



A Classic Solution to ARGESIM Benchmark C9R 'Extended Fuzzy Control' using MATLAB / Simulink and Fuzzy Control Toolbox

Igor Škrjanc, University of Ljubljana, Slovenia; igor.skrjanc@fe.uni-lj.si
Felix Breiteneker, Vienna University of Technology, Austria

Simulator: MATLAB is an environment for Numerical vector and matrix computations with a high level programming language. Simulink is MATLAB's simulator, which allows to model a dynamic system by graphical blocks and to simulate it in the time domain. MATLAB and Simulink come with additional tools, Toolboxes and Blocksets, for special purposes/applications. The *Fuzzy Toolbox* supports design, modelling and simulation of fuzzy control in MATLAB and in Simulink. (WWW.MATHWORKS.COM).

Modelling. The model of the overall system is composed in Simulink in a modular manner (Figure 1): submodels for the fuzzy control (Fuzzy Logic Controller) and for the plant (model2tank), reference as table input (Setup h2), disturbance as step function input (Disturbance), and z-transformation blocks for discrete control action. The two-tank system is implemented with standard Simulink blocks (Figure 2); there for the blocks of type Abs, Sign, Switch zero crossing detection is enabled, so that discontinuous changes are synchronised with the ODE solver (ODE23).

The *Fuzzy Logic Toolbox* provides all features to design the fuzzy control. Starting with the basic *FIS editor* (Figure 3) to construct the structure of fuzzy inference system, on command line or within the GUI the fuzzy control can be parametrised and views: *mfedit* to edit membership functions, *ruleedit* is the rule editor and parser, *ruleview* to view the rules and fuzzy inference diagram, and *surfview* which is the output surface viewer.

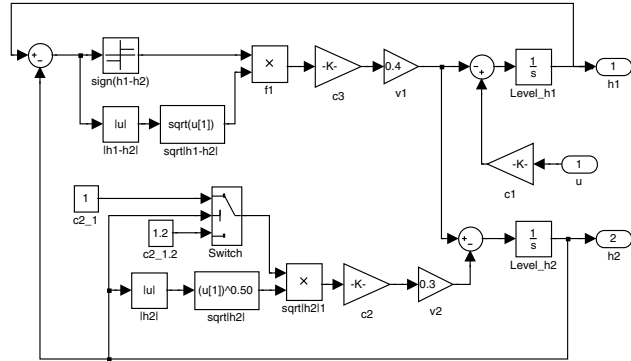


Figure 2: Simulink submodel for two-tank system.

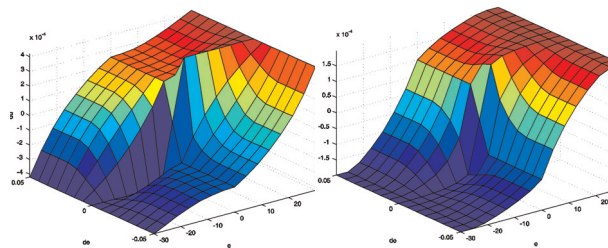


Figure 3: Control surfaces I-FC1 (left) and I-FC2 (right).

A - Task: Controller Surfaces. The fuzzy controller can be used directly in MATLAB. It is not necessary to generate a surface plot by a MATLAB program which calls the defined fuzzy controller, because the FIS editor offers *surfview* for this purpose. Figure 5 shows the control surfaces for I-FC1 and I-FC2: they seem very similar, but they have different scales ($u_{I-FC1} \sim 2 \cdot u_{I-FC2}$); qualitative differences occur on the 'plateau' left above and at the slope at right (different flexion). The fuzzy controller is stored as structure, and all control parameters are stored in a .fis file.

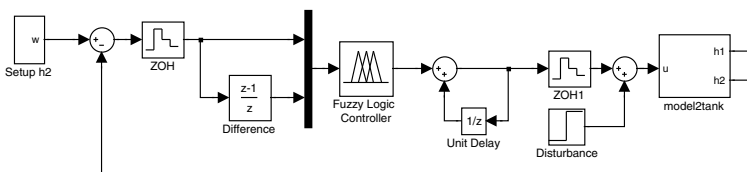


Figure 1: Simulink model of the overall system.

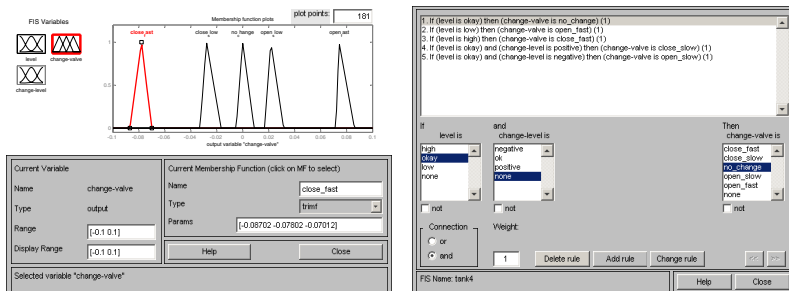


Figure 4: Editor for membership functions and for rule base.

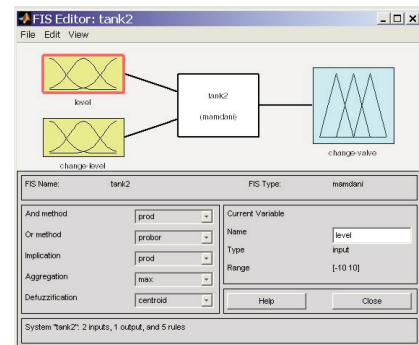


Figure 3: Basic editor for fuzzy control.

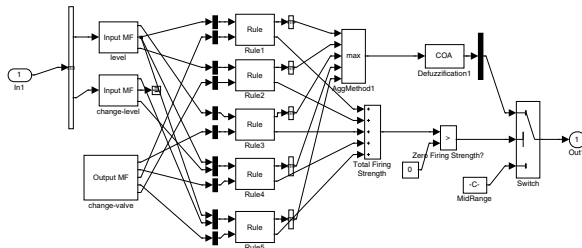


Figure 5: Simulink submodel fuzzy controller, automatically generated from .fis file.

B-Task: Transient Response. In Simulink, the block Fuzzy Controller reads the .fis file and sets up internally a Simulink submodel (Figure 5), again with submodel blocks consisting of basic Simulink blocks and function blocks, so that fuzzification of inputs, inference, and defuzzification for output is done completely at Simulink level (necessary for real-time).

Comparing the results for controllers I-FC1 and I-FC2 (Figure 6 and Figure 7), it can be seen that smooth division of output space in case of I-FC1 results in better control (no overshoot, faster settling time). Both controllers are of integral type and can reject the disturbance, which appear at the input of the process or at the output of the process.

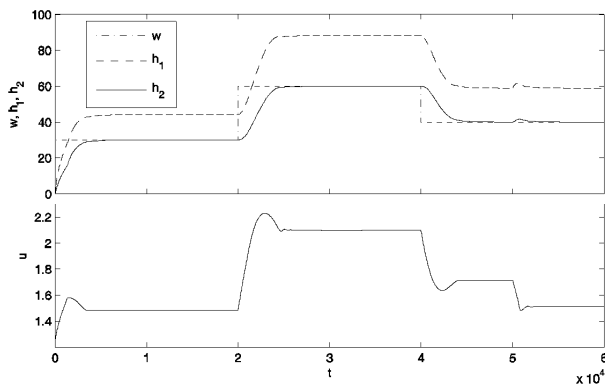


Figure 6: State variables h_1 , h_2 and reference signal (upper) and control signal (lower) over time for I-FC1.

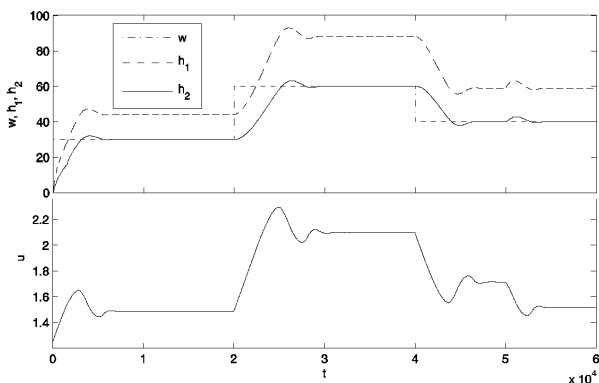


Figure 7: State variables h_1 , h_2 and reference signal (upper) and control signal (lower) over time for I-FC2.

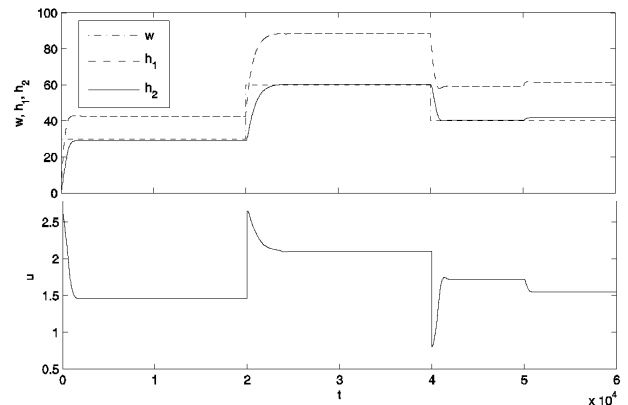


Figure 9: State variables h_1 , h_2 and reference signal (upper) and control signal (lower) over time for P-FC.

C-Task: Comparison with Proportional Fuzzy Controller. The proportional fuzzy controller P-FC is designed as before with the MATLAB *FIS editor*, creating also the control surface for P-FC: Figure 8 shows a much smoother control surface for P-FC. The Simulink model for this control is simpler, because no integral action has to be modelled.

Time response for the P-FC controller is quite different (Figure 8). The control can neither reject the input disturbance nor it can compensate the disturbance at $t = 50.000$ s (control error remains). But P-FC control reacts much faster than I-FC control.

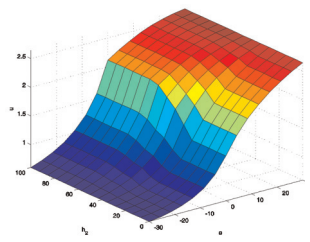


Figure 8: Control surface for P-FC

Résumé: This classic MATLAB / Simulink solution (Rel. 2006a) benefits from the comfortable features of the Fuzzy Control Toolbox. At MATLAB level, the toolbox allows to design all fuzzy controllers and to generate the control surfaces within a GUI. For modelling the plant, and for embedding the discrete control, standard simulink blocks are used, supplemented by the fuzzy control block, which automatically generates the Simulink model for the fuzzy control. Results are documented with MATLAB plot features.

Corresponding Author: I. Škrjanc, igor.skrjanc@fe.uni-lj.si
Laboratory of Modelling, Simulation and Control (LMSC)
Faculty of Electrical Engineering, Univ. of Ljubljana
Tržaška 25, 1000 Ljubljana, Slovenia
Felix Breiteneker, Inst. f. Analysis and Scientific Computing, Vienna University of Technology,
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

Received: June 25, 2006

Revised: October 23, 2006; November 30, 2006

Accepted: December 15, 2006