wrench $\mathbf{w} = [f_x, f_y, f_z, 0, 0, 0]^T$, the feasibility of the corresponding input vector \mathbf{u} is verified using the input allocation described in section III-A. The forces f_x , f_y and f_z are each varied between a lower and upper bound. The bounds are increased until the feasible force set no longer grows, ensuring that all feasible wrenches are encompassed. The wrench is considered feasible if no rotor exceeds its maximum thrust, so $f_{i,min} < f_{r,i} < f_{i,max}$.

The same process is repeated for the torque, but instead of setting \mathbf{f} to zero, it is set to $\mathbf{f} = [0, 0, mg]^T$, in order to compensate for gravity. Similar to before, the feasibility of \mathbf{u} is verified for each wrench $[0, 0, mg, \tau_x, \tau_y, \tau_z]^T$, where each torque τ_x , τ_y and τ_z is bounded. Again, the bounds are increased until the feasible set no longer grows, meaning that all feasible torques have been encompassed. The result can be seen in figure 4.

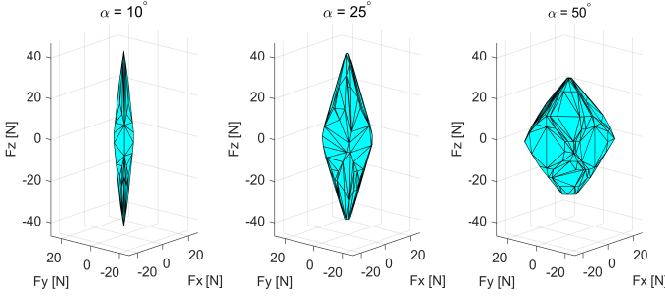


Fig. 3. Convex hull of feasible force set with ${}^B\boldsymbol{\tau} = [0, 0, 0]^T$ visualized for $\alpha = 10^\circ$, $\alpha = 25^\circ$ and $\alpha = 50^\circ$.

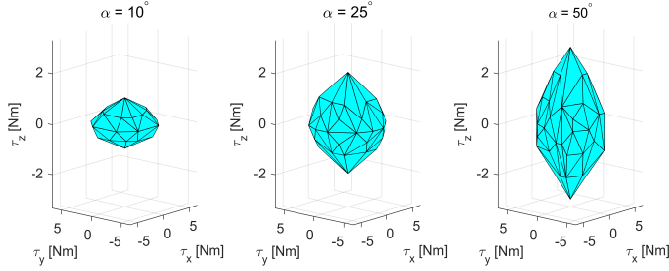


Fig. 4. Convex hull of feasible torque set visualized with ${}^B\boldsymbol{f} = [0, 0, mg]^T$ for $\alpha = 10^\circ$, $\alpha = 25^\circ$ and $\alpha = 50^\circ$.

E. Platform properties

Depending on α , the fundamental properties of the platform change. When $\alpha = 0^\circ$, all motor normals are aligned coplanarly and colinearly, which means that the platform is equivalent to two stacked quadrotors. In this configuration, the platform cannot exert any lateral forces without also changing its attitude. This coupling between translational and rotational dynamics means that the platform is underactuated: not all degrees of freedom (DOFs) can be controlled independently. Another extreme case is when $\alpha = 90^\circ$, in this case the platform can only exert lateral forces. The tradeoff between vertical and lateral force can be seen in figure 5. This figure shows the maximum lateral and vertical force versus α . As seen before in figure 3, the maximum feasible lateral force becomes larger for larger α , and the maximum feasible vertical force decreases as α increases. The platform can only compensate for gravity (and thus fly) if $f_{z,max} > mg$, which is indicated as the green range in figure 5. At $\alpha \approx 78.2^\circ$, the platform is no longer able to compensate for gravity as $f_{z,max} < mg$.

The actuation capability of the platform can formally be described by the rank of the wrench map. When $\alpha = 0^\circ$, $\text{rank}\{\mathbf{F}\} = 4$, which means the platform is underactuated, as only 4 DOFs can be controlled independently. When $\alpha \neq 0^\circ$ and $\alpha \neq 90^\circ$, the OmniMorph is overactuated, which means that $\text{rank}\{\mathbf{F}\} = 6$ and that there are more inputs than DOFs. Overactuation implies that there are multiple inputs that leads to the same output, which complicates the input allocation. The platform becomes omnidirectional when a sphere of radius mg fits inside the feasible force set in figure 3, which means that the platform can sustain gravity in any orientation. These platform properties are discussed in detail in [6].

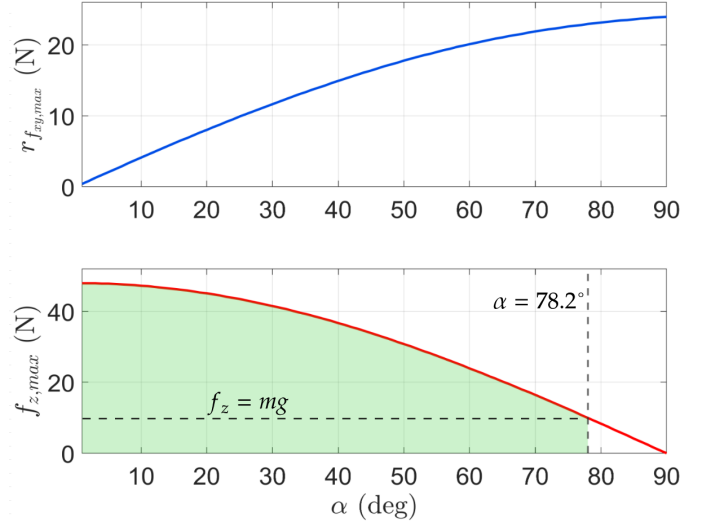


Fig. 5. Top: $r_{f_{xy}}$ versus α for $f_z = 0$, where $r_{f_{xy}}$ is the radius of largest inner circle in the convex hull in figure 3 on the horizontal plane $f_z = 0$ centered at the origin. This is a conservative measure of the maximum feasible lateral force. Bottom: maximum feasible f_z versus α for $f_x = f_y = 0$. The maximum α at which hover is possible is indicated by the dashed line ($\alpha = 78.2^\circ$).

For the remainder of this paper, the focus will lie on the range of alpha where the platform is sufficiently fully actuated and where it is able to fly. This range is defined as $\alpha \in [6^\circ, 73^\circ]$. The lower bound ensures that the platform is fully actuated. The angle $\alpha = 6^\circ$ corresponds to a feasible lateral force that is at least 10% of the peak feasible lateral force at $\alpha = 90^\circ$. The upper bound accounts for the fact that the platform can no longer sustain gravity beyond $\alpha = 78.2^\circ$. The upper bound is selected at $\alpha = 73^\circ$ which ensures that when the platform is sustaining gravity, this uses at most 90% of the total feasible vertical force, leaving a margin of at least 10%, similar to the lower bound.

F. Model parameters

Table I below shows all relevant model parameters.

Variable	Description	Value
c_f	Thrust coefficient	$1.1 \cdot 10^{-4} \text{ N}/(\text{rad}/\text{s})^2$
c_τ	Drag coefficient	$2.2 \cdot 10^{-6} \text{ Nm}/(\text{rad}/\text{s})^2$
m	Platform mass	1.0 kg
J	Platform inertia matrix	$\text{diag}(1.16, 1.13, 1.13) \cdot 10^{-2} \text{ kg m}^2$
D	Distance from COM to rotor	0.2165 m
$f_{i,min}$	Minimum rotor force	-6.0 N
$f_{i,max}$	Maximum rotor force	6.0 N

TABLE I
MODEL PARAMETERS

III. CONTROLLER DESIGN

In order to track a time-varying 6D reference trajectory, a suitable controller needs to be implemented. The structure of the system with the platform model with the controller can be seen in figure 6. The reference trajectory will consist of a desired position, velocity and acceleration ($\mathbf{p}_d, \dot{\mathbf{p}}_d, \ddot{\mathbf{p}}_d$) and a desired attitude, angular velocity and angular acceleration ($\mathbf{R}_d, \boldsymbol{\omega}_d, \dot{\boldsymbol{\omega}}_d$). The position error is defined as $\mathbf{e}_p = \mathbf{p} - \mathbf{p}_d$, and the velocity error is defined as $\mathbf{e}_v = \dot{\mathbf{p}} - \dot{\mathbf{p}}_d$. The

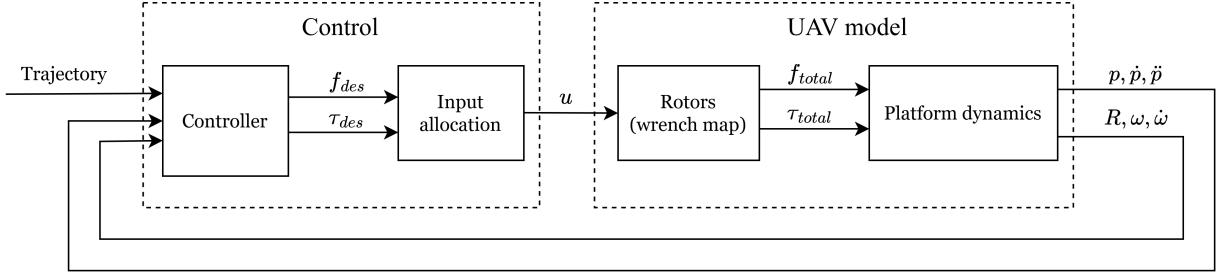


Fig. 6. Structure of the platform model along with the controller.

integral position error is defined as $e_i = \int_0^t e_p d\tau$. The attitude and angular velocity errors are defined as (10) and (11), respectively [7]. The integral attitude error is defined as $e_{Ri} = \int_0^t e_R d\tau$.

$$e_R = \frac{1}{2}(\mathbf{R}^T \mathbf{R}_d - \mathbf{R}_d^T \mathbf{R})^\vee \quad (10)$$

$$e_\omega = \mathbf{R}^T \mathbf{R}_d \omega_d - \omega \quad (11)$$

Where $(\cdot)^\vee$ maps a skew symmetric matrix to the associated 3-dimensional vector. The desired force is calculated with the use of PID-control as follows:

$${}^B \mathbf{f}_{des} = {}^W \mathbf{R}_B^T m(\ddot{\mathbf{p}}_d + g\hat{\mathbf{z}} + \mathbf{K}_p \mathbf{e}_p + \mathbf{K}_i \mathbf{e}_i + \mathbf{K}_d \mathbf{e}_v) \quad (12)$$

Where $\mathbf{K}_p, \mathbf{K}_i, \mathbf{K}_d \in \mathbb{R}^{3 \times 3}$ are the diagonal position tracking gain matrices. This control law for desired force consists of gravity compensation, the feedforward of desired acceleration and the actual dynamics that rejects disturbances. The desired torque is calculated with the use of PID-control as follows:

$${}^B \boldsymbol{\tau}_{des} = \boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega} + \mathbf{J}(\mathbf{K}_R \mathbf{e}_R + \mathbf{K}_{Ri} \mathbf{e}_{Ri} + \mathbf{K}_\omega \mathbf{e}_\omega) \quad (13)$$

Where $\mathbf{K}_R, \mathbf{K}_{Ri}, \mathbf{K}_\omega \in \mathbb{R}^{3 \times 3}$ are the diagonal attitude tracking gain matrices. The control laws correspond to those proposed in [4], with the exception that the sign of the error is flipped and that the PID terms are now multiplied by the mass and inertia. The PID terms are scaled by the mass and inertia in order to scale the gains according to the amount of force required to accelerate. For example, for a larger mass, the controller needs to apply more force to reach the same acceleration. This could also be accounted for by scaling the gains themselves, but as \mathbf{J} is very small, this would result in gain values that are impractical to work with.

A. Input allocation

The output of the controller is the desired wrench. A combination of inputs \mathbf{u} needs to be found such that the wrench generated by the platform is equal to the desired wrench \mathbf{w}_{des} . This means that the equation $\mathbf{w}_{des} = \mathbf{F}\mathbf{u}$ needs to be solved for \mathbf{u} . As \mathbf{F} is a non-square matrix, it cannot be inverted directly, so another method needs to be used. A simple approach is using the pseudo-inverse, however, this may lead to an infeasible \mathbf{u} , even if \mathbf{w}_{des} is feasible [2].

Another way to solve this equation is by approaching it as an optimization problem, such as in [8]. The optimization aims

to minimize the cost function $\|\mathbf{u}\|_2^2$, which is the squared 2-norm of the input vector. This optimization is subject to the following constraints:

$$\begin{cases} \mathbf{F}\mathbf{u} = \mathbf{w}_{des} \end{cases} \quad (14)$$

$$\begin{cases} \mathbf{u}_{min} < \mathbf{u} < \mathbf{u}_{max} \end{cases} \quad (15)$$

Where $\mathbf{u}_{min} \in \mathbb{R}^8$ and $\mathbf{u}_{max} \in \mathbb{R}^8$ are the lower and upper bound of the input vector related to the minimum and maximum rotor force $f_{i,min}$ and $f_{i,max}$ according to $u_{i,min} = \frac{f_{i,min}}{c_f}$ and $u_{i,max} = \frac{f_{i,max}}{c_f}$. The equality constraint (14) ensures that \mathbf{u} results in the desired wrench and the inequality constraint (15) ensures that \mathbf{u} is feasible. The minimization of the norm of the input ensures that the rotor thrusts are as small as possible while respecting the constraints, which is optimal for efficiency. This optimization can be solved using quadratic programming.

B. Trajectory generation

The position and attitude of the platform will track a certain trajectory. It is useful to be able to set the final displacement and the maximum acceleration of this trajectory. The maximum acceleration is important to set, as the constraints of the system are imposed on the force, which is related to acceleration. This section presents a generic trajectory $x(t)$ that can be used as a reference for any DOF of the platform. The trajectory is defined as the third order polynomial in (16), such that $x(0) = 0$, $x(\tau) = \Delta x$, $\dot{x}(0) = 0$ and $\dot{x}(\tau) = 0$, where Δx is the final displacement and τ is the duration.

$$x(t) = \Delta x \cdot \left(-2\left(\frac{t}{\tau}\right)^3 + 3\left(\frac{t}{\tau}\right)^2\right) \quad (16)$$

The maximum acceleration occurs at $t = 0$ and at $t = \tau$ and is equal to $|\ddot{x}_{max}| = \frac{6\Delta x}{\tau^2}$. This means that the maximum acceleration of the trajectory can easily be set by setting the duration to $\tau = \sqrt{\frac{6\Delta x}{|\ddot{x}_{max}|}}$. For example, figure 7 shows $x(t)$, $\dot{x}(t)$ and $\ddot{x}(t)$ for $\Delta x = 1m$ and $|\ddot{x}_{max}| = 2m/s^2$.

IV. GAIN SCHEDULING

A. Simplified case for 1D position tracking

In order to clearly demonstrate the limitations that need to be overcome, it is helpful to look at a fundamental case first, before diving into the full 6D model with additional complications. This simplified case needs the following main properties to resemble the properties of the multirotor platform:

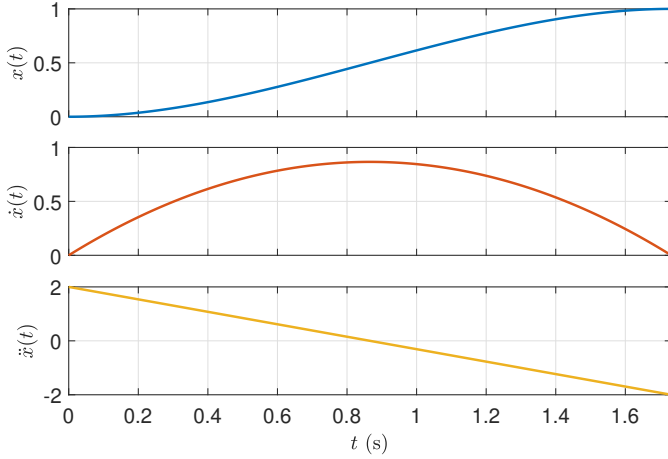


Fig. 7. Generated trajectory for $\Delta x = 1m$, $\ddot{x}_{max} = 2m/s^2$ ($\tau = \sqrt{3}$ s).

- The range of the actuator in the system needs to be limited.
- The operating point of the system must change the control authority that the actuator has on the system.

A setup that has these properties and resembles the multirotor platform, but is much simpler than the multirotor platform, is a cart on rails that can move freely along a single axis. There is a propeller attached to the cart that can tilt by angle β (to prevent confusion with α , which is the tilt angle of the main platform). When $\beta = 0^\circ$, no force can be applied along the movement axis, whereas when $\beta = \pm 90^\circ$, full force can be applied. The setup can be seen in figure 8. The thrust that the rotor can generate is limited to $\pm 6N$, equal to the thrust limit of the multirotor actuators. The mapping from the rotor thrust f_r to applied horizontal force is given by $f_x = \sin(\beta)f_r$, so the maximum horizontal force is $|f_{x,max}(\beta)| = \sin(\beta) \cdot 6N$, which is plotted in figure 9.

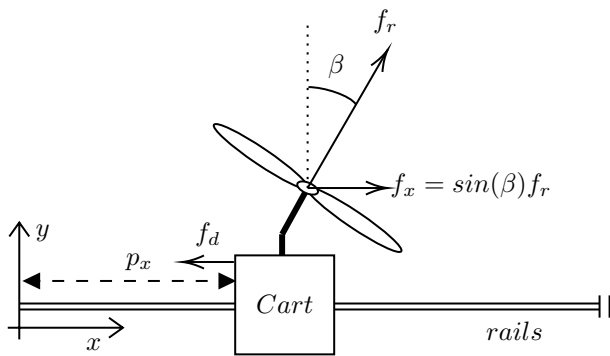


Fig. 8. Overview of the 1D cart setup with a propeller. Here p_x is the position along the rails, f_r is the rotor thrust, $f_x = \sin(\beta)f_r$ is the horizontal projection of the rotor thrust and f_d is the disturbance force.

A PID controller is added to the system, that aims to track the desired position $p_{x,d}$. The position error is defined as $e_p = p_{x,d} - p_x$, where p_x is the position along the movement axis. The control law is defined as (17).

$$f_r = \frac{1}{\sin(\beta)}(k_p e_p + k_d \dot{e}_p + k_i \int_0^t e_p d\tau) \quad (17)$$

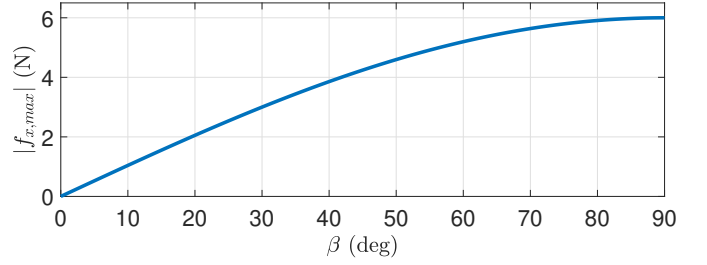


Fig. 9. Maximum feasible $|f_x|$ versus β of 1D cart, where $|f_{r,max}| = 6N$.

Where k_p , k_i and k_d are the controller gains. The factor $\frac{1}{\sin(\beta)}$ acts as the input allocation.

To obtain the set of baseline gains $k_{p,base}$, $k_{i,base}$ and $k_{d,base}$ at the baseline tilt angle $\beta_B = 15^\circ$, the gains are tuned by trial and error for a unit step input to $p_{x,d}$. The result can be seen in figure 10, which shows that the system properly tracks a step reference input.

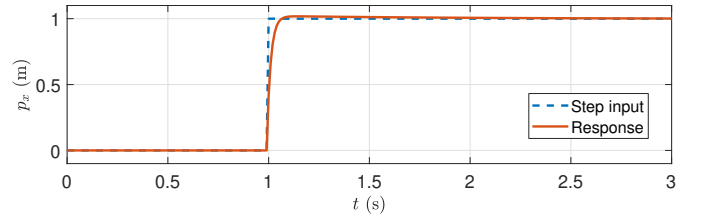


Fig. 10. Position response of the cart with the baseline tuned controller to a step input at $t = 1$ for $\beta_B = 15^\circ$, $k_{p,base} = 10$, $k_{i,base} = 4$ and $k_{d,base} = 7$.

For the remainder of this section, the desired position will be kept at zero ($p_{x,d} = 0$) and input will be supplied to the disturbance input. Figure 11 shows the response to the step disturbance input $f_d(t) = 0.9 \cdot f_{x,max}(\beta_B) \cdot u(t)$. The magnitude of this step represents 90% of the feasible horizontal force at the baseline $\beta_B = 15^\circ$. This step size is selected in this way such that the peak of f_r is close to $f_{r,max}$, without actually saturating. In figure 11, the top plot shows the position over time, which gets deflected before returning to 0. The middle plot shows the thrust generated by the rotor together with its limit. The bottom plot shows the rotor thrust projected onto the horizontal axis, along with the disturbance input for which it is compensating. After some time, the horizontal force projection of the rotor thrust converges to the disturbance force, in order to keep $p_x = 0$.

Figure 11 only shows the response for a single value of β . In order to visualize what happens to the response for different values of β , a large number of response plots are stacked alongside each other to form the surface in figure 12, which shows the position and rotor thrust over time for different values of β . In this surface plot, the magnitude of the disturbance step is scaled in proportion to the feasible horizontal force at the current β , according to $f_d(t) = 0.9 \cdot f_{x,max}(\beta) \cdot u(t)$. Note that the magnitude of the step is a function of β .

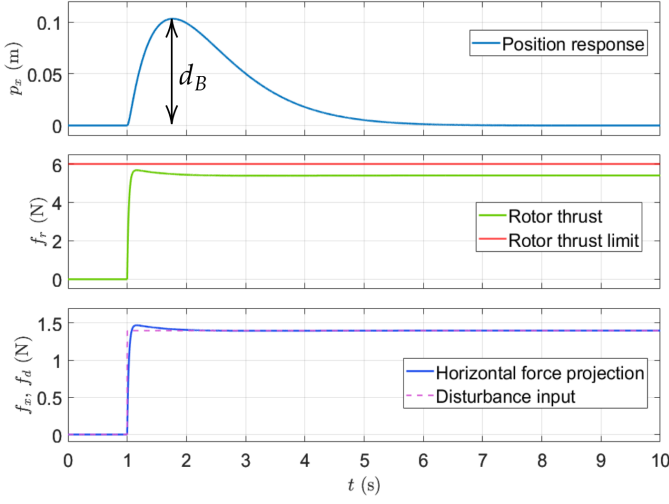


Fig. 11. Response to a step disturbance input with a step size of $0.9 \cdot f_{x,max}(15^\circ)$ at $\beta = 15^\circ$.

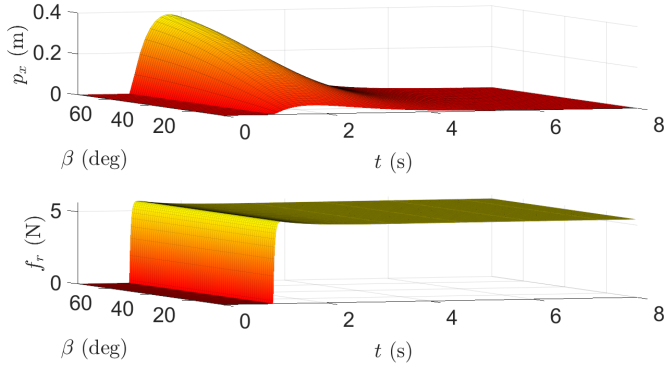


Fig. 12. Position p_x and rotor thrust f_r over time for different values of $\beta \in [15^\circ, 90^\circ]$ in response to a step disturbance with a varying step size of $0.9 \cdot f_{x,max}(\beta)$.

In figure 12, the rotor thrust response is equal for each β , since the magnitude of the disturbance increases in proportion to the increase in projection from f_r to f_x . The increase in disturbance and increase in projection exactly cancel each other. When β increases, the maximum force that can be applied horizontally increases, giving the controller larger authority. But still, the position deflection becomes larger when β and the disturbance become larger. This shows that the system is not fully exploiting the increase in capability that comes from changing β . If the controller were effectively exploiting the system's full potential, the peak in position deflection would remain constant as β and f_d increase. Currently, the controller is unaware of the change in β , so there is no way for it to adapt to the enhanced capabilities.

This demonstrates the limitation of this system: a change in operating point of the system changes certain capabilities, but the controller does not adapt to this operating point, so the system cannot be pushed to its limits. This limitation is also present in the actual multirotor platform, which means that finding a solution for the simplified 1D case can provide valuable insights for utilizing a similar solution for the multirotor. The next section will focus on modifying the controller

such that it does adapt to the operating point.

B. Proposed methods

As demonstrated in the previous section, the controller does not adapt to the operating point, preventing it from pushing the system to its limits. A solution to this problem could be to vary the gains depending on the operating point, formally known as gain scheduling. This section will focus on finding the gains as a function of the operating point such that the performance is maximized for each operating point. The two following points lay the foundation for finding a solution:

- The controller is tuned for a baseline operating point β_B , the gains being $k_{p,base}$, $k_{i,base}$ and $k_{d,base}$. This means that for all gain functions, one point is already known.
- The ratio between the gains will be kept constant. This narrows the problem down from finding three independent functions, to just finding one function.

The metric that will be used to measure if the system is fully exploiting its potential is the peak of the position deflection. When increasing the disturbance in proportion to the maximum feasible force at a certain β , it should be possible to keep the peak position deflection equal for each β , as the capability of the system increases by the same amount as what is asked of it. The goal of the solution is to find the gains that keep the peak deflection d equal to d_B across $\beta \in [5^\circ, 90^\circ]$, where d_B is the baseline deflection (figure 11). Or, in more intuitive language, flattening the curve of peak position deflection versus β .

Method 1. A simple solution to this problem is by looking at the shape of the peak deflection without gain scheduling, as in figure 12. Note how the peak deflection changes with the same shape as $f_{x,max}(\beta)$ with which the magnitude of the disturbance step force is scaled (see figure 9). This gives rise to the idea to scale the gains in proportion to $f_{x,max}(\beta)$ as well, in order to cancel the shape and thus flattening the curve. In this case, the gain functions $k_p(\beta)$, $k_i(\beta)$ and $k_d(\beta)$ will be sinusoidal functions, since $f_{x,max}(\beta)$ is sinusoidal as well. The gain functions can be solved for by substituting the baseline points and assuming that the gains are maximal at $\beta = 90^\circ$, since that is the operating point with maximal control authority. The gain functions become:

$$\begin{aligned} k_p(\beta) &= \frac{k_{p,base}}{\sin(\beta_B)} \cdot \sin(\beta) \\ k_i(\beta) &= \frac{k_{i,base}}{\sin(\beta_B)} \cdot \sin(\beta) \\ k_d(\beta) &= \frac{k_{d,base}}{\sin(\beta_B)} \cdot \sin(\beta) \end{aligned} \quad (18)$$

Where β_B is the baseline β for which the baseline gains have been tuned.

Method 2. The second solution consists of an algorithm resembling a gradient descent. The algorithm searches the best gains for each β by iterating over $\beta \in [5^\circ, 90^\circ]$ with step size $\Delta\beta$. For each beta, the gains are tweaked to match the position deflection response as well as possible to the baseline response. After each variation of the gains, the algorithm will

determine the current peak position deflection d , in order to determine $\Delta d = d - d_B$, which represents the difference between the current deflection d and the baseline deflection d_B (see figure 13). This Δd is the metric that will be used to compare the current response to the baseline response and needs to be minimized. Again, the magnitude of the disturbance step input is $0.9 \cdot f_{x,max}(\beta)$, such that the step size scales in proportion to the maximum feasible horizontal force at the current β .

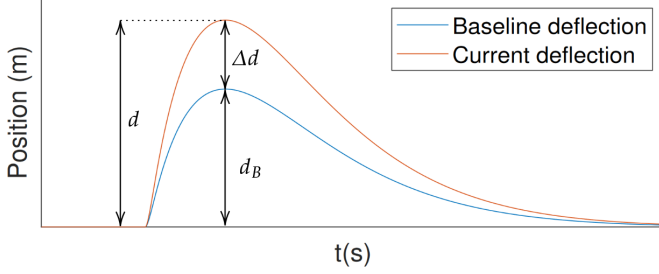


Fig. 13. Overview of d , d_B and Δd .

The algorithm starts by increasing and decreasing the gains with respect to the baseline and checks which direction decreases $|\Delta d|$. It keeps stepping in that direction until the sign of Δd flips, indicating that the deflection error has crossed zero and thus the gains with the smallest $|\Delta d|$ have been found. The step size Δk_p is only defined for k_p , since k_i and k_d can be scaled along with k_p , since the ratio between the gains stays equal. Each step, the gains are scaled as follows:

$$\begin{aligned} k_p &= k_{p,prev} \pm \Delta k_p \\ k_i &= \frac{k_p}{k_{p,base}} k_{i,base} \\ k_d &= \frac{k_d}{k_{d,base}} k_{d,base} \end{aligned} \quad (19)$$

Where $k_{p,prev}$ is k_p from the previous step, and $\frac{k_p}{k_{p,base}}$ represents the factor with which k_p has increased with respect to $k_{p,base}$ and thus sets the amount with which k_i and k_d also need to be scaled. The step Δk_p is either added or subtracted, depending on the direction that has been determined before. The minimal $|\Delta d|$ is achieved in either the step before or after the zero crossing, so they are compared and the gains corresponding to the smallest $|\Delta d|$ are stored.

After finding the gains for a certain β , it varies β by $\Delta\beta$ and repeats the process. However, the algorithm no longer starts at the baseline gains, but at the previously found gains, since the gains at the new β most likely lie closer to the previous set of gains than the baseline gains. The algorithm starts at the baseline β_B and increments β until it hits an upper bound (in this case 90°), after which it returns to β_B and decrements β until it hits a lower bound (in this case 5°).

C. Comparing methods

The gains resulting from both methods proposed in the previous section can be seen in figure 14. The baseline tuning

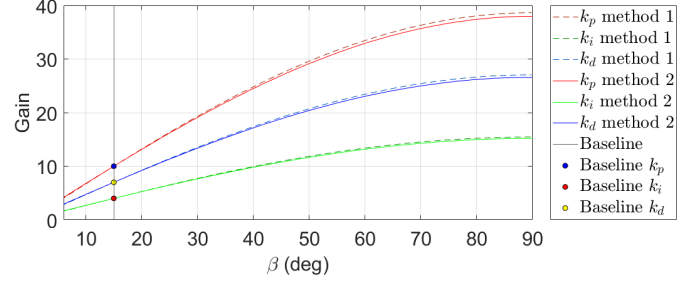


Fig. 14. Resulting gains for Method 1 (simple) and Method 2 (algorithm), along with baseline gains at $\beta_B = 15^\circ$. For the algorithm, $\Delta k_p = 0.05$, $\Delta\beta = 0.5$ and the disturbance step size is $0.9 \cdot f_{x,max}(\beta)$.

is also indicated. At the baseline, the gains resulting from both methods are naturally the same, as they use this as a common starting point. It is clear that the general shape is very similar. However, the gains calculated by the algorithm are higher below the baseline and lower above the baseline in comparison to the gains calculated with the simple method.

Figure 15 shows the position response over time for different values of β , similar to the top surface in 12, but now with the gains scheduled according to both methods. It is clear that the ridge of the deflection has become flat, indicating that the gain scheduling has ensured that the system is operating much closer to its limits at each tilt angle than before.

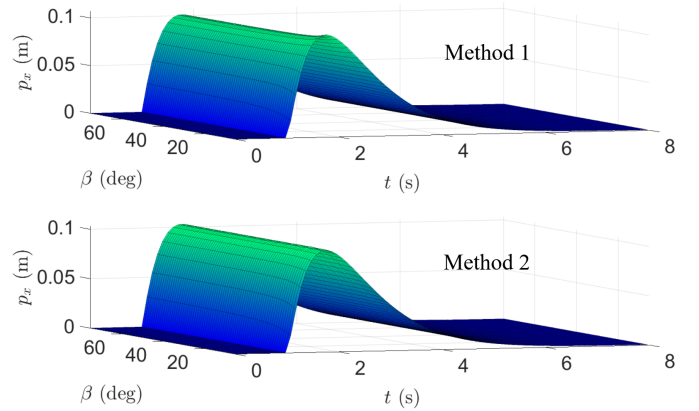


Fig. 15. Position response over time for different values of $\beta \in [6^\circ, 90^\circ]$ in response to a step disturbance with a varying step size of $0.9 \cdot f_{x,max}(\beta)$, with the controller gains scheduled according to Method 1 (top) and Method 2 (bottom).

The numerical metric $\Delta d_{res}(\beta)$ is used to indicate how well the peak deflection d matches to the baseline peak deflection d_B at a certain β and is defined as $\Delta d_{res}(\beta) = d(\beta) - d_B$. This metric essentially describes how flat the ridge of the resulting position over time versus beta surface has become. Figure 16 shows $\Delta d_{res}(\beta)$ for both methods.

Figure 16 clearly shows that the error of the simple method is much greater than the algorithm method. The residual Δd_{res} of the simple method also shows a very distinct shape, which resembles the shape of the function $\frac{1}{\sin(\beta)}$, which coincidentally is the input allocation factor in (17). This shows that Method 1 misses a corrective term that compensates for the change in force projection from f_r to f_x . Method 2 does

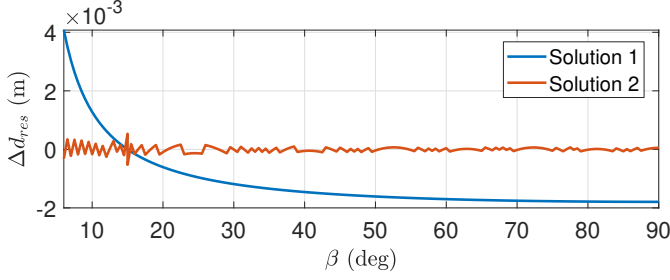


Fig. 16. Residual deflection $\Delta d_{res}(\beta)$ vs β for both methods. For the algorithm, $\Delta k_p = 0.05$ and $\Delta\beta = 0.5$.

not have this problem, as it will always try to match the position response to the baseline, so this effect is automatically accounted for. The residual Δd_{res} of the algorithm method is fairly constant around zero and could theoretically be decreased further by running the algorithm with a smaller Δk_p .

As a metric of the total performance of each method, the mean square of $\Delta d_{res}(\beta)$ is determined for $\beta \in [5^\circ, 90^\circ]$, which is calculated as follows:

$$MS(\Delta d_{res}) = \frac{1}{n} \sum_{i=1}^n (\Delta d_{res}[\beta_i])^2 \quad (20)$$

Where $MS(\Delta d_{res})$ is the mean square of $\Delta d_{res}(\beta)$ for $\beta \in [5^\circ, 90^\circ]$, n is the amount of values of β for which the gains are calculated and β_i is the i^{th} value of β .

Using the data from figure 16, for Method 1 $MS(\Delta d_{res}) = 2.64 \cdot 10^{-6}$ and for Method 2 $MS(\Delta d_{res}) = 1.45 \cdot 10^{-8}$. This clearly indicates that Method 2 performs much better than Method 1.

In conclusion, both methods greatly improve the amount of capability that the system can exploit. While Method 2 performs much better, Method 1 can be deployed with great ease, as it only requires the relation between the feasible force and the operating point and a set of baseline gains to properly scale the relation.

Method 2 also only requires the relation between feasible force and operating point and a set of baseline gains, but it clearly much more involved to implement. Next to this, the quality of the result it delivers depends on the parameters Δk_p and $\Delta\beta$. The smaller these parameters, the more accurate the results but the longer the algorithm takes to run. Although Method 2 is very computationally heavy, the gains only need to be found once, after which they can be stored and simply retrieved from a lookup table.

D. Applying methods to multirotor platform

So far, only the 1D case has been considered. This section will focus on transferring the insights gained from the simplified case to the actual multirotor platform. For the sake of comprehensibility, the remainder of this section will only focus on the position tracking to prevent discussing six dimensions simultaneously. However, these principles can be extended to accommodate the attitude tracking as well. There are a few major differences between the 1D case and the real platform:

- In the 1D case, the mapping from the desired force to the rotor input is one-to-one. For the multirotor, however, this mapping is much more involved. There is not a unique solution for \mathbf{u} that lead to the desired wrench \mathbf{w}_{des} .
- In the 1D case, the gain scheduling methods have been developed under the assumption that the cart does not wish to track a trajectory, it just wants to remain stationary. However, the multirotor wishes to track a certain trajectory next to rejecting disturbances.
- There is no gravity in the 1D case, which is of course present for the multirotor. For the 1D case, when there is no disturbance and the cart is not accelerating, $f_x = 0$ and so $u = 0$. However, for the multirotor, when there is no disturbance and the cart is not accelerating, the vertical force (in world frame) is always equal to $f_z = mg$ in order to compensate for gravity.

There are some important conclusions that can be drawn from the differences above, especially from the second and third. For the 1D cart there is no reference trajectory, so no desired acceleration. Also, there is no gravity. However, for the multirotor, gravity is always acting and there is a certain desired acceleration in accordance with the reference trajectory. In order to achieve this acceleration, the platform needs to apply a force. This force takes up part of the total feasible force, so there is less force that remains for disturbance rejection. The limits visualized by the convex hull in figure 3 represent the limits imposed on the total force \mathbf{f}_{total} that the platform applies. The total force \mathbf{f}_{total} that the platform wishes to generate is a sum of the force required to achieve the desired acceleration, the force required to sustain gravity, and the force required to reject disturbances, as can be seen in the control law in (12). So when part of the force is used to sustain gravity or to track a trajectory, less force remains that can be used for disturbance rejection. The sum of the forces that are known (gravity and trajectory) are denoted by \mathbf{f}_k . The force required for the disturbance rejection is unknown beforehand, as the disturbances cannot be predicted. What can be examined, however, is the force that remains for rejecting disturbances.

The maximum vertical force available for disturbance rejection is denoted by $f_{d,v,max}$ and is defined as the shortest distance along a vertical line (in body frame) from \mathbf{f}_k to the top or bottom surface of the feasible force set. The maximum lateral force available for disturbance rejection is denoted by $r_{d,l,max}$ and is defined as the radius of the largest circle that fits in the feasible force set on the horizontal (in body frame) plane $f_z = (\mathbf{f}_k)_z$. The center of the circle is located at \mathbf{f}_k . Next, a few cases will be presented in order to demonstrate how the gain scheduling could be shaped for the OmniMorph.

- The first simple case considers only sustaining gravity, in which case $\mathbf{f}_k = {}^W\mathbf{R}_B^T[0, 0, mg]^T$.
- Next, the case is considered where there is a non-zero desired vertical acceleration ${}^W\ddot{p}_{z,des}$ along with gravity that needs to be sustained, in which case $\mathbf{f}_k = {}^W\mathbf{R}_B^T[0, 0, m(g + {}^W\ddot{p}_{z,des})]^T$. In this case, the force remaining for vertical disturbance rejection is altered by the additional force required to achieve ${}^W\ddot{p}_{z,des}$.

However, when introducing a non-zero desired vertical acceleration, not only the vertical bounds are influenced, but the lateral bounds also change. The reason is that the center of circle with which the lateral bounds are determined is shifted vertically in the feasible force set, so its radius can change. So even though only ${}^W\ddot{p}_{z,des}$ is introduced, this can also influence the lateral limits.

- The final case considers sustaining gravity, along with both a non-zero desired vertical acceleration and a non-zero desired lateral acceleration, such that $\mathbf{f}_k = {}^W\mathbf{R}_B^T m[{}^W\ddot{p}_x, {}^W\ddot{p}_y, g + {}^W\ddot{p}_{z,des}]^T$. In this case, even though only the desired lateral acceleration has changed with respect to the previous case, $f_{d,v,max}$ will also change. This is because the lateral components of \mathbf{f}_k change, so the distance along a vertical line (in body frame) from \mathbf{f}_k to the top or bottom surface of the feasible force set can also change. The lateral force bound also changes, as the center point of the circle has moved laterally, so the radius of the largest inner circle will change.

The cases above demonstrate the complex interplay between the known force \mathbf{f}_k and the vertical and lateral force that is available for disturbance rejection. In addition to this, the shape of the feasible force set is a function of α . This means that $r_{d,l,max}$ and $f_{d,v,max}$ are a function of both α and \mathbf{f}_k .

As demonstrated for the 1D case, the gains will be closely related to the force that is available for disturbance rejection. For the OmniMorph, this implies that the gains will be closely related to $r_{d,l,max}$ and $f_{d,v,max}$, which implies that the gains will also be a function of both α and \mathbf{f}_k . So instead of only determining the gains for a range of α such as in the 1D case, the gains now need to be found for the each point in the feasible force set for each value of α .

In order to implement the methods for finding the gains from the 1D case, the relation between the maximum feasible force available for disturbance rejection and α is required. For the 1D case this is just one relation, but now there are two relations: $r_{d,l,max}$ versus α and $f_{d,v,max}$ versus α . Not only are there now two relations, but these relations are different for each point \mathbf{f}_k in the feasible force set.

For Method 1, the relations are required to scale the gains, as well as to scale the magnitude of the disturbance input in order to quantitative measure the performance. Method 2 requires the relations in order to scale the magnitude of the disturbance input while running the algorithm, as well as for the quantitative performance measurement. These relations can be found for a certain point \mathbf{f}_k as follows:

- $f_{d,v,max}$ versus α can be found by finding the shortest distance along a vertical line (in body frame) from \mathbf{f}_k to the top or bottom surface of the feasible force set, for each value of α .
- $r_{d,l,max}$ versus α can be found by finding the radius of the largest circle that fits inside the feasible force set, on the horizontal plane ${}^W\mathbf{f}_z = (\mathbf{f}_k)_z$, for each value of α .

An important note is that $r_{d,l,max}$ and $f_{d,v,max}$ actually represent a quite conservative measure of the force bounds. For $r_{d,l,max}$, this is the case since the inner circle will limit

its radius to the shortest distance from the center of the circle to any point on the surface of the feasible force set. For example, consider the case where the platform is accelerating in the $+x$ direction. This means that the platform can reject less disturbance in the $-x$ direction, since the circle has moved closer to the surface of the feasible force set in the $+x$ direction. However, the platform now actually has more force available for disturbance rejection in the $+x$ direction. Regrettably, by using the simple measure $r_{d,l,max}$, the disturbance rejection will be limited in all lateral directions, even though some directions could perform much better. The same holds for $f_{d,v,max}$: this measure only considers the shortest distance from \mathbf{f}_k to either the upper or lower surface of the feasible force set, meaning that the disturbance rejection in both directions is limited to the performance of the direction with the least capability. A possible solution to overcome this limitation, and to further push the OmniMorph to its limits, would be to modify the controller such that it has two sets of gains for each DOF. In this way, the performance in one direction does not hinder the performance in the other direction. This needs to be studied in more detail and is left open as possible future work.

Due to time constraints imposed on this thesis, adapting the gain scheduling to be a function of both α and \mathbf{f}_k is unfortunately not feasible. However, in order to demonstrate the effectiveness of scheduling the gains of the OmniMorph, the next section will focus on scheduling the gains as a function of α for a single point \mathbf{f}_k . This point is chosen according to the conditions under which the platform will likely often be used: level flight without acceleration. This could be stationary hover or level flight with a constant velocity, the main point being that $\ddot{\mathbf{p}}_{des} = \mathbf{0}$ such that no force is used to track a trajectory, but only to sustain gravity and reject disturbances. Due to the ease of implementation, Method 1 will be used to find the gains.

E. Scheduled position tracking for level, unaccelerated flight

This section will focus on finding the gains as a function of α for a single point $\mathbf{f}_k = {}^W\mathbf{R}_B^T[0, 0, mg]^T$ in the feasible force set, where $\ddot{\mathbf{p}}_{des} = \mathbf{0}$, so the platform is not intending to accelerate. Besides this, the platform is kept level. If the attitude of the platform would not be kept level, the gravitational force would be split into a vertical component and a lateral component in a ratio determined by ${}^W\mathbf{R}_B$. In order to prevent this complication, the platform will simply be kept level for now, such that \mathbf{f}_k will always appear at the same location within the feasible force set, since ${}^W\mathbf{R}_B^T = \mathbf{I}$.

Just like for the 1D case, the platform is tuned for the baseline angle $\alpha_B = 15^\circ$ such that the rotor thrusts are close to saturation, but do not actually saturate. Figure 17 shows the position \mathbf{p}_x over time for different values of α in response to a step disturbance force along the x-axis, similar to figure 12. The magnitude of the disturbance step force is set to $0.9 \cdot r_{d,l,max}(\alpha)$. Note that the magnitude is a function of α . The result is similar to the 1D case: the peak deflection becomes larger for a larger disturbance step at larger α , once again meaning that the system is not fully exploiting its

capabilities. The response of p_z over time for different values of α in response to a downward step disturbance force along the z-axis can be seen in figure 18, for which the magnitude of the disturbance is set to $0.9 \cdot f_{d,v,max}(\alpha)$. The disturbance force is applied downward, as this is the direction in which the available force is actually limited, since part of the feasible vertical force is already used for gravity compensation.

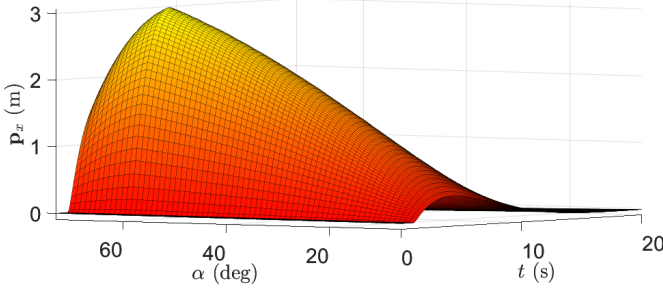


Fig. 17. Response of p_x over time for $\alpha \in [6^\circ, 73^\circ]$ in response to a lateral (in x-direction) step disturbance with a varying step size of $0.9 \cdot r_{d,l,max}(\alpha)$, with the controller gains equal to the baseline gains at each α .

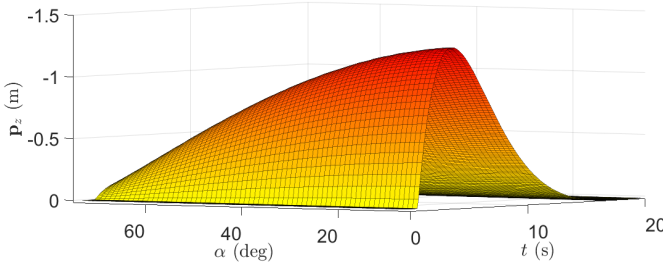


Fig. 18. Response of p_z over time for $\alpha \in [6^\circ, 73^\circ]$ in response to a vertical step disturbance with a varying step size of $0.9 \cdot f_{d,v,max}(\alpha)$, with the controller gains equal to the baseline gains at each α . Note that the z-axis is inverted in this plot, the deflection is downward.

In order to find the gain functions, Method 1 will be used. For the 1D case, the gains can be directly defined as a function of the maximum feasible force, as the maximum feasible force itself is a simple, known function. However, in the case of the OmniMorph, the mapping from the input u to the wrench w is not one-to-one, but is computed by the more involved input allocation. This means that the maximum feasible force cannot be expressed easily as a function of α , but needs to be computed numerically. This difficulty is encountered before in section II-D, when calculating the feasible force set. To overcome this complication, $f_{d,v,max}$ and $r_{d,l,max}$ will be computed numerically for a discretized range of α , such that the resulting data can be interpolated.

The result of the computation of $r_{d,l,max}$ and $f_{d,v,max}$ for a discretized range of α can be seen in figure 19. The data of $r_{d,l,max}$ is similar to $r_{f_{xy}}$ in figure 5, but is computed at $f_z = mg$ instead of $f_z = 0$. The data of $f_{d,v,max}$ is similar to $f_{z,max}$ in figure 5, but is shifted down by mg . An important difference can be observed between $r_{f_{xy}}$ and $r_{d,l,max}$, which is that $r_{f_{xy}}$ keeps increasing for increasing α , but $r_{d,l,max}$, which includes gravity, starts decreasing beyond $\alpha = 61.0^\circ$. Note that the shape of $r_{d,l,max}$ and $f_{d,v,max}$ versus alpha depends on

which point f_k it is calculated for, figure 19 simply shows a single variation of the relations for $f_k = {}^W R_B^T [0, 0, mg]^T$.

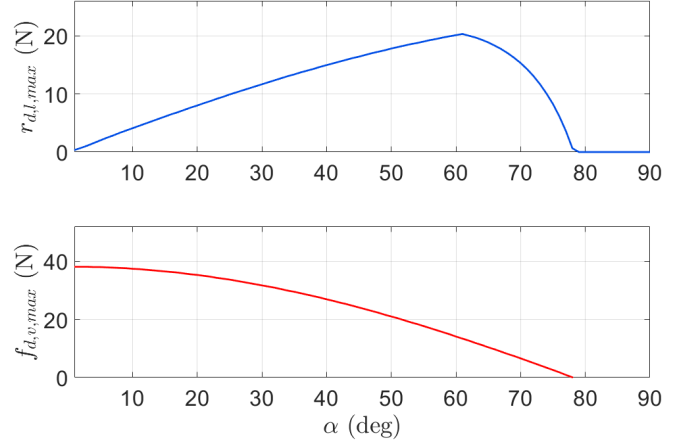


Fig. 19. Maximum lateral force available for disturbance rejection $r_{d,l,max}$ versus α (top) and maximum vertical force available for disturbance rejection $f_{d,v,max}$ versus α (bottom) at $f_k = [0, 0, mg]$. At $\alpha = 61.0^\circ$, f_{xy} starts dropping and at $\alpha = 78.2^\circ$, it is no longer possible to hover.

The gains can be found as a function of α by interpolating $r_{d,l,max}$ and $f_{d,v,max}$ and scaling them such that the gains have the proper baseline value at $\alpha_B = 15^\circ$. The resulting gains can be seen in figure 20. The diagonal gain matrices are defined as functions of α :

$$\begin{aligned} K_p(\alpha) &= \text{diag}(k_{p,xy}(\alpha), k_{p,xy}(\alpha), k_{p,z}(\alpha)) \\ K_i(\alpha) &= \text{diag}(k_{i,xy}(\alpha), k_{i,xy}(\alpha), k_{i,z}(\alpha)) \\ K_d(\alpha) &= \text{diag}(k_{d,xy}(\alpha), k_{d,xy}(\alpha), k_{d,z}(\alpha)) \end{aligned} \quad (21)$$

Where the gains for the x and y direction are equal, as the the lateral force bound is found by means of a symmetric circle.

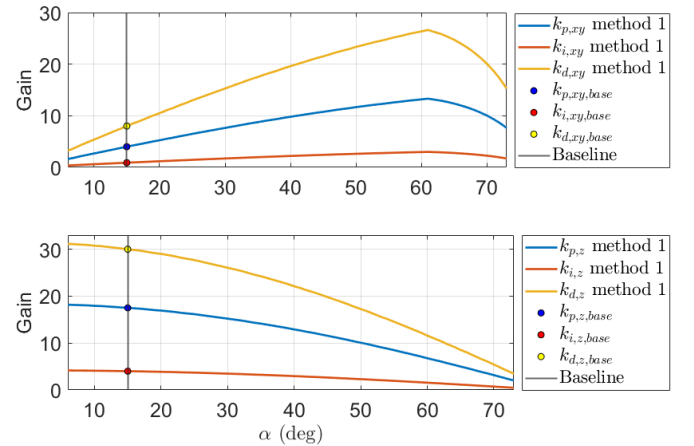


Fig. 20. Lateral and vertical gains for $\alpha \in [6^\circ, 73^\circ]$, based on $r_{d,l,max}$ and $f_{d,v,max}$ according to Method 1, such that at the baseline α_B , the gain functions match to the baseline gains. The baseline gains are $k_{p,xy,base} = 4$, $k_{i,xy,base} = 0.9$, $k_{d,xy,base} = 8$, $k_{p,z,base} = 17.5$, $k_{i,z,base} = 4$ and $k_{d,z,base} = 30$.

Figure 21 shows the position in x direction over time for different values of α , similar to the surface in figure 17, but now with the gains scheduled in proportion to $r_{d,l,max}$, according

to Method 1. It is clear that the ridge of the deflection has become much flatter, indicating that the gain scheduling has successfully ensured that the system is operating much closer to its limits at each tilt angle than before. Figure 22 shows the gain scheduled response of the position in z direction over time for different values of α . The ridge of the deflection in the z direction has also become a lot flatter than the ridge with constant gains in figure 18, indicating that much more of the capability is exploited.

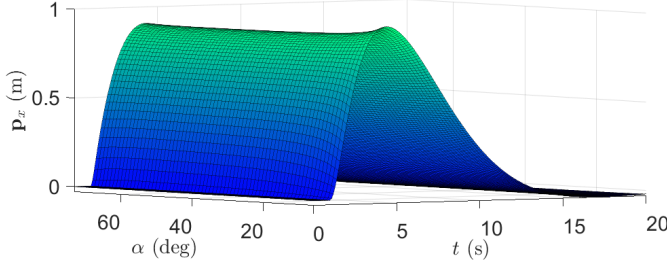


Fig. 21. Response of p_x over time for $\alpha \in [6^\circ, 73^\circ]$ in response to a lateral step disturbance (in x -direction) with a varying step size of $0.9 \cdot r_{d,l,max}(\alpha)$, with the controller gains scheduled in proportion to $r_{d,l,max}$ according to Method 1.

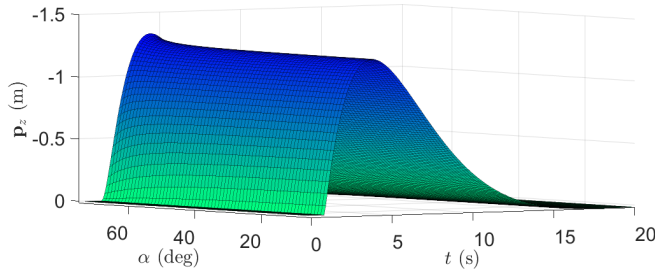


Fig. 22. Response of p_z over time for $\alpha \in [6^\circ, 73^\circ]$ in response to a vertical step disturbance with a varying step size of $0.9 \cdot f_{d,v,max}(\alpha)$, with the controller gains scheduled in proportion to $f_{d,v,max}$ according to Method 1. Note that the z -axis is inverted in this plot, the deflection is downward.

Similar to the 1D case, Method 1 does not fully flatten the ridge of the position deflection: there is an increase in deflection when the feasible force and the magnitude of the applied disturbance approach zero. This is visible in both figure 21 for low values of α in and figure 22 for high values of α . This is in line with the results in figure 16 for the 1D case, which also shows that the residual deflection becomes large as the feasible force and the disturbance magnitude approach zero. Method 2 is expected to be able to solve this deviation, but due to limited time, Method 2 has not yet been implemented to integrate with the OmniMorph model, so for now it is unclear how Method 2 performs for the OmniMorph.

F. Strategy for extending methods

Unfortunately, due to time constraints, not all ideas that have been presented have been implemented, so there are no full results available. However, the previous section has demonstrated that Method 1 is successful in exploiting the capabilities of the OmniMorph, albeit for a single point f_k . The gain functions can be extended to become a function of f_k

by repeating exactly what was done in the previous section, each time for a different point f_k in the feasible force set. This results in a 3-dimensional grid of gain functions, between which can be interpolated, just like $r_{d,l,max}$ and $f_{d,v,max}$ have been interpolated after calculating them for a discretized range.

Up to this point, only the position tracking has been considered. However, the principles should be transferable to the attitude control as well. However, there may be some complications due to the nonlinearities present in the attitude dynamics, so this needs to be studied in more depth.

V. CONCLUSION

This thesis has studied the design of a linear controller for the OmniMorph that automatically adapts to the operating point to push the platform to its limits. A model of the OmniMorph has been used to study the capabilities and limitations of the platform at different tilt angles. This model was the foundation for designing a linear controller that allows the platform to track a time-varying 6D trajectory.

In order to develop gain scheduling methods, a simplified 1D case was analyzed first, which provided valuable insights into the possible solutions. Based on this analysis, two gain scheduling methods were proposed, both of which were effective at pushing the platform to its limits. The simple method is easier to deploy, but offers limited performance. The optimization algorithm is more complex, but it offers much better performance. After developing these methods, the adaptation of these techniques to the OmniMorph has been studied. Although the techniques have not yet been fully adapted to accommodate the OmniMorph, the partial results show promising progress.

VI. FUTURE WORK

Due to time constraints, the gain scheduling methods have not yet been fully adapted to the OmniMorph. The gain functions for the OmniMorph can be further expanded such that they also adapt to the force that is already used for compensating gravity and following a trajectory. Besides this, the gain scheduling can be expanded to encompass the attitude controller as well. Once each channel of the controller is gain scheduled, the behaviour of the complete system can be studied.

An interesting idea that can be studied further for pushing the OmniMorph to its limits is to employ less conservative methods for quantifying the maximum available force for disturbance rejection. This allows the controller to perform optimally in each direction, without restrictions that are imposed by limitations of other directions.

ACKNOWLEDGMENT

I would like to sincerely thank Antonio Franchi and Youssef Aboudorra for their roles in my committee, and for the invaluable feedback that I have received. I would also like to thank Wouter Kuijper for his role as a member of my committee. Also, special thanks go to Youssef Aboudorra for providing guidance and feedback throughout the project.

REFERENCES

- [1] F. Ruggiero, V. Lippiello, and A. Ollero, “Aerial Manipulation: A Literature Review,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 1957–1964, July 2018. Conference Name: IEEE Robotics and Automation Letters.
- [2] D. Brescianini and R. D’Andrea, “Design, modeling and control of an omni-directional aerial vehicle,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3261–3266, May 2016.
- [3] S. Park, J. Her, J. Kim, and D. Lee, “Design, modeling and control of omni-directional aerial robot,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1570–1575, Oct. 2016. ISSN: 2153-0866.
- [4] M. Ryll, D. Bicego, M. Giurato, M. Lovera, and A. Franchi, “FAST-Hex—A Morphing Hexarotor: Design, Mechanical Implementation, Control and Experimental Validation,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, pp. 1244–1255, June 2022. Conference Name: IEEE/ASME Transactions on Mechatronics.
- [5] Y. Aboudorra, C. Gabellieri, Q. Sablé, and A. Franchi, “Modelling, Analysis and Control of OmniMorph:an Omnidirectional Morphing Multi-rotor UAV,” May 2023. arXiv:2305.16871 [cs, eess].
- [6] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, “Design of multirotor aerial vehicles: A taxonomy based on input allocation,” *The International Journal of Robotics Research*, vol. 40, pp. 1015–1044, Aug. 2021. Publisher: SAGE Publications Ltd STM.
- [7] A. Franchi, R. Carli, D. Bicego, and M. Ryll, “Full-Pose Tracking Control for Aerial Robotic Systems With Laterally Bounded Input Force,” *IEEE Transactions on Robotics*, vol. 34, pp. 534–541, Apr. 2018. Conference Name: IEEE Transactions on Robotics.
- [8] D. A. Santos and J. A. Bezerra, “On the control allocation of fully actuated multirotor aerial vehicles,” *Aerospace Science and Technology*, vol. 122, p. 107424, Mar. 2022.