COMPARATIVE STUDY OF CONVENTIONAL AND GENETIC ALGORITHMS IN ADAPTIVE SIGNAL PROCESSING AND CONTROL [a]

Innokenti Semoushin* **and Maria A. Fedorova***

**School of Mathematics and Mechanics, Ulyanovsk State University*
42 Leo Tolstoy Street, Ulyanovsk 432970, Russia

fax: +7 8422 41 2340 e-mail: `SemushinIV@ulsu.ru`

Abstract

A comparative study is made of genetic algorithms (GA) and conventional numeric algorithms (NA) for the purpose of filter optimization in a class of adaptive stochastic systems known as trackers. The need for using NA or GA is associated with uncertainty in covariances of the noises driving a reference signal model (RSM) (a shaping filter) and a controlled plant (CP).

[a] This work was supported by research grant No. T02-03.2-3427 from the Russian Ministry of Education and Science.

1 INTRODUCTION

The increasing interest in synthetic systems capable to maintain high performance under uncertainty and unpredictability has heightened the need for borrowing the mechanisms of adaptation from Biology, Evolution and Genetics suitable to improve the behavior of the system under synthesis. That such is the case is suggested by the fact that the standard mathematical programming models, that are synthetic as well, are not able sometimes to guarantee good results: the swiftness of response, short settling time and high steady-state accuracy for low computational cost.

To an increasing degree many non-standard approaches such as Fuzzy Technology, Neural Networks and Evolutionary Computing are merged in the area of “Intelligent Techniques” or “Computational Intelligence” (CI). The last three decades have witnessed a very strong growth of CI. Since the early seventies of the last century these techniques have also been applied to a large variety of problems to deliver efficient solutions and benefits to the users. Thus, the joint usage of expert system technology and fuzzy logic enables data base size to be cut in tens [?]. Another approach to intelligent system optimization and learning is closely associated with developing the combined technologies of fuzzy neural structures [?, ?]. Modern special purpose program-instrumental tools make possible not only simulate, in all details, the system under construction, but evaluate efficiency of the adopted design solutions based on one or other intelligent technology, as well. For example, the application soft package *WinFACT* (Windows Fuzzy and Control Tools) provides a means for going from a classic design to a neural network version of the fuzzy control being constructed through it [?].

At the same time, some distrust as to the novel, non-standard approaches to system optimization remains a fact within applied and computational mathematics community. This fact stimulates a comparative study of conventional and genetic approaches to the problem of eliminating uncertainty during the system life cycle. The genetic methods are of special interest because of their ability to treat the complex optimization problems using a simple and relatively inexpensive model.

In this paper, a problem of discrete time filter optimization both in open and close loop is considered as applied to trackers [**?**]. Two approaches to filter optimization within a stochastic control system operating under uncertainty are experimentally compared:

1) The auxiliary performance index approach based on classic numerical optimization methods [**?**] and
2) Non-numerical filter optimization using genetic algorithms [**?**, **?**].

The outline of the paper is as follows. Section **??** describes the problem including the monitored system, Kalman filter and adaptive filter. A general framework for Kalman filter identification is given in Section **??**. Section **??**

contains a short description of genetic algorithms. Some computational experiment results are shown in Section **??**. Finally, Section **??** concludes the paper.

2 PROBLEM DESCRIPTION

To construct and study the adaptive tracker, first express the problem in terms of the augmented state process [a)]

$$x_{\mathrm{a}}(\cdot) \doteq \begin{bmatrix} x(\cdot) \\ x_{\mathrm{r}}(\cdot) \end{bmatrix} \in \mathbb{R}^n$$

as generated in discrete time by

$$\begin{bmatrix} x(t_{i+1}) \\ x_{\mathrm{r}}(t_{i+1}) \end{bmatrix} = \begin{bmatrix} \Phi & 0 \\ 0 & \Phi_{\mathrm{r}} \end{bmatrix} \begin{bmatrix} x(t_i) \\ x_{\mathrm{r}}(t_i) \end{bmatrix} + \begin{bmatrix} \Psi \\ 0 \end{bmatrix} u(t_i) + \begin{bmatrix} w(t_i) \\ w_{\mathrm{r}}(t_i) \end{bmatrix} \qquad (1)$$

$$\mathrm{E}\left\{ \begin{bmatrix} w(t_i) \\ w_{\mathrm{r}}(t_j) \end{bmatrix} \begin{bmatrix} w^T(t_i) & {w_{\mathrm{r}}}^T(t_j) \end{bmatrix} \right\} = \begin{bmatrix} Q & 0 \\ 0 & Q_{\mathrm{r}} \end{bmatrix} \delta_{i,j} \qquad (2)$$

[a)] The notation "$\doteq$" instead of "$=$" is meant in this paper to indicate "equality by definition".

with a control input $u(\cdot)$ and zero mean white noises $w(\cdot)$, $w_{\mathrm{r}}(\cdot)$, and observed by

$$z_{\mathrm{a}}(\cdot) \doteq \begin{bmatrix} z(\cdot) \\ z_{\mathrm{r}}(\cdot) \end{bmatrix} \in \mathbb{R}^m, \quad z_{\mathrm{a}}(t_i) = \begin{bmatrix} z(t_i) \\ z_{\mathrm{r}}(t_i) \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & C_{\mathrm{r}} \end{bmatrix} x_{\mathrm{a}}(t_i) + \begin{bmatrix} v(t_i) \\ v_{\mathrm{r}}(t_i) \end{bmatrix} \tag{3}$$

$$\mathrm{E}\left\{ \begin{bmatrix} v(t_i) \\ v_{\mathrm{r}}(t_j) \end{bmatrix} \begin{bmatrix} v^T(t_i) & {v_{\mathrm{r}}}^T(t_j) \end{bmatrix} \right\} = \begin{bmatrix} R & 0 \\ 0 & R_{\mathrm{r}} \end{bmatrix} \delta_{i,j} \tag{4}$$

with zero mean white noises $v(\cdot)$, $v_{\mathrm{r}}(\cdot)$, where $\delta_{i,j}$ stands for Kroneker's delta function.

The tracker is designed to regulate the tracking error

$$e(t_i) \doteq C_{\mathrm{a}} \begin{bmatrix} x(t_i) \\ x_{\mathrm{r}}(t_i) \end{bmatrix} \quad \text{with} \quad C_{\mathrm{a}} \doteq \begin{bmatrix} C & \vdots - C_{\mathrm{r}} \end{bmatrix} \tag{5}$$

to zero which is to be accomplished by minimization of the mean quadratic cost

$$J_{\mathrm{control}} \doteq \lim_{t_0 \to -\infty} \; \mathrm{E} \left\{ \frac{1}{2} \sum_{j=0}^{i} \begin{bmatrix} x_{\mathrm{a}}(t_j) \\ u(t_j) \end{bmatrix}^T \begin{bmatrix} X_{\mathrm{a}} & S_{\mathrm{a}} \\ S_{\mathrm{a}}^T & U \end{bmatrix} \begin{bmatrix} x_{\mathrm{a}}(t_j) \\ u(t_j) \end{bmatrix} \right\} \tag{6}$$

where $X_a \doteq C_{\mathrm{a}}^T Y C_{\mathrm{a}}$ and $S_a \doteq C_{\mathrm{a}}^T S$ with S allowed to be nonzero, and some constant $Y > 0$ and $U > 0$, so that the composite matrix in (**??**) is positive semidefinite. Thus, the tracker must cause a controlled variable $y_{\mathrm{c}}(\cdot) = Cx(\cdot)$ to track some target (*reference*) variable $y_{\mathrm{r}}(\cdot) = C_{\mathrm{r}} x_{\mathrm{r}}(\cdot)$.

Restricting our attention to time-invariant system, we can seek the steady state (constant) gains G_{c1} and G_{c2} in the full-state feedback law of

$$u(t_i) = -\begin{bmatrix} G_{\mathrm{c1}} & \vdots & G_{\mathrm{c2}} \end{bmatrix} \begin{bmatrix} x(t_i) \\ x_{\mathrm{r}}(t_i) \end{bmatrix} = -G_{\mathrm{c1}} x(t_i) - G_{\mathrm{c2}} x_{\mathrm{r}}(t_i) \; . \tag{7}$$

This law [**?**] could be used if we had perfect knowledge of both $x(t_i)$ and $x_{\mathrm{r}}(t_i)$. As it is not the case, one employs the Kalman filter synthesis to generate estimates $\hat{x}(t_i^+)$ and $\hat{x}_{\mathrm{r}}(t_i^+)$ and substitute them in (**??**) for $x(t_i)$ and $x_{\mathrm{r}}(t_i)$, correspondingly. To obtain the filter decoupled into two totally independent filters, we require $x(t_0)$ and $x_{\mathrm{r}}(t_0)$ to be uncorrelated; the same requirement is adopted with respect to $w(\cdot)$ and $w_{\mathrm{r}}(\cdot)$ in (**??**) and $v(\cdot)$ and $v_{\mathrm{r}}(\cdot)$ in (**??**). Further, to guarantee existence of the steady state filters, we assume that Φ_{r} is *asymptotically stable* (has all of its eigenvalues strictly within the unit circle on the complex plane), the pair $(\Phi, Q^{1/2})$ is *stabilizable*, the pair (Φ, H) is *observable*, and the pair (Φ, Ψ) is *controllable*. Also, to exploit the Auxiliary Performance Index (API) approach for the optimal filters identification [**?**, **?**], we assume that both subsystems in (**??**), (**??**) are given in the *standard observable form*, SOM.

In these assumptions, the Steady-State Kalman Filter (SSKF) is presented by

$$\begin{aligned}\hat{x}_{\mathrm{a}}(t_{i+1}^{-}) &= \Phi_{\mathrm{a}}\hat{x}_{\mathrm{a}}(t_i^{+}) + \Psi_{\mathrm{a}}u(t_i), &(8)\\ \hat{x}_{\mathrm{a}}(t_i^{+}) &= \hat{x}_{\mathrm{a}}(t_i^{-}) + K_{\mathrm{a}}[z_{\mathrm{a}}(t_i) - H_{\mathrm{a}}\hat{x}_{\mathrm{a}}(t_i^{-})] &(9)\end{aligned}$$

with Φ_{a} and Ψ_{a} readily seen from (**??**), H_{a} from (**??**) and K_{a} computed through the forward Riccati recursion [**?**] for use $\hat{x}_{\mathrm{a}}(t_i^{+}) = [\hat{x}(t_i^{+})^T \vdots \hat{x}_{\mathrm{r}}(t_i^{+})^T]^T$ instead of $x_{\mathrm{a}}(t_i)$ in (**??**).

The problem we consider is caused by the fact of parameter uncertainty inherent to (**??**), (**??**), (**??**) and (**??**). In this work we concentrate on one level of uncertainty:

Case 1: Four matrices in the system description, namely Q, R, Q_{r} and R_{r} in (**??**) and (**??**), allow for the dependence on an uncertainty vector $\theta \in \Theta$ [b)]. Each particular value of θ specifies a *mode*.

[b)] Θ is a compact set where the SSKF (**??**), (**??**) exists.

Estimates of the Kalman gains K and $K_{\rm r}$ in $K_{\rm a} = {\rm diag}[K,\ K_{\rm r}]$ will be denoted correspondingly as $\bar{K}$ and $\bar{K}_{\rm r}$ in $\bar{K}_{\rm a} = {\rm diag}[\bar{K},\ \bar{K}_{\rm r}]$ within a suboptimal filter (SOF)

$$\begin{aligned} \bar{x}_{\rm a}(t_{i+1}^-) &= \Phi_{\rm a}\bar{x}_{\rm a}(t_i^+) + \Psi_{\rm a}u(t_i), && (10) \\ \bar{x}_{\rm a}(t_i^+) &= \bar{x}_{\rm a}(t_i^-) + \bar{K}_{\rm a}[z_{\rm a}(t_i) - H_{\rm a}\bar{x}_{\rm a}(t_i^-)] && (11) \end{aligned}$$

and as $\tilde{K}$ and $\tilde{K}_{\rm r}$ in $\tilde{K}_{\rm a} = {\rm diag}[\tilde{K},\ \tilde{K}_{\rm r}]$ within an adaptive filter (AF)

$$\begin{aligned} \tilde{x}_{\rm a}(t_{i+1}^-) &= \Phi_{\rm a}\tilde{x}_{\rm a}(t_i^+) + \Psi_{\rm a}u(t_i), && (12) \\ \tilde{x}_{\rm a}(t_i^+) &= \tilde{x}_{\rm a}(t_i^-) + \tilde{K}_{\rm a}[z_{\rm a}(t_i) - H_{\rm a}\tilde{x}_{\rm a}(t_i^-)] && (13) \end{aligned}$$

All the AF defined by (**??**), (**??**) vary in the algorithm of adaptation (selection) of $\tilde{K}_{\rm a}$. Two algorithms for this are compared: conventional numeric (NA) and genetic (GA).

3 GENERAL FRAMEWORK FOR IDENTIFICATION

Conventional identification methods such as stochastic approximation (or Robbins-Monroe procedure), least-squares method and many others are widely presented in the literature [?]. For comparison study, we select one of them known as Auxiliary Performance Index (API) method developed for adaptive filtering [?] and extended for control problems [?].

To tailor this paper with [?, ?], re-denote the variables of (??), (??) as follows

$$g(t_{i+1|i}) \doteq \tilde{x}_{\mathrm{a}}(t_{i+1}^{-}) \ , \quad g(t_{i|i}) \doteq x_{\mathrm{a}}(t_i^{+}) \ , \quad \eta(t_{i|i}) \doteq z_{\mathrm{a}}(t_i) - H_{\mathrm{a}}\tilde{x}_{\mathrm{a}}(t_i^{-}) \tag{14}$$

and denote the design (adjustable) AF parameter as the column vector

$$\hat{\theta} \doteq \mathrm{col}[\, \underbrace{\hat{\theta}_{11}, \ldots, \hat{\theta}_{1m}; \cdots ; \hat{\theta}_{n1}, \ldots, \hat{\theta}_{nm}}_{\text{all } p \text{ entries of } \tilde{K}_{\mathrm{a}} \text{ ordered by rows}} \,] \in \mathbb{R}^p \ , \quad \hat{\theta}_{ij} \doteq [\tilde{K}_{\mathrm{a}}]_{ij} \ , \quad p = mn \ . \tag{15}$$

As it is known, API method boils down to MPE[a)] method except one fea-

[a] Minimum Prediction Error

ture: it exploits the idea of equivalent replacing the *Original Performance Index* being Direct but Inaccessible (DbI) one, by an *Auxiliary Performance Index* being Accessible albeit Indirect (AaI). The main requirement for this replacement is that OPI and API be *equimodal*, that is, they have one and the same minimizing argument. Thus, based on API approach, we do the following:

1. Construct the Sensitivity Model (SM) by partial differentiating equations (**??**)–(**??**) to obtain the recursive equations for the sensitivity functions

$$\mu_{jl}(t_{i+1|i}) \doteq \frac{\partial}{\partial \hat{\theta}_{jl}} g(t_{i+1|i}) \ , \quad j = 1, \ldots, n \ ; \quad l = 1, \ldots, m$$

2. For $j = 1, \ldots, n$; $l = 1, \ldots, m$, compute

$$\begin{aligned} \eta^{i|i-1}_{i-s+1|i-s} &= [\eta^T(t_{i-s+1|i-s}) \ | \ \cdots \ | \ \eta^T(t_{i|i-1})]^T \\ \varepsilon(t_i, \hat{\theta}) &= \mathscr{P}(\tilde{K}_{\mathrm{a}}) \eta^{i|i-1}_{i-s+1|i-s} \end{aligned} \tag{16}$$

$$\begin{aligned}\frac{\partial \eta(t_{i|i-1})}{\partial \hat{\theta}_{jl}} &= -H_{\mathrm{a}}\mu_{jl}(t_{i|i-1}) \\ \frac{\partial \varepsilon(t_i,\hat{\theta})}{\partial \hat{\theta}_{jl}} &= \left[\frac{\partial}{\partial \hat{\theta}_{jl}}\mathscr{P}(\tilde{K}_{\mathrm{a}})\right]\eta^{i|i-1}_{i-s+1|i-s} + \mathscr{P}(\tilde{K}_{\mathrm{a}})\frac{\partial}{\partial \hat{\theta}_{jl}}\eta^{i|i-1}_{i-s+1|i-s} \qquad (17)\end{aligned}$$

where $\mathscr{P}(\cdot)$ is a procedure defined in [**?**] as follows.

Definition 1. *Let* $s, p_1, \ldots, p_m \in \mathbb{N}$ *satisfy*

$$s \doteq \max(p_1, p_2, \ldots, p_m) \leq p_1 + p_2 + \ldots + p_m = n$$

and A *be an* $ms \times k$ *matrix,* $k \in \mathbb{N}$*, composed of* s *submatrices with* m *rows and* k *columns each. Denote the* j*-th row of the* i*-th submatrix by* a_i^j

and rearrange all these ms rows in order to form a new $s \times km$ matrix

$$A_T \doteq \begin{bmatrix} a_1^1 & \dots & a_1^m \\ \vdots & \ddots & \vdots \\ a_s^1 & \dots & a_s^m \end{bmatrix} .$$

Then $\mathscr{S}(A)$, which is the $n \times k$ matrix, is called the $\mathscr{S}$-transform of the matrix A, provided its n rows are obtained by taking the elements a_i^j from A_T and placing them into $\mathscr{S}(A)$ as rows in the following order: $a_1^1, a_2^1, \dots, a_{p_1}^1,\ a_1^2, a_2^2, \dots, a_{p_2}^2,\ \dots,\ a_1^m, a_2^m, \dots, a_{p_m}^m$.

Definition 2. *Let $(\Phi_\star, H_\star)$ be matrices in the SOM such that $\Phi_\star$ is of a block companion form whose non-trivial (non-zero or non-unit) elements lie in the rows with numbers $p_1,\ p_1 + p_2,\ \dots,\ p_1 + p_2 + \dots + p_m = n$, where p_j are the given partial observability indices of $(\Phi_\star, H_\star)$, and $H_\star$ is the $m \times n$ matrix whose ij-th element is 1 if $j = p_1 + p_2 + \dots + p_{i-1} + 1$*

and 0 *otherwise.*

Then $\mathscr{P}(D)$, *which is the* $(n \times sm)$*-matrix, is called the* $\mathscr{P}$*-transform of an* $(n \times m)$*-matrix* D, *provided*

$$\mathscr{P}(D) \doteq \mathscr{S}(S_\theta(H_\star, \Phi_\star, \Phi_\star D))$$

with the $(sm \times sm)$*-matrix*

$$S_\theta(H_\star, \Phi_\star, G) \doteq \begin{bmatrix} I & 0 & \cdots & 0 \\ H_\star G & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_\star \Phi_\star^{s-2} G & H_\star \Phi_\star^{s-3} G & \cdots & I \end{bmatrix}, \quad G = \Phi_\star D \qquad (18)$$

3. Arrange the sensitivity functions defined by (**??**) into the $n \times p$ sensitivity

matrix

$$S(t_i) \doteq \left[\frac{\partial\varepsilon(t_i,\hat{\theta})}{\partial\hat{\theta}_{11}}, \dots, \frac{\partial\varepsilon(t_i,\hat{\theta})}{\partial\hat{\theta}_{1m}}, \cdots, \frac{\partial\varepsilon(t_i,\hat{\theta})}{\partial\hat{\theta}_{n1}}, \dots, \frac{\partial\varepsilon(t_i,\hat{\theta})}{\partial\hat{\theta}_{nm}} \right] \quad (19)$$

4. Construct the Gradient Model with an exponential smoothing factor β

$$\begin{aligned} \mathcal{G}(t_i) &= S^T(t_i)\varepsilon(t_i,\hat{\theta}) \quad (20) \\ \hat{\mathcal{G}}(t_i) &= \beta\hat{\mathcal{G}}(t_{i-1}) + (1-\beta)\mathcal{G}(t_i) \end{aligned}$$

5. Check the Stability Condition (SC): $\rho[(I - \tilde{K}_{\rm a}H_{\rm a})\Phi_{\rm a}] < 1$ for AF and SM where $\rho[\cdot]$ stands for the spectral radius of matrix $[\cdot]$. In doing so, the designer should have the characteristic polynomial for matrix $(I - \tilde{K}_{\rm a}H_{\rm a})\Phi_{\rm a}$, that is written as

$$q(\lambda) \doteq b_0\lambda^n + b_1\lambda^{n-1} + \cdots + b_{n-1}\lambda^1 + b_n, \quad b_0 > 0$$

with $b_0, \dots, b_n$ expressed as some functions $b_k = b_k(\hat{\theta})$ of the design parameter (**??**), and then use Jury's criterion [**?**, Sect. 3.3]. If SC is met, we write $SC(\hat{\theta}) = \text{true}$, and $SC(\hat{\theta}) = \text{false}$ otherwise, and we call this procedure by `checkSC`$(\hat{\theta})$.

6. Use an **Adaptation Procedure** (AP) to update the parameter (**??**) within an effective operating range of t_i from t_{start} to t_{stop}. For AP, one can use different optimum seeking methods able to minimize a performance index taken to characterize the quality of AF (**??**)–(**??**). For such index, we use in this paper the API

$$J(\tilde{\theta}) \doteq \frac{1}{2}\,\mathrm{E}\left\{\varepsilon(t_i,\hat{\theta})^T\varepsilon(t_i,\hat{\theta})\right\} \tag{21}$$

with the generalized residual (**??**). Using the Simplified Least Squares (SLS) to minimize (**??**), the AP looks as follows:

```
**if** $(t_{\text{start}} \leq t_i < t_{\text{stop}})$ **then**
    **begin if** $(t_{\text{start}} = t_i)$ **then**
        **begin** $k := 0$; $\Lambda = I$ **end else** $k := k + 1$;
        **if** $(k > 0 \ \& \ k = 0 \bmod s)$ **then**
        **begin** $\tau := [k - 1/s]$; **for** $j = 1$ **to** $p$ **do**
            $\lambda^{(j)}_{\tau+1} := \lambda^{(j)}_{\tau} + \left\| \frac{\partial r(t_i, \hat{\theta}(\tau))}{\partial \hat{\theta}_j} \right\|^2$; $\Lambda_{\tau+1} = \text{diag}\ \{\lambda^{(j)}_{\tau+1}\}$;        $\backslash\star$ SLS $\star\backslash$
            $\pi := \hat{\theta}(\tau) - \Lambda^{-1}_{\tau+1} \hat{\mathcal{G}}(t_i)$;        $\backslash\star$ Num$\star\backslash$
            checkSC($\pi$);
            **if** $(SC(\pi))$ **then** $\hat{\theta}(\tau + 1) := \pi$
        **end**
    **end**
```

Remark 1. *Changing the line marked by* $\backslash\star$ SLS$\star\backslash$, *one can go to another minimum seeking method. For instance, by using*

$$\lambda_{\tau+1}^{(j)} := \lambda_{\tau}^{(j)} + 1$$

one switches to the Simple Stochastic Approximation $\backslash\star$ SSA$\star\backslash$. *Such cases, together with the line marked by* $\backslash\star$ Num$\star\backslash$, *form a numeric method to optimize* $\hat{\theta}$.

Remark 2. *Changing only the two marked lines, one can go to a genetic method.*

Remark 3. *Due to* $s-1$ *one sample delays in the framework of Fig.* **??**, *the time pace for AP has to be set* s *times lower than for the system. To do this, the AP timer* τ *is introduced in the above AP through the truncation function* $[\cdot]$.

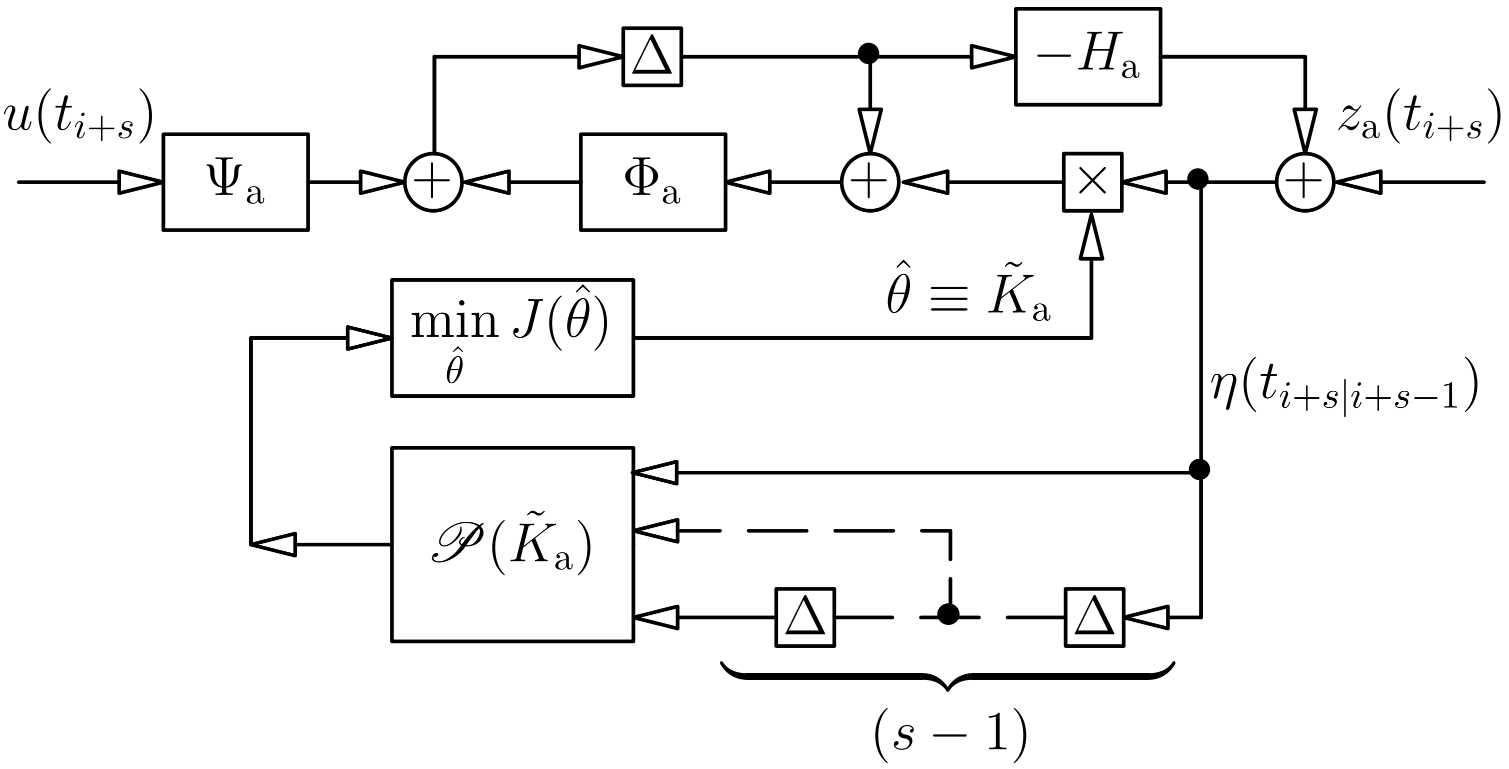

Figure 1: The general framework for using the API. Δ denotes a one sample memory (unit delay).

4 THE SIMULATED GENETIC ALGORITHMS [?, ?, ?, ?]

A genetic algorithm (GA) differs from other search techniques by the use of concepts taken from natural genetics and evolution theory.

The basic element processed by a GA is the string formed by concatenating a substring, each of which is a binary code of a parameter of search space [**?**]. Thus each string represents a possible solution to the problem. The GA works with a set of strings, called the population. This population then evolves from generation to generation through the application of genetic operators. A GA in its simplest form uses three operators: Replication, Crossover and Mutation.

Replication. Replication is based on the principle of survival of the fittest. A fitness, $F(i)$, is assigned to each individual in the population where high numbers mean good fit. The fitness function can be any nonlinear, non-differentiable, discontinuous, positive function, because the algorithm only needs a fitness assigned to each string. We will first consider the construction of the intermediate population from the current population. In the first generation the current population is also the initial population. After calculating $F(i)/\bar{F}$ (where $\bar{F}$ is the average value of the fitness) for all the strings in the current population, selection is carried out. In the canonical genetic algorithm, the probability that strings in the current population are copied (i. e., replicated) and placed in the intermediate generation is proportional to their fitness.

There are a number of ways to do selection. We might view the population as mapping onto a roulette wheel, where each individual is represented by a space that proportionally corresponds to its fitness. By repeatedly spinning the roulette wheel, individuals are chosen using "stochastic sampling with replacement” to fill the intermediate population.

A selection process that will more closely match the expected fitness values is a “remainder stochastic sampling”. For each string i where $F(i)/\bar{F}$ is greater than 1.0, the integer portion of this number indicates how many copies of that string are directly placed in the intermediate population. All strings (including those with $F(i)/\bar{F}$ less than 1.0) then place additional copies in the intermediate population with a probability corresponding to the fractional portion of $F(i)/\bar{F}$. For example, a string with $F(i)/\bar{F} = 1.36$ places 1 copy in the intermediate population, and then receives a 0.36 chance of placing a second copy. A string with a fitness of $F(i)/\bar{F} = 0.54$ has a 0.54 chance of placing one string in the intermediate population.

Crossover: Replication directs the search toward the best existing individuals but does not create any new individuals. In nature, an offspring is rarely an exact clone of a parent, it usually has two parents and inherits genes from both. The main operator to work on the parents is crossover, which is applied with a certain probability, called crossover rate (p_c). Crossover takes two individuals from intermediate population, and cuts their chromosome strings at some randomly chosen position, to produce two "head" segments, and two "tail" segments. The tail segments are then swapped over to produce two new full length chromosomes:

$$\left\{\begin{array}{cccccccc} 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & \vdots & 1 & 1 & 1 & 1 \end{array}\right. \times \Rightarrow \quad \begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}$$

This is known as single-point crossover.

A two-point crossover operator uses two randomly chosen crossover points.

Strings exchange the segment that falls between these two points:

$$\left\{\begin{array}{ccccccccc} 0 & 0 & \vdots & 0 & 0 & 0 & \vdots & 0 & 0 \\ 1 & 1 & \vdots & 1 & 1 & 1 & \vdots & 1 & 1 \end{array}\right. \times \Rightarrow \begin{array}{ccccccc} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{array}$$

Mutation: Mutation is applied to each child individually after crossover. It randomly alters each gene with a small probability (typically 0.001). Following example shows the third gene of the chromosome being mutated.

$$0 \;\; 0 \; \vdots \; 0 \; \vdots \; 0 \;\; 0 \Rightarrow 0 \;\; 0 \; \vdots \; 1 \; \vdots \; 0 \;\; 0$$

5 SOME COMPUTATIONAL EXPERIMENT RESULTS

A sample of GA behavior within the Adaptation Procedure of Section **??** is shown on Figure **??**. For simulation, the tracker described in [**?**] has been chosen. In this example, we have the following constant quantities:

$$\Phi = [0.82],\ \ \Phi_{\mathrm{r}} = [0.61],\ \ \Psi = [0.18],\ \ C = [1],\ \ C_{\mathrm{r}} = [1],\ \ Y = [1],\ \ U = [1]$$

and $S_{\mathrm{a}}^T = [0 \vdots 0]$ for the LQG controller (**??**) synthesis. This gives

$$G_{\mathrm{c1}} = [0.36]\ ,\quad G_{\mathrm{c2}} = [-0.19]\ .$$

For the Kalman filter synthesis we have the following data:

$$H = [1],\ \ Q = [0.084\sigma^2],\ \ Q_{\mathrm{r}} = [0.63\sigma_{\mathrm{r}}^2],\ \ \sigma^2 = \mathit{var},\ \ \sigma_{\mathrm{r}}^2 = \mathit{var},\ \ R = \mathit{var},\ \ R_{\mathrm{r}} = \mathit{var}$$

where "*var*" stands for the variable quantities. While simulating, we specify the two phases named "*before*" and "*after*" (before and after a parameter switch). The phase "*before*" takes 300 time instants and phase "*after*" takes

10000 time instants.

In these experiments, simulation parameters are as follows:

1. Number of Iterations, (NI) 10000
2. Number of Experimental Samples (NES) to average results, ... 100
3. Simulated numeric algorithms and their parameters:
   (a) Robbins-Monroe procedure, ... SSA
   (b) Simplified Least Squares, ... SLS
   (c) Exponential smoothing parameter β, ... 0.5
4. Simulated genetic algorithm:
   (a) Chromosome Length, (CL) 7
   (b) Power (size) of population, (PP) 50

(c) Eliticity Factor [a)], (EF) 2

(d) Mutation Probability Rate, (MPR) 0.100

(e) Selection Mode, (SM) .... remainder stochastic sampling

(f) Crossover Mode, (CM) ....two-point

In experiments, we consider the following two cases.

Case 1. See Figures **??** to **??**.

“*before*”						“*after*”					
σ^2	σ_{r}^2	R	R_{r}	K	K_{r}	σ^2	σ_{r}^2	R	R_{r}	K	K_{r}
10.0	4.0	4.0	1.0	0.287	0.736	10.0	4.0	0.1	10.0	0.900	0.258

[a)] Number of chromosomes passing from current population into future population without any operations (selection-crossover-mutation) as being the best ones.

Case 2. See Figures **??** to **??**.

"*before*"						"*after*"					
σ^2	σ_{r}^2	R	R_{r}	K	K_{r}	σ^2	σ_{r}^2	R	R_{r}	K	K_{r}
10.0	4.0	4.0	10.0	0.287	0.258	10.0	4.0	0.1	1.0	0.900	0.736

The Integral Percent Error has been determined (with averaging over the NESs) by

$$\text{(a)IPE}(t_i) \doteq \frac{\|\tilde{K}_{\mathrm{a}}(t_i) - K_{\mathrm{a}}\|}{\|K_{\mathrm{a}}\|} 100\% \text{ or } \quad \text{(b)IPE}(t_i) \doteq \frac{\|\tilde{K}(t_i) - K\|}{\|K\|} 100\% \tag{22}$$

In the other portion of experiments, we modify the parameters: NI to 5000, NES to 50, CL to 5, PP to 30, and MPR to 0.200. Corresponding results for Case 1 are shown in Figures **??** to **??** [b)]. The IPE is shown in the figures

[b)] Figures **??** to **??** have been shown in Appendix.

for all three methods: SSA by red lines, SLS by blue lines and GA by green lines.

As can be seen from these and many other experiments that were conducted, in the majority of cases, the averaged GA behavior provides the lower IPE levels. However it is important to note, that the individual GA behavior is much more fluctuating than the individual numeric methods behavior.

The consistent difference between these types of methods, that was to be expected, lies in the very mechanism of their behavior. Numeric algorithms are sequential in operation while genetic algorithms are parallel. The latter require a set of individuals forming the current population of adaptive filters, each individual filter working through the mechanism of Figure **??**. Large size of population causes the computation load to rise so that real time data processing may occur to be difficult to sustain.

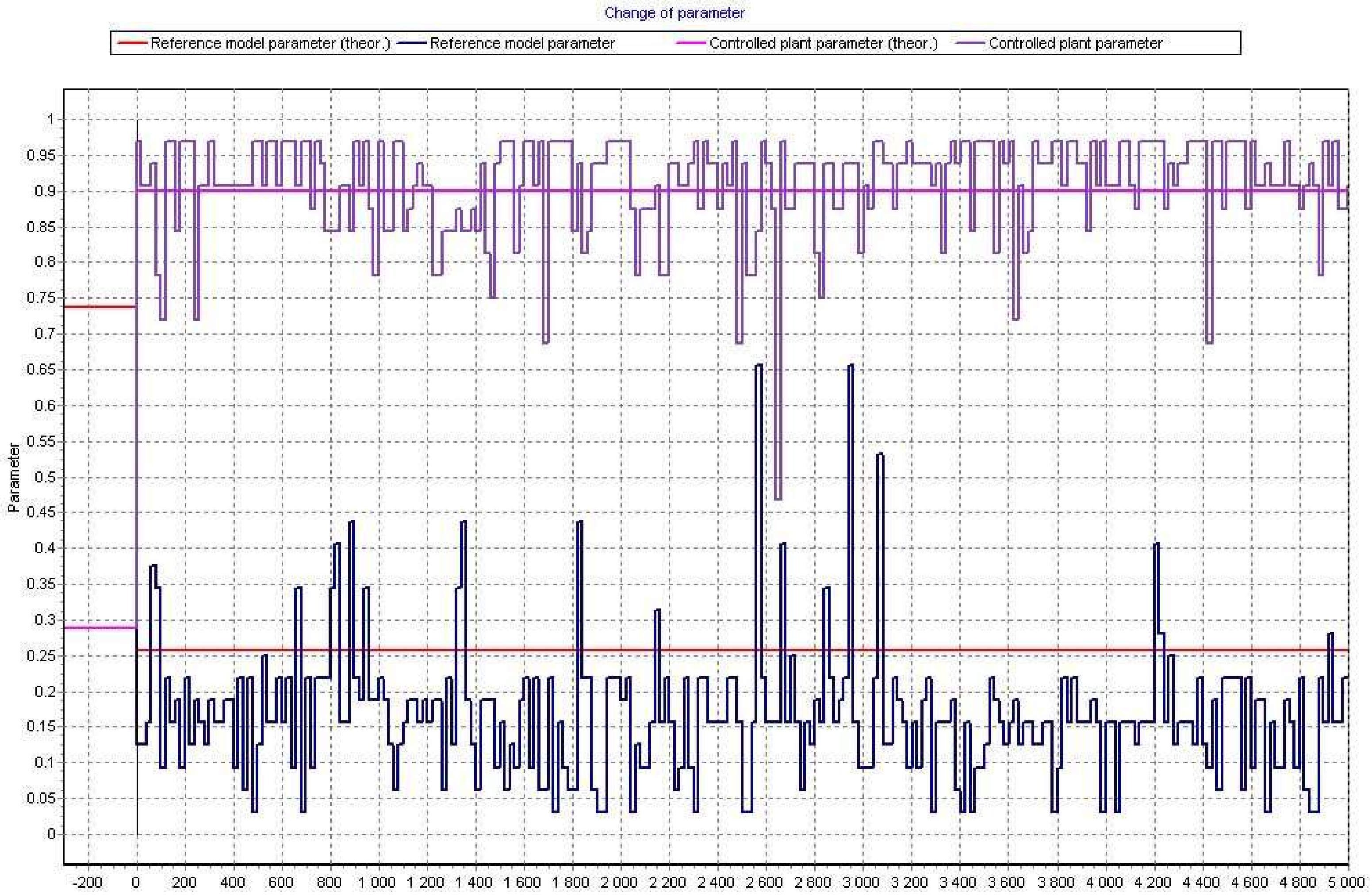

Figure 2: Behavior of the modified GA parameters for Case 1.

6 ...

6.1 The Sensor equation

References

[1] Hjalmarsson, H., Gevers, M., Gunnarsson, S. & Lequin O. 1998. Iterative feedback tuning: theory and applications. *IEEE Control Systems Magazine* **18**: 26–41.

[2] Basseville, M., Nikiforov, I.: Detection of Abrupt Changes: Theory and Applications. Prentice-Hall Inc., New York (1993)

[3] Landau, I.D. (ed.): Identification des Systemes. Les Bases de l'Identification des Systemes. Hermes, Paris (2001)

[4] Winner, R.I., Pennell, J.P., Bertrend, H.E. & Slusarczuk, M.M.G. 1988. *The Role of Concurrent Engineering in Weapons System Acquisition*, IDA Report R-338. Alexandria, VA: Institute for Defense Analyses.

[5] Maybeck, P.S. 1982. *Stochastic models, estimation, and control* **3**. Paris San Diego San Francisco Sao Paulo Sydney Tokyo Toronto: Academic

Press.

[6] Caines, P.: Linear Stochastic Systems. John Willey & Sons, New York Chichester Brisbane Toronto Singapore (1988)

[7] Semoushin, I.V. 1974. On One Approach to Fault Detection in Linear Dynamical Systems with Possible Disturbances. *Automatics and Computing Techniques, The Latvian SSR Academy of Sciences* **4**: 24–30.

[8] Ponyrko, S.A. & Semoushin, I.V. 1973. On the Choice of Start-Stop Algorithm while Minimizing the Square Mean Performance Index. *Autometria, The USSR Academy of Sciences* **2**: 68–74. [*in Russian*]

[9] Soderstrom, T.: Identification of stochastic linear systems in presence of input noise. Automatica **17** (1981) 713–725

[10] Ansay, P., Gevers, M., Wertz, V.: Closed-loop or open-loop models in identification for control? In: Proc. of the European Control Conference

(1999) CD-ROM F544

[11] Semoushin, I.V.: Adaptive Identification and Fault Detection Methods in Random Signal Processing. Saratov University Publishers, Saratov (1985) [*in Russian*]

[12] Semoushin, I.V.: Adaptive control for stochastic linear plants under conditions of uncertainty. Nonlinear Dynamical Systems – Qualitative Analysis and Control: Collected Papers, The Institute for System Analysis of the Russian Academy of Sciences **2** (1994) 104–110

[13] Semoushin, I.V., Tsyganova, J.V.: Indirect error control for adaptive filtering. In: Neittaanmaki, P., Tiihonen, T., Tarvainen, P. (eds.): Proc. of the 3rd European Conference on Numerical Mathematics and Advanced Applications. World Scientific, Singapore New Jersey London Hong Kong (2000) 333–340

[14] Semoushin, I., Gorokhov, O.: Computational processes in iterative control design. In: Sloot, P.M.A., Kenneth Tan, C.J., Dongarra, J.J., Hoekstra, A.G. (eds.): Computational Science – ICCS 2002. Lecture Notes in Computer Science, Vol. 2329. Springer-Verlag, Berlin Heidelberg New York Barcelona Hong Kong London Milan Paris Tokyo (2002) 186–195

[15] Semoushin, I.V.: The design of active type systems for the adaptive control and its application to Inertial Navigation Systems. Nonlinear Dynamical Systems – Qualitative Analysis and Control: Collected Papers, The Institute for System Analysis of the Russian Academy of Sciences **2** (1994) 110–115

[16] Lancaster, P., Rodman, L.: Algebraic Riccati Equations. Clarendon Press, Oxford (1995)

[17] Antoniou, A.: Digital filters: analysis and design. MdGraw-Hill, Inc., New York (1979)

[18] Gorokhov, O. Yu.: Investigation of the jointly performed operations of detection, identification and adaptation in discrete stochastic systems. Candidate of Science dissertation (Advisor I.V. Semoushin). Ulyanovsk State University, Ulyanovsk (2002)

[19] Kondratiev, A.E., Fatyanova, O.A.: Start-stop algorithms for adaptive filtering. In: Boulyarski, S.V. (ed.): Young Researchers' Collected Papers of the Ulyanovsk State University. UlSU Publishers, Ulyanovsk (2001) 6–7

[20] Semoushin, I.V. & Polosenko, V.P. 2002. Iterative Cooperative CAD for Avionics Navigation Systems. In R. Goncalves, R. Roy, A. Steigler-Garcao (eds), *Advances in Concurrent Engineering; Proc. intern. conf., Cranfield, July 27–31 2002*: 753–761. Lisse, Abington, Exton (PA), Tokyo: Balkema.

[21] Maybeck, P.S. 1982. *Stochastic models, estimation, and control* **3**. Paris

San Diego San Francisco Sao Paulo Sydney Tokyo Toronto: Academic Press.

[22] Caines, P.: Linear Stochastic Systems. John Willey & Sons, New York Chichester Brisbane Toronto Singapore (1988)

[23] Semoushin, I.V. 1974. On One Approach to Fault Detection in Linear Dynamical Systems with Possible Disturbances. *Automatics and Computing Techniques, The Latvian SSR Academy of Sciences* **4**: 24–30.

[24] Ponyrko, S.A. & Semoushin, I.V. 1973. On the Choice of Start-Stop Algorithm while Minimizing the Square Mean Performance Index. *Autometria, The USSR Academy of Sciences* **2**: 68–74. [*in Russian*]

[25] Soderstrom, T.: Identification of stochastic linear systems in presence of input noise. Automatica **17** (1981) 713–725

[26] Ansay, P., Gevers, M., Wertz, V.: Closed-loop or open-loop models in

identification for control? In: Proc. of the European Control Conference (1999) CD-ROM F544

[27] Semoushin, I.V.: Adaptive Identification and Fault Detection Methods in Random Signal Processing. Saratov University Publishers, Saratov (1985) [*in Russian*]

[28] Semoushin, I.V.: Adaptive control for stochastic linear plants under conditions of uncertainty. Nonlinear Dynamical Systems – Qualitative Analysis and Control: Collected Papers, The Institute for System Analysis of the Russian Academy of Sciences **2** (1994) 104–110

[29] Semoushin, I.V., Tsyganova, J.V.: Indirect error control for adaptive filtering. In: Neittaanmaki, P., Tiihonen, T., Tarvainen, P. (eds.): Proc. of the 3rd European Conference on Numerical Mathematics and Advanced Applications. World Scientific, Singapore New Jersey London Hong Kong (2000) 333–340

[30] Semoushin, I., Gorokhov, O.: Computational processes in iterative control design. In: Sloot, P.M.A., Kenneth Tan, C.J., Dongarra, J.J., Hoekstra, A.G. (eds.): Computational Science – ICCS 2002. Lecture Notes in Computer Science, Vol. 2329. Springer-Verlag, Berlin Heidelberg New York Barcelona Hong Kong London Milan Paris Tokyo (2002) 186–195

[31] Semoushin, I.V.: The design of active type systems for the adaptive control and its application to Inertial Navigation Systems. Nonlinear Dynamical Systems – Qualitative Analysis and Control: Collected Papers, The Institute for System Analysis of the Russian Academy of Sciences **2** (1994) 110–115

[32] Lancaster, P., Rodman, L.: Algebraic Riccati Equations. Clarendon Press, Oxford (1995)

[33] Antoniou, A.: Digital filters: analysis and design. MdGraw-Hill, Inc., New York (1979)

[34] Gorokhov, O. Yu.: Investigation of the jointly performed operations of detection, identification and adaptation in discrete stochastic systems. Candidate of Science dissertation (Advisor I.V. Semoushin). Ulyanovsk State University, Ulyanovsk (2002)

[35] Kondratiev, A.E., Fatyanova, O.A.: Start-stop algorithms for adaptive filtering. In: Boulyarski, S.V. (ed.): Young Researchers' Collected Papers of the Ulyanovsk State University. UlSU Publishers, Ulyanovsk (2001) 6–7

[36] Semoushin, I.V. & Polosenko, V.P. 2002. Iterative Cooperative CAD for Avionics Navigation Systems. In R. Goncalves, R. Roy, A. Steigler-Garcao (eds), *Advances in Concurrent Engineering; Proc. intern. conf., Cranfield, July 27–31 2002*: 753–761. Lisse, Abington, Exton (PA), Tokyo: Balkema.