A Global Strategy for Tailsitter Hover Control

Robin Ritz and Raffaello D'Andrea

Abstract We present a nonlinear hover controller for a small flying wing tailsitter vehicle, which enables recovering to hover from a large set of initial conditions. The proposed attitude control law is obtained by solving an optimal control problem, with the objective of correcting large attitude errors by turning primarily around the vehicle's strongly actuated axis. Solutions for a set of initial attitudes are precomputed and stored in a lookup table. For each controller update, the optimal inputs are read from this table, and applied to the system in an MPC-like manner. Simulation results indicate that this control method is able to perform recoveries to hover from any initial attitude, given that the initial velocity of the vehicle is below a certain limit. Further, the performance of the control strategy is demonstrated on a small tailsitter vehicle in the ETH Zurich Flying Machine Arena.

1 Introduction

There is an increasing demand to apply hover-capable flying machines to long-range and long-endurance missions. Traditional multicopters use rotors to produce lift and overcome gravity, which is, compared to fixed-wing airplanes, inefficient in terms of both required energy per distance flown and required energy per flight time, and thus a limiting factor of the operating range and flight duration [1]. In order to overcome this limitation, powered lift aircrafts [2] such as the so-called *tailsitter* vehicle [3, 4] have been suggested. A tailsitter is able to take off and land vertically on its tail with the nose and thrust direction pointing upwards. For fast forward flight, the vehicle

Robin Ritz
Institute for Dynamic Systems and Control, ETH Zürich, Sonneggstrasse 3, 8092 Zürich, CH
e-mail: rritz@ethz.ch

Raffaello D'Andrea
Institute for Dynamic Systems and Control, ETH Zürich, Sonneggstrasse 3, 8092 Zürich, CH
e-mail: rdandrea@ethz.ch

tilts to a near-horizontal attitude resulting in a more efficient lift production with conventional wings. Compared to other powered lift aircraft types (such as tiltrotors [5, 6] or tiltwings [7, 8]), the major advantage of a tailsitter is its mechanical simplicity; no mechanism for changing the direction of the propulsive system has to be added, saving weight and reducing susceptibility to malfunctions. As a result of the availability of cheap, lightweight electronic components and the numerous potential applications of such small hybrid vehicles[1], many researchers and companies have recently started research programs exploring the capabilities of these flying machines. For example, the company Transition Robotics is selling the tailsitter vehicle Quadshot for the hobby and research market [9]. Last August, the team of Google's Project Wing tested a tailsitter prototype for a packet delivery service vehicle.[2] However, recently Google announced that the tailsitter wing design approach was scrapped; the project leaders came to the conclusion that it is still too difficult to control such a vehicle in a reliable and robust manner.[3]

Over the last decades, the research community has developed many successful control strategies for small unmanned aerial vehicles including quadrocopters and conventional fixed-wing airplanes (see for example [10, 11] and references therein). However, relatively little attention has been paid to small powered lift aircrafts such as tailsitters, where the large flight envelope and the highly nonlinear dynamics introduce additional challenges for control design. The problem of attitude control for tailsitters or similar vehicles has been addressed, for example, in [12, 13, 14, 15, 16, 17]. In order to avoid singularities, typically a quaternion representation for the vehicle's attitude is used, combined with linear feedback on the quaternion error vector to obtain the desired body rates. However, alternative approaches of representing and controlling attitude exist, such as the resolved tilt-twist method leading to better tracking performance when large attitude errors occur [15, 16]. As computational units become more powerful and less expensive, model predictive control (MPC) [18] has become a viable approach for controlling systems with fast dynamics such as small flying machines; first results of an MPC-based controller for a tailsitter in hover flight have been published in [19].

In this paper, we address the problem of designing a nonlinear hover controller for a small flying wing tailsitter vehicle, which should be capable of recovering to hover from any initial attitude. The challenges of this task lie in the fact that typically the vehicle's actuators operate close to their saturation limits, and in the fact that the rotation axis along the wing is weakly actuated and might be dominated by aerodynamic torques. As mentioned above, most traditional tailsitter attitude control methods are based on linear quaternion feedback or similar strategies, which provide under some assumptions global asymptotic stability. However, for large attitude errors these methods have difficulties to properly account for the different magnitudes of attainable torques around the different rotation axes. Due to the nonlinear nature of the attitude dynamics, these algebraic feedback laws typically do not result

[1] We refer *hybrid vehicles* to vehicles that provide both hover-capabilities and wings for aerodynamic lift production.

[2] http://www.bbc.com/news/technology-28964260/

[3] http://blogs.wsj.com/digits/2015/03/17/google-working-on-new-drone-after-wing-design-failed/

in an optimal maneuver. In order to overcome this limitation, we propose a control strategy which plans trajectories in such a way that the vehicle exploits its strongly actuated axis (the axis actuated by the propellers) if it has to recover from large attitude errors. The control law is obtained by solving an optimal control problem that minimizes a quadratic cost with a particular structure. Since the computation is not feasible in real-time, solutions for a set of initial attitudes are precomputed and stored in a map. For each controller update, the optimal inputs are then read from this lookup table, and fed to the system in an MPC-like manner. We do not prove stability of the presented controller, however the performance of the control strategy is analyzed in a simulation environment and the results indicate that the tailsitter is able recover to hover from any initial attitude, given that the initial velocity does not exceed a certain limit. Furthermore, the effectiveness of the control strategy is demonstrated in the ETH Zurich Flying Machine Arena.

The remainder of the paper is structured as follows: In Section 2 we present a dynamic model for a small flying wing tailsitter. Section 3 introduces a nonlinear hover control strategy for the tailsitter vehicle. Simulations and experimental results are presented in Section 4, and we conclude in Section 5.

2 Flying Wing Tailsitter Model

In this section, we derive a model for the dynamics of a small flying wing tailsitter vehicle. Due to the large operating range and complex aerodynamic forces and torques acting on the vehicle, deriving an adequate dynamic model for a tailsitter is a challenging task. Typically, the complex aerodynamic properties are obtained either by CFD methods [20, 17], by measurement series covering all relevant operating points [21], or by first-principles models combined with heuristics that capture some of the unmodeled effects [22]. Since herein we focus on control design, we follow a similar approach as proposed in [22] and derive a first-principles model of the considered tailsitter vehicle.

The vehicle is actuated by two propellers, one in front of each wing, and two flaps located at the wings' trailing edges. An illustration of the tailsitter is shown in Fig.1. The control inputs are the propeller forces $f_{prop,l}$ and $f_{prop,r}$, and the flap angles $\delta_{flap,l}$ and $\delta_{flap,r}$, and all four control inputs are subject to saturations:

$$\begin{aligned} f_{prop,min} \leq f_{prop,l}, f_{prop,r} \leq f_{prop,max}, \\ \delta_{flap,min} \leq \delta_{flap,l}, \delta_{flap,r} \leq \delta_{flap,max}. \end{aligned} \tag{1}$$

We assume that the vehicle's airspeed is small, such that the range of attainable propeller forces $[f_{prop,min}, f_{prop,max}]$ can be considered to be constant. The propellers are of fixed-pitch type and the motors are not able to reverse direction mid-flight, meaning that the propellers cannot produce negative thrust, i.e. $f_{prop,min} > 0$

We introduce a body-fixed coordinate frame B with origin at the vehicle's center of mass, as shown in Fig. 1. The z-axis of the body frame B is aligned with

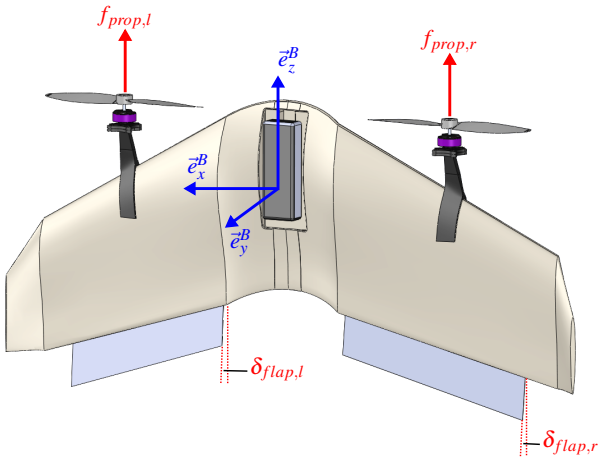

Fig. 1 Illustration of a flying wing tailsitter vehicle. The vehicle is actuated by two propellers that produce forces along the body z-axis, and two flaps that produce aerodynamic forces by deflecting the airflow over the wings.

the thrust direction, the x-axis points along the left wing, and the y-axis completes the right-handed coordinate system. We denote unit vectors along the axes of the body frame as $\vec{e}_x^B$, $\vec{e}_y^B$, and $\vec{e}_z^B$, respectively. The position of the vehicle's center of mass relative to an inertial coordinate frame I, expressed in this inertial frame I, is denoted as ${}_I\vec{p} = (p_x, p_y, p_z)$. To simplify notation, vectors may be expressed as n-tuples $\vec{x} = (x_1, x_2, \ldots, x_n)$, with dimension and stacking clear from context. The tailsitter's attitude is described by a unit quaternion $\vec{q} = (q_0, q_1, q_2, q_3)$ that represents a rotation from the inertial frame I to the body-fixed frame B. For more information on representing attitudes with unit quaternions, see for example [23] and references therein. For the remainder of the paper, unless otherwise stated, we will express all quantities in the body frame B. If a vector $\vec{x}$ is expressed in the inertial frame I, the notation ${}_I\vec{x}$ will be used. The translational velocity of the body frame B relative to the inertial frame I is denoted as $\vec{v} = (v_x, v_y, v_z)$, and the rotational body rates are denoted as $\vec{\omega} = (\omega_x, \omega_y, \omega_z)$. The position and attitude kinematics are

$$\begin{aligned} {}_I\dot{\vec{p}} &= R^T \vec{v}, \\ \dot{\vec{q}} &= \frac{1}{2} W^T \vec{\omega}, \end{aligned} \tag{2}$$

where R denotes the rotation matrix from the inertial frame I to the body frame B, and W is the quaternion rate matrix [23]. The vehicle is modeled as a rigid body with mass M and rotational inertia J, and the dynamics are thus given by the Newton-Euler equations:

$$\begin{aligned} M\dot{\vec{v}} &= \vec{f}_{tot} - \vec{\omega} \times M\vec{v}, \\ J\dot{\vec{\omega}} &= \vec{m}_{tot} - \vec{\omega} \times J\vec{\omega}, \end{aligned} \tag{3}$$

where $\vec{f}_{tot}$ and $\vec{m}_{tot}$ denote the external force and torque vector, respectively, acting on the vehicle. The external force is modeled as

$$\vec{f}_{tot} = \vec{f}_{air} + (f_{prop,l} + f_{prop,r})\vec{e}_z^B - Mg\vec{e}_z^I, \tag{4}$$

where $\vec{f}_{air}$ denotes the aerodynamic force, g is the gravitational acceleration, and $\vec{e}_z^B$ and $\vec{e}_z^I$ are the unit vectors along the z-axis of the corresponding frame.

Similarly, the external torque acting on the vehicle is given by

$$\vec{m}_{tot} = \vec{m}_{air} + x_{prop}(f_{prop,r} - f_{prop,l})\vec{e}_y^B + \kappa_{prop}(f_{prop,r} - f_{prop,l})\vec{e}_z^B, \tag{5}$$

where $\vec{m}_{air}$ denotes the aerodynamic torque, x_{prop} is the x-axis offset of the propellers, and κ_{prop} is the propeller's torque-to-thrust ratio (the left propeller rotates counter-clockwise, and the right propeller rotates clockwise). Note that we assume that the offset of the propellers in the y-direction is negligible.

Typically, the aerodynamic force $\vec{f}_{air}$ and torque $\vec{m}_{air}$ are complex functions of the vehicle state and control input. However, in order to keep the problem tractable and applicable to the MPC-like control strategy that will be introduced in Section 3, we apply the following first-principles aerodynamic model: For each wing the aerodynamic force and torque is computed separately, assuming that the air velocity is uniform over the wing, and assuming that there is no cross coupling between the left and right wing. First, we compute the total velocity of the corresponding wing:

$$\vec{v}_{wing} = \vec{v} - \vec{p}_{wing} \times \vec{\omega}, \tag{6}$$

where $\vec{p}_{wing}$ denotes the reference position of the wing relative to the body frame B. The propellers of the tailsitter are mounted in front of the wings; consequently the air is accelerated along the negative body z-axis if the propeller produce a positive thrust force. For simplicity, we assume that the entire wing is in the propeller streamtube. For low airspeeds, the z-component of the total airspeed including propeller-induced speedup can be approximated using Momentum Theory [24]:

$$v_{wing,tot,z} = \sqrt{\frac{2f_{prop}}{\rho_{air}A_{prop}} + v_{wing,z}^2}, \tag{7}$$

where ρ_{air} denotes the density of air, and A_{prop} is the propeller area. We neglect that the Momentum Theory approximation (7) becomes less accurate as $v_{wing,z}$ becomes negative. The angle of attack α_{wing} and the reference airspeed $v_{wing,ref}$ of the corresponding wing are then defined as [25]

$$\alpha_{wing} = -\text{atan2}(v_{wing,y}, v_{wing,tot,z}), \quad v_{wing,ref} = \sqrt{v_{wing,y}^2 + v_{wing,tot,z}^2}. \tag{8}$$

The aerodynamic force is modeled as

$$\vec{f}_{wing} = \vec{c}_{air}(\alpha_{wing}, \delta_{flap}) v^2_{wing,ref}, \tag{9}$$

where $\vec{c}_{air}$ denotes a coefficient vector, and the arguments $(\alpha_{wing}, \delta_{flap})$ are stated explicitly in order to highlight that the coefficients are a function of angle of attack and flap angle. When modeling the aerodynamic coefficients $\vec{c}_{air}$, we make two simplifying assumptions; 1) the wing does not produce a force component along the body x-axis, and 2) the flap deviates the airflow behind the wing by a small angle proportional to the flap angle deviation δ_{flap}:

$$\vec{c}_{air}(\alpha_{wing}, \delta_{flap}) = \left(c_y(\alpha_{wing}) + c_{y,\delta}\,\delta_{flap}\right)\vec{e}^{\,B}_y + c_z(\alpha_{wing})\vec{e}^{\,B}_z, \tag{10}$$

where the function $c_y(\alpha_{wing})$ corresponds to the lift coefficient of the wing, the constant $c_{y,\delta}$ describes first order effects of flap angle deviations on the lift coefficient, and the function $c_z(\alpha_{wing})$ corresponds to the drag coefficient of the wing. Similarly, the aerodynamic torque is modeled as

$$\vec{m}_{wing} = \vec{d}_{air}(\alpha_{wing}, \delta_{flap}) v^2_{wing,ref}, \tag{11}$$

with

$$\begin{aligned}\vec{d}_{air}(\alpha_{wing}, \delta_{flap}) = &\left(d_x(\alpha_{wing}) + d_{x,\delta}\,\delta_{flap}\right)\vec{e}^{\,B}_x + d_y(\alpha_{wing})\vec{e}^{\,B}_y \\ &+ \left(d_z(\alpha_{wing}) + d_{z,\delta}\,\delta_{flap}\right)\vec{e}^{\,B}_z,\end{aligned} \tag{12}$$

where the functions $d_x(\alpha_{wing})$, $d_y(\alpha_{wing})$, and $d_z(\alpha_{wing})$ are wing characteristics, and the constants $d_{x,\delta}$ and $d_{z,\delta}$ describe first order effects of flap angle deviations. This completes the derivation of the first-principles model for the aerodynamic effects acting on one of the tailsitter's wings. The total aerodynamic forces and torques yield

$$\begin{aligned}\vec{f}_{air} &= \vec{f}_{wing,left} + \vec{f}_{wing,right}, \\ \vec{m}_{air} &= \vec{m}_{wing,left} + \vec{m}_{wing,right}.\end{aligned} \tag{13}$$

Note that due to symmetry considerations the aerodynamic coefficients $c_{(\cdot)}$ and $d_{(\cdot)}$ are identical for both wings, except for some sign changes where appropriate.

3 Control Strategy

In this section, we present a control strategy for the tailsitter modeled in Section 2. First, a desired attitude and a desired thrust force is computed, and subsequently an attitude controller computes desired body rates in order to track the desired attitude. The desired body rates are then fed to an inner control loop that computes actuator commands.

3.1 Desired Attitude and Thrust Force

As proposed in [26], a desired acceleration ${}_I\vec{a}_{des}$ is computed based on position error $\Delta\vec{p}$ and velocity error $\Delta\vec{v}$. (The errors are expressed in the inertial frame I.) The control loop is shaped such that for each coordinate the system behaves like a second-order system with some desired time constant and damping ratio. In order to compensate for modeling errors and external disturbances such as wind, an additional integral state $\dot{\vec{s}} = \Delta\vec{p}$ is added. Thus, the desired acceleration is given by

$$ {}_I\vec{a}_{des} = K_s\vec{s} + K_p\Delta\vec{p} + K_v\Delta\vec{v}, \tag{14} $$

where the control gains K_s, K_p, and K_v are computed such that the desired closed-loop properties are met. Using the substitution ${}_I\vec{f}_{tot} = M_I\vec{a}_{des}$, the desired thrust vector can then be computed by rearranging (4):

$$ {}_I\vec{f}_{thrust,des} = (f_{prop,l} + f_{prop,r})\,{}_I\vec{e}_z^{\,B} = M_I\vec{a}_{des} + Mg\,{}_I\vec{e}_z^{\,I} - {}_I\vec{f}_{air}. \tag{15} $$

Note that, in order to compute the desired thrust vector, the controller needs an estimate of the current aerodynamic force ${}_I\vec{f}_{air}$, which might be an estimate or simply the current nominal value. Since the thrust force acts along $\vec{e}_z^{\,B}$, we choose the desired attitude such that the body z-axis is aligned with the desired thrust direction. A desired attitude that aligns actual and desired thrust axis is given by

$$ \vec{q}_{thrust,des} = \left(\cos\left(\theta_{des}/2\right), \vec{n}_{des}\sin\left(\theta_{des}/2\right)\right), \tag{16} $$

where θ_{des} is the desired tilt angle, and $\vec{n}_{des}$ the desired tilt rotation direction:

$$ \theta_{des} = \arccos\left({}_I\vec{e}_z^{\,I} \cdot {}_I\vec{e}_{z,des}^{\,B}\right), \quad \vec{n}_{des} = \frac{{}_I\vec{e}_z^{\,I} \times {}_I\vec{e}_{z,des}^{\,B}}{||{}_I\vec{e}_z^{\,I} \times {}_I\vec{e}_{z,des}^{\,B}||}, \tag{17} $$

where the desired body z-axis is given by

$$ {}_I\vec{e}_{z,des}^{\,B} = \frac{{}_I\vec{f}_{thrust,des}}{||{}_I\vec{f}_{thrust,des}||}. \tag{18} $$

After this tilt alignment, we can rotate the vehicle around its z-axis without changing the thrust direction, hence we can choose an arbitrary yaw angle ψ_{des}. The desired attitude yields

$$ \vec{q}_{des} = \vec{q}_{yaw,des} \cdot \vec{q}_{thrust,des}, \tag{19} $$

where $(\cdot)$ denotes the quaternion multiplication and $\vec{q}_{yaw,des}$ is given by

$$ \vec{q}_{des,yaw} = \left(\cos\left(\psi_{des}/2\right), 0, 0, \sin\left(\psi_{des}/2\right)\right). \tag{20} $$

The desired thrust force is given by the magnitude of the desired thrust vector:

$$ f_{thrust,des} = ||{}_I\vec{f}_{thrust,des}||. \tag{21} $$

3.2 Attitude Control

As mentioned above, one of the challenges when designing a tailsitter attitude controller is to cope with the limited torques that can be produced around the body x-axis, since the corresponding lever arm is small and the flap saturation boundaries are relatively tight. In addition, large undesired aerodynamic torques may act on the x-axis, which further complicates the controller design. On the other hand, the attainable torques around the y-axis are relatively high, since this torque is produced by the propellers' differential thrust. In the following, we propose a method for computing desired body rates in order to control the vehicle's attitude, while turning preferably around the better actuated y-axis.

The attitude error is given by

$$\vec{q}_{err} = \vec{q}_{est} \cdot (\vec{q}_{des})^{-1}, \tag{22}$$

where $\vec{q}_{est}$ denotes the current estimated attitude of the vehicle. For convenient notation, we will drop the error subscript in the following, i.e. we define $\vec{q} = \vec{q}_{err}$. We assume that an inner control loop perfectly tracks the body rates $\vec{\omega}$, such that we can directly set the body rates without any delay or dynamics. As mentioned in Section 2, the error quaternion kinematics are given by

$$\dot{\vec{q}} = \frac{1}{2} W^T \vec{\omega}. \tag{23}$$

The objective of the proposed attitude controller is to align the vehicle's thrust axis (i.e. the body z-axis) with the desires thrust direction. (The remaining degree of freedom, i.e. the yaw angle of the vehicle, is controlled separately and not considered here.) We define the tilt angle θ to be the angle between the desired and actual thrust direction; it is given by

$$\theta = \arccos(q_0^2 - q_1^2 - q_2^2 + q_3^2). \tag{24}$$

In order to control the tilt angle θ to zero, we choose the following cost function to be minimized:

$$J = \int_{t_0}^{t_f} c_\theta \theta^2 + (c_x + c_{x,\theta}\theta^2)\omega_x^2 + c_y\omega_y^2 + c_z\omega_z^2 \, \mathrm{dt}, \tag{25}$$

where $c_{(\cdot)}$ indicates constant, positive weight parameters, and the different terms are explained in the following: The first term $c_\theta\theta^2$ quadratically penalizes the tilt angle that should be controlled to zero. The second term $(c_x + c_{x,\theta}\theta^2)\omega_x^2$ penalizes the control effort around the x-axis. As mentioned above, the attainable torques around this axis are subject to tight bounds; therefore we want to avoid that large tilt errors are corrected by turning around this axis. Thus, the weight on the control input ω_x is not constant, but contains a term that grows quadratically with the tilt angle θ, such that large errors are corrected by turning mainly around the other two axes. Finally,

the third and fourth term $c_y\omega_y^2$ and $c_z\omega_z^2$, respectively, penalize the inputs around the remaining two axes.

For a given initial attitude error $\vec{q}_{ini}$, the optimal control inputs $\vec{\omega}^*$ solve the optimization problem

$$\begin{aligned} &\text{minimize} \;\; J \\ &\text{subject to} \;\; \dot{\vec{q}} = \frac{1}{2} W^T \vec{\omega}, \\ &\qquad\qquad \vec{q}(t_0) = \vec{q}_{ini}, \\ &\qquad\qquad \vec{\omega}(t) \in \mathbb{R}^3 \; \forall \, t \in [t_0, t_f]. \end{aligned} \tag{26}$$

In order to simplify this optimization problem, we leverage Pontryagin's Minimum Principle [27] to derive necessary conditions for optimality, which are then used for computing candidate optimal solutions. The Hamiltonian of the above problem is given by

$$H = c_\theta \theta^2 + (c_x + c_{x,\theta}\theta^2)\omega_x^2 + c_y\omega_y^2 + c_z\omega_z^2 + \frac{1}{2}\vec{\lambda}^T W^T \vec{\omega}, \tag{27}$$

where $\vec{\lambda} = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ denotes the costate vector. The costate equation $\dot{\vec{\lambda}} = -\nabla_{\vec{q}} H$ yields

$$\begin{aligned} \dot{\lambda}_0 &= -\tfrac{\mathrm{d}}{\mathrm{d}q_0} H = (\lambda_0 k - \lambda_1\omega_x - \lambda_2\omega_y - \lambda_3\omega_z)/2, \\ \dot{\lambda}_1 &= -\tfrac{\mathrm{d}}{\mathrm{d}q_1} H = (-\lambda_1 k + \lambda_0\omega_x - \lambda_3\omega_y + \lambda_2\omega_z)/2, \\ \dot{\lambda}_2 &= -\tfrac{\mathrm{d}}{\mathrm{d}q_2} H = (-\lambda_2 k + \lambda_3\omega_x + \lambda_0\omega_y - \lambda_1\omega_z)/2, \\ \dot{\lambda}_3 &= -\tfrac{\mathrm{d}}{\mathrm{d}q_3} H = (\lambda_3 k - \lambda_2\omega_x + \lambda_1\omega_y + \lambda_0\omega_z)/2, \end{aligned} \tag{28}$$

with the shorthand notation

$$k = \frac{8(c_\theta + c_{x,\theta}\omega_x^2)\theta}{\sqrt{1 - (q_0^2 - q_1^2 - q_2^2 + q_3^2)^2}}. \tag{29}$$

Since the final state $\vec{q}(t_f)$ is free and costless, the costates satisfy the final condition $\vec{\lambda}(t_f) = (0,0,0,0)$. According to the Minimum Principle, the optimal inputs $\vec{\omega}^*$ minimize the Hamilton over the set of attainable controls. We do not impose any constraints on the body rates $\vec{\omega}$, and the cost function is quadratic in $\vec{\omega}$ with positive weights $c_{(\cdot)}$. Thus, an expression for the optimal body rates $\vec{\omega}^*$ can be obtained by setting the derivative of the Hamiltonian with respect to $\vec{\omega}$ to zero:

$$\begin{aligned} \omega_x^* &= \frac{-\lambda_1 q_0 + \lambda_0 q_1 + \lambda_3 q_2 - \lambda_2 q_3}{4(c_x + c_{x,\theta}\theta^2)}, \\ \omega_y^* &= \frac{-\lambda_2 q_0 - \lambda_3 q_1 + \lambda_0 q_2 + \lambda_1 q_3}{4c_y}, \\ \omega_z^* &= \frac{-\lambda_3 q_0 + \lambda_2 q_1 - \lambda_1 q_2 + \lambda_0 q_3}{4c_z}. \end{aligned} \tag{30}$$

By substituting the expression for the optimal body rates (30) into the quaternion kinematics (23), the optimization problem (26) can be written as a boundary value problem (BVP):

$$\begin{aligned} \dot{\vec{q}} &= \frac{1}{2} W \vec{\omega}^*, \quad \dot{\vec{\lambda}} = -\nabla_{\vec{q}} H, \\ \vec{q}(t_0) &= \vec{q}_{ini}, \quad \vec{\lambda}(t_f) = \vec{0}. \end{aligned} \tag{31}$$

Thus, we can obtain candidate optimal solutions to the optimization problem (26), by numerically solving BVP (31).[4]

On an Intel i7-3520M processor, it typically takes between 10 s and 30 s to compute a solution to BVP (31), which is not fast enough for real-time applications. Therefore, we compute trajectories for a set of initial attitudes $\{\vec{q}_{ini}\}$ and create a lookup table for the candidate optimal body rates $\vec{\omega}^*$ at the beginning of the trajectory. Since in the cost function (25) the yaw angle is not penalized, we can arbitrarily rotate the inertial reference frame I around its z-axis without changing the body rates $\vec{\omega}^*$ of the corresponding solution to BVP (31). Consequently, we can always rotate the reference frame I such that the quaternion error around the z-axis is zero, i.e. $q_{ini,3} = 0$. Hence, the state space for which control inputs need to be computed is two-dimensional. Each point in this reduced two-dimensional state space is defined by a tilt angle $\theta \in [0, \pi]$ and a tilt direction $\phi \in [0, 2\pi]$, where a tilt rotation around the x-axis corresponds to $\phi = 0$, and a tilt rotation around the y-axis corresponds to $\phi = \pi/2$. Symmetry considerations indicate that it is sufficient to compute body rates for $\phi \in [0, \pi/2]$, and map these solutions onto the full range $\phi \in [0, 2\pi]$ using appropriate coordinate transformations. We thus choose a uniformly sampled set over the space $\{(\theta, \phi) \mid \theta \in [0, \pi], \phi \in [0, \pi/2]\}$, which defines the set of initial attitudes $\{\vec{q}_{ini}\}$ for which solutions to BVP (31) are computed.

Fig. 2 shows the resulting candidate optimal body rates $\vec{\omega}^*$ as function of tilt angle θ and tilt direction ϕ, for a map that was computed as described above. For each controller update, the desired body rates $\vec{\omega}_{des}$ for the current attitude error are read from the precomputed maps using linear interpolation and subsequently sent to the inner control loop. As we can see in Fig. 2, the maps for the desired body rates $\vec{\omega}_{des}$ are relatively smooth. Hence, we could approximate the desired body rates by fitting some functions with a small number of parameters into the maps. For example in an onboard implementation where memory is limited, this approach might be beneficial.

3.3 Body Rate Control

An inner body rate controller tracks the desired body rates $\vec{\omega}_{des}$ using rate gyroscope measurements $\vec{\omega}_{meas}$. First, the range of allowed propeller forces is adjusted to the current flight situation, and then the four actuator commands are computed.

[4] Computations are executed with Matlab [28], using the function *'bvp4c'*.

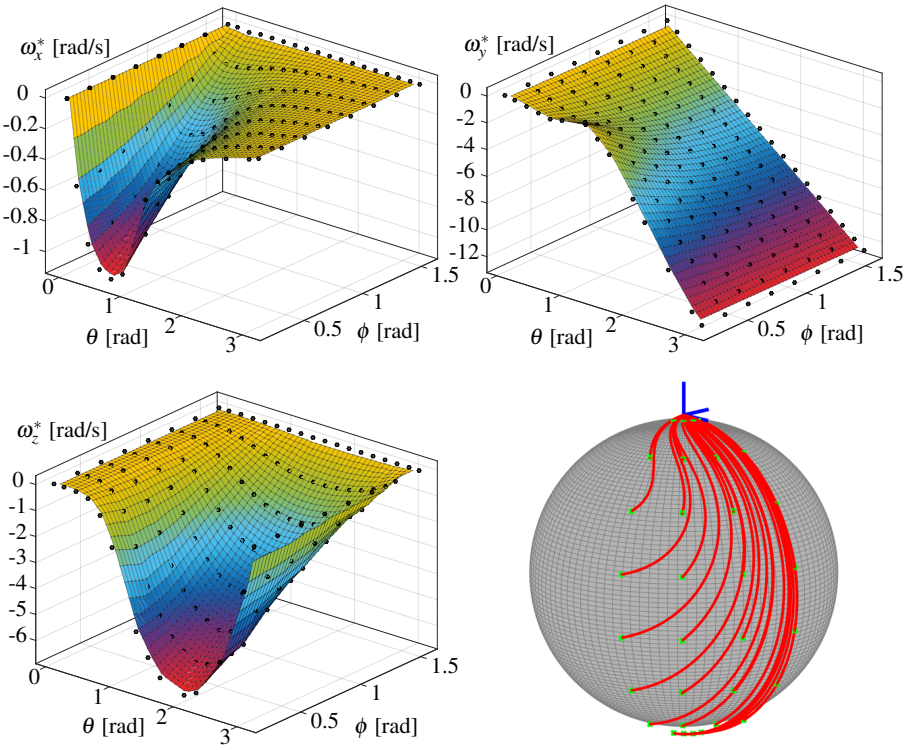

Fig. 2 Map of candidate optimal body rates $\vec{\omega}^*$ for x-axis (top left), y-axis (top right), and z-axis (bottom left). The black dots indicate elements of $\{\vec{q}_{ini}\}$ for which BVP (31) has been solved. The bottom right drawing shows a selection of candidate optimal tilt trajectories. The top of the sphere corresponds to $\theta = 0$, and the bottom to $\theta = \pi$. We can observe that the trajectories do not correspond to the shortest rotation towards zero tilt, because the body rates have different weights in the cost function (25).

3.3.1 Propeller Force Boundaries

As mentioned in Section 2, due to actuator saturations the attainable propeller forces are constrained to the range $f_{prop} \in [f_{prop,min}, f_{prop,max}]$. However, in order to ensure that the flaps do not loose effectiveness, we pose two additional constraints on the propeller forces: Firstly, since the torques produced by the flaps scale quadratically with the reference airspeed $v_{wing,ref}$ (as defined in (8)), we choose a minimum required reference airspeed over the wings. For each wing, this defines an additional lower bound on the propeller force f_{prop}, which can be computed using (7) and (8) and depends on the current velocity and body rates of the vehicle. Secondly, we choose a maximum allowable angle of attack α_{wing} (as defined in (8)). Doing so, we can avoid that the flaps loose their effectiveness due to stall phenomena at large angles of attack. Again, this defines an additional lower bound on the propeller force f_{prop} given by (7) and (8) and depending on the current state of the vehicle.

Even though these two additional constraints might narrow the range of allowed propeller forces considerably, experimental results show that, to some extent, the benefits of effective flaps outweigh this drawback.

3.3.2 Actuator Commands

The body rate controller is designed such that the elements of the body rate error $(\vec{\omega}_{meas} - \vec{\omega}_{des})$ follow three decoupled first order systems with desired time constants $\vec{\tau}_\omega = (\tau_{\omega,x}, \tau_{\omega,y}, \tau_{\omega,z})$. The total desired torque acting on the vehicle is obtained by rearranging the angular dynamics (3), which yields

$$\vec{m}_{des} = J(\vec{\omega}_{des} - \vec{\omega}_{meas})/\vec{\tau}_\omega + \vec{\omega}_{meas} \times J\vec{\omega}_{meas}, \tag{32}$$

where the division is executed element-wise. By inspection of (5) and (11), we find that the total torque around the body y-axis is given by

$$m_{tot,y} = x_{prop}(f_{prop,r} - f_{prop,l}) + d_y(\alpha_{wing,l})v^2_{wing,ref,l} - d_y(\alpha_{wing,r})v^2_{wing,ref,r}, \tag{33}$$

and does not depend on the flap angles $\delta_{flap,l}$ and $\delta_{flap,r}$. Note that both α_{wing} and $v_{wing,ref}$ depend on the corresponding propeller force f_{prop}. The two propeller forces $f_{prop,r}$ and $f_{prop,l}$ can thus be computed such that the two conditions

$$m_{tot,y} = m_{des,y}, \quad f_{prop,l} + f_{prop,r} = f_{thrust,des}, \tag{34}$$

are satisfied. However, due to actuator saturations, we might not be able to satisfy both conditions of (34). In this case, we adapt the desired thrust $f_{thrust,des}$ such that we can achieve the desired torque $m_{des,y}$; intuition and experiments indicate that it is more important to align the thrust axis, rather than keep the desired thrust value. The flap angles $\delta_{flap,l}$ and $\delta_{flap,r}$ are then computed such that the remaining two desired torques are achieved, i.e. such that

$$m_{tot,x} = m_{des,x}, \quad m_{tot,z} = m_{des,z}. \tag{35}$$

Again, if we cannot satisfy both conditions, we adapt $m_{des,z}$ such that we can achieve $m_{des,x}$ which is crucial for aligning the thrust axis and thus considered more important.

Note that, as we can see from (9), the aerodynamic force $\vec{f}_{air}$ depends on the actuator inputs, hence the actual aerodynamic force may differ from the nominal or estimated value we have used in (15) to compute the desired thrust vector $\vec{f}_{thrust,des}$. However, we assume that the difference is negligible.

4 Results

In this section, we present simulation and experimental results of the proposed tailsitter hover control strategy.

4.1 Simulation Results

During controller design we made some significant assumptions, in order to keep the problem tractable. We assume, for example, that the inner control loop perfectly tracks the desired body rates even though the actuators saturate quickly, which might have a significant effect on the behavior of the vehicle. Therefore, we verify the reliability performance of the proposed controller by simulating its behavior for a variety of initial conditions. In particular, we simulate the vehicle's dynamics for a set of 10'000 random initial conditions with the following probability distribution: The magnitude of the initial velocity vector $\vec{v}_{ini}$ is uniformly distributed between 0 m/s and 5 m/s, and its direction is uniformly distributed among all possible directions. Similarly, the magnitude of the initial body rates $\vec{\omega}_{ini}$ is uniformly distributed between 0 rad/s and 10 rad/s, and its direction is uniformly distributed among all directions. Further, the initial attitude $\vec{q}_{ini}$ is uniformly distributed over the whole attitude space, and the initial position $\vec{p}_{ini}$ is chosen to be the origin. For each of the 10'000 simulations, we draw a random sample for the initial state, and subsequently simulate the vehicle's dynamics forward in time. The task of the vehicle is to recover and to fly back to the origin. We consider the maneuver to be successful, if this task is achieved within reasonable thresholds. We found that for all 10'000 simulations the recovery is successful, indicating that for the chosen range of initial states, the proposed control strategy is able to cope with the neglected effects. However simulations also show that for larger initial velocities, due to the dominant aerodynamic forces and torques, the vehicle might enter a tumbling motion pattern and is not always able to stabilize. Thus, for these regions of the state space, the proposed hover controller is not suitable.

4.2 Experimental Results

The experiments are carried out in the ETH Zurich Flying Machine Arena [26]. The vehicle consists of a Robbe Mini Wing RC styrofoam airframe, a PX4 FMU flight computer [29], two MKS DS65K servos actuating the flaps, two Hacker A05-13S brushless electric motors driving the propellers, two ZTW Spider 12A ESCs with SimonK firmware [30], communication radios, and a LiPo battery. The testbed provides an infrared motion-capture system, which is used for estimating the tailsitter's current state. The outer control loop that computes desired body rates, as presented in Section 3, is implemented offboard and runs on a desktop workstation. The de-

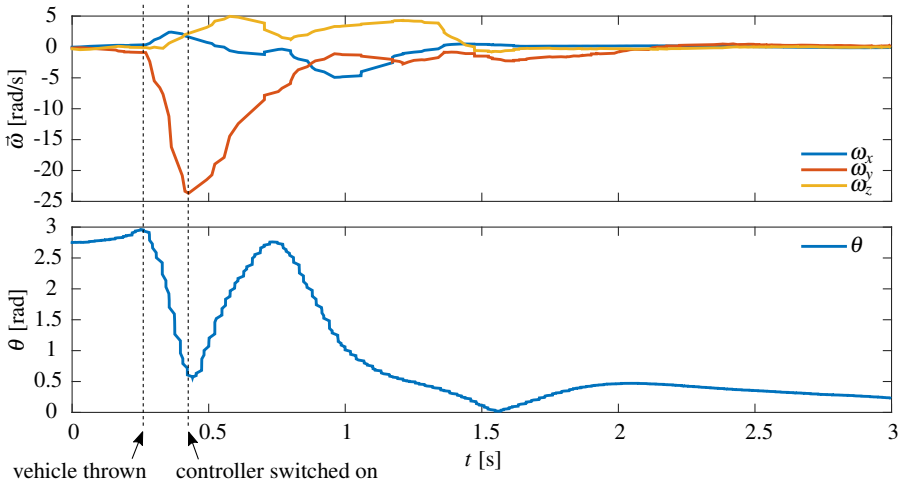

Fig. 3 Body rate and tilt angle trajectories of an example recovery maneuver. The vehicle is thrown into the air and the controller is switched on at the indicated points in time. Roughly 1 s after the controller has been switched on, the vehicle has recovered to near-hover flight.

sired body rates are sent over wireless radios to the vehicle with a rate of 50 Hz. Hence, the inner body rate controller is implemented onboard, and it runs at a rate of 1000 Hz.

In order to show the performance of the control strategy, we execute recovery maneuvers from arbitrary initial states back to hover flight. Therefore, the vehicle is manually thrown into the air, and at a predefined height of 2.5 m, the controller is switched on. Fig. 3 shows the body rate and tilt trajectories of such a recovery maneuver. Further, a video showing a series of throws and subsequent recoveries back to hover can be found on *www.youtube.com/watch?v=JModZfnVAv4*. The experiments indicate that for most of these manual throws the vehicle is able to recover; the only observed reason for failure is the limited space of the Flying Machine Arena, i.e. the vehicle bounces into the floor during the recovery maneuver.

Because of the relatively large drag coefficient perpendicular to the wings, a hovering tailsitter is susceptible to horizontal wind gusts. In order to analyze the performance of the proposed control strategy for such a situation, we simulate an external disturbance using a fan that creates an airstream with a velocity of about 2 m/s. The direction of the airstream is aligned with the body y-axis, i.e. perpendicular to the wings. Fig. 4 shows the response of the vehicle to such a simulated wind gust.

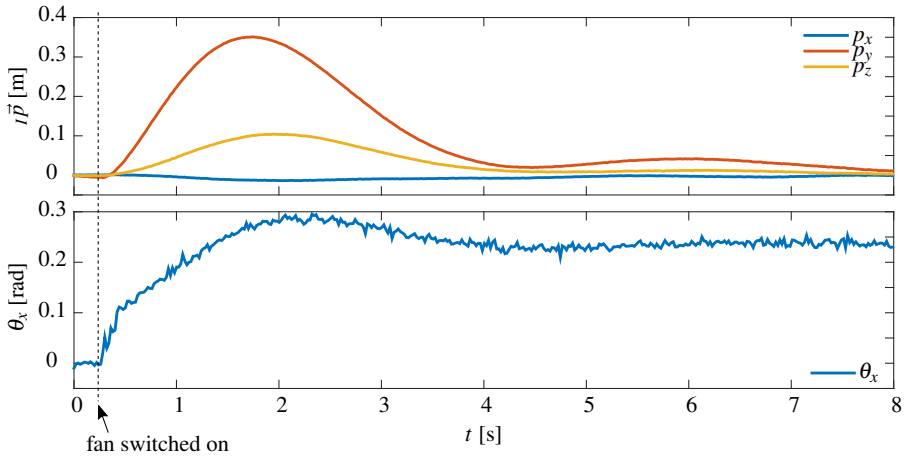

Fig. 4 Position and pitch angle trajectories when a wind gust hits the vehicle. Initially, the vehicle is hovering in resting air and the body and inertial frames are aligned. At the indicated point in time a fan is switched on, which results in an airstream of about 2 m/s along the y-axis. The plot on the top shows that due to the integrator term of the position controller, the vehicle is able to compensate the external disturbance without a steady-state position error. In the bottom plot, θ_x denotes the pitch angle around the body x-axis, and we can observe that the vehicle pitches in order to counteract the external disturbance.

5 Conclusion

An approach for a tailsitter hover controller has been presented, and its performance has been analyzed in simulation and verified in experiments. The experimental results indicate that the proposed tailsitter hover control strategy performs well under external aerodynamic disturbances, and is able to stabilize the vehicle from any initial attitude, given that the initial velocity is below a certain limit.

Future work includes the extension of the control method such that it can be applied to high-velocity maneuvers, for example by adding a high-level trajectory planner. Further, we would like to investigate if a proof of stability can be established.

References

1. A. Filippone, *Flight performance of fixed and rotary wing aircraft*. Elsevier, 2006.
2. W. H. Deckert and J. A. Franklin, "Powered-lift aircraft technology," NASA, Tech. Rep., 1989.
3. J. P. Campbell, "Research on VTOL and STOL aircraft in the United States," in *Advances in Aeronautical Sciences: Proceedings of the first International Congress in the Aeronautical Sciences, Madrid, 8-13 September, 1958*, vol. 2. Pergamon Press, 1959.
4. R. J. Woods, "Convertiplanes and other VTOL aircraft," SAE Technical Paper, Tech. Rep., 1957.

5. K. G. Wernicke, "Tilt prop-rotor composite research aircraft," DTIC Document, Tech. Rep., 1968.
6. R. L. Lichten, "Some aspects of convertible aircraft design," *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, vol. 16, no. 10, 2012.
7. J. Stuart, "Tilt wing propelloplane potentialities," *Journal of the American Helicopter Society*, vol. 4, no. 1, 1959.
8. L. P. Tosti, "Flight investigation of the stability and control characteristics of a 1/4-scale model of a tilt-wing vertical-take-off-and-landing aircraft," NASA, Tech. Rep., 1959.
9. P. Sinha, P. Esden-Tempski, C. A. Forrette, J. K. Gibboney, and G. M. Horn, "Versatile, modular, extensible VTOL aerial platform with autonomous flight mode transitions," in *IEEE Aerospace Conference*. IEEE, 2012.
10. C. Powers, D. Mellinger, and V. Kumar, "Quadrotor kinematics and dynamics," in *Handbook of Unmanned Aerial Vehicles*. Springer, 2014.
11. P. Sujit, S. Saripalli, and J. Borges Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles," *Control Systems, IEEE*, vol. 34, no. 1, 2014.
12. N. B. Knoebel and T. W. McLain, "Adaptive quaternion control of a miniature tailsitter UAV," in *American Control Conference (ACC)*. IEEE, 2008.
13. E. N. Johnson, A. Wu, J. C. Neidhoefer, S. K. Kannan, and M. A. Turbe, "Flight-test results of autonomous airplane transitions between steady-level and hovering flight," *Journal of guidance, control, and dynamics*, vol. 31, no. 2, 2008.
14. K. Kita, A. Konno, and M. Uchiyama, "Transition between level flight and hovering of a tail-sitter vertical takeoff and landing aerial robot," *Advanced Robotics*, vol. 24, no. 5-6, 2010.
15. T. Matsumoto, K. Kita, R. Suzuki, A. Oosedo, K. Go, Y. Hoshino, A. Konno, and M. Uchiyama, "A hovering control strategy for a tail-sitter VTOL UAV that increases stability against large disturbance," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010.
16. J. M. Beach, M. E. Argyle, T. W. McLain, R. W. Beard, and S. Morris, "Tailsitter attitude control using resolved tilt-twist," in *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014.
17. Y. Jung, S. Cho, and D. H. Shim, "A comprehensive flight control design and experiment of a tail-sitter UAV," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2013.
18. E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.
19. P. Anderson and H. Stone, "Predictive guidance and control for a tail-sitting unmanned aerial vehicle," in *Information, Decision and Control (IDC)*. IEEE, 2007.
20. R. H. Stone, "Aerodynamic modeling of the wing-propeller interaction for a tail-sitter unmanned air vehicle," *Journal of Aircraft*, vol. 45, no. 1, 2008.
21. G. E. Erickson, "High angle-of-attack aerodynamics," *Annual review of fluid mechanics*, vol. 27, no. 1, 1995.
22. N. B. Knoebel, S. R. Osborne, D. O. Snyder, T. W. McLain, R. W. Beard, and A. M. Eldredge, "Preliminary modeling, control, and trajectory design for miniature autonomous tailsitters," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2006.
23. J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, 2006.
24. W. Johnson, *Helicopter theory*. Courier Corporation, 2012.
25. P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics (Aiaa Education)*. American Institute of Aeronautics & Astronautics, 2003.
26. S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, 2014.
27. H. P. Geering, *Optimal Control with Engineering Applications*. Springer, 2007.
28. The MathWorks Inc., "Matlab, Version R2012a (7.14.0.739)," 2012.
29. "PX4 FMU," www.pixhawk.ethz.ch/px4/modules/px4fmu, accessed 2015-04-27.
30. "SimonK - Open Source Firmware for ATmega-based Brushless ESCs," https://github.com/sim-/tgy, accessed 2015-04-27.