

# Validation Techniques Applied on the Saab Gripen Fighter Environmental Control System Model

Robert Hällqvist<sup>1</sup> Magnus Eek<sup>1</sup> Ingela Lind<sup>1</sup> Hampus Gavel<sup>1</sup>

<sup>1</sup>Saab Aeronautics, Linköping, Sweden

## Abstract

The Environmental Control System (ECS) of the Saab Gripen fighter provides a number of vital functions, such as provision of coolant air to the avionics, comfort air to the cockpit, and pressurization of the aircraft fuel system. To support system design, a detailed simulation model has been developed in the Modelica-based tool Dymola. The model needs to be a “good system representation”, during both steady-state operation and relevant dynamic events, if reliable predictions are to be made regarding cooling performance, static loads in terms of pressure and temperature, and various other types of system analyses. A framework for semi-automatic validation of the ECS model against measurements is developed and described in this paper. The framework extends a proposed formal methodology of semi-automatic model validation against in-situ measurements to the model development process implemented at Saab. Applied methods for validating the model in steady-state operation and during relevant dynamic events are presented in detail. The developed framework includes automatic filtering of measurement points defined as steady-state operation and visualization techniques applied on validation experiments conducted in the previously mentioned points. The proposed framework both simplifies continuous validation throughout the system development process and enables a smooth transition towards a more independent verification and validation process.

*Keywords: Verification and Validation, Coverage, Domain of Validity, Historical Data Validation*

## 1 Introduction

Model-Based System Engineering (MBSE) is playing an increasingly important role at Saab Aeronautics. Modeling and Simulation (M&S) is used already in early system development phases to increase the understanding of complex, highly integrated, and strongly coupled systems' behavior. In addition, a growing number of design decisions are taken relying on simulation results as simulating a system under investigation often is less costly than physical testing of the

actual system (Carlsson, Andersson, Gavel and Ölvander, 2012). Such an outspoken strategy imposes high demands on the simulation models to cover their *intended use*, see section 2.2. Model Verification and Validation (V&V) then become critical activities.

Model validation is an iterative process that continues for at least as long as the system that the model represents is under development. There is therefore a need to automate the validation process in order to assess the model validity with respect to the current system configuration in a convenient manner. As long as model validation requires a significant manual engineering effort, V&V activities will be “rare events” during system development. A high degree of automation is a necessary prerequisite for a continuous model V&V process. Increasing the level of automation would significantly simplify the execution of V&V activities, allowing for an increase in simulation result credibility with relatively little effort.

The methodology presented in this paper is a result of automation efforts performed when validating the Gripen Environmental Control System (ECS) model statically and the dynamic events for which the model needs to be a good system representation according to its intended use. The presented semi-automatic framework for ECS model validation is a proposed formal extension to the model development process implemented at Saab, see Figure 1. The model validation presented here is performed against existing in-situ measurements on a system level. The presented framework is developed specifically for the Gripen ECS simulation model. However, much of the work is considered to be generic and may hopefully guide validation efforts of other physics-based system simulation models for which *in-situ measurements* exist.

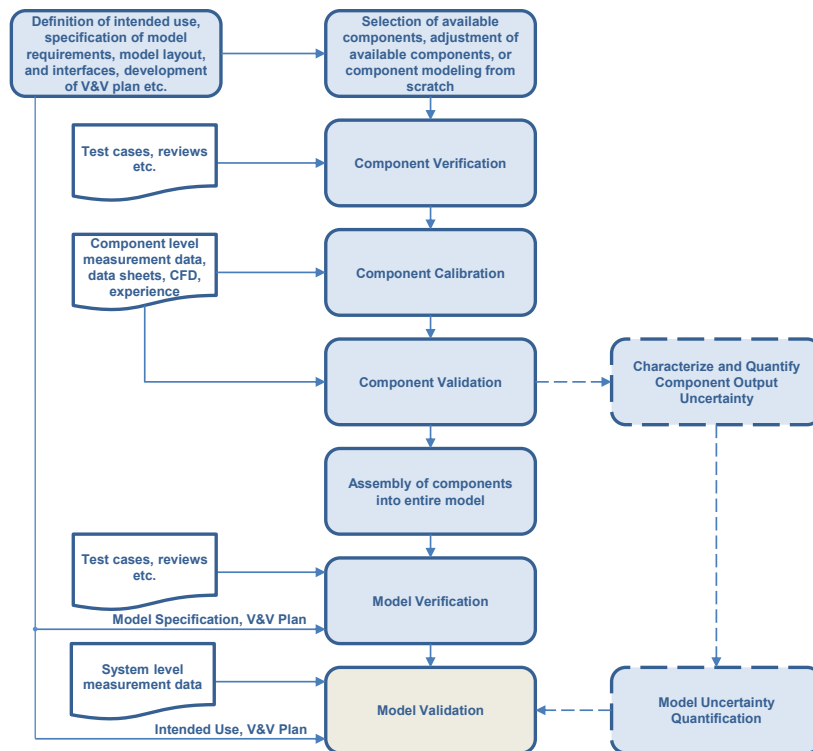
## 2 Theoretical Background

To provide a context for the presented validation methodology, the simulation model development workflow is visualized in Figure 1 (Carlsson,

2013). This workflow quantifies the model development process implemented at Saab Aeronautics, beginning with definitions of intended use and requirements specifications, all the way to system level model validation against reference data. A serious validation effort of equation-based simulation models requires such a bottom-up approach. Extensive system level validation should therefore preferably not be performed until after at least a sub-model by sub-model validation has been executed. Prior to the

sub-model validation, validation should be performed on a component level.

This paper presents a pragmatic decomposition of the *Model Validation* step of the model development process, see Figure 1. Existing validation measures and techniques are modified and applied, rendering a formal proposal of a semi-automatic validation procedure of physics based models.



**Figure 1:** A general process for simulation model specification, development, V&V, and uncertainty quantification (Carlsson, 2013).

## 2.1 The concepts of Verification and Validation

The concepts of verification and validation (V&V) are fundamental to the work presented in this paper. As there are several available definitions of verification and validation, the interpretations related to the work presented here are explained in the following paragraphs.

The term validation is interpreted as the process of determining the model's validity within its range of usage, i.e. to determine if the model represents the physical system in the operational point, and during the dynamic events of interest, "well enough". The term "well enough" is here defined by relevant validation measures in relation to the model's intended use. This is generally in line with the definitions provided by NASA (2008), SISO (2013), and the

DoD (2007). There are numerous techniques for model validation with varying levels of formality. This paper focuses on model validation using measurement data from the real-world system, sometimes referred to as *historical data validation* or *predictive validation*, depending on the order of execution (Sargent, 2010).

In practice, model verification can be difficult to separate from model validation. However, in this work verification is related to ensuring a correct implementation and that the model requirements are met.

In order to ensure objectivity in V&V, it is important to have a certain degree of independence in the V&V process. What is a suitable level of independence varies depending on for example the type of organization, application, model's intended use, and the level

of formality required. Arthur and Nance (2000) defines independent Verification and Validation (IV&V) as “a series of technical and management activities performed by someone other than the developer of a system with the objective of: improving the quality of the system, and assuming that the delivered product satisfies the users operational needs.”

## 2.2 Model Intended Use

A model’s purpose (or intended use) should serve as input to the model’s development process as well as later model level validation activities, see Figure 1. The intended use should also serve as the foundation when deciding which modeling technique to implement when developing the model. The model’s intended use should therefore be considered a prerequisite to model development and latter activities (Carlsson, 2013). However, it is often practically impossible to develop a complete set of well-defined intended uses in the initial phase of model development. In later phases, when the model is available to users, new areas of use of the model will probably be found. It is nonetheless important to obtain as complete a picture of the intended use as possible prior to model development (Carlsson et al., 2012).

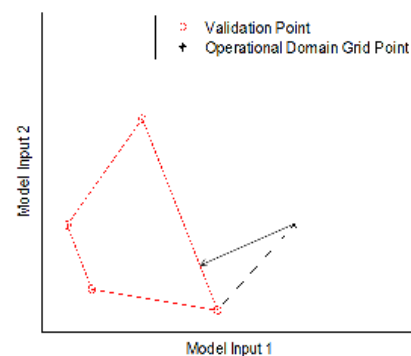
## 2.3 Model Coverage

A simulation model is intended to operate within some domain of operation. This domain of operation is usually limited by the physical bounds of the model inputs; for example, an aircraft system simulation model dependent on aircraft Mach number and altitude is normally not intended to produce valid results outside of the aircraft’s flight envelope. Model coverage is here referred to as a measure of how well the model’s operational domain is covered by a given set of validation experiments (the model’s domain of validity). Four criteria that any model coverage metric should take into account are: conducting validation experiments at an untested model operational point should always improve coverage; diverse validation experiments yield better coverage than clustered validation settings; regions of extrapolation result in a degradation of coverage; the metric should be objective (Atamturktur, Egeberg, Hemez and Stevens, 2015). The sensitivity-adjusted nearest neighbor metric to quantify coverage proposed by Atamturktur, Hemez, Unal and Williams (2009) accounts for all of the listed criteria except the penalization of extrapolation. This suggested metric is modified to include a penalty regarding extrapolation, a metric favoring dispersed

validation experiments over clustered. The resulting coverage description is given as

$$\eta_c = \frac{1}{g} \sum_{i=1}^g \min(d_{E,i}) + d_{ZI,i} \quad (1)$$

which provides an objective measure of model coverage accounting for all four of the previously described criteria (Atamturktur et al., 2015). The coverage measure is denoted  $\eta_c$  in Equation 1. The metric decreases with increasing coverage of the operational domain. If every point within the operational domain is validated, then the metric is zero. The total number of grid points within the model’s domain of operation is denoted  $g$ . The coverage metric is normalized by the total number of grid points ensuring that the metric remains unaffected by the resolution of the domain of operation. The area encapsulated by the red line in Figure 2 represents a hypothetical model domain of validity resulting from validation simulations conducted with settings corresponding to the circles in the figure. The distance  $d_{E,i}$  is the length from grid point  $i$  to the closest validation point, see the dashed line in the figure.



**Figure 2:** Operational domain grid point depicted outside of a hypothetical model domain of validity

The distance  $d_{ZI,i}$  denotes the shortest distance between grid point  $i$  and the model domain of validity, see the solid arrow in the figure. This distance serves as an extrapolating penalty, punishing clustered arrangements of validation experiments.

## 3 Validation Measures

The validation measures define how well the model represents the corresponding physical system in terms of its dynamics and statics.

The measures implemented in the developed framework are partitioned into a set of steady-state and dynamic measures. The steady-state measures describe the model’s accuracy with

respect to the true system in steady-state operational points. Steady-state operation is here considered operation where all model inputs affecting observable system states vary within pre-defined bounds, see section 5.6.

The dynamic measures describe the accuracy of modeled dynamic behavior compared to measurement data from the true system. Which dynamic events to consider for validation are determined by the model's intended use.

### 3.1 Signal Level Measures

Typical validation measures used to quantify a model's steady-state validity on a signal level are:

- Absolute errors. The difference between simulated values of validation quantities and the corresponding measured values
- Relative errors. The relative error is defined as the absolute error normalized with the measured value.

Typical validation measures used to quantify a model's dynamic validity are:

- Relative error of overshoot/undershoot during a step in the reference
- Comparisons of simulation and reference data settling times during a step in the reference or during an event pushing the investigated signal away from its set-point
- Comparisons of rise time during a step in the reference or during an event pushing the investigated signal away from its set-point
- Comparisons of oscillations frequencies and amplitudes.

### 3.2 System Level Measures

As described in section 2.3, coverage is a quantity describing to what degree the model has been validated within its domain of operation. Coverage as defined in Equation 1 is in itself a relevant metric quantifying to which extent the model has been validated. Moreover, the metric may be modified to account for experimental

uncertainty (Egeberg, Atamturktur and Hemez, 2013). This modification is the source of inspiration for the system level validation metric implemented in the ECS model's validation methodology

$$\eta_c = \frac{1}{g} \sum_{i=1}^g \min(d_{E,i})(1 + V_E) + d_{Zl,i} \quad (2)$$

where  $V_E$  represents any relevant validation metric, for example the worst value (on a system level) of any static validation measure in each operational point  $E$ . In Equation 2,  $V_E$  is a validation measure that decreases with increasing model accuracy relative to the true system. Keeping  $V_E$  as generic as possible is advantageous if the measure is to be compared in between models. This modified coverage metric significantly reduces the amount of information needed to overview the assessed model validity.

## 4 Industrial Application Example: The Environmental Control System

### 4.1 System Description

At large, the Gripen Environmental Control System provides its subscribers with the desired amount of air, conditioned to the correct temperature and pressure. Subscribers to ECS conditioned air are typically the fuel system, on board oxygen generating system, anti-g system, avionics, etc. These subscribers often have different requirements regarding mass flow, temperature, and pressure of the supplied air. The difference in requirements is addressed as the air is extracted from the ECS, to the individual subscribers, at different points within the system.

The ECS input air is bleed from the engine or the Auxiliary Power Unit depending on the aircraft's operational point. This air is conditioned through a series of heat exchangers, a compressor-turbine set-up, a condenser, a series of valves, and controlling software. A schematic view of the system at hand is presented in Figure 3.

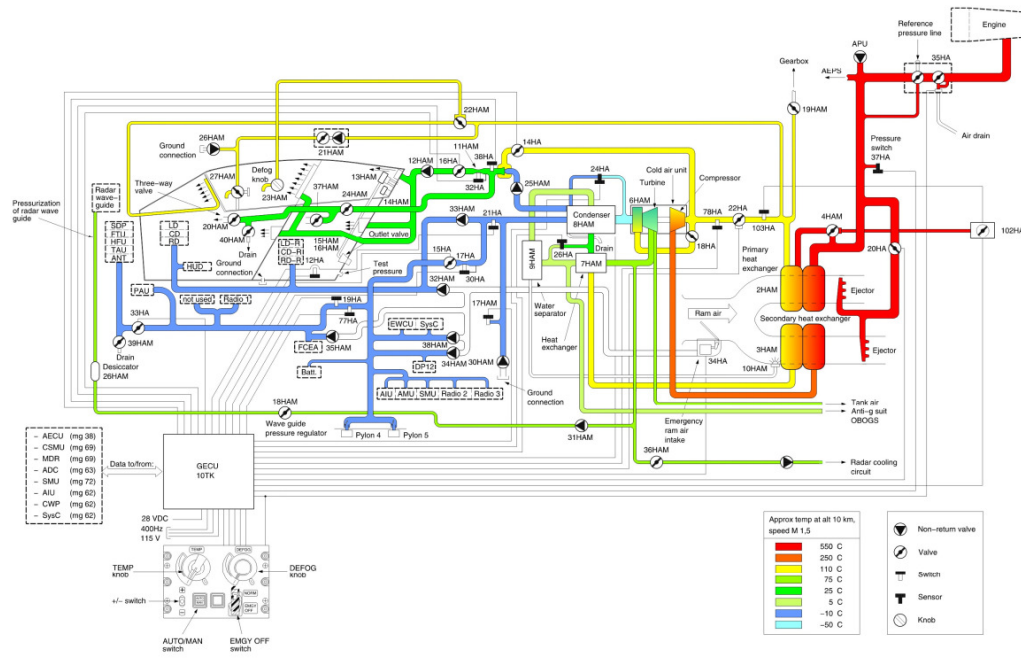


Figure 3: Schematic overview of P-ECS.

### 4.2 Simulation Model

The available ECS models are developed in the Modelica-based modeling tool Dymola. The ECS models were developed implementing the workflow depicted in Figure 1 and described in (Andersson and Carlsson, 2012, Carlsson et al., 2012). A total of three different fidelity level models representing the ECS exist:

1. High fidelity physics-based model for use in the Dymola simulation environment
2. High fidelity physics-based model for export to simulators without real-time performance.
3. Low fidelity model for export to Hardware In the Loop (HIL) and soft simulators with real-time performance.

The intended uses of the three ECS models are defined by a set of use cases. Based on these use cases, the intended uses of the detailed ECS model for use in Dymola can be summarized as follows:

- Conceptual design concerning H/W and S/W
- Export model benchmarking. The high fidelity ECS model should work as data supplier for V&V activities of less detailed system models

- Analysis of system static and transient pressures, temperatures, and mass flows during hardware malfunctions
- Analysis of system static and transient pressures, temperatures, and mass flows during software faults
- Prediction of system static cooling performance and pressure load levels
- Identification of oscillations and transients occurring as a result of software events or rapid changes in flight conditions.

These use cases are general in nature, which causes problems when quantifying the model accuracy bounds specifying if the model under investigation represents the true system “well enough”.

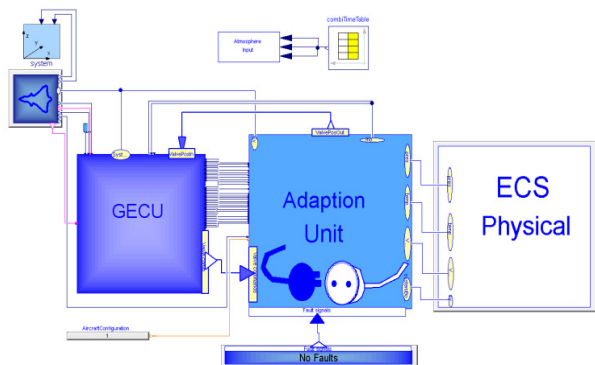
The validation strategies presented in this paper are applied on model no. 1. This particular model is a high fidelity Modelica model intended to represent the true systems statics as well as selected dynamic events. The ECS is a large, non-linear, MIMO system containing strong cross-couplings and widely varying time constants, aspects that are accounted for in the model. The resulting model has more than 100 inputs and 100 outputs, approximately 9000 time varying variables, and 14 non-linear systems of equations. The model is very computationally expensive, not only as a consequence of the previously mentioned characteristics, but also as a result of the incorporated controlling software.

The controlling software is sampled at 30Hz and events are generated at the time of each sample.

Model no. 2 is somewhat simplified compared to no. 1 as it needs to comply with a fixed step solver. The model's time constants are deliberately increased in order to reduce the model's stiffness, which enables the previously mentioned solver type to be used. The primary consequence is that the model's representation of system dynamics is reduced in accuracy compared to the true system and the non-export high fidelity model.

Finally, model no. 3 is designed to represent the system principal behavior in order to achieve real-time performance for use in simulators. Its physics are severely simplified and the model mainly supplies nominal values, depending on operational point, of system key quantities. Even though greatly simplified, this model is of great use during early software development as well as system fault simulations and integration testing.

The closed loop system representation includes sub-models of the system hardware as well as the controlling software developed in Simulink, see Figure 4.



**Figure 4:** Top level of closed-loop ECS Dymola model.

The sub-model *ECS Physical* is a replaceable model representing the physical part of the ECS. Any of the three ECS simulation models defined at the beginning of section 4.2 can be simulated in this closed loop environment as they are declared as replaceable classes with identical interfaces to their surrounding environment.

The controlling software is integrated into the closed loop model by using a static library (\*.lib) file referencing the compiled software object files. The static library is referenced in Dymola through the available annotations feature, which allows the user to link in their own C-libraries (Dassault, 2013).

The component outlined with an aircraft in the left corner of Figure 4 reads data regarding the aircraft's operational point from a comma-separated value (\*.csv) file, see section 5.3 for a more detailed description of how the flight conditions are specified. Atmospheric boundary conditions are incorporated into the closed loop simulation environment through the *systems* component located in the top left corner of Figure 4. This component provides interpolated values of ambient pressure, temperature, and humidity according to pre-defined atmospheric profiles.

The model of the physical parts of the ECS is shown in Figure 5. The model's graphical layout is designed to resemble the schematics of Figure 3 to the extent possible.

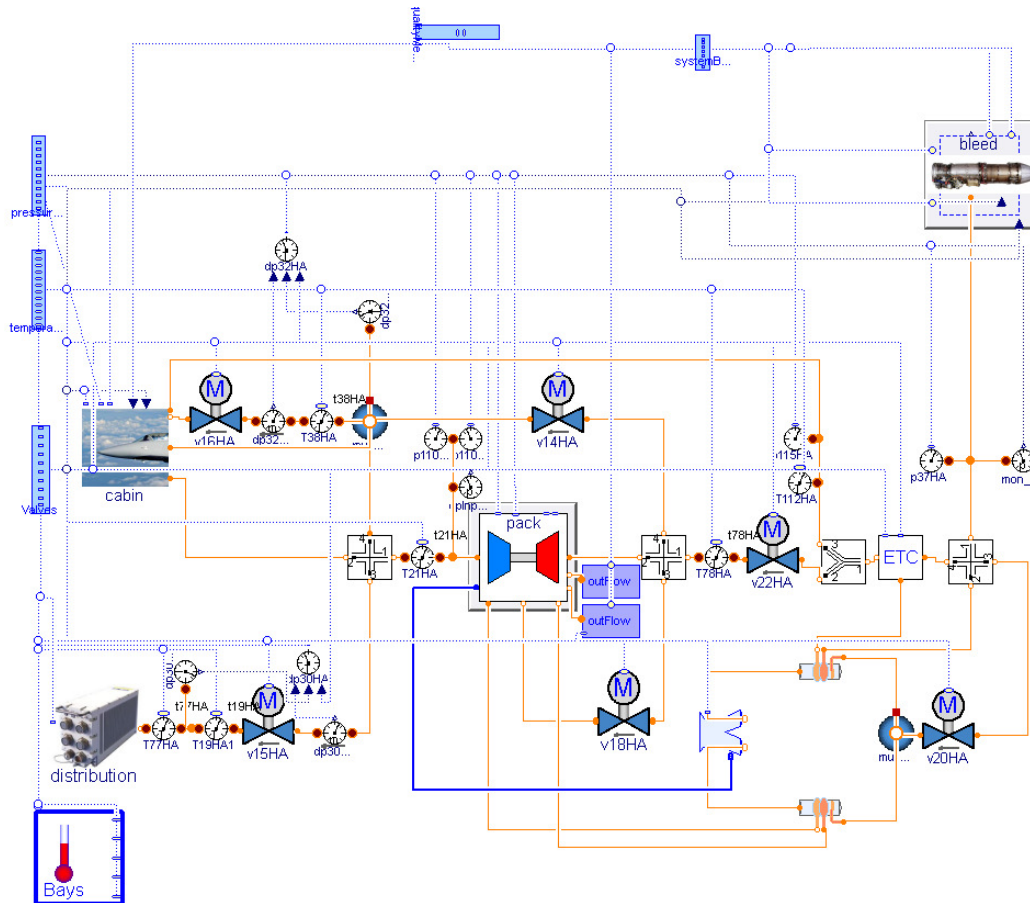


Figure 5: The detailed physical ECS hardware model.

### 5 Proposed Validation Framework

Using the ECS model as a guiding example, this section presents the developed framework for model validation using measurement data. As is, executing a validation simulation and adding the resulting validation points to the domain of validity is a semi-automatic process. The workflow of this process is presented in Figure 6. This workflow is a proposed generic decomposition of the final step *Model Validation* of the model development process presented in Figure 1. The seven different steps are described in detail in this section.

The validation framework is developed in Matlab as it provides a convenient platform for pre- and post-processing of data. Dassault Systèmes supplies m-functions along with the standard installation of Dymola that enables an interface between Matlab and Dymola allowing a complete validation simulation to be executed from Matlab.

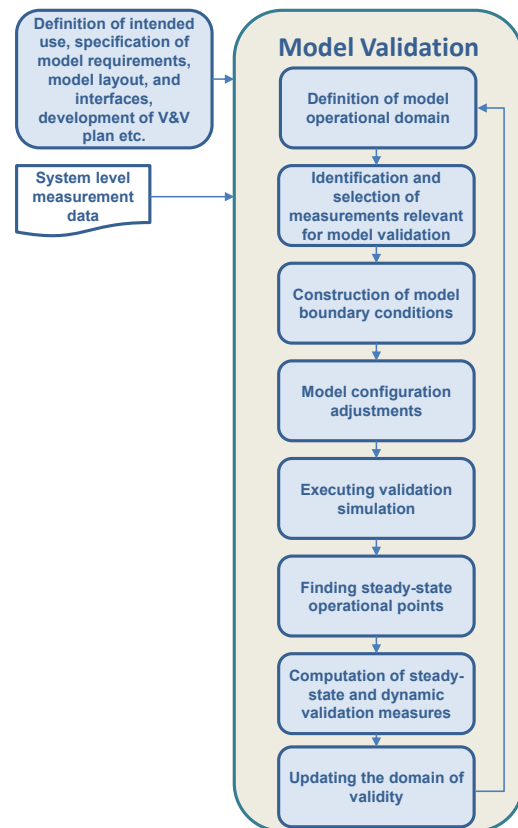


Figure 6: Proposed validation framework.

### 5.1 Definition of model operational domain

The model's operational domain is spanned by all input variables affecting observable system states. Which these model inputs are should be specified prior to any system level validation activity, they are essential when finding relevant reference data and a necessity when computing coverage as well as the system level validation metric presented in section 3.2.

In the case of the high-fidelity ECS model, these inputs are partitioned into atmospheric inputs (ambient pressure, temperature, and humidity) and inputs controllable by the pilot (altitude, Mach number, and Power Lever Angle). Furthermore, the atmospheric conditions are omitted from the definition of ECS operational domain as they are assumed to remain constant during flight at constant altitude. Atmospheric models are used to get the coupling between ambient conditions and altitude, which model is applied is dependent on the test site ambient ground conditions on the day of the test, see section 5.3.

### 5.2 Identification and selection of measurements relevant for model validation

Naturally, measurement data containing the steady-state working points where the model needs to be validated should be used for validation. The ECS needs to be a good system representation throughout the aircraft's operational envelope as the model's intended use does not specify especially significant areas of the system domain of operation. Operational points scattered throughout the envelope are therefore an advantage in order to maximize coverage, see section 2.3.

Measurement data including events generating transient phenomena or oscillations in pressures, mass flows or temperatures are relevant to validation. As the model's purpose includes identification of oscillations occurring as a result of software events or rapidly changing flight conditions, model validity during flight conditions knowingly resulting in system oscillations and transients is crucial. A typical event resulting in pressure oscillations propagated throughout the ECS is the switch of bleed supply air from auxiliary power unit to engine. This event, and others like it, are of particular interest.

The available missions feasible for ECS model validation have not been flown specifically to generate ECS model validation data. Even though a plethora of available measurements

exist, they stem from differently configured aircraft with varying measurement set-ups, factors that need to be considered when identifying measurements for validation purposes. These missions are scanned for the steady-state points and dynamic events of interest with the help of tools within the presented framework. The developed Matlab framework includes functions aiding in the identification process, functions presenting maximum/minimum values of model inputs affecting observable system states as well as their maximum/minimum time derivatives. The information regarding time derivatives is used to identify dynamic phenomena, for example steep dives, bleed supply switches, etc. The developed functions also provide the steady-state points found in each scanned mission. The availability of new steady-state points enables computation of the coverage increase connected to validation simulations in these operational points.

The available in-situ measurements contain information regarding pressure levels, mass flows, and temperatures from sensors distributed throughout the system. Like all measured quantities, these measurements are not exact. No extensive effort has been made to quantify the present measurement uncertainties. The information available in sensor data-sheets is investigated and these uncertainties are determined to be small in comparison to the known model errors.

### 5.3 Construction of model boundary conditions

Necessary model boundary conditions are automatically generated from the measurements selected for validation. Information regarding Mach number, Power Lever Angle (PLA) and bleed pressure, and altitude are logged during flight. Engine bleed temperature levels are computed by combining the relation describing adiabatic compression (Burden, 2009), an example of a non-linear addition to the model,

$$\frac{T_1}{T_2} = \left(\frac{P_1}{P_2}\right)^{\frac{\gamma-1}{\gamma}} \quad (3)$$

with known data of compressor efficiency. The inlet and outlet temperatures are denote  $T_1$  and  $T_2$  respectively in Equation 3. The specific heat is represented by  $\gamma$ , and the inlet and outlet pressure by  $P_1$  and  $P_2$ , respectively. Once established, the model input boundary conditions are stored in a \*.csv file which is a convenient format for Dymola to read during model compilation.



As described in section 4, atmospheric boundary conditions are incorporated into the simulation through pre-defined atmospheric models. However, the atmospheric conditions (during any given flight to be used for validation activities) are rarely close enough to any of the available models. Offsets regarding humidity, ambient pressure, and ambient temperature can therefore easily be defined. Ground level data regarding ambient conditions are used to specify the offsets. Such data is available from the Swedish Meteorological and Hydrological Institute (SMHI, 2015).

#### 5.4 Model configuration adjustments

Considering the ECS, model adjustments matching the ECS model to the system configuration used when producing measurements for validation are necessary, both in terms of hardware and controlling software. For example, different aircrafts are equipped with different avionics equipment. This affects the software control set-point of coolant flow as well as the avionics distribution pressure drop characteristics. Such model modifications tuning the model to the specific aircraft configuration should be separated from model calibration against measurements.

#### 5.5 Execution of validation simulation

Simulating a flown mission is done through Matlab using the m-function *dymolaM.m* which executes a command specified as input to the function in Dymola.

The implemented order of succession when executing a mission simulation is as follows:

1. The model is translated along with the boundary conditions generated from in-situ measurements (see Section 5.3) of the considered mission by calling the Dymola function *translateModel*.
2. The result file content is specified through the Dymola function *experimentSetupOutput*. Dymola is capable of storing data regarding variable derivatives, states, inputs, outputs, etc., which if all are enabled result in large result files. If long simulations with high resolution outputs are to be simulated, disabling saving of non-important data is essential in order to avoid memory problems.
3. The mission is simulated through the command *simulateModel*.

The used Dymola functions are described in (Dassault, 2013).

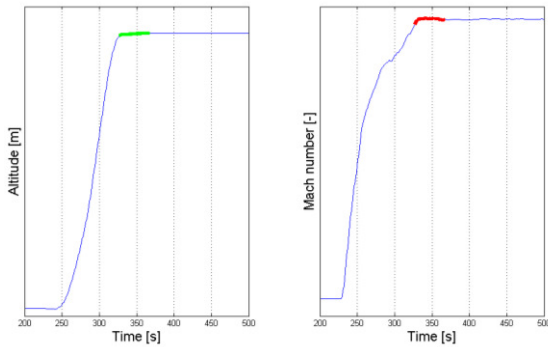
The simulation output is easily read into the Matlab workspace using the *dymload* function. The data structure of the loaded results is somewhat non-intuitive; however, the function *dymget* extracts simulation results for a specific variable from the workspace results.

#### 5.6 Finding steady-state points

Static points are automatically filtered from the measurements. Steady-state points are here referred to as operational points where the conditions on the standard deviations,

$$\begin{aligned}\sigma(Alt) &\leq 10 \\ \sigma(Mach) &\leq 0.009 \\ \sigma(PLA) &\leq 1.5,\end{aligned}\tag{3}$$

of all (by the pilot) controllable closed-loop model inputs: altitude, Mach number, and PLA, are fulfilled. The conditional bounds of Equation 8 are determined through input sensitivity analysis and Subject Matter Experts (SME's) at Saab. A minimum steady-state time span is established which is significantly longer than the system time constants. If the conditions regarding standard deviation are fulfilled for the minimum steady-state time span, this time span is considered to be steady-state. The conditions are computed sequentially in time until they are no longer fulfilled. A continuous steady-state time span is identified by means of a sliding window that monitors the steady-state constraints of Equation 3. The resulting steady-state values are computed as the mean value of the identified continuous time span. Time series plots of altitude and Mach number during part of a flown mission are presented in Figure 7. The conditions on standard deviation regarding altitude are fulfilled for the specified time span (see the left-hand side of the figure); however, the conditions on Mach number are not fulfilled. This particular time span is therefore not defined as a steady-state time span. This filtering algorithm is verified on sets of measurements known to contain static points where it is deemed to fulfill its purpose.

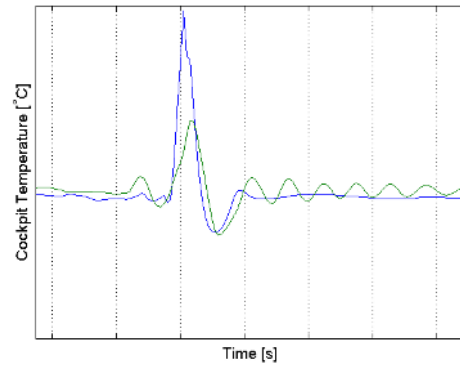


**Figure 7:** 300-second time span of measurements from flown mission used for model validation.

### 5.7 Computation of steady-state and dynamic validation measures

The static signal level and system level validation measures, specified in section 3, are automatically computed (through tools developed within the context of the framework) for each found steady-state operational point along with the coverage metric specified in Equation 1. The validation metric  $V_E$  implemented in the system level metric of Equation 2 is a weighted sum of all relative errors between model and true system. The weight is implemented to emphasize the impact of large relative errors on the metric. A drawback of such a definition is that some subjectivity is introduced to the measure; the metric is therefore also computed omitting the weight. These measures provide a comprehensible overview of the model's static validity.

The model intended use serves as the foundation when selecting time segments for dynamic validation. In practice, specific representative events known to produce dynamic phenomena are investigated as the signal level dynamic measures are computed at manually selected time segments. The post-processing framework plots the mission time evolution of all relevant variables used in the validation process. Transients of interest are found manually from time series plots of measurements and simulation results and the signal level dynamic validation metrics are computed. Such a time series plot of measured and simulated values of cockpit temperature is provided in Figure 8.

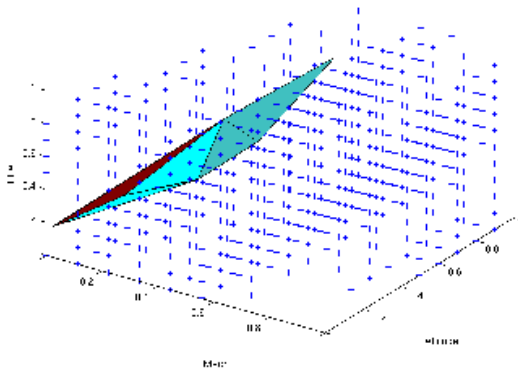


**Figure 8:** Cockpit temperature during part of a flown mission: green represents measurement data, and blue model simulation results.

Since the simulation results of cockpit temperature have a further dampened character than the corresponding measured quantity, comparing settling times is suitable. Other dynamic measures relevant to this particular event are difference in maximum amplitude, and oscillation frequency.

### 5.8 Adding to the model domain of validity

All data relevant for post-processing of a simulated mission is available through the steps described in section 5.1 through 5.7. All computed relative and absolute errors in system mass flows, temperatures, and pressure are appended to a \*.csv file containing all previously validated steady-state operational points, including information regarding model coverage and the computed system level static validation metric. The file contains all the information necessary to formulate the model's steady-state domain of validity. Figure 9 illustrates a static domain of validity plotted on top of the operational domain. This figure provides information regarding model coverage at a less detailed level than the coverage metric specified by Equation 1 as it is unaffected by changes in validation domain density. Such a figure serves as an intuitive complement to the above metric.



**Figure 9:** The static validation domain of the ECS model.

As is, one dynamic domain of validity needs to be specified for each investigated event or dynamic phenomenon. The signal level validation metrics are the sole source of information regarding model dynamic validity. Updating such dynamic domains of validity is a tedious process and requires a significant amount of manual effort and system knowledge. However, no alternative method is implemented at this point in time.

## 6 Discussion and Conclusions

In this paper, a framework for semi-automatic model validation using measurement data from a real-world system is proposed. To ensure industrial applicability, the framework is based on experience from an extensive validation of a detailed system simulation model. The framework covers validation of both steady-state and transient behavior.

In order to reduce the effort required in thorough model validation activities, an increased degree of automation is essential as models need to be repeatedly validated throughout the system development process if well-justified model-based design decisions are to be made. The proposed framework provides a semi-automatic workflow that simplifies the shift towards a more independent V&V approach. Traditional validation measures and techniques suitable for automatic validation have been selected, modified, and implemented in the framework. As a result of the developed methodology, less strict requirements on V&V personnel system expertise are called for during V&V activities. Relieving model developers of repetitive work not only increases objectivity in model validation, but also frees experienced model developers for other tasks that require their knowledge of the modeled system. Parts of the developed framework are currently being used at Saab Aeronautics during validation of

other detailed physics-based simulation models for which measurement data from rig testing exists.

To further increase the degree of automation in the ECS model validation procedure, the selection of an atmospheric model for a specific validation experiment needs to be further automated. No automatic coupling between weather data and model boundary conditions currently exists. Furthermore, the main issues to resolve before the process can be fully automatic are related to selection of aircraft software configuration. There is currently no method to automatically modify the controlling software when matching the model to a specific aircraft configuration.

In addition to the issues listed earlier, improved definitions of dynamic events relevant to validation need to be specified in the model's intended use if the validation of transient behavior is to be further automated. If one or more dynamic operational domains can be clearly and meaningfully specified, there is great potential to automate the identification of dynamic events and the updating of the corresponding dynamic validation domain(s).

The current knowledge regarding present measurement uncertainties in ECS flight data is solely based on supplier information. A pragmatic approach to determining the measurement uncertainties on a more detailed level should be studied in the future.

## Acknowledgements

The research leading to these results has received funding from Saab Aeronautics and the National Aviation Engineering Research Programme (NFFP) jointly driven by the Swedish Armed Forces, the Swedish Defence Materiel Administration (FMV), and the Swedish Governmental Agency for Innovation Systems (VINNOVA), (NFFP6 2013-01211).

The authors would like to thank Anders Järlestål for his invaluable input regarding aircraft environmental control systems and Carl-Philip Forss for his help with refining the developed validation framework

## References

- Andersson, H., Carlsson, M., *Saab Aeronautics Handbook for Development of Simulation Models: Public Variant*, LIU-IEI-R--12/00159--SE, Linköping University, Linköping, Sweden, 2012.
- Arthur, J.D., Nance, R.E., *Verification and validation without independence: a recipe for failure*. 2000

*Winter Simulation Conference*, Orlando, FL, USA, 2000.

Atamturktur, H.S., Hemez, F.M., Unal, C., Williams, B.J., Predictive maturity of computer models using functional and multivariate output, *Proceedings of the 27th International Modal Analysis Conference (IMAC-XXVII)*, Orlando, FL, USA, 2009.

Atamturktur, S., Egeberg, M.C., Hemez, F.M., Stevens, G.N., Defining coverage of an operational domain using a modified nearest-neighbor metric, *Mechanical Systems and Signal Processing*, **50–51**: p. 349-361, 2015.

Burden, T., *Termodynamik med kompressibel strömning*, Vol. Ver. 1.2, Institutionen för mekanik, Kungliga Tekniska Högskolan (KTH), 2009.

Carlsson, M., *Methods for Early Model Validation: Applied on Simulation Models of Aircraft Vehicle Systems*, Tekn. Lic. no 1591, Linköping University, Linköping, Sweden, 2013.

Carlsson, M., Andersson, H., Gavel, H., Ölvander, J., Methodology for Development and Validation of Multipurpose Simulation Models, *Proceedings of the 50th AIAA Aerospace Sciences Meeting*, Nashville, TN, USA, 2012.

Dassault, *Dymola User Manual*, Vol. Vol. 1, Ver. 14, p. 586, Dassault Systèmes, 2013.

DoD, *Department of Defence Directive Number 5000.59*, Under Secretary of Defense for Acquisition, Technology, and Logistics USD(AT&L), US Department of Defence (DoD), 2007.

Egeberg, M.C., Atamturktur, S., Hemez, F.M., Defining coverage of a domain using a modified nearest-neighbor metric, *Proceedings of the Society for Experimental Mechanics Series*, 2013.

NASA, *Standard for Models and Simulations*, NASA-STD-7009, National Aeronautics and Space Administration, Washington, DC, USA, 2008.

Sargent, R.G., Verification and Validation of Simulation Models, *Proceedings of the 2010 Winter Simulation Conference*, Baltimore, MD, USA, 2010.

SISO, *GM-VV Vol. 3: Reference Manual, Reference for Generic Methodology for Verification and Validation (GM-VV) to Support Acceptance of Models, Simulations and Data*, SISO-REF-039-2013, Simulation Interoperability Standards Organization, 2013.

SMHI 2015, *Swedish Meteorological and Hydrological Institute (SMHI)*, [www.smhi.se](http://www.smhi.se).