**Nationaal Lucht- en Ruimtevaartlaboratorium**

National Aerospace Laboratory NLR


NLR-TP-2014-287

# Flatness Based Trajectory Generation For A Helicopter UAV

S. Taamallah

**Nationaal Lucht- en Ruimtevaartlaboratorium**
National Aerospace Laboratory NLR

Anthony Fokkerweg 2
P.O. Box 90502
1006 BM  Amsterdam
The Netherlands
Telephone +31 (0)88 511 31 13
Fax +31 (0)88 511 32 10
www.nlr.nl

**Nationaal Lucht- en Ruimtevaartlaboratorium**
National Aerospace Laboratory NLR

## Executive summary

# Flatness Based Trajectory Generation For A Helicopter UAV

**Problem area**
A main challenge, in the realm of vehicle guidance systems, consists in the on-line computation of accurate optimal trajectories. In particular, for high-bandwidth plants such as helicopter Unmanned Aerial Vehicles (UAVs), stringent real-time timing constraints may often need to be met.

**Description of work**
It is the purpose of this paper to present a novel trajectory planner framework, anchored in the combined paradigms of differential flatness and neural networks, and allowing for a computationally tractable on-line use, in a hard real-time environment.

**Results and conclusions**
The proposed approach is tested for the case of power-off, or autorotative, landing trajectories for a small-scale helicopter UAV.

**Applicability**
Development of flight control systems for small-scale UAV helicopters.

**Report no.**
NLR-TP-2014-287

**Author(s)**
S. Taamallah

**Report classification**
UNCLASSIFIED

**Date**
July 2014

**Knowledge area(s)**
Helikoptertechnologie

**Descriptor(s)**
Unmanned Aerial Vehicle (UAV)
Helicopter Trajectory Planning

This report is based on a presentation held at the AIAA. Guidance, Navigation, and Control Conference, August 19 – 22, 2013, Boston, Massachusetts.

# Flatness Based Trajectory Generation For A Helicopter UAV

**Nationaal Lucht- en Ruimtevaartlaboratorium**, National Aerospace Laboratory NLR

Anthony Fokkerweg 2, 1059 CM Amsterdam,
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands
Telephone +31 88 511 31 13, Fax +31 88 511 32 10, Web site: www.nlr.nl

**Nationaal Lucht- en Ruimtevaartlaboratorium**
National Aerospace Laboratory NLR

NLR-TP-2014-287

# Flatness Based Trajectory Generation For A Helicopter UAV

S. Taamallah

This report is based on a presentation held at the AIAA Guidance, Navigation, and Control Conference, August 19 – 22, 2013, Boston, Massachusetts.



| | |
|---|---|
| Customer | National Aerospace Laboratory NLR |
| Contract number | ---- |
| Owner | NLR |
| Division NLR | Aerospace Systems and Applications |
| Distribution | Unlimited |
| Classification of title | Unclassified |
| | July 2014 |

Approved by:

| Author | Reviewer | Managing department |
|---|---|---|
| Date: 08/07/2014 | Date: 14/07/2014 | Date: 17/7/2014 |

# Flatness Based Trajectory Generation For A Helicopter UAV

Skander Taamallah*†

*National Aerospace Laboratory (NLR), 1059CM Amsterdam, The Netherlands*

**A main challenge, in the realm of vehicle guidance systems, consists in the on-line computation of accurate optimal trajectories. In particular, for high-bandwidth plants such as helicopter Unmanned Aerial Vehicles (UAVs), stringent real-time timing constraints may often need to be met. Hence it is the purpose of this paper to present such a novel trajectory planner framework, anchored in the combined paradigms of differential flatness and neural networks, and allowing for computationally tractable optimal control problems. We conclude by presenting simulation examples, for the case of a helicopter UAV in autorotation, that demonstrates the applicability of our proposed approach.**

## I. Introduction

Designing an advanced vehicle guidance system, or Trajectory Planner (TP), may often be tantamount to solving complex optimization problems. For complex plants, most nonlinear constrained optimization problems are typically computationally intensive (real-time computation), and/or memory intensive (off-line computation), hence solving such optimization problems may often lead to computational difficulties or intractability.[1] Hence a plethora of technical avenues, addressing these planning problems, have been researched over the past years, such as: (i) cell decomposition, (ii) potential fields, (iii) roadmaps and hybrid systems, (iv) inverse dynamics and differential flatness, (v) Mixed Integer Linear Programming (MILP), (vi) Receding Horizon Control (RHC), (vii) optimal control, and (viii) evolutionary/genetic algorithms, with specific benefits for each method.[2–6]

In this paper, we have chosen to base our TP upon the concept of differential flatness. The seminal ideas of differential flatness were introduced in the early 90s in Ref. 7–9 as part of a paradigm in which certain differential algebraic representations of dynamical systems are equivalent. Indeed, flatness can be seen as a a subclass of the set of controllable nonlinear systems,[10] or as a system's geometric property,[11] independent of coordinate choice, or as a Lie-Bäcklund equivalence property,[12,13] where a complete parametrization of all system variables - inputs, states, and outputs - may be given in terms of a finite set of independent variables, called flat outputs, and a finite number of their derivatives.[13,14]

Flatness comes with two important benefits. First, it offers a particularly well adapted framework for solving inverse dynamics problems.[11,15] Indeed, flatness implies the absence of so-called zero dynamics, allowing for a one-to-one correspondence between trajectories of the input-state system and trajectories of the flat output, in which case the nonlinear system can be feedback linearized using endogenous dynamic feedback.[12] This allows trajectory generation and tracking for non-minimum phase systems by exact linearization.[16,17] Second, and in the framework of nonlinear optimal control, flat parameterizations result in optimization problems with fewer variables,[18] i.e. by the complete elimination of the dynamical constraints. In this case, a trajectory generation problem is transformed from a dynamic to an algebraic one, in which the flat outputs are parametrized over a space of basis functions, for which the generation of feasible trajectories is reduced to a classical algebraic interpolation or collocation problem.[11,19] This allows, in principle, for

*R&D Engineer, Aircraft Systems Department, National Aerospace Laboratory (NLR), 1059CM Amsterdam, The Netherlands.

†Ph.D. Student, Delft Center for Systems and Control (DCSC), Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, 2628CD Delft, The Netherlands.

significant computational advantage [a].

It is in general difficult to determine whether a given nonlinear system is flat, although several methods for constructing flat outputs have been documented in the literature.[14,22–24] As an example it is known that a system's Huygens center of oscillations may qualify as a flat output.[8,16,17] Further rules include also the following: (i) all linear systems are flat, (ii) all nonlinear systems which are static and dynamic feedback linearizable are flat, (iii) fully actuated systems are flat, and (iv) finally under-actuated systems may or may not be flat. With regard to applications, it was shown that simplified dynamics of aircraft and Vertical Take-Off and Landing (VTOL) aircraft are flat,[15,25–28] simplified helicopter dynamics is flat,[14,29–32] simplified quadrotor dynamics is flat,[33–37] simplified planetary lander dynamics is flat,[38] and simplified reentry vehicle dynamics is also flat,[39] whereas more realistic vehicle models are in general non-differentially flat, e.g. Ref. 10, 14 for the helicopter case. For additional interesting contributions in the application of flatness towards trajectory planning, the reader is referred to: (i) Ref. 40 for a multi-vehicle system, (ii) Ref. 9, 41 for the case where the motion is not subject to inequality constraints, and (iii) Ref. 42–44 for the case where inequality constraints have been added.

### I.A. Problem Statement

The purpose of this paper is to present a TP concept, based on a high-fidelity model plant, that computes optimal trajectories subject to system and environment constraints, with the following specifications

(1) The model shall be such that accurate Open-Loop (OL) optimal trajectories may be obtained.

(2) The TP shall be based upon the concept of differential flatness, as to retain high computational efficiency, e.g. for on-line use in a hard real-time environment where stringent timing constraints may need to be met, especially for high-bandwidth systems.

Our model is defined by a low-order thirteen-states vector [b]

$$\mathbf{x} = \begin{pmatrix} x_N & x_E & x_Z & \phi & \theta & \psi & u & v & w & p & q & r & \Omega_{MR} \end{pmatrix}^T \tag{1}$$

with the states being defined in Appendix A. Here, we have only retained the rigid-body equations of motions and the main rotor RPM (to allow for the computation of autorotative flight conditions). Indeed, the higher-order main rotor phenomena (dynamic inflow and blade flap/lag[45,46]) may be taken into account through either (i) their corresponding steady-state expressions, and/or (ii) the addition of empirical coefficients resulting in a grey-box modeling paradigm.[47,48] By so doing we have reduced the model order, with the obvious advantage of great computational savings, see also our discussion in Ref. 49. The bandwidth of the neglected dynamics is generally higher than the bandwidth of the vehicle flight mechanics, and higher than any flight mechanics closed-loop controller bandwidth. Hence, and on the grounds of this time-scale separation principle,[50] the lack of high frequency modeling detail becomes typically justifiable and acceptable for flight mechanics applications,[1] while enduring a minimal loss in accuracy and fidelity.

Regarding the inputs, a helicopter uses four control inputs defined as

$$\mathbf{u}_\theta = \begin{pmatrix} \theta_0 & \theta_{TR} & \theta_{1c} & \theta_{1s} \end{pmatrix}^T \tag{2}$$

also given in Appendix A. Now, and even with the modeling simplifications outlined here-above, high-fidelity helicopter models may still be rather complex. In fact, high-fidelity helicopter models based on the states given in $\mathbf{x}$, and plant inputs $\mathbf{u}_\theta$, are known to be non-differentially flat. To circumvent this difficulty, a standard approach has often consisted in progressively simplifying the model until it indeed becomes flat. Now, rather than generating optimal trajectories based upon such simplified representations,

[a] Note that, in the presence of constraints, flatness parameterization implies a path constraint on the flat outputs, resulting from complex transformations of the control and/or state regions. These transformations may lead to a loss of convexity, which may be detrimental to real-time optimal control computations.[18,20,21] However, it is our experience that for complex, high-order, highly nonlinear plants, the benefits from the elimination of the dynamical constraints outweigh the disadvantages due to path constraints on the flat outputs.

[b] Vectors in this paper are printed in boldface.

we propose here an alternative and novel approach, consisting in (i) using total aerodynamic forces and moments as plant inputs, resulting in a system which is exactly flat, and (ii) relying upon a separate model - a simplified Inverse-Simulation (IS),[51–53] Neural Networks (NN)[54,55] based transcription - to account for the relationship between $\mathbf{u}_\theta$ and the plant's total forces and moments. This approach will be illustrated for the more demanding case of helicopter UAV autorotation,[56] although the outlined machinery may easily be transposed to any other trajectory problem.

## II. Modeling

As mentioned in the previous section, to derive the flat parametrization we will use a model having total aerodynamic forces and moments as plant inputs. For the three-Dimensional (3D) aerodynamic forces experienced by the fuselage Center of Gravity (CG), in the body frame $F_b$, we have

$$\mathbf{F}^b_{aero,G_{Fus}} = (F^b_{Xaero,G_{Fus}} \quad F^b_{Yaero,G_{Fus}} \quad F^b_{Zaero,G_{Fus}})^T \tag{3}$$

Similarly, the total 3D moments, expressed at the fuselage[c] CG, in frame $F_b$, are given by

$$\mathbf{M}^b_{G_{Fus}} = (M^b_{X,G_{Fus}} \quad M^b_{Y,G_{Fus}} \quad (N^b_{(MR)} + \bar{N}^b_{(MR)}))^T \tag{4}$$

with the total yaw term $M^b_{Z,G_{Fus}} = (N^b_{(MR)} + \bar{N}^b_{(MR)})$, where the component $N^b_{(MR)}$ represents the torque contribution from the Main Rotor (MR) only, i.e. a torque induced by the blade lift and drag projections in the plane of rotation, and $\bar{N}^b_{(MR)}$ being the contribution from all other sub-systems. This sub-division is necessary to later be able to account for MR Rotations per Minute (RPM) dynamics. Hence, we can define a seven-inputs control vector as

$$\mathbf{u} = \left( F^b_{Xaero,G_{Fus}} \quad F^b_{Yaero,G_{Fus}} \quad F^b_{Zaero,G_{Fus}} \quad M^b_{X,G_{Fus}} \quad M^b_{Y,G_{Fus}} \quad N^b_{(MR)} \quad \bar{N}^b_{(MR)} \right)^T \tag{5}$$

Now, classical Newtonian mechanics and the fundamental relationship of kinematics provide us with the standard twelve-state rigid body equations of motion. Following notations of Ref. 57, we get

$$\begin{pmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_Z \end{pmatrix}^o = \begin{pmatrix} V_N \\ V_E \\ V_Z \end{pmatrix}^o \qquad \begin{pmatrix} V_N \\ V_E \\ V_Z \end{pmatrix}^o = \mathbb{T}_{ob}. \begin{pmatrix} u \\ v \\ w \end{pmatrix}^b \tag{6}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}^b = - \begin{pmatrix} q.w - r.v \\ r.u - p.w \\ p.v - q.u \end{pmatrix}^b + g. \begin{pmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{pmatrix}^b + \frac{\mathbf{F}_{aero,G_{Fus}}{}^b}{m_{Fus}} \tag{7}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix}^b = \mathbb{I}^{-1}_{Fus}. \left( \mathbf{M}^b_{G_{Fus}} - \begin{pmatrix} p \\ q \\ r \end{pmatrix}^b \times \left( \mathbb{I}_{Fus}. \begin{pmatrix} p \\ q \\ r \end{pmatrix}^b \right) \right) \tag{8}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}^b = \begin{bmatrix} 1 & \sin\theta.\frac{\sin\phi}{\cos\theta} & \sin\theta.\frac{\cos\phi}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} . \begin{pmatrix} p \\ q \\ r \end{pmatrix}^b \tag{9}$$

[c]Note that fuselage inertia and fuselage CG are used here rather than vehicle inertia and vehicle CG, since in the moments term $\mathbf{M}^b_{G_{Fus}}$ we have already accounted for rotor moments due to main rotor inertial loads.

$$\mathbb{T}_{ob} = \begin{bmatrix} \cos\theta\cos\psi & \sin\theta\sin\phi\cos\psi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\phi\sin\psi \\ \sin\psi\cos\theta & \sin\theta\sin\phi\sin\psi + \cos\psi\cos\phi & \sin\theta\cos\phi\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{10}$$

with the states and parameters being defined in Appendix A.

Further, the total forces can be split into contributions from the Tail Rotor (TR), Fuselage (Fus), Vertical Tail (VT), and Horizontal Tail (HT)

$$\mathbf{F}^b_{aero,G_{Fus}} = \begin{pmatrix} F_{Xaero,G_{Fus}} \\ F_{Yaero,G_{Fus}} \\ F_{Zaero,G_{Fus}} \end{pmatrix}^b = \begin{pmatrix} F_{x_{MR}} \\ F_{y_{MR}} \\ F_{z_{MR}} \end{pmatrix}^b + \begin{pmatrix} F_{x_{TR}} \\ F_{y_{TR}} \\ F_{z_{TR}} \end{pmatrix}^b + \begin{pmatrix} F_{x_{Fus}} \\ F_{y_{Fus}} \\ F_{z_{Fus}} \end{pmatrix}^b + \begin{pmatrix} F_{x_{VT}} \\ F_{y_{VT}} \\ F_{z_{VT}} \end{pmatrix}^b + \begin{pmatrix} F_{x_{HT}} \\ F_{y_{HT}} \\ F_{z_{HT}} \end{pmatrix}^b \tag{11}$$

Whereas the total moments, which also include the components due to the non-collocation of the vehicle CG and fuselage CG, are given by

$$\begin{aligned} \mathbf{M}^b_{G_{Fus}} = \begin{pmatrix} M_{X,G_{Fus}} \\ M_{Y,G_{Fus}} \\ M_{Z,G_{Fus}} \end{pmatrix}^b = \begin{pmatrix} M_{x_{MR}} \\ M_{y_{MR}} \\ M_{z_{MR}} \end{pmatrix}^b + \begin{pmatrix} M_{x_{TR}} \\ M_{y_{TR}} \\ M_{z_{TR}} \end{pmatrix}^b + \begin{pmatrix} M_{x_{Fus}} \\ M_{y_{Fus}} \\ M_{z_{Fus}} \end{pmatrix}^b + \begin{pmatrix} M_{x_{VT}} \\ M_{y_{VT}} \\ M_{z_{VT}} \end{pmatrix}^b + \begin{pmatrix} M_{x_{HT}} \\ M_{y_{HT}} \\ M_{z_{HT}} \end{pmatrix}^b \\ + \begin{pmatrix} -y_{Fus}.F_{Zaero,G_{Fus}} + z_{Fus}.F_{Yaero,G_{Fus}} \\ -z_{Fus}.F_{Xaero,G_{Fus}} + x_{Fus}.F_{Zaero,G_{Fus}} \\ -x_{Fus}.F_{Yaero,G_{Fus}} + y_{Fus}.F_{Xaero,G_{Fus}} \end{pmatrix}^b \end{aligned} \tag{12}$$

These forces and moments may be computed from comprehensive helicopter models, such as in Ref. 45–47.

Next, the MR yaw-moment $M^b_{z_{MR}}$ component includes also terms due to the non-alignment, in the body $(x, y)$ plane, between the main rotor hub and the fuselage CG.

$$M^b_{z_{MR}} = N^b_{(MR)} + x_H.F^b_{y_{MR}} - y_H.F^b_{x_{MR}} \tag{13}$$

Now we rewrite Eq (12), using Eq (13), as

$$\mathbf{M}^b_{G_{Fus}} = \begin{pmatrix} M_{X,G_{Fus}} \\ M_{Y,G_{Fus}} \\ M_{Z,G_{Fus}} \end{pmatrix}^b = \begin{pmatrix} M_{X,G_{Fus}} \\ M_{Y,G_{Fus}} \\ N_{(MR)} + \bar{N}_{(MR)} \end{pmatrix}^b \tag{14}$$

with

$$\bar{N}^b_{(MR)} = x_H.F^b_{y_{MR}} - y_H.F^b_{x_{MR}} + M^b_{z_{TR}} + M^b_{z_F} + M^b_{z_{VT}} + M^b_{z_{HT}} - x_{Fus}.F^b_{Yaero,G_{Fus}} + y_{Fus}.F^b_{Xaero,G_{Fus}} \tag{15}$$

Next, the main rotor RPM dynamics is related to the available and required power by the following expression[58]

$$N_b.I_b.\Omega_{MR}.\dot{\Omega}_{MR} = P_{shaft} - P_{req} \tag{16}$$

with $P_{shaft}$ being the available shaft power, $P_{req}$ the required power to keep the vehicle aloft, and all other terms defined in Appendix A. This latter is the sum of MR induced and profile power, TR induced and profile power, power plant transmission losses, vehicle parasite power (i.e. drag due to fuselage, landing skids, rotor hub, etc), and finally MR, TR, and fuselage aerodynamic interference losses.[59]

For the case of autorotation, following for instance an engine failure, a first-order response in $P_{shaft}$ is generally assumed to represent the power decay,[60,61] we have

$$\dot{P}_{shaft} = -\frac{P_{shaft}}{\tau_p} \tag{17}$$

with $\tau_p$ a to-be-identified time constant. For the required power $P_{req}$, we simplify the model by only considering the contributions from the MR as

$$P_{req} \simeq P_{MR} = N_{(MR)}.\Omega_{MR} \tag{18}$$

If, at engine failure, we were to assume instantaneous power loss $P_{shaft} = 0$, then from Eq (16) and Eq (18) we obtain

$$\dot{\Omega}_{MR} = -\frac{N_{(MR)}}{N_b.I_b} \tag{19}$$

### II.A. Helicopter Control Inputs: Inverse-Simulation Based Neural Networks Model

A trajectory optimization based on a flat parametrization of the problem at hand will allow us to put trajectory constraints and additional bounds on the plant states $\mathbf{x}$, as defined in Eq (1), and on the inputs to the flat model $\mathbf{u}$, as defined in Eq (5), as both $\mathbf{x}$ and $\mathbf{u}$ depend on the flat outputs. However, rather than bounding $\mathbf{u}$, we actually need to be able to bound the helicopter control inputs $\mathbf{u}_\theta$, as defined in Eq (2). Hence, the idea here consists in finding a relation between $\mathbf{u}_\theta$ and the flat outputs, by expressing $\mathbf{u}_\theta$ as a function of $\mathbf{u}$ and possibly $\mathbf{x}$. In other words, we aim at finding smooth and continuous-time nonlinear mappings, $g_{\theta_0}(\cdot)$, $g_{\theta_{TR}}(\cdot)$, $g_{\theta_{1c}}(\cdot)$, and $g_{\theta_{1s}}(\cdot)$ s.t.

$$\mathbf{u}_\theta = \begin{pmatrix} \theta_0(t) \\ \theta_{TR}(t) \\ \theta_{1c}(t) \\ \theta_{1s}(t) \end{pmatrix} = \begin{pmatrix} g_{\theta_0}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) \\ g_{\theta_{TR}}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) \\ g_{\theta_{1c}}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) \\ g_{\theta_{1s}}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) \end{pmatrix} \tag{20}$$

with $\tilde{\mathbf{x}} \in \mathbb{R}^{\tilde{n}_x}$, $\tilde{\mathbf{u}} \in \mathbb{R}^{\tilde{n}_u}$, $\tilde{n}_x \leq 13, \tilde{n}_u \leq 7$ representing either, the full state and control vectors respectively as given in Eq (1) and Eq (5), or a subset of these vectors. As can be seen from Eq (20), the chosen framework is somewhat reminiscent of Inverse-Simulation (IS) schemes, namely determining the control inputs that allow a helicopter model to fly a specified maneuver. A wide plethora of highly-effective, yet computationally intensive, IS methods have been devised over the years, i.e. so-called differential, integration-based, or global methods.[51–53,62] They generally consist in solving non-linear optimization problems, subject to system's dynamical constraints and desired trajectories.

In our case we are interested in finding a computationally tractable approach to Eq (20), since this latter will be used during the trajectory optimization process. The approach chosen in this paper is to base Eq (20) on a simple, static, approximate, yet computationally fast, Neural Networks (NN) model. These latter have indeed found a wide range of applications in control theory. Moreover, under mild assumptions on continuity and boundedness, a network of two layers, the first being hidden sigmoid and the second linear, can be trained to approximate any Input-Output (IO) relationship arbitrarily well, provided the number of neurons $L$ in the hidden layer is high enough.[54,55] Hence, we propose here to anchor the $g_{\theta_i}(\cdot), i \in \{0, TR, 1c, 1s\}$ modeling within the NN paradigm, as follows

$$
\begin{aligned}
&\forall i \in \{0, TR, 1c, 1s\} \\
&\theta_i(t) = g_{\theta_i}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) = C_{\theta_i}.\mathbf{s}_{\theta_i}(t) \\
&\mathbf{s}_{\theta_i}(t) = W_{o_{\theta_i}}.\kappa\big(W_{x_{\theta_i}}\tilde{\mathbf{x}}(t) + W_{u_{\theta_i}}\tilde{\mathbf{u}}(t) + W_{b_{\theta_i}}\big)
\end{aligned} \tag{21}
$$

where $W_{o_{\theta_i}} \in \mathbb{R}^{1\times L}$ and $W_{x_{\theta_i}} \in \mathbb{R}^{L\times \tilde{n}_x}$, $W_{u_{\theta_i}} \in \mathbb{R}^{L\times \tilde{n}_u}$, contain the output and hidden layer weights respectively. Further, $W_{b_{\theta_i}} \in \mathbb{R}^{L}$ contains the sets of biases in the hidden layer, $C_{\theta_i} \in \mathbb{R}$ contains the output linear map, and $\kappa(\cdot)$ is the activation function, taken as a continuous, diagonal, differentiable, and bounded static sigmoid nonlinearity.

## III. Flatness Formulation

We suppose that a plant's Nonlinear Model (NM), derived from first-principles, is available and given by

$$
\forall t \geq 0 \qquad \dot{\mathbf{x}}(t) = f\big(\mathbf{x}(t), \mathbf{u}(t)\big) \qquad \mathbf{y}(t) = \mathbf{x}(t) \tag{22}
$$

with $f(\cdot)$ a deterministic, Continuous-Time (CT) function of class $C^{\infty}$. Further, we have $\mathbf{x}(t) \in \mathcal{P}_x \subset \mathbb{R}^{n_x}$ the plant state, $\mathbf{y}(t) \in \mathcal{P}_y \subset \mathbb{R}^{n_y}$ the plant output, $\mathbf{u}(t) \in \mathcal{P}_u \subset \mathbb{R}^{n_u}$ the control input, $t$ the time variable, and $(\mathcal{P}_x, \mathcal{P}_y, \mathcal{P}_u)$ some compact sets. In this paper we will encompass our discussion within the CT framework since flatness has originally been defined in this framework.

**Definition 1 (Ref. 13)** *The explicit system given by Eq (22) is differentially flat if and only if there exists a flat output $\mathbf{z}(t) \in \mathcal{P}_z \subset \mathbb{R}^{n_u}$, two integers $r$ and $s$, a mapping $\psi : \mathbb{R}^{n_x} \times (\mathbb{R}^{n_u})^{s+1} \to \mathbb{R}^{n_u}$ of rank $n_u$ in a suitably chosen open subset, a mapping $\phi_0 : (\mathbb{R}^{n_u})^{r+1} \to \mathbb{R}^{n_x}$ of rank $n_x$ in a suitably chosen open subset, and a mapping $\phi_1 : (\mathbb{R}^{n_u})^{r+2} \to \mathbb{R}^{n_u}$ of rank $n_u$ in a suitably chosen open subset, such that $\mathbf{z} = \psi(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \cdots, \mathbf{u}^{(s)})$ implies that $\mathbf{x} = \phi_0(\mathbf{z}, \dot{\mathbf{z}}, \cdots, \mathbf{z}^{(r)})$, $\mathbf{u} = \phi_1(\mathbf{z}, \dot{\mathbf{z}}, \cdots, \mathbf{z}^{(r+1)})$, and the differential equation $\frac{d\phi_0(\cdot)}{dt} = f(\phi_0(\cdot), \phi_1(\cdot))$ are identically satisfied.*

Next, we show that our helicopter model, as outlined in Section II, becomes flat with the following seven states as flat outputs

$$
\mathbf{z} = \begin{pmatrix} x_N & x_E & x_Z & \phi & \theta & \psi & \Omega_{MR} \end{pmatrix}^T \tag{23}
$$

With this choice, the remaining six states of $\mathbf{x}$ in Eq (1), and all seven inputs of $\mathbf{u}$ in Eq (5), can be expressed in terms of the flat outputs $\mathbf{z}$ and their derivatives.

From Eq (6) and Eq (10) we obtain

$$
\begin{pmatrix} u \\ v \\ w \end{pmatrix}^b = \begin{pmatrix} -V_Z \sin\theta + V_N \cos\theta \cos\psi + V_E \cos\theta \sin\psi \\ -V_N \sin\psi \cos\phi + V_E \cos\psi \cos\phi + V_N \sin\theta \sin\phi \cos\psi + V_E \sin\theta \sin\phi \sin\psi + V_Z \sin\phi \cos\theta \\ V_Z \cos\phi \cos\theta + V_N \sin\phi \sin\psi - V_E \sin\phi \cos\psi + V_N \sin\theta \cos\psi \cos\phi + V_E \sin\theta \sin\psi \cos\phi \end{pmatrix} \tag{24}
$$

where, we have used $[\dot{x}_N \ \dot{x}_E \ \dot{x}_Z]^T = [V_N \ V_E \ V_Z]^T$. Now, from Eq (9) we get

$$
\begin{pmatrix} p \\ q \\ r \end{pmatrix}^b = \begin{pmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta \\ -\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta \end{pmatrix} \tag{25}
$$

From Eq (7) and Eq (24), and further using the derivative of Eq (24), we obtain for the force inputs

$$\begin{pmatrix} F_{Xaero,G_{Fus}} \\ F_{Yaero,G_{Fus}} \\ F_{Zaero,G_{Fus}} \end{pmatrix}^b = ...$$
$$m_{Fus}.\begin{pmatrix} g.\sin\theta - \dot{V}_Z\sin\theta + \dot{V}_E\cos\theta\sin\psi + \dot{V}_N\cos\theta\cos\psi \\ -g.\sin\phi\cos\theta - \dot{V}_N\sin\psi\cos\phi + \dot{V}_Z\sin\phi\cos\theta + \dot{V}_E\cos\psi\cos\phi + \dot{V}_E\sin\theta\sin\phi\sin\psi + \dot{V}_N\sin\theta\sin\phi\cos\psi \\ -g.\cos\phi\cos\theta + \dot{V}_Z\cos\phi\cos\theta + \dot{V}_N\sin\phi\sin\psi - \dot{V}_E\sin\phi\cos\psi + \dot{V}_N\sin\theta\cos\psi\cos\phi + \dot{V}_E\sin\theta\sin\psi\cos\phi \end{pmatrix} \tag{26}$$

Now from Eq (8) and Eq (25), and further taking the derivative of Eq (25), we obtain for the moments inputs

$$\begin{gathered} M^b_{X,G_{Fus}} = A.(\ddot{\phi} - \dot{\theta}\dot{\psi}\cos\theta - \ddot{\psi}\sin\theta) \\ -F.(\ddot{\theta}\cos\phi + \ddot{\psi}\sin\phi\cos\theta - \dot{\theta}\dot{\phi}\sin\phi + \dot{\psi}(\dot{\phi}\cos\phi\cos\theta - \dot{\theta}\sin\phi\sin\theta)) \\ -E.(-\ddot{\theta}\sin\phi - \dot{\theta}\dot{\phi}\cos\phi + \ddot{\psi}\cos\phi\cos\theta - \ddot{\psi}(\dot{\phi}\sin\phi\cos\theta + \dot{\theta}\cos\phi\sin\theta)) \\ +(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta)(-E.(\dot{\phi} - \dot{\psi}\sin\theta) - D.(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta) + C.(-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)) \\ -(-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)(-F.(\dot{\phi} - \dot{\psi}\sin\theta) + B.(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta) - D.(-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)) \end{gathered} \tag{27}$$

$$\begin{gathered} M^b_{Y,G_{Fus}} = -F.(\ddot{\phi} - \dot{\theta}\dot{\psi}\cos\theta - \ddot{\psi}\sin\theta) \\ +B.(\ddot{\theta}\cos\phi + \ddot{\psi}\sin\phi\cos\theta - \dot{\theta}\dot{\phi}\sin\phi + \dot{\psi}(\dot{\phi}\cos\phi\cos\theta - \dot{\theta}\sin\phi\sin\theta)) \\ -D.(-\ddot{\theta}\sin\phi - \dot{\theta}\dot{\phi}\cos\phi + \ddot{\psi}\cos\phi\cos\theta - \ddot{\psi}(\dot{\phi}\sin\phi\cos\theta + \dot{\theta}\cos\phi\sin\theta)) \\ +(-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)(A.(\dot{\phi} - \dot{\psi}\sin\theta) - F.(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta) - E.(-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)) \\ -(\dot{\phi} - \dot{\psi}\sin\theta)(-E.(\dot{\phi} - \dot{\psi}\sin\theta) - D.(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta) + C.(-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)) \end{gathered} \tag{28}$$

And using Eq (14) and Eq (19) we get

$$N^b_{(MR)} = -N_b.I_b.\dot{\Omega}_{MR} \tag{29}$$

$$\begin{gathered} \bar{N}^b_{(MR)} = -E.(\ddot{\phi} - \dot{\theta} * \dot{\psi}\cos\theta - \ddot{\psi}\sin\theta) \\ -D.(\ddot{\theta}\cos\phi + \ddot{\psi}\sin\phi\cos\theta - \dot{\theta}\dot{\phi}\sin\phi + \dot{\psi}(\dot{\phi}\cos\phi\cos\theta - \dot{\theta}\sin\phi\sin\theta)) \\ +C.(-\ddot{\theta}\sin\phi - \dot{\theta}\dot{\phi}\cos\phi + \ddot{\psi}\cos\phi\cos\theta - \ddot{\psi}(\dot{\phi}\sin\phi\cos\theta + \dot{\theta}\cos\phi\sin\theta)) \\ +(\dot{\phi} - \dot{\psi}\sin\theta)(-F.(\dot{\phi} - \dot{\psi}\sin\theta) + B.(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta) - D.(-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)) \\ -(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta)(A.(\dot{\phi} - \dot{\psi}\sin\theta) - F.(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta) - E.(-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)) \\ +N_b.I_b.\dot{\Omega}_{MR} \end{gathered} \tag{30}$$

### III.A. Flat Output Parametrization

To transform the problem from an infinite-dimensional one to a finite-one, a parametrization of the flat outputs over a space of basis functions is required. Here numerous alternatives are available, e.g. generic polynomial parameterizations have been addressed in Ref. 13, 14, 63, 64, spline parameterizations[65–67] have been applied in Ref. 34, 43, 68–72, whereas pseudospectral parameterizations have been used in Ref. 20, 38. In this paper, and with a view to using the most straightforward approach, we simply apply elementary polynomials parametrization as done in Ref. 13, 14. We get

$$\mathbf{z}(t) = \left( \sum_{i=0}^{n} a_{(i,1)}.t^i \ , \ \dots \ , \ \sum_{i=0}^{n} a_{(i,n_u)}.t^i \right)^T \tag{31}$$

with $t$ the time, and $a_{(i,j)}, i = 0, ..., n, j = 1, ..., n_u$ the to-be-identified polynomial coefficients, where from Ref. 13 we have $n \geq 2(r+1)+1$, with $r$ as given in Definition 1. For our helicopter model, as outlined in Section II with $n_u = 7$, and for the flat outputs, as given in Eq (23), we obtain $n = 5$ for each one of the first six flat outputs, and $n = 3$ for flat output $\Omega_{MR}$.

## IV. Optimal Trajectory Generation

We consider the following problem, consisting in minimizing a cost functional $J(a_{(i,j)}, \mathbf{u}_\theta, T_o, T_f)$, with the coefficients $a_{(i,j)}, i = 0, ..., n, j = 1, ..., n_u$ given in Eq (31), and the control inputs $\mathbf{u}_\theta$ defined in Eq (2), and computed from Eq (21). Further, the independent time variable $t$ is defined over the time domain $\Omega = (T_o, T_f)$, where the final time $T_f$ may be free or fixed. In the general problem formulation, the cost functional $J(\cdot)$ has contributions from a fixed cost $\Phi(\cdot)$, a running cost over time $\int_\Omega \Psi(\cdot)dt$, and is defined as

$$\begin{aligned} &J(\mathbf{x}(\mathbf{z}), \mathbf{u}(\mathbf{z}), \mathbf{u}_\theta, T_o, T_f) := \Phi(\mathbf{x}(\mathbf{z}), \mathbf{u}(\mathbf{z}), \mathbf{u}_\theta, T_o, T_f) + \textstyle\int_\Omega \Psi(\mathbf{x}(\mathbf{z}), \mathbf{u}(\mathbf{z}), \mathbf{u}_\theta, t)dt \\ &\mathbf{z}(a_{(i,j)}, t) = \left( \textstyle\sum_{i=0}^{n} a_{(i,1)}.t^i \ , \ \dots \ , \ \sum_{i=0}^{n} a_{(i,n_u)}.t^i \right)^T \end{aligned} \tag{32}$$

where, in this paper, the running cost $\int_\Omega \Psi(\cdot)dt$ is approximated by computing it at $K$ collocation points, evenly spaced on $\Omega$, resulting in set $\Omega_K$. Next, the cost functional $J(\cdot)$ is subject to initial and final-time boundary inequality conditions given by

$$\begin{aligned} &B_o\big(\mathbf{x}(\mathbf{z}(a_{(i,j)}, T_o)), \mathbf{u}(\mathbf{z}(a_{(i,j)}, T_o)), \mathbf{u}_\theta(T_o)\big) \quad \leq 0 \\ &\mathbf{z}(a_{(i,j)}, T_o) = \left( \textstyle\sum_{i=0}^{n} a_{(i,1)}.T_o^i \ , \ \dots \ , \ \sum_{i=0}^{n} a_{(i,n_u)}.T_o^i \right)^T \\ &B_f\big(\mathbf{x}(\mathbf{z}(a_{(i,j)}, T_f)), \mathbf{u}(\mathbf{z}(a_{(i,j)}, T_f)), \mathbf{u}_\theta(T_f)\big) \quad \leq 0 \\ &\mathbf{z}(a_{(i,j)}, T_f) = \left( \textstyle\sum_{i=0}^{n} a_{(i,1)}.T_f^i \ , \ \dots \ , \ \sum_{i=0}^{n} a_{(i,n_u)}.T_f^i \right)^T \end{aligned} \tag{33}$$

which may describe the initial and final trimmed flight conditions, while checking for actuators range limitations. Conjointly any algebraic trajectory inequality constraints, derived from the $K$ collocation points, are given by

$$\begin{aligned} &T\big(\mathbf{x}(\mathbf{z}(a_{(i,j)}, t_k)), \mathbf{u}(\mathbf{z}(a_{(i,j)}, t_k)), \mathbf{u}_\theta(t_k)\big) \quad \leq 0 \quad t_k \in \Omega_K \\ &\mathbf{z}(a_{(i,j)}, t_k) = \left( \textstyle\sum_{i=0}^{n} a_{(i,1)}.t_k^i \ , \ \dots \ , \ \sum_{i=0}^{n} a_{(i,n_u)}.t_k^i \right)^T \quad k = 1, ..., K \end{aligned} \tag{34}$$

For generality, the boundary and trajectory constraints Eq (33)-Eq (34) have been expressed as inequality constraints, equality constraints may simply be enforced by equating upper and lower bounds. Further, in Eq (32)-Eq (34) the functions $\Phi(\cdot)$, $\Psi(\cdot)$, $B_o(\cdot)$, $B_f(\cdot)$, and $T(\cdot)$ are assumed to be sufficiently smooth, i.e. at least $C^2$.

Note that the trajectory constraints, presented here-above, have a threefold objective: (i) account for vehicle's inherent physical and flight envelope limitations (bounds on speeds, attitude, and main rotor RPM), (ii) account for environmental constraints (the helicopter cannot descend below ground), and (iii) check for actuators dynamic and range limitations.

Finally, the solution to the trajectory planning gives the flat output polynomial coefficients $a_{(i,j)}, i = 0, ..., n, j = 1, ..., n_u$, which minimize the cost functional $J(\cdot)$, while enforcing the here-above predefined constraints Eq (33)-Eq (34)

$$\begin{aligned} &\hat{a}_{(i,j)} := \arg\min_{a_{(i,j)} \in \mathbb{R}} J(\mathbf{x}(\mathbf{z}), \mathbf{u}(\mathbf{z}), \mathbf{u}_\theta, T_o, T_f) \\ &\mathbf{z}(a_{(i,j)}, t) = \left( \textstyle\sum_{i=0}^{n} a_{(i,1)}.t^i \ , \ \dots \ , \ \sum_{i=0}^{n} a_{(i,n_u)}.t^i \right)^T \end{aligned} \tag{35}$$

## V. Simulation Results

We implemented our trajectory planner in a MATLAB® environment,[73] for the case of a two-bladed *Align T-REX* helicopter, with a 0.9 m rotor radius and a total mass of 7.75 kg, see Appendix B. First, the NN model for Eq (21) is chosen to be based upon the following inputs

$$\begin{aligned}
\theta_0 &:= g_{\theta_0}(w, \Omega_{MR}, F^b_{Zaero,G_{Fus}}, N^b_{(MR)}) \\
\theta_{TR} &:= g_{\theta_{TR}}(v, p, r, \phi, F^b_{Yaero,G_{Fus}}, M^b_{X,G_{Fus}}, \bar{N}^b_{(MR)}) \\
\theta_{1c} &:= g_{\theta_{1c}}(u, v, p, q, \phi, \theta, \Omega_{MR}, F^b_{Xaero,G_{Fus}}, F^b_{Yaero,G_{Fus}}, M^b_{X,G_{Fus}}, M^b_{Y,G_{Fus}}) \\
\theta_{1s} &:= g_{\theta_{1s}}(u, v, p, q, \phi, \theta, \Omega_{MR}, F^b_{Xaero,G_{Fus}}, F^b_{Yaero,G_{Fus}}, M^b_{X,G_{Fus}}, M^b_{Y,G_{Fus}})
\end{aligned} \tag{36}$$

with 4-neurons for $g_{\theta_0}(\cdot)$, 7-neurons for $g_{\theta_{TR}}(\cdot)$, and 11-neurons each for $g_{\theta_{1c}}(\cdot)$ and $g_{\theta_{1s}}(\cdot)$, and all these functions being based upon feedforward networks, with a hyperbolic tangent activation transfer function in the hidden layer, and backpropagation training for the weights and biases.

To train the NN model, we have used a high-order, nonlinear, helicopter flight dynamics model, based upon the work done in Ref. 45, 46, although a Flightlab®[74] model applies equally well. We simulate an instantaneous engine failure starting from hover[d], and then apply several sine-sweeps from 0.1 to 3 Hz, with varying amplitudes, for a duration of 3 seconds, on each control input channel separately. Now, the very low vehicle roll inertia, and the low pitch inertia (see Appendix B) have resulted in very noisy roll and pitch rates, due to blade flapping. Indeed, a power spectral density analysis revealed the presence of a high energy component around the (Nb=2)/Rev harmonic[e]. While the nominal RPM is of 1350, resulting in a 45 Hz 2/Rev frequency, the main rotor RPM is expected to reach the [750 - 1000] lower RPM range during autorotation. In that case a 750 RPM would result in vibrations at around 25 Hz frequency.

Hence, to facilitate the NN training, we had to low-pass the noisy roll and pitch rates, and the x- and y-forces and moments, with a zero-phase shift[f], digital Butterworth filter. Based upon the previous discussion, we decided to have less than 3 dB of ripple in the passband, defined from 0 to 20 Hz, and at least 30 dB of attenuation in the stopband, defined from 25 Hz to the Nyquist frequency (here 50 Hz in our case), which resulted in an 11th order Butterworth filter.[76]

Note that the NN model obtained in Eq (36) is rather approximate. First, due to the few, and relatively short, trajectories used to train the NN, and second, because of its simplistic, and static nature. Hence, we will only make use of this NN model during the trajectory optimization process, namely only to check the boundary conditions of helicopter control inputs as defined by Eq (33) and Eq (34). In other words, once the optimal trajectory has been computed, we will not generate the corresponding optimal input in order to use it in a feedforward control scheme. Only the optimal states will subsequently be used as reference setpoints for a feedback trajectory tracker.

The nonlinear optimization of Section IV may be chosen to minimize the rates of input $\mathbf{u}$ (as a reminder $\mathbf{u}$ is the input to the flat model, defined in Eq (5)), even though the presence of such a cost objective is not mandatory. Indeed, we noticed that the computational time can be substantially reduced if one were to search for feasible trajectories only, rather than optimal ones. Hence, in the sequel only feasible trajectories will be presented. Further we use $K = 6$ collocation points, evenly spaced between the initial and final tines, to enforce various trajectory constraints on states (e.g. tail rotor clearance when flaring) and control inputs. For a more in-depth review of cost objectives and trajectory constraints, see the results presented in Ref. 49, 77. Further, the optimization problem of Eq (35) is solved with the MATLAB function *fmincon* of the Optimization Toolbox, based upon an Interior Point (IP) method.[78–84]

We present next simulation results for an autorotative landing, with engine failure starting from a hover initial condition. The following constraints on initial states $\mathbf{x}_i$, final states $\mathbf{x}_f$, and trajectory, have also been

[d] To enhance the NN modeling accuracy, one should gather additional data corresponding to a range of starting conditions, by gridding the flight envelope domain.

[e] The harmonic is approximately at Nb/Rev since our main rotor is modeled as an articulated rotor, with hinge stiffness and offsets. These latter aspects will slightly shift the natural frequency of the flap motion away fro its nominal Nb/Rev value.[75]

[f] To avoid the introduction of delays

added. Initial $\mathbf{x}_i$ and final $\mathbf{x}_f$ states are given by

$$\begin{aligned}\mathbf{x}_i &= \begin{pmatrix} 0m & 0m & 30m & 0^\circ & 0^\circ & 0^\circ & 0m/s & 0m/s & 0m/s & 0^\circ/s & 0^\circ/s & 0^\circ/s & 1350RPM \end{pmatrix}^T \\ \mathbf{x}_f &= \begin{pmatrix} Free & Free & 0.5m & 0^\circ & 0^\circ & 180^\circ & 0m/s & 0m/s & 0m/s & 0^\circ/s & 0^\circ/s & 0^\circ/s & Free \end{pmatrix}^T \end{aligned} \tag{37}$$

with the following states constraints

$$\begin{aligned}\mathbf{x}_{min} = \begin{pmatrix} -200m & -200m & 0.5m & -48^\circ & -48^\circ & -360^\circ & ... \\ -5m/s & -2m/s & -10m/s & -100^\circ/s & -100^\circ/s & -100^\circ/s & 945RPM\ (=70\%.\Omega_{MR_{100\%}}) \end{pmatrix}^T \end{aligned} \tag{38}$$

$$\begin{aligned}\mathbf{x}_{max} = \begin{pmatrix} 200m & 200m & 50m & 48^\circ & 48^\circ & 360^\circ & ... \\ 20m/s & 2m/s & 15m/s & 100^\circ/s & 100^\circ/s & 100^\circ/s & 1485RPM\ (=110\%.\Omega_{MR_{100\%}}) \end{pmatrix}^T \end{aligned} \tag{39}$$

and for both cases, we have the input constraints

$$\begin{aligned}\mathbf{u}_{\theta_{min}} &= \begin{pmatrix} -5^\circ & -30^\circ & -7^\circ & -7^\circ \end{pmatrix}^T \\ \mathbf{u}_{\theta_{max}} &= \begin{pmatrix} 15^\circ & 31^\circ & 8^\circ & 8^\circ \end{pmatrix}^T \end{aligned} \tag{40}$$

Figure 1 and Figure 2 show the evolution of the body states and corresponding inertial components, Figure 3 shows the evolution of the main rotor RPM, whereas Figure 4 and Figure 5 visualize the required control inputs. A quick scan of the control input time-histories reveals that the identified NN model is rather inaccurate, in particular the tail rotor collective input is too high. Hence, additional training data need to be included, e.g. by adding trajectories starting from autorotative initial conditions (the current model is based on only four very short trajectories, starting from hover). On the other hand, the states behave as expected, e.g. the 180° yaw turn is clearly recognizable.

## VI. Conclusion

In this paper we have presented a novel trajectory planner framework, anchored in the combined paradigms of differential flatness and Neural Network (NN), and allowing for a computationally tractable determination of optimal trajectories. The proposed approach was tested for the case of power-off, or autorotative, landing trajectories for a small-scale helicopter UAV. Our preliminary encouraging results invite further application of the here-presented approach. In particular, the required accuracy of the NN model warrants further investigations. Indeed, how to obtain just sufficiently rich data sets, for NN training, remains an open issue. Further research will also be dedicated towards the design of feedback trajectory trackers, in order to evaluate the complete guidance and control system.

## Appendix A: Nomenclature

Vectors in this paper are printed in boldface $\mathbf{X}$ and are defined in $\mathbb{R}^3$. A vector is qualified by its subscript while its superscript denotes the projection frame. Matrices are written in outline type $\mathbb{M}$. All frames are 3-D orthogonal and right-handed. Transformation matrices are denoted as $\mathbb{T}_{ij}$, with the two suffices signifying from frame $F_j$ to frame $F_i$.

- Frames
  - $F_o$ Vehicle carried normal earth frame (z-axis > 0 up)
  - $F_b$ Body (vehicle) frame (z-axis > 0 down)
- Angles between frames
  - $\psi$ Azimuth angle (yaw angle, heading)
  - $\theta$ Inclination angle (pitch angle, or elevation)
  - $\phi$ Bank angle (roll angle)
- Position
  - $x_N, x_E, x_Z$ Coordinates of vehicle Center of Gravity (CG) in $F_o$ frame
  - $x_H, y_H, z_H$ Position of MR hub wrt fuselage CG in $F_b$ frame
  - $x_{Fus}, y_{Fus}, z_{Fus}$ Position of fuselage CG wrt vehicle CG in $F_b$ frame
- Linear velocities are denoted $\mathbf{V}$ and their components $u, v, w$
  - $\mathbf{V}_{k,G}$ Kinematic velocity of the vehicle CG
  - $u_k^o = V_N$ $x$ component of $\mathbf{V}_{k,G}$ on $F_o$, $V_N$ North velocity
  - $v_k^o = V_E$ $y$ component of $\mathbf{V}_{k,G}$ on $F_o$, $V_E$ East velocity
  - $w_k^o = V_Z$ $z$ component of $\mathbf{V}_{k,G}$ on $F_o$, $V_Z$ Vertical velocity
  - $u_k^b = u$ $x$ component of $\mathbf{V}_{k,G}$ on body frame $F_b$
  - $v_k^b = v$ $y$ component of $\mathbf{V}_{k,G}$ on body frame $F_b$
  - $w_k^b = w$ $z$ component of $\mathbf{V}_{k,G}$ on body frame $F_b$
- Angular velocities are denoted $\mathbf{\Omega}$ and their components $p, q, r$
  - $p_k^b = p$ Roll velocity (roll rate) of the vehicle relative to the earth (frame $F_E$)
  - $q_k^b = q$ Pitch velocity (pitch rate) of the vehicle relative to the earth
  - $r_k^b = r$ Yaw velocity (yaw rate) of the vehicle relative to the earth
  - $\Omega_{MR}$ MR instantaneous angular velocity
- Acceleration
  - $g$ Acceleration due to gravity
- Mass and Inertia
  - $m_{Fus}$ Fuselage mass
  - $\mathbb{I}_{Fus} = \begin{bmatrix} A & -F & -E \\ -F & B & -D \\ -E & -D & C \end{bmatrix}$ Fuselage inertia
- Main Rotor (MR) properties
  - $N_b$ Number of blades
  - $I_b$ Blade 2nd mass moment (inertia about rotor shaft)
- Control Inputs
  - $\theta_0$ MR blade root collective pitch
  - $\theta_{1c}$ MR lateral cyclic pitch
  - $\theta_{1s}$ MR longitudinal cyclic pitch
  - $\theta_{TR}$ TR blade pitch angle

## Appendix B: Align T-REX Physical Parameters

| | Name | Parameter | Value | Unit |
|---|---|---|---|---|
| Environment | Air density | $\rho$ | 1.2367 | $kg/m^3$ |
| | Static temperature | $T$ | 273.15 + 15 | $K$ |
| | Specific heat ratio (air) | $\gamma$ | 1.4 | |
| | Gas constant (air) | $R$ | 287.05 | $J/kg.K$ |
| | Gravity constant | $g$ | 9.812 | $m/s^2$ |
| Vehicle | Total mass | $m$ | 7.75 | $kg$ |
| | Inertia moment wrt $x_b$ | $A$ | 0.0705 | $kg.m^2$ |
| | Inertia moment wrt $y_b$ | $B$ | 0.4760 | $kg.m^2$ |
| | Inertia moment wrt $z_b$ | $C$ | 0.2855 | $kg.m^2$ |
| | Inertia product wrt $x_b$ | $D$ | 0 | $kg.m^2$ |
| | Inertia product wrt $y_b$ | $E$ | 0.0018 | $kg.m^2$ |
| | Inertia product wrt $z_b$ | $F$ | 0 | $kg.m^2$ |
| Main Rotor | Direction of rotation | $\Gamma$ | CW (-1) | |
| | Number of blades | $N_b$ | 2 | |
| | Nominal angular velocity | $\Omega_{MR_{100\%}}$ | 131.37 | $rad/s$ |
| | Rotor radius from hub | $R_{rot}$ | 0.9 | $m$ |
| | Blade mass | $M_{bl}$ | 0.208 | $kg$ |
| | Spring restraint coef. due to flap | $K_{S_\beta}$ | 163.8 | $N.m/rad$ |
| | Distance between hub and flap hinge | $\Delta_e$ | 0.32 | $m$ |
| Tail Rotor | Number of blades | 2 | | |
| | Nominal angular velocity | $\Omega_{TR_{100\%}}$ | 612.61 | $rad/s$ |
| | Rotor radius from rotor hub | $R_{rot_{TR}}$ | 0.14 | $m$ |
| Actuators | MR collective | $\theta_0$ | [-5,15].$\pi$/180 | $rad$ |
| | TR collective | $\theta_{TR}$ | [-30,31].$\pi$/180 | $rad$ |
| | MR lateral cyclic | $\theta_{1c}$ | [-7,8].$\pi$/180 | $rad$ |
| | MR longitudinal cyclic | $\theta_{1s}$ | [-7,8].$\pi$/180 | $rad$ |
| | MR collective rate | $\dot{\theta}_0$ | [-52,52].$\pi$/180 | $rad/s$ |
| | TR collective rate | $\dot{\theta}_{TR}$ | [-120,120].$\pi$/180 | $rad/s$ |
| | MR lateral cyclic rate | $\dot{\theta}_{1c}$ | [-56,56].$\pi$/180 | $rad/s$ |
| | MR longitudinal cyclic rate | $\dot{\theta}_{1s}$ | [-56,56].$\pi$/180 | $rad/s$ |

## Appendix C: Simulation Results

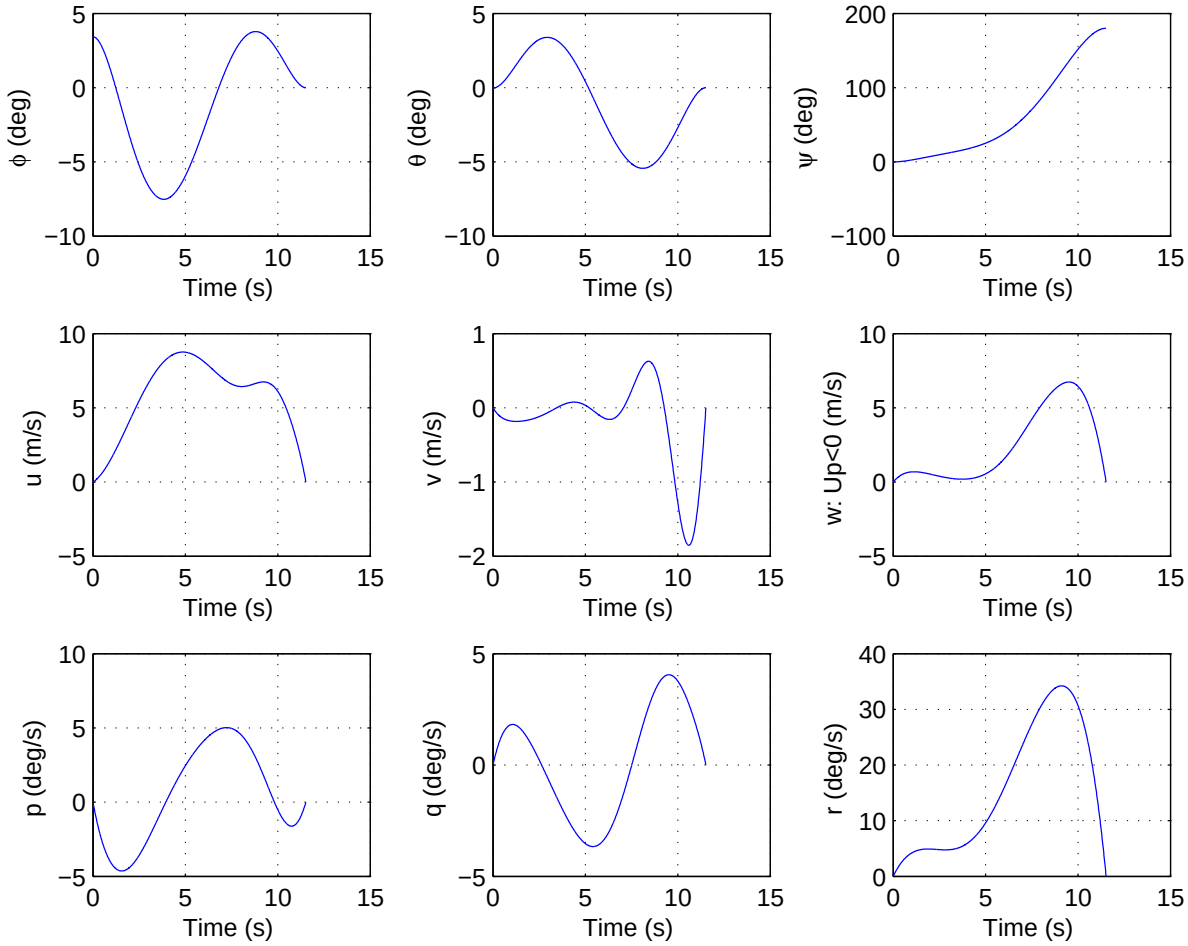


**Figure 1. Vehicle response in body frame**

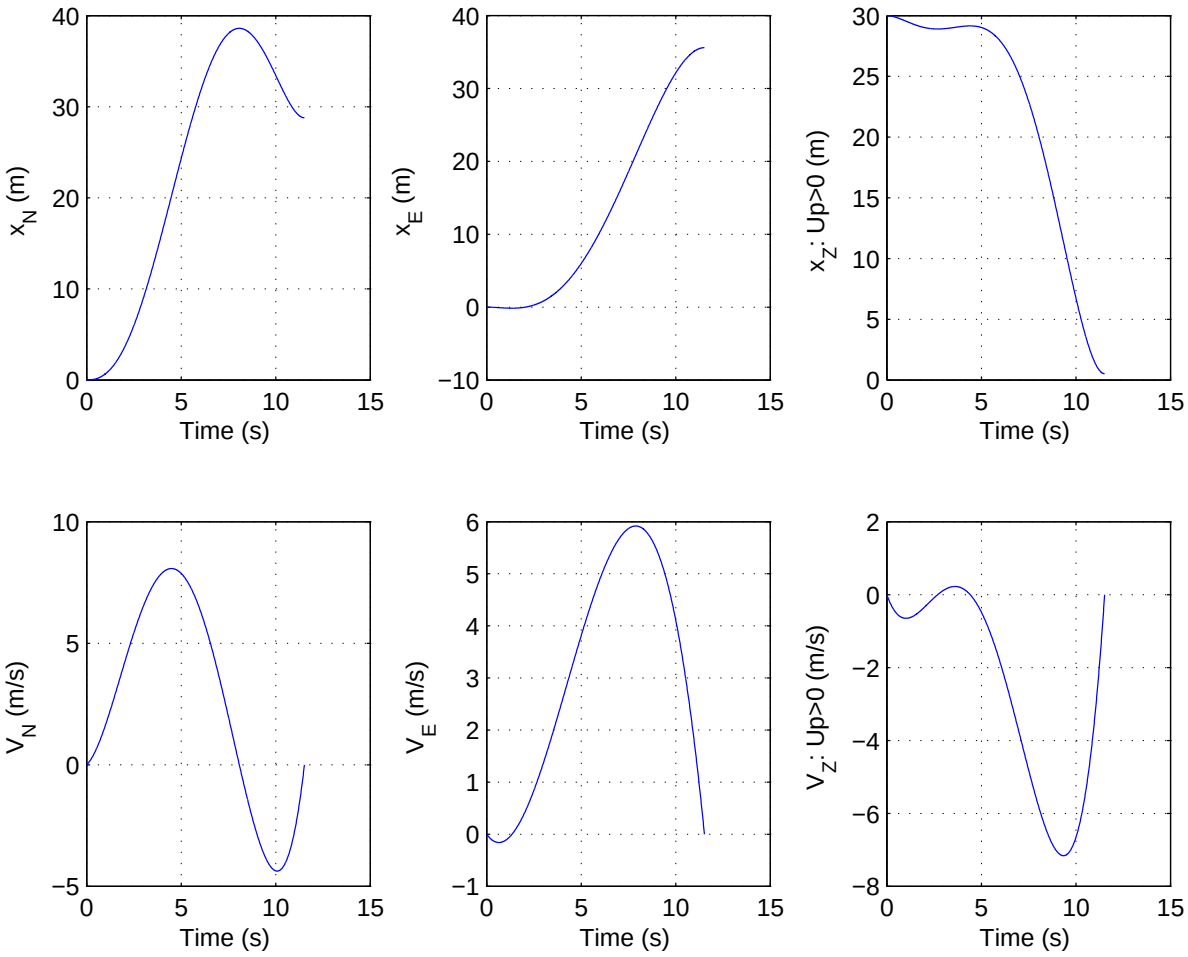


**Figure 2. Vehicle response in inertial frame**

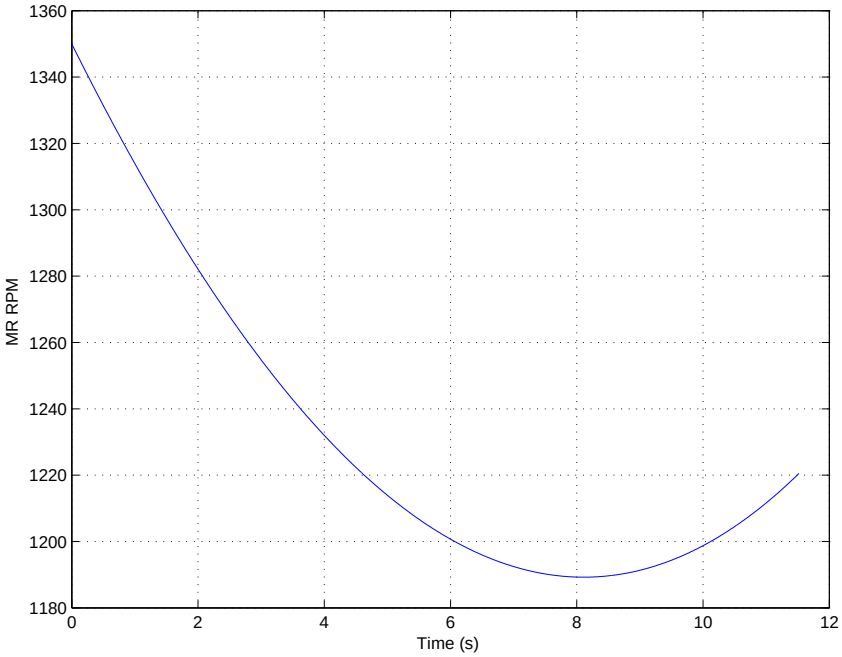


**Figure 3. MR RPM**

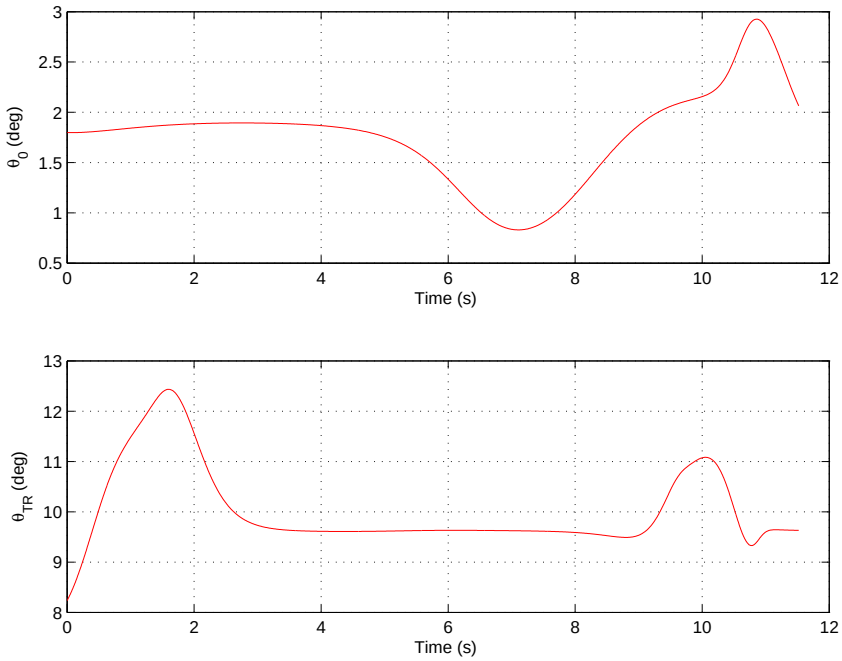


**Figure 4. MR & TR collective control inputs**

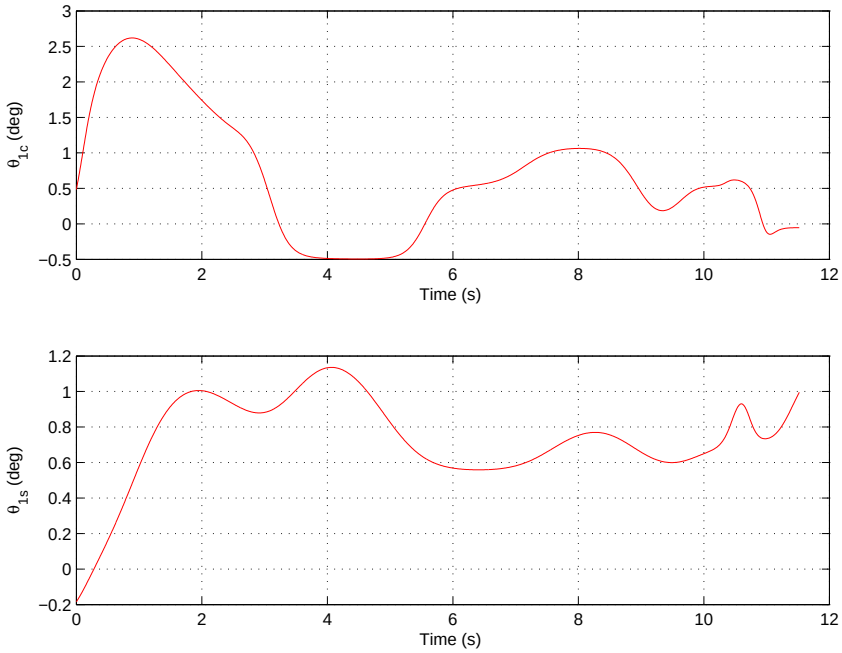


**Figure 5. MR Lat./Long. cyclic control inputs**

## References


[1]Bottasso, C. L., Croce, A., Leonello, D., and Riviello, L., "Optimization of Critical Trajectories for Rotorcraft Vehicles," *J. of the Am. Helicopter Soc.*, 2005, pp. 165–177.

[2]Latombe, J. C., *Robot Motion Planning*, Kluwer Int. Series in Engineering and Computer Science, Mass., 1991.

[3]Latombe, J. C., "Motion Planning: A Journey of Robots, Molecules, Digital Actors, and Other Artifacts," *Int. J. of Robotics Research*, Vol. 18, No. 11, 1999, pp. 11191128.

[4]LaValle, S. M., *Planning Algorithms*, Cambridge University Press, Cambridge England, 2006.

[5]Goerzen, C., Kong, Z., and Mettler, B., "A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance," *J. Intell. Robot Syst.*, Vol. 57, 2010, pp. 65–100.

[6]Dadkhah, N. and Mettler, B., "Survey of Motion Planning Literature in the Presence of Uncertainty: Considerations for UAV Guidance," *J. Intell. Robot Syst.*, Vol. 65, 2012, pp. 233–246.

[7]Fliess, M., "Generalized Controller Canonical Forms for Linear and Nonlinear Dynamics," *IEEE Trans. on Automatic Control*, Vol. 35, 1990, pp. 9941001.

[8]Fliess, M., Levine, J., Martin, P., and Rouchon, P., "Sur les systemes non lineaires differentiellement plats," *C.R. Acad. Sci. Paris*, Vol. 315, No. 1, 1992, pp. 619–624.

[9]Fliess, M., Levine, J., Martin, P., and Rouchon, P., "Flatness and Defect of Nonlinear Systems: Introductory Theory and Examples," *Int. J. of Control*, Vol. 61, No. 6, 1995, pp. 13271361.

[10]Sira-Ramirez, H., Castro-Linares, R., and Liceaga-Castro, E., "A Liouvillian Systems Approach For The Trajectory Planning-Based Control Of Helicopter Models," *Int. J. Of Robust And Nonlinear Control*, Vol. 10, 2000, pp. 301320.

[11]Martin, P., Murray, R. M., and Rouchon, P., "Flat Systems, Equivalence And Trajectory Generation," Tech. rep., CDS, California Institute of Technology, 2003.

[12]Fliess, M., Lvine, J., Martin, P., and Rouchon, P., "A Lie-Backlund Approach to Equivalence and Flatness of Nonlinear Systems," *IEEE Trans. on Automatic Control*, Vol. 44, 1999, pp. 922937.

[13]Levine, J., *Analysis and Control of Nonlinear Systems: A Flatness-Based Approach*, Springer, 2009.

[14]Sira-Ramirez, H. and Agrawal, S. K., *Differentially Flat Systems*, Marcel Dekker Inc., 2004.

[15]Lu, W. C., Duan, L., Hsiao, F. B., and Camino, F. M., "Neural Guidance Control for Aircraft Based on Differential Flatness," *AIAA J. of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 892–898.

[16]Martin, P., Devasia, S., and Paden, B., "A Different Look at Output Tracking: Control of a VTOL Aircraft," *Automatica*, Vol. 32, No. 1, 1996, pp. 101–107.

[17]Saeki, M. and Sakaue, Y., "Flight Control Design for a Nonlinear Non-Minimum Phase VTOL Aircraft via Two-Step Linearization," *IEEE Conf. on Decision and Control*, 2001.

[18]Ross, I. M. and Fahroo, F., "A Unified Computational Framework for Real-Time Optimal Control," *IEEE Conf. on Decision and Control*, 2003.

[19]Murray, R. M., Hauser, J., Jadbabaie, A., Milam, M. B., Petit, N., Dunbar, W. B., and Franz, R., *Online control customization via optimization-based control. In T. Samad and G. Balas, Editors, Software-Enabled Control: Information Technology for Dynamical Systems*, IEEE Press, 2003.

[20]Ross, I. M. and Fahroo, F., "Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems," *IEEE Trans. on Automatic Control*, Vol. 49, No. 8, 2004, pp. 1410–1413.

[21]Ross, I. M. and Fahroo, F., "Issues in the Real Time Computation of Optimal Control," *Mathematical and Computer Modelling*, Vol. 43, 2006, pp. 1172–1188.

[22]Murray, R., Rathinam, M., and Sluis, W., "Differential Flatness of Mechanical Control Systems: A Catalog of Prototype Systems," *ASME Int. Mechanical Engineering Congress*, 1995.

[23]Rathinam, M., *Differentially Flat Nonlinear Control Systems*, Ph.D. thesis, California Institute of Technology, 1997.

[24]Chetverikov, V. N., "New Flatness Conditions For Control Systems," *IFAC Nonlinear Control Systems*, 2001.

[25]Martin, P., "Aircraft Control Using Flatness," *Proc. CESA Lille*, 1996.

[26]Hauser, J. and Hindman, R., "Aggressive Flight Maneuvers," *IEEE Conf. on Decision and Control*, 1997.

[27]Cazaurang, F., Bergeon, R., and Lavigne, L., "Modelling of Longitudinal Disturbed Aircraft Model by Flatness Approach," *AIAA Guidance Navigation and Control Conf.*, 2003.

[28]Ailon, A., "Optimal Path and Tracking Control of an Autonomous VTOL Aircraft," *Mediterranean Conf. on Control and Automation*, 2006.

[29]van Nieuwstadt, M. and Murray, R. M., "Outer Flatness: Trajectory Generation For A Model Helicopter," *Europ. Control Conf.*, 1997.

[30]Koo, T. J. and Sastry, S., "Output Tracking Control Design Of A Helicopter Model Based On Approximate Linearization," *IEEE Conf. on Decision and Control*, 1998.

[31]Koo, T. J., Hoffmann, F., Sinopoli, B., and Sastry, S., "Hybrid Control of An Autonomous Helicopter," *IFAC Workshop on Motion Control*, 1998.

[32]Koo, T. J. and Sastry, S., "Differential Flatness Based Full Authority Helicopter Control Design," *IEEE Conf. on Decision and Control*, 1999.

[33]Cowling, I. D., Yakimenko, O. A., Whidborne, J. F., and Cook, A. K., "A Prototype of an Autonomous Controller for a Quadrotor UAV," *Europ. Control Conf.*, 2007.

[34]Bouktir, Y., Haddad, M., and Chettibi, T., "Trajectory Planning for a Quadrotor Helicopter," *Mediterranean Conf. on Control and Automation*, 2008.

[35]Mellinger, D. and Kumar, V., "Minimum Snap Trajectory Generation and Control for Quadrotors," *International Conf. on Robotics and Automation*, 2011.

[36]Formentin, S. and Lovera, M., "Flatness-Based Control of a Quadrotor Helicopter via Feedforward Linearization," *IEEE Conf. on Decision and Control*, 2011.

[37]Chamseddine, A., Youmin, Z., Rabbath, C. A., Join, C., and Theilliol, D., "Flatness-Based Trajectory Planning/Replanning for a Quadrotor Unmanned Aerial Vehicle," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 48, No. 4, 2012, pp. 2832–2848.

[38]Desiderio, D. and Lovera, M., "Flatness-Based Guidance For Planetary Landing," *Am. Control Conf.*, 2010.

[39]Morio, V., Cazaurang, F., and Vernis, P., "Flatness-Based Hypersonic Reentry Guidance of a Lifting-Body Vehicle," *Control Engineering Practice*, Vol. 17, No. 5, 2009, pp. 588–596.

[40]Chauvin, J., Sinegre, L., and Murray, R. M., "Nonlinear Trajectory Generation for the Caltech Multi-vehicle Wireless Testbed," *Europ. Control Conf.*, 2003.

[41]van Nieuwstadt, M. J. and Murray, R. M., "Real Time Trajectory Generation for Differentially Flat Systems," *Int. J. of Robust and Nonlinear Control*, Vol. 18, No. 11, 1998, pp. 9951020.

[42]Faiz, N., Agrawal, S. K., and Murray, R. M., "Trajectory Planning of Differentially Flat Systems with Dynamics and Inequalities," *AIAA J. of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 219227.

[43]Milam, M. B., Mushambi, K., and Murray, R. M., "A New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems," *IEEE Conf. on Decision and Control*, 2000.

[44]Milam, M. B., Franz, R., and Murray, R. M., "Real-Time Constrained Trajectory Generation Applied to a Flight Control Experiment," *IFAC World Conf.*, 2002.

[45]Taamallah, S., "Small-Scale Helicopter Blade Flap-Lag Equations of Motion For A Flybarless Pitch-Lag-Flap Main Rotor," *AIAA Modeling and Simulation Technologies Conf.*, 2011.

[46]Taamallah, S., "Flight Dynamics Modeling For A Small-Scale Flybarless Helicopter UAV," *AIAA Atmospheric Flight Mechanics Conf.*, 2011.

[47]Taamallah, S., "Low-Order Modeling For A Small-Scale Flybarless Helicopter UAV, A Grey-Box Time-Domain Approach," *AIAA Atmospheric Flight Mechanics Conf.*, 2012.

[48]Taamallah, S., "Identification Of A Nonlinear Grey-Box Helicopter UAV Model (accepted for publication)," *AIAA Atmospheric Flight Mechanics Conf.*, 2013.

[49]Taamallah, S., "Optimal Autorotation With Obstacle Avoidance For A Small-Scale Flybarless Helicopter UAV," *AIAA Guidance, Navigation and Control Conf.*, 2012.

[50]Ardema, M. D. and Rajan, N., "Separation of Time Scales in Aircraft Trajectory Optimization," *J. of Guidance, Control, and Dynamics*, Vol. 8, No. 2, 1985, pp. 275278.

[51]Thomson, D. G. and Bradley, R., "Development and Verification of an Algorithm for Helicopter Inverse Simulation," *Vertica*, Vol. 14, No. 2, 1990, pp. 185–200.

[52]Hess, R. A. and Gao, C., "A Generalized Algorithm for Inverse Simulation Applied to Helicopter Maneuvering Flight," *J. of the Am. Helicopter Soc.*, Vol. 16, No. 5, 1993, pp. 3–15.

[53]Thomson, D. G. and Bradley, R., "Inverse Simulation as a Tool for Flight Dynamics Research Principles and Applications," *Progress in Aerospace Sciences*, Vol. 42, No. 3, 2006, pp. 174–210.

[54]Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, 1989, pp. 359–366.

[55]Norgaard, M., Ravn, O., Poulsen, N., and Hansen, L., *Neural Networks for Modelling and Control of Dynamic Systems*, Springer-Verlag, London, 2000.

[56]Taamallah, S., "A Qualitative Introduction to the Vortex-Ring-State, Autorotation, and Optimal Autorotation," *Europ. Rotorcraft Forum*, 2010.

[57]Boiffier, J. L., *The Dynamics of Flight The Equations*, John Wiley & Sons, Chichester, England, 1998.

[58]Prouty, R. W., *Helicopter Performance, Stability, and Control*, Krieger Publishing Company, Malabar, Florida USA, 1995.

[59]AviationToday, *Ask Ray Prouty: Gross Weights Effect on Autorotation*, Danbury CT ., U.S.A., 2007.

[60]Carlson, E. B., *Optimal Tiltrotor Aircraft Operations During Power Failure*, Ph.D. thesis, University of Minnesota, 1999.

[61]Aponso, B. L., Bachelder, E. N., and Lee, D., "Automated Autorotation for Unmanned Rotorcraft Recovery," *AHS Int. Specialists' Meeting On Unmanned Rotorcraft*, 2005.

[62]Celi, R., "Optimization-Based Inverse Simulation of a Helicopter Slalom Maneuver," *AIAA J. of Guidance, Control, and Dynamics*, Vol. 23, No. 2, 2011, pp. 289–297.

[63]Levine, J. and Nguyen, D., "Flat Output Characterization For Linear Systems Using Polynomial Matrices," *Systems & Control Letters*, Vol. 48, 2003, pp. 6975.

[64]Henrion, D. and Lasserre, J. B., "LMIs For Constrained Polynomial Interpolation With Application In trajectory Planning," *Systems & Control Letters*, Vol. 55, 2006, pp. 473477.

[65]Piegl, L. and Tiller, W., *The NURBS Book (2nd Ed.)*, Springer, 1997.

[66]de Boor, C., *A Practical Guide to Splines*, Springer-Verlag, New York, 2001.

[67]Egerstedt, M. and Martin, C., *Control Theoretic Splines: Optimal Control, Statistics, and Path Planning*, Princeton Series in Applied Mathematics, Princeton University Press, 2010.

[68]Caigny, J. D., Demeulenaere, B., Swevers, J., and de Schutter, J., "Optimal Design Of Spline-Based Feedforward For Trajectory Tracking," *Am. Control Conf.*, 2007.

[69]Suryawan, F., Dona, J. A. D., and Seron, M. M., "Methods for Trajectory Generation in a Magnetic-Levitation System Under Constraints," *Mediterranean Conf. on Control and Automation*, 2010.

[70]Suryawan, F., Dona, J. A. D., and Seron, M. M., "On Splines And Polynomial Tools For Constrained Motion Planning," *Mediterranean Conf. on Control and Automation*, 2010.

[71]Dai, R., "B-Splines Based Optimal Control Solution," *AIAA Guidance, Navigation and Control Conf.*, 2010.

[72]Suryawan, F., Dona, J. A. D., and Seron, M. M., "Flatness-based Minimum-time Trajectory Generation for Constrained Linear Systems Using B-Splines," *18th IFAC World Congress*, 2011.

[73]MathWorks, http://www.mathworks.com/, Natick MA., U.S.A.

[74]ART, http://www.flightlab.com/, Mountain View CA., U.S.A.

[75]Johnson, W., *Helicopter Theory*, Dover Publications Inc., NY, USA, 1994.

[76]Rabiner, L. R. and Gold, B., *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs NJ, 1975.

[77]Taamallah, S., Bombois, X., and Van den Hof, P. M. J., "Optimal Control For Power-Off Landing Of A Small-Scale Helicopter A Pseudospectral Approach," *Am. Control Conf.*, 2012.

[78]Vanderbei, R. J. and Shanno, D. F., "An Interior-Point Algorithm for Nonconvex Nonlinear Programming," *Computational Optimization and Applications*, Vol. 13, 1999, pp. 231–252.

[79]Byrd, R. H., Hribar, M. E., and Nocedal, J., "An Interior Point Algorithm for Large-Scale Nonlinear Programming," *SIAM J. on Control and Optimization*, Vol. 9, No. 4, 1999, pp. 877–900.

[80]Biegler, L. T., Ghattas, O., Heinkenschloss, M., and van Bloemen Waanders, B., *Large-Scale PDE-Constrained Optimization*, Springer, Berlin, 2003.

[81]Byrd, R. H., Nocedal, J., and Waltz, R. A., "KNITRO: An Integrated Package for Nonlinear Optimization," *Large-Scale Nonlinear Optimization, Springer-Verlag*, 2006, pp. 35–59.

[82]Wachter, A., *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*, Ph.D. thesis, Carnegie-Mellon University, 2002.

[83]Wachter, A. and Biegler, L. T., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57.

[84]Biegler, L. T. and Zavala, V. M., "Large-Scale Nonlinear Programming Using IPOPT: An Integrating Framework for Enterprise-Wide Optimization," *Computers and Chemical Engineering*, Vol. 33, 2008, pp. 575–582.