

Trajectory Tracking by the Interacting Multiple Model Algorithm: Genetic Approach to Improve the Performance

Dmitrii A. Bedin and Alexey G. Ivanov

N.N. Krasovskii Institute of Mathematics and Mechanics
of the Ural Branch of the Russian Academy of Sciences (IMM UB RAS)
Ekaterinburg, Russia

Email: bedin@imm.uran.ru, iagsoft@imm.uran.ru

Abstract—In the paper, we consider an application of the genetic approach to improve the performance of the trajectory tracking procedure. Our aim is to reduce the horizontal position estimation errors. The procedure is based on the well known Interacting Multiple Model algorithm and has many parameters to be adjusted, e.g., elements of the transition probability matrix. A genetic tuning algorithm is elaborated and some numerical experiments on simulated aircraft trajectories are made.

I. INTRODUCTION

The process of aircraft trajectory tracking [1] can be represented as follows. A track of some aircraft consists of a sequence of measurements $\{z_j\}_{j=1}^n$, where n is the total length of the sequence. The measurement z_j at an instant t_j contains the random error w_j and is related to the vector $x(t_j)$ of the true aircraft position. We consider only the simplest case of the horizontal plane motion of aircraft (thus, the vectors x , z , and w belong to \mathbf{R}^2) and the additive random error:

$$z_j = x(t_j) + w_j. \quad (1)$$

There are two types of the random errors: the usual random errors and the outliers, which have another probabilistic distribution with greater covariance characteristics and a systematic component. The outliers are rare events with small probability.

Any real-life estimate \hat{x}_j of the aircraft position vector $x(t_j)$ is a function of the history of measurements $\{z_k : t_k \leq t_j\}$ and *a priori* information such as properties of the aircraft motion and the random errors. It is desirable to decrease a “distance” between the true state $x(t_j)$ and the estimate \hat{x}_j . Commonly, the “distance” is the “total” mean squared error (MSE) $\mathbf{E} \{\|\hat{x}_j - x(t_j)\|^2\}$, but, in some cases, it is useful to control other metrics.

Aircraft trajectories are not equal in terms of the quality of estimation. The MSE of the position estimates depends on the MSE of the measurements: the higher the latter, the higher the former. This is the reason to introduce [2] the normalized position error (NPE) as the ratio between the MSE of the position estimates and the MSE of the measurements. Moreover, the parts of one trajectory are not equal: in the simplest case, the measurement MSE can vary in time or

depend on the position of the aircraft with respect to the radar. So, the NPE has to be considered as a function of time.

Another important fact: it is much simpler to make an estimate with small MSE in straight line segments of motion than in circular ones. The most difficult thing for estimating is the transitions between segments of the different motion types. Usually, the MSE of a trajectory tracking algorithm has the “peak” after the maneuver starts. Also, there is a time required to decrease the MSE from the peak value to the level that is typical for the steady motion.

This work is a result of collaboration with NITA, LLC, which is the leading Russian company in the area of air traffic management (ATM) systems. The trajectory tracking complex by NITA is based on the well-known interacting multiple model (IMM) algorithm. The engineers of the company in their work observe all the typical behaviours of the tracking estimates as “peaks” and address a natural question how to “smooth” them and improve the performance. The algorithm behaviour depends on many parameters of the IMM method. Thus, the problem of setting the parameters of IMM arises.

II. PARAMETERS OF THE TRAJECTORY TRACKING ALGORITHM

The real trajectory tracking algorithm of the NITA company consists of two parts: the preprocessor, which is responsible for correct mixing of data from different sensors and robustness against outliers, and the IMM filtering procedure whose main purpose is to produce a good estimate of the current aircraft position despite of its maneuvers.

We consider the preprocessor part as a black box: it was given by the NITA company as an executable file and we could only change its parameters. There are discrete and continuous parameters; their column vector will be denoted as θ^{pre} .

In contrast, the next part (i.e., the IMM procedure) has an explicit description. The vector of its parameters will be labeled as θ^{imm} . All the parameters are continuous.

The core of the IMM procedure [1], [3] are several Kalman filters, each of which corresponds to its own motion model. These filters produce the estimates at the last measurement instant, then the algorithm produces the joint estimate.

The algorithm of the NITA company deals with several models of the aircraft dynamics, which are widely used in practice (see [1], [4], [5]): constant velocity (CV) motion, constant acceleration (CA), and coordinated turn (CT). Since the CA and CT dynamics have a nonlinear description, the IMM extended Kalman filter (IMM EKF) modification is used. Denote the total number of the models as K .

All the dynamics can be represented [4] by the equation

$$\xi_{j+1}^{(k)} = f^{(k)}(t_{j+1} - t_j, \xi_j^{(k)}) + g^{(k)}(t_{j+1} - t_j, \xi_j^{(k)})\nu^{(k)},$$

where $\xi_j^{(k)} \in \mathbf{R}^{d_k}$ is the state vector of k th model at the instant t_j which includes the position vector $x(t_j)$ of the aircraft, its velocity vector, and such specific variables as angular speed or acceleration; t_{j+1} and t_j are successive time instants; $f^{(k)}$ is the transitional function of the k th dynamics; $g^{(k)}$ is the process noise function of the k th dynamics; and $\nu^{(k)} \in \mathbf{R}^{m_k}$ is the random process noise whose characteristics are specific for the k th dynamics.

The parameters θ^{imm} of the IMM method consist of two main parts. The first one is the process noise covariances $Q_{(k)} = \mathbf{E}\{\nu^{(k)}\nu^{(k)\top}\}$. In our algorithm, a simple version is used:

$$Q_{(k)} = \text{diag}\left\{\sigma_{1(k)}^2, \dots, \sigma_{m_k(k)}^2\right\}.$$

The second part of θ^{imm} is the parameters of the transition probability matrix (TPM). Since the time intervals between consecutive measurements vary, the transition probabilities also have to vary. We use a simple version of this dependence:

$$P(t_{j+1} - t_j) = Ie^{-(t_{j+1}-t_j)/\lambda_P} + P_\infty(1 - e^{-(t_{j+1}-t_j)/\lambda_P}),$$

where P is the TPM, P_∞ is the limit value of TPM as the time interval tends to infinity, I is the identity matrix, and λ_P is the constant of the convergence rate.

The constant λ_P is included into θ^{imm} directly, but this is not the case for P_∞ . Indeed, every row p_k of P_∞ is a list of the transition probabilities p_{kj} from the k th dynamics to the j th one with the constraints:

$$p_{kj} \in [0, 1], \quad \sum_{j=1}^K p_{kj} = 1. \quad (2)$$

Such a set is known as a simplex. Equality constraints (2) are inconvenient for numerical optimization, if we use p_{kj} as parameters in a straightforward way. But we can simplify this problem using some representation of p_{kj} by means of other parameters η_{kj} with the box constraints.

There are two useful facts. First, the uniform distribution over simplex (2) can be modelled by the Dirichlet distribution $\text{Dir}(\alpha_1 = 1, \dots, \alpha_n = 1)$, which is described by n exponential random values [6], [7]:

$$\begin{aligned} \eta'_{kj} &\sim \text{Exp}(1) \implies \\ &\implies \left(\frac{\eta'_{k1}}{\sum_{j=1}^K \eta'_{kj}}, \dots, \frac{\eta'_{kn}}{\sum_{j=1}^K \eta'_{kj}} \right) \sim \text{Dir}(1, \dots, 1). \end{aligned}$$

Second, using the inverse transform sampling [8], the exponential random value can be represented as follows: $\eta'_{kj} = -\log \eta_{kj}$ where $\eta_{kj} \sim \mathbf{U}(0, 1)$ (the uniform distribution over the interval $(0, 1]$). On the basis of these facts, we use the following representation for the probabilities

$$p_{kj} = \frac{-\log \eta_{kj}}{-\sum_{\ell=1}^K \log \eta_{k\ell}}, \quad (3)$$

where $\eta_{kj} \in (0, 1]$ are directly adjusted parameters that are included in θ^{imm} . This representation is convenient and can be found, for example, in [9].

If the adjusted parameters η_{kj} , $j = \overline{1, K}$, spread uniformly over the box $(0, 1] \times \dots \times (0, 1]$, then the probabilities p_{kj} , $j = \overline{1, K}$, also spread uniformly over simplex (2). In other words, this representation allows us to avoid an ‘‘artificial’’ concentration of p_{kj} in some part of the simplex.

The joint vector θ^{imm} is

$$\theta^{imm} = [\sigma_{1(1)} \quad \dots \quad \sigma_{m_K(K)} \quad \lambda_P \quad \eta_{11} \quad \dots \quad \eta_{KK}]^\top.$$

The total parameter vector θ consists of θ^{imm} and θ^{pre} .

III. PARAMETER ADJUSTMENT METHODOLOGY

The application of IMM filtering procedure to the trajectory tracking has been widely presented in literature [1], [2], [4], [9]–[11]. In one part of the publications, the parameters are chosen *a priori* and do not change during the IMM work [2], [12], [13]. In another part, adaptive versions of the IMM algorithm are considered, where parameters are adjusted while the tracking process runs [9], [14].

In our work, we keep a non-adaptive strategy and consider the parameters of IMM as *a priori* chosen (and they do not change during tracker executes) because the current trajectory tracking program of the NITA company is based on a non-adaptive version of the IMM method. Our aim is to improve the existent tracking complex, but not to develop a new one. So, we try to find the ‘‘best’’ or ‘‘good’’ parameters that enable the tracking program to work well in all the motion segments of all aircraft types.

The elements of TPM can be the parameters most affecting the behavior of the IMM algorithm. There are several publications [4], [9], [15] devoted to the design of TPM. The authors of [15] suggest a generic Bayesian framework for the online TPM adjustment and suppose three practical numerical schemes. In [9], the TPM is modelled by the Dirichlet distribution like (3) and analytical formulas are suggested to integrate of the common density of the state, mode, and TPM sequence.

We optimize not only the TPM elements but also many different heterogeneous parameters such as process noise covariances and discrete tuning constants of the IMM preprocessor θ^{pre} . Thus, we have to use a simple optimization method that can handle both continuous and discrete variables and does not rely on the existence of the cost function derivatives in its work. A genetic algorithm matches these conditions [16], [17], so we decided to use it.

The genetic approach has been used for the parameter adjustment in the trajectory tracking problems. For example,

in [12], the authors describe a version of the IMM method, namely the Fuzzy IMM (FIMM), and adjust the process noise covariances using the genetic algorithm. In [13], the genetic algorithm sets the switching parameters in the variable structure IMM. Genetic algorithms in these papers are used in off-line regime: the parameters do not change as the tracker executes. Unlike this, in [14], the genetic algorithm is used on-line for the data association. All these examples have a common point: the genetic algorithm optimizes parameters that have a complex and unobvious influence on the performance. This is very close to our case.

The difficult point in our study is to define what “the best” and “good” tracking quality are. There are many criteria of the trajectory tracking quality in ATM practice. We follow EUROCONTROL documents [18], [19] and the criteria in them as basic. In our investigation, we consider only the position error, which is the difference between the true position of the aircraft and the estimated one.

The position errors along and across the trajectory are distinguished [18] and the performance criteria in these directions are distinctly formulated [19]. Moreover, different quality norms stand for the different motion segments and transitions between segments [19]. The measure of these errors in the standard [19] is the root mean squared (RMS) error.

We decided to use such norms as the reference in our quality criteria. Following the standard [19], for each measurement instant t_j , we can assign the norm values for deviation between the ideal trajectory position $x(t_j)$ and the estimated one \hat{x}_j : $\tilde{\sigma}_j^l$ for the deviation along the trajectory and $\tilde{\sigma}_j^n$ for the across deviation. The norm values depend on the motion type of the ideal trajectory $x(\cdot)$ at the measurement instant t_j and are adjusted in accordance with the standard deviation of the measurement noise. Our analog of the NPE is based on the ratios between the tracking error value and the quality norm:

$$r_j^l = \frac{|e_j^l \top (\hat{x}_j - x(t_j))|}{\tilde{\sigma}_j^l}, \quad r_j^n = \frac{|e_j^n \top (\hat{x}_j - x(t_j))|}{\tilde{\sigma}_j^n}. \quad (4)$$

Here, e_j^l is the unit vector along a certain trajectory $x(\cdot)$ at the point $x(t_j)$ and e_j^n is the corresponding unit vector for the across direction. Their calculations are based on the velocity vector of the trajectory $x(\cdot)$ at t_j .

Ideally, we want to decrease the root mean squared ratios

$$\sigma^l = \sqrt{\mathbf{E}_x \mathbf{E}_{t_j} \mathbf{E}_{w_j} \{(r_j^l)^2\}}, \quad \sigma^n = \sqrt{\mathbf{E}_x \mathbf{E}_{t_j} \mathbf{E}_{w_j} \{(r_j^n)^2\}}, \quad (5)$$

where \mathbf{E}_x , \mathbf{E}_{t_j} , \mathbf{E}_{w_j} are expectations relative to the trajectories $x(\cdot)$, their measurement instants t_j , and the measurement random errors w_j , respectively. But the distribution over the trajectories $x(\cdot)$ is unknown and there is a problem with the distribution over w_j since, in the presence of outliers, we do not know exactly their properties.

Usually, the expectations in (5) are substituted by the means over a sufficiently wide set $\{x(\cdot)\}$ of trajectories. In the present work, for simplicity, we decide to use one but a long trajectory that has all the types of motion segments and their transitions and a wide range of the speed regimes. To minimize the risk of

the overfitting (see [20]), we use one trajectory as the training set (the training trajectory) but check the quality on some other test trajectory.

In order to substitute expectation over the measurement errors in (5), a set (or package) of the measurement tracks $\{\{z_{ij}\}_{j=1}^{n_i}\}_{i=1}^N$ is considered. Every track i is the sequence of measurements $\{z_{ij}\}_{j=1}^{n_i}$ obtained by a specific run of the measurement noise $\{w_{ij}\}_{j=1}^{n_i}$ at the measurement instants $\{t_{ij}\}_{j=1}^{n_i}$. The training and test trajectories have their own measurement track packages.

Finally, after substitutions in (5), we have these total RMS errors in the along and across directions:

$$\hat{\sigma}^l = \sqrt{\frac{1}{N} \sum_{i=1}^N \frac{1}{n_i} \sum_{j=1}^{n_i} (r_{ij}^l)^2}, \quad \hat{\sigma}^n = \sqrt{\frac{1}{N} \sum_{i=1}^N \frac{1}{n_i} \sum_{j=1}^{n_i} (r_{ij}^n)^2}. \quad (6)$$

To show separately the behaviour on the motion segments and their connections, we use the RMS error as a function of time along the trajectory $x(\cdot)$:

$$\hat{\sigma}^l(x(\cdot), t) = \sqrt{\frac{1}{N(t)} \sum_{i \in I(t)} (r_{ij}^l)^2}. \quad (7)$$

Here, each instant t is considered to belong to the union $\cup_{i,j} t_{ij}$ of all the measurement instants, $I(t) = \{i : t_{ij} = t\}$ are the indices of t_{ij} that coincide with t , and $N(t) = \#I(t)$ is the cardinality ($\#$) of this set. In addition, a quantile analogue is considered in order to check the guaranteed (with ε probability) level of deviations:

$$q^{\varepsilon}(x(\cdot), t) = \min \left\{ q : \frac{\#\{i \in I(t) : r_{ij}^l \leq q\}}{N(t)} \geq \varepsilon \right\}. \quad (8)$$

The functions $\hat{\sigma}^n(x(\cdot), t)$, $q^{n\varepsilon}(x(\cdot), t)$ for across deviations are defined in a similar way.

There are a lot of criteria important for the practice. Thus, the multicriterial optimization technique is needed as, for example, in [21].

IV. GENETIC PROCEDURE

The genetic algorithm is based on modeling the process of the natural evolution with the inclusion of elements of directional search [16], [17]. The scheme of the algorithm workflow is presented in Fig. 1. In this section, we describe subprocedures of the algorithm according to their sequence in the total procedure and discuss their details. The term “genotype” denotes the parameter vector θ of the tracking algorithm when the term “gene” stands for some specific parameter θ^k . An “individual” a corresponds to a point θ_a in the space of the parameter vectors. The term “generation” stands for the iteration number of the genetic algorithm workflow and the set of all individuals at this iteration.

A. Initialization

The initialization of the algorithm is performed from one individual. After the creation of one individual, the population starts to spread: a certain number of individuals is generated,

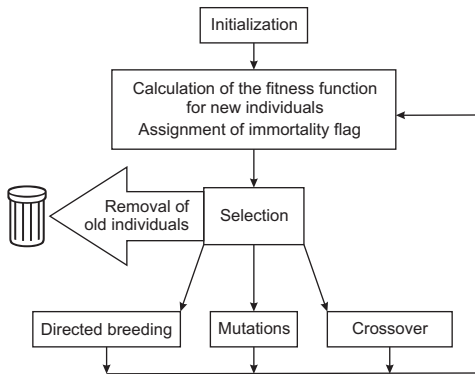


Fig. 1. Genetic algorithm flowchart

whose genes differ from the parameters of the first individual by the current standard step of variation of this gene (or, sometimes, by 1/3 of the current step).

B. Evaluation of New Individuals

In every generation, new individuals are created. For each newly formed individual, the quality criteria are evaluated using the trajectory tracking program.

To speed up the calculations, we use parallel computation of individuals. For debugging, we use a personal computer (with OS Windows) with a multi-core processor. For the main experiment we use the “Uran” supercomputer of N.N. Krasovskii Institute of Mathematics and Mechanics (with OS Linux).

The evaluation process is based on the one ideal training trajectory, which represents the typical motion of a civil aircraft, and a package of N measurement tracks. The tracks differ in the measurement time step and the characteristics of the random error.

For each track in the package, ratios (4) are calculated for all the measurements of the trajectory. Based on these data, the genetic algorithm calculates the values of the individuals’ criteria. For robustness, we do not use RMS criterion (6) exactly in the training process, but use its analogs that are based on the mean and maximum absolute values.

The following criteria are used: the average over the package of the maximum over time in the along and across channels:

$$c_2 = \frac{1}{N} \sum_{i=1}^N \max_j r_{ij}^l, \quad c_3 = \frac{1}{N} \sum_{i=1}^N \max_j r_{ij}^n,$$

the average of the package in the along and across channels:

$$c_4 = \frac{1}{N} \sum_{i=1}^N \frac{1}{n_i} \sum_{j=1}^{n_i} r_{ij}^l, \quad c_5 = \frac{1}{N} \sum_{i=1}^N \frac{1}{n_i} \sum_{j=1}^{n_i} r_{ij}^n,$$

the total average of the package:

$$c_6 = \frac{c_4 + c_5}{2},$$

and the maximum over package of the average over time in both channels:

$$c_7 = \max_i \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{r_{ij}^l + r_{ij}^n}{2}.$$

The criterion c_1 is used in the algorithm as the fitness function h . It is the complex empirical criterion that takes into account several criteria with weights:

$$h = c_1 = \frac{5}{12} \left(\frac{c_2 + c_3}{10} + 2c_6 + \frac{c_7}{5} \right).$$

C. Immortal Individuals Assignment

The individuals that are good by any criteria have to be preserved in the population. To do this, these individuals are marked by “immortality” flags. Two levels of “immortality” are introduced: the “best immortality” and the “good immortality”.

After all the individuals are evaluated, the extreme values of the criteria for the entire population are determined:

$$C_{k \min} = \min_{a \in A} c_k(a), \quad C_{k \max} = \max_{a \in A} c_k(a), \quad k = \overline{1, 7}.$$

Here, A is the set of all individuals in the given generation; $c_k(a)$ is the value of criterion c_k for the individual a .

If the minimum of any criterion is reached at an individual, then this individual is marked with the “best immortality” flag. The individual a of “good immortality” is such that at least one criterion value $c_k(a)$ satisfies the inequality

$$c_k(a) \leq \lambda_{im} C_{k \max} + (1 - \lambda_{im}) C_{k \min}.$$

Here, λ_{im} is the parameter of proximity of the “good immortality” to the “best immortality”, the same for all the criteria.

The initial value of the proximity parameter λ_{im} is equal to 0.01. During further work of the algorithm, if the number of individuals of “good immortality” increases greatly in some generation, then the value of the parameter decreases by half.

D. Selection

During selection, the “old” individuals (the age of the individual is greater than the current established level T_L) are destroyed, except for those which have the “immortality” flag. If the number of individuals exceeds the assigned maximum value after destruction of the “old” individuals, the “elderly” individuals (for which the difference between the current age and T_L is small) can be destroyed.

The maximum assigned age T_L for an individual a is calculated at each generation on the base of $h(a)$:

$$T_L(a) = 1 + M_A \frac{C_{1 \max} - h(a) + 1}{C_{1 \max} - C_{1 \min}}.$$

Here, the constant M_A is the maximum possible age (the parameter of the algorithm).

E. Crossover

Any two individuals whose age are greater than 2 can be crossed over. Pairs of individuals chosen by the random number generator participate in the crossing gene exchange with the creation of a new individual: each gene of the new individual is randomly inherited either from the parent 1 or from the parent 2. If the child is a clone of one of the parents, then the individual is not formed.

F. Directed Breeding

Two random chosen individuals with different fitness function values can participate in breeding. Denote the individual with the best fitness as b and the individual with the worst fitness as w . The values of the i th gene are θ_b^i and θ_w^i , respectively. The value of this gene for the child in the breeding process is chosen randomly in the interval $[\frac{1}{2}(\theta_w^i + \theta_b^i), 2\theta_b^i - \theta_w^i]$ from a uniform distribution over it.

G. Mutation

At every generation, the immortal individuals mandatorily undergo the mutation process while other individuals undergo mutations only with some probability.

If the current age of an individual is less than the maximum assigned age T_L , then the individual undergoes an “usual” mutation: the value of each gene is randomly changed by a value no greater than the current step of variation of this gene.

To revive genetic diversity in the case of “stagnation” of the population, the individual can be subjected to “micromutation” or “macromutation” if the age of the individual exceeds T_L (it is possible for the immortal individuals).

In the micromutation, the value of each gene is changed by a random value with the magnitude $1/100$ of the current step of variation of this gene.

When the macromutation occurs, the value of only one randomly selected gene is changed by a random value with the magnitude 100 of the current step of gene variation.

H. Adaptive Choice of the Basic Variation Step of Each Gene

The basic step size of gene variation is adjusted depending on the history of variations. If the new individual has a better fitness value than its parent and it has been created with the step size different from the basic one, then, further, this step size will be assigned as the basic. Otherwise, the basic step is not changed.

I. Intermediate Populational “Shaking”

The “shaking” is used when the evolution stalls. In shaking, all the individuals that are not “immortal” are removed. For the remained ones, the genetic algorithm starts new spreading that is similar to the initialization from the first individual. The steps of variation of these descendants are changed to the initial step of variation, and then the genes are varied. The “best immortality” individuals produce more descendants than the “good immortality” ones. The intermediate shaking algorithm is called up every 100th generation.

V. NUMERICAL EXPERIMENTS

The training was performed on a package of 20 measurement tracks ($N = 20$) built on the basis of the single ideal trajectory with duration 2430 s. The RMS for the random measurement errors depended on the track label i as follows: $\sigma_i = 6i$ m, $i = \overline{1, 20}$, hence covered the range from 6 to 120 m. Tracks with the labels i greater than 10 contain outliers. The probability of the outliers is $1/20$, and the RMS error of the outliers is five times greater than the regular RMS. The tracks

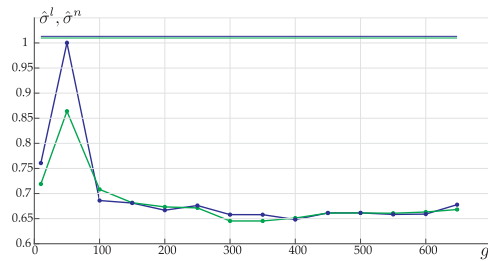


Fig. 2. Total RMS error as a function of the generation number g in along (green) and across (blue) channels, for the test package

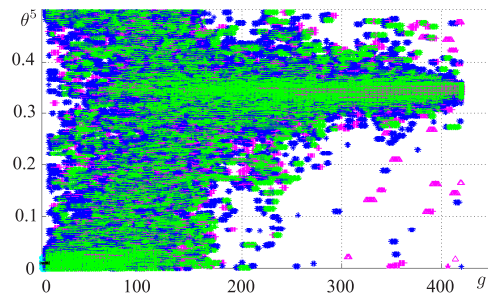


Fig. 3. Time dynamics of gene θ^5 . Gene value as a function of the generation number g

with odd numbers have the measurement time step 4 s (the airport zone is simulated according to [19]) and the tracks with even numbers have the measurement time step 12 s (the en-route radar is simulated according to [19]).

The initial point of the evolution process was the “IMM0” parameters, which are close to the parameters used by the NITA company in its tracking program. The evolution lasted during 648 generations and was stopped when the total RMS criterion (6) became to increase confidently on the test trajectory as one can see in Fig. 2.

Fig. 2 shows the results for the test package of tracks: the total RMS errors in the across (7) and along channels are shown as functions of the generation number. The horizontal lines at the top of the figure are the criterion levels for the measurements. They are depicted for reference. This package consisted of 120 tracks with duration 1280 s and the measurement RMS error σ_i covered the range $[10, 120]$. Half of the trajectories had outliers, the measurement time step was either 4 s or 12 s. The parameters that are best for this generation according to the c_6 criterion were used to restore tracks in the test package.

An example of a noticeable change in the distribution of the gene θ^5 depending on generation number g is shown in Fig. 3. The initial value of gene θ^5 was 0.01. At the 419th generation, the immortal (and others) individuals of the population were concentrated around the value of 0.35.

Fig. 4 shows graphs of the 95% quantiles $q^{l,0.95}$ as functions of the measurement time for the test track package. This package consists of 240 tracks with duration 2070 s and with σ_i covering the range $[5, 105]$. The most of the trajectories are with outliers, the step between consecutive measurements

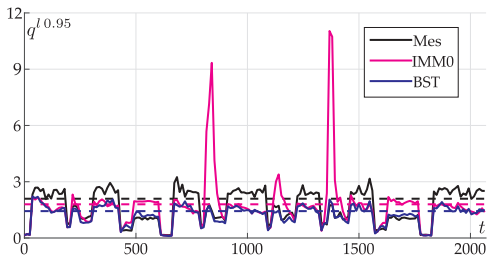


Fig. 4. 95% quantile $q^{l0.95}$ as a function of the measurement time in the along channel, for the test package

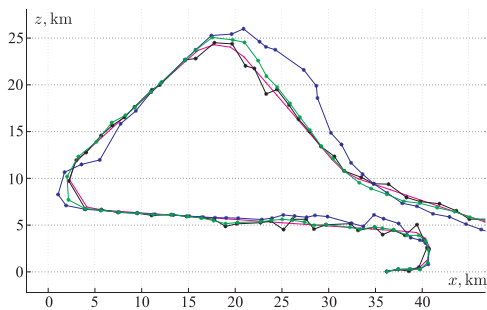


Fig. 5. A fragment of the test trajectory on plane: the ideal trajectory (magenta), the measurements (black), and the tracking results before (blue) and after (green) optimization

is 12 s. The black lines (Mes) are for the level of measurements. The magenta color (IMM0) shows the results for the initial IMM0 parameters θ . The dark blue lines (BST) are the results obtained for the best parameters by the c_6 criterion for the 350th generation of evolution.

In Fig. 5, a fragment of plane with these test trajectories is shown. The ideal test trajectory depicted with magenta and has straight line, circular, and acceleration motion sections. The measurements are shown as black stars. The blue line is a one realization of the tracking result with IMM0 parameters before optimization. The green line corresponds to the tracking result with optimized BST parameters.

VI. CONCLUSION

As a result of the numerical experiment, we have found the set of the parameter vectors θ (the set of the “immortal” individuals) that outperform the initial IMM0 parameters. The tracking algorithm with these parameters shows less relative (with respect to the norms [19]) RMS error in the horizontal plane in both along and across channels. The errors are decreased both in the total value per trajectory and in the problematic parts of the trajectories, i.e. in the connections of the motion segments. The influence of outliers is also decreased.

However, we are aware of the fact that we used too small both training and test sets in our numerical experiment. In our future work, we are going to use bigger sets and add the real data from radars.

ACKNOWLEDGMENT

We would like to thank NITA, LLC (<http://www.nita.ru>) for the problem formulation and the fruitful collaboration.

REFERENCES

- [1] Y. Bar-Shalom and W. D. Blair, *Multitarget-Multisensor Tracking: Applications and Advances, Vol. III*. Norwood, MA: Artech House, 2000.
- [2] A. Munir, J. A. Mirza, and A. Q. Khan, “Parameter adjustment in the turn rate models in the interacting multiple model algorithm to track a maneuvering target,” in *Proc. IEEE International Multi Topic Conference (INMIC 2001), Technology for the 21st Century*, Lahore, Pakistan, 2001, pp. 262–266.
- [3] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking. Part V. Multiple-model methods,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.
- [4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory, Algorithms, and Software*. New York: Wiley, 2001.
- [5] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking. Part I. Dynamic models,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.
- [6] W. J. Reed, “Random points in a simplex,” *Pacific J. Math.*, vol. 54, no. 2, pp. 183–198, 1974.
- [7] S. S. Wilks, *Mathematical Statistics*. New York–London: J. Wiley and Sons, 1962.
- [8] L. Devroye, *Non-Uniform Random Variate Generation*. New York: Springer-Verlag, 1986.
- [9] A. Doucet and B. Ristic, “Recursive state estimation for multiple switching models with unknown transition probabilities,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 1098–1104, 2002.
- [10] W. Youn and H. Myung, “Robust interacting multiple model with modeling uncertainties for maneuvering target tracking,” *IEEE Access*, vol. 7, pp. 65 427–65 443, 2019.
- [11] G. Soldi, F. Meyer, P. Braca, and F. Hlawatsch, “Self-tuning algorithms for multisensor-multitarget tracking using belief propagation,” *IEEE Transactions on Signal Processing*, vol. 67, no. 15, pp. 3922–3937, 2019.
- [12] B. J. Lee, J. B. Park, H. J. Lee, and Y. H. Joo, “Fuzzy-logic-based IMM algorithm for tracking a manoeuvring target,” in *IEEE Proc. – Radar, Sonar and Navigation*, vol. 152, no. 1, Feb. 11, 2005, pp. 16–22.
- [13] A. Vahabian, A. K. Sedigh, and A. Akhbardeh, “Optimal design of the variable structure imm tracking filters using genetic algorithms,” in *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.*, vol. 2, 2004, pp. 1485–1490 Vol.2.
- [14] M. A. Zaveri, S. N. Merchant, and U. B. Desai, “Tracking multiple point targets using genetic interacting multiple model based algorithm,” in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. 3, 2004, pp. III–917.
- [15] V. P. Jilkov and X. R. Li, “Online Bayesian estimation of transition probabilities for markovian jump systems,” *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1620–1630, Jun. 2004.
- [16] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Heidelberg: Springer-Verlag, 1996.
- [17] D. Whitley, “A genetic algorithm tutorial,” *Stat. Comput.*, no. 4, pp. 65–85, 1994.
- [18] *EUROCONTROL Specification for ATM Surveillance System Performance*, Std. [Online]. Available: <https://www.eurocontrol.int/publication/eurocontrol-specification-atm-surveillance-system-performance-esassp>
- [19] *EUROCONTROL Standard for Radar Surveillance in En-Route Airspace and Major Terminal Areas*, Std. [Online]. Available: <https://www.eurocontrol.int/publication/eurocontrol-standard-radar-surveillance-en-route-airspace-and-major-terminal-areas>
- [20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2013, Springer Series in Statistics.
- [21] J. S. Carlton, “Propeller design,” in *Marine Propellers and Propulsion (Fourth Edition)*, J. S. Carlton, Ed. Amsterdam, The Netherlands: Butterworth-Heinemann, 2019, ch. 22, pp. 469–497.