

IMPLEMENTATION OF A FUEL CELL COOLING SUBSYSTEM CONTROLLER REALIZED WITH AN AVIONIC CONTROL UNIT

Lois Picka, Cornelia Bänsch, Dominik Murschenhofer
German Aerospace Center (DLR)

Institute of Engineering Thermodynamics, Energy System Integration
Pfaffenwaldring 38-40, 70569 Stuttgart, Germany
Tel.: +49 711 6862 8514
lois.picka@dlr.de

Abstract

The present paper focuses on the implementation of an avionic control unit, i.e. the integrated modular avionics (IMA), into the cooling subsystem of a polymer electrolyte membrane (PEM) fuel cell system. The implementation involves the control of testbed components ensuring interface compatibility, PI temperature control, serial communication between devices and the programming of the IMA specific operating system. From this, an implementation method is developed that can be applied to the remaining fuel cell subsystems. Further investigations of the IMA controller provide estimations on whether the entire fuel cell system control is feasible on the IMA.

The IMA was implemented as a temperature controller into an experimental setup closely resembling the stand-alone cooling subsystem of a PEM fuel cell system. The necessary interfaces for the temperature control were identified and a PI controller was designed. The influences of the dead time delay of the system, the actuator model inaccuracies, the anti-windup methods and the modeling of the predictor were analyzed and a suitable solution for the design of a fast while precise and stable PI controller was crafted. The programming for the IMA operating system, which requires detailed configuration with resource budgeting and scheduling was completed and testing and analysis of the PI controller was carried out.

The IMA has proven to be capable to control the thermal system and the implementation method is promising to work for other fuel cell subsystems as well.

1. INTRODUCTION

With the rising technology readiness and commercial availability of PEM fuel cell systems, their applicability to zero emission powertrains becomes increasingly promising. Especially in aviation, fuel cell systems can fill the technological gap of powerful, efficient and lightweight zero emission power sources. Yet, their implementation is challenging due to difficult operating conditions at high altitude. Moreover, the fuel cell system including its control unit are subject to stringent safety requirements when it comes to aircraft certification. To address the challenges associated to the control unit, the present work focuses on the implementation of a commercial avionic control system, i.e. the integrated modular avionics, to utilize the certifiability and already established hardware safety. Likewise, the IMA's operating system ensures the required safety for a deterministic and reliable controller software.

2. TESTBED

The PEM fuel cell system can be divided into four subsystems: the anode system for hydrogen supply, the cathode system for air supply, the electric system to monitor and control the electric power and the thermal system for temperature control. To engineer an IMA fuel cell system controller, the controllers for these four subsystems are developed in isolation and merged afterwards.

The first system to be controlled via the IMA will be the thermal system. This subsystem is the first choice, since the

number of components and interfaces is manageable, while on the other hand a variety of the available interfaces is used. The tasks of the controller in this subsystem are similar to what will be needed for the other more complex subsystem controllers. Therefore, the thermal system is a suitable platform to develop an implementation method for the IMA, that can be used for the other subsystems later on.

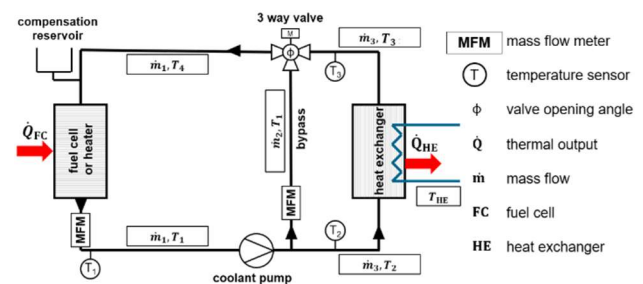


Figure 1 P&ID of the thermal system testbed

The piping and instrumentation diagram (P&ID) of the thermal system testbed is shown in Figure 1. The actuator for the temperature control is the 3-way valve. With it, the coolant flow is divided and directed either through the heat exchanger to cool the system or the bypass, if no cooling is needed. Thus, the cooling power is controlled by the mass flow through the heat exchanger.

The experimental setup, which was realized according to the P&ID in Figure 1, is similar to the cooling system of real fuel cell systems. However, in the current setup the fuel cell

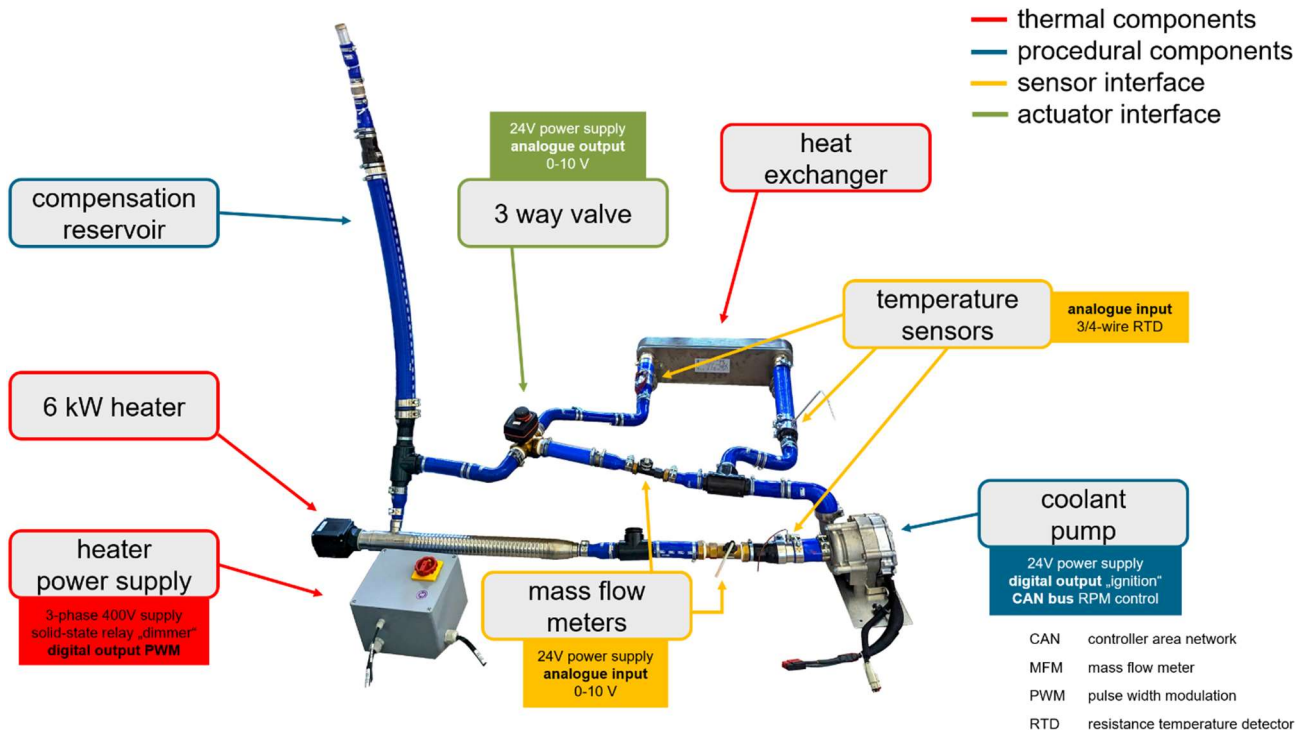


Figure 2 Experimental setup of the thermal system with components and their interfaces to the IMA controller

stack is replaced by a 6 kW electric heater. A picture of the testbed is shown in Figure 2.

Temperature sensors and mass flow meters are installed for feedback control. Although, in the current state of the test bed, the mass flow meters are not yet fully implemented and calibrated. The control temperature, which resembles the heater temperature, is measured closely behind the heater by the temperature probe T1, as seen in Figure 1.

3. PI CONTROLLER

A PI controller is implemented to control the heater temperature. It is a disturbance variable control that reaches and maintains the set temperature. By actuating the opening angle of the 3-way valve, an increase or decrease of coolant mass flow through the heat exchanger results. The cooling power and therefore temperature change is controlled by the PI controller. An actuator model transforms the output of the controller, which is the desired temperature change to reach the set temperature, to the opening angle. The actuator model is described in Section 3.2.

3.1. Control quality

To determine, if the controller improves during the development, the following control quality parameters will be analyzed, as seen in Figure 3:

- The rise time T_r that measures the speed at which 90% of the set temperature is reached
- The settling time T_ϵ , when the set temperature is reached within an acceptable tolerance band of 2ϵ , also if oscillation occurs.
- The height of the overshoot Δh , which should be kept at a minimum
- The permanent control deviation $e(\infty)$

All these parameters contribute to control deviation $e(t)$ with the deviation defined as:

$$e(t) = h_{\text{target}} - h(t), \quad (1)$$

where h_{target} and $h(t)$ denote the setpoint and the actual, respectively.

The integrated squared control deviation J as in Equation (2) can be used to compare controller variations [1]. The lower the value, the higher the control quality and the better the controller, provided that all other specific criteria for control quality parameters are met.

$$J = \int_0^\infty e^2(t) dt \quad (2)$$

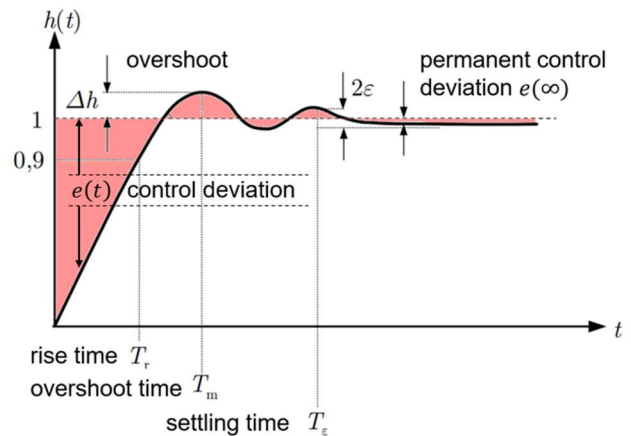


Figure 3 Important parameters to determine control quality with the control deviation $e(t)$ as in Equation (1) with $h_{\text{target}} = 1$. The control deviation over time is marked as red.

3.2. Actuator model

The 3-way valve, which is the actuator in the thermal system, controls temperature change by opening the valve and directing coolant through the heat exchanger. The PI controller therefore adjusts cooling according to a desired temperature change, depending on the temperature deviation from the setpoint temperature. The actuator model transforms the desired temperature change to the corresponding valve voltage.

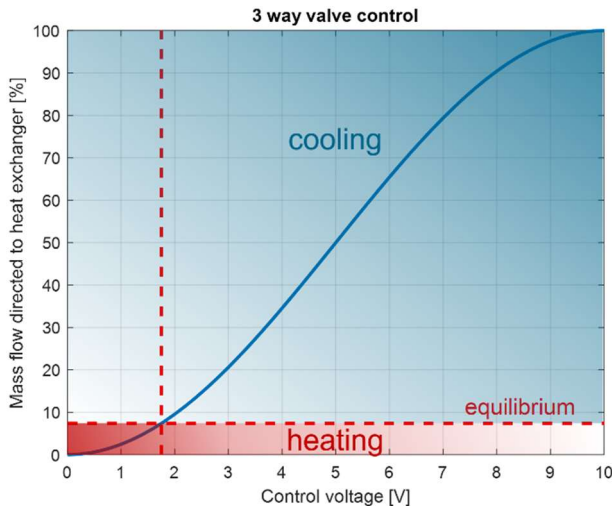


Figure 4 Mass flow directed to the heat exchanger (blue curve) as a function of valve control voltage. The valve voltage of 0 to 10 V corresponds to the full valve opening angle range of 0 to 90°. Equilibrium for 50°C setpoint is at 7.4 % heat exchanger mass flow.

The temperature change depends on the opening angle of the 3-way valve. If the valve is fully closed, i.e. the entire coolant mass flow bypasses the heat exchanger, no cooling takes place and the system is heated by the 6 kW heater. When the valve is fully open, 100% of mass flow is directed through the heat exchanger. In this case, the system is simultaneously heated by the 6 kW heater and cooled with maximum heat exchanger output. Between fully closed and open the heat exchanger mass flow through the valve follows a sinusoidal function according to the datasheet [2]. This function is shown in Figure 4.

Somewhere between the closed and open valve exists an opening angle, where cooling and heating output are equal. This is the equilibrium where the temperature change is zero, also marked in Figure 4 and Figure 5. Since the cooling power of the heat exchanger is not measurable without the mass flow meters, the point of equilibrium is empirically determined. The maximum possible cooling power of the heat exchanger is therefore extrapolated from the equilibrium. Although the estimation of the maximum heat exchanger output is not precise, the valve voltage of the controller output will be close to the equilibrium when approaching the temperature setpoint. The equilibrium differs over temperature and has to be analyzed for different setpoints. The empirically determined equilibrium at the setpoint of 50 °C was at 7.4 % heat exchanger mass flow in Figure 4 or 1.75 V control voltage in Figure 5. The equilibrium is below 10% heat exchanger mass flow, because the heat exchanger in the testbed is oversized compared to the heater.

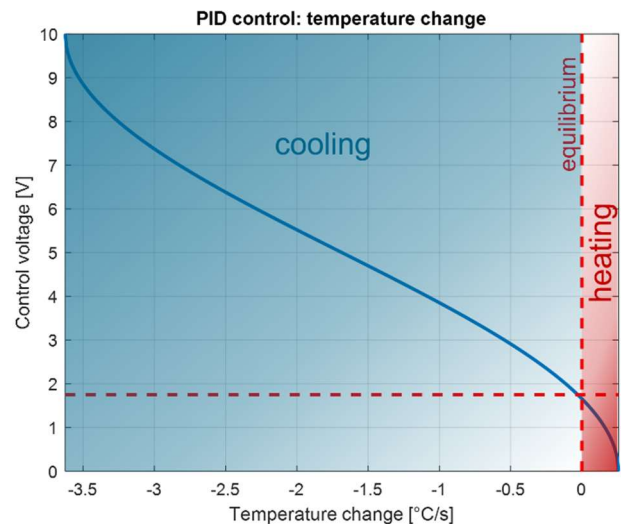


Figure 5 Control voltage of the valve (blue curve) as a function of the desired temperature change of the PI controller. Equilibrium for 50°C setpoint is at 1.75 valve control voltage (15.8 ° opening angle).

The function in Figure 5 shows the required valve control voltage to achieve the desired temperature change by the PI controller. This function is derived from the empirically determined temperature change while only heating and the extrapolated heat exchanger output.

3.3. Dead time

The dead time is the time elapsed between setting the valve voltage and the system's response measured by the temperature sensor. The main effects that contribute to this are the coolant transport delay and the slow valve control time. The valve can turn from fully closed to fully open, i.e. 90°, in 15 s [3]. Especially, when fast valve reactions are needed, e.g. when setting a new set temperature or when reaching the set temperature and a fast reaction is needed to prevent overshooting, the valve turning speed reaches the limit. Another effect is the heating or cooling delay when the heat capacity of components absorb heat until they are on coolant temperature.

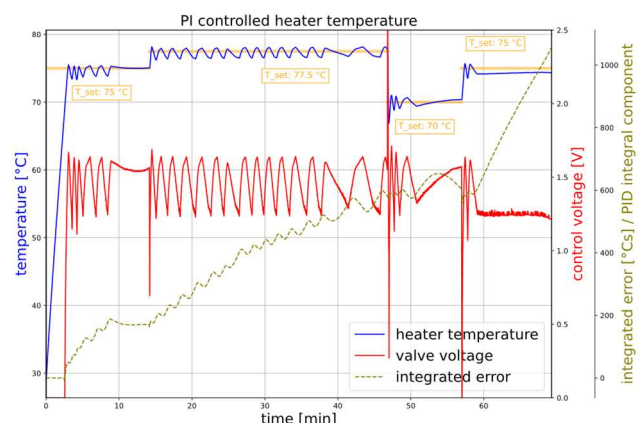


Figure 6 Test run of the IMA controller in the thermal system testbed with temperature (blue, left ordinate), valve control voltage (red, first right ordinate) and the integrated control deviation (olive, second right ordinate), starting with a step response from room temperature to 75 °C set temperature continuing with steps to 77.5°C, 70°C and 75°C set temperature.

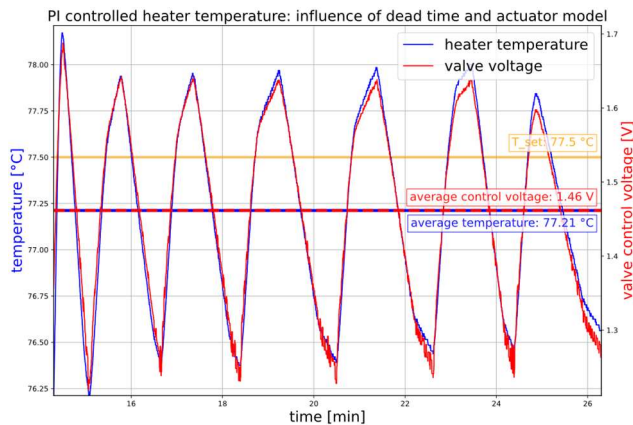


Figure 7 Excerpt of Figure 6. PI controlled temperature (blue, left ordinate) and valve control voltage (red, right ordinate) oscillating around the setpoint temperature of 77.5 °C, caused by instabilities due to dead time and an offset in the equilibrium of the actuator model. The offset in the equilibrium causes the controller to cool more than necessary on average.

The dead time can lead to instabilities, when the PI control gains are too high or the actuator model has inaccuracies. If the PI controller cools or heats too fast to reach the set temperature, but only receives the sensor response that it has overshoot the target temperature after the dead time delay, it will not stop at the setpoint and start oscillating. This oscillation can be seen in Figure 7. The frequency of the oscillation stretches out due to the integral component of the controller. In some cases, the oscillation can be cancelled, if the integral component of the PI controller corrects the average temperature of the oscillating temperature closer to the setpoint temperature, which leads to lesser overreaction of the high gain proportional component until the temperature settles, as seen in Figure 8.

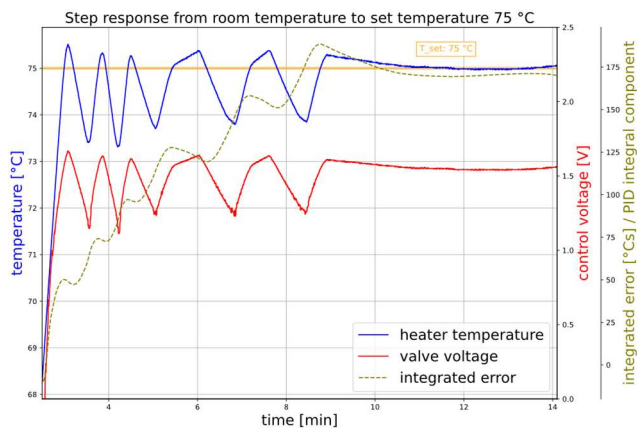


Figure 8 Excerpt of Figure 6. Step function of the PI temperature controlled thermal system from room temperature to 75 °C with the heater temperature (blue, left ordinate), the valve control voltage (red, right ordinate) and the integrated error of the integral component (green, second right ordinate). The increasing integral component of the PI controller causes the temperature to settle.

3.4. Predictor

A predictor is a tool for controller design that can counteract the dead time delay. With the use of a system model, the predictor, if accurate, calculates the future temperature that

will be measured after the dead time delay. Therefore, the system acts as if there was no dead time. In this case, the estimated temperature ahead of dead time will replace the actual value of the control temperature [1].

A simple extrapolation of the actual temperature change for the estimated dead time is used in the model at this stage of controller development. The extrapolation time is a function of the actual temperature gradient becoming lesser when the gradient converges to zero temperature change, i.e. when settling on the setpoint temperature.

3.5. Anti-windup

Integral windup occurs, when actuators reach their output limit and the PI controller, despite not being achievable, demands higher actuator output. The actuator is saturated but the integration of the deviation for the integral component continues. This error integration is disproportional, since the system cannot act as fast in reality as in theory as the integration continues longer until reaching the setpoint. The disproportional growth of the integral component is called windup and takes time to decrease with negative deviation integration after overshooting the setpoint, which causes overshooting and long deviation [4].

This is also the reason why anti-windup has an essential role for the integral component. Anti-windup engages, when the PID controller demands valve voltages, that are out of the specified limits of the valve. This happens either when the valve turning rate is exceeded or when the valve opening angle is at its limit, i.e. for longer fast heating periods when the valve cannot go below 0 V for the fully bypassed heat exchanger. The anti-windup corrects the error integration, if saturation occurs and therefore prevents integral windup.

The tracking anti-windup is the method used in this controller, which subtracts the over the limit exceeding control signal from the integrated error to reduce windup [4]. The block diagram in Figure 9 shows the control loop design with PI controller, actuator model, saturation and anti-windup and the predictor model.

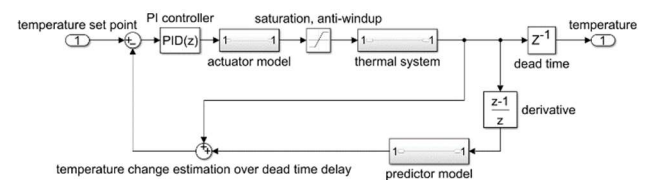


Figure 9 Block diagram of the PI controlled testbed with actuator model, anti-windup, and predictor model

4. INTEGRATED MODULAR AVIONICS

4.1. Implementation

The IMA operating system, on which the controller program will run, ensures deterministic software by a strict configuration process. Program tasks are assigned to so called partitions, which have a prescheduled cycle time and duration, pre-allocated memory and reserved interfaces. That way, partitions cannot interfere with each other to prevent unforeseen errors. A possibility to have IMA internal data exchange is the RAM memory port. The resulting program is therefore deterministic, since all programs and interface usages are scheduled and known before even starting the IMA. In case of changes, i.e. turning on the controller and setting a new temperature setpoint, the

possible change of program is also preplanned. The partitions run within minor frames (MIF), i.e. the time frames within which the partitions are scheduled. In this project the minor frame has a duration of 1 ms, of which approximately 500 μ s can be used for the partitions. The remaining time is needed for the operating system and memory synchronization, as illustrated in Figure 10.

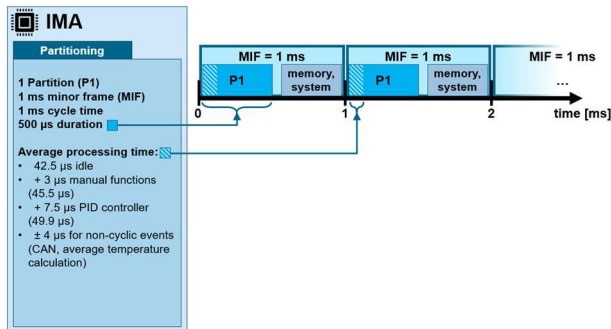


Figure 10 Illustration of the controller partitioning scheduled within the minor frames (MIF) on the IMA

The Programming is done in C and the code is compiled in an IMA specific tool chain, to check for correct configuration before uploading to the IMA. With coordinated configuration, partitions can be developed totally independently. Thus, different partitions of fuel cell subsystem controllers can be developed and combined on the IMA afterwards, if the configurations and resource budgeting matches.

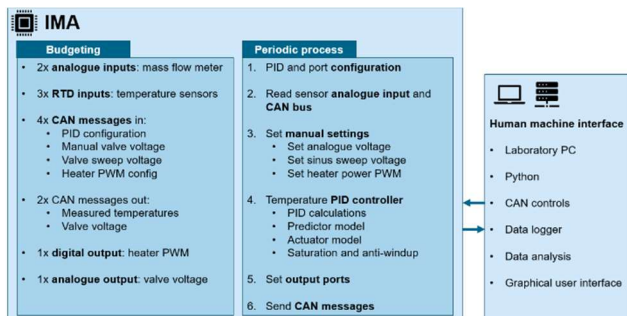


Figure 11 Schematic of the IMA interface usage (left), the IMA program flow (middle) and the human machine interface (right)

The required interfaces between IMA and the testbed, and between IMA and the human machine interface (HMI) are listed in Figure 11. To control the temperature, the analogue output to the 3-way valve and the temperature sensor for feedback is needed. The heater power is set by the IMA with a controlled solid-state relay. Additional sensors, are added for analysis and future control functions.

The HMI runs on a separate laboratory PC and communicates to the IMA via CAN bus. It contains a graphical user interface to monitor sensor and actuator values and a data logger to acquire data for analysis. Temperature set points for the PID controller are sent to the IMA and testing functions for manual valve control or sinus sweeps for dynamic system testing can be applied.

4.2. Limitations

When it comes to the programming of the IMA, the interface usage and scheduling have to be preplanned. Therefore, the following limitations of the IMA have to be considered. Analogue and digital interfaces update every 1 ms cycle

according to the MIF duration. The resulting 1 kHz sample rate is fast enough for this application. Thus, the more relevant limitation is the number of available interfaces used.

Another limitation is the CAN bus usage. The longest possible CAN message with maximum stuffing bits is 129 bits long. At the here used baud rate of 500 kbit/s, a maximum of 3 messages per cycle can be sent without buffer overflow. The messages are sent within one partition every 100 ms. If more communication is needed, e.g. for additional data logging or the addition of partitions, partition scheduling and CAN message cycle time can be redesigned, to optimize the bus usage. The CAN bus is used for compatibility with the test bed and HMI devices. The Arinc 664 protocol can be used alternatively to increase the bandwidth.

Complex, processing intensive software can be difficult to realize within the 500 μ s time frames. This has to be kept in mind when programming actuator or predictor models for the PID controller.

Since the available interfaces, sample rate and CAN communication are within these limits, a single partition is used for the thermal subsystem controller with 500 μ s duration and 1 ms cycle time. The actual duration of one program run is about 42.5 μ s long on average in idle mode, i.e. only reading and writing CAN-data and keeping the outputs constant. The manual test functions add about 3 μ s and the PI controller adds approximately 7.5 μ s processing time. Since CAN messages and sensor data are only sent every 100 ms to the data logger, the program duration varies slightly between partition runs. However, on average the program requires about 50 μ s, i.e. a tenth of the available time frame.

4.3. Partitioning estimation for the entire fuel cell system

The entire fuel cell system will probably exceed the aforementioned limitations. According to the last comparable fuel cell system test bed, the minimally needed interfaces are: more than 20 analog in- and outputs, more than 5 temperature sensors, more than 10 digital in- and outputs and at least 2 CAN busses. A single IMA cannot supply all necessary interfaces. Therefore, either an interface expansion hardware has to be found, with hardware safety according to aircraft requirements or multiple IMA modules will be utilized.

When using multiple IMAs for the fuel cell system, additional communication between the IMAs is needed to connect the subsystem controls, which depend on each other. The Arinc 664 protocol can be used between IMAs for fast data exchange, since the CAN bus is otherwise limited in bus usage and channels. Whether the number of CAN channels is enough, depends on the test bed and controllers. With the additional bus load for data logging and HMI controls, the cycle time will have to be adjusted, to give all acquired data enough time to be transferred. This cycle time adaptation should not be problematic, unless multiple CAN controlled actuators needed a very high sample rate. The software of the entire fuel cell system becomes more complex by introducing message and user event queues that have to be synchronized. A state machine for fuel cell system operation will be necessary. With up to 16 PID controllers, which partially will need underlying models, the processing time increases to over 500 μ s duration, exceeding the time frame of a single MIF. The reduction of partition cycle times frees process time between partition cycles and can be done if stability with lower sample rates

is ensured.

The resource budgeting and scheduling of the partitions can be optimized to evenly distribute program tasks over multiple IMAs. Sorting partitions, that will depend on many interfaces, while others will need more processing time is key to exhaust possible resources. Overall, the usage of multiple IMAs and partitions introduces more overhead in communication and synchronization of the controllers and requires thorough planning.

5. CONCLUSIONS

In the present paper, the implementation of an avionic control unit, i.e. the IMA, to the thermal subsystem of a PEM fuel cell system was investigated. A testbed as shown in Figure 1 and Figure 2 was designed and built, that closely resembles a real fuel cell cooling system, in which the IMA was implemented.

The resulting IMA implementation method is universal and can be applied to other fuel cell subsystems. Although the thermal system controller does not reach the limitations of the IMA, number of interfaces, CAN bus load and processing time were examined to estimate the possible limitations, if the entire fuel cell system controller will be implemented on the IMA. While the additional processing time and CAN bus load can be handled with appropriate partitioning, i.e. efficient programming, scheduling and resource budgeting of the partitions, the number of interfaces for the entire fuel cell system will exceed the available number of interfaces of a single IMA. It is therefore necessary to investigate possibilities for interface extensions that meet the safety requirements for aircraft or to extend the control system to several IMAs.

Data analysis and observations of the system response to different PI control, actuator model, anti-windup and predictor parameters have shown challenges and possible solutions to issues with instability due to dead time delay [5] and control imprecision. The PI controller tests have shown, that empirically determined parameters are not precise enough, if highest precision and speed should be achieved. It is necessary to implement mass flow meters for the system models that require cooling power measurement. With a refined system model, a more advanced predictor like the smith predictor will be a good choice to improve the controllers speed and precision despite a long dead time delay [5]. More details in less significant effects like the heat output to and from the system's thermal capacity can then be further considered in the system model.

The cooling subsystem with its varieties of interfaces and tasks necessary for operation including PI control, is well suited as a model for a universal implementation method. It will continue to be useful for PID temperature controller optimization and testing. The IMA has shown the capability to work as a controller for the thermal system and beside the interface limitation is promising to control additional fuel cell subsystems.

6. OUTLOOK

To improve temperature control precision, the already installed mass flow meters will be integrated into the control loop. With them, the actual cooling power can be measured as additional feedback for the PI controller. Additionally, with a resulting more advanced system model, the smith predictor will be applied to reduce the influence of dead time delay [5].

After improving the temperature PID controller, multi-variable control with valve opening angle and coolant pump RPM will be implemented to enable further optimizations.

Acknowledgements

Thanks to the German Federal Ministry for Economic Affairs and Climate Action for funding the project EnaBle (20M1905).

References

- [1] Lunze, J. (2020), *Regelungstechnik 1*, 12. Edition. Springer Vieweg
- [2] ESBE, (2020), *Stellmotor Serie ARA600* [data sheet]
- [3] ESBE, (2020), *Mischer Serie VRG130* [data sheet]
- [4] Bohn, C. (1995), *An Analysis Package Comparing PID Anti-Windup Strategies*, IEEE Control Systems
- [5] Ingimundarson, A. (2002), *Performance comparison between PID and dead-time compensating controllers*, Journal of Process Control 12, Elsevier Science Ltd.