

# Adaptive Reference-Augmented Predictive Control, with Application to the Reflexive Control of Unmanned Rotorcraft Vehicles

Carlo L. Bottasso, Roberto Nicasastro, Barbara Savini, and Luca Riviello

C.L. Bottasso, R. Nicasastro, B. Savini and L. Riviello are with the Department of Aerospace Engineering, Politecnico di Milano, Milano, 20156 Italy.

Paper presented at the 2007 American Control Conference, New York, NY, USA, July 11–13, 2007, with the title 'Adaptive Reference-Augmented Predictive Control'.

# Adaptive Reference-Augmented Predictive Control, with Application to the Reflexive Control of Unmanned Rotorcraft Vehicles

## Abstract

We describe a novel adaptive non-linear model predictive controller which is based on the idea of neural-augmentation of reference elements, both at the level of the reduced model and at the level of the control action. The new methodology is primarily motivated by the desire to consistently incorporate existing legacy modeling and control techniques into an adaptive non-linear, yet real-time-capable, control framework.

At the level of the model, a reference model is augmented using an adaptive neural element. Kalman filtering is used for identifying on-line the free parameters of the neural network, with the goal of maximizing the prediction fidelity of the model with respect to the plant. At the level of the control strategy, a reference solution is augmented using a second adaptive neural element. The augmenting control network is trained on-line for correcting the reference control action and promoting it to the solution of the underlying non-linear model predictive problem.

The resulting neural-augmented control strategy is non-linear, yet it is real-time capable in the sense that it requires a fixed number of operations at each step. Furthermore, it substantially eases the adaption process, since the neural elements must only be trained to capture the defects of the reference legacy elements, which is small if such reference elements are adequate.

The proposed procedures are demonstrated in a virtual environment with the help of the classical model problem of the double inverted-pendulum, and with the more challenging reflexive control of an autonomous helicopter.

## Index Terms

Predictive control, adaptive control, neural networks, mobile robots, real time systems.

## NOMENCLATURE

$J$	Cost function
$T_p$	Length of the prediction window
$T_s$	Length of the steering window
$m$	Moment resultant
$p_c$	Free control parameters
$p_m$	Free model parameters
$r$	Position vector
$s$	Force resultant
$u$	Control inputs
$v$	Linear velocity vector

$x$	States
$y$	Outputs
$\alpha$	Direction cosine matrix
$\lambda$	Co-states
$\omega$	Angular velocity vector
$t$	Time
$t_0$	Initial time
$a$	Scalar
$\mathbf{a}$	Vector
$\mathbf{A}$	Matrix
$\mathbf{a}, \mathbf{b}$	Matrix of partial derivatives $[\partial a_i / \partial b_j]$
$(\cdot)^*$	Goal to-be-tracked quantity
$(\cdot)^T$	Transpose
$(\cdot)^{\mathcal{A}}$	Vector components in the $\mathcal{A}$ triad
$(\cdot)_c$	Control element quantity
$(\cdot)_m$	Reduced model quantity
$(\cdot)_p$	Parametric quantity
$(\cdot)_{\text{ref}}$	Reference quantity
$\dot{(\cdot)}$	Derivative with respect to time, $d \cdot / dt$
$\widetilde{(\cdot)}$	Plant quantity

## I. INTRODUCTION AND MOTIVATION

Model predictive control is based on the on-line solution of an optimal control problem, which seeks to minimize some appropriate cost function [1]. The problem is formulated in terms of a reduced model of the plant, that allows one to predict the future response of the system on a given time horizon. The computed control actions are then used to steer the plant on a short time window. At the next time step, the open-loop optimal control problem is solved again on a prediction window shifted forward in time (receding horizon control).

If the reduced model is linear, the cost is quadratic and the prediction horizon of infinite length, one has the well known linear-quadratic-regulator (LQR) problem. Optimal state- or output-feedback gains can be pre-computed by numerically solving off-line suitable non-linear matrix problems [2], [3]. When the problem is non-linear (non-linear model predictive (NMP) control), the solution can not in general be pre-computed and has to be obtained on-line by numerical means on a finite horizon (see [4] and references therein for stability results in this case). Although there are efficient methods to solve this class of optimal control problems [5], these are typically not yet fast enough for real-time applications. Even more importantly, such algorithms are iterative by nature, and hence it is difficult if not impossible to guarantee the satisfaction of a hard real-time schedule, since the number of operations required for computing the solution at each step is not known a priori.

It is clear that the applicability of non-linear model predictive controllers to a variety of control problems of industrial interest is crucially dependent on:

- 1) The ability to develop good non-linear reduced models of the plant, i.e. on the ability of the reduced model to perform faithful predictions of the plant response; notice that faithfulness of the model to the plant is the key property for proving non-linear stability of NMP control [1], the finite horizon length issue being a lesser concern [4].
- 2) The ability to solve on-line the resulting open-loop optimal control problem in real-time, i.e. fast enough and with a fixed number of operations.

In this paper we propose an approach which tries to address both of these concerns through the concept of *augmentation of reference* solutions, both at the level of the reduced model and at the level of the control actions. For this reason, the proposed approach is here termed Reference-Augmented Predictive Control (RAPC).

For the reduced model, the idea is as follows. A problem-dependent *reference approximate model* of the plant is augmented with an adaptive element, a neural network in the present case, which is trained on-line to capture the “defect” between the solution of the approximate reference model and the real response of the plant [6].

For the control actions, the procedure is as follows. At each time step, a *reference control solution* is provided by a suitable control method, which is the LQR in this paper but could be any controller that works well for the problem at hand and guarantees a minimum level of acceptable performance. These approximate control actions are now *augmented* by a second adaptive element, which is trained on-line to approximate the “defect” between the reference and the NMP solutions. The adaptive element can be chosen as any suitable parametric function, as for example a neural network in the case of the present paper. Previous work in this class of controllers is described in [7].

A major motivation behind RAPC is the observation that, in all control application fields, there is a wealth of knowledge and legacy methodologies which are known to perform reasonably well for a given problem. In the design of new advanced control techniques and in the search for better performance and improved capabilities, it is clearly undesirable and wasteful to neglect valuable available prior knowledge. RAPC allows one to exploit available legacy methods, embedding them in a non-linear model predictive control framework. For example, in the field of rotorcraft flight mechanics, there is vast experience on the modeling of the vehicle using combined blade element theory and inflow modeling techniques [8], [9]. Yet, given the complexity of rotorcraft aeromechanics and the necessity to limit the numerical cost of flight mechanics models, there will always be physical processes which are not captured or not resolved well by the available models. The reduced model augmentation of RAPC provides a way to improve an existing flight mechanics model, by trying to approximate the effects of unmodeled or unresolved physics on flight mechanics. Similarly, there is a wealth of experience in the design of flight control systems using a variety of linear control techniques [2]. For highly non-linear systems such as helicopters, the control augmentation of RAPC provides a consistent way to design a non-linear controller building on linear ones which are known to provide a minimum level of performance about certain linearized operating conditions.

A part from the motivations above, the present approach has the following highlights:

- It is adaptive at the level of the model and at the level of the control laws. The two are coupled, but they can be trained independently on-line, which eases this process; this is in contrast with other approaches, such as for example adaptive critic methods [10], where the adaptive elements are intimately coupled and must be trained simultaneously. Through the proposed procedure, we are able to separate the control and model identification problems using two different sets of adaptive elements to accomplish the two different tasks. Different training strategies can be used, as well as different update frequencies. For example, in this work the model network parameters are identified using a Kalman filter, while the control network parameters are identified using a steepest descent approach.
- Both controller and reduced model are fully non-linear, yet the numerical implementation of the adaptive controller requires the exact same number of operations at each time step. This means that one can guarantee the satisfaction of a hard real-time schedule (clearly, assuming that sufficient computing resources are available).
- Since the procedure is based on reference solutions, a minimum level of performance is guaranteed even at the beginning of the process and before any adaption has taken place. Since reasonable predictions and reasonable control actions are generated by the reference elements, there is no need for pre-training of the networks, a major hurdle in other network-based or adaptive control approaches [10]. In fact, all examples presented in this paper were successfully solved without any pre-training, by simply starting from small random numbers for the unknown parameters of the adaptive elements.
- If the reference solutions are in the neighborhood of the true ones, the defect that the networks are in charge of approximating is small, and this substantially eases the identification process. This is in contrast with other approaches which delegate to the networks the approximation of the whole solution [11].
- Adaptivity of controllers is often seen as a topic of academic interest, but of little applicability in real life, such as to flight control systems, for which one must be able to guarantee a certain minimum level of performance throughout the operating envelope of the controlled system. Although we do not address this problem in the present paper, we believe that RAPC gives a way to address this issue: by monitoring the magnitude of the defects which are identified on-line, i.e. by verifying how “far” we are going from the reference solutions, we hypothesize that appropriate strategies can be designed for detecting anomalous behaviors of the adaptive processes and for system diagnostics.

The paper is organized as follows. Section II formulates the problem of non-linear model predictive control, and introduces the relevant notation. Next, Section III describes the adaptive reduced model identification procedure based on the augmentation of a reference reduced model with the equation defect. Section IV describes the augmentation of a reference control strategy, so as to approximate the solution of the underlying non-linear model predictive control problem in a way which is adaptive and real-time capable. The key contribution of this section is the analysis of the functional dependence of the minimizing optimal control on the problem data, which forms the basis for the proposed approximation. Finally, Section V presents results and applications of the new controller. The methodology is first tested on the classical double inverted-pendulum model problem, and then it is demonstrated

for the reflexive control of a high performance autonomous rotorcraft.

## II. NON-LINEAR MODEL PREDICTIVE CONTROL

The basic principle of non-linear model predictive control is illustrated in Fig. 1. At the current time  $t_0$ , a non-linear reduced model  $\mathcal{M}$  is used for predicting the future behavior of the plant  $\widetilde{\mathcal{M}}$  in terms of its states  $\mathbf{x} \in \mathbb{R}^n$  under the action of the control inputs  $\mathbf{u} \in \mathbb{R}^m$ . An open-loop optimal control problem is solved for the reduced model on a finite horizon (the prediction window  $[t_0, t_0 + T_p]$ ). The cost of this optimization problem is in general

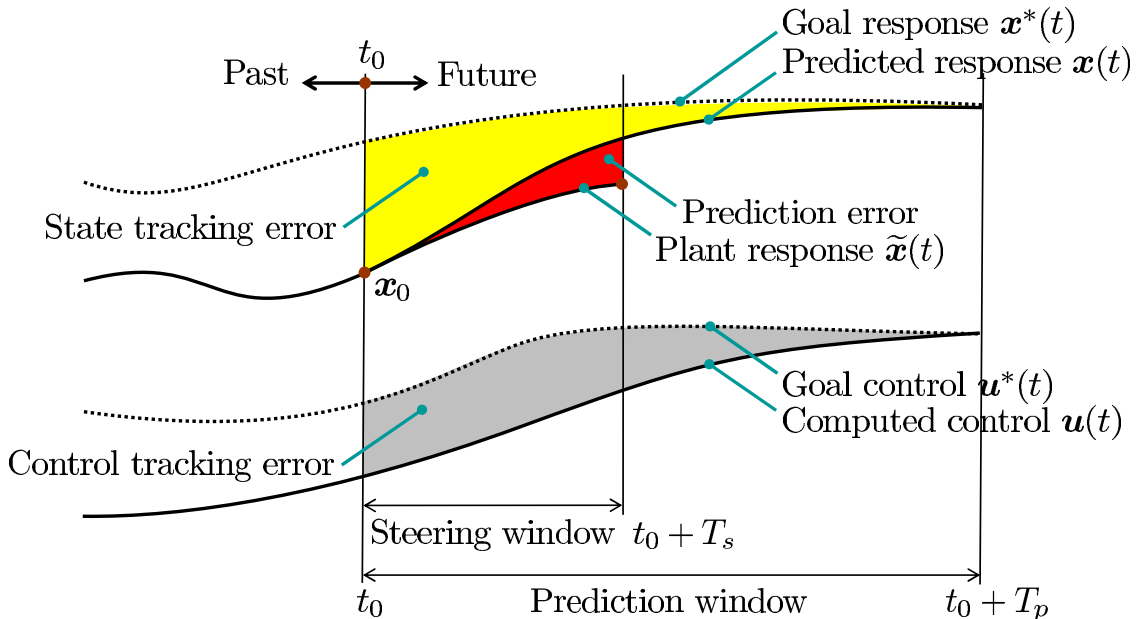


Fig. 1. Model predictive control

The controls computed by the optimizer are now used for steering the plant, but only on a short time horizon up to time  $t_0 + T_s$ , as soon as a new measurement becomes available. In fact, due to the presence of disturbances and the inevitable mismatch between reduced model and plant, the actual response of the system  $\tilde{\mathbf{x}}(t)$  will drift away from the predicted one  $\mathbf{x}(t)$ . Once the plant has reached the end of the steering window  $[t_0, t_0 + T_s]$  under the action of the computed control inputs, the model predictive optimization problem is solved again, looking ahead in the future over the prediction horizon shifted forward in time. This procedure results in a feedback, receding horizon approach.

The future control actions are computed by solving the following open-loop optimal control problem:

$$\min_{\mathbf{u}, \mathbf{x}, \mathbf{y}} J = \int_{t_0}^{t_0 + T_p} L(\mathbf{y}, \mathbf{u}) dt, \quad (1a)$$

$$\text{s.t.: } \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = 0, \quad t \in [t_0, t_0 + T_p], \quad (1b)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad (1c)$$

$$\mathbf{y} = \mathbf{g}(\mathbf{x}), \quad t \in [t_0, t_0 + T_p]. \quad (1d)$$

The tracking cost is noted  $J$ , whose integrand  $L(\cdot, \cdot) : \mathbb{R}^l \times \mathbb{R}^m \rightarrow \mathbb{R}_{\geq 0}$  is defined here as

$$L(\mathbf{y}, \mathbf{u}) = (\mathbf{y} - \mathbf{y}^*)^T \mathbf{Q}(\mathbf{y} - \mathbf{y}^*) + (\mathbf{u} - \mathbf{u}^*)^T \mathbf{R}(\mathbf{u} - \mathbf{u}^*), \quad (2)$$

where  $\mathbf{y}^*$  and  $\mathbf{u}^*$  are desired goal outputs and goal controls, respectively, while  $\mathbf{Q} \geq 0$  and  $\mathbf{R} \geq 0$  are symmetric design matrices of coefficients which weight the output and control tracking errors. The plant non-linear reduced model is expressed by (1b), where  $\mathbf{f}(\cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ , while (1c) provides the initial conditions at the beginning of the prediction window. Finally, (1d) defines the system outputs  $\mathbf{y}$ , where  $\mathbf{g}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^l$ .

### III. REFERENCE-AUGMENTED REDUCED MODEL IDENTIFICATION

The tracking performance of the controller (1) and, ultimately, its non-linear stability, critically hinge on the fidelity of the reduced model to the plant [1]. More precisely, consider both the plant  $\widetilde{\mathcal{M}}$  and its reduced model  $\mathcal{M}$  as expressed by (1b) to be subjected to the same given input signal  $\mathbf{u} : t \mapsto \mathbf{u}(t)$  for  $t \in [t_0, t_0 + T_p]$ , starting from the same initial conditions  $\mathbf{x}_0$  at time  $t_0$ . The ensuing response of the plant can be written as

$$\widetilde{\mathbf{x}}(t) = \widetilde{\phi}_u(\mathbf{x}_0, t), \quad (3)$$

where  $\widetilde{\phi}_u(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  is the plant state flow, a non-linear time dependent function which, for a given control input signal  $\mathbf{u}$ , maps the initial conditions  $\mathbf{x}_0$  into the state response  $\widetilde{\mathbf{x}}(t)$ ; similarly, the response of the reduced model will be

$$\mathbf{x}(t) = \phi_u(\mathbf{x}_0, t), \quad (4)$$

$\phi_u(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  being the corresponding reduced model state flow. Ideally, plant and reduced model should produce the same response, i.e.

$$\mathbf{x}(t) = \widetilde{\mathbf{x}}(t) \quad \forall t \in [t_0, t_0 + T_p]. \quad (5)$$

When this condition holds for generic initial conditions  $\mathbf{x}_0$  and control input signals  $\mathbf{u}$  within the operational envelope of the plant, we say that the reduced model has perfect prediction capabilities.

Unfortunately, in all cases of practical interest the functional form of a reduced model (1b) with perfect prediction capabilities is unknown. To address this issue, consider first a given suitable reduced model, developed on the knowledge of the physical processes governing the response of the plant. Such model is termed here a *reference reduced model* and it can be written in general as

$$\mathbf{f}_{\text{ref}}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = 0, \quad (6)$$

where  $\mathbf{f}_{\text{ref}}(\cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ . When integrated with initial conditions  $\mathbf{x}_0$  under the action of a given input signal  $\mathbf{u}$ , this reduced model produces a response  $\mathbf{x}(t) \neq \widetilde{\mathbf{x}}(t)$ , which differs from the actual one due to the modeling errors of (6) with respect to the real plant dynamics.

Another way of looking at this problem is to say that the actual plant response  $\tilde{\mathbf{x}}(t)$  associated with  $\mathbf{u}(t)$  and  $\mathbf{x}_0$  does not satisfy the reduced model equations, or, in symbols

$$\mathbf{f}_{\text{ref}}(\dot{\tilde{\mathbf{x}}}, \tilde{\mathbf{x}}, \mathbf{u}) \neq 0. \quad (7)$$

However, it is always possible to write

$$\mathbf{f}_{\text{ref}}(\dot{\tilde{\mathbf{x}}}, \tilde{\mathbf{x}}, \mathbf{u}) = \mathbf{d}(\tilde{\mathbf{x}}, \mathbf{u}), \quad (8)$$

where  $\mathbf{d}(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is termed the *model equation defect function*. In other words, the defect is that unknown function which, augmenting the reference element, ensures that the plant response satisfies the reduced model equations, thereby generating a perfect matching between the states of the reduced model and the corresponding states of the plant, when the two are subjected to the same inputs starting from the same initial conditions. If we knew function  $\mathbf{d}$ , we could write the reduced model of problem (1) as

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = \mathbf{f}_{\text{ref}}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) - \mathbf{d}(\mathbf{x}, \mathbf{u}) = 0, \quad (9)$$

and the prediction of the system states  $\mathbf{x}$  would exactly match the corresponding values of the plant. Notice that one might include in the defect also the dependence on the derivatives of the states up to some suitable order, for example for modeling delays or other effects not included in the reference model.

Although  $\mathbf{d}$  is unknown, under assumptions of sufficient smoothness it is possible to construct an approximation of it using a parametric function  $\mathbf{d}_p$ , approximation which can be written as

$$\mathbf{d}(\mathbf{x}, \mathbf{u}) = \mathbf{d}_p(\mathbf{x}, \mathbf{u}, \mathbf{p}_m) + \boldsymbol{\varepsilon}_m, \quad (10)$$

where  $\mathbf{p}_m \in \mathbb{R}^{p_m}$  are free model parameters to be identified in order to minimize the approximation error  $\boldsymbol{\varepsilon}_m$ , and where  $\mathbf{d}_p(\cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{p_m} \rightarrow \mathbb{R}^n$ .

In this work we use as parametric function a single-hidden-layer neural network with  $n_h$  neurons, which has the following functional form

$$\mathbf{d}_p(\mathbf{x}, \mathbf{u}, \mathbf{p}_m) = \mathbf{W}_m^T \boldsymbol{\sigma}(\mathbf{V}_m^T \mathbf{i}_m + \mathbf{a}_m) + \mathbf{b}_m, \quad (11)$$

where  $\mathbf{i}_m = (\mathbf{x}^T, \mathbf{u}^T)^T \in \mathbb{R}^{n+m}$  is the network input vector,  $\mathbf{W}_m \in \mathbb{R}^{n_h \times n}$  and  $\mathbf{V}_m \in \mathbb{R}^{(n+m) \times n_h}$  are matrices of synaptic weights,  $\mathbf{a}_m \in \mathbb{R}^{n_h}$  and  $\mathbf{b}_m \in \mathbb{R}^n$  are the network biases,  $\boldsymbol{\sigma}(\bullet) = (\dots, \sigma(\cdot), \dots)^T$  is the vector-valued function of sigmoid activation functions  $\sigma(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ . The model parameters are defined as the network weights and biases, i.e.

$$\mathbf{p}_m = (\dots, W_{m_{ij}}, \dots, V_{m_{ij}}, \dots, a_{m_i}, \dots, b_{m_i}, \dots)^T, \quad (12)$$

with  $\mathbf{p}_m \in \mathbb{R}^{p_m}$ ,  $p_m = n_h(2n + m + 1) + n$ . The universal approximation property of neural networks [12] ensures that the functional reconstruction error  $\boldsymbol{\varepsilon}_m$  in (10) can be bounded as  $\|\boldsymbol{\varepsilon}_m\| \leq C_\varepsilon$  for any  $C_\varepsilon > 0$ , for some appropriately large number of hidden neurons  $n_h$ , assuming that  $\mathbf{d}$  is sufficiently smooth.

With these choices, the reduced model of (1b) can be written as

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}, \mathbf{p}_m) = \mathbf{f}_{\text{ref}}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) - \mathbf{d}_p(\mathbf{x}, \mathbf{u}, \mathbf{p}_m) = 0. \quad (13)$$

There are several ways to identify on-line the free parameters  $\mathbf{p}_m$  of this reduced model. In the present work we use an extended Kalman filter. The dynamics equations for the model parameters are simply

$$\dot{\mathbf{p}}_m = \mathbf{w}_{pn}, \quad (14)$$

where  $\mathbf{w}_{pn}$  is the white process noise, and the ‘‘measurement’’ equation is

$$\mathbf{z} = \mathbf{f}_{\text{ref}}(\dot{\tilde{\mathbf{x}}}, \tilde{\mathbf{x}}, \mathbf{u}) - \mathbf{d}_p(\tilde{\mathbf{x}}, \mathbf{u}, \mathbf{p}_m) + \mathbf{w}_{mn}, \quad (15)$$

where  $\mathbf{w}_{mn}$  is the white measurement noise.

#### IV. REFERENCE-AUGMENTED PREDICTIVE CONTROL

##### A. Functional Form of the Optimal Control Solution

In this section we develop an analysis of problem (1), in order to determine the functional dependence of the minimizing optimal control signal. This will then be used later on for developing suitable approximations.

The optimal control problem (1) can be transformed into a boundary value problem in the temporal domain  $[t_0, t_0 + T_p]$ . To this end, first eliminate the outputs from (1a) using (1d), and consider the constraints (1b) written in the parametric form (13). Next, augment the cost  $J$  of (1a) with the constraints through a set of Lagrange multipliers  $\boldsymbol{\lambda} \in \mathbb{R}^n$  (co-states). This gives

$$\hat{J} = \int_{t_0}^{t_0+T_p} (L(\mathbf{y}(\mathbf{x}), \mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}, \mathbf{p}_m)) dt, \quad (16)$$

where  $\hat{J}$  is the augmented cost. Finally, the stationarity of the augmented cost is enforced, which yields the following system of coupled ordinary differential equations with their boundary conditions:

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}, \mathbf{p}_m) = 0, \quad t \in [t_0, t_0 + T_p], \quad (17a)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad (17b)$$

$$-\frac{d(\mathbf{f}_{,\dot{\mathbf{x}}}^T \boldsymbol{\lambda})}{dt} + \mathbf{f}_{,\mathbf{x}}^T \boldsymbol{\lambda} + \mathbf{y}_{,\mathbf{x}}^T L_{,\mathbf{y}} = 0, \quad t \in [t_0, t_0 + T_p], \quad (17c)$$

$$\boldsymbol{\lambda}(t_0 + T_p) = 0, \quad (17d)$$

$$L_{,\mathbf{u}} + \mathbf{f}_{,\mathbf{u}}^T \boldsymbol{\lambda} = 0, \quad t \in [t_0, t_0 + T_p]. \quad (17e)$$

The system can be partitioned into three main sets of equations:

- 1) Equation (17a) represents the reduced model dynamics, with initial conditions provided by (17b). This initial value problem can not in general be solved in closed form in terms of the state function  $\mathbf{x}(t)$  for a generic input signal  $\mathbf{u}$  and initial conditions  $\mathbf{x}_0$ , especially since we are not making any specific assumptions on the functional form of (17a). However, by inspection of the equations, we can at least write symbolically

$$\mathbf{x}(t) = \boldsymbol{\phi}_u(\mathbf{x}_0, t), \quad (18)$$

for  $t \in [t_0, t_0 + T_p]$ , which asserts that the state function depends, for a given input function  $\mathbf{u}(t)$ , on the initial conditions and time through the operator  $\phi_u(\cdot, \cdot)$  of (4).

- 2) Equation (17c) are the adjoint equations in the co-states  $\boldsymbol{\lambda}$ , with final conditions given by (17d). Here again, it is not possible in general to solve this final value problem in closed form in terms of the co-states, but by inspection of the equations and considering (2), we can write symbolically

$$\boldsymbol{\lambda}(t) = \boldsymbol{\theta}_{x,u}(\mathbf{y}^*(t), t), \quad (19)$$

i.e. the co-state function depends, for given input  $\mathbf{u}(t)$  and state  $\mathbf{x}(t)$  functions, on the goal output function and time through a co-state flow function  $\boldsymbol{\theta}_{x,u}(\cdot, \cdot) : \mathbb{R}^l \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ .

- 3) Finally, (17e) are the transversality conditions, which can be solved in terms of the control function. Similarly to the previous cases, by inspection of the equation we can write symbolically

$$\mathbf{u}(t) = \boldsymbol{\psi}_{x,\lambda}(\mathbf{u}^*(t), t), \quad (20)$$

i.e. the control function depends, for given state  $\mathbf{x}(t)$  and co-state  $\boldsymbol{\lambda}(t)$  functions, on the goal input function and time through some operator  $\boldsymbol{\psi}_{x,\lambda}(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ .

Inserting now (18) and (19) into (20), we can write the optimal control function  $\mathbf{u}(t)$  which solves problem (17e) as

$$\mathbf{u}(t) = \boldsymbol{\chi}(\mathbf{x}_0, \mathbf{y}^*(t), \mathbf{u}^*(t), t), \quad (21)$$

where  $\boldsymbol{\chi}(\cdot, \cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$  is an unknown non-linear time dependent operator. This shows that, given a reduced model and optimization cost, the optimal control is a sole function of the initial conditions, of the goal output and goal control functions, and of time.

### B. Systems with Symmetries: the Case of Flight Mechanics

The expression given in (21) for the minimizing optimal control which solves problem (1) can be detailed further for systems possessing symmetries. Specifically, in this paper we analyze the case where model  $\mathcal{M}$  represents a flying vehicle (see for example [13] for a treatment in the context of Lie group theory).

First, consider an inertial frame of reference denoted by a North-East-Down (NED) triad of unit vectors  $\mathcal{E} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ , and a body-attached local frame of reference denoted by a triad of unit vectors  $\mathcal{B} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$  with origin at the center of gravity of the vehicle. In the following, the notation  $(\cdot)^{\mathcal{A}}$  denotes the components of a vector or tensor in the  $\mathcal{A}$  triad.

Consider the following form of the vehicle equations of motion (1b):

$$m(\dot{\mathbf{v}}^{\mathcal{B}} + \boldsymbol{\omega}^{\mathcal{B}} \times \mathbf{v}^{\mathcal{B}}) = mg \mathbf{e}_3^{\mathcal{B}} + \mathbf{s}_{\text{aero}}^{\mathcal{B}}(\mathbf{v}^{\mathcal{B}}(\tau), \boldsymbol{\omega}^{\mathcal{B}}(\tau), \mathbf{u}(\tau), \boldsymbol{\pi}(\tau)), \quad (22a)$$

$$\mathbf{J}^{\mathcal{B}} \dot{\boldsymbol{\omega}}^{\mathcal{B}} + \boldsymbol{\omega}^{\mathcal{B}} \times (\mathbf{J}^{\mathcal{B}} \boldsymbol{\omega}^{\mathcal{B}}) = \mathbf{m}_{\text{aero}}^{\mathcal{B}}(\mathbf{v}^{\mathcal{B}}(\tau), \boldsymbol{\omega}^{\mathcal{B}}(\tau), \mathbf{u}(\tau), \boldsymbol{\pi}(\tau)), \quad (22b)$$

$$\dot{\mathbf{r}}^{\mathcal{E}} = \boldsymbol{\alpha}(\mathbf{e}) \mathbf{v}^{\mathcal{B}}, \quad (22c)$$

$$\dot{\mathbf{e}} = \mathbf{L}^{-1}(\mathbf{e}) \boldsymbol{\omega}^{\mathcal{B}}, \quad (22d)$$

where  $\mathbf{v}^{\mathcal{B}} = (u, v, w)^T$  and  $\boldsymbol{\omega}^{\mathcal{B}} = (p, q, r)^T$  are the body-attached components of the linear and angular velocity vectors, respectively,  $m$  being the mass and  $\mathbf{J}$  the inertia dyadic. The inertial components of the position vector are noted  $\mathbf{r}^{\mathcal{E}}$ , and  $\mathbf{e} = (\psi, \theta, \phi)^T$  are the Euler angles in the 3-2-1 sequence. The vehicle attitude is described in terms of the direction cosine matrix  $\boldsymbol{\alpha}(\mathbf{e})$ , parameterized in terms of the Euler angles, and  $\mathbf{L}(\mathbf{e})$  is the matrix which relates the body components of the angular velocity to the time rates of the Euler angles.

The body-attached components of the aerodynamic/propulsive forces  $\mathbf{s}_{\text{aero}}$  and moments  $\mathbf{m}_{\text{aero}}$  appearing on the right hand side of (22a,22b) depend on the body-attached components of the linear and angular velocity vectors, on the control inputs  $\mathbf{u}$  and on a number of physical parameters  $\boldsymbol{\pi}$ . The dependence on these quantities is through a retarded time  $\tau$ ,  $0 \leq \tau \leq t$ , which accounts for the memory effects of aerodynamic processes. This is typically approximated with a Taylor expansion arrested at the first order which has the effect of turning the delay differential equations (22) into ordinary differential equations.

It is readily verified [14] that (22) is unaffected by translations in the  $\mathbf{e}_1$ - $\mathbf{e}_2$  plane and by rotations about  $\mathbf{e}_3$ . Strictly speaking, translations along  $\mathbf{e}_3$  imply a change of air properties, and hence of the parameters  $\boldsymbol{\pi}$ , although this change is slow and can be neglected for small changes of altitude. Hence, the dynamics of the vehicle are invariant with respect to translations and to rotations about the local vertical.

This means that, given an input signal  $\mathbf{u}$  and given initial conditions on the linear and angular velocities, roll and pitch angles

$$\mathbf{v}^{\mathcal{B}}(t_0) = \mathbf{v}_0^{\mathcal{B}}, \quad (23a)$$

$$\boldsymbol{\omega}^{\mathcal{B}}(t_0) = \boldsymbol{\omega}_0^{\mathcal{B}}, \quad (23b)$$

$$\phi(t_0) = \phi_0, \quad (23c)$$

$$\theta(t_0) = \theta_0, \quad (23d)$$

the system will generate the same response  $\mathbf{v}^{\mathcal{B}}(t)$ ,  $\boldsymbol{\omega}^{\mathcal{B}}(t)$ ,  $\phi(t)$ ,  $\theta(t)$ , and  $\psi(t) - \psi_0$ ,  $\mathbf{r}^{\mathcal{B}}(t) - \mathbf{r}_0^{\mathcal{B}}$ , *irrespectively* of the initial conditions on heading and position

$$\psi(t_0) = \psi_0, \quad (24a)$$

$$\mathbf{r}^{\mathcal{B}}(t_0) = \mathbf{r}_0^{\mathcal{B}}, \quad (24b)$$

i.e. for any  $\psi_0$  and  $\mathbf{r}_0^{\mathcal{B}}$ . In other words,  $\mathbf{v}^{\mathcal{B}}(t)$ ,  $\boldsymbol{\omega}^{\mathcal{B}}(t)$ ,  $\phi(t)$ ,  $\theta(t)$ ,  $\psi(t) - \psi_0$ ,  $\mathbf{r}^{\mathcal{B}}(t) - \mathbf{r}_0^{\mathcal{B}}$  are invariant quantities, while  $\psi(t)$  and  $\mathbf{r}^{\mathcal{B}}(t)$  (or  $\mathbf{r}^{\mathcal{E}}(t)$ ) are not.

Therefore, we partition the state vector  $\mathbf{x}$  into invariant and non-invariant states,

$$\mathbf{x} = (\mathbf{x}_I^T, \mathbf{x}_{NI}^T)^T, \quad (25)$$

where the invariant states are  $\mathbf{x}_I = (\mathbf{v}^{\mathcal{B}T}, \boldsymbol{\omega}^{\mathcal{B}T}, \theta, \phi)^T$  and the non-invariant ones are  $\mathbf{x}_{NI} = (\psi, \mathbf{r}^{\mathcal{B}T})^T$ . Clearly, the optimal control function of (21) can only depend on the initial values of the invariant states, while it can not depend on the initial values of the non-invariant ones, for the invariance properties of the underlying mechanical system  $\mathcal{M}$  to be preserved.

A similar argument applies to the goal outputs  $\mathbf{y}^*$  which appear in (21). Suppose for example to define the system outputs as  $\mathbf{y} = (V, v, V_z, p, q, \Omega, \mathbf{e}^T, \mathbf{r}^{\mathcal{B}T})^T$ , where  $V = \|\mathbf{v}\|$  is the vehicle speed,  $V_z$  is the climb speed, and  $\Omega$  is the turn rate. Suppose now to specify a goal function  $\mathbf{y}^*(t)$ . This means that one is trying to track both invariant quantities ( $V^*(t), v^*(t), V_z^*(t), p^*(t), q^*(t), \Omega^*(t), \theta^*(t), \phi^*(t)$ ) and non-invariant ones ( $\psi^*(t), \mathbf{r}^{*\mathcal{B}}(t)$ ). However, given the invariance of the system with respect to translations of the origin and rotations about the vertical, it is clear that the optimal control which solves (1) can not depend on the non-invariant goal quantities  $\psi^*(t)$  and  $\mathbf{r}^{*\mathcal{B}}(t)$ , but only on their invariant counterparts  $\psi^*(t) - \psi_0$  and  $\mathbf{r}^{*\mathcal{B}}(t) - \mathbf{r}_0^{\mathcal{B}}$ . Therefore, here again we partition the output vector  $\mathbf{y}$  into invariant and non-invariant outputs, i.e.

$$\mathbf{y} = (\mathbf{y}_I^T, \mathbf{y}_{NI}^T)^T. \quad (26)$$

In conclusion, the minimizing optimal control function  $\mathbf{u}(t)$  which solves problem (1) when model  $\mathcal{M}$  is a vehicle, can be written as

$$\mathbf{u}(t) = \chi_s(\mathbf{x}_{I_0}, \mathbf{y}_I^*(t), \mathbf{y}_{NI}^*(t) - \mathbf{y}_{NI_0}, \mathbf{u}^*(t), t), \quad (27)$$

where  $\chi_s(\cdot, \cdot, \cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$  is a non-linear time dependent operator. This functional form asserts that, given a reduced model and optimization cost, the optimal control is a sole function of the initial conditions of the invariant states, of the goal invariant outputs, of the goal non-invariant outputs shifted in the origin, of the goal controls and of time.

### C. Parametric Approximation of the Optimal Control Solution

The functional form of the optimal control given in (21) (or (27) for flight mechanics problems) can be used for developing approximations to problem (1). However, in the spirit of the approach described in the introduction, we do not approximate (21) directly, but we first transform it into a reference-augmented form.

To this end, consider a reference (known) control action  $\mathbf{u}_{\text{ref}}(t), t \in [t_0, t_0 + T_p]$ , as computed by using some suitable available controller (for example, an LQR controller, as in the present work). If we assume that the reference control function depends, for a given state function  $\mathbf{x}(t)$ , on the goal outputs, goal controls and time, i.e.

$$\mathbf{u}_{\text{ref}}(t) = \zeta_x(\mathbf{y}^*(t), \mathbf{u}^*(t), t), \quad (28)$$

using (18) to eliminate the dependence on  $\mathbf{x}(t)$ , we find

$$\mathbf{u}_{\text{ref}}(t) = \gamma(\mathbf{x}_0, \mathbf{y}^*(t), \mathbf{u}^*(t), t), \quad (29)$$

where  $\gamma(\cdot, \cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ , which shows that the reference control depends at the most on the same quantities as the optimal control (21). Hence, using (29), it is always possible to write the optimal control  $\mathbf{u}(t)$  of (21) as

$$\mathbf{u}(t) = \mathbf{u}_{\text{ref}}(t) + \mathbf{v}(\mathbf{x}_0, \mathbf{y}^*(t), \mathbf{u}^*(t), t). \quad (30)$$

The term  $\mathbf{v}(\cdot, \cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$  represents the *control function defect*, i.e. that unknown function which, augmenting the reference solution, promotes it to the solution of the optimal control problem (1).

Clearly,  $\mathbf{v}$  is unknown in general, or otherwise we would know the solution of the optimal control problem in closed form. However, it is possible to construct an approximation of it in terms of a parametric function  $\mathbf{v}_p$ :

$$\mathbf{v}(\mathbf{x}_0, \mathbf{y}^*(t), \mathbf{u}^*(t), t) = \mathbf{v}_p(\mathbf{x}_0, \mathbf{y}^*(t), \mathbf{u}^*(t), t, \mathbf{p}_c) + \varepsilon_c, \quad (31)$$

where  $\mathbf{p}_c \in \mathbb{R}^{p_c}$  are free control parameters to be identified on-line in order to minimize the functional reconstruction error  $\varepsilon_c$ , and  $\mathbf{v}_p(\cdot, \cdot, \cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \times \mathbb{R}^{p_c} \rightarrow \mathbb{R}^m$ . Inserting (31) into (30), we can write the parameterized unknown control function as

$$\mathbf{u}(t) \approx \mathbf{u}_{\text{ref}}(t) + \mathbf{v}_p(\mathbf{x}_0, \mathbf{y}^*(t), \mathbf{u}^*(t), t, \mathbf{p}_c). \quad (32)$$

#### D. On-line Identification of the Control Parameters

Notice that in the infinite dimensional optimization problem (1), the unknowns are represented by the state  $\mathbf{x}(t)$ , co-state  $\boldsymbol{\lambda}(t)$  and control  $\mathbf{u}(t)$  functions, which must be computed so as to satisfy (17) and hence minimize the cost  $J$ . However, once the control function is put into the reference-augmented form (30) and the parametric approximation (31) is introduced, the resulting controls (32) cease to be independent variables, and become themselves functions of other quantities, and in particular of the states (through the reference control) and of the free control parameters. Considering this fact and imposing once again the stationarity of the augmented cost (16) with respect to all free variables, one obtains a new set of coupled ordinary differential equations, which write in this case

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}, \mathbf{p}_m) = 0, \quad t \in [t_0, t_0 + T_p], \quad (33a)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad (33b)$$

$$-\frac{d(\mathbf{f}_{,\dot{\mathbf{x}}}^T \boldsymbol{\lambda})}{dt} + (\mathbf{f}_{,\mathbf{x}} + \mathbf{u}_{\text{ref},\mathbf{x}}^T \mathbf{f}_{,\mathbf{u}})^T \boldsymbol{\lambda} + \mathbf{y}_{,\mathbf{x}}^T L_{,\mathbf{y}} + \mathbf{u}_{\text{ref},\mathbf{x}}^T L_{,\mathbf{u}} = 0, \quad t \in [t_0, t_0 + T_p], \quad (33c)$$

$$\boldsymbol{\lambda}(t_0 + T_p) = 0, \quad (33d)$$

$$\int_{t_0}^{t_0+T_p} \mathbf{v}_{,\mathbf{p}_c}^T (L_{,\mathbf{u}} + \mathbf{f}_{,\mathbf{u}}^T \boldsymbol{\lambda}) dt = 0. \quad (33e)$$

Equations (33) can be used for the on-line identification of the unknown parameters  $\mathbf{p}_c$ , through the following iterative process:

- 1) *State prediction.* Integrate the current estimate of the reduced model equations (33a) forward in time over the prediction window starting at the actual initial condition (33b). The initial value problem is solved under the action of the control inputs based on the augmented control function (32) evaluated in correspondence of the latest available estimate of the control parameters  $\mathbf{p}_c$ .
- 2) *Co-state prediction.* Using the augmented control function and the states computed at the previous step, integrate the adjoint equations (33c) backward in time starting from the final conditions (33d).

- 3) *Control parameter update.* The states and co-states computed at the two previous steps satisfy by construction all equations of the governing boundary value problem (33), except for the transversality condition

$$\hat{J}_{,\mathbf{p}_c} = 0, \quad (34)$$

i.e. (33e). Once this remaining condition is satisfied by all unknowns, the parameterized control becomes optimal, since the state and co-state equations and the boundary conditions are already satisfied. In order to seek the eventual enforcement of the transversality condition, the current estimate of the control parameters is corrected using the steepest descent rule:

$$\mathbf{p}_c = \mathbf{p}_c - \eta_c \hat{J}_{,\mathbf{p}_c}, \quad (35)$$

where  $\eta_c > 0$  is the step length. Based on this new estimate, the control action is updated using (32). Clearly, notice that by using (35), the control parameters are modified only as long as (34) is not satisfied.

- 4) *Plant steering.* Feed the computed controls to the plant, steering it on the window  $[t_0, t_0 + T_s]$ .  
 5) *Model parameter update.* Every  $N$  steps,  $N \geq 1$ , update the model parameter estimate using the procedure of Section III.  
 6) Update initial time as  $t_0 = t_0 + T_s$ , update initial conditions, and repeat from 1.

For regulation problems, the procedure above is based on the assumption that the time (i.e. the number of steps and hence of steepest descent corrections (35)) required in order to correctly identify the control parameters  $\mathbf{p}_c$  which satisfy the transversality condition (34), is small compared with the characteristic to-be-controlled time scales of the plant. In such a case in fact, the control becomes optimal (all governing equations (33) are satisfied) well before the system has been regulated. A similar reasoning applies to tracking problems as well. We have observed that this condition is easily verified in practise, and the identification time of the optimal control is typically quite small, as shown in the results section below.

### E. Neural-Network Based Implementation

In this work, the parametric approximation of the control function defect is based on a single-hidden-layer neural network.

To simplify the training of the control network, we reduce its input space in two ways. First, we drop the dependence of the parametric function on the time history of the goal quantities, by simply retaining their initial value at time  $t_0$ :

$$\mathbf{v}_p(\mathbf{x}_0, \mathbf{y}^*(t), \mathbf{u}^*(t), t, \mathbf{p}_c) \approx \mathbf{v}_p(\mathbf{x}_0, \mathbf{y}^*(t_0), \mathbf{u}^*(t_0), t, \mathbf{p}_c). \quad (36)$$

Notice that, even in the presence of the above approximation, the effect of time varying goal outputs is still felt by the controller during the co-state prediction phase (equations (33c,33d)), while time varying goal inputs are felt during the solution of (33c).

Second, the dependence of the control defect on time is rendered using shape functions which interpolate temporal nodal values. This can be done in several different ways. In this work we have found useful to partition the prediction

time window as

$$t_0 < t_1 < \dots < t_{M-1} \equiv t_0 + T_p, \quad (37)$$

where  $t_k$ ,  $k = (0, M-1)$ , are the temporal nodes; uniform or non-uniform distributions of the nodes can be used, depending on the application. The control function defect is then interpolated using, for example, linear shape functions between each couple of consecutive nodes; at the given time  $\tau$  within the prediction window, the control defect is interpolated as

$$\mathbf{v}_p(\mathbf{x}_0, \mathbf{y}^*(t_0), \mathbf{u}^*(t_0), \tau, \mathbf{p}_c) \approx (1 - \xi) \mathbf{v}_{p_k}(\mathbf{x}_0, \mathbf{y}^*(t_0), \mathbf{u}^*(t_0), \mathbf{p}_c) + \xi \mathbf{v}_{p_{k+1}}(\mathbf{x}_0, \mathbf{y}^*(t_0), \mathbf{u}^*(t_0), \mathbf{p}_c), \quad (38)$$

where  $\mathbf{v}_{p_k}(\cdot, \cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}^{p_c} \rightarrow \mathbb{R}^m$  is the control defect nodal value function at time  $t_k$ , the node index  $k$  is such that  $t_k \leq \tau < t_{k+1}$ , and  $\xi = (\tau - t_k)/(t_{k+1} - t_k)$ ,  $0 \leq \xi \leq 1$ . Clearly, shape functions other than linear can be used to interpolate the nodal values.

Finally, each nodal value of the time-discrete parametric control function defect is associated with one of the outputs of a multi-output neural network, whose form can be written

$$\mathbf{o}_c = \mathbf{W}_c^T \sigma(\mathbf{V}_c^T \mathbf{i}_c + \mathbf{a}_c) + \mathbf{b}_c, \quad (39)$$

where  $\mathbf{o}_c = (\mathbf{v}_{p_0}^T, \mathbf{v}_{p_1}^T, \dots, \mathbf{v}_{p_{M-1}}^T)^T \in \mathbb{R}^{mM}$  are the network outputs,  $\mathbf{i}_c = (\mathbf{x}_0^T, \mathbf{x}^{*T}(t_0), \mathbf{u}^{*T}(t_0))^T \in \mathbb{R}^{2n+m}$  are the network inputs, while  $\mathbf{W}_c \in \mathbb{R}^{n_h \times mM}$ ,  $\mathbf{V}_c \in \mathbb{R}^{(2n+m) \times n_h}$ ,  $\mathbf{a}_c \in \mathbb{R}^{n_h}$  and  $\mathbf{b}_c \in \mathbb{R}^{mM}$  are the network weights and biases which represent the free control parameters, i.e.

$$\mathbf{p}_c = (\dots, W_{c_{ij}}, \dots, V_{c_{ij}}, \dots, a_{c_i}, \dots, b_{c_i}, \dots)^T, \quad (40)$$

with  $\mathbf{p}_c \in \mathbb{R}^{p_c}$ ,  $p_c = n_h(2n + m(M+1) + 1) + mM$ .

## V. RESULTS AND APPLICATIONS

### A. Double Inverted-Pendulum Model Problem

We consider the classical problem of the stabilization of a double inverted-pendulum on a cart, as shown in Fig. 2. The system has three degrees of freedom, the horizontal position of the cart  $\vartheta_0$  and the angles  $\vartheta_1$  and  $\vartheta_2$  of the two bars with respect to the vertical direction, so that  $\mathbf{x} = (\dot{\vartheta}^T, \vartheta^T)^T$ ,  $\vartheta = (\vartheta_0, \vartheta_1, \vartheta_2)^T$ ,  $n = 6$ . The system has one single control ( $m = 1$ ) represented by the horizontal force  $u$  applied to the cart, and the outputs coincide with the states,  $l = n$ ,  $\mathbf{y} = \mathbf{x}$ . The two uniform bars have length  $L_1 = 0.5$  m and  $L_2 = 0.75$  m and mass  $M_1 = 0.5$  kg and  $M_2 = 0.75$  kg, while the cart has mass  $M_0 = 1.5$  kg. The bars are initially aligned and both form an angle of 25 deg with the vertical. The problem is to regulate the system to the origin (the vertical unstable equilibrium) by minimizing the cost (2), where  $\mathbf{y}^* = (0, 0, 0, 0, 0, 0)^T$ ,  $\mathbf{u}^* = 0$ ,  $\mathbf{Q} = \text{diag}(20, 700, 700, 5, 50, 50)$  and  $R = 1$ .

1) *Choice of the Prediction Window and Identification of Control Parameters:* In the first set of tests, we do not perform any model adaptivity, and we identify on-line the sole control parameters. The reference controller is of the LQR type, obtained by linearization of the equations of motion about the vertical position. The steering window is set to  $T_s = 0.02$  sec; the state and co-state predictions are performed by discretizing (33a) and (33c)

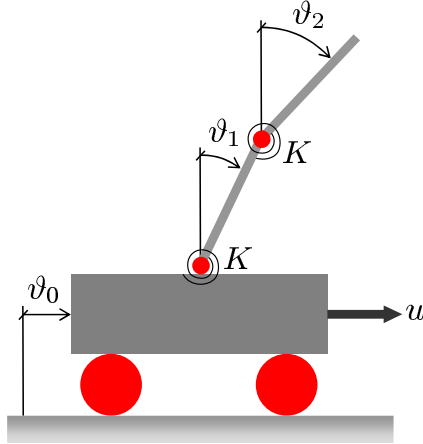


Fig. 2. Double inverted-pendulum on a cart

with the forward Euler scheme, with a time step  $\Delta t = 0.02$  sec. The control network has  $n_h = 3$  neurons and  $M = 5$  outputs (temporal nodes of the control function defect), uniformly distributed on the prediction window.

In order to select the length of the prediction window  $T_p$ , we define the total regulation error as

$$E = \int_0^T L dt, \quad (41)$$

where  $T$  is the time necessary in order to regulate the system and  $L$  is given in (2). Figure 3 shows the normalized total regulation error  $E/E_{\text{LQR}}$  vs. the length of the prediction window  $T_p$ , where  $E_{\text{LQR}}$  is the error obtained with the sole LQR reference controller. The figure shows that RAPC recovers the regulation error of the LQR controller if the prediction window is slightly longer than 4.0 sec, after which further improvements in the regulation error are only marginal. Based on these results, in the following examples we always used a value  $T_p = 3.5$  sec, which is a good compromise between computational efficiency and performance.

Next, we assess the ability of the proposed procedure to produce optimal controls which solve problem (1). Figure 4 shows with a solid line the time history of the 2-norm of the violation of the transversality condition (34). Recall that the satisfaction of the transversality condition implies that the solution is optimal, in the sense that it satisfies all coupled governing differential equations (33) originating from (1). The same plot reports also the time history of the angle  $\vartheta_2$  of the second bar using a dashed line. The plot shows that the controller quickly achieves

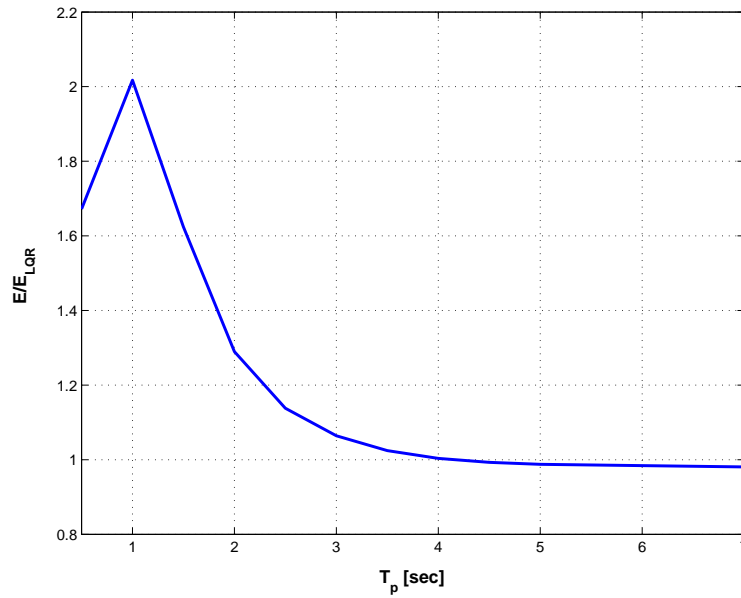


Fig. 3. Double inverted-pendulum on a cart. Total normalized regulation error  $E/E_{LQR}$  vs. length of the prediction window  $T_p$

optimality, well before the system is regulated, as it appears clearly by comparing the characteristic times of the two curves.

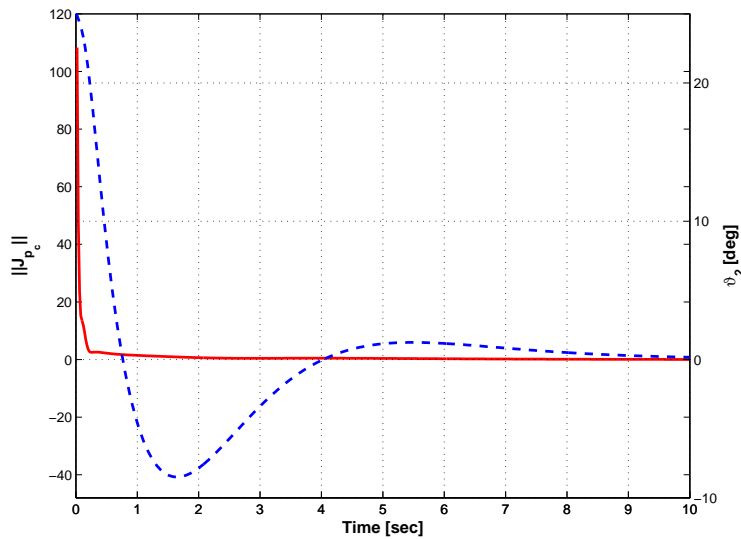


Fig. 4. Double inverted-pendulum on a cart. Time history of the 2-norm of the violation of the transversality condition (34) (solid line), and time history of the angle of the second bar (dashed line)

2) *Effect of Model Adaption:* In order to demonstrate the effects of model adaption, in this second set of tests we introduce a mismatch between plant and reduced model. More precisely, the plant now includes two springs

which apply destabilizing torques to the two bars; these torques are not present in the reduced model. The first torque  $Q_1 = K(\pi/2 - \vartheta_1)$  is applied between the first bar and the cart, while  $Q_2 = K(\pi + \vartheta_1 - \vartheta_2)$  is applied between the second and the first bars. Goal of the example is to show whether this mismatch between plant and reduced model can be compensated by the proposed model identification procedure.

The numerical experiments were conducted as follows. The stiffness  $K$  of the springs was progressively increased starting from 0 up to  $K_{\max} = 0.05 \text{ Nm/rad}$ , thus producing increasing errors between reduced model and plant; notice that these errors affect both the equilibrium condition and the dynamic behavior of the system. For each value of  $K$ , the new plant equilibrium  $\tilde{x}_0$  was numerically evaluated by solving the system of equations  $\tilde{f}(\tilde{x}, \tilde{x}, \tilde{u}) = 0$ , setting  $\dot{\tilde{x}} = 0$  and  $\tilde{u} = 0$ .

At first, we computed the optimal gains of a LQR controller based on the linearization of the plant about the new equilibrium point, including the effects of the springs. This first LQR controller, labelled  $\text{LQR}_{\text{exact}}$  in the following, was used for providing a benchmark result in the absence of modeling errors.

Next, the optimal gains of a second LQR controller were computed linearizing the reduced model, which is completely unaware of the presence of the springs, about the vertical position  $x = 0$ . This second LQR was used as the reference control element of RAPC with reduced model adaption, using  $n_h = 3$  neurons for the model defect network. For each experiment at every given value of the spring stiffness, we initialized the model and control parameters to small random values. All the experiments were performed with the same parameters as in the first set of tests, except for the goal output which in this case is the true equilibrium point of the plant  $y^* = \tilde{x}_0$ , which differs from the vertical because of the presence of the springs.

The results are summarized in Fig. 5, which shows the total regulation error as a function of the stiffness  $K$  of the two springs. The solid line represents the regulation error of RAPC, while the dashed line reports the regulation error of  $\text{LQR}_{\text{exact}}$ . Apparently, the proposed controller is not only able to compensate the model mismatch generated by the springs, but it also achieves a significant performance improvement with respect to the benchmark provided by the exact LQR implementation.

For the case  $K = K_{\max}$ , Fig. 6 shows the improved prediction capabilities achieved with reduced model identification. The top part of the figure reports the  $\dot{\vartheta}_1$  angular velocity, while the bottom part shows the  $\dot{\vartheta}_2$  angular rate. The solid lines show the time histories obtained by integrating the plant equations on each steering window under the effects of the  $\text{LQR}_{\text{exact}}$  control action. The dashed lines show the results obtained by the integration on each steering window of the sole reference model, again with the same control input signal and starting from the same initial conditions. Finally, the dash-dotted lines give the results obtained by integration of the reduced model (reference plus neural defect correction), again from the same initial conditions and with the same control inputs. It appears that there is an excellent correspondence between plant and augmented model shortly after a fast transient. The improvement with respect to the use of the sole reference model is clearly apparent, showing the effect of the model augmentation to improve the prediction performance.

Figure 7 shows the adaption process through yet another point of view, reporting with a solid line the norm of the model reconstruction error  $\|\varepsilon_m\|$  of (10) as a function of time. The same figure also plots with a dashed line

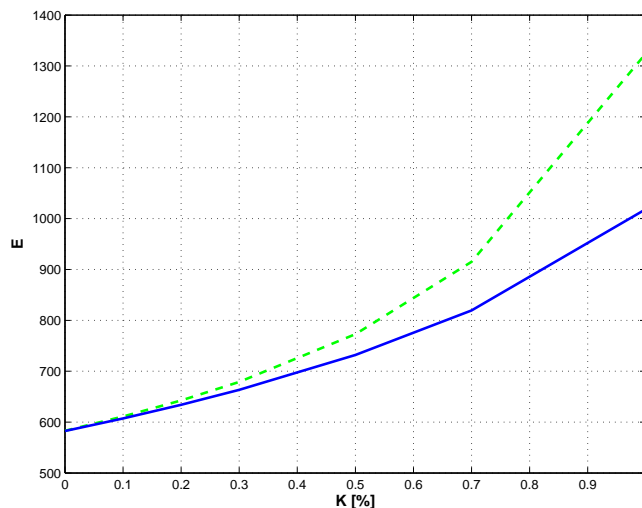


Fig. 5. Double inverted-pendulum on a cart. Total regulation error  $E$  vs. spring stiffness  $K$ . RAPC: solid line;  $LQR_{\text{exact}}$ : dashed line

the time history of the angular rate  $\dot{\vartheta}_2$  for the second bar. By examining the figure, one can appreciate that the neural element is able to identify the defect quickly (i.e. in a time which is small when compared to the plant response time scale), in turn improving the prediction fidelity of the reduced model. The rapidity of the adaption is certainly also due to the reasonably good performance of the reference model, which makes the defect a relatively small quantity; this makes the adaption a faster and easier task.

### B. Reflexive Control of an Unmanned Rotorcraft

In this section we consider the application of RAPC to the reflexive control of an autonomous rotorcraft vehicle. The numerical experiments are conducted in a high fidelity virtual environment, which allows to more precisely quantify the performance of the controller. Application in flight to a small size rotorcraft UAV is underway, and the results will be discussed in a forthcoming paper.

The plant is a vehicle model which represents a medium-size multi-engine four-bladed articulated-rotor utility helicopter, validated with respect to flight data in a previous research effort. The mathematical model of the helicopter is as given in §IV-B. The helicopter controls are defined as  $\mathbf{u} = (A_1, B_1, \theta_{0MR}, \theta_{0TR})^T$ ,  $m = 4$ , where  $A_1$ ,  $B_1$  are the lateral and longitudinal cyclics, respectively, while  $\theta_{0MR}$  is the main rotor collective and  $\theta_{0TR}$  is the collective of the tail rotor. The main and tail rotor forces and moments are computed by combining actuator disk and blade element theory, considering a uniform inflow [8]. 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 model considers simple corrections to account for compressibility effects and for the downwash angle at the tail due to the main rotor. The model includes process and measurement noise models and delays, and the vehicle states are reconstructed on-line using

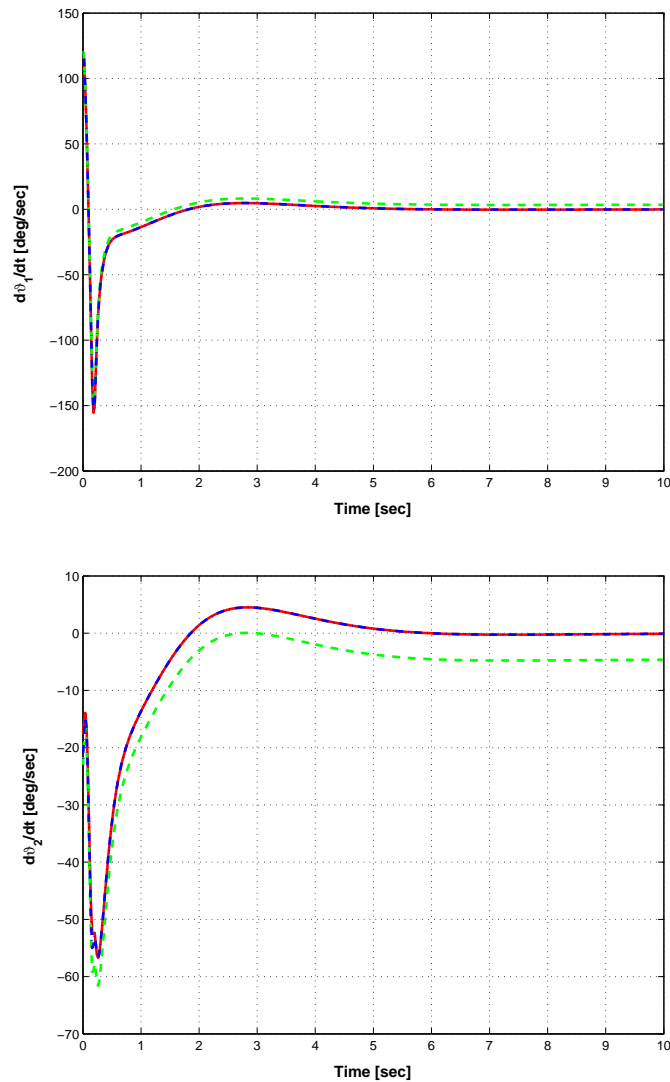


Fig. 6. Double inverted-pendulum on a cart. Time history of the angular rates  $\dot{\theta}_1$  (top) and  $\dot{\theta}_2$  (bottom), for  $K = 0.05$ . Plant: solid line; reference model: dashed line; augmented model: dash-dotted line

an extended Kalman filter.

The vehicle is piloted along an aggressive flight routine, where the vehicle starts in straight and leveled flight at 30 m/s, decelerates to 20 m/s, and then performs a 30 m climb, a 90 deg right turn, a 30 m descent, and finally an acceleration to 30 m/s; the flight sequence is represented in Fig. 8.

The goal outputs and goal controls of the flight routine are computed using the planning procedures of Bottasso et al. [14]; using this approach, the vehicle is either in a trim or in a maneuver condition, the latter being defined as a finite-time transition between two trims [13]. The time history of the vehicle states and associated control inputs along each maneuver are computed by solving maneuver optimal control problems for the non-linear reduced model

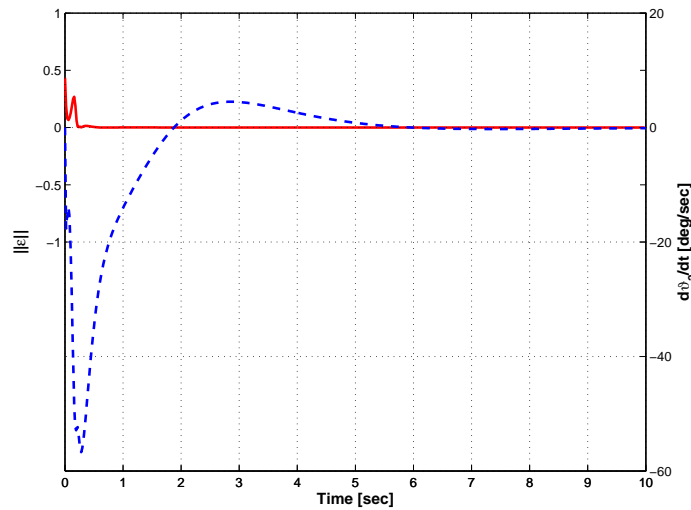


Fig. 7. Double inverted-pendulum on a cart. Time history of the norm of the model reconstruction error  $\|\epsilon_m\|$  (solid line), and time history of the second bar angular rate (dashed line), for  $K = 0.05$

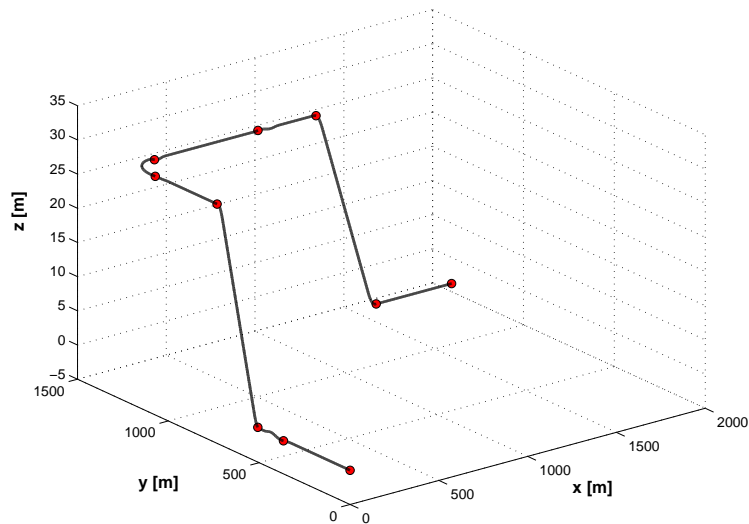


Fig. 8. Flight routine for the autonomous helicopter problem

of the vehicle, using a direct transcription approach, while in trim conditions vehicle states and control inputs are computed solving non-linear trim problems (cfr. [14] for details).

The reference control element is based on an output-feedback LQR, using only invariant outputs (equation (26)). At all trim conditions of the flight routine, the optimal LQR control gains are pre-computed off-line using the procedure of [3], based on a linearization of the vehicle equations of motion at that trim condition. The output-

feedback LQR reference control law is computed at 50 Hz as

$$\mathbf{u}_{\text{ref}} = \mathbf{u}^* - \mathbf{K}(\mathbf{y}_I - \mathbf{y}_I^*), \quad (42)$$

where the goal quantities are obtained by first finding the closest point on the to-be-tracked trajectory, and then interpolating outputs and controls as computed by the planning procedures at that point. When the goal point is along a maneuver, the optimal output-feedback gain matrix  $\mathbf{K}$  is computed using the linearized vehicle model in the arrival trim.

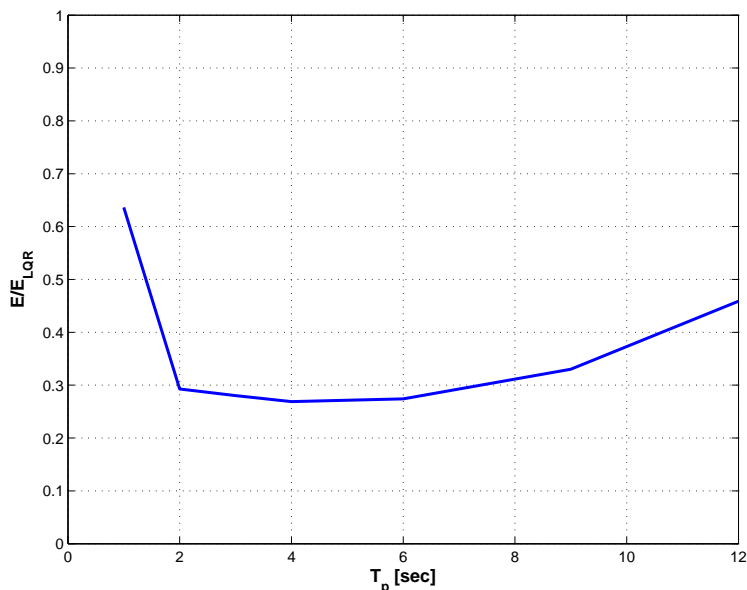


Fig. 9. Autonomous helicopter problem. Normalized tracking error  $E/E_{LQR, \alpha=0}$  vs. length of the prediction window  $T_p$

1) *Tuning of the Controller:* Several simulations were conducted in order to optimize the choice of the various algorithmic parameters. Detailed results are not reported here for the sake of brevity, but for the model adaption process the experiments lead to the choice of  $n_h = 3$  neurons for the model network and an activation frequency of model adaption  $N = 10$  Hz (step 5 in the iterative process of §IV-D). For the control process, the choice fell on four single-output control networks (one for each control) with  $n_h = 3$  neurons each, a steering window of 0.02 sec and a time step for the integration of the state and co-state equations  $\Delta t = 0.02$  sec.

The choice of the prediction window size  $T_p$  was based on the tracking cost, which is defined as

$$E = \int_{T_0}^T L dt, \quad (43)$$

where  $T$  is the total duration of the flight routine, and  $T_0 = 2$  sec is used to exclude the initial transient. Figure 9 reports the tracking error as a function of the length of the prediction window  $T_p$ . The tracking error for RAPC is normalized with respect to the tracking error obtained with the sole use of the reference LQR controller. The figure shows that with a prediction window of 2 sec, RAPC already improves the performance of the reference controller,

reaching a minimum around  $T_p \approx 5 \div 6$  sec. The best performance corresponds to an improvement in the tracking cost of more than 70% with respect to the output-feedback LQR. Based on these results, the prediction window was selected as  $T_p = 3$  sec for the following examples.

2) *Performance Assessment in the Presence of Modeling Errors:* Next, we verify here again in the context of this more complex example the ability of the proposed procedures to improve the reference control in the presence of modeling errors. To this end, we consider the complex flight routine tracking problem introduced above. The problem is now parameterized by using a factor  $\alpha$ . For each value of  $\alpha$ , the reduced model as well as the goal outputs and goal controls remain the same, whereas the plant is affected by errors parameterized in  $\alpha$  for the air density and the vehicle mass. More precisely, if the reduced model uses an air density  $\rho$ , the plant uses a density  $\rho(1 - \alpha)$ , and if the reduced model has a total vehicle mass  $M$ , the plant uses a mass  $M(1 + \alpha)$ . Hence, the factor  $\alpha$  measures the mismatch between reduced model and plant.

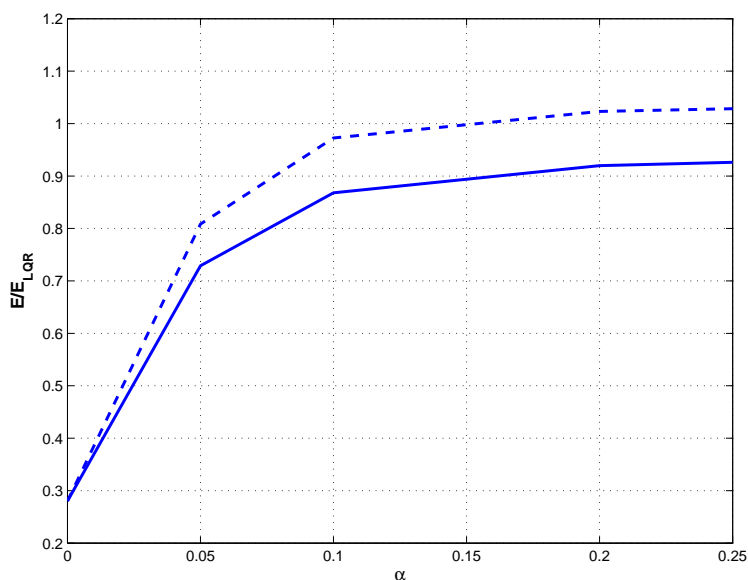


Fig. 10. Autonomous helicopter problem. Normalized tracking error  $E/E_{LQR}$  vs. model mismatch parameter  $\alpha$ . RAPC with model adaption: solid line; RAPC without model adaption: dashed line

Similarly to the case of the double-inverted pendulum, the simulations were conducted as follows. The mismatch factor  $\alpha$  was progressively increased starting from  $\alpha = 0$ . At each value of  $\alpha$ , the optimal gains of the output-feedback LQR were computed based on linearizations of the reduced model, in correspondence of the current value of  $\alpha$  and hence affected by modeling errors. This controller was first used alone and then as the reference control element of RAPC.

The results of these simulations are reported in Fig. 10, which gives the normalized tracking error  $E/E_{LQR}$  as a function of the model mismatch parameter  $\alpha$ . The solution obtained by RAPC without model adaption is shown using a dashed line, while the results for RAPC with model adaption are plotted using a solid line.

This figure indicates several interesting facts. First of all, in the absence of modeling errors ( $\alpha = 0$ ), the use of non-linear model predictive control allows for a reduction of the tracking error in excess of 70% with respect to the LQR case. Although this remarkable improvement decreases with increasing values of  $\alpha$ , it remains substantial for an ample margin of the model mismatch parameter. Furthermore, even without model adaption, RAPC with the sole on-line identification of the control parameters is able to outperform the LQR controller up to  $\alpha \approx 0.15$ . Therefore, the control-adaptive features of RAPC seem to allow for a partial mitigation of modeling errors, in the sense that the controller seems to be able to cope with the limitations of its reduced model by correspondingly adapting the control law. Finally, the introduction of model adaption has a beneficial effects, as expected, and results in a further reduction of the tracking error with respect to the LQR case.

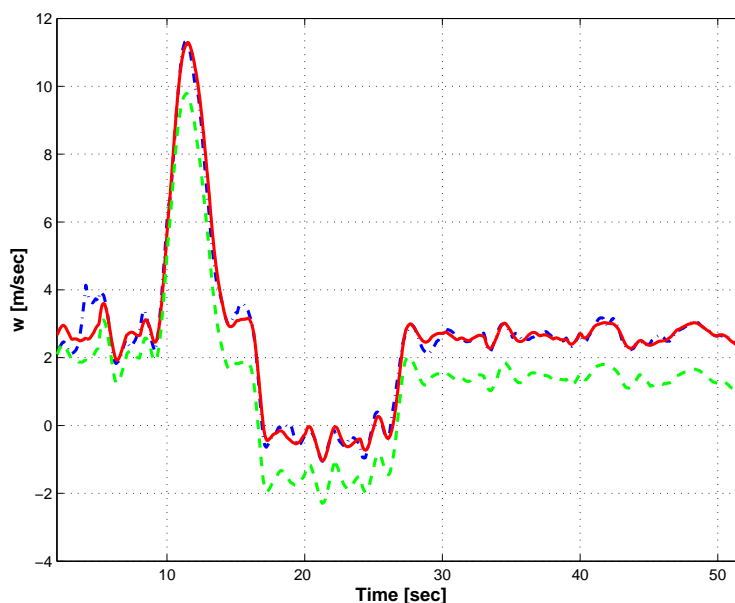


Fig. 11. Autonomous helicopter problem. Time history of the linear body-frame velocity component  $w$ , for  $\alpha = 0.2$ . Plant: solid line; reference model: dashed line; augmented model: dash-dotted line

In analogy with what done for the pendulum problem, in Fig. 11 we report the improved prediction capabilities of the augmented adaptive model. In particular the figure shows the linear body frame velocity component  $w$ , obtained in the case  $\alpha = 0.2$  when different models are integrated in time with the same initial condition and the same control inputs. The difference between the augmented model (dash-dotted line) and the reference model alone (dashed line) is significant, the former capturing well the plant dynamics soon after a very short transient. Similar results are obtained for the other vehicle states, but are not reported here for the sake of brevity.

To show the local behavior of the controllers, for the case  $\alpha = 0.1$ , Fig. 12 reports the time history of the goal turn rate and the corresponding quantity obtained by the tracking helicopter; the LQR case is shown with a dashed line, while the RAPC case with model adaption is plotted with a dash-dotted line. These results show here again the improved tracking capabilities of RAPC as compared to the LQR.

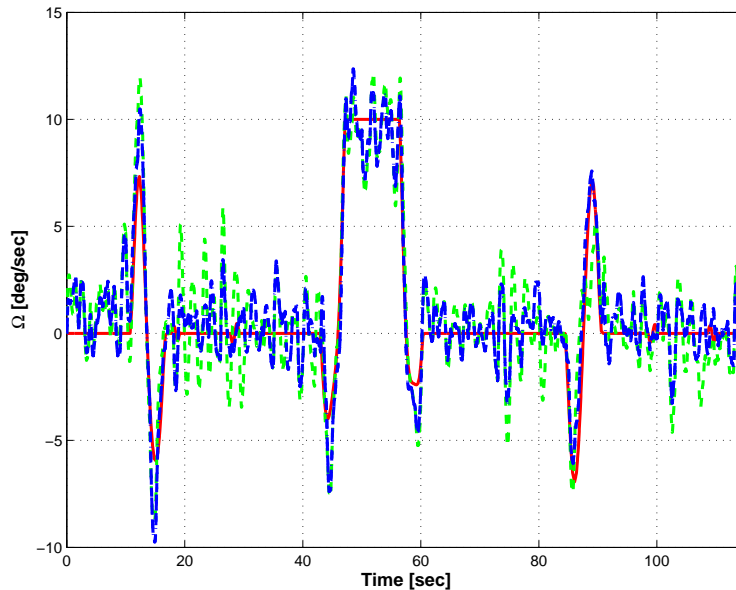


Fig. 12. Autonomous helicopter problem. Time history of goal turn rate and corresponding quantity for the tracking helicopter, for a model mismatch parameter  $\alpha = 0.1$ . Goal: solid line; LQR: dashed line; RAPC with model adaption: dash-dotted line

3) *Trajectory Tracking Performance Assessment*: The LQR controller based on (42) is subject to drift and heading errors, since it only tracks invariant quantities. To correct this behavior, we use a slower outer loop operating at 5 Hz, based on a simple proportional-integral (PI) controller, which reacts to positions errors (i.e. distance between current and projected positions along the goal trajectory) and heading errors [14]. The gains of the compensator are computed off-line using an optimization technique. Specifically, the tracking error of the non-invariant quantities is considered to be a sole function of the control gains of the compensator. This error is minimized by using a gradient-based optimization technique coupled with a response surface.

Notice that the RAPC controller does not need a drift and heading compensator, since it can track both invariant and non-invariant outputs. Position and heading tracking is simply achieved by turning on the tracking cost function weights of (2) acting on the corresponding outputs.

In this section we compare the trajectory tracking performance of RAPC and of the compensator-enhanced LQR controller. Figure 13 reports the tracking performance in the case of model error with  $\alpha = 0.15$ , showing the goal, LQR and RAPC trajectories. Furthermore, Fig. 14 gives a more quantitative view, by plotting the norm of the distance  $\mathbf{d}$  from the goal trajectory,  $\mathbf{d} = \mathbf{r} - \mathbf{r}^*$ . From the figures it appears that, since the model mismatch acts on mass and density, its effect is predominantly manifest on the altitude tracking. The figures also show that the LQR controller, although augmented with a drift and heading compensator whose gains were very carefully optimized as explained above, has a hard time to correct for the model error; on the other hand, RAPC improves the LQR tracking performance in excess of 50%.

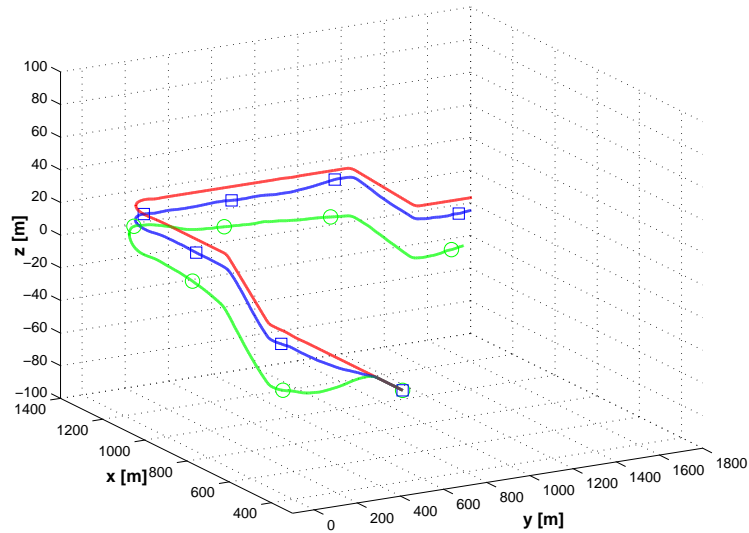


Fig. 13. Autonomous helicopter problem. Goal trajectory and corresponding quantity for the tracking helicopter, for a model mismatch parameter  $\alpha = 0.15$ . Goal: solid line; LQR and PI compensator: circle marker; RAPC with model adaption: square marker

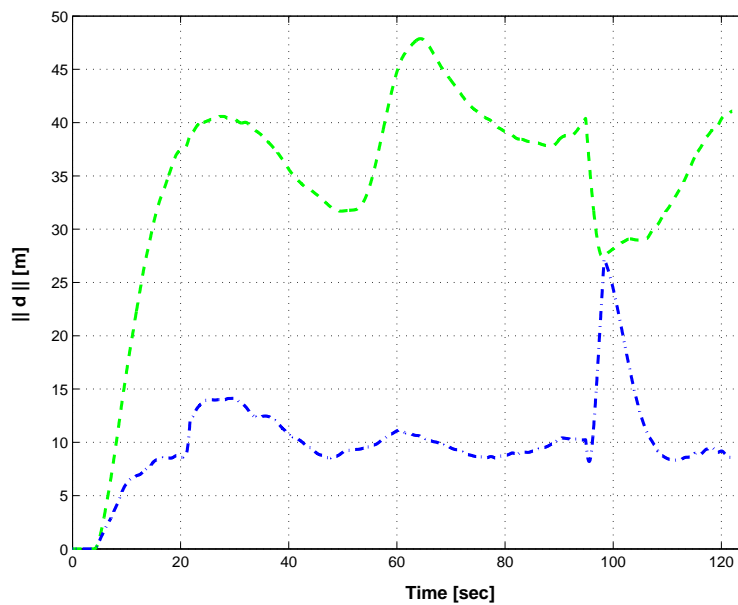


Fig. 14. Autonomous helicopter problem. Norm of the distance from the goal trajectory, for a model mismatch parameter  $\alpha = 0.15$ . LQR and PI compensator: dashed line; RAPC with model adaption: dash-dotted line

## VI. CONCLUSIONS

We have described a novel adaptive non-linear model predictive control approach, based on the idea of augmenting reference elements with parametric functions both at the level of the reduced model and at the level of the control action.

The reduced model identification takes a reference reduced model and approximates its defect, i.e. it reconstructs that unknown function which, when put on the right hand side of the reference model, is satisfied by the measured plant states. This approach amounts to a black-box identification of the physics which are not captured or not well resolved by the reference model. An alternative but somewhat related approach would be to insert black-box approximators in specific parts of the reference model, where the true physics are known not to be well represented. Both approaches can be seen as ways to boost the accuracy of existing legacy analytical models, so as to improve the predictive fidelity of the reduced model to the plant.

The control function identification takes a reference control action and promotes it to the solution of the underlying optimal control problem. Key to this approach is an analysis of the governing equations which indicates the functional dependence of the unknown control function on the current and desired states. Once the dependence of the unknown control function has been identified, it is possible to construct an approximation of it using a parametric function. Here again, this approach can be seen as a way to boost existing control solutions.

The numerical experiments demonstrated the basic features of the proposed procedures, both for a classical model problem and for the control of an autonomous high performance rotorcraft. The tests, although limited for now to the sole virtual environment, show a very pronounced improvement of the reference solutions when modeling errors are present. In all cases, the identification of the reduced model and of the control parameters was fast compared to the characteristic to-be-controlled time scales of the plant, which is crucial for the effectiveness of the procedures. We believe that much of the robustness and good performance that we have found is due to the very basic idea of the proposed control framework, i.e. on the availability of reference elements. These in fact ensure reasonable performance of the controller even during the initial transient, before identification of the parameters has taken place, easing and speeding up the identification process itself. We remark in fact that all problems studied here were solved without any pre-training, a major hurdle in other adaptive network-based approaches.

The software is currently being ported on-board a small autonomous rotorcraft in the lab of the authors, which will allow for further testing of the procedures in flight.

#### ACKNOWLEDGMENTS

The authors acknowledge many stimulating discussions with Dr. Alessandro Croce.

#### REFERENCES

- [1] R. Findeisen, L. Imland, F. Allgöwer, and B. Foss, "State and output feedback nonlinear model predictive control: An overview," *European Journal of Control*, vol. 9, pp. 190–206, 2003.
- [2] B. Stevens and F. Lewis, *Aircraft Control and Simulation*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2003.
- [3] D. Moerder and A. Calise, "Convergence of a numerical algorithm for calculating optimal output feedback gains," *IEEE Transactions on Automatic Control*, vol. 30, pp. 900–903, 1985.
- [4] G. Grimm, M. Messina, S. Tuna, and A. Tell, "Model predictive control: For want of a local control Lyapunov function, all is not lost," *IEEE Transactions on Automatic Control*, vol. 50, pp. 546–558, 2005.
- [5] J. Betts, *Practical Methods for Optimal Control using Non-Linear Programming*. Philadelphia, PA, USA: SIAM, 2001.

- [6] C. 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*, vol. 195, pp. 7052–7072, 2006.
- [7] E. Wan and A. Bogdanov, "Model predictive neural control with applications to a 6 dof helicopter model," in *Proceedings of IEEE American Control Conference*, vol. 1, Arlington, VA, June 2001, pp. 488–493.
- [8] R. Prouty, *Helicopter Performance, Stability, and Control*. Malabar, FL, USA: R.E. Krieger Publishing Co., 1990.
- [9] W. Johnson, *Helicopter Theory*. New York, NY, USA: Dover Publications, 1994.
- [10] S. Ferrari and R. Stengel, "Online adaptive critic flight control," *Journal of Guidance, Control, and Dynamics*, vol. 27, pp. 777–786, 2004.
- [11] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, 1990.
- [12] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer feed-forward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
- [13] E. Frazzoli, "Robust hybrid control for autonomous vehicle motion planning,," Ph.D. dissertation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, 2001.
- [14] C. Bottasso, D. Leonello, and B. Savini, "Path planning for autonomous vehicles by trajectory smoothing using motion primitives," *IEEE Transactions on Control Systems Technology*, 2007, under review.