Signal Restoration in Uncertainty Conditions Based on Evolving F-Transform

Teslenko Nataliya[1]

Control Systems Research Laboratory ,Kharkov National University of Radio Electronics, Lenina av.14, Kharkov, Ukraine.

ntntp@ukr.net

Abstract. *The main idea of appeared in last years F-transform consists in fuzzy partition of initial space to a set of crossed fuzzy subspaces and local approximation of initial function in these subspaces using fuzzy inference means. It was shown in this work that there exists similarity between General Regression Neural Network and F-transform, although the first one concerned to "memory-based" systems, and the second one to "optimization-based" systems. This similarity allowed developing adaptive procedure of discrete F-transform computation. Using of cubic splines as membership functions of adaptive F-transform is proposed in this work. This allows working with a small number of observations, i.e. in situations when the number of points is comparable with number of membership functions. In this case convenient triangular functions don't fit well, because they provide only piecewise-linear approximation and give poor accuracy. For practical tasks when sequential signals processing there may be a priory unknown the interval, on which the function is defined. In addition known a priory interval boundaries could be changed during the object functioning. Therefore, the number of F-transform basis functions will change. In this case it is necessary to adjust all the F-transform components that are associated with additional (added) basis function. So, the inverse F-transform computation procedure becomes evolving, since it adapts the structure to newly incoming signals.*

Keywords

Adaptive F-transform, cubic splines basis function, "memory-based" systems, principle "neurons-at-data-points", unknown function defining interval.

1 Introduction

A lot of real applications associated with nonstationary systems and objects modeling (emulation, control, pattern recognition, prediction) must learn and adapt their parameters and structure continuously in real time, since they interacts with undefined object functioning environment. In this case the modeling of the system must be evolving.

At present time evolving neuro-fuzzy systems are the objects of great attention in the information processing tasks [1]. One of these tasks is the construction of approximating models, where along with conventional regression analysis methods and means of different transformations (Fourier, Laplace) recently appeared F-transform is used [2,3].

The essence of F-transform consists in replacement of continuous function given on closed interval of real numbers by its discrete representation (using direct F-transform). After that using inverse F-transform the discrete representation converts to in the space of continuous functions. The main idea of this transformation is in fuzzy partition of initial space to the set of intersecting fuzzy subsets and local approximation of initial function in these subsets using the means of fuzzy inference.

However, conventional F-transform is intended for stationary signals processing, that are defined on the bounded interval. The boundaries of this interval are known a priory, which restrict considerably its capabilities for real problems solving. It is possible to overcome the lacks of F-transform using the ideas of adaptive and evolving systems of computational intelligence, that use nonconventional learning principles.

2 F-transform basis functions

As a rule, in the tasks dealing with information processing, such as approximation task, the function $f(x)$ is defined in separate points $x(1), y(1), x(2), y(2), ..., x(k), y(k), ..., x(l), y(l)$. In this case discrete F-transform is used in a form [2]

$$F_j(l) = \frac{\sum_{k=1}^{l} y(k)\mu_j(x(k))}{\sum_{k=1}^{l} \mu_j(x(k))} = \frac{\sum_{k=1}^{l} f(x(k))\mu_j(x(k))}{\sum_{k=1}^{l} \mu_j(x(k))}, \; j = 1,2,...,h, \; h \le l, \tag{1}$$

where $y = f(x)$ – original function, $\mu^h = (\mu_1(x),...,\mu_j(x),...,\mu_h(x))^{\mathrm{T}}$ – membership functions of F-transform, that meet the conditions:

$\mu_j(x) : [a,b] \to [0,1]$, $\mu_j(c_j) = 1$;

$\mu_j(x) = 0$, if $x \notin (c_{j-1}, c_j)$, that is membership functions have compact receptive field;

$\mu_j(x)$ are continuous;

$\mu_j(x)$ grow on the interval $[c_{j-1}, c_j]$ and diminish on $[c_j, c_{j+1}]$;

$$\sum_{j=1}^{h} \mu_j(x) = 1. \tag{2}$$

Also notice, that in F-transform [2] the location of nodes c_j in no way doesn't connected with patterns $x(k)$ location as far as they disposed on the uniform lattice. The structure of F-transform itself, if it is examined from a position of neural networks learning, can be referred to the optimization based systems.

In the simplest case, when nodes-centers c_j disposed irregularly, triangular constructions can be used as membership functions in the form

$$\begin{cases} \mu_j^R(x) = \dfrac{c_{j+1} - x}{c_{j+1} - c_j}, \\ \mu_{j+1}^L(x) = \dfrac{x - c_j}{c_{j+1} - c_j}, \\ x \in [c_j, c_{j+1}], \; j = 1,2,...,h-1. \end{cases} \tag{3}$$

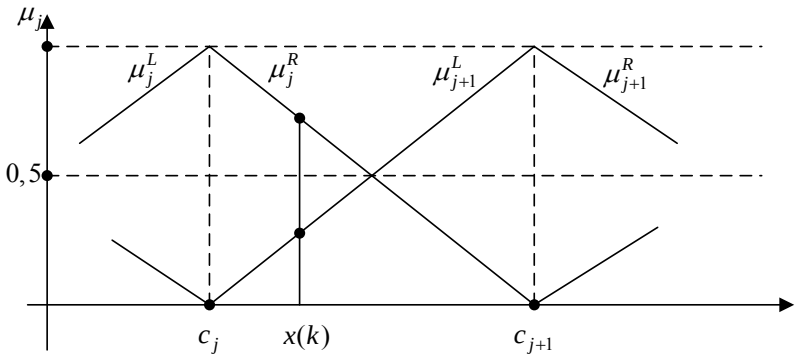


Fig. 1. Triangular membership functions.

In the majority of practical applications it is desirable to approximate experimental points not by a polyline, but by a smooth curve. It is best for these purposes fit cubic splines, i.e. a segments of cubic parabolas of special form. The interval of approximation breaks up into small segments, on which the function defines as cubic polynomials. Approximation with cubic splines allows to work with a small number of observations, i.e. in situations when the

number of points is comparable with number of membership functions. In this case convenient triangular functions don't fit well, because they provide only piecewise-linear approximation and give poor accuracy.

Membership functions were chosen as cubic splines and defined by expressions [4]

$$\begin{cases} \mu_j^R(x) = 0,25\left(2 - 3\dfrac{2x - c_{j+1} - c_j}{c_{j+1} - c_j} + \left(\dfrac{2x - c_{j+1} - c_j}{c_{j+1} - c_j}\right)^3\right), \\ \mu_{j+1}^L(x) = 0,25\left(2 + 3\dfrac{2x - c_{j+1} - c_j}{c_{j+1} - c_j} - \left(\dfrac{2x - c_{j+1} - c_j}{c_{j+1} - c_j}\right)^3\right), \\ x \in [c_j, c_{j+1}],\ j = 1,2,...,h-1. \end{cases} \quad (4)$$

These functions provide Ruspini unity partition, i.e. fulfillment of condition (2).

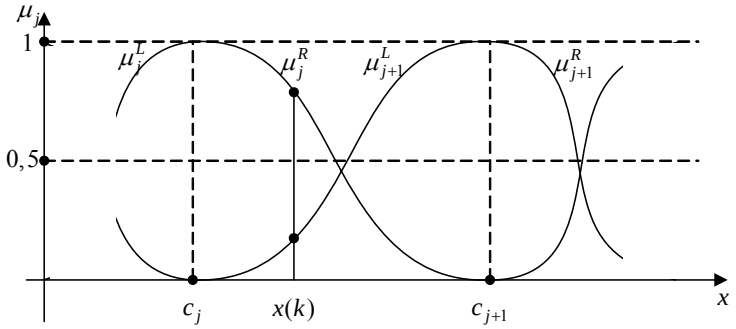


Fig. 2. Cubic splines membership functions.

The first derivatives of these functions are

$$\begin{cases} (\mu_j^R(x))' = -\dfrac{1,5}{c_{j+1} - c_j}\left(1 + \left(\dfrac{2x - c_{j+1} - c_j}{c_{j+1} - c_j}\right)^2\right), \\ (\mu_{j+1}^L(x))' = \dfrac{1,5}{c_{j+1} - c_j}\left(1 - \left(\dfrac{2x - c_{j+1} - c_j}{c_{j+1} - c_j}\right)^2\right). \end{cases}$$

It is obvious, that first derivatives for both functions are continuous on the close intervals $[c_j, c_{j+1}]$ and the second derivatives of these functions are

$$(\mu_j^R(x))'' = (\mu_{j+1}^L(x))'' = -6\left(\frac{2x - c_{j+1} - c_j}{(c_{j+1} - c_j)^3}\right).$$

The second derivatives for both functions are equal and also continuous on the close intervals $[c_j, c_{j+1}]$.

3 Adaptive F-transform

Inverse F-transform defines by the expression

$$\hat{y}_F(k) = f_{F,h}(x(k)) = \sum_{j=1}^{h} F_j(l)\mu_j(x(k)) = (F^h(l))^{\mathrm{T}}\mu^h(x(k)), \quad (5)$$

at that exists such a number of nodes h_ε, that

$$\left| f(x(k)) - f_{F,h_\varepsilon}(x(k)) \right| < \varepsilon, \tag{6}$$

(here ε – arbitrary small positive number), i.e. inverse F-transform approximates the initial function arbitrarily exactly.

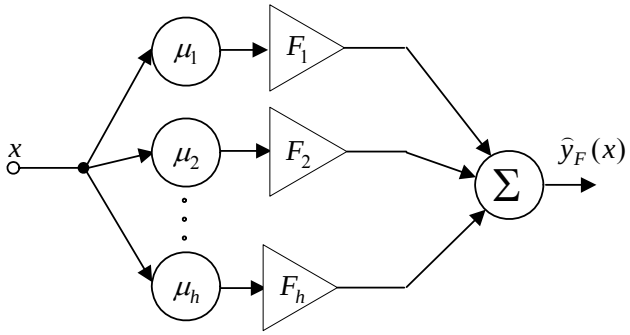


Fig. 3. Inverse F-transform model.

The F-transform technique is characterized by its compactness and convenience, but somehow or other this technique is based on the principle of learning with optimization ("optimization-based" systems) and membership functions disposed on the uniform lattice and do not associate with input data.

The alternative to the systems, learning of which is based on optimization are the systems, that are based on memory or "memory-based" systems. The most known representative of these systems is General Regression Neural Network (GRNN) [5]. The learning of GRNN is realized very easy and consists in one-pass placing the centers of kernel functions in the observation points (principle "neurons-at-data-points").

It is essentially that GRNN can learn in real time, processing the data $x(k), y(k)$, $k = 1,2,...,l,...$, that income sequentially.

GRNN learning process includes adjustment of centers in the form

$$\begin{cases} a_j(k) = a_j(k-1) + y(k), \quad j = 1,2,...,h; \quad k = 1,2,...,l, \\ b_j(k) = b_j(k-1) + 1, \\ c_j(k) = \dfrac{k-1}{k} c_j(k-1) + \dfrac{1}{k} x(k), \quad \text{if} \quad \left| x(k) - c_j(k-1) \right| \le r, \end{cases} \tag{7}$$

$$\begin{cases} a_j(k) = a_j(k-1), \\ b_j(k) = b_j(k-1), \\ c_j(k) = c_j(k-1), \ \text{if} \quad \left| x(k) - c_j(k-1) \right| > r. \end{cases} \tag{8}$$

If the characteristics of function that should be approximated change in time then in procedure (7) one can add exponential forgetting in the form

$$\begin{cases} a_j(k) = \dfrac{\tau - 1}{\tau} a_j(k-1) + \dfrac{1}{\tau} y(k), \quad j = 1,2,...,h; \quad k = 1,2,...,l, \\ b_j(k) = \dfrac{\tau - 1}{\tau} b_j(k-1) + \dfrac{1}{\tau}, \\ c_j(k) = \dfrac{\tau - 1}{\tau} c_j(k-1) + \dfrac{1}{\tau} x(k), \quad \text{если} \quad \left| x(k) - c_j(k-1) \right| \le r, \end{cases} \tag{9}$$

where τ - time constant of exponentially decaying function. The procedure (8) remains without changes.

General regression neural network as a learning result realizes the mapping

$$\hat{y}_G(x) = \frac{\sum_{j=1}^{h} a_j(l)\varphi_j(D_j,\sigma_j)}{\sum_{j=1}^{h} b_j(l)\varphi_j(D_j,\sigma_j)} = \frac{\sum_{j=1}^{h} a_j\varphi_j(D_j,\sigma_j)}{\sum_{j=1}^{h} b_j\varphi_j(D_j,\sigma_j)}, \tag{10}$$

where $D_j = |x_i - c_j|$, σ_j - the width parameter of kernel function.

When the signal $x(p)$, that doesn't belong to learning sample, is feeding on the network input the response appears on its output

$$\hat{y}_G(x(p)) = \hat{y}_G(p) = \frac{\sum_{j=1}^{h} a_j\varphi_j(D_j(p),\sigma_j)}{\sum_{j=1}^{h} b_j\varphi_j(D_j(p),\sigma_j)}, \tag{11}$$

that belongs to the optimal regression curve, which approximates the initial function.

When the signal $x(p)$, is feeding on the F-transform input the response appears on its output

$$\hat{y}_F(x(p)) = \hat{y}_F(p) = \sum_{j=1}^{h} F_j(l)\mu_j(x(p)), \tag{12}$$

that approximates initial function too, but different from (11).

At the same time one can notice that if instead of expression (7) for clusters parameters adjusting use the relations

$$\begin{cases} a_j(k) = \frac{k-1}{k}a_j(k-1) + \frac{1}{k}y(k), & j = 1,2,...,h; \quad k = 1,2,...,l, \\ b_j(k) = \frac{k-1}{k}b_j(k-1) + \frac{1}{k}, \\ c_j(k) = \frac{k-1}{k}c_j(k-1) + \frac{1}{k}x(k), & \text{если} \quad |x(k) - c_j(k-1)| \le r, \end{cases} \tag{13}$$

and instead of kernel functions $\varphi(\cdot)$ - basis membership functions (3), with regard for (2) the transformations (11) and (132) become structurally equivalent [6,7], although the first of them is gotten as a result of applying the procedure based on the memory and the second – as a result of solving optimization problem.

This similarity allowed to develop adaptive procedure of discrete F-transform computation, realize sequential data processing, associate the disposition of membership functions centers with observations disposition and adapt the number of these functions during the learning process. The required approximation accuracy ε and influence area r are the only needed initial data.

As a result of such a system learning there will be generated a finite set of basis functions, the number of which will be unchangeable.

4 Evolving F-transform

For practical tasks when sequential signal processing there may be a priory unknown the interval, on which the function is defined. In addition known a priory interval boundaries could be changed during the plants under consideration functioning. Therefore, the number of F-transform basis functions will change. In this case it is necessary to adjust all the F-transform components that are associated with additional basis function. So, the inverse F-transform computation procedure becomes evolving, since it adapts the structure to newly incoming signals.

When the first sample point is fed to the F-transform input the basis membership function with center $c_1 = x(1)$ is forming and then the inverse F-transform is computed to the single basis function. The next point $x(2)$ is examined for belonging to the influence area of the function with center c_1. If this point is situated out of the influence area of existing basis function, so that the distance from the basis function center to new observation exceeds some influence

radius, that is given a priory ($|x(2)-c_1|>r$), then the second basis function $\mu_2(x)$ is formed, so that $x(2)=c_2$ and F-transform is computed for two basis functions according to the expression

$$F_j(p)=\frac{\sum_{p=1}^{k} y(p)\mu_j(x(p))}{\sum_{p=1}^{k} \mu_j(x(p))}, \tag{14}$$

where $k=1,...,l$; $j=1,...,h$.

The inverse F-transform in this case is defined by expression

$$\hat{y}_F(k)=f_{F,h}(x(k))=\sum_{j=1}^{h} F_j(p)\mu_j(x(k))=(F^h(p))^{\mathrm{T}}\mu^h(x(k)), \tag{15}$$

where $p=1,...,k$.

If the point $x(2)$ is situated in the basis function $\mu_1(x)$ influence area, then the adjustment of this function center disposition is made according to the expression

$$c_j(k)=(1-\alpha)c_j(k-1)+\alpha x(k), \tag{16}$$

(here $0\le\alpha\le 1$ – smoothing parameter) and F-transform components are improved by formula (14).

This going on either up to learning sample exhaustion or continuously in real time. The F-transform structure and parameters are adjusted and specified as new observation income. At the same time when new basis function is added the F-transform components are recomputed for all observations.

5 Numerical experiment

For illustration of proposed methods functioning the function

$$f(x)=\sin((x-1)(x-2)(2x-3)+6) \tag{17}$$

was chosen on the interval [0; 2,5] and consider its approximation using standard [2], adaptive [4,6] and evolving F-transform with triangle and spline membership functions.

In the table 1 the results of functions (17) approximation using different F-transform types are represented. As approximation error the expression

$$\varepsilon=\frac{1}{p}\sum_{p=1}^{k}\left|f(x(k))-f_{F,h_\varepsilon}(x(k))\right|$$

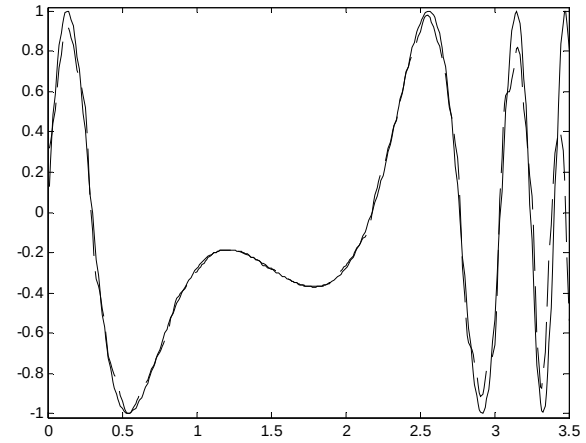
is used.

Tab.1. Error of function approximation using standard, adaptive and evolving F-transform

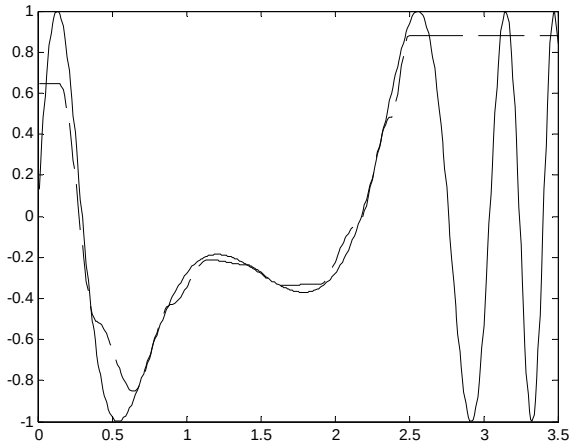
F-transform type		**Approximation error**
Standard F-transform		0,5768
Adaptive F-transform	Triangle membership functions	0,1016
	Cubic splines membership functions	0,0881
Evolving F-transform	Triangle membership functions	0,0904
	Cubic splines membership functions	0,0795

Consider also the problem of the function (17) approximation using adaptive and evolving F-transform (fig.5) to illustrate the properties of evolving F-transform. Function's values, that are fed to processing will go beyond the a prior given interval boundaries [0;2,5].

From the fig.4 it is obvious that the evolving F-transform can approximate the function, that go beyond the interval boundaries, since it has ability to place basis functions centers in the case when new data are situated outside the interval and to re-compute the F-transform components, that are associated with new center.



a) Evolving F-transform with cubic splines membership functions



b) Adaptive F-transform with cubic splines membership functions

Fig.4. Processing of function's (17) values, that go beyond the interval boundaries using evolving and adaptive F-transform (solid line – initial function, dashed line – its approximation).

6 Conclusion

Evolving F-transform, that allows to adapt not only parameters, but also structure of F-transform to variable plant characteristics is proposed in this paper. The possibility to rebuild the F-transform structure allows to increase the function approximation accuracy in the consecutive information processing mode. In addition it is shown the ability of evolving F-transform to process the data, that are situated outside the a prior given interval boundaries.

References

[1] Kasabov N.: Evolving Connectionist Systems: Methods and Applications in Bioinformatics. *Brain Study and Intelligent Machines.* – London: Springer – Verlag. - 2002. – 307 p.

[2] Perfilieva I.: Fuzzy transforms: Theory and Applications. *Fuzzy Sets and Systems*. – 157. – 2006. – P. 993-1023.

[3] Perfilieva I.: Fuzzy transforms. *Trans. on Rough Sets.* – Berlin: Springer-Verlag. – 2004. – P63-81

[4] Teslenko N., Deineko A.: Adaptive F-transform with cubic splines membership functions. *Modern computer systems and networks, development and application: 4-th Int. Scientific and Technical Conf. ACSN-2009*. – 2009. – P.186-188.

[5] Seng T.L., Khalid M., Yusof R., Omatu S.: Adaptive neuro-fuzzy control system by RBF and GRNN neural networks. *J. of Intelligent and Robotic Systems*. – 1998. – 23. – P.267-289.

[6] Bodyanskiy Ye., Teslenko N.: Adaptive F-transform on the basis of general regression neural network. *Automatic Control Adaptive Systems*. – vol. 10(30). – 2007. – P.25-31 (in Russian).

[7] Bodyanskiy Ye., Teslenko N., Teslenko S.: Adaptive real-time F-transform in approximation tasks. *Automation: Problems, Ideas, Decisions: Int. Scientific and Technical Conf.* – 2008. – P.86-89 (in Russian).