ROTORCRAFT TRIM BY A NEURAL MODEL-PREDICTIVE AUTO-PILOT

A Thesis
Presented to
The Academic Faculty

by

Luca Riviello

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Aerospace Engineering

School of Aerospace Engineering
Georgia Institute of Technology
April 2005

ROTORCRAFT TRIM BY A NEURAL MODEL-PREDICTIVE AUTO-PILOT

Approved by:

Prof. Carlo L. Bottasso, Chair,
Daniel Guggenheim School of
Aerospace Engineering,
Georgia Institute of Technology

Prof. Olivier Bauchau,
Daniel Guggenheim School of
Aerospace Engineering,
Georgia Institute of Technology

Prof. Dewey H. Hodges,
Daniel Guggenheim School of
Aerospace Engineering,
Georgia Institute of Technology

Date Approved: 14 April 2005

To my old friends and my whole family (I'm coming back).

To the new "American" friends, I will always remember this great time, and I will bother you in the future.

And to my love.

Thank you.

$\mathcal{L}$.

ACKNOWLEDGEMENTS

I'd like here to thank my committee: it has been wonderful to work with you, I hope I will deserve other opportunities like this in the future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Description
$\widetilde{(\bullet)}$	Quantity of the system (comprehensive model)
$(\bullet)^{\text{reg}}$	Quantity of the model predictive regulation problem
$(\bullet)^{\text{steer}}$	Quantity of the steering problem
$(\bullet)^{\text{adapt}}$	Quantity of the model adaption problem
$(\bullet)^*$	Given or desired value
$\| \bullet \|_2$	Euclidean norm
$\widetilde{\boldsymbol{x}}$	System states
$\widetilde{\boldsymbol{\lambda}}$	System Lagrange multipliers
$\widetilde{\boldsymbol{u}}$	System controls
$\widetilde{\boldsymbol{y}}$	System outputs
$\boldsymbol{y}$	Reduced model states
$\boldsymbol{u}$	Reduced model outputs
$\boldsymbol{p}$	Reduced model parameters
$\dot{(\bullet)} = \mathrm{d}(\bullet)/\mathrm{d}t$	Derivative with respect to time
$\mathrm{diag}(\bullet)$	Operator $\mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ transforming a vector in a square matrix with the vector elements on the diagonal
t	Time
T_i	Initial time
T_f	Final time
J	Cost function
T	Rotor revolution period

Symbol	Description
$(\bullet)_h$	Discretized quantity, as obtained by the use of a numerical method
$\mathcal{T}_h$	Computational grid on the interval $[T_i, T_f]$
K	Element (time step) of $\mathcal{T}_h$

SUMMARY

In this work we investigate the use of state-of-the-art tools for the regulation of complex, non-linear systems to improve the methodologies currently applied to trim comprehensive virtual prototypes of rotors and rotorcrafts.

Among the several methods that have been proposed in the literature, the auto-pilot approach has the potential to solve trim problems efficiently even for the large and complex vehicle models of modern comprehensive finite element-based analysis codes. In this approach, the trim condition is obtained by adjusting the controls so as to virtually "fly" the system to the final steady (periodic) flight condition. Published proportional auto-pilots show to work well in many practical instances. However, they cannot guarantee good performance and stability in all flight conditions of interest. Limit-cycle oscillations in control time histories are often observed in practice because of the non-linear nature of the problem and the difficulties in enforcing the constant-in-time condition for the controls.

To address all the above areas of concern, in this research we propose a new auto-pilot, based on non-linear model-predictive control (NMPC). The formulation uses a non-linear reference model of the system augmented with an adaptive neural element, which identifies and corrects the mismatch between reduced model and controlled system.

The methodology is tested on the wind-tunnel trim of a rotor multibody model and compared to an existing implementation of a classic auto-pilot. The proposed controller shows good performance without the need of a potentially very expensive tuning phase, which is required in classical auto-pilots. Moreover, model-predictive control provides a framework for guaranteeing stability of the non-linear closed-loop system, so it seems to be a viable approach for trimming complete rotorcraft comprehensive models in free-flight.

CHAPTER I

INTRODUCTION

In the aeronautical field, the word "trim" is adopted to indicate the aircraft control settings, attitude and cargo disposition necessary to obtain a desired steady flight condition. For a fixed wing aircraft, a steady flight condition can be easily characterized for example by some constant values of the linear and angular velocities in a body attached frame. On the contrary, a rotorcraft is characterized by rotating surfaces subjected to aerodynamic loads, so it is always excited by harmonic external forces. Thus for a rotorcraft no equilibrium conditions exist such that the rates of the aircraft states are constant. However, the controls and the attitude can be changed to achieve a particular periodic orbit. On this orbit, the rotorcraft is characterized by periodic solutions of all the states and loads, though the controls are constant in time. This flight condition can be characterized in terms of the average over a whole rotor revolution of some components of the states and loads.

In solving the trim problem we are interested in starting from these desired (average) values and compute the constant controls and the attitude that make the rotorcraft fly in the corresponding steady (periodic) condition. In practice, for rotary wing applications we are interested in either trimming a rotor model constrained to the ground or a complete aircraft model in free-flight. In the first case, the trim problem consists in finding the rotor controls that produce desired average values of a number of components of the hub loads equal to the number of controls. For free-flight simulation, for instance for a conventional rotorcraft configuration with constant rotor speed, we look for the (constant) main and tail rotor controls and two (periodic) attitude angles[1] that produce desired values of the body linear and angular rates in a body axis frame.

The solution of the trim problem is the starting point for estimating performance, rotor loads and vibratory level. Moreover, the accurate determination of the vehicle trim

[1] generally pitch and yaw angles, while the average roll is chosen *a priori*.

settings plays a crucial role in the analysis of the flight mechanics characteristics of the vehicle. In fact, the dynamic and aeroelastic stability, the handling qualities and the control system design are commonly analyzed by perturbing the system about the periodic orbit corresponding to the trim condition. In other words, the analysis is based on a set of perturbation equations which strongly depend upon the steady solution that is perturbed. So, an inaccurate trim computation potentially undermines the validity of all the approaches which perturb the system about a steady condition. Moreover complexity and non-linearity of comprehensive rotorcraft models make rotor loads and vibratory level analyses strongly affected by the computed trim solution.

Trim settings can not be estimated using flight test data because of the unavoidable mismatch between the virtual prototype and the real system. Because of modelling approximations, measured controls cannot trim a numerical model, which would simply drift away from the desired periodic solution, or even diverge in free-flight cases. The primary consequence is that one must compute the trim for every model used for stability, loads, vibrations or performance analyses. Reference [20] provides an exhaustive analysis of rotorcraft trim and presents a complete list of relevant references on this topic.

Rotary wing vehicles are complex aeroelastic systems which are modeled with comprehensive finite element-based analysis tools [2, 24]. These tools provide the ability to render with a high level of detail the various sub-systems of the vehicle, including the most complex one, the rotor system. Rotorcraft codes are coupled with time-accurate aerodynamic models, ranging from dynamic inflow [21] to free-wake models [5] all the way to computational fluid dynamics, which is based on first principles and can account for the main rotor wake distortion, the interactions of shed vortex filaments with rotor blades, fuselage and tail rotor, or the dynamic stall effects on the retreating blade. Hence, rotorcraft aeroelastic analyses are typically based on complex, large, highly non-linear, multi-field models. In this research, our main concern is the efficient trim of such models. We will discuss in the following sections how the complexity and the non-linearity of comprehensive rotorcraft models call for advanced trim methodologies.

1.1 Rotorcraft Trim Formulation

1.1.1 Governing Equations

We want to represent the rotorcraft by a virtual prototype $\widetilde{\mathcal{M}}$. A general form of the governing equations for this system is the following:

$$\widetilde{\boldsymbol{f}}(\dot{\widetilde{\boldsymbol{x}}}, \widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{u}}) = 0, \tag{1a}$$

$$\widetilde{\boldsymbol{c}}(\dot{\widetilde{\boldsymbol{x}}}, \widetilde{\boldsymbol{x}}) = 0. \tag{1b}$$

The first set of equations, (1a), represents the equations of dynamic equilibrium and the kinematic equations, and the second set, (1b), represents possible holonomic and non-holonomic constraint conditions. The vector $\widetilde{\boldsymbol{x}} \in \mathbb{R}^{n_x}$ denotes the system states, $\widetilde{\boldsymbol{\lambda}} \in \mathbb{R}^{n_\lambda}$ are the Lagrange multipliers associated with multibody formulations in which the constraints (1b) are enforced, $\widetilde{\boldsymbol{u}} \in \mathbb{R}^{n_u}$ are the controls, and finally $\widetilde{\boldsymbol{f}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\lambda} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $\widetilde{\boldsymbol{c}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_\lambda}$.

The states $\widetilde{\boldsymbol{x}}$ can include displacements, rotations, linear and angular velocities, and possible internal states describing engine, actuator and damper dynamics. Controls $\widetilde{\boldsymbol{u}}$ may represent actuator inputs (voltages, valve apertures, etc.), applied forces, throttle position, and controlled joint relative displacements and rotations. If $\widetilde{\mathcal{M}}$ includes flexible components, the ordinary differential-algebraic equations (1) are obtained as the result of the discretization in the space dimension of the governing partial differential-algebraic equations. In this case, the states $\widetilde{\boldsymbol{x}}$ include degrees of freedom associated with the nodes of the spatial grids of the flexible components and/or modal amplitudes. For coupled problems, $\widetilde{\boldsymbol{x}}$ may also include states modeling the interacting fields, as for example aerodynamic states, and equations (1) will include appropriate discretizations of the corresponding partial-differential equations.

1.1.2 Trim Conditions and Constraints

As we mentioned before, one of the basic concepts of trim is that the controls must be constant in time. The *trim conditions* for the controls $\widetilde{\boldsymbol{u}}$ can then be written as

$$\dot{\widetilde{\boldsymbol{u}}} = 0, \quad \forall\, t. \tag{2}$$

Let us define a set of system outputs $\widetilde{\boldsymbol{y}} \in \mathbb{R}^{n_y}$ such that

$$\widetilde{\boldsymbol{y}} = \frac{1}{T}\int_t^{t+T} \widetilde{\boldsymbol{g}}(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{u}})\,\mathrm{d}t, \tag{3}$$

where T is the rotor period, and $\widetilde{\boldsymbol{g}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$. The exact physical meaning of the outputs $\widetilde{\boldsymbol{y}}$ will depend on the specific problem under consideration. In the case of free-flight applications, the outputs will typically represent the average of some global vehicle states which describe its gross motion, such as position, orientation, linear and angular velocity of a vehicle-embedded frame with respect to an inertial frame of reference. For example, for an embedded frame associated with a flexible fuselage model, the frame will be a floating frame of reference that describes the average fuselage rigid body states. In the simpler case of a rotor connected to the ground, the outputs $\widetilde{\boldsymbol{y}}$ will typically be some components of the hub loads expressed in the inertial frame.

The trimmed steady flight condition can then be defined by specifying desired values for the outputs of the system:

$$\widetilde{\boldsymbol{y}} = \boldsymbol{y}^*, \quad \forall\, t. \tag{4}$$

We will refer to this expression as the *trim constraints* on the outputs $\widetilde{\boldsymbol{y}}$.

As a matter of fact, we will always be able to satisfy a number of trim constraints lower or equal to the number of controls, so that $n_y \leq n_u$. If $n_y = n_u$ and a trim solution exists, the trim constraints implicitly define the trim controls and attitude, and the solution is generally unique. On the contrary, if $n_y < n_u$ we can encounter a whole set of solutions, among which we can look for the one that optimizes a specific merit function, for example the power required in trim or the average yaw angle.

1.1.3 Periodicity Conditions

As previously mentioned, trim corresponds to a particular solution for the rotorcraft system, characterized by the harmonic response of all its states. For simple rotor models, every degree of freedom passes through the same state every rotor revolution, so

$$\widetilde{\boldsymbol{x}}(t+T) = \widetilde{\boldsymbol{x}}(t), \quad \forall\, t. \tag{5}$$

We call these relations *periodicity conditions*. Reference [20] treats the problem of unknown rotor period in detail. Here, we want to emphasize that a rotorcraft system is often trimmed in a condition of non-zero average linear or angular velocities. Therefore, some of the states are not strictly periodic. Consider for instance a rotorcraft trimmed to non-zero average linear velocities expressed in an inertial frame. The corresponding position coordinates have in this case average values growing linearly with time. So, we generalize conditions (5) in the following way:

$$\widetilde{\boldsymbol{x}}(t+T) = \widetilde{\boldsymbol{x}}(t) + \widetilde{\boldsymbol{z}}, \quad \forall\, t, \tag{6}$$

where $\widetilde{\boldsymbol{z}}$ is a function of states and controls that renders the variation of $\widetilde{\boldsymbol{x}}$ in trim over the rotor period T. Reference [20] indicates these as "quasi-periodicity conditions." For the degrees of freedom that have zero average velocity over a period, the last term vanishes, so they simply respect the periodicity conditions (5).

1.2 Trim Solution Strategies

Some of the several methods for rotorcraft trim appeared in the specialized literature over the last thirty years are the following:

- In the *harmonic balance method*, the states are described by superposition of harmonics with integer multiples of the rotor frequency. For each harmonic, all the terms with that frequency are equated, and it is possible to solve for the amplitudes of all the harmonics for every degree of freedom in trim.

- In *quasi-Newton methods*, the controls are initially guessed, the system is integrated in time until transients decay and the error on the trim constraints is checked. A Newton method is applied to update the controls and iterate. This approach generally requires the adoption of springs connected to the ground and is not indicated for unstable systems.

- The *periodic shooting* is based on a Newton-like iterative method. The equations of motion are integrated for a rotor period for given initial conditions and (constant)

controls, then the error on the trim constraints is checked and the initial conditions and controls are updated.

- A fourth strategy is to discretize the problem in time, explicitly enforce trim and periodicity constraints and assemble a large non-linear algebraic problem, to be solved again in an iterative fashion. This is sometimes referred to as the *finite elements in time* method[2].

The last approach often found is that of auto-pilots, which is the one here considered. In this case, we augment the system with a control law that closes the loop and steers the system towards the desired trim condition. This control law can be, for example, a set of dynamic equations coupled with those of the system, as in the case of the classic auto-pilots presented in the literature so far. Differently from the previous approaches and because of this control law, the controls have their own dynamic behavior and no explicit constraint is enforced to have constant controls when the trim constraints are fulfilled, so limit cycles are often observed on the controls. The presence of these limit cycles indicates that the desired steady-state condition has not been reached. On the other hand, auto-pilots are extremely efficient when they are applied to very large and complex systems, since they depend on the number of controls and outputs of the system and not on the number of degrees of freedom.

1.3 Scope of the Research and Highlights of the Investigated Methodology

In this research, we propose the formulation of a new auto-pilot based on non-linear model-predictive control. This approach uses a reduced model of rotorcraft, which must be able to adequately approximate the controlled system. Thus, an adaptive neural element is adopted to match model and system responses.

The new auto-pilot has the following features:

- NMPC provides a framework for guaranteeing stability of the non-linear closed-loop system.

[2]We honestly consider the name *finite elements in time* inappropriate, since it refers to the applied discretization in time rather than the actual adopted solution strategy.

- The use of a non-linear reduced model for the prediction of the system response typically implies superior performance of the controller with respect to other approaches.

- The solution specifically accounts for the presence of the constant-in-time constraints on the control actions in the prediction problem. The effect of constraints is difficult to incorporate in other control approaches, and this often results in limit-cycle oscillations of the inputs, as previously noted.

- The fidelity of the reduced model to the plant is crucial for the performance of the model predictive approach. As the rotorcraft is flown, the reduced model learns the characteristics of the system through its adaptive nature. Hence, there is no need of tuning the model to different vehicles or to different flight conditions.

- Given the powerful characteristics of neural networks for the identification of non-linear dynamical systems [17], the controller is applicable to virtually any rotorcraft mathematical model without modifications or the need for manual tuning, including models with complex aerodynamic effects.

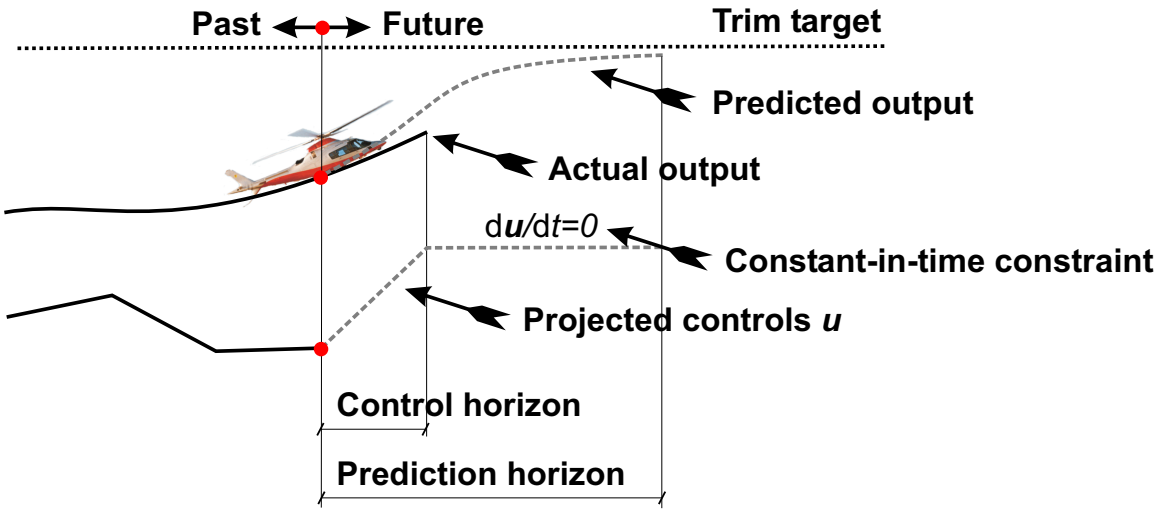

Figure 1: The model-predictive auto-pilot.

The basic principle of the non-linear model-predictive auto-pilot is illustrated in Figure 1. A non-linear reduced model of the vehicle is used for predicting the future behavior of the plant, i.e. the rotorcraft model, under the action of the control inputs $\boldsymbol{u}$. An open-loop

optimal control problem is solved for the reduced model on a finite horizon (the prediction window). For trim problems, the optimization cost function is defined as the square of the violation of the trim constraints.

The optimizer accounts for possible input and output constraints that may need to be satisfied. In particular, the control actions can vary in time only within the control horizon, while they become constant-in-time from the end of the control horizon to the end of the prediction horizon. Hence, the solution of the open-loop model-predictive regulation problem yields control actions that generate predicted outputs which follow the prescribed target in some optimal, constraint satisfying, way.

The controls computed by the optimizer are now used to steer the plant, but only on a short time horizon (typically equal to the control horizon). In fact, due to the inevitable mismatch between reduced model and plant, the actual outputs will drift away from the predicted ones. Once the plant has reached the end of the steering window 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.

Our formulation of the problem uses an adaptive non-linear reduced model which is composed of two elements:

1. A simple analytic reference model of the rotor that is responsible for capturing the gross features of the plant response, thereby easing the training of the adaptive element. In this work, the reference model computes the rotor forces and moments by combining actuator disk and blade element theory, considering a uniform inflow. The rotor attitude is evaluated by means of quasi-steady flapping dynamics with a linear aerodynamic damping correction [22].

2. A single-hidden-layer neural network, whose role is to approximate the error between the prediction of the reference model and the actual response measured on the plant. The neural network is trained on-line as the rotorcraft model is flown by the NMP

controller, using an error-correction learning algorithm that adjusts the network parameters [11].

As we already mentioned, commonly used auto-pilots can be very difficult to use. In fact, a proportional controller can work nicely for the problem for which it was tuned. However, it must be carefully tuned to show a stable behavior on a wide range of conditions. Since no rules generally exist to compute proper controller parameters to guarantee stability, most of this tuning phase is based on trial and error.

Therefore, in developing the proposed methodology, attention was concentrated on constructing a robust tool that could be applied to the largest possible class of problems. Moreover, this tool had to need minimal information coming from simulations of the comprehensive model. The following formal description of the proposed controller justifies the superior performance and robustness it will show in Chapter 4:

- The stability requirement is addressed by the NMPC methodology. It guarantees the convergence of the non-linear closed-loop system to the desired target under the assumptions of an infinite prediction horizon and that the reduced model perfectly matches the controlled system.

- Clearly, the practical implementation of the methodology implies the use of finite prediction and control horizons. Different approaches to achieve closed-loop stability using finite horizon lengths exist. A very common idea is the introduction of *stability constraints*. For example, the system can be required to reach the target solution in finite time (at the end of the prediction) by enforcing proper constraints, the so called *zero terminal constraints*, on the outputs. Another possibility is to enforce a *terminal region constraint* (the output is required to reach a ball about the solution at the end of the prediction) and/or a *terminal penalty term* in the cost function, which weights the distance of the states from the target at the end of the time window. A different path is followed in the *quasi-infinite horizon* approach. In this case, by using a locally stabilizing linear control law, it is possible to compute off-line the terminal region and a terminal penalty matrix to be used in solving the NMPC problem. See

Reference [12] for a list of references on this subject.

- On the other hand, a perfect match of reference model and controlled system can not be expected in general. So, although a good reference model certainly eases its work, the adaptive element present in the reduced model plays a central role. For the neural structure used as adaptive element in this research, the fundamental result of the universal approximation property (see Reference [13]) implies that the approximation error can be made arbitrarily small by changing the number of internal units (neurons) of the network. Therefore, given a particular structure of the network, a set of parameters of the adaptive element exist that give the desired accuracy of the reduced model.

- Analytically, the adaptive technique applied in this research cannot be proved to be effective in finding this combination of network parameters and in guaranteeing the stability of the closed-loop system. However, published theoretical results on this topic show that proper adaptive rules can be constructed to ensure tracking stability. For example in Reference [15], feedback control based on a neural adaptive element is used to make robot manipulators track desired outputs. A tuning algorithm is therein adopted to show that states, tracking error and network weights are uniformly ultimate bounded, which implies in practice the convergence of all these quantities to a bounded region. The effect of different adaptation algorithms on the proposed predictive controller will be investigated in future works.

he stability requirement is addressed as we said by the NMPC methodology, which guarantees the convergence of the non-linear closed-loop system to the desired target under the assumptions of an infinite prediction horizon and that the reduced model perfectly matches the controlled system [12]. Clearly, the practical implementation of the methodology implies the use of a finite time horizon, which is however a reasonable approximation that does not prejudice the stability of the system in most cases. On the other hand, a perfect match of reduced model and controlled system can not be expected in general. So, although a good reference model certainly eases its work, the adaptive element present in the reduced

model plays a central role. For the neural structure used for the adaptive element, the fundamental result of the universal approximation property (presented in Reference [13]) implies that the approximation error can be made arbitrarily small by changing the number of internal units (neurons) of the network. However, the adaptive technique applied in this research can not be proved to be effective in finding the combination of network parameters minimizing this error. The results obtained in practice, gathered in Chapter 4, seem in any case to be particularly promising and motivate at the end this research effort.

1.4 Contents of the Thesis

The plan of this thesis is as follows. In Chapter 2 we formulate the model-predictive control problem. In doing this, we assume the knowledge of a "good" reduced model. We also discuss the discretization in time and solution of the predictive control problem. Chapter 3 is devoted to the construction of adaptive reduced models to accurately approximate the controlled system. The numerical results and analyses of the proposed methodology are gathered in Chapter 4, while in Chapter 5 we draw some conclusions and we discuss possible future steps of this research effort. Details of the rotor reference model used are briefly shown in Appendix A. Appendix B presents a classic simplified approach for rotorcraft performance analysis that will be useful in the numerical test phase, and Appendix C gives some highlights on the classical auto-pilot approach. We close the work extending the proposed auto-pilot to free-flight applications in Appendix D, for future research on this topic.

CHAPTER II

MODEL PREDICTIVE CONTROL

Non-linear model-predictive control arose in the 1970s and grew in the 1980s as a practical and sound methodology for the control of non-linear systems. A predictive controller predicts the future behavior of the plant using a reduced model, and finds the control actions necessary for the regulation of the plant solving an optimal control problem. The control strategy is applied to the system for a finite time window, then it is computed again by solving a new optimization problem. NMPC is very general and powerful, with respect to similar linear control strategies, since it allows to use nonlinear models of the plant and to explicitly enforce constraints on both state and control variables. The price for these advantages is a superior complexity and a high computational cost. Indeed, one needs to solve on-line an optimization problem at each control sampling interval to produce a prediction of the controls necessary to minimize the objective function.

In the past years, NMPC has been widely used in the chemical process industries and in other fields in which long time scale systems must be controlled or regulated with extreme stability and efficiency. At the present time, the control community is gradually moving towards real-time applications of the MPC by taking advantage of the available computational power, see for instance Reference [6].

In this chapter we will recast the rotorcraft trim problem introduced in Section 1.1 as a model-predictive regulation problem. We will focus our attention on the formulation and solution of the predictive control problem, and leave the discussion on the reduced model to Chapter 3.

2.1 Model Predictive Regulation

NMPC can be exploited for controlling a system in a variety of "optimal ways," depending on the objective function that is optimized in the process. As an example, for regulation

problems the controlled system is steered in order to track a desired output time history that is constant in time. So, it is quite straightforward to define the objective function as the error between the reduced model output and the desired one. In trimming a rotorcraft model for instance, one wants to regulate the outputs $\widetilde{\boldsymbol{y}}$ of the plant $\widetilde{\mathcal{M}}$, defined in equation (3), to the desired values $\boldsymbol{y}^*$, as expressed by the trim constraints (4).

NMPC uses a reduced model $\mathcal{M}$ for predicting the future behavior of the plant $\widetilde{\mathcal{M}}$ under the action of the control inputs $\boldsymbol{u}$. An open-loop optimal control problem is solved for the reduced model on a finite horizon, the *regulation window*. The cost function to be minimized translates the distance between the prediction of the model response and the desired response, and can weight, in case, some form of control activity. The optimizer accounts for possible input and output constraints that may need to be satisfied. In particular, it allows to enforce the trim condition of constant controls, expressed in equation (2). The solution of the open-loop model predictive regulation problem yields control actions that generate predicted outputs which follow the prescribed reference $\boldsymbol{y}^*$ in some optimal way that satisfies the constraints.

The controls computed by the optimizer are used for steering the plant $\widetilde{\mathcal{M}}$, but only on a short time horizon, the *steering window*. Indeed, due to the inevitable mismatch between reduced model and plant, the actual outputs will drift away from the predicted ones. Once the plant has reached the end of the steering window under the action of the computed control inputs, the model predictive regulation problem is solved again, looking ahead in the future over the prediction horizon shifted forward in time. This procedure results is a feedback, receding horizon approach.

2.2 Formulation of the Model Predictive Regulation Problem

Let us assume that the reduced model $\mathcal{M}$ is governed by the system of ordinary differential equations

$$\boldsymbol{f}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = 0, \tag{7}$$

representing a specific mathematical model based on insight on the nature of the plant, i.e. of system (1). The functional dependence can be related to the details that one wants to represent by means of the reduced model. We assume the parameters $\boldsymbol{p}^*$ to be given as a result of the adaption of the model. In Chapter 3 this topic will be investigated in detail.

Let $t = T_i^{\text{reg}} = T_i^{\text{steer}}$ be the current time, which is also the beginning of the regulation and steering windows, while $T_f^{\text{reg}} = T_i^{\text{reg}} + \Delta T^{\text{reg}}$ is the end of the regulation window of size ΔT^{reg}. Given initial conditions on the plant states $\widetilde{\boldsymbol{x}}(T_i^{\text{reg}}) = \widetilde{\boldsymbol{x}}_i$, which induce the output initial conditions $\widetilde{\boldsymbol{y}}_i = \widetilde{\boldsymbol{h}}(\widetilde{\boldsymbol{x}})|_{t=T_i^{\text{reg}}}$, the future control actions are found from the solution of the following model-predictive regulation problem:

$$\min_{\boldsymbol{y},\boldsymbol{u}} J^{\text{reg}}, \tag{8a}$$

$$\text{with: } J^{\text{reg}} = \int_{T_i^{\text{reg}}}^{T_f^{\text{reg}}} M(\boldsymbol{y}, \boldsymbol{y}^*, \boldsymbol{u})\, \mathrm{d}t, \tag{8b}$$

$$\text{s.t.: } \boldsymbol{f}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = 0, \tag{8c}$$

$$\boldsymbol{g}^{\text{reg}}(\boldsymbol{y}, \boldsymbol{u}) \in [\boldsymbol{g}_{\min}^{\text{reg}}, \boldsymbol{g}_{\max}^{\text{reg}}], \tag{8d}$$

$$\boldsymbol{y}(T_i^{\text{reg}}) = \widetilde{\boldsymbol{y}}_i. \tag{8e}$$

The regulation cost, J^{reg}, is computed as the integral over the regulation window of the function

$$M(\boldsymbol{y}, \boldsymbol{y}^*, \boldsymbol{u}) = ||\boldsymbol{y} - \boldsymbol{y}^*||_{\boldsymbol{S}_y^{\text{reg}}} + ||\boldsymbol{u}||_{\boldsymbol{S}_u^{\text{reg}}} + ||\dot{\boldsymbol{u}}||_{\boldsymbol{S}_{\dot{u}}^{\text{reg}}}, \tag{9}$$

where the first term accounts for the regulation error, while the second and third terms are quadratic terms in the control actions and control rates, respectively. These quantities are measured in the norms $||\bullet||_{\boldsymbol{S}_y^{\text{reg}}} = (\bullet) \cdot \boldsymbol{S}_y^{\text{reg}}(\bullet)$, $||\bullet||_{\boldsymbol{S}_u^{\text{reg}}} = (\bullet) \cdot \boldsymbol{S}_u^{\text{reg}}(\bullet)$ and $||\bullet||_{\boldsymbol{S}_{\dot{u}}^{\text{reg}}} = (\bullet) \cdot \boldsymbol{S}_{\dot{u}}^{\text{reg}}(\bullet)$ respectively, based on the scaling matrices $\boldsymbol{S}_y^{\text{reg}}$, $\boldsymbol{S}_u^{\text{reg}}$ and $\boldsymbol{S}_{\dot{u}}^{\text{reg}}$. The last two terms are typically used for ensuring smooth control policies, through appropriate choices of the weighting matrices. The solution of the optimization problem satisfies the reduced model governing equations (7), together with additional possible input and output constraints.

Since we are interested in enforcing the trim conditions expressed by equation (2), we enforce zero control velocities on the time interval $(T_c^{\text{reg}}, T_f^{\text{reg}})$ by means of the constraint

$$\dot{\boldsymbol{u}}(t) = 0, \quad T_i^{\text{reg}} < T_c^{\text{reg}} \leq t \leq T_f^{\text{reg}}, \tag{10}$$

which can be easily appended to the set of constraints (8d). T_c^{reg} represents the end of the *control horizon*, over which the controls $\boldsymbol{u}$ are free to change.

If the constant-in-time condition for the controls is enforced over a time interval larger than one rotor period, i.e. $T_f^{\text{reg}} - T_c^{\text{reg}} > T$, we are exactly asking that the model be trimmed at the end of the regulation window. The use of an adaptive element in the model allows the convergence of the behavior of the model to that of the system, as we will see in Chapter 3, and the enforcement of the trim condition not only for the former but also for the latter.

The prediction phase is followed by the plant steering phase. Consider the known controls $\boldsymbol{u}^*(t)$ as obtained from the solution of problem (8) above, with $t \in \Omega^{\text{steer}} = (T_i^{\text{steer}}, T_f^{\text{steer}})$, where $T_f^{\text{steer}} = T_i^{\text{steer}} + \Delta T^{\text{steer}}$ is the end of the steering window of size ΔT^{steer}. Under the action of the controls $\boldsymbol{u}^*$, the plant $\widetilde{\mathcal{M}}$ is advanced forward in time starting from the current state $\widetilde{\boldsymbol{x}}_i$. This steering phase amounts to the solution of the following initial value problem:

$$\widetilde{\boldsymbol{f}}(\dot{\widetilde{\boldsymbol{x}}}, \widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{\lambda}}, \boldsymbol{u}^*) = 0, \tag{11a}$$

$$\widetilde{\boldsymbol{c}}(\dot{\widetilde{\boldsymbol{x}}}, \widetilde{\boldsymbol{x}}) = 0, \tag{11b}$$

$$\widetilde{\boldsymbol{x}}(T_i^{\text{steer}}) = \widetilde{\boldsymbol{x}}_i, \tag{11c}$$

which yields a solution in terms of $\widetilde{\boldsymbol{x}}(t)$ and $\widetilde{\boldsymbol{\lambda}}(t)$ for $t \in \Omega^{\text{steer}}$. The solution at the end of the steering window, $\widetilde{\boldsymbol{x}}(T_f^{\text{steer}})$, provides the initial condition for the next regulation problem, and consequently for the next steering problem.

2.3 Numerical Solution of the Model Predictive Regulation Problem

The numerical solution of the model predictive regulation problem (8) can be obtained very efficiently by the direct transcription method [4]. The governing equations of the reduced model are discretized on a computational grid of the regulation window using an appropriate numerical method. This defines a set of discrete unknown state and control parameters on the computational grid. Next, the constraint conditions and the problem cost function are expressed in terms of the discrete parameters. This in turn defines a non-linear discrete parameter optimization problem, i.e. a non-linear programming problem

(NLP). The numerical solution of this discrete problem approximates the solution of its infinite-dimensional counterpart, problem (8).

To describe the numerical solution of the model predictive problem in more precise terms, let us consider the following partition of the temporal domain $\Omega^{\text{reg}} = (T_i^{\text{reg}}, T^{\text{reg}})$

$$T_i^{\text{reg}} \equiv t_0 < t_1 < \ldots < t_{n-1} < t_n \equiv T_f^{\text{reg}}, \tag{12}$$

where $n \geq 1$, and $t_i = t_{i-1} + h^i$, $i = 1, \ldots, n$. A grid $\mathcal{T}_h^{\text{reg}}$ is associated with this partition. $\mathcal{T}_h^{\text{reg}}$ is made of n elements, the generic element being labelled K. Each element spans a time interval $T^i = [t_i, t_{i+1}]$, and has a left vertex ∂K^L associated with time t_i and a right vertex ∂K^R associated with time t_{i+1}. Typically, the size of each element in the model predictive regulation problem is held constant throughout the grid, and hence we simply have $h^i = h = (T_f^{\text{reg}} - T_i^{\text{reg}})/n$.

Given a numerical integration scheme, we indicate with the symbols $\boldsymbol{y}_h$, $\boldsymbol{u}_h$ the finite dimensional approximations that the numerical scheme makes of the infinite dimensional unknown fields $\boldsymbol{y}(t)$, $\boldsymbol{u}(t)$. The restriction of these approximations to the generic grid element K is written $\boldsymbol{y}_h|_K$ and $\boldsymbol{u}_h|_K$. The state approximations evaluated on the right vertex of an element K_i are equal to the state approximations evaluated on the left vertex of the neighboring element K_{i+1}, i.e.

$$\boldsymbol{y}_h|_{\partial K_i^R} = \boldsymbol{y}_h|_{\partial K_{i+1}^L}. \tag{13}$$

This condition expresses the continuity of the states at the element interfaces, in the sense that it provides the initial conditions on each element as the value of the final states on the preceding element. In general, there is not a similar condition on the controls, since there are no initial conditions on this field, so that the control approximations $\boldsymbol{u}_h$ should be discontinuous across element interfaces.

On the computational grid $\mathcal{T}_h^{\text{reg}}$, $\boldsymbol{y}_h$ and $\boldsymbol{u}_h$ can be regarded as functions of some discrete parameters $\boldsymbol{y}_d$, $\boldsymbol{u}_d$, i.e. $\boldsymbol{y}_h = \boldsymbol{y}_h(\boldsymbol{y}_d)$ and $\boldsymbol{u}_h = \boldsymbol{u}_h(\boldsymbol{u}_d)$ on $\mathcal{T}_h^{\text{reg}}$. These discrete parameters depend on the specific numerical integration scheme. In general, the vector of discrete state parameters $\boldsymbol{y}_d$ will contain the state unknowns at the grid nodes, $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$. Furthermore, some schemes, as for example the Runge-Kutta and finite element methods, will also

have an additional n_s internal unknowns (stages) per time step $\boldsymbol{y}_i^1, \boldsymbol{y}_i^2, \ldots, \boldsymbol{y}_i^{n_s}$. Therefore, the vector of discrete state parameters can in general be written as

$$\boldsymbol{y}_d = (\ldots, \boldsymbol{y}_i, \boldsymbol{y}_i^1, \boldsymbol{y}_i^2, \ldots, \boldsymbol{y}_i^{n_s}, \boldsymbol{y}_{i+1}, \ldots)^T. \tag{14}$$

Similarly for the vector of discrete control parameters $\boldsymbol{u}_d$, according to the numerical method.

The discretized version of the model predictive regulation problem can be written as

$$\min_{\boldsymbol{y}_d, \boldsymbol{u}_d} J_h^{\mathrm{reg}}, \tag{15a}$$

$$\text{with: } J_h^{\mathrm{reg}} = \int_{T_0^{\mathrm{reg}}}^{T^{\mathrm{reg}}} M(\boldsymbol{y}_h, \boldsymbol{y}_h^*, \boldsymbol{u}_h)\, \mathrm{d}t, \tag{15b}$$

$$\text{s.t.: } \boldsymbol{f}_h(\boldsymbol{y}_h|_K, \boldsymbol{u}_h|_K, \boldsymbol{p}^*) = 0 \quad \forall K \in \mathcal{T}_h^{\mathrm{reg}}, \tag{15c}$$

$$\boldsymbol{g}_h^{\mathrm{reg}}(\boldsymbol{y}_h|_K, \boldsymbol{u}_h|_K) \in [\boldsymbol{g}_{\min,K}^{\mathrm{reg}}, \boldsymbol{g}_{\max,K}^{\mathrm{reg}}] \quad \forall K \in \mathcal{T}_h^{\mathrm{reg}}, \tag{15d}$$

$$\boldsymbol{y}_h(T_i^{\mathrm{reg}}) = \widetilde{\boldsymbol{y}}_i. \tag{15e}$$

In problem (15), the first set of constraint conditions (15c) represents a discretized version of the reduced model governing equations (7) on each element K of the computational grid. These equations are coupled together through the gluing conditions (13) that induce a banded sparsity pattern to the problem. The second set of constraints (15d) represents a discretized version of the input and output constraints of problem (8), again expressed on each grid element K. Finally, the cost J_h in (15b) represents a discretized version of J as given in (8b), the integral being evaluated through some appropriate quadrature rule. Discrete cost and constraints are all functions of the vectors of discrete state and control parameters $\boldsymbol{y}_d$, $\boldsymbol{u}_d$, which represent the unknowns of the optimization problem.

In this work, the transcription process is based on the finite element in time formulation of References [7, 9].

The plant steering phase amounts to the solution of the initial value problem (11) with known control actions. $\widetilde{\mathcal{T}}_h^{\mathrm{steer}}$ is the grid used for advancing forward in time the comprehensive model in Ω^{steer}. Note that the typical time step size in $\widetilde{\mathcal{T}}_h^{\mathrm{steer}}$ is in general smaller than the typical time step size in $\mathcal{T}_h^{\mathrm{reg}}$, since much finer solution scales need to

be resolved in this case. Furthermore, in some applications it can be useful to adapt the time step size to the local characteristics of the solution using some error control method. It is also clear that the numerical method used for integrating the comprehensive model equations can in general differ from the numerical method used for the discretization of the optimal control problem. Therefore, there is a need to map the controls $\boldsymbol{u}_h^*$ obtained on $\mathcal{T}_h^{\text{reg}}$ from the solution of problem (15) onto the steering grid $\mathcal{T}_h^{\text{steer}}$. This mapping depends on the grids and on the numerical methods used in the prediction and steering phases, and will be simply indicated here with the notation

$$\boldsymbol{u}_h^*|_{\widetilde{\mathcal{T}}_h^{\text{steer}}} = \mathcal{P}(\boldsymbol{u}_h^*|_{\mathcal{T}_h^{\text{reg}}}), \tag{16}$$

where $\mathcal{P}(\bullet)$ is an appropriate projection operator.

The discretized version of the steering problem (11) can be written as

$$\widetilde{\boldsymbol{f}}_h(\widetilde{\boldsymbol{x}}_h|_K, \widetilde{\boldsymbol{\lambda}}_h|_K, \boldsymbol{u}_h^*|_K) = 0 \quad \forall K \in \widetilde{\mathcal{T}}_h^{\text{steer}}, \tag{17a}$$

$$\widetilde{\boldsymbol{c}}_h(\widetilde{\boldsymbol{x}}_h|_K) = 0 \quad \forall K \in \widetilde{\mathcal{T}}_h^{\text{steer}}, \tag{17b}$$

$$\widetilde{\boldsymbol{x}}_h(T_i^{\text{steer}}) = \widetilde{\boldsymbol{x}}_i. \tag{17c}$$

The discrete equations (17a,17b) are solved on each element sequentially. For the first element, the initial conditions are provided as (17c), while for all subsequent elements they are given by the gluing conditions (13). Finally, the outputs are obtained as

$$\widetilde{\boldsymbol{y}}_h|_K = \widetilde{\boldsymbol{h}}(\widetilde{\boldsymbol{x}}_h)|_K, \quad \forall K \in \widetilde{\mathcal{T}}_h^{\text{steer}}. \tag{18}$$

As part of the tasks of this research, we will use a multibody rotor prototype as comprehensive model. The numerical integration of the multibody dynamics equations is based on the non-linearly unconditionally stable energy decaying scheme described in Reference [3] and references therein.

The formulated model predictive control approach imply the following approximations:

1. The predicted and actual outputs will differ because of modeling errors, since the reduced model will in general be able to only approximate, but not render exactly, the dynamics of the plant outputs. This error can be controlled by a judicious choice

of the reduced model and by adapting the model parameters, as described in the following chapter.

2. The prediction of the future behavior of the system is carried out on a finite horizon, rather than up to T_∞. Shorter horizons imply smaller computational costs in the solution of the model predictive regulation problem. On the other hand, longer horizons imply improved stability of the controller and better regulation performance. Hence, there is a trade-off among these conflicting requirements.

3. The control actions are recomputed by the optimizer only after a finite time interval ΔT^{steer}. Since predicted and plant models differ because of point (1) above, the outputs of system $\widetilde{\mathcal{M}}$ will drift away from the predicted ones under the action of the controls. Clearly, the greater the steering window, the greater the effect of this drift will be. On the other hand, longer steering windows imply a smaller number of solutions of the model predictive regulation problem for a given maneuver duration, and hence a reduced computational cost. Hence, here again there is a trade-off between these contrasting requirements.

CHAPTER III

ADAPTIVE REDUCED MODEL

For the solution of the NMPC problem (8) we assumed the knowledge of a reduced model that is a good approximation of the controlled system. One could think to simply formulate the regulation problem (8) using $\mathcal{M} = \widetilde{\mathcal{M}}$. However, the direct solution of the resulting optimal control problem is not numerically attractive and acceptable in practice, because of the large number of states characterizing a comprehensive model. A practical and feasible approach is the adoption of a reduced model $\mathcal{M}$ based on classic and compact blade element models, see Reference [22]. This idea has been recently applied with satisfactory results to the steering of rotorcraft models in free-flight [8].

Nevertheless, in order to properly trim a rotorcraft prototype by a predictive controller, we need a reduced model able to adequately represent the system behavior at least locally, i.e. about the trimmed solution, in terms of flexibility, unsteady aerodynamic effects and inflow dynamics. Since no model exists that satisfies these constraints at a reasonable computational cost, one is induced to look in the direction of the identification of the behavior of the complete comprehensive model.

System identification is concerned with the mathematical characterization of a system. Let us think of the system $\widetilde{\mathcal{M}}$ as an operator, belonging to a class $\widetilde{\mathbb{M}}$, that maps the space of the possible input signals into the space of the output signals. These signals are typically vectors of functions of time of suitable regularity. The target of the identification problem is to find an operator $\mathcal{M}$, belonging to a class $\mathbb{M} \subset \widetilde{\mathbb{M}}$, which approximates $\widetilde{\mathcal{M}}$ in a desired sense. So, if $\widetilde{\boldsymbol{u}}$ and $\widetilde{\boldsymbol{y}}$ are respectively the input and output signals of $\widetilde{\mathcal{M}}$, we look for $\mathcal{M} \in \mathbb{M}$ such that its outputs $\boldsymbol{y}$ respect

$$\left\|\widetilde{\boldsymbol{y}} - \boldsymbol{y}\right\| = \left\|\widetilde{\mathcal{M}}(\widetilde{\boldsymbol{u}}) - \mathcal{M}(\widetilde{\boldsymbol{u}})\right\| \leq \varepsilon, \tag{19}$$

where ε is the maximum acceptable approximation error. In our case, we are interested in

finding a model $\mathcal{M}$ such that it shares with the actual rotorcraft system $\widetilde{\mathcal{M}}$ those dynamic characteristics that are relevant for trim.

An identification methodology is characterized by the class of identification models $\mathbb{M}$ and the method to determine $\mathcal{M}$. Desired accuracy, analytical tractability, computational efficiency and simplicity in finding $\mathcal{M}$ are the aspects that concur in the choice of the methodology for each application.

In the following sections, we briefly introduce the reference model adopted for this research. Then, we discuss its augmentation in order to get the reduced model described by equation (7), that here we repeat

$$\boldsymbol{f}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = 0,$$

described in terms of a set of states $\boldsymbol{y}$, controls $\boldsymbol{u}$ and parameters $\boldsymbol{p}$. Indeed, the dependence on the vector of parameters $\boldsymbol{p}$ is introduced to perform the identification. A particular combination $\boldsymbol{p}^*$ fully characterizes the reduced model $\mathcal{M}$ among the models $\mathcal{M}(\boldsymbol{p})$ that form the class $\mathbb{M}$ of operators. These parameters are optimized to ensure proper matching between reduced and full outputs, when the reduced and full systems are subjected to the same inputs. Analytically, the reduced model is determined in order to obtain

$$\boldsymbol{y} \approx \widetilde{\boldsymbol{y}} \quad \text{for} \quad \boldsymbol{u} = \widetilde{\boldsymbol{u}}. \tag{20}$$

3.1 Reference Model

The reduced model is obtained as a reference model augmented by an element catching the deficiencies between the reference and the full plant. By reference model we want here to indicate a specific mathematical model based on insight on the nature of the system (1). This model can be expressed in the form

$$\boldsymbol{f}_{\text{ref}}(\dot{\hat{\boldsymbol{y}}}, \hat{\boldsymbol{y}}, \boldsymbol{u}) = 0, \tag{21}$$

where $\hat{\boldsymbol{y}}$ are the outputs of the reference model subjected to the control time history $\boldsymbol{u}$. The level of detail of the reference model is clearly a crucial aspect. On one side, the solution of the optimal control problem (8) implies that the whole reduced model must be

computationally very efficient, so the reference model cannot be complex. On the other hand, a more refined reference model clearly eases the identification process since it is typically "closer" to the rotorcraft system, in the sense previously discussed. Appendix A summarizes the reference model adopted for this research, which is based on blade element theory, with uniform inflow computed by means of momentum theory [10].

We can notice that the controls of the plant, $\widetilde{\boldsymbol{u}}$, might have physical meanings that differ from those of the reduced model controls $\boldsymbol{u}$, since the two models describe the system at two different levels of detail. For example, the controls $\widetilde{\boldsymbol{u}}$ might include the valve apertures of three hydraulic actuators connected to the swash-plate of a detailed multibody model of the rotor system, while the reduced model controls will include the rotor collective, longitudinal and lateral cyclic angles. These two sets of controls, although clearly different, eventually determine the pitch setting of each rotor blade at any given instant of time, which in turn determines the forces and moments generated by the rotor. Irrespectively of the details and of the problem considered, it is reasonable to assume that it will be always possible to map one set of controls into the other, $\widetilde{\boldsymbol{u}} = \boldsymbol{m}(\boldsymbol{u})$ and viceversa, $\boldsymbol{u} = \boldsymbol{m}^{-1}(\widetilde{\boldsymbol{u}})$. We will just consider $\widetilde{\boldsymbol{u}} = \boldsymbol{u}$ in the following.

3.2 Reference Model Augmentation and Error Identification

We will show now two possible approaches for the augmentation of the reference model and the construction of the reduced model to be used in the model-predictive auto-pilot.

3.2.1 Output Error Identification

A first possibility is to identify the error between the system and the reference model outputs when they are subjected to the same input. Let us define the reference model as in equation (21), where $\hat{\boldsymbol{y}}$ is the vector of outputs of the reference model subjected to the same controls as the system, i.e. $\boldsymbol{u} = \widetilde{\boldsymbol{u}}$. We can now write

$$\boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{e}, \tag{22}$$

where the output error $\boldsymbol{e}$ is a function of time. If we enforce equation (20) and substitute $\boldsymbol{y} = \widetilde{\boldsymbol{y}}$, the vector $\boldsymbol{e}$ becomes the measure of the mismatch we want to identify.

Under proper assumptions on its regularity, the dynamics of $\boldsymbol{e}$ can be described by a (unknown) system of differential equations

$$\boldsymbol{h}(\boldsymbol{e}^{(n)}, \dots, \boldsymbol{e}, \boldsymbol{u}) = 0, \tag{23}$$

where we used the notation $\boldsymbol{e}^{(n)}$ to indicate the derivative of order n of the output error with respect to time, and $\boldsymbol{e}^{(n)}, \dots, \boldsymbol{e}$ to indicate the sequence of the derivatives of order decreasing form n to zero. This system is completed by proper initial conditions for the error and its derivatives, $\boldsymbol{e}^{(k)}(T_i) = \boldsymbol{e}_0^{(k)}, \quad k = 0, \dots, n-1$. By construction, the operator $\boldsymbol{h}$ is able to generate values of $\boldsymbol{e}$ such that $\boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{e} = \widetilde{\boldsymbol{y}}$.

The operator $\boldsymbol{h}$ can be characterized as belonging to an infinite-dimensional class of operators $\mathbb{H}$. We approximate this operator by choosing a finite-dimensional class, subset of $\mathbb{H}$, parameterized in $\boldsymbol{p}$, so that

$$\boldsymbol{h}_p \in \mathbb{H}_p(\boldsymbol{p}) \subset \mathbb{H}, \tag{24}$$

where $\boldsymbol{h}_p$ approximates $\boldsymbol{h}$ as desired, namely

$$\boldsymbol{h} = \boldsymbol{h}_p + \boldsymbol{\varepsilon}, \tag{25}$$

$\boldsymbol{\varepsilon}$ indicating the approximation error. As a result, we have the following reduced model:

$$\boldsymbol{f}_{\text{ref}}(\dot{\hat{\boldsymbol{y}}}, \hat{\boldsymbol{y}}, \boldsymbol{u}) = 0, \tag{26a}$$

$$\boldsymbol{h}_p(\boldsymbol{e}_p^{(n)}, \dots, \boldsymbol{e}_p, \boldsymbol{u}, \boldsymbol{p}^*) = 0, \tag{26b}$$

$$\boldsymbol{y}_p = \hat{\boldsymbol{y}} + \boldsymbol{e}_p, \tag{26c}$$

where $\boldsymbol{y}_p$ approximates $\widetilde{\boldsymbol{y}}$ as desired.

3.2.2 Equation Defect Identification

A more appealing formulation is obtained by writing the reduced model governing equations (7) as follows:

$$\boldsymbol{f}_{\text{ref}}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}) - \boldsymbol{d}(\boldsymbol{y}^{(n)}, \dots, \boldsymbol{y}, \boldsymbol{u}) = 0, \tag{27}$$

where again we used $\boldsymbol{y}^{(n)}$ to indicate the derivative of order n of the outputs with respect to time. We define the (unknown) operator $\boldsymbol{d}$ as the defect of equations (21) when the condition (20) is enforced, i.e. $\boldsymbol{u} = \widetilde{\boldsymbol{u}}$ and $\boldsymbol{y} = \widetilde{\boldsymbol{y}}$. Thus, the operator respects

$$\boldsymbol{d}(\widetilde{\boldsymbol{y}}^{(n)}, \ldots, \widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ref}}(\dot{\widetilde{\boldsymbol{y}}}, \widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{u}}). \tag{28}$$

This form of the reduced model ensures the matching of reduced and full states when it is subjected to the same inputs as the plant. The functional dependence of the defect is related to the dynamic characteristics of the system governed by equations (1).

The operator $\boldsymbol{d}$ belongs to an infinite-dimensional class of non-linear functions mapping the derivatives of the states $\boldsymbol{y}$ and the controls $\boldsymbol{u}$ into a vector of defects, formally

$$\boldsymbol{d} \in \mathbb{D}. \tag{29}$$

Since this operator is unknown, we want to identify it by choosing an approximating operator $\boldsymbol{d}_p$ belonging to a finite-dimensional class $\mathbb{D}_p$ parameterized in $\boldsymbol{p}$, so that

$$\boldsymbol{d}_p \in \mathbb{D}_p(\boldsymbol{p}) \subset \mathbb{D}. \tag{30}$$

and

$$\boldsymbol{d} = \boldsymbol{d}_p + \boldsymbol{\varepsilon}. \tag{31}$$

Here again $\boldsymbol{\varepsilon}$ represents the approximation error.

The resulting reduced model is in this case

$$\boldsymbol{f}_{\text{ref}}(\dot{\boldsymbol{y}}_p, \boldsymbol{y}_p, \boldsymbol{u}) - \boldsymbol{d}_p(\boldsymbol{y}_p^{(n)}, \ldots, \boldsymbol{y}_p, \boldsymbol{u}, \boldsymbol{p}^*) = 0, \tag{32}$$

with $\boldsymbol{y}_p$ being the desired approximation of $\widetilde{\boldsymbol{y}}$.

3.3 Approximation by Neural Networks and Discretization

In the last five decades artificial neural networks, or simply neural networks (NNs), have changed from promising model of parallel processing inspired by biological systems to a widely used technique for the approximation of non-linear maps. Here, we do not want to emphasize the biological implications of neural networks. On the contrary, we simply

recognize the valuable theoretical and practical results allowed by NNs in the fields of system identification and control. Indeed, neural models are powerful tools for the approximation of non-linear dynamical systems since they allow a constructive decomposition of a non-linear map in a finite-dimensional space of non-linear functions. Because of this feature, a neural element is chosen to approximate the mismatch between reference model and plant and to perform the identification.

Let us summarize the reduced models obtained in the previous sections, taking $n = 1$ for simplicity: for the output error identification case

$$\boldsymbol{f}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = \begin{cases} \boldsymbol{f}_{\text{ref}}(\dot{\hat{\boldsymbol{y}}}, \hat{\boldsymbol{y}}, \boldsymbol{u}) &= 0, \\ \boldsymbol{h}_p(\dot{\boldsymbol{e}}, \boldsymbol{e}, \boldsymbol{u}, \boldsymbol{p}^*) &= 0, \\ \boldsymbol{e} - (\boldsymbol{y} - \hat{\boldsymbol{y}}) &= 0, \end{cases}$$

and for the defect identification case

$$\boldsymbol{f}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = \boldsymbol{f}_{\text{ref}}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}) - \boldsymbol{d}_p(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = 0,$$

where we dropped the subscripts from the variables for uniformity with equation (7). NNs are parametric non-linear maps that can be used to render the approximating operator present in each of the two approaches [17].

3.3.1 Recurrent Neural Network Approximating the Output Error

For the output error, we can use a recurrent network to render the system of differential equations $\boldsymbol{h}_p(\dot{\boldsymbol{e}}, \boldsymbol{e}, \boldsymbol{u}, \boldsymbol{p}^*) = 0$. Recurrent networks have a peculiar feedback form, corresponding to a time-varying map, such that the outputs of the network at each instant in time have an influence on the values at the following instants. Thus, even subjected to constant controls ($\boldsymbol{u}$, in this case) the network generates time-varying output signals (here $\boldsymbol{e}$). Recurrent NNs are typically expressed in explicit and discretized form [18] as

$$\boldsymbol{e}_{i+1} = \boldsymbol{W}^T \boldsymbol{\sigma}(\boldsymbol{V}_e^T \boldsymbol{e}_{i+1}^{\phi} + \boldsymbol{V}_u^T \boldsymbol{u}_{i+1}^{\phi} + \boldsymbol{b}_V) + \boldsymbol{b}_W, \tag{33}$$

where $\boldsymbol{W}$, $\boldsymbol{V}_e$ and $\boldsymbol{V}_u$ are matrices of synaptic weights, $\boldsymbol{b}_V$ and $\boldsymbol{b}_W$ are vectors of biases, and

$$\boldsymbol{\sigma}(\boldsymbol{\phi}) = (\sigma(\phi_1), \ldots, \sigma(\phi_{N_h}))^T$$

is the vector-valued function of activation functions $\sigma(\bullet)$, typically chosen as the sigmoid function, for the N_h processing elements (neurons). We used $\boldsymbol{e}_{i+1}^{\phi}$ and $\boldsymbol{u}_{i+1}^{\phi}$ to indicate the arrays

$$\boldsymbol{e}_{i+1}^{\phi} = \begin{bmatrix} \boldsymbol{e}_i \\ \\ \boldsymbol{e}_{i-1} \end{bmatrix}, \quad \boldsymbol{u}_{i+1}^{\phi} = \begin{bmatrix} \boldsymbol{u}_i \\ \\ \boldsymbol{u}_{i-1} \end{bmatrix}.$$

In the present case, the considered structure is the NNARX, which is often applied to identify dynamical systems with negligible noise [18]. In practice, this implies the dependence of the values at each time only upon the network inputs and outputs at previous time instants. Moreover, we choose a structure for the approximation space including two previous values. Clearly, the resulting approximation error depends upon the order of the system of differential equations (23), which we do not know *a priori*. Typically, the ability of this structure to render the dynamical system under study is checked *a posteriori*, and in case it is augmented with other values back in time.

3.3.2 Static Neural Network Approximating the Equation Defect

In the equation defect identification, the operator $\boldsymbol{d}_p$ is a static map between its inputs and outputs. Thus, we can use a static network to render the approximation space, as follows:

$$\boldsymbol{d}_p(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}) = \boldsymbol{W}^T \boldsymbol{\sigma}(\boldsymbol{V}_{\dot{y}}^T \dot{\boldsymbol{y}} + \boldsymbol{V}_y^T \boldsymbol{y} + \boldsymbol{V}_u^T \boldsymbol{u} + \boldsymbol{b}_V) + \boldsymbol{b}_W. \tag{34}$$

As mentioned before, the network inputs must include as many derivatives of $\boldsymbol{y}$ as it is necessary to guarantee a sufficiently small approximation error. The reduced model with defect identification in the form (34) can be easily discretized as in equation (15c), in explicit or implicit form depending on the numerical scheme. This aspect implies a more flexible formulation with respect to the output error identification. Moreover, the characterization of a simple static map instead of a dynamic one makes this second approach potentially more efficient and the identification task easier.

3.3.3 Properties of the Neural Representation

In both cases, we adopt a single-hidden-layer feedforward network structure. This is made evident by the presence of one layer of N_h neurons connected to the network inputs and

outputs by the interconnection weights $\boldsymbol{V}$ and $\boldsymbol{W}$, respectively. Indeed for this particular structure, the universal approximation property of neural networks [13] ensures that the approximation error can be made arbitrarily small, i.e. it can be bounded as $||\boldsymbol{\varepsilon}||_2 \leq C_\varepsilon$ for any $C_\varepsilon > 0$, for some appropriately large number of hidden neurons N_h. This feature makes single-hidden-layer feedforward networks widely accepted tools for identification of dynamical systems.

Finally, the reduced model parameters $\boldsymbol{p}$ are defined as the synaptic weights and biases of the network, for example

$$\boldsymbol{p} = (\ldots, W_{ij}, \ldots, V_{ij}, \ldots, b_{V,i}, \ldots, b_{W,i}, \ldots)^T. \tag{35}$$

3.4 Model Adaptation

As we mentioned at the beginning of this chapter, the reduced model parameters are optimized in order to minimize the distance between $\mathcal{M}$ and $\widetilde{\mathcal{M}}$ in some norm. Proper matching between predicted and actual outputs can be obtained by subjecting reduced model and plant to the same control actions, and by tuning the reduced model parameters so that some measure of mismatch between the two is minimized. The neural network is trained with an error-correction learning algorithm, where the network parameters are adjusted to minimize some error E between the network outputs and the desired outputs. For the two cases previously discussed, if $\widetilde{\boldsymbol{y}}^*$ are the plant outputs obtained for given control inputs $\boldsymbol{u}^*$, the error can be defined as a function of the parameters $\boldsymbol{p}$ at each time instant as follows:

- for the output error identification

$$E(t) = \left\|\boldsymbol{e}^* - \boldsymbol{e}_p^*\right\|_2 = \left\|(\widetilde{\boldsymbol{y}}^* - \hat{\boldsymbol{y}}^*) - \boldsymbol{e}_p^*\right\|_2, \tag{36}$$

  where $\hat{\boldsymbol{y}}^*$ and $\boldsymbol{e}_p^*$ are obtained using the reference model (21) and the neural approximation (33), respectively, with given controls $\boldsymbol{u}^*$.

- for the equation defect identification

$$E(t) = \left\|\boldsymbol{d}(\dot{\widetilde{\boldsymbol{y}}}^*, \widetilde{\boldsymbol{y}}^*, \boldsymbol{u}^*) - \boldsymbol{d}_p(\dot{\widetilde{\boldsymbol{y}}}^*, \widetilde{\boldsymbol{y}}^*, \boldsymbol{u}^*, \boldsymbol{p})\right\|_2. \tag{37}$$

Several methods are available for solving this optimization problem [11]. Let us indicate with $\boldsymbol{u}_h^*$ some given control inputs, and with $\widetilde{\boldsymbol{y}}_h^*$ the resulting outputs of the comprehensive model obtained by solving numerically equations (17) and (18). The model adaptation problem can be formulated as the following optimization problem

$$\min_{\boldsymbol{y}_d, \boldsymbol{p}} J_h^{\text{adapt}}, \tag{38a}$$

$$\text{with: } J_h^{\text{adapt}} = \int_{T_i^{\text{adapt}}}^{T_f^{\text{adapt}}} E(\boldsymbol{y}_h, \widetilde{\boldsymbol{y}}_h^*, \boldsymbol{u}_h^*)\, \mathrm{d}t, \tag{38b}$$

$$\text{s.t.: } \boldsymbol{f}_h(\boldsymbol{y}_h|_K, \boldsymbol{u}_h^*|_K, \boldsymbol{p}, T) = 0 \quad \forall K \in \mathcal{T}_h^{\text{adapt}}. \tag{38c}$$

The problem is defined over a time interval starting at T_i^{adapt} and ending at T_f^{adapt}, and solved numerically through transcription on a grid $\mathcal{T}_h^{\text{adapt}}$. The discretized reduced model equations (38c) on each grid element provide a set of constraints for the optimization. The cost function (38b) is an integral measure of the overall mismatch error.

In this research, we choose the initial and final times to be $T_i^{\text{adapt}} = T_i^{\text{steer}}$ and $T_f^{\text{adapt}} = T_f^{\text{steer}}$, i.e. they coincide with the initial and final time, respectively, of the current steering problem. The idea is to use the local information provided by the last steering of the plant to correct the current estimate $\boldsymbol{p}_{\text{curr}}$ of the parameters. An updated value of the parameters $\boldsymbol{p}_{\text{new}}$ can be obtained with the back-propagation algorithm [23], i.e. using the steepest-descent search direction

$$\boldsymbol{p}_{\text{new}} = \boldsymbol{p}_{\text{curr}} - \eta \frac{\partial J_h^{\text{adapt}}}{\partial \boldsymbol{p}}\bigg|_{\boldsymbol{p}_{\text{curr}}}, \tag{39}$$

where η is the so called "learning rate," which represents a tunable parameter to control the network adaption speed.

CHAPTER IV

NUMERICAL RESULTS

In this chapter we discuss the numerical investigations conducted using the proposed auto-pilot.

In doing this, we will respect the conceptual order we followed in testing the procedure.

- First, in Section 4.1 we check the ability of the auto-pilot to practically achieve a desired trimmed condition. For this preliminary test we use a simplified rigid rotor model.
- Then, in Section 4.2, we apply the procedure to a rotor multibody model:
  - As a first step (Subsection 4.2.2), we present a comparison of the proposed output error identification and equation defect identification approach, introduced in Chapter 3, for different operative conditions.
  - Finally (Subsection 4.2.3), we compare the predictive auto-pilot and an available implementation of the classic auto-pilot algorithm. The latter is optimized for different conditions, and several measures are introduced to show the superior performance of the designed NMP auto-pilot.

4.1 Preliminary Tests

This section is devoted to the introduction of the wind-tunnel rotor trim problem and the discussion of some preliminary tests we made to explore the potentiality of the NMP auto-pilot. The chosen approach will be the one of the output error identification introduced in Chapter 3.

The problem considers the wind-tunnel trim for three desired values of the rotor force components in the fixed frame. The plant is a rigid four-bladed rotor attached to the ground. Only flap and pitch hinges, with an offset from the rotor rotation axis, are present

and the blades have uniform inertial and aerodynamic characteristics, with no built-in twist. The pitch of each blade as a function of time can be easily expressed in terms of the rotor controls, which are the collective, and longitudinal and lateral cyclic. For simplicity, we used strip theory to describe the aerodynamics of the system.

4.1.1 Implementation Aspects

In the proposed algorithm, the prediction phase consists in solving the regulation problem (8), which is tackled by means of the direct transcription and translated in problem (15). In practice, this problem can be viewed as an optimization problem with generic nonlinear cost and constraints. This kind of problems, generally called nonlinear programming (NLP) problems, are potentially large, but they can be efficiently solved in different ways. In our case, the solution is performed using a sequential quadratic programming (SQP), Newton-like method (see Reference [1]). Since the numerical solution of optimization problems can be highly sensitive to badly scaled unknowns and constraints, in practice we generally look for a solution in terms of scaled variables. In the present case we can define the scaled states $\boldsymbol{y}_s$ and scaled controls $\boldsymbol{u}_s$ as follows:

$$\boldsymbol{y}_s = \boldsymbol{Y}_v^{-1}\left(\boldsymbol{y} - \boldsymbol{y}_a\right), \quad \boldsymbol{u}_s = \boldsymbol{U}_v^{-1}\left(\boldsymbol{u} - \boldsymbol{u}_a\right), \tag{40}$$

with

$$\boldsymbol{Y}_v = \mathrm{diag}(\boldsymbol{y}_v), \quad \boldsymbol{U}_v = \mathrm{diag}(\boldsymbol{u}_v), \tag{41}$$

$\boldsymbol{y}_a, \boldsymbol{u}_a, \boldsymbol{y}_v$ and $\boldsymbol{u}_v$ being vectors of scaling values which produce scaled vectors $\boldsymbol{y}_s$ and $\boldsymbol{u}_s$ of order $\mathcal{O}(1)$. The choice of these scaling vectors is generally guided by the following considerations: taking for example the state i, if y_{a_i} is chosen as the state time average and y_{v_i} as the maximum deviation of the state from the average, then the absolute value of y_{s_i} is always less than or equal to 1.

In the following, the chosen rotor reference model is a simple analytical representation based on blade element theory, which we report in Appendix A. This reference model can be formally written as

$$\boldsymbol{f}_{\mathrm{ref}}(\dot{\hat{\boldsymbol{y}}}, \hat{\boldsymbol{y}}, \boldsymbol{u}) = \boldsymbol{f}_{\mathrm{ref}}(\hat{\boldsymbol{y}}, \boldsymbol{u}) = \hat{\boldsymbol{y}} - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{u})\,\mathrm{d}t = 0,$$

since no internal states are present and the forces produced by the rotor are considered functions only of the controls $\boldsymbol{u}$.

4.1.2 Formulation of the Output Error Identification

As reference model, we use a simple analytical representation based on the blade element theory, which we report in Appendix A. This model can be formally written as

$$\boldsymbol{f}_{\text{ref}}(\dot{\hat{\boldsymbol{y}}}, \hat{\boldsymbol{y}}, \boldsymbol{u}) = \boldsymbol{f}_{\text{ref}}(\hat{\boldsymbol{y}}, \boldsymbol{u}) = \hat{\boldsymbol{y}} - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{u})\,\mathrm{d}t = 0,$$

since the reference model does not have internal states and the forces produced by the rotor are considered functions only of the controls $\boldsymbol{u}$. The reduced model reads in this case

$$\boldsymbol{f}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = \begin{cases} \hat{\boldsymbol{y}} - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{u})\,\mathrm{d}t &= 0, \\ \boldsymbol{h}_p(\dot{\boldsymbol{e}}, \boldsymbol{e}, \boldsymbol{u}, \boldsymbol{p}^*) &= 0, \\ \boldsymbol{e} - (\boldsymbol{y} - \hat{\boldsymbol{y}}) &= 0, \end{cases}$$

For the neural network training, we can compute the output error between system and model as

$$\boldsymbol{e} = \widetilde{\boldsymbol{y}}^* - \hat{\boldsymbol{y}} = \widetilde{\boldsymbol{y}}^* - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{u}^*)\,\mathrm{d}t = \boldsymbol{f}_{\text{ref}}(\widetilde{\boldsymbol{y}}^*, \boldsymbol{u}^*). \tag{42}$$

4.1.3 Test 1: Hover

At first, we tested the algorithm with zero far field velocity. We arbitrarily set the collective, and longitudinal and lateral cyclic to 4.5, -1.5 and 0.5 deg, respectively, and we simulated forward in time the system response throughout the ensuing transient until a periodic solution was reached. The time average over the rotor period of the force components was then measured; this provided the target desired values for the trim problem. In fact, since at this point one knows that certain assigned controls produce known values of the force components, one can validate the controller by checking whether it is able to find the solution starting from some different initial setting of the controls.

Next, we set the three controls to zero and we let the auto-pilot steer the rotor model. During flight to the trim set, we limited the maximum control rates to 10 deg/sec. The ability offered by the predictive control approach to constrain the control signal as desired

has the potential to avoid undesired aerodynamic effects when dealing with complex aerodynamic models. The controller was activated with a frequency of 4/rev and a prediction horizon of 5 rotor revolutions. The recurrent neural network used 20 neurons in the internal layer, it was initialized with small random values, and was trained adaptively throughout the maneuver.

Higher activation frequencies and longer prediction windows would be preferable to ensure greater controller stability, however the chosen combination allowed a less burdensome computation in developing and testing the methodology.

The resulting time histories of rotor control inputs are given in Figure 2. It can be observed that the auto-pilot quickly steers the system back to the desired control values, indicated in the figure with dashed lines. No limit-cycles are observed in the solution. At the beginning, the controller operates at the maximum allowed control rate, then the adaptive element of the reduced model allows the controller to exactly catch the correct values of the controls and the target outputs (Figure 3), which are approximately obtained at the tenth revolution. In Figure 4 and 5 we show the time histories of the objective function J^{reg} of the regulation problem (8), which represents the prediction of the violation of the trim constraints (4), and the actual constraint violation, respectively. After a quick transient, both quantities converge to a small value, which is the residual error due to numerical discretization and to the discrete action of the regulator. This residual error seems to be compatible with the required accuracy of the considered problems. Finally, Figure 6 shows the actual reaction forces, which are averaged on a rotor revolution to obtain the outputs.

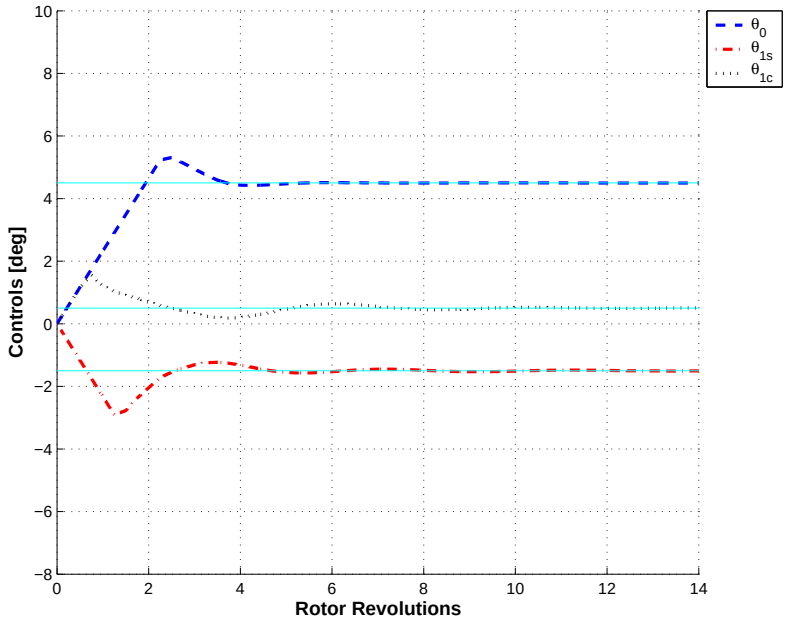

Figure 2: Preliminary tests: NMP auto-pilot control time histories for zero far field velocity (solid lines: target values).

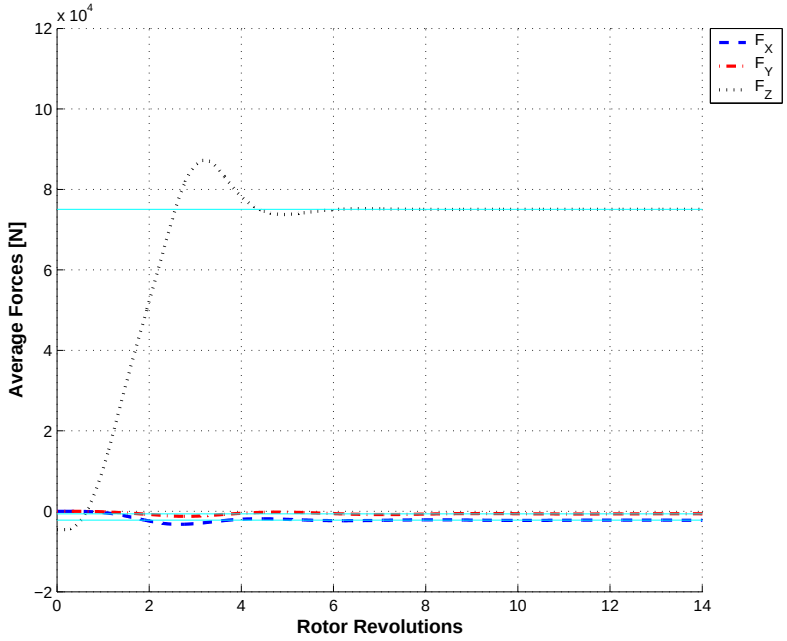

Figure 3: Preliminary tests: NMP auto-pilot average force time histories for zero far field velocity (solid lines: target values).

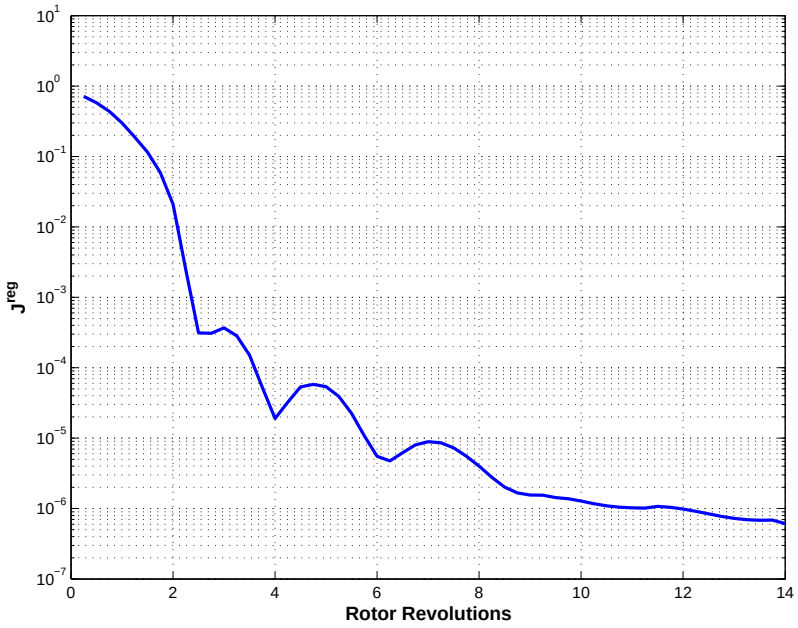

Figure 4: Preliminary tests: time history of the (scaled) cost function J^{reg} for zero far field velocity.

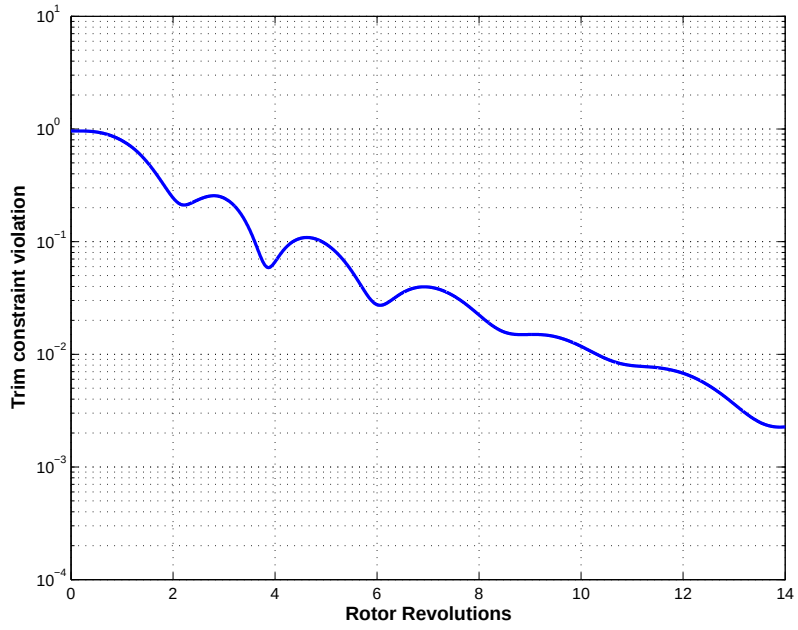

Figure 5: Preliminary tests: time history of the (scaled) violation of the trim constraints for zero far field velocity.

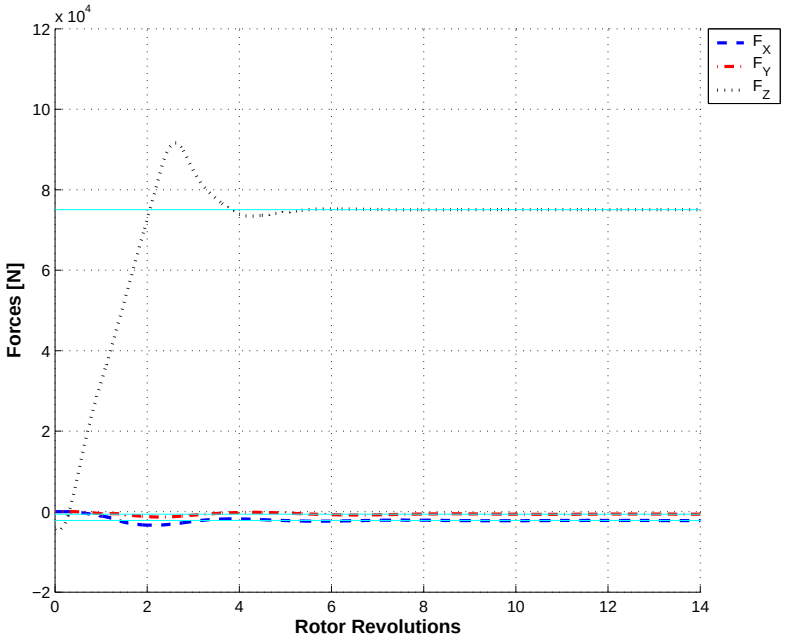

Figure 6: Preliminary tests: NMP auto-pilot force time histories for zero far field velocity (solid lines: target values).

4.1.4 Test 2: Forward Flight

The algorithm has also been tested with a far field velocity of 50 m/sec directed perpendicularly to the rotor axis of rotation. We proceeded as in the previous example: we set the three controls to 8.0, -6.0 and 2.0 deg, respectively, we simulated forward in time, measuring the average of the force components over the rotor period, and finally we set the three controls to zero and we activated the auto-pilot. Control rates were limited to 10 deg/sec, the controller activation frequency was 4/rev and a prediction horizon of 5 rotor revolutions was used. The recurrent neural network is similar to that of the previous case and is adaptively trained during the whole control phase.

The results for this second example are given in Figures 7-11. The observed trends are similar to those of the previous trim, with convergence to the solution obtained at about the fourteenth revolution.

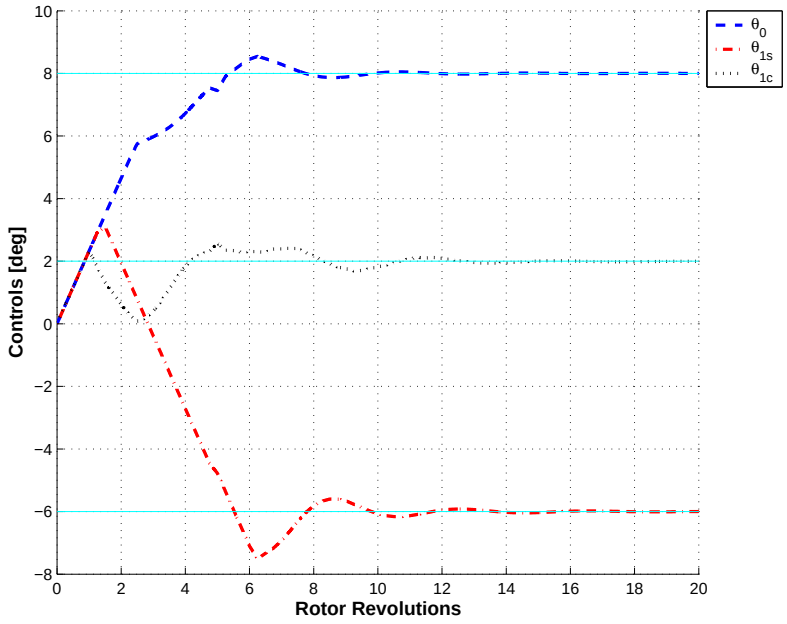

Figure 7: Preliminary tests: NMP auto-pilot control time histories for a far field velocity of 50 m/sec (solid lines: target values).

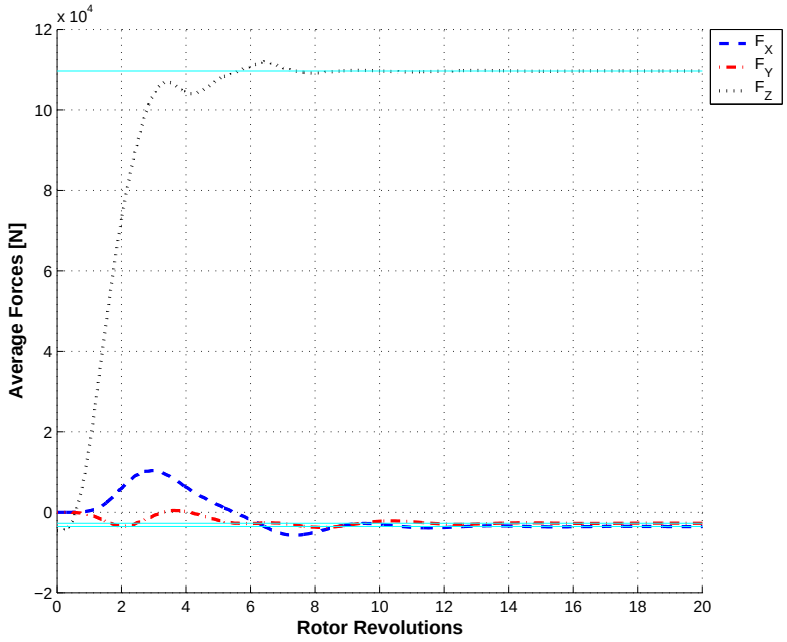

Figure 8: Preliminary tests: NMP auto-pilot average force time histories for a far field velocity of 50 m/sec (solid lines: target values).

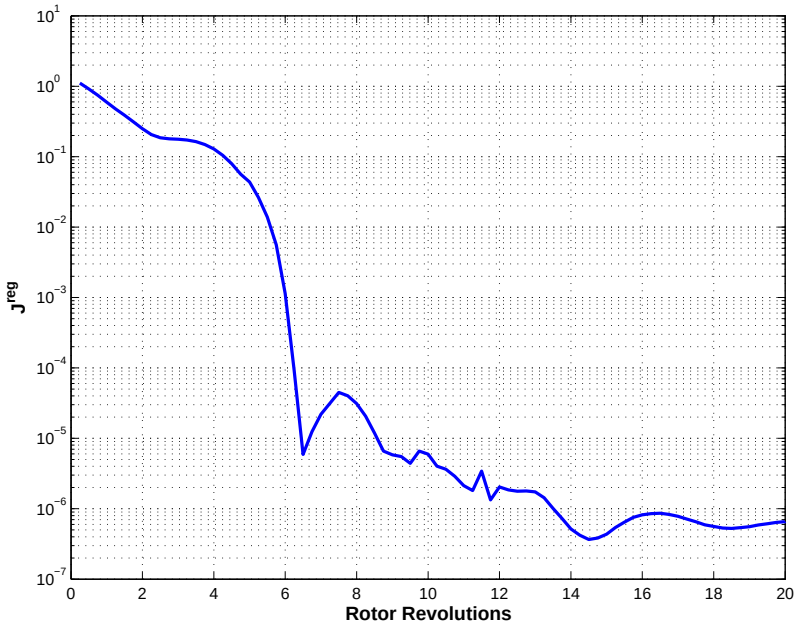

Figure 9: Preliminary tests: time history of the (scaled) cost function J^{reg} for a far field velocity of 50 m/sec.

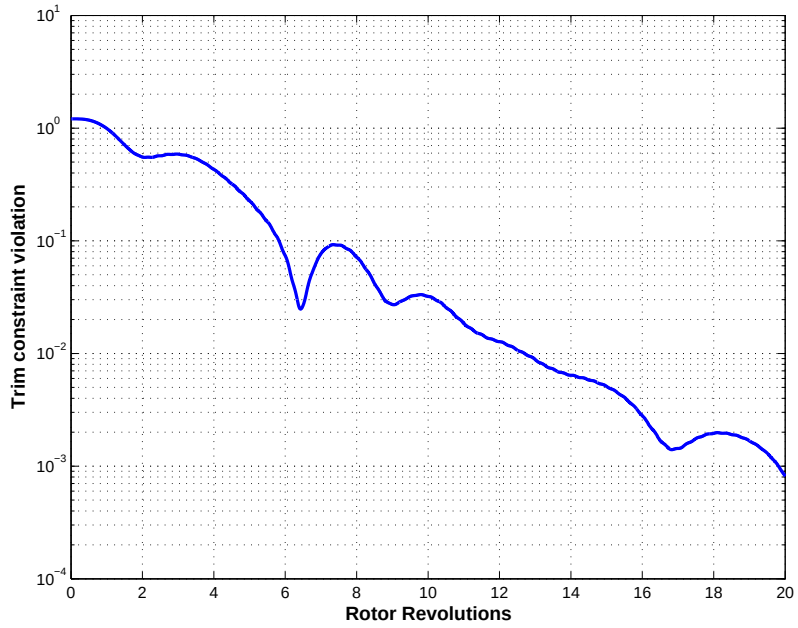

Figure 10: Preliminary tests: time history of the (scaled) violation of the trim constraints for a far field velocity of 50 m/sec.

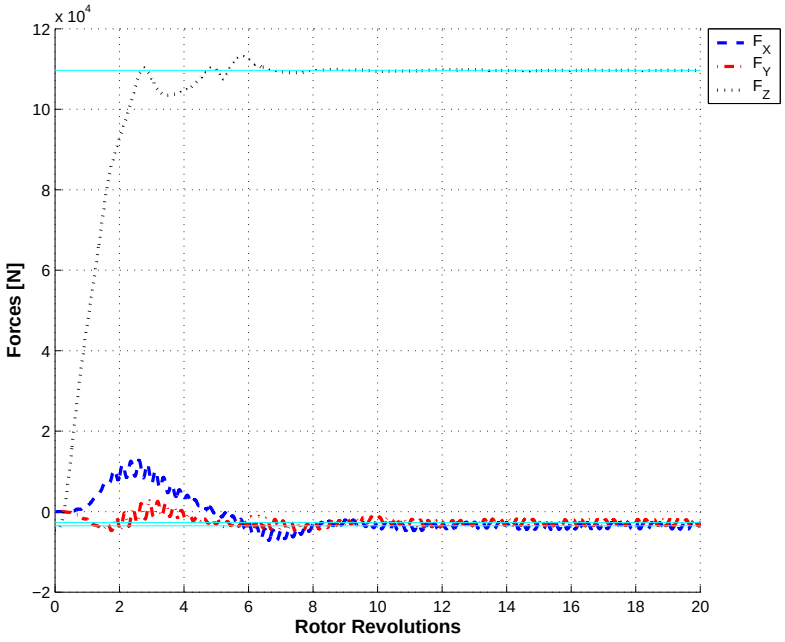

Figure 11: Preliminary tests: NMP auto-pilot force time histories for a far field velocity of 50 m/sec (solid lines: target values).

4.2 Wind-Tunnel Trim of a Rotor Multibody Model

In this Section, a multibody model of the rotor of the UH-60 utility helicopter, based on the finite element multibody approach described in References [2, 3], is used to perform two different tasks: at first, to study the formulations with output error identification and equation defect identification introduced in Chapter 3; then, to compare the behavior of the proposed NMP auto-pilot and an available implementation of the classic auto-pilot described in Reference [19]. The four-bladed articulated rotor model is characterized by the typical flap-lag-pitch configuration (from the hub to the blade), with three coincident hinges offset from the rotor shaft axis. The aerodynamic and structural properties of the blades are provided by means of look-up tables. The rotor controls are in this case the rotation of the pitch hinge of each blade, which can be related to the rotor collective θ_0, the longitudinal cyclic θ_{1s} and the later cyclic θ_{1c} as follows:

$$\theta_i(\psi) = \theta_0 + \theta_{1s} \sin\left(\psi - \frac{\pi}{2}\, i\right) + \theta_{1c} \cos\left(\psi - \frac{\pi}{2}\, i\right), \quad i = 1, 2, 3, 4,$$

where ψ is the azimuthal angle of the rotor and each value of i indicates one particular blade. At the beginning of each test, the collective, longitudinal and lateral cyclic are set to 4.01 deg (0.07 rad), 0.0 and 0.0 deg, respectively, and the rotor response follows the periodic orbit related to this constant controls. This condition is obtained by simulating the system forward in time with constant controls until all the transients decay. Then, the chosen controller is activated.

Higher accuracy of the model could be achieved by using proper aerodynamic modules to render the rotor wake effect. However, we will simply use strip theory to represent the rotor aerodynamics. Despite this lack of accuracy, we can consider this multibody model a good test-bed for the proposed auto-pilot, because of the presence of flexible elements with complex geometric, inertial and aerodynamic properties.

4.2.1 Formulation of the Equation Defect Identification

Here again we use the analytical reference model of Appendix A. We recall that this model can be formally expressed in the following way:

$$\boldsymbol{f}_{\text{ref}}(\hat{\boldsymbol{y}}, \boldsymbol{u}) = \hat{\boldsymbol{y}} - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{u})\,\mathrm{d}t = 0,$$

where the outputs $\hat{\boldsymbol{y}}$ are the rotor force components averaged on a revolution. Using the equation defect identification, the reduced model reads

$$\boldsymbol{f}(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = \boldsymbol{y} - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{u})\,\mathrm{d}t - \boldsymbol{d}_p(\dot{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = 0.$$

In order to train the neural network, we have to measure the model equation defect, as follows

$$\boldsymbol{d}_p(\dot{\widetilde{\boldsymbol{y}}}^*, \widetilde{\boldsymbol{y}}^*, \boldsymbol{u}^*, \boldsymbol{p}^*) = \boldsymbol{f}_{\text{ref}}(\widetilde{\boldsymbol{y}}^*, \boldsymbol{u}^*) = \widetilde{\boldsymbol{y}}^* - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{u}^*)\,\mathrm{d}t. \tag{43}$$

4.2.2 Comparison of Output Error and Equation Defect Identification

We are ready to investigate possible differences between the output error identification strategy, which exploits a recurrent neural network to perform the task, and the equation defect identification strategy, which is associated to a static neural network trained to catch the measured defect. Here again the controller activation frequency is equal to 4/rev, and

the control rates are limited to 10 deg/sec. The prediction horizon is equal in this case to 3 rotor revolutions.

Both networks are characterized by 20 neurons in the internal layer, and they are initialized with small random values and trained adaptively throughout the maneuver. In particular, the neural networks are modified right after the plant steering phase and before a new control activation, using the last available datum obtained integrating the multibody model. At each update, we use an adaptive approach for the learning rate η. Starting from an initial value $\eta = 0.3$, we update the network weights as described in Section 3.4, then we check the error $\boldsymbol{e}^* - \boldsymbol{e}_p^*$, for the output error case, or $\boldsymbol{d} - \boldsymbol{d}_p$, for the equation defect case, of the updated network on the last datum. If an increment in one of the components of this error is registered with respect to the error of the old network weights on the same data, the update is rejected, the learning rate is decreased and the process is repeated. In general, this procedure was rarely necessary.

To test the algorithm on a realistic practical problem, we compute the three approximate rotor force components required to trim the UH-60 as a function of the flight speed, following the relations summarized in Appendix B. We now show some results for values of the advance ratio μ equal to 0 and 0.35.

- $\mu = 0$. At first, we can appreciate the similarity between Figure 12 and 13 and the results of Figure 2. The performance of the MPC auto-pilot applied to the multibody model is almost the same as the one registered with the simplified rigid rotor model. Comparing now the output error and equation defect identification, the time histories of all the quantities are analogous (see Figures 12-13, 14-15 and 16-17), though the latter seems to perform slightly better.

- $\mu = 0.35$. Also in this case the two different implementations of the predictive auto-pilot seem to behave in a similar manner. The controller seems to behave in a very stable and efficient way, achieving the trim condition in about 15 revolutions. The controls of the equation defect identification implementation appear a bit more smooth (Figures 18-19), producing a slightly better trend of the outputs. Figures 20-21 and

22-23 show the time histories for the outputs and the actual forces, respectively.

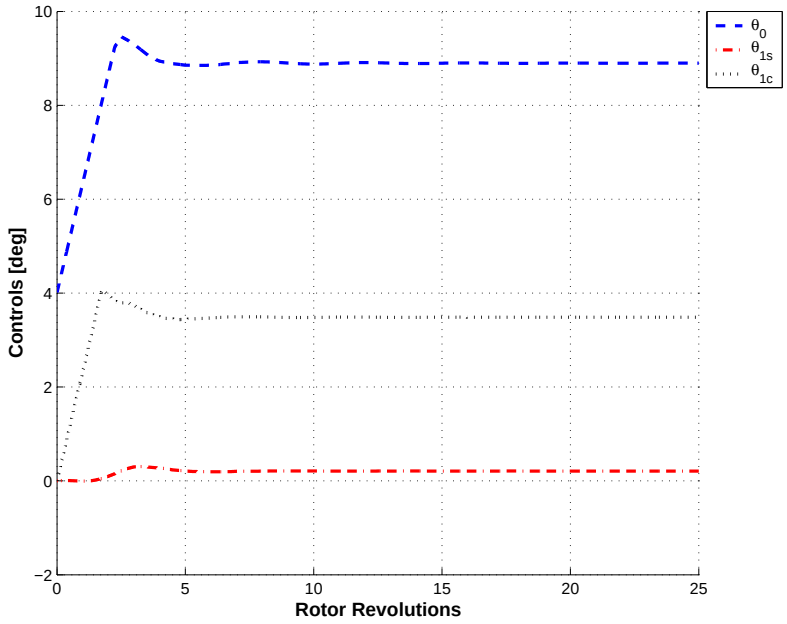

Figure 12: Control time history for the NMP auto-pilot with output error identification, $\mu = 0$.

These numerical experiments elucidate that the two proposed approaches do not radically differ one from one another when dealing with the considered problem. If we recall equation (42) and (43), it is clear that we are identifying the same signal, $\widetilde{\boldsymbol{y}}^* - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{u}^*)\,\mathrm{d}t$, using two different approaches. Moreover, since the desired trim condition corresponds to certain constant values of the controlled system outputs, the identification process consists mainly in finding the constant shift of the reference model outputs such that, subjected to the trim controls, it produces the same outputs as the plant. Thus, because of the structure of the adaptive elements, given in equations (33) and (34), the principal task of the neural networks is to identify the proper vector of biases $\boldsymbol{b}_W$, in order to obtain this correspondence. This discussion justifies the fact that for the present example the two adaptive elements work and behave in a very similar manner.

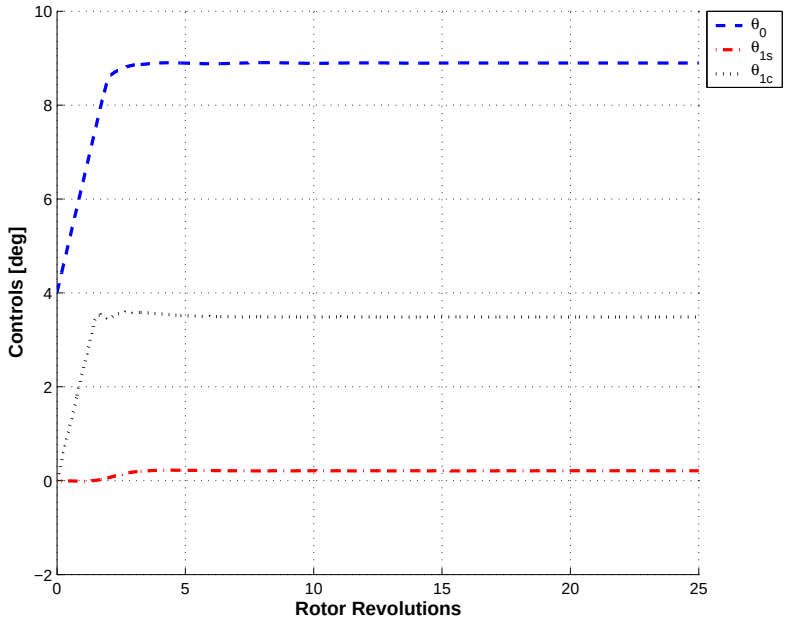

Figure 13: Control time history for the NMP auto-pilot with equation defect identification, $\mu = 0$.

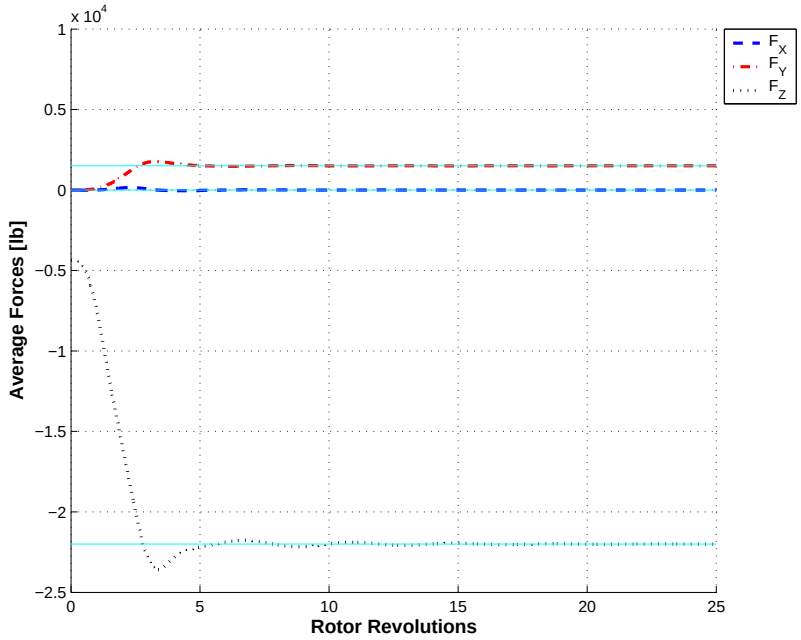

Figure 14: Output time history for the NMP auto-pilot with output error identification, $\mu = 0$ (solid lines: target values).

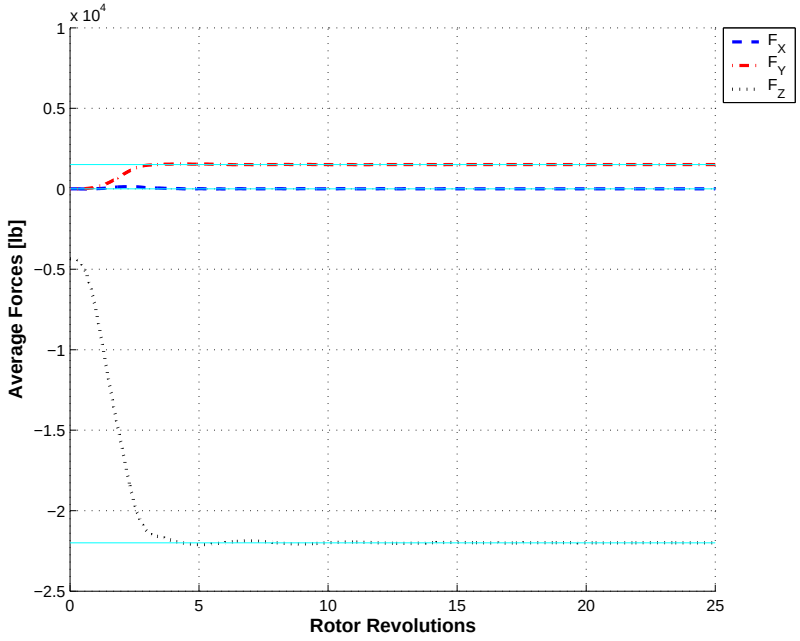

Figure 15: Output time history for the NMP auto-pilot with equation defect identification, $\mu = 0$ (solid lines: target values).

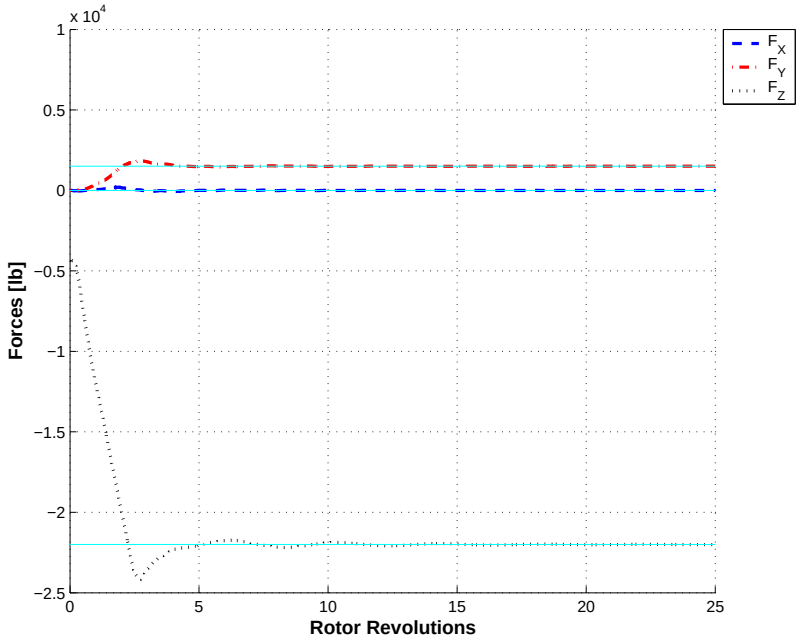

Figure 16: Force component time history for the NMP auto-pilot with output error identification, $\mu = 0$ (solid lines: target values).

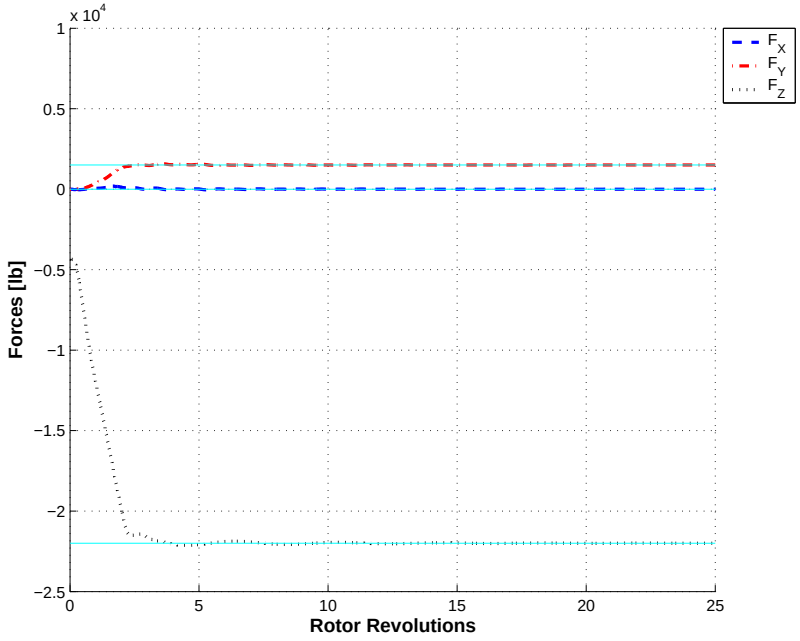

Figure 17: Force component time history for the NMP auto-pilot with equation defect identification, $\mu = 0$ (solid lines: target values).

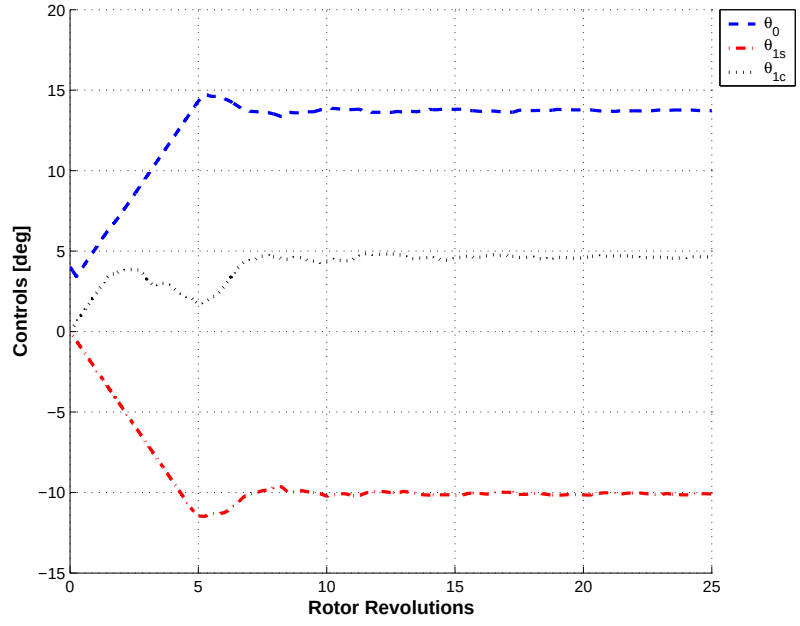

Figure 18: Control time history for the NMP auto-pilot with output error identification, $\mu = 0.35$.

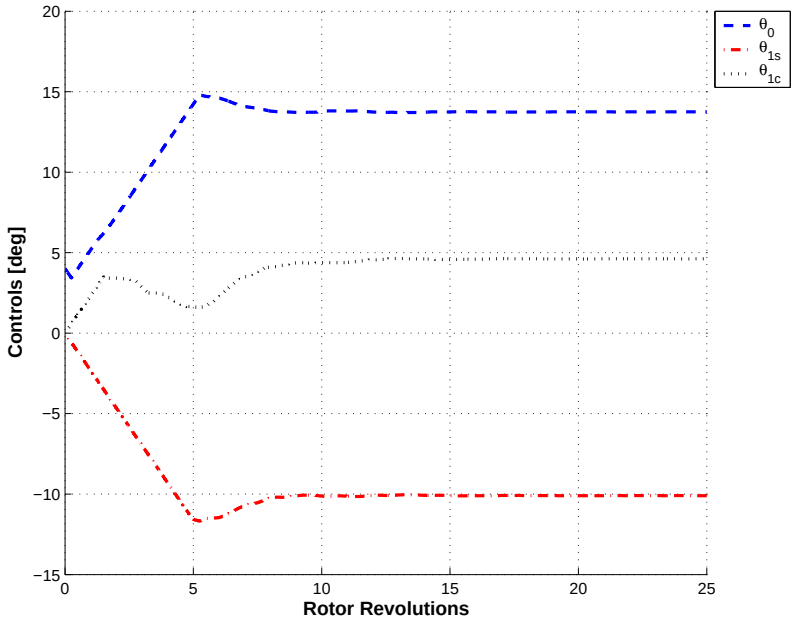

Figure 19: Control time history for the NMP auto-pilot with equation defect identification, $\mu = 0.35$.

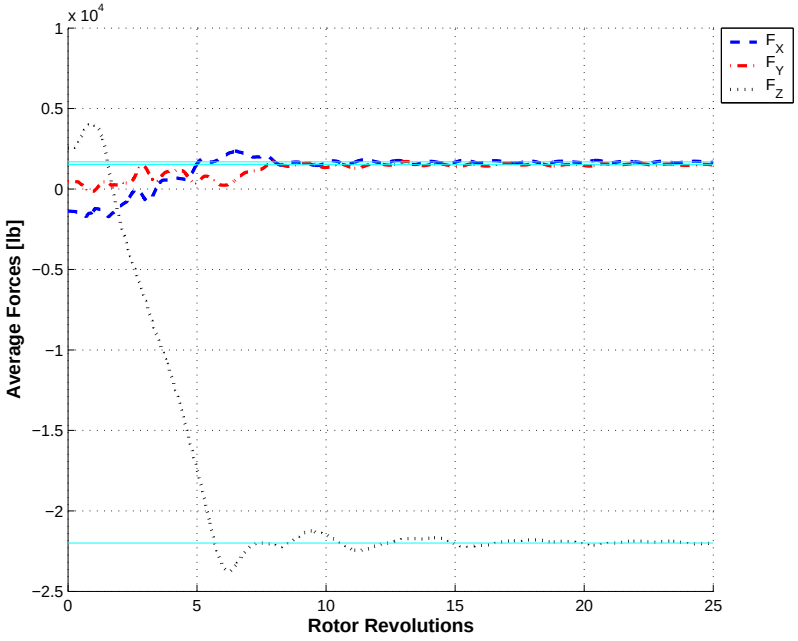

Figure 20: Output time history for the NMP auto-pilot with output error identification, $\mu = 0.35$ (solid lines: target values).

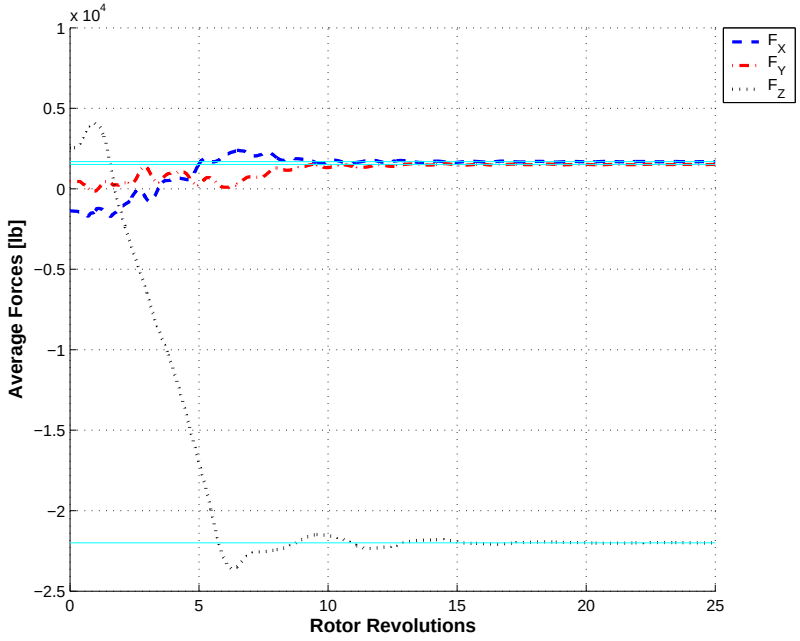

Figure 21: Output time history for the NMP auto-pilot with equation defect identification, $\mu = 0.35$ (solid lines: target values).

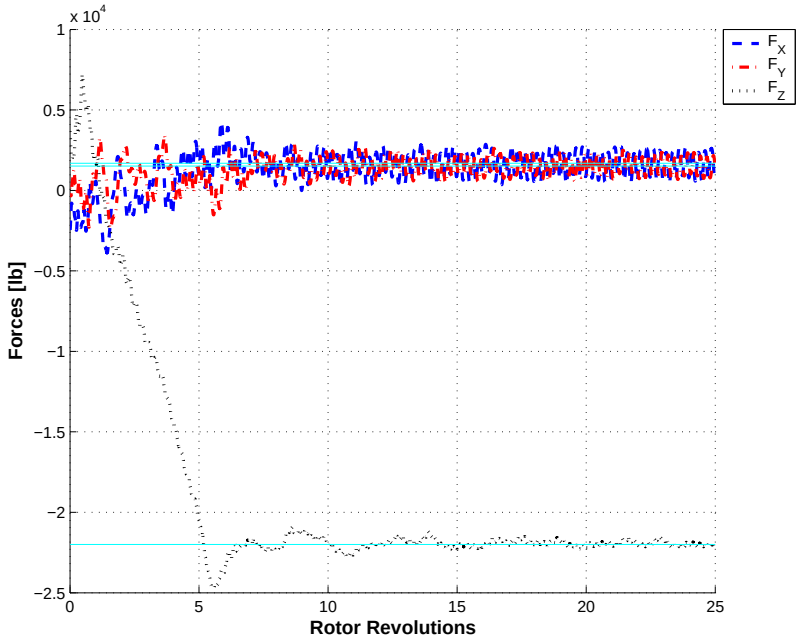

Figure 22: Force component time history for the NMP auto-pilot with output error identification, $\mu = 0.35$ (solid lines: target values).

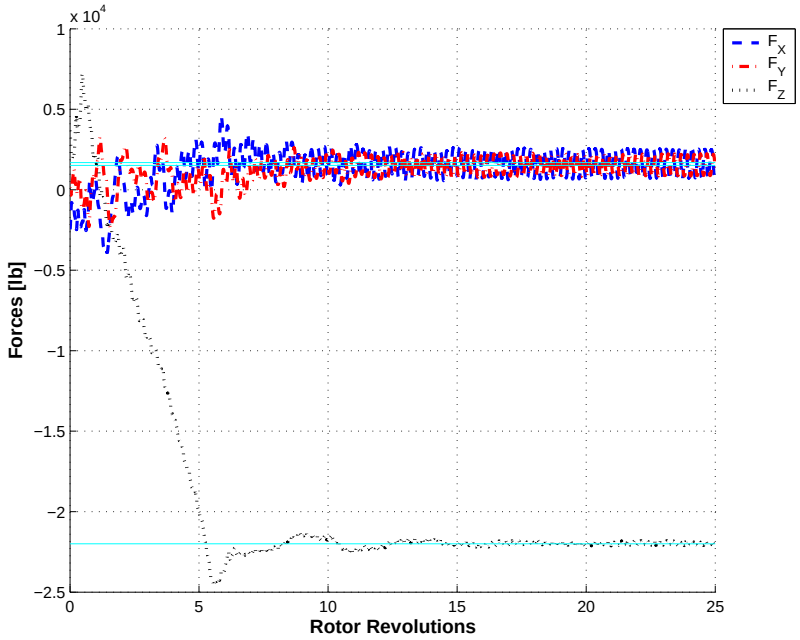

Figure 23: Force component time history for the NMP auto-pilot with equation defect identification, $\mu = 0.35$ (solid lines: target values).

4.2.3 Complete Trim Analysis and Comparison with Classic Auto-pilot

We are now ready to extensively study the behavior of the proposed auto-pilot comparing its performance with that of a more traditional auto-pilot, whose formulation is reported in Appendix C. The multibody model previously described is steered to an estimated trim condition, computed according to Appendix B, for different values of the far field velocity. In each simulation, a fixed value of advance ratio, μ, is chosen starting from 0 for the first simulation and increasing it of 0.05 each time, up to a maximum of 0.35. At the beginning of each test, the rotor collective, longitudinal and lateral cyclic are set to 4.01 deg (0.07 rad), 0.0 and 0.0 deg, respectively.

Several preliminary tests where performed to tune the classic auto-pilot gain matrix $\boldsymbol{G}$ for this particular problem. In the following, we will refer to two different situations:

- *classic auto-pilot A*, tuned for trimming the rotor with maximum performance (shortest time) at $\mu \approx 0.2$, with $\boldsymbol{g}_0 = (0.6, 0.6, 2.0)^T$;

- *classic auto-pilot B*, optimized for $\mu \approx 0.25$ and corresponding in our tests to $\boldsymbol{g}_0 = (0.2, 0.2, 2.0)^T$.

In both cases, the matrix $\boldsymbol{J}$ is initially computed by perturbing the system about the initial condition, according to equation (50). In particular, we perturbed one control at time of a small quantity Δ for 4 revolutions, measuring the changes in the outputs and extracting one column of the matrix at time.

To let all the transients decay, we simulated a number of rotor revolutions equal to 25 for the NMP auto-pilot, 30 for the auto-pilot A and 102 for the auto-pilot B. The NMP auto-pilot used for these tests is the one with equation defect identification, with controller activation frequency equal to 4/rev, prediction horizon of 3 revolutions and control rates limited to 10 deg/sec. Also in this case the neural network has 20 neurons in its hidden layer and a starting learning rate set to 0.3.

4.2.3.1 Trim Time

In order to measure the ability of the proposed auto-pilot to accurately and quickly trim the system, a criterion is required to indicate when the trim solution is achieved, within a desired tolerance. Let us define

$$\varepsilon_{\text{con}}(t) = \|\widetilde{\boldsymbol{y}}_s(t) - \boldsymbol{y}_s^*\|_2 \,,$$

which is a possible measure of the error committed on the trim constraints (4) as a function of time. The outputs are here scaled according to Subsection 4.1.1, using the scaling factors given in Table 1 and 2.

Table 1: Input scaling factors.

	$\boldsymbol{u}_a$[deg]	$\boldsymbol{u}_v$[deg]
θ_0	20	20
θ_{1s}	0	10
θ_{1c}	0	10

In this test, we conventionally define the *trim time* as

$$T_{\text{trim}} : \varepsilon_{\text{con}}(t) \leq \varepsilon_{\text{con}}^{\max}, \forall t \geq T_{\text{trim}}.$$

Table 2: Output scaling factors.

	$\boldsymbol{y}_a$[lb]	$\boldsymbol{y}_v$[lb]
F_X	0	2000
F_Y	0	2000
F_Z	-22000	22000

The quantity $\varepsilon_{\text{con}}^{\max}$ is the maximum allowed error on the trim constraints, which strongly affects the value of the trim time for a given system output time history.

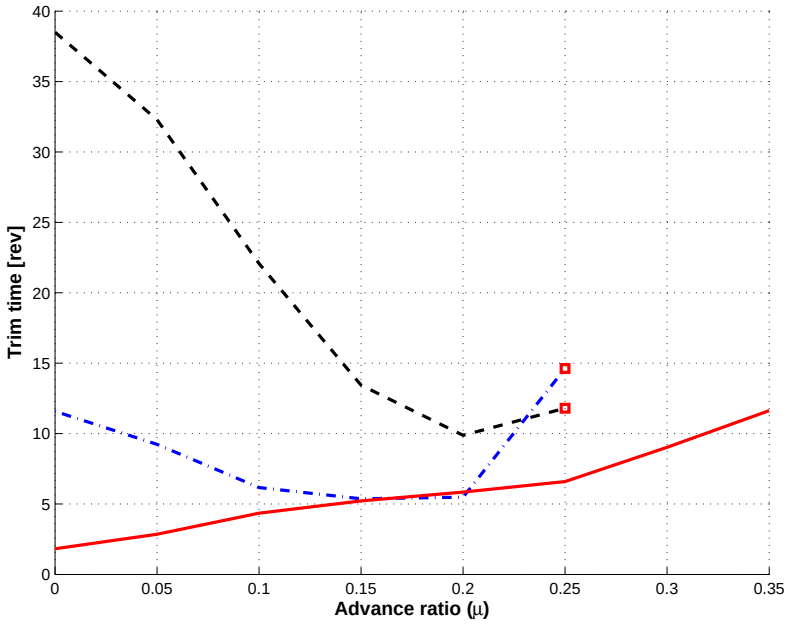

Figure 24: Trim time for $\varepsilon_{\text{con}}^{\max} = 0.05$. Solid line: NMP auto-pilot; dash-dotted line: classic auto-pilot A; dashed line: classic auto-pilot B.

In Figure 24, 25 and 26 we summarize the resulting trim time for $\varepsilon_{\text{con}}^{\max} = 0.05$, $\varepsilon_{\text{con}}^{\max} = 0.02$ and $\varepsilon_{\text{con}}^{\max} = 0.01$, respectively. The dash-dotted and dashed lines represent the classic auto-pilot A and B, respectively, and the solid line indicates the proposed predictive auto-pilot. Clearly, the request of a lower error $\varepsilon_{\text{con}}^{\max}$ increases the trim time for all the controllers, such that

$$T_{\text{trim}}|_{\varepsilon_{\text{con}}^{\max}=0.05} < T_{\text{trim}}|_{\varepsilon_{\text{con}}^{\max}=0.02} < T_{\text{trim}}|_{\varepsilon_{\text{con}}^{\max}=0.01}. \tag{44}$$

In each figure, close to the values of μ for which they have been calibrated, the classic

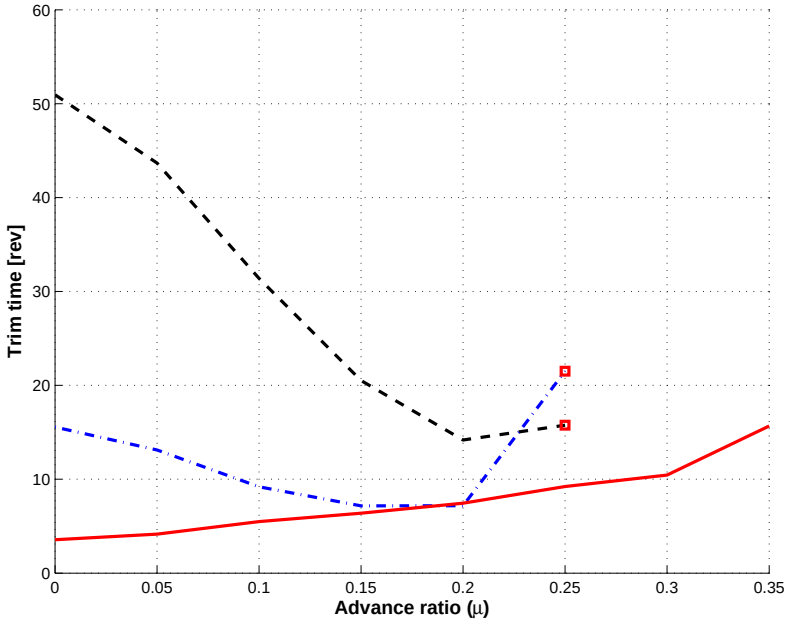

Figure 25: Trim time for $\varepsilon_{\text{con}}^{\max} = 0.02$. Solid line: NMP auto-pilot; dash-dotted line: classic auto-pilot A; dashed line: classic auto-pilot B.

auto-pilots perform nicely and their trim time is definitely similar to the one of the NMP auto-pilot. However, their overall behavior is poor, with almost unacceptable trim times for low advance ratios. Moreover, for $\mu > 0.25$ it was not possible to trim the rotor, so the square symbols in the figures indicate the classic auto-pilot stability limit found for this problem. Both this aspects are related to the trade-off existing between performance and stability: higher gains give a better behavior for small values of μ but they can make the auto-pilot unstable at higher ratios. Therefore, different tests have been made decreasing the gains to find a combination able to stabilize the controller beyond $\mu = 0.25$. Unfortunately, no solution to this problem was found.

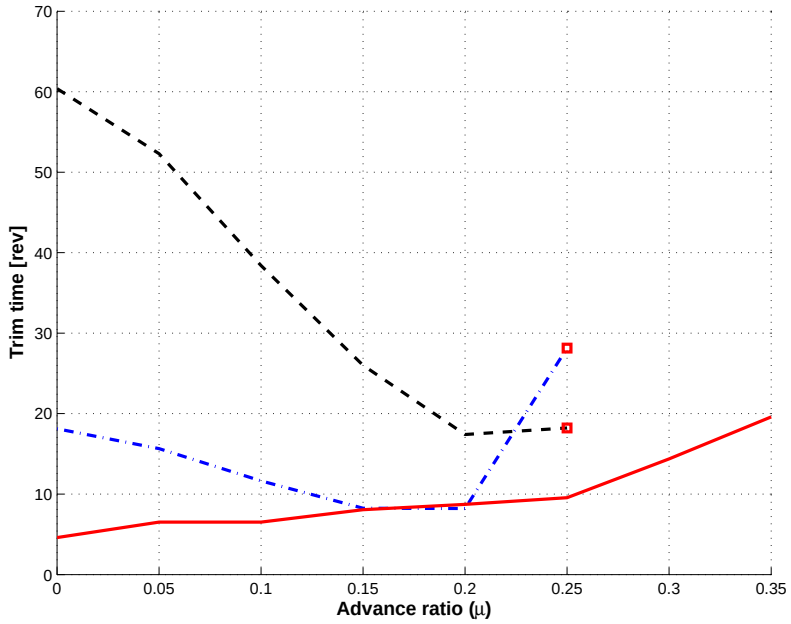

Figure 26: Trim time for $\varepsilon_{\text{con}}^{\max} = 0.01$. Solid line: NMP auto-pilot; dash-dotted line: classic auto-pilot A; dashed line: classic auto-pilot B.

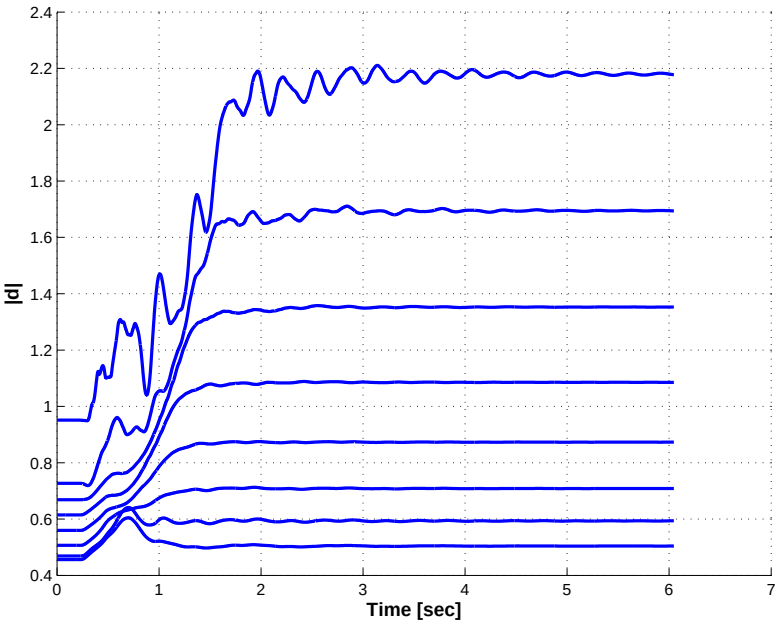

Figure 27: Time history of the (scaled) defect $\|\boldsymbol{d}\|$ for different values of the advance ratio μ (from 0 to 0.35 with steps of 0.05, from bottom to top).

Finally, Figure 27 shows the time histories of the actual equation defect $\boldsymbol{d}$ as measured during the tests with the NMP auto-pilot, for different values of μ (from 0 to 0.35 from bottom to top). After a transient, the defect to be identified converges to a constant vector, which however increases in norm with μ. Thus, for high advance ratios, a longer time is required to identify $\boldsymbol{d}$ and higher trim times must be expected for the proposed controller, as pointed out by Figures 24-26.

4.2.3.2 Accuracy of the Computed Trim Controls

For the considered problem, solutions of the trim problem in terms of the controls are not available. However, they can be estimated by considering the values reached by the auto-pilots at the end of the different simulations. Table 3 summarizes the results for the predictive and the two classic auto-pilots, respectively. As already mentioned, the NMP controller is the only one stable up to $\mu = 0.35$, so we take it as reference. Let us now measure the error of the classic auto-pilot solutions at the end of the simulations with respect to the final solution of the predictive auto-pilot. The Table shows that the maximum of the absolute value of this error is always less than 0.0037 deg, so we can consider the reference solution of the predictive controller pretty accurate.

Table 3: Trim solution (in degrees) at the end of the simulations: NMP auto-pilot (top), classic auto-pilot A (center) and classic auto-pilot B (bottom).

μ	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35
θ_0	8.897	8.931	9.100	9.454	10.031	10.887	12.083	13.748
θ_{1s}	0.210	-0.364	-1.126	-2.076	-3.314	-4.949	-7.146	-10.096
θ_{1c}	3.487	3.181	2.559	2.300	2.394	2.785	3.504	4.620
θ_0	8.898	8.931	9.100	9.454	10.030	10.887	-	-
θ_{1s}	0.209	-0.365	-1.126	-2.075	-3.313	-4.947	-	-
θ_{1c}	3.484	3.181	2.559	2.300	2.395	2.787	-	-
θ_0	8.898	8.931	9.100	9.454	10.030	10.885	-	-
θ_{1s}	0.208	-0.365	-1.126	-2.075	-3.313	-4.943	-	-
θ_{1c}	3.484	3.181	2.559	2.300	2.395	2.789	-	-

Let's now consider the difference between the controls at $t = T_{\text{trim}}$ for different maximum

allowed error on the trim constraints:

- Setting $\varepsilon_{\text{con}}^{\max} = 0.05$, the solution in terms of controls is pretty inaccurate for all the auto-pilots (see Table 4). The error with respect to the reference solution is of magnitude up to 0.6 deg, which can not be considered acceptable, for all the controllers.

- For $\varepsilon_{\text{con}}^{\max} = 0.02$, the quality of the solution is superior, however the error reaches sometimes 0.1 deg (see Table 5). The two classic auto-pilots, especially auto-pilot B, seem to behave slightly better, though the accuracy of the controllers is practically the same.

- In the case $\varepsilon_{\text{con}}^{\max} = 0.01$, the solution has always an error quite lower than 0.1 deg (see Table 6). No substantial difference exists in the accuracy of the controllers.

4.2.3.3 Analysis of the Results

Figures 24-26 underline pros and cons of the use of a classic auto-pilot: if correctly tuned, it can be very efficient, however this tuning process can require a few simulations and is hardly automatable. On the other hand, a model predictive control strategy can incorporate initial information about the system, while guaranteeing stability and performance. This also means that the initial evaluation of the sensitivity matrix $\boldsymbol{J}$, needed in the classic auto-pilot just to extract a linear approximation of the behavior of the system, is avoided, saving further computational time.

The improvement in terms of trim time obtained by using the proposed auto-pilot seems not to influence the quality of the solution in terms of controls. For the problem analyzed here, a tolerance on the trim constraints $\varepsilon_{\text{con}}^{\max} < 0.02$ seems to be necessary to get a solution with accuracy greater than 0.1 deg on the rotor controls.

Table 4: Control error (in degrees) at trim time for $\varepsilon_{\text{con}}^{\max} = 0.05$: NMP auto-pilot (top), classic auto-pilot A (center) and classic auto-pilot B (bottom).

μ	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35
θ_0	6.71e-1	2.26e-2	3.75e-2	2.99e-2	3.04e-2	3.70e-2	5.73e-2	4.03e-2
θ_{1s}	1.74e-1	1.62e-1	1.38e-1	1.19e-1	1.20e-1	1.27e-1	3.48e-2	3.48e-2
θ_{1c}	3.02e-2	1.87e-1	3.03e-1	2.69e-1	2.84e-1	3.14e-1	2.07e-1	1.13e-1
θ_0	1.04e-2	4.07e-2	2.14e-1	3.65e-1	3.77e-1	1.22e-2	-	-
θ_{1s}	8.07e-2	7.52e-2	8.09e-2	1.40e-1	2.12e-1	2.20e-2	-	-
θ_{1c}	1.83e-1	2.03e-1	2.00e-1	1.07e-1	1.02e-1	1.11e-2	-	-
θ_0	1.13e-4	5.71e-3	1.01e-2	5.84e-3	8.81e-2	3.32e-2	-	-
θ_{1s}	9.96e-2	8.82e-2	6.96e-2	1.32e-2	2.36e-1	9.54e-2	-	-
θ_{1c}	2.32e-1	2.31e-1	2.34e-1	2.49e-1	2.48e-1	4.45e-2	-	-

Table 5: Control error (in degrees) at trim time for $\varepsilon_{\text{con}}^{\max} = 0.02$: NMP auto-pilot (top), classic auto-pilot A (center) and classic auto-pilot B (bottom).

μ	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35
θ_0	1.62e-2	1.91e-2	1.64e-3	1.09e-3	2.72e-2	2.57e-2	2.31e-2	1.10e-2
θ_{1s}	1.13e-3	4.33e-2	3.54e-2	2.34e-2	1.98e-2	1.07e-2	1.34e-2	7.21e-3
θ_{1c}	8.65e-2	9.23e-2	1.15e-1	1.02e-1	1.15e-1	8.41e-2	8.13e-2	2.50e-2
θ_0	4.63e-4	4.86e-3	4.65e-2	1.40e-1	1.43e-1	2.52e-3	-	-
θ_{1s}	2.87e-2	1.87e-2	3.60e-3	5.05e-2	6.76e-2	6.49e-3	-	-
θ_{1c}	7.03e-2	6.73e-2	5.64e-2	2.87e-2	3.47e-2	1.30e-2	-	-
θ_0	8.43e-4	2.35e-3	4.10e-3	1.97e-3	1.64e-2	4.85e-3	-	-
θ_{1s}	4.02e-2	3.54e-2	2.69e-2	6.59e-3	5.51e-2	1.85e-2	-	-
θ_{1c}	9.35e-2	9.17e-2	9.37e-2	9.74e-2	9.43e-2	2.36e-5	-	-

Table 6: Control error (in degrees) at trim time for $\varepsilon_{\text{con}}^{\max} = 0.01$: NMP auto-pilot (top), classic auto-pilot A (center) and classic auto-pilot B (bottom).

μ	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35
θ_0	3.17e-3	8.57e-3	4.69e-3	1.62e-2	2.07e-2	1.83e-2	1.33e-2	3.37e-3
θ_{1s}	1.35e-2	4.49e-3	5.12e-3	7.24e-4	6.72e-3	7.12e-3	5.83e-3	3.88e-3
θ_{1c}	3.84e-2	1.39e-2	4.92e-2	4.34e-2	5.67e-2	6.82e-2	2.39e-2	5.30e-3
θ_0	2.11e-4	2.38e-4	1.08e-2	8.10e-2	8.53e-2	6.97e-3	-	-
θ_{1s}	1.78e-2	1.19e-2	5.17e-3	4.36e-2	5.96e-2	1.80e-2	-	-
θ_{1c}	3.90e-2	3.23e-2	1.80e-2	1.67e-2	2.80e-2	5.88e-3	-	-
θ_0	1.10e-3	1.25e-3	2.07e-3	9.55e-4	5.60e-3	1.06e-3	-	-
θ_{1s}	2.03e-2	1.78e-2	1.33e-2	4.65e-3	2.07e-2	2.19e-3	-	-
θ_{1c}	4.73e-2	4.53e-2	4.71e-2	4.74e-2	5.05e-2	3.50e-3	-	-

4.2.4 Convergence Analysis for Classic and Proposed Auto-pilots

As a final test, we conceptually repeated the procedure followed in Section 4.1, investigating this time the behavior of both the proposed formulations and the classic auto-pilot A. We chose the control values 10.0, -6.0 and 5.0 deg for the collective, longitudinal and lateral cyclic pitch angles, respectively. Then, we simulated forward in time the multibody model, we left all the transients decay and we eventually measured the trim outputs $\boldsymbol{y}^*$.

For each of the considered auto-pilots, we started again the simulation setting θ_0, θ_{1s} and θ_{1c} to 0.07, 0.0 and 0.0 rad, with the rotor following the corresponding trimmed orbit, and we activated the controller. Here again, the different quantities are scaled using the values of Tables 1 and 2.

Since the exact vector of trim control values, indicated with $\boldsymbol{u}^*$, is known in this case, it is possible to measure the error

$$\varepsilon_{\text{ctr}}(t) = \|\boldsymbol{u}(t) - \boldsymbol{u}^*\|_2 .$$

In Figures 28-30 all the three auto-pilots exhibit a similar trend, giving an error less than 6.0e-3 deg after 20 revolutions. In Figures 31-33 slightly better results for the proposed auto-pilots, in terms of convergence of the constraint violation, can be observed. The corresponding time histories for the objective function J^{reg} of the predictive regulation

problem (8) are finally shown in Figures 34 and 35. Control and output time histories respect the general behavior of Figures 12-17, therefore they are omitted.

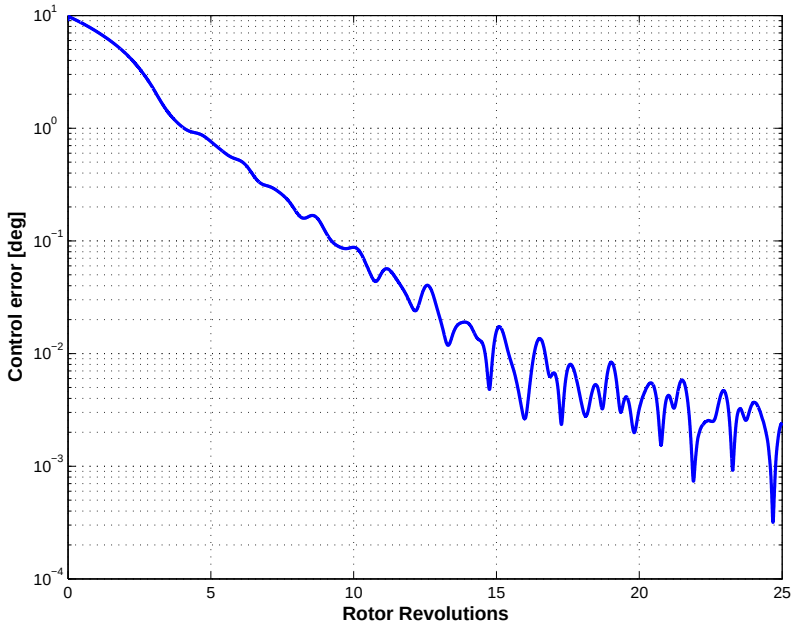

Figure 28: Time history of the control error ε_{ctr} for the classic auto-pilot A, $\mu = 0.2$.

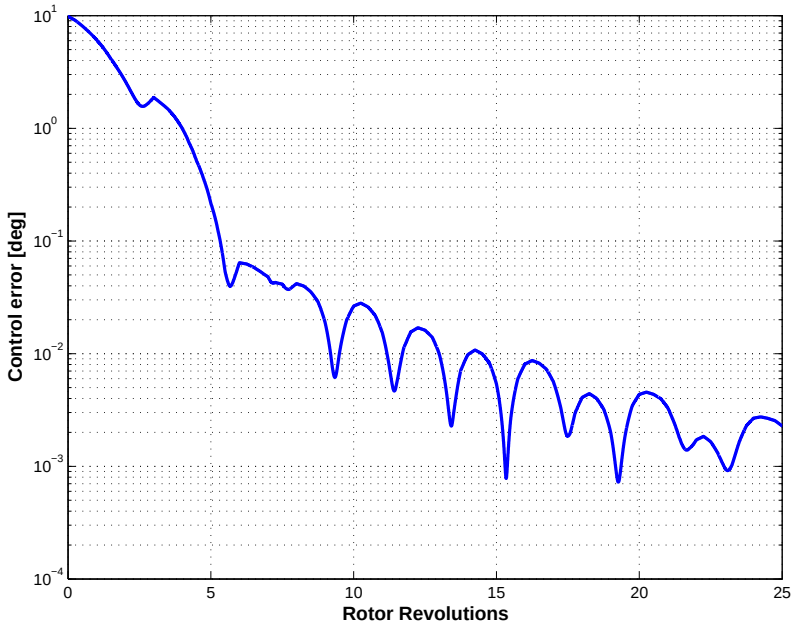

Figure 29: Time history of the control error ε_{ctr} for the proposed auto-pilot with output error identification, $\mu = 0.2$.

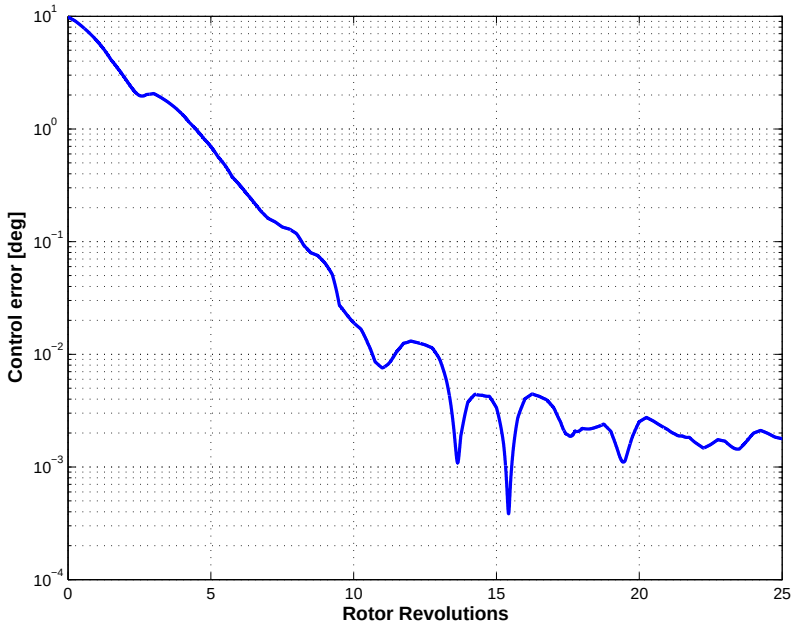

Figure 30: Time history of the control error ε_{ctr} for the proposed auto-pilot with equation defect identification, $\mu = 0.2$.

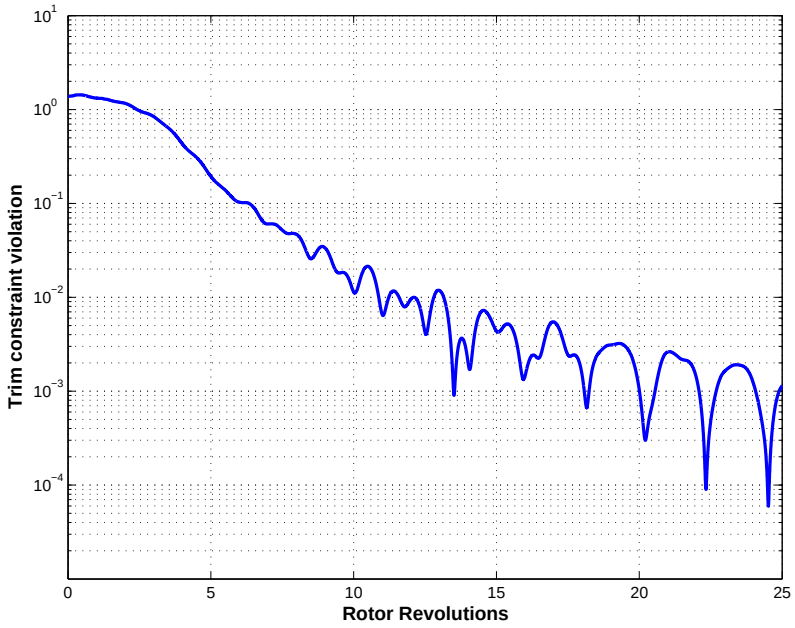

Figure 31: Time history of the (scaled) violation of the trim constraints for the classic auto-pilot A, $\mu = 0.2$.

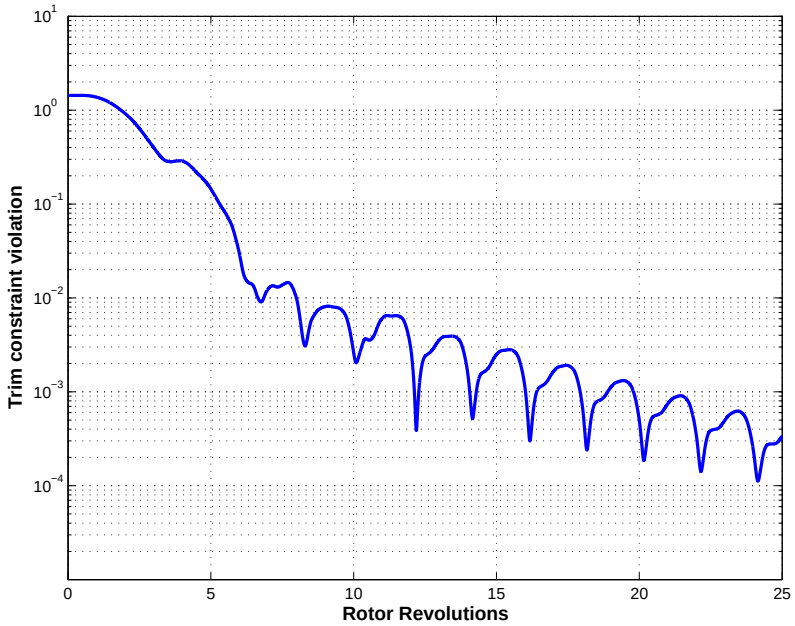

Figure 32: Time history of the (scaled) violation of the trim constraints for the proposed auto-pilot with output error identification, $\mu = 0.2$.

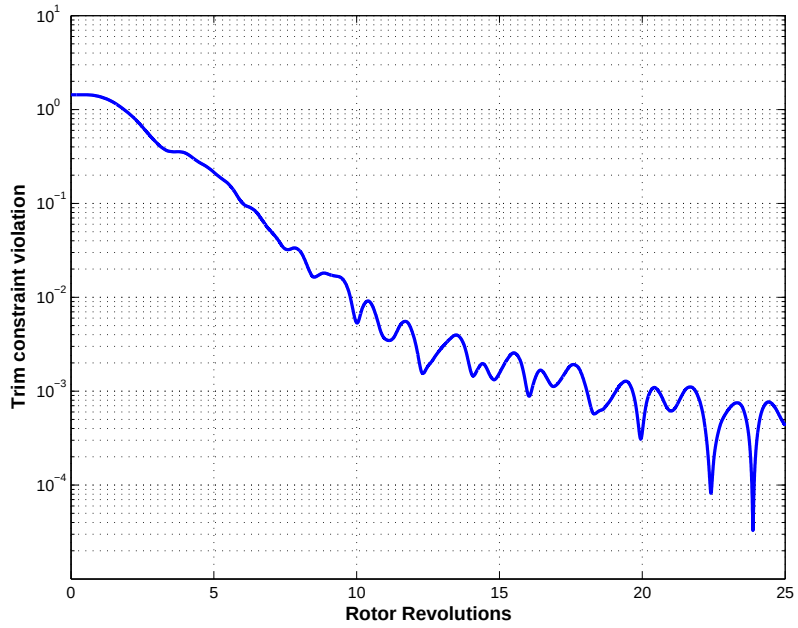

Figure 33: Time history of the (scaled) violation of the trim constraints for the proposed auto-pilot with equation defect identification, $\mu = 0.2$.

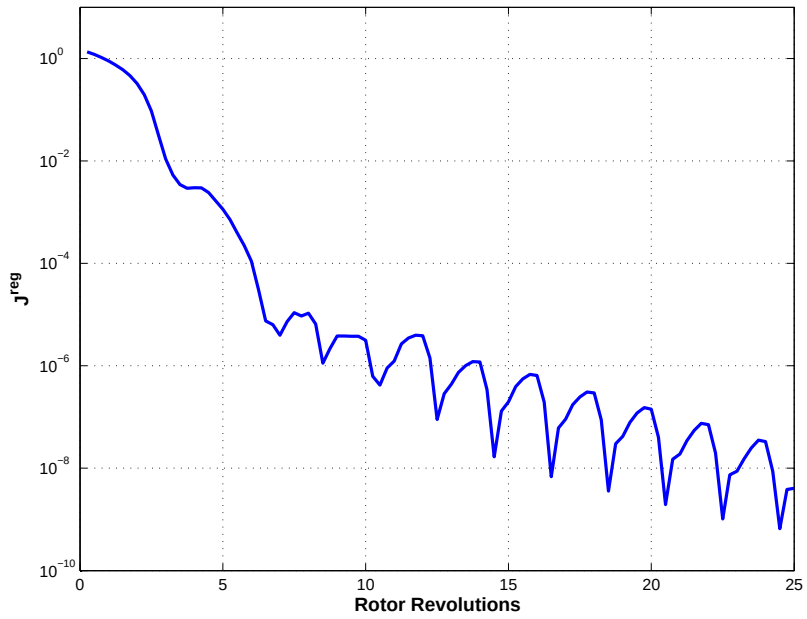

Figure 34: Time history of the (scaled) cost function J^{reg} for the proposed auto-pilot with output error identification, $\mu = 0.2$.

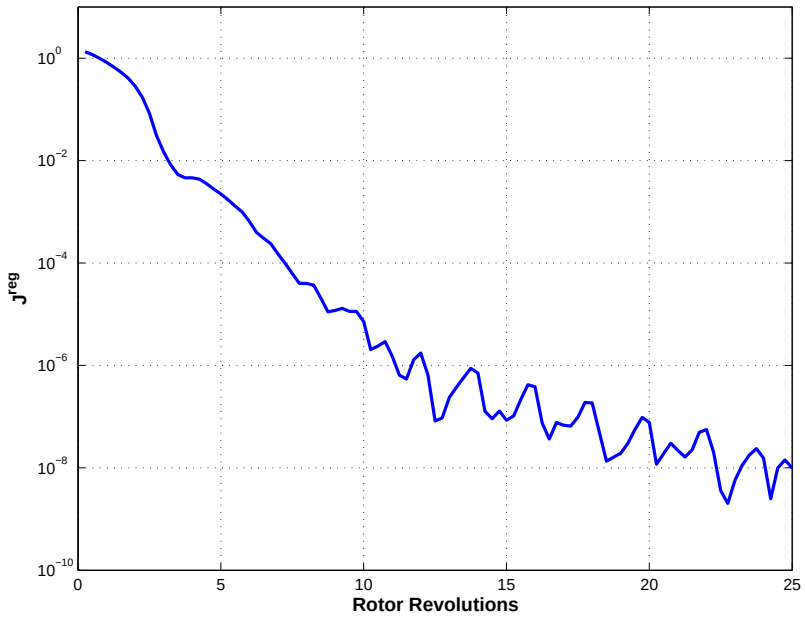

Figure 35: Time history of the (scaled) cost function J^{reg} for the proposed auto-pilot with equation defect identification, $\mu = 0.2$.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

Increasingly advanced tools are available for the modeling and simulation of complex dynamical systems. Fixed and rotary wing aircrafts can be represented by means of flexible, non-linear virtual models subjected to airloads coming from the coupling with advanced aerodynamic codes. The evaluation of the equilibria of these systems is still the cornerstone for performance and stability analysis. Unfortunately, although model trim could appear a conceptually simple problem, the modelling generality of modern codes is making it more and more burdensome.

In particular, rotorcraft models are typically unstable and characterized by harmonic forcing loads. Therefore, the trim problem translates in finding some particular constant controls so that the system response follows a desired periodic orbit. Several numerical methods have been proposed to enforce the compatibility of the solution with the system dynamics, the constraint of constant controls and the periodicity. However, the efficiency of most of these methodology degrades with the complexity of the system since the trim problem dimension is function of the total number of degrees of freedom.

The auto-pilot approach follows a different path: a control strategy is designed to produce control signals to virtually "fly" the system to the trim solution. In its classic formulation, however, no constant-in-time condition for the controls is considered. Moreover in practice the classic auto-pilot must be carefully tuned to guarantee stability and a reasonable efficiency.

In this work, we proposed a new auto-pilot based on the non-linear model-predictive control technique. NMPC framework allows to design control strategies which can guarantee stability, and the computed solution is typically very efficient since it is based on the solution of an optimal control problem (prediction) for a reduced model. Stability and efficiency are strictly related to this model. Proper reference models are easily available today and they

can be augmented by means of adaptive elements to increase their accuracy. Here, the chosen reference model has been based on analytical relations derived from blade element theory, and the identification task has been pursued by means of neural networks.

The obtained controller was able to preliminarily trim a simplified, rigid four-bladed rotor. Then, it was applied to a multibody finite element model of the UH-60 rotor, using and comparing two different formulations for the identification.

Finally, the trim of the same multibody model for a wide range of far field velocity conditions has been obtained using the proposed and a classic auto-pilot. This final test showed that, if well tuned, the classic auto-pilot performs almost as well as the more refined NMP one. On the other hand, even the use of a pretty simple reference model gives the predictive control superior stability properties, while the adaptive element allows convergence. Even more important, the information incorporated in the reduced model allows one to circumvent the necessity of tuning the controller, which is sometimes a very hard and time demanding task with classic auto-pilots. The main cost of the proposed methodology is a higher computational effort, which is acceptable also because, for very complex systems, it represents a little percentage of the cost of the overall simulation.

The results gathered in this thesis show that model-predictive control is a promising approach for trimming complex models of rotors, and possibly rotorcrafts. Future efforts should be in the direction of extending these results to more complex and demanding systems, like multibody models coupled with refined free wake or CFD procedures, or free flying complete rotorcrafts. In particular, the impact of the mismatch between reference model and controlled system on the auto-pilot stability and the trim solution accuracy must be carefully investigated. In this sense, different methodologies and formulations for the identification process could be analyzed, trying to extract as much synthetic information as possible from the data available from the controlled system.

APPENDIX A

ROTOR BLADE ELEMENT MODEL

In this appendix we give a brief overview of the analytic blade element rotor model [10, 14, 22] we adopt as reference model for this research.

Assuming that the shaft is aligned with the gravity acceleration and the Z axis of an inertial frame, the rotor force components can be expressed in this frame as

$$F_X = -T \sin a_{1s} - H \cos a_{1s},$$

$$F_Y = -T \sin b_{1s},$$

$$F_Z = -T \cos a_{1s} + H \sin a_{1s},$$

where T and H are rotor thrust and H-force, the lateral flapping angle with respect to the shaft is noted b_{1s} and the longitudinal flapping angle a_{1s}, which corresponds to the tip-path-plane angle of attack α_{tpp} in the present case.

The rotor thrust coefficient c_t and H-force coefficient c_h can be related to the piloting controls. To this end, the blade pitch, neglecting the effects of higher harmonics, can be written as

$$\theta = \theta_0 + \theta_{1s} \sin\psi + \theta_{1c} \cos\psi.$$

The rotor advance and inflow ratios can be written in the tip-path-plane reference as

$$\mu_{\text{tpp}} = \frac{V \cos\alpha_{\text{tpp}}}{\omega R}, \qquad \lambda_{\text{tpp}} = \frac{V \sin\alpha_{\text{tpp}}}{\omega R} - \frac{u}{\omega R},$$

and transformed to the no-feathering reference through the relations

$$\mu_{\text{nf}} = \quad \mu_{\text{tpp}} \cos a_{1\,\text{nf}} + \lambda_{\text{tpp}} \sin a_{1\,\text{nf}}, \qquad \lambda_{\text{nf}} = -\mu_{\text{tpp}} \sin a_{1\,\text{nf}} + \lambda_{\text{tpp}} \cos a_{1\,\text{nf}},$$

u being the uniform induced velocity, V the wind speed, ω and R rotor rotational speed and radius, while $a_{1\,\text{nf}}$, $b_{1\,\text{nf}}$ are the longitudinal and lateral, respectively, flapping angles with respect to the no-feathering plane, which are related to the same quantities measured

in the shaft reference as

$$a_{1s} = a_{1\,\mathrm{nf}} + \theta_{1s}, \qquad b_{1s} = b_{1\,\mathrm{nf}} - \theta_{1c}.$$

The H-force coefficient in the same plane can at this point be expressed as

$$\frac{c_{h_{\mathrm{nf}}}}{\sigma} = \frac{c_d\,\mu_{\mathrm{nf}}}{4} + \frac{c_{l_\alpha}}{2}\left(\frac{1}{3}\,a_{1\,\mathrm{nf}}\,\theta_0 + \frac{3}{4}\,\lambda_{\mathrm{nf}} a_{1\mathrm{nf}} - \frac{1}{2}\,\mu_{\mathrm{nf}}\,\theta_0\,\lambda_{\mathrm{nf}} + \frac{1}{4}\,\mu_{\mathrm{nf}}\,a_{1\,\mathrm{nf}}^2\right).$$

Similarly, the thrust coefficient is

$$\frac{c_{t_{\mathrm{nf}}}}{\sigma} = \frac{c_{l_\alpha}}{4}\left[\left(\frac{2}{3} + \mu_{\mathrm{nf}}^2\right)\theta_0 + \lambda_{\mathrm{nf}}\right].$$

In the previous relations, the blade airfoil lift curve slope is noted c_{l_α}, while σ is the rotor solidity and c_d the rotor blade section drag coefficient. The force coefficients can be transformed to the tip-path-plane reference as

$$\begin{aligned} c_{t_{\mathrm{tpp}}} &= \quad c_{t_{\mathrm{nf}}}\cos a_{1\,\mathrm{nf}} + c_{h_{\mathrm{nf}}}\sin a_{1\,\mathrm{nf}}, \\ c_{h_{\mathrm{tpp}}} &= -c_{t_{\mathrm{nf}}}\sin a_{1\,\mathrm{nf}} - c_{h_{\mathrm{nf}}}\cos a_{1\,\mathrm{nf}}. \end{aligned}$$

Finally, using momentum theory the induced velocity u is expressed implicitly in the following way:

$$c_{t_{\mathrm{tpp}}} = \frac{2\,u}{\omega R}\sqrt{\mu_{\mathrm{tpp}}^2 + \lambda_{\mathrm{tpp}}^2}.$$

Neglecting higher harmonics, the blade flapping angle is

$$\beta(\psi) = a_0 - a_{1s}\cos\psi - b_{1s}\sin\psi,$$

where ψ is the azimuthal position of the blade. Imposing the equilibrium of the blade around the flap hinge gives the coning angle, namely

$$a_0 = \frac{2\,\rho c R^2}{m_b}\,\frac{c_t}{\sigma} - \frac{3\,g}{\omega^2 R}\,\frac{1}{2 + e_\beta},$$

where c and m_b are respectively the blade chord and mass, ρ the air density, g the acceleration of gravity and e_β the flap hinge offset. The steady-state solution for blade flapping [10, 14] gives

$$\begin{aligned} a_{1\,\mathrm{nf}} &= \frac{\mu_{\mathrm{nf}}}{1 - \mu_{\mathrm{nf}}^2/2}\left(\frac{8}{3}\,\theta_0 + 2\lambda_{\mathrm{nf}}\right) + \frac{8}{\gamma}\,\frac{\epsilon}{1 - \mu_{\mathrm{nf}}^2/2}\,b_{1\,\mathrm{nf}}, \\ b_{1\,\mathrm{nf}} &= \frac{4/3\mu_{\mathrm{nf}}\,a_0}{1 + \mu_{\mathrm{nf}}^2/2} - \frac{8}{\gamma}\,\frac{\epsilon}{1 + \mu_{\mathrm{nf}}^2/2}\,a_{1\,\mathrm{nf}}, \end{aligned}$$

where

$$\epsilon = x_{CG_{bl}} \frac{m_b e_\beta R^2}{J_\beta},$$

and

$$\gamma = \frac{\rho c_{l_\alpha} c R^4}{J_\beta}$$

is the Lock number. The non-dimensional position of the blade center of gravity is noted $x_{CG_{bl}}$, and J_β is the flapping moment of inertia of the blade.

APPENDIX B

ROTORCRAFT APPROXIMATE PERFORMANCE

Following the indications of [16], we can estimate the average forces required to trim a UH-60 as a function of the flight speed, in order to produce values $\widetilde{\boldsymbol{y}}$, functions of the advance ratio, to which the rotor of Chapter 4 can be trimmed.

For simplicity, let's define

$$c_w = \frac{W}{\rho\, A\, (\Omega R)^2},$$

where W is an estimate of the helicopter weight, ρ the air density, A the reference area (area of the rotor disk), Ω the rotor rotational speed and R the rotor radius. From the momentum theory applied to the rotor disk, we can obtain the thrust coefficient and assume it be equal to c_w:

$$c_t = -2\,\lambda_i \sqrt{\mu^2 + \lambda^2} = c_w,$$

where μ is the advance ratio, λ the inflow ratio and $\lambda_i = \frac{u}{\Omega R}$ the induced inflow. In this way, we can easily estimate the required thrust for trim,

$$T = \rho\, A\, (\Omega\, R)^2\, c_t,$$

and the related inflow velocity u.

In first approximation, the required power for trim can be evaluated from

$$c_p = \frac{K\, c_w^2}{2\sqrt{\mu^2 + \lambda^2}} + \frac{\sigma\, c_{d_0}}{8}\left(1 + k_B\, \mu^2\right) + \frac{f}{A}\,\mu^3,$$

c_p being the power coefficient, K a power correction factor generally chosen equal to 1.15, c_{d_0} the rotor sectional drag coefficient, k_B the so called Bennet correction factor [14, p. 184], f the fuselage equivalent flat plate area, considered equal to 22 ft^2 (see for instance Reference [25]). We are now ready to calculate the require power,

$$P = \rho\, A\, (\Omega\, R)^3\, c_p,$$

and derive the rotor required torque as follows

$$Q = \frac{P}{\Omega}.$$

By this set of relations, we can estimate the average of the three force components required to trim the helicopter:

$$F_X = \tfrac{1}{2}\,\rho\,(\Omega\,R)^2\,f, \tag{45}$$

$$F_Y = Q/x_{TR}, \tag{46}$$

$$F_Z = -W. \tag{47}$$

APPENDIX C

CLASSIC AUTO-PILOT

In the auto-pilot approach, a feedback control is designed to steer the system. In this method, the controls are characterized by a dynamic behavior, obtained by measuring the error on the trim constraints (4) and consequently changing the system inputs. For the classic auto-pilot in particular, properly tuned gain factors allow this error to decrease and the system outputs to converge to the trimmed solution. The discrete dynamic equations for the controls can be written as follows:

$$\widetilde{\boldsymbol{u}}_f = \widetilde{\boldsymbol{u}}_i + \Delta t\, \boldsymbol{J}^{-1}\boldsymbol{G}\left(\boldsymbol{y}^* - \widetilde{\boldsymbol{y}}\right), \tag{48}$$

where i and f stand for *initial* and *final*, respectively, and $\boldsymbol{y}^* - \widetilde{\boldsymbol{y}}$ is the error on the trim constraints. The matrix $\boldsymbol{G} = \mathrm{diag}(\boldsymbol{g}_0)$ represents the gain factors, which are carefully tuned in order to obtain a stable solution with the desired performance.

Finally, $\boldsymbol{J}$ indicates the sensitivity matrix of the outputs with respect to the inputs, namely

$$\boldsymbol{J} = \frac{\partial \widetilde{\boldsymbol{y}}}{\partial \boldsymbol{u}}. \tag{49}$$

In practice, this matrix is evaluated by perturbation, for example about the reference output value $\widetilde{\boldsymbol{y}}$:

$$\boldsymbol{J} = \left[\frac{\boldsymbol{y}_1 - \widetilde{\boldsymbol{y}}}{\Delta_1}, \frac{\boldsymbol{y}_2 - \widetilde{\boldsymbol{y}}}{\Delta_2}, \ldots, \frac{\boldsymbol{y}_{n_u} - \widetilde{\boldsymbol{y}}}{\Delta_{n_u}}\right], \tag{50}$$

where the values $\widetilde{\boldsymbol{y}}_i$ are obtained by perturbing one input at time by a quantity Δ_{n_i}. This aspect is quite critical since proper values for the perturbation factors Δ_{n_i} must be chosen. Moreover, this perturbations on the controls must be applied one at time, letting the transients decay and measuring the perturbed outputs. This strategy is clearly unfeasible when dealing with unstable systems.

APPENDIX D

EXTENSION OF THE PROPOSED AUTO-PILOT TO FREE-FLYING ROTORCRAFT TRIM

D.1 Formulation for Free-flying Rotorcraft Model

The aeroelastic model of an entire rotorcraft can be formalized by means of equation (1), here recalled:

$$\begin{aligned} \widetilde{\boldsymbol{f}}(\dot{\widetilde{\boldsymbol{x}}}, \widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{u}}) &= 0, \\ \widetilde{\boldsymbol{c}}(\dot{\widetilde{\boldsymbol{x}}}, \widetilde{\boldsymbol{x}}) &= 0. \end{aligned}$$

As already pointed out in Section 1.1, free-flight trim deals with the average gross motion of the vehicle, representing its flight mechanics response. A possible approach is to consider as flight mechanics states of the rotorcraft the position, orientation, and linear and angular rates of a frame attached to the fuselage, with respect to an inertial frame. In this case, the outputs can be defined as

$$\widetilde{\boldsymbol{y}} = \frac{1}{T} \int_{t-T}^{t} \widetilde{\boldsymbol{g}}(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{u}}) \, \mathrm{d}t,$$

$$\widetilde{\boldsymbol{g}}(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{u}}) = \left(\widetilde{U}, \widetilde{V}, \widetilde{W}, \widetilde{P}, \widetilde{Q}, \widetilde{R}, \widetilde{k} \right)^T,$$

where $\widetilde{U}, \widetilde{V}, \widetilde{W}$ and $\widetilde{P}, \widetilde{Q}, \widetilde{R}$ are the linear and angular velocity components of the body-attached frame expressed in the inertial reference. The function $\widetilde{k} = \widetilde{k}(\widetilde{\boldsymbol{x}})$ is arbitrarily specified to close the trim problem, since the unknown of the problems, in the case of a conventional helicopter, are the average attitude angles and the trim controls

$$\boldsymbol{u} = (\delta_c, \delta_e, \delta_a, \delta_p)^T, \tag{51}$$

$\delta_c, \delta_e, \delta_a$ and δ_p being the generic pilot collective, longitudinal, lateral and pedal controls. For example, if we desire to trim to straight level flight with zero average roll angle and

flight speed $\bar{U}$, we can define $\widetilde{k}(t) = \widetilde{\Phi}(t)$ and specify $\widetilde{\boldsymbol{y}} = \boldsymbol{y}^* = (\bar{U}, 0, 0, 0, 0, 0, 0)^T$ as trim constraints.

As reference model, it is possible to use a rigid body model of the rotorcraft with main and tail rotors described by means of the blade element theory with uniform inflow, see for example Reference [8]. In this case, the reference model reads

$$\dot{\boldsymbol{x}} - \boldsymbol{\varphi}_{\text{ref}}(\boldsymbol{x}, \boldsymbol{u}) = 0, \tag{52a}$$

$$\boldsymbol{y} - \frac{1}{T}\int_t^{t+T} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})\,\mathrm{d}t = 0. \tag{52b}$$

with

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = (U, V, W, P, Q, R, k)^T,$$

where $k(t)$ is the counterpart of $\widetilde{k}(t)$ for the reference model. In this case, internal states for the reference model, indicated as $\boldsymbol{x}$, are present. The whole system of equations can be summarized as

$$\boldsymbol{f}_{\text{ref}}(\boldsymbol{y}, \dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}) = 0,$$

which mimics equation (21).

A possible identification strategy to augment this reference model is to correct the outputs as follows:

$$\boldsymbol{y} - \frac{1}{T}\int_{t-T}^{t} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})\,\mathrm{d}t - \boldsymbol{d}_p(\boldsymbol{y}^{(n)}, \ldots, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}^*) = 0, \tag{53}$$

where as in Section 3.2 the approximating operator $\boldsymbol{d}_p$ is parameterized in $\boldsymbol{p}$.

D.2 Formulation for Rotorcraft Model Constrained on the Longitudinal Plane

In some cases, it can be interesting to study the behavior of a rotorcraft performing maneuvers only in a longitudinal plane. Here, we use the term *longitudinal* to identify, chosen a vector perpendicular to the direction of the gravity acceleration, a plane orthogonal to this vector. In practice, we are considering a system like the one shown in Figure 36. A planar joint located at the helicopter center of mass prevents translations perpendicular to

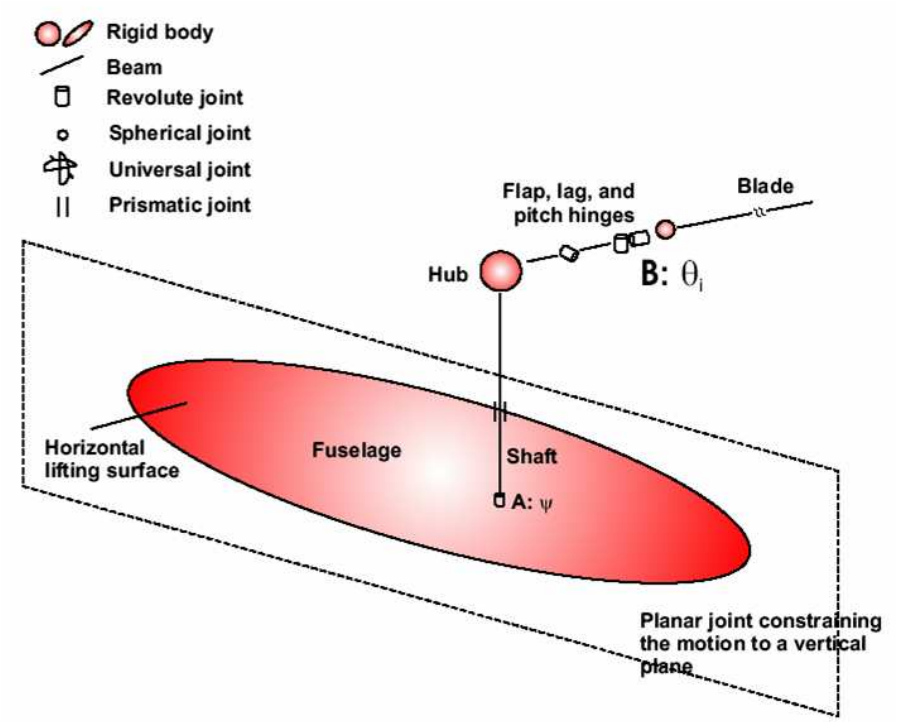

Figure 36: Schematic representation of the rotorcraft multibody model used in Reference [8].

the longitudinal plane, and the only rotational degree of freedom left to the (rigid) fuselage is the pitch attitude with respect to the inertial reference frame.

In this case the trim problem can be recast as follows. The controls are still those of equation 51. Moreover, the pitch attitude is free, so it is possible to choose

$$\widetilde{\boldsymbol{g}}(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{u}}) = \left(\widetilde{U}, \widetilde{W}, \widetilde{Q}, \widetilde{F}_Y, \widetilde{M}_Z\right)^T,$$

where $\widetilde{F}_Y$ and $\widetilde{M}_Z$ indicate this time the reaction force and moment components of the planar constraint of the multibody model, expressed in the inertial frame X, Y, Z. A *quasi-trim constraint* can be obtained by specifying $y_4^* = 0$ and $y_5^* = 0$. In fact, a free-flying trimmed rotorcraft "pretending" to remain in the longitudinal plane will show periodic displacements and rotations out of it. However, using the proposed formulation, the model of Figure 36 will fly in a condition very close to free-fly trim.

Also in this case, the reference model must follow the structure of the multibody model.

Using a rigid body, blade element model as the one of system (52), we will have

$$\dot{\boldsymbol{x}} - \boldsymbol{\varphi}_{\mathrm{ref}}(\boldsymbol{x}, \boldsymbol{u}) = 0,$$
$$\boldsymbol{y} - \tfrac{1}{T}\int_t^{t+T} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})\,\mathrm{d}t = 0,$$

where the states $\boldsymbol{x}$ describe the model degrees of freedom in the longitudinal plane, and

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = (U, W, Q, F_Y, M_Z)^T.$$

Here again the formulation given by equation (53) can be applied to identify the mismatch between reference and multibody model.

REFERENCES

[1] Barclay, A., Gill, P. E., and Rosen, J. B., "SQP methods and their application to numerical optimal control," Tech. Rep. NA 97-3, Department of Mathematics, Univerisity of California, San Diego, CA, 1997.

[2] Bauchau, O. A., Bottasso, C. L., and Nikishkov, Y. G., "Modeling rotorcraft dynamics with finite element multibody procedures," *Mathematics and Computer Modeling*, vol. 33, pp. 1113–1137, 2001.

[3] Bauchau, O. A., Bottasso, C. L., and Trainelli, L., "Robust integration schemes for flexible multibody systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 395–420, 2003.

[4] Betts, J. T., *Practical Methods for Optimal Control Using Non-Linear Programming*. Philadelphia, PA: SIAM, 2001.

[5] Bhagwat, M. J. and Leishman, J. G., "Stability, consistency and convergence of time marching free-vortex rotor wake algorithms," *Journal of the American Helicopter Society*, vol. 46, pp. 59–71, 2001.

[6] Bhattacharya, R., Balas, G. J., Kaya, M. A., and Packard, A., "Nonlinear receding horizon control of an F-16 aircraft," *Journal of Guidance, Control and Dynamics*, vol. 25, pp. 924–931, 2002.

[7] Bottasso, C. L., "A new look at finite elements in time: a variational interpretation of Runge-Kutta methods," *Applied Numerical Mathematics*, vol. 25, pp. 355–368, 1997.

[8] Bottasso, C. L., Croce, A., Leonello, D., and Riviello, L., "Unsteady trim for the simulation of maneuvering rotorcraft with comprehensive models," *Journal of the American Helicopter Society*, 2005. Under review.

[9] Bottasso, C. L., Micheletti, S., and Sacco, R., "The discontinuous Petrov-Galerkin method for elliptic problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 3391–3409, 2002.

[10] Bramwell, A. R. S., *Helicopter dynamics*. New York, NY: John Wiley & Sons, 1976.

[11] Fausett, L., *Fundamentals of Neural Networks*. New York, NY: Prentice Hall, 1994.

[12] Findeisen, R., Imland, L., Allgöwer, F., and Foss, B. A., "State and output feedback nonlinear model predictive control: an overview," *European Journal of Control*, vol. 9, no. 2–3, pp. 190–207, 2003.

[13] Hornik, K., Stinchombe, M., and White, H., "Multi-layer feed-forward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[14] Johnson, W., *Helicopter theory*. New York, NY: Dover Publications, 1994.

[15] Kim, Y. H. and Lewis, F. L., *High-Level Feedback Control with Neural Networks*. Singapore: World Scientific Publishing, 1998.

[16] Leishman, G., *Principles of Helicopter Aerodynamics*. Cambridge, MA: Cambridge University Press, 2000.

[17] Narendra, K. and Parthasarathy, K., "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, 1990.

[18] Nørgaard, M., Ravn, O., Poulsen, N. K., and Hansen, L. K., *Neural Networks for Modelling and Control of Dynamic Systems*. London, UK: Springer-Verlag, 2000.

[19] Peters, D. A., "Flap-lag stability of helicopter rotor blades in forward flight," *Journal of the American Helicopter Society*, vol. 20, 1975.

[20] Peters, D. A. and Barwey, D., "A general theory of rotorcraft trim," *Mathematical Problems in Engineering*, vol. 2, pp. 1–34, 1996.

[21] Peters, D. A. and He, C., "Finite state induced flow models. part II: three-dimensional rotor disk," *Journal of Aircraft*, vol. 32, pp. 323–333, 1995.

[22] Prouty, R. W., *Helicopter Performance, Stability, and Control*. Malabar, FL: R.E. Krieger Publishing Co., 1990.

[23] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning internal representations by error propagation," No. 1 in Parallel Distributed Processing, ch. 8, pp. 318–362, Cambridge, MA: M.I.T. Press, 1986.

[24] Rutkowski, M., Ruzicka, G. C., Ormiston, R. A., Saberi, H., and Jung, Y., "Comprehensive aeromechanics analysis of complex rotorcraft using 2GCHAS," *Journal of the American Helicopter Society*, vol. 40, pp. 3–17, 1995.

[25] Yeo, H., Bousman, W. G., and Johnson, W., "Performance analysis of a utility helicopter with standard and advanced rotors," in *Aerodynamics, Acoustics, Test and Evaluation Technical Specialist Meeting*, (San Francisco, CA), American Helicopter Society, January 2002.