2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Dynamic Feedback Linearization of Flying Wings With Real-time Newton-Raphson Iterations

Tom Lefebvre†, Jolan Wauters and Guillaume Crevecoeur

***Abstract*—This paper details development of a dynamic feedback linearization controller tailored to trajectory tracking of hybrid UAVs with a (tailsitter) flying wing topology. First, a differential flatness transform is presented using a simplified aerodynamic model with negligible lateral forces. The proposed controller derives from the flatness transform. For every state we can determine a collection of flat trajectories that correspond with that state. From that collection, we choose a flat trajectory that converges smoothly to the reference flat trajectory and apply the flat inverse dynamics to compute a suitable control input. To remedy the lack of an explicit inverse of the forward dynamics, we propose a dynamic inverse mapping approach which keeps the control algorithm computationally affordable. We evaluate and compare the control architecture with state-of-the-art cascade control in simulation.**

I. Introduction

Over the last decade tailsitters have become a prominent member of the family of hybrid Unmanned Aerial Vehicles (UAVs). Other members include quadplanes [1, 2, 3] and tiltwings. Hybrid UAVs support both hovering as well as forward flight capabilities. Such designs allow sufficient manoeuvrability when operating in confined spaces and support vertical take-off and landing. Second, they preserve the possibility for fast and energy efficient forward flight when their surrounding and application permit. Therewith hybrid UAVs aim to exceed the range and speed limitations faced by rotorcraft, that are strictly specialized to hovering [4].

Tailsitter or flying wings present a lean and lightweight design suitable to accommodate the requirements of a myriad of hybrid UAV applications. The concept distinguishes itself by its simplistic mechanical design. These vehicles consists of a single aerodynamic surface with two control surfaces to generate the required lift to support forward flight. Further they are equipped with an axisymmetric propulsion system that allows the system to hover when the vehicle is pitched vertically [5]. Therefore their operating regime is determined entirely by the attitude of the system [6, 7]. No mechanism is needed to change the direction of the propulsion system which allows to reduces weight and reduces susceptible malfunctions [3, 6]. Most tailsitter concepts described in the literature also lack a fuselage nor are they equipped with a tail or vertical aerodynamic surfaces.

Due to the lack of vertical aerodynamic surfaces, tailsitters require active directional stabilization [6, 7, 8]. In general, these unconventional UAV designs are difficult to control, especially due to the imbalance between relevant quantities that describe spatial configuration and limited control inputs.

†Corresponding author.
T. Lefebvre, J. Wauters and G. Crevecoeur are with the Dynamic Design Lab (D^2LAB) of the Department of Electromechanical, Systems and Metal Engineering, Ghent University, B-9052 Ghent, Belgium. E-mail: {tom.lefebvre,jolan.wauters,guillaume.crevecoeur}@ugent.be.
T. Lefebvre, J. Wauters and G. Crevecoeur are member of core-lab MIRO, Flanders Make, Belgium.

In recent years, the research community has proposed various control architectures. The main challenge is to realise both position control as well as attitude control with an underactuated system. There exists no direct relationship between the aerodynamic force acting on the system and the control inputs, impeding position control in particular. Further, as a result of the different operating modes, finding a control architecture that can face every mode, a so called global control design, is especially challenging [6, 9].

Several control architectures have been proposed based on the concept of Incremental Nonlinear Dynamic Inversion [6, 8, 10, 11]. INDI methods rely less on a model though they do rely on high-frequency acceleration measurements and are therefore susceptible to measurement inaccuracies. Successful stabilization can also be realised using MPC architectures which are, however, typically associated with a higher computational load [12, 13, 14]. Another line of work proposes cascaded controllers [7, 9]. Cascaded controllers are based on a similar principle to thrust vectoring, only here the thrust (and lift) direction is determined by the attitude of the vehicle. To that end the position controller modifies the reference attitude into a desired attitude that orients the thrust towards the reference position [7]. Clearly this approach renders the attitude control subordinate to that of the position.

Interestingly, the merit turns out to be in the proverbial fault. Due to the absence of vertical aerodynamic surfaces, most models do not consider a lateral force which can be exploited by the controller [6, 7, 9, 15]. Ritz et al. exploited this observation in a cascaded architecture [9]. Tal et al. pointed out that the tailsitter is therefore a differentially flat system [6]. Flatness is a resourceful property for both the analysis and controller synthesis of nonlinear systems [16, 17, 18]. The flatness property implies that there exists a differentially independent coordinate that fully parametrizes the differentially dependent state and input. Well known flat UAVs are quadcopters which already lead to a variety of dedicated flatness based controllers, e.g. [19, 20, 21].

Tal et al. combined the flatness property with INDI [6]. In this work we propose a novel dynamic feedback linearization strategy which poses lower demands on the onboard instrumentation. For every state we can calculate a set of flat trajectories that map to that state. From that set we choose that flat trajectory that converges smoothly to the reference trajectory and apply the flat inverse dynamics to compute a suitable control input. Our control strategy requires inversion of the forward differential map. Instead of relying on exact numerical inversion every time interval, we propose a dynamic inverse mapping approach which keeps the overall control algorithm computationally affordable.

We evaluate and compare the control architecture with state-of-the-art cascade control in simulation.

II. Modelling

We follow the approach by [6]. In the following $\mathrm{rot}_\alpha, \alpha \in \{x, y, z\}$ and e_α respectively represent the basic rotation matrix and unit vector about the α-axis. Further we denote the elements of a vector $w \in \mathbb{R}^3$ by subscripting, i.e. w_α.

A. Kinematics and dynamics

To represent the kinematics of flight we make use of a global (and inertial) and a local frame, see Fig. 1.

The spatial configuration of the UAV is represented by the position and attitude of the local frame w.r.t. the global frame. The position is represented with Cartesian coordinate vector $p \in \mathbb{R}^3$. The UAV's linear velocity, denoted $v \in \mathbb{R}^3$ in global coordinates, is governed by the differential equation, $v = \dot{p}$. To represent the attitude of the system we use the yaw-roll-pitch convention with yaw angle, ψ, roll angle, ϕ, and, pitch angle, θ. The angles are gathered in an angle vector $q \in \mathbb{R}^3$. The relation between the angular velocity in local coordinates, $\omega''' \in \mathbb{R}^3$, and the time derivative of the angle vector, $\dot{q}$, is governed by the equation

$$\omega''' = \mathrm{rot}_y^\top(\theta)\mathrm{rot}_x^\top(\phi)e_z\dot{\psi} + \mathrm{rot}_y^\top(\theta)e_x\dot{\phi} + e_y\dot{\theta} \tag{1}$$

Throughout we will use the convention of accentuating vectorial physical quantities such as velocities and forces depending on the frame we have expressed them in. Starting with a vector $w \in \mathbb{R}^3$ in the global frame we define $w' = \mathrm{rot}_z^\top(\psi)w$, $w'' = \mathrm{rot}_x^\top(\phi)w'$ and $w''' = \mathrm{rot}_y^\top(\theta)w''$, i.e. w''' is expressed in the local frame.

The flight dynamics are governed by the Newton-Euler equations of motion. The dynamic state of the UAV system is gathered in the vector $\xi = (p, q, v, \omega''') \in \mathbb{R}^{12}$.

$$\begin{aligned} m\dot{v} + mg &= f \\ \mathrm{I}\dot{\omega}''' + \omega''' \times \mathrm{I}\omega''' &= \tau''' \end{aligned} \tag{2}$$

Here $f \in \mathbb{R}^3$ represents the force exerted on the aircraft expressed in the global frame, whereas $\tau''' \in \mathbb{R}^3$ represents the torque exerted on the craft expressed in the local frame. The matrix $\mathrm{I} = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$ denotes the inertia tensor and m the mass of the system.

The system is equipped with two propellers, generating forces T_1 and T_2, and, two control surfaces, δ_1 and δ_2. These 4 control inputs are gathered in the input vector, $\upsilon \in \mathbb{R}^4$. The forces, f, and, τ''', are parametrized using ϕ-theory [4]. We adopt the aerodynamic model from [22] which is a simplified version of [6] yet is sufficient for this preliminary study.

$$\begin{aligned} f''' &= \begin{pmatrix} T_1 + T_2 - k_D v_x'''\|v\| \\ 0 \\ -k_L v_z'''\|v\| \end{pmatrix} \\ \tau''' &= \begin{pmatrix} k_1 v_x'''\|v\|(\delta_2 - \delta_1) \\ k_2 v_x'''\|v\|(\delta_1 + \delta_2) \\ k_3 v_x'''\|v\|(\delta_2 - \delta_1) + T_2 l - T_1 l \end{pmatrix} \end{aligned} \tag{3}$$

where k_D and k_L are aerodynamic lift and drag constants that must be evaluated empirically, k_1, k_2 and k_2 are aerodynamic propulsion constants and l denotes the symmetric position of the propellers. We refer to sec. IV-A for values.

Remark that we consider a simplified form for the aerodynamics. This assumption has the specific advantage that the corresponding dynamics are now *differentially flat*. The assumption is justified by the lack of vertical aerodynamic surfaces as was motivated in the introduction.

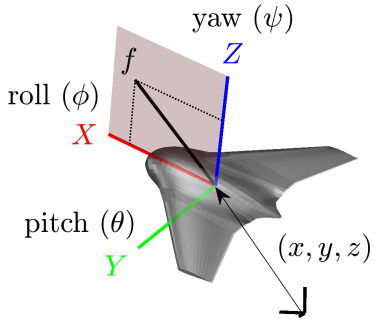

Fig. 1: Inertial (black) and body-fixed (coloured) frames, geometry of aerodynamic and propulsion force, f, and angular motion terminology.

***Assumption** 1: The second component of the aerodynamic force, f''', expressed in the local frame is zero.*

In conclusion the Newton-Euler equations can be organised in a conventional state-space representation which yields the following ordinary differential equation

$$\dot{\xi} = g(\xi, \upsilon) = g_\xi(\xi) + g_\upsilon(\xi)\upsilon \tag{4}$$

B. Differential flatness

Differential flatness is a structural property of a class of nonlinear dynamical systems. If a system is *flat*, it is implied that all differentially dependent system variables – states and inputs – can be written in terms of a specific set of differentially independent variables, and their time derivatives, the so called flat output [17].

***Definition** 1 (Differential flatness [16]): The system, $\dot{\xi} = g(\xi, \upsilon)$, with state $\xi \in \Xi \subset \mathbb{R}^{n_\xi}$ and input $\upsilon \in \Upsilon \subset \mathbb{R}^{n_\upsilon}$, is differentially flat if there exists a variable $\sigma \in \Sigma \subset \mathbb{R}^{n_u}$, whose components are differentially independent, and operators Ξ and Υ such that*

$$\begin{aligned} \xi &= \Xi(\sigma, \dot{\sigma}, \ddot{\sigma}, \dots) \\ \upsilon &= \Upsilon(\sigma, \dot{\sigma}, \ddot{\sigma}, \dots) \end{aligned} \tag{5}$$

Intuitively, the flat output, $\sigma(t)$, can be interpreted as a minimal dynamical representation of any feasible state-input trajectory, $\{\xi(t), \upsilon(t)\} \in \mathcal{G}$ of the system. The feasible state-input function space, $\mathcal{G}$, defined as $\mathcal{G} = \{\{\xi(t), \upsilon(t)\} | \xi \in \Xi, \upsilon \in \Upsilon : \dot{\xi} = g(\xi, \upsilon)\}$, is a subspace from the function space $\Xi \times \Upsilon$ so that any function element satisfies the dynamic constraint, $\dot{\xi} = g(\xi, \upsilon)$. Likewise, $\{\Xi, \Upsilon\}[\cdot] : \Sigma \mapsto \mathcal{G}$.

As shown by Tal et al. [6], following assumption 1 the tailsitter is differentially flat with flat coordinate. The flatness of the tailsitter is a direct consequence of assumption 1.

$$\sigma = (p, \psi) = (x, y, z, \psi) \tag{6}$$

In the remainder of this section, the differential flatness of the tailsitter is demonstrated. Calculation of v and $\dot{v}$ as a function of the derivatives of p is trivial. From the expression for $\dot{v}$ we can calculate the force vector f expressed in global coordinates. By choice, we have access to the yaw angle, ψ. As such we can compute f'. Following the geometry of the problem and the assumed lack of a second force component of f''', f' must then lie in the roll plane and therefore

$$\phi = -\arctan\frac{f'_y}{f'_z} \tag{7}$$

Once we know ϕ we can calculate f''. Likewise we can calculate v''. Then we can solve for θ and T from expression (3). Here T denotes the total thrust force, $T_1 + T_2$.

$$\begin{aligned} \theta &= \arctan \tfrac{-f_z'' - k_L v_z'' \|v\|}{f_x'' + k_L v_x'' \|v\|} \\ T &= (-f_z'' - k_D v_z'' \|v\|) s_\theta + (f_x'' + k_D v_x'' \|v\|) c_\theta \end{aligned} \tag{8}$$

Now that we have access to all the angles, we invoke the kinematics of the problem to determine ω'''. From ω''' and (2), we can calculate τ''' which is related to the deflection of the control surfaces and allows to determine the distribution of the total thrust force over the two propellers.

$$\begin{aligned} T_1 &= \tfrac{1}{2}T + \tfrac{1}{2l}\left(\tfrac{k_3}{k_1}\tau_x''' - \tau_z'''\right) \\ T_2 &= \tfrac{1}{2}T - \tfrac{1}{2l}\left(\tfrac{k_3}{k_1}\tau_x''' - \tau_z'''\right) \\ \delta_1 &= \tfrac{1}{2v_x''' \|v\|}\left(\tfrac{\tau_y'''}{k_2} - \tfrac{\tau_x'''}{k_1}\right) \\ \delta_2 &= \tfrac{1}{2v_x''' \|v\|}\left(\tfrac{\tau_x'''}{k_1} + \tfrac{\tau_y'''}{k_2}\right) \end{aligned} \tag{9}$$

It follows that we have found expressions for ξ and υ as a function of the derivatives of σ.

$$\begin{aligned} \xi &= \Xi(\sigma, \dot\sigma, \ddot\sigma, \dddot\sigma) = \Xi[\sigma] \\ \upsilon &= \Upsilon(\sigma, \dot\sigma, \ddot\sigma, \dddot\sigma, \ddddot\sigma) = \Upsilon[\sigma] \end{aligned} \tag{10}$$

Provided some reference signal, $\sigma_d(t)$, we can thus efficiently evaluate the corresponding reference state and input signals $\xi_d(t) = \Xi[\sigma_d(t)]$ and $\upsilon_d(t) = \Upsilon[\sigma_d(t)]$.

III. Control design

In this section we propose a novel trajectory tracking control for the tailsitter system. First, we recapitulate some theoretical and technical results reflecting on the design of feedback controllers for flat systems. These results guarantee the existence of dynamic feedback linearization strategy, however do not specify how to derive it. Second, we present an intuitive argument to design two controllers that are consistent with the theoretical guarantees.

As part of the described solution approach, it will be required to invert the mapping, $\Xi : \mathcal{Z} \mapsto \mathcal{Z}$, preferably analytically. To alleviate the task of finding the exact inverse, we propose a novel dynamic model inversion approach that accomplishes the required function inverse in real-time.

A. Technical background

First consider the following definition of dynamic feedback linearisable systems.

***Definition** 2 (Dynamic feedback linearization [23]): System (4) is dynamic feedback linearizable, if there exist auxiliary states, $\gamma \in \mathcal{R}^{n_\gamma}$; a dynamic feedback, with $\nu \in \mathbb{R}^{n_\upsilon}$*

$$\begin{aligned} \dot\gamma &= g_\gamma(\xi, \gamma) + g_\nu(\xi, \gamma)\nu \\ \upsilon &= \alpha(\xi, \gamma) + \beta(\xi, \gamma)\nu \end{aligned}$$

and an extended state transformation, $\chi = \eta(\xi, \gamma)$ such that $\chi \in \mathbb{R}^{n_\xi + n_\gamma}$ and the extended system

$$\begin{aligned} \dot\xi &= g_\xi(\xi) + g_\upsilon(\xi)\alpha(\xi, \gamma) + g_\xi(\xi)\beta(\xi, \gamma)\nu \\ \dot\gamma &= g_\gamma(\xi, \gamma) + g_\nu(\xi, \gamma)\nu \end{aligned}$$

satisfies, in the new extended state, χ

$$\dot\chi = \mathrm{A}\chi + \mathrm{B}\nu$$

where the linear system (A, B), is in Brunovský form.

When a system is (dynamic) feedback linearisable, it is straightforward to design a tracking controller, $\upsilon_d(t)$, that is asymptotically stable towards the reference, $\chi_d(t)$, in the extended state space, through construction of a tracking controller for the system (A, B), and nonlinear transformation, η. The principle is demonstrated with linear feedback. The gain matrix K can be determined e.g. through LQR design.

$$\begin{aligned} \upsilon(t, \xi(t), \gamma(t)) &= \alpha(\xi(t), \gamma(t)) + \beta(\xi(t), \gamma(t))\nu(t, \xi(t), \gamma(t) \\ \nu(t, \xi(t), \gamma(t) &= \nu_d(t) - \mathrm{K}\Delta\chi(t) \\ \Delta\chi(t) &= \chi(t) - \chi_d(t) \\ &= \eta(\xi(t), \gamma(t)) - \chi_d(t) \end{aligned}$$

Second, as proven by [24], it holds that any flat system is dynamic feedback linearisable. The proof does not detail how to determine the auxiliary, γ.

***Theorem** 1: System (4) is dynamic feedback linearizable if and only if it is differentially flat.*

Finally, the following two Lemmas are useful to analyse the nominal stability of the proposed control strategy.

***Lemma** 1 (Lemma 4.5 from [25]): Assume $g(\xi, \upsilon)$ is continuously differentiable and globally Lipschitz. If $\dot\xi = g(\xi, 0)$ has a global asymptotically stable equilibrium point at the origin, then the system $\dot\xi = g(\xi, \upsilon)$ is input-to-state stable.*

***Lemma** 2 (Lemma 4.6 from [25]): If the systems $\dot\xi_1 = g_1(\xi_1, \xi_2)$ and $\dot\xi_2 = f_2(\xi_2, \upsilon)$ are input-to-state stable, then their cascade is input-to-state stable. Consequently, if $\dot\xi_1 = g_1(\xi_1, \xi_2)$ is input-to-state stable and the origin of $\dot\xi_2 = f_2(\xi_2, \upsilon)$ is globally asymptotically stable, then the origin of the cascade is globally asymptotically stable.*

B. Dynamic feedback linearization

Flatness implies that any smooth feasible state-action trajectory, that is $\{\xi(t), \upsilon(t)\} \in \mathcal{G}$, is equivalent to or parametrized entirely by the flat trajectory, $\sigma(t) \in \Sigma$. This suggests that for any given state, $\xi(t)$, there exists a set of flat trajectories that are consistent with that state. The remaining degrees of freedom are determined by $\upsilon(t) \in \Upsilon$. Our proposed tracking control strategy makes use of this observation by choosing that trajectory from the set that converges smoothly to the desired flat trajectory, $\sigma_d(t)$. To calculate the required input, $\upsilon(t)$, we can again rely on (10).

The set of flat trajectories that the system may occupy for given $\xi(t)$ can be determined by considering the expression $\xi = \Xi(\sigma, \dot\sigma, \ddot\sigma, \dddot\sigma)$. For given $\xi(t)$, we can find $\sigma(t)$ and higher order time derivatives, parametrizing all flat trajectories that visit that state. Put differently, we have to invert the map, $\Xi(\cdot)$. Note that there are generally more arguments in the map than there are variables in the state ξ. Therefore we can select a subset of variables, ζ, so to establish a unique inverse function. We will refer to this subset as the *dependent variables*. The remaining, parametrising variables are denoted with γ and will be referred to as the *independent variables*. More formally we have $\zeta \subset (\sigma, \dot\sigma, \ddot\sigma, \dddot\sigma)$, $\gamma = (\sigma, \dot\sigma, \ddot\sigma, \dddot\sigma) \backslash \zeta$ and $\xi = \Xi(\zeta, \gamma)$ where ζ and γ are so that $\zeta = \Xi^{-1}(\xi, \gamma)$. Remark that the elements of ζ are a design choice.

For the present system, it holds $\xi = \Xi(p, \dot p, \ddot p, \dddot p, \psi, \dot\psi)$. Then we argue as follows. The variables p, $\dot p$, ψ and $\dot\psi$ are elements of, or are directly related to, the state. They are therefore identified as dependent variables. At this point we have several options. Most can be disregarded due to symmetry considerations so that only two possibilities remain.

a) **Ψ-*inverse*:** We can append the dependent variables with $\ddot{p}$ and $\dddot{z}$, rendering $\ddot{x}$ and $\ddot{y}$ independent. For reasons elaborated later on, we refer to this choice as the Ψ-inverse.

$$(\underbrace{p, \dot{p}, \psi, \dot{\psi}, \ddot{p}, \dddot{z}}_{\zeta}) = \Xi_{\Psi}^{-1}(\underbrace{p, \dot{p}, q, \dot{q}}_{\xi}, \underbrace{\ddot{x}, \ddot{y}}_{\gamma}) \in \mathbb{R}^{12} \tag{11}$$

b) **Θ-*inverse*:** We can also append the dependent variables with $\ddot{x}$, $\ddot{y}$, $\dddot{x}$ and $\dddot{y}$, rendering $\ddot{z}$ and $\dddot{z}$ independent variables. We refer to this choice as the Θ-inverse.

$$(\underbrace{p, \dot{p}, \psi, \dot{\psi}, \ddot{x}, \ddot{y}, \dddot{x}, \dddot{y}}_{\zeta}) = \Xi_{\Theta}^{-1}(\underbrace{p, \dot{p}, q, \dot{q}}_{\xi}, \underbrace{\ddot{z}, \dddot{z}}_{\gamma}) \in \mathbb{R}^{12} \tag{12}$$

The expression above implies that for any given state, $\xi(t)$, and whatever but given independent variables, $\gamma(t)$, the corresponding state trajectory is parametrized by any flat trajectory, that is consistent with $\zeta(t) = \Xi^{-1}(\xi(t), \gamma(t))$. It follows that we can design an artificial controller, ν, that steers the flat trajectory – consistent with, $\xi(t)$ – to the desired flat trajectory, $\sigma_d(t)$. Then, taking into account the differential dependencies, these observations suggests that we can choose 4, differentially independent, tracking controllers that steer each flat coordinate to the desired trajectory $\sigma_d(t)$[1]

$$\begin{aligned} \ddddot{\alpha} = \nu_\alpha = \ddddot{\alpha}_d - \sum_{i=0}^{4} \mathrm{K}_\alpha^i \Delta\alpha^{(i)} \\ \ddot{\psi} = \nu^\psi = \ddot{\psi}_d - \sum_{i=0}^{2} \mathrm{K}_\psi^i \Delta\psi^{(i)} \end{aligned} \tag{13}$$

with $\alpha \in \{x, y, z\}$ and where

$$\Delta\alpha^{(i)} = \alpha^{(i)} - \alpha_d^{(i)} \tag{14}$$

To calculate the required input, υ, we can then again rely on the flat expressions

$$\upsilon = \Upsilon(\dot{p}, \ddot{p}, \dddot{p}, \ddddot{p}, \psi, \dot{\psi}, \ddot{\psi}) = \Upsilon(\zeta, \gamma, \nu) \tag{15}$$

The present control architecture can be recognized as a dynamic feedback linearisation approach with

$$\chi = (p, \dot{p}, \ddot{p}, \dddot{p}, \psi, \dot{\psi}) \tag{16}$$

Remark that we cannot and need not measure the independent variables, γ. These can be evaluated by integrating the appropriate elements from ν. As opposed to INDI, the proposed architecture is thus a state feedback controller.

C. Real-time function inversion

Even for the simplified aerodynamic model used in this paper, calculation of the inverse map, Ξ_{Ψ}^{-1} and Ξ_{Θ}^{-1}, is nontrivial. Stripping the problem from the trivial mappings from ξ to p, $\dot{p}$, ψ and $\dot{\psi}$, we can focus on finding the mapping from ϕ, $\dot{\phi}$, θ and $\dot{\theta}$ to $\ddot{p}$ and $\dddot{z}$ either $\ddot{x}$, $\ddot{y}$, $\dddot{x}$ and $\dddot{y}$.

To that end, we may reconsider equations (7) and (8)

$$\begin{aligned} t_\phi &= \tfrac{s_\psi \ddot{x} - c_\psi \ddot{y}}{\ddot{z} + g} \\ t_\theta &= \tfrac{-\sqrt{1+t_\phi^2}(\ddot{z}+g) - \frac{k_L}{m}(s_\phi(s_\psi \dot{x} - c_\psi \dot{y}) + c_\phi \dot{z})\|v\|}{c_\psi \ddot{x} + s_\psi \ddot{y} + \frac{k_L}{m}(c_\psi \dot{x} + s_\psi \dot{y})\|v\|} \end{aligned} \tag{17}$$

Derivation to time, yields 2 more equations. The resulting system of 4 can then be solved for the dependent variables, γ.

[1]Note that it is possible to include the integrated error in the linear feedback control design of ν. This can be achieved by introducing yet another auxiliary variable, ι, so that $\dot{\iota} = \Delta\sigma$. Then the sum in the expression for ν runs from $i = -1$. The LQR design can be updated accordingly.

Unfortunately, these calculations turn out to be hard and other approaches need be devised. One possibility is to rely on numerical inversion techniques however this puts heavy demands on the on-board computational platform. Thus we propose an other strategy altogether.

For the Ψ-inverse it is possible to solve the first 3 equations for $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$ as a function of ξ, γ and $\dddot{z}$. Substitution of these expressions into the fourth equation yields an implicit constraint for $\dddot{z}$ parametrized by ξ and γ. This equation cannot be solved for $\dddot{z}$ explicitly. A similar problem arose for the Θ-inverse. Due to symmetry considerations here we can solve the first 2 equations for $\ddot{x}$ and $\ddot{y}$ and substitute them into the third and fourth equation which then could not be solved explicitly for $\dddot{x}$ and $\dddot{y}$.

Generally, we solve explicitly for a subset, ζ' of ζ as a function of the state, ξ, the independent variables, γ, and the remaining dependent variables, ζ''.

$$\zeta' = k'(\xi, \gamma, \zeta'') \tag{18}$$

The remaining subset, ζ'', is determined implicitly.

$$0 = k''(\xi, \gamma, \zeta'') \tag{19}$$

Then to determine ζ'', we introduce an additional variable λ as a proxy and propose the following exponential decay equation to regulate the constraint when it deviates from zero.

$$\dot{k}'' = -\alpha k'' \tag{20}$$

Substituting λ for ζ'', expanding the left-hand side of the equation as a function of $\dot{\lambda}$, $\dot{\xi}$ and $\dot{\gamma}$ and solving for $\dot{\lambda}$ yields the following control law

$$\dot{\lambda} = -(\partial_\lambda k'')^{-1} \left(\alpha k'' + \partial_\xi k'' \dot{\xi} + \partial_\gamma k'' \dot{\gamma} \right) \tag{21}$$

The 1th term of this control law reads as a real-time Newton-Raphson iteration. The 2nd and 3rd term compensate for the simultaneous change in the state variables ξ and independent variables γ.

D. Complete control architecture

The ingredients from the previous section can be gathered into the following control strategy. As a result of the dynamic map inversion strategy, the auxiliary dynamic state variable, γ, is extended with the dynamic variable, λ, resulting into a dynamic feedback strategy with $n_\gamma + n_\lambda$ virtual variables. The gain matrix, Γ, select the first two entries from ν.

$$\begin{aligned} \dot{\gamma} &= \Gamma\nu \\ \dot{\lambda} &= -(\partial_\lambda k'')^{-1} \left(\alpha k'' + \partial_\xi k'' \dot{\xi} + \partial_\gamma k'' \dot{\gamma} \right) \\ \dot{\xi} &= g_\xi(\xi) + g_\upsilon(\xi)\upsilon \\ \zeta' &= k'(\xi, \gamma, \lambda) \\ \upsilon &= \Upsilon(\zeta', \gamma, \lambda, \nu) \\ \nu &= \nu_d - \sum\nolimits_i \mathrm{K}_i \Delta\sigma^{(i)} \end{aligned} \tag{22}$$

E. Stability analysis of dynamic inverse mapping

We discuss here briefly the nominal stability of the proposed control strategy in an attempt to quantify its region of attraction. To that end, note that we can analyse the system in its linear coordinate space.

$$\dot{\chi} = \mathrm{A}\chi + \mathrm{B}\beta^{-1}(\xi, \gamma)(\upsilon - \alpha(\xi, \gamma)) \tag{23}$$

Now, instead of substituting the exact dynamic feedback linearization, that is $\upsilon = \alpha(\chi) + \beta(\chi)\mathrm{K}\chi$, the following approximate policy needs to be considered, which depends on the auxiliary variable, λ

$$\upsilon = \hat{\alpha}(\xi, \gamma, \lambda) + \hat{\beta}(\xi, \gamma, \lambda)\mathrm{K}\hat{\eta}(\xi, \gamma, \lambda) \tag{24}$$

Combination of these two differential equations yields the following perturbed partial feedback linearized system.

$$\begin{aligned} \dot{\chi} &= (\mathrm{A} - \mathrm{BK})\chi + \mathrm{B}\delta(\chi, \lambda) \\ \dot{\lambda} &= g_\lambda(\lambda, \chi) \end{aligned} \tag{25}$$

The coupled system is a cascade connection of $\dot{\lambda} = g_\lambda(\lambda, \chi)$, with χ as input, and $\dot{\chi} = (\mathrm{A} - \mathrm{BK})\chi + \mathrm{B}\delta(\chi, \lambda)$, with $\delta(\chi, \lambda)$ as input. It follows from Lemma 2 if the system, $\dot{\lambda} = g_\lambda(\lambda, \chi)$, is input-to-state stable, so is the system, $\dot{\chi} = (\mathrm{A} - \mathrm{BK})\chi + \mathrm{B}\delta(\chi, \lambda)$, with δ as input.

So our stability analysis reduces to analysing the input-to-state stability of the system, $\dot{\lambda} = g_\lambda(\lambda, \chi)$. Consider therefore the following Lyapunov function

$$V(\lambda) = \tfrac{1}{2}k''(0, 0, \lambda)^2 \tag{26}$$

so that

$$\dot{V} = k'' \cdot \partial_\lambda k'' \cdot \dot{\lambda} = -\alpha(k'')^2 \tag{27}$$

It follows that if $(k'')^2$ is a class $\mathcal{KL}$ function and furthermore k'' is globally Lipschitz in (ξ, γ, λ) the system is input-to-state stable according to Lemma 1. Unfortunately the function k'' does only exhibit the required properties locally, so that the Lemma only applies locally. This implies that the dynamic map inversion is not a globally stabilizing method.

The actual region of attraction is highly application dependent and is beyond the scope of the present study.

F. Singularities

Flatness based controllers tend to suffer from singularities [26]. We list here two known singularities.

Both the Ψ- and Θ-inverses exhibit a singular manifold where the inverse is not defined. The technical reason is that, the independent variables can no longer be independent on those manifolds, this on account of the geometry of the problem, implying a destruction of the set. The Ψ-inverse is singular for $\phi = 0$, $\dot{\psi} = 0$ and $\dot{\phi} = 0$. Hence, if the forward flight of tailsitter is restricted to the yaw-roll plane. The Θ-inverse is singular on the manifold $\theta = 0$. That is when the tailsitter is flying perfectly level.

The singularities cannot be avoided since they are determined by the dependent variables. This means that we can only resolve the issue by actively steering the dependent variables away from the singularities.

IV. Simulations

A. Experiment

Simulation results are presented for a conceptual tailsitter design closely resembling the commercially available UX5 with a wingspan of $1\,\mathrm{m}$. The geometry is visualized in the scheme in Fig. 1. Aerodynamic coefficients were evaluated using MIT's Athena Vortice Lattice method. The coefficients are assumed fixed and determined at cruise conditions with $\alpha = 6^\circ$ and $v_x = 22\,\mathrm{m/s}$ (i.e. about roughly $80\,\mathrm{km/h}$). Following this approach the aerodynamic coefficients are given in Table I. The mechanical parameters are given in Table II. All model parameters are adopted from [22].

TABLE I: Aerodynamic coefficients

k_L [kg/m]	k_D [kg/m]	k_1 [kg]	k_2 [kg]	k_3 [kg]
$6.14\text{E}{-1}$	$2.66\text{E}{-4}$	$4.29\text{E}{-2}$	$1.46\text{E}{-2}$	$2.3\text{E}{-2}$

TABLE II: Mechanical parameters

m [kg]	I_{xx} [kg m^2]	I_{yy} [kg m^2]	I_{zz} [kg m^2]	l [m]
2.5	$8.35\text{E}{-3}$	$1.18\text{E}{-2}$	$1.99\text{E}{-2}$	$2\text{E}{-1}$

TABLE III: Control allocation for different reference paths.

	acrobatic turn	ascending turn	landing	take-off
CSFL	✓	✓	✓	✓
Ψ-DFL+NIM	✓	✓		
Ψ-DFL+RIM	✓	✓		
Θ-DFL+NIM		✓	✓	✓
Θ-DFL+RIM		✓	✓	✓

a) Reference paths: We compare 4 reference paths. The reference paths are visualized in Fig. 2. We consider an acrobatic left turn, a continuously ascending left turn, a landing manoeuvre and a take-off manoeuvre. The acrobatic turn crosses the singular manifold of the Θ-inverse (zero-crossing of θ halfway). The landing and a take-off manoeuvres are embedded in the singular manifold of the Ψ-inverse. Details on the path parametrization can be found in appendix A.

b) Control architectures: We compare our methods with a state-of-the-art cascaded static feedback linearization controller [9]. We refer to appendix B for details. We compare with the Ψ- and Θ-variant of the proposed dynamic feedback linearization approach. To assess the effect of the dynamic inverse mapping, we use numerical inversion to evaluate the entire dependent variable, ζ, explicitly. Henceforth we denote the cascaded controller from Ritz as CSFL. The Ψ- and Θ-variants are denotes Ψ- and Θ-DFL. When the inverse is computed numerically +NIM is added, real-time inversion is denoted +RIM. The control allocation for each path is illustrated in Table III. The CSFL is tuned to obtain two critically damped 2nd order systems. We use time constants $T_p = 1\mathrm{s}$ and $T_q = 1\text{E}{-1}\mathrm{s}$, for the position controller and attitude controller respectively. The DFLs are tuned using LQR design with scalar weighting matrices. The linearized state penalties are $1\text{E}3$ and $1\text{E}6$ for, x and y, and, z, respectively. The linearized input penalty was set to 1 irrespectively. The dynamics of the ψ subsystem are tuned to obtain a critically damped 2nd order system with time constant $T_\psi = 1\text{E}{-2}\mathrm{s}$. The DFL architectures were tuned empirically to obtain roughly the same mean squared error on the position path as the CSFL.

c) Uncertainties: We consider 2 types of uncertainty. An initial state bias, $\xi(0) \sim \mathcal{N}(\xi_d(0), \Sigma_{\xi\xi})$, and, exogenous input perturbation, $\mathrm{d}\upsilon \sim \mathcal{N}(0, \Sigma_{\upsilon\upsilon})$. The magnitude of the perturbations was tuned to obtain a SNR of roughly $10\mathrm{dB}$.

$$\begin{aligned} \Sigma_{\xi\xi} &= \mathrm{diag}(\mathrm{I}_3 \cdot 1\text{E}2, \mathrm{I}_3 \cdot 1\text{E}{-4}, 0, 0, 0, 0, 0, 0) \\ \Sigma_{\upsilon\upsilon} &= \mathrm{diag}(\mathrm{I}_2 \cdot 1\text{E}{-2}, \mathrm{I}_2 \cdot 1\text{E}{-8}) \end{aligned} \tag{28}$$

B. Results

We performed 50 experiments for each reference path and each allocated control architecture. The mean squared errors (MSE) for different state subspaces, particularly p, v, q and ω, are visualised in Fig. 3. A representative example of the tracking performance is given in Fig. 4.

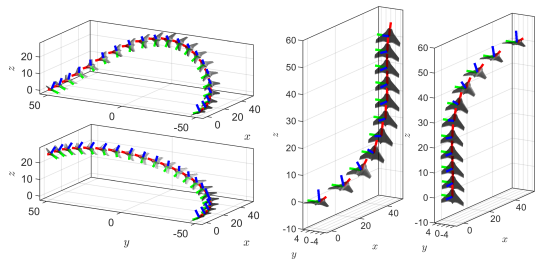

Fig. 2: Illustration of different reference paths. Top: acrobatic turn (1), bottom: ascending turn (2), right: landing (3), and, far right: take-off (4).

The CSFL control is slightly more performant than the DFL approaches with respect to the MSE. The difference however is insignificant compared to the overall magnitudes of the signals. Furthermore, the position MSE is dominated by the large initial state bias which the CSFL architecture corrects more aggressively. This however has an adverse affect on the other state variables. Indeed each of the DFL architectures outperforms the CSFL architecture with respect to the 3 other MSE values. The most significant improvement can be observed for the angular velocities. This is a direct result of the principle behind the CSFL approach where the attitude control is subordinate to the position control and rapid alignment of the UAVs attitude with the reference attitude is required to obtain the desired thrust vectoring behaviour. The cascaded approach explicitly decomposes these task whereas they remain integrated in the presented approach. This mechanism is nicely illustrated in the individual example depicted in Fig. 4. This effect is absent with the DFL approaches since here no direct attitude control is pursued but rather the system is tracked exponentially in the linearized state-space. Finally we remark that the used of the dynamic inversion technique has little negative effect and causes only a slight increase in MSE.

V. Conclusion

The authors acknowledge that the proposed control architectures need to face the test of physical experiment. The results presented are indicative of its effectiveness as far as theoretical guarantees and simulation can be indicative for practical successes. Nevertheless, we argue dynamic feedback linearization can be an efficient design approach to address the control challenges associated to stabilizing tailsitters.

We conclude with some remarks that may be of value to future practitioners. In this study, a relatively easy aerodynamic model has been adopted, however, we note that there are no restrictions towards the model complexity. As long as the lateral force is negligible, the model is flat and our approach remains applicable. When the complexity of the model increases it is even unlikelier that the exact inverse maps exist. This observation accentuates the purposefulness of the proposed dynamic inverse map approach. Furthermore, we have found tuning of the dynamic feedback linearization strategies significantly more convenient. Finally we note that any feedback linearization strategy (including ours) is susceptible to model inaccuracies. Fortunately the literature is larded with adaptive and learning methods, so that model parameters can be estimated in real-time from linear and angular acceleration measurements. Alternatively steady-state errors can be compensated by adopting PID schemes which are straightforward to adopt in the DFL architecture.

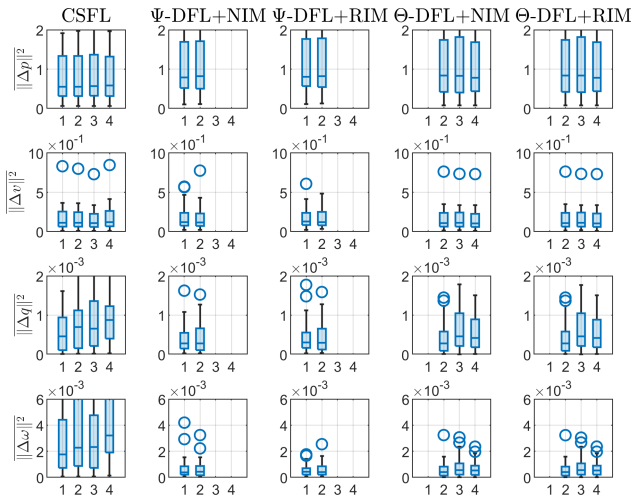

Fig. 3: MSE for varying controllers and different reference paths. The reference path is denoted by the horizontal axis. Different control architectures are compared horizontally.

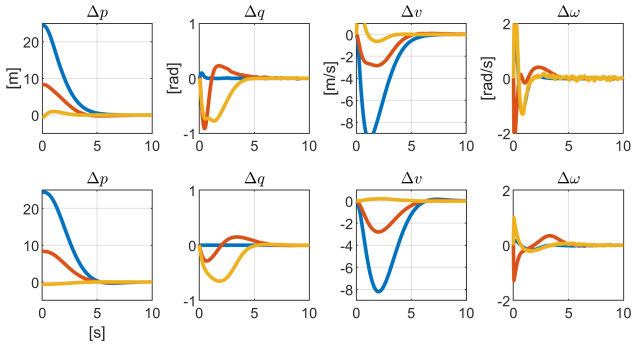

Fig. 4: Representative example of the reference tracking performance for the take-off manoeuvre. Top: CSFL, bottom: Θ-DFL-RIM.

Acknowledgements

This work is supported by the ETF project BORNE.

Appendix A
Reference Paths

The following reference paths were used with $R = 50\text{m}$ and $T = 10\text{s}$.

Acrobatic turn We set $x_d(t) = R\sin\left(\frac{\pi}{T}t\right)$, $y_d(t) = -R\cos\left(\frac{\pi}{T}t\right)$, $z_d(t) = \frac{R}{4} - \frac{R}{4}\cos\left(\frac{2\pi}{T}t\right)$ and $\psi_d(t) = \arctan\frac{\dot{y}_d(t)}{\dot{x}_d(t)} = \frac{\pi}{T}t$.

Ascending turn We set $x_d(t) = R\sin\left(\frac{\pi}{T}t\right)$, $y_d(t) = -R\cos\left(\frac{\pi}{T}t\right)$, $z_d(t) = \frac{R}{4} - \frac{R}{4}\cos\left(\frac{\pi}{T}t\right)$ and $\psi_d(t) = \arctan\frac{\dot{y}_d(t)}{\dot{x}_d(t)} = \frac{\pi}{T}t$.

Landing The landing manoeuvre takes place in the xz-plane. We fit 3$^{\text{th}}$ order polynomials to the x and z curve so that $x(0) = z(0) = 0$ and $x(T) = z(T) = R$. Additionally we constraint the input and output velocities so that $v_x(0) = 20\text{m}\,\text{s}^{-1}$, $v_x(T) = 1\text{m}\,\text{s}^{-1}$, $v_z(0) = 1\text{m}\,\text{s}^{-1}$ and $v_z(T) = 5\text{m}\,\text{s}^{-1}$. This simulates the deceleration manoeuvre from forward flight to the hovering regime.

Take-off The same procedure is used as for the landing manoeuvre with following velocity constraints $v_x(0) = 1\text{m}\,\text{s}^{-1}$, $v_x(T) = 20\text{m}\,\text{s}^{-1}$, $v_z(0) = 5\text{m}\,\text{s}^{-1}$ and $v_z(T) = 1\text{m}\,\text{s}^{-1}$. This simulates the acceleration manoeuvre from hovering to the forward flight regime.

APPENDIX B
CASCADED STATIC FEEDBACK LINEARIZATION

The control strategy from Ritz et al. is discussed here briefly [9]. The strategy can be described as a cascaded static feedback linearization approach. First a desired force, f_c, is calculated based on the static feedback linearization principle.

$$\begin{aligned} f_c(\xi, \xi_d, \dot{\xi}_d) &= m\dot{v}_c(\xi, \xi_d, \dot{\xi}_d) + mg \\ \dot{v}_c(\xi, \xi_d, \dot{\xi}_d) &= \dot{v}_d - \mathrm{K}_p^1(v - v_d) - \mathrm{K}_p^0(p - p_d) \end{aligned} \tag{29}$$

Second, an instantaneous desired attitude, $\mathrm{R}_c \in \mathfrak{SO}(3)$ with $\mathfrak{SO}(3)$ the special orthogonal group, is calculated along the reference flight path. The aim of the desired attitude is to rotate the vehicle so that the aerodynamic force, f, aligns with the desired force, f_c. The desired attitude, R_c, is calculated using ψ_d. The desired roll and pitch are calculated using (7) and (8) substituting the desired force, f_c, for f. Ritz et al. used the additional condition $\psi_d = \arctan \frac{\dot{y}_d}{\dot{x}_d}$. Then, based on R_c, a desired moment, m_c, is calculated again according the static feedback linearization principle.

$$\tau_c'''(\xi, \xi_d, \dot{\xi}_d) = \mathrm{I}\dot{\omega}_c'''(\xi, \xi_d, \dot{\xi}_d) + \omega''' \times \mathrm{I}\omega''' \tag{30}$$

To calculate the desired control velocity, $\dot{\omega}_c'''$, Ritz et al. propose nonlinear MPC which they implement using look-up tables. For small errors, they state that $\dot{\omega}_c''' \approx \mathrm{K}_q^0 \Delta q$, $\Delta\hat{q} = \mathrm{R}(q)^{-1}\mathrm{R}_c$ with $\hat{\cdot}$ the skew-symmetric operator. We found empirically that the following control is also stabilizing

$$\dot{\omega}_c'''(\xi, \xi_d, \dot{\xi}_d) = \mathrm{K}_q^1(\omega''' - \omega_d''') + \mathrm{K}_q^0 \Delta q \tag{31}$$

The control force and torque, f_c and τ_c respectively, are then transformed into the control inputs T_1, T_2, δ_1 and δ_2 relying on the flat inverse dynamics, see eq. (8) and (9). Only upon achieving the instantaneous desired attitude, the desired force can be applied. Therefore it is important that the time constant of the attitude controller is chosen a magnitude smaller than that of the position controller.

REFERENCES

[1] Haowei Gu, Ximin Lyu, Zexiang Li, Shaojie Shen, and Fu Zhang. Development and experimental verification of a hybrid vertical take-off and landing (vtol) unmanned aerial vehicle (uav). In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 160–169. IEEE, 2017.

[2] Philipp Hartmann, Carsten Meyer, and Dieter Moormann. Unified velocity control and flight state transition of unmanned tilt-wing aircraft. *Journal of guidance, control, and dynamics*, 40(6):1348–1359, 2017.

[3] Adnan S Saeed, Ahmad Bani Younes, Chenxiao Cai, and Guowei Cai. A survey of hybrid unmanned aerial vehicles. *Progress in Aerospace Sciences*, 98:91–105, 2018.

[4] Leandro R Lustosa, François Defaÿ, and Jean-Marc Moschetta. Global singularity-free aerodynamic model for algorithmic flight control of tail sitters. *Journal of Guidance, Control, and Dynamics*, 42(2):303–316, 2019.

[5] Boyang Li, Jingxuan Sun, Weifeng Zhou, Chih-Yung Wen, Kin Huat Low, and Chih-Keng Chen. Transition optimization for a vtol tail-sitter uav. *IEEE/ASME transactions on mechatronics*, 25(5):2534–2545, 2020.

[6] Ezra A. Tal and Sertac Karaman. Global trajectory-tracking control for a tailsitter flying wing in agile uncoordinated flight. In *AIAA Aviation 2021 Forum*, page 3214, 2021.

[7] Romain Chiappinelli and Meyer Nahon. Modeling and control of a tailsitter uav. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 400–409. IEEE, 2018.

[8] Ewoud JJ Smeur, Murat Bronz, and Guido CHE de Croon. Incremental control and guidance of hybrid aircraft applied to a tailsitter unmanned air vehicle. *Journal of Guidance, Control, and Dynamics*, 43(2):274–287, 2020.

[9] Robin Ritz and Raffaello D'Andrea. A global controller for flying wing tailsitter vehicles. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2731–2738. IEEE, 2017.

[10] P Smith. A simplified approach to nonlinear dynamic inversion based flight control. In *23rd Atmospheric Flight Mechanics Conference*, page 4461, 1998.

[11] Barton Bacon and Aaron Ostroff. Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4565, 2000.

[12] Thomas J Stastny, Adyasha Dash, and Roland Siegwart. Nonlinear mpc for fixed-wing uav trajectory tracking: Implementation and flight experiments. In *AIAA guidance, navigation, and control conference*, page 1512, 2017.

[13] Boyang Li, Weifeng Zhou, Jingxuan Sun, Chih-Yung Wen, and Chih-Keng Chen. Development of model predictive controller for a tail-sitter vtol uav in hover flight. *Sensors*, 18(9):2859, 2018.

[14] Leonard Bauersfeld, Lukas Spannagl, Guillaume JJ Ducard, and Christopher H Onder. Mpc flight control for a tilt-rotor vtol aircraft. *IEEE Transactions on Aerospace and Electronic Systems*, 57(4):2395–2409, 2021.

[15] Robin Ritz and Raffaello D'Andrea. A global strategy for tailsitter hover control. In *Robotics Research: Volume 1*, pages 21–37. Springer, 2017.

[16] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6):1327–1361, 1995.

[17] Gerasimos G Rigatos. Differential flatness theory and flatness-based control. In *Nonlinear Control and Filtering Using Differential Flatness Approaches*, pages 47–101. Springer, 2015.

[18] Tom Lefebvre, Sander De Witte, Thomas Neve, and Guillaume Crevecoeur. Differential flatness of slider–pusher systems for constrained time optimal collision free path planning. *J. Dyn. Sys., Meas., Control*, 145(6):061001, 2023.

[19] Ezra Tal and Sertac Karaman. Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness. *IEEE Transactions on Control Systems Technology*, 29(3):1203–1218, 2020.

[20] Melissa Greeff and Angela P Schoellig. Flatness-based model predictive control for quadrotor trajectory tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6740–6745. IEEE, 2018.

[21] Rumit Kumar, Alireza Nemati, Manish Kumar, Rajnikant Sharma, Kelly Cohen, and Franck Cazaurang. Tilting-rotor quadcopter for aggressive flight maneuvers using differential flatness based flight controller. In *Dynamic Systems and Control Conference*, volume 58295, page V003T39A006. American Society of Mechanical Engineers, 2017.

[22] Jolan Wauters, Tom Lefebvre, and Guillaume Crevecoeur. Multi-objective co-design for mission-specific development of unmanned aerial systems. In *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 17–24. IEEE, 2023.

[23] Hong-Gi Lee. Linearization of nonlinear control systems. *Linearization of Nonlinear Control Systems*, pages 1–589, 2022.

[24] Jean Lévine. On the equivalence between differential flatness and dynamic feedback linearizability. *IFAC Proceedings Volumes*, 40(20):338–343, 2007.

[25] H.K. Khalil. *Nonlinear Control*. Always Learning. Pearson Education Limited, 2014.

[26] Yirmeyahu J Kaminski, Jean Lévine, and François Ollivier. Intrinsic and apparent singularities in differentially flat systems, and application to global motion planning. *Systems & Control Letters*, 113:117–124, 2018.