

# **FLECS: FUNCTIONAL LIBRARY OF THE ENVIRONMENTAL CONTROL SYSTEM – A SIMULATION TOOL FOR THE SUPPORT OF INDUSTRIAL PROCESSES**

**Dieter Scholz<sup>1</sup>, Christian Müller\*<sup>1</sup>, Tim Giese<sup>2</sup>, Carsten Erdmann<sup>3</sup>**

<sup>1</sup>Department of Automotive and Aeronautical Engineering  
Hamburg University of Applied Sciences  
Berliner Tor 9, 20099 Hamburg, Germany

<sup>2</sup>Validation & Verification, Airbus Deutschland GmbH  
Kreetslag 10, 21129, Hamburg, Germany

<sup>3</sup>CeBeNetwork GmbH  
Hein-Saß-Weg 36, 21129 Hamburg, Germany

[info@ProfScholz.de](mailto:info@ProfScholz.de)

## **Abstract**

An Environmental Control System (ECS) in an aircraft ensures the correct air temperature and pressure for passengers and crew. The ECS development process requires the simulation of the ECS system functionality in order to find an optimum system configuration and to support sizing of components. FLECS, the "Functional Model Library of the Environmental Control System", is programmed to support this development process. It offers a library of generic models of ECS components and the aircraft cabin. The library is based on the commercial standard simulation software MATLAB/Simulink. FLECS allows the simulation of a network topology of a system, combining heat and mass flows. Fundamental to the simulation are generalized volume and generalized resistance elements – mass flow resistances as well as heat flow resistances for conduction, convection and radiation. The mass flow of predominantly dry air may contain water vapor and CO<sub>2</sub>. A highly dynamic sample simulation from FLECS components was compared with results from another verified simulation program: Both results were in good agreement. Various Graphical User Interfaces (GUI) in FLECS support everyday use in the industrial environment.

## **1 INTRODUCTION**

**Environmental control system:** Heating, cooling, ventilation, distribution and purification of the air as well as control of temperature, pressure and humidity are the tasks of an environmental control system (ECS) on board of aircrafts. Already in the early stage of the air-

craft development, different system architectures need to be compared. Trade studies need to be performed to evaluate one system architecture against another alternative, in order to be able to choose the most suitable one in the end [1]. The many functions of an ECS require many components, which need to be sized and tuned to each other in an optimum way.

**Simulation:** The ECS development as just described is achieved today with the help of simulation. Three main areas of simulation in the ECS development may be differentiated:

- the calculation of three-dimensional (3D) flow fields in components of the environmental control system or the aircraft cabin by means of Computational Fluid Mechanics (CFD) [2],
- the simulation of one-dimensional (1D) flow in air distribution networks,
- the functional simulation of the environmental control system and the cabin.

An overview of the application of different areas of simulation in ECS is given in [3]. The *functional simulation* is the topic of the project FLECS. It encompasses especially the description of thermodynamics, mechanics and control aspects of the ECS and its components with an emphasis on their dynamics and interaction.

**Functional library:** The project's aim is to support design activities for innovative air conditioning systems in future passenger aircraft. In order to achieve this, a model library based on the commercial standard software MATLAB/Simulink [4] [5] is programmed. Finally, the library will contain generic simulation models of all relevant components that can be found in the air conditioning system and of thermally relevant components in the aircraft cabin. Information on the FLECS project in [6] and [7].

**New technologies** in the area of ECS in aircrafts are e.g. bleedless aircraft, vapor cycle system, new control concepts, increase of cabin air humidity and galley cooling. These new technologies require standard components as well as new ones that are modeled in the FLECS project. In this way a functional ECS simulation is built up that ensures the safe introduction and integration of these new technologies in the aircraft.

**Literature review:** A review [8] showed that many approaches exist of 1D, 3D or functional simulation that have been or could be applied to ECS development. Publications with aircraft ECS applications are e.g. [9] [10] [11]. The goals of FLECS are fully met in none of the contribution found in [8].

## 2 IMPLEMENTATION OF FLECS IN THE INDUSTRIAL PROCESS OF ENVIRONMENTAL CONTROL SYSTEM DEVELOPMENT

### 2.1 Product Development: The V-Model in Aircraft System Design

The development process of an aircraft environmental control system follows the widely used V-model [12]. In general the V-model is associated with software development. The application of the V-model in aircraft system development requires further substantiation that is briefly described here and in *Figure 1*.

Starting with the *aircraft top level requirements*, system **requirements** are derived to fulfill the aircraft top level requirements and to add the necessary level of detail for a complete specification of the respective system. The system specification process comprises three different levels of detail. *High level system requirements*, stating global system behavior in the aircraft context are followed by a *functional system specification*. The functional specification is a breakdown of all functions a system has to fulfill, without giving a solution how this function is to be implemented. The implementation down to signal level (for the software parts of a system) or down to a single piece of equipment (for the hardware parts) is described in the *detail specification*.

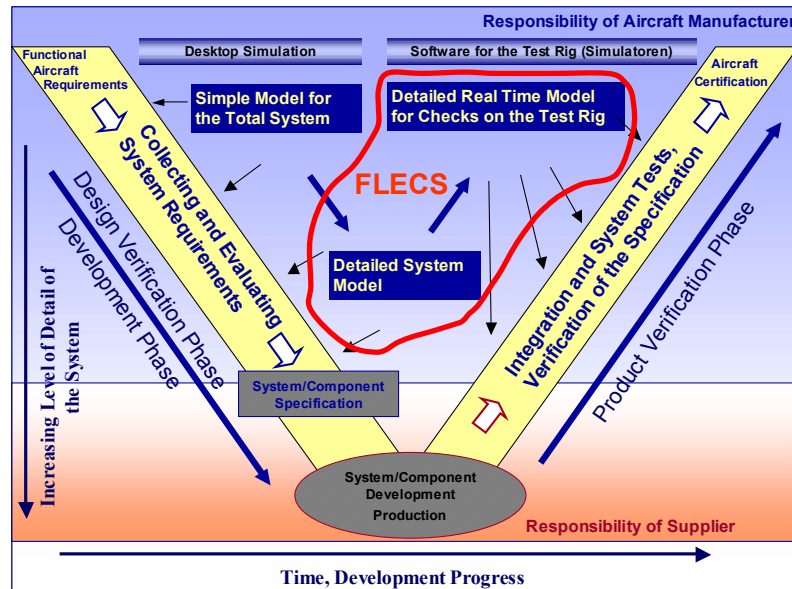


Figure 1 - Simulation in the system development process of an aircraft manufacturer

During the specification process an iterative process of **Design Verification** is carried out, to check if the more and more detailed design fulfills the requirements of the different specification levels. Right after first software and hardware prototypes have been developed, the iterative **Product Verification** processes start. This is to verify, if the now available products comply with the requirements they have been developed to. Usually information and responsibility for a certain task is handed over to another person, department or company. Nevertheless, the final product is integrated into an aircraft, for which the aircraft manufacturer is held responsible.

## 2.2 Functional Simulation

Functional simulation is a mean to support the aforementioned activities in all phases of the design and to track the development progress. To be supportive to the design process, the simulation activities shall be traceable and as seamless as possible. During early aircraft program concept phases there is the need to **investigate a large number of potential system architectures** in a short time frame with the help of simulation, to find the best candidates for an *optimum architecture*. After down selection of system architectures, simulation models need to be further detailed giving a better insight into the expected system behavior. It shall be possible to enhance the simulation model used for **concept studies** by simply adding more detail without changing the architecture of the simulation model. During this detailing process the number of contributors to such a model is increasing, as more technical disciplines get deeply involved (e.g. thermodynamics group as well as controller design group).

To be compatible to such a process **FLECS follows a strictly modular approach**. Sub-components representing generic physical mechanisms or software functions are assembled to components, representing a piece of aircraft equipment or a certain controller function. Components are grouped into component classes covering major aircraft functions to structure the resulting library. Components of the library are available in three different detail levels "*Pre-Design*", "*Simple Dynamic*" and "*Detailed Dynamic*", to cope with the different needs during different design phases. Interfaces of the components are standardized to ensure easy replacement of components with different detail levels.

If the design process enters the phase of prototypes, there is the need of product verification. For **hardware in the loop (HIL)** testing of controllers, simulated equipment is needed

to close the loop. HIL testing takes place in a real-time environment, which needs *real-time capable* simulation models. Such models are integrated on the test rig as C-code, compiled and linked on the target platform. With the chosen simulation tool for FLECS, MATLAB/Simulink, C-Code can automatically be generated. To fulfill real time constraints for the generated C-Code, the user has to use components up to ‘Simple Dynamic’ level. In addition with a small set of overall modeling guidelines, the usage of these components ensures the later use of desktop models for HIL activities, without adding additional burden to the users of the library.

System designers have the need to use simulation models in two different ways: An **interactive mode**, where the model reacts to interactive user inputs. This is especially useful for design verification activities, when models are used to get a better insight of complex system behavior. On the other hand, there is a need to run a large number of predefined test cases in **batch mode**, to ensure proper system functionality in the whole operational envelope of the system. Furthermore it needs to be ensured, that during system development already verified functions are still working properly (regression testing). Within the FLECS project interactive mode as well as batch mode functionality has been developed (see section 4).

As models get more and more complex during system development, **simulation performance** is an issue in a desktop environment. Surprisingly there are fewer performance constraints on HIL test rigs, because the modular structure of the model allows easy distribution of sub-model parts on to different processor cards. A similar principle is applied to the desktop environment: Within the FLECS project, methods are developed to generate C-Code for sub-models, which include the numerical solver for the sub-model. Different integration step sizes are applied for the different sub-models. This approach results in an optimized use of CPU time, as every sub-model is solved with the most appropriate solver and time step. For future applications it is envisaged to run the sub-models on different CPUs, allowing the creation of comprehensive *Software in the Loop* test environment.

### 3 MODELING APPROACH AND IMPLEMENTATION IN MATLAB/SIMULINK

#### 3.1 Component Classes, Components and Models

Each component of the Environmental Control System has to be mapped by a MATLAB/Simulink model block. Several **component classes** were defined for FLECS:

- Ambient conditions
- Aircraft boundaries
- Flow resistances
- Flow and pressure sources
- Volumes
- Area models
- Mixing unit
- Heat exchangers
- Air cycle machine and air compressor
- Ram flow
- Vapor cycle systems
- Sensors
- Controls

Each FLECS component class contains a number of **models** for each aircraft **components** considered. The real world behavior has to be reproduced by a physical description using the appropriate functions. To assure high flexibility and a large application range, modeling has to be as general as possible. This leads to three different **model types** "*Detailed Dynamic*". Simplifications of the models introduced in a second step, lead to the types "*Simple Dynamic*" and "*Pre-Design*" that enhance stability of the simulation and reduce simulation time.

### 3.2 Functions

A dynamic simulation shows the evolution of system parameters with time. The time evolution is expressed by a set of **state equations** (functions). The time series is defined by a time step  $\Delta t$ . At each defined point in time the state equation for each **state variable** has to be solved.

Three different **types of functions** that describe the components have to be distinguished:

1.) Functions that are defined by time-dependant rate equations, which can be expressed in the form  $dy(t)/dt = f(y(t);t)$ . In general  $f(y(t);t)$  is not only a function of  $y(t)$ , but also a function of other time-dependant variables. Equations according to this type are solved by an integrator and are only solvable, if the initial value for each variable is known.

2.) Functions that are defined by quasi steady equation in the form  $y(t) = f(t)$ . These equations are solvable without any knowledge of initial values.

3.) Functions that are of steady state type  $y(t) = const$ .

The interaction of functions of type 1.) and 2.) can cause **stability problems**. Therefore the inner structure of each model block has to be optimized and well tested to ensure numeric stability.

MATLAB/Simulink provides a software platform, which enables to solve functions of type 1.) with two different **solver types**: fixed step solvers and variable step solvers. For the model blocks developed, it was ensured that both solver types work satisfactorily for different processes with different time constants.

**Volumes as well as heat rate and flow resistances** are at the heart of FLECS. They are the **backbone of an ECS simulation**. For this reason, details in this paper are limited to volumes and resistances. They were used to build up highly dynamic simulations. Such simulations with many components were performed to check the stability and real time capability of FLECS

### 3.3 Network Topology

FLECS allows the simulation of a network topology. The network topology is a rather universal topology and includes e.g. parallel and series systems. Model blocks can be linked together with signal lines that carry the state variables. In Simulink, it is necessary to connect two neighboring blocks A and B in the topology by two unidirectional signal lines. One signal line sends a number of output variables from A to B, the second line sends output back from B to A. An example of a **combined heat and mass flow system** is shown in *Figure 2* (Note: signal lines are not shown in *Figure 2*). Model blocks are:

- volumes or nodes (nodes have no mass storage capability),
- flow resistances or heat resistances,
- pressure sources, mass flow sinks (that are boundary conditions of the network).

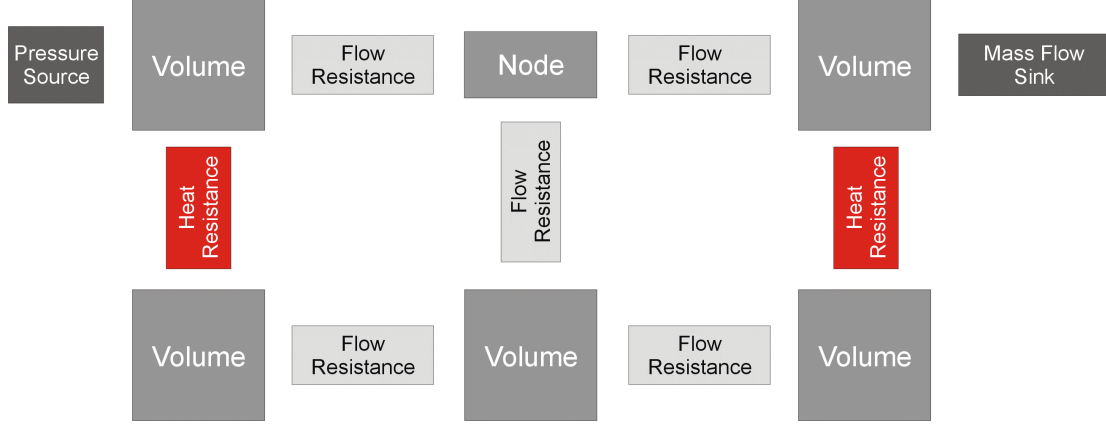


Figure 2 - Example of a network topology of a simulation with heat and mass flows

### 3.4 Basic Volume

The component class "volumes" in FLECS is characterized by constant volumes (isochore systems). State variables are temperature  $T$ , density  $\rho$  and pressure  $p$ .

The physical behavior of volume elements is based on the **first law of thermodynamics**. FLECS is built for applications where the flow velocities are much smaller than the velocity of sound. In such applications, the kinetic energy at the start and at the end state of a system can be neglected. Differences in height are neglected. In such applications, the potential energy at the start and at the end of a system can be neglected. In this case the differential form of the first law of thermodynamics is

$$\frac{dH(t)}{dt} = (\dot{Q}(t) + \dot{H}^{in-out}(t)) + V(t) \frac{dp(t)}{dt} + \underbrace{\left( p(t) \frac{dV(t)}{dt} \right)}_{=0} \quad (1)$$

The heat transfer rate  $\dot{Q}$  is characterized by the external heat transfer processes and will be discussed in Subsection 3.7.  $\dot{H}^{in-out}$  specifies the enthalpy flow, which enters into the system, respectively leaves the system. Eq. 1 can be converted into a **state equation of the temperature  $T$**  using the definition of the specific enthalpy  $h$  ( $h=H/m$ ) and the relation  $dh/dt = c_p \cdot dT/dt$  ( $c_p$ : specific heat capacity at constant pressure):

$$\frac{dT(t)}{dt} = \frac{1}{V \rho(t) c_p} (\dot{Q}(t) + \dot{H}^{in-out}(t)) - \frac{T(t)}{\rho(t)} \frac{d\rho(t)}{dt} \quad (2)$$

The **state equation of the density  $\rho$**  can be derived from the **continuity equation**

$$\frac{d\rho(t)}{dt} = \frac{1}{V} \frac{dm(t)}{dt} = \frac{\dot{m}^{in}(t) - \dot{m}^{out}(t)}{V} \quad (3)$$

The state variables pressure, density and temperature are coupled by the equation of state. For air flow in ECS the ideal gas equation applies. The **state equation for the pressure  $p$**  follows from the **ideal gas law**

$$p(t) = R T(t) \rho(t) \quad (4)$$

### 3.5 Generalized Volume with Air, Water and CO<sub>2</sub>

In general, the airflow is composed not only of dry air, but also of a fraction water vapor and a fraction CO<sub>2</sub>. The input and output variables (*Table 1*) for a volume element have to be adapted to account for all constituents: air, water and CO<sub>2</sub>. The composition of the gas mixture cannot be assumed constant, it is rather a time dependant variable. Therefore the different **contents of water vapor ( $x_{WV}$ ), CO<sub>2</sub> ( $x_{CO2}$ ) and water ( $x_{H2O}$ )** have to be considered separately.  $x_{WV}$ ,  $x_{CO2}$  and  $x_{H2O}$  are the ratio of the mass of the corresponding substance and the mass of the dry air. Hence:  $x_{WV} = m_{WV} / m_{Air}$ ,  $x_{CO2} = m_{CO2} / m_{Air}$ ,  $x_{H2O} = m_{H2O} / m_{Air}$ .

*Table 1 - Input and output variables for a volume with air, water vapor and CO<sub>2</sub>*

input variables	output variables
temperature	pressure
mass flow of air	density
water vapor content	temperature
CO <sub>2</sub> content	mass of air
water content	water vapor content
	CO <sub>2</sub> content
	water content

The pressure inside the volume has to be separated into the **partial pressures** of dry air, water vapor and CO<sub>2</sub>.

The **gas constants  $R$  and specific heat capacities  $c_p$**  of these different gases are taken from [13] (valid under standard conditions):

$$R^{Air} = 287.058 \text{ J/(kg}\cdot\text{K)}, \quad R^{WV} = 461.523 \text{ J/(kg}\cdot\text{K)}, \quad R^{CO2} = 188.924 \text{ J/(kg}\cdot\text{K)},$$

$$c_p^{Air} = 1005 \text{ J/(kg}\cdot\text{K)}, \quad c_p^{WV} = 1870 \text{ J/(kg}\cdot\text{K)} \text{ and } c_p^{CO2} = 830 \text{ J/(kg}\cdot\text{K)}.$$

In the case of **condensation** the gas has a gain of enthalpie  $\Delta h_C$ , in the case of **evaporation** there is a loss of enthalpie  $\Delta h_E$  inside the water. The specific enthalpies of condensation  $\Delta h_C$  and evaporation  $\Delta h_E$  have the same value and are only functions of the temperature. Inside a wet gas an amount of latent heat is stored.

- During the phase change: gas  $\rightarrow$  liquid (condensation) the energy  $m_C \Delta h_C$  ( $m_C$ : mass of condensed water) is released. Therefore the temperature of the gas in a closed adiabatic system increases.
- During the phase change: liquid  $\rightarrow$  gas (evaporation) the free water has to enrage the energy  $m_E \Delta h_E$  ( $m_E$ : mass of evaporated water) and the temperature in a closed adiabatic system decreases.
- *Eq. 5* considers *condensation*: Mixing is thought of consisting of two independent and consecutive steps:
  - Step 1: Mixing of constituents without condensation.  
The resulting temperature is  $T$ .  
Constituents:  $m_{Air}$ ,  $m_{WV}$ ,  $m_{CO2}$ ,  $m_{H2O}$ .
  - Step 2: Instantaneously condensation or evaporation.  
The resulting temperature is  $T' > T$   
Constituents:  $m_{Air}$ ,  $m_{WV}' = m_{WV}^S(T')$ ,  $m_{CO2}$ ,  $m_{H2O}' = m_{H2O} + (m_{WV} - m_{WV}^S(T'))$

$$\begin{aligned}
& \underbrace{(m_{Air} c_p^{Air} + m_{WV} c_p^{WV} + m_{CO_2} c_p^{CO_2}) T}_{\text{term 1}} + \underbrace{m_{H_2O} c^{H_2O} T}_{\text{term 2}} = \\
& \underbrace{(m_{Air} c_p^{Air} + m_{WV}^S(T') c_p^{WV} + m_{CO_2} c_p^{CO_2}) T'}_{\text{term 3}} - \underbrace{(m_{WV} - m_{WV}^S(T')) \Delta h_C}_{\text{term 4}} + \\
& \underbrace{m_{H_2O} c^{H_2O} T' + (m_{WV} - m_{WV}^S(T')) c^{H_2O} T'}_{\text{term 5}}
\end{aligned} \quad (5)$$

**Term 1** is the enthalpy of the gas after step 1 (mixing).

**Term 2** is the enthalpy of the liquid or frozen water after step 1 (mixing).

For liquid water the specific heat is  $c^{H_2O} = 4183 \text{ J/(kg K)}$ .

For frozen water the specific heat is  $c^{H_2O} = 2060 \text{ J/(kg K)}$ .

**Term 3** is the enthalpy of the gas after step 2 (condensation).

The temperature of the gas mixture has risen to  $T'$ .

**Term 4** is the condensation heat.

In good approximation  $\Delta h_C$  is set to the value of 2500000 J/kg.

**Term 5** is the enthalpy of the liquid or frozen water after step 2 (condensation.)

- *Eq. 5* cannot be solved analytically. Only a numerical solution for a given parameter set can be calculated. An iterative setup has to be chosen (see *Eq. 6*)

Estimation :  $T' = T$

$$T_0' \rightarrow m_{WV}^S(T_0') \xrightarrow{\text{Eq.5}} T_1' \rightarrow m_{WV}^S(T_1') \xrightarrow{\text{Eq.5}} T_2' \rightarrow \dots \quad (6)$$

This loop is executed until a certain break condition is satisfied.

- For the *evaporation* process an analogous enthalpy equation can be derived.

### 3.6 The Generalized Resistance

Flow resistances fix the flow properties in the network. These resistance elements are always linked to a block, which has a defined pressure, density and temperature. The occurrence of a pressure drop  $\Delta p$  in the resistance elements involves a certain mass flows  $\dot{m}$ . The enthalpy flow is defined by a mass flows  $\dot{m}$ , the temperature  $T$  and the specific heat of the transported gas mixture. The input respectively output values are shown in *Table 2*.

*Table 2 - Input and output variables for a flow resistance*

input variables	output variables
pressure	pressure
density	density
temperature	temperature
water vapor content	mass flow air
CO <sub>2</sub> content	flow velocity
water content	water vapor content
	CO <sub>2</sub> content
	water content



The **mass flows**  $\dot{m}$  is calculated from the average flow velocity  $v$ . The density  $\rho_{FR}$  of the flow resistance is assumed to remain constant over the whole length of the flow resistance. Therefore Eq. 7 is limited to systems with incompressible flow properties.

$$\dot{m}(t) = A \rho_{FR}(t) v(t) \quad (7)$$

The **average flow velocity**  $v$  is caused by a pressure drop  $\Delta p$  at the apertures that pushes the gas through the flow component with a cross section  $A$ .

$$v(t) = \sqrt{|\Delta p(t)| \frac{2}{\rho_{FR}} \frac{1}{\zeta_G(\text{Re}(t))}} \quad (8)$$

The average velocity  $v$  is dependant on the **minor loss coefficient**  $\zeta_G$ .  $\zeta_G$  can be divided into the friction coefficient  $\lambda(\text{Re}(t))$  of the flow and into a minor loss coefficient  $\zeta$  related to the design of the flow element (see Eq. 9).  $D$  is the diameter and  $L$  the length of the flow element, assuming that the profile is circular. [14] [15]

$$\zeta_G(\text{Re}(t)) = \frac{L}{D} \lambda(\text{Re}(t)) + \zeta \quad (9)$$

The developed flow model is able to simulate a continuous transition between laminar and turbulent flow. The **friction coefficient**  $\lambda(\text{Re})$  is given in [14] as

$$\begin{aligned} \text{laminar flow : } \lambda(\text{Re}) &= \frac{64}{\text{Re}}, \text{ Re} \leq 2320 \\ \text{turbulent flow : } \lambda(\text{Re}) &= 0.3164 \cdot \text{Re}^{-0.25}, \text{ Re} > 2320 \\ \text{mixed flow : } \lambda(\text{Re}) &= \begin{cases} \frac{64}{\text{Re}}, \text{ Re} \leq 1187.384382 \\ 0.3164 \cdot \text{Re}^{-0.25}, \text{ Re} > 1187.384382 \end{cases} \end{aligned} \quad (10)$$

The **Reynolds number**  $\text{Re}$  depends on the average velocity  $v$ .  $\eta$  is the average dynamic viscosity of the air mixture.

$$\text{Re}(t) = \frac{v(t) D \rho(t)}{\eta} \quad (11)$$

The fact, that the average flow velocity  $v$  is a function of the Reynolds number  $\text{Re}$  (see Eq. 8 and 9) but  $\text{Re}$  is also a function of  $v$  (see Eq. 11) requires an **iterative solution**:

$$\begin{aligned} &\text{Estimation : } v_0 \\ &v_0 \rightarrow \text{Re}_0 \rightarrow v_1 \rightarrow \text{Re}_1 \rightarrow \dots \end{aligned} \quad (12)$$

This loop is executed until a certain break condition is satisfied.

Under special conditions in a general flow resistance also compressible flow properties have to be assumed. With the assumption that adiabatic compression respectively adiabatic expansion occurs, the following **compressible mass flow equation** can be used [14] [16].

$$\dot{m} = \frac{A p}{\sqrt{T}} \sqrt{\frac{\gamma}{R}} \text{Ma} \left(1 + \frac{\gamma-1}{2} \text{Ma}^2\right)^{\frac{\gamma+1}{2(\gamma-1)}} \quad (13)$$

The Mach number  $\text{Ma}$  is estimated for a given pressure drop by the incompressible flow equation *Eq. 9* and  $\text{Ma} = v a$  with  $a = \sqrt{\gamma \bar{R} T}$  the speed of sound ( $\bar{R}$ : average gas constant). The pressure and the temperature are taken from neighboring blocks. The specific heat ratio  $c_p/c_V$  is called  $\gamma$ .

### 3.7 Heat Transfer Modeling

Heat transfer units play an important part inside the heat exchangers of an air conditioning system. Three different heat transfer processes can be distinguished: conduction, convection and radiation of heat.

The heat transfer rates for heat **conduction** with a thermal conductivity  $\alpha$  [J/(s·m·K)] is expressed by *Eq. 14*.  $A$  is the effective surface via which the heat transfer takes place. The thickness of the wall is  $b$ .

$$\dot{Q}^{\text{Cond}} = \alpha A \frac{T_1 - T_2}{b} \quad (14)$$

The **radiation** process with the Stefan-Boltzmann constant  $\sigma = 5.67 \cdot 10^{-8}$  J/(s·m<sup>2</sup>·K) and the emissivity  $\varepsilon$ ,  $0 \leq \varepsilon \leq 1$  can be expressed by *Eq. 15*.

$$\dot{Q}^{\text{Rad}} = \varepsilon \sigma A (T_S^4 - T^4) \quad (15)$$

*Equations 13 and 14* can be embedded in quasi-steady model blocks.

**Convection** can be described by a convection heat transfer coefficient  $\lambda$  [J/(s·m<sup>2</sup>·K)]. The heat is transferred from a medium with the temperature  $T_1$  to a surface  $A$  with temperature  $T_S$  and back from the surface to a medium with a temperature  $T_2$ . The heat transfer rates can be expressed by

$$\begin{aligned} \dot{Q}^{\text{Conv},1} &= \lambda A (T_S - T_1) \\ \dot{Q}^{\text{Conv},2} &= \lambda A (T_S - T_2) \end{aligned} \quad (16)$$

*Equations 15* have to be embedded in a dynamic model block, because a dynamic state equation for the temperature  $T_S$  has to be solved.

### 3.8 Example Simulations

With help of Subchapter 3.7 a set of different model blocks can be created. Each model block is saved within the FLECS database (see *Figure 3*, Database). From this database a set of model blocks can be chosen and a simulation can be built up. The mode of action will be discussed by means of two simulations.

In *Figure 3* a simulation of a **dynamic heat exchanger** is shown. The heat exchange consists of two volume blocks and a heat transfer unit. A constant mass flow (1 kg/s) enters respectively leaves the system. The cold side works with a temperature of 293.15 K and the hot side is set to a temperature of 373.15 K. An input mask fixes all the important parameters of the different blocks. A scope window can observe the results. The dynamic of the output temperature (see *Figure 3*, Result) is only given by the heat transfer via the conduction heat

transfer unit. In the case of maximum heat transfer efficiency the output temperature is the average between the cold and the hot side, knowing that a constant enthalpy flow enters the system and both mass flows are equal. For the chosen parameter set the heat transfer efficiency is much lower than 100%. The example given here neglects any thermal capacities of the heat exchanger wall material, usually dominating the dynamics of the heat transfer process. With the fully developed library this could be added easily.

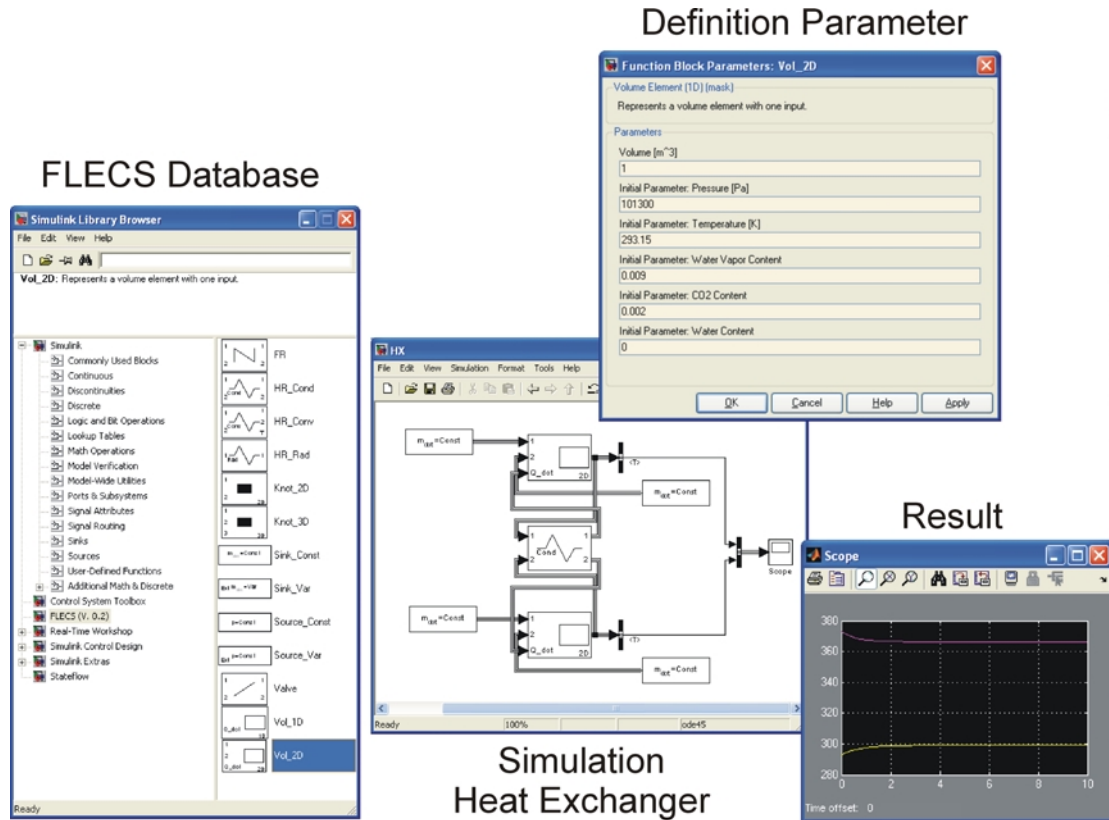


Figure 3 - The FLECS database and a simulation of a dynamic heat exchanger

A **highly dynamic enthalpy flow system** is shown in Figure 4. During such simulation the stability and the dynamic performance of the developed model blocks can be checked. The shown T-joint is composed as a combination of volume elements and flow resistances. Via the output a constant mass flow (1 kg/s) leaves the system. The pressure at the input rises up within 2 s from 320000 Pa to 650000 Pa (see Figure 4, right graph, black curve). As a result the gas mixture inside the system is compressed by the pressure wave. After the pressure has reached the maximum value, the pressure falls back within 2 s to a value of 280000 Pa. As a consequence the gas is able to expand. A mass flow curve moves through the system of the form shown in Figure 4, right graph, red curve. One output of the T-joint is closed by an end piece. The mass flow inside the end piece is shown in Figure 4, left graph, red curve. In the time range marked with a blue circle the mass flow changes the sign. The occurrence of small mass flows normally causes instabilities during a simulation. The model blocks used are optimized so that such systems are stable, independently of what kind of solver type is used.

In order to check of the dynamic behavior of the model blocks, the output temperature is compared with the temperature curve of a Flowmaster [17] Simulation (Figure 5). The setup of the simulation within the two programs is comparable. In contrast to the integration

of differential equations in the FLECS/Simulink model blocks, Flowmaster uses the hydraulic transient analysis method [18].

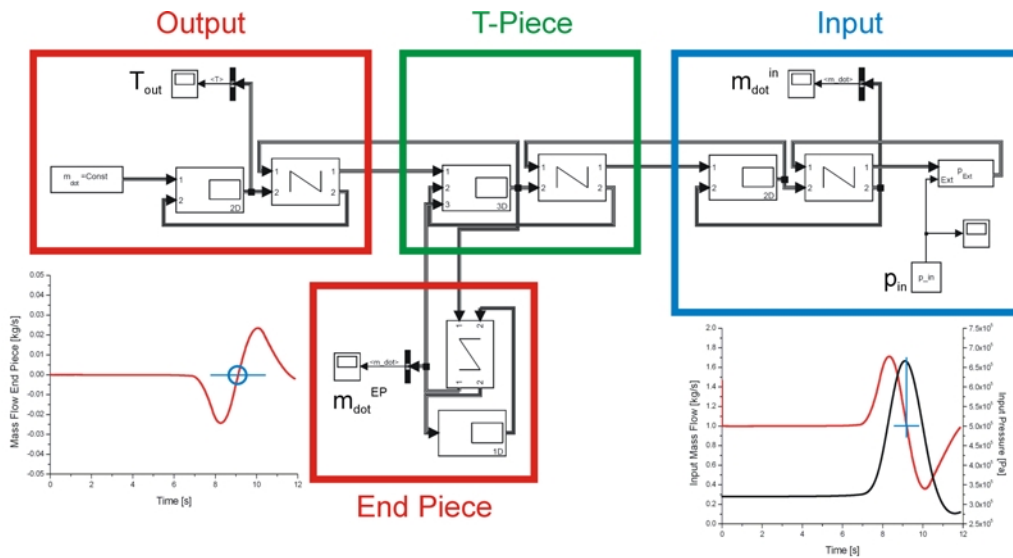


Figure 4 – Simulation of a dynamic enthalpy flow system

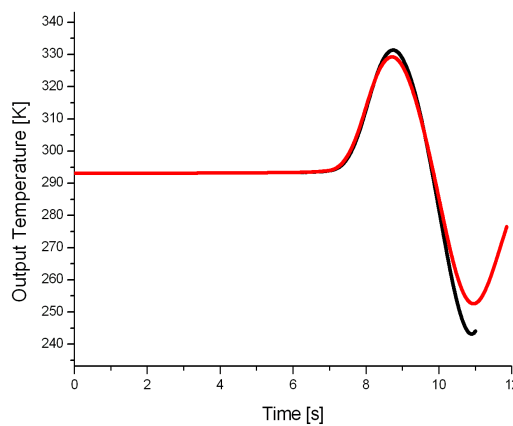


Figure 5 – Comparison of the output temperature of a simulation with the FLECS model blocks (red curve) and a Flowmaster simulation (black curve)

## 4 SOFTWARE ARCHITECTURE AND GRAPHICAL USER INTERFACE

### 4.1 Save-Restore-Init Functionality

In order to provide the user with the necessary convenience for every-day use, one has to provide easy ways to both save the simulation's states in a file at any time as well as load saved states from a file into a simulation. It is possible to save different levels of detail (Figure 6, left):

- only initialization parameters
- the technically relevant values/states
- every value in the whole model, down to internal variables.

The user may want to save a simulation state with different intentions:

- to get a simple snapshot for documentation or analysis
- to save the initial parameters for further configurations of this or similar models
- to save the model's variables when the simulation has reached a steady state (these values can later be used as a starting point for further simulations, e.g. to avoid the system's settling time, for comparison of simulation results after small changes)
- to save the complete simulation state to be able to interrupt the simulation at any time and restart later from the current state.

As a Simulink model is a strictly hierarchical and self-descriptive construct. So it is possible to store all information within this model in a structurally identical MATLAB structure (struct). This is a bijective mapping, so it is easily possible to re-load the data set to reproduce the simulation's state. Additionally, the data can be saved in MATLAB's own binary data format (MAT-file). Thus, it can also be accessed externally via MATLAB, or it may be converted into another data format using MATLAB's internal converters for access in other programs. It is important that the mapping algorithm is flexible and robust enough to cope with differences in the model structure during the loading process. Only those parts of the data tree are updated that are identical in structure and path, all other will be left untouched. Thus, the user can use a data file generated with one model and use it to initialize a second, modified model.

## 4.2 Convenient Library Use in Different Scenarios

Three scenarios are implemented for the use of FLECS. Running a simulation ...

- ... on a desktop computer, e.g. during model development
- ... on a desktop computer, but with higher performance requirements
- ... in a real-time environment (HIL)

**The first requirement** is the standard way to use a MATLAB/Simulink model, so no limitations result here.

**The second requirement** was met by choosing MATLAB's "rapid simulation target" (RST). This compilation target is performance optimized, but is still running on a desktop computer (that is, not in the HIL environment). GUI access to the simulation is still available within the RST.

**The third requirement** is the strictest; it defined the requirements for the complete library development. All coding uses only elements that can be compiled into a HIL target without limitations. Especially we allowed only embedded MATLAB code blocks within the FLECS library.

## 4.3 User Maintainability and User Extendibility

In order to allow the user to not only use but also maintain and modify the FLECS, the user's competences was estimated and the languages and techniques used for coding were limited to those known at the user's site. An automatic documentation system (MATLAB Report Generator) was implemented that keeps the documentation up-to-date with little effort.

## 4.4 Development of Different GUIs

GUI controls had to be developed for 3 types of usage:

1. Simulation controls

2. Simulation of crew interaction
3. Visualization of the temperature distribution in the cabin

**Type 1:** A GUI was developed that allowed access to central control features of a FLECS library model: e.g. initialization, simulation controls, call of models, save functionalities (*Figure 6, left*). The GUI also features a scrollable logging area as well as a status line.

**Type 2:** The GUI mimics the hardware user interface in the plane's cockpit (*Figure 6, right*). As the elements of MATLAB's GUI Development Environment (GUIDE) extension are limited, functionally equivalent elements are chosen, e.g. replacing knobs by sliders.

**Type 3:** A simplified sketch of a longitudinal fuselage cross section is shown with color coded temperature distribution. This is mainly for demonstration purposes and for general tests. No extended user input functionality is present here.

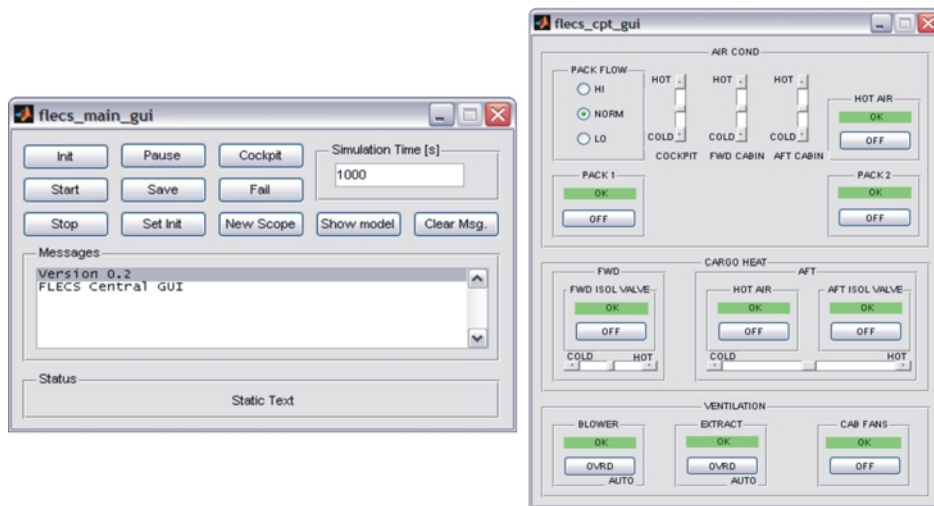


Figure 6 - The FLECS central simulation control panel (left) and the simplified human machine interface (right)

## 4.5 Optimization of Simulation Performance

Future work is planned on an optimisation of simulation performance in the HIL scenario. Big simulation models can tend to become numerically unstable when they run with a variable step solver. A fix step solver, however, may be too slow with the necessary small step size. The fastest sub-model dictates the step size. The solution of the problem consists of running two different sub-models with different solvers. This approach is supported in Simulink.

## 5 SUMMARY

The FLECS database contains generic standard components as well as specialized and detailed aircraft ECS components modeled with MATLAB/Simulink. Algorithms of these blocks are defined in the most general way and are based on equations from thermodynamics. Heat and mass flow systems in a network structure can be simulated. The paper presented systems consisting of volume elements and flow resistances.

Two different simulation setups are discussed: 1.) A dynamical simulation of a heat exchanger shows the interaction of different block types and can be used as example for a cross-linked heat transfer and flow system. 2.) The dynamics and the numerical stability are

checked with a simulation of a T-joint, which is exposed to a steep pressure increase. The result of this simulation is in good agreement with a solution from Flowmaster.

Within the FLECS database an input mask defines each model block. A central GUI that ensures convenient save-restore-init functionality can control the overall simulation. In addition a human machine interface is built, which enhances the user-friendliness of the simulation database.

## 6 REFERENCES

- [1] SCHOLZ, Dieter: Aircraft Systems. In: DAVIES, Mark: *The Standard Handbook for Aeronautical and Astronautical Engineers*. New York : McGraw-Hill, 2003
- [2] WESSELING, Pieter: *Principles of Computational Fluid Dynamics*. Springer : Berlin, 2000
- [3] SCHOLZ, Dieter: FLECS – Funktionale Modellbibliothek des Environment Control System. HWF: Workshop der Initiative Luftfahrtstandort Hamburg (Hamburg, 26th January 2006). Presentation, Hamburg, 06-01-26. - URL: <http://FLECS.ProfScholz.de>
- [4] URL: <http://www.mathworks.com/products/matlab/description1.html> (2007-01-12)
- [5] URL: <http://www.mathworks.com/products/simulink/description1.html> (2007-01-12)
- [6] URL: <http://FLECS.ProfScholz.de> (2007-01-12)
- [7] SCHOLZ, Dieter: FLECS – Funktionale Modellbibliothek des Environment Control System. In: *mobiles*. Dep. of Automotive and Aeronautical Engineering, Hamburg University of Applied Sciences, 2005, p. 113
- [8] WERNER, Tom: *Literaturrecherche : Verfahren und Programme zur Berechnung von Luftsyste-men*, Dep. of Automotive and Aeronautical Engineering, Hamburg University of Applied Sciences, 2006. – URL: <http://bibliothek.ProfScholz.de>
- [9] OEHLER, Bettina: Modeling and Simulation of Global Thermal and Fluid Effects in an Aircraft Fuselage. In: SCHMITZ, G. (Ed.): *4th International Modelica Conference* (Hamburg University of Technology 2005). Hamburg : TUHH, Department of Thermodynamics, pp. 497-506
- [10] ZIEGLER, Shayne; SHAPIRO, Steven: FLOWMASTER : Computer Simulation of an Aircraft Environmental Control System. UKIP MEDIA & EVENTS: *Aerospace Testing Expo 2006* (Anaheim, California, 8th-10th November 2005). – A presentation from the company Flowmaster USA Inc.
- [11] HE, Jun; ZHAO, Jing-quan: Dynamic Simulation of the Aircraft Environmental Control System. In: *Chinese Journal of Aeronautics*, **Vol. 14** (2001), No. 3, pp. 129-133
- [12] URL: <http://de.wikipedia.org/wiki/V-Modell> (2007-01-12)
- [13] TAUSCH, Michael; VON WACHTENDONK, Magdalene: *Chemie SII : Stoff - Formel - Umwelt*. Bamberg : Buchners, 1993
- [14] BOHL, Willi: *Technische Strömungslehre : Stoffeigenschaften von Flüssigkeiten und Gasen, Hydrostatik, Aerostatik, Inkompressible Strömungen, Kompressible Strömungen, Strömungsmess-technik*. Würzburg : Vogel, 2005
- [15] BÖSWIRTH, Leopold: *Technische Strömungslehre*. Wiesbaden : Vieweg, 2005
- [16] URL: <http://www.lerc.nasa.gov/WWW/K-12/airplane/mflchk.html> (2007-01-12)
- [17] URL: <http://www.flowmaster.com/flowmaster/index.html> (2007-01-12)
- [18] WYLIE, E. B.; STREETER, V. L.: *Fluid Transients in Systems*. Englewood : Prentice Hall, 1993