Filter tuning system using fuzzy logic

Moreira-Tamayo, O.; Pineda de Gyvez, J.; Sanchez-Sinencio, E.

Published in:
Electronics Letters

DOI:
10.1049/el:19940598

Published: 01/01/1994

Citation for published version (APA):
We can conclude from Table 1 that the noise reduction factors of the two alternatives are almost the same, while, as was mentioned earlier, the comb alternative suggested in this Letter is far more efficient (low complexity structure).

Conclusions: A mixed configuration of a simple to implement ESS scheme in FSSAs was introduced. The suggested scheme replaces the error feedback structure with a comb-ESS structure which achieves substantial noise reduction while minimizing the complexity of the error spectrum shaping network, thus making it very attractive to implement.

References
1. GOLD, D., and UR, H.: 'Quantization noise analysis and error feedback implementation in frequency sampling FIR filters', Signal Processing, 1993, 30, pp. 103-113

Filter tuning system using fuzzy logic
O. Moreira-Tamayo, J. Pineda de Gyvez and E. Sanchez-Sinencio

Indexing terms: Fuzzy logic, Fuzzy set theory, Filters, Tuning

The authors describe an expert system for tuning filters using fuzzy logic. The proposed system adjusts the filter components in order to meet the given window specifications. The system is described and experimental results are presented for a lowpass filter implemented with OTAs.

Introduction: Specifications in the design of an analogue filter are often expressed by window constraints. Fig. 1 shows a typical window. Fitting the frequency response within window constraints is complicated for the following reasons:

(i) The adjustable components usually produce nonlinear changes in the frequency response of the filter.

(ii) The variation of one component modifies several characteristics of the filter.

(iii) The implemented circuit contains parasitic components and other nonideal effects.

The proposed system uses a fuzzy logic approach taking advantage of the fact that a window specification can allow any curve as long as it is within the window. Therefore, when the filter is tuned it does not have to match exactly the approximation curve because the filter is tuned to fit within the window. Once it is within the window the system can move the filter to approximate the desired function (optimisation process).

![Fig. 2 Filter tuning system](image)

**Fig. 2** Filter tuning system

Filter tuning system: The proposed method consists of an expert system which allows the tuning of the filter based on the same procedure that an expert would use in practice. A fuzzy logic system of this type requires the measurement of variables to be controlled. In our case, we need to measure the magnitude of the input and output signals. In practice a frequency response plot is obtained by using measurement equipment. The system is simplified by using the minimum number of measurement points in the frequency response. For a second order filter we determined that we need to know the frequency response at only three points: \(|H(w)|, |H(0)|, \text{and } |H(\omega_0)|\). Therefore, three sinusoidal test signals are applied at the frequencies previously mentioned. Fig. 2 shows the system diagram.

![Fig. 3 Second order biquadratic lowpass filter](image)

**Fig. 3** Second order biquadratic lowpass filter

Filter: The case study was a second order lowpass filter implemented with OTAs [5]. Fig. 3 shows the circuit. The transfer function is as follows:

\[
H(s) = \frac{V_o(s)}{V_i(s)} = \frac{\frac{s^2}{2}}{s^2 + \frac{s^2}{2} + \frac{1}{2}} = \frac{\frac{A_o^2}{2}}{s^2 + s\frac{Q}{2} + \frac{1}{2}} \tag{1}
\]

The main advantage of using OTAs for this type of application is that the transconductance can be programmed by the current supplied to the OTA, allowing the filter parameters to be changed. This can readily be seen in eqn. 1. Even though the relation between the supplied current and the transconductance is logarithmic, the system can tune the filter correctly regardless of the non-linearities. It can also be seen in eqn. 1 that by modifying one component several filter parameters are changed, e.g. if either \(c_1\) or \(c_2\) are changed, the filter parameters \(A_o, Q, \text{and } |H(0)|\) are also changed.

Tuning procedure: The dependence that exists between the controllable attribute and the filter parameter. Expert system: In classical logic, sets are defined in a crisp manner, i.e. an element either belongs to a set or does not belong to it. In fuzzy logic, a membership value between 0 and 1 is assigned to each of the elements of the set, where 0 means that an element does not belong to the set and 1 means that the element totally belongs to the set. Fig. 4 shows a set of membership functions used to quantify the deviation of the curve from the window specifications.
The system processes the fuzzy logic quantities by using the Tagaki and Sugeno if-then rules [4]. The conjunction and is computed by obtaining the maximum of the membership values of the variables. The consequence of each rule is a constant term (a singleton). Thus, the defuzzified output is calculated as the weighted average of the individual rule's output. Once the output is obtained the system modifies the filter parameters accordingly. The system then applies the test signals again and repeats the same process until the frequency response is within the window. A control section applies the test signals sequentially and records the magnitude measurements. The expert system was implemented using a digital computer.

**Results:** Fig. 5 shows the results for two different initial conditions of the filter. The output shown was obtained when the filter response met the window constraints. For curve (i) the system tuned the filter after 14 iterations. For curve (ii) the system tuned the filter after nine iterations. If the system continues iterating it tries to approach the optimum points.

**Conclusion:** The proposed system is able to tune second order filters and approximate them to have maximum attenuation in the passband. This system implementation is intended to show that fuzzy logic can be used to tune filters or to tune other electronic circuits or systems provided the availability of a manual procedure.

© IEE 1994

Fast motion estimation with spatial transformation

K.T. Tan, T.N. Goh and M. Ghanbari

**Inducing terms:** Motion estimation, Image processing

A fast search method for motion estimation with spatial transformations is introduced. While the full-search method requires the number of transformation mappings with a displacement of $\pm 1$ of each pixel per frame, this peak-to-peak signal to motion-compensated noise ratio is, on the average, 3dB better than that of the conventional full-search block-matching algorithm.

**Introduction:** Motion estimation based on the block-matching algorithm (BMA) is popular due to its simplicity and efficiency. However, the block-matching method assumes that the motion of each pixel is independent of the others and contains purely uniform translational motion. Hence, any rotational or complex motions cannot be tracked efficiently. In block matching with spatial transformation (BMST) [1], complex motions are compensated for by mapping the current block of pixels onto a quadrilateral in the previous frame using transformations defined by a set of mapping parameters [2]. The parameters of the spatial transformation are obtained by solving a set of equations relating the co-ordinates of the current block to the best-matched quadrilateral in the previous frame [2]. Similar to the BMA, the best-matched quadrilateral is found by displacing each vertex of the previous block, so that the minimum distortion inside a search window is identified. The application of the full-search method in finding the minimum distortion is computationally intensive, because the four vertices of the quadrilateral have to be displaced independently. In this Letter, a fast method for reducing the computational load is presented.

**Computational load:** The search for the best-matched quadrilateral in the previous frame requires the appropriate displacement of the quadrilaterals' vertices. With the full-search method at the vertices of the quadrilateral and constrained to a maximum displacement of $\pm 1$ of each pixel per frame, there are $(2^4 + 1)^4 = 104$ possible displacements for each vertex. For four vertices, there are $4(2^4 + 1)^4 = 8$ possible quadrilateral candidates. The mapping count increases at an exponential rate of 8 and, for $r = 4$ pixels per frame, it approximates to $7 \times 10^5$ mappings per search block. This quadrilateral count is reduced by employing fast-search techniques at the vertices. Similar to the conventional BMA, logarithmic step-search methods can