BUILDING A MIMETIC MODEL OF A NONLINEAR DYNAMIC SYSTEM

Mihai PASCADI[1], Bogdan TIGANOAIA[2]

The paper extends the results introduced in [1] in the attempt to provide a management instrument that would allow economic, technologic, and other types of systems' modeling based on trajectory / behavior monitoring. The conditions imposed in building the model in [1] are being relaxed in this paper, with the benefit of accelerating the model development. Steps ahead are also made in analyzing the convergence of the model to the "to be modeled" system. Numeric illustration confirms (in a 2-dimensional space) both that a good approximation of the model may be achieved based on the new results and that the real and the modeled systems have similar trajectories.

Keywords: nonlinear dynamic systems, mimetic model, economic and technical systems

1. Introduction

"The theory of dynamical systems is very broad and is extremely active in terms of research. It also depends substantially on most of the main areas of mathematics." [2]. The last 20 to 30 years have seen an intense development both in theory, in quantitative and qualitative analysis such as presented in [2] and [3] but also in applications like (technical, economic, social, bio, geo) systems engineering and control, as shown in [4], including modeling, simulation and systems optimization. "Today, we are facing major technological and social challenges, such as global warming, nuclear catastrophe, cyber security, worldwide shortage of potable water, and violent religious extremism. Addressing these problems will not only require knowledge in technologies and social sciences but also a deep understanding of the systems and their behaviors." [8]. Examples of dynamic modelling cover the entire spectrum of scientific and technological applications: biochemistry such as in [9], systems control [10] or quantum optics [11]. The range of modeling methods is also wide: Fuzzy Equations [12], Deep Boltzmann machines [13], convolutional neural networks [14], wavelet networks [15] etc. The conclusions chapter of ths paper also comments on two applications of the method in economy and technical systems management.

[1] Avantera s.r.l., Bucharest, Romania, E-mail: mihai.pascadi@avantera.ro

[2] Faculty of Entrepreneurship, Business Engineering and Management, University POLITEHNICA of Bucharest, Romania, E-mail: bogdantiganoaia@gmail.com

The purpose of this paper is to improve a previous method [1] and to analyze the ability of a specific mimetic model to fit a modeled system.

It would be ideal to understand the evolutions taking place around us by:

1. watching a system’s trajectory (memorizing its coordinates in time),
2. building a model based on the above observations,
3. making predictions based on the built model.

This kind of approach could be called a mimetic approach as the task is to try to build a model that would imitate the real system in its behavior. According to [8], such a model would be classified as a Black Box model (one does not need to understand what’s in the system). Of course, one may ask if the simple observation of a system would be sufficient in order to model it (are there any hidden parameters? – and if so, how do they influence the system’s evolution?). For the purpose of this paper, there will be used the supposition that the system is completely characterized by its visible parameters.

2. Building the model

[1] proposes approximating a nonlinear dynamical system by approximating its vector field system function F.

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}) \qquad (2.1)$$

where $\boldsymbol{x}$, $\boldsymbol{F(x)}$ ϵ D⊂ $\mathbf{R}^n$ (also called the phase space).

The approximating function proposed in [1] was supposed to be gradually built by observing the approximated system and collecting information regarding its behavior in time:

$x_1, \dots, x_T$, where, $x_t \in \mathbf{R}^n$ – called the approximation points (2.2)

and $x_{t+1} - x_t = \tilde{F}(x_t) * (t_{t+1} - t_t)$

with $F = (f_1, \dots, f_n)$ and $\tilde{F}(x_t)$ approximating F in x_t.

Regarding the approximation of the components of $F = (f_1, \dots, f_n)$, various methods used for approximation of a function can be used (two of which mentioned here):

1. Uniform approximation - for every continuous function on [a, b], the following norm is defined: ||f||= $\max_{x\in[a,b]}$ |f(x)|. The best polynomial approximation (also called minimax polynomial) of a continuous function f is p* n (x) with the following propriety: ||f- p* n (x)|| = $\min_{\text{pn}\,\in\pi\text{n}}$ ||f - p n (x)|| = $\min_{\text{pn}\,\in\pi\text{n}} \max_{x\in[a,b]} |\text{f(x)} - \text{pn (x)}|$ [7]. Characterization theorem: p* n (x) is an uniform approximation if e(x) = f(x) - p* n (x) has n+2 extreme values (+E and –E), with alternation of signs between 2 consecutive extreme values. Cebasev polynomials are used in the uniform approximation because these polynomials fulfil the characterization theorem. For a continuous function f, the minimax polynomial is: p* n (x) =$\sum_{p=0}^{n} Cp\; Tp(x)$, where

Cp are the coefficients of Taylor series with Cebasev polynomials of the function f(x) - p* n (x) and Tp(x) is Cebasev polynomial. Better approximations of minimax polynomial are obtained by using Rémes algorithms. More details about this type of approximation in [7].

2. Least Squares - discrete data sets are commonly involved in technical calculations. The source of the data may be experimental observations or numerical computations [5]. Least squares is a general class of methods for fitting observed data to a theoretical model function. In the general setting there are given a set of data and some class of functions, F. The goal then is to find the "best" f$\in F$ to fit the data to y = f(x). Usually the class of functions F will be determined by some small number of parameters; the number of parameters will be smaller (usually much smaller) than the number of data points. The theory here will be concerned with defining "best" and examining methods for finding the "best" function to fit the data [6]. (Ordinary) Least Squares Best Approximant - The least-squares best approximant to a set of data, x, y from a class of functions, F, is the function f *$\in F$ that minimizes the l2 norm of the error. That is, if f * is the least squares best approximant, then

$$\|y - f^*(x)\|^2 \quad = \min_{f \in \mathcal{F}} \|y - f(x)\|^2$$

The general assumption is that the minimum is unique. This method is sometimes called the ordinary least squares method. It assumes there is no error in the measurement of the data x, and usually admits a relatively straightforward solution. Least squares can be used in the following versions: continuous approximation (trigonometric least square; Cebasev least square) and discrete approximation (trigonometric least square; Cebasev least square).

As exemplification, the continuous least square approximation g*(x) of f(x) on C([a, b]) is defined by

$$\int_a^b w(x) * [f(x) - g^*(x)]^2 \, \mathrm{dx} = \min_{g \epsilon G} \int_a^b w(x) \, \cdot [f(x) - g(x)]^2 \, dx$$

More details about this type of approximation in [6] and [7].

In approximating the components of $F = (f_1, \dots, f_n)$, the following function was proposed as a sum of n-dimensional, bell-shaped functions:

$$\tilde{f}_i = \sum_{t=1}^{T} \varphi_t, \text{ with } \tilde{f}_i, \varphi_t \in \mathrm{R}. \tag{2.3}$$

where

$$\varphi_t \stackrel{\text{def}}{=} \frac{a_t}{(b_t * (x - x_t)^2 + 1)} \tag{2.4}$$

and

a_t, b_t are parameters while x_t is one of T approximating points in $\boldsymbol{R}^n$ and $(x - x_t)^2$ is the scalar product of the difference vector $x - x_t$ with itself, x_t is an approximation point and x is another point in $\boldsymbol{R}^n$ where the approximated value of F is to be calculated.

In order to determine the above parameters, [1] proposed the following method and conditions: (i) The approximating bell-shaped functions were grouped in pairs that had to approximate the “highest mountain” and the “deepest valley”, during each iteration; (ii) the “height” of the up-ward oriented bell should coincide with the maximum value of the f_i component of **F** while the depth of the down-ward oriented bell should coincide with the minimum value of the f_i component of **F**; (iii) The value of the sum function of the pair of bells should be the arithmetic average of the maximum value and minimum value of f_i in the middle of the segment that unites in $\boldsymbol{R}^n$ the position of the absolute minimum point and respectively absolute maximum point; (iv) the value of the b parameters is the same for the up-ward and down-ward oriented bell functions.

The above (i) condition was expressed as follows:

$$f_i^1(x) = \varphi_{mn} + \varphi_{mx} = \frac{a_{mn}}{(b_{mn} * (x - x_{mn})^2 + 1)} + \frac{a_{mx}}{(b_{mx} * (x - x_{mx})^2 + 1)} \tag{2.5}$$

while (ii) was:

$$f_i^1(x_{mn}) = y_{mn} \tag{2.6}$$

$$f_i^1(x_{mx}) = y_{mx} \tag{2.7}$$

3. Relaxing the imposed conditions to the approximating function

As opposed to the approach in [1], (iii) and (iv) are given up while keeping (i) and (ii) i.e. (2.5), (2.6) and (2.7).

By making the following notations: (3.1)

$$k_n = b_{mn} * (x_{mx} - x_{mn})^2 + 1 \triangleq b_{mn} * \delta^2 + 1$$

and

$$k_x = b_{mx} * (x_{mn} - x_{mx})^2 + 1 \triangleq b_{mx} * \delta^2 + 1$$

Determining a_{mn} and a_{mx} is reduced to resolving:

$$\begin{pmatrix} 1 & {k_x}^{-1} \\ {k_n}^{-1} & 1 \end{pmatrix} \begin{pmatrix} a_{mn} \\ a_{mx} \end{pmatrix} = \begin{pmatrix} y_{mn} \\ y_{mx} \end{pmatrix} \tag{3.2}$$

The results are:

$$a_{mn} = \frac{k_n}{k_n * k_x - 1} * (k_x * y_{mn} - y_{mx}) \,, \tag{3.3}$$

and

$$a_{mx} = \frac{k_x}{k_n * k_x - 1} * (k_n * y_{mx} - y_{mn}) \tag{3.4}$$

while keeping in mind that b_{mn}, b_{mx} can be established based on additional conditions as it may be convenient.

4. Studying the convergence of the approximating method

4.1 The algorithm

Changes to the algorithm described in [1] are related to the changes above in chapter 3 and as well to the distance used to measure the approximation error. The approximation is being developed after monitoring the system and collecting the values of the f_i components of the system's function $\boldsymbol{F}$ **(**which $= \dot{\boldsymbol{x}}$) in a number of discrete points the systems' trajectory passes through: $(\boldsymbol{x}, \dot{\boldsymbol{x}})_t, t = 1, T$. A Residual function (still to be approximated function) that initially equals f_i in the known points is being iteratively diminished in absolute value by subtracting the pairs of bell functions as determined and described above in chapter 2. For convergence evaluation purposes, known f_i components may be considered; in this case, a stop criterion may be defined: if d(f_i, $\widetilde{f_\imath}$) < ℮$_a$ (acceptable error), then stop; otherwise, continue to gather information (increase T) and iterate.

4.2 The approximation error and the associated distances

The approximation error may be defined component by component, based on a distance between functions defined on $D \subset \mathbb{R}^n$.

The absolute error is given by d(f_i, $\widetilde{f_t}$),

$$℮ = \mathrm{d}(f_i, \widetilde{f_t}) \tag{4.2.1}$$

where f_i is the i-th component of $\boldsymbol{F} = (f_1, \dots, f_n)$ and $\widetilde{f_t}$ is the t-th iteration of the approximation of f_i. A relative approximation error could be defined as well.

In order to prove convergence, any suitable distance may be used. For instance, if the intention is measure error all over D, the bellow proposed distance d_σ is the most relevant, especially as a stop criterion for the algorithm. For convergence demonstration purposes a different distance, d_x was suitable to be used (see please Lemma 3, in the next chapter).

The first proposed distance d_σ is:

$$d_\sigma(f, g) = \sum_{x \epsilon D} |f - g| \tag{4.2.2}$$

"d" is a distance if considering f,g,h: $D \subset \mathbb{R}^n \rightarrow \mathbb{R}$:

Indeed:

1. As $|f(x) - g(x)| \geq 0 \; \forall x \in D,\; \sum_{x \epsilon D} |f - g| \geq 0 \; so$ d(f, g) ≥ 0
2. $d(f, g) = 0 \;\Leftrightarrow\; \sum_{x \epsilon D} |f - g| = 0 \;\Leftrightarrow\; |f(x) - g(x)| = \;0 \quad \forall x \in D \; so \; f = g \;\forall\; x \in D$
3. $d(f, g) = \sum_{x \epsilon D} |f - g| = \sum_{x \epsilon D} |g - f| = d(g, f)$
4. For $x \in D$, the following inequalty is true:

$$|f - h| \leq |f - g| + |g - h|$$

By summing up to the left and to the right, after all $x \in D$, the result is:

$\sum_{x \in D} |f - h| \leq \sum_{x \in D} |f - g| + \sum_{x \in D} |g - h|$, and consequently:

$d_\sigma(f, g) = \sum_{x \epsilon D} |f - g|$ is a distance.

The second distance proposed is:

$$d_x(f,g) = \max_{x\epsilon D}|f-g| \tag{4.2.3}$$

For the second distance $d_x(f,g)$:

1. As $|f(x)-g(x)| \geq 0\ \forall x \in D,\ \max_{\boldsymbol{x}\in \mathrm{D}}|f(x)-g(x)| \geq 0\ so$ $\mathrm{d(f,g)} \geq 0$
2. $d(f,g)=0 \Leftrightarrow \max_{\boldsymbol{x}\in \mathrm{D}}|f(x)-g(x)| = 0\ considering\ that\ |f(x)-g(x)| \geq 0\ \forall x \in D\ so\ |f(x)-g(x)| = 0\ and\ after\ explicitating\ the\ module, f = g\ \forall\ x \in D$
3. $d(f,g) = \max_{\boldsymbol{x}\in \mathrm{D}}|f(x)-g(x)| = \max_{\boldsymbol{x}\in \mathrm{D}}|g(x)-f(x)| = d(g,f)$
4. Let's suppose that an arbitrary point $x \in D$ is chosen, where always:

$$|f-h| \leq |f-g| + |g-h|$$

because, depending on the order of f,g,h, , the following cases stand:

f,g,h $f-g \leq h-g+h-f \Leftrightarrow 2g \leq 2h$, true (considered case f ≤ g ≤h)

g,f,h $g-f \leq h-g+f-h \Leftrightarrow 2f \leq 2h$, true (considered case g ≤ f ≤h)

g,h,f $f-g \leq h-g+f-h \Leftrightarrow 0 \leq 0$, true

f,h,g $g-f \leq g-h+h-f \Leftrightarrow 0 \leq 0$, true

h,f,g $g-f \leq g-h+h-f \Leftrightarrow 0 \leq 0$, true

h,g,f $f-g \leq g-h+f-h \Leftrightarrow 2h \leq 2g$, true, (considered case h ≤ g ≤f)

Let $x_{f,h}$ so that $\left|f(x_{f,h})-h(x_{f,h})\right| = \max_{\boldsymbol{x}\in \mathrm{D}}|f(x)-g(x)|$; the above relationship remains true in this point as well:

$$|f-h| = \max_{\boldsymbol{x}\in \mathrm{D}}|f(x)-g(x)| \leq |f-g|+|g-h| \\ \leq \max_{\boldsymbol{x}\in \mathrm{D}}|f(x)-g(x)| + \max_{\boldsymbol{x}\in \mathrm{D}}|g(x)-h(x)|$$

The conclusion is:

$d(f,g) = \max_{\boldsymbol{x}\in \mathrm{D}}|f(x)-g(x)|$ is a distance.

4.3 The convergence of the approximating method

The convergence analysis is based on the relationship between two successive iterations of the Residual function (still to be approximated function):

$$\widetilde{R_t} = \widetilde{R_{t-1}} - \emptyset_{t-1} \tag{4.3.1}$$

Where $\emptyset_t(x)$ is the t-th iteration of the f_i of $\boldsymbol{F}$

The codomain of $\widetilde{R_{t-1}} \triangleq \mathrm{C}(\widetilde{R_{t-1}})$ is superior bounded by $R_{t-1,x}$ and inferior bounded by $R_{t-1,n}$. $\widetilde{R_{t-1}}$ reaches the values R_x and R_n at least in x_{mn}, respectively x_{mx}.

5. Three Lemmas on the convergence of the approximating method

Lemma 1

$$0 \in \mathrm{C}(\widetilde{R_t}), \forall t > 0, t \in \boldsymbol{N} \tag{5.1}$$

(The Codomain of the Residual function after the t-th iteration $\mathrm{C}(\widetilde{R_t})$ includes the value “0”.)

Indeed, as $\widetilde{R_t} = \widetilde{R_{t-1}} - \emptyset_{t-1}$

and through construction:

$\emptyset_{t-1}(x_n) = \min_{x \in D} \widetilde{R_{t-1}}$ with x_n so chosen that $\widetilde{R_{t-1}}(x_n) = \min_{x \in D} \widetilde{R_{t-1}}$

and

$\emptyset_{t-1}(x_x) = \max_{x \in D} \widetilde{R_{t-1}}$ $with\ x_x$ so chosen that $\widetilde{R_{t-1}}(x_x) = \max_{x \in D} \widetilde{R_{t-1}}$

It results that $\widetilde{R_t}(x_n)$ =0 and respectively $\widetilde{R_t}(x_x)$ =0 (the approximating function $\emptyset_{t-1}$ was designed to reduce the maximum and respectively minimum values of the Residual Function $\widetilde{R_{t-1}}$ to “0” in two points where it reaches its minimum and maximum value.

Lemma 2

$$\forall x \in D, x \neq x_{ea} \in EA = \{x \in D, x \neq x_n \wedge x \neq x_x \,|\, \widetilde{R_{t-1}}(x) = R_{t-1,x}\ or\ \widetilde{R_{t-1}}(x) = R_{t-1,n}\}, \tag{5.2}$$

$\forall\ \widetilde{R_{t-1}}(x), \exists b_{mn}, b_{mx} \in \boldsymbol{R}_+\ a.$î.:

$$R_{t-1,n} < \widetilde{R_t}(x) = \widetilde{R_{t-1}}(x) - \emptyset_{t-1}(x) < R_{t-1,x}$$

In other words:

The Residual function after the t-th iteration: $\widetilde{R_t}(x)$ with the exception of its t-1 iteration absolute maximum and minimum points remains strictly bounded between the minimum and maximum values of $\widetilde{R_{t-1}}$:

$$R_{t-1,n} < \widetilde{R_t}(x) < R_{t-1,x}$$

In fact, for (i) $x \in D \setminus EA$, and for (ii) b_{mn}, b_{mx} conveniently chosen the “amplitude” of the codomain of the Residual function decreases from one iteration to the next.

This means that the successive iterations of the Residual function are lower and upper bounded in a clipper that is closing and (based on Lemma 1) contains always the value “0”.

However, based on the above the only conclusion is that the Residual function has a limit (but not necessarily the constant function “0”).

In order to proof that:

$R_{t-1,n} < \widetilde{R_t}(x) = \widetilde{R_{t-1}}(x) - \emptyset_{t-1}(x), \forall x$ in the considered set,

it is sufficient:

(i) to remark that the most unfavorable situation is when $\emptyset_{t-1}(x)$ is positive and

(ii) to show that in this latter case the value of $\emptyset_{t-1}(x)$ can be diminished as much as one desires by modifying b_{mn} $și$ b_{mx} so that the above inequality becomes/is true (knowing that $\widetilde{R_{t-1}}(x) > R_{t-1,n}$) on the considered set D\EA:

$$\forall\varepsilon > 0, \exists b_{mn}(\varepsilon)\ or\ b_{mx}(\varepsilon)\ so\ that\ \emptyset_{t-1}(x) < \varepsilon \qquad (*)$$

Remark: (*) is implied by $\lim_{b_x\to\infty}\lim_{b_n\to\infty}\emptyset_t(x) = 0$

For simplicity, the following notations are made:
$b_n \triangleq b_{mn}, b_x \triangleq b_{mx}, a_n \triangleq a_{mn}, a_x \triangleq a_{mx}, x_x \triangleq x_{mx}, x_n \triangleq x_{mn}, y_n \triangleq y_{mn}, y_x \triangleq y_{mx}$
corresponding to the relationships (2.5),...,(2.4).

Knowing (2.5) and (2.1), (2.3) and (2.4) one may remark that:

$$\lim_{b_n\to\infty} a_n = y_n - y_x * {k_x}^{-1} = y_n - y_x * (b_x * \delta^2 + 1)^{-1}$$

$$\lim_{b_x\to\infty} a_n = y_n$$

$$\lim_{b_x\to\infty} a_x = y_x - y_n * {k_n}^{-1} = y_x - y_n * (b_n * \delta^2 + 1)^{-1}$$

$$\lim_{b_n\to\infty} a_x = y_x$$

and thus,

$$\lim_{b_x\to\infty}\lim_{b_n\to\infty} a_n = y_n$$

$$\lim_{b_x\to\infty}\lim_{b_n\to\infty} a_x = y_x$$

while:

$$\lim_{b_x\to\infty}\lim_{b_n\to\infty}\emptyset_t(x) =$$

$$\frac{\lim_{b_x\to\infty}\lim_{b_n\to\infty} a_n}{\lim_{b_x\to\infty}\lim_{b_n\to\infty}(b_n * (x - x_n)^2 + 1)} + \frac{\lim_{b_x\to\infty}\lim_{b_n\to\infty} a_x}{\lim_{b_x\to\infty}\lim_{b_n\to\infty}(b_x * (x - x_x)^2 + 1)} =$$

$$\frac{y_n}{\infty} + \frac{y_x}{\infty} = 0$$

Consequently, for b_n, b_x sufficiently large, $\emptyset_{t-1}(x)$ becomes sufficiently small so that the relationship (*) is fulfilled.

Similarly, for the second part of the relationship to be proven (6.1):
$\widetilde{R_t}(x) = \widetilde{R_{t-1}}(x) - \emptyset_{t-1}(x) < R_{t-1,x}$, $\forall x$ in the considered set,
it is sufficient:

(i) to remark that the most unfavorable situation is when $\emptyset_{t-1}(x)$ is negative and

(ii) to show that in this latter case the absolute value of $\emptyset_{t-1}(x)$ can be diminshed as much as needed by modifying b_{mn} $și$ b_{mx} so that the above inequality becomes/is true (knowing that $\widetilde{R_{t-1}}(x) < R_{t-1,x}$) on the considered set D\EA:

$$\forall\varepsilon > 0, \exists b_n(\varepsilon)\ or\ b_x(\varepsilon)\ so\ that\ \emptyset_{t-1}(x) < \varepsilon \qquad (**)$$

Similarly to the (*) case, the following calculation is made

$$\lim_{b_x\to\infty}\lim_{b_n\to\infty}|\emptyset_t(x)| \quad =$$

$$\left|\frac{\lim\limits_{b_x\to\infty}\lim\limits_{b_n\to\infty} a_n}{\lim\limits_{b_x\to\infty}\lim\limits_{b_n\to\infty}(b_n*(x-x_n)^2+1)}+\frac{\lim\limits_{b_x\to\infty}\lim\limits_{b_n\to\infty} a_x}{\lim\limits_{b_x\to\infty}\lim\limits_{b_n\to\infty}(b_x*(x-x_x)^2+1)}\right|=$$

$$\left|\frac{y_n}{\infty}+\frac{y_x}{\infty}\right|=0$$

Consequently, (**) is fulfilled.

The decrease of the amplitude (maximum value-minimum value) of the codomain of the Residual function on $x \in D\backslash EA$ from one iteration to the next is a direct consequence of the strict inequality:

$$R_{t-1,n} < \widetilde{R_t}(x) < \; R_{t-1,x}$$

which implies that on the considered domain:

$$R_{t-1,n} < R_{t,n} < R_{t,x} < R_{t-1,x}$$

Lemma 3

If for all “t” iterations of the approximation algorithm,

$$card(EA_n^t) = K_n < \aleph_0$$

respectively

$$card(EA_x^t) = K_x < \aleph_0,$$

the approximation process remains convergent.

Above:

$$EA_n^{t-1} = \{x_{en} \in EA, \left|\text{a. î.} \widetilde{R_{t-1}}(x_{en}) = R_{t-1,n}\right.\},$$

$$EA_x^{t-1} = \{x_{ex} \in EA, \left|\text{a. î.} \widetilde{R_{t-1}}(x_{ex}) = R_{t-1,x}\right.\}$$

Otherwise, the authors do not have at this moment a method to characterize the approximation process.

The subset of minimum extreme points is to be separated from

$$EA = \{x \in D, x \neq x_n \wedge x \neq x_x \left| \widetilde{R_{t-1}}(x) = R_{t-1,x} \; sau \; \widetilde{R_{t-1}}(x) = R_{t-1,n}\right.\}$$

corresponding to the t-1 iteration,

$$EA_n^{t-1} = \{x_{en} \in EA, \left|\text{a. î.} \widetilde{R_{t-1}}(x_{en}) = R_{t-1,n}\right.\} .$$

The Residual function (still to approximate) at the “t” iteration is determined with the relationship

$$\widetilde{R_t}(x) = \widetilde{R_{t-1}}(x) - \emptyset_{t-1}(x)$$

Let ”C” be the image of EA_n^{t-1} through $\widetilde{R_{t-1}}(x) - \emptyset_{t-1}(x)$. C shall include values given by:

$$\widetilde{R_t}(x) = R_{t-1,n} - \emptyset_{t-1}(x_{en})$$

In the unfavorable case where

$$\emptyset_{t-1}(x_1) > 0$$

all the values in C shall be smaller than $R_{t-1,n}$ and this iteration shall increase the approximation error associated to the d_x distance.

$$R_{t-1,n} > \min(\mathrm{C}) = R_{t,n}$$

In this case, the preimage of $R_{t,n}$ shall be the new EA_n^t.

The possible cases for EA_n^t are:

1. a finite number of elements $card(EA_n^t) = K < \aleph_0$
2. an infinite set of elements $(card(EA_n^t) \geq \aleph_0)$

In case 1., during the next iteration:

(i) the approximation error will increase (d_x)

(ii) $card(EA_n^t)$ shall decrease by at least 1 (the element for which the Residual function becomes 0) and new points in EA_n^t cannot appear (by applying Lemma 2 for the points in D\EA), and

(iii) in the most unfavorable case, all the $K - 1$ elements will become elements of EA_n^{t+1}.

Supposing that the following iterations will occur as well in the most unfavorable scenario, (which is improbable but not impossible), during the next K-1 iterations, one by one, the extreme minimum values smaller than $R_{t-1,n}$ and equal among them shall be eliminated. The next extreme minimum point(s) shall be all greater than $R_{t-1,n}$ and the approximation error at the step “t+K” shall become smaller than the one at step t-1.

In the second case, the number of extreme values is infinite, and one may not exclude particular situations where after each supplementary iteration the error defined based d_x would increase, the approximation process being divergent.

The above reasoning was applied for the set of extreme minimum points, but the reasoning is similar in the case of the set of extreme maximum points.

In conclusion, with the exception of some particular situations characterized as above, the approximation process is convergent over the entire set EA.

6. The relationship between the approximation error of F and the modeled system’s trajectory

Trajectories’ separation speed

As the basic idea of the mimetic modeling is to approximate the system’s evolution by approximating its system function $\boldsymbol{F}$, a legitimate question is: “given an approximation error of F, what would be the approximation error of the system’s trajectory?”

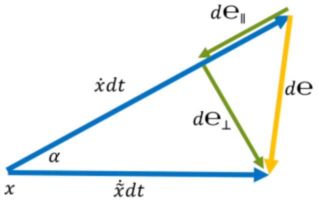

Fig. 6.1: $\dot{x}dt$ – elementary displacement of the approximated system; $\dot{\tilde{x}}dt$ – elementary displacement of the approximating system; $d\mathrm{e}$ - trajectory elementary error; $d\mathrm{e}_{\perp}$ - trajectory radial error; $d\mathrm{e}_{\parallel}$ - trajectory tangential error

The trajectory elementary error in determining the trajectory in a given point x is: (6.1)

$$d\mathrm{e} = \dot{\tilde{x}}dt - \dot{x}dt$$

where $\dot{\tilde{x}} = \tilde{F}$ and $\dot{x} = F$, which implies:

$\frac{d\mathrm{e}}{dt} = \dot{\tilde{x}} - \dot{x} = \tilde{F} - F$, thus:

$$\dot{\mathrm{e}} = \tilde{F} - F \tag{6.2}$$

Fig. 6.1 represents the separation rate (speed) between the two trajectories (the approximated and approximating systems' trajectories) in a given point "x". The above put in words would be: ***the trajectories' separation speed in each point $x \in D$ is precisely the difference between the approximated and approximating system functions which coincides with the approximation error of F;*** ($\dot{\mathrm{e}}, \tilde{F}, F, \in \mathbb{R}^n$). Separation speed is different from point to point and different initial conditions generate different real and modeled trajectories.

There are two kinds of trajectory error characterizations of interest: (i) for a particular trajectory – what is the resulting trajectory error? (ii) a global characterization of D regarding the trajectory errors. The simplest way to measure a particular trajectory absolute error is to measure the distance between the final positions indicated by the two trajectories (real and approximated). The relative error would be given by the ratio between the absolute error and the distance between the final and initial positions on the real trajectory. The global characterization of the trajectory error could be based on considering an average error within D (the average "ave" value of the trajectories' separation speed). The interpretation in this case is based on the most unfavorable case (as if the separation of the trajectories is cumulated all over D and there is no compensation between errors).

$$\underset{x\in D}{\mathrm{ave}}\left\|\dot{\mathrm{e}}\right\| = \underset{x\in D}{\mathrm{ave}}\left(\left\|\tilde{F} - F\right\|\right) \tag{6.3}$$

In order to be able to specify what means low separation vs high separation speeds a reference is needed. A convenient reference is the average (ave()) speed of the real system which is given by F.

$$\dot{\mathrm{e}}_{ave,rel} = \frac{\underset{x\in D}{\mathrm{ave}}\left(\left\|\tilde{F} - F\right\|\right)}{\underset{x\in D}{\mathrm{ave}}\left(\left\|F\right\|\right)} \tag{6.4}$$

Fig. 6.1 illustrates the two errors – the tangential error:

$$\dot{\mathrm{e}}_{\parallel} = \|\dot{x}\| - \left\|\dot{\tilde{x}}\right\| \cos\alpha \tag{6.5}$$

which represents the separation speed along the trajectory (the model moves slower or faster than the real one) while

$$\dot{\mathrm{e}}_{\perp} = \left\|\dot{\tilde{x}}\right\| \sin\alpha \tag{6.6}$$

represents the radial separation speed from the trajectory (the model moves on another path than the real one).

Based on 4.5 and 4.6 there may be defined the average relative values:

$$\dot{\mathrm{e}}_{\|,med,rel} = \frac{\underset{x\in D}{\operatorname{ave}}(\|\dot{x}\| - \|\dot{\tilde{x}}\| \cos\alpha)}{\underset{x\in D}{\operatorname{ave}}(\|F\|)} \tag{6.7}$$

$$\dot{\mathrm{e}}_{\perp,med,rel} = \frac{\underset{x\in D}{\operatorname{ave}}(\|\dot{\tilde{x}}\| \sin\alpha)}{\underset{x\in D}{\operatorname{ave}}(\|F\|)} \tag{6.8}$$

7. Numeric illustration

7.1 The convergence of the approximating method and mimetic model to the approximated model (2-dimensions).

Fig. 7.1.1 (next page) illustrates the comparison between the approximated function (f1,f2) in the upper left and right areas with the approximating function $(\tilde{f}_1, \tilde{f}_2)$ in the lower left and right areas but as well the two systems' trajectories. The figure also displays the approximation errors for $f_1{}^{\sim}$, $f_2{}^{\sim}$ as well as the trajectory error and the radial and tangential trajectory errors (separations). Fig 7.1.2 presents an enlarged view of the trajectories.

F(f_1, f_2) and its approximation$\tilde{F}(\tilde{f}_1, \tilde{f}_2)$ look alike (red = high values, green = low values), that the approximation error is reasonably low after only 20 approximation iterations for $F^{\sim}$ and that a particular trajectory of the mimetic system, based on randomly generated initial conditions, is as well a reasonably good approximation of the real trajectory after 100 integration iterations.

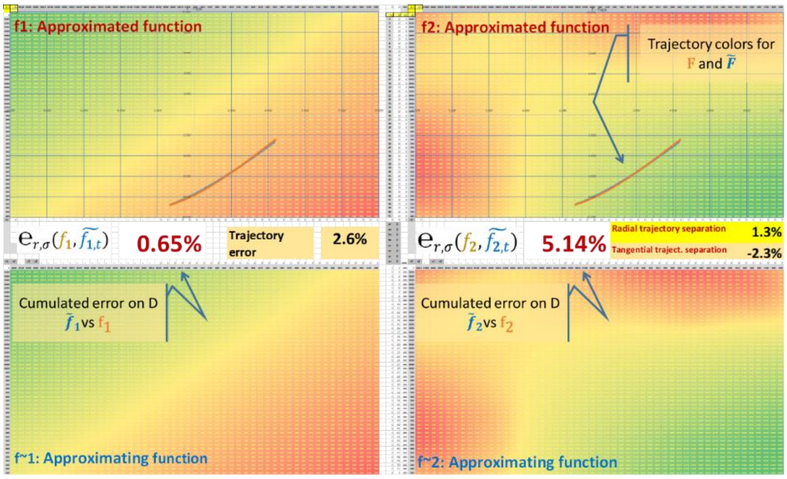

Fig 7.1.1 Convergence of the approximating method to the approximated model, including trajectories (F in brown, $\tilde{F}$ in light blue, 20 approximation iterations).

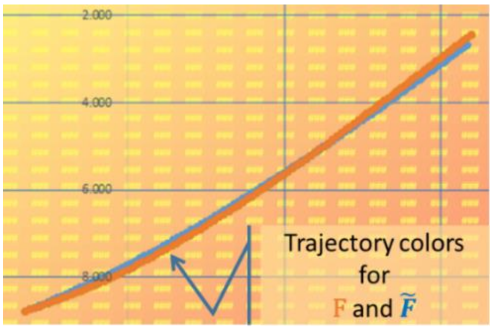

Fig 7.1.2 The enlarged trajectories presented in Fig 7.1.1 (brown – real system, blue – approximating system).

Fig 7.1.1 also shows F(f_1, f_2) and its approximation $\tilde{F}(\tilde{f}_1, \tilde{f}_2)$ look alike (red = high values, green = low values), and therefore the approximation error is reasonably low after only 20 approximation iterations for $\tilde{F}$ and that a particular trajectory of the mimetic system, based on randomly generated initial conditions, is as well a reasonably good approximation of the real trajectory after 100 integration steps.

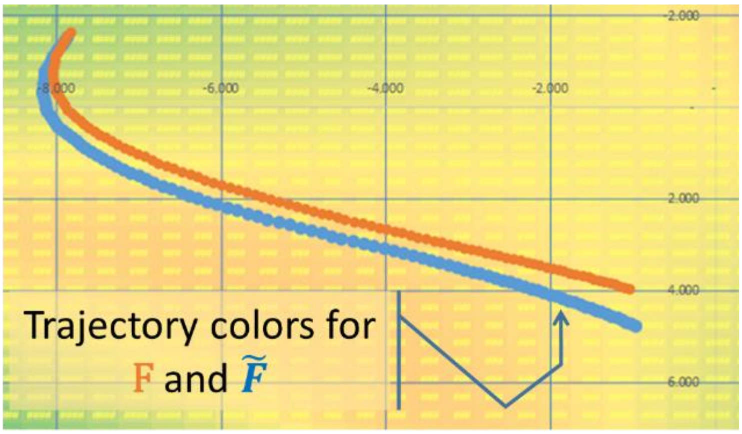

Fig. 7.1.3 A second illustration of trajectory approximation for randomly generated F and initial conditions. For this second example, the trajectory error is 9% after 100 integration steps, (6.4% radial, 6.3% tangential separation errors). Overall cumulated (on the definition domain D) errors for $\tilde{\boldsymbol{f}}_1$ and $\tilde{\boldsymbol{f}}_2$ in aproximating F(f_1, f_2) are 22.13%, respectively 8.33%.

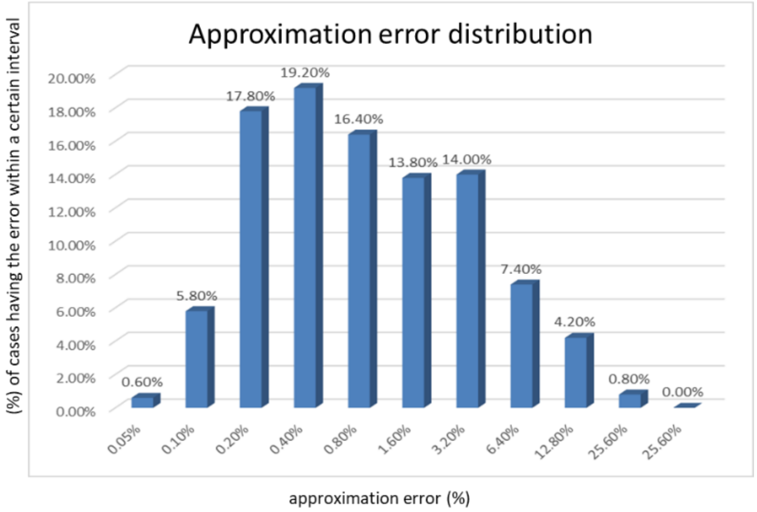

Fig. 7.1.4 Approximating F by $\tilde{F}$ distribution error for 500 randomly generated cases (horizontal axis is logarithmic) shows a reasonably low error after only 20 approximation iterations. Maximum error on this lot was 16.66%. The average error was 1.45%.

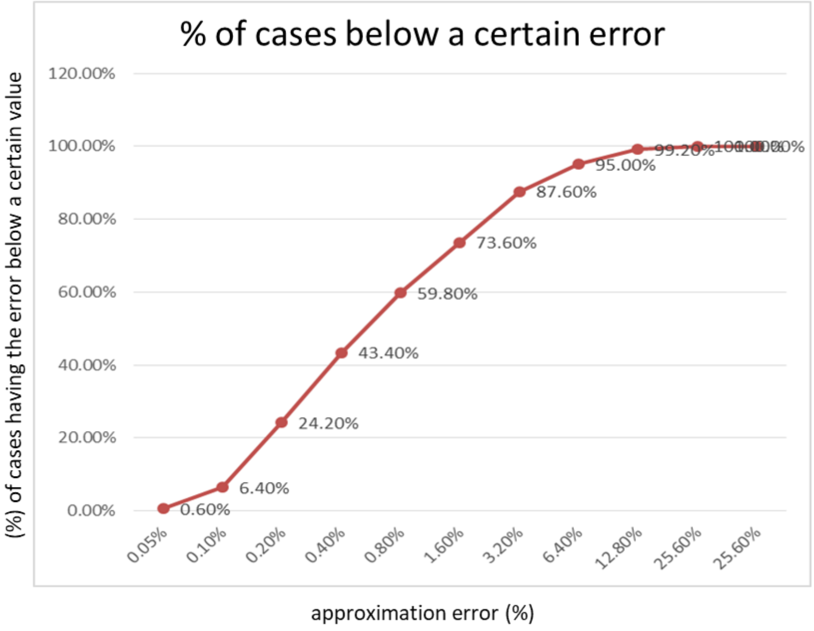

Fig. 7.1.5 Approximating F by $\tilde{F}$ percentage of cases below a certain error for the above distribution of 500 cases. 87.6% of the cases have a relative error below 3.2% (after only 20 approximation iterations). 95% of the cases had an error below 6.4%.

In evaluating the trajectory error (named ERRF/LTr), the considered the error is given by the ratio between

- the distance between the real and approximated trajectory ofter 100 integration steps and
- the trajectory length.

As the next page picture shows, based on the analysis of 300 cases, for a small number of cases the error was higher than 96% which corresponds to almost total unpredictability.

However, a large percentage of the cases (64%) had an error less than 12% after 100 integration steps of the trajectory, which (in our opinion) is remarcable.

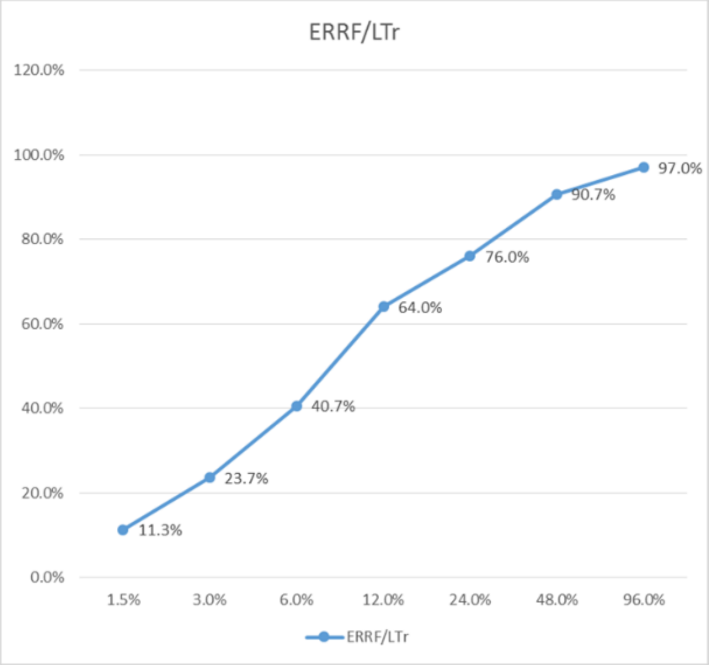

Fig. 7.1.6 Trajectory separation errors for 300 cases after 100 integration steps, based on a 20 iterations approximation of the vector field of the dynamic system.
64% of all the trajectories had an relative error (compared to the real trajectory) that was less than 12% after 100 integration steps.

7.2 Illustrating Lemma 2: monotonously decreasing the amplitude of the codomain of the Residual function by making a convenient choice of $\boldsymbol{b_n}, \boldsymbol{b_x}$[3]

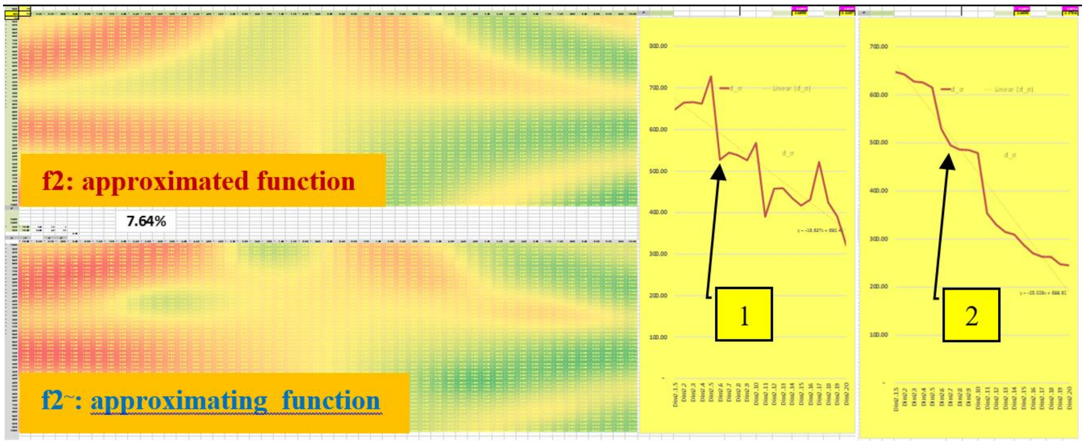

Fig. 7.2.1 shows the effect of conveniently choosing b_n,b_x on the monotony of the amplitude of the codomain of the Residual function x∈D\EA

Fig. 7.2.1 above (right side "1") shows the approximation error of one of F's components after n iterations when using relationships established in [1]. Based on Lemma 2, by using 3.3 and 3.4 (with modified values of b_n și b_x) the result is a monotonously decreasing error corresponding to a monotonously decreasing amplitude of the codomain of the residual function (right side "2").

8. Conclusions

The first conclusion is that numerical simulations (500 and 300 randomly generated cases) support both

- the convergence of the approximating method and mimetic model to the approximated model and
- the monotonous decrease of the amplitude of the codomain of the Residual function by making a convenient choice of $\boldsymbol{b_n}, \boldsymbol{b_x}$.

As part of this paper, a theoretical proof has been provided (numerically confirmed) that:

- the codomain amplitude of the residual (still to be approximated) function monotonously decreases
- the codomain allways includes the value ”0”.

This means that the approximation error decreases to a limit value close to ”0”, which may or may not be ”0”.

There has been also shown the relationship between

- the approximation error of the model function F and

[3] sufficiently large values, according to Lemma 2

- the error of the approximating trajectory of the system.

Numerical simulations support as well as the idea that once the model function F is well approximated, the trajectories of the system and its approximating model are similar and show low errors even after 100 steps of integration of the differential equations that describe them.

Applications studied by the authors (to be published) show that at least for lower frequency economic multidimensional economic signals (e.g. the moving average of the 30 exchange rates published by the National Bank of Romania) the increase/decrease predictions are surprisingly accurate by using a similar mimetic modeling approach. This could lead to the hypothesis that applications may include predictions regarding the macro-economic indicators (such as cash flows between the main economic nodes of an economic system – which reflect the average evolution of the companies in a node), even though external parameters such as political decisions, general trust in the economy, catastrophic events, etc could change in an unpredictible way the evolution (unless a model of these external factors influence is also developed). Technical applications studied by the authors included as well industrial climate evolution predictions (Pressure, Temperature and Humidity for sensitive areas of a pharmaceutical plant). The almost 43200 trajectory steps analyzed could be ***fully*** mimetically modeled within a 51-dimensional space (16 history points + the present point)*3. The particularity of this system was that it behaved as if there were a number of hidden parameters that changed its behavior in different passages through a given point (PTH had a two decimal digits precision).

The above two categories of applications (predictions for lower frequency economic signals and predictions for systems that seem to have hidden parameters) are open ways to further research.

REFERENCES

[1] Pascadi Mihai (2015), “A Model for Nonlinear Dynamic Systems with Application Potential in Management”, Proceedings of the IETEC-BRCEBE Conference, Sibiu, Romania. Copyright © M. Pascadi, 2015

[2] Barreira, Luis, Valls Claudia, “Dynamical Systems – An Introduction”, Universitext, DOI 10.1007/978-1-4471-4835-7_1, Springer-Verlag London 2013, p. v;

[3] Jost, Jurgn, “Dynamical Systems – Examples of Complex Behaviour”, Springer-Verlag Berlin Heidelberg 2005, p. 1-6 ;

[4] Ghosh, Asish, “Dynamic Systems for Everyone”, DOI 10.1007/978-3-319-10735-6_6, Springer International Publishing Switzerland 2015, p. 40-61, 83-126

[5] Kiusalaas, Jaan, “Numerical Methods in Engineering with MATLAB”, second edition, Cambridge University Press, ISBN: 978-0-521-19133-3, 2010, p. 125-136.

[6] Pav, Steven Elliot, “Numerical Methods Course Notes”, Version 0.11, (UCSD Math 174), Department of Mathematics, MC0112, University of California, 2004-2006, p 121-135.

[7] Iorga, Valeriu, “Numerical Methods Course Notes, Department of Computer Science”, University Politehnica of Bucharest, accessed in 2016.

[8] Ghosh, Asish, “Dynamic Systems for Everyone – Understanding How Our World Works”, second edition, DOI 10.1007/978-3-319-43943-3, Springer International Publishing Switzerland 2017, p. ix, p. 103-108.

[9] Gordon, Shane E., Weber, Daniel K., Downton, Matthew T, Wagner, John, Perugini, Matthew A. (2016) “Dynamic Modelling Reveals ‘Hotspots’ on the Pathway to Enzyme Substrate Complex Formation”, PLoS Comput Biol 12(3): e1004811. doi:10.1371/journal.pcbi.1004811.

[10] Wei, Wei, “On Disturbance Rejection for a Class of Nonlinear Systems”, Hindawi, Complexity, Volume 2018, Article ID 1212534, https://doi.org/10.1155/2018/1212534, 2018

[11] Buonaiuto, Giuseppe, Cancellieri, Emiliano, and Whittaker David M., “Quantum optics of nonlinear systems in cascade”, arXiv:1805.09272v1 [quant-ph] 23 May 2018

[12] Raheleh Jafari, Wen Yu, “Uncertainty Nonlinear Systems Modeling with Fuzzy Equations”, Conference: 2015 IEEE 16th International Conference on Information Reuse and Integration, DOI: 10.1109/IRI.2015.36, 2015

[13] Wen Yu, de la Rosa, Erick “Deep Boltzmann machine for nonlinear system modelling”, International Journal of Machine Learning and Cybernetics, DOI: 10.1007/s13042-018-0847-0, 2018

[14] Lopez, Mario, Wen Yu, “Nonlinear system modeling using convolutional neural networks”, 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), DOI: 10.1109/ICEEE.2017.8108894, 2017

[15] Postalcioglu, Seda, Becerkli, Yaşar, “Wavelet networks for nonlinear system modeling”, Neural Computing and Applications 16(4):433-441, DOI: 10.1007/s00521-006-0069-3, January 2017