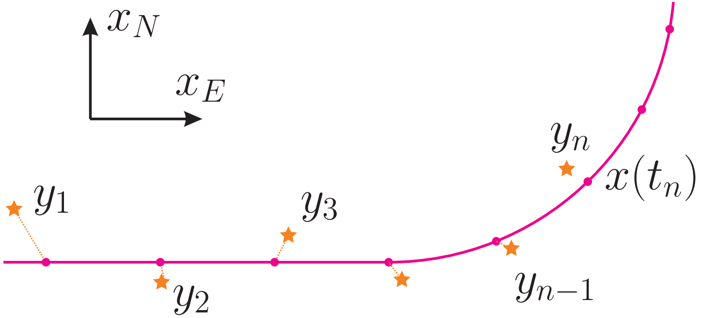
Multicriteria Genetic Optimization Procedure for Trajectory Tracking by the Interacting Multiple Model Algorithm

Dmitrii A. Bedin **Alexey G. Ivanov**

N.N. Krasovskii Institute of Mathematics and Mechanics
(IMM UB RAS)

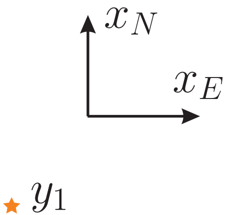
MMSC 2020:
Workshop on Mathematical Modeling and Scientific Computing: Focus on complex processes and systems
Technical University of Munich,
Munich, Germany, November 19-20, 2020

Outline

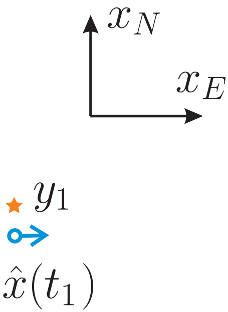


Our optimization object is the IMM (Interacting Multiple Model) algorithm of trajectory tracking (the tracker). In trajectory tracking, the subjects are a moving object and the trajectory of its motion. The trajectory can be curvylinear. The measurements of the object position with random noise appear consequently at discrete time instants.

Trajectory Tracking

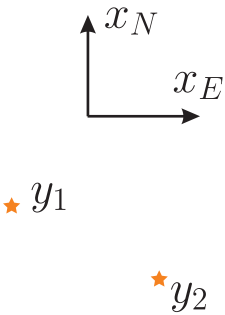


The measurements of the object position with random noise appear consequently at discrete time instants.



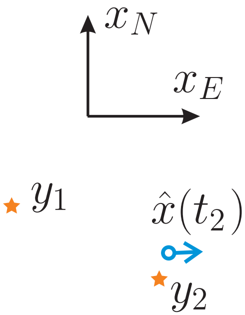
The IMM algorithm is a <u>*filtering*</u> *procedure. Using measurements up to the current time instant, it elaborates the esitmate of the position and the velocity. Now, at the instance* t_1*, there is a single measurement* y_1 *and the estimate* $\hat{x}(t_1)$ *(blue circle with arrow) can use only this measurement.*

Trajectory Tracking



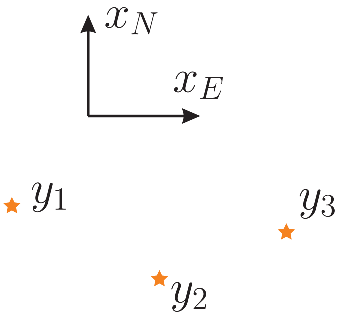
Now, there are two measurements y_1 *and* y_2.

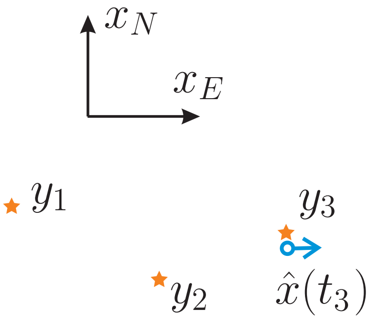
Trajectory Tracking



Now, there are two measurements y_1 and y_2. The estimate $\hat{x}(t_2)$ uses both of them.

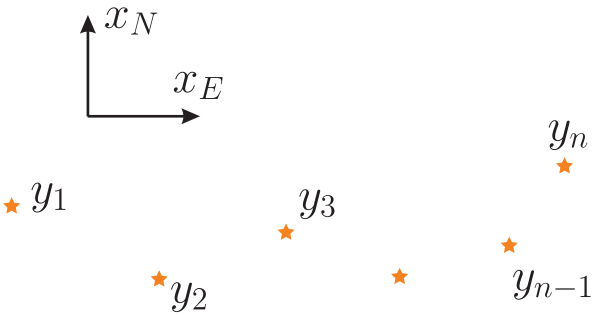
Trajectory Tracking



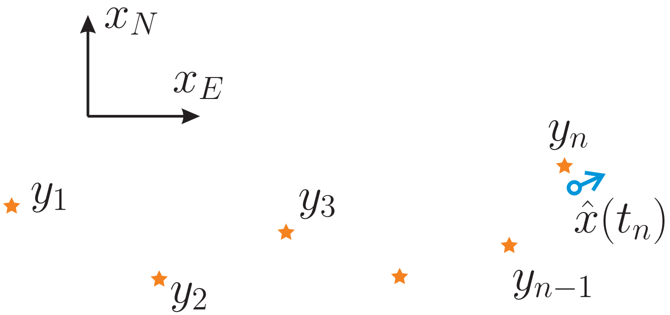


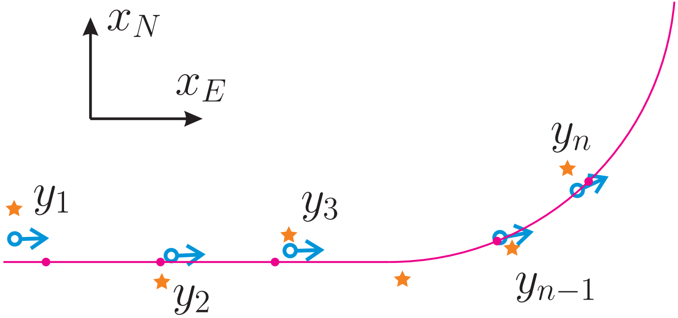
And so on.

Trajectory Tracking



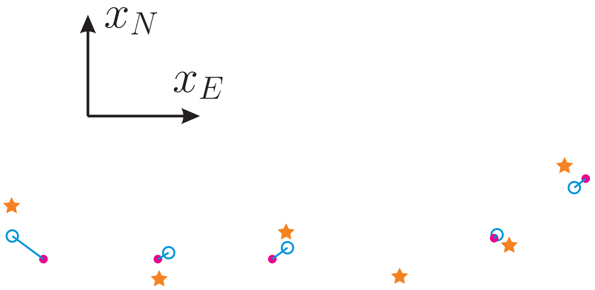
Trajectory Tracking





The estimates (blue circles with arrows) differ from the true positions (magenta dots) at the corresponding instants. These differences are subject of optimization. The less the differences are, the better the tracking algorithm is.

Trajectory Tracking



The estimates (blue circles with arrows) differ from the true positions (magenta dots) at the corresponding instants. These differences are subject of optimization. The less the differences are, the better the tracking algorithm is.

Trajectory Tracking

Trajectories can be very different, a tracking algorithm has to handle all possible trajectories. For example, it has to recognize the turn in this trajectory after the rectilinear motion section. If not, the errors will be large. The IMM handles this difficulty well.

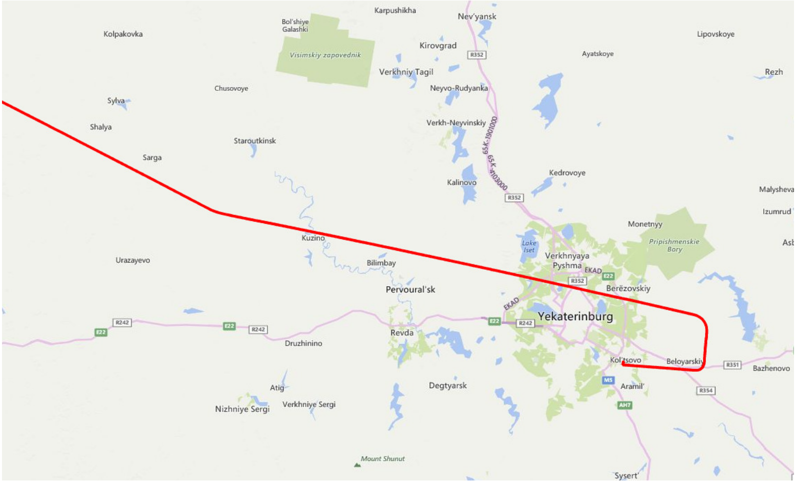
Trajectory Tracker by NITA

Our optimization object is the trajectory tracking program by the NITA company

http://nita.ru/en/main

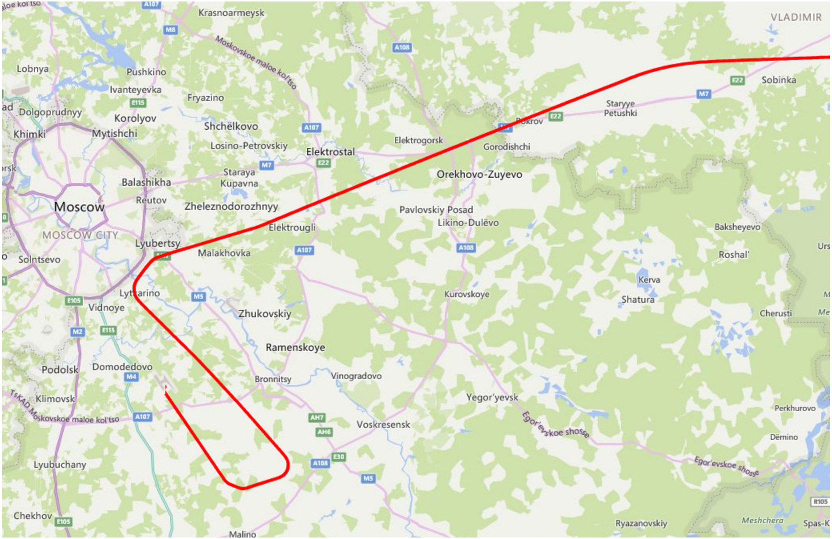
We collaborate with the NITA company and optimize their trajectory tracker. This company is the leader in Russia in the Air Traffic Management (ATM) solutions.

Planar Trajectories of Aircraft



This is a real aircraft trajectory from `flightradar24.com`.

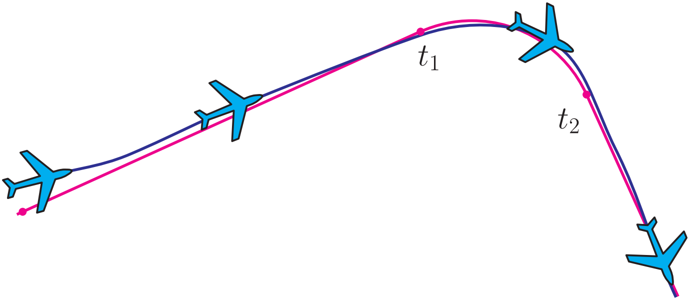
Planar Trajectories of Aircraft



This is a real aircraft trajectory from `flightradar24.com`.

Planar Trajectories of Aircraft

A trajectory of aircraft is close to a sequence of straight line and circular arc segments.



This is a simple, but sufficient model of the motion that we use.

Planar Trajectories of Aircraft

Approximate planar dynamics:

$$\begin{cases} \dot{x}_N = v \cos\varphi\,, \\ \dot{x}_E = v \sin\varphi\,, \\ \dot{\varphi} = u/v\,, \\ \dot{v} = w\,. \end{cases}$$

- x_N, x_E are the Cartesian coordinates in the plane;
- v, φ are the aircraft speed and course;
- u and w are the tangential and orthogonal accelerations, (unknown controls!)

Typical motion types:

- $u(t) = 0$, $w(t) = 0$ — constant velocity (CV);
- $u(t) = \text{const}$, $w(t) = 0$ — coordinated turn (CT);
- $u(t) = 0$, $w(t) = \text{const}$ — constant acceleration (CA).

Measurements

The measureable part of the state vector: the "geometrical" part

$$x_G = \begin{bmatrix} x_N \\ x_E \end{bmatrix}.$$

The measurement vector

$$y = \begin{bmatrix} y_N \\ y_E \end{bmatrix} = \begin{bmatrix} x_N \\ x_E \end{bmatrix} + \begin{bmatrix} w_N \\ w_E \end{bmatrix} = x_G + \eta\,.$$

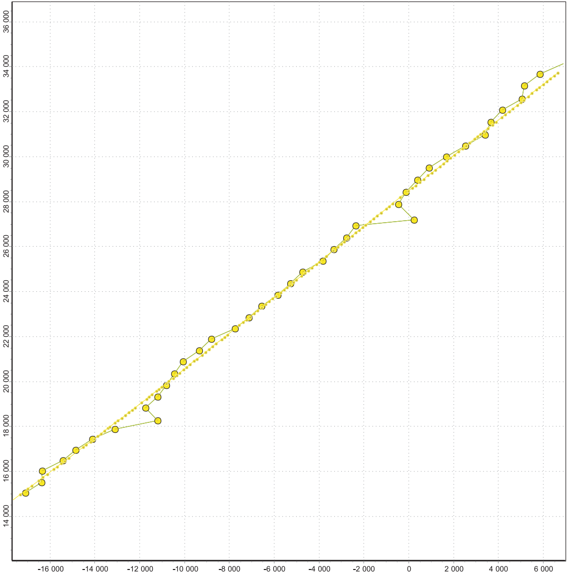
Measurements are made at discrete time instants

$$y(t_i) = x(t_i) + \eta(t_i)\,, \qquad t_i \in \{t_1, t_2, t_3, \ldots, t_n\}$$

with random errors. For example,

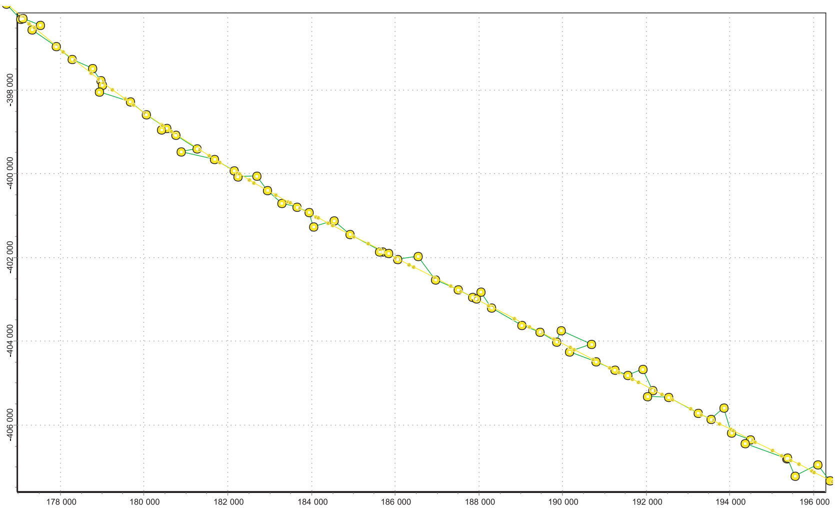
$$\eta(t_i) \sim \mathcal{N}(0, H_i).$$

Measurements



This is a fragment of a real measurement track.

Measurements



This is a fragment of a real measurement track.

Filtering Problem

It is needed to make an estimate $\hat{x}(t_n)$ (or $\hat{x}_n$) close to $x(t_n; u(\cdot), w(\cdot))$ using the history of measurements up to the instant t_n

$$Y_n = \{y(t_i) : \; i = 1, \ldots, n\} \; .$$

The estimator $\hat{x}(t_n)$ is a response to Y_n:

$$\hat{x}(t_n) = \hat{x}(t_n; Y_n) \, .$$

Performance criterion usually is

$$\mathbf{MSE}\left\{\hat{x}(t_n)\right\} = \mathbf{E}\left\{\Big(\hat{x}(t_n; Y_n) - x(t_n; u(\cdot), w(\cdot))\Big)^2\right\} .$$

Filtering Problem

- *Gustafsson, F.* Adaptive filtering and change detection. Wiley: 2000.
- *Li, X. R., Jilkov, V.* A survey of maneuvering target tracking. Part IV: Decision-based methods // Proc. SPIE. 2002. Vol. 4728, pp. 511–534.
- *Li, X. R., Jilkov, V.* A survey of maneuvering target tracking. Part V: Multiple-model methods // IEEE Trans. on AES. 2005. No. 4, pp. 1255–1321.
- *Bar-Shalom, Y., Li, X. R., Kirubarajan, T.* Estimation with Application to Tracking and Navigation: Theory, Algorithms, and Software. New York: Wiley, 2001.
- *Blom, H., Bar-Shalom, Y.* The interacting multiple model algorithm for systems with Markovian switching coefficients // IEEE Transactions on Automatic Control, Vol. 33, No. 8, 1988, pp. 780–783.
- *Gustafsson, F., et al.* Particle filters for positioning, navigation, and tracking // IEEE Transactions on Signal Processing, 2002. No. 50 (2), pp. 425–437.

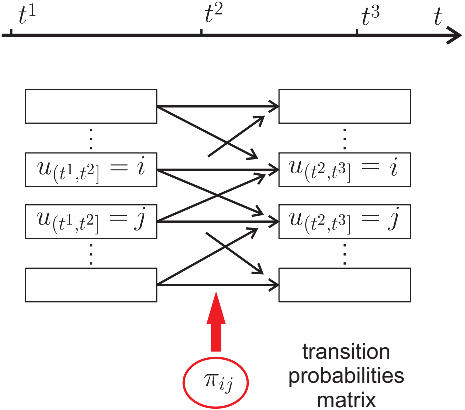
The filtering problem for the trajectory tracking is classic. There are a lot of publications about it.

Interacting Multiple Model Algorithm

Today, the IMM, Interacting Multiple Model algorithm, is the state-of-the-art solution for real-life problems with abruptly changing trajectories. Therefore, the IMM is widely applied in ATM where the air traffic controller side has to be ready for unscheduled sudden maneuvers of an aircraft.

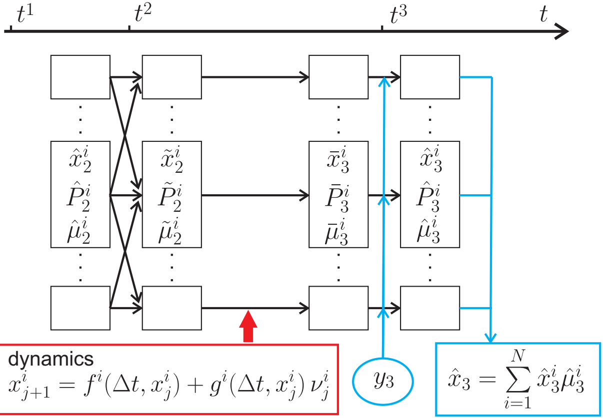
Interacting Multiple Model Algorithm

This is the scheme of IMM. Its core is the set of different models. Each model corresponds to some motion regime.
The Transition Probabilities Matrix (TPM) determines the switches between regimes (this is the parameter of the algorithm).



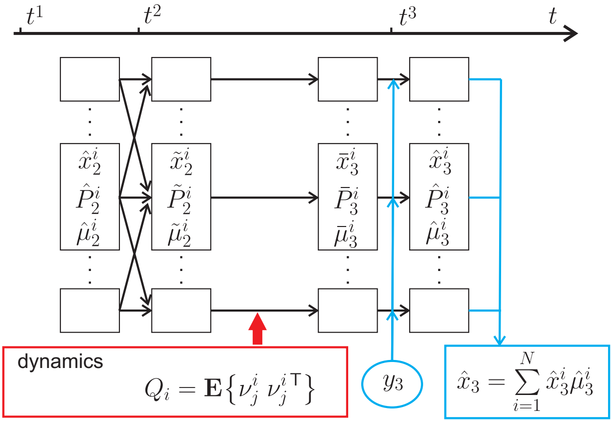
Interacting Multiple Model Algorithm

This is the scheme of IMM. Its core is the set of different models. Each model corresponds to some motion regime. Each model has its own dynamics and elaborates the partial estimate $\hat{x}_i$ of x at the last time instant (here, it is t_3). The final estimate is a mix of these partial ones. The jth dynamics include the system noise ν_j with a covariance matrix Q_j.

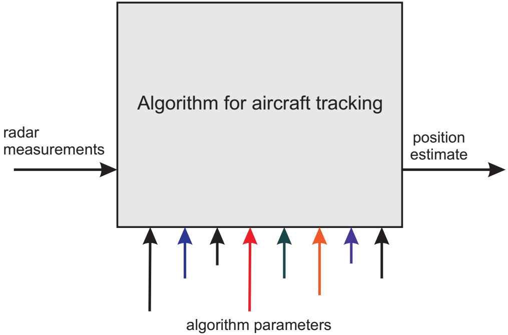


Interacting Multiple Model Algorithm

Each model has its own dynamics and elaborates the partial estimate $\hat{x}_i$ of x at the last time instant (here, it is t_3). The final estimate is a mix of these partial ones. The jth dynamics include the system noise ν_j with a covariance matrix Q_j.



Black Box Concept



Any trajectory tracking algorithm can be imagined as a black box: the radar measurements are at input, the position estimates are at output. There are some parameters influencing the algorithm behavior and the estimation quality. We optimize the tracker parameters without using any specific features of the IMM algorithm. We keep such a straightforward approach 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 real ATM system.

Standards by EUROCONTROL

- EUROCONTROL Specification for ATM Surveillance System Performance, Std.

  https://www.eurocontrol.int/publication/eurocontrol-specification-atm-surveillance-system-performance-esassp

- 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

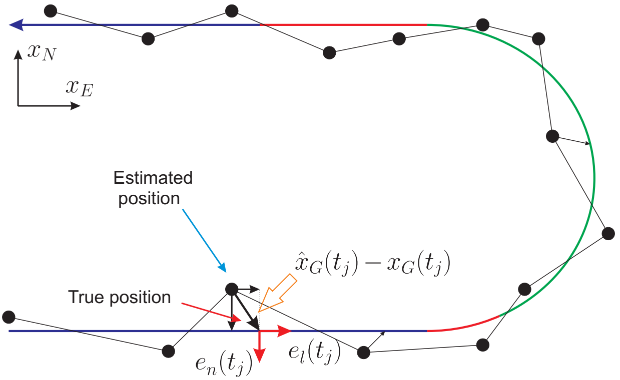
How do we assess the quality of estimation?
In ATC, there are many norms and specifications about the quality of estimation.
We follow Eurocontrol specifications in general.

Standards by EUROCONTROL

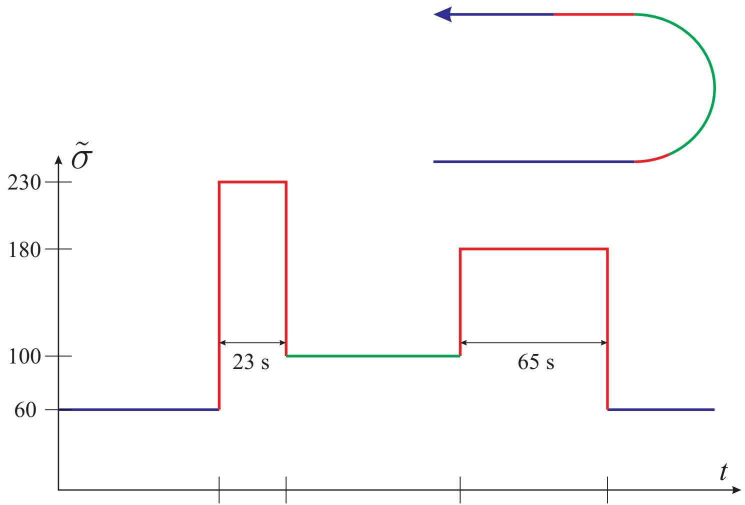
type of motion	en-route airspace: $\Delta t_z = 12$ s, $\sigma_r = 140$ m		major terminal area: $\Delta t_z = 4$ s, $\sigma_r = 70$ m	
	along deviation	across deviation	along deviation	across deviation
initialization phase	$\sigma_b = 999$ m $\Delta t = 30$ c	$\sigma_b = 999$ m $\Delta t = 30$ c	$\sigma_b = 999$ m $\Delta t = 10$ c	$\sigma_b = 999$ m $\Delta t = 10$ c
acceleration-free motion $u = 0,\ w = 0$	$\sigma_b = 120$ m	$\sigma_b = 120$ m	$\sigma_b = 60$ m	$\sigma_b = 60$ m
along acceleration $u = 0,\ w \neq 0$	$\sigma_b = 285$ m	$\sigma_b = 145$ m	$\sigma_b = 180$ m	$\sigma_b = 60$ m
turning $u \neq 0,\ w = 0$	$\sigma_b = 180$ m	$\sigma_b = 180$ m	$\sigma_b = 100$ m	$\sigma_b = 100$ m
transition $u = 0,\ w = 0 \to u \neq 0,\ w = 0$	$\sigma_b = 240$ m $\Delta t = 35$ s	$\sigma_b = 375$ m $\Delta t = 35$ s	$\sigma_b = 140$ m $\Delta t = 24$ s	$\sigma_b = 230$ m $\Delta t = 24$ s
transition $u \neq 0,\ w = 0 \to u = 0,\ w = 0$	$\sigma_b = 160$ m $\Delta t = 70$ s	$\sigma_b = 200$ m $\Delta t = 85$ s	$\sigma_b = 110$ m $\Delta t = 65$ s	$\sigma_b = 180$ m $\Delta t = 65$ s
transition $u = 0,\ w = 0 \to u = 0,\ w \neq 0$	$\sigma_b = 425$ m $\Delta t = 60$ s	$\sigma_b = 220$ m $\Delta t = 68$ s	$\sigma_b = 310$ m $\Delta t = 50$ s	$\sigma_b = 120$ m $\Delta t = 60$ s
transition $u = 0,\ w \neq 0 \to u = 0,\ w = 0$	$\sigma_b = 999$ m $\Delta t = 60$ s	$\sigma_b = 999$ m $\Delta t = 68$ s	$\sigma_b = 999$ m $\Delta t = 50$ s	$\sigma_b = 9999$ m $\Delta t = 60$ s

This is the table of prescribed by the EUROCONTROL standard upper limits σ_b of root mean squared error (RMSE) in the lateral and longitudinal channels in dependence on the type of motion segment (acceleration free, turning, transition between turning and acceleration free, etc.). Also, the upper limits on the transitional duration time Δ_t are given.

Esimate—Truth Differences

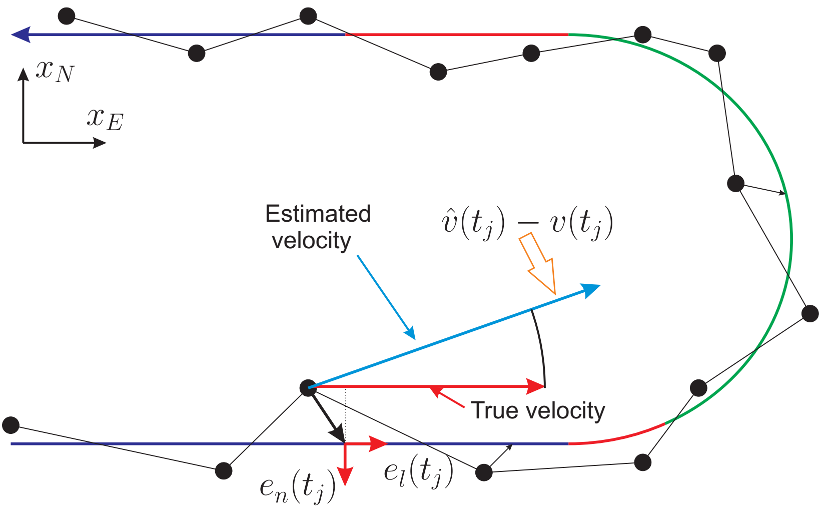


In this picture, the true trajectory is the thick multicolor line line, and the estimated positions are the black dots. The true trajectory is known during the simulations or, maybe, it can be measured by more accurate sensor (in some cases, GPS can be such a sensor). In the standards (by EUROCONTROL or in Russians standards too), it is conventional to project the difference between the estimate $\hat{x}_G$ *and the true* x_G *positions onto directions along* e_l *and across* e_n *the trajectory, in other words, onto the longitudinal and lateral directions.*
In the straight line motion segments (blue parts of the solid line), the upper RMSE limits are strict. In the turns (green segment), the limits are relaxed. The transitional segments are red. The durations of transitional processes are limited too.



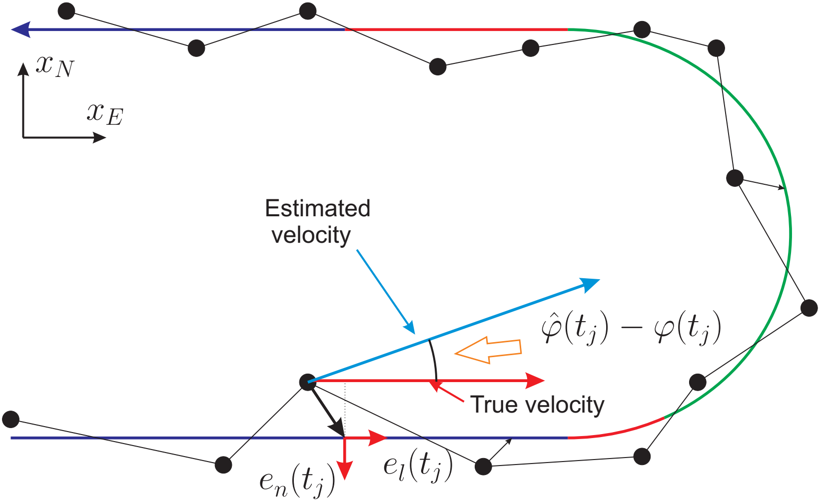
We can consider the upper RMSE limit $\tilde{\sigma}$ as a function of time t along the trajectory. Then, we can compare the differences $\hat{x}_G - x_G$ and their root mean squared error with it.

Esimate—Truth Differences



Also, in the standards, there are upper RMSE limits on differences of the velocity magnitude v

Esimate—Truth Differences



Also, in the standards, there are upper RMSE limits on differences of the velocity magnitude v
and the course φ.

The Standards Criticism

There are some questions about the usage of the standards.

1. *The accuracies of sensors are very different and differ from the ones described in the standards. What do we do if the sensor error is one half of the standard value? The rate of measurement arrival also varies.*
2. *The aircraft have different maneuverability. This influences the estimate error level and duration of the transitional processes. What formulas can describe this dependence?*
3. *Transitions between some motion segments are not described in the standards. Where can we get the limit values for these transitions?*

Cramér–Rao Lower Bound as a Reference

An attempt to answer the questions above leads us to the use of the Cramér–Rao lower bound (CRLB) instead of the upper RMSE limits from the standards as the reference accuracy.

The CRLB is the lower bound for the mean squared error and the covariance matrix of the error (in a vector case) for the unbiased estimates.

Real trajectory tracking algorithms usually produce biased estimates. Nevertheless, the CRLB value describes well potential accuracy of these algorithms.

The CRLB has an essential advantage: its value can be evaluated explicitly using the true trajectory and characteristics of the sensor system.

Cramér–Rao Lower Bound as a Reference

$$\mathbf{E}\{\hat{x}(t_n; Y_n)\} \equiv x(t_n; \theta),$$

$$u(t) = u(t; \theta), \qquad w(t) = w(t; \theta).$$

$$J(t_n; \theta) = \sum_{i=1}^{n} \frac{\partial x(t_i; \theta)}{\partial \theta}^{\mathsf{T}} (H_i)^{-1} \frac{\partial x(t_i; \theta)}{\partial \theta}.$$

$$\mathbf{E}\left\{ \left(\hat{x}(t_n; Y_n) - x(t_n; \theta)\right) \left(\hat{x}(t_n; Y_n) - x(t_n; \theta)\right)^{\mathsf{T}} \right\} \succcurlyeq$$
$$\succcurlyeq \left(\frac{\partial x(t_n; \theta)}{\partial \theta} \right) J(t_n; \theta)^{-1} \left(\frac{\partial x(t_n; \theta)}{\partial \theta} \right)^{\mathsf{T}}.$$

If the controls u and w are paramtrized by some parameters θ, then we can calculate the partial derivatives of the true states $x(t_i)$ and the Fisher information matrix J using the measurement noise covariances H_i. After matrix inversion, we get the lower bound formula. Here, $\succcurlyeq$ denotes the partial order in symmetric matrices in terms of positive semi-definiteness (Loewner order).

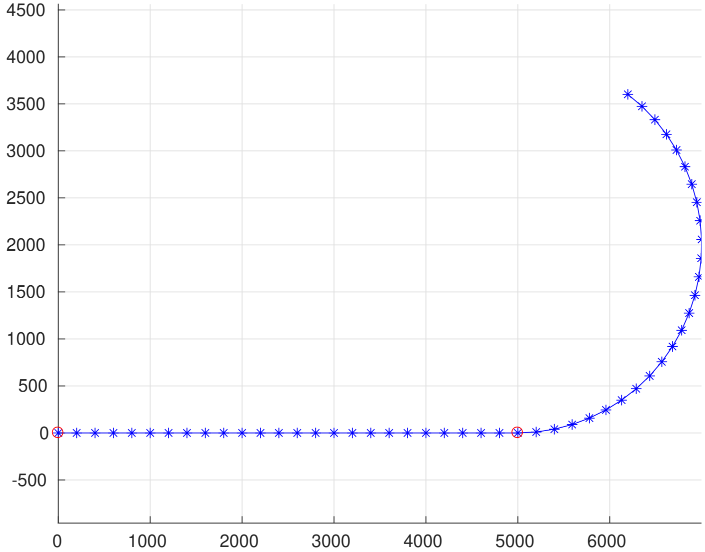
Cramér–Rao Lower Bound as a Reference

$$\begin{cases} \dot{x}_N = v\cos\varphi\,, \\ \dot{x}_E = v\sin\varphi\,, \\ \dot{\varphi} = u/v\,, \\ \dot{v} = w\,. \end{cases}$$

$$\theta = \begin{bmatrix} {\theta_0}^{\mathsf{T}} & u_0 & w_0 & t^1 & u_1 & w_1 & t^2 & u_2 & w_2 & \ldots \end{bmatrix}^{\mathsf{T}}.$$

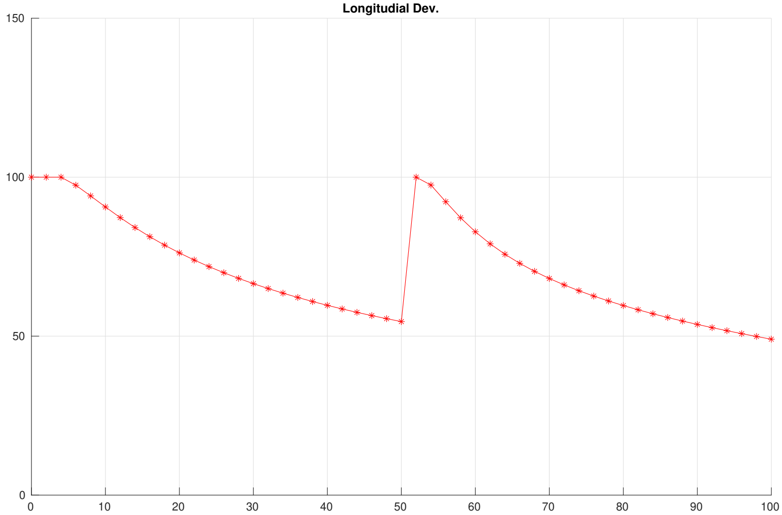
For example, the parameters θ of the controls u and w can consist of the values u_i and w_i of a piecewise-constant function of the time together with the switching instants t^i.

IMM Method vs. CRLB



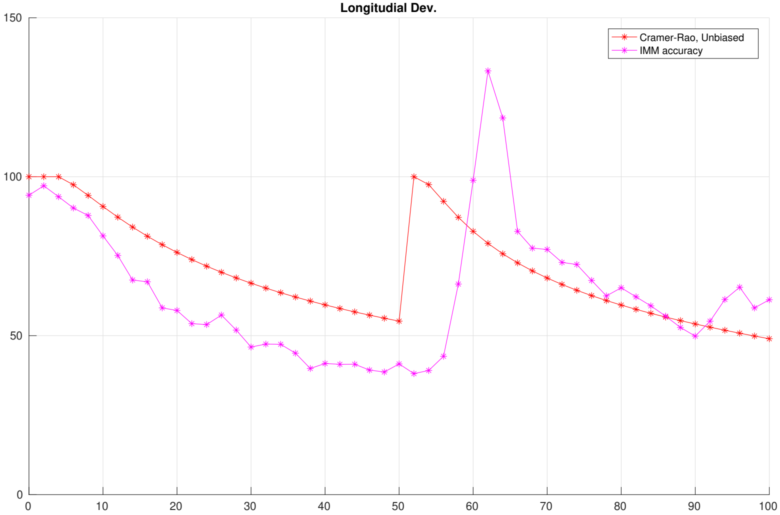
For this simpe trajectory with one straight-line and one coordinated turn segments . . .

IMM Method vs. CRLB



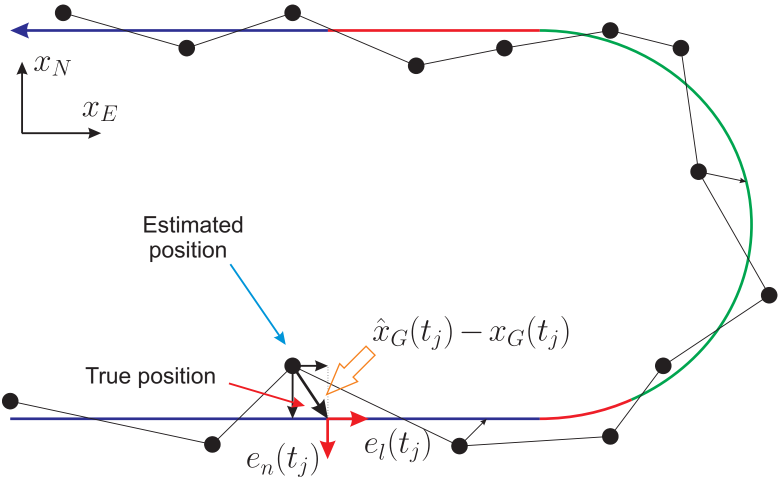
...we have this graph of the CRLB of the RMSE in the longitudinal channel.

IMM Method vs. CRLB



The accuracy of the real IMM algorithm is near CRLB, as you can see in this graph. The ratio of these values is constant almost always.

Performance Criteria



The tracking quality criteria can be formulated using the CRLB as follows. We consider the relative longitudinal and lateral deviations and compare them with the CRLB values "projected" onto the corresponding directions.

Performance Criteria

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

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

$$c_l = \sqrt{\mathbf{E}\left\{X_l^2(t)\right\}}.$$

We consider the relative longitudinal and normal deviations X_l *and* X_n *and want to use the expectations of mean squared deviations. (Instead of them, the empirical mean squared deviations are used in the algorithm.)*

Performance Criteria

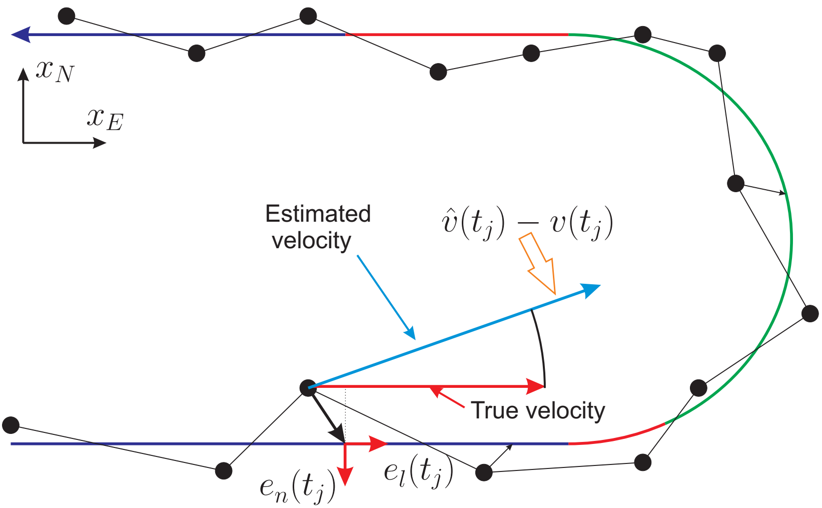
$$x_n(t_i) = e_n^\mathsf{T}(t_i)(\hat{x}(t_i) - x(t_i)),$$

$$X_n(t_i) = \frac{x_n(t_i)}{\sqrt{e_n^\mathsf{T}(t_i)J(t_i)e_n(t_i)}},$$

$$c_n = \sqrt{\mathbf{E}\left\{X_n^2(t)\right\}}.$$

We consider the relative longitudinal and normal deviations X_l and X_n and want to use the expectations of mean squared deviations. (Instead of them, the empirical mean squared deviations are used in the algorithm.)

Performance Criteria



Similarly, we consider the relative deviations of course and velocity magnitude.

Performance Criteria

$$x_v(t_i) = e_v^{\mathsf{T}}(t_i)(\hat{x}(t_i) - x(t_i)),$$

$$X_v(t_i) = \frac{x_v(t_i)}{\sqrt{e_v^{\mathsf{T}}(t_i)J(t_i)e_v(t_i)}},$$

$$c_v = \sqrt{\mathbf{E}\left\{X_v^2(t)\right\}}.$$

Similarly, we consider the relative deviations of course and velocity magnitude.

Performance Criteria

$$x_\varphi(t_i) = e_\varphi^\mathsf{T}(t_i)(\hat{x}(t_i) - x(t_i)),$$

$$X_\varphi(t_i) = \frac{x_\varphi(t_i)}{\sqrt{e_\varphi^\mathsf{T}(t_i) J(t_i) e_\varphi(t_i)}},$$

$$c_\varphi = \sqrt{\mathbf{E}\left\{X_\varphi^2(t)\right\}}.$$

Similarly, we consider the relative deviations of course and velocity magnitude.

Performance Criteria

The Mahalanobis distance in the "geometric" coordinates is

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

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

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

$$c_{2d} = \sqrt{\mathbf{E}\left\{X_{2d}^2(t)\right\}} \qquad c_{4d} = \sqrt{\mathbf{E}\left\{X_{4d}^2(t)\right\}}.$$

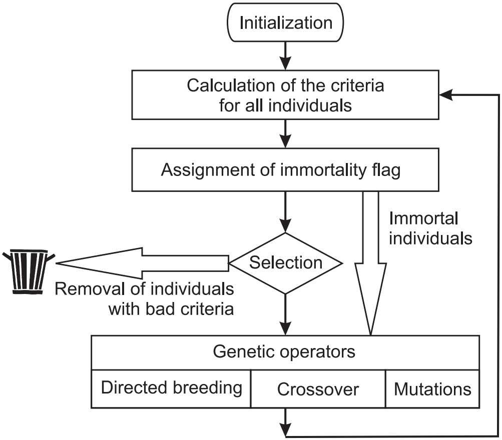
Additionally, we formulate the complex criteria c_{2d} and c_{4d}. They measure the relative deviations in many channels simultaneously.

The Problem

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

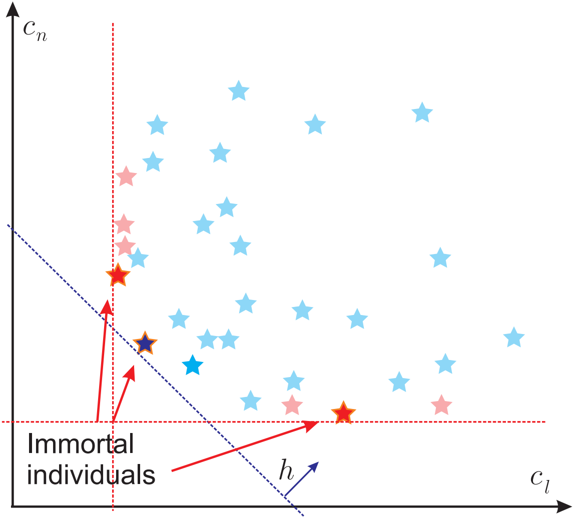
We have a multicriteria optimization problem with the parameter vector a. There are 16 parameters of the IMM implementation by NITA; therefore, $a \in \mathbb{R}^{16}$*. All the parameters are subject of box constraints.*

Genetic Algorithm



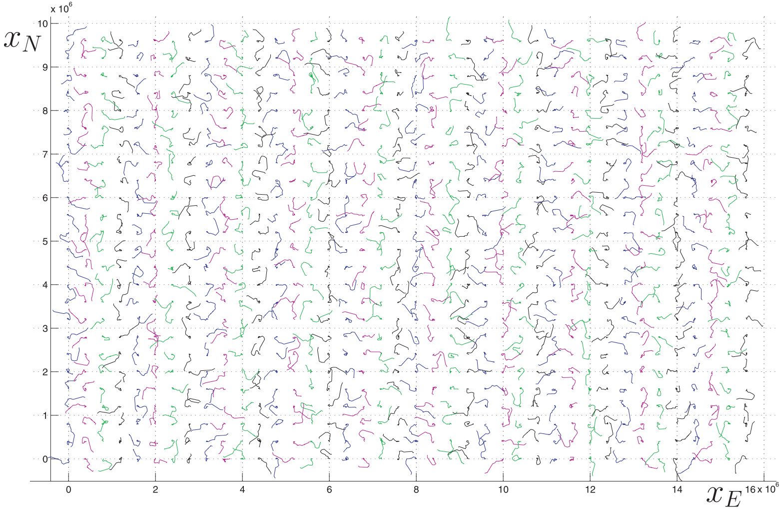
We use a genetic optimization algorithm since the parameters of the NITA IMM have complex and non-differentiable infulence on the performance metrics. The operations of the genetic algorithm (the directed breeding, the crossover, and the mutation) are quite usual. There is a population consisting of individuals. Every individual is connected with some point a *in the space* $\mathbb{R}^{16}$ *of parameters to be optimized (of the trajectory tracker in our case).*

Multicriteria Optimization



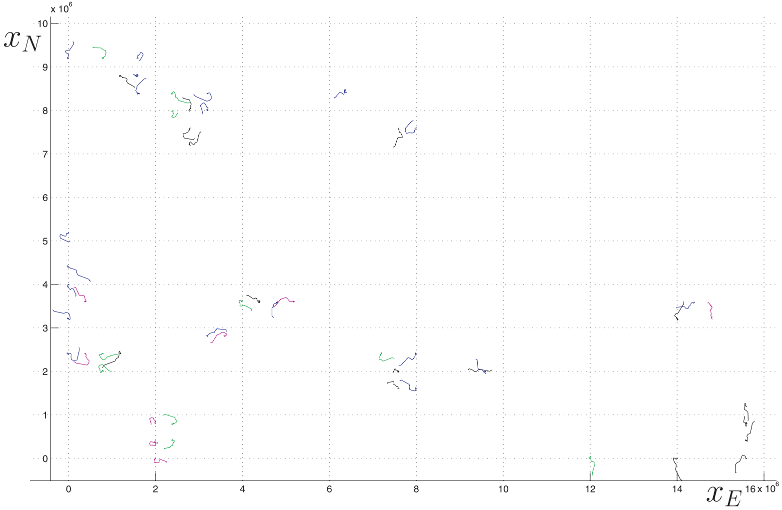
In this figure, the individuals are depicted in the criteria space (the criteria values are assigned to each individual). Since we have many criteria, we can have many minimum points. These individuals are marked as immortal.

One Batch per Generation



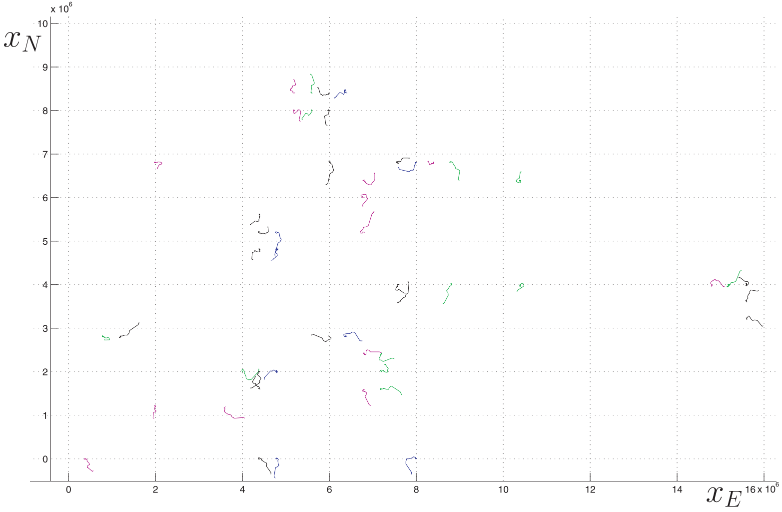
We use a big dataset for optimization.
We want to calculate the estimates of the criteria for each individual using the whole set, but it will be very expensive from the computational point of view.

One Batch per Generation



Instead of that, at every generation of the evolution process, we calculate the "preliminary" estimates on a small batch and then upgrade them at the succeeding generations. In the figure, a batch is shown ...

One Batch per Generation



. . . and this is another batch.

Estimates of the Criterion. Selection

The "true" criterion value c_l is the expectation

$$c_l = \sqrt{\mathbf{E}\left\{X_l^2(t)\right\}}.$$

We replace the expectation by its plug-in estimator

$$\hat{c}_{l,n}^2 = \frac{1}{n}\sum_{i=1}^{n} X_{l,i}^2\,, \quad \hat{c}_{l,n} = \sqrt{\hat{c}_{l,n}^2}.$$

N is the number of the trajectories $x_k(\cdot)$,
N_k is the number of the measurements in $x_k(\cdot)$,
$n = \sum_{k=1}^{N} N_k$ is the total number of the measurements.

$$\hat{c}_{l,n} \xrightarrow[n\to\infty]{\mathbf{P}} c_l\,, \qquad \mathbf{E}\left\{\hat{c}_{l,n}\right\} = c_l.$$

Estimates of the Criterion. Selection

We use the normal-based confidence interval

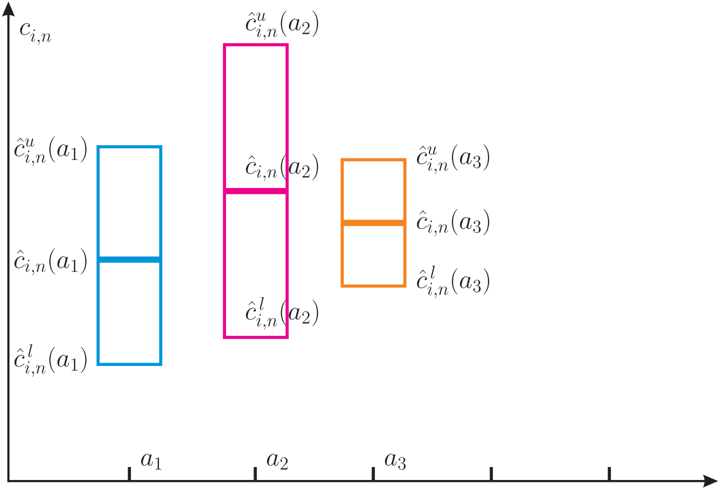
$$[\hat{c}^l_{l,n}, \hat{c}^u_{l,n}] = \left[\hat{c}_{l,n} - \hat{\mathsf{se}}_n(\hat{c}_{l,n}) z_{\alpha/2}, \hat{c}_{l,n} + \hat{\mathsf{se}}_n(\hat{c}_{l,n}) z_{\alpha/2}\right],$$

where $z_{\alpha/2}$ is the $(1-(1-\alpha)/2)$-quantile of $\mathcal{N}(0,1)$, and

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

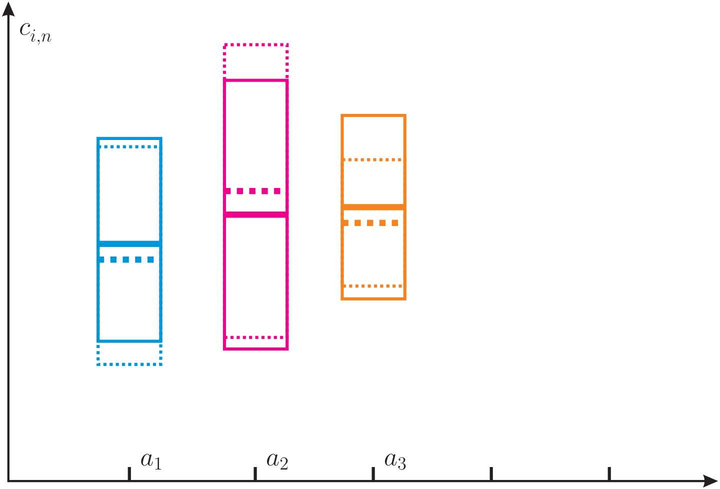
We estimate the criteria using simulations. There are random errors in their values.
For this reason, the lower and upper confidence bounds $\hat{c}^l_{i,n(a)}(a)$, $\hat{c}^u_{i,n(a)}(a)$ are used in the selection procedure to prevent deletion of the individuals whose current values of $\hat{c}_i$ are slightly worse than others due to random influence.
Note that the number of samples $n(a)$ in the estimates $\hat{c}_i(a)$ and their bounds $\hat{c}^l_{i,n(a)}(a)$, $\hat{c}^u_{i,n(a)}(a)$ depends on the individual a since the "lifetime" of the individuals can differ.

Estimates of the Criterion. Selection



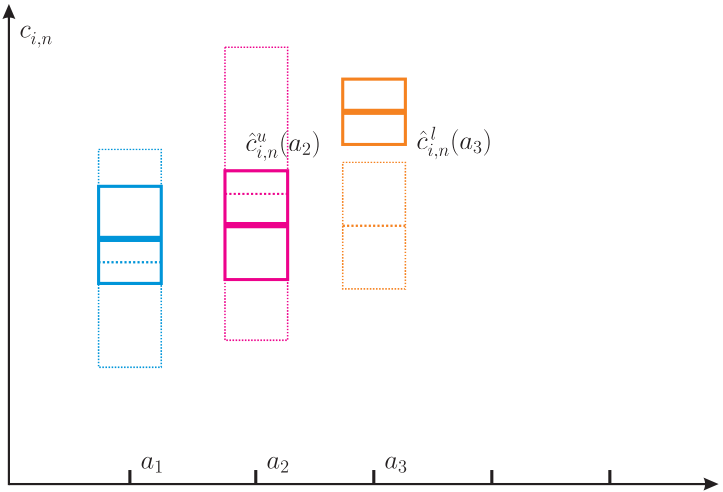
The criterion estimates $\hat{c}_i$ are depicted for three individuals a_1, a_2, and a_3 with the upper and lower bounds $\hat{c}^l_i$, $\hat{c}^u_i$ of the confidence intervals.

Estimates of the Criterion. Selection



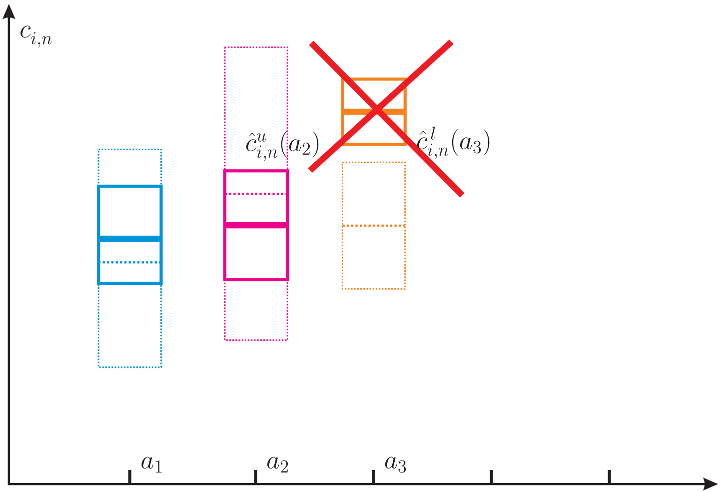
During simulations, the estimates $\hat{c}_i$ and the confidence intervals $\hat{c}_i^l$, $\hat{c}_i^u$ are changing. The width of an interval $\hat{c}_i^u - \hat{c}_i^l$ shrinks whlie the center $\hat{c}_i$ drifts.

Estimates of the Criterion. Selection



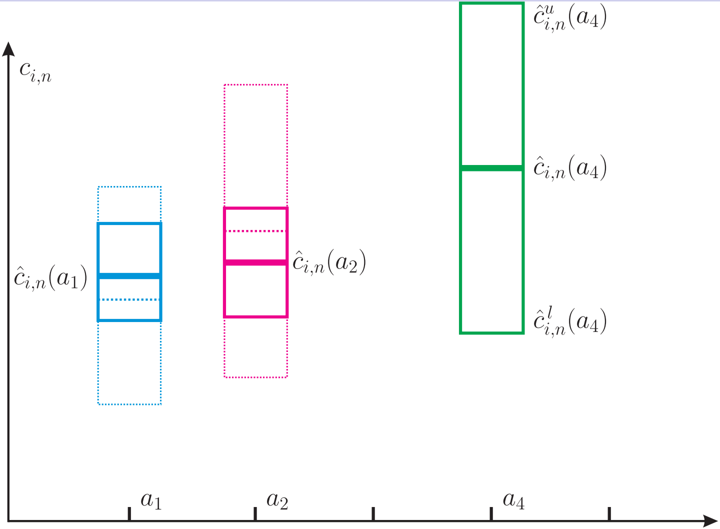
During simulations, the estimates $\hat{c}_i$ and the confidence intervals $\hat{c}_i^l$, $\hat{c}_i^u$ are changing. The width of an interval $\hat{c}_i^u - \hat{c}_i^l$ shrinks whlie the center $\hat{c}_i$ drifts.

Estimates of the Criterion. Selection



The individual a_3 is removed from the population if its lower confidence bound $\hat{c}^l_i(a_3)$ is greater than the minimum of upper confidence bounds $\min_{a\in P} \hat{c}^u_i(a)$.

Estimates of the Criterion. Selection



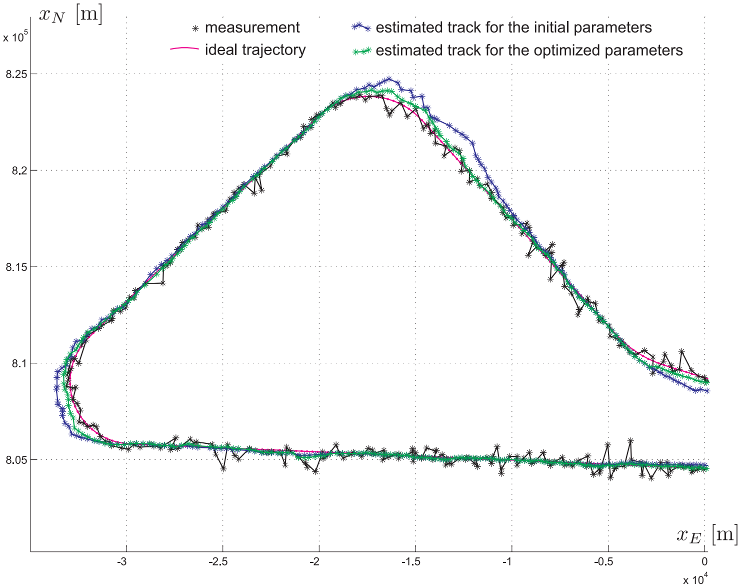
A new individual a_4 has the confidence interval $[\hat{c}^l_{i,n(a_4)}(a_4), \hat{c}^u_{i,n(a_4)}(a_4)]$ that is wider than the intervals of the "older" individuals since $n(a_4) \ll n(a_1), n(a_2)$. Despite the estimate $\hat{c}_i(a_4)$ is the worst, the individual a_4 is kept in the population.

Results on 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	2.86341	3.71362	3.2436	3.32135	5.11065	8.78238
Optimized	0.90609	0.94095	1.15279	1.18451	1.33594	2.64482

In the optimization, we use the test-train split as usual in machine learning. One trajectory set is for optimization (the train set) and another one is for validation. They consist of different trajectories.
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 the Table, the criteria values on the test trajectories are shown. All the criteria have lower values after optimization.

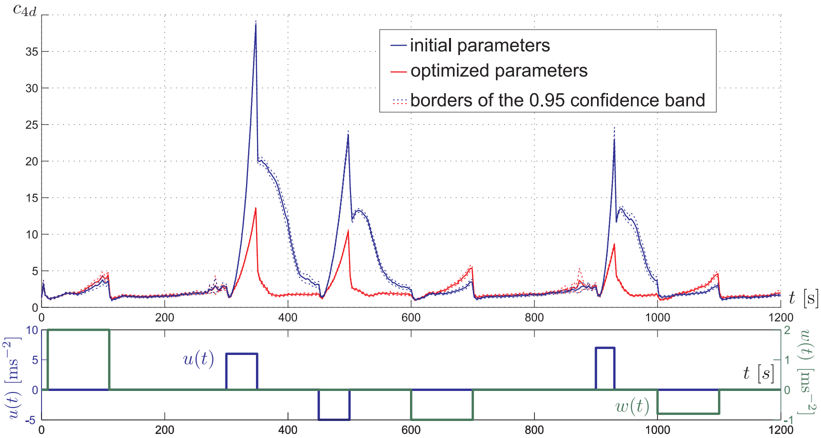
A Test Trajectory in Plane



A Test Trajectory in Plane

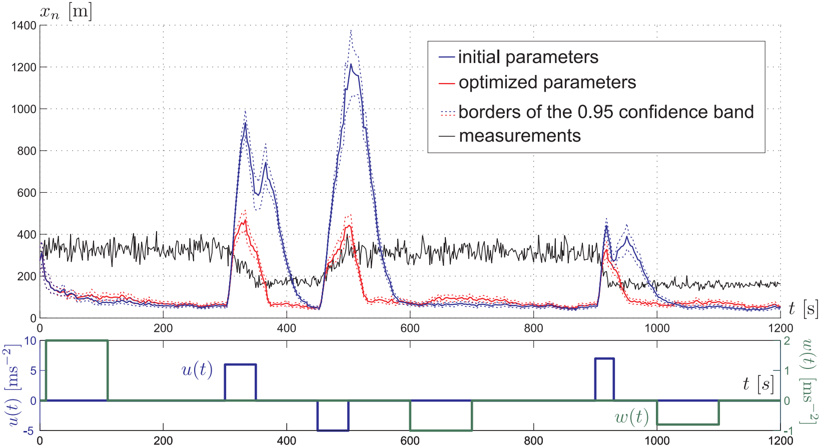
This is a fragment of some trajectory in the plane. The true simulated trajectory is magenta. The measurements are black. The blue line corresponds to the initial parameters. The green one is for the optimized parameters. You see that this line is closer to the true trajectory, especially near the turns.

A Test Trajectory in Plane



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

A Test Trajectory in Plane



The absolute (in meters) RMS deviation of x_n
in the lateral channel as a function of time.