Performance Evaluation of Multiple Target Tracking in the Absence of Reference Data

Robin Schubert, Horst Klöden, Gerd Wanielik
Professorship of Communications Engineering
Chemnitz University of Technology
Chemnitz, Germany
robin.schubert@etit.tu-chemnitz.de

Stephan Kälberer
Radar Algorithm
Magna Electronics Europe GmbH & Co KG
Ottobrunn, Germany
Stephan.Kaelberer@magnaelectronics.eu

Abstract – *Evaluating the performance of Multiple Target Tracking algorithms is a crucial requirement for the design and validation of different applications. Most of the available evaluation metrics require knowledge about the true state of the estimated situation. However, in most practical applications, such reference data are not available. Thus, a system of performance evaluation metrics which do not require reference data is proposed in this paper. The presented measures evaluate the detection performance, accuracy, and quality of the system output. The approach is evaluated based on simulated data and demonstrated on the example of an automotive tracking application.*

Keywords: Performance evaluation, MTT

1 Introduction

For many applications which rely on sensor measurements, Multiple Target Tracking (MTT) algorithms are a crucial part of the data processing chain. On the one hand, these techniques aim at estimating the number of targets in the sensor's field of view which is in general not known in advance. On the other hand, they are capable of estimating the state of each object, which may contain different quantities such as position or velocity. Finally, MTT techniques can associate an identifier to each entity in order to separate different objects from each other.

Despite the importance of MTT algorithms, there is no standardized design methodology available. In fact, an MTT designer has to choose among a considerable number of different filtering techniques which all have their specific advantages and disadvantages. Furthermore, an appropriate statistical model of the system under consideration needs to be derived. Finally, the MTT system needs to be parameterized, which may be considered as one of the most important steps of the design process.

For all those tasks, an objective *performance evaluation* of the chosen approach is necessary. On the one hand, a relative evaluation can be used to compare different algorithms, models, and parameters with each other in order to identify the optimal approach for the targeted application. On the other hand, an absolute performance evaluation is also necessary in order to verify that the implementation complies with given requirements.

An important issue for evaluation algorithms is the availability of *reference data*; that is, data about the true situation which is estimated by the MTT system. Reference data may include the number of objects, their identifiers, and their true states. Possible sources of such data are simulations or alternative sensors whose performance is some magnitudes higher than that of the system under evaluation.

The problem of performance evaluation has been widely studied in literature, including questions about its definition [1] and the relationship between the performance of a system and the complexity of the perceived situation [2]. In [3], it is shown that a majority of researchers are aware of the fact that reference data (which is sometimes also referred to as *ground truth*) are not available in most practical problems. However, a considerable amount of the available literature is focusing on performance metrics which at least partly require reference data (e.g. [4]) or try to generate such data from the measurements themselves [5].

Among the evaluation metrics which do not rely on reference data are *information theoretic measures* [6] which are based on quantities like information rates or entropy. While such techniques may provide important contributions to the evaluation process, they often require detailed knowledge about the system under evaluation (in particular about the nature of the involved probability density functions).

In this paper, a performance metric system for MTT algorithms is proposed which attempts to cope with unavailable reference data. For that, the fundamentals of MTT are firstly described in section 2. After that, performance metrics based on reference data are reviewed in section 3.1. Section 3.2 attempts to define equivalents for those metrics in the absence of reference data. The correlations between those two groups are evaluated based on simulations in section 4. Finally, an automotive case study is provided in section 5. The paper concludes with a discussion of the benefits and limitations of the proposed metric system in section 6.

2 Fundamentals of MTT

Multiple Target Tracking aims to detect, identify, and estimate the state of objects within a pre-defined region. The number of objects in that region is generally unknown. The state of an object at time k is described by a vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$ in an n_x-dimensional state space which is chosen upon the tracking problem specifics. Dimensions of the state space may contain for example kinematic states (e.g. position, velocity, or acceleration) or object related characteristics (e.g. a scattering coefficient). State estimation in real time applications is usually based on recursive estimator structures. In addition, this case requires each state to contain all previously received information, i.e. the *Markov condition*

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, ..., \mathbf{x}_0) \tag{1}$$

has to be fulfilled. The dynamic behavior of the objects can be modeled by a *state transition equation*, for target tracking sometimes also referred to as *motion model*,

$$\mathbf{x}_k = \mathbf{g}_a(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \tag{2}$$

that depends on the preceding state, only. The uncertainty of the stochastic model is represented by zero-mean process noise $\mathbf{v}_k$ with a known probability density function (pdf) $p(\mathbf{v}_k)$ and variance $var(\mathbf{v}_k) = \mathbf{Q}_k$.

Typically, the state vector $\mathbf{x}_k$ is not observable; however, it is assumed that each object causes a sensor detection $\mathbf{y}_k \in \mathbb{R}^{n_y}$ that is related to $\mathbf{x}_k$ by a known measurement model

$$\mathbf{y}_k = \mathbf{g}_c(\mathbf{x}_k, \mathbf{w}_k), \tag{3}$$

where $\mathbf{w}_k$ denotes zero-mean measurement noise with a known pdf $p(\mathbf{w}_k)$ and variance $var(\mathbf{w}_k) = \mathbf{R}_k$. The sequence of measurements received up to time step k is denoted by $\mathbf{Y}_k \widehat{=} \{\mathbf{y}_i, i = 1, ..., k\}$.

Figure 1 shows a generalized structure of a recursive MTT that contains the entities data association, track management, state filter, and calculation of gates. The sensors are not considered to be a part of the tracker; they rather act as interface between tracker and environment. It is assumed that there is a signal preprocessing step that creates object detections from the sensor raw data, i.e., the signal preprocessing decides which measurements originated from objects of interest.

At first, the tracker has to decide which of the new observations can be associated to already detected objects from former time steps. Assume the tracker has a prediction of the measurement $\hat{\mathbf{y}}_k = \mathbf{g}_c(\hat{\mathbf{x}}_{k|k-1}, 0)$ that an already known object should cause at the new time step. Next, a gating criteria is applied that defines a valid region around the predicted measurement. Observations that fall into the gate of a track are potential candidates to be associated with that track. Typically, rectangular or elliptical gates are used. A subsequent step decides within the set of these candidates about the final observation-to-track associations. For this, different association rules can be applied. The simplest rule uses the nearest neighborhood approach that assigns the statistically closest measurement to the track. More advanced rules allow the assignment of multiple measurements to a single track and the assignment of a single measurement to multiple tracks [7].

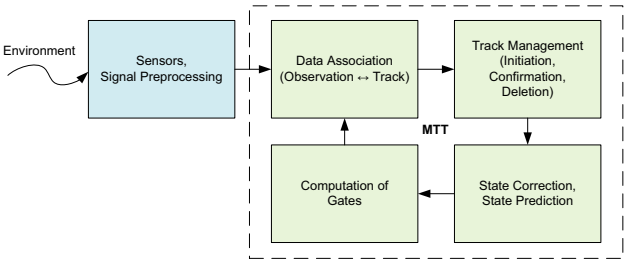

Figure 1: General Structure of a Multiple Target Tracker

After the data association step, the tracker decides how to deal with tracks in the future. Tracks can be created, for example, if observations could not be associated to an existing track. Tracks can be deleted, for example, if no new observations were assigned for a couple of consecutive time steps. Tracks can be confirmed, for example, if the probability of existence exceeds a certain threshold.

If new observations have been assigned to a track, the information of the measurement has to be incorporated. For this reason, a state filter is implemented. The filter algorithm accomplishes two steps

1. State Correction: Based on the measurement model, the new measurement is incorporated to correct the pdf of the predicted state $p(\mathbf{x}_k|\mathbf{Y}_{k-1})$ to $p(\mathbf{x}_k|\mathbf{Y}_k)$. The first iteration uses an initial distribution $p(\mathbf{x}_0)$.
2. State Prediction: Based on the motion model, the pdf of the state $p(\mathbf{x}_{k+1}|\mathbf{Y}_k)$ of the next time step is predicted.

Using the predicted pdf $p(\mathbf{x}_k|\mathbf{Y}_{k-1})$ and the likelihood distribution $p(\mathbf{y}_k|\mathbf{x}_k)$ that can be derived from the measurement model, the gates of the next time step can be defined. They build the basis of the next data association step.

The lack of a general approach for designing MTTs causes the need for a system of metrics that rates the performance of the actual design.

3 Performance Evaluation of MTTs

This section presents performance measures that can be used for performance evaluation of any multiple target tracker (MTT). Figure 2 shows an alternate view of the general structure of an MTT. One entity contains gating, data association and track management whereas the other two represent the filter steps of state correction and state prediction. It is assumed that the data at each output of an entity is accessible for the evaluator.

Algorithm-independent performance measures are usually based on the knowledge of reference data [8, 9, 10]. An overview of such quantities is given in the next subsection. Afterwards, performance measures are introduced that can be calculated without the knowledge of reference data.

3.1 Reference Data based Measures

The tracker performance is significantly dependent on the quality of the sensor observations. Thus, knowledge of the input data quality can be helpful to interpret other performance measures.

Based on detection theory, the probability of detection (PD) and the false alarm rate (FAR) can be determined.

$$FAR = \frac{N_{NC}}{n_T} \tag{4}$$

$$PD = \frac{N_C}{N_{TC}} \tag{5}$$

N_{NC} : *number of false detections*
n_T : *total number of frames*
N_{TC} : *theoretical number of detections*
N_C : *number of true detections*

Furthermore, the accuracy of the measurements can be evaluated by comparison with their theoretical values. The difference can be characterized by a constant error (Bias) and the mean squared error (MSE) of every dimension of the measurement space.[1]

$$Bias(\mathbf{y}) = \mathbf{y}_k - \mathbf{g}_c(\mathbf{x}_k, 0) \tag{6}$$

$$MSE(\mathbf{y}) = (\mathbf{y}_k - \mathbf{g}_c(\mathbf{x}_k, 0))\,(\mathbf{y}_k - \mathbf{g}_c(\mathbf{x}_k, 0))^T \tag{7}$$

In analogy to the input data evaluation, the detection performance can be evaluated on track level. The track probability of detection (TPD) calculates the probability that an object causes a confirmed track. The track false alarm rate (TFAR) calculates the average number of false tracks that exist at a time step. Since information of several time steps is used to maintain a track, the track detection performance (TPD, TFAR) should improve in comparison to the input detection performance (PD, FAR).

$$TPD = \frac{\sum_{i=1}^{N_{TT}} N_{D,i}}{\sum_{i=1}^{N_{TT}} N_{TD,i}} \tag{8}$$

$$TFAR = \frac{\sum_{i=1}^{N_T} L_i - \sum_{i=1}^{N_{TT}} N_{D,i}}{n_T} \tag{9}$$

N_{TT} : *total number of objects*
N_T : *total number of tracks*
$N_{D,i}$: *object i causes a track in $N_{D,i}$ frames*
$N_{TD,i}$: *object i exists in $N_{TD,i}$ frames*
L_i : *length of track i (number of frames)*

Additionally, the track detection performance is influenced by effects that occur over time. Track fragmentation describes the sequential deletion and creation of multiple tracks for one object. The track fragmentation rate (TFR) measures the rate of this event normalized over track length. Furthermore, track switching that describes the event of a track changing its tracked object may occur. The rate of this event normalized over object life time is defined as track switching rate (TSR).

[1] All following equations of the bias, mean, and smoothness require the calculation of the *sample mean*. For the sake of brevity, this is not explicitly denoted in the following.

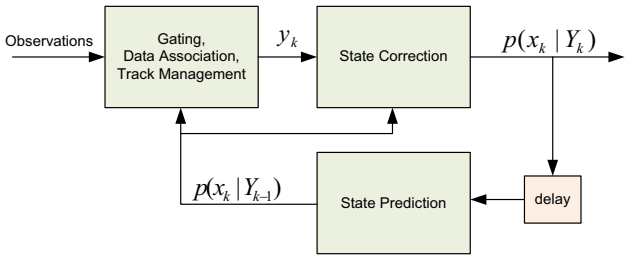

Figure 2: Alternative view of an MTT structure

$$TFR = \frac{1}{N_{TT}} \sum_{i=1}^{N_{TT}} \frac{\max(N_{TR,i} - 1,\, 0)}{N_{TD,i}} \tag{10}$$

$$TSR = \frac{1}{N_T} \sum_{i=1}^{N_T} \frac{\max(N_{O,i} - 1,\, 0)}{L_i} \tag{11}$$

$N_{TR,i}$: *number of tracks created for object i*
$N_{O,i}$: *number of objects tracked by track i*

The performance measures TPD, TFAR, TFR, and TSR give detailed information about the track detection performance. To characterize the track detection performance as a whole, it is helpful to determine a cumulative value. For this, a track management Score value is suggested in [8]. The Score value is increased at every time step for every object that is tracked by the same track. It is decreased at every time step by the number of existing false tracks. To simplify the interpretation of the absolute value, the Score can be normalized by its theoretical maximum.

Besides track detection performance, the track accuracy can be determined. This can be achieved by comparing the estimated track states with its theoretical values. The constant error (Bias) and the mean squared error (MSE) can be defined for every state in state space. The challenge is now to reduce the number of performance measures by finding meaningful cumulative values without knowing the dimensions of the states. This paper suggests to determine the difference between estimated and reference state $\Delta\mathbf{y}_k$ in measurement space. With the help of the input data accuracy, a weighted average can be calculated.

$$Bias(\mathbf{x}) = \left(\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k\right) \tag{12}$$

$$MSE(\mathbf{x}) = \left(\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k\right)\left(\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k\right)^T \tag{13}$$

$$Bias_c(\mathbf{x}) = \sum_{i=1}^{n_y} \left| \frac{Bias^{(i)}(\mathbf{g}_c(\mathbf{x}, 0))}{Bias^{(i)}(\mathbf{y})} \right| \tag{14}$$

$$MSE_c(\mathbf{x}) = \Delta\mathbf{y}_k^T MSE^{-1}(\mathbf{y}) \Delta\mathbf{y}_k \tag{15}$$

$$\Delta\mathbf{y}_k = \mathbf{g}_c(\hat{\mathbf{x}}_{k|k}, 0) - \mathbf{g}_c(\mathbf{x}_k, 0) \tag{16}$$

3.2 Measures without Reference Data

With reference data available, the tracking result can be compared with reality. Based on this comparison, measures characterizing the difference between these two can be defined. However, reference data is often unknown and the performance measures introduced cannot be determined. Nevertheless, it is possible to find performance measures that act as estimate of the aforementioned. It should be noted that these estimates contain presumptions that may not always hold. However, in most situations these values are useful indicators for the actual tracker performance.

At first, an estimate of the track detection performance has to be found. Empirical studies have shown that poor detection performance causes a short average track length L. The average track length usually decreases,

- if a track is only partially detected (TPD↓);
- if more track fragmentation occurs (TFR↑);
- if more false tracks exist (TFAR↑).[2]

However, using solely the average track length for measuring track detection performance may lead to erroneous results: Neglecting short tracks would automatically lead to a higher detection performance. For this reason, this paper suggests to combine the average track length (L) with the probability of association (PA) to get a cumulative measure (Score*) for the track detection performance. The probability of association denotes the probability that a sensor observation can be assigned to a track.

$$L = \frac{1}{N_T}\sum_{i=1}^{N_T} L_i \tag{17}$$

$$PA = \frac{N_A}{N} \tag{18}$$

$$Score^*(L, PA) = L \cdot PA^{c_g}, \quad c_g \geq 0^3 \tag{19}$$

N_A : *number of detections associated with a track*
N : *total number of detections*
c_g : *weighting parameter*

In order to determine the track accuracy, only the measurements can be used as reference. Every state estimate is projected into measurement space and compared to its assigned measurement. The difference $\Delta\hat{\mathbf{y}}_k$ can be characterized by the constant error (Bias*) and the mean squared error (MSE*) of every dimension in measurement space. A cumulative value for the MSE* can be calculated with the help of the a priori known measurement noise covariance matrix $\mathbf{R}$.

$$Bias^*(\mathbf{g}_c(\mathbf{x}, 0)) = \Delta\hat{\mathbf{y}}_k \tag{20}$$

$$MSE^*(\mathbf{g}_c(\mathbf{x}, 0)) = (\Delta\hat{\mathbf{y}}_k)\,(\Delta\hat{\mathbf{y}}_k)^T \tag{21}$$

$$MSE_c^*(\mathbf{g}_c(\mathbf{x}, 0)) = \Delta\hat{\mathbf{y}}_k^T \mathbf{R}^{-1} \Delta\hat{\mathbf{y}}_k \tag{22}$$

$$\Delta\hat{\mathbf{y}}_k = \mathbf{g}_c(\hat{\mathbf{x}}_{k|k}, 0) - \mathbf{y}_k \tag{23}$$

Comparing the tracks with measurements holds the risk of not catching the case of overfitting, i.e. tracks are constructed by solely connecting measurements. Although this may have a sufficient accuracy in some cases, the tracker works poorly if objects are closely spaced or if object detections are missed. Additionally, the tracker would fail to work for applications that need precise knowledge of non-observable states or need a reliable state prediction. To prevent overfitting, further criteria are needed.

One criterion may be the smoothness of a track. It is assumed that the sample rate of the sensors is sufficiently high so that adjacent state changes of the object movement are strongly correlated. The smoothness defines a measure that is lowered if state changes in a window of n adjacent samples are non-monotonic. The measure ranges between 0 and 1 and can be calculated for all states in state space. Since these values are non-dimensional, the average of all smoothness values can be used as cumulative value. In case of overfitting, the smoothness gives relatively small values.

$$S(\hat{\mathbf{x}}^{(i)}) = \frac{\left|\hat{\mathbf{x}}_{k|k}^{(i)} - \hat{\mathbf{x}}_{k-n|k-n}^{(i)}\right|}{\sum_{i=k-n+1}^{k}\left|\hat{\mathbf{x}}_{i|i}^{(i)} - \hat{\mathbf{x}}_{i-1|i-1}^{(i)}\right|} \tag{24}$$

$$S_c(\hat{\mathbf{x}}) = \frac{1}{n_x}\sum_{i=1}^{n_x} S(\hat{\mathbf{x}}^{(i)}) \tag{25}$$

Another criterion can be found by evaluating the prediction performance of the tracker. This can be achieved by comparing the predicted measurement $\mathbf{g}_c(\hat{\mathbf{x}}_{k+n|k}, 0)$ calculated at time k with the real measurement $\mathbf{y}_{k+n}$. Again, a cumulative value can be found by using the measurement noise covariance matrix $\mathbf{R}$. A tracker with good tracking performance shows usually a relatively small value for the mean squared prediction error PMSE*.

$$PMSE^*(\mathbf{g}_c(\mathbf{x}, 0)) = (\Delta\tilde{\mathbf{y}}_k)\,(\Delta\tilde{\mathbf{y}}_k)^T \tag{26}$$

$$PMSE_c^*(\mathbf{g}_c(\mathbf{x}, 0)) = \Delta\tilde{\mathbf{y}}_k^T \mathbf{R}^{-1} \Delta\tilde{\mathbf{y}}_k \tag{27}$$

$$\Delta\tilde{\mathbf{y}}_k = \mathbf{g}_c(\hat{\mathbf{x}}_{k+n|k}, 0) - \mathbf{y}_{k+n} \tag{28}$$

The definition of the PMSE* value raises the question how to choose the prediction time n. On the one hand side, n should be chosen to be very small so that $\mathbf{x}_k$ and $\mathbf{x}_{k+n}$ are strongly correlated. On the other hand side, if n is chosen to be too small the measure may be affected negatively if the measurement noise is not ideally zero-mean.

[2] The track length of false tracks is typically shorter because the probability of maintaining a track based on false alarms decreases with its length.

[3] The parameter c_g realizes the compromise between track length and probability of association; simulations have shown that in many cases the trivial choice of $c_g = 1$ gives already meaningful results.

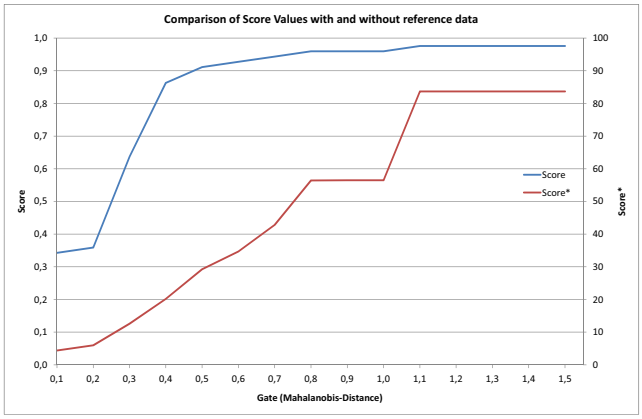

Figure 3: Comparison of the measures *Score* (using reference data) and *Score** (not using them) for different gates.

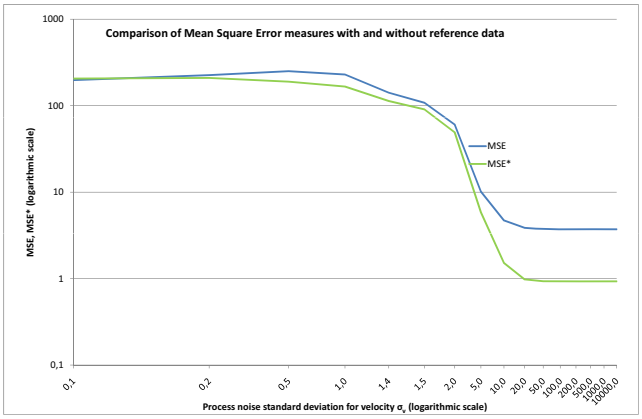

Figure 4: Comparison of the MSE measures with and without using reference data.

4 Simulative Analysis

4.1 Methodology

In order to validate the proposed performance measures, a simulation environment for automotive radars has been developed and applied. This simulator is able to generate the following data:

- Ego motion (velocity, acceleration, yaw rate)
- Road geometry (e.g. curvature, number of lanes)
- Object states (e.g. ID, position, velocity, yaw rate)
- Noisy[4] radar observations (e.g. range, angle, Doppler)

The radar observations serve as input for the MTT under evaluation, which is based on [11]. This MTT applies the Unscented Kalman Filter in combination with a Constant Turn Rate and Velocity model for estimating the states of the tracked objects. Furthermore, the Sequential Probability Ratio Testing (see [9]) is used for existence estimation.

4.2 Comparison of Performance Measures

The tracker performance can be calculated directly by comparing the tracker result with reference data. When no reference data is available, only performance estimates can be formulated. In the following, it will be analyzed how these estimates relate to the reference data based performance measures.

In section 3, a Score value was introduced that measures the detection performance of a tracker. Furthermore, it was stated that Score* defines an estimate of the detection performance. Figure 3 shows a simulation where the detection performance was determined for different gate sizes while all other parameters remained constant. It can be seen that the two measures Score and Score* are correlated. However, it should be noted that both values have different quantities and, therefore, the absolute values are not comparable. That is, Score* is an estimate of the detection performance, but not an estimate of the Score value.

[4]The simulator does not apply raytracing algorithms.

Moreover, it should be noted that the two performance measures are defined differently. As a result, the sensitivity to different detection errors is not necessarily the same. For example, other simulations have shown that the Score value is much less sensitive to track fragmentation than the estimated Score* value. This is the main reason for differences between Score and Score*.

Finally, section 3 proposed the (cumulative) MSE as a measure of track accuracy. It was claimed that an accuracy estimate MSE* can be formulated without the knowledge of ground truth data. Figure 4 shows a simulation where the track accuracy was determined for different variances of the process noise of the velocity σ_v. Clearly, the two accuracy measures MSE and MSE* are correlated. Theoretically, the relationship MSE* $= 1+$ MSE should hold for a sufficiently high number of samples. However, the measurement noise covariance matrix $\mathbf{R}$ must be known precisely. Usually, only an approximate of $\mathbf{R}$ is known. Thus, the quantities of the MSE* value should not be compared to the MSE value. The MSE* should rather be treated as alternate measure of the track accuracy.

In addition, it can be seen that a high process noise variance σ_v causes a significant difference between the MSE and the MSE* value. The high process noise leads to overfitting, i.e. the tracks are constructed by connecting the incoming measurements. As a result, the MSE* value experiences a strong decrease and, thereby, erroneously suggests a better accuracy. This is the major drawback of the MSE* estimate.

In conclusion, simulations have shown that the proposed performance measures determined without reference data are strongly correlated to the reference data based measures. Especially if the tracker works relatively close to optimality, valuable information can be gained. However, it should be noted that performance measures without the usage of reference data do have limitations, i.e., special conditions (e.g. overfitting) may lead to incorrect results.

5 Application

MTTs are integral components for many surveillance applications. As an example, the Automatic Cruise Control (ACC) will be analyzed. In contrast to ordinary cruise control systems, the ACC is able to detect the distance to the next vehicle and adjust the ego vehicle velocity to maintain a predefined distance.

For this application, the design of an MTT is not straightforward. At first, a motion model for vehicles has to be defined. For example, literature suggests the Constant Velocity Model (CV), the Constant Turning Rate and Velocity Model (CTRV), or the Constant Turning Rate and Acceleration Model (CTRA) [12].

Next, a filter algorithm has to be selected. The nonlinear filtering problem can be solved for example with an Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), or Particle Filter (PF). Also, rules for data association and track management have to be defined.

Finally, the multiple target tracker has to be parametrized, i.e., the process noise, the measurement noise, and all algorithm specific parameters (e.g. gate size) have to be specified. Typically, these design decisions are made on the basis of experience and empirical studies. To rate different tracker designs, performance measures are needed that reflect the system performance.

To demonstrate the practicability of the performance measures defined in section 3, simulations with an ACC system were made. The system is based on detections made by a doppler radar[5] $\mathbf{y} = (r, \varphi, \dot{r})^T$. Furthermore, a CTRV model with the state vector[6] $\mathbf{x} = (x, y, \gamma, v, \omega)^T$ that assumes constant velocity v and constant turning rate ω was applied. Both the measurement noise and the process noise were assumed to be Gaussian. To filter the object states, an Unscented Kalman Filter algorithm (UKF) was implemented.

With these design decisions made, the question how to choose good parameters is remaining. Different parameter sets are shown in table 1 and include the size of the gate (measured by the Mahalanobis distance), the process noise and the measurement noise.

[5] All measurements are relative to the own vehicle (distance r, angle of detection φ, and doppler $\dot{r}$).

[6] The states are x, y, γ are also defined relative to the own vehicle, while v and ω are absolute values.

Table 1: Parameter sets which are used for calculating different performance measures. The parameters consist of the size of the gate (as Mahalanobis distance), the process noise (σ_v and σ_ω), and the measurement noise (σ_r, σ_φ, and $\sigma_{\dot{r}}$).

	p_1	p_2	p_3	p_4	p_5
Gate	2.14	0.1	0.3	2.14	2.14
σ_v	5	5	5	1.5	20
σ_ω	0.2	0.2	0.2	0.2	0.2
σ_r	0.25	0.25	0.25	0.25	0.25
σ_φ	0.0087	0.0087	0.0087	0.0087	0.0087
$\sigma_{\dot{r}}$	0.2	0.2	0.2	0.2	0.2

Tables 2 and 3 show the performance measures for a simulated traffic scenario with these parameters. Based on observation, the tracker parametrized with p_1 shows almost optimal detection performance. This can be confirmed by the performance measures obtained. The Score value takes on almost the optimal value of 1. Also, the track probability of detection (TPD) is almost 1, whereas the track false alarm rate (TFAR) is 0. Additionally, no track fragmentation (TFR) and track switching (TSR) occurs. In contrast to the reference data based measures, the absolute values of the performance measures do only have significance if compared to other parameterizations.

Parameter set p_2 makes the track management process too selective by choosing a small gate size. Thus, it can be observed that the birth probability of tracks is too small and that tracks are deleted prematurely. As a result, TPD is relatively low and track fragmentation (TFR) occurs. Thus, the track management Score is decreased. In comparison to p_1, also the Score* estimate is decreased. That results from a smaller track length (L) due to track fragmentation and a smaller probability of association (PA) due to a small track probability of detection (TPD).

Likewise, parameter set p_3 (which defines a gate size smaller than p_1, yet larger than p_2) leads to a smaller detection performance (Score). In this case, the set of parameters causes the tracks to diverge from the true target. The tracker creates a new track when the former one exceeds a certain distance. The old track survives few more time steps before being deleted. Conclusively, TFR and TFAR are increased while TPD is acceptable. Again, this case can be caught without the knowledge of reference data. The decreased Score* value is caused by a smaller value of L while PA is constant due to an optimal TPD.

Visualizing the tracking result of parameter set p_1 shows a satisfying accuracy measured with MSE and MSE*, respectively. In contrast, parameter set p_4 leads to substantial differences between the state of the track and the true target due to the decreased process noise. Thus, both accuracy measures MSE and MSE* are significantly increased. This effect is caused by a lower process noise of the velocity component, which prevents the filter from fast adaptions to the measurements.

Parameter set p_5 seems to have a better accuracy than

Table 2: Performance measures for three different sets of parameters p_1, p_2, and p_3.

	p_1	p_2	p_3
Score	0.9758	0.3427	0.6371
TPD	0.9760	0.4600	0.9760
TFAR	0	0.1190	0.5397
TFR	0	0.0520	0.0600
TSR	0	0	0
Score*	83.6838	4.3698	12.5790
PA	0.6859	0.5042	0.6854
L	122.00	8.6666	18.3529

parameter set p_1. However, observing the tracking result shows the case of overfitting due to high process noise, i.e., the tracks seem to be constructed by connecting the incoming measurements. Though this is not reflected by the accuracy measures MSE or MSE*, this case can be detected with the help of the decreased track smoothness (S).

In conclusion, it could be shown that the system of performance measures is applicable to rate different parameterizations. Also, it was shown that conclusions can be drawn without the knowledge of reference data.

6 Conclusions

In this paper, an attempt was made to define performance measures for Multiple Target Trackers without having reference data available. This proposal is based on a set of well-known measures which assess detection performance, track accuracy, and quality. It was shown in section 3.2 how such measures can be partially estimated.

In addition, it was shown under simulative conditions that the detection performance can in many cases be approximated by the measure Score*. Likewise, the correlation between the true MSE and its estimate were illustrated. Despite these results, it is important to emphasize that performance measures without reference data can *per definitionem* not achieve the same normative significance as their reference data based counterparts. Examples for potential pitfalls have been given in the previous sections.

However, it could be shown in section 5 that the proposed measures can still be helpful for practical applications. During the evaluations, it turned out that human designers tend to optimize the parameters mainly with respect to the detection performance; that is, the parameters are tuned in such a way that a high detection rate and low false alarms are achieved. On the other hand, human designers usually fail to optimize the track accuracy (measured for example by the MSE). For these quantities, the proposed measures proved to be a helpful tool for parameterization.

Future work will mainly deal with the problem of automatic parameterization of MTT. The performance measures presented in this paper may serve as a starting point for research in this field. However, much effort needs to be spend in order to develop general, application-independent techniques for designing MTTs.

Table 3: Performance measures for three different sets of parameters p_1, p_4, and p_5.

	p_1	p_4	p_5
MSE	10.1641	108.0972	3.8719
MSE*	5.8504	90.7355	1.5179
S	0.6091	0.6652	0.5080

References

[1] A. Kott and W. Milks. Approaches to validation of information fusion systems. In *12th International Conference on Information Fusion*, pages 882 –889, 2009.

[2] Chee-Yee Chong. Problem characterization in tracking/fusion algorithm evaluation. *Aerospace and Electronic Systems Magazine, IEEE*, 16(7):12 –17, 2001.

[3] J. van Laere. Challenges for IF performance evaluation in practice. In *12th International Conference on Information Fusion*, pages 866 –873, 2009.

[4] R. Canavan, C. McCullough, and W.J. Farrell. Track-centric metrics for track fusion systems. In *12th International Conference on Information Fusion*, pages 1147 –1154, 2009.

[5] H. Leung, Zhijian Hu, and M. Blanchette. Evaluation of multiple radar target trackers in stressful environments. *Aerospace and Electronic Systems, IEEE Transactions on*, 35(2):663 –674, Apr 1999.

[6] Huimin Chen, Genshe Chen, E.P. Blasch, P. Douville, and K. Pham. Information theoretic measures for performance evaluation and comparison. In *12th International Conference on Information Fusion*, pages 874 –881, 2009.

[7] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and data association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[8] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[9] S. Blackman. *Multiple-Target Tracking with Radar Applications*. Norwood, MA: Artech House, 1986.

[10] S. Coraluppi, D. Grimmet, and P. de Theije. Benchmark Evaluation of Multistatic Trackers. In *9th International Conference on Information Fusion*, 2006.

[11] E. Richter, R. Schubert, and G. Wanielik. Radar and Vision based Data Fusion - Advanced Filtering Techniques for a Multi Object Vehicle Tracking System. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 120–125, 2008.

[12] R. Schubert, E. Richter, and G. Wanielik. Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking. In *11th International Conference on Information Fusion*, 2008.