

# Design and optimal control of a tiltrotor micro aerial vehicle for efficient omnidirectional flight

Mike Allenspach<sup>1\*</sup>, Karen Bodie<sup>1\*</sup>, Maximilian Brunner<sup>1\*</sup>, Luca Rinsoz<sup>1</sup>, Zachary Taylor<sup>1</sup>, Mina Kamel<sup>1</sup>, Roland Siegwart<sup>1</sup> and Juan Nieto<sup>1</sup>

## Abstract

Omnidirectional micro aerial vehicles are a growing field of research, with demonstrated advantages for aerial interaction and uninhibited observation. While systems with complete pose omnidirectionality and high hover efficiency have been developed independently, a robust system that combines the two has not been demonstrated to date. This paper presents the design and optimal control of a novel omnidirectional vehicle that can exert a wrench in any orientation while maintaining efficient flight configurations. The system design is motivated by the result of a morphology design optimization. A six degrees of freedom optimal controller is derived, with an actuator allocation approach that implements task prioritization, and is robust to singularities. Flight experiments demonstrate and verify the system's capabilities.

## Keywords

Aerial robotics, optimal control, omnidirectional MAV, tiltrotor, design optimization

## Multimedia

A supporting video showcasing experiments can be accessed at <https://youtu.be/mBi9mOQaZzQ>

## 1 Introduction

Omnidirectional micro aerial vehicles (MAVs) present a compelling solution for future applications of aerial robots. Full actuation allows for *force-omnidirectionality* (force and torque tracking in six DOF), decoupling the translational and rotational dynamics of the system, and permitting stable interaction with the environment. Such a system proposes significant functional advantages over the traditional underactuated MAV, which provides only four controllable degrees of freedoms (DoFs) by nature of the aligned propeller axes.

With the added criterion that force and torque in any direction must compensate for the system's mass, the result is complete *pose-omnidirectionality*, where a system can achieve uninhibited aerial movement and robust tracking of six DOF trajectories. This extension offers a unique advantage for aerial filming and 3D mapping, as well as configuration-based navigation in constrained environments.

A dominant struggle in the development of MAVs is reaching a compromise between performance and efficiency. For robust aerial interaction and high control authority in omnidirectional flight, performance can be represented by the force and torque control volumes of a system. For pose-omnidirectional platforms, the force envelope must exceed gravity in all directions with an additional buffer to maintain dynamic movement. Countering this performance goal is the desire for high efficiency and longer flight times, which is compromised in systems that generate high internal forces (thrust forces which counteract each other), or add additional weight for actuation.

Within the past 5 years, substantial growth has occurred in the field of fully actuated omnidirectional MAVs, from the emergence of these systems to their application in realistic inspection scenarios. We consider two dominant categories of platform actuation: fixed rotor, and tiltrotor platforms. The fixed rotor OMAV offers a mechanically simple design to achieve full actuation, creating a thrust vector by varying propeller speeds of fixedly tilted rotors. However, any interaction wrench exerted on the environment from a stable hovering pose creates a proportionately significant amount of internal force, which directly detracts from flight efficiency. As a result, orienting the propellers to prefer efficient hover flight and a higher payload directly reduces the capability to generate lateral force for interaction. Several fixed rotor platforms that achieve full pose omnidirectionality have been developed in recent years (Brescianini and D'Andrea 2016; Park et al. 2018; Staub et al. 2018), while other platforms achieve omnidirectional wrench generation from a defined hover orientation (Ryll et al. 2018; Wopereis et al. 2018; Ollero et al. 2018). Other concepts have extended the theory of fixed rotor platforms (Tognon and Franchi 2018), as interest in these systems grows.

---

\* Authors contributed equally to the work

<sup>1</sup>ETH Zürich, Switzerland

## Corresponding author:

Karen Bodie, Maximilian Brunner  
Autonomous Systems Lab, ETH Zürich,  
Leonhardstrasse 21, 8092 Zurich, Switzerland.

Email: {karen.bodie, maximilian.brunner}@mavt.ethz.ch

Tiltrotor platforms can individually tilt rotor groups with additional actuation, and can achieve optimal hover efficiency in the absence of external disturbances, when all propeller thrust vectors are aligned against gravity. Tiltrotor systems have been demonstrated in the form of a quadrotor (Falconi and Melchiorri 2012; Ryll et al. 2015) with limited roll and pitch, and more recently in the form of a hexarotor (Kamel et al. 2018). These platforms achieve force omnidirectionality with high hover efficiency in specific poses, at the cost of additional inertia and mechanical complexity, but force-omnidirectionality is marginal or impossible in some body orientations. Another concept that reduces complexity by coupling the tilt axes to one or two motors has been evaluated in simulation (Ryll et al. 2016; Morbidi et al. 2018), providing versatility in force generation and efficiency, but without pose-omnidirectionality.

When designing a system to target the application of aerial workers, versatility plays a large role. Navigation efficiency and payload capacity are often required, while new tasks of omnidirectional interaction demand high force capabilities in all directions. The tilt-rotor OMAV provides a promising solution, having omnidirectional capabilities while maintaining the ability to revert to an efficient hover. While additional motors add complexity and weight, we can take advantage of the highly overactuated system to prioritize tasks in the allocation of actuator commands.

In this paper we present the design of an omnidirectional tiltrotor dodecacopter, a jerk-level LQRI optimal controller for six DOF control of a floating base system, and task prioritization in actuator allocation. Various experiments show controller performance, and demonstrate handling of singularities and individual tilt arm control.

## 1.1 Contributions

- Design of a novel OMAV that achieves both force- and pose-omnidirectionality with highly dynamic capabilities, while maintaining high efficiency in hover. A tiltrotor design optimization tool is described and made available open source.
- A full state optimal controller is designed in the form of a jerk-level LQRI, which takes into account tilt-motor commands.
- An allocation strategy is developed to prioritize tracking in 6 DOF, while completing additional tasks in the null space of the overactuated system.
- Experiments demonstrate the tracking performance of the LQRI as compared to a benchmark PID controller. Further tests show singularity handling and cable unwinding as secondary tasks while tracking a full pose trajectory.

## 1.2 Outline

This paper extends upon the work presented at the 2018 International Symposium on Experimental Robotics (ISER 2018) (Bodie et al. 2018) and (Bodie et al. 2019), and is structured as follows. Section 2 states assumptions and notation, and derives a general model of the tiltrotor OMAV. Section 3 describes the parametric design the system subject to a cost function, and compares the system's theoretical capabilities to other state-of-the-art

concepts. Section 4 details the LQRI optimal control implementation, and describes the allocation of 6 DOF wrench tracking commands, and additional secondary tasks, to 18 independent actuators. Section 5 describes the hardware of the prototype system and the experimental setup. In section 6, results from experimental flights are presented and discussed. Finally, section 7 offers concluding remarks and directions for future work.

## 2 System Modeling

In this section we present definitions and notation used throughout the paper, define assumptions, and develop a generalized tiltrotor MAV model. We further discuss singularity cases that arise in generalized tiltrotor systems.

### 2.1 Definitions and notation

In the present work, we consider a general rigid-body model for a tilt-rotor aerial vehicle. Refer to table 1 for definitions of common symbols used throughout the paper. We use following conventions for coordinate frames:

- Inertial world frame  $\mathcal{F}_W$
- Body frame  $\mathcal{F}_B$  attached to the COM of the MAV with origin  $O_B$
- Rotor frame  $\mathcal{F}_{r_i}$

Figure 1 illustrates the frames and angles that we use when describing a tiltrotor MAV. The rotating axis of each rotor group,  $x_{R_i}$ , is defined as the axis of the arm coming out of the main body, and intersects the body origin,  $O_B$ . This arm may be positioned at arbitrary angles  $\gamma_i, \beta_i$  which respectively define the rotation about the  $z_B$  axis, and inclination from the  $z_B$ -plane.

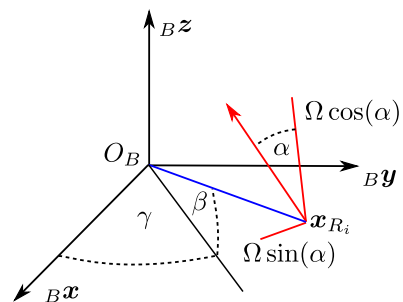


Figure 1. Illustration of arm angles.

$\mathbf{R}_{WB} \in SO(3)$  denotes a rotation matrix expressing the orientation of  $\mathcal{F}_B$  with respect to  $\mathcal{F}_W$ .

${}^B\boldsymbol{\omega}_{WB} \in \mathbb{R}^3$  denotes the angular velocity of  $\mathcal{F}_B$  with respect to  $\mathcal{F}_W$ , expressed in  $\mathcal{F}_B$ . We can then express the orientation kinematics of a frame  $\mathcal{F}_B$  as

$$\dot{\mathbf{R}}_{WB} = \mathbf{R}_{WB} [{}^B\boldsymbol{\omega}_{WB}]_{\times} \quad (1)$$

with  $[\cdot]_{\times} \in SO(3)$  as the skew symmetric matrix of the vector  $\cdot \in \mathbb{R}^3$ . The angular acceleration of the body frame is written as  ${}^B\boldsymbol{\psi}_{WB}$  and the angular jerk as  ${}^B\boldsymbol{\zeta}_{WB}$ .

Furthermore, we denote the position of the MAV in the frame  $\mathcal{F}_W$  as  ${}^W\mathbf{p}$ . The body velocity, acceleration, and jerk are described as  ${}^W\mathbf{v}$ ,  ${}^W\mathbf{a}$ , and  ${}^W\mathbf{j}$ , respectively.

The time derivative of a vector  ${}_B \mathbf{a}$  in a rotating frame  $\mathcal{F}_B$  is written as follows

$$\frac{d}{dt} ({}_B \mathbf{a}) = {}_B (\dot{\mathbf{a}}) \quad (2)$$

and can be expanded to

$${}_B (\dot{\mathbf{a}}) = {}_B \dot{\mathbf{a}} + [{}_B \boldsymbol{\omega}_{WB}]_{\times} {}_B \mathbf{a} \quad (3)$$

## 2.2 Assumptions

The following assumptions are adopted to simplify the model:

- The entire platform can be modeled as a rigid body.
- Thrust and drag torques are proportional to the square of the rotors angular speed, and rotors are able to achieve desired speeds  $\omega_i$  with negligible transients.
- The primary axes of the system correspond with the principal axes of inertia. Products of inertia are considered negligible.
- The dynamics of tilt motors are independent of the rotational speed of rotors.
- The tilting axis of each propeller group intersects the body origin.
- The thrust and drag torques produced by each rotor are independent, i.e. there is no airflow interference.

## 2.3 Rigid body model

The system dynamics are derived by the Newton-Euler approach under the stated assumptions, resulting in the following equations of motion expressed in the body-fixed frame:

$$\begin{bmatrix} m\mathcal{I}_3 & 0 \\ 0 & \mathbf{J}_B \end{bmatrix} \begin{bmatrix} {}_B \dot{\mathbf{v}} \\ {}_B \dot{\boldsymbol{\omega}}_{WB} \end{bmatrix} = - \begin{bmatrix} [{}_B \boldsymbol{\omega}_{WB}]_{\times} m {}_B \mathbf{v}_B \\ [{}_B \boldsymbol{\omega}_{WB}]_{\times} \mathbf{J}_B {}_B \boldsymbol{\omega}_{WB} \end{bmatrix} + \begin{bmatrix} {}_B \mathbf{f} \\ {}_B \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} m {}_B \mathbf{g} \\ 0 \end{bmatrix} \quad (4)$$

where  $m$  and  $\mathbf{J}$  are the mass and inertia of the MAV,  $[{}_B \mathbf{f}^T \quad {}_B \boldsymbol{\tau}^T]^T$  are forces and torques resulting from the actuation, and  $m {}_B \mathbf{g}$  is the force resulting from gravity.

## 2.4 Aerodynamic force model

For any geometry of a tiltrotor MAV we can find a relation between the individual rotor forces  $\mathbf{f}_{r,i}$  and torques  $\boldsymbol{\tau}_{r,i}$  and the total body force and torque vector  ${}_B \mathbf{w}$  that is generated at its center of mass.

As previously presented by Kamel et al. (2018), we assume that the force and torque produced by a rotor can be described as

$$f_i = c_F \omega_i^2 \quad (5a)$$

$$\tau_i = c_M \omega_i^2 \quad (5b)$$

Since both the force and torque are directly proportional to the squared rotor speed, we define the vector of squared rotor speeds:

$$\boldsymbol{\Omega} = \begin{bmatrix} \Omega_1 \\ \vdots \\ \Omega_n \end{bmatrix} = \begin{bmatrix} \omega_1^2 \\ \vdots \\ \omega_n^2 \end{bmatrix} \quad (6)$$

Considering a tiltrotor model,  $\alpha$  represents the active rotational joint angle about an arm axis. We define the *instantaneous allocation matrix*  $\mathbf{A}_\alpha$  that is a nonlinear function of the current tilt angles  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]$  with  $n$  being the number of tilt arms. Using a matrix multiplication we can relate the body force and torque directly with the squared rotor speeds:

$${}_B \mathbf{w} = \begin{bmatrix} {}_B \mathbf{f} \\ {}_B \boldsymbol{\tau} \end{bmatrix} = \mathbf{A}_\alpha \boldsymbol{\Omega} \quad (7)$$

For convenience we can also write this equation with a static allocation matrix  $\mathbf{A}$  that is independent of the varying tilt angles and that only depends on the geometry of the MAV:

$${}_B \mathbf{w} = \mathbf{A} \tilde{\boldsymbol{\Omega}} \quad \mathbf{A} \in \mathbb{R}^{6 \times n}, \tilde{\boldsymbol{\Omega}} \in \mathbb{R}^{2n} \quad (8)$$

$$\tilde{\boldsymbol{\Omega}} = \begin{bmatrix} \sin(\alpha_i) \Omega_i \\ \cos(\alpha_i) \Omega_i \\ \vdots \\ \sin(\alpha_n) \Omega_n \\ \cos(\alpha_n) \Omega_n \end{bmatrix} \quad \forall i \in \{1 \dots n\} \quad (9)$$

The elements of the vector  $\tilde{\boldsymbol{\Omega}}$  correspond to the lateral and vertical components of each squared rotor speed in the respective rotor unit frame.

Refer to appendix A.5 for a general formulation of the static allocation matrix  $\mathbf{A}$ .

## 2.5 Tilt motor model

As proposed by Kamel et al. (2018) we model the tilt angle dynamics using a first order damped system, i.e.

$$\dot{\alpha}_i = \frac{1}{\tau_\alpha} (\alpha_{i,ref} - \alpha_i) \quad (10)$$

with  $\alpha_{i,ref}$  being the reference,  $\alpha_i$  the actual tilt angle of arm  $i$ , and  $\tau_\alpha$  the time constant of the tilting motion. This accounts for dynamic effects as well as physical limitations of the servo motors that are otherwise unmodeled.

## 2.6 Singularity cases for tiltrotor MAVs

As previously identified by Bodie et al. (2018), an OMAV can encounter two types of singularity cases that result from the allocation model.

The first type occurs when the tilt angles  $\alpha_i$  are aligned in a way that leads to a rank reduction of the instantaneous allocation matrix  $\mathbf{A}_\alpha$ . This corresponds to a state in which instantaneous controllability of select forces and torques is lost due to tilt motor delay. We refer to this type of singularity as a *rank reduction singularity*. It has been studied thoroughly by Morbidi et al. (2018).

The second type, which we refer to as a *kinematic singularity*, occurs when rotor thrusts cannot contribute to the desired body wrench  $\mathbf{w}_B$ , which leads to tilt angles not being uniquely defined. This condition has been described by Bodie et al. (2018).

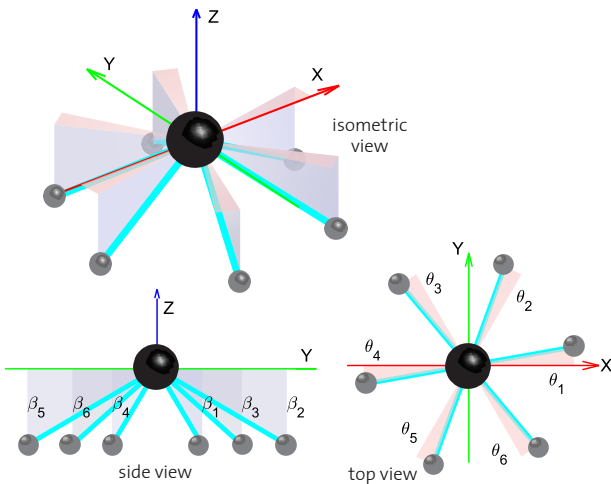
## 3 Tiltrotor morphology design

The tiltrotor MAV offers compelling advantages toward the goal of versatility in force exertion and hover efficiency.

**Table 1.** Symbols and definitions for a tilt-rotor aerial vehicle.

Symbol	Definition
$\mathcal{F}_W : \{O_W, x_W, y_W, z_W\}$	inertial frame: origin and primary axes
$\mathcal{F}_B : \{O_B, x_B, y_B, z_B\}$	robot body-fixed frame: origin and primary axes
$\mathcal{F}_{r_i} : \{O_{r_i}, x_{r_i}, y_{r_i}, z_{r_i}\}$	$i^{\text{th}}$ rotor unit frame: origin and primary axes
$R_{ab}$	Orientation of $\mathcal{F}_b$ expressed in $\mathcal{F}_a$
$A, A_\alpha$	Static allocation matrix and instantaneous allocation matrix, respectively
$m$	mass
$J$	inertia matrix
$r_{com}$	center of mass offset, given in the body frame
$\alpha$	vector of tilt angles
$\omega$	vector of angular velocities of the rotors
$\Omega$	vector of squared rotor speeds
$\tilde{\Omega}$	vector of lateral and vertical components of $\Omega$
$f$	force vector
$\tau$	torque vector
$Bw$	total actuation wrench in the body frame
$wg = [0 \ 0 \ g]^\top$	gravity acceleration vector, $g = 9.81 \text{ m s}^{-2}$
$wP$	position
$wv$	linear velocity
$wa$	linear acceleration
$wj$	linear jerk
$B\omega_{WB}$	angular velocity
$B\psi_{WB}$	angular acceleration
$B\zeta_{WB}$	angular jerk

However, despite these advantages, tiltrotor systems also come with drawbacks such as additional actuation mass and complexity, limited rotation due to possible arm cable windup, and the presence of singularity cases that are not encountered in a fixed rotor system. Therefore, morphology design is important to ensure that the resulting platform meets performance requirements. This section describes the morphology design approach, compares this resulting morphology to other state of the art omnidirectional systems, and justifies the choice of a tiltrotor OMAV.

**Figure 2.** Definitions of angles  $\theta$  and  $\beta$  for the tilt-rotor morphology optimization.

### 3.1 Evaluation metrics

We aim to achieve the following design goals:

- Fully actuated system in any hover orientation.

- High force and torque capabilities in all directions.
- High efficiency hover in at least one orientation.

To evaluate the dynamic capabilities of force- and pose-omnidirectionality, as well as hover efficiency, we define the following metrics:

#### 3.1.1 Force and torque envelopes

Volumes corresponding to the maximum reachable forces ( $f_{vol}$ ) and torques ( $\tau_{vol}$ ) of the system expressed in  $\mathcal{F}_B$  can be computed by feeding commands forward through the actuator allocation matrix (see eq. (8)). We refer to these volumes as *force* and *torque envelopes*. Maximum, minimum, and mean values for the envelopes are calculated ( $f_{max}, f_{min}, f_{mean}$  and  $\tau_{max}, \tau_{min}, \tau_{mean}$ ) and used to evaluate dynamic capabilities. The force envelope is computed in the absence of torque, and the torque envelope is computed in the presence of a static hover force. Torque envelopes presented here include a hover force aligned with  $z_B$ , though other directions have been evaluated to verify full pose omnidirectionality in all hover conditions for optimized tiltrotor platforms.

#### 3.1.2 Force efficiency index, $\eta_f$

As an efficiency metric of omnidirectional force exertion, we use a *force efficiency index*  $\eta_f \in [0, 1]$  as originally defined by Ryll et al. (2016). The index represents the ratio of the desired body force magnitude to the sum of individual rotor group thrust magnitudes, as expressed in eq. (11).

$$\eta_f = \frac{\|Bf_d\|}{\sum_{i=1}^n Bf_{r_i}} \quad (11)$$

When  $\eta_f = 1$ , no internal forces are present, and all acting forces are aligned with the desired force vector  $f_d$ . While these internal forces should be reduced for efficient flight,

they also allow for instantaneous disturbance rejection, since thrust vectoring can be achieved by changing only rotor speeds.

A *torque efficiency index* can be computed similarly in eq. (12) to evaluate the efficiency of maximum torque commands, considering the body moment due to propeller forces acting at distance  $l$  from  $O_B$ . Torque contributions due to rotor drag are assumed to be negligible, since they are a significantly smaller multiple of the thrust force.

$$\eta_\tau = \frac{\|B\tau_d\|}{l \sum_{i=1}^n B f_{r_i}} \quad (12)$$

### 3.1.3 System Inertia

The inertial properties of the platform translate generalized force to acceleration, and therefore shape the agility of the flying system. Reducing inertia increases agility, and can be achieved by reducing the total mass  $m$ , and bringing the mass closer to the body origin  $O_B$ . Dynamic tiltrotor inertial effects are neglected for this analysis. The mass and inertia used at each optimization step are parametrically computed based on a simplified system geometry and realistic component masses, as derived in appendix A.1.

## 3.2 Optimization

We present a tiltrotor morphology optimization tool, made available open source<sup>1</sup>. This tool performs parametric optimization of a generalized tiltrotor model as shown in fig. 2, subject to a cost function of force and torque exertion, agility and efficiency metrics. We define fixed angles  $\theta, \beta \in \mathbb{R}^n$  as  $z_B$ - and  $y_B$ -axis angular deviations from a standard multicopter morphology with arms evenly distributed in the  $z_B$ -plane. Rotor groups can tilt actively about  $x_{r_i}$ , and are controlled independently. For the purposes of this paper, we only optimize over  $\theta$  and  $\beta$ . Although the tool allows for additional optimization over number of arms  $n$  and arm length  $l$ , we set these parameters in advance to 6 and 0.3 m respectively.

We consider two cost functions to evaluate an omnidirectional tiltrotor system: one that prefers unidirectional flight efficiency while requiring omnidirectional flight, and another that maximizes omnidirectional force and torque capabilities. The first cost function is defined as a maximization of the force envelope in one direction (here, unit vector  $\hat{z}_B$ ), while requiring a minimum force in all directions, expressed as

$$\begin{aligned} & \min_{\theta, \beta} (-f_{max} \hat{z}_B) \\ \text{s.t.} & \\ & f_{min} > mg \\ & -\frac{\pi}{2} < \theta_i < \frac{\pi}{2} \\ & -\frac{\pi}{2} < \beta_i < \frac{\pi}{2} \\ & \omega_{min} < \omega_i < \omega_{max}, \quad \forall i \in \{1 \dots n\} \end{aligned} \quad (13)$$

The second cost function seeks to maximize omnidirectional capability by maximizing the minimum force and torque envelope values in all body directions, i.e. for all surface point directions on a unit sphere. This result will

**Table 2.** Optimized morphologies and simplified metrics

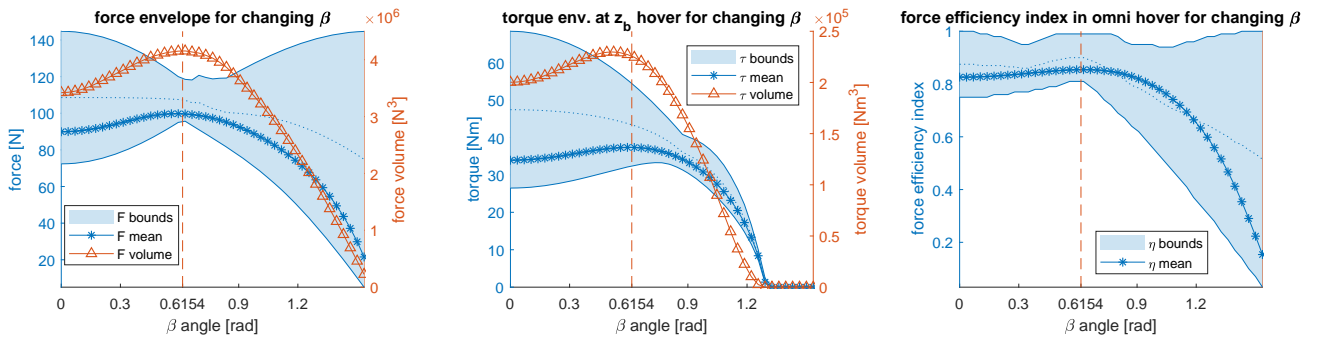
property	cost function 1, eq. (13)
$\beta$ [deg]	[0, 0, 0, 0, 0, 0]
$\theta$ [deg]	[0, 0, 0, 0, 0, 0]
mass [kg]	4.0
inertia* [kgm <sup>2</sup> ]	{0.0725, 0.0725, 0.1439}
$\{f_{min}, f_{max}\}$ [N]	{72.4, 144.7}
$f_{vol}$ [N <sup>3</sup> ]	$3.5 \cdot 10^6$
$\{\tau_{min}, \tau_{max}\}$ [Nm]	{26.5, 68.4}
$\tau_{vol}$ [Nm <sup>3</sup> ]	$2.1 \cdot 10^5$
$\eta_f$ at hover {min, max}	{0.75, 1.0}
property	cost function 2, eq. (14)
$\beta$ [deg]	$35.26 \cdot [1, -1, 1, -1, 1, -1]$
$\theta$ [deg]	[0, 0, 0, 0, 0, 0]
mass [kg]	4.0
inertia* [kgm <sup>2</sup> ]	{0.0959, 0.0959, 0.0972}
$\{f_{min}, f_{max}\}$ [N]	{96.5, 118.2}
$f_{vol}$ [N <sup>3</sup> ]	$4.2 \cdot 10^6$
$\{\tau_{min}, \tau_{max}\}$ [Nm]	{28.7, 44.0}
$\tau_{vol}$ [Nm <sup>3</sup> ]	$1.7 \cdot 10^5$
$\eta_f$ at hover {min, max}	{0.82, 1.0}

\*Primary components of inertia are presented, products of inertia are assumed negligible.

guarantee omnidirectional hover if the system's parameters provide a sufficient thrust to weight ratio.

$$\begin{aligned} & \min_{\theta, \beta} (-f_{min}, -\tau_{min}) \\ \text{s.t.} & \\ & f_{min} > mg \\ & -\frac{\pi}{2} < \theta_i < \frac{\pi}{2} \\ & -\frac{\pi}{2} < \beta_i < \frac{\pi}{2} \\ & \omega_{min} < \omega_i < \omega_{max}, \quad \forall i \in \{1 \dots n\} \end{aligned} \quad (14)$$

The resulting optimized morphologies and metrics are presented in table 2. The first optimization function results in a tiltrotor that takes a standard hexacopter morphology. Force in a single direction is maximized along  $z_B$ , and system properties are sufficient to ensure hover in any orientation. The second optimization has multiple solutions of the same result result metrics, placing rotor groups  $O_{r_i}$  at the vertices of an arbitrarily oriented octahedron. We list the  $\beta$  values when  $\theta$  are 0. Results show that both systems can sustain omnidirectional hover with additional force capability, and have omnidirectional force and torque envelopes, as seen by the  $f_{min}, \tau_{min}$ . We further evaluate the evolution of performance metrics between solutions, with  $\theta$  values fixed at 0, and  $\beta = \beta \cdot [1, -1, 1, -1, 1, -1]$  for changing  $\beta$ . Results are shown in fig. 3, where we see a clear maximization of the minimum reachable force in the left plot. Considering the goal of versatility for efficient flight with omnidirectional capabilities, the  $\beta = 0$  solution presents high reachable forces with a maximum efficiency



**Figure 3.** Variation of force and torque envelope metrics (max and min bounds, mean and volume) with changing  $\beta$ , the most efficient hover solution is at  $\beta = 0$  and the maximized force/torque envelope is at  $\beta = 0.6154$  rad (35.26 deg).

index, and sufficiently large minimum reachable forces and torques for agile flight in 6 DOF.

### 3.3 Comparison

We compare capabilities of the two optimized tiltrotor OMAV results to other state-of-the-art omnidirectional micro aerial vehicles (OMAVs). Platforms selected for comparison are the fixed rotor *tilt-hex* realized for aerial interaction applications (Ryll et al. 2019) and the fixed rotor omnidirectional platforms derived and realized by Brescianini and D’Andrea (2016); Park et al. (2016).

For all platforms compared in fig. 4, we show result metrics in table 3 relative to the first tiltrotor platform. For intuitive comparison of the second optimized tiltrotor result, the configuration is rotated such that axes align with the fixed rotor platform in fig. 4 d).

Force and torque envelopes are colored with the efficiency index of the maximum achievable values in each direction. The third column shows the force efficiency index for each achievable hover direction, plotted on a unit sphere. As expected, the reachable force and torque envelopes for the two tiltrotor systems are much larger than their fixed rotor counterparts. Designs a) and c) achieve high forces in level hover, while designs b) and d) show a more uniform distribution of omnidirectional force. Due to its tilting rotors, design a) still has better omnidirectional force and torque characteristics than the fixed rotor design d) which is optimized for omnidirectional force.

The tiltrotor implementation of a standard hexacopter design promises a versatile and capable morphology solution. Additional weight and complexity can be justified by significantly improved performance metrics. Independent tilting of each rotor group results in overactuation: 12 inputs to control 6 DOF. The controller can act in the null space of the allocation to assign secondary tasks. We can further justify the tilt-rotor version of a standard hexacopter morphology for perception applications, where the dual unobstructed hemispheres of the  $z_B$  plane allow for a large field of view.

As previously discussed, dynamic inertial effects of tilting rotor groups were not taken into account for the optimization. As such, we choose to augment the system with double propeller groups to increase total thrust and to balance rotational inertia about the tilting axis. Counter-rotating propellers in each rotor group further reduce gyroscopic

effects. Upper rotor directions alternate with adjacent arms to reduce the unmodelled effects of dual rotor groups.

Coordinate systems for the platform are described in fig. 5, with the body and rotor group frames  $\mathcal{F}_B, \mathcal{F}_{r_i}$ , and definitions of the fixed arm spacing angles  $\gamma = \frac{\pi}{6} \cdot [1, 3, 5, 7, 9, 11]$  and tilting angles  $\alpha_i$ . Individual rotor angular velocities  $\omega_i$  for  $i \in (1, 6)$  represent upper rotors and  $i \in (7, 12)$  represent lower rotors.

## 4 Control

### 4.1 Overview

The control problem consists of tracking a 6 DOF reference trajectory given by  $(\mathbf{p}_d(t), \mathbf{R}_d(t)) \in \mathbb{R}^3 \times \text{SO}(3)$ . Previous works Bodie et al. (2019); Kamel et al. (2018); Ryll et al. (2015) exploit the rigid body model from eq. (4) in combination with the aerodynamic force model from eq. (8) to compute actuator commands based on a thrust vectoring control. Force and torque commands are computed from position and attitude errors respectively and actuator commands can be resolved using the Moore-Penrose pseudo-inverse of the static allocation matrix.

$$\tilde{\Omega} = \mathbf{A}^\dagger \begin{bmatrix} B\mathbf{f} \\ B\boldsymbol{\tau} \end{bmatrix} \quad \mathbf{A}^\dagger \in \mathbb{R}^{n \times 6} \quad (15)$$

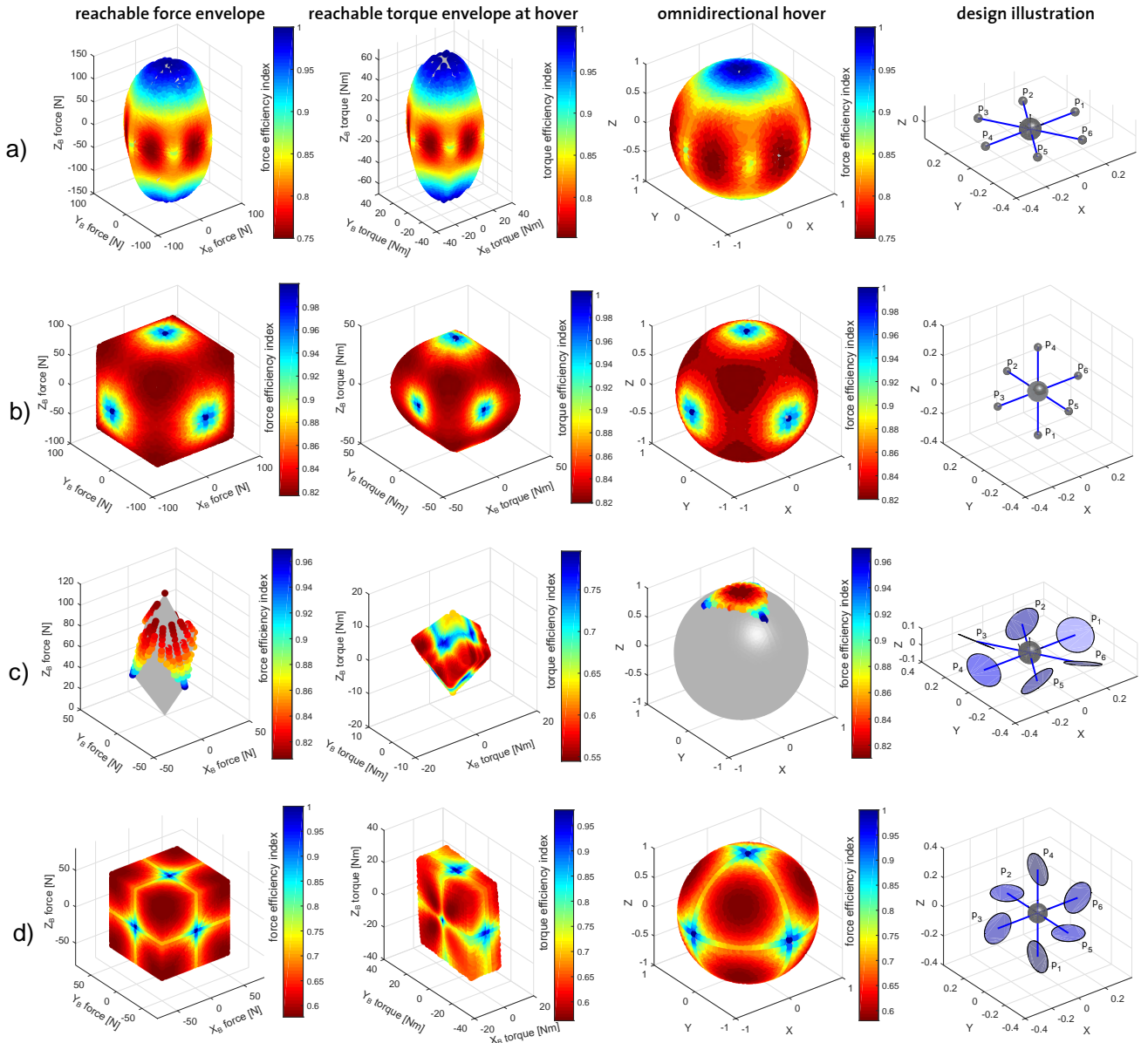
While this method shows good performance in most flight scenarios, it exhibits some disadvantages:

1. Singularities in the allocation matrix need to be avoided or handled explicitly
2. Rotor and tilt arm dynamics are not considered by the allocation

While controllers for overactuated MAVs usually operate on the acceleration level, we present two controllers that generate linear and angular jerk commands rather than acceleration commands. Specifically, we present

- an LQRI controller based on a rigid body model assumption, and
- a PID controller that generates acceleration commands with subsequent numerical differentiation to obtain jerk.

The main motivation to use jerk rather than acceleration commands is that this allows us to access the tilt angle dynamics of the rotating arms. Therefore we also present



**Figure 4.** Reachable force envelope, reachable torque envelope at  $z_B$  hover, omnidirectional hover directions colored by force efficiency index, all expressed in  $\mathcal{F}_B$  and a morphology design illustration are shown for (a) a tiltrotor optimized for unidirectional efficiency, (b) a tiltrotor that maximizes  $f_{min}$  and  $\tau_{min}$ , (c) a fixed rotor design from Ryll et al. (2019), and (d) a fixed rotor design from Brescianini and D’Andrea (2016); Park et al. (2016).

**Table 3.** Comparison of relative omnidirectional system properties corresponding to morphologies in fig. 4

property*	tiltrotor a)	tiltrotor b)	fixed rotor c)	fixed rotor d)
inertia** $\{m, I_{xx}, I_{yy}, I_{zz}\}$	reference	$\{1.0, 1.33, 1.33, 0.67\}$	$\{0.77, 0.99, 0.99, 0.99\}$	$\{0.77, 1.32, 1.32, 0.67\}$
force $\{\min, \max, \text{vol}\}$	reference	$\{1.33, 0.81, 1.20\}$	$\{0, 0.81, 0.01\}$	$\{0.67, 0.58, 0.26\}$
torque $\{\min, \max, \text{vol}\}$	reference	$\{1.14, 0.66, 0.95\}$	$\{0.21, 0.17, 0.01\}$	$\{0.31, 0.53, 0.17\}$
$\eta_f$ at hover $\{\min, \max\}$	reference	$\{1.09, 1.0\}$	$\{1.08, 0.97\}$	$\{0.7, 1.0\}$

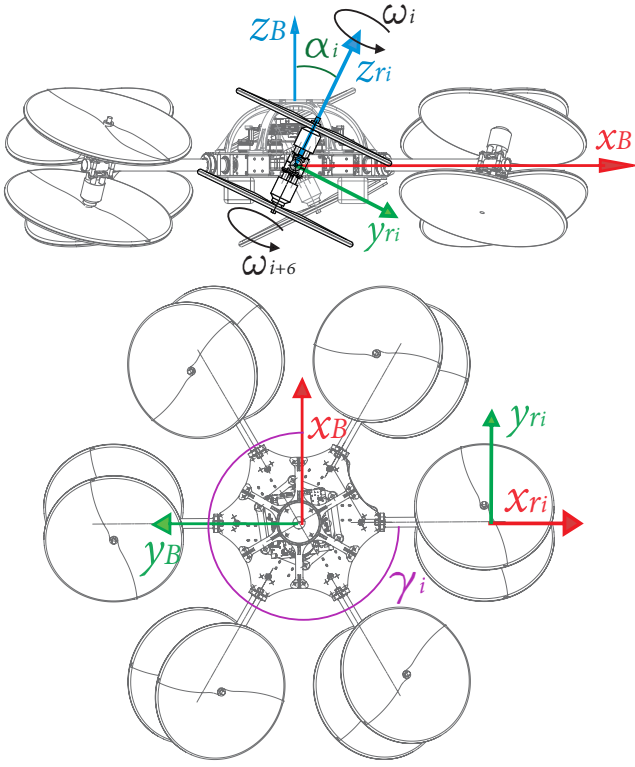
\*All values are expressed relative to the first tiltrotor system.

\*\*Primary components of inertia are presented, products of inertia are assumed negligible.

a new method of allocation, based on optimizing over the differential actuation commands  $(\dot{\alpha}, \dot{\omega})$  rather than the absolute actuation commands. We will refer to this as the *differential allocation*. This approach accounts for limited changes in tilt angles and rotor speeds, and can achieve almost arbitrary sets of tilt angles.

Figure 6 gives a high-level overview of the controller block diagram. LQRI or PID can be used interchangeably in the control block to compute jerk commands that are resolved to actuator commands by the differential allocation.

In this section, we will use the following notation:  $\mathbf{u} = [{}_B \mathbf{j}^T \quad {}_B \dot{\boldsymbol{\zeta}}^T]^T$  denotes the commanded body jerk which corresponds to the input to the differential allocation.  $\bar{\mathbf{u}}$



**Figure 5.** Coordinate frames and variables for  $i^{\text{th}}$  rotor group  $r_i$  and body  $B$ . Principal axes are shown, as well as rotor group tilt angle  $\alpha_i$ , rotor angular velocity  $\omega_i$ , and arm spacing angle  $\gamma_i$ .

denotes the input after feedback linearization, and  $\tilde{\mathbf{u}} = [\dot{\omega}^T \ \dot{\alpha}^T]^T$  is the set of differential actuator commands.

## 4.2 LQRI controller

We use an LQRI controller to optimize the system dynamics according to the following infinite time optimization problem:

$$\begin{aligned} \min_{\tilde{\mathbf{u}}(t)} \int_0^\infty \mathbf{e}(t)^T \mathbf{Q} \mathbf{e}(t) + \tilde{\mathbf{u}}(t)^T \mathbf{R} \tilde{\mathbf{u}}(t) dt & \quad \mathbf{e} \in \mathbb{R}^n \\ & \quad \tilde{\mathbf{u}} \in \mathbb{R}^m \\ & \quad \mathbf{Q} \in \mathbb{R}^{n \times n} \\ & \quad \mathbf{R} \in \mathbb{R}^{m \times m} \end{aligned} \quad (16)$$

s.t.  $\dot{\mathbf{e}}(t) = f(\mathbf{e}(t), \tilde{\mathbf{u}}(t))$

where  $\mathbf{e}$  is the error vector that is to be minimized,  $\tilde{\mathbf{u}}$  is the control input to the system,  $f(\cdot)$  is the function describing the state evolution, and  $\mathbf{Q}$ ,  $\mathbf{R}$  are weighting matrices for the error and control input.

In the following sections we present the underlying system model and the inputs that are used for optimization.

### 4.2.1 System jerk dynamics

The dynamics of the model are obtained by differentiating the dynamic equations of a rigid body model:

$$\begin{aligned} \frac{d}{dt} (m {}_B \mathbf{a}) &= \frac{d}{dt} ({}_B \mathbf{f} + \mathbf{R}_{BW} \mathbf{w} \mathbf{g}) \\ m {}_B (\dot{\mathbf{a}}) &= {}_B (\dot{\mathbf{f}}) - [{}_B \boldsymbol{\omega}_{WB}] \times \mathbf{R}_{BW} \mathbf{w} \mathbf{g} \end{aligned} \quad (17)$$

where  ${}_B (\dot{\mathbf{a}} = {}_B \mathbf{j})$  is the jerk in the body frame. Similarly we differentiate the angular acceleration:

$$\begin{aligned} \frac{d}{dt} (\mathbf{J}_B \dot{\boldsymbol{\psi}}_{WB} + [{}_B \boldsymbol{\omega}_{WB}] \times \mathbf{J}_B \boldsymbol{\omega}_{WB}) &= \\ \frac{d}{dt} (\boldsymbol{\tau}_B - [{}_{\mathbf{r}_{com}}] \times {}_B \mathbf{f}) & \end{aligned} \quad (18)$$

and obtain the following angular jerk equation of motion:

$$\begin{aligned} \mathbf{J}_B \dot{\boldsymbol{\psi}}_{WB} + 2 [{}_B \boldsymbol{\omega}_{WB}] \times \mathbf{J}_B \boldsymbol{\psi}_{WB} \\ + [{}_B \boldsymbol{\psi}_{WB}] \times \mathbf{J}_B \boldsymbol{\omega}_{WB} + [{}_B \boldsymbol{\omega}_{WB}] \times^2 \mathbf{J}_B \boldsymbol{\omega}_{WB} \\ = {}_B (\dot{\boldsymbol{\tau}}) - [[{}_B \boldsymbol{\omega}_{WB}] \times \mathbf{r}_{com}] \times {}_B \mathbf{f} \\ - [{}_{\mathbf{r}_{com}}] \times {}_B (\dot{\mathbf{f}}) \end{aligned} \quad (19)$$

Using the kinematics from eq. (3) we can also write the angular jerk explicitly as

$${}_B \zeta_{WB} = {}_B (\dot{\boldsymbol{\psi}}_{WB}) = {}_B \dot{\boldsymbol{\psi}}_{WB} + [{}_B \boldsymbol{\omega}_{WB}] \times {}_B \boldsymbol{\psi}_{WB} \quad (20)$$

### 4.2.2 Error vector

We define the error state vector as a concatenation of linear and angular differences between the state and the reference. Reference variables are denoted with a subscript  $(\cdot)_d$ .

$$\mathbf{e} = \begin{bmatrix} W \mathbf{e}_p \\ W \mathbf{e}_{p,i} \\ W \mathbf{e}_v \\ W \mathbf{e}_a \\ B \mathbf{e}_R \\ B \mathbf{e}_{R,i} \\ B \mathbf{e}_\omega \\ B \mathbf{e}_\psi \end{bmatrix} = \begin{bmatrix} W \mathbf{p} - W \mathbf{p}_d \\ \int W \mathbf{e}_p dt \\ W \mathbf{v} - W \mathbf{v}_d \\ W \mathbf{a} - W \mathbf{a}_d \\ \frac{1}{2} (\mathbf{R}_{WB_d}^T \mathbf{R}_{WB} - \mathbf{R}_{WB}^T \mathbf{R}_{WB_d})^\vee \\ \int B \mathbf{e}_R dt \\ {}_B \boldsymbol{\omega}_{WB} - \mathbf{R}_{BW} W \boldsymbol{\omega}_{WB_d,d} \\ B \mathbf{e}_\psi \end{bmatrix} \quad (21)$$

with  $\mathbf{e} \in \mathbb{R}^{24}$ . Note that the angular velocity reference  $W \boldsymbol{\omega}_{WB_d,d}$  denotes the reference angular velocity of the reference body frame with respect to the world frame. The geometric attitude error

The error dynamics are derived as:

$$W \dot{\mathbf{e}}_p = W \mathbf{e}_v \quad (22a)$$

$$W \dot{\mathbf{e}}_{p,i} = W \mathbf{e}_p \quad (22b)$$

$$W \dot{\mathbf{e}}_v = W \mathbf{e}_a \quad (22c)$$

$$\begin{aligned} W \dot{\mathbf{e}}_a &= \frac{1}{m} \mathbf{R}_{WB} \left( {}_B (\dot{\mathbf{f}}) - [{}_B \boldsymbol{\omega}_{WB}] \times \mathbf{R}_{BW} m \mathbf{w} \mathbf{g} \right) \\ &\quad - W \dot{\mathbf{j}}_d \end{aligned} \quad (22d)$$

$$\begin{aligned} B \dot{\mathbf{e}}_R &= \frac{1}{2} \underbrace{(\text{tr} (\mathbf{R}_{WB}^T \mathbf{R}_{BW_d}) \mathcal{I}_3 - \mathbf{R}_{WB}^T \mathbf{R}_{WB_d})}_{\mathbf{A}_{R\Omega}(\mathbf{R}_{WB}, \mathbf{R}_{WB_d})} B \mathbf{e}_\omega \end{aligned} \quad (22e)$$

$$B \dot{\mathbf{e}}_{R,i} = B \mathbf{e}_R \quad (22f)$$

$$B \dot{\mathbf{e}}_\omega = B \mathbf{e}_\psi \quad (22g)$$

$$B \dot{\mathbf{e}}_\psi \quad (22h)$$

Refer to appendix A.2, eq. (42) for a full formulation of the angular acceleration error dynamics.



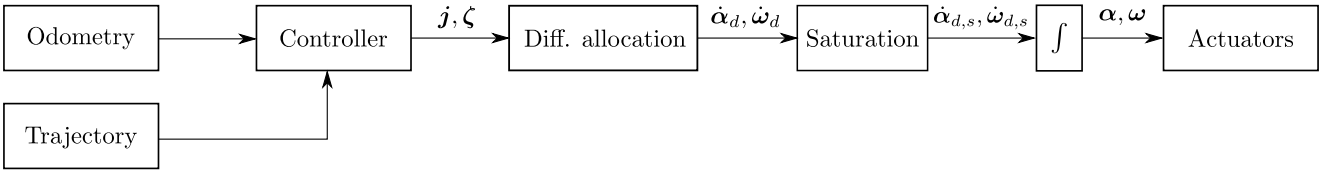


Figure 6. Control block diagram.

#### 4.2.3 Feedback linearization

We use feedback linearization to simplify the error dynamics, specifically eq. (22d) and eq. (22h). We introduce the virtual input vector  $\bar{\mathbf{u}} \in \mathbb{R}^6$  and define the total force derivative as follows:

$$\begin{aligned} {}_B(\dot{\mathbf{f}}) &= [{}_B\boldsymbol{\omega}_{WB}]_{\times} \mathbf{R}_{BW} m_W \mathbf{g} + m \mathbf{R}_{WB} W \dot{\mathbf{j}}_d \\ &+ m \mathbf{R}_{WB} \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \end{bmatrix} \end{aligned} \quad (23)$$

We follow the same principle for the angular dynamics, the full derivation is shown in appendix A.3, eq. (43).

Using the definitions from eq. (23) and eq. (43) together with eq. (22d) and eq. (22h) we obtain the simplified error dynamics for linear and angular acceleration:

$$W \dot{\mathbf{e}}_a = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \end{bmatrix} \quad {}_B \dot{\mathbf{e}}_\psi = \begin{bmatrix} \bar{u}_4 \\ \bar{u}_5 \\ \bar{u}_6 \end{bmatrix} \quad (24)$$

#### 4.2.4 Linearization

Having performed the feedback linearization, we obtain following nonlinear error dynamics:

$$\begin{aligned} \dot{\mathbf{e}} &= f(\mathbf{e}, \bar{\mathbf{u}}) \\ \frac{d}{dt} \begin{bmatrix} W \mathbf{e}_p \\ W \mathbf{e}_{p,i} \\ W \mathbf{e}_v \\ W \mathbf{e}_a \\ {}_B \mathbf{e}_R \\ {}_B \mathbf{e}_{R,i} \\ {}_B \mathbf{e}_\omega \\ {}_B \mathbf{e}_\psi \end{bmatrix} &= \begin{bmatrix} W \mathbf{e}_v \\ W \mathbf{e}_p \\ W \mathbf{e}_a \\ \mathbf{A}_{R\Omega} \begin{bmatrix} \bar{u}_1 & \bar{u}_2 & \bar{u}_3 \end{bmatrix}^T \\ {}_B \mathbf{e}_\omega \\ {}_B \mathbf{e}_R \\ {}_B \mathbf{e}_\psi \\ \mathbf{A}_{R\Omega} \begin{bmatrix} \bar{u}_4 & \bar{u}_5 & \bar{u}_6 \end{bmatrix}^T \end{bmatrix} \end{aligned} \quad (25)$$

As the attitude error dynamics are nonlinear, we linearize them at each time step around zero attitude error, i.e.  ${}_B \mathbf{e}_R = \mathbf{0}$ , to obtain a linear representation of the form

$$\dot{\mathbf{e}} = \mathbf{A} \mathbf{e} + \mathbf{B} \bar{\mathbf{u}} \quad \mathbf{A} \in \mathbb{R}^{24 \times 24}, \mathbf{B} \in \mathbb{R}^{24 \times 6} \quad (26)$$

Where  $\mathbf{A}$  is the state transition matrix and  $\mathbf{B}$  is the input matrix.

#### 4.2.5 Optimal gain computation

The optimal control inputs can then be computed using the gain  $\mathbf{K}_{LQRI}$ :

$$\bar{\mathbf{u}} = -\mathbf{K}_{LQRI} \mathbf{e} \quad \bar{\mathbf{u}} \in \mathbb{R}^6, \mathbf{K}_{LQRI} \in \mathbb{R}^{6 \times 24} \quad (27)$$

To find the optimal LQRI gain matrix, we solve the continuous time algebraic Riccati equation (CARE):

$$0 = \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} \quad (28)$$

and then use it's solution  $\mathbf{P} \succ 0$ ,  $\mathbf{P} \in \mathbb{R}^{24 \times 24}$  according to:

$$\mathbf{K}_{LQRI} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (29)$$

A stability proof can be found in appendix A.4.

### 4.3 PID controller

In this section we present a PID controller that yields rigid body jerk commands. The jerk commands are computed by numerically differentiating acceleration commands that are computed from a PID controller as in Bodie et al. (2018); Kamel et al. (2018).

Using the current as well as the reference state of the system, reference accelerations are obtained as follows:

$$W \mathbf{a} = \mathbf{K}_p W \mathbf{e}_p + \mathbf{K}_v W \mathbf{e}_v + \mathbf{K}_{p,i} W \mathbf{e}_{p,i} \quad (30a)$$

$${}_B \boldsymbol{\psi} = \mathbf{K}_R {}_B \mathbf{e}_R + \mathbf{K}_\omega {}_B \mathbf{e}_\omega + \mathbf{K}_{R,i} {}_B \mathbf{e}_{R,i} \quad (30b)$$

The error terms are defined as in eq. (21) and the matrices  $\mathbf{K}_{(\cdot)}$  are diagonal tuning matrices. The reference accelerations are then numerically differentiated and rotated to obtain the body jerk commands:

$$\mathbf{u} = \begin{bmatrix} {}_B \dot{\mathbf{j}} \\ {}_B \dot{\boldsymbol{\zeta}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{BW} W \dot{\mathbf{a}} \\ {}_B \dot{\boldsymbol{\psi}} \end{bmatrix} \quad (31)$$

### 4.4 Differential actuator allocation

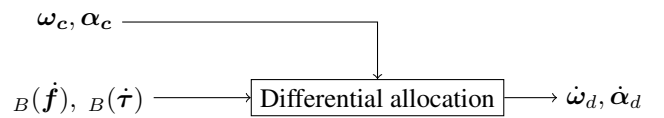


Figure 7. Differential allocation.

The LQRI and PID controllers presented above return jerk commands that need to be executed by the platform. Extending the work from Bodie et al. (2018) and based on work by Ryll et al. (2015), we present a weighted differential allocation of actuator commands, and include task prioritization in the null space.

Taking the time derivative of the standard allocation, we obtain the relation between the derivative of the body

wrench, actuator controls and their derivatives:

$$\begin{aligned}
\frac{d}{dt} \begin{pmatrix} {}_B \mathbf{f} \\ {}_B \boldsymbol{\tau} \end{pmatrix} &= \frac{d}{dt} (\mathbf{A} \tilde{\boldsymbol{\Omega}}) \\
&= \frac{d}{dt} \begin{pmatrix} \mathbf{A} \begin{bmatrix} \omega_1^2 \sin(\alpha_{c,1}) \\ \omega_1^2 \cos(\alpha_{c,1}) \\ \vdots \\ \omega_{12}^2 \sin(\alpha_{c,6}) \\ \omega_{12}^2 \cos(\alpha_{c,6}) \end{bmatrix} \\ \vdots \\ \dot{\omega}_{d,1} \\ \vdots \\ \dot{\omega}_{d,12} \\ \dot{\alpha}_{d,1} \\ \vdots \\ \dot{\alpha}_{d,6} \end{pmatrix} \\
&= \underbrace{\mathbf{A} \cdot \Delta \mathbf{A}(\omega_c, \alpha_c)}_{\tilde{\mathbf{A}}} \cdot \underbrace{\begin{bmatrix} \dot{\omega}_{d,1} \\ \vdots \\ \dot{\omega}_{d,12} \\ \dot{\alpha}_{d,1} \\ \vdots \\ \dot{\alpha}_{d,6} \end{bmatrix}}_{\tilde{\mathbf{u}} \in \mathbb{R}^{18}}
\end{aligned} \quad (32)$$

The vectors  $\omega_c, \alpha_c$  contain the current actuator commands, and the vector  $\tilde{\mathbf{u}} = [\dot{\omega}_d^T \ \dot{\alpha}_d^T]^T$  represents the desired derivatives of the actuator commands, i.e. rotor accelerations and tilt angle velocities.

Note that eq. (32) requires the derivative of the body wrench as input, while the controller returns a desired body jerk. Just like on the acceleration level, these two quantities are related by the body inertia properties:

$$\begin{bmatrix} {}_B \dot{\mathbf{f}} \\ {}_B \dot{\boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} m \mathcal{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix} \begin{bmatrix} {}_B \mathbf{j} \\ {}_B \boldsymbol{\zeta} \end{bmatrix} \quad (33)$$

Since the differential allocation matrix  $\tilde{\mathbf{A}}$  is in  $\mathbb{R}^{6 \times 18}$ , we have a high dimensional nullspace that can be exploited when computing  $\tilde{\mathbf{u}}$  using its pseudoinverse. In order to gain control over this freedom, we formulate following optimization problem:

$$\begin{aligned}
\min_{\tilde{\mathbf{u}}} & \|\mathbf{W}(\tilde{\mathbf{u}} - \tilde{\mathbf{u}}^*)\|_2 \\
\text{s.t.} & \begin{bmatrix} {}_B \dot{\mathbf{f}} \\ {}_B \dot{\boldsymbol{\tau}} \end{bmatrix} = \tilde{\mathbf{A}} \tilde{\mathbf{u}},
\end{aligned} \quad (34)$$

where  $\mathbf{W} \in \mathbb{R}^{18 \times 18}$  is a weighting matrix and  $\tilde{\mathbf{u}}^*$  are optimal rotor acceleration and tilt velocity values. The analytical solution to the problem is given as:

$$\tilde{\mathbf{u}} = \tilde{\mathbf{u}}^* + \mathbf{W} \tilde{\mathbf{A}}^T (\tilde{\mathbf{A}} \mathbf{W} \tilde{\mathbf{A}}^T)^{-1} \left( \begin{bmatrix} {}_B \dot{\mathbf{f}} \\ {}_B \dot{\boldsymbol{\tau}} \end{bmatrix} - \tilde{\mathbf{A}} \tilde{\mathbf{u}}^* \right) \quad (35)$$

Using the optimization from eq. (34), we are able to choose optimal differential controls  $\tilde{\mathbf{u}}^*$  freely. Here we present our method of finding a  $\tilde{\mathbf{u}}^*$  that achieves an efficient control configuration and that allows arm unwinding mid-flight.

Based on the current control inputs  $\alpha_c$  and  $\omega_c$ , we compute the resulting body wrench. Using the pseudo inverse of the static allocation matrix (eq. (8)), we can compute the optimal tilt angles  $\alpha^*$  and rotor speeds  $\omega^*$ . The optimal differential controls  $\tilde{\mathbf{u}}^*$  are then computed from the difference and are assigned unwinding velocities  $v_{\dot{\omega}}$  and  $v_{\dot{\alpha}}$ :

$$\dot{\omega}^* = \text{sign}(\omega^* - \omega_c) v_{\dot{\omega}} \quad (36a)$$

$$\dot{\alpha}^* = \text{sign}(\alpha^* - \alpha_c) v_{\dot{\alpha}} \quad (36b)$$

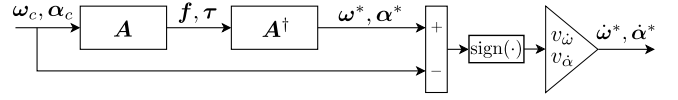


Figure 8. Optimal control allocation.

After having computed the optimal control inputs we use eq. (35) to find the desired differential control inputs  $\tilde{\mathbf{u}}$ . In a next step they are saturated according to predefined limits on maximum rotor and tilt angle velocities and subsequently integrated over time (see fig. 6). The resulting commands for tilt angles  $\alpha$  and for rotor speeds  $\omega$  are then sent to the actuators.

Kinematic singularities are handled inherently by the inclusion of  $\dot{\alpha}$  in the differential actuator allocation. Rank reduction singularities are still present, and we evaluate them by computing the condition number of the instantaneous allocation matrix,  $\kappa(\mathbf{A}_\alpha)$ , defined as

$$\kappa(\mathbf{A}_\alpha) = \frac{\sigma_{max}(\mathbf{A}_\alpha)}{\sigma_{min}(\mathbf{A}_\alpha)} \quad (37)$$

where  $\sigma_{max}$  and  $\sigma_{min}$  are the maximum and minimum singular values of  $\mathbf{A}_\alpha$ .

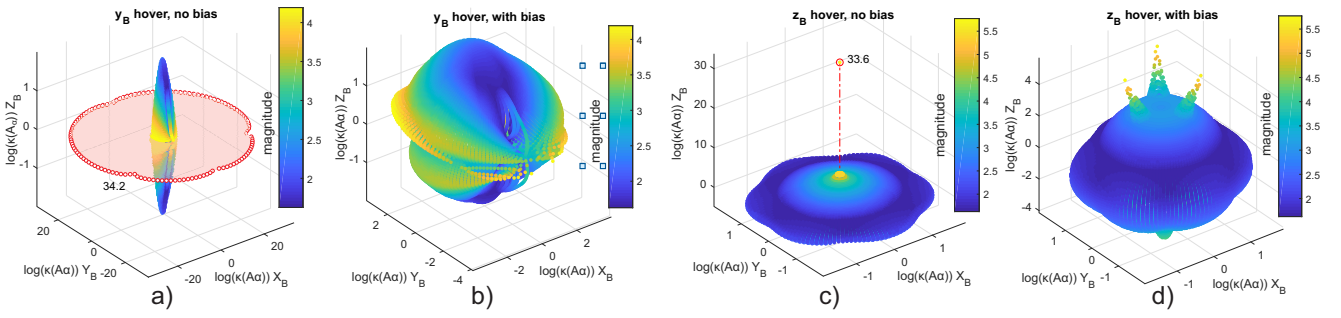
Figure 9 plots the log of  $\kappa(\mathbf{A}_\alpha)$  over a force envelope of magnitude  $mg$  in addition to a static hover force in  $y_B$  and  $z_B$  directions. An allocation of the current wrench according to eq. (8) produces the envelopes shown in a) and c), with high condition numbers in the  $z_B$ -plane, and along the  $z_B$ -axis respectively. High condition numbers occur where  $\sigma_{min}(\mathbf{A}_\alpha)$  is significantly reduced, indicating loss of rank and therefore loss of instantaneous controllability in at least one DOF.

We can address the above-mentioned rank reduction in the formulation of  $\alpha^*$ , incorporating a bias term  $\alpha_{bias}$  derived according to the alignment of  $\mathbf{f}_d$  in the body frame. The detailed derivation of  $\alpha_{bias}$  is presented by Bodie et al. (2018). The addition of this bias term to  $\alpha^*$  results in the log( $\kappa(\mathbf{A}_\alpha)$ ) envelopes in fig. 9(b,d). In these plots, values of  $\kappa_{max}(\mathbf{A}_\alpha)$  are significantly reduced, from exp(34.2) to exp(4.19) in  $y_B$  hover, and exp(33.6) to exp(5.77) in  $z_B$  hover.

## 5 Experimental Setup

### 5.1 System hardware

The experimental platform modeled after the morphology presented in section 3 is a 12 rotor MAV with 6 tilttable arms, shown in fig. 10. Primary system properties and components are reported in table 4, and described below. The system layout consists of equally spaced arms along the  $z_b$ -plane, with two rotors per arm to balance the rotational inertia. Two KDE 2315XF-885 motors per arm with 9in (228.6 mm) propellers provide sufficient thrust (11 N per motor), to enable dynamic flight in the least efficient configurations. Each arm is independently tilted by a Dynamixel XL430 servo actuator, located in the base to reduce system inertia. Upper and lower propellers counter-rotate such that drag torques from the propellers are approximately canceled, and the gyroscopic moment on the tilting mechanism



**Figure 9.** The log of the condition number  $\kappa(\mathbf{A}_\alpha)$  is plotted in  $\mathcal{F}_B$  for the cases of  $y_B$  (a,b) and  $z_B$  (c,d) hover with an envelope of additional gravitational force in all directions. Envelopes with a tilt bias compensation on  $\alpha$  (b,d) significantly reduce the maximum condition numbers. Condition number magnitudes are indicated on the color scale, with  $\kappa_{max}(\mathbf{A}_\alpha)$  indicated in (a,c)



**Figure 10.** Image of the prototype system.

is minimized, reducing the effort of the tilt motor. The controller and state estimator operate on an onboard Intel NUC i7, which sends direct actuator commands to a Pixhawk flight controller. All rotors and tilting servos, as well as a high resolution IMU, are connected to the flight controller. The system is powered by two 3800 mAh 6S LiPo batteries. The total system mass is 4.27 kg, and it can generate over 130 N of force at maximum thrust in the horizontal hover configuration.

## 5.2 Test environment

All experiments are performed in a laboratory environment, where an external Vicon motion capture system wirelessly sends high-accuracy odometry information to the onboard computer. The test area provides a space of approximately 4x4 meters in size and 3 meters in height. Motion capture pose estimates are taken at 20 Hz and fused by an EKF with the onboard IMU at a rate of 102 Hz. Reference trajectories are sent to the controller from a ground control station, or from the onboard computer. The trajectories are polynomials that are generated offline based on discrete waypoints. They contain smooth reference commands for position, velocity, acceleration, as well as attitude, angular velocity, and angular acceleration, and are sampled at 100 Hz. A block diagram of the experimental system setup is shown in fig. 11).

## 5.3 Acceleration estimation

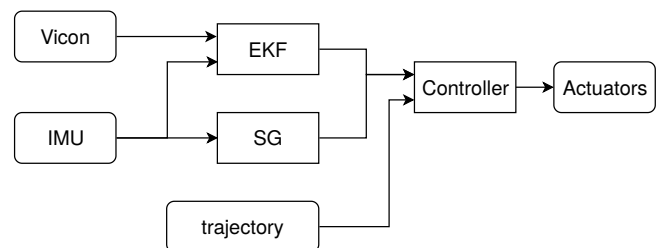
The suggested jerk level LQRI requires linear and angular acceleration estimates to find an optimal gain matrix. We use a first-order SG filter to smooth raw IMU acceleration and

**Table 4.** Main system components and parameters

component	part number	qty
onboard computer	Intel NUC i7	1
flight controller	Pixhawk mRo	1
rotor	KDE 2315XF-885	12
propeller	Gemfan 9x4.7	12
ESC	T-motor F45A 3-6S	12
tilt motor	Dynamixel XL430-W250	6
IMU	ADIS16448	1
battery	3800 mAh 6S LiPo	2

parameter	value	units
rotor groups	6	
arm length, $l$	0.3	[m]
total mass	4.27	[kg]
inertia*	{0.086, 0.088, 0.16}	[kgm <sup>2</sup> ]
diameter	0.83	[m]
$f_{r_i, max}$	11	[N]
$c_f$	$7.1e-6$	[Ns <sup>2</sup> /rad <sup>2</sup> ]
$\omega_{max}$	1250	[rad/s]

\*Primary components of inertia are obtained from CAD model.



**Figure 11.** System setup: Vicon and IMU measurements are fused by an EKF, and combined with the Savitzky-Golay (SG) filtered angular acceleration estimate for complete state estimation. The controller processes the state estimate and reference trajectory, and sends actuator commands.

angular velocity data. The smoothed angular velocity data is differentiated numerically to obtain angular acceleration estimates.

**Table 5.** Controller parameters for experimental tests

parameter	value	parameter	value
<b>LQRI</b>		<b>PID</b>	
$k_p$	200	$k_p$	5
$k_{p,i}$	50	$k_{p,i}$	0.3
$k_v$	100	$k_v$	1.0
$k_a$	0	$k_R$	3.5
$k_R$	100	$k_{R,i}$	0.3
$k_{R,i}$	100	$k_\omega$	0.8
$k_\omega$	200	<b>Allocation</b>	
$k_\psi$	0	$k_\alpha$	1000
$r_f$	(1.0, 1.0, 0.2)	$v_\alpha$	1
$r_\tau$	(1.0, 1.0, 1.0)	$v_\omega$	250

## 5.4 Controller parameters

Controller parameters and gains used in experimental tests are listed in table 5.

## 6 Experimental Results

### 6.1 Trajectories and data collection

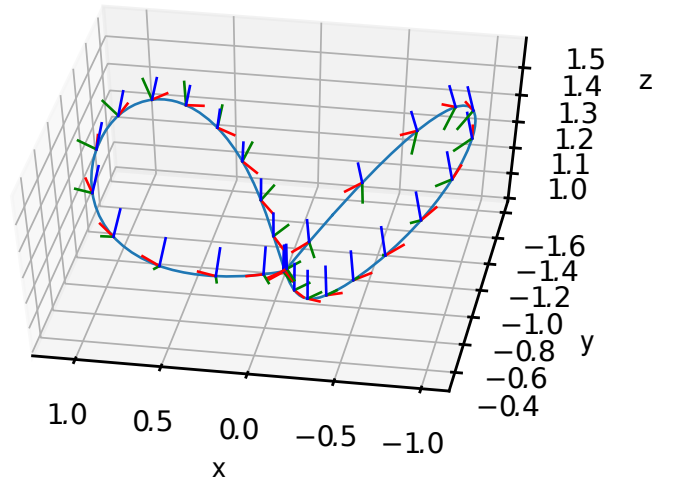
In order to evaluate and to compare tracking performance of the controllers, several experimental trajectories are designed and tested. Videos of the experimental trials can be found in our supporting multimedia content.

We first define a reference trajectory that covers large parts of the 6-dimensional pose space. The trajectory can be described as a figure eight with varying height and attitudes. Figure 12 illustrates the trajectory with some reference positions and attitudes. We use three different adaptations of the trajectory to evaluate controller performance, specifically:

- Trajectory (a), low angle: The maximum tilt angle does not exceed 30 degrees, the duration of the trajectory is 29.4 seconds.
- Trajectory (b), high angle: The maximum reference tilt angle does not exceed 80 degrees and the duration is 29.4 seconds.
- Trajectory (c), fast tracking: Using a maximum tilt angle of 30 degrees, the duration of the trajectory is 10.7 seconds.

Another set of trajectories target the singular configurations. The first commands a lateral translation and rotation while in both a kinematic and rank reduced singular state, namely at 90 degrees roll, and a second demonstrates transition through the kinematic singularity.

- Trajectory (d), singular translation: The trajectory transitions to a 90° roll, translates, reverses direction, and rotates about the  $y_B$ -axis. The duration of the trajectory is 36.1 s.
- Trajectory (e), cartwheel: The trajectory transitions to a 90° roll, then makes two complete rotations about the  $z_B$ -axis while translating in a circular trajectory. The duration of the trajectory is 35.5 s.



**Figure 12.** Reference figure eight trajectory used for performance evaluation.

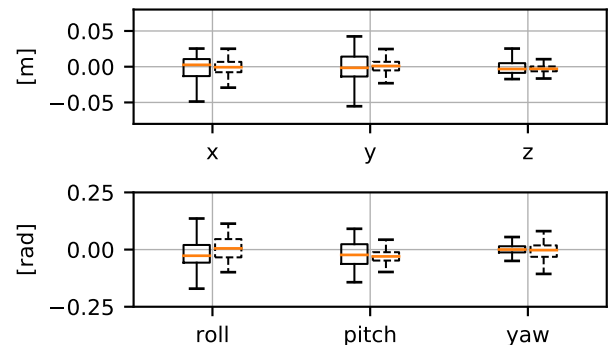
Two additional trajectories are designed to evaluate the secondary unwinding task with a complete rotation about a specified axis:

- Trajectory (f), roll flip: complete rotation about the  $x_B$  axis. The duration of the trajectory is 16 s.
- Trajectory (g), pitch flip: complete rotation about the  $y_B$  axis. The duration of the trajectory is 8 s.

Position and attitude errors throughout the section are taken from data collected over the complete duration of a single iteration of each trajectory, for both LQRI and PID controllers. Since we are interested in exploiting the omnidirectional capabilities of the system and thus track all 6 axes of a trajectory, errors are given separately in all axes.

Position errors are given in the world frame in meters, attitude errors are given with respect to the reference attitude in radians.

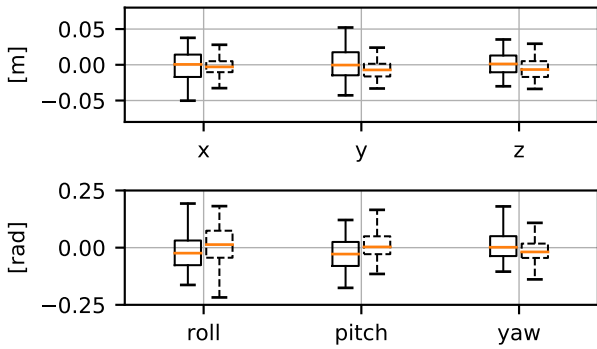
All errors are evaluated in boxplots, showing the median, the upper and lower quartile, as well as the 1.5 IQR.



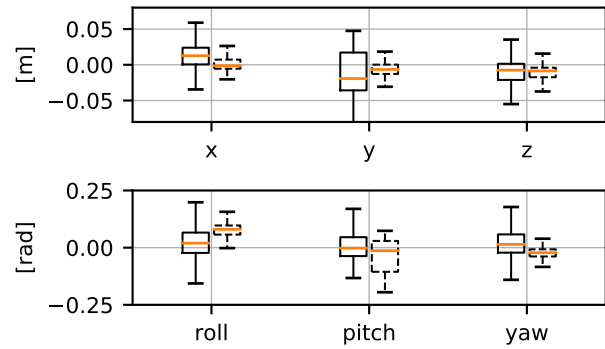
**Figure 13.** Position and attitude errors of LQRI (solid) and PID (dashed) for trajectory (a), low angle figure eight.

### 6.2 Full pose tracking

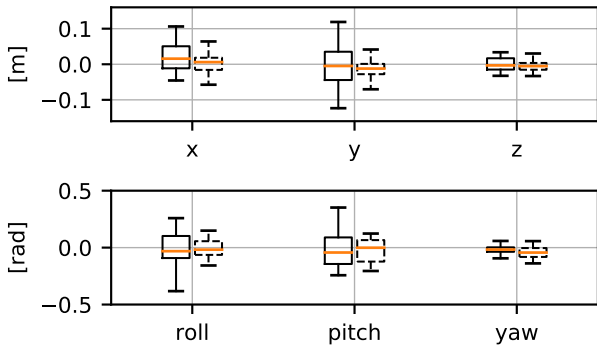
While both control approaches demonstrate stable tracking, fig. 13 to fig. 15 show a better tracking performance under the PID control law. We identify two major sources for the higher



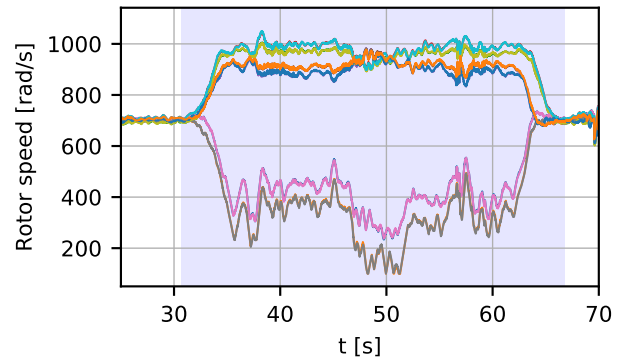
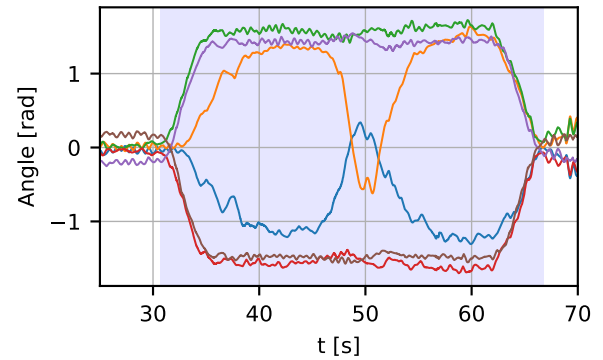
**Figure 14.** Position and attitude errors of LQRI (solid) and PID (dashed) for trajectory (b), high angle figure eight.



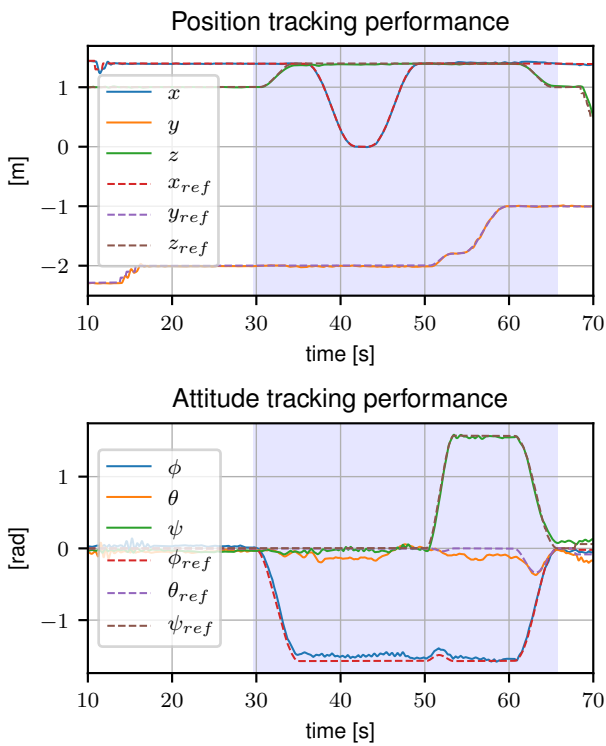
**Figure 17.** LQRI vs PID tracking errors during trajectory (d), singular translation.



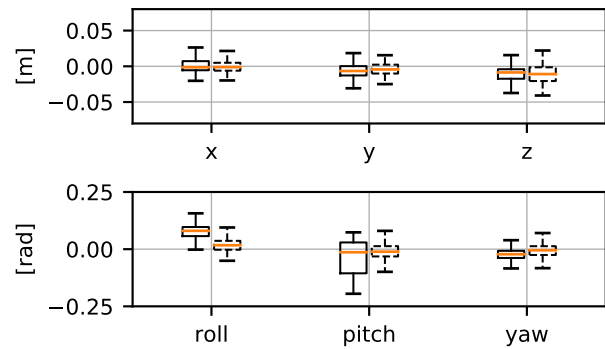
**Figure 15.** Position and attitude errors of LQRI (solid) and PID (dashed) for trajectory (c), fast figure eight.



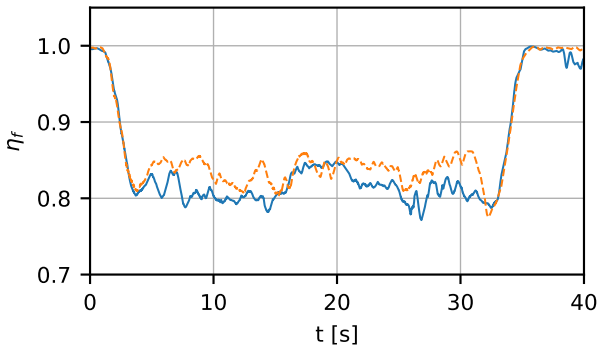
**Figure 18.** PID direct controls during trajectory (d), singular translation.



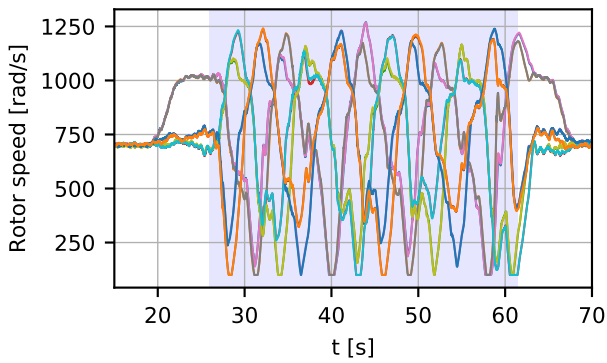
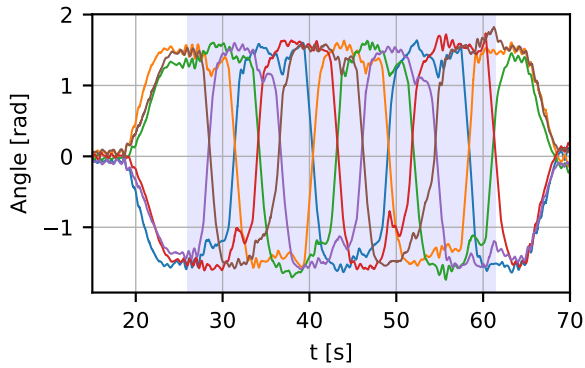
**Figure 16.** Position and attitude errors of LQRI (solid) and PID (dashed) for trajectory (d), singular translation.



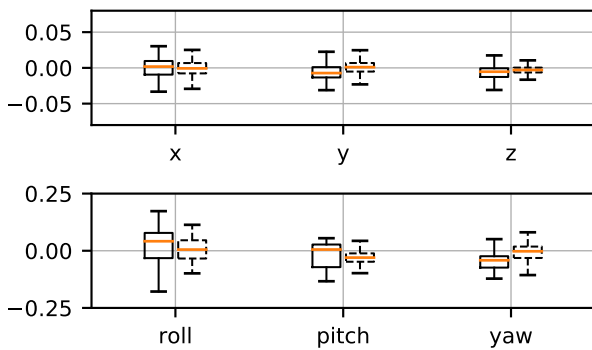
**Figure 19.** Tracking accuracy of PID with and without singularity handling, of trajectory (d), singular translation.



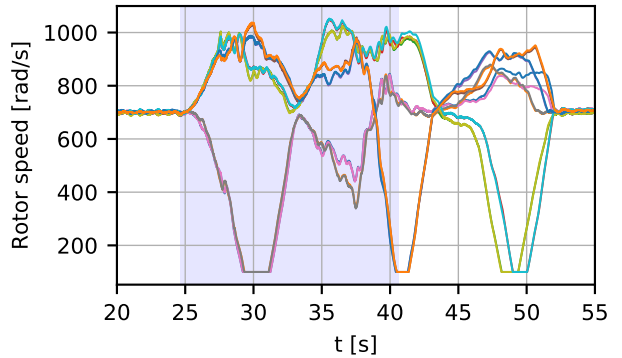
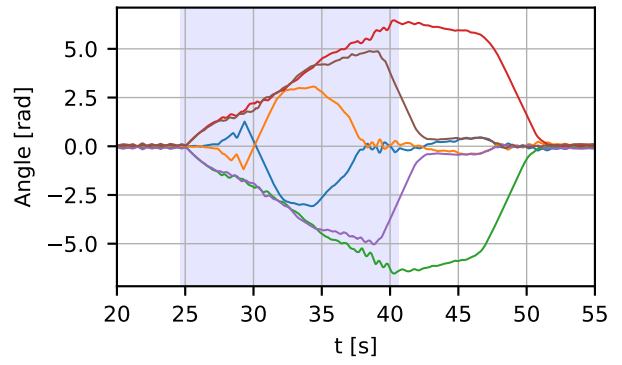
**Figure 20.** Force efficiency index of PID, with (solid) and without (dashed) singularity handling, of trajectory (d), singular translation.



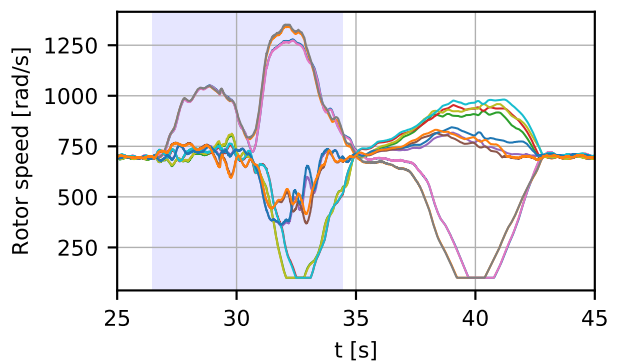
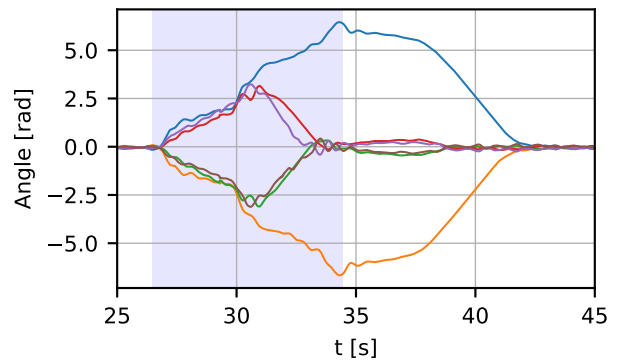
**Figure 21.** Trajectory (e) actuator commands during a cartwheel.



**Figure 22.** Trajectory (a) tracking errors with and without unwinding arms (solid and dashed lines, respectively).



**Figure 23.** Trajectory (f) actuator commands during a roll flip. The duration of the flip trajectory is highlighted.



**Figure 24.** Trajectory (g) actuator commands during a pitch flip. The duration of the flip trajectory is highlighted.

tracking error of the LQRI controller. Firstly, all components of the LQRI error state  $e$  are ultimately dependent on the linear or angular acceleration errors, according to the dynamic matrix  $A$  from eq. (26). Consequently, the

controller strongly relies on precise acceleration estimates to perform well. However, due to the introduced delay in the SG filter and the numerical differentiation, these estimates may be noisy and inaccurate. Secondly, the LQRI controller dynamics (especially the feedback linearisation) are based on the system model presented in section 2. That being said, several assumptions made are likely to result in unmodeled forces in the system, e.g. configuration-based airflow interference, and additional dual-rotor effects. We expect that these unmodeled disturbances lead to additional tracking error. While the integrator action might compensate these effects to a certain extent, it also reduces the ability to track fast trajectories because of the introduced delay.

### 6.3 Singularity handling

The proposed differential allocation is capable of inherently dealing with *kinematic singularities* as described by Morbidi et al. (2018). The described optimization ensures finite tilt velocities and thereby allow stable flight, as shown in figs. 16 and 17.

Figure 20 shows how the proposed *rank reduction singularity* handling procedure reduces the force efficiency index by increasing internal forces. This configuration, while slightly less efficient, allows instantaneous omnidirectional force generation, avoiding the singularity condition. Results in fig. 19 do not show an improvement in tracking performance with additional singularity handling. This can be explained by small correction terms of force and torque that are constantly present, so the system is rarely if ever in a true singular state.

### 6.4 Unwinding

We illustrate the advantage of the differential allocation by performing trajectory (a) while commanding four arms to perform a full rotation. This is achieved by starting the experiment with four arms set to  $\alpha = 2\pi$ , then activating the unwinding task at the start of the trajectory.

Figure 22 illustrates tracking errors during a standard trajectory, comparing with and without unwinding 4 out of 6 arms. The comparison in fig. 22 shows, that despite the full unwinding of 4 arms, the tracking performance is only affected slightly. Figure 24 shows that the unwinding is done sequentially. As the trajectory starts and the pitch increases, two arms are unwound completely in the first half of the trajectory. The opposite two arms are unwound in the second half. This sequential unwinding of arms results from the optimization of the differential allocation in combination with the given trajectory and has not been specifically designed.

As a comparison, the roll flip control inputs in fig. 23 shows that the optimization triggers the unwinding only after the flip (blue background) has been completed. Similarly, during pitch flip (actuator commands shown in fig. 24) it can be observed how the 2 arms that are aligned with the y axis perform a full rotation while the remaining 4 arms can already unwind during the second half of the flip. Unwinding for the remaining two arms continues afterwards in steady hover. The delay of unwinding is influenced by the choice of  $v_{\dot{\alpha}}, v_{\dot{\omega}}$  and weighting matrix  $\mathbf{W}$ , which are selected based on a compromise of responsive unwinding and overall

system stability. The weighting matrix  $\mathbf{W}$  for the differential allocation has been set to be  $\text{blkdiag}(\mathcal{I}_{12}, k_{\alpha} \cdot \mathcal{I}_6)$ .

In all three cases, the rotor speeds are ramped down to avoid force inversion. However, because the rotor speeds and therefore the rotor thrusts are not upper bounded in the optimization, the procedure could result in the simultaneous unwinding of more arms than minimally necessary for hover. Additionally, the differential allocation does not incorporate knowledge of the future trajectory and might thus start unwinding, even if it turns out to be unfavourable later on.

### 6.5 Complexity and efficiency

The differential actuator allocation demonstrates prioritization of 6 DOF tracking, while optimally exploiting the null space to fulfill secondary tasks. Results show successful kinematic singularity rejection and stable arm unwinding for different trajectories. That being said, the mentioned optimality comes at the cost of higher computational load, since a high dimensional matrix inversion needs to be performed at each iteration. We can justify this computational load with a high performance onboard computer, which completes an iteration of the control loop in 3 ms on average.

The additional complexity of the actuation can be justified by comparing to other morphological designs. The added complexity of tilting rotor groups allows force- and pose-omnidirectionality, enabling the system to track 6DOF trajectories with highly dynamic capabilities. This is not possible with regular underactuated multicopters. Fixed rotor systems that enable full actuation directly select a tradeoff of flight efficiency and omnidirectional force capabilities via their design morphology.

The proposed tiltrotor system takes advantage of both highly efficient flight and omnidirectional capabilities, at the expense of additional weight and actuation complexity. Morphology optimization results presented in section 3 quantify these metrics. Verified system performance in experimental flights and the opportunity of null space task prioritization further strengthen the case for the tiltrotor OMAV.

## 7 Conclusion

In this paper, the complete system design and optimal control for a novel efficient and versatile tiltrotor OMAV has been presented. We have open sourced a morphology optimization tool for design of a tiltrotor OMAVs, and used it to demonstrate the advantages of the system. A new controller has been derived and implemented, based on a jerk-level LQRI and a differential actuator allocation. We have further integrated secondary tasks to trajectory tracking into the actuation null space, such as maximizing hover efficiency, avoiding cable windup, and singularity handling. Through various experiments we have compared tracking of the LQRI and PID controllers, and verified the capabilities of task prioritization in the actuator allocation.

Regarding future work, we intend to improve the system model, and perform a diligent system identification, the importance of which is discussed in section 6. We expect that higher model accuracy will improve the LQRI performance. The integration of constraints in actuator allocation, and consideration of further secondary tasks, are also the subject

of future work. Acceleration estimation has a significant influence on the controller, and will be evaluated in more detail. The controller presented in this paper sets the stage for further optimization over the trajectory horizon, e.g. using a nonlinear MPC.

### Acknowledgements

This work was supported by funding from ETH Research Grants, the National Center of Competence in Research (NCCR) on Digital Fabrication, NCCR Robotics, and Armasuisse Science and Technology.

### Notes

1. Tiltrotor morphology optimization tool is available open source at [github.com/ethz-asl/tiltrotor\\_morphology\\_optimization](https://github.com/ethz-asl/tiltrotor_morphology_optimization)

### References

- Bodie K, Brunner M, Pantic M, Walser S, Pfändler P, Angst U, Siegwart R and Nieto J (2019) An Omnidirectional Aerial Manipulation Platform for Contact-Based Inspection. In: *Proceedings of Robotics: Science and Systems*.
- Bodie K, Taylor Z, Kamel M and Siegwart R (2018) Towards Efficient Full Pose Omnidirectionality with Overactuated MAVs. In: *International Symposium on Experimental Robotics*. Springer.
- Brescianini D and D'Andrea R (2016) Design, modeling and control of an omni-directional aerial vehicle. In: *ICRA 2016*. IEEE, pp. 3261–3266.
- Falconi R and Melchiorri C (2012) Dynamic model and control of an over-actuated quadrotor uav. *IFAC Proceedings Volumes* 45(22): 192–197.
- Kamel M, Verling S, Elkhatib O, Sprecher C, Wulkop P, Taylor Z, Siegwart R and Gilitschenski I (2018) The Voliro Omniorientational Hexacopter: An Agile and Maneuverable Tilttable-Rotor Aerial Vehicle. *IEEE Robotics & Automation Magazine* 25(4): 34–44.
- Lee T (2012) Robust Adaptive Attitude Tracking on  $SO(3)$  With an Application to a Quadrotor UAV. *IEEE Transactions on Control Systems Technology* 21(5): 1924–1930.
- Morbidi F, Bicego D, Ryll M and Franchi A (2018) Energy-efficient trajectory generation for a hexarotor with dual-tilting propellers. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6226–6232.
- Ollero A, Heredia G, Franchi A, Antonelli G, Kondak K, Sanfeliu A, Viguria A, Martinez-De Dios JR, Pierri F, Cortés J et al. (2018) The AEROARMS Project: Aerial Robots with Advanced Manipulation Capabilities for Inspection and Maintenance. *IEEE Robotics and Automation Magazine* .
- Park S, Her J, Kim J and Lee D (2016) Design, modeling and control of omni-directional aerial robot. In: *IROS 2016*. IEEE, pp. 1570–1575.
- Park S, Lee J, Ahn J, Kim M, Her J, Yang GH and Lee D (2018) ODAR: Aerial Manipulation Platform Enabling Omnidirectional Wrench Generation. *IEEE/ASME Transactions on Mechatronics* 23(4): 1907–1918.
- Ryll M, Bicego D and Franchi A (2016) Modeling and control of FAST-Hex: a fully-actuated by synchronized-tilting hexarotor. In: *IROS 2016*. IEEE, pp. 1689–1694.
- Ryll M, Bicego D and Franchi A (2018) A Truly Redundant Aerial Manipulator exploiting a Multi-directional Thrust Base. *IFAC-PapersOnLine* 51(22): 138–143.
- Ryll M, Bühlhoff HH and Giordano PR (2015) A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation. *IEEE Transactions on Control Systems Technology* 23(2): 540–556.
- Ryll M, Muscio G, Pierri F, Cataldi E, Antonelli G, Caccavale F, Bicego D and Franchi A (2019) 6D interaction control with aerial robots: The flying end-effector paradigm. *The International Journal of Robotics Research* : 0278364919856694.
- Staub N, Bicego D, Sablé Q, Arellano V, Mishra S and Franchi A (2018) Towards a Flying Assistant Paradigm: the OTHex. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 6997–7002.
- Tognon M and Franchi A (2018) Omnidirectional aerial vehicles with unidirectional thrusters: Theory, optimal design, and control. *IEEE Robotics and Automation Letters* 3(3): 2277–2282.
- Wopereis H, van de Ridder L, Lankhorst TJ, Klooster L, Bukai E, Wuthier D, Nikolakopoulos G, Stramigioli S, Engelen JB and Fumagalli M (2018) Multimodal Aerial Locomotion: An Approach to Active Tool Handling. *IEEE Robotics & Automation Magazine* .



## A Appendix

### A.1 Morphology Optimization Inertia Calculations

We consider a constant core mass  $m_{c, const}$  that allows for complete on-board autonomy, an additional core mass  $m_a$  to provide tiltrotor actuation for each of  $n$  rotor groups, and a mass for each rotor group  $m_{r_i}$ . The mass of each rigid arm tube  $m_t$  is a function of its length  $L$  and length normalized mass  $m_{t, norm}$ .

$$m_c = m_{c, const} + n \cdot m_a \quad (38a)$$

$$m_t = m_{t, norm} \cdot L \quad (38b)$$

$$m = m_c + \sum_{i=1}^n (m_{r_i} + m_t) \quad (38c)$$

The core inertia is modeled as a solid cylinder centered at  $O_B$  with radius  $r_c$  and height  $h_c$ . Rigid tilt arms that connect to propeller groups are modeled as cylindrical tubes, with radii  $r_1, r_2$  and length  $L$ . For fairness of comparison, we consider that all systems have single rotor groups with origin  $O_{r_i}$ , and a tilt axis  $x_{r_i}$  aligned with the corresponding arm axis. The inertia of each rotor group is modeled as a cylinder of radius  $r_{r_i}$  and height  $h_{r_i}$ , with inertial values in  $y$  and  $z$  axes averaged to approximate a system independent of tilt. Values are chosen based on components presented in section 5.1.

$$\mathbf{J}_c = \text{diag} \left( \begin{bmatrix} \frac{1}{12} m_c (3r_c^2 + h_c^2) \\ \frac{1}{12} m_c (3r_c^2 + h_c^2) \\ \frac{1}{2} m_c r_c^2 \end{bmatrix} \right) \quad (39a)$$

$$\mathbf{J}_t = \text{diag} \left( \begin{bmatrix} \frac{1}{2} m_t r_c^2 \\ \frac{1}{12} m_t (3(r_1^2 + r_2^2) + L^2) \\ \frac{1}{12} m_t (3(r_1^2 + r_2^2) + L^2) \end{bmatrix} \right) \quad (39b)$$

$$\mathbf{J}_{r_i} = \text{diag} \left( \begin{bmatrix} \frac{1}{12} m_{r_i} (3r_{r_i}^2 + h_{r_i}^2) \\ \frac{1}{2} (\frac{1}{12} m_{r_i} (3r_{r_i}^2 + h_{r_i}^2) + \frac{1}{2} m_{r_i} r_{r_i}^2) \\ \frac{1}{2} (\frac{1}{12} m_{r_i} (3r_{r_i}^2 + h_{r_i}^2) + \frac{1}{2} m_{r_i} r_{r_i}^2) \end{bmatrix} \right) \quad (39c)$$

Total system inertia is then computed using the parallel axis theorem to express all inertias in  $\mathcal{F}_B$ .

$$\mathbf{p}_t = [\frac{L}{2} \quad 0 \quad 0]^\top \quad (40a)$$

$$\mathbf{p}_{r_i} = [L \quad 0 \quad 0]^\top \quad (40b)$$

$${}_{O_B} \mathbf{J}_t = \mathbf{J}_t + m_t \|\mathbf{p}_t\|^2 \mathcal{I}_3 - \mathbf{p}_t \mathbf{p}_t^\top \quad (40c)$$

$${}_{O_B} \mathbf{J}_{r_i} = \mathbf{J}_{r_i} + m_{r_i} \|\mathbf{p}_{r_i}\|^2 \mathcal{I}_3 - \mathbf{p}_{r_i} \mathbf{p}_{r_i}^\top \quad (40d)$$

$$\mathbf{R}_{Br_i} = \mathbf{R}_z(\gamma_i) \mathbf{R}_z(\theta_i) \mathbf{R}_x(\beta_i) \quad (40e)$$

$$\mathbf{J} = \mathbf{J}_c + \sum_{i=1}^n (\mathbf{R}_{Br_i} ({}_{O_B} \mathbf{J}_t + {}_{O_B} \mathbf{J}_{r_i}) \mathbf{R}_{Br_i}^\top) \quad (41)$$

### A.2 Angular acceleration dynamics

The dynamics of the angular acceleration error can be expanded to

$$\begin{aligned} {}_B \dot{\mathbf{e}}_\psi &= {}_B (\dot{\psi}_{WB}) + [{}_B (\dot{\omega}_{WB})]_\times \mathbf{R}_{BW} W \omega_{WB_d} + [{}_B \omega_{WB}]_\times \dot{\mathbf{R}}_{BW} W \omega_{WB_d} \\ &\quad + [{}_B \omega_{WB}]_\times \mathbf{R}_{BW} W (\dot{\omega}_{WB_d}) - \dot{\mathbf{R}}_{BW} W \psi_{WB_d} - \mathbf{R}_{BW} W (\dot{\psi}_{WB_d}) \\ &= \mathbf{J}^{-1} \left( {}_B (\dot{\tau}) - [[{}_B \omega_{WB}]_\times \mathbf{r}_{com}]_\times {}_B \mathbf{f} - [\mathbf{r}_{com}]_\times {}_B (\dot{\mathbf{f}}) \right) \\ &\quad \mathbf{J}^{-1} \left( -2 [{}_B \omega_{WB}]_\times \mathbf{J} {}_B \psi_{WB} - [{}_B \psi_{WB}]_\times \mathbf{J} {}_B \omega_{WB} - [{}_B \omega_{WB}]_\times^2 \mathbf{J} {}_B \omega_{WB} \right) \\ &\quad + [{}_B \omega_{WB}]_\times {}_B \psi_{WB} + [{}_B \psi_{WB}]_\times \mathbf{R}_{BW} W \omega_{WB_d} - [{}_B \omega_{WB}]_\times \mathbf{R}_{BW} W \omega_{WB_d} \\ &\quad + 2 [{}_B \omega_{WB}]_\times \mathbf{R}_{BW} W \psi_{WB_d} - \mathbf{R}_{BW} W \zeta_{WB_d} \end{aligned} \quad (42)$$

### A.3 Linearization

Feedback linearization of angular acceleration error dynamics:

$$\begin{aligned}
{}_B(\dot{\boldsymbol{\tau}}) = & [[{}_B\boldsymbol{\omega}_{WB}]_{\times} {}_B\mathbf{r}_{com}]_{\times} {}_B\mathbf{f} + [{}_B\mathbf{r}_{com}]_{\times} {}_B(\dot{\mathbf{f}}) + 2[{}_B\boldsymbol{\omega}_{WB}]_{\times} \mathbf{J} {}_B\boldsymbol{\psi}_{WB} \\
& + [{}_B\boldsymbol{\psi}_{WB}]_{\times} \mathbf{J} {}_B\boldsymbol{\omega}_{WB} + [{}_B\boldsymbol{\omega}_{WB}]_{\times}^2 \mathbf{J} {}_B\boldsymbol{\omega}_{WB} \\
& + \mathbf{J} (-[{}_B\boldsymbol{\omega}_{WB}]_{\times} {}_B\boldsymbol{\psi}_{WB} - [{}_B\boldsymbol{\psi}_{WB}]_{\times} \mathbf{R}_{BW} \mathbf{I} \boldsymbol{\omega}_{WB_d} + [{}_B\boldsymbol{\omega}_{WB}]_{\times}^2 \mathbf{R}_{BW} \mathbf{I} \boldsymbol{\omega}_{WB_d}) \\
& + \mathbf{J} (-2[{}_B\boldsymbol{\omega}_{WB}]_{\times} \mathbf{R}_{BW} \mathbf{I} \boldsymbol{\psi}_{WB_d} + \mathbf{R}_{BW} \mathbf{I} \dot{\boldsymbol{\zeta}}_{WB_d}) + \mathbf{J} \begin{bmatrix} \bar{u}_4 \\ \bar{u}_5 \\ \bar{u}_6 \end{bmatrix}
\end{aligned} \tag{43}$$

Using eq. (43) in combination with eq. (42) yields a feedback linearized dynamic system of the angular acceleration error dynamics, see eq. (24).

Finally, we can write a linear representation of the error dynamics as follows:

$$\frac{d}{dt} \begin{bmatrix} W\mathbf{e}_p \\ W\mathbf{e}_{p,i} \\ W\mathbf{e}_v \\ W\mathbf{e}_a \\ B\mathbf{e}_R \\ B\mathbf{e}_{R,i} \\ B\mathbf{e}_\omega \\ B\mathbf{e}_\psi \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & \mathcal{I} & 0 & 0 & 0 & 0 & 0 \\ \mathcal{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} & 0 \\ 0 & 0 & 0 & 0 & \mathcal{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_A \begin{bmatrix} W\mathbf{e}_p \\ W\mathbf{e}_{p,i} \\ W\mathbf{e}_v \\ W\mathbf{e}_a \\ B\mathbf{e}_R \\ B\mathbf{e}_{R,i} \\ B\mathbf{e}_\omega \\ B\mathbf{e}_\psi \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \mathcal{I} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \mathcal{I} \end{bmatrix}}_B \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \\ \bar{u}_4 \\ \bar{u}_5 \\ \bar{u}_6 \end{bmatrix} \tag{44}$$

where  $\mathcal{I}$  and 0 are identity and zero matrices in  $\mathbb{R}^{3 \times 3}$ .

#### A.4 Stability proof of the proposed LQRI controller

We prove the stability of the proposed control law eq. (27) using Lyapunov stability theory. The closed loop system can be written as

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{B}\mathbf{K}_{LQRI})\mathbf{e} \tag{45}$$

We define the Lyapunov function:

$$V(\mathbf{e}) = \mathbf{e}^T \mathbf{P} \mathbf{e} \quad V(\mathbf{e}) \in \mathbb{R} \tag{46}$$

According to Lyapunov stability theory, the system is asymptotically stable, if  $\dot{V}(\mathbf{e}) < 0$ ,  $\forall \mathbf{e} \neq 0$ . The time derivative is computed to be:

$$\begin{aligned}
\dot{V}(\mathbf{e}) &= \dot{\mathbf{e}}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \dot{\mathbf{e}} \\
&= \mathbf{e}^T (\mathbf{A} - \mathbf{B}\mathbf{K}_{LQRI})^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} (\mathbf{A} - \mathbf{B}\mathbf{K}_{LQRI}) \mathbf{e} \\
&= -\mathbf{e}^T (\mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}) \mathbf{e}
\end{aligned} \tag{47}$$

Note that due to  $\mathbf{Q} \succeq 0$ ,  $\mathbf{R} \succ 0$ ,  $\mathbf{P} \succ 0$  and the structure of  $\mathbf{B}$ , the matrix  $\mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}$  is always positive definite. Therefore, we have  $\dot{V}(\mathbf{e}) < 0$ ,  $\forall \mathbf{e} \neq 0$  and the closed loop system is asymptotically stable.

Since real system dynamics are nonlinear, the closed loop system needs to be written as follows:

$$\dot{\mathbf{e}} = f(\mathbf{e}, \mathbf{u} = -\mathbf{K}_{LQRI}\mathbf{e}) \tag{48}$$

with the Taylor series of the function  $f$  evaluated at the origin:

$$f(\mathbf{e}, -\mathbf{K}_{LQRI}\mathbf{e}) = \mathbf{A}\mathbf{e} - \mathbf{B}\mathbf{K}_{LQRI}\mathbf{e} + \mathbf{o}|_0(\mathbf{e}^T \mathbf{e}) \tag{49}$$

As above, we can derive the time derivative of the Lyapunov function:

$$\begin{aligned}
\dot{V}(\mathbf{e}) &= \dot{\mathbf{e}}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \dot{\mathbf{e}} \\
&= -\mathbf{e}^T (\mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}) \mathbf{e} + 2\mathbf{e}^T \mathbf{P} \mathbf{o}|_0(\mathbf{e}^T \mathbf{e})
\end{aligned} \tag{50}$$

For the above expression to be  $< 0$ , the following must hold:

$$\begin{aligned}
2\mathbf{e}^T \mathbf{P} \mathbf{o}|_0(\mathbf{e}^T \mathbf{e}) &\stackrel{!}{<} \mathbf{e}^T (\mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}) \mathbf{e} \\
\|2\mathbf{e}^T \mathbf{P} \mathbf{o}|_0(\mathbf{e}^T \mathbf{e})\| &\stackrel{!}{<} \|\mathbf{e}^T (\mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}) \mathbf{e}\| \\
2\|e\| \|\mathbf{P}\| \|\mathbf{o}|_0(\mathbf{e}^T \mathbf{e})\| &\stackrel{!}{<} \|e\|^2 \lambda_{\min}(\mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P}) \\
\frac{\|\mathbf{o}|_0(\mathbf{e}^T \mathbf{e})\|}{\|e\|} &\stackrel{!}{<} \frac{\lambda_{\min}(\mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P})}{2\|\mathbf{P}\|}
\end{aligned} \tag{51}$$

where  $\lambda_{\min}(\cdot)$  denotes the minimum eigenvalue of  $(\cdot)$ . Looking at the dynamics in eq. (25), it is clear that all higher order terms are coming from  $\dot{e}_{R,B}$ , i.e.  $\|\mathbf{o}|_0(e^T e)\| = \|\mathbf{o}_{R|0}(e^T e)\|$ . Lee (2012) shows that

$$\|\dot{e}_{R,B}\| \leq \frac{3}{\sqrt{2}} \|{}_B e_\omega\| \quad (52)$$

This can be extended to

$$\begin{aligned} \|\dot{e}_{R,B}\| &\leq \frac{3}{\sqrt{2}} \|{}_B e_\omega\| \\ \|\|{}_B e_\omega + \mathbf{o}_{R|0}(e^T e)\|\| &\leq \frac{3}{\sqrt{2}} \|{}_B e_\omega\| \\ \|\|\mathbf{o}_{R|0}(e^T e)\|\| - \|{}_B e_\omega\| &\leq \frac{3}{\sqrt{2}} \|{}_B e_\omega\| \\ \|\|\mathbf{o}_{R|0}(e^T e)\|\| &\leq \frac{3 + \sqrt{2}}{\sqrt{2}} \|{}_B e_\omega\| \end{aligned} \quad (53)$$

Plugging this into the equation above, we finally get the asymptotic stability condition:

$$\frac{3 + \sqrt{2}}{\sqrt{2}} \frac{\|{}_B e_\omega\|}{\|e\|} \stackrel{!}{<} \frac{\lambda_{\min}(\mathbf{Q} + \mathbf{PBR}^{-1}\mathbf{B}^T\mathbf{P})}{2\|\mathbf{P}\|} \quad (54)$$

Looking at eq. (54), we see that a small angular velocity error  ${}_B e_\omega$  will make the condition more likely to be fulfilled and thus increase the overall system stability. This can be achieved to some extent by increasing the corresponding weights in the  $\mathbf{Q}$  matrix and thereby changing the objective function for eq. (16) accordingly. Real-world experiments have confirmed this behaviour, as generally speaking, higher angular velocity gains lead to less oscillations.

Note however, that eq. (54) is a very conservative and only sufficient condition, meaning that stability might be achieved, even when it is violated.

### A.5 General formulation of the static allocation matrix

For a general tiltrotor MAV with  $n$  arms we can write the relation between the body force and torque and the rotor force components as follows:

$$\begin{bmatrix} {}_B \mathbf{f} \\ {}_B \boldsymbol{\tau} \end{bmatrix} = \mathbf{A} \tilde{\boldsymbol{\Omega}} \quad (55)$$

where the rotor force components  $\mathbf{u}$  are defined as the lateral and vertical components of the squared rotor speeds:

$$\tilde{\boldsymbol{\Omega}} = \begin{bmatrix} \sin(\alpha_i)\Omega_i \\ \cos(\alpha_i)\Omega_i \\ \vdots \\ \sin(\alpha_n)\Omega_n \\ \cos(\alpha_n)\Omega_n \end{bmatrix} \quad \forall i \in \{1 \dots n\} \quad (56)$$

The static allocation matrix  $\mathbf{A}$  can then be written as follows:

$$\mathbf{A} = c_f \begin{bmatrix} \sin(\gamma_i) & -\sin(\beta_i) \cos(\gamma_i) & \dots \\ -\cos(\gamma_i) & -\sin(\beta_i) \sin(\gamma_i) & \dots \\ 0 & \cos(\beta_i) & \dots \\ -s_j c_d \sin(\gamma_i) + l_x \sin(\beta_i) \cos(\gamma_i) & l_x \sin(\gamma_i) + s_i c_d \sin(\beta) \cos(\gamma) & \dots \\ s_i c_d \cos(\gamma_i) + l_x \sin(\beta_i) \sin(\gamma_i) & -l_x \cos(\gamma_i) + s_i c_d \sin(\beta) \sin(\gamma) & \dots \\ -l_x \cos(\beta_i) & -s_i c_d \cos(\beta_i) & \dots \end{bmatrix} \quad (57)$$

where  $\gamma_i$  are the angles between the body  $x$  axis and arm  $i$  in the body  $x$ - $y$ -plane,  $\beta_i$  are the angles between the body  $x$ - $y$  plane and arm  $i$ , as illustrated in fig. 1.  $c_f$  is the rotor thrust coefficient,  $c_d$  is the rotor drag coefficient (relative to  $c_f$ ), and  $s_i$  are rotor spin directions of rotors attached to arms  $i$ .