International Symposium on Coupled Methods in Numerical Dynamics
FESB, Split, Croatia, September 16th-19th 2009.

A Unified Approach to Trajectory Optimization and Parameter Estimation in Vehicle Dynamics

C.L. Bottasso*, Fabio Luraghi and Giorgio Maisano

Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale,
Via La Masa 34, 20156 Milano, Italy

Abstract In this paper, trajectory optimization and parameter estimation problems in vehicle dynamics are first defined using a unified notation. Next, it is shown that both can be formulated as two-point boundary value constrained optimization problems defined over a temporal domain of known or unknown duration. We further show that both can be discretized in time using the same algorithms, obtaining a constrained algebraic optimization problem which is formally identical in the two cases, and which can be solved using standard techniques. The unified formulation is exploited for developing a general purpose computer code for the solution of both classes of problems with applications in rotorcraft flight mechanics.

Key words: *Optimization, trajectory optimization, parameter estimation, multibody dynamics, flight mechanics, rotorcraft vehicles.*

1. Introduction

In this paper we formulate a unified approach to the problems of trajectory optimization and parameter estimation in vehicle dynamics.

In a trajectory optimization or Maneuver Optimal Control Problem (MOCP), one is interested in computing an extremal maneuver for a given vehicle model; the sought solution minimizes a cost function while satisfying given constraints which translate various requirements including the boundaries of the performance envelope of the vehicle [1, 2].

In a Parameter Estimation Problem (PEP), one is interested in finding estimates of some free quantities in a parametric model, which best fit some given measurements of the response of the actual vehicle gathered during experimental observations [3].

Both problems amount to two-point boundary value optimization problems in the time domain: the former is formulated for a given model and has the

*Correspondence to: Carlo L. Bottasso, Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, Via La Masa 34, 20156 Milano, Italy. E-mail: `carlo.bottasso@polimi.it`; Tel.: +39-02-2399-8315; Fax: +39-02-2399-8334.

vehicle inputs as primary unknowns, while the latter is formulated for given inputs and associated measured responses and has the model parameters as primary unknowns. In the MOCP case, the temporal duration of the maneuver is often unknown; this however does not cause any conceptual difficulty, since the duration itself can be treated as an additional unknown of the problem. When discretized in the time domain, both problems can be turned into non-linear constrained algebraic optimization problems, for which efficient solution techniques are available.

The unified formulation of the two classes of problems was exploited for developing a general purpose software program. The code has been specifically developed for applications in rotorcraft flight mechanics, although it could be readily modified for the analysis of other vehicles. The procedures implemented in the code support efficiently the use of vehicle models of varying complexity, a measure which is here defined so as to account for both the number of degrees of freedom in the model and the ratio of the fastest scales in the solution to the duration of the maneuver or experimental observation (which is related to the number of time steps necessary for covering the temporal domain). The bulk of the code is shared between both applications, with differences relegated to a few percent points of the total line count of the program and due to the different definitions of cost functions, constraints, and unknowns for the two cases; clearly, this leads to substantial savings in code development, debugging, validation and maintenance.

The paper starts by formulating MOCPs and PEPs as optimization problems, using a single common notation. Next, the discretization of the resulting two-point boundary value problems is described, leading to a standard algebraic constrained optimization. Techniques for handling models of varying complexity are described and compared. Finally, we briefly review applications in rotorcraft flight mechanics which cover both classes of problems.

2. Vehicle models

Consider a generic multi-physics multibody vehicle model $\mathcal{M}$ described in terms of a set of non-linear differential algebraic equations written as

$$\boldsymbol{f}_{\mathrm{SD}}(\dot{\boldsymbol{x}}_{\mathrm{SD}}, \boldsymbol{x}_{\mathrm{SD}}, \boldsymbol{\lambda}, \boldsymbol{x}_{\mathrm{C}}, \boldsymbol{u}, \boldsymbol{p}, \boldsymbol{w}, t) = \boldsymbol{0}, \tag{1a}$$

$$\boldsymbol{c}(\boldsymbol{x}_{\mathrm{SD}}, t) = \boldsymbol{0}, \tag{1b}$$

$$\boldsymbol{f}_{\mathrm{C}}(\dot{\boldsymbol{x}}_{\mathrm{C}}, \boldsymbol{x}_{\mathrm{C}}, \boldsymbol{x}_{\mathrm{SD}}, \boldsymbol{u}, \boldsymbol{p}, \boldsymbol{w}, t) = \boldsymbol{0}, \tag{1c}$$

where $\boldsymbol{x}_{\mathrm{SD}}$ are the structural dynamics states, $\boldsymbol{\lambda}$ are constraint-enforcing Lagrange multipliers, $\boldsymbol{x}_{\mathrm{C}}$ are the states describing the coupled fields (e.g. aerodynamic, hydraulic, etc.), and $\boldsymbol{u}$ is the control input vector. Equations (1) also depend on a set of model parameters $\boldsymbol{p}$, which are unknown in the case of parameter estimation problems whereas should be regarded as known in all other cases. Finally, vector $\boldsymbol{w}(t)$ models other exogenous inputs and disturbances acting on the system (e.g., gusts and air turbulence). Equations (1a) group together the equations of dynamic equilibrium and the kinematic equations. Equations (1b) represent mechanical joint constraint equations, while (1c) are the coupled physics governing equations.

The equations of motion of model $\mathcal{M}$ can be written in a more compact form as

$$\boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}, \boldsymbol{w}, t) = \mathbf{0}, \tag{2}$$

where for the sake of notational simplicity, but with no loss of generality, we have eliminated the Lagrange multipliers, $\boldsymbol{x}$ now grouping together the structural dynamics and the coupled field states. The model definition is completed by specifying a set of outputs

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}), \tag{3}$$

which typically represent some global vehicle states describing its gross motion, or other quantities useful for formulating maneuver or parameter estimation problems on model $\mathcal{M}$.

3. The maneuver optimal control problem

A general MOCP [4, 1] for model $\mathcal{M}$ can be formulated as:

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{u},T} \quad J^{\text{MOCP}}(\boldsymbol{y}, \boldsymbol{u}, T, T_i), \tag{4a}$$

$$\text{s.t.:} \quad \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}^*, \boldsymbol{w}^*, t) = \mathbf{0}, \tag{4b}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}), \tag{4c}$$

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, t, T, T_i) \leq \mathbf{0}. \tag{4d}$$

The problem is defined over the interval $\Omega = [0, T]$, $t \in \Omega$, where the final time T is typically unknown and must be determined as part of the solution. Specific events might be associated with unknown time instants T_i, $0 < T_i < T$ (for example, the jettisoning of part of the cargo or other instantaneous conditions).

J^{MOCP} indicates the to-be-minimized cost, which, depending on the problem, might account for maneuver duration, control activity, fuel consumption, etc., or some other given function of interest that typically expresses an index of performance of the vehicle.

The maneuver definition is completed by providing a set of problem-dependent equality and inequality constraints (equations (4d)) which translate the operating envelope of the vehicle, the performance and procedural requirements as dictated by norms and regulations (for example, in the case of the certification of flying vehicles), and all other necessary maneuver-defining constraints. All such constraints are typically expressed in terms of the outputs $\boldsymbol{y}$. Finally, equations (4d) also include initial and final conditions on the vehicle states $\boldsymbol{x}$.

Notice that the problem is formulated for a given vehicle model, i.e. for fixed values of the model parameters $\boldsymbol{p} = \boldsymbol{p}^*$, where the symbol $(\cdot)^*$ indicates a know assigned value. Similarly, if exogenous inputs are present, these are also known, so that $\boldsymbol{w}(t) = \boldsymbol{w}^*(t)$.

4. The parameter estimation problem

A general PEP for the parametric model $\mathcal{M}(\boldsymbol{p})$ can be formulated as

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{p}} \quad J^{\mathrm{PEP}}(\boldsymbol{z}-\boldsymbol{y}), \tag{5a}$$

$$\text{s.t.:} \quad \boldsymbol{f}(\dot{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{u}^*,\boldsymbol{p},\boldsymbol{w},t)=\boldsymbol{0}, \tag{5b}$$

$$\boldsymbol{y}=\boldsymbol{h}(\boldsymbol{x}), \tag{5c}$$

$$\boldsymbol{g}(\boldsymbol{p})\leq\boldsymbol{0}, \tag{5d}$$

where $\boldsymbol{z}$ are measurements of the outputs gathered at N discrete sampling time instants t_k during the experimental test,

$$\boldsymbol{z}(t_k)=\boldsymbol{y}(t_k)+\boldsymbol{v}(t_k). \tag{6}$$

The available measures are affected by noise $\boldsymbol{v}$ with covariance $\boldsymbol{R}_k = E[\boldsymbol{v}_k\boldsymbol{v}_k^T]$, $E[\cdot]$ being the expected value operator. The presence of measurement noise, together with the possible presence of a process noise term $\boldsymbol{w}$ for modeling disturbances acting on the system (e.g., air turbulence), makes the problem of a stochastic nature. Hence, the to-be-minimized cost function J^{PEP} is typically a statistical measure of the match between measured quantities $\boldsymbol{z}$ and model outputs $\boldsymbol{y}$.

A Maximum Likelihood estimator is obtained by choosing

$$J^{\mathrm{PEP}}=\det(\boldsymbol{R}), \tag{7}$$

where $\boldsymbol{R}=1/N\sum_{k=1}^{N}\big(\boldsymbol{z}(t_k)-\boldsymbol{y}(t_k)\big)\big(\boldsymbol{z}(t_k)-\boldsymbol{y}(t_k)\big)^T$. Alternatively, a weighted Least Squares estimator is obtained if

$$J^{\mathrm{PEP}}=\frac{1}{2}\sum_{k=1}^{N}\big(\boldsymbol{z}(t_k)-\boldsymbol{y}(t_k)\big)\boldsymbol{W}\big(\boldsymbol{z}(t_k)-\boldsymbol{y}(t_k)\big)^T, \tag{8}$$

where $\boldsymbol{W}$ is a weight matrix. This method can be seen as a particular case of the Maximum Likelihood method for known measurement noise covariance matrix, $\boldsymbol{W}=\boldsymbol{R}^{-1}$ [3]. In the Filter Error Method [3, 5], the system states obtained by integrating model (5b) are corrected by a Kalman filter, whose role is to stabilize the integration around the measurements and to account for the presence of process noise; details are omitted for brevity, but the PEP can still be expressed in a form resembling (5).

Inequality (5d) enforces possible constraints on the model parameters. Such constraints ensure that the estimated parameters lie within acceptable bounds and do not take at convergence values which are non-physical.

Notice that in this case the model inputs are known and fixed to the values $\boldsymbol{u}(t_k)=\boldsymbol{u}^*(t_k)$ measured during the experimental test (values in between the sampling instants may be interpolated, if necessary). Similarly, the temporal domain $\Omega=[0,T^*]$ is also known.

When the system $\mathcal{S}$ (the plant) is unstable, it is usually artificially stabilized by means of a control system (CS) (see Figure 1). The presence of the feedback loop will in general tend to suppress the external excitation $\boldsymbol{u}_p$ supplied to the system during the experimental test; furthermore it correlates inputs and outputs, and induces a stochastic excitation into the system generated by the process noise term. Nonetheless, parameter estimation performed from data collected in closed-loop is the only method that can be used in most practical cases when dealing with unstable vehicles for obvious safety reasons.

Closed-loop parameter estimation can be performed using the direct approach [6], in which one ignores the feedback by using the sole measurements of plant inputs and outputs, i.e. on the basis of the input/output relation $\boldsymbol{u}/\boldsymbol{y}$. This situation is illustrated in Figure 1. Notice that this approach does not require knowledge of the CS. Therefore, there is no possible effect of regulator modeling approximations on the quality of the results, and the method is applicable even when the CS is not available, for example because covered by proprietary rights.

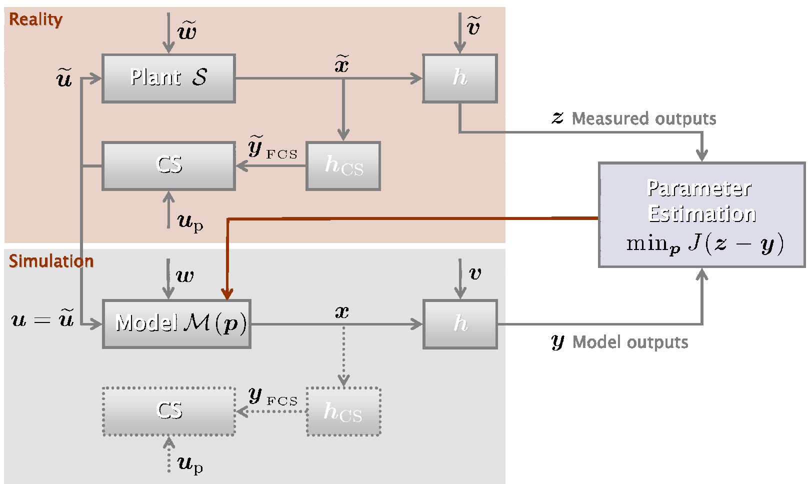

Figure 1. Parameter estimation using the direct approach. Here, quantities related to the plant (the true system, as opposed to a model of it) are indicated using the notation $\widetilde{(\cdot)}$.

5. Solution techniques for two-point boundary value optimization problems

Problems (4) and (5) are two-point boundary value optimization problems, and both can be solved using the same techniques. In this work, we use the so-called direct optimization approach [7], which is based on the idea of eliminating the temporal dimension by discretization, this way obtaining a non-linear discrete optimization (or non-linear programming, NLP) problem. This can in general be written as

$$\min_{\boldsymbol{\pi}} \quad J^{\mathrm{NLP}}(\boldsymbol{\pi}), \tag{9a}$$

$$\text{s.t.} \quad \boldsymbol{a}(\boldsymbol{\pi}) = \boldsymbol{0}, \tag{9b}$$

$$\boldsymbol{b}(\boldsymbol{\pi}) \leq \boldsymbol{0}, \tag{9c}$$

where $\boldsymbol{\pi}$ is a set of algebraic unknowns, and J^{NLP} is a scalar objective function which represents an approximation of the cost of equation (4a) or (5a). The equality constraints (9b) are generated by the discretization of the equations of motion (4b) or (5b), while the inequality constraints (9c) by the discretization of the constraints (4d) or (5d). The specific form of the vector of algebraic unknowns and of the constraints depends on the method used for performing the discretization of problems (4) and (5), as detailed below.

The NLP problem (9) is solved using sequential quadratic programming (SQP), with gradients computed by perturbation [8]. For robustness, it is usually convenient to consider a scaled version of problem (9), where the NLP variables $\boldsymbol{\pi}$ are replaced by scaled quantities $\hat{\boldsymbol{\pi}} = \mathrm{diag}(\boldsymbol{w}_\pi)\boldsymbol{\pi}$, $\boldsymbol{w}_\pi$ being scaling coefficients chosen so that the new unknowns are $\hat{\boldsymbol{\pi}} \approx \mathcal{O}(1)$. Likewise, also the constraints (9b,9c) are similarly scaled.

5.1. Problem complexity

The discretization of problems (4) and (5) to yield an equivalent NLP problem must be made by taking into account the underlying complexity of the problems to be solved. In the context of the present discussion, a meaningful measure C of the *complexity* of problems (4) and (5) is given by the expression

$$C = \kappa \frac{T}{\tau}, \tag{10}$$

where κ is the number of unknown components in the problem, T is the length of the temporal domain and τ is the characteristic time length associated with the fastest solution scale component that one needs to resolve.

For the MOCP case, κ is the number of states plus the number of inputs and outputs. For sophisticated models (as for example aeroelastic models with flexible components and refined aerodynamics), the number of states is significantly larger than the number of inputs (which is usually very small for most vehicles) and of outputs (which is also typically a rather small number). For the PEP case, κ is the number of states, outputs and unknown model parameters; here again, for sophisticated models, the number of states dominates since just a rather limited number of model parameters are typically observable and can be estimated using a given set (or sets) of measurements.

Problems of modest complexity use vehicle models with a small number of unknown quantities κ, and a solution is sought which captures only those components of the response which are slow compared to the overall duration (T/τ small). On the contrary, problems of high complexity use models with many unknown quantities (κ large, typically because of a large number of states) which capture fine scale response components, that are possibly very fast compared to the length of the temporal domain (T/τ large).

5.2. Direct transcription

Using the direct transcription approach, the time interval Ω is partitioned as $0 = t_0 < t_1 < \ldots < t_N = T$, where the generic time element, noted $\Omega^i = [t_i, t_{i+1}]$, $i = (0, N-1)$, has size $h^i = t_{i+1} - t_i$. Quantities associated with the generic element vertex i are indicated using the notation $(\cdot)_i$, while quantities associated with the generic element i are labeled $(\cdot)^i$. The time step size is a function of the final time when T is unknown, i.e. $h^i = h^i(T)$.

In each time element Ω^i, the governing equations (4b) or (5b) are discretized using a suitable numerical method. Considering a low order scheme without internal stages for notational simplicity, the discrete equations write

$$\boldsymbol{f}_h(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i, \boldsymbol{u}^i, \boldsymbol{p}, \boldsymbol{w}, h^i) = \boldsymbol{0}, \qquad i = (0, N-1), \tag{11}$$

where $\boldsymbol{f}_h$ is an algorithmic approximation of function $\boldsymbol{f}$, $\boldsymbol{x}_i$, $\boldsymbol{x}_{i+1}$ are the values of the state vector at t_i and t_{i+1}, respectively, while $\boldsymbol{u}^i$ represents the value of the control vector within the step. Clearly, in the MOCP case $\boldsymbol{p}$ is known, and the NLP variables are chosen as:

$$\boldsymbol{\pi} = (\boldsymbol{x}_{i=(0,N)}, \boldsymbol{u}^{i=(0,N-1)}, T), \tag{12}$$

i.e. they are defined as the discrete values of states and controls on the computational grid, and the final time. On the other hand, for the PEP case $\boldsymbol{u}^i$ is known and the NLP variables are:

$$\boldsymbol{\pi} = (\boldsymbol{x}_{i=(0,N)}, \boldsymbol{p}). \tag{13}$$

In both cases, the optimization cost (equation (4a) or (5a)) is discretized in terms of $\boldsymbol{\pi}$, obtaining the discrete cost J^{NLP} of equation (9a). Then, the model and output equations (4b,4c) or (5b,5c) are discretized over each step using equations (11), and become the set of NLP equality constraints appearing in equations (9b). Finally, all other problem constraints and bounds, equations (4d) or (5d), are expressed in terms of the NLP variables $\boldsymbol{\pi}$ and become the NLP inequality constraints of equations (9c).

The resulting problem is potentially large; in fact, the problem size is proportional to the complexity C defined in equation (10), i.e. it grows not only with the number of states, but also with the number of time steps N, which increases with the ratio T/τ. This means that, although the problem is very sparse and has a banded nature due to the fact that each time step couples together only a few of the discrete unknows, the direct transcription approach is applicable only to problems of not excessive complexity.

On the other hand, the method is typically very robust; since the equations of motion are discretized on the temporal grid and retained as constraints, the solution is much more stable than in the case of the shooting methods described next.

5.3. Direct multiple shooting

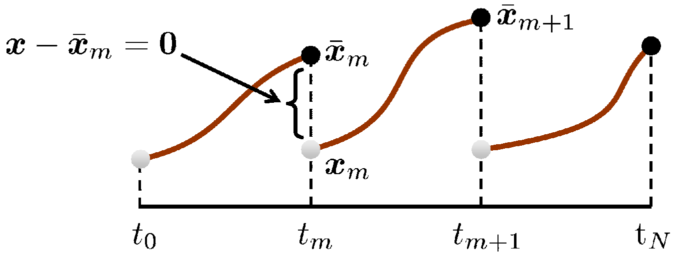

Figure 2. Basic principle of the multiple shooting method.

The time domain Ω is partitioned as $0 = t_0 < t_1 < \ldots < t_M = T$ with $\Omega^m = [t_m, t_{m+1}]$, $m = (0, M-1)$, where each Ω^m is a shooting segment, as illustrated in Figure 2. Quantities associated with the generic vertex j between segments are indicated using the notation $(\cdot)_j$, while quantities associated with the generic segment k are labeled $(\cdot)^k$. For the MOCP case, the controls are discretized in each shooting segment Ω^m as $\boldsymbol{u}^m(t) = \sum_{i=1}^{N_c^m} s_i(t)\boldsymbol{u}_i^m$, where $s_i(t)$ are basis functions, in particular cubic splines in the present implementation, and $\boldsymbol{u}_i^m$ are N_c^m unknown discrete control values. Control approximations are confined on each shooting segment, instead of considering interpolations across segment boundaries; this has the effect of decreasing the computational cost of finite differencing by increasing the problem sparsity. Constraints are enforced at the shooting segment boundaries to enforce the continuity of the controls up to C^1.

For the MOCP case, the set of NLP variables is chosen as:

$$\boldsymbol{\pi} = (\boldsymbol{x}_{m=(0,M)}, \boldsymbol{u}_{i=(1,N_c^m)}^{m=(0,M-1)}, T), \tag{14}$$

i.e. they are defined as the discrete values of the states at the interfaces between shooting segments, the discrete values of the controls within each segment, and the final time. For the PEP case, the set of NLP variables are:

$$\boldsymbol{\pi} = (\boldsymbol{x}_{m=(0,M)}, \boldsymbol{p}). \tag{15}$$

Next, the governing ODEs (4b) or (5b) are marched in time within each shooting segment Ω^m, starting from the initial conditions provided by the values of the states $\boldsymbol{x}_m$ at the left boundary of the segment. The effect of the forward integration is to generate a discrete time history of states within Ω^m, which we label $\boldsymbol{x}_i^m$, $i = (1, N^m)$, where N^m is the number of steps taken in that segment. The last value of this sequence is named $\bar{\boldsymbol{x}}_{m+1} = \boldsymbol{x}_{N^m}^m$, and represents the new value of the state variables at the right boundary of the shooting segment. Segments are then glued together by imposing the following equality constraints

$$\boldsymbol{x}_m - \bar{\boldsymbol{x}}_m = \boldsymbol{0}, \qquad m = (2, M). \tag{16}$$

In the direct multiple shooting case, the cost of equation (4a) or (5a) is discretized in terms of $\boldsymbol{\pi}$ and evaluated using the segment time histories $\boldsymbol{x}_i^m$; this yields the discrete cost J^{NLP} of equation (9a). Next, the gluing conditions (16) are used to express the set of NLP equality constraints appearing in equations (9b). All other problem constraints and bounds, equations (4d) or (5d), are expressed in terms of the NLP variables $\boldsymbol{\pi}$ and become the NLP inequality constraints of equations (9b).

Multiple shooting segments are introduced for reducing the potential instabilities encountered using a single shooting arc. With a shooting method, adaptive step procedures can be used to advance the solution in time within each segment, so that highly accurate solutions can be obtained. The size of the NLP problem clearly depends on the number of segments but, since the segments are consecutive, the resulting problem is sparse. Even more importantly, the size of the NLP problem does not depend on the number of internal steps taken in each segment. This is crucial if the vehicle model $\mathcal{M}$ has fast dynamic components which need small time steps to be resolved. Hence, shooting-based methods can handle problems with higher values of the complexity C than in the case of transcription methods.

5.3.1. Direct single-multiple shooting

For problem in rotorcraft flight mechanics, we have observed that the satisfaction of the multiple shooting gluing constraints can be particularly difficult and usually ends up dominating the problem. This is not surprising, since the rotor generates most of the aerodynamic forces acting on the vehicle and even small variations in its states may imply large variations in the resulting forces, which hinders the satisfaction of the gluing constraints. We have found that these problems can be alleviated by using multi-time scale arguments [9]. In fact, the rotor states (both structural and aerodynamic) are significantly faster than the flight mechanics ones. Thus, since the multiple shooting treatment of these fast states is the main cause of the two aforementioned issues, i.e. raise in computational cost and difficulty in satisfying gluing constraints, one can think of treating slow and fast scales using different methods.

The procedure uses a multiple shooting approach for the slow states. This is crucial, since with single shooting small changes early in the trajectory can produce dramatic effects at the end of it; clearly, the problem is exacerbated when analyzing unstable systems, which is often the case when considering rotorcraft vehicles. Hence, the multiple shooting treatment of slow scales avoids the blow up of the solution. On the contrary, the fast scales are treated using a single shooting approach, as depicted in Figure 3. This does not compromise the robustness of the procedure, since fast scales will not diverge if slow ones do not; hence, the stabilizing effect produced by the multiple shooting treatment of slow scales is felt also at the level of the fast ones.

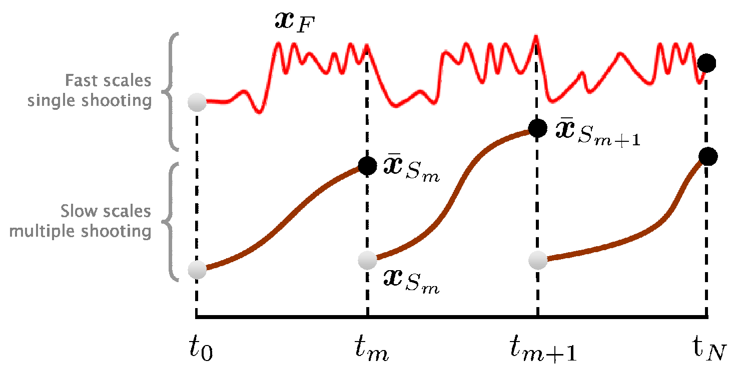

Figure 3. Hybrid single-multiple shooting approach.

With such a hybrid single-multiple shooting approach, the size of the resulting NLP problem is substantially reduced and so is the total computational cost. Furthermore, there are no gluing constraints to be enforced for the fast rotor states, since only the slow states need to be glued together at the shooting interfaces. This has the effect of greatly increasing the robustness of the procedure, and the convergence speed. The detailed mathematical formulation of the single-multiple shooting method is given in reference [9].

5.3.2. State dependent constraints

One possible issue with the multiple shooting approach is the solution of MOCP problems with state dependent constraints. In fact, only state variables at the shooting segment boundaries enter into the definition of the optimization unknowns (see the definition of $\boldsymbol{\pi}$ given in equation (14)). Therefore, one can not directly impose constraints or bounds on the values assumed by the states within the shooting segments.

A simple but approximate way of dealing with this problem is to break the segment where violations are detected into small ones, and continue with refinement until no more internal violations are found or when they become smaller than some given acceptable tolerance.

6. Software architecture

We have exploited the similarities between MOCPs and PEPs for developing a general purpose software program for rotorcraft applications named STOP (System Identification and Trajectory Optimization Program). The architecture of the code is shown in Figure 4.

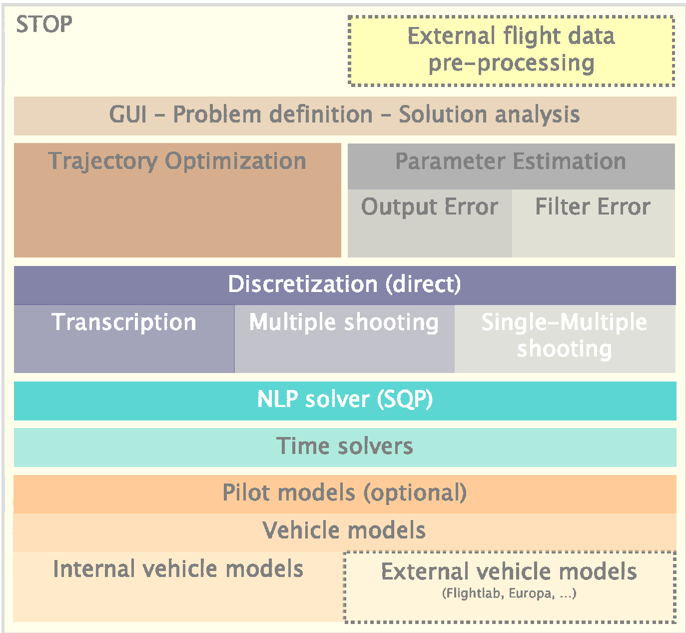

Figure 4. Architecture of the STOP code.

A graphical user interface supports the definition of MOCPs and PEPs. Both classes of problems share the same direct optimization strategies, which include direct transcription and direct multiple and hybrid single-multiple shooting so as to support a broad range of vehicle models of varying complexity. This layer generates a NLP problem, which is solved using a SQP algorithm. Time marching can be based either on algorithms available in external vehicle models, or with built-in explicit or implicit time solvers. The vehicle models include an optional layer that models the pilot, which is useful in certain MOCP applications for computing maneuvers considering pilot-in-the-loop effects [10]. The vehicle itself can be modeled using an internal model, or by external simulators through a generic interface which supports all necessary operations.

7. Applications and results

7.1. MOCP: Category-A continued take-off after engine failure for a tilt-rotor

We consider the vertical take-off from a confined area for the tilt-rotor ERICA [11] under Category-A certification requirements, as described in [12]. The vehicle model is implemented with the `FLIGHTLAB` [13] code, and it is based on three-dimensional rigid body dynamics, with two four-bladed gimballed rotors with Peters-He dynamic inflow.

The optimization cost function (4a) takes the form

$$J = h(T) + \frac{1}{T}\int_0^T \dot{\boldsymbol{u}} \cdot \boldsymbol{W}\dot{\boldsymbol{u}}\,\mathrm{d}t, \tag{17}$$

where $h(T)$ is the altitude loss at the lowest point in the maneuver, measured from the initial trimmed hovering condition. The maneuver definition is completed by the exit conditions, which specify a positive climb gradient and force the rotor speed back to its nominal value; furthermore, bounds on the vehicle control inputs and their rates are enforced throughout the maneuver.

A series of snapshots from the computed maneuver are shown in Figure 5.

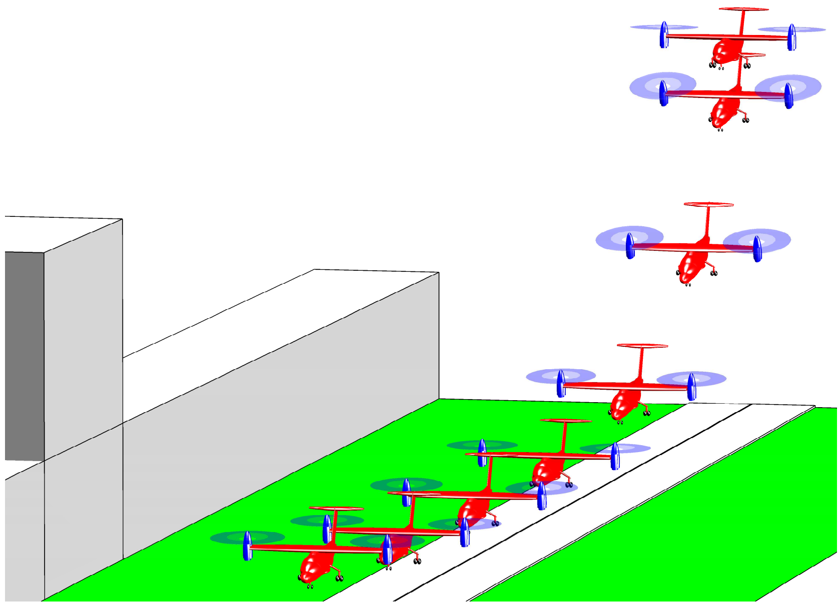

Figure 5. Continued take-off of a tilt-rotor, after loss of power from one of the engines.

The computed control time histories are given in Figure 6. Notice how a quick stick-forward input is applied in the initial phase of the maneuver to promote a fast forward acceleration of the vehicle to gain speed and hence reduce the required power, as expected.

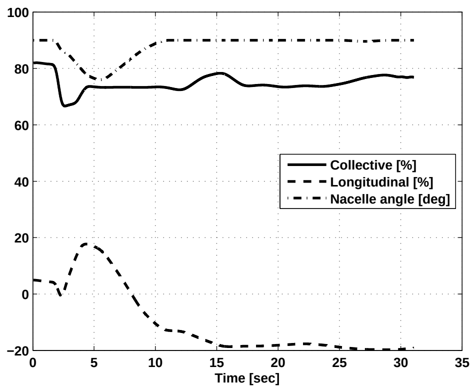

Figure 6. Category-A continued take-off: time history of control inputs.

7.2. PEP: identification of a small autonomous vehicle using simulation data

We consider the problem of parameter estimation for a small-size unmanned rotorcraft vehicle; full details on this problem are given in reference [14]. Model equations are derived from first principles based on the three-dimensional rigid body equations of motion. Rotor forces and moments are computed analytically by combining actuator disk and blade element theory, considering a uniform inflow (see reference [15]). The model also include a Bell-Hiller stabilizing flybar.

We perform a virtual experiment considering the vehicle trimmed in forward flight at a very low advance ratio, $\mu = 0.04$. Data used for estimation is generated by integrating the equations of motion of the vehicle under the action of a LQR regulator, with successive excitation of the main rotor longitudinal cyclic, main rotor lateral cyclic, main rotor collective, and tail rotor collective using a 1-1-2-3 sequence [3]. The generated output signals are corrupted by adding a white noise.

Figure 7 shows some of the typical results obtained for this problem with the Maximum Likelihood method. The time histories show a very good agreement between the simulated flight data and the computed response for converged parameter estimates.

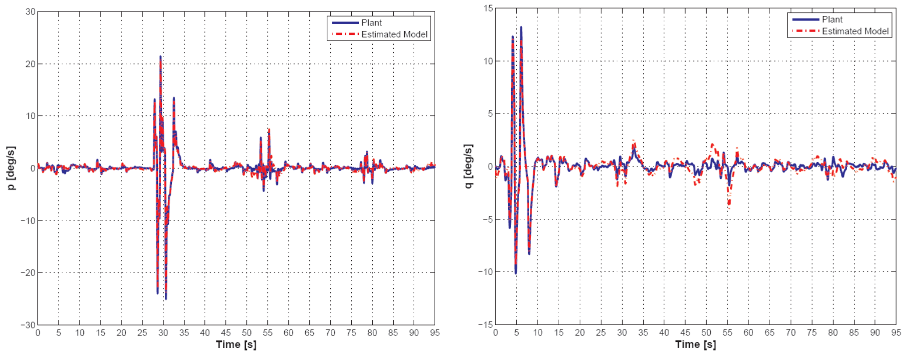

Figure 7. Left: roll rate; right: pitch rate. Solid line: simulated flight data. Dash-dotted line: computed response for converged parameter estimates.

8. Conclusions

We have formulated a unified approach to trajectory optimization and parameter estimation in vehicle dynamics. The unified approach has enabled the design and implementation of a common computer program for the solution of both classes of problems, reducing the time and effort of software development, debugging, validation and maintenance.

The common thread between the solution of the two problem classes is the discretization in the temporal domain, which leads in all cases to standard algebraic constrained optimization problems. We have argued that such discretization must be carried out in a way which reflects the level of complexity (as defined in this work) of the problem. More specifically, experience has shown that the direct transcription method is the more robust of the approaches, but can typically only be applied to problems of low to moderate complexity. On the other hand, problems of higher complexity should be treated using shooting-based methods, which however need some care in their usage regarding the presence of possible instabilities, the satisfaction of the gluing constraints and the enforcement of state inequalities. In all cases, scaling of the unknowns and of the governing equations eases the problem by improving its conditioning.

The approach described herein lends itself to the use of black-box models of the vehicle, which is often useful in practical applications, and in fact the software has been successfully interfaced with existing industrial rotorcraft codes.

Future efforts will concentrate on further expanding the features of the code and on its extensive use and validation on cases of industrial relevance.

Acknowledgements

This research is funded by Agusta-Westland, with Marco Cicalè serving as technical monitor. The authors gratefully acknowledge the contribution of A. Maffezzoli, D. Muffo, A. Ragazzi, F. Scorcelletti and N. William.

References

[1] C.L. Bottasso, C.-S. Chang, A. Croce, D. Leonello, and L. Riviello. Adaptive planning and tracking of trajectories for the simulation of maneuvers with multibody models. *Computer Methods in Applied Mechanics and Engineering*, 195:7052–7072, 2006.

[2] C.L. Bottasso. Solution procedures for maneuvering multibody dynamics problems for vehicle models of varying complexity. In C.L. Bottasso, editor, *Multibody Dynamics — Computational Methods and Applications*, Computational Methods in Applied Sciences. Springer-Verlag, Dordrecht, The Netherlands, 2008. ISBN 978-1-4020-8828-5.

[3] R.V. Jategaonkar. *Flight Vehicle System Identification. A Time Domain Approach.* AIAA Progress in Astronautics Aeronautics, Reston, VA, 2006.

[4] C.L. Bottasso, A. Croce, D. Leonello, and L. Riviello. Optimization of critical trajectories for rotorcraft vehicles. *Journal of the American Helicopter Society*, 50:165–177, 2005.

[5] C.L. Bottasso, F. Luraghi, and G. Maisano. Time-domain parameter estimation for first-principle rotorcraft models using recursive and batch procedures: Formulation and preliminary results. Technical Report DIA-SR 09-05, Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, 2009.

[6] U. Forssell and L. Ljung. Closed-loop identification revisited. *Automatica*, 35:1215–1241, 1999.

[7] J.T. Betts. *Practical Methods for Optimal Control Using Non-Linear Programming.* SIAM, Philadelphia, 2006.

[8] A. Barclay, P.E. Gill, and J.B. Rosen. SQP methods and their application to numerical optimal control. Report NA 97-3, Department of Mathematics, University of California, San Diego, 1997.

[9] C.L. Bottasso and G. Maisano. A new single-multiple shooting method for trajectory optimization of complex rotorcraft models. 2009. In preparation for submission.

[10] C.L. Bottasso, G. Maisano, and F. Scorcelletti. Trajectory optimization of rotorcraft including pilot models, with applications to ADS-33 MTEs, Cat-A procedures and engine off landings. In *American Helicopter Society 65th Annual Forum and Technology Display*, Gaylord Texan Convention Center, Grapevine, TX, USA, May 27–29 2009.

[11] F. Nannoni, G. Giancamilli, and M. Cicalè. ERICA: the European advanced tiltrotor. In *27th European Rotorcraft Forum*, September 11–14 2001.

[12] Anonymous. Advisory circular 29-2C, certification of transport category rotorcraft. Technical report, Federal Aviation Administration, Department of Transportation, 1999.

[13] Advanced Rotorcraft Technology Inc., 1685 Plymouth Street, Suite 250, Mountain View, CA 94043, USA, `http://www.flightlab.com`.

[14] A. Maffezzoli. Procedures for the estimation of model parameters for a small rotorcraft UAV. Master's thesis, Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, 2009.

[15] R.W. Prouty. *Helicopter performance, stability, and control.* R. E. Krieger Publishing Co., Malabar, FL, 1990.