

Multicriteria Genetic Optimization Procedure for Trajectory Tracking by the Interacting Multiple Model Algorithm ^{*} ^{**}

Dmitrii A. Bedin^[0000-0002-3790-2943] and
Alexey G. Ivanov^[0000-0002-5852-7273]

N.N. Krasovskii Institute of Mathematics and Mechanics
of the Ural Branch of the Russian Academy of Sciences,
Yekaterinburg, Russia
{bedin, iagsoft}@imm.uran.ru

Abstract. Trajectory tracking algorithms have many parameters that influence their behavior. Adjustment of these parameters is complex because of their heterogeneous nature and the fact that the tracking quality depends on the parameters in a complicated and mutual manner. In the air traffic control area, there are many tracking quality criteria. An optimization program for this problem is elaborated on the basis of the genetic approach. With its use, the tracking quality of the real trajectory tracker has been significantly improved.

Keywords: Multicriteria optimization · Trajectory tracking · Genetic algorithm.

1 Problem Overview

Trajectory tracking algorithms are used in applied problems of surveillance of a moving object. During surveillance, observations are made at discrete time instants $\{t_j\}$. At a time instant t_i , a part of an object state vector $x(t_i)$ (in the formula, the “geometrical” part $x_G(t_i)$ of x is considered) is observed:

$$y(t_i) = x_G(t_i) + \eta(t_i).$$

The measurement $y(t_i)$ includes the random error $\eta(t_i)$. A trajectory tracking algorithm maps the sequence $\{y(t_i)\}_{i=1}^n$ of the measurements up to the current instant t_n to an estimate $\hat{x}(t_n)$ of the state vector $x(t_n)$.

The classical algorithm for the trajectory tracking is the Kalman filter [1]. However, in the case of a maneuvering object, the problem becomes more complicated and demands more complex algorithms. In the area of air traffic management (ATM), the interacting multiple model (IMM) algorithm is the state-of-the-art solution [1, 7]. The IMM is based on the hidden Markov model approach.

* The work was performed as part of research conducted in the Ural Mathematical Center.

** Copyright ©2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The authors collaborate with the NITA company (Russia, Saint-Petersburg, <http://nita.ru>) which has the leader position in Russia in the ATM solutions. This company elaborated its own trajectory tracking program on the basis of the IMM algorithm. So, NITA is interested in optimizing the program work.

In the present paper, we consider the trajectory tracking program as a black box and optimize its parameters without using any specific features of the IMM algorithm (in contrast to, for example, [4]). We keep the straightforward approach to optimization as we do not want to deviate from the existent peculiarities and parameters of the NITA program since they are grounded by practical demands of the existent ATM system.

Unfortunately, we cannot fully describe the IMM implementation in the NITA program and the IMM parameter list since we have the non-disclosure agreement. But the structure of the IMM variables' dependence on parameters is as follows. There are two main parts of the parameters: the first part governs the process noise covariance matrices Q_k (for all $k = \overline{1, m}$, i.e., for all the dynamics in the IMM method); the second one tunes the transition probabilities matrix Π (TPM), whose elements $\pi_{k\ell}$ are the probabilities of switches from k th to the ℓ th dynamics. Due to heterogeneity of the parameters and peculiarities of the parametrization, the influence of the parameters on the estimate sequence $\{\hat{x}(t_i)\}$ is complex. For this reason, we decide to use the genetic approach [8] to the optimization. Genetic algorithms have been applied earlier to the IMM parameter adjustment, for example, in [10]. The authors also created a program for optimization of a trajectory tracker by the genetic algorithm (see [2]). In the present work, the program has been adapted to the real NITA program.

In the optimization, simulated tracks are used. The model dynamics of an aircraft is as follows:

$$\begin{cases} \dot{x}_N = v \cos \varphi, \\ \dot{x}_E = v \sin \varphi, \\ \dot{\varphi} = \frac{u}{v}, \\ \dot{v} = w. \end{cases} \quad (1)$$

Here, x_N , x_E are the north and east coordinates of the aircraft in the plane, they make up together the “geometric” coordinate vector $x_G(t_i) = [x_N(t_i) \ x_E(t_i)]^T$. The symbols v and φ are the velocity magnitude and the path angle, they make up together the vector $x_V(t_i) = [v(t_i) \ \varphi(t_i)]^T$ of the velocity elements. The u and w are the lateral and longitudinal accelerations.

This dynamics is compliant with the common concepts of the aircraft motion [1]. The motion with dynamics (1) with constant values of $u(t)$ and $w(t)$ is well known: if $u = w = 0$, it is the constant velocity (CV) motion; in the case of $u \neq 0, w = 0$, the motion is termed as the coordinated turn (CT); and if $u = 0, w \neq 0$, it is the constant acceleration (CA) [1, 7]. We use only these cases in our simulations with analytic integration of equations (1) from [3]. The NITA implementation of the IMM is based on the CV, CT, and CA dynamics as well.

2 Performance Criteria and Optimization Problem

2.1 Performance Criteria in Target Tracking

In the ATM, there are conventional standards of the trajectory tracking quality. All of them are based on the analysis of the differences $\{\hat{x}(t_i) - x(t_i)\}$ between the estimated $\hat{x}(t_i)$ and the true $x(t_i)$ states, which are collected along the trajectory. Following the standard [12], these differences are projected to the along and across directions with respect to the trajectory, making the so-called longitudinal and lateral channels. Also, it is conventional to consider the differences of the magnitude of velocity and the path angle. The standard [13] introduces norms on deviations in the channels in terms of the root mean squared (RMS) error depending on the segment of the aircraft motion and the zone of observation (an airfield or en-route zone). In addition, in [13], there are directives on the durations for the transitional process after the start and the end of some maneuver.

Unfortunately, there are some questions about the usage of the standards in real and model circumstances.

1. The accuracy of sensors is very different and differs from the one described in the standards.
2. The maneuverability of aircraft is different too. This influences the estimate error level and duration of the transitional processes.
3. Transitions between some motion segments are not described in the standards.

An attempt to answer these questions leads the authors to the use of the Cramér–Rao lower bound (CRLB) instead of the standards.

2.2 Cramér–Rao Lower Bound as Universal Standard

The CRLB is the lower bound for the mean squared error (in a scalar case) and the covariance matrix of the error (in a vector case) for the unbiased estimates. The following matrix inequality holds [6]:

$$\mathbf{E} \{ (\hat{x}(t_i) - x(t_i))(\hat{x}(t_i) - x(t_i))^T \} \succcurlyeq J(t_i).$$

The sign \succcurlyeq denotes a partial order for symmetric matrices. Let $A, B \in \mathbb{S}^{m \times m}$, $x \in \mathbb{R}^m$, then $A \succcurlyeq B \Leftrightarrow x^T A x \geq x^T B x$. The matrix $J(t_i)$ reflects the information in the preceding measurements $\{y_j\}_{j=1}^i$ and can be calculated in a recurrent way (see [9]).

Though the estimates of the real algorithms including the IMM are biased, the CRLB shows the adequate behavior of the estimation error according to maneuvers and time. In the segments of a constant control, the CRLB specifies the error level that is hard to outperform. In Figure 1, the CRLB (red) and RMS deviation of the IMM algorithm (magenta) for the longitudinal channel are shown for the trajectory with one switch from the CV motion ($u = w = 0$)

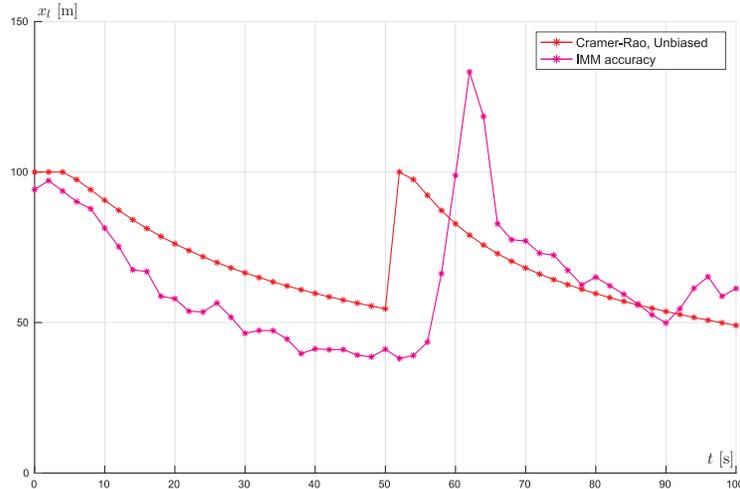


Fig. 1. The CRLB and the RMS deviation for one simulated trajectory.

to the CT motion ($u \neq 0$, $w = 0$) at $t = 50$ s. As one can see in the figure, the decrease of the RMS error of the IMM filter after the maneuver is in accordance with the CRLB behavior.

A recurrent procedure is elaborated that calculates the matrices $\{J(t_i)\}$ for model tracks in the simulations. The analytical integrals from [3] for dynamics (1) are used in it.

As the criteria for the optimization program, we decide to use the RMS deviation for values of this type:

$$X_l(t_i) = \frac{x_l(t_i)}{\sqrt{e_l^T(t_i)J(t_i)e_l(t_i)}},$$

where

$$x_l(t_i) = e_l^T(t_i)(\hat{x}(t_i) - x(t_i)).$$

Here, X_l is the relative longitudinal deviation, that is, the ratio of the “absolute” (in meters) longitudinal deviation $x_l(t_i)$ and the corresponding “projection” of the CRLB matrix onto the longitudinal channel defined by the longitudinal direction vector $e_l(t_i) \in \mathbb{R}^4$. The vector $e_l(t_i)$ consists of the tangential vector to the curve $x_G(\cdot)$ in the plane x_N, x_E in the first two coordinates and two zeros in the second two coordinates (the x_V elements). The analogous formulas are introduced for the lateral (X_n), the velocity magnitude (X_v), and the path angle (X_φ) channels using the corresponding directional vectors (e_n, e_v, e_φ).

Besides these “one-dimensional” criteria, we introduce additional two ones. The RMS deviation of the Mahalanobis distance in the “geometric” coordinates

is

$$X_{2d}(t_i) = \sqrt{(\hat{x}_G(t_i) - x_G(t_i))^T (J_G(t_i))^{-1} (\hat{x}_G(t_i) - x_G(t_i))},$$

and the RMS deviation of the Mahalanobis distance in the total state vector space is

$$X_{4d}(t_i) = \sqrt{(\hat{x}_i - x(t_i))^T (J(t_i))^{-1} (\hat{x}_i - x(t_i))}.$$

Here, $J_G(t_i)$ is the upper-left block (two first rows and columns) of the matrix $J(t_i)$, $\hat{x}_G(t_i)$ is the “geometric” coordinates estimate (two first elements of \hat{x}).

The total number of the criteria ($c_l, c_n, c_v, c_\varphi, c_{2d}, c_{4d}$) is $n_c = 6$. In the case of iterating the criteria, they will be indexed as $c_i, i \in \overline{1, n_c}$.

2.3 Confidence Interval of Criterion

In this section, we consider the longitudinal channel (deviation X_l , criterion c_l) as an example. Calculations for other channels (criteria $c_n, c_v, c_\varphi, c_{2d}, c_{4d}$) can be made in the same way.

The “true” criterion value c_l is the expectation

$$c_l = \sqrt{\mathbf{E} \{X_l^2(t)\}},$$

with averaging over the distribution of random measurement errors, measurement instants, and probable trajectories.

For practical usage, we replace the expectation by its plug-in estimator \hat{c}_l (see [11]). It is the empirical sample average

$$\hat{c}_{l,n}^2 = \frac{1}{\sum_{k=1}^N N_k} \sum_{k=1}^N \sum_{j=1}^{N_k} X_l^2(t_j, x_k(\cdot)) = \frac{1}{n} \sum_{i=1}^n X_{l,i}^2, \quad \hat{c}_{l,n} = \sqrt{\hat{c}_{l,n}^2}. \quad (2)$$

Here, N is the number of the trajectories $x_k(\cdot)$ that have been processed up to the instant of \hat{c}_l calculation, N_k is the number of the measurements of $x_k(\cdot)$, $n = \sum_{k=1}^N N_k$ is the total number of the measurements. The X_l values are indexed in two ways: firstly, by the trajectory and the time instant as $X_l(t_j, x_k(\cdot))$, secondly, by the total index i over all the trajectories as $X_{l,i}$. The $\hat{c}_{l,n}$ satisfies (see [11])

$$\hat{c}_{l,n} \xrightarrow[n \rightarrow \infty]{\mathbf{P}} c_l, \quad \mathbf{E} \{\hat{c}_{l,n}\} = c_l.$$

Since $\hat{c}_{l,n} \neq c_l$, we need a confidence interval for $\hat{c}_{l,n}$ for precise calculations. An interval $[\hat{c}_{l,n}^l, \hat{c}_{l,n}^u] \ni \hat{c}_{l,n}$ is a $1 - \alpha$ confidence interval if

$$\liminf_{n \rightarrow \infty} \mathbf{P} \{c_l \in [\hat{c}_{l,n}^l, \hat{c}_{l,n}^u]\} \geq 1 - \alpha.$$

In our work, we use the normal-based confidence interval [11] in the form

$$[\hat{c}_{l,n}^l, \hat{c}_{l,n}^u] = [\hat{c}_{l,n} - \hat{\mathbf{s}}\mathbf{e}_n(\hat{c}_{l,n})z_{\alpha/2}, \hat{c}_{l,n} + \hat{\mathbf{s}}\mathbf{e}_n(\hat{c}_{l,n})z_{\alpha/2}],$$

where $z_{\alpha/2}$ is the $(1 - (1 - \alpha)/2)$ -quantile of the standard normal distribution $\mathcal{N}(0, 1)$, and $\hat{\text{se}}_n(\hat{c}_{l,n})$ is the estimated standard error of $\hat{c}_{l,n}$, which can be calculated as follows:

$$\hat{m}_{l,n}^{(4)} = \frac{1}{n} \sum_{i=1}^n X_{l,i}^4, \quad \hat{\text{se}}_n(\hat{c}_{l,n}) = \frac{1}{2\hat{c}_{l,n}\sqrt{n}} \sqrt{\hat{m}_{l,n}^{(4)} - (\hat{c}_{l,n}^2)^2}.$$

2.4 Optimization Problem

Denote the parameter vector by a . In our case, there are 16 parameters of the IMM implementation by NITA; therefore, $a \in \mathbb{R}^{16}$. All the parameters are subject of box constraints: $a_{\min} \leq a \leq a_{\max}$. The multicriteria optimization problem can be formulated as follows:

$$\begin{cases} (c_l(a), c_n(a), c_v(a), c_\varphi(a), c_{2d}(a), c_{4d}(a)) \rightarrow \min, \\ a_{\min} \leq a \leq a_{\max}. \end{cases} \quad (3)$$

3 Genetic Program

The genetic optimization algorithm is based on the evolution process simulation. In these terms, there is a population $P = \{a\}$ consisting of individuals. Every individual a is connected with some point in the space of parameters to be optimized (of the trajectory tracker in our case).

In Figure 2, the principal scheme of our genetic optimization program is shown. The scheme is standard enough [8] and only slightly differs from the one in our earlier work [2]. The main differences are that the new program is intended for big data sets processing and that the criteria contain uncertainties (the confidence intervals).

The ‘‘true’’ criterion values in (3) cannot be calculated; therefore, the estimates $\hat{c}_{i,n(a)}(a)$ are used instead:

$$\begin{cases} (\hat{c}_{l,n(a)}(a), \hat{c}_{n,n(a)}(a), \hat{c}_{v,n(a)}(a), \hat{c}_{\varphi,n(a)}(a), \hat{c}_{2d,n(a)}(a), \hat{c}_{4d,n(a)}(a)) \rightarrow \min, \\ a_{\min} \leq a \leq a_{\max}. \end{cases}$$

Note that, here, the number of samples n for the estimates $\hat{c}_{i,n(a)}(a)$ depends on a : $n = n(a)$. This reflects the fact that the estimates \hat{c}_i of the criteria are calculated using different subsets of the whole training trajectory set $\mathcal{X} = \{x_k(\cdot)\}_{k=1}^N$.

In the program, we refuse the idea to calculate the estimates \hat{c}_i for a newly formed individual a using the whole set \mathcal{X} . The set \mathcal{X} is expected to be very big; therefore, if we did so, the computational load would be too large. Instead of that, at every generation b (this is the epoch of the evolution process), we calculate the ‘‘preliminary’’ estimates $\hat{c}_{i,N_{\text{batch}}}(a)$ on a small batch $\mathcal{X}_b = \{x_k(\cdot)\}_{k=N_{\text{batch}}b+1}^{N_{\text{batch}}(b+1)}$ and then upgrade them at the succeeding generations using formulas (2).

The estimates $\hat{c}_{i,n(a)}(a)$ of the criteria are influenced by random errors in data; therefore, they have uncertainties, which are especially big if the number $n(a)$ is small. For this reason, the lower and upper confidence bounds $\hat{c}_{i,n(a)}^l(a)$,

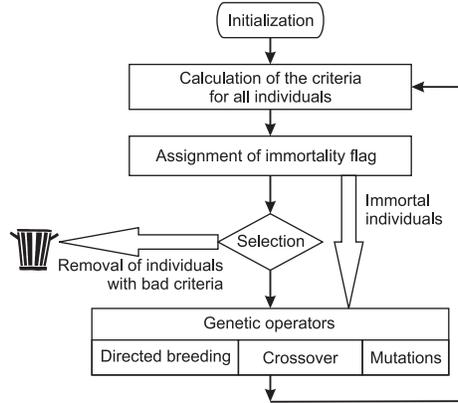


Fig. 2. Genetic algorithm flowchart.

$\hat{c}_{i,n(a)}^u(a)$ are used in the selection procedure in order to prevent deletion of the individuals that might have better values of \hat{c}_i in the future. Other operations of the genetic algorithm (the directed breeding, the crossover, and the mutation) do not differ significantly from those in [2].

3.1 Selection

The main difference of the new genetic program from the previous one is in the selection procedure. At the first step, immortal individuals $P_d = \{a_d\}$ are assigned. An individual a_d is assigned to be immortal if it is the minimum point of some criterion or its upper confidence bound

$$\exists i \in \overline{1, n_c} : \quad a_d = \operatorname{argmin}_{a \in P} \hat{c}_{i,n}(a) \quad \text{or} \quad a_d = \operatorname{argmin}_{a \in P} \hat{c}_{i,n}^u(a).$$

Only non-immortal individuals of age greater than 3 participate in the second step of the selection. An individual a_s is removed from the population if its lower confidence bound of some criterion is greater than the population minimum of upper confidence bound of this criterion

$$\exists i \in \overline{1, n_c} : \quad \hat{c}_{i,n}^l(a_s) > \min_{a \in P} \hat{c}_{i,n}^u(a). \tag{4}$$

In some cases (at the initial steps), condition (4) is not checked for all the criteria, but only for the criterion c_{4d} .

4 Simulation Results

There are two sets of simulated trajectories. The first one is the training set for optimization, the second one is the test set for validation of the optimization

of Sciences. Using 16 CPU cores, the total computation time were 58 hours. The computations was stopped at the 2320th generation of the evolution.

The test set consists of 400 trajectories which are created independently in the same way as the training ones (but without random variations of measurement instants). In Table 1, the criteria values on the test trajectories are shown.

Table 1. The values of the criteria on the test set.

	$\hat{c}_{l,n}$	$\hat{c}_{n,n}$	$\hat{c}_{v,n}$	$\hat{c}_{\varphi,n}$	$\hat{c}_{2d,n}$	$\hat{c}_{4d,n}$
Initial parameters	2.86341	3.71362	3.2436	3.32135	5.11065	8.78238
Optimized parameters	0.90609	0.94095	1.15279	1.18451	1.33594	2.64482

An example of the trajectory tracker output in the plane x_N, x_E for one test trajectory is shown in Fig. 4. The blue graph corresponds to the initial parameters, the green one corresponds to the optimized parameters at the generation 2320 (the best of the c_{4d} criterion), the black dots are the measurements, and the true (ideal) trajectory is the magenta solid line. The estimated track for the optimized parameters is evidently closer to the true trajectory than that for the initial parameters. In Figure 5, the graphs of the RMS deviation of X_{4d} is shown

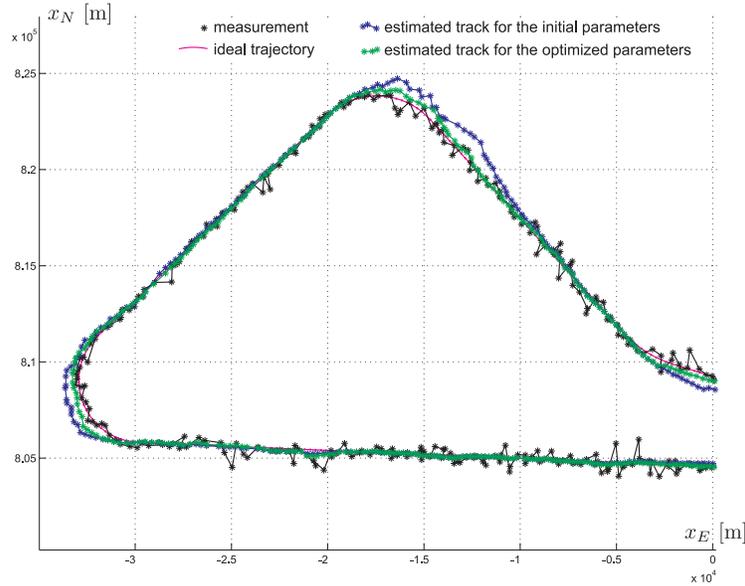


Fig. 4. A fragment of some test trajectory in the plane x_N, x_E .

as a function of time along the trajectory shown in Fig. 4. The averaging of X_{4d} is performed over 100 realizations of the random errors with the same grid of measurement instants. The blue solid line is for the initial parameters of the tracker and the red one is for the optimized parameters at the generation 2320. The dashed lines are the 0.95-confidence bands. The subfigure below the main figure shows the controls $u(t)$ (green) and $w(t)$ (blue). Using this subfigure, one can see that all the places where the behavior of $X_{4d}(t)$ changes are connected with the switches of the controls.

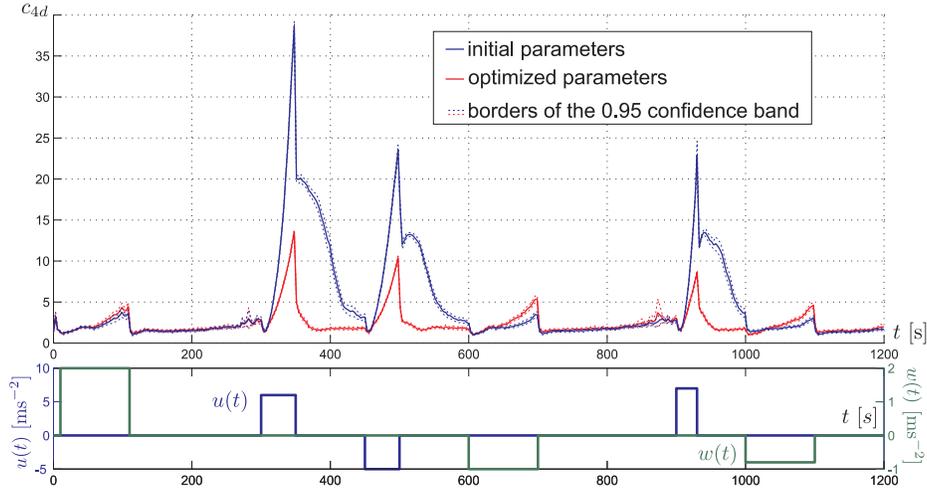


Fig. 5. Graphs of the RMS deviation of X_{4d} along some test trajectory as a function of time.

In Figure 6, the analogous RMS error graph for the absolute (in meters, without division by the value of the CRLB) lateral deviations $x_n(t)$ are shown (the trajectory is the same). As in the case of Fig. 5, the initial parameters correspond to the blue solid line and the optimized ones correspond to the red line. The dashed lines show the 0.95-confidence bands. The black line is for the RMS error of the measurements.

5 Conclusion

As a result of computer simulations, we can say that the trajectory tracking program with the optimized parameters shows a smaller RMS error (both absolute and relative) in each channel and has a faster transition process.

In further work on this project, we plane to use real ATC data for the training and test trajectory sets. Another point important for our study is the selection procedure, since its current version claims too high requirements on individuals.

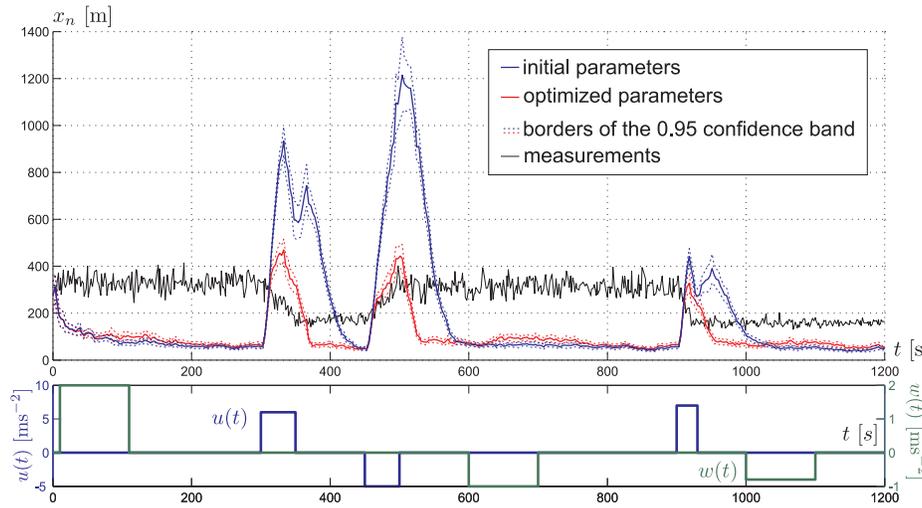


Fig. 6. The absolute (in meters) RMS deviation in the lateral channel.

References

1. Bar-Shalom, Y., Blair, W.D.: Multitarget-multisensor tracking: Applications and advances. Vol. III. Artech House, Norwood, MA (2000)
2. Bedin, D., Ivanov, A.: The use of a genetic algorithm for parameter adjustment of the multi-hypothesis aircraft tracking algorithm. In: 2019 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), pp. 1–4. Saint Petersburg, Russia (2019)
3. Bedin, D.A., Patsko, V.S., Fedotov, A.A., Belyakov, A.V., Stokov, K.V.: Restoration of aircraft trajectory from inaccurate measurements. *Automation and Remote Control* **71**(2), 185–197 (2010)
4. Doucet, A., Ristic, B.: Recursive state estimation for multiple switching models with unknown transition probabilities. *IEEE Transactions on Aerospace and Electronic Systems* **38**(3), 1098–1104 (2002)
5. Hastie, T., Tibshirani, R., Friedman, J.: *The elements of statistical learning: Data mining, Inference, and Prediction*. Springer, New York (2013)
6. Lehmann, E.L., Casella, G.: *Theory of point estimation*. 2nd edn. Springer, New York (1998)
7. Li, X.R., Jilkov, V.P.: Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems* **41** (4), 1255–1321 (2005)
8. Michalewicz, Z.: *Genetic algorithms + Data structures = Evolution programs*. Springer-Verlag, Berlin, Heidelberg (1996)
9. Šimandl, M., Královec, J., Tichavský, P.: Filtering, predictive, and smoothing Cramér–Rao bounds for discrete-time nonlinear dynamic systems. *Automatica* **37** (11), 1703–1716 (2001)
10. Vahabian, A., Sedigh, A.K., Akhbardeh, A.: Optimal design of the variable structure IMM tracking filters using genetic algorithms. In: *Proceedings of the 2004*

- IEEE International Conference on Control Applications, pp. 1485–1490. Taipei, Taiwan (2004)
11. Wasserman, L.: All of nonparametric statistics. Springer, New York (2006)
 12. EUROCONTROL Specification for ATM Surveillance System Performance, Std. <https://www.eurocontrol.int/publication/eurocontrol-specification-atm-surveillance-system-performance-esassp>. Last accessed 10 Sep 2020
 13. EUROCONTROL Standard for Radar Surveillance in En-Route Airspace and Major Terminal Areas, Std. <https://www.eurocontrol.int/publication/eurocontrol-standard-radar-surveillance-en-route-airspace-and-major-terminal-areas>. Last accessed 10 Sep 2020