Application of an Optimization-based Design Process for Robust Autoland Control Laws

Gertjan Looye[*], Hans-Dieter Joos[†]and Dehlia Willemsen[‡]

German Aerospace Center
DLR-Oberpfaffenhofen, Institute for Robotics and Mechatronics
D-82234 Wessling, Germany
E-mails: {gertjan.looye, dieter.joos, dehlia.willemsen}@dlr.de

May 7, 2001

Abstract The design of a robust autoland controller using an optimization based design process is described. A modular controller architecture is developed first. Inner loops are based on Dynamic Inversion, speed and longitudinal path tracking loops are based on the Total Energy Control System. Functions for lateral path tracking, ILS guidance, flare and runway alignment are based on classical PID structures. The free parameters in the controller functions are tuned using multi-objective optimization. Performance criteria are directly derived from the imposed design requirements and computed from linear or nonlinear analysis (e.g. eigenvalues, simulations). Robustness is addressed via a multi-model approach, via optimization criteria (e.g. gain/phase margins), and, as a new contribution, via statistical criteria computed from on-line Monte-Carlo analysis. The design process and selected architecture have been applied to a wide-body as well as a small passenger aircraft to demonstrate controller robustness and efficiency of the process. In both cases, JAR-AWO specifications had to be met. The latter design was successfully flight tested.

1. Introduction

The development of flight control laws, especially for automatic landing, is a demanding task, since high safety standards have to be met before operational use under Cat IIIb/c conditions is allowed. Control law robustness to variations in atmospheric, airport, equipment, aircraft loading parameters etc. has to be proven via extensive Monte-Carlo analysis and flight tests. The current development process is to a large extent manual and involves considerable trial-and-error to finally meet the certification requirements. Neglecting tolerances in model parameters cause additional design cycles for each model update. Especially in the implementation phase this may become very costly.

For this reason the EU funded a project aiming at reducing development costs by improving efficiency of the design process as well as reducing the number of design cycles by improving controller robustness. This project was called *Robust and Efficient Autoland control Laws design* (REAL). Industry and research institutes from France, Germany and the Netherlands participated in this project. Two design teams were asked to propose a design process that fulfils the following criteria:

[*] Research Engineer, PhD candidate

[†] Research Engineer, PhD

[‡] Research Engineer, DLR-Braunschweig, Institute for flight system dynamics

- systematic incorporation of design requirements in the design,
- explicit consideration of uncertainties and parameter variations from the start,
- the resulting controller structure should be visible and physically understandable,
- the process should allow for automation to a maximum extent.

Each design process was sequentially applied to two benchmark problems based on respectively a model called RealCAM (REAL Civil Aircraft Model), representing a large passenger aircraft, and the DLR experiment aircraft ATTAS (Advanced Technologies Testing Aircraft System). The mission involves an ILS approach and flare until touch down of the main wheels. The design specifications were mainly based on JAR-AWO[4] certification requirements. Design activities had to be documented in detail, allowing industrial partners to assess the design process criteria. The designs for the transport aircraft were extensively assessed for robustness, the designs for ATTAS were flight tested[2]. The ATTAS designs had to be performed in a short time frame to prove design process efficiency.

The design process proposed by DLR is based on multi-objective optimization of free parameters in a pre-specified controller structure. The process is and iterative approach involving modeling, analysis and sub-task selection, tuning and compromising via multi-objective optimization, and closed-loop assessment[7]. Controller architecture definition is based on the designer's experience and may be an existing controller structure. Robustness is addressed via a multi-model approach, via optimization criteria (e.g. gain and phase margins), and, as a new contribution, statistical criteria from on-line Monte-Carlo analysis. The latter implies that in each iteration step a large number of landing simulations is performed with randomly selected input parameters. Statistical criteria are derived that are used as optimization criteria.

The process is largely automated by using appropriate software tools for modeling, simulation, optimization, and data handling that interact via direct data transfer and automatic code generation mechanisms.

In this paper the design process is introduced in section 2. Then the steps are described as applied to the autoland controller design for the RealCAM benchmark. In section 3 the aircraft model and the adopted control law structure are described. In section 4 and 5, the strategy for tuning the over-all structure via optimization sub-tasks is explained. The optimization of the flare mode is then described in detail. After a discussion of assessment results in section 7, conclusions are drawn in section 8.

2. Design process

The process structure[7] that was used for the design of the autoland control laws is depicted in Figure 1. The execution of the principal steps (to the left in the block diagram) and the supporting software tools (to the right) will be explained in the following sections, as they are performed for the RealCAM benchmark design. Controller synthesis involves two independent steps: controller architecture selection (a *Modeling* activity) and controller parameter tuning. The key variables in the process are the free controller parameters T (Tuners) and the variable model parameters p that have to be coped with. The parameters represent aircraft configurations, environment conditions, and uncertainties in the model. MOPS (Multi-Objective Parameter Synthesis[6]) is an advanced optimization tool. It forms the core of the Matlab-based MOPS-environment, which additionally provides routines for problem-definition, on-line visualization, data handling, and Graphical User Interfaces (GUIs). The process and supporting software tools have been successfully used in many aerospace and robotics applications (see Ref.[6] for a list of references).

3. Modeling

3.1 Plant model development

For analysis and design computations, parameterized models of the aircraft dynamics and environment (e.g. atmosphere, airport) are required. As

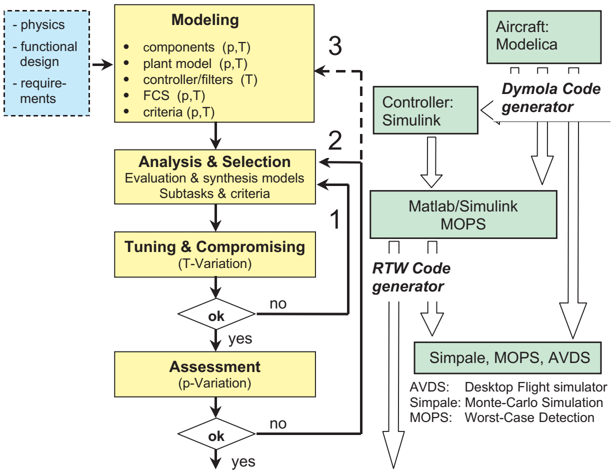

Figure 1: Design process with supporting software tools

a modeling platform, the object-oriented modeling software Modelica/Dymola[3] is used, in which a flight-mechanics class library[14] is available. The aircraft model is depicted in Fig. 2. The model components are implemented one-to-one as software objects. The underlying modeling language, Modelica[13] is equation-based. The depicted interconnections represent kinematic constraints and energy flows. Within Dymola, the model is composed and processed symbolically. Simulation code can be generated automatically for several engineering environments. For the two benchmark problems, the RealCAM and the ATTAS models (which will not be described here) were implemented and simulation code was generated for use in Matlab/Simulink^{TM}.

The available inputs for the controller are aileron, elevator, rudder, and throttles ($\delta_A, \delta_E, \delta_R, \delta_{T1,2}$). The available sensors are airspeed (calibrated V_{cas} and true V_{tas}), ground speed V_g, angular rates p, q, r, attitude angles ϕ, θ, ψ, load factors n_x, n_y, n_z, track angle χ, flight path angle γ, vertical speed V_Z, angle of attach α, localizer and glide slope signals ϵ_{LOC}, ϵ_{GS}, radio altitude H_{ra}, barometric altitude H_{baro}, aircraft mass m, and the centre of gravity location x_{cg}. For ATTAS also the mean engine fan shaft speed N_1 is available to enable backlash compensation.

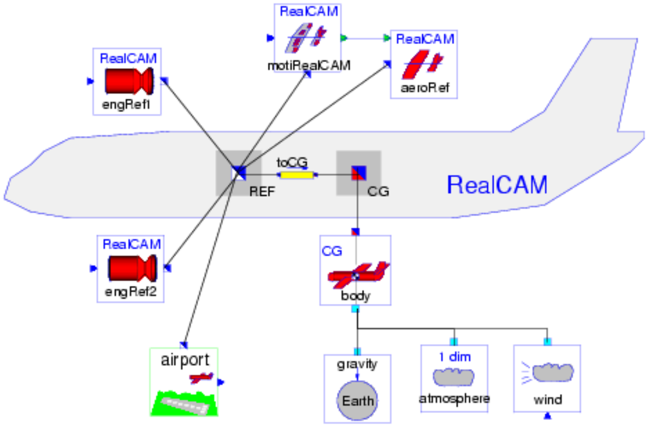

Figure 2: Aircraft model in Dymola

3.2 Controller architecture selection

Controller synthesis consists of architecture selection and controller parameter tuning. In principle, these steps are independent. The controller is selected by the designer and therefore open to incorporate available knowledge and experience. The controller architecture is specified in Matlab/Simulink. The Simulink add-on Real-Time Workshop^{TM} (RTW) is used for implementation of the controller in the assessment tools (section 7) as well as the ATTAS ground based simulator and ATTAS itself[2] (second benchmark).

The selected structure for the autoland controller is depicted in Figure 3. Three main loops can be identified, separated by the vertical dashed lines: stability and command augmentation, path tracking, and guidance. The flare and align modes engage shortly before touch down in order to respectively reduce the sink rate to an acceptable level, and to align the aircraft with the runway centre line in case of cross wind (decrab).

Inner loops
The task of the inner loops is to improve stability and to achieve robust tracking of inner loop com-

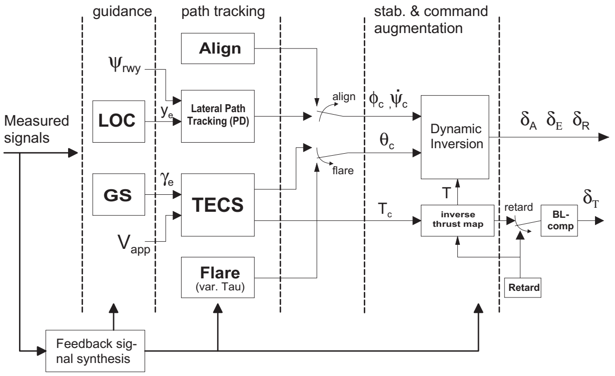

Figure 3: Autoland controller architecture (Note, for block inputs, only command or error signals shown)

mand variables (ϕ_c, θ_c, $\dot{\psi}_c$). Roll angle and heading rate commands are co-ordinated when used for lateral flight path control (during approach), whereas $\dot{\psi}_c$ alone allows for direct heading control during the align phase. The inner loops were designed with Dynamic Inversion. Inverse model equation compensate the nonlinear aircraft dynamics, resulting in decoupled command responses without the need for additional gain scheduling. The design of the inner loops is discussed in detail in Ref.[12].

Longitudinal tracking and glide slope mode

For longitudinal flight path and speed tracking during the approach, the Total Energy Control System (TECS[10]) is used, see Figure 4. The TECS-architecture offers pilot-like decoupled tracking of speed and flight path angle commands. The input signals are aerodynamic flight path angle $\gamma_a \approx -V_Z/V_{cas}$ and acceleration $\dot{V}/g$. The subscript e denotes an error signal from a commanded value. TECS controls pitch attitude $\delta\theta_c$ and thrust (per unit weight, T_c/W). An inverse thrust map is used to generate appropriate throttle commands (Figure 3). The integral gains K_{TI} and K_{EI} usually have magnitude of 0.4. The proportional gains K_{TP} and K_{EP} are typically 1. The additional gain K_F allows for shifting control priority to flight path tracking, which is preferred during the approach. The speed loop is opened in case thrust saturates. A detailed description of TECS can be found in Ref.[10]. In our design, the gains are considered as free parameters for fine-tuning. The feedback signal $\hat{\dot{V}}/g$ is obtained from a complementary filter[11]:

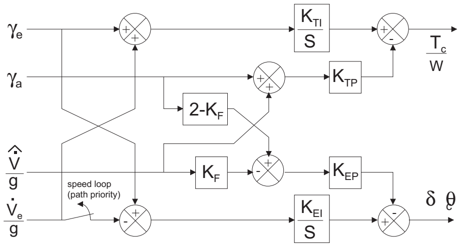

Figure 4: TECS controller structure

$$\hat{\dot{V}} = \frac{s}{\tau_V s + 1} V_{cas} + \frac{\tau_V s}{\tau_V s + 1} \dot{V} \qquad (1)$$

where V_{cas} is the calibrated airspeed, and $\dot{V}$ is time derivative of the inertial speed that is computed from Inertial Reference System signals. A good compromise between speed tracking in wind shear and throttle activity in turbulence is obtained for τ_V=10 s. The acceleration command $\dot{V}_c/g$ is computed from proportional feedback of the calibrated airspeed. In connection with the glide slope mode, the flightpath angle error is obtained from[11]:

$$\gamma_e = \frac{1}{\hat{V}_{cas}} \left[k_h \frac{1}{\tau_h s + 1} \Delta\tilde{h} - \hat{\dot{h}} \right] \qquad (2)$$

where $\hat{V}_{cas}$ is a complementary signal obtained using the filter in eq. 1, $\Delta\tilde{h}$ the height error estimated from the glide slope signal. This value is filtered with time constant τ_H in order to remove high frequency noise. $\hat{\dot{h}}$ is the estimated vertical speed with respect to the glide slope, obtained using a complementary filter:

$$\hat{\dot{h}} = \frac{s}{\tau_{\dot{h}} s + 1} \Delta\tilde{h} + \frac{\tau_{\dot{h}} s}{\tau_{\dot{h}} s + 1} (V_g \sin(-3\pi/180) + V_Z)$$

The filter results in an unbiased estimate of the vertical speed error, even when the reference glide angle is not exactly 3 deg, and propagation of high-frequency noise from radio signal is prevented. The variables k_h, τ_h and $\tau_{\dot{h}}$ are tuning parameters.

Lateral tracking and localizer mode

For lateral path tracking, co-ordinated roll angle and heading rate commands are used:

$$\phi_c = k_{\dot{y}} \dot{y}_e, \qquad \dot{\psi}_c = \frac{g}{\hat{V}_{cas}} \phi_c$$

For ATTAS additional PI feedback of the lateral load factor n_y was added in order to achieve unbiased tracking of the localizer in case of lateral wind shear. The lateral speed with respect to the localizer beam is obtained in exactly the same fashion as γ_e in the longitudinal case (eq. 2):

$$\dot{y}_e = k_y \frac{1}{\tau_y s + 1} \Delta\tilde{y} - \hat{\dot{y}} \tag{3}$$

where $\Delta\tilde{y}$ is the lateral position error estimated from the localizer signal, and $\hat{\dot{y}}$ is the estimated lateral speed with respect to the localizer beam:

$$\hat{\dot{y}} = \frac{s}{\tau_{\dot{y}} s + 1} \Delta\tilde{y} + \frac{\tau_{\dot{y}} s}{\tau_{\dot{y}} s + 1} V_g \sin(\chi - \psi_{rwy})$$

where the runway heading ψ_{rwy} is entered by the pilot, but is only accurate to 1 deg. The filter results in an unbiased estimate of the lateral speed error, and propagation of noise from radio signal is limited to lower frequencies. The variables k_y, $k_{\dot{y}}$, τ_y and $\tau_{\dot{y}}$ are tuning parameters.

Flare law

For the flare law, the so-called variable Tau principle[9] was chosen (Fig. 5). It is initiated at a fixed radio altitude (H_{flare}) and results in low dispersions of the runway touch down point. To the generated pitch angle command a feed forward part is added, computed from the trimmed attitude at flare start. The latter value is estimated by low-pass filtering the pitch command from TECS during the approach, and holding the value from flare initiation (Fig. 5). The feed forward command consists of a constant part (via gain K_{FW}) and a rising part via a rate limiter and the gain K_{Rate}.

The vertical speed $\hat{\dot{h}}$ is obtained by complementary filtering the radio altitude H_{ra} and V_Z, resulting in a runway referenced signal with low noise content. From the difference $\hat{\dot{h}} + V_Z$ an additional feed forward command is generated via K_{RW}, anticipating a possible runway slope. During flare the thrust is retarded at a constant rate $\dot{\delta}_{T_c}$, such that throttles reach idle position at touchdown. The retard rate is therefore proportional to the ground speed.

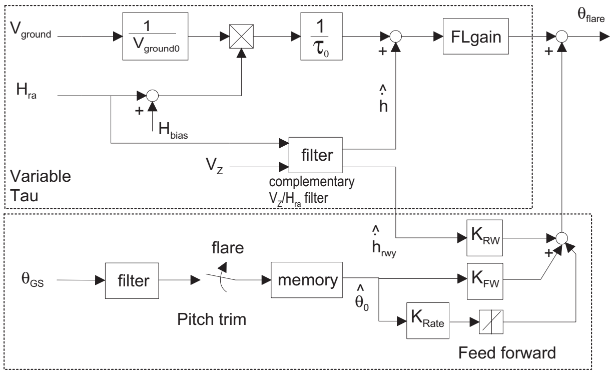

Figure 5: Flare architecture

Align mode

Laterally, a classical align mode takes over from the lateral path tracking mode in order to align the aircraft with the runway centre line in case of cross wind, while keeping lateral deviation small. The lateral position is controlled via roll angle commands, while the aircraft heading is steered via heading rate commands:

$$\begin{aligned} \phi_c &= k_\chi(\hat{\psi}_{rwy} - \chi) + k_{fw}(\hat{\psi}_{rwy} - \psi_i) \\ \dot{\psi}_c &= k_\psi(\hat{\psi}_{rwy} - \psi) \end{aligned}$$

The actual runway heading $\hat{\psi}_{rwy}$ is estimated from a moving average of the track angle during the approach. Note that lateral position control is only via track angle feedback. In addition, a feed forward roll command is generated from the difference between the runway heading and the aircraft heading at initiation of align, ψ_i.

3.3 Criteria modeling

For optimization, design requirements have to be formulated in a quantitative way, so that they can be computed from nonlinear or linear closed-loop analyses (e.g. simulation responses, or eigenvalues). Criteria computation is done in Matlab macros. The provided criteria in the two benchmark problems were mostly in numerical form (see Ref.[2]), so that this modeling task was relatively easy. Criteria are modeled such, that a smaller value indicates better performance. The computed criteria are of different order of magnitude and have different levels of importance in the optimization. For this reason, scaling is applied. This can be done via divisions by demand values, or via so-called good-bad values[5]. By dividing a criterion by its demand value, values less than 1 indicate satisfactory solutions, whereas values greater than 1 (criterion > demand) are considered as unsatisfactory. The scaling by good-bad values is illustrated in Fig. 6. The demand is that the landing distance from the runway threshold is between 300 and 500 m. Any value between 350 and 450 m is considered equally good. In that case the criterion is scaled to zero. Below 350 and above 450 m, the scaled value increases linearly, such that a value of 1 is reached at 300 and 500 m respectively. Any value below 300 and above 500 m is considered bad. In the example the current landing distance is 257 m (obtained from a nonlinear simulation), resulting in a (bad) criterion value of 1.86. The optimization of the flare mode and involved criteria will be described in more detail in section 6. Tuning of the inner loops is described in Ref.[12].

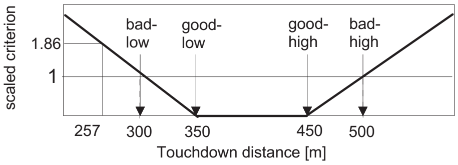

Figure 6: Criteria scaling with good-bad values

4. Analysis and sub-task definition

The first objective of this step is to analyze the open-loop aircraft dynamics for different flight conditions and parameter sets. The analysis results are used to make a proper selection of design models for the multi-model set that is to be considered during optimization[8].

Sub-task definition involves the break-down of the over-all controller optimization task into optimization sub-tasks. Especially in case of complex systems, this is necessary in order to keep the task tractable. In this application, the sub-tasks follow directly from the adopted modular controller structure (Fig. 3): for each component an optimization sub-task is defined to tune its free parameters. For inner and tracking loop optimization, criteria such as rise time, overshoot, damping, gain and phase margins are used. The criteria for the guidance, flare, and align modes are based on JAR-AWO[4] requirements.

In principle, the controller components can be tuned individually. However, performance can be considerably improved by taking interactions between the components into account. The MOPS environment allows design sub-tasks to be augmented into a single optimization task. In this way it is possible to optimize several components simultaneously. Table 1 illustrates the procedure that was used to tune the longitudinal loops. In first instance, the longitudinal parameters in the Dynamic Inversion controller are optimized. Next, the sub-task of the longitudinal tracking controller (basically TECS) is augmented. During optimization of this component, the inner loop (DI) parameters may be adjusted if required. In that case, the criteria in the inner loop sub-task are set as inequality constraints. In this way the effect of adjusting inner loop parameters on inner loop criteria can be controlled. This is useful in case the same inner loops are used for several autopilot modes. In the same fashion, the inner, tracking, and guidance design sub-tasks can be augmented. Table 2 lists all available sub-tasks, as well as the

	Design task:		
active tuning parameters:	inner loops	tracking controller	guidance controller
inner loops	✓	(✓)	(✓)
tracking controller		✓	(✓)
guidance controller			✓

Table 1: Sequential tuning strategy. The parameters of a sub-task with (✓) can be adjusted if necessary

type of analysis for criteria assessment.

name	contr. mode	analysis
DILON	inner loop longitudinal	linear (eigenvalues) nonlinear simulations
DILAT	inner loop lateral	linear (eigenvalues) nonlinear simulations
TECS	path tracking longitudinal	linear (eigenvalues) nonlinear simulations
LOC	localizer align	nonlinear simulations
GS	glide slope	nonlinear simulations
FLARE	flare	nonlinear simulations
MC	flare glide slope	Monte-Carlo analysis

Table 2: Available sub-tasks for optimization

5. Tuning and compromising

Tuning & compromising is based on a multi-criteria/multi-model parameter tuning facility which provides a systematic way for optimization based control law tuning by directly specifying bounds and demands on controller specifications and handling qualities as well as physical control implementation constraints[5]. Robustness to variations in model parameters and operating conditions is covered basically by the multi-model formulation. For a given controller structure, the free parameters of the controller, T, are automatically tuned by an optimizer to their best values satisfying the specifications and flying/handling quality demands. The multi-objective tuning of parameters is achieved by using the optimization tool MOPS (Multi-Objective Parameter Synthesis[6]). During optimization, intermediate results are visualized in so-called parallel co-ordinates and in user-defined graphs. This will be illustrated in section 6.

6. Optimization of the flare law

As an example, the parameter tuning of the flare law for the RealCAM with MOPS is discussed. The control structure of this mode is depicted in Fig. 5. The following parameters will be optimized: flare initiation height H_{flare}, throttle retard height H_{retard}, H_{bias}, FLgain, the feed forward parameters $\mathrm{K_{RW}}$, $\mathrm{K_{FW}}$, $\mathrm{K_{Rate}}$ and the rate limit. The flare time constant τ_0 is computed analytically from:

$$\tau_0 = (H_{flare} + H_{bias})/(V_{g0} * \tan(3 * \pi/180))$$

where V_{g0} is a mean ground speed, which is set to 65 m/s. In the undisturbed case, this results in transient-free switching between the glide slope and the flare mode.

Two classes of criteria are considered: deterministic and stochastic criteria. The deterministic criteria are listed in Table 3. Those are computed from a nonlinear landing simulation. The scaling of the criteria using good-bad values (section 3.3) is given in Table 4. In some cases (e.g. *B4*), the bad-low and good-low values are omitted and the good-high value is zero. The criteria value is then simply divided by the bad-high value.

Name	Specification description	Computation
xtd	touchdown point	$x_{td}(t_{td})$
vztd	vert. touch-down speed	$\dot{H}_{ra}(t_{td})$
B4	$\dot\theta$ may not change sign	$1 - \min_{t_1 \le t \le t_{td}}\{q(t)\}$
B6	$\dot V_Z$ may not change sign	$1 + \max_{t_1 \le t \le t_{td}}\{\dot V_Z\}$
t_{td} = touchdown time, t_1 = flare init time + 2 s		

Table 3: Deterministic flare criteria

Name	Bad low	Good low	Good high	Bad high	Type
xtd [m]	300	350	450	600	min
vztd [m/s]	-3.5	-3.0	-2.0	-1.5	min
B4 [rad/s]	–	–	0	1.15	constr
B6 [m/s]	–	–	0	1.15	constr

Table 4: Scaling of deterministic flare criteria

Table 5 show the applied scalings for the stochastic criteria. For three risk parameters the mean value,

Name	Bad low	Good low	Good high	Bad high	Type
meanHTP60 [m]	4	5	7	8	min
stdevHTP60 [m]	–	–	0	1.3	min
riskHTP60 [-]	–	–	0	1/6	constr
meanXTD [m]	250	350	400	450	min
stdevXTD [m]	–	–	0	100	min
riskXTD [-]	–	–	0	1/6	constr
meanVZTD [m/s]	-6	-4	-2	-1.5	min
stdevVZTD [m/s]	–	–	0	1.2	min
riskVZTD [-]	–	–	0	1/6	constr

Table 5: Scaling of stochastic flare criteria

the standard deviation, and the probability of exceeding a limit value are computed. These risk parameters are: the height of the main gear over the runway at 60 m from the threshold (HTP60) to assess the risk of short landings, the touchdown point on the runway (XTD) to assess the risk of long landings, and the vertical speed with respect to the runway surface (VZTD), to assess hard landings. Their risk values directly correspond to JAR-AWO criteria. For example, the probability of touchdown at more than 915 m from the threshold must be less than 10^{-6}. For optimization, the criterion is formulated as follows:

$$riskXTD = -1/\log(P(XTD > 915))$$

In combination with handling the criterion as a constraint, the bad-high scaling of 1/6 implies, that the probability should not exceed 10^{-6}. The mean values and standard deviations of the risk parameters are not prescribed, but sensible values were specified.

The stochastic criteria are computed via Monte-Carlo analysis. Each analysis involves 400 landing simulations. For each landing, 16 parameters (related to mass, centre of gravity location, wind, temperature, runway elevation and slope, ILS properties, etc.) are selected randomly, according to prescribed statistical properties. The risk parameters are computed from each individual simulation. After the simulations, their distributions and cumulative distributions are determined. These are assumed to be Gaussian. During the REAL project, a fast Matlab program was developed for Monte-Carlo analysis, called SIMPALE.

Figures 7 and 8 show the progress of the optimization (some iterations are hidden for better visibility, identical line styles correspond to the same iteration). The total computation time was about 18 hours on a PC (P-II 400 Mhz). All scaled criteria values are represented in so-called parallel co-ordinates (Fig. 7). All scaled criterion values are plotted on an individual axis and connected through a line (i.e. one graph corresponds to one tuning parameter set). The horizontal line indicates a value of one. Criteria values below this line are considered satisfactory. Parallel co-ordinates give quick insight in the optimization progress, criteria that are hard to satisfy, and criteria that conflict and thus have to be compromised. It can be seen that after 24 iterations (fat line), all criteria values are below one, thus all objectives are fulfilled. The right half of Fig. 8 shows the development of nominal landing simulations. The left half of Fig. 8 depicts cumulative distribution functions for the three risk parameters. The latter can be interpreted as follows (XTD example, fat curve): the probability of landing beyond 600 m is $10^{-2.2}$. The graph should not enter the dark region. It can be seen, that this is achieved by the optimization. Using statistical criteria in optimization was found extremely useful, since JAR-AWO robustness criteria could be addressed directly, and fulfilled.

7. Assessment

The purpose of the assessment is to detect hidden weaknesses in the designed autoland system. To this end, three approaches were used:

1. Parameter studies

The MOPS environment not only supports optimization or manual entering of parameters, but also parameter studies. The tuning parameters T are frozen and selected model parameters p are varied instead. The same criteria are computed and visualized as in the optimization. One possibility is

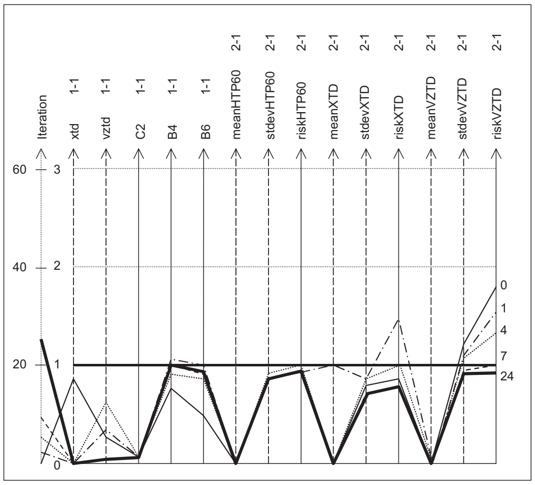

Figure 7: Optimization progress, parallel coordinates (dashed arrow = minimize criterion, solid arrow = inequality constraint on criterion)

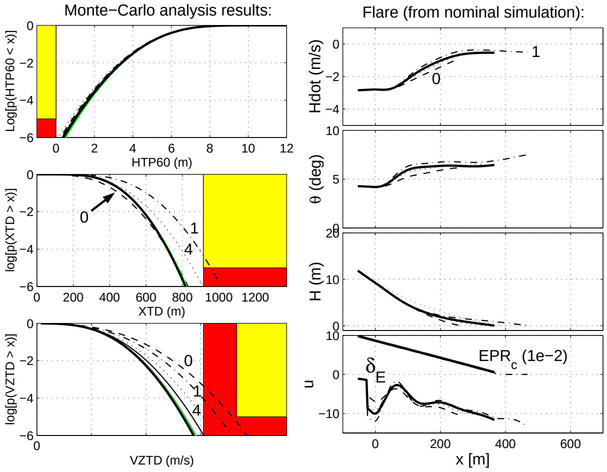

Figure 8: Optimization progress, Monte-Carlo analysis and nonlinear simulations

to find worst-case parameter by anti-optimization, in which case a selected criterion is to be maximized for the selected parameter set[1]. Another, frequently used method is grid-based parameter studies, in which case parameter values are taken at fixed points within their allowable ranges.

2. Monte-Carlo assessment

Monte-Carlo analysis is used to assess the probability of exceeding limits on risk parameters (e.g. vertical speed at touchdown), derived from 2000 landings under statistical variation of a large number of model parameters (wind, mass, runway slope, etc.). The maximum probabilities are specified by JAR-AWO. The case in which all model parameters vary around their mean value is called average risk analysis. In case of a limit-risk assessment, one of the model parameters is set to its maximum or minimum. The allowed limit values as well as their probabilities are then somewhat higher. Worst-case parameters can be extracted from the analysis results for closer analysis, or can even be added to a multi-model set in a new optimization.

3. Desktop flight simulator / 3D visualization

Qualitative assessment can be performed using the desktop flight simulator AVDS (Aviator Visual Design Simulator[15]). In this tool control laws as well as the aircraft model can be implemented (Figure 1). This was used to qualitatively assess the inner loops. AVDS also has an animation mode, in which data from simulations or flight tests can be imported and visualized in 3D.

Design iteration loops

The design process depicted in Fig. 1 has three main iteration loops. Loop 1 mainly involves adjustment of scalings on criteria, for example to steer trade-off between conflicting requirements by adapting their relative importance. This is also performed in loop 2, but in addition, assessment results can be used to update model cases in a multi-model set for optimization. Loop 3 involves modification in the models. This may be the plant model, e.g. for implementing more detailed model components, the controller model, in case architectural changes are required, e.g. addition of complementary filters or an integrator, or criteria models, e.g. when new issues show up in the assessment, that should be considered in the optimization as criteria.

The final control laws have been extensively tested using the methods mentioned above. As an example, Fig. 9 shows cumulative distribution plots for the afore mentioned longitudinal, as well as the lateral risk parameters YTP (touchdown distance from runway centre line), PHI (roll angle at touch down), and SSTP (slip angle at touch down). All requirements for average risks have been fulfilled. Based on assessment with the design model, all imposed criteria could be met, except for two limit-risk probabilities. Monte-Carlo assessment indicated that for maximum head wind or maximum cross wind, the probabilities of exceeding some of the risk parameters are too high. This is caused by the associated levels of turbulence (proportional to the wind). Most likely, the controller structure needs further refinements. Unfortunately, the time frame of the project did not allow to investigate this. In Ref.[2] final assessment results for the second benchmark (based on ATTAS) will be discussed in more detail.

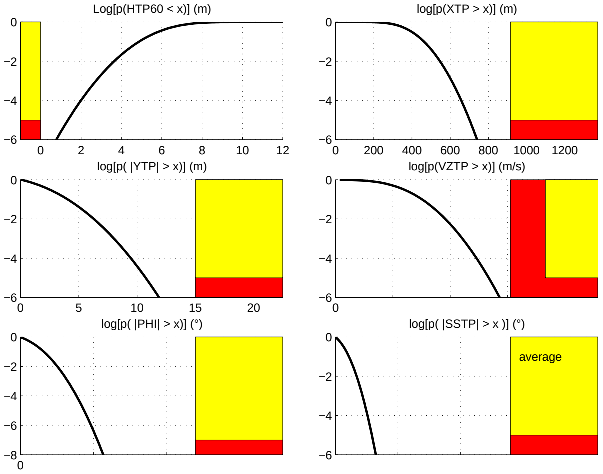

Figure 9: Monte-Carlo analysis results for final controller (average risks)

8. Conclusions

An optimization-based design process was described and applied to the design of autoland control laws. The process fulfills all criteria listed in section 1. Multi-objective optimization allows for incorporating a broad spectrum of criteria, computed from linear and nonlinear closed loop analysis. This allows for systematic incorporation of design requirements. Explicit consideration of uncertainties and parameter variations is addressed via a multi-model approach and optimization of statistical criteria from on-line Monte-Carlo analysis. In addition, robustness to unspecified uncertainties can be incorporated via gain and phase margins as design criteria. In principle, controller structure selection and parameter optimization are independent. The designer is able to incorporate experience-based structures that are visible and physically understandable. Finally, the process is automated to a large extent using optimization, and appropriate software tools that interact via data transfer and automatic code generation. The resulting designs fulfill nearly all imposed performance and robustness criteria. Only maximum head wind and cross wind limit risks are violated due to the associated turbulence levels.

The design process as well as the selected controller architecture were applied to design an autoland system for ATTAS (second benchmark problem). This was done in a short amount of time, proving redesign efficiency of the applied process. The resulting control laws were flight tested. Assessment results, the flight test procedure and flight test results are described in Ref.[2].

Acknowledgements

The RealCAM model implementation in Dymola/Modelica (and all underlying libraries) was performed by Dieter Moormann from the same institute as the first author. The extensively used Matlab-based Monte-Carlo assessment tool SIMPALE was developed by the REAL-project partner ONERA. Reports from, and discussions with Tony Lambregts (FAA) have been of invaluable help. The REAL project was sponsored by the Commission of the European Comunity under contract number BRPR-CT-98-0627.

References

[1] J. Bals, W. Fichter, and M. Surauer. Optimisation of magnetic and angular-momentum control for low earth orbit satellites. In *3rd ESA International Conference on Spacecraft Guidance, Navigation and Control Systems, ESTEC, Noorwijk, NL*, 1996.

[2] M. Bauschat, W. Mönnich, D. Willemsen, and G. Looye. Flight testing Robust Autoland Control Laws. In *Proceedings of the AIAA Guidance, Navigation and Control Conference 2001, Montreal CA*, 2001.

[3] H. Elmqvist. Object-oriented modeling and automatic formula manipulation in Dymola. In *Scandinavian Simulation Society SIMS*, Kongsberg, Norway, 1993.

[4] Joint Aviation Authorities Committee. Joint Aviation Requirements, All Weather Operations. Technical report, JAAC, 1996.

[5] H.-D. Joos. A methodology for multi-objective design assessment and flight control synthesis tuning. *Aerospace Science and Technology*, 3(3):161–176, April 1999.

[6] Hans-Dieter Joos. *In: J.F. Magni, S. Bennani, J. Terlouw, Eds. Robust Flight Control, a Design Challenge.*, chapter Multi-Objective Parameter Synthesis (MOPS), pages 13 – 21, 199 – 217. Lecture Notes in Control and Information Sciences, Vol 224. Springer-Verlag, 1997.

[7] H.D. Joos, A. Varga, R. Finsterwalder, and J. Bals. Eine integrierte optimierungsbasierte Entwurfsumgebung für Flugregelungsaufgaben. *Automatisierungstechnik AT*, Juni 1999.

[8] G. Kreisselmeier and R. Steinhauser. Systematische Auslegung von Reglern durch Optimierung eines vektoriellen Gütekriteriums. *Regelungstechnik, Heft 3*, pages 76 –79, 1979.

[9] A A. Lambregts. Avoiding the pitfalls in automatic landing control system design. *AIAA Paper 82-1599*, 1982.

[10] A A. Lambregts. Integrated system design for flight and propulsion control using total energy principles. *AIAA Paper 83-2561*, 1983.

[11] A A. Lambregts. Vertical flight path and speed control autopilot design using total energy principles. *AIAA Paper 83-2239*, 1983.

[12] Gertjan Looye and Hans-Dieter Joos. Design of Robust Dynamic Inversion Control Laws using Multi-Objective Optimization. In *Proceedings of the AIAA Guidance, Navigation and Control Conference 2001, Montreal CA*, 2001.

[13] S.E. Mattsson, M. Otter, and H. Elmqvist. Modelica Hybrid Modeling and Efficient Simulation. In *Proceedings of the of the 38th IEEE Conference on Decision and Control, CDC'99*, December 1999. Also available from http://www.modelica.org.

[14] D. Moormnann, P.J. Mosterman, and G. Looye. Object-oriented computational model building of aircraft flight dynamics and systems. *Aerospace Science and Technology*, 3(3), April 1999.

[15] S.J. Rasmussen and S.G. Breslin. AVDS: a Flight Systems Design Tool for Visualization and Engineer-in-the-Loop Simulation. AIAA 97-3467. 1997.