

# RECONFIGURABLE CONTROL FOR VTOL UAV SHIPBOARD LANDING

David G. Ward, Jeffrey F. Monaco, and John D. Schierman

*Barron Associates, Inc., 1160 Pepsi Place, Suite 300, Charlottesville, VA 22901*

## Abstract

This paper presents an intelligent reconfigurable control system for autonomous landing of a vertical takeoff and landing (VTOL) uninhabited air vehicle (UAV) on a destroyer helicopter deck. The reconfigurable control approach leverages previous inner-loop reconfigurable control work for manned fixed-wing aircraft whereby 1. polynomial neural networks model the nominal nonlinear response characteristics of the UAV, 2. on-line regularized parameter identification estimates models of off-nominal behavior owing to uncertainty or failures, and 3. indirect-adaptive model-predictive control provides consistent inner-loop responses. In this work, the inner-loop controller is tasked with tracking velocity and angular rate commands and the authors have added an optimum-path-to-go (OPTG) autopilot that computes approach and landing trajectories. These trajectories are computed in real time and minimize a weighted combination of time and UAV accelerations; such a cost function allows the autopilot to be less aggressive in cases where failures have reduced the UAV control authority. Nonlinear simulation results show that, unlike non-adaptive controllers, the intelligent controller is consistently able to land the UAV with very small miss distances, even in cases of multiple failures, severe turbulence, high seas, and uncertainties in the aerodynamic data.

## Introduction

Of current interest for naval applications is automated shipboard landing of Vertical Take-Off/Landing (VTOL) Uninhabited Air Vehicles (UAV) in the presence of uncertainties, disturbances, and failures. Control laws are currently designed off-line with extensive analysis, simulation, iteration, and redesign. This traditional approach is often too burdensome to address all possible failure scenarios. The objective of the current task was to develop control design algorithms that execute on-

line in real-time, resulting in a controller that reacts to changes or events as they occur and tailors the control laws to yield the maximum possible performance.

In previous work, Barron Associates, Inc. (BAI) demonstrated a reconfigurable inner-loop control design that provides close tracking of flying-qualities models and continuous control law adaptation, including reconfiguration for single and multiple unforeseen effector impairments. This controller was successfully demonstrated in a series of VISTA/F-16 flight tests that culminated with a smooth landing in crosswind conditions with a (simulated) completely missing left horizontal tail surface.<sup>1</sup> The reconfigurable controller correctly adapted to the changes in the stability and control derivatives and avoided what would have been a destabilizing failure. By concentrating on control design algorithms rather than a specific control law, the resulting self-designing flight control system is more easily extensible to new tasks, missions, aircraft modifications, and weapons, as well as to other aircraft. Recently, an extended version of this reconfigurable control system was successfully demonstrated on the Air Force Wright Laboratory's Reconfigurable Systems for Tailless Fighter Aircraft (RESTORE) program. Challenges included significant numbers of effectors that generate moments in multiple axes and have nonlinear interactions. Additionally, the airframe was unstable in both the longitudinal and directional axes.

The current problem of automated VTOL/UAV shipboard landings necessitated the development of outer-loop trajectory optimization and control approaches in addition to the inner-loop control algorithms. Toward this end, a baseline Optimum-Path-To-Go (OPTG) autopilot that works in concert with a reconfigurable inner-loop controller was developed. The OPTG approach allows rapid on-line real-time computation of (approximately) optimal trajectories for a nonlinear system. To the extent that the inner-loop controller is able to achieve consistent performance, an OPTG trajectory will remain optimal. Because the OPTG solutions are re-computed at

---

Copyright © 1999 by Barron Associates, Inc. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

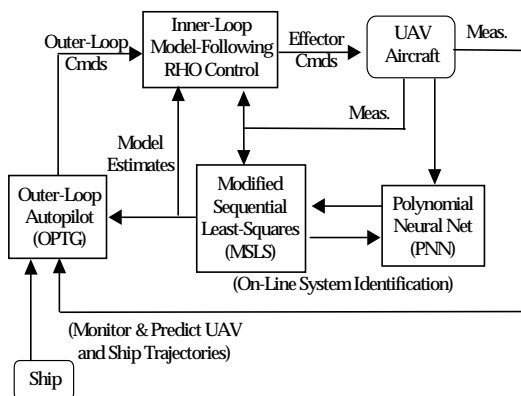


Fig. 1: Reconfigurable UAV Control System

each update, the autopilot loop accounts for suboptimality that arise due to ballistic (steady) winds, turbulence, and imperfect model following in a manner similar to that of the Receding-Horizon-Optimal (RHO) controller.<sup>2,3</sup> In fact, the OPTG autopilot can be thought of as a special case of RHO control.

The overall control architecture is shown in Fig. 1. The individual elements in the figure will be discussed in the next sections, followed by a presentation of simulation results.

### Simulation System Description

A nonlinear six-degree-of-freedom UAV simulation for ship landing was developed. This simulation was based on software and data provided by the Navy and Texas A&M; however, extensive modifications were incorporated including 1. actuator nonlinearities, 2. a large variety of actuator failure modes, 3. the ability to add uncertainties to the aerodynamic data, 4. nonlinear dynamics including time-varying air data and inertial cross couplings, 5. a Dryden gust model, and 6. ship-motion simulation.

For this study, BAI used aerodynamic data and other physical parameters representative of a YAV-8B aircraft. The seven effectors used by the aircraft are left and right stabilators, pitch nozzle angle, left and right ailerons, rudder and aft thrust. The vehicle's translational and angular velocities were considered the measurements used by the parameter identification routine as well as the controller. These measurements are considered a realistic set and have been used in previous flight test work.<sup>1</sup>

Body-axes six degree-of-freedom force and moment equations, complete kinematics and the full nonlinear navigation equations were incorporated

into the simulation. The kinematic and navigation equations are necessary to control the UAV trajectory for the shipboard approach and landing tasks. Accurate modeling of atmospheric turbulence is important in the quantitative design and assessment of the intelligent control algorithms; therefore, stochastic gusts and ballistic (steady) winds were incorporated into the simulation using the widely accepted Dryden gust model.<sup>4</sup> Body-axes wind gusts were added to the body-axes (linear and angular) acceleration equations, and ballistic (steady) winds were incorporated directly as biases on the navigation equations.

Frequently, a control design that appears to work well will break down in the presence of actuator dynamics. Some of the most troubling dynamics are actuator rate limits; these introduce phase lag and make the closed-loop system nonlinear. Therefore, nonlinear actuator dynamics in the form of position and rate limits on all actuators were incorporated. While an actuator transfer function could have been incorporated, BAI has found that it is frequently the actuator limits (especially rate limits) that have the most profound adverse effects on closed-loop system performance. Because all aerodynamic data were expressed in terms of percentage-of-maximum-deflection, actuator position limits are  $\pm 100\%$ . Actuator rate limits were chosen such that it takes (a minimum of) 20 controller steps (at 100 Hz) to transition between maximum and minimum effector displacement.

In addition to rate and position limits, capability was added to introduce failures in each of the actuators. There are two primary types of failures: stuck and missing actuators. For a stuck actuator, the time of failure and position at failure can be specified. Additionally, some or all of an actuator can be specified as missing. The analyst can specify any combination of failures and failure times via a simulation control file.

BAI employed the Navy's *PREDICT*<sup>5</sup> ship motion simulation software to simulate ship dynamics for a variety of sea states, headings, and wind conditions. The software requires the user specify detailed information about the hull and appendage particulars of the ship to simulate, the coordinates of points of interest on the ship, nominal ship speed and (wave-relative) heading, and information about the types of seas which would be present in the simulation, including significant wave height, modal period, short-crested vs. long-crested waves, etc.

Numerical values of ship parameters were used that were representative of a CG82-class destroyer. A large data base of ship simulation results was generated and used directly with the UAV simulation. Because the UAV simulation was executed repeatedly during its development and evaluation, it was important that the ship motion be different from one simulation run to the next. Otherwise, the wave motion is not truly random, and the controller may learn to anticipate specific waves and not be robust to wave motion in general. To overcome this problem, each time the UAV simulation was executed, a random access point in the ship motion time-series data was chosen to correspond to the initial simulation time. Thus, from run to run, the specific ship motions were different and appeared to be random to the controller.

### Reconfigurable Inner Loop Control

Traditional control design methods frequently require specialized knowledge, substantial off-line analysis, and extensive real-time tuning and validation (often accompanied by numerous iterations of large and complex pre-specified gain schedules). Design and development expenses can mushroom when one requires *a priori* control modes designed to handle the large universe of possible anomalies that may be experienced during the system service life. Rather than rely on numerous control system implementations based on pre-hypothesized impairment or airframe damage scenarios, the proposed reconfigurable controller uses an indirect-adaptive control architecture. The algorithm incorporates a time-varying model of the current system dynamics using a combination of neural networks and a novel on-line parameter identification technique that tracks rapidly time-varying parameters and is robust to adverse conditions of low excitation or correlated inputs (the latter frequently being the case with use of large numbers of effectors). The identified model of the current system dynamics is communicated to an on-line control design module and is used to predict future system states. The control module then computes a set of effector commands that will achieve stable responses as close as possible to a set of desired responses (model-following). The ability of the intelligent controller to reconfigure a system rapidly after single or multiple impairments, making effective use of residual effector authority, greatly enhances survivability.

The model-following design approach has the

potential to reduce the time and expense of developing advanced controllers for new systems, operational conditions, configurations, mission requirements, and failure modes. Additionally, the model-following approach means that to the pilot or, in this case, autopilot, the inner-loop dynamics remain consistent across operating conditions and a variety of failures. Thus, the autopilot loop can be designed to perform optimally on the nominal dynamics and be assured that the closed-loop system will perform as expected under a much wider range of circumstances. The remainder of this section describes each component of the inner-loop intelligent control system.

#### 1. Receding-Horizon Optimal (RHO) Control

Receding-Horizon Optimal (RHO) control is a form of model-predictive control, in which a finite-time optimal control solution is computed that minimizes the error between predicted plant responses and the desired plant responses.<sup>2,3</sup> This minimization results in a sequence of optimal control commands, and the first command, corresponding to the current time, is applied to the system. At the next control update, rather than applying the second command in the open-loop optimal command sequence, the finite horizon optimization is completely recalculated using up-to-date estimates of the plant dynamics, desired control, and system state. In this way, the open-loop finite-horizon optimal control problem becomes closed-loop, and the optimization horizon is said to *recede* because the controller never applies the commands corresponding to the end of the horizon.

A RHO controller shares a number of advantages with LQ control techniques, especially stability and robustness.<sup>6</sup> However, unlike infinite-time LQ control, a receding-horizon controller *anticipates* desired responses and optimizes tracking over a short horizon. This quality adds desirable phase lead to the controller and makes it attractive for multi-input multi-output (MIMO) control problems where one is interested in achieving specific *transient* responses as well as certain *steady-state* responses.<sup>7</sup> In fact, one paper recently argued that, for problems that are not inherently linear-time-invariant (LTI), receding-horizon control is “the only viable controller synthesis method.”<sup>8</sup>

In RHO, the nonlinear aircraft equations of motion are represented as time-varying linear equations of the form:

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)\delta(t) + d(t) \\ y(t) &= C(t)x(t) + D(t)\delta(t) + e(t)\end{aligned}\quad (1)$$

where the states consist of the body-axes translational velocities and angular rates. Or,  $x = [u, v, w, p, q, r]$ .  $\delta$  is an array of effectors,  $y$  is the vector of outputs the controller is trying to track,  $d$  and  $e$  are bias vectors included to account for linearizations at non-trim flight conditions. The on-line parameter identification algorithm provides the matrices  $A$ ,  $B$ ,  $C$ , and  $D$ , which are time-varying functions of the current states, effector commands, flight condition, failure scenario, etc.

The RHO control law uses the above model of the desired system transient responses to *predict and anticipate* the system behavior and generate commands to track the models optimally. The RHO gains are calculated to minimize the cost function:

$$r^T Q_f r \Big|_{t=t_f} + \frac{1}{2} \int_{t=t_0}^{t_f} (r^T Q_t r + x_i^T Q_i x_i + \delta^T Q_u \delta) dt \quad (2)$$

where  $r = y_m - Cx$ ,  $y_m$  is the desired response and  $x_i$  is a vector of integral-error states. Note the combination of penalties on tracking error, integrated tracking error, actuation, and final output error. The Riccati Differential Equations (RDE's) required to solve for the control law which minimizes this cost function can be derived using calculus of variations. Furthermore, methods exist which can solve the RDE's in real time with limited computing bandwidth.<sup>1</sup>

For the UAV application, it is of interest to control ground speed, vertical velocity, and angular rates. Thus,  $y = [V, \dot{z}, p, q, r]$ . Therefore, the matrix  $C$  in the output equation relates this response vector to the state vector given above.

Given that the UAV has seven independent effectors, it is not unreasonable to control five outputs; however it should be noted that the fewer outputs one cares about tracking, the more redundancy is available for reconfiguration. In this case, should there not be enough control authority to track five independent quantities, the controller will forego tracking of particular variables based on the relative penalty weights. In the current implementation, the emphasis is on tracking the angular rates because

control of these is necessary for stability.

## 2. Modified Seq. Least-Squares (MSLS)

On-line parameter identification is required to account for unforeseen changes due to failure, novel system configurations, etc. A Modified Sequential Least-Squares (MSLS)<sup>9</sup> algorithm is used here for the parameter identification problem. The *nonlinear* system dynamics are represented by the *linear time-varying* state equations given earlier, where the nonlinear system is *linearized* at each update. The reduction of a complex nonlinear system to a suitable piecewise-linear approximation, and stability and robustness proofs regarding such approximations can be found in the literature.<sup>10,11</sup>

The standard normal equations are typically used to find the least-squares solution. However, there are two significant difficulties with on-line parameter estimation using traditional regression approaches. These are: (1) data collinearities and (2) time-varying parameters. Data collinearities occur when any of the input variables to a system being identified can be represented as linear combinations of other input variables. Such conditions frequently occur in systems with more effector than control degrees-of-freedom. Time-varying parameters can arise either during changes in operating condition (slow variations) or during impairment, change of system state, or flight in a nonlinear regime (fast variations). It can be shown that the pseudo inverse in the normal equation is nearly "singular" if either of these cases occurs.

A number of batch and nonlinear regression algorithms address these singularities through a process known as *regularization*,<sup>12,13,14,15</sup> and the MSLS approach is one such method with certain advantages. Regularization is achieved by including additional constraints in the cost function to be minimized. A simplified representation of the MSLS cost function is:

$$\begin{aligned}J(\theta) &= \frac{1}{2} \sum_{n=t_0}^t r(n)^T r(n) \\ &+ q(n)^T W_0 q(n) + p(n)^T W_1 p(n)\end{aligned}\quad (3)$$

$$\begin{aligned}r(n) &= y(n) - \theta^T(n-1)\phi(n) \\ q(n) &= \theta(n) - \theta(n-1) \\ p(n) &= M\theta(n) - k\end{aligned}\quad (4)$$

where  $y$  is the measurements,  $\theta$  represents the estimated parameters, and  $\phi$  is the vector of system states and control inputs. The three terms, as ordered, in the above cost function represent:

1. The standard least squares cost function.
2. *Temporal* constraints which penalize parameter estimates that deviate from previous estimate. This results in a smoothing over time, but, as will be shown, does not hinder the ability to track rapidly varying parameters. *Temporal* constraints have the effect of reducing the data window size during periods of high excitation, resulting in rapid parameter tracking.<sup>16</sup>
3. *Spatial* constraints, which arise from imposing the linear constraint relationship  $K = M\Theta$ , penalize parameter estimates that diverge from *a priori* estimates of their true values (i.e. nominal (*unfailed*) values). These estimates can be constant, or they can be provided by on-board neural network system models, discussed next.

### 3. Polynomial Neural Networks (PNNs)

A Polynomial Neural Network (PNN) estimation model was used to construct the spatial constraints in the parameter identification algorithm. BAT's employed its own *GNOSIS* software package to synthesize the PNN's from a numerical database of desired input-output relationships. *GNOSIS* selects those variables most useful for the modeling task from the list of available candidate inputs, and delivers a PNN with an optimum level of complexity. *GNOSIS* has been applied successfully to a number of challenging problems in a wide variety of applications.<sup>17</sup>

Polynomial networks are compositions of Kolmogorov-Gabor (KG) multinomials.<sup>18</sup> Kolmogorov and Gabor each demonstrated the near universality of multinomials in representing physical processes, including dynamical systems. The KG multinomial is

$$y = a_0 + \sum_i a_i x_i + \sum_i \sum_j a_{ij} x_i x_j + \sum_i \sum_j \sum_k a_{ijk} x_i x_j x_k + \dots \quad (5)$$

and can model any analytic, single-valued transformation.<sup>19</sup> However, note that as the number of potential inputs grows, the size of even a moderate degree multinomial containing all the inputs grows exponentially. Fortunately, the dimensionality of

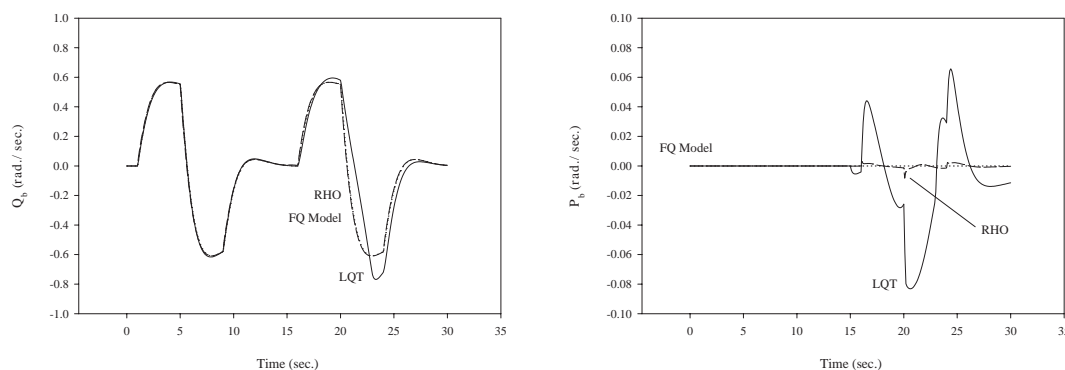
the true problem is frequently much smaller, and *GNOSIS* uses structure-learning techniques which result in reduced order multinomials to represent the system. This is accomplished by growing the model in "chunks", using a small number of inputs in each chunk, and utilizing an information-theoretic approach to select useful input variables and reduce the likelihood of overfit.

### 4. Inner-Loop Preliminary Results

Of special interest is RHO control response under control effector failures. Figure 2 compares a point-design (trim) optimal controller (denoted LQT) with the indirect-adaptive control system. The two controllers are designed using identical cost-function weighting matrices and are identical for a linear time-invariant (LTI) system without failures. For the current UAV application, performance of the two systems prior to a failure is qualitatively similar, despite stability and control derivatives that vary with flight condition, nonlinear inertial cross-couplings, and orientation-dependent gravitational effects. Following a failure, however, there are pronounced differences between the two systems. The maneuver of Fig. 2 is a pair of back-to-back pitch-rate doublets. Here, the left elevon and the left aileron become completely ineffective 15 sec. into the maneuver. Note that for this preliminary experiment MSLS was not used, and the controllers were each supplied full knowledge of the failures (i.e. the "truth" model was supplied). In Fig. 2, the pitch-rate response of the RHO control law follows the reference model (denoted FQ model) closely, and there is little discernable difference between the pre-impairment and post-impairment flying qualities. In the case of the point-design linear-quadratic controller, post-impairment tracking of the desired pitch-rate is compromised.

Prior to the failure, both the RHO controller and the point-design LQ controller de-couple the longitudinal and lateral/directional motions appropriately; there are no roll-rate excursions prior to 15 sec. During the execution of the second pitch doublet, the uncommanded roll rate associated with the RHO control system is minimal. Contrariwise, the post-impairment pitch rate and roll rate are more strongly coupled for the point-design LQ controller. Here, the maximum uncommanded roll-rate is (approximately) an order of magnitude more severe than that of the RHO controller.

These results demonstrate the utility of the on-



**Fig. 2: RHO vs. LQ Pitch and Roll Rate Responses with Ineffective Left Elevon and Left Aileron**

line time-varying gain computations performed by RHO when compared to fixed or gain scheduled controllers. Again, it was assumed here that the controller had knowledge regarding the actual values of the stability and control derivatives before and after impairment.

The next step in this experiment was to incorporate the MSLS parameter identification. The parameter identification module can easily estimate the parameter value when it is close to the *a priori* estimate. The real challenge is how easily will the identification algorithm *ignore* the spatial constraint when a failure occurs. To test this, several single and multiple effector failures were evaluated, and representative estimates of key control derivatives are provided in Fig. 3.

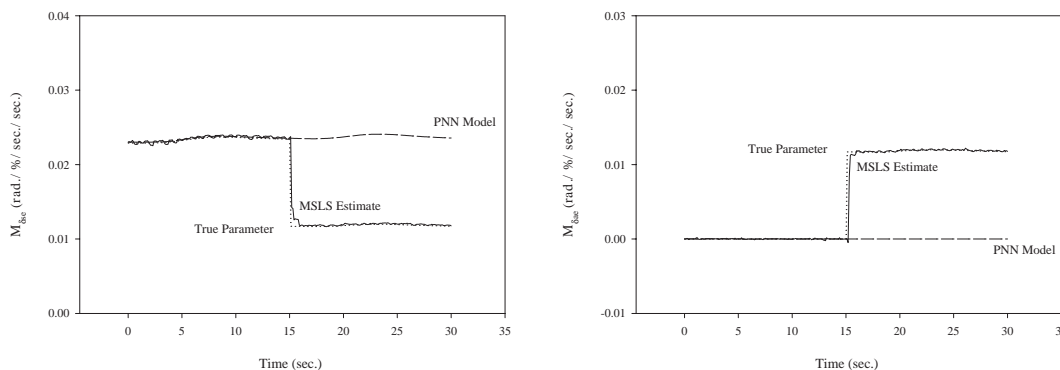
The parameters in Fig. 3 are the contributions of symmetric and asymmetric elevon, respectively, to pitching moment for the back-to-back pitch doublet maneuver. Again, control surface failure occurs 15 sec. into the maneuver. Prior to the failure, there is good agreement between the true parameter, the MSLS estimate, and the *a priori* PNN spatial constraint in each case. Following the failure, the identification algorithm is able to ignore the (now erroneous) PNN spatial constraints and track the true parameters closely. Furthermore, note that no “active noise injection” or manual excitation of the system is required for this identification method, as is often necessary for other approaches. Finally, note that real-time versions of the MSLS algorithm exist and have been used successfully in flight tests.<sup>10</sup>

In summary, BAI has implemented on-line parameter identification logic that is suitable for reconfiguration of the UAV in the presence of arbitrary sin-

gle and multiple failures. The chosen identification algorithm is a regularized least-squares technique developed by the authors. Temporal constraints are used to discourage sample-to-sample parameter variations, and PNN models of nominal (unimpaired) stability derivatives, control derivatives, and equation-of-motion biases are used as spatial constraints in MSLS. By continuously identifying the dynamics of the UAV via a regularized on-line system identification algorithm, the integrated controller is able to provide consistent inner-loop responses regardless of UAV health. Thus, by providing inner-loop reconfiguration, the outer-loop strategies remain near-optimal, even in the presence of unforeseen failures. Any suboptimality that results will depend on the degree to which the inner-loop controller can no longer reconfigure to achieve the desired responses. Even in suboptimal situations, it is expected that the outer-loop controller will usually perform well.

### Opt.-Path-to-Go (OPTG) Autopilot

The difficult shipboard UAV landing task required the development of a suitable outer-loop controller to generate the inner-loop commands. Toward this end, BAI developed an Optimal-Path-To-Go (OPTG) approach that is a natural extension of RHO control. A modern multivariable control architecture could have been developed that utilizes the effectors directly to achieve desired trajectories. However, it was decided to remain within a more traditional framework of inner/outer-loop control. The advantages of this framework include: 1. the inner-loop controller architecture can control and stabilize unstable modes directly, 2. separate inner- and



**Fig. 3: Symmetric Elevon and Asymmetric Elevon Pitching Moment Effectivenesses**

outer-loop controllers can take advantage of inherent frequency separation between the loops to provide more robust control responses, 3. components of the partitioned design can be more readily transferred into existing systems, and 4. it is easier to leverage decades of experience in the two domains of inner-loop control and outer-loop guidance.

The OPTG guidance law uses the calculus of variations to compute intercept trajectories that are optimal with respect to a particular cost function. For the shipboard landing task, it is important to achieve approaches in minimal time that do not require excessive UAV accelerations at any time during the trajectory. The latter is always a good idea, but becomes especially important when one designs a reconfigurable control system to operate under (significantly) reduced control authority. For the purpose of computing minimum-time trajectories with minimal accelerations, one can represent the desired UAV responses as a first-order system with respect to commanded accelerations, and the UAV accelerations can be minimized. It is noted that the desired *flying qualities* of the UAV are chosen to be first-order. No requirement of linear dynamics of the *actual* UAV system has been imposed.

For each Earth axis,  $i$ , one can define the UAV equations of motion as:

$$\begin{aligned}\dot{x}_i &= v_i \\ \dot{v}_i &= a_i \\ \dot{a}_i &= 1/\tau_i(u_i - a_i)\end{aligned}\quad (6)$$

where  $x_i$ ,  $v_i$ ,  $a_i$  are the positions, velocities, and accelerations along the  $i$  axis,  $\tau_i$  is the time constant

providing a rough approximation of the UAV dynamics, and  $u_i$  is the acceleration command.

The cost function may be written as:

$$\mathcal{J} = \int_{t_0}^{t_f} \left(1 + \frac{1}{2}u^T \mathcal{R}u\right) dt \quad (7)$$

Because the commands enter at the acceleration level, the cost function allows a direct trade-off (via adjustment of  $\mathcal{R}$ ) between minimum-time responses and minimum commanded accelerations. Again, note that this development is a special case of the system used by the RHO controller.

The landing task was comprised of two sequential steps. First, the UAV is guided to an artificial waypoint  $10ft.$  above and  $10ft.$  aft of the center of the helicopter landing pad. After the UAV is *sufficiently* close to the waypoint, the landing phase is initiated. During the landing, the UAV attempts to null the two-dimensional offset between its center of gravity and the center of the ship helicopter pad. Drastic motions are discouraged during the landing phase via increased actuation penalties in the performance index. During the development of the OPTG autopilot it was discovered that this two-step approach is necessary to achieve the landing task. Afterwards, the authors found that this is *precisely* how pilots accomplish shipboard helicopter landing tasks.

Once the optimal Earth-axes trajectory is computed, inner-loop UAV commands must be generated. This process involves two steps: 1. converting the North-East-Down trajectory to desired Euler-angle rates, and 2. converting the desired Euler-angle rates into body-axes angular rate commands.

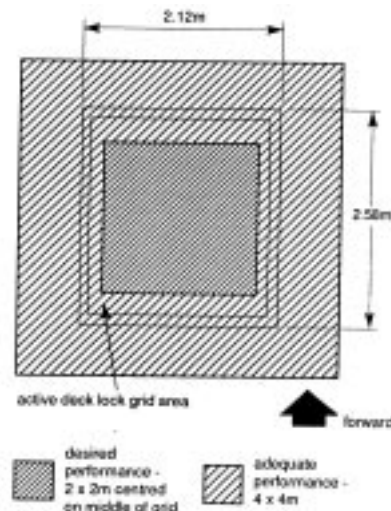
In summary, the desired Earth-axes responses are

realized by controlling the UAV airspeed directly (via the RHO architecture) and the inertial flight-path indirectly (via a sequence of transformations that yield equivalent body-axes angular rate commands). Each contributes to the UAV following the *optimal* inertial flight path closely. The efficacy of this approach is bolstered further by the extensive (closed-loop) results summarized in the next section.

### Simulation Results

The authors developed a library of simulation control files to evaluate the ability of the integrated reconfigurable control system to land a UAV on a ship's helicopter deck. The library is comprised of 9,450 combinations of UAV initial conditions (velocity, distance from ship and altitude), ship initial conditions, and sea states (ship velocity, orientation, wave direction, etc.). In excess of 18,500 trajectories were simulated to obtain statistically significant measures of the UAV's performance.<sup>20</sup> The experiments discussed herein were selected because they capture the essence of all the experiments and allow the closed-loop performance to be described concisely. The UAV landing task was evaluated for: 1. offset-correction landings with five unique initial offsets evenly distributed between  $\pm 300$  *ft.*, 2. initial UAV altitudes of 150 *ft.* and 300 *ft.*, 3. significant differential velocities between the UAV and ship, 4. wave incidences on the ship that vary from head seas to starboard beam seas, 5. wave intensities, 6. ballistic (steady) winds that vary in direction (from Northerly to Easterly) and magnitude, 7. calm, light, and moderate turbulence levels, 8. single and multiple effector impairments, (both missing as well as locked aerodynamic control surfaces), and 9. uncertainties in the aerodynamic data.

In each of the tables presented below, the first row gives the statistics of the root sum-of-squares (RSS) offset between the UAV center of gravity and the center of the helicopter pad at landing. Rows 2-4 present the difference between ship and UAV component velocities at landing, and row 5 summarizes the final RSS velocity difference (i.e., differential speed). Thus, the data in rows 2-3 provide a measure of the shear on the UAV landing gear, and the data in row 4 give a measure of the impact severity. The remaining rows of the table show statistics for the final body-axes angular rates of the UAV and the trial duration statistics. Finally, each table shows the percentage of simulated landings that achieve *desired* performance and the percentage of



**Fig. 4: Helicopter Pad Geometry and Landing Performance Measures**

simulated approaches that achieve *adequate* performance. Desired performance is defined by a final RSS offset that is less than 1 *m.* (3.28 *ft.*), and adequate performance is defined by a final RSS offset that is less than 2 *m.* (6.56 *ft.*) The measures of desired and acceptable performance are shown in Fig. 4.<sup>21</sup>

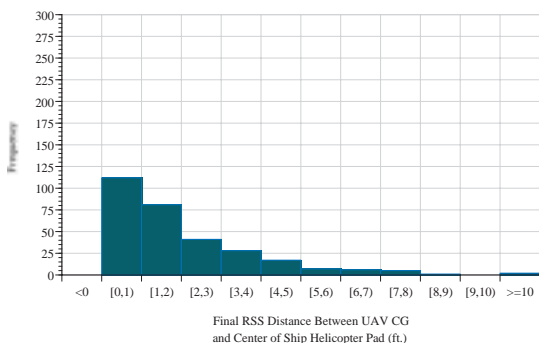
In the following discussion, the suffix *-IS* denotes that the experiment was conducted using the ideal control system, i.e., perfect knowledge of the UAV dynamics was available to the controller. Although this is never the case in reality, it provided a benchmark for system performance. The suffix *-NR* denotes that the experiment was conducted using the baseline non-reconfigurable controller, i.e., PNN approximations of the *nominal* (unfailed) UAV dynamics were available to the controller, and the suffix *-RE* denotes that the experiment was conducted with the reconfigurable control system, i.e., on-line parameter estimation techniques were invoked to identify the UAV dynamics and communicate the information to the control computations.

Table 1 and Fig. 5 show benchmark (ideal-system controller) results for one of the experiments which reflects a very challenging shipboard landing task, characterized by ballistic (steady) winds, moderate turbulence, uncertainties in the aerodynamic data, and multiple effector impairments. The effector impairments are defined as 50 % left elevon, 50 % left aileron, and 0 % rudder.

Here, the closed-loop performance is quite acceptable with mean and standard deviation RSS offset

**Table 1: Summary Statistics of the Ideal System for Experiment XV-IS**

	<u>Winds</u> Commensurate with Sea State <u>Turbulence</u> Dryden (Moderate)	<u>Aerodynamic Uncertainties</u> Mean = 0% Standard Deviation = 33%	<u>Effector Impairments</u> 50% Left Elevon 50% Left Aileron 0% Rudder
Summary Statistics (N = 300 Cases)			
	Mean	Standard Deviation	Minimum      Maximum
$\Delta X_{RSS} (ft.)$	1.9865	1.9149	0.0865      16.2825
$\Delta V_{\dot{x}_i} (ft./sec.)$	-0.0610	1.6983	-8.0383      20.2724
$\Delta V_{\dot{y}_i} (ft./sec.)$	-0.2482	1.3804	-5.5083      6.4933
$\Delta V_{\dot{z}_i} (ft./sec.)$	-1.0975	1.4527	-9.0042      0.0211
$\Delta V_{\dot{RSS}} (ft./sec.)$	1.8618	2.1670	0.0173      21.0616
$p_i (rad./sec.)$	-0.0009	0.0092	-0.0299      0.0281
$q_i (rad./sec.)$	0.0004	0.0016	-0.0052      0.0061
$r_i (rad./sec.)$	0.0003	0.0088	-0.0318      0.0358
$T_i (sec.)$	143.4911	24.1672	105.6700      360.0000
	$\Delta X_{RSS} < 1 \text{ meter}$	82.00%	
	$\Delta X_{RSS} < 2 \text{ meters}$	97.00%	



**Fig. 5: Final RSS Offset between UAV CG and Center of Helicopter Pad for Experiment XV-IS**

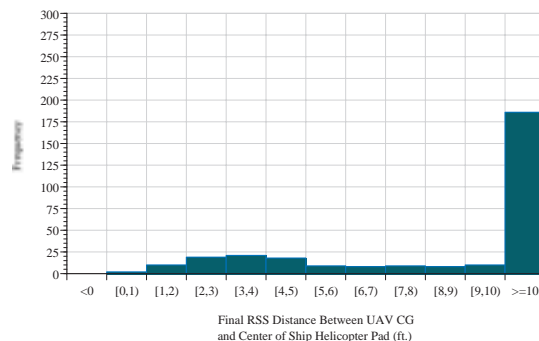
distances significantly less than 3 ft. Despite the simultaneous partial or complete failure of three aerodynamic control surfaces compounded by winds and moderate turbulence, the final body-axes angular rates are all well within  $\pm 3.0 \text{ deg./sec.}$ , and the final body-axes pitch rate is significantly less.

Table 2 and Fig. 6 summarize the corresponding results for the *non-reconfigurable* controller. Here, the baseline controller has no knowledge of the effector impairments.

The mean and standard deviation of the final RSS offset are 847 ft. and 2143 ft., respectively. The maximum RSS offset is approximately 7600 ft. The

**Table 2: Summary Statistics of the Non-Reconfigurable Control System for Experiment XV-NR**

	<u>Winds</u> Commensurate with Sea State <u>Turbulence</u> Dryden (Moderate)	<u>Aerodynamic Uncertainties</u> Mean = 0% Standard Deviation = 33%	<u>Effector Impairments</u> 50% Left Elevon 50% Left Aileron 0% Rudder
Summary Statistics (N = 300 Cases)			
	Mean	Standard Deviation	Minimum      Maximum
$\Delta X_{RSS} (ft.)$	846.5997	2143.2947	0.2108      7597.5800
$\Delta V_{\dot{x}_i} (ft./sec.)$	1.7082	20.9057	-87.9476      120.2930
$\Delta V_{\dot{y}_i} (ft./sec.)$	0.3814	25.8165	-99.0724      118.6430
$\Delta V_{\dot{z}_i} (ft./sec.)$	-9.8020	12.2606	-68.1739      0.2996
$\Delta V_{\dot{RSS}} (ft./sec.)$	23.6865	28.1142	0.0820      134.2500
$p_i (rad./sec.)$	0.1352	0.9836	-3.1514      3.3514
$q_i (rad./sec.)$	0.0747	0.6608	-1.8947      3.2044
$r_i (rad./sec.)$	-0.0909	0.5647	-2.7602      1.8467
$T_i (sec.)$	117.9109	40.1004	21.0700      239.2200
	$\Delta X_{RSS} < 1 \text{ meter}$	12.67%	
	$\Delta X_{RSS} < 2 \text{ meters}$	28.33%	

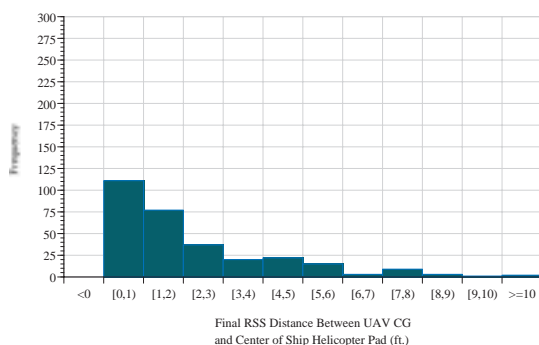


**Fig. 6: Final RSS Offset between UAV CG and Center of Helicopter Pad for Experiment XV-NR**

final body-axes angular rates approach or exceed  $180 \text{ deg./sec.}$  on multiple occasions. In short the UAV is uncontrollable for a significant fraction of the simulated trajectories and is not able to accomplish the landing task consistently. A meager 12.67 % of the cases satisfy the desired performance measure, and only 28.33 % of the cases achieve adequate performance. The non-reconfigurable system is not acceptable because it might damage the ship or injure the ship's crew during a landing attempt. Other experiments (not presented here) showed that the non-reconfigurable controller is robust to slight

**Table 3: Summary Statistics of the Reconfigurable Control System for Experiment XV-RE**

	<i>Winds</i> Commensurate with Sea State <i>Turbulence</i> Dryden (Moderate)	<i>Aerodynamic Uncertainties</i> Mean = 0% Standard Deviation = 33%	<i>Effector Impairments</i> 50% Left Elevon 50% Left Aileron 0% Rudder
Summary Statistics (N = 300 Cases)			
	Mean	Standard Deviation	Minimum
$\Delta X_{RSS}$ (ft.)	2.2096	2.0893	0.1528
$\Delta V_{x_i}$ (ft./sec.)	-0.0014	1.9842	-13.4991
$\Delta V_{y_i}$ (ft./sec.)	-0.2783	1.4627	-5.4859
$\Delta V_{z_i}$ (ft./sec.)	-1.4441	1.8232	-8.7698
$\Delta V_{RSS}$ (ft./sec.)	2.3257	2.4786	0.0446
$p_i$ (rad./sec.)	0.0004	0.0106	-0.0266
$q_i$ (rad./sec.)	0.0001	0.0018	-0.0117
$r_i$ (rad./sec.)	0.0000	0.0094	-0.0361
$T_i$ (sec.)	141.1600	17.3012	103.1700
	$\Delta X_{RSS} < 1 \text{ meter}$	77.00%	
	$\Delta X_{RSS} < 2 \text{ meters}$	94.33%	

**Fig. 7: Final RSS Offset between UAV CG and Center of Helicopter Pad for Experiment XV-RE**

modeling errors owing to imperfect PNN estimates, moderate levels of aerodynamic uncertainties, and non-drastic effector impairments. However, when more severe impairment conditions are coupled with minimal uncertainty in the aerodynamic data, the *non-reconfigurable* controller is not robust and is not able to maintain UAV stability.

The final experiment to be discussed is that of using the reconfigurable controller for the corresponding landing task, and Table 3 and Fig. 7 summarize the results for this case.

The distribution of RSS offsets at landing is comparable (slightly inferior) to that of the *IS* controller.

The mean and standard deviation of the RSS offset distance increase by 0.23 *ft.* and 0.17 *ft.* for the most demanding task. Note that the body-axes angular rates at landing are all within  $\pm 2.5 \text{ deg./sec.}$  and are similar to the angular rates for the *IS* controller of Experiment XV-*IS*.

Although PNN models of the *nominal* UAV dynamics are used as spatial constraints in the MSLS parameter identification algorithm, the RSS offsets at landing for Experiment XV-*RE* are noticeably improved over the control system architecture that relies entirely on the same neural networks to calculate and issue the control commands (Experiment XV-*NR*). The mean, standard deviation, and maximum RSS offsets are orders of magnitude less than those of the non-reconfigurable controller. It is apparent that the MSLS algorithm is identifying the anomalous operating conditions, and the inner-loop controller is correcting for the off-nominal behaviors. Recall, it was asserted that the PNN spatial constraints in the MSLS algorithm are ignored when the estimates are erroneous because of unforeseen damage events. The simulation results above support this assertion.

## Conclusions and Recommendations

A complete nonlinear simulation, an adaptive reconfigurable inner-loop controller (including neural networks and on-line parameter identification), and an optimum-path-to-go autopilot were developed for autonomous UAV landing on a destroyer helicopter deck. The adaptive intelligent controller uses redundant moment-generating capabilities of the unfailed effectors to improve performance significantly. Even in cases of multiple failures, severe turbulence, high seas, and uncertainties in the aerodynamic data, the intelligent controller is consistently able to land the UAV on the helicopter deck with reasonably small miss distances (offset between UAV center of gravity and center of helicopter landing pad) and impact velocities. The performance of a conventional fixed-structure (non-adaptive) or gain-scheduled controller degrades significantly in the presence of failures, uncertainties, and turbulence. Finally, the modular inner-loop and outer-loop control architecture can be readily incorporated into new applications.

Navy UAV programs will require highly innovative designs with reliable control algorithms in the presence of uncertainties, failures, and large disturbances. UAV development costs must be

significantly less than those associated with manned aircraft. This requirement dictates: 1. improved development tools to reduce labor expense, 2. control algorithms that can be designed using less reliable aerodynamic data, and 3. control algorithms that can reconfigure on-line in the event of effector failure to reduce the costs associated with actuator redundancy management. The work described herein serves as a foundation to address these challenges. The need exists to perform further development and testing on the complete UAV control system. Specifically, future research should: 1. address control reconfiguration under sensor failures, 2. evaluate the control system on an actual UAV investigating effects of structural modes, sensor dynamics, and computational time delays, and 3. investigate the utility of feeding failure information to the OPTG autopilot for on-line re-computation of optimal strategies.

### Acknowledgments

This work was funded by the Naval Surface Warfare Center (NSWC), Contract N00178-98-C-1009. Tech. Monitor: John Bibel, NSWC Dahlgren Division; Topic Sponsor: Dr. Allen Moshfegh, Office of Naval Research.

### References

- <sup>1</sup>D. Ward, J. Monaco, R. Barron, and R. Bird, "Self-Designing Controller: Design, Simulation and Flight Test Evaluation," Final Report: WL-TR-97-3095, Barron Associates, Inc., Wright-Patterson AFB, OH, Nov. 1996.
- <sup>2</sup>C. Garcia and D. Pretz, "Model predictive control: Theory and practice - a survey," *Automatica*, vol. 25, pp. 335-348, 1989.
- <sup>3</sup>H. Michalska and D. Mayne, "Robust receding horizon control of constrained nonlinear systems," vol. 38, pp. 1623-1633, Nov. 1993.
- <sup>4</sup>F. Hoblit, *Gust Loads on Aircraft: Concepts and Applications*. Washington, DC: American Institute of Aeronautics and Astronautics, 1988.
- <sup>5</sup>T. Smith and C. Bennett, "User's Manual for the Ship Motions Prediction Applications Manager - PREDICT," Final Technical Rept. CRDKNSWC/HD-0936-06, Carderock Division Naval Surface Warfare Center, New London, CT, Feb. 1994.
- <sup>6</sup>R. Bitmead, M. Gevers, and V. Wertz, *Adaptive Optimal Control*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1990.
- <sup>7</sup>B. Anderson and J. Moore, *Optimal Control: Linear Quadratic Methods*. Englewood Cliffs, NJ: Prentice Hall International, 1989.
- <sup>8</sup>P. Chandler, M. Pachter, and M. Mears, "System identification for adaptive and reconfigurable control," *Guidance, Control, and Dynamics*, vol. 18, pp. 516-24, 1995.
- <sup>9</sup>D. Ward, J. Monaco, and M. Bodson, "Development and flight testing of a parameter identification algorithm for reconfigurable control," *Guidance, Control, and Dynamics*, vol. 21, pp. 948-956, 1998.
- <sup>10</sup>D. Ward, J. Monaco, R. Barron, R. Bird, J. Virnig, T. Landers, "Self-Designing Controller: Design, Simulation, and Flight Test Evaluation," Final Technical Rept. for AFOSR, Contract F49620-94-C-0087, Barron Associates, Inc., Nov. 1996.
- <sup>11</sup>R. Murray, J. Doyle, J. Marsden, S. Wiggins, G. Balas, and B. Morton, "Robust nonlinear control theory with applications to aerospace vehicles." Summary Progress Report for AFOSR-PRET Grant F49620-95-1-0419, Mar. 1997.
- <sup>12</sup>T. Söderström and P. Stoica, *System Identification*. Englewood Cliffs, NJ: Prentice Hall, 1989.
- <sup>13</sup>V. Cherkassky, J. Friedman, and H. Wechsler, *From Statistics to Neural Networks, Theory and Pattern Recognition Applications*. Berlin: Springer-Verlag, 1993.
- <sup>14</sup>S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*. Philadelphia: SIAM, 1991.
- <sup>15</sup>L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1983.
- <sup>16</sup>M. Bodson, "An adaptive algorithm with information-dependent data forgetting," in *Proc. 1995 American Control Conf.*, (Seattle), June 1995.
- <sup>17</sup>D. Ward, "Generalized networks for complex function modeling," in *IEEE Trans. on Systems, Man, & Cybernetics*, vol. SMC-94, Oct. 1994.
- <sup>18</sup>R. Barron, A. Mucciardi, F. Cook, J. Craig, and A. Barron, *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, S.J. Farlow (Ed.), ch. Adaptive learning networks: Development and application in the United States of algorithms related to GMDH, pp. 25-65. New York: Marcel Dekker, Inc., 1984.
- <sup>19</sup>D. Gabor, "Communication theory and cybernetics," in *Trans. of IRE*, vol. CT-1, p. 19, 1954.
- <sup>20</sup>D. Ward, J. Monaco, R. Barron, and L. Funderburg, "Intelligent Reconfigurable Control for Systems with Multiple Effectors," Contract N00178-98-C-1009, Barron Associates, Inc., Dahlgren, VA, May 1998.
- <sup>21</sup>G. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. Washington, DC: American Institute of Aeronautics and Astronautics with Blackwell Science, Ltd., 1996.