Path Planning for Autonomous Vehicles by Trajectory Smoothing using Motion Primitives

Carlo L. Bottasso, Domenico Leonello, and Barbara Savini

C.L. Bottasso, L. Leonello and B. Savini are with the Department of Aerospace Engineering, Politecnico di Milano, Milano, 20156 Italy.
Path Planning for Autonomous Vehicles by Trajectory Smoothing using Motion Primitives

Abstract

We present a novel planning strategy which is applicable to high performance unmanned aerial vehicles. The proposed approach takes as input a three-dimensional sequence of way-points connected by straight flight trim conditions, and “smooths” it in an optimal way with the goal of making it compatible with the vehicle dynamics. The smoothing step is achieved by selecting appropriate sequences of alternating trims and maneuvers from within a pre-computed library of motion primitives. The resulting extremal trajectory is compatible with the vehicle and therefore trackable with small errors; furthermore, it is guaranteed to stay within the flight envelope boundary, alleviating the need for flight envelope protection systems. Yet it can be computed in real-time using closed form expressions, all non-linearities due to the vehicle model being confined to the stored library of motion primitives. The new method is demonstrated for the aggressive maneuvering of a helicopter.

Index Terms

Optimal control, motion-planning, aircraft navigation, mobile robots, real time systems.

NOMENCLATURE

\( J \)  Cost function  
\( K \)  Time element  
\( M_{ij} \)  Maneuver departing from trim \( T_i \) and arriving in trim \( T_j \)  
\( T_i \)  \( i \)th trim trajectory  
\( W_i \)  \( i \)th way-point  
\( \Delta t \)  Maneuver duration  
\( \Delta z \)  Change of altitude  
\( R(\cdot) \)  Rotation tensor whose argument is the rotation vector \( \cdot \)  
\( R_{\mathcal{E} \to \mathcal{B}} \)  Rotation tensor bringing triad \( \mathcal{E} \) into triad \( \mathcal{B} \)  
\( W_u \in \mathbb{R}^{m \times m} \)  Control input weight matrix  
\( W_y \in \mathbb{R}^{l \times l} \)  Output weight matrix  
\( m \)  Moment resultant  
\( p \)  Rotation parameters  
\( r \)  Position vector  
\( s \)  Force resultant  
\( t_r \)  Unit vector between way-point \( W_r \) and \( W_{r+1} \)  
\( u \in \mathbb{R}^m \)  Control inputs
Linear velocity vector $\boldsymbol{v}$
State vector $\boldsymbol{x} \in \mathbb{R}^n$
Outputs $\boldsymbol{y} \in \mathbb{R}^l$
Direction cosine matrix $\boldsymbol{\alpha}$
Angular velocity vector $\boldsymbol{\omega}$
Rotation vector $\boldsymbol{\psi}$
Track angle $\chi$
Distances from way-point along segments $\xi, \eta$
Time grid element size $h$
Time $t$
Inertial frame of origin $\mathcal{O}$ and triad $\mathcal{E}$
Body-attached frame of origin $\mathcal{P}$ and triad $\mathcal{B}$
Computational grid $\mathcal{T}_h$
Goal to-be-tracked quantity $\cdot^*$
Transpose $\cdot^T$
Vector components in the $\mathcal{A}$ triad $\cdot^A$
Skew-symmetric tensor associated with the vector argument $\cdot_\times$
Discretized quantity, as obtained by the use of a numerical method $\cdot_h$
Vector quantity referred to pole $\mathcal{P}$ $\cdot_\mathcal{P}$
Quantity relative to trim $\mathcal{T}_i$ $\cdot_{\mathcal{T}_i}$
Derivative with respect to time $\cdot'$

I. INTRODUCTION AND MOTIVATION

The flight control systems of unmanned aerial vehicles (UAVs) are often based on fairly simple navigation strategies, for example obtained by interpolation of way-points. It is difficult with such simplified strategies to guarantee that the trajectory will be effectively trackable by the reflexive control layer of the vehicle, and that the resulting motion will be compatible with the flight envelope limits. This way one can not in general guarantee small tracking errors; furthermore, the flight control system has to be augmented with suitable flight envelope protection systems [1], which result in additional complexity to the software of the vehicle.

A few approaches have been recently developed to generate more sophisticated trajectories [2]–[9].

Frazzoli et al. [5] proposed an approach based on motion primitives. The dynamics of the vehicle are transcribed into a finite dimensional space, whereby the vehicle can only occupy one of two possible state types: a trim or a maneuver, the latter being defined as a finite-time transition between two trims. Maneuvers can be obtained by experimental flight data or by solving an optimal control problem using a reduced vehicle model; therefore, maneuvers and trims are by design compatible with the vehicle or with a suitably faithful model of it. This means that sequences of trims and maneuvers will be trackable with small errors, and will be, up to the tracking errors,
within the vehicle flight envelope. However the problem of navigation using motion primitives is not simple to solve, especially in real-time, since one has to find the optimal sequence of alternating trims and maneuvers and the time spent in each trim, a non-linear hybrid optimal control problem for which practical, general and efficient solution methods are still lacking.

Anderson et al. [3] proposed a way-point-based trajectory smoothing methodology, using a kinematic model of the vehicle. Using this approach one can produce extremal trajectories that transition between straight-line path segments in a time-optimal fashion. In other words, the problem of planning is broken into three main steps: first, one determines a set of candidate way-points, for example by discretization of the motion space using quad-trees, triangular meshes, probabilistic road-maps, etc. [6]; next, a path is constructed by connecting a subset of the way-points, for example using the $A^*$ search algorithm; finally, the straight line paths are “smoothed” using a kinematic model of the vehicle. This divide-and-conquer approach greatly simplifies the planning problem; however, since only an extremely simple kinematic model of the vehicle is used for the smoothing step, here again there can be no assurance in general that the resulting path will be trackable with small error by the vehicle, nor that the ensuing motion will be confined within the vehicle flight envelope.

In this work, we describe a computational procedure for the navigation problem of UAVs which is a combination of the two previous approaches. The new proposed procedure has general applicability, for example to both fixed and rotary wing vehicles, but it is demonstrated here for the sole latter case.

In particular, as in Ref. [3] we first compute an optimal sequence of way-points in 3-D space connected by straight flight trim conditions, which is then smoothed. However, the smoothing step is here accomplished using a model of the vehicle based on motion primitives, i.e. a highly faithful non-linear vehicle model rather than a simple kinematic model. This critical smoothing step ensures the compatibility of the resulting motion plan with the vehicle dynamics, and hence we term it “trajectory compatibilization”. The proposed compatibilization strategies can be constructed so as to be optimal, by minimizing some cost. Specifically, here we consider compatibilizations which require the least number of maneuvers and which minimize either the time required to perform the transition or some measure of the deviation of the path from the way-points, although other strategies are also possible.

Trims and maneuvers used in this work are generated off-line using a high-fidelity non-linear model of the aircraft. Maneuvers are formulated as optimal control problems and computed using a direct transcription approach (see Ref. [10]–[12]). The resulting describing quantities are stored in a suitable data structure and used in real-time for the trajectory compatibilization.

One of the key highlights of this new methodology is that, although it is based on sophisticated non-linear models of the vehicle which help ensure small tracking errors and partial flight envelope protection, still the solution of the planning problem can be performed in closed form. This means that the procedure implies, contrary to most non-linear approaches, a fixed number of operations, and hence one can guarantee a hard-real time schedule. This key property is due to the fact that model non-linearities are by design relegated to the motion primitive library, and hence hidden to the planner, while the trajectory compatibilization is a purely kinematic problem which can be solved analytically.
The paper is organized as follows. First, we describe the motion primitives and their numerical determination in Section II. In particular, §II-A introduces the necessary background information on the vehicle model and the notation used throughout this work. Next, §II-B discusses the trim problem. Although the problem of computing trimmed flight conditions is well known in the aerospace literature, here we give a complete kinematic characterization of trim trajectories for an arbitrary non-linear model of the vehicle which, to our knowledge, is not offered elsewhere. Maneuvers are discussed in §II-C; first we formulate maneuvers in a clear mathematical sense as solutions of optimal control problems in §II-C1, and then we briefly describe an efficient procedure for their numerical computation in §II-C2. The section is concluded by the quantization of the system dynamics using motion primitives, discussed in §II-D. Section III describes the problem of path planning by smoothing for the quantized reduced system. More specifically, we identify three necessary trajectory compatibilization primitives, the first involving a turn at a way-point, discussed in §III-A, and the other two addressing the problem of climb and descent, treated in §III-B. Numerical examples which demonstrate the proposed procedures are presented in IV, while conclusions are discussed in Section V.

II. MOTION PRIMITIVES

A. Vehicle Model and Notation

We consider an inertial frame of reference $\mathcal{F}_{O,E}$, centered at point $O$ and denoted by a triad of unit vectors $E = (e_1, e_2, e_3)$, $e_1$ pointing North, $e_2$ pointing East and $e_3$ pointing down (NED navigational system). A body-attached local frame of reference $\mathcal{F}_{P,B}$ has origin in the generic material point $P$ of the vehicle and has a triad of unit vectors $B = (b_1, b_2, b_3)$.

The equations of balance of linear and angular momentum (Euler’s equations) of the vehicle expressed in terms of body-attached components can be written as

\begin{align}
\dot{l}^B &+ \omega^B \times l^B - s^B = 0, \\
\dot{h}_P^B &+ v_P^B \times l^B + \omega^B \times h_P^B - m_P^B = 0,
\end{align}

where the linear momentum is $l = mv_P + S_P^T \omega$ and the angular momentum is given by $h_P = S_P v_P + J_P \omega$. Letting $\rho v$ be the vehicle density, $m = \int_V \rho v \, dV$ is the mass, $S_P = \int_V \rho v r \, dV$ is the first moment of inertia, and $J_P = -\int_V \rho v r x r_y \, dV$ is the inertia dyadic.

The symbol $(\cdot)$ denotes a derivative with respect to time $t$.

Here and in the following, the notation $(\cdot)^A$ denotes components in the generic $A$ triad. If $R_{E \rightarrow B}$ is the rotation tensor that brings triad $E$ into triad $B$, then the components of a generic vector $a$ in the two triads are related as $a^E = \alpha a^B$, $a^B = \alpha^T a^E$, where $\alpha$ is the direction cosine matrix, $\alpha = R_{E \rightarrow B} = R_{E \rightarrow B}^T$. Furthermore, we use the notation $a_x$ to indicate the skew-symmetric tensor associated with $a$, and $(\cdot)^T$ to indicate the transpose operation.

We denote with $r_P = (P - O)$ the position vector of the vehicle reference point $P$ with respect to the origin of the inertial frame $O$, and with $v_P$ its linear velocity vector. The time rates of change of the position vector
components are related to the components of the linear velocity vector as

\[ \dot{r}_P^E = v_P^E = \alpha v_P^B. \]  

(2)

Indicating with \( \omega \) the angular velocity vector of triad \( B \) with respect to triad \( E \), Poisson’s equations express the time rates of change of the direction cosine matrix as

\[ \dot{\alpha} = \omega_E \times \alpha = \alpha \omega^B_x. \]  

(3)

The force resultant \( s \) acting on the vehicle can be written as

\[ s = s_{\text{grav}} + s_{\text{aero}}, \]  

(4)

where \( s_{\text{grav}} = mg e_3 \) is the gravitational force, and \( s_{\text{aero}} \) is the resultant of the aerodynamic and propulsive forces.

The moment resultant \( m_P \) can be written as

\[ m_P = r_{PG} \times s_{\text{grav}} + r_{PA} \times s_{\text{aero}} + m_{A,\text{aero}}, \]  

(5)

where the aerodynamic/propulsive forces have resultant moment \( m_{A,\text{aero}} \) about the reference point \( A \), \( G \) is the center of gravity, and \( r_{PG} \), \( r_{PA} \) are the distance vectors between point \( P \) and points \( G \) and \( A \), respectively.

To simplify the notation, in the following we drop the subscript \((\cdot)_P\) which indicates the reference pole on the vehicle, specifically from the linear velocity, the position vector, the angular momentum and the moment resultant.

The functional form of the aerodynamic/propulsive forces and moments at the generic time instant \( t \) is as follows:

\[ s_{\text{aero}}^B(t) = s_{\text{aero}}^B(w^B(\tau), u(\tau), \pi(\tau)), \]  

(6a)

\[ m_{A,\text{aero}}^B(t) = m_{A,\text{aero}}^B(w^B(\tau), u(\tau), \pi(\tau)), \]  

(6b)

with \( 0 \leq \tau \leq t \). The expressions above indicate that the body-attached components of the aerodynamic and propulsive forces and moments depend on the body-attached components of the linear and angular velocity vectors (where we have introduced the generalized velocities \( w = (v, \omega) \) in the interest of brevity), to account for the generation of such forces and moments by the relative wind impinging on the body.

The generalized forces depend also on the control inputs \( u \), whose nature depends on the specific vehicle (fixed wing, rotary wing) and its configuration, and on a number of physical parameters \( \pi \), which typically include the air density \( \rho \), the Mach (\( M \)) and Reynolds (\( Re \)) numbers, etc.

Expressions (6) include also the memory effects of aerodynamics, by specifying that the instantaneous values of the generalized forces at time \( t \) depend on the current value and on the previous history \((0 \leq \tau \leq t)\) of generalized velocities, control inputs and parameters. These delay effects are typically approximated by a truncated expansion in Taylor series, which can be written

\[ s_{\text{aero}}^B(t) \approx s_{\text{aero}}^B(w^B(t), \dot{w}^B(t), u(t), \pi(t)), \]  

(7a)

\[ m_{A,\text{aero}}^B(t) \approx m_{A,\text{aero}}^B(w^B(t), \dot{w}^B(t), u(t), \pi(t)). \]  

(7b)
It is readily verified that (1) are unaffected by translations in the $e_1$-$e_2$ plane and by rotations about $e_3$ (while translations along $e_3$ will imply a change of air properties, and hence of the parameters $\pi$, although this change is slow and can be neglected for small changes of altitude). This means that all solutions of such equations, which either belong to the class of trim trajectories (discussed in \S II-B) or to the class of maneuvers (treated in \S II-C), can be rigidly translated so as to originate from an arbitrary point $A$ and rotated about the local vertical so as to depart from $A$ with an arbitrary track angle $\chi$, where

$$\chi = \cos^{-1}\left(\frac{v_{12}}{|v_{12}|} \cdot e_1\right),$$

$v_{12}$ being the projection of the velocity vector $v$ at $A$ onto the $e_1$-$e_2$ plane, i.e.

$$v_{12} = (I - e_3 \otimes e_3)v.$$  

The translational and rotational invariance of the motion primitives will be the key property used in the following for assembling alternating sequences of trim trajectories and maneuvers into a complete flight path.

**B. Trim Trajectories**

It has been argued that all human-built vehicles are designed such they possess relative equilibria [13], in order to reduce the pilot workload. Flying vehicles do in fact possess relative equilibria, which are termed in the aerospace literature as *trim trajectories*. Along trim trajectories, the body-attached components of the linear and angular velocities as well as the control inputs are constant-in-time:

$$v^B(t) = \text{const.} = v^B_T,$$

$$\omega^B(t) = \text{const.} = \omega^B_T,$$

$$u(t) = \text{const.} = u_T,$$

$\forall t \in [0, \infty]$, i.e. a relative equilibrium can be held for an indefinite duration. In practice however $t$ can not be too large, since the change of mass and the movement of the center of gravity due to the consumption of fuel makes most flying vehicles non-time-invariant systems, for which relative equilibria do not strictly exist. In the above expressions, the notation $(\cdot)_T$ indicates constant quantities relative to the generic trim condition $T$.

1) Kinematic Characterization of Trim Trajectories: Using (10), it is possible to characterize the kinematic nature of trim trajectories.

To this end, let us consider a trimmed vehicle whose orientation at time $t = 0$ is given by the direction cosine matrix $\alpha_T$. The orientation of the vehicle at all successive instants of time can be obtained by integrating Poisson’s equations (3) subjected to the initial conditions $\alpha_T$:

$$\dot{\alpha} = \alpha\omega^B_{T_x},$$

$$\alpha(0) = \alpha_T.$$
Since the components $\omega_B^T$ are constant by definition of a trim trajectory, (11) represent an initial value problem for a system of constant coefficient linear ordinary differential equations, whose solution is readily found as

$$\alpha(t) = \alpha_T R(\omega_T^R t) = R(\omega_T^R t)\alpha_T, \quad (12)$$

where

$$R(\psi) = \exp^{\psi_x} = \sum_{k=0}^{\infty} \frac{\psi_x^k}{k!} = I + \frac{\sin \varphi}{\varphi} \psi_x + \frac{1 - \cos \varphi}{\varphi^2} (\psi_x)^2, \quad (13)$$

$R(\psi)$ being the rotation tensor corresponding to a rotation vector $\psi = \varphi \mathbf{k}$ of magnitude $\varphi$ and unit axis $k$ (see for example Borri et al. [14] and references therein for a more detailed treatment on the problem of parameterizing rotations).

Consider now the translational equilibrium of the vehicle, (1a), which on a trim trajectory become

$$\omega_B^T \times l_B^T - mg e_3^B - s_{\text{aero}}^B = 0. \quad (14)$$

The first (gyroscopic) term in the above equation is constant by definition of trim trajectory, i.e. $\omega_B^T \times l_B^T = \text{const}$. The third (aerodynamic and propulsive) force term is also constant since at trim, using (7a), we have

$$s_{\text{aero}}^B = s_{\text{aero}}^B(\omega_T^B, 0, u_T, \rho, M, Re) = \text{const}. \quad (15)$$

Hence the second term in (14) must also be constant-in-time, i.e. $e_3^B = \text{const}$. Consequently, using (12) we find

$$e_3^B = \alpha_T R^T(\omega_T^R t)e_3^T = \text{const}. \quad (16)$$

which implies that the vehicle angular velocity in a trim trajectory must be parallel to the local vertical, $\omega \parallel e_3$.

This, together with condition (10b), allows one to write the angular velocity vector as

$$\omega = \omega_T e_3, \quad (17)$$

where $\omega_T$ is the constant turn rate.

Next, consider the vehicle linear velocity, whose components obey the following relationship:

$$v^T(t) = \alpha(t)v_T^B. \quad (18)$$

Using (12) we find

$$v^T(t) = R(\omega_T e_3 t)v_T^T, \quad (19)$$

where $v_T^T = \alpha_T v_T^B$ are the components measured in the inertial frame of the linear velocity vector at time $t = 0$. The previous equation states that along a trim trajectory the linear velocity vector rotates about the local vertical at the constant turn rate $\omega_T$. 

Finally, the vehicle trajectory can be computed by integrating the kinematic equations (2). If \( r(0) \) is the position vector of the vehicle at time \( t = 0 \), its location at the generic time instant \( t \) is obtained as

\[
\begin{align*}
r^E(t) &= r^E(0) + \left( \int_0^t \alpha(\tau) \, d\tau \right) v^B_T, \\
&= r^E(0) + \left( \int_0^t R(\omega_T e_3 \tau) \, d\tau \right) \alpha_T v^B_T, \\
&= r^E(0) + S(\omega_T e_3 t) v^B_T,
\end{align*}
\]

where

\[
S(\psi) = \sum_{k=0}^{\infty} \frac{(\psi \times)^k}{(k+1)!} = I + \frac{1 - \cos \varphi}{\varphi^2} \psi \times + \frac{1}{\varphi^2} \left( 1 - \sin \varphi \right) (\psi \times)^2
\]

is the tensor which relates the angular velocity and the rotation vector and its rates, i.e. \( \omega = S(\psi) \dot{\psi} \). In deriving (20c), we have made use of the fact that for the generic vector \( a \) we have [14]

\[
S(ta) = \frac{1}{t} \int_0^t R(\tau a) \, d\tau.
\]

Expression (20c) for \( r(t) \) represents a constant pitch helicoid with axis parallel to \( e_3 \).

This completes the kinematic characterization of trim trajectories.

2) Numerical Solution of the Trim Problem: Based on the above discussion, it is apparent that in order to fully determine a trim trajectory one needs to compute the 6 unknown components of the linear and angular velocity vectors, \( v^B_T \) and \( \omega^B_T \), and the unknown attitude of the vehicle at time \( t = 0 \) as described by the direction cosine matrix \( \alpha_T \). The direction cosines of the vehicle can be expressed in terms of a suitable set of rotational parameters \( p_T \), i.e.

\[
\alpha_T = \alpha_T(p_T),
\]

for example the commonly used Euler angles \( \psi, \theta, \phi \) in the 3-2-1 sequence. Irrespectively of the specific choice of rotational parameters, the independent parameters must be three in number because of Euler’s Rotation Theorem. Recall however that the heading of the vehicle at \( t = 0 \) is arbitrary, for the previously discussed invariance of all solutions of the governing equations to rotations about \( e_3 \).

A part from the above 9 unknown kinematic parameters, the \( m \) values of the controls \( u_T \) (typically being \( m = 4 \) for both fixed wing aircrafts and helicopters) are also unknown and need to be computed. This brings the total number of unknowns for the trim problem to 13.

The numerical solution of the trim problem can be formulated in several different ways. For example, in the present work we consider the following non-linear system of algebraic equations:

\[
\begin{align*}
\omega^B_T \times l^B_T - mg \alpha_T^T e_3^e - s^B_{aero} &= 0, \\
v^B_T \times l^B_T + \omega^B_T \times h^B_T - r^B_{PG} \times m g \alpha_T^T e_3^e - r^B_{PA} \times s^B_{aero} - m^B_{A,aero} &= 0, \\
\alpha_T v^B_T &= v^B_T, \\
\alpha_T \omega^B_T &= \omega_T e_3^e
\end{align*}
\]
\[ p(v_B^T, \omega_B^T, p_T, u_T) = 0. \] (24e)

The first two vector equations, \((24a,24b)\), are Euler’s translational and rotational equilibrium equations at trim.

The third vector equation, \((24c)\), assigns the desired linear velocity of the vehicle by specifying its components in the inertial frame, \(v_E^T\). Notice that by doing so we remove the arbitrariness in the vehicle track; in this work we conventionally always choose \(v_E^T\) such that \(\chi = 0\), where \(\chi\) is given in (8), so that the vehicle is initially flying North. The reason for such a choice will become clear in the following.

The fourth vector equation, \((24d)\), assigns the desired value of the turn rate, \(\omega_T\).

Finally, the fifth scalar equations, \((24e)\), brings the number of equations in system \((24)\) to 13, matching the number of unknowns. This equation expresses the so called “piloting condition”, which in general is some given function of the problem unknowns. A typical example is the requirement to fly with no side-slip, which would be expressed with the equation \(v_T \cdot b_2 = 0\).

There are clearly other ways to formulate the problem. For example, one could use a two-angle parameterization which explicitly accounts for the fact that \(\psi\) is arbitrary, and specify the vehicle speed \(V = |v|\) and climb/descent rate \(V_z = v \cdot e_3\) instead of the three inertial components provided in \((24c)\). This would have the effect of eliminating one unknown and one equation. Clearly these two formulations are perfectly equivalent.

Notice that system \((24)\) does not contain any information about the flight envelope and performance boundaries of the vehicle. Hence, after solving the problem, one must verify that the computed trim trajectory is feasible for the vehicle, by checking that the resulting thrust or power is indeed available, that the load factor is within the admissible limits, etc.

An alternative way to solve the trim problem is to formulate it as a constrained optimization problem, which writes

\[
\min_{v_B^T, \omega_B^T, p_T, u_T} J^{\text{trim}}(v_B^T, \omega_B^T, p_T, u_T),
\]

s.t.:

\[
\omega_B^T \times l_B^T - mg\alpha_T e_3 - s_{aero}^B = 0,
\]

\[
v_B^T \times l_B^T + \omega_B^T \times h_B^T - r_{PG}^B \times m\alpha_T e_3 - r_{PA}^B \times s_{aero}^B - m_{A,aero}^B = 0,
\]

\[
\alpha_T v_B^T = v_E^T,
\]

\[
\alpha_T \omega_B^T = \omega_T e_3,
\]

\[
g(v_B^T, \omega_B^T, p_T, u_T) \leq 0,
\]

where \(J^{\text{trim}}\) is the cost function which one seeks to minimize, a possible example being the thrust, while \((25b–25e)\) are the same as in the square trim problem \((24)\) and enforce the equilibrium and the desired linear and angular velocity of the vehicle. Finally, \((25f)\) represents all those inequality constraints which are needed in order to express the operational and performance limits of the vehicle.
C. Maneuvers

Maneuvers have been previously defined as finite-time transitions between two trims. Here we consider the mathematical formulation of the maneuver problem, and its numerical solution.

1) Mathematical Formulation of the Maneuver Problem: It is clear that, given a start trim and an arrival trim, there is an infinite number of ways to transition between the two. A way to remove this arbitrariness is to formulate a maneuver as an optimal control problem [11], [12], where one minimizes a cost (time, fuel consumption, etc.) which in general is some given function of the vehicle states and control inputs. The solution of the optimization problem must satisfy the dynamic and kinematic equations of the vehicle, the initial and final conditions corresponding to the start and arrival trims, and all other equality and inequality constraints which need to be met in order to satisfy given performance and procedural requirements.

More precisely, consider a starting trim \( T_i \), kinematically characterized by the body components of the linear velocity vector \( v_{Ti}^B \), the turn rate \( \omega_{Ti} \), and the rotational parameters \( p_{Ti} \), under the previously recalled assumption that the vehicle is flying North (null track angle) at time \( t = 0 \), which is the initial instant of the maneuver.

Furthermore, consider an arrival trim \( T_j \), characterized by the kinematic quantities \( v_{Tj}^B, \omega_{Tj} \) and \( p_{Tj} \). The arrival trim is reached at the completion of the maneuver at the unknown time \( \Delta t \).

The maneuver \( M_{ij} \) originating in \( T_i \) and arriving in \( T_j \) is the solution of the following optimal control problem:

\[
\begin{align*}
\min_{v^B(t), \omega^B(t), p(t), u(t)} & \quad J_{\text{maneuver}} = \Phi(v^B, \omega^B, p, r^E)_{\Delta t} + \int_0^{\Delta t} L(v^B, \omega^B, p, r^E, u, \dot{u}) \, dt, \\
\text{s.t.:} & \quad l^B + \omega^B \times l^B - mg\alpha T e^E_3 - s^B_{\text{aero}} = 0, \\
& \quad \dot{h}^B + v^B \times l^B + \omega^B \times h^B - r^B_{PC} \times mg\alpha T e^E_3 - r^B_{PA} \times s^B_{\text{aero}} - m^B_{A, \text{aero}} = 0, \\
& \quad r^E = \alpha v^B, \\
& \quad \dot{p} = S_{\alpha^{-1}} \omega^B, \\
& \quad g(v^B, \omega^B, p, r^E, u, \dot{u}, \Delta t) \leq 0, \\
& \quad v^B(0) = v_{Tj}^B, \\
& \quad \omega^B(0) = \alpha_T \omega_{Tj} e^E_3, \\
& \quad p(0) = p_{Tj}, \\
& \quad r^E(0) = 0, \\
& \quad \dot{u}(0) = 0, \\
& \quad v^B(\Delta t) = v_{Tj}^B, \\
& \quad \omega^E(\Delta t) = \alpha_T(\Delta t) \omega_{Tj} e^E_3, \\
& \quad \theta(\Delta t) = \theta_{Tj}, \\
& \quad \phi(\Delta t) = \phi_{Tj},
\end{align*}
\]
\[ \dot{u}(\Delta t) = 0. \] (26p)

The cost function \( J_{\text{maneuver}} \) has in general both a boundary term \( \Phi \) and an integral term \( L \). The latter includes also the control inputs \( u \) and often their temporal rates \( \dot{u} \), in order to penalize an excessive control activity.

An optimal solution is sought such that a number of equality and inequality constraints are satisfied, as detailed in the following.

In particular, (26b,26c) are the equations of dynamic equilibrium of the vehicle, while (26d,26e) represent the linear and angular kinematic conditions, \( S^p \) being the matrix operator corresponding to the set of rotation parameters \( p \) which express the body components of the angular velocity in terms of the rotation parameters and their rates.

All maneuver-defining, envelope-protection constraints are grouped in (26f).

The initial conditions of departure from trim \( T_i \) are expressed by (26g–26k). More in particular, notice that with (26i) we fix the initial orientation of the vehicle to the one of the departing trim, i.e. the vehicle begins the maneuver heading North. Furthermore, we conventionally fix the initial position at the origin of the inertial frame in (26j). Finally, although mathematically the controls should carry no boundary conditions since they appear as algebraic variables in the equations of motion of the vehicle, we include (26k) to avoid the appearance of jumps in the control time histories between the initial trim control values and the maneuver controls at time \( t = 0^+ \).

The final boundary conditions of arrival to trim \( T_j \) are expressed by (26l–26p). Notice that (26n) and (26o) specify the pitch and roll of the vehicle in the arrival trim. The heading, or alternatively the final track angle, can be constrained or let free; this point will be raised again and made clearer later on. Finally, (26p) avoids the appearance of jumps in the controls between the end of the maneuver at time \( t = \Delta t \) and the arrival trim.

The solution of problem (26) yields the maneuver duration \( \Delta t \), and the time history of the vehicle states \( v^B(t), \omega^B(t), p(t), r^E(t) \) and control inputs \( u(t) \forall t \in [0, \Delta t] \), which minimize \( J_{\text{maneuver}} \) and satisfy all constraints.

Within maneuver \( M_{ij} \), the vehicle undergoes an incremental displacement \( \Delta r_{ij} \), so that its final position entering trim \( T_j \) will be

\[ r(\Delta t) = r(0) + \Delta r_{ij}. \] (27)

Furthermore, the vehicle attitude will undergo an incremental rotation \( \Delta R_{ij} \) that brings the direction cosine matrix \( \alpha_{Ti} \) at the departing trim \( T_i \) into the arrival direction cosine matrix \( \alpha(\Delta t) \), i.e.

\[ \alpha(\Delta t) = \Delta R_{ij} \alpha_{Ti}. \] (28)

Notice that in general the arrival attitude \( \alpha(\Delta t) \) is not equal to \( \alpha_{T_j} \), since with the latter symbol we express the attitude at trim \( T_j \) when the vehicle is flying North (see §II-B2), and the two differ by the change of track \( \Delta \chi_{ij} \) during the maneuver, i.e.

\[ \alpha(\Delta t) = R(\Delta \chi_{ij} e_3) \alpha_{T_j}. \] (29)

Letting \( v^E(\Delta t) = \alpha(\Delta t)v^E_{ij} \) be the inertial velocity components at the end of the maneuver, the change of track within the maneuver is computed as

\[ \Delta \chi_{ij} = \cos^{-1} \left( \frac{v_{12}(\Delta t)}{|v_{12}(\Delta t)|} \cdot e_1 \right), \] (30)
where $v_{12}(\Delta t) = (I - e_3 \otimes e_3)v(\Delta t)$. If the change of track is let free in problem (26), then its value $\Delta x_{ij}$ can be computed a posteriori once the problem has been solved using (30). Since maneuvers with changes of track different from zero can cause some complications, as explained below, one can include the request $\Delta x_{ij} = 0$ among the constraints of problem (26).

2) Numerical Solution of the Maneuver Optimal Control Problem: There are several efficient numerical methods for the solution of the maneuver optimal control problem (26) [15]. In the present work, we use the direct transcription approach described in detail in Refs. [10]–[12]. The vehicle governing equations (26b–26e) are discretized on a computational grid of the temporal domain using an appropriate numerical method. This defines a set of discrete unknown state and control parameters on the computational grid. Next, the constraint conditions and the problem cost function are expressed in terms of the discrete parameters. This in turn defines a non-linear discrete parameter optimization problem, whose numerical solution approximates the solution of its infinite-dimensional counterpart (26).

To make things more precise, we partition the temporal domain as

$$0 \equiv t_0 < t_1 < \ldots < t_{N-1} < t_N \equiv \Delta t,$$

where $N \geq 1$, and $t_{i+1} = t_i + h_i$, $i = 0, \ldots, N - 1$. A grid $T_h$ is associated with this partition. $T_h$ is made of $N$ elements, the generic element being labelled $K$. Each element spans a time interval of size $h_i(\Delta t)$ which is a function of the unknown maneuver duration $\Delta t$, and has a left vertex $\partial K^L$ associated with time $t_i$ and a right vertex $\partial K^R$ associated with time $t_{i+1}$.

Let us define for the sake of a more compact notation the vehicle state vector $x = (v^B_T, \omega^B_T, p^T, r^E_T)^T$. Given a numerical integration scheme, we indicate with the symbols $x_h, u_h$ the finite dimensional approximations that the numerical scheme makes of the infinite dimensional unknown fields $x(t), u(t)$. On the computational grid $T_h$, $x_h$ and $u_h$ can be regarded as functions of some discrete parameters $x_d, u_d$, i.e. $x_h = x_h(x_d)$ and $u_h = u_h(u_d)$ on $T_h$. These discrete parameters depend on the specific numerical integration scheme and will contain the state unknowns at the grid nodes and possibly additional internal stages. Similarly for the vector of discrete control parameters $u_d$, according to the numerical method. In this work, we use the finite element method described in Ref. [10].

The restriction of the approximations $x_h$ and $u_h$ to the generic grid element $K$ is written $x_h|_K$ and $u_h|_K$, respectively. The state approximations evaluated on the right vertex of an element $K_i$ are equal to the state approximations evaluated on the left vertex of the neighboring element $K_{i+1}$, i.e.

$$x_h|_{\partial K^R_i} = x_h|_{\partial K^L_{i+1}}.$$  

(32)

This condition expresses the continuity of the states at the element interfaces, in the sense that it provides the initial conditions on each element as the value of the final states on the preceding element. In general, there is not a similar condition on the controls, since there are no initial conditions on this field, so that the control approximations $u_h$ should be discontinuous across element interfaces.
The discretized version of the maneuver optimal control problem can at this point be written as

\[
\min_{x_d, u_d} J_{\text{maneuver}}^h = \Phi_h(x_h)\big|_{0}^{\Delta t} + \int_0^{\Delta t} L_h(x_h, u_h) \, dt,
\]  

(33a)

s.t.:

\[
\begin{align*}
& f_h(x_h|K, u_h|K, \Delta t) = 0 \quad \forall K \in T_h, \\
& g_h(x_h|K, u_h|K, \Delta t) \leq 0 \quad \forall K \in T_h, \\
& \psi(x_h(0), u_h(0)) = \psi_0, \\
& \psi(x_h(\Delta t), u_h(\Delta t)) = \psi_{\Delta t}. \\
\end{align*}
\]

(33b)–(33e)

In problem (33), the cost \(J_{\text{maneuver}}^h\) represents a discretized version of the cost \(J_{\text{maneuver}}\) of problem (26), the integral term being evaluated through some appropriate quadrature rule. Furthermore, the first set of constraint conditions (33b) represents a discretized version of the vehicle governing equations, (26b–26e), written on each element \(K\) of the computational grid. These equations are coupled together through the gluing conditions (32) that induce a banded sparsity pattern to the problem. The second set of constraints (33c) represents a discretized version of the maneuver-defining constraints (26f), again expressed on each grid element \(K\). Finally, (33d) and (33e) represent the initial and final conditions, respectively. Discrete cost and constraints are all functions of the vectors of discrete state and control parameters \(x_d, u_d\), which represent the unknowns of the optimization problem.

Problem (33) is a non-linear programming problem which is solved in this work using the sequential quadratic programming method [16].

D. Quantization of System Dynamics using Motion Primitives

The dynamics of a vehicle can be quantized by selecting a suitable finite number of motion primitives from within the two classes of trim trajectories and maneuvers described above [13]. In other words, instead of introducing a discretization of states and controls on a temporal grid using a numerical method, as done for example in the direct transcription approach of §II-C2, the system dynamics are restricted to the switching between a finite number of motion primitives stored in a motion library, always alternating between trims and maneuvers.

This quantization implies an approximation: not all trims and maneuvers which are possible solutions of the vehicle equations of motion can be realized by the quantized dynamical system. In fact, the quantized system will only be able to realize those motion primitives which are stored in the motion library. In this sense the computed solution will be sub-optimal. Nonetheless, it should be noticed that the quantized system is a very powerful approximation of reality, in that it satisfies \textit{by design} the vehicle dynamics and its envelope boundaries (clearly, up to the fidelity of the model to the plant). Furthermore, depending on the available hardware, it is possible in practice to use quite dense quantizations; in fact, the data to be stored for each trim is of extremely small size, while the one for each maneuver is also quite compact and can be further compressed in case of need, for example using smooth interpolations based on splines. Finally, it should be remarked that even non-autonomous vehicles are often operated at discrete trim values for a number of reasons, and the ability to be able to fly at \textit{each} possible point within the flight envelope does not seem to be particularly useful in practice.
In this work, the motion library is computed off-line solving problems (24) and (33) numerically for a suitable number of trim trajectories and connecting maneuvers. The resulting kinematic quantities are stored in a data structure, and retrieved on-line by the path smoother described in Section III.

We remark that by transcribing the system dynamics into a motion library, all difficulties related to the solution of non-convex optimization problems, model non-linearities, the satisfaction of constraints, etc. are relegated to the off-line phase, where the use of iterative and computing intensive algorithms does not pose insurmountable problems.

III. PATH PLANNING BY SMOOTHING

Assume that a path has been defined by an appropriate planner as a sequence of way-points connected by straight flight segments, the constant flight speed on each connecting segment being also known and belonging to the straight flight trim trajectories stored in the motion library. The smoothing procedure described in the following has the goal of compatibilizing the given path with the vehicle dynamics. This is achieved by finding an optimal switching sequence in the motion library such that the quantized system traces an extremal trajectory which follows closely the given straight segment path. The switching sequences considered in the following are optimal in the sense that they use the minimum possible number of maneuvers, and they minimize either the total transition time or some measure of the deviation from the assigned way-points.

Clearly, in order for the divide-and-conquer procedure (way-point planning followed by compatibilization) to yield consistent approximations of the solution of a (possibly constrained) optimal planning problem, the criteria for solving the way-point planning and the compatibilization problem should match. To make this point clearer, assume for example that we need to plan a minimum time trajectory, which satisfies path constraints (e.g. avoidance of obstacles), together with flight envelope protection constraints. Then, one would start by solving a minimum time way-point planning problem which accounts for the obstacle avoidance constraints. At this level, it is important to notice that the successive smoothing step implies that the vehicle will deviate from the resulting way-point connecting straight segments. The deviations can however be computed up-front on the knowledge of the motion primitives (see below for details), so that one can account for this effect while solving the way-point tracking problem, effectively ensuring the avoidance of all obstacles for the smoothed trajectory. Next, given the time-optimal path-constraint-satisfying way-point-based trajectory, one smooths it using the proposed trajectory compatibilization procedure, using time-optimal transitions. These transitions, by design, satisfy the flight envelope protection constraints. Therefore, by this procedure one obtains a flight path that satisfies the path and envelope constraints and is time-(sub)optimal, in the sense described above. Notice that in this conceptual example, time (or other, depending on need) optimal criteria have been used for solving both the way-point planning and the subsequent smoothing steps, and that obstacle constraints have been treated at the level of way-point planning (with buffers which account for deviations from straight segments), while the flight envelope constraints have been included in the compatibilization. This way one has approximated the solution of a non-linear constrained planning problem by decomposing it into two manageable and real-time capable sub-problems.
Notice on the other hand, that the matching of optimization criteria at the different levels is not strictly necessary unless one insists in computing optimal solutions. In other words, one has the freedom to, for example, plan the way-points for, say, minimum path length, and then smooth the trajectory using some different criterion; this will not yield a length-(sub)optimal solution, but it would still represent a sensible way to smooth and compatibilize a trajectory so as to make it compatible with the vehicle flight envelope constraints. In fact, it can be hypothesized that in certain applications it might be preferable to ensure compatible rather than optimal flight paths. Therefore, the procedure proposed here has the flexibility to be adjusted so as to answer to the many different path planning requirements of UAVs, while ensuring that feasible trajectories are always generated even for agile and high performance vehicles operating in proximity of their flight envelope boundaries.

We consider in following three main trajectory compatibilization primitives, which are described in §III-A and §III-B. For helicopters, which can come to a full stop and hover in place, there are additional primitives which are not detailed here for the sake of brevity, although we will briefly mention them below.

A. Turn

The first trajectory compatibilization primitive involves a turn at a way-point, as depicted in Fig. 1.

The vehicle is flying in a straight trimmed flight condition $T_i$ between way-point $W_{r-1}$ and way-point $W_r$, and needs to transition to the straight trim trajectory $T_k$ between way-point $W_r$ and way-point $W_{r+1}$. The transition between the two trim trajectories involves a known change of track $\Delta \chi$.

The transition that is performed with the minimum number of maneuvers is obtained by first switching to the pre-computed maneuver $M_{ij}$, leading the vehicle from the trim condition $T_i$ to the new trim condition $T_j$, which is

![Fig. 1. First smoothing primitive: trajectory compatibilization of a turn at a way-point.](image)
a trimmed turn. This new trim trajectory is held for the unknown duration (coasting time [13]) \( t_{T_j} \), until maneuver \( M_{jk} \) is initiated. This maneuver leads the vehicle from \( T_j \) to the goal trim \( T_k \). With reference to Fig. 1, the switching points between the motion primitives of this sequence are labelled \( A \), \( B \), \( C \) and \( D \). Here and in the following, trim trajectories are shown using solid lines, while maneuvers are shown using dashed lines.

The direction cosine matrices expressing the vehicle attitude at the switching points are readily found to be

\[
\alpha_A = R(\chi_0 e_3)\alpha_{T_i},
\]

(34a)

\[
\alpha_B = \Delta R_{ij}\alpha_A = R((\chi_0 + \Delta \chi_{ij})e_3)\alpha_{T_j},
\]

(34b)

\[
\alpha_C = R(\omega_{T_j}e_3 t_{T_j})\alpha_B = R((\chi_0 + \Delta \chi - \Delta \chi_{ij})e_3)\alpha_{T_j},
\]

(34c)

\[
\alpha_D = \Delta R_{jk}\alpha_C = R((\chi_0 + \Delta \chi)e_3)\alpha_{T_k},
\]

(34d)

where \( \chi_0 \) is the initial track of the vehicle while in \( T_i \), \( \Delta \chi_{ij} \) and \( \Delta \chi_{jk} \) are the changes of track due to maneuvers \( M_{ij} \) and \( M_{jk} \), respectively, while \( \omega_{T_j} \) is the turn rate of the trimmed turn \( T_j \).

Using (34b) and (34c), we get

\[
R(\omega_{T_j}e_3 t_{T_j}) = R((\chi_0 + \Delta \chi - \Delta \chi_{ij})e_3)\alpha_{T_j}^{T^T}(\chi_0 + \Delta \chi_{ij})e_3,
\]

(35)

which yields the coasting time \( t_{T_j} \):

\[
t_{T_j} = \frac{\Delta \chi - \Delta \chi_{ij} - \Delta \chi_{jk}}{\omega_{T_j}}.
\]

(36)

This expression shows that, assuming that the heading changes \( \Delta \chi_{ij} \) and \( \Delta \chi_{jk} \) are independent of \( \omega_{T_j} \), the minimum coasting time in \( T_j \) is obtained for \( \omega_{T_j} \) maximum. Since a discrete number of trimmed turns is stored in the motion library, this simply implies picking the one that has the maximum turn rate. The total transition time between the switching points \( A \) and \( D \) is

\[
t_{AD} = \Delta t_{ij} + t_{T_j} + \Delta t_{jk}.
\]

(37)

Under the similar hypothesis that the maneuver durations \( \Delta t_{ij} \) and \( \Delta t_{jk} \) are independent of \( \omega_{T_j} \), the condition \( \min t_{T_j} \) implies \( \min t_{AD} \), and consequently the whole transition is extremal (minimum time) for the maximum possible turn rate in \( T_j \).

In reality, it is easy to remove the above introduced hypothesis of maneuver heading changes and duration independent of the turn rate \( \omega_{T_j} \). This can be useful if the flying strategy of the maneuvers stored in the motion library is particularly aggressive, when in fact one typically observes that the higher the turn rate the higher the heading changes needed to enter and leave the turn. In this case, a precise determination of the optimal turn and its associated extremal coasting time requires the evaluation of (36) and (37) for all candidate turns among those stored in the motion library, and then picking the one which yields the smallest value for \( t_{AD} \). In this context, a candidate trim trajectory is one which is reachable from the current trim \( T_i \), i.e. it is connected to it by an existing maneuver in the motion library, and which implies a turn in the correct direction (right for \( \Delta \chi > 0 \), left for \( \Delta \chi < 0 \)). The number of candidate trims is small in a typical motion library, and it is known at the time of
creation of the library; consequently, here again the number of operations needed to determine the optimal strategy can be computed up-front, and hence the procedure can be applied in real-time.

Maneuvers with non-null heading changes may cause some complications, which are discussed next. Consider in fact the following condition

\[
\text{sign}(\omega_T) = \text{sign}(\Delta \chi - \Delta \chi_{ij} - \Delta \chi_{jk}),
\]

which enforces a positive coasting time \( t_T \) in (36). When maneuver heading changes are null, the condition is obviously satisfied by performing a turn in the same sense as the required heading change. However, when maneuver heading changes are not null, (38) might not be satisfied, especially for small values of \( \Delta \chi \); for example, this will happen when the change of heading required to enter the turn with maneuver \( M_{ij} \) is already larger than the total heading change between \( T_i \) and \( T_k \). This could be solved by setting \( \Delta \chi = \Delta \chi + 2\pi \), i.e. by requiring the vehicle to perform a complete loop. A more practical solution is to generate maneuvers to and from turning trim trajectories for which the change of heading is null. As discussed previously in §II-C2, this can be easily achieved by enforcing the condition \( \Delta \chi_{ij} = 0 \) among the constraints of the maneuver optimal control problem (26). In this work, all maneuvers were generated with this condition.

The distances \( \xi \) and \( \eta \), shown in Fig. 1, which identify the points \( A \) and \( D \) where the transition is respectively initiated and terminated, are also easily computed in closed form in terms of known kinematic quantities. Let in fact be \((W_i - O)\) the position vector of the generic \( i \)th way-point \( W_i \) with respect to the origin \( O \) of the inertial frame \( F_{O,E} \). The unit tangent between way-point \( W_r \) and \( W_{r+1} \) is then

\[
t_r = \frac{(W_{r+1} - W_r)}{|(W_{r+1} - W_r)|},
\]

and similarly for the unit tangent \( t_{r-1} \) between \( W_{r-1} \) and \( W_r \).

Again with reference to Fig. 1, the closure of the polygon \( W_r ABCD \) gives:

\[
\xi t_{r-1} + \eta t_r = \Delta r_{AB} + \Delta r_{BC} + \Delta r_{CD}.
\]

The inertial components of the displacements for each one of the three motion primitives are

\[
\begin{align*}
\Delta r^e_{AB} &= R(\chi_0 e_3) \Delta r^f_{ij}, \\
\Delta r^e_{BC} &= R((\chi_0 + \Delta \chi_{ij}) e_3) S(\omega_T e_3 t_T) \alpha_T v^g_{T_j} T_j, \\
\Delta r^e_{CD} &= R((\chi_0 + \Delta \chi_{ij} + \omega_T t_T) e_3) r^g_{T_j}.
\end{align*}
\]

The displacement components for the two maneuvers \( M_{ij} \) and \( M_{jk} \) as stored in the motion library are denoted with the symbol \((\cdot)^L\); recall in fact that each maneuver is computed off-line such that it initially heads North. These components are then rotated to account for the necessary headings at points \( A \) and \( C \) in (40a) and (40c). The displacement components for the trim trajectory \( T_j \) is computed using (20c) and rotated to account for the heading at point \( B \) in (40b).

The polygon closure (39) can be written in matrix form as

\[
\begin{bmatrix}
  t^e_{r-1} & t^e_r \\
\end{bmatrix}
\begin{bmatrix}
  \xi \\
  \eta \\
\end{bmatrix} = \Delta r^e,
\]

(41)
where the right hand side is \( \Delta r^E = \Delta r^E_{AB} + \Delta r^E_{BC} + \Delta r^E_{CD} \), which is readily computed using (40). Solving for the unknown distances \( \xi \) and \( \eta \), we get

\[
\begin{bmatrix}
\xi \\
\eta
\end{bmatrix} = \frac{1}{\sin^2 \gamma_r} \begin{pmatrix}
(t^E_{r-1} - \cos \gamma_r t^E_r)^T \\
(-\cos \gamma_r t^E_{r-1} + t^E_r)^T
\end{pmatrix} \Delta r^E,
\]

where

\[\cos \gamma_r = t_{r-1} \cdot t_r\] (43)

is the angle formed by the two unit tangents along the way-point connecting segments, as shown in Fig. 1.

The previous analysis assumed that the way-points are sufficiently spaced so as to accommodate the transition. This can be verified by checking the conditions

\[\xi + \hat{\eta} \leq |(W_r - W_{r-1})|,\] (44a)
\[\eta + \hat{\xi} \leq |(W_{r+1} - W_r)|,\] (44b)

where \( \hat{\eta} \) is the distance from way-point \( W_{r-1} \) of the termination point of the previous smoothing, and \( \hat{\xi} \) the distance from way-point \( W_{r+1} \) of the initial point of the successive one, as shown in Fig. 1.

When this is not possible for all candidate trimmed turns stored in the motion library, it is necessary to turn in the opposite direction, i.e. to set \( \Delta \chi = 2\pi - \Delta \chi \), all the previous expressions remaining valid and the overall procedure unchanged. In this case, the resulting trajectory is as shown in Fig. 2.

Clearly, for helicopters an alternative strategy might be to come to a hover, turn in place until the desired heading change is achieved, and then accelerate back to the desired straight flight trim \( T_k \).

An extremal minimum time strategy for the transition was discussed previously, and used (36,37). An alternative strategy may be based on (42). For example, one may seek a strategy which implies the least deviation from way-point \( W_r \). This can be implemented by defining a cost function \( J_{\text{dist}} = J_{\text{dist}}(\xi, \eta) \), for example \( J_{\text{dist}} = \xi^2 + \eta^2 \), which measures the deviation from the way-point. Evaluating \( J_{\text{dist}} \) for each one of the possible candidate trims, one can then pick the trim trajectory which yields the smallest value for the cost. Even in this case, being the candidate trims known up-front, the procedure is implementable according to a hard real-time schedule.

B. Climb and Descent

The results of the previous section are valid also for trims \( T_i \) and \( T_k \) which imply climbing or descending flight, and in this case trim \( T_j \) will be a climbing or descending turn. However, since the climb and descent speeds stored in the motion library are finite and discrete, it is in general not possible to directly connect two successive way-points at two different and arbitrary altitudes. Hence, it is necessary to have a strategy to fly from one way-point to another, the two being located at different altitudes, using one of the discrete climb or descent speeds available in the motion library. The second and third smoothing primitives address this problem.

The second smoothing primitive is depicted in Fig. 3. Here the vehicle is flying in the trimmed straight and level flight condition \( T_i \) and needs to increase its altitude of the quantity

\[\Delta z = (W_r - W_{r-1}) \cdot e_3,\] (45)
Fig. 2. First smoothing primitive: trajectory compatibilization of a turn for closely spaced way-points.

Fig. 3. Second smoothing primitive: trajectory compatibilization of a climb/descent.

In order to reach way-point $W_r$, going back to $T_i$ at the new altitude. This transition is realized with a minimum number of maneuvers by first using maneuver $M_{ij}$ in order to reach the steady climb (or descent) trim $T_j$. Next, maneuver $M_{ji}$ brings back the vehicle to the steady level trim $T_i$ at the new altitude. The switching points between the motion primitives are labelled $A$, $B$, $C$ and $D$, as shown in the figure.
The required change of altitude can also be expressed as
\[ \Delta z = \Delta r \cdot e_3, \]
(46)
where \( \Delta r \) is the total vehicle displacement during the transition. The total displacement can be evaluated as the sum of the displacements of maneuver \( M_{ij} \) from \( A \) to \( B \), of the climbing trim trajectory \( T_j \) from \( B \) to \( C \), and of maneuver \( M_{ji} \) from \( C \) to \( D \), i.e.
\[ \Delta r = \Delta r_{AB} + \Delta r_{BC} + \Delta r_{CD}. \]
(47)
The maneuver displacements are stored in the motion library in terms of the components \( \Delta r_{L_{ij}} \) and \( \Delta r_{L_{ji}} \). The displacement components for the trim trajectory are computed using (20c) and read
\[ \Delta r_{L_{BC}} = \alpha T_j v_{B_T_j} - T_j, \]
(48)
Inserting the above expressions into (46), we get the coasting time \( t_{T_j} \) for climb \( T_j \)
\[ t_{T_j} = \frac{\Delta z - \Delta z_{ij} - \Delta z_{ji}}{V_{z,T_j}}, \]
(49)
where \( \Delta z_{ij} = \Delta r_{ij} \cdot e_3 \), \( \Delta z_{ji} = \Delta r_{ji} \cdot e_3 \) are the changes of altitude due to maneuvers \( M_{ij} \) and \( M_{ji} \), respectively, while \( V_{z,T_j} = \alpha T_j v_{T_j} - e_3 \) is the climb/descent speed in trim \( T_j \). To avoid difficulties, we consider only requests of altitude change \( \Delta z \) which are large enough to guarantee \( \text{sign}(\Delta z) = \text{sign}(\Delta z - \Delta z_{ij} - \Delta z_{ji}) \). Assuming that \( \Delta z_{ij} \) and \( \Delta z_{ji} \) are independent of \( V_{z,T_j} \), the extremal minimum time transition is obtained for \( |V_{z,T_j}| \) maximum. Since a discrete number of straight climb/descent trims is stored in the motion library, this simply implies picking the one that has the maximum vertical speed. Finally, the total transition time is obtained as
\[ t_{AD} = \Delta t_{ij} + t_{T_j} + \Delta t_{ji}. \]
(50)
Here again it is easy to remove the above hypothesis of maneuvers independent of the vertical speed, by simply evaluating (49,50) for all candidate climb/descent trims and then picking the one which yields the minimum total transition time \( t_{AD} \). It is also clear that other, non-minimum time strategies, can be easily implemented.

As shown in Fig. 3, the horizontal projection of the transition must fit within the two way-points, accounting also for the possible turn smoothings which terminate and initiate at distances \( \hat{\eta} \) and \( \hat{\xi} \), respectively, from way-points \( W_{r-1} \) and \( W_{r} \). Clearly, maneuver \( M_{ij} \) can not begin before a distance \( \hat{\eta} \) from way-point \( W_{r-1} \), and maneuver \( M_{ji} \) must finish before a distance \( \hat{\xi} \) from way-point \( W_{r} \) is reached, which leads to the following condition
\[ |\Delta r_{12}| \leq |(W'_r - W'_{r-1})| - \hat{\eta} - \hat{\xi}, \]
(51)
where \( \Delta r_{12} \) is the projection of the total transition displacement onto the \( e_1\cdot e_2 \) plane, i.e.
\[ \Delta r_{12} = (I - e_3 \otimes e_3) \Delta r. \]
(52)
Hence, the initiation point \( A \) of the transition, whose distance from \( W_{r-1} \) is labelled \( \xi \), is arbitrary as long as the following condition is satisfied
\[ \hat{\eta} \leq \xi \leq |(W'_r - W'_{r-1})| - |\Delta r_{12}| - \hat{\xi}. \]
(53)
When this is not possible, i.e. when the climb gradients stored in the motion library and therefore available to the planner are not high enough for the required change of altitude to take place between the two way-points, then the transition can be achieved using the third smoothing primitive considered in this work, which is depicted in Fig. 4.

In this case, the transition between the straight and level trim $T_i$ on the first segment and the straight and level trim $T_l$ on the second segment which involves the minimum number of maneuvers is as follows. First, maneuver $M_{ij}$ leads from the straight level trim $T_i$ to the climbing/descending turn $T_j$. This is kept for a time $t_{T_j}$, until the resulting change of altitude, together with the change of altitude of the subsequent maneuver $M_{jk}$, equals the required value $\Delta z$. Then, maneuver $M_{jk}$ leads the vehicle to the level turning trim $T_k$, which is held until the required change of track $\Delta \chi$ for the complete transition is achieved. Finally, maneuver $M_{kl}$ leads the vehicle to the trim trajectory $T_l$. In other words, the idea is to climb/descend first on a spiral until the change of altitude has been met, and then to enter a level turn to adjust the heading until the second segment is met.

The switching points between the motion primitives are labelled $A$, $B$, $C$, $D$, $E$ and $F$, and are shown in the figure.

The coasting time for trim $T_j$ is readily computed using arguments similar to the ones of the previous case, and
reads
\[ t_{T_j} = \frac{\Delta z - \Delta z_{ij} - \Delta z_{jk} - \Delta z_{kl}}{V_{z,T_j}}. \] (54)

The direction cosine matrices expressing the attitude of the vehicle at the switching points of the transition are
\[
\begin{align*}
\alpha_A &= R(\chi_0 e_3)\alpha_T, \\
\alpha_B &= \Delta R_{ij}\alpha_A = R((\chi_0 + \Delta \chi_{ij})e_3)\alpha_T, \\
\alpha_C &= R(\Delta \chi_{T_j} e_3)\alpha_B, \\
\alpha_D &= \Delta R_{jk}\alpha_C = R((\chi_0 + \Delta \chi_{ij} + \Delta \chi_{T_j} + \Delta \chi_{jk})e_3)\alpha_T, \\
\alpha_E &= R(\omega_{T_k} e_3 t_{T_k})\alpha_D = R((\chi_0 + \Delta \chi - \Delta \chi_{kl})e_3)\alpha_T, \\
\alpha_F &= \Delta R_{kl}\alpha_E = R((\chi_0 + \Delta \chi)e_3)\alpha_T,
\end{align*}
\]

where \( \Delta \chi_{T_j} \) is the change of track due to holding \( T_j \) for \( t_{T_j} \) seconds, which is computed as
\[
\Delta \chi_{T_j} = \omega_{T_j} t_{T_j} - \text{floor} \left( \frac{\omega_{T_j} t_{T_j}}{2\pi} \right) 2\pi.
\] (56)

Using (55d) and (55e), we get
\[
R(\omega_{T_k} e_3 t_{T_k}) = R((\chi_0 + \Delta \chi - \Delta \chi_{kl})e_3)\alpha_T \alpha_T^T R^T((\chi_0 + \Delta \chi_{ij} + \Delta \chi_{T_j} + \Delta \chi_{jk})e_3),
\] (57)

which yields the coasting time \( t_{T_k} \) for the level turn trim \( T_k \)
\[
t_{T_k} = \frac{\Delta \chi - \Delta \chi_{ij} - (\omega_{T_k} t_{T_j} - \text{floor}(\omega_{T_j} t_{T_j}/2\pi)2\pi) - \Delta \chi_{jk} - \Delta \chi_{kl}}{\omega_{T_k}}.
\] (58)

Finally, the total transition time is computed as
\[
t_{AF} = t_{ij} + t_{T_j} + t_{jk} + t_{T_k} + t_{kl}.
\] (59)

Here again, similarly to the previous cases, the extremal minimum time transition is obtained by choosing \( T_j \) with the maximum climb gradient and \( T_k \) with the maximum turn rate, when neglecting the dependence of the maneuvers on the vertical speed and turn rate of the trim trajectories, or by evaluating (54,59) for all candidate trims and then picking the ones which yield the minimum value for \( t_{AF} \).

The condition of positive coasting time \( t_{T_k} \) reads
\[
\text{sign}(\omega_{T_k}) = \text{sign}(\Delta \chi - \Delta \chi_{ij} - (\omega_{T_k} t_{T_j} - \text{floor}(\omega_{T_j} t_{T_j}/2\pi)2\pi) - \Delta \chi_{jk} - \Delta \chi_{kl}).
\] (60)

Even in case, the enforcement of such condition can be obtained by either storing maneuvers in the motion library such that they have null heading changes, or by setting \( \Delta \chi = \Delta \chi + 2\pi \), i.e. by requiring the vehicle to complete one extra loop.

The points of initiation and termination of the transition, points \( A \) and \( F \), are determined in terms of their distances \( \xi \) and \( \eta \) from the way-point projection \( W' \) onto the \( e_1 \cdot e_2 \) plane, as shown in Fig. 4. The closure of the polygon \( W'' ABCDEF \) gives
\[
\xi t_{\tau-1} + \Delta \xi e_3 + \eta t_{\tau} = \Delta r_{AB} + \Delta r_{BC} + \Delta r_{CD} + \Delta r_{DE} + \Delta r_{EF},
\] (61)
where the unit vectors along the path segments, \( t_r \) and \( t_{r-1} \), are now contained in the \( e_1-e_2 \) plane, and the inertial components of the vehicle displacements for each motion primitive in the transition are

\[
\Delta r_{EAB}^f = R(\chi_0 e_3)\Delta r_{ij}^\ell, \quad (62a)
\]

\[
\Delta r_{EBC}^f = R\left( (\chi_0 + \Delta \chi_{ij})e_3 \right) S(\omega_T e_3 t_{Tj}) \alpha_T v_B^T t_{Tj}, \quad (62b)
\]

\[
\Delta r_{CD}^f = R\left( (\chi_0 + \Delta \chi_{ij} + \omega_T t_{Tj})e_3 \right) r_{jk}^\ell, \quad (62c)
\]

\[
\Delta r_{DE}^f = R\left( (\chi_0 + \Delta \chi_{ij} + \omega_T t_{Tj} + \Delta \chi_{jk})e_3 \right) S(\omega_T e_3 t_{Tk}) \alpha_T v_B^T t_{Tk}, \quad (62d)
\]

\[
\Delta r_{EF}^f = R\left( (\chi_0 + \Delta \chi_{ij} + \omega_T t_{Tj} + \Delta \chi_{jk} + \omega_T t_{Tk})e_3 \right) r_{kl}^\ell. \quad (62e)
\]

Writing (61) in matrix form we get again (41), where the right hand side is now defined as \( \Delta r^f = \Delta r_{AB}^f + \Delta r_{BC}^f + \Delta r_{CD}^f + \Delta r_{DE}^f + \Delta r_{EF}^f - \Delta z e_3^f \). The solution is again expressed by (42), where the angle \( \gamma_r \) is shown in Fig. 4.

Even in this case, it is possible to base other, non-minimum time, strategies on the knowledge of the distances \( \xi \) and \( \eta \).

Finally, we notice that for a helicopter an alternative strategy is to come to a hover and then climb vertically, accelerating to the straight flight goal trim once the new altitude has been achieved. This and other transitions can be easily computed following a similar reasoning, and are not detailed here for brevity.

IV. NUMERICAL RESULTS

In order to demonstrate the procedures discussed in this paper, we report here on a series of experiments conducted in a high-fidelity virtual environment. While work towards flight testing on a small size UAV is well underway, the analysis of the proposed algorithm in a virtual environment is important since it allows for a far more precise quantification of its performance, and in particular of the tracking errors. The vehicle considered for the tests is a medium-size multi-engine four-bladed articulated-rotor utility helicopter. This aircraft is not in the class of the typical small size rotorcraft vehicles considered in most of the recent UAV literature, but it was used here because its model was carefully validated in previous research efforts with respect to flight data.

The mathematical model of this helicopter is based on (1), where the main and tail rotor forces and moments are computed by combining actuator disk and blade element theory, considering a uniform inflow. The rotor attitude is evaluated by means of quasi-steady flapping dynamics with a linear aerodynamic damping correction. Look-up tables are used for the quasi-steady aerodynamic coefficients of the vehicle lifting surfaces (horizontal and vertical tails), and the aerodynamic model includes corrections to account for compressibility effects and for the downwash angle at the tail due to the main rotor. Detailed descriptions of this class of rotorcraft models are available in textbooks, such as for example Ref. [17].

We first quantized the vehicle dynamics by creating a motion library for the example helicopter, as described in §IV-A. Next, a complex path was defined by prescribing a sequence of way-points and by assigning flight speeds along the connecting segments. Then, the smoothing procedures described in this paper were used to compatibilize the trajectory using motion primitives from within the library. Finally, the compatibilized trajectory was tracked...
by the helicopter model in the presence of disturbances, using as reflexive controller an output-feedback Linear Quadratic Regulator (LQR) augmented with a drift and heading compensator, whose implementation is briefly described in §IV-B. The results of the tracking experiments are reported in §IV-C.

A. Motion Library

A small size library of motion primitives was computed numerically, solving problem (24) for the determination of the trim trajectories, and problem (26) for the connecting maneuvers. All trim trajectories were computed using null side-slip as the piloting condition. For all maneuvers, the cost function was defined as $J_{\text{maneuver}} = \Delta t + \int_0^{\Delta t} u^T W_u u\, dt$, which defines a minimum time transition with a term penalizing excessive control effort through the symmetric matrix $W_u > 0$ of weight coefficients, and the optimal control problem included the null track angle change $\Delta \chi_{ij} = 0$ among the constraints. The helicopter controls are defined as $u = (A_1, B_1, \theta_{\text{MR}}, \theta_{\text{TR}})^T$, where $A_1, B_1$ are the lateral and longitudinal cyclics, respectively, while $\theta_{\text{MR}}$ is the main rotor collective and $\theta_{\text{TR}}$ is the collective of the tail rotor.

The library includes, for the trim trajectories, a hover and, for three different values of the flight speed (slow, moderate and fast), straight and level flight trims, straight climb and descent, and left and right level turns. At each value of the flight speed, maneuvers connect to and from each straight and level flight condition with the other trims at the same flight speed. Transition between the flight speeds is allowed by acceleration and deceleration maneuvers connecting to and from each straight and level flight condition with all other straight and level flight conditions in the library.

The maneuver optimal control problems were solved on uniform grids. To facilitate and speed up convergence, the computation was initiated on coarse grids made of only two elements; in fact, very crude discretizations have the effect of decreasing the sensitivity of the numerical procedure to imprecise initial guesses. The solution obtained on the initial grid was then projected onto a finer grid obtained by uniform sub-division of the initial one, and a new solution was computed. This procedure was continued until a sufficient refinement in the grid was achieved. A further improvement in terms of robustness was obtained by scaling all problem unknowns [10].

To give an example of the maneuvers obtained by solving problem (26), Fig. 5 shows the time histories of the roll (left) and pitch (right) angles, computed for a maneuver from a fast straight and level trim to a fast right level turn. Notice how the request of null track change induces a pronounced non-minimum phase roll response.

B. Reflexive Control Layer

The simulation environment used for tracking includes models of the vehicle actuators and sensors, realistic values of all sources of process and measurement noise and delays. The vehicle state variables are reconstructed using an extended Kalman filter. The LQR controller operates on the outputs $y = (|v|, v, V_z, p, q, \Omega, \phi, \theta)^T$, i.e. on the vehicle speed, $y$ body-attached velocity component, climb speed, roll and pitch angular velocities, turn rate, and roll and pitch angles, respectively, by minimizing the tracking cost

$$J_{\text{track}} = \int_z^{\infty} \left( (y - y^*)^T W_y (y - y^*) + (u - u^*)^T W_u (u - u^*) \right)\, dt,$$

(63)
Tracking is performed by first sampling states and controls of the planned sequence of trims and maneuvers. This yields a sequence of points $P^*_i$, where all the desired vehicle states and associated control inputs are known. These “goal” quantities are indicated in the following with the notation $(\cdot)^*$. Each time the reflexive controller is called, the closest point $P^*$ along the to-be-tracked trajectory is found by solving the following problem

$$r^* = (1 - \nu)r_i^* + \nu r_{i+1}^*,$$

$$\nu = \frac{(r - r^*) \cdot (r_{i+1}^* - r_i^*)}{|r_{i+1}^* - r_i^*|^2},$$

whose solution is readily found as

If $\nu > 1$ (respectively, $\nu < 0$), the closest point is searched in another segment by increasing (respectively, decreasing) the node counter $i$. The to-be-tracked and actual trajectories are shown in Fig. 6, the former using a solid line and the latter using a dashed line. Based on the knowledge of the coefficient $\nu$, the vehicle states and controls are interpolated at point $P^*$ using $(\cdot)^* = (1 - \nu)(\cdot)_i^* + \nu(\cdot)_{i+1}^*$, to yield the goal values $y^*$ and $u^*$ at the current time instant.

The output-feedback LQR control law is computed at 50 Hz as

$$u_{LQR} = u^* - K(y - y^*).$$

The non-linear helicopter model is linearized in every trim condition of the motion library, and the optimal output-feedback gain matrix $K$ is computed based on the procedure of Ref. [18]. When the goal point is along a maneuver, the optimal control gains are those computed using the linearized vehicle model in the arrival trim.
The weight matrices $W_y$ and $W_u$ of the cost $J^{\text{track}}$ are assumed to be diagonal, and their entries are determined using an optimization technique. At first, a complex path is planned that uses as many primitives as possible from the library. Next, the vehicle model is flown so as to track this path under the action of the output-feedback controller. The tracking error (63) of this path following problem is then interpreted as the cost for the optimization, this cost being considered a sole function of the entries of the matrices $W_y$ and $W_u$. The tracking error is minimized by using a gradient-based optimization technique coupled with a response surface for the approximation of the cost function.

To improve the tracking performance, a slower outer loop operating at 1 Hz is introduced to reduce the drift and heading errors, which are not seen by the LQR controller since the outputs it operates on are all necessarily invariant quantities.

The drift and heading compensation is based on a simple proportional-integral (PI) controller. Assuming that the $B_1$, $A_1$ and $\theta_{\text{hbr}}$ controls are primarily responsible for the generation of control forces in the local body-attached directions $b_1$, $b_2$ and $b_3$, respectively, the drift-compensated control inputs are computed as

$$
\begin{bmatrix}
B_1 \\
A_1 \\
\theta_{\text{hbr}}
\end{bmatrix} =
\begin{bmatrix}
B_1 \\
A_1 \\
\theta_{\text{hbr}}
\end{bmatrix}_{\text{LQR}} - \text{diag}(K_P^{B_1}, K_P^{A_1}, K_P^{\theta_{\text{hbr}}}) \alpha^T d^e - \text{diag}(K_I^{B_1}, K_I^{A_1}, K_I^{\theta_{\text{hbr}}}) \alpha^T \int_{t-\Delta t}^{t} d^e(\tau) \, d\tau,
$$

i.e. the corrective action is induced by the body-attached components of the position error $d = r - r^*$ through the proportional gains $K_P^{B_1}$, $K_P^{A_1}$, $K_P^{\theta_{\text{hbr}}}$ and the integral gains $K_I^{B_1}$, $K_I^{A_1}$, $K_I^{\theta_{\text{hbr}}}$. 

Fig. 6. Goal to-be-tracked (solid line) and actual (dashed line) trajectories, and their projections onto the $e_1$-$e_2$ plane.
Similarly, considering that the $\theta_{0_{\text{TR}}}$ control input is primarily responsible for the generation of control moments about the $b_3$ body-attached direction, the heading-compensated control input is computed as

$$
\theta_{0_{\text{TR}}} = \theta_{0_{\text{TR,LQR}}} - K_P^{\theta} \alpha_{33} (\chi - \chi^*) - K_I^{\theta} \alpha_{33} \int_{t-\Delta t}^{t} (\chi(\tau) - \chi^*(\tau)) \, d\tau,
$$

where $\chi$ and $\chi^*$ are computed with (8) using $v$ and $v^*$, respectively, while $K_P^{\theta}$, $K_I^{\theta}$ are the heading proportional and integral control gains, respectively, and $\alpha_{33} = \cos \theta \cos \phi$ projects the heading error into a yaw error in the body-attached frame.

Similarly to the case of the LQR parameters $W_y$ and $W_u$, even the drift and heading compensator gains are computed using an optimization technique. In this case the optimization cost is defined as the tracking error of the non-invariant quantities, i.e.

$$
J_{\text{track}} = \int_{t}^{\infty} (dT + w(\chi - \chi^*)^2) \, dt,
$$

where $w > 0$ is a weight factor. This cost is considered a sole function of the control gains of the compensator, which is minimized using the same technique described before.

C. Trajectory Tracking Performance

To test the performance of the proposed methodology, a highly aggressive trajectory was defined by the following sequence: acceleration from 20 m/s to 30 m/s, 30 m climb, 90 deg right turn at maximum rate, straight and level flight at 30 m/s, 90 deg right turn at maximum rate, 30 m descent, deceleration back to 20 m/s.

In the following, we consider both a non-compatibilized trajectory with instantaneous switches between the trim conditions, and the one obtained by the proposed smoothing. Both the compatibilized and non-compatibilized trajectories were tracked using the output-feedback LQR controller with drift and heading compensation.

Figure 7 shows the goal to-be-tracked compatibilized trajectory, as well as the trajectories obtained by the vehicle as steered by the LQR reflexive controller when tracking both the compatibilized and the non-compatibilized goal quantities. Notice that, to better highlight the differences, the axes of the three-dimensional plot are not to scale.

Figure 8 reports the history of the normalized local tracking errors for the three body-attached components of the angular velocity, which are here defined as

$$(|p(t)| - |p^*(t)|)/||\omega||_{\infty}, (|q(t)| - |q^*(t)|)/||\omega||_{\infty}, (|r(t)| - |r^*(t)|)/||\omega||_{\infty},$$

where $||\omega||_{\infty}$ is the maximum value of the angular speed throughout the simulation.

The plots show that in the non-compatibilized case the LQR reflexive layer generates far larger tracking errors, and it eventually diverges. This behavior is due to the high aggressiveness of the goal trajectory, with its rapid switching from one trim to the next which is hard to follow for the reflexive controller, especially if the trajectory has not been smoothed in order to make it compatible with the vehicle dynamics.

For the same problem, Fig. 9 shows the yaw rate time history. The solid line shows the time history of the goal to-be-tracked signal, while the dashed line gives the yaw rate time history generated by the tracking vehicle. The left figure shows the compatibilized case; the tracking error is quite small, and the desired signal is followed well throughout the whole duration of the simulation. The right part of the figure shows the results obtained for the non-compatibilized case.
To quantify the effects on the tracking errors of the measurements and process noise used in the previous simulations, the same example was run without noise. Figure 10 shows the obtained yaw rate time histories in this case, which should be compared with the ones of Fig. 9. It appears that the non-compatibilized case still shows substantial tracking errors, which are solely due in this case to the fact that the LQR is trying to track a non-physical trajectory.
Partial Flight Envelope Protection: Finally, we show the partial flight envelope protection offered by the proposed compatibilization. The protection is partial in the sense that although the planned trajectory is guaranteed to remain within the flight envelope, tracking will never be perfect due to modeling assumptions and the presence of disturbances and noise, so that a further protective layer should also be implemented at the level of the tracking controller to avoid leaving the flight envelope.

Figure 11 shows the time history of the load factor experienced by the vehicle during tracking in the presence of noise and disturbances, using a dashed line; the load factor corresponding to the goal planned trajectory is plotted using a solid line. The load factor experienced by the tracking helicopter when following the compatibilized
trajectory is shown in the left part of the figure, while the non-compatibilized solution is shown in the right part of the same figure. It appears that the compatibilized solution closely follows the reference signal. In fact, the maximum load factor throughout the simulation is \( n \approx 1.7 \) g, which is only marginally larger than the maximum prescribed value \( n_{\text{max}} = 1.5 \) which was included among the constraints of all maneuver defining problems (26). On the other hand, in the non-compatibilized case the vehicle far exceeds the limit value of the load factor, generating two spikes at the exit of the turns which reach \( n \approx 2.8 \), before finally diverging.

One might be tempted to think that the problem will be alleviated by using some smooth interpolation of the goal outputs between two successive trims, rather than the abrupt switches used here. To test this hypothesis, we interpolated all quantities at the interface between two successive trims. More precisely, consider trims \( T_i \) and \( T_j \) and let \( t_s \) be the switching time between the former and the latter. The instantaneous switch is now removed by performing a smooth interpolation on a time interval \( \Delta t \), using the following expression

\[
a(t) = a_{T_i} + (a_{T_j} - a_{T_i}) \frac{1 - \cos(\pi \zeta)}{2},
\]

where \( a(t) \) is the interpolated quantity at time \( t \), and

\[
\zeta = \frac{t - t_s + \Delta t/2}{\Delta t}, \quad t_s - \Delta t/2 \leq t \leq t_s + \Delta t/2, \quad 0 \leq \zeta \leq 1,
\]

is the interpolation coefficient. The quantity \( \Delta t \) represents the support of the interpolation: by increasing \( \Delta t \) one realizes the transition between the initial value \( a_{T_i} \) and the final one \( a_{T_j} \) over a larger time interval. Clearly, such time intervals have to be chosen so that the coasting times remain positive; for the problem at hand, this poses a constraint on the interval which should be \( \Delta t \leq 4.48 \) sec, the upper value being used in the results reported below.

The load factor experienced by the tracking helicopter when following the goal quantities interpolated using (70) is shown in Fig. 12. Similarly to the previous case, the goal planned quantity is plotted using a solid line and the
Fig. 12. Time history of load factor for the goal trajectory (solid line) and corresponding quantity for the tracking helicopter (dashed line), when using interpolation (70).

one realized by the tracking helicopter using a dashed line.

It appears that, although in this case the tracking controller does not eventually lose control of the vehicle, the load factor substantially exceeds the desired limits, reaching $n \approx 2.2$. Therefore, although by interpolating between successive trims one can make the tracking problem somewhat simpler, it is clear that the obtained trajectory may still not be compatible with the vehicle dynamics; this in turn can lead the vehicle to leave its flight envelope, unless specific additional protection systems are implemented.

V. CONCLUSIONS

We have proposed a novel planning strategy for high performance aggressive autonomous vehicles. The procedure is based on the idea of smoothing in an optimal fashion a given trajectory expressed in terms of a sequence of way-points and connecting straight trim flight conditions.

Departing from similar approaches which simply use crude kinematic models, the smoothing is here obtained by using a transcribed version of the vehicle dynamics expressed in terms of motion primitives.

The resulting extremal trajectories are compatible by design with the vehicle model embedded in the motion primitives. Since the primitives can be based on off-line computations performed on very sophisticated vehicle models or obtained experimentally, they provide for a potentially very faithful representation of the vehicle dynamics in a compact and small-dimensional yet semantically very rich space. This means that the resulting trajectories will be by design confined within the flight envelope of the vehicle, and will be trackable with small error by suitable reflexive controllers. These properties, which are particularly relevant to the case of high performance vehicles, are not shared by smoothing approaches based on cruder (purely kinematic) vehicle models.

Although the procedures can be based on non-linear models of the vehicle, yet the planning algorithm can be solved in closed form, yielding simple explicit expressions for all problem unknowns, effectively enabling an
implementation satisfying hard real-time constraints. This important feature is achieved by the quantization of the system dynamics into a finite dimensional library of motion primitives, which effectively shields the real-time planner from the complexity of generating maneuvers and trim trajectories.

The proposed method has been demonstrated using a simulated helicopter model, and is currently being ported to the on-board system of a rotorcraft UAV in the lab of the authors.

REFERENCES