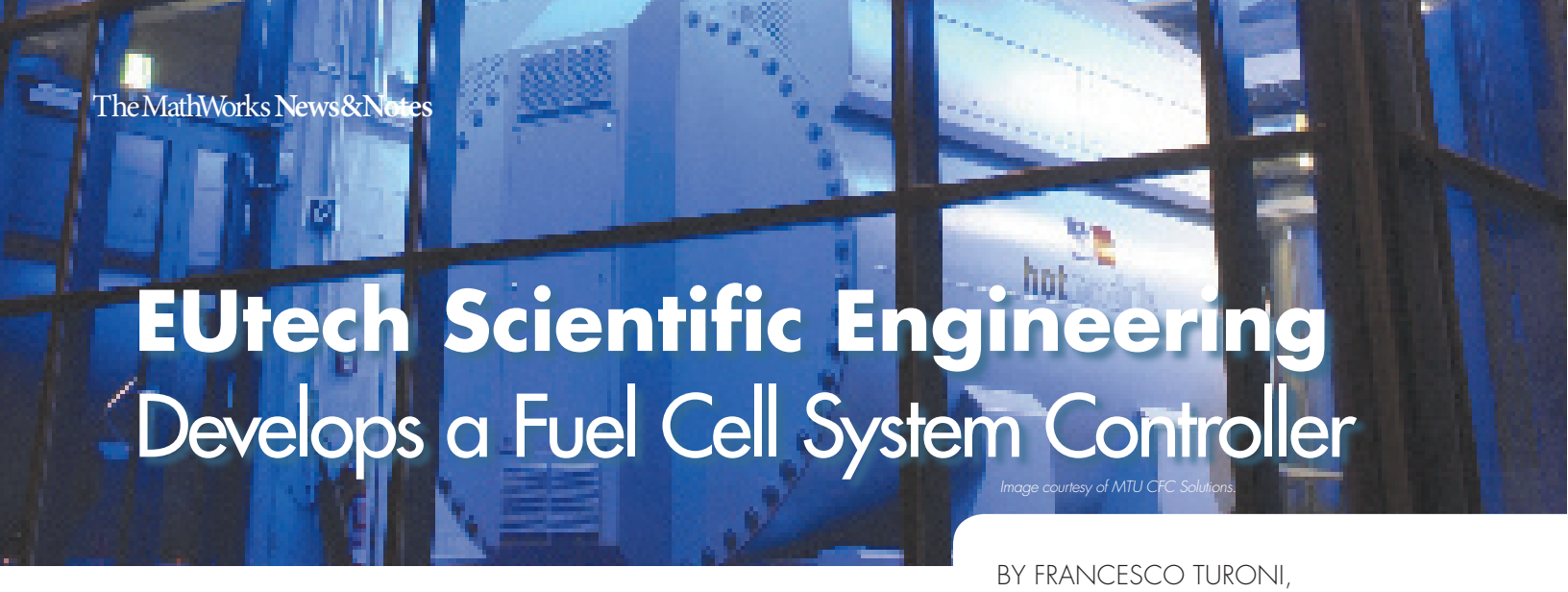


EUtech Scientific Engineering Develops a Fuel Cell System Controller

Image courtesy of MTU CFC Solutions.

BY FRANCESCO TURONI, ABAS SADATSAKAK, MICHAEL MLYNSKI, ALEXANDER HLAWENKA, AND MICHAEL SCHREIBER, EUTECH

To comply with the Kyoto Protocol, energy companies are seeking more efficient power generation strategies. One approach is to combine electricity and heat generation in household micro-cogeneration plants using proton exchange membrane fuel cells (PEM-FCs).

PEM-FCs present complex control challenges. The gas mixture entering the fuel cell must meet strict requirements at all times, and the load dynamics must not affect or destabilize the operation of the stack. Faulty operating conditions can destroy the stack. The gas generation and clean-up units and their heat integration constitute major parts of the fuel cell system, adding complexity to the control design problem.

At EUtech Scientific Engineering, we had to build a fuel cell control system before the fuel cell plant was available, which meant that we could not test the controller in the actual plant under real operating conditions. We overcame this problem by using Model-Based Design and rapid prototyping: We used a Simulink® model of the fuel cell to design the controller. Simulation and automatic code generation enabled us to perform intricate test sequences quickly, repeatedly, and economically.

Modeling the Fuel Cell System

Working in Simulink, we built the fuel cell system using thermodynamic models from FClib, our fuel cell components library. FClib is fully compatible with Simulink and xPC Target, and thus readily supports hardware-in-the-loop (HIL) simulations.

The fuel cell system contains dozens of components. To facilitate development, we divided it into eight functional subsystems:

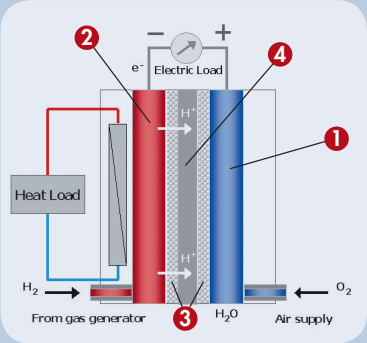
- Burner
- Process media supply
- Steam reformation
- Gas clean-up
- Fuel cell stack
- Electrical system and converter
- Cooling cycle
- Properties

Because we would be using the fuel cell model for evaluation and optimization, we took into account the principles of physical conservation, and included extensive thermodynamic balancing functionality. Mass and energy balances at the component and system level are generated automatically with a step time short enough to capture the plant dynamics.

Designing the Controller

The fuel cell system exhibits strong nonlinear and discontinuous behavior. The sequence control of the start-up proce-

Proton Exchange Membrane Fuel Cell Systems



A PEM-FC system recombines hydrogen and oxygen at the cathode (1), simultaneously inducing an electric current and generating heat. The anode, the negative post of the fuel cell (2), conducts the electrons to an external circuit. At the cathode (the positive post), oxygen is distributed to the surface of the catalyst (3).

The cathode conducts the electrons from the external circuit to the catalyst, where they recombine with the hydrogen ions and oxygen to form water. The electrolyte is the proton exchange membrane (4), a specially treated material that conducts positively charged ions and blocks electrons. Usually, several fuel cells are combined to form a fuel cell stack.

dure runs through several states in which air, steam, and then gas flow through the system. This results in abrupt and discontinuous property changes and requires state-dependent resetting of control parameters. Furthermore, the chemical reactions taking place in the gas generator and the gas clean-up, which are nonlinear in nature, must be taken into account.

Given the complexity of the fuel cell system, we decided to develop the controller in stages, focusing on the individual subsystems and then merging them using a Simulink library of essential control elements. The control system included four functional groups:

- State machine
- Closed-loop controls
- Open-loop controls
- Alarms

The fuel cell stack subsystem and the gas generator operate in many different states. The operating state is determined by the nature of the working fluid (either gas or steam), by load conditions, and by the operating mode (such as normal, start-up, shutdown, and emergency). We modeled the control system's supervisory logic using a state machine developed in Stateflow®.

The various control loops are tightly coupled, but the effects of their interdependence can be reduced by clearly defined state logic. To ensure smooth transitions, we adjusted the control parameters for each state. In most cases we found that a well-tuned PI controller with anti wind-up was sufficient to ensure smooth behavior.

Since the fuel cell stack exhibits a pronounced nonlinear behavior, classical PID controllers would not work for the fuel cell. Therefore we implemented a MIMO scalar fuzzy controller, relaxing the practical tuning requirements. We used state-dependent mappings created in Stateflow to establish the set points (desired states) of the actuators (Figure 1).

The Kyoto Protocol

The Kyoto Protocol is an amendment to the international treaty on climate change, assigning mandatory targets for the reduction of greenhouse gas emissions to signatory nations. Countries that ratify this protocol commit to reduce their emissions of carbon dioxide and other greenhouse gases, or engage in emissions trading.

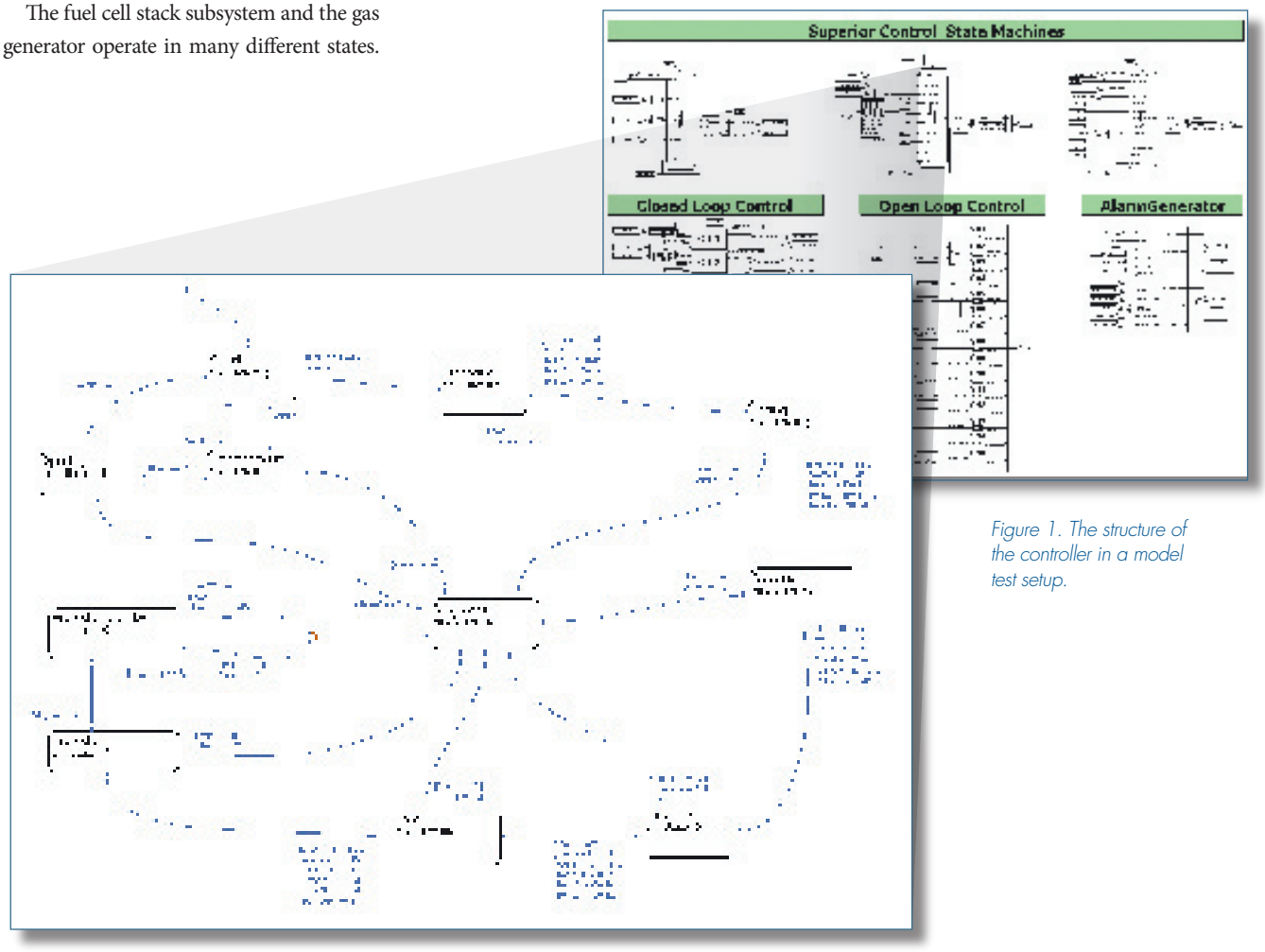


Figure 1. The structure of the controller in a model test setup.

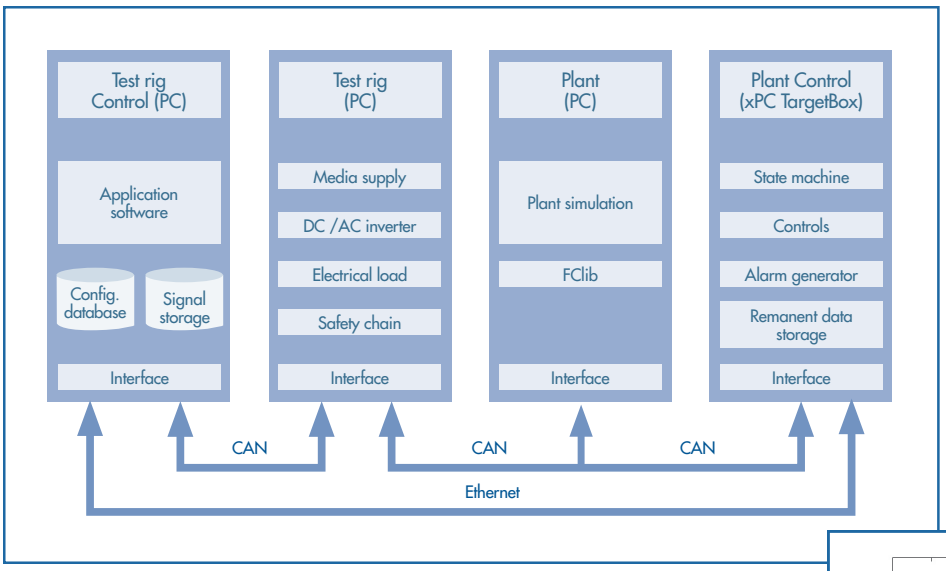


Figure 2. In the HIL setup, control of the fuel cell system (plant) is separated from control of the test rig.

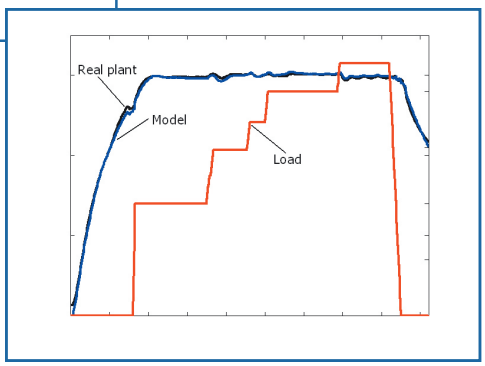


Figure 3. Temperature control simulation results.

Designing the Alarm Subsystem

The alarms not only handle the various emergency conditions; they also ensure that the prototype fuel cell system operates within safe limits under all circumstances. Cost, as well as safety, is an important design consideration: some components are unique and therefore expensive and time-consuming to replace.

To facilitate alarm management, we implemented an alarm-handling system that lets the operator enable, disable, and parameterize the alarm functions online. The application checks all available hardware and software signals and generates an alarm definition table in which the alarm conditions are defined. This powerful functionality is especially helpful when new and complicated systems are tested in the field under severe time restrictions.

Generating Embedded Code

After thoroughly testing the fuel cell process model and its control system, we generated the embedded software using Real-Time Workshop® and Stateflow® Coder. To do this we separated the control system model from the process model along the I/O interfaces. Separation was easy because data conversion blocks transform the signal values of the CAN Bus to physical values, and vice versa.

We compiled and deployed the automatically generated C code using xPC Target, a real-time operating environment. The extended control functionality required powerful target hardware. We selected the PC-compatible xPC TargetBox®. Communication between the control system and process (plant model) was via CAN-Bus, minimizing cabling efforts.

Hardware-in-the-Loop Testing

After building and installing the executable program on the embedded system, we were ready to begin testing the control software. The real process was not yet available, however, and the risk of damaging the expensive prototype hardware with untested software was too high.

The most realistic approach was HIL (Figure 2). For this approach to work, the process model would have to be accurate enough to allow us to test the embedded software under all relevant modes of operation.

We set up the real-time process simulation using a regular industrial PC. Our HIL simulations traversed the control system's entire operational envelope under normal as well as extreme conditions. We executed test scripts automatically in batch runs, saving hours of development time. Code reviews and tests showed that the generated code was efficient.

Our simulation results mapped closely to measured results. Figure 3 shows how well the temperature of the gas generation unit is controlled under extreme load variations.

For More Information

- EUtech
  www.eutech-scientific.de
- Control Design
  www.mathworks.com/res/control_design

Resources

VISIT
www.mathworks.com

TECHNICAL SUPPORT
www.mathworks.com/support

ONLINE USER COMMUNITY
www.mathworks.com/matlabcentral

DEMOS
www.mathworks.com/products/demos

TRAINING SERVICES
www.mathworks.com/training

THIRD-PARTY PRODUCTS
www.mathworks.com/connections

WORLDWIDE CONTACTS
www.mathworks.com/contact

E-MAIL
info@mathworks.com

91480V00 05/07