JOURNAL OF AIRCRAFT
Vol. 41, No. 4, July–August 2004

Preliminary Model Matching of the EMBRAER 170 Jet

Sergio Cavalcanti*
Empresa Brasileira de Aeronáutica S. A., 12227-901 São José dos Campos, Brazil
and
Marco Papini†
Opal-RT Technologies, Montreal, Quebec H3K 1GK, Canada

The development of the Empresa Brasileira de Aeronáutica (EMBRAER) 170 Jet has benefited from an expanded modeling and simulation capability at EMBRAER. The aerodynamics model and its accuracy became an important part of the design development phase. The structure of this Simulink model, how the model was matched to flight test, some of the problems encountered, and the advantages and disadvantages of the matching techniques used are discussed. RT-LAB/DINAMO, the parameter estimation software used, is also presented.

Introduction

THE Empresa Brasileira de Aeronáutica (EMBRAER), 170 Jet, a 70-passenger airliner, is the most recent aircraft to be developed at EMBRAER, São José dos Campos, Brazil. It made its first flight early in 2002, and it will receive certification mid-2003. It presents new challenges due to its fly-by-wire flight-control system, the expanded system integration, and the increased number of failure analyses required by certification agencies worldwide.

To overcome some of these issues, EMBRAER chose to utilize a model-centered (or model-based) design methodology. Simulation models would be utilized in all phases of the design process and would be deployed from desktop analysis to hardware-in-the-loop (HIL) test rigs. Furthermore, a single model would be used (or some of its pieces) in as many different applications as possible, minimizing model maintenance and maximizing efficiency.

To use a single model for many different applications, such as flight dynamics analysis, training, control law design, functional hazard analysis and HIL test rigs, the model had to be matched beyond standard requirements and the complexity extended beyond the norm. The aerodynamics, for example, now had to encompass regions of the flight envelope not explored in conventional airplanes, such as a poststall, and the control system's complexity had to exceed that of typical single application models, for example, the inclusion of hydraulic failures.

New Concept in the Use of Simulation

Just a decade ago, most flight dynamics simulation models were built with the focus on the need for training simulators. During this period, the interim level C and level D requirements (simulator certification levels) were the targets of model matching. New developments, however, have considerably changed this concept. Modern aircraft are frequently using fly-by-wire flight controls and integrated systems. Consequently, the flight-controls team needs fairly sophisticated controls and aerodynamics models to design the fly-by-wire law. The handling qualities team needs the same kind of models to guarantee design objectives. The engine team needs a complete engine model to evaluate integration issues. The systems team needs to analyze all possible combinations of failure cases, the autopilot subcontractor needs the most accurate aerodynamics to assure specifications are met, and so on. The demands on the simulation model have grown substantially.

At EMBRAER, the modeling process started very early in the design process with an initial model based on wind-tunnel data and flight-controls specifications. Over the course of the aircraft program, the initial model was expanded to include new systems, updated to reflect current design specifications and modified to mirror the actual implementation. The maturing of the program did not simplify the models, as one might expect. Indeed, the eventual deployment of the models in HIL test rigs required unprecedented complexity to assure that all signals expected by the hardware were, in fact, available and realistic.

Another great challenge in the development of such sophisticated simulation models has become the accuracy of their prediction, especially given the extremes (flight envelope and configuration) to which these models would be run. All components of the model had to be, and were, matched to reality. However, the focus of this paper will be mainly limited to a description of the methodology and tools by which the aerodynamics model was matched to flight-test data.

Integrated Simulation Model

The integrated simulation model was entirely built with Simulink/MATLAB® (MathWorks, Natick, Massachusetts). The main block diagram in Simulink is shown in Fig. 1. It consists of several sub-systems, each kept and maintained in special libraries and sublibraries. All subsystems pass their information to the other modules and post the results on the main bus, which is used to output data.

The subsystems (Fig. 1) necessary for the preliminary model are autopilot, pilot controller, atmosphere, simulator interface, controls (flight controls), hydraulics, ground handling, aerodynamics, engine, sensors, equations of motion, and mass properties. In this example, only 10 of the 12 blocks pass information to the output bus.

Aerodynamics Library

The aerodynamics library is prepared in a format that is also suited to loads calculations. With this purpose, the main sublibraries are as follows: 1) WingBodyNacellePylon 2) Horizontal Tail 3) Vertical Tail 4) Ground Effects 5) Dynamics 6) MassFlow (Engine Inlet) 7) Mach Buffet, and 8) Hinge Moments.

Every sublibrary contains the evaluation of the six aerodynamic coefficients (C_D, C_Y, C_L, C_R, C_M, and C_N), which are then added in the sublibrary aero totals. A modular view of the Simulink aerodynamic library is presented in Fig. 2.

Received 18 June 2003; presented as Paper 2003-5534 at the Atmospheric Flight Mechanics Conference, Austin, TX, 11 August 2003; revision received 18 September 2003; accepted for publication 22 September 2003.

*Senior Engineer, Flight Mechanics, Av. Brig. Faria Lima, 2170, CEP; Sergio.Cavalcanti@embraer.com.br. Senior Member AIAA.

†Manager, Aerospace Modeling and Simulation, 1751 Richardson, Suite 2525; Marco.papini@opal-rt.com.

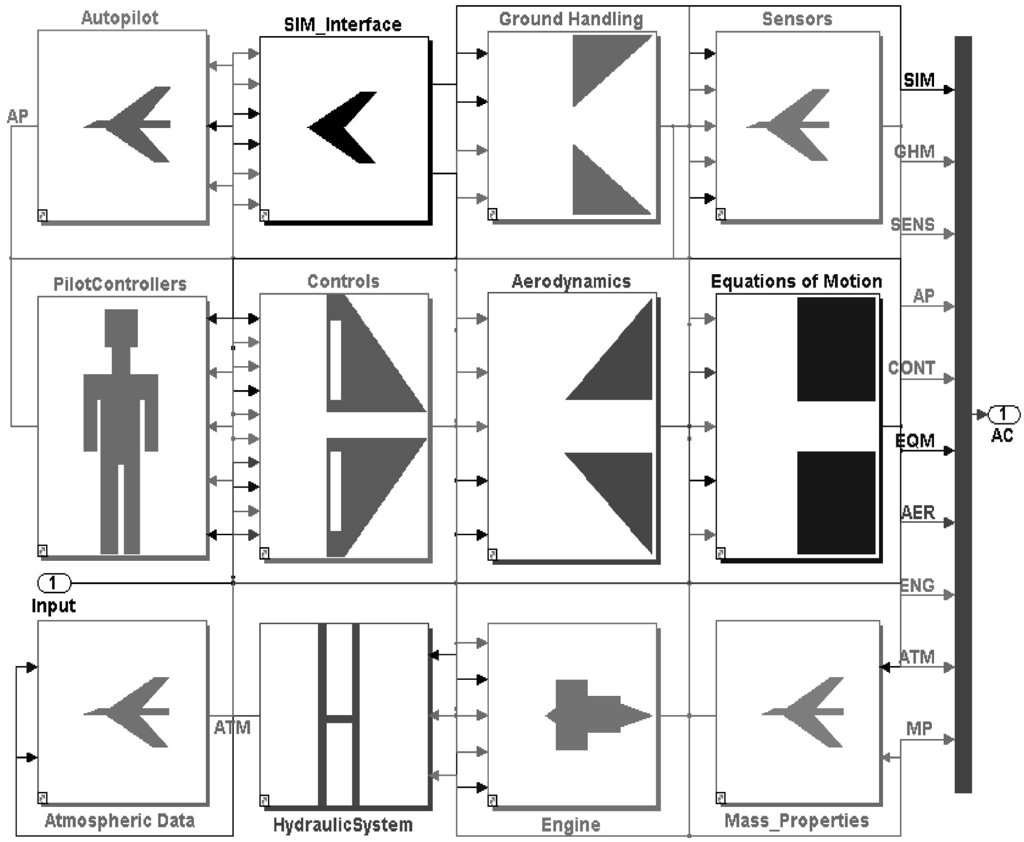

Fig. 1 Integrated simulation model in Simulink.

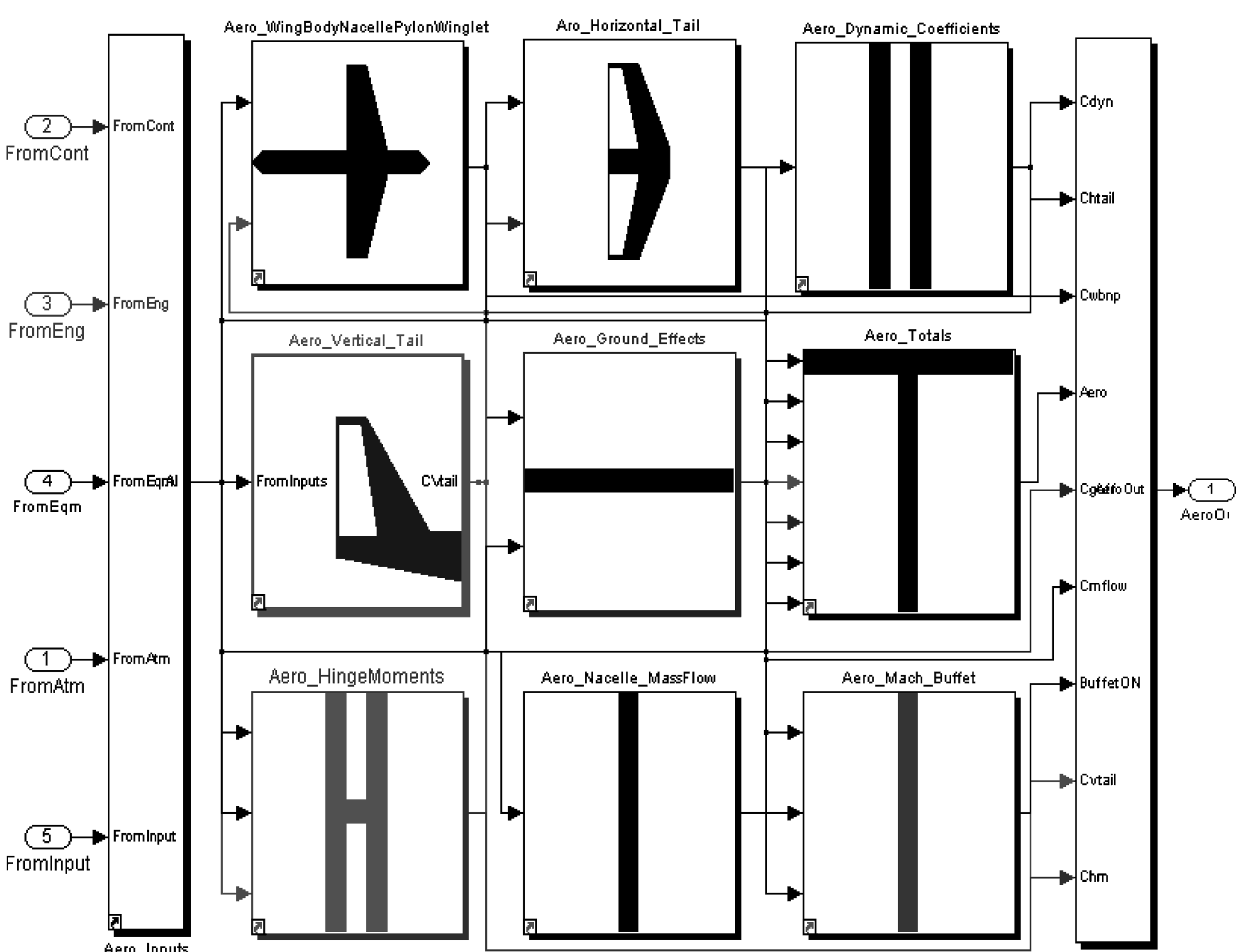

Fig. 2 Aerodynamics model in Simulink.

Validation

Validation of each module is independently achieved with different techniques. The modules that required the most extensive validation were controls, engine, and aerodynamics.

The controls module was validated against flight-test data by the use of the so-called iron bird, a system test rig that is a hybrid of simulation and hardware. To interface the simulation to actual aircraft components, the models were compiled with MATLAB/RTW and RT-LAB (OPAL-RT Technologies, Montreal, Quebec, Canada). DINAMO (also from OPAL-RT) was then used to trim the simulations, inject inputs, and run the simulations in real time.

The Engine model was validated with special flight tests and separate system test rigs. RT-LAB was used in these rigs to run the models in real time and for debugging purposes.

The validation of the aerodynamic model, the main objective of this paper, was confirmed with flight-test data. Here, new tools had to be built to simplify this laborious task.

Parameter Estimation

The task presented to the OPAL-RT Technologies team was to build software capable of tuning the detailed nonlinear simulation model built in Simulink to match flight-test time histories. This software had to be as automated as possible to deal with the enormous amount of data expected; in the peak of the flight tests, six prototypes were used, with at least three of them providing data that required analysis and parameter tuning. Furthermore, high accuracy and consistency was required for the myriad of applications of the model.

Initial attempts to do parameter estimation (PE) entirely within the Simulink environment made it painfully obvious that the process was too slow to handle the enormous amounts of data required for model matching; the model had to be run outside MATLAB. To accomplish the task, it was decided that the Simulink model would be compiled with existing software, then interfaced to a new PE module.

Generation of the C-code and compilation was achieved through OPAL-RT's main product, RT-LAB[1] and MATLAB's RTW. The benefits of using RT-LAB were that it produced very fast code; that it provided access to internal model parameters, inputs, and outputs; and that it had the capability to distribute the task over multiple personal computers, if required.

Another in-house code (DINAMO[2]) was used for runs of the precompiled Simulink models. DINAMO is an aeronautics specific code that has the capability to configure trims, create input time histories, and produce batch runs. The PE algorithms would be inserted into DINAMO to produce an integrated aeronautics tool.

Experience with model matching dictated that the problem be split into two pieces, static coefficients first and then dynamic coefficients. Static coefficients could be determined from trims, whereas dynamic coefficients could be extracted from maneuvers. However, a different type of PE algorithm was then required for each type of coefficient.

Trims are static conditions and have single-valued outputs; thus, they are not well suited to least-square PE algorithms. Furthermore, trim computations often produce large discontinuous jumps in solution space from one trim iteration to the next. Accordingly, the algorithm selected for matching trim conditions was the Nelder–Mead simplex. This algorithm is known to have a slow convergence rate, but excellent robustness in the presence of discontinuities.

For the dynamic coefficients, maneuvers (time histories) provided a great deal of data at each run, but took a long time to execute. Thus, least-squares methods were a natural choice as long as a good convergence rate could be achieved. For these tasks, the Levenberg–Marquardt algorithm was chosen, mainly for its excellent convergence rate.

Both algorithms are well documented in the literature,[3,4] but, for completeness, a brief description is included hereafter.

PE Algorithms

The Nelder–Mead simplex (NM) is a very widely used and simple algorithm for solving nonlinear unconstrained optimization problems. It is a downhill, direct search, local minima algorithm that does not require the computation of any derivatives.

Given N parameters to determine and an objective function of the form

$$F(\boldsymbol{x}) = \sum (y_m - y_t)^2$$

where the y_m are the desired model outputs to match the y_t flight-test trim values and $\boldsymbol{x}$ is a vector of dimension N, the algorithm begins by creating $N+1$ vertices in N-dimensional space, describing a nonzero volume geometric figure, referred to as a simplex. The objective function is evaluated for each vertex of the simplex and the vertices sorted according to the magnitude of the objective function. Then, by a set of simple mathematical transformations (reflection, expansion, contraction, and shrinkage) the worst vertices (those associated with the largest objective function) are moved to new points that produce smaller objective function values. In this way, the simplex is said to move toward a minima around which it will contract to the solution.

The Levenberg–Marquardt algorithm (LM) is also a local minimization algorithm. Its classic form is intended for unconstrained nonlinear least-squares problems, but versions also exist for solving constrained minimization problems. As a least-squares algorithm, the objective function is of the form

$$F(\boldsymbol{x}) = \sum (\boldsymbol{y}_m - y_f)^2$$

where $\boldsymbol{y}_m$ is a vector of n model observations in time corresponding to $n \times y_f$ flight-test observations; $\boldsymbol{x}$ is a vector of N unknowns, $n \gg N$. The algorithm consists of a clever mathematical trick that merges the equations of a gradient search algorithm (steepest descent) with the equations of a Newton search algorithm to produce the following equation:

$$(J^T J + \lambda I)\boldsymbol{x} = -J^T f$$

Where J is the Jacobian of $f(\boldsymbol{x}) = (\boldsymbol{y}_m - y_f)$, I is the identity matrix, and λ is the Marquardt parameter. When $\lambda = 0$, a pure Newton step is taken; when $\lambda \gg 0$, a pure downhill step is taken. Many methods exist for computing λ producing subtle variations of the algorithm. The one chosen here is based on the computation of the trust region for $\boldsymbol{x}$, which guarantees convergence at every iteration with minimal computations.

DINAMO Interface for Parameter Estimation

Model Preparation for Parameter Estimation

The aerodynamics model is prepared for the parameter estimation task by inserting MATLAB parameters at any point where adjustment is needed. To make the parameters distinguishable for the PE module, a naming convention of PEG_ (for gains) and PEC_ (for bias) prefixes was adopted (Fig. 3). These values are typically initialized to 1 and 0, respectively.

Creating DINAMO Files

To create one PE run with either algorithm of the PE module, three files must be created:

1) The trim configuration file set the model to the same configuration as the flight-test data (DINAMO trim interface).

2) The PE file contains the algorithm to use, its termination tolerances, inputs, outputs, and parameters (DINAMO parameter interface).

3) The flight-test data file contains data to be matched.

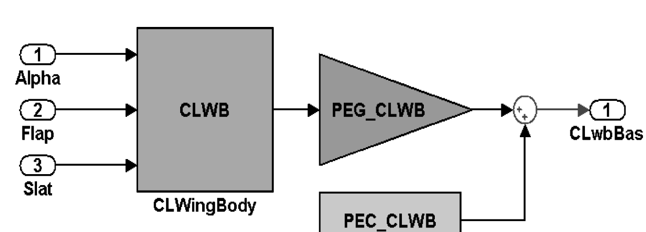

Fig. 3 Parameter gain and bias.

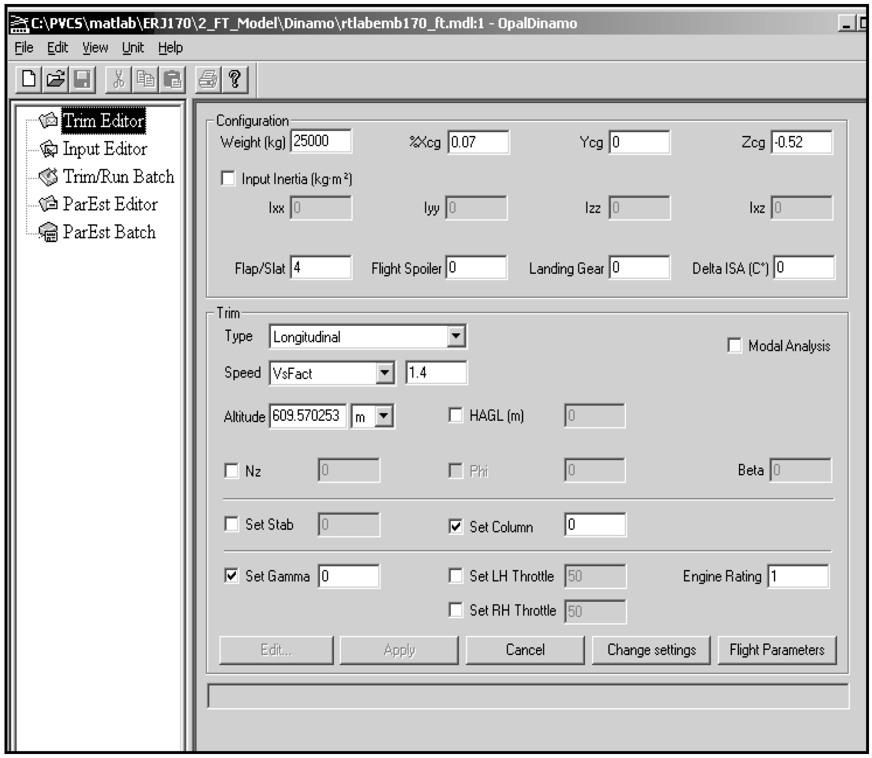

Fig. 4 DINAMO trim editor.

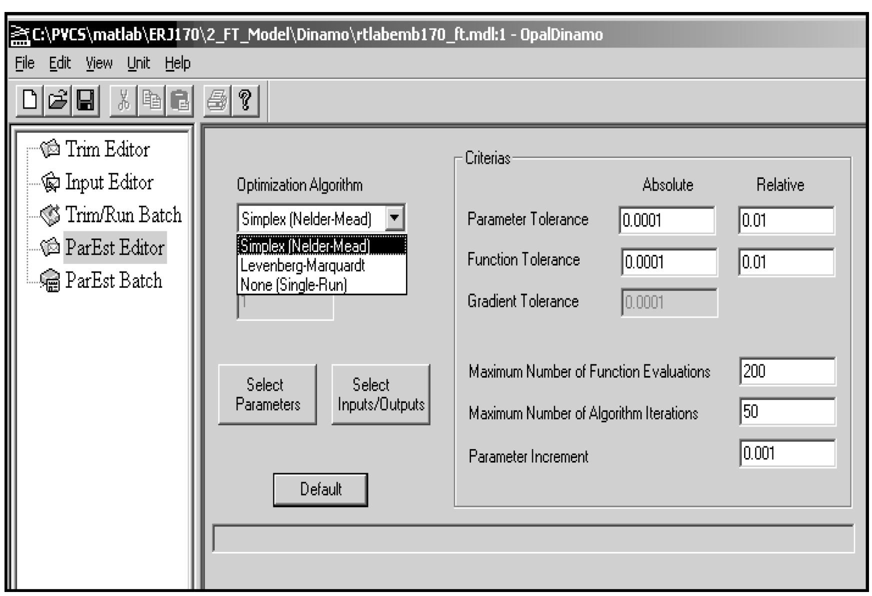

Fig. 5 DINAMO PE editor.

User interface panels are provided to simplify the task of creating these files.

DINAMO Trim Editor

The trim editor allows for the setup of the trim configuration (Fig. 4), including weight, c.g., flaps, and flags. The configuration is stored as a file for latter use.

DINAMO PE Editor

Another interface is the PE editor (Fig. 5). Capabilities for choosing the PE method, setting the PE termination criteria, and the variables for the PE case are provided.

From the PE editor, the Parameter and input/output panels can be accessed.

Within the parameter panel, the parameters to be used by the algorithm (Fig. 6) can be selected. Similarly, to select the output time histories to be matched and the input time histories to excite the simulation, the input/output selection panel (Fig. 7) is used. All of the data entered through these panels are collected and saved in a PE configuration file for latter use.

DINAMO PE Batch Panel

Within the batch panel (Fig. 8), the trim configuration, PE configuration, and flight-test data files are collected together to form one PE case. DINAMO's batch capability is then exploited to organize and run a batch of PE cases. The individual PE runs can be organized into a sequence and the sequence can then be stored. The different types of PE runs can be freely mixed in forming the batch.

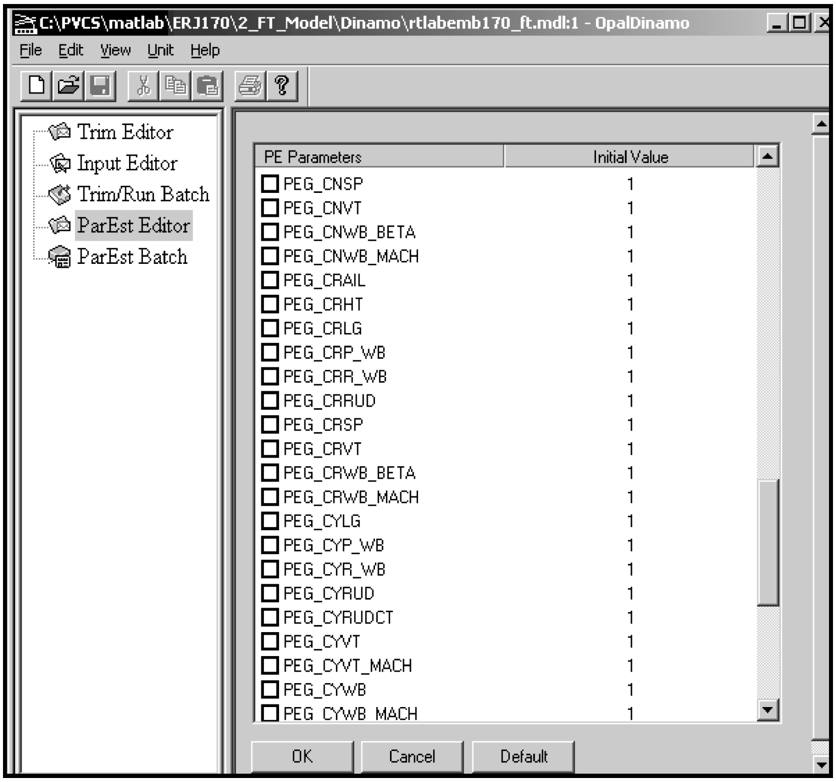

Fig. 6 DINAMO parameter gain/bias selection.

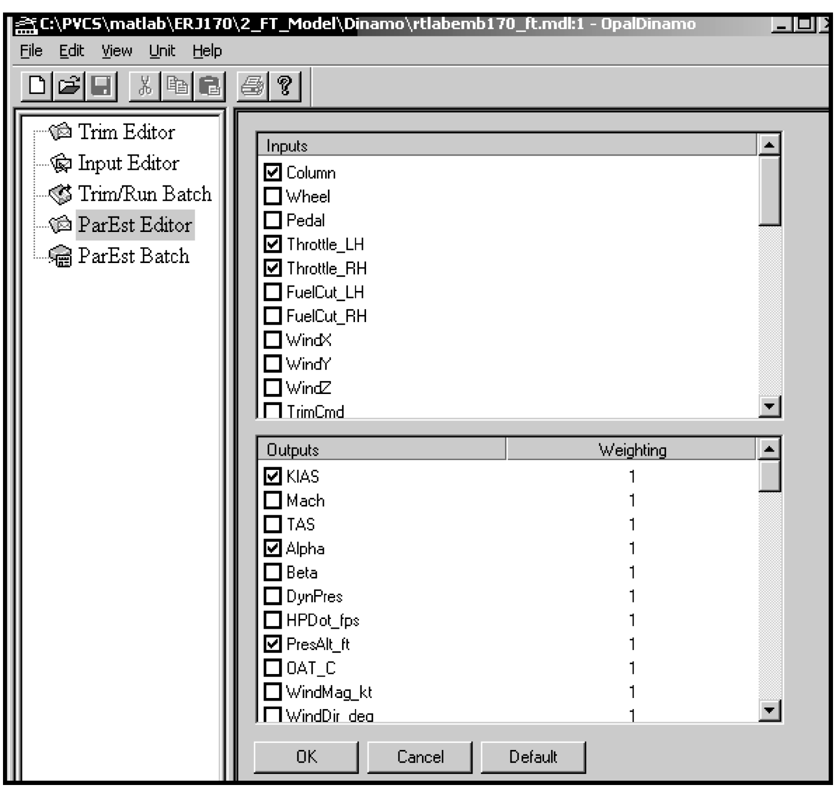

Fig. 7 DINAMO input/ouput selection.

PE Output

The output produced by the PE module is in the form of text files. The files contain information regarding the progress of the algorithms, termination conditions, final values, and associated statistical information. The type of data produced is dependent on the algorithm selected.

The NM algorithm prints the parameters, outputs, and objective function values at each iteration. (These are also available in a separate file for easy plotting.) After the final iteration, the algorithm prints the termination condition, comparison of model vs flight-test outputs, and the sensitivities of the objective function to the parameters.

The LM algorithm prints the parameters, outputs, and objective function values at each iteration. (These are also available in a separate file for easy plotting.) After the final iteration, the algorithm prints the termination condition, parameter, values, sensitivities of

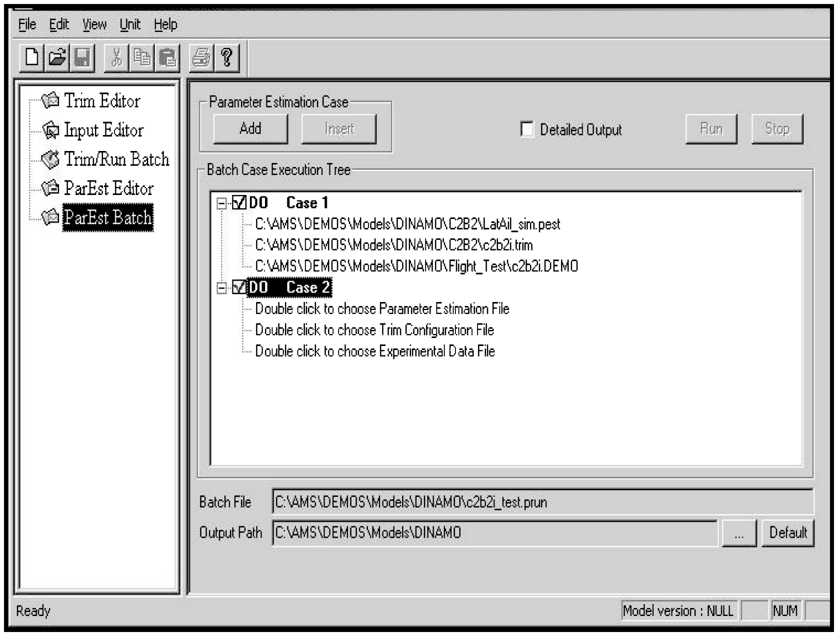

Fig. 8 DINAMO PE batch panel.

the objective function to the parameters, and the estimated error (rms) in the parameter values. A separate file is also produced to simplify plotting of the PE match. The file contains the time histories of the selected outputs, both from the model and flight test, as well as the residuals from these outputs (the absolute difference between model and flight test).

Flight-Test Validation

To explain the methodology used to match the flight dynamics model to flight-test data, the maneuvers used have are separated into four categories: 1) ground maneuvers (takeoff, landing, etc.); 2) lateral handling qualities; 3) longitudinal handling qualities, and 4) high angle of attack (stall, windup, and pullup).

All maneuvers are matched by the use, as much as possible, of direct inputs from flight tests. In some cases, however, a controller is used to keep the inputs close to the flight-test values. Figures 4–8 show examples of flight-test matches.

Wind and turbulence effects were included for all maneuvers and were computed from flight data with airspeed and inertial velocity information.

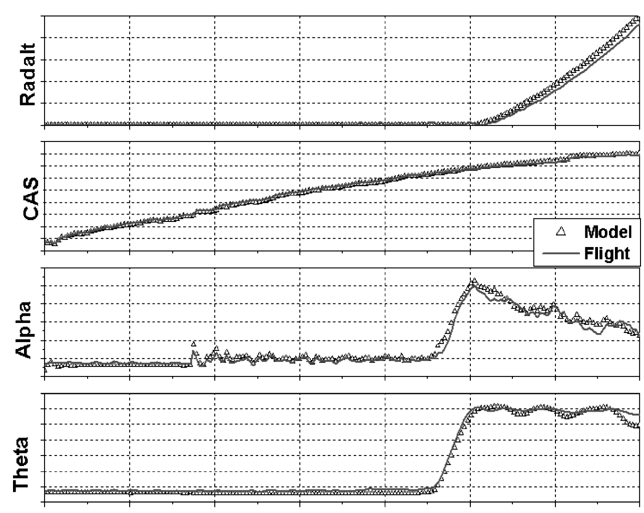

Fig. 9 EMBRAER 170 Jet normal takeoff match.

Ground Maneuvers

Normally, for Interim Level C (ILC), only three takeoffs, three landings (normal, crosswind, and one engine), a minimum control speed on ground (VMCG), and a ground effect maneuver is necessary. However, as explained before, this model has another objective that is more comprehensive than the ILC scope. Thus, for the preliminary matching of this kind of model, takeoffs (Fig. 9) and landings at all flap settings are necessary; VMCG is also done for all takeoff flap settings.

For the ground effect, the shallow approach (1.5-deg approach) is used instead of the classical fly-by maneuvers. The reason is that the shallow approach, if properly done, provides a continuous way of tuning the ground-effect model that is more in line with the PE methodology used. There are other reasons: The fly-by is a more risky maneuver, it is very difficult to perform correctly, and it only provides data for tuning one single height above ground. In fact, the authors would go so far as to suggest that this maneuver should be removed from the requirements and more emphasis placed on the shallow approach.

Another important aspect of the ground maneuvers matching is the correct modeling of the runway characteristics. Correct runway profile and ground profile before runway threshold has to be used particularly if radio altimeter information, instead of a differential global positioning system, was used during the match.

Lateral Handling Qualities

For the lateral handling qualities, two phases are performed that can be described as static and dynamic matches.

In the first phase, the NM algorithm is used to tune the trimmed points for steady heading sideslips (SHSS) and one-engine out. For SHSS, many points are used, varying beta, airspeed, and flap; more than 20 sideslips were used for the preliminary match. For OEO, three different ways of performing the maneuver are utilized: fixing beta, aileron, or rudder. With this approach, the amount of flight-test data is greatly increased; however, the accuracy of the validation is extraordinary.

In the second phase, the dynamic tests were matched with the LM algorithm. Dutch roll (Fig. 10), aileron, and rudder responses are the basic dynamic maneuvers used.

Longitudinal Handling Qualities

Longitudinal matching is also performed in two phases. The static part is performed with the trims before any maneuver is matched. Speed, flap, and speed brake settings were varied, and trims were

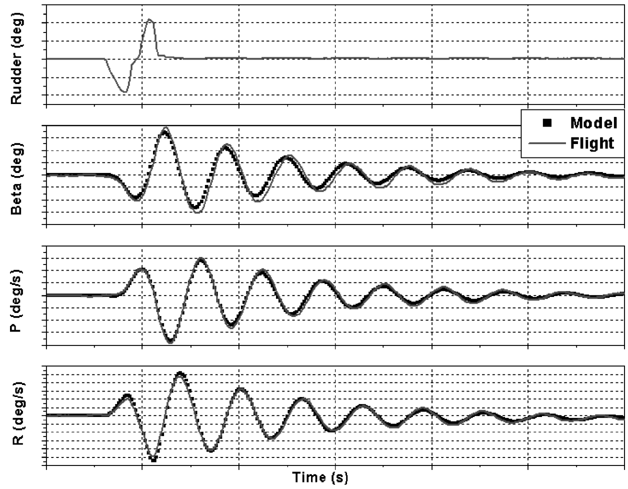

Fig. 10 EMBRAER 170 Jet dutch roll match.

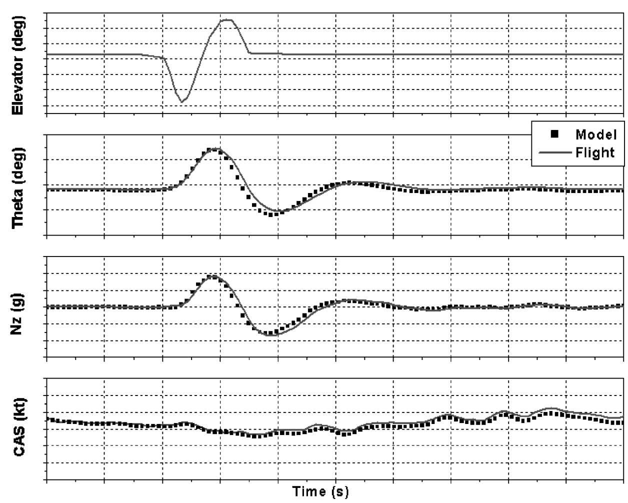

Fig. 11 EMBRAER 170 Jet short-period match.

performed at idle and maximum power settings to adjust the thrust effects on the model. To use as many test points as possible, test pilots were told to hold a trim for at least 5 s before each maneuver. With this simple procedure, it was possible to collect more than 1000 trim points for the flight-test database. This enabled the accurate adjustment of the static part of the model throughout the flight envelope.

The dynamic tests included short-period, elevator response, and configuration changes, namely, landing gear, flap, speed brake, and power effects.

The example given in Fig. 11 shows the short-period match, which, although not perfect, is well within the frequency and damping tolerance required for this maneuver.

High Angle of Attack

This became a special category for the EMBRAER 170 Jet due to the special angle-of-attack limiting function in the fly-by-wire system. Because this function exists throughout the Mach envelope, a number of special tests were required to adjust the C_L, C_D, and C_M curves at all Mach.

For stall tests (Fig. 12), a special requirement was to match stall beyond the wing–body stall angle to obtain data for the design of the angle-of-attack (AOA) limiter. Initially, this proved too difficult to do with the automated procedure. However, by adding gains and biases for restricted alpha regions, this was also made possible.

For high Mach (above 0.5), experience showed that the best tests to use were the windup turns. These tests are performed at constant speed, which allows the line-per-line correction of the aerodynamic tables (functions of Mach). The windups produce a variation of AOA by adjusting the bank angle and varying load factor. They proved to be easy to match with the automated procedure. In the EMBRAER 170 Jet program, windup turns were performed from low speed to Mach 0.82, up to the maximum alpha. The quality of these tests allowed for an exceptional match of the basic longitudinal coefficients.

Table 1 Typical NM PE results

Parameter	Tuned gain	Sensitivity	Outputs	Model	Flight test
PEG_CLwb	0.956	1.078	Alpha	2.553	2.549
PEG_CMwb	1.004	1.405	Stab	−2.708	−2.710
PEG_CLelev	1.101	1.127	Elevator	0.176	0.215

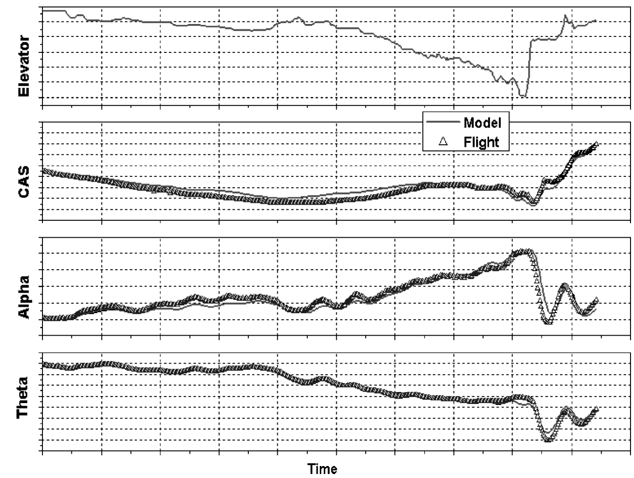

Fig. 12 EMBRAER 170 Jet stall match.

Use of DINAMO for PE

Because DINAMO was the basic tool used for the flight-test matching, and it was being used for the first time, a critical review of its use is necessary.

Trim Matches

The static matches were performed using the NM algorithm. The NM is easy to use and provides very fast matches. The procedure for each case is to use the DINAMO interface (Fig. 6) to select the gains (PEG_) and biases (PEC_) that most affect the aerodynamic coefficients of interest and then let the algorithm do the work. A typical result of a run with the NM is shown in Table 1 (not EMBRAER 170 Jet data).

Its use proved to be extremely valuable because of the amount of data that was available. DINAMO can be used to match the trims by automatically varying the parameters inside the model, but it also allows the verification of the results by permitting single runs at chosen parameters values. To be most effective, however, the possible variation of each parameter has to be bounded and specified by the user.

Time History Matches

Time histories were matched with the LM algorithm. By the use of the DINAMO PE interface, the inputs, outputs, and parameters are easily selected. Table 2 lists examples of the choices of inputs/outputs and parameters for typical maneuvers. If necessary, the initial conditions, as well as bounds, are chosen. The use of bounds is important because otherwise the algorithm can converge to values that are not realistic.

Typical results of an LM iteration are shown in Table 3 (not EMBRAER 170 Jet data). A critical review of the final parameters values is very important. Because of the very nature of the optimization algorithms, a local minimum may be found that is not satisfactory. If so, a change of parameter choices with, as always, a good understanding of the dynamics of that specific maneuver, may lead to a better solution.

An LM run produces three types of results for every output: the flight-test time history, the simulation model time history, and the

Table 2 Typical input/output and parameter choices

Maneuver	Basic inputs	Matched outputs	Coefficients
Short period	Elevator	CAS	CL_elev
		Theta	CM_q
		Alpha	CM_wb
		Nz	
Dutch roll	Rudder	Beta	CN_beta
		Phi	CR_beta
		P	CY_beta
		R	CY_rud
Takeoff	Throttle elevator	CAS	CL_geff
		Alpha	CD_geff
		Theta	CM_geff
		Radalt	
Stall	Elevator	CAS	CL_wb
		Alpha	CD_wb
		Theta	CM_wb
			CL_elev

Table 3 Typical LM PE results

Parameters	Tuned gain	Sensitivity	Error
PEG_CLwb	1.015	0.127	0.0005
PEG_CMwb	0.987	0.015	0.0034
PEG_CLelev	1.056	0.135	0.0007

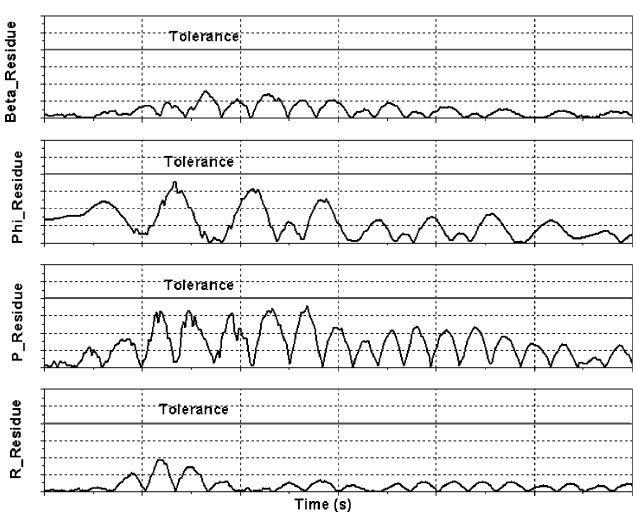

Fig. 13 Typical tolerances for a Dutch roll match.

absolute value of the error between flight test and model. This enables immediate plotting of the tolerances to evaluate the accuracy of the matching (Fig. 13).

The LM algorithm is most useful when great amounts of data are available. It provides the first approximation of corrections that must be applied to the model. The final tuning has to be done, in most cases by single runs, which is possible with the DINAMO PE module. The single-run capability of DINAMO is particularly useful, not only for evaluation of the accuracy of the final matching, but also for getting an idea of the sensitivity of the matching to the parameters chosen.

Conclusions

The simulation model has become the backbone of the engineering design and analysis for the aircraft design team. The need for an accurate prediction model and for a more extensive and comprehensive validation has become mandatory. Because of the enormous quantity of data that needs to be analyzed to obtain the best possible match, automated software procedures become necessary. For this purpose OPAL-RT has developed an interface to its main software RT-LAB, called DINAMO. This interface takes care of trimming the model, running in batch mode, and running with optimization algorithms, either for trim points derived with the NM algorithm, or for time histories derived with the LM algorithm. Both algorithms can produce good results, but, in certain cases, the LM algorithm needs to be complemented with hand tuning. Both algorithms need the capability to establish bounds for parameters to avoid convergence toward undesirable local minima.

The parameter estimation tools discussed were used to tune and update the flight dynamics model of the EMBRAER 170 Jet during the design and certification phase. Some preliminary results have been presented in this paper, as has a review of the methodology used.

Acknowledgment

Ravindra Jategaonkar of DLR, German Aerospace Research Center, Institute of Flight Systems, Brunswick, Germany organized and coordinated the present *Journal of Aircraft* Special Focus Issue.

References

[1]"RT-LAB, Version 6.2, User's Manual," OPAL-RT Technologies, Montreal, Quebec, Canada.

[2]"DINAMO Version 6, User's Manual," OPAL-RT Technologies, Montreal, Quebec, Canada.

[3]Lagarais, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E., "Convergence Properties of the Nelder–Mead Simplex in Low Dimensions," *SIAM Journal on Optimization*, Vol. 9, No. 1, 1998, pp. 112–147.

[4]More, J. J., "The Levenberg–Marquardt Algorithm: Implementation and Theory," *Lecture Notes in Mathematics 630*, Springer-Verlag, Berlin, 1977, pp. 105–116.