DESIGN OF A SHIPBOARD RECOVERY SYSTEM FOR A SHROUDED-FAN UAV

Giulio Avanzini , Guido de Matteis
Turin Polytechnic, Dept. of Aeronautical and Space Engineering ,
University of Rome "La Sapienza", Dept. of Mechanics and Aeronautics

Keywords: *UAV, automated ship landing, rotary–wing*

Abstract

An automatic ship recovery system for a lightweight, shrouded-fan Uninhabited Aerial Vehicle (UAV) is presented. The guidance and control functions are specifically designed for robust and computationally efficient operation of the flight control system of a lightweight vertical take–off and landing (VTOL) platform that provides three-dimensional mobility to dedicated sensors and payload. Automatic recovery in the presence of ship–deck motion is realized by a trajectory generation and tracking system the performance of which is analyzed and tested using computer based simulation.

1 Introduction

Automated operations of VTOL UAVs from ships are of relevant interest in order to provide the aerial system with improved characteristics in terms of reliability, safety and ship crew security, particularly in adverse weather conditions [1–3]. Envisioned applications of ship based UAVs include environmental monitoring using a lidar fluorosensor apparatus for sea diagnostic purposes [4], autonomous package delivery and collection in remote areas, and autonomous aerial mapping and obstacle detection.

The objective of this study is the development and testing by numerical simulation of an automatic ship recovery system for a lightweight rotary–wing type UAV for civil applications the characteristics of which are illustrated in detail in Refs. 5–7.

The UAV system comprises a ground control station (GCS) and the air vehicle and sensors, including a daylight TV camera, navigation sensors, and the UV laser transmitter, receiving telescope and an array of detectors, acquisition and controlling electronics used for detection of natural and pollutant components in the marine environment. The GCS capability include UAV manual control and flight planning and monitoring with real-time flight data and geographical information display, lidar sensor suite management and control, and TV camera control.

The air platform is a 1.9 m diameter, shrouded-fan UAV using counter rotating rotors for propulsion. Thrust and attitude control is provided by six 0.5 m long rotor blades, linearly tapered with 10 deg twist and 10 cm maximum chord, operating at a speed of 3,000 rpm. The rotors are driven by three two-stroke, air-cooled one–cylinder engines, each rated at 14 maximum horsepower at 11,000 rpm, located at the bottom of the central hub. Maximum performances are 50 kt and 2,000 m for airspeed and service ceiling, respectively, and 2 hr endurance.

The UAV is designed to operate at sea from the support ship Italica of the Italian Antarctic Program, shown in Fig. 1 with the aerial platform, that is presently used as a oceanographic ship as well as to transport person-

nel, equipment and consumables from New Zealand to the main Italian Antarctic base at Terranova Bay. The 130 m long ship has a displacement of 5,665 tons and a 10 x 10 landing pad located 10 m over the water surface.

Fig. 1 View of UAV recovery.

In our approach, the principal task of the recovery system is to guide and hold the vehicle into a specified position, 0.5 m over the landing pad center. We assume that the second and final phase of the landing, that is to bring to zero the vertical distance from the ship deck, is accomplished by manual control. Design objective is the development of approach and landing capability using supervisory control so that the vehicle can be safely operated by personnel with limited training, with reduced workload even in adverse weather conditions. The system consists of three integrated modules, that is, an off-line trajectory generator, a guidance algorithm and an inverse simulation algorithm for real-time command generation. The general architecture of a similar system was presented in Ref. 8, where ship landing of a helicopter model is dealt with and the above elements are discussed in detail.

We consider the implementation of the aforementioned modules in the UAV flight control system, where a robust dynamic controller at the inner control level is used for vehicle stabilization and response uncoupling [9]. Further elements in the onboard avionic suite are a GPS aided Inertial Navigation System (INS) and a control augmentation system providing attitude command/hold in pitch/roll axes and rate command in yaw axis response characteristics. Higher level control tasks, that is, ground speed, altitude, heading and rate of climb command/hold are provided by SISO loops that generate reference commands for the inner controller.

A sub-optimal four-dimensional flight path is determined by a computationally efficient trajectory optimization algorithm where UAV operational limits as well as geometric constraints such as ship over-deck structure are accounted for. The desired trajectory is defined in the inertial frame and, in this phase, the observed averaged ship motion due to seaway is taken into consideration so that at the end of the approach the vehicle maintains a fixed position with respect to the ship-deck. Next, the guidance algorithm generates velocity commands to drive the vehicle onto the specified trajectory and, finally, an inverse simulation method [10] is adopted to determine feedforward commands for trajectory tracking. Errors induced by external disturbances and model uncertainties are accounted for by the inner control loop. A sketch of the ground and flight segments of the recovery system is shown in Fig. 2.

Six degree-of-freedom real-time, computer simulation is used to test design performance and robustness and to assess the ability of guidance and control system and, in the final phase, of the remote pilot, to land the vehicle on the ship deck. To this end, a complete nonlinear model of the UAV is used [7], featuring a detailed aerodynamic database, actuator dynamics with rate and position saturation and engine dynamics and related feedback loop for rotor rpm control.

Ship motion caused by sea waves is accounted for by sine functions for heave, roll and pitch displacements where amplitude and frequency are assigned as functions of ship inertial characteristics and initial conditions (speed and heading), and wave height and di-

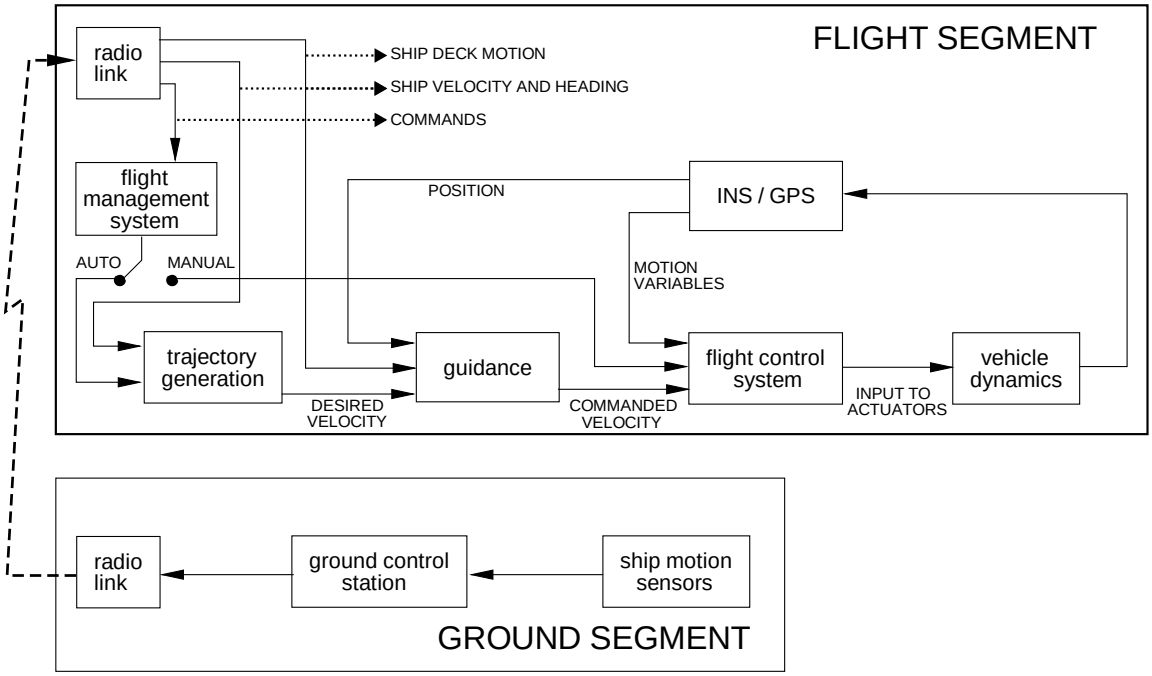

Fig. 2 Schematic of the shipboard recovery system.

rection. Note that we assume that landing point position and ship orientation, as determined by ship motion sensors, are transmitted to the UAV.

In all the cases, the GCS hardware is in the simulation loop to allow for an efficient management of the whole recovery maneuver where manual control is requested only for the final touch-down, when the guidance and control system has already brought the vehicle in a fixed position relative to the landing pad.

In what follows, a concise description of the UAV model is reported in Section 2. Next, the modules for trajectory calculation, guidance and inverse simulation are discussed in Section 3. Results for a simulated recovery maneuver are presented and analyzed in Section 4. A section of conclusions ends the paper.

2 Vehicle Model

In this Section the main features of the UAV model are illustrated in short. Reference should be made to Refs. 6 and 7 for a detailed description of the air vehicle mathematical modelization.

We use a 6 DOF baseline model of the rigid UAV where the aerodynamic force and moment of the airframe are provided in tabular form. Rotor thrust and moment are determined by blade element theory assuming quasi–static inflow dynamics and neglecting the interaction of rotor inflow with the shroud. Ground effect on the aerodynamic actions exerted by rotors and shroud is not taken into consideration.

Control of the two swashplates is realized by a set of 6 actuators commanded by 4 input channels for collective pitch (δ_C), longitudinal (δ_B) and lateral (δ_A) cyclic and differential collective (δ_D). Actuator dynamics are modeled by second–order transfer functions with natural frequency and damping coefficient equal, respectively, to 12.6 rad s^{-1} and 0.85, cascaded with a constant delay equal to 0.03 s.

The vehicle control system is based on two hierarchically nested layers where stability augmentation and response uncoupling (in-

ner layer) and control augmentation (outer layer) are carried out. The inner control loop uses a robust multivariable controller the design of which was developed in the framework of μ–synthesis control theory [11] whereas attitude–command/attitude–hold (ACAH) response characteristics are obtained by simple SISO loops at the outer level [9].

Four–axis manual control is provided by a single–axis joystick for vertical velocity control and a three–axes joystick for reference commands on pitch/roll/yaw angles.

The three engines are modeled by first–order lags with a time constant of 0.01 s. A RPM governor that consists of a lead–lag network with input and output given by rotor speed Ω and throttle actuator command, respectively, realizes precise control of Ω when the auto–throttle autopilot function is engaged [7].

3 Guidance and Control Functions

As we said, the recovery system is composed of three software modules for (i) trajectory generation, (ii) computation of guidance commands to keep the vehicle on the prescribed flight path and (iii) generation of commands on heave velocity and attitude angles as reference inputs for the outer control loops.

3.1 Trajectory generation

The off–line trajectory generation procedure, presented in Ref. 8, if briefly recalled heretofore. An admissible flight path in the navigation frame $\mathcal{F}_\mathcal{N}$, that is, a frame centered on the surface of a non rotating, flat Earth with axes pointing North, East and Down is expressed as

$$\begin{aligned} \boldsymbol{R}^N_{V_{\mathrm{DES}}} &= \boldsymbol{R}^N_{S_0} + \boldsymbol{L}_{NS_0}\boldsymbol{R}^S_{V_{\mathrm{opt}}} + k(d)\left[\Delta\boldsymbol{R}^N_S \right. \\ &+ \left. \left(\boldsymbol{L}_{NS} - \boldsymbol{L}_{NS_0}\right)\boldsymbol{R}^S_{\mathrm{LP}}\right] \end{aligned} \tag{1}$$

where $\boldsymbol{R}^N_S$ is the ship position vector, $\boldsymbol{R}^S_{V_{\mathrm{opt}}}$ is the sub–optimal flight–path calculated in the ship–fixed frame $\mathcal{F}_\mathcal{S}$, $\boldsymbol{R}^S_{\mathrm{LP}}$ gives the landing pad coordinates in $\mathcal{F}_\mathcal{S}$ and $\boldsymbol{L}_{NS}$ is the transformation matrix from $\mathcal{F}_\mathcal{S}$ to $\mathcal{F}_\mathcal{N}$. Next, $\Delta\boldsymbol{R}^N_S$ is the vertical perturbation of ship motion with respect to a straight course (as perturbations in the horizontal plane are neglected), $\boldsymbol{R}^S_{\mathrm{LP}}$ is the position vector of landing pad and the subscript 0 indicates calm sea, whereas the third term in Eq. (1) is used to lock the UAV position onto the ship deck motion when the distance $d = \left|\boldsymbol{R}^N_V - \boldsymbol{R}^N_S\right|$ is shorter than a prescribed length d_0.

In this respect, when linear and angular displacements of the flight deck due to waves are present, the UAV and ship motions are somewhat coupled as the distance d decreases. This is accomplished using the weight function $k(d) \in [0,1]$, defined as

$$\begin{aligned} k &= \exp\left[-\left(\frac{d-d_0}{10d_0}\right)^2\right] \quad \text{for } d > d_0 \\ k &= 1 \quad \text{for } d \le d_0 \end{aligned} \tag{2}$$

From Eq. (1), the desired velocity is expressed as

$$\begin{aligned} \dot{\boldsymbol{R}}^N_{V_{\mathrm{DES}}} &= \dot{\boldsymbol{R}}^N_{S_0} + \boldsymbol{L}_{NS_0}\dot{\boldsymbol{R}}^S_{V_{\mathrm{opt}}} \\ &+ k(d)\left[\Delta\dot{\boldsymbol{R}}^N_S + \boldsymbol{L}_{NS}\tilde{\boldsymbol{\omega}}^S_S\boldsymbol{R}^S_{\mathrm{LP}}\right] \end{aligned} \tag{3}$$

being $\boldsymbol{\omega}^S_S$ the ship angular motion in $\mathcal{F}_\mathcal{S}$.

As for the calculation of the UAV flight path $\boldsymbol{R}^S_{V_{\mathrm{opt}}}$, the coordinates are parametrized using Legendre polynomials, boundary conditions are assigned on position and velocity at the initial and final time, and suitable nonlinear constraints are enforced on $\boldsymbol{R}^S_V$, $\dot{\boldsymbol{R}}^S_V$ and $\ddot{\boldsymbol{R}}^S_V$ to account for limitations on vehicle performance and/or flight path geometry for obstacle avoidance. Next, a constrained optimization problem is solved where the final time of the trajectory is minimized even though, for the sake of computational efficiency in practical applications, feasible (i.e. satisfying all the constraints) rather than optimal solutions are pursued [8].

3.2 Guidance algorithm

In order to drive the UAV onto the prescribed flight path from the actual position, a velocity command is calculated at the beginning of each discretization interval Δt, to be used as the desired output for the inverse simulation algorithm that, in the same time frame Δt, generates a constant reference command for the outer controller [12].

The velocity command $\dot{\boldsymbol{R}}^N_{V_{\rm COM}}$ is expressed as

$$\dot{\boldsymbol{R}}^N_{V_{\rm COM}}(t) = \dot{\boldsymbol{R}}^N_{V_{\rm DES}}(t) - K_G\left[\boldsymbol{R}(t_{k-1}) - \boldsymbol{R}^N_{V_{\rm DES}}(t_{k-1})\right] \quad (4)$$

where $t_{k-1} \leq t \leq t_k$, $t_k = k\Delta t$ and K_G is a guidance gain [8].

3.3 Command generation

Reference inputs, namely $\dot{\boldsymbol{R}}^N_{V_{\rm des}}$, $\phi_{\rm des}$, $\theta_{\rm des}$ and $\psi_{\rm des}$ at the outer control level are computed as functions of $\dot{\boldsymbol{R}}^N_{V_{\rm COM}}$ by an inverse simulation method, the characteristics of which are presented in Ref. 10. The method uses a simplified dynamical model of the vehicle with linear aerodynamics, where the so–called slow states $(\boldsymbol{V}, \boldsymbol{R}) \in \mathcal{R}^6$ are accounted for, being $\boldsymbol{V}$ and $\boldsymbol{R}$ the UAV velocity and position vectors, respectively. The simplified model was obtained from the linearization of the vehicle equations of motion in the reference condition of trimmed level flight at 9 $\rm ms^{-1}$, the same condition used for the synthesis of the inner MIMO robust controller. The pseudo–control variables of the model are the collective pitch and Euler's angles. Figure 3 shows the structure of the feedback loops and the inputs to inner and outer controllers. In the figure, it is $\Delta\boldsymbol{R}^N_V = \boldsymbol{R} - \boldsymbol{R}^N_{V_{\rm DES}}$ and $\boldsymbol{V}^N_V \equiv \dot{\boldsymbol{R}}^N_V$.

Reference signals for the desired value of Euler's angles and vertical velocity component are determined at each discretization interval by a constrained optimization procedure where the constrains $\beta = 0$ on the sideslip angle is also enforced. Use of a power series expansion for the trajectory in the Frenet triad [13] allows to formulate an algebraic problem for the vehicle trajectory dynamics. The attitude (fast) state variables are obtained at each discretization interval from a piecewise linear interpolation of the Euler's angles within the timeframe Δt. Assuming that the angular rates achieve their steady state values within a smaller timescale, the linear set of equations $\dot{p} = 0$, $\dot{q} = 0$, and $\dot{r} = 0$ is solved with respect to the fast control variables, namely δ_A, δ_B, and δ_D, the effect of which on the slow timescale dynamics cannot be neglected when dealing with rotorcraft [10].

This approach is very efficient and stable from the computational point of view and provides accurate calculation of reference commands at each discretization interval where the number of iteration of the sequential quadratic programming algorithm used for the solution of the minimization problem is always very low and, more important as far as real–time implementation of the algorithm is concerned, convergence problems are practically eliminated.

4 Results and Discussion

The simulation code is entirely built using Simulink by MathWorks Inc. The UAV model as well as the software modules for trajectory generation, guidance and inverse simulation are written in C-code and implemented in the Simulink environment via the S-function interface [14]. In particular, we remark that sub–optimal trajectory is computed off–line at the beginning of the simulation, whereas a discrete–time S-function, where the sampling interval is equal to Δt, is used for the computation of pseudo–commands.

As for the implementation of trajectory generation and tracking algorithms in the flight computer, automatic generation of C code that can run as a stand–alone application is achieved by means of Real Time Workshop software [14]. As a result, testing and validation of software modules is carried out in the

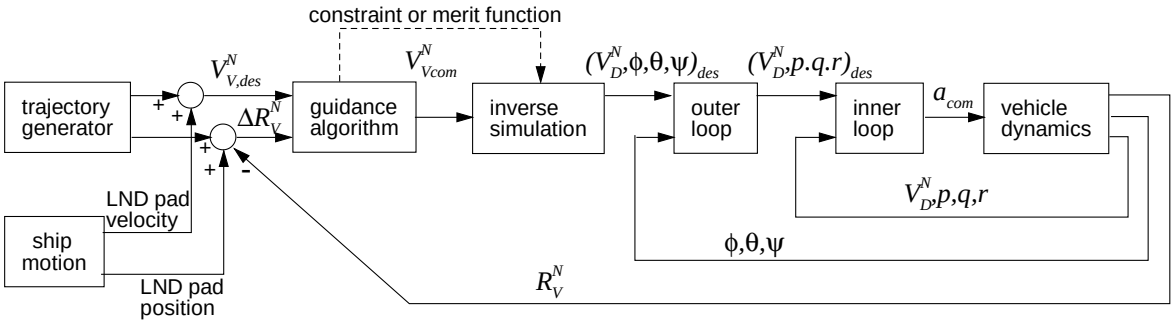

Fig. 3 Block diagram of trajectory generation and tracking system.

real–time, hardware–in–the–loop (HITL) simulation facility described in Ref. 7.

Several geometric constraints are enforced in the trajectory generation procedure in order to avoid possible interference of vehicle flight path with ship over–deck structure and rigging [8].

For the calculation of the landing trajectory we assume that the ship is in forward motion on a straight course at a constant speed $\dot{R}^N_{N_S} = 5$ m s^{-1}, and seaway causes heave, roll and pitch motions that are represented by sine functions. In particular, amplitude and period of forcing functions are (3 m, 51 s), (30 deg, 40 s) and (10 deg, 47 s) for heave, roll and pitch, respectively.

At the initial time ($t_i = 0$) the UAV is in steady–state rectilinear flight at an altitude of 40 m and velocity $V = 9$ m s^{-1}, heading North, in a backward (500 m) and lateral (150 m to the right) location with respect to the ship. Atmospheric wind is not considered in the simulation.

Maximum speed of the UAV and maximum rate of descent are constrained to 10 and 2.5 m s,$^{-1}$ respectively. Further constraints concern the maximum admissible value for the module of the UAV acceleration vector relative to the ship motion, set to zero at the initial and final (t_f) times of the maneuver and equal to 7 m s^{-2} for $t_i < t < t_f$.

As for the constant parameters in the problem, the distance d_0 in Eqs. (2) is equal to 15 m and the gain in Eq. (4) is $K_G = 0.8$ s^{-1}. Finally, the discretization interval is $\Delta t = 0.25$ s, and the time step for the 4th–order Runge–Kutta algorithm used in the forward simulation of the nonlinear UAV model is equal to 0.01 s.

Results for the recovery maneuvers are reported in Figs. 4–7 where time histories of significant state and control variables are shown. In particular, Fig. 4 shows the rather smooth reduction of flight speed to match the ship cruise velocity. UAV position in $\mathcal{F}_N$ also varies smoothly to acquire the same motion of the landing pad (dashed lines) as perturbed by seaway.

Figure 5, where the Euler's angles are reported vs. time, shows that the negative pitch angle at trim is increased to decrease speed, as expected, whereas small amplitude oscillations on roll angle are induced by the necessity of tracking the position specified at the final time, that is fixed with respect to the landing pad. More relevant is the variation of yaw angle and, in this case, performance of feedforward command generation system could be improved by relaxing the constraint on β so as to allow sideslip as the vehicle is sufficiently close to the ship.

Plots of control displacements in Fig. 6 shows the oscillatory behavior induced by the ship motion that is somehow locked after 160 s in the maneuver. Longitudinal cyclic pitch command is reduced for obtaining the commanded lower flight speed, while the variation of differential collective to balance the rotor

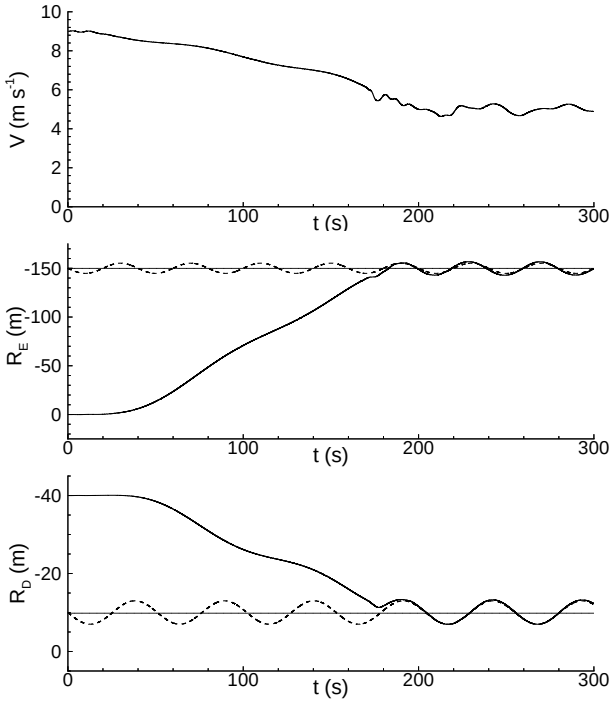

Fig. 4 Flight speed modulus (V), lateral ($\boldsymbol{R}_E$) and vertical ($\boldsymbol{R}_D$) coordinates in $\mathcal{F}_N$ vs. time (t).

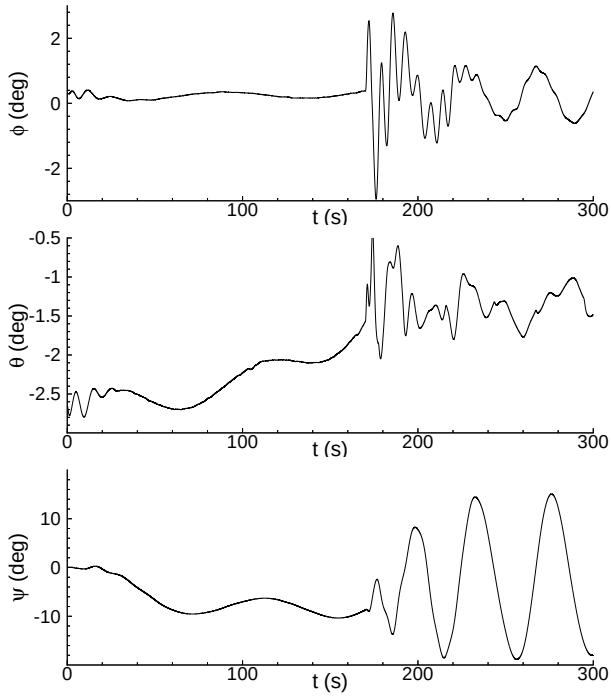

Fig. 5 Roll (ϕ), pitch (θ) and yaw (ψ) angles vs. time.

torque is practically negligible.

Finally, Fig. 7 reports the UAV position error with respect to the specified flight path $\Delta\boldsymbol{R}_V^N$. It is apparent that, as desired, in the initial phase of the recovery maneuver the error remains sizeable as the ship motion is ignored by the guidance system. When the distance from the ship is about 15 m, the vehicle accurately tracks the trajectory specified in the ship–fixed reference frame and the error is brought to very low values.

5 Conclusions

In this paper a shipboard recovery system for a VTOL UAV has been presented. Trajectory generation and guidance modules were developed to provide reference commands for the inner/outer–loop robust controller of the vehicle.

Nonlinear simulation of a recovery maneuver in the presence of ship motion due to seaway shows that the system is consistently able to accurately drive the UAV onto a specified location over the landing pad with negligible relative motion with respect to the ship.

Adopted algorithms for trajectory generation and feedforward command evaluation address relevant issues concerning required computational time, and accuracy and convergence characteristics of the optimization and inverse simulation algorithms. Software development in Simulink environment allowed evaluation of the control laws in a HITL simulation facility where the UAV stability and control augmentation system already resides in the flight computer.

Ongoing activity is dedicated to further development and testing of the UAV flight management system. Specifically, future research will address the problem of architectural design of the complete control system, including the automatic recovery capability, for onboard implementation, and of sensor failure recognition and control reconfiguration.

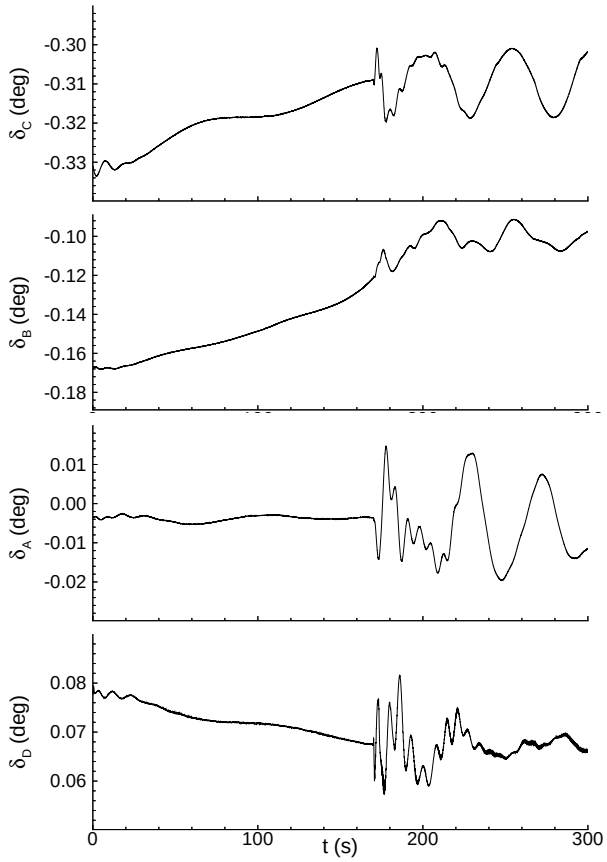

Fig. 6 Control commands vs. time.

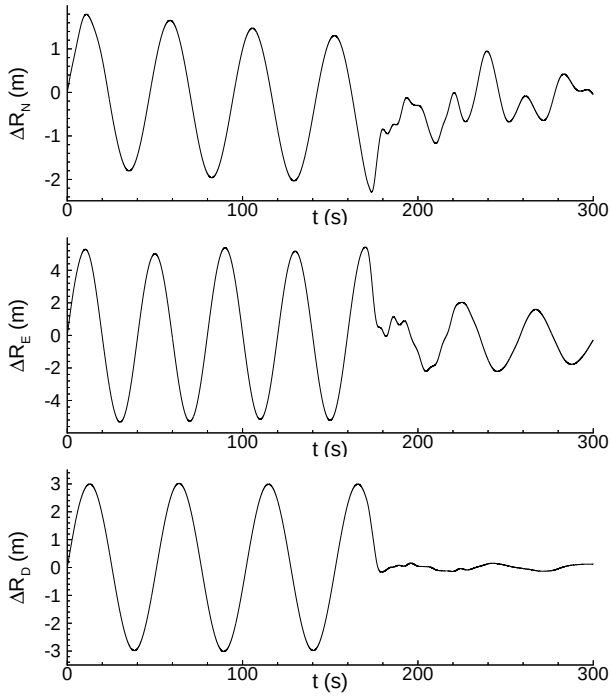

Fig. 7 Position error vs. time.

6 Acknowledgements

This work was supported by the Italian Ministry of University and Scientific Research (MIUR).

7 References

1. Weidel M, Alles W. Automatic flight control system for an unmanned helicopter: system design and flight test results. *AGARD Flight Mechanics Panel Symposium*, Turin, Italy, May 1994, pp. 15.1-15.9.

2. Bole M. Design of an automatic landing system for twin rotor vertical take–off and landing unmanned air vehicle. Ph.D. Thesis, Dept. Mechanical Engineering, Concordia University, Montreal, Nov. 1999.

3. Pelletier M, Giroux R. On–line trajectory generation for autonomous VTOL–UAV maneuvers. *AIAA Guidance, Navigation amd Control Conference and Exhibit*, Denver, Paper 2000–4168, Aug. 2000.

4. Avanzini G, et al. A remotely piloted platform for environmental monitoring in antarctica. *Fifth International Airborne Remote Sensing Conference*, San Francisco, USA, Aug. 2001.

5. Avanzini G, D'Angelo S, De Matteis G. Design and development of a VTOL uninhabited aerial vehicle. *XVI Conference of the Italian Association of Aeronautics and Astronautics (AIDAA)*, Palermo, Italy, Sept. 2001.

6. Avanzini G, D'Angelo S, and de Matteis G. Performance and stability of a ducted-fan uninhabited aerial vehicle. sl 39th Aerospace Sciences Meeting and Exhibit, Reno, USA, Paper 2001-0844, Jan. 2001.

7. Avanzini G, D'Angelo S, de Matteis G. Modelling and simulation of a shrouded-fan UAV for environmental monitoring. *AIAA AIAA's 1st Conference & Workshop on Unmanned Aerospace Vehicles, Systems, Technologies and Operations*, Portsmouth, VA, Paper 2002-3464, May 2002.

8. Avanzini G, Busato A, de Matteis G. Trajectory generation and tracking for ship–deck landing of a VTOL vehicle. *AIAA Atmospheric Flight Mechanics Conference*, Montreal, Canada, Paper 2001-4004, Aug. 2001.

9. Avanzini G, de Matteis G, Fresta F. Robust multivariable control of a shrouded–fan uninhabited aerial vehicle. *AIAA Atmospheric Flight Mechanics Conference*, Monterey, California, Paper 2002-4703, Aug. 2002.

10. Avanzini G, de Matteis G. Two-timescale inverse simulation of a helicopter model. *Journal of Guidance, Control and Dynamics*, Vol. 24, No. 2, pp. 330-339, 2001.

11. Zhou K, Doyle J, Glover K. *Robust and Optimal Control*. Prentice–Hall, Englewood Cliffs, NJ, 1996.

12. de Matteis G, de Socio L, Leonessa A. Solution of aircraft inverse problems by local optimization. *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 3, pp. 567–571, 1995.

13. Avanzini G, de Matteis G, de Socio L. Natural description of aircraft motion. *Journal of Guidance, Control and Dynamics*, Vol. 21, No. 2, pp. 229–233, 1998.

14. *Simulink: Dynamic System Simulation for Matlab*. The Mathworks Inc., Chap. 3, 1997.