EPiC Series in Computing

Volume 104, 2024, Pages 36–50

Proceedings of 3rd International Workshop on
Mathematical Modeling and Scientific Computing

Optimizing Trajectory Tracking Algorithms by Evolutionary Procedure on Real Data with Simulated Measurements

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, Yekaterinburg, Russia
bedin@imm.uran.ru, iagsoft@imm.uran.ru

Abstract

The problem of optimizing trajectory tracking algorithms is considered. Based on measurements of a moving object, such algorithms iteratively make estimates of its state. These algorithms contain parameters that affect the quality of their work, for example, the noise variances in a mathematical model of the object's dynamics. A multicriteria evolutionary optimization algorithm for such parameters is proposed based on genetic procedures. We also elaborate a procedure for using this algorithm on real data in which random measurement errors are simulated along the real trajectory. The system of criteria is proposed that assesses both the total mean square deviation of the trajectory tracking algorithm's output and the quality of its transition processes after a change of the object's motion mode. The algorithm was tested on model and real air traffic data.

1 Introduction

This study focuses on optimizing trajectory tracking algorithms. The problem statement and the associated formulas overlap with the authors' previous works [4, 3], and in some cases, formulas are repeated for consistency in notation.

Trajectory tracking algorithms estimate the position of a moving object, as well as its velocity vector and possibly other motion parameters. These estimates are based on the history of object position measurements up to the current time instant. A classical example of such an algorithm is the Kalman Filter (KF) [1]. One of the most widely used trajectory tracking algorithms for an aircraft as a moving object is the Interacting Multiple Model (IMM) algorithm [21]. In the rest of the text, we will consider specifically this case of a moving object.

Let us assume that the aircraft's position is measured at discrete time instants t_i. Various types of measurement systems exist. Radars are typical for aircraft [1]. The relationship between the aircraft position and the measurement $y(t_i)$ can be described as follows:

$$y(t_i) = x_G(t_i) + w_{i,k(t_i)}. \tag{1}$$

V.L. Turova, A.E. Kovtanyuk and J. Zimmer (eds.), MMSC 2024 (EPiC Series in Computing, vol. 104), pp. 36–50

Here, x_G is the geometric part of the state vector, it represents the aircraft position and consists of the north and east coordinates: $x_G = [x_N\ x_E]^\mathsf{T} \in \mathbf{R}^2$; $k(t_i)$ is the identifier, which a measurement system (radar) active at time t_i. The random measurement errors $w_{i,k}$ are assumed to be centered $\mathbf{E}\{w_{i,k}\} = 0$, with a covariance matrix $W_k(t_i) = \mathbf{E}\left\{w_{i,k} w_{i,k}^T\right\}$, which depends on the time t_i. Generally, the variance of random errors for radar measurements depends on the relative positioning of the aircraft and the radar. However, this dependence is relatively weak [1], and we can reasonably assume that $W_k(x_G(t_i)) \approx W_k(\chi)$ for any possible $\chi \in \mathbf{R}^2$ near $x_G(t_i)$. Given the fact, that $y(t_i)$ is also near $x_G(t_i)$, one can compute matrices using $y(t_i)$ instead of $x_G(t_i)$ for any t_i. Therefore, we can assume that the covariance matrices $W_k(t_i)$ are known in advance and depend only on the measurement number: $V_i \coloneqq W_{k(t_i)}(y(t_i))$.

The trajectory tracking algorithm processes a set of measurements $\{y_i\}_{i=1}^n$ available up to the last measurement time t_n, to produce an estimate $\hat{x}_{out}(t_n)$ of part $x_{out}(t_n)$ of the motion parameters of the aircraft, that someone is interested in [1, 12]. For example, there can be position x_G, velocity vector $v = \dot{x}_G$ and other motion parameters as acceleration, turn rate and so on: $x_{out} = \left[x_G{}^\mathsf{T}\ v^\mathsf{T}\ \ldots\right]^\mathsf{T}$. For simplicity, we consider only the case $x_{out} = x_G$.

Typically, each algorithm has parameters that influence its performance. For example, the simplest tracking algorithm, the so-called alpha-beta filter [1, 14], has two parameters: α, β, each with a simple interpretation. The filter follows these equations:

$$\begin{aligned} \hat{x}_G(t_i) &= \bar{x}_G(t_i) + \alpha\left(y(t_i) - \bar{x}_G(t_i)\right), \\ \hat{v}(t_i) &= \bar{v}(t_i) + \beta\left(y(t_i) - \bar{x}_G(t_i)\right)/dt_i. \end{aligned} \tag{2}$$

Here, $\bar{x}_G(t_i), \bar{v}(t_i)$ are the predicted values of position $x_G(t_i)$ and velocity $v(t_i) = \dot{x}_G(t_i)$, respectively, and $dt_i = t_i - t_{i-1}$. Predictions are made using the Constant Velocity (CV) model [20]:

$$\begin{aligned} \bar{x}_G(t_i) &= \hat{x}_G(t_{i-1}) + \hat{v}(t_{i-1})dt_i, \\ \bar{v}(t_i) &= \hat{v}(t_{i-1}). \end{aligned} \tag{3}$$

The form of equations (2) makes interpretation of α and β clear: α is the coefficient of position update and β is the velocity update coefficient. Depending on the values of α and β, the algorithm exhibits different behaviors. Setting $\alpha = 1$ results in a trivial estimate where the last measurement is taken as the output $\hat{x}_G(t_i) = y(t_i)$. Conversely, when $\alpha = 0$, the estimate coincides with the prediction $\bar{x}_G(t_i)$. Thus, values of α close to 1 correspond to strong coalescence with measurements, while values close to 0 correspond to over-reliance on previous predictions. A similar interpretation can be done for the connection between the coefficient β and the velocity vector.

Fig. 1 illustrates examples of algorithm behavior for different α values (and fixed β for simplicity). Here, α are chosen from set $\{0.13, 0.6, 0.8, 0.9\}$ and $\beta = 0.01$. The filter runs here on real aircraft measurements. The aircraft moves from left to right, black dashed line shows "almost true" aircraft's trajectory: the data from precise measurement system (namely, GPS), circles are positions on the true trajectory at the radar measurement instants (they are made by interpolation). The solid black line depicts the radar measurements. The zigzag pattern is conditioned by the radar random errors (see equation (1)). The differences in the algorithm's behavior for different α are evident. The value of $\alpha = 0.13$ smooths the data effectively, and closely approximates the true aircraft trajectory while the aircraft moves straight. However, the transition to circular motion causes a "catastrophic" event: the algorithm loses the trajectory and only comes close to it after a long period of time. Algorithms with $\alpha = 0.6, 0.8, 0.9$ (red,

green and magenta lines, respectively) are very close to radar measurements throughout the entire motion, without losing tracking the true trajectory at the start of the circular motion segment. This clearly shows that as the value of α increases, the estimated trajectory becomes closer to the measurements, but further away from the true trajectory due to higher variance. On the other hand, smaller values of α result in a smoother estimated trajectory with a closer distance to the true one. However, these smaller values also cause a poorer response to changes in motion mode (from a straight-line to a circular in this example).

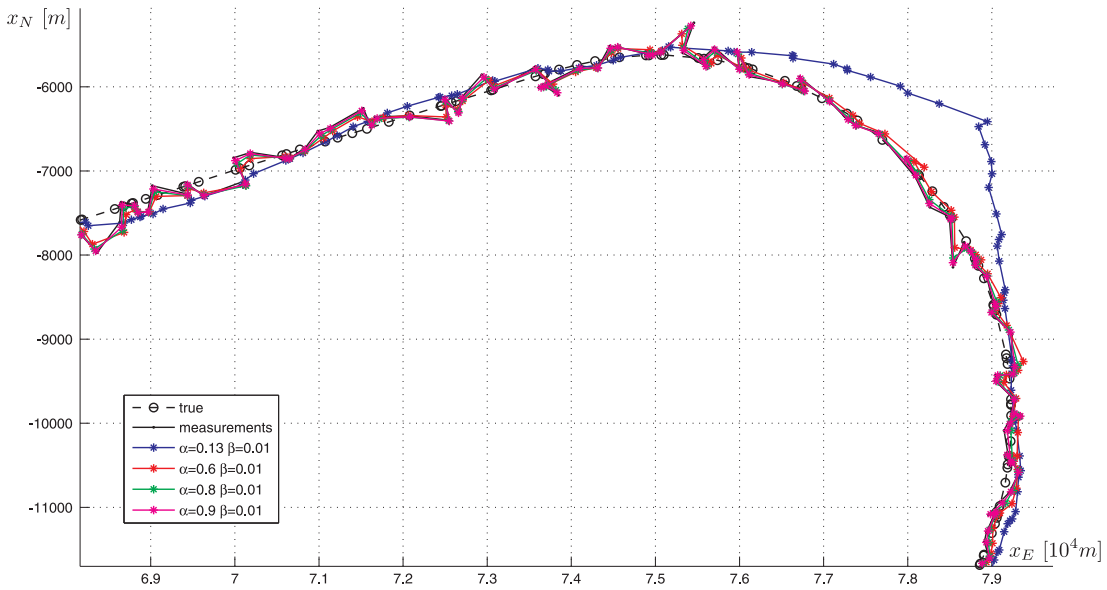

Figure 1: Trajectory tracking using the alpha-beta filter with different values of α.

We see that the selection of α and β values influences tracking performance, and there are at least two meanings of the word "performance": the first can be called "track smoothness" (the output track of estimates has less variance than the measurements) and the second is connected to the ability to overcome sudden changes in the aircraft's motion mode.

This problems of dependence on parameters and two meanings of tracking quality arise for other algorithms too. In [2], authors presented simulations, which show, that the model lying behind KF and its parameters influence on mean square error (MSE) in position and velocity estimations. Additionally it is showed there that IMM algorithm usually outperform KF and its MSE also has dependence on algorithm's structure (which motion models are used in it). The simulations in [2] clearly shows that the MSE value usually has a peak after a maneuver's onset, and the value of this peak can depend on the structure or the parameters in another way as the MSE level during stationary motion time period. Analogous simulations and conclusions exist in [12, 22]. In standards of EUROCONTROL (the central organization in Europe for coordination and planning of air traffic control), such as in [10], performance criteria are not formulated only in terms of total MSE along whole trajectory, but also in the transitional properties of the MSE as a function of time.

There are a lot of publications devoted to parameter adjustment for tracking algorithms. For example, choosing of IMM algorithm parameters was considered in [1, 6, 22, 17, 24, 8]. In one part of them, adaptive versions of IMM algorithm is described [24, 8] where parameters

are changed during the tracking process. In the other part, the parameters are chosen before tracking process by a procedure, which optimizes some cost function [6, 17]. There are examples of genetic or evolutionary algorithms use in optimization [15, 23]. The parameter tuning of basic KF algorithm is also wide presented in literature [6, 25, 5]. For the alpha-beta filter, parameter adjustment problem can be partial solved analytically [11, 14] (only in the case of equal dt_i and $W_i = \sigma^2 I = \text{const}$).

Previously, our research team developed an evolution procedure for optimizing trajectory tracking algorithms in non-adaptive way as in [6, 17, 15, 23]. And this evolution procedure was implemented to the IMM tracker with particular structure [4, 3]. The optimization methodology for this procedure is based on the use of artificial data. The work with real data seemed us to be very complicated due to presence of outliers in the measurement, systematic errors of radars. In addition, modeling makes it easier to generate large amounts of data and, for artificial measurements, there is no problem with what the true trajectory is, to compare estimation results. Unfortunately, when we tried to apply the optimization results in a tracking system with real-world aircraft traffic, we found that the tracking quality only increased slightly or not at all. Also this runs on real data showed some problems with transitions between motion modes like that the alpha-beta filter with $\alpha = 0.13$ demonstrates in Fig. 1.

To overcome these problems, we decided to develop a methodology for optimizing trajectory tracking algorithms based on real data, as well as to introduce criteria that directly penalize poor transitions between motion modes. In this paper, we present our new methodology and the new criteria. We then show the work of the updated evolutionary procedure using the simplest alpha-beta filter as the optimization object. This was chosen because there are only two parameters in it, and all calculations can be done quickly.

2 Mean square criteria

The performance of trajectory tracking algorithms is typically evaluated using the mean square error (MSE) between the algorithm's estimates and the true positions at the same time instants on a trajectory. The MSE is generally assessed in different "channels." Traditionally, these channels include longitudinal deviation tng, lateral deviation ort, velocity magnitude vel, and track angle ang [10]. The particular level of such criterion at a certan point on a trajectory of an aircraft depends not only the aircraft's motion parameters, but also on the overall surveillance conditions, i.e. the variances of random radar errors, the frequency of measurement updates, the distance between the point to the radars and so on. In EUROCONTROL standard [10], a set of detailed scenarios is proposed, specifying all these conditions and defining reference accuracy levels for the algorithms. According to EUROCONTROL, algorithm designers must make the MSE lower than these accuracy levels, without necessarily optimizing beyond them. Subsequently, it is assumed that either simulations are carried out in accordance with these scenarios, or only those real trajectories that fully match the scenario conditions are considered. In [3], the authors discussed the problems with such accuracy levels and proposed, instead of predefined scenarios, using the Cramér–Rao lower bound [16] as the reference accuracy for arbitrary simulation conditions.

If the motion model is unknown or the trajectory of a real aircraft is considered, the Cramér–Rao bound cannot be used. However, algorithms can still be compared with each other. For example, the Kalman filter for the Constant Velocity (CV) model or another well-investigated algorithm with predictable behavior can be used as a reference algorithm. The authors propose that, when working with real data, empirical covariance matrices $\hat{V}_{et}(t)$ (and one-dimensional empirical root mean square deviations $\hat{\sigma}_{et,ch}$) be constructed for a selected reference algorithm

et using simulation for each measurement time.

In [3], it was proposed to compare the difference between the estimated $\hat{x}(t)$ and the true state vector $x(t)$ to the reference value (the the Cramér–Rao bound was used as a reference there), calculate the deviations in relative units, and use them as the basis for evaluation criteria. Now, we can use $\hat{V}_{et}(t)$ instead of it and calculate the relative deviations $\delta_{et,ch}(t)$ in a simiar as in [3] way:

$$\delta_{et,ch} = \frac{h_{ch}(\hat{x}) - h_{ch}(x)}{\hat{\sigma}_{et,ch}}, \quad \delta_{et,2d}(t) = (\hat{x}_G(t) - x_G(t))^{\mathsf{T}} \hat{V}^{-1}_{et,G}(t) \, (\hat{x}_G(t) - x_G(t)) \,. \tag{4}$$

Here, h_{ch} is the scalar function for "projecting" onto the one-dimensional channel ch (for example, it can be inner product to the specific vector); $2d$ refers to the deviation in the observed two-dimensional position on the plane; $\hat{V}_{et,G}(t)$ is a submatrix of the empirical covariance matrix $\hat{V}_{et}(t)$ for the reference algorithm, associated with the geometric position on the plane; $\hat{\sigma}_{et,ch}(t)$ is the reference MSE level for channel ch, derived from $\hat{V}_{et}(t)$:

$$\hat{\sigma}^2_{et,ch}(t) = \left(\frac{\partial h_{ch}}{\partial x}\right) \hat{V}_{et}(t) \left(\frac{\partial h_{ch}}{\partial x}\right)^{\mathsf{T}}.$$

The final formula for the mean square criteria, based on the reference algorithm, consists of the time-averaged values of $\delta_{cr,ch}(t)$ or $\delta_{et,ch}$ along a single trajectory $x(t)$, and then averaged over the set of all trajectories $X = \{x(\cdot)\}$ in the training set.

Currently, a scheme is used that simulates a subset of random measurement realizations $y_\omega(t)$ along a single true trajectory $x(\cdot)$. In this case, the averaging is also performed over random realizations:

$$J_{ch}(X, \Omega) = \frac{1}{n_\Omega \sum\limits_{x(\cdot)\in X} n_{x(\cdot)}} \sum_{x(\cdot)\in X} \sum_{t_i \in T_{x(\cdot)}} \sum_{\omega\in\Omega} \delta^2_{et,ch,x(\cdot),\omega}(t_i). \tag{5}$$

Here, Ω is the set of random realizations ω with a total count of n_Ω; $T_{x(\cdot)}$ are the measurement time instants for trajectory $x(\cdot)$, and $n_{x(\cdot)}$ is their number; $\delta_{et,ch,x(\cdot),\omega}(t_i)$ is the value of $\delta_{et,ch}(t_i)$ calculated using formula (4) for trajectory $x(\cdot)$ based on the random measurement realization ω (the algorithm's responses $\hat{x}(t_i)$ depend on the random realization of measurements).

3 Transitional criteria

It is known that at the boundaries of stationary motion segments, tracking algorithms have an error spike, associated with the time required by the algorithm (rather a certain number of measurements) "to understand" the end of the previous segment [2, 12]. As an example, we can again refer to Fig. 1. In some cases, the spike is indeed caused by the time needed to recognize the inconsistency between the current motion and the model from the previous segment. Such algorithms were discussed, for example, in [12, 19]. In other cases, such as for the IMM algorithm, time is required to adjust the model probability vector [21, 1].

Mean square criteria only partially describe the behavior of trajectory tracking algorithms. Real trajectories typically include long segments of stationary motion (mostly straight-line), and the mean square error reflects behavior mainly on these segments, proportional to their duration in the overall motion. Transition process segments are relatively short, so the algorithm's behavior on them has little impact on the mean square criteria. This is further compounded by the fact that, during transition processes, the MSE of the reference algorithm has higher values

than in stationary segments [2, 21, 22]. Thus, in formula (4), the denominator becomes large, and the same deviation (e.g., in meters) is assessed much lower during a transition process segment.

The term of transition processes can be formulated in many ways, but most often it can be associated with situations when the estimate loses the property to be centered relative to the true trajectory (as for alpha-beta filter with $\alpha = 0.13$ in Fig. 1). That is, at these time instants, the output state vector has a bias:

$$b(t) = \mathbf{E}\{\hat{x}_{out}(t)\} - x_{out}(t).$$

The authors decided to use the time integral of the bias as a criterion for the quality of transition processes. As with the MSE criteria, the bias is proposed to be divided into "channels," similar to those introduced in Section 2.

In EUROCONTROL standard [10], the criteria for evaluating the quality of the transition process are the peak MSE after the onset of a new motion segment and the period during which MSE decreases from its peak to a specified level above the required MSE for stationary segments. The authors believe that minimizing the integral of bias $b(t)$ along the trajectory minimizes the EUROCONTROL criterion. The idea is as follows: the informal formula states that $\text{MSE}(t) = b^2(t) + \text{Variance}(t)$ [16], and there is an observation that the variance of "good" trajectory algorithms does not vary rapidly over time, including transitions. Another observation is that, on stationary motion segments, the bias of "good" algorithms is sufficiently small. Therefore, the peak of MSE is largely determined by the peak of bias, which is small on stationary segments. Minimizing the peaks of bias can decrease the peaks of MSE, and the integral of the bias is a characteristic that takes into account the height and width of the peaks.

It should be noted that a small bias does not imply it is zero. For instance, the Kalman filter, designed for straight-line constant velocity (CV) motion, has unavoidable bias on circular motion segments (the coordinated turn (CT) model [20]). However, the magnitude of this bias is quasi-stationary and much smaller than the error spike caused by a mode change. More complex algorithms that take into account variability of the dynamics, such as the IMM with both CV and CT dynamics, will have an advantage in this case: on both stationary segments, CV and CT, the IMM will have near-zero bias [1, 2, 22].

For transition process criteria, the bias estimates $\hat{b}$ are used instead of b and they projected also onto individual channels. The bias estimate $\hat{b}$ in channel ch at time instant t_i is calculated as follows:

$$\hat{b}_{ch}(t_i) = h_{ch}(x(t_i)) - \frac{1}{n_\Omega}\sum_{\omega\in\Omega} h_{ch}(\hat{x}_\omega(t_i)), \qquad \hat{b}_{2d}(t_i) = x_G(t_i) - \frac{1}{n_\Omega}\sum_{\omega\in\Omega} \hat{x}_{G,\omega}(t_i). \tag{6}$$

Here, the formula $\frac{1}{n_\Omega}\sum_{\omega\in\Omega}$ denotes empirical averaging over random realizations. The estimate of bias in channel ch for the reference algorithm et is further in similar fashion denoted as $\hat{b}_{ch,et}$.

Unlike the criteria (5), comparison with the reference at every time instant along the trajectory, as in formula (4), does not have much meaning. The reason is that, for different algorithms, the peak bias $b(t)$ will occur at different time instants t, and the behavior of $b(t)$ as a function of t can vary significantly: some algorithms show a monotonic decrease, while others oscillate. On the other hand, absolute values (e.g., in meters) are unsuitable for the same reasons as for MSE criteria—due to the magnitude can significantly vary in accordance with the observation scenario. Taking both considerations into account, the authors decided to divide trajectories into relatively short stationary motion segments and, for each segment, calculate the ratio of the integrals of the absolute value of the bias, for two biases of estimates obtained using the

tested and reference algorithms:

$$r_{j,x(\cdot),ch} = \frac{\sum\limits_{t_i \in T_j(x(\cdot))} \left|\hat{b}_{ch}(t_i)\right| dt_{i+1}}{\sum\limits_{t_i \in T_j(x(\cdot))} \left|\hat{b}_{ch,et}(t_i)\right| dt_{i+1}}. \tag{7}$$

Here, $T_j(x(\cdot))$ represents consecutive time moments belonging to stationary segment j. For $2d$ channel, the formula should be modified slightly: modulus $|\cdot|$ should be replaced by Euclidean norm $\|\cdot\|$ for $\hat{b}_{2d}$ from (6).

For more accurate results when n_Ω is small, it is better to use different sets of random realizations Ω for a tested algorithm and the reference one (in the numerator and the denominator of (8)).

The overall criterion value is obtained by averaging over segments and trajectories:

$$I_{ch}(X,\Omega) = \frac{1}{\sum\limits_{x(\cdot)\in X} m_{ch,x(\cdot)}} \sum_{x(\cdot)\in X} \sum_{j \in M_{ch,x(\cdot)}} r_{j,x(\cdot),ch}. \tag{8}$$

Here, $M_{ch,x(\cdot)}$ is the set of segments in trajectory $x(\cdot)$ for channel ch, and $m_{ch,x(\cdot)}$ is the number of such segments.

For lateral deviation *ort* and track angle *ang* channels, the segments are determined as follows. For real data, data from Automatic Dependent Surveillance–Broadcast system (ADS-B) [9] is used. Trajectories with uninterrupted data of track angle φ are selected, and piecewise-linear approximation to these data is applied, with variation of the number of possible intervals of linear dependency pattern. The intervals' endpoints of the approximation procedure are considered as the intervals of constant value of lateral accleration u_{ort}. For piecewise-linear approximation, the authors used the package [13].

The package requires an initial approximation for stable work. This was obtained as follows. First, an estimate of lateral acceleration $\hat{u}_{ort}$ is calculated for the trajectory using the numerical derivative $\hat{\dot{\varphi}}$ of the track angle and the measured by ADS-B value of velocity magnitude v:

$$\hat{\dot{\varphi}}(t_i) = \frac{\varphi(t_{i+10}) - \varphi(t_i)}{t_{i+10} - t_i}, \qquad \hat{u}_{ort}(t_i) = v(t_i)\hat{\dot{\varphi}}(t_i),$$

(it should be noted that v and φ are measurements, and their accuracy is assumed to be good, with outliers removed by preliminary procedures). This estimate is then smoothed using a 20-second sliding window, and the resulting smoothed values are compared to a threshold of 0.1 m/s^2. This threshold is used to determine the presence or absence of straight-line motion in accordance with EUROCONTROL [10]. The time instants t_i where the threshold is exceeded are used as the initial approximation for the piecewise-linear approximation procedure.

A similar procedure is used for the longitudinal deviation *tng* and speed magnitude *vel* channels. The difference is that longitudinal acceleration $\hat{u}_{tng}$ is estimated as:

$$\hat{u}_{tng} = \frac{v(t_{i+10}) - v(t_i)}{t_{i+10} - t_i}.$$

The smoothing period and threshold values are the same as for $\hat{u}_{ort}$ (corresponding to [10]). For the $2d$ channel, the segments are formed by intersecting the segments from the lateral *ort* and longitudinal *tng* channels.

Fig. 2 shows the track angle φ (in degrees, °) for one of the real aircraft trajectories (a typical passenger jet flight). The blue dots represent the values from the corresponding ADS-B field. The magenta line shows the numerical derivative $\hat{\varphi}$ (in radians per second) of φ, and the orange line represents the smoothed $\hat{u}_{ort}$ (in m/s^2), computed using the given formula. The blue line indicates the points where the smoothed $\hat{u}_{ort}$ exceeded the threshold in either the positive or negative direction. The green line shows the piecewise-linear approximation of track angle φ, and the green points correspond to the interval boundaries.

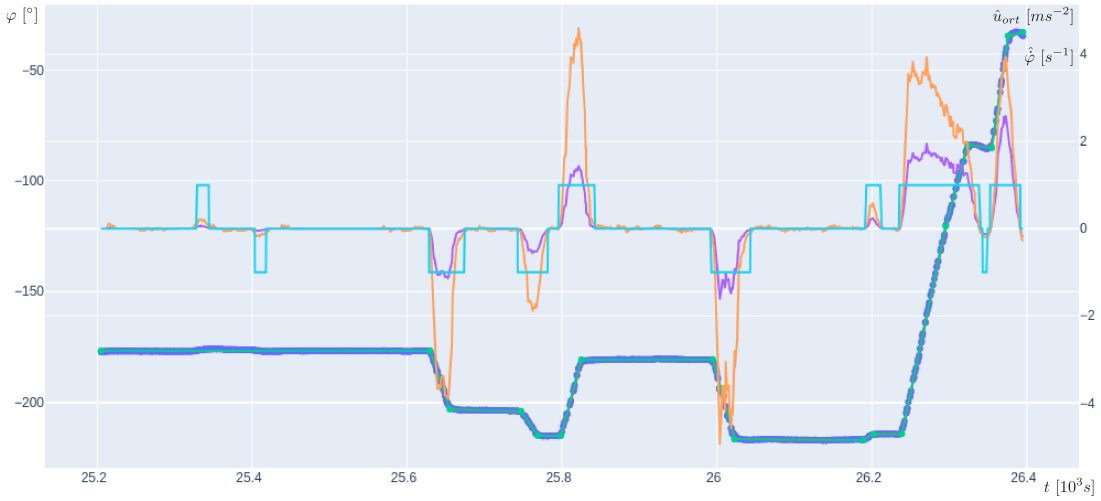

Figure 2: Example of segment finding in the lateral *ort* channel. Track angle φ measurements are shown as blue dots, and the green line represents the piecewise-linear approximation.

4 Evolutionary procedure

The algorithms being optimized have parameters. Let us call a fixed point a in the parameters' space as an "individual," as, for example, in [7]. Each "individual" a corresponds to an object in evolutionary procedure, for which all the criteria are calculated and evolutionary operators are applied.

The entire problem can be summarized as follows:

$$\begin{cases} (J_{ch_1}(a), I_{ch_1}(a), J_{ch_2}(a), I_{ch_2}(a), \ldots) \underset{a}{\to} \min, \\ a \in \mathcal{A}, \end{cases}$$

where ch_i is in the list of channels [*tng*, *ort*, *vel*, *ang*, *2d*], and $\mathcal{A}$ is a set of constraints to a values. Next, we make optimization only for the *2d* channel criteria, so the problem can be written shorter:

$$\begin{cases} (J_{2d}(a), I_{2d}(a)) \underset{a}{\to} \min, \\ a \in \mathcal{A}. \end{cases}$$

We elaborated the following scheme for optimizing trajectory tracking algorithms using real data and the evolutionary procedure, which we made before [4, 3].

1. Among real trajectories, those with long segments of ADS-B signal recordings are selected (the specification for ADS-B parameters can be found in [9]). From these segments, the longest one is selected with a maximum ADS-B data gap of 20 seconds. Additionally, it is required that the data includes not only position but also speed and track angle information. Confirmed high accuracy of ADS-B is also required: the PIC (Position Integrity Category) field must have a value greater than or equal to 7. This way the general set of trajectories $\mathcal{X}$ is created.

2. For the time instants t_i of the real radar measurements, the aircraft's motion parameters are fitted by linear interpolation from ADS-B measurements. Based on the known characteristics of the radar systems, covariance matrices W_i at t_i are constructed and recorded.

3. A simulation is performed in which $n_{\Omega'}$ random realizations of measurement errors are generated. For each realization, the estimated trajectory $\hat{x}_{et}(t_i)$ is calculated for the reference algorithm. The average track of the reference algorithm's estimates is computed, along with the reference MSE matrix $\hat{V}_{et}(t_i)$, for use in formula (4), and the bias esitmates $\hat{b}_{ch,et}$, for formula (7), are calculated.

4. The optimization process starts with an initial set of individuals $A_0 = \{a\}$. At each step l of the optimization, computations are performed with a subset X of the general set of trajectories $\mathcal{X}$. For each trajectory $x(\cdot) \in X$, and for each individual from the current set $a \in A_l$, a simulation is performed with n_Ω random realizations of measurement errors and measurement tracks $\{y_\omega(t_i)\}_{t_i \in T_{x(\cdot)}}$ are generated. For each of these tracks, the algorithm a processes the data, computes the corresponding estimate track $\{\hat{x}(t_i)\}_{t_i \in T_{x(\cdot)}}$, and calculates the criteria (5), (8).

5. The entire population undergoes non-dominated sorting [7], after which individuals with clearly non-optimal criteria values are identified. In dependence on age of an individual, the individual is identified as "bad" if it has rank after the non-dominated sorting is greater than 3 (for an usual "not young" individual) or 1 (for an individual, for which all the trajectories in the dataset are used in calculations, so called an "elder" individual). These individuals are discarded and do not participate in further calculations (but for the "young" individuals the discarding is postponed). The details of the sorting are in the authors' previous works [3, 4].

6. Evolutionary algorithm steps are performed, which are described in details in the authors' previous works [3, 4].

Steps 4–5 are repeated until the Pareto front stabilizes, which means, that all the individuals in the Pareto front are the same during a some preconditioned number of steps.

5 Work example

As an example how the evolutionary procedure work, we present the results of the optimiziation the alpha-beta filter, which is maybe the simplest trajectory tracking algorithm. There are complex versions of the alpha-beta filter exist [1], which are essentially simplified (and precomputed offline) versions of the Kalman filter, designed for certain constant dynamics. However, we consider the simplest version of the algorithm [14] with the formulas provided above: (2), (3).

This filter contains only two parameters, α and β, with a simple constraint:

$$\alpha, \beta \in [0, 1].$$

The downside of such a simple algorithm is that it does not account for measurement covariances V_i, and therefore, it cannot have equal effectiveness across different observation scenarios. Another known drawback is its poor handling of maneuvers [1], such as coordinated turns (CT) and accelerations [20]. Nevertheless, this simple example is useful for illustrating the work of the multi-criteria procedure and visually demonstrating the dependence of the criteria on the parameters.

Unfortunately, direct implementation of formula (2) leads to unstable work for some parameter subset (for β near 1 and α small). In several simulations, the filter estimate $\hat{x}$ "runs away" suddenly to infinity. It was connected to the dt_i in denominator in formula (2) for $\hat{v}$ update. When dt_i and dt_{i+1} for consecutive steps are very different, there can be a "jump" in the velocity estimate. To overcome this difficulty, we modified formula (2) slightly in the part of $\hat{v}$ update:

$$\hat{v}(t_i) = \bar{v}(t_i) + \mathrm{Sat}_{C_v}\Big(\beta \left(y(t_i) - \bar{x}_G(t_i)\right) / dt_i\Big), \tag{9}$$

where the saturation function defined as

$$\mathrm{Sat}_C(x) = \begin{cases} x, & \|x\| \le C, \\ \frac{x}{\|x\|} C, & \|x\| > C, \end{cases}$$

and C_v is the upper bound for velocity update. We used $C_v = 100$ m/s.

There are many publications that present analytical derivations for choosing the parameters of the alpha-beta filter. However, we found that in these analyses, only the case $dt_i = dt = \mathrm{const}$ and $W_i = \sigma_w^2 I = \mathrm{const}$ is considered. The most cited paper on this topic is [14], in which the steady-state filter conditions are investigated, they yield the formula for coefficients:

$$\Lambda^2 = \frac{\beta^2}{1-\alpha}, \qquad \beta = 2(2-\alpha) - 4\sqrt{1-\alpha}. \tag{10}$$

Here, Λ is the so-called tracking index. The formula of it is as follows:

$$\Lambda = \frac{\sigma_a (dt)^2}{\sigma_w}, \tag{11}$$

where σ_a is the intensity of the dynamic Gaussian white noise in acceleration, σ_w is the standard deviation of the measurement noise.

The starting set of individuals A_0 consisted of a single individual $a_0 = (\alpha = 1, \beta = 1)$. The optimization dataset included 156 real civil jet trajectories provided by NITA, LLC. NITA also provided its trajectory tracking algorithm, which we treated as a black box, and used as the reference algorithm in our evolutionary procedure, step 3. The number of random realizations for the reference algorithm in formulas (4), (7) was equal to 1000.

Optimization was performed based on two criteria: MSE in geometric 2-d position J_{2d} and the integral of the absolute mean deviation in geometric 2-d position I_{2d}. The evolutionary algorithm computations were performed on the Uran supercomputer at the IMM UB RAS [18], using 200 workers, with a computation time of 104 hours. A total of 800 generations were processed. Each generation worked with one real data trajectory, and 10000 random measurement realizations were generated. The population limit was 2000 individuals.

In Fig. 3, the best individuals from "elders" (in the sense of step 5 of the evolutionary algorithm) are shown on the plane of criteria J_{2d} and I_{2d}. The final Pareto front, calculated after generation 800, is displayed in blue. The points corresponding to earlier Pareto fronts from generations 170 and 350 are shown in magenta and red, respectively. The green marker corresponds to the "zero individual" (starting point) a_0 with the parameters $\alpha = 1$, $\beta = 1$. The evolution of the Pareto front was very rapid: the best individuals from 170 generations and 350 lie near the final front from 800 generation.

In Fig. 4, the Pareto optimal set of "elders" is shown in the plane of parameters α, β (blue crosses). The green line depicts the theoretical relationship between β and α by second equation in (10). The Pareto set not coincide with the theoretic manifold, but the part with $\beta > 0.5$ seems to be close to it. The reason why they do not coincide is due to the fact that this theoretical relationship was developed for the case where dt_i and W_i are constant, while in real data, dt_i varies as well as W_i. Also, obtaining the relations (10), (11) in [11, 14] did not imply any direct optimization of MSE or other criterion.

Figures 5 and 6 show the dependence of the criteria J_{2d} and I_{2d} on the parameters α (Fig. 5) and β (Fig. 6) for the "elders" comprising the Pareto front.

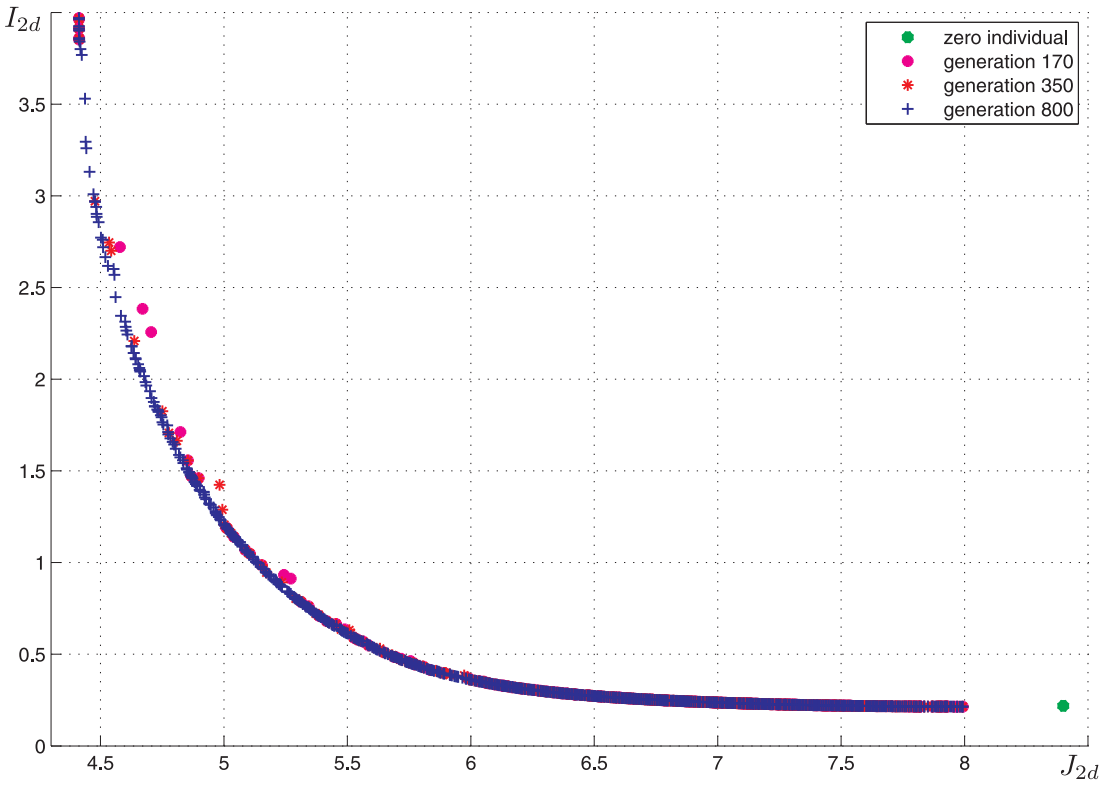

Figure 3: The Pareto front on the plane of criteria J_{2d} and I_{2d} as a result of optimization. Generations 170 (magenta), 350 (red), and 800 (blue, the final generation) are shown. The green point is the initial individual.

In Fig. 7, the work of the algorithms is demonstrated on an artificial test trajectory. A magnified fragment of the trajectory is shown. The true trajectory (ADS-B measurements) is

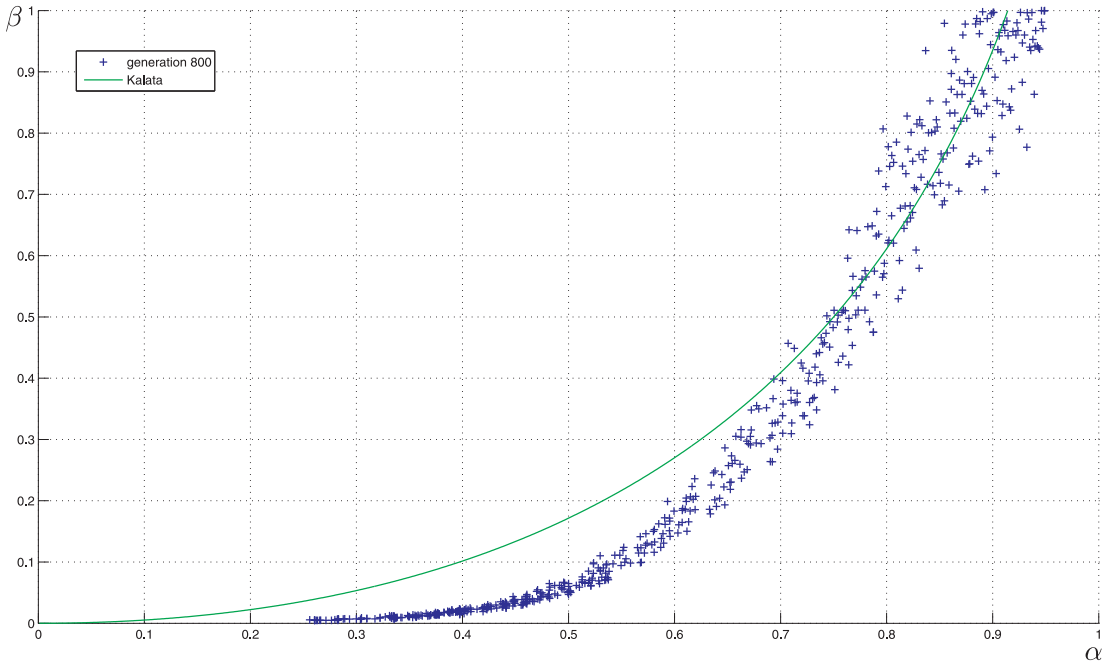

Figure 4: The Pareto optimal set of individuals (blue crosses) in the plane of parameters α and β. The green line shows the theoretical relationship between β and α, as described by second of equations (10).

indicated by a black dashed line with circles at the measurement times. The measurements are represented by a solid black line. Two "individuals" from the Pareto optimal set are shown: $(\alpha = 0.2605, \beta = 0.004782)$ (red line), $(\alpha = 0.4907, \beta = 0.07217)$ (green line), and one parameter pair, which was previously presented in Fig. 1 $(\alpha = 0.6, \beta = 0.01)$ (blue line). These two Pareto optimal parameters produce smoother estimates than the non-optimized case. The red line has the smoothest trajectory, but it deviates significantly from the measurements in the circular motion segment and has significant bias. The blue line has the largest variance. The green line balances variance and bias well.

6 Conclusions

The new methodics for optimization of trajectory tracking algorithms with use of real aircraft trajectories and simulated measurements was implemented to the simplest tracking algorithm, alpha-beta filter. The Pareto front and the Pareto optimal set for the algorithm's parameters were found.

The authors thank LLC "NITA" for providing real navigation system data and for their collaboration in designing the tracking algorithms.

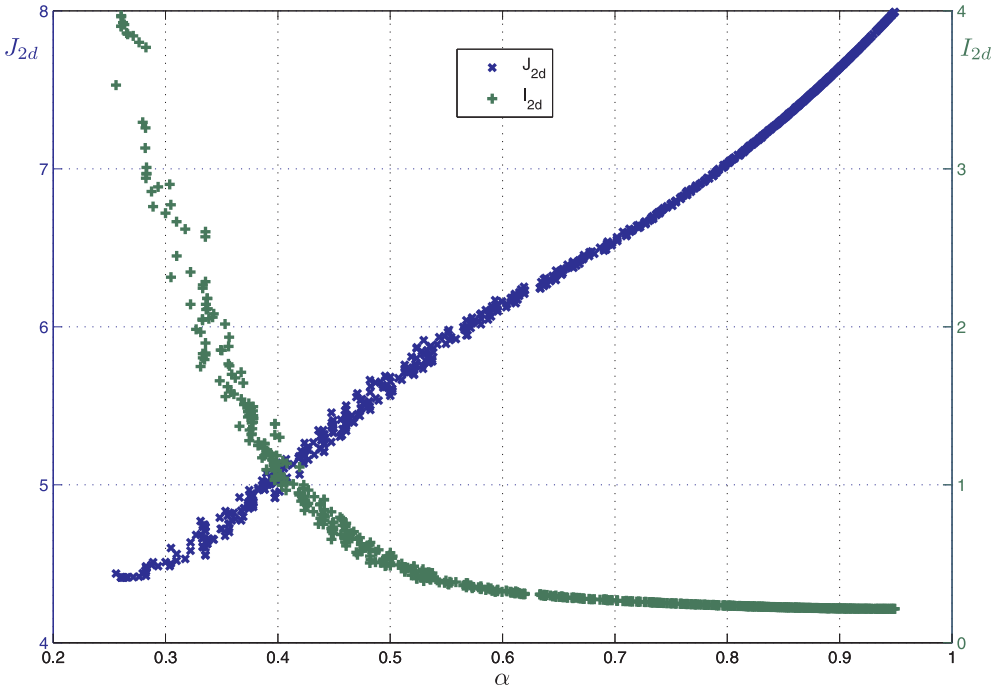

Figure 5: Dependence of the criteria J_{2d} and I_{2d} on parameter α.

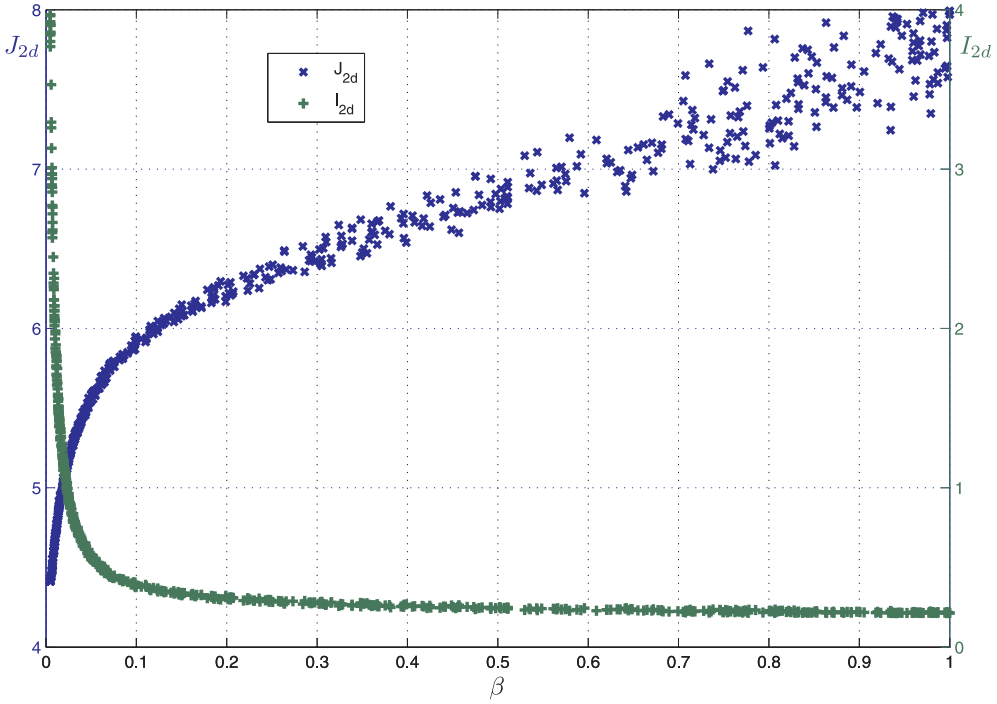

Figure 6: Dependence of the criteria J_{2d} and I_{2d} on parameter β.

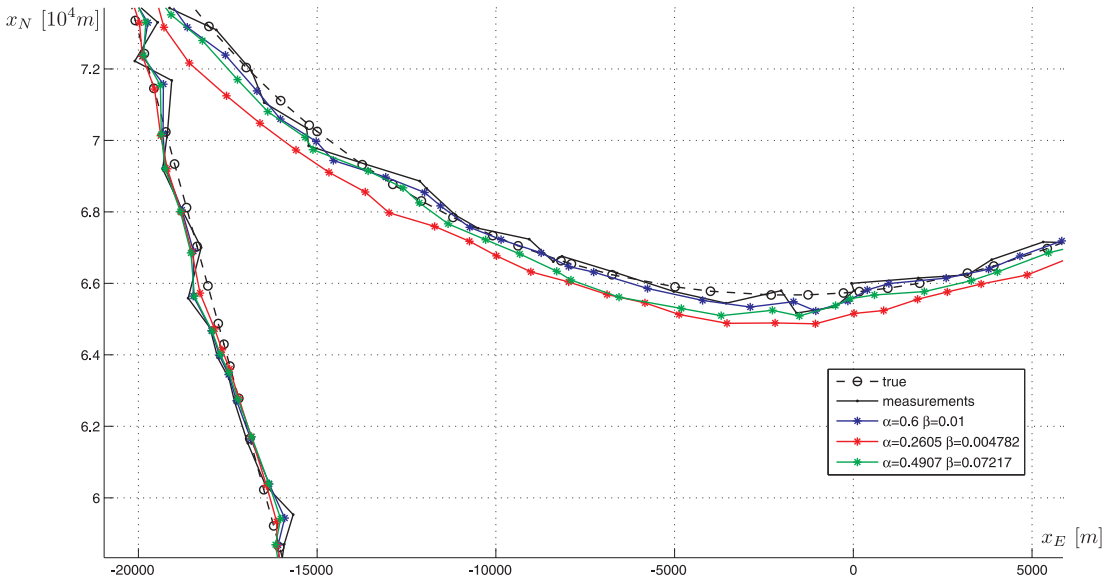

Figure 7: Reconstructed trajectory of the alpha-beta filter for parameters found in optimization.

References

[1] J. Bar-Shalom and X. Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. 1995.

[2] Y. Bar-Shalom and W. D. Blair. *Multitarget-Multisensor Tracking: Applications and Advances. Vol. III*. Artech House, 2000.

[3] D. A. Bedin and A. G. Ivanov. Multicriteria genetic optimization procedure for trajectory tracking by the interacting multiple model algorithm. In V. Turova, A. Kovtanyuk, H.G. Bock, F. Holzapfel, and E. Kostina, editors, *Proceedings of the Workshop on Mathematical Modeling and Scientific Computing: Focus on Complex Processes and Systems—dedicated to the memory of Nikolai Botkin*, volume 2783, pages 17–28. CEUR-WS.org, 2020. https://ceur-ws.org/Vol-2783/paper02.pdf.

[4] D. A. Bedin and A. G. Ivanov. Trajectory tracking by the interacting multiple model algorithm: Genetic approach to improve the performance. In F. Gini, M.S. Greco, V. Carotenuto, and L. Pallotta, editors, *Proceedings of 2020 IEEE Radar Conference (RadarConf20)*, pages 1–6, 2020. https://doi.org/10.1109/RadarConf2043947.2020.9266635.

[5] S. Bittanti and S.M. Savaresi. On the parametrization and design of an extended Kalman filter frequency tracker. *IEEE Transactions on Automatic Control*, 45(9):1718–1724, 2000.

[6] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House Inc, 1999.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 04 2002.

[8] A. Doucet and B. Ristic. Recursive state estimation for multiple switching models with unknown transition probabilities. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):1098–1104, 2002.

[9] EUROCONTROL. Eurocontrol specification for surveillance data exchange — ASTERIX Part 12. Category 21: ADS-B target reports. https://www.eurocontrol.int/sites/default/files/2021-12/asterix-adsbtr-cat021-part12-v2-6.pdf.

[10] EUROCONTROL. Eurocontrol standard for radar surveillance in en-route airspace and major terminal areas. https://www.eurocontrol.int/publication/eurocontrol-standard-radar-surveillance-en-route-airspace-and-major-terminal-areas, 1997. [Online; accessed 8-September-2024].

[11] J. Gray and W. Murray. A derivation of an analytic expression for the tracking index for the Alpha-Beta-Gamma filter. *IEEE Transactions on Aerospace and Electronic Systems*, 29:1064 – 1065, 08 1993.

[12] F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, 2000.

[13] C. F. Jekel and G. Venter. pwlf: A python library for fitting 1d continuous piecewise linear functions. https://github.com/cjekel/piecewise_linear_fit_py, 2019. [Online; accessed 7-September-2024].

[14] P. R. Kalata. The tracking index: A generalized parameter for α-β and α-β-γ target trackers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-20(2):174–182, 1984.

[15] 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*, volume 152, pages 16–22, February 11, 2005.

[16] E. L. Lehmann and G. Casella. *Theory of point estimation. 2nd ed. Springer texts in statistics*. Springer, 1998.

[17] 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*, pages 262–266, Lahore, Pakistan, 2001.

[18] IMM UrB RAS. Uran supercomputer. http://parallel.uran.ru/node/419.

[19] X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking: decision-based methods. In O.E. Drummond, editor, *Proceedings of SPIE—The International Society for Optical Engineering*, volume 4728, pages 511–534. International Society for Optics and Photonics, SPIE, 08 2002. https://doi.org/10.1117/12.478535.

[20] X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, 2003.

[21] X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321, 2005.

[22] I. Simeonova and T. Semerdjiev. Specific features of IMM tracking filter design. *Information & Security: An International Journal*, 9:154–165, 2002.

[23] A. Vahabian, A. K. Sedigh, and A. Akhbardeh. Optimal design of the variable structure IMM tracking filters using genetic algorithms. In S. Hara, Li-Chen Fu, Shuzhi Sam Ge, T. Samad, M. Sebek, and S. Engell, editors, *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.*, volume 2, pages 1485–1490 Vol.2, 2004.

[24] M. A. Zaveri, S. N. Merchant, and U. B. Desai. Tracking multiple point targets using genetic interacting multiple model based algorithm. In Wu-Sheng Lu, editor, *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, volume 3, pages III–917, 2004.

[25] A. Zhang and M. M. Atia. An efficient tuning framework for Kalman filter parameter optimization using design of experiments and genetic algorithms. *NAVIGATION: Journal of the Institute of Navigation*, 67(4):775–793, 2020.