	OPEN ETCS	

openETCS “Determine Train Location” Procedure

Author : Uwe Steinke Date : 2014-12-17

Subject :
OpenETCS WP3
- provisional -

Document Review : ☑
Design Review : ❒
Other :

Name	Position	Company / Department

Distribution to:

Name	Position	Company / Department
datadictionary@openetcs.org; openETCS srs-analysis@openetcs.org		

TABLE OF CONTENT

1. OPENETCS "DETERMINE TRAIN LOCATION" PROCEDURE

1.1 REFERENCES

UNISIG Subset_026 version_3.3.0

Chapter 3.6 : Location Principles, Train Position and Train Orientation

1.2 OBJECT

This document specifies the mathematical relationships needed for train position calculation.

This document version reflects only those equations that have been used within the CalculateTrainPosition SCADE model implementation at https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/TrainPosition/CalculateTrainPosition.

2. BASIC INACCURACY CALCULATIONS

2.1 ADDITION OF INDEPENDENT TOLERANCE VALUES

l is a triple of a nominal value l_{nom} with a minus tolerance Δl_{min} (negative) and a plus tolerance Δl_{max} (positive) so that

$$l_{nom} + \Delta l_{min} \le l \le l_{nom} + \Delta l_{max} .$$

The addition of 2 such values leads to

$$(l_{1,nom} + \Delta l_{1,min}) + (l_{2,nom} + \Delta l_{2,min}) \le (l_1 + l_2) \le (l_{1,nom} + \Delta l_{1,max}) + (l_{2,nom} + \Delta l_{2,max}) .$$

For the addition of more values in general:

$$\sum_i (l_{i,nom} + \Delta l_{i,min}) \le \sum_i (l_i) \le \sum_i (l_{i,nom} + \Delta l_{i,max}) \qquad [[\ 2.1\ ]]$$

This equation in a different notation

$$\sum_i (l_{i,nom}) + \sum_i (\Delta l_{i,min}) \le \sum_i (l_i) \le \sum_i (l_{i,nom}) + \sum_i (\Delta l_{i,max})$$

... which means that the tolerances of the sum equals the sum of the tolerances.

This applies only, if each of the values has its own tolerances independent from each other.

2.2 SUBTRACTION OF TOLERANCE VALUES

The substraction of 2 tolerance affected values leads to

$$(l_2 - l_1)_{min} \le (l_2 - l_1) \le (l_2 - l_1)_{max} \qquad [[\ 2.2\ ]]$$

The minimum and maximum tolerance limits are generated by the suitable combination of min/max of values 1 and 2:

$$(l_2 - l_1)_{nom} + (\Delta l_{2,min} - \Delta l_{1,max}) \le (l_2 - l_1) \le (l_2 - l_1)_{nom} + (\Delta l_{2,max} - \Delta l_{1,min})$$

2.3 DISTANCES BETWEEN LINKED ELEMENTS (BALISEGROUPS, ...)

The rules of chapter 2.1 and 2.2 refer to distances between elements along the track in general.

But for distances between linked elements, there is an important difference: Linked elements like balises are – as specified in Subset 026-3.6 – thought to be positioned on an absolutely correct nominal position with known min/max accuracy around the nominal position.

Therefore, the tolerances of 2 and more linking distances between balises must not be summed up as calculated in chapter 2.1. Instead, only the positioning inaccuracies of the first and the last balise group in a chain of linking distances is relevant for distance calculation:

$$\sum_1^n (l_{i,nom}) + (\Delta l_{1,min}) + (\Delta l_{n,min}) \le \sum_1^n (l_i) \le \sum_1^n (l_{i,nom}) + (\Delta l_{1,max}) + (\Delta l_{n,max}) \qquad [[\ 2.3\ ]]$$

3. ODOMETRY MODEL

The odometry subsystem is assumed to provide its own inaccuracies with the nominal, minimum and maximum travelled distance since system start as inputs to the OBU software:

- o_{nom} = nominal value
- o_{min} = minimum value
- o_{max} = maximum value

At system start all values are 0.

$$l_{odometry,nom,t=0} = l_{odometry,min,t=0} = l_{odometry,max,t=0} = 0$$

The distance between two odometry locations is presumed to

$$d_{nom} = (o_2 - o_1)_{nom} = o_{2,nom} - o_{1,nom}$$
$$d_{min} = (o_2 - o_1)_{min} = o_{2,min} - o_{1,min} \qquad [[\ 3.1\ ]]$$
$$d_{max} = (o_2 - o_1)_{max} = o_{2,max} - o_{1,max}$$

After odometry distances have be calculated from odometry outputs in this way, they can be used in subsequent uncertainty calculations by applying [[2.1]] and [[2.2]].

The original outputs of the odometry must never be manipulated by the OBU software.

4. LOCATIONS, POSITIONS AND COORDINATE SYSTEMS

4.1 LOCATION AND POSITION DEFINITION

For the rest of this document, we will differentiate locations from positions:

- Location: Locality of a stationary element on track side like balise group, signal, profile element etc.
- Position: Current locality of a mobile object, i. e. the train position.

4.2 COORDINATE SYSTEMS

The track can be seen as a one-dimensional coordinate system.

4.2.1 *The True Coordinate System*

The track with its dimension is the truth and makes up the true coordinate system. Its origin can be chosen arbitrarily, because the train operates on distances only.

All stationary elements on the track are placed on a true location; the train is at any time on a true position.

True locations and true positions are not affected by inaccuracies. They are exactly where they are.

We will designate all true locations, positions and distances with the subscripted string “true”:

$$pos_{true} , loc_{true} , d_{true}$$

4.2.2 *Apparent Coordinate Systems*

The train is unable to recognize the true coordinate system directly and without any deviation. It “sees” its position and the locations of track elements by using the odometry, by detecting balise groups und using linking information.

The trains perspective of the track and the true coordinate system is not unbiased: track elements appear on locations different from their true locations. These are “apparent” positions, locations and distances as seen by the appropriate observer, in our case the OBU.

4.2.2.1 The Odometry Coordinate System

Each location and position within the true coordinate system has a corresponding locality measured by the train’s odometry, as soon as the train already passed it. Therefore, the odometry coordinate system typically ranges from where the train trip started up to the current train position, but not to any elements ahead.

We will designate locations, positions and distances measured via the odometry with the subscripted string “odo”:

$$pos_{odo}, \qquad loc_{odo} , \qquad d_{odo}$$

Due to the uncertainties of odometry measurement, these values are afflicted by inaccuracies.

4.2.2.2 The Linking Coordinate System

Balise groups (BGs) marked as linked and with linking information available creates a coordinate system with (relative) locations of BGs and announced track elements. Linked BGs are used as the only location references and form a discrete coordinate systems with no location information in between. Since linked BGs with linking data might not be available on the whole track and comprise linking holes, the linking coordinate system covers the track only partially.

The BGs are not mounted exactly at their intended true location, but within a $\pm q\_locacc$ range. In addition, the train does not detect them exactly where they are mounted, but with an $\pm centerDetectionInaccuracy$ range caused by balise antenna reception. In contrast, linking distances d_{link} announced via linking information received from the track are per definition exact.

Distances between BGs have to be calculated according to [[2.3]]. The uncertainties of the first and the last element in the linking chain are simply the sum of two inaccuracy values from q_locacc and $centerDetectionInaccuracy$ according to [[2.1]].

4.2.2.3 The OBU Coordinate System

While calculating the location of BGs, track elements and the current train position, the calculateTrainPosition functions merges the odometry and the linking coordinate system into one OBU coordinate system. As the both coordinate systems it is derived from, it is an apparent coordinate system working on apparent locations and positions and deviates from the true coordinate system. It is private to and only known by the OBU.

The origin of the OBU coordinate system can be chosen arbitrarily; to set it to 0 at system start up is a suitable choice. The only requirement is, that

> distances originating from the odometry and the linking coordinate system must not be degraded ineligibly. [[4.1]]

Therefore, a best-of method for calculating locations, distances and the train position is used:

> <u>Best-of calculation strategy:</u> [[4.2]]

1. The calculation is performed section by section from a reference location (e. g. the LRBG) up to questioned target location (e. g. BG location). The section borders are the locations of linked BGs in between, where the availability of odometry or linking information changes.
2. If only odometry information is available for a track section, the calculation is based on odometry only by applying [[3.1]].
3. If only linking information is available for a track section, the calculation is based on linking information only by applying [[2.3]].
4. If both – odometry and linking information – is available for a track section, the location / distance / position is calculated based on both. Both calculations lead to two different, regularly overlapping uncertainty ranges. The intersection of both is the resulting uncertainty range for the questioned locality. Its nominal value is adjusted to the center of the range.

5. The section lengths are then accumulated from the appropriate reference location (e. g. the LRBG) up to questioned target location (e. g. BG location) according to [[2.1]].

The orientation of the coordinate system is defined by current train orientation.

The train starts at location 0 at system start up.

The OBU coordinate system is preserved as long as the train is in operation; a reset of the coordinate system is permitted only when the OBU is restarted or all location and position information can be deleted.

5. CALCULATION ALGORITHMS

This chapter describes the algorithm in detail, how the CalculateTrainPosition function computes the locations of linked and unlinked BGs and the train position.

Note:

- All calculation results are related to the OBU coordinate system (chapt. 4.2.2.3).
- Without further notice, all variables used in this chapter uniquely comprise uncertainties and are represented by a set of a nominal value with a minimum (typical negative) and a maximum inaccuracy deviation as stated in chapt. 2. This applies even to linking distances that are exact; in this case, the min/max deviations are set 0. All calculations are based on [[2.1]], [[2.2]], [[2.3]], for the nominal values and the uncertainties.

$$\overline{variable} = nominalValue_{\Delta min}^{\Delta max} \qquad [[\ 5.1\ ]]$$

- Calculation with these variables means calculation with ranges. The nominal value is regularly set into the middle of the range, but – contrary to the range – irrelevant for safe train position and safe distance calculation.
- A bit different handled are distance calculations based on pure odometry values. Here, distances measured by odometry are calculated and converted via [[3.1]] into distances according to [[5.1]] and subsequently processed with [[2.1]], [[2.2]], [[2.3]].

$$\overline{odoValue} = odoNominalValue_{omin}^{omax} \qquad [[\ 5.2\ ]]$$

- The OBU stores the odometry values of all BG location as an “odometry stamp”. This allows distance calculations based on the odometry whenever needed later on:

$$\overline{ododistance} = nominalValue_{\Delta min}^{\Delta max} = \left(\overline{odo_2} - \overline{odo_2} \right) \qquad [[\ 5.3\ ]]$$
$$= \left(odoNominal_2 - odoNominal_1 \right)_{o2min-\,o1min}^{o2max\,-\,o1max}$$

5.1 THE TRAIN POSITION AT OBU SYSTEM START

At start of the OBU system, the actual train position in the OBU coordinate system is arbirtrarily set to 0:

$$\overline{pos_{train}(@systemStart)} = 0_{-0}^{+0} \qquad [[\ 5.4\ ]]$$

5.2 THE TRAIN PASSES A LINKED BALISE GROUP

When the train passes a linked BG, the locations of all BGs are recalculated by performing the following steps subsequently:

1. Calculation of the passed BGs location
2. (Re-)calculation of all announced linked BG locations ahead
3. Recalculation of all linked BG locations with linking information available astern
4. Recalculation of all unlinked and linked without linking information BG locations

This will be elaborated in the following subchapters.

5.2.1 Calculation of the passed BG location

When the train passes a linked BG, its location is calculated with reference to the previously passed linked BG:

$$\overline{loc_{passedBG}} = \overline{loc_{prevPassedLinkedBG}} + intersectionOf\left(\overline{ododistance_{BG-prevBG}} \parallel \overline{linkingdistance_{BG-prevBG}}\right) \quad [[\ 5.5\ ]]$$

Since both BGs must have been passed, the odometry information is available anyway, the linking information only, if the currently passed BG was announced via linking. The $intersectionOf$ function is implemented according to [[4.2]].

We know, that the passed linked BG has been mounted on the track with an inaccuracy range given

- by Q_LOCACC as part of the linking information,
- by Q_NVLOCACC, if there is no linking information available, or
- a default value, if there is no Q_NVLOCACC defined by national values.

Let

$$\overline{qLocAcc} = chosenInaccuracy\left(\overline{Q_{LOCACC}} \parallel \overline{Q_{NVLOCACC}} \parallel \overline{Q_{DefaultLOCACC}}\right) \quad [[\ 5.6\ ]]$$

be the inaccuracy chosen from these options.

The balise detection between the balise and the train is affected by antenna affects and leads to a center detection inaccuracy:

$$\overline{centerDetectionInacc} \quad [[\ 5.7\ ]]$$

Because the train is not able to differentiate between BG mounting and balise detection antenna uncertainties, it will find the BG within a range given by the $qLocAcc$ and $centerDetectionInacc$. We can set the just passed BG within this inaccuracy window around the nominal location from [[5.5]]. This is feasible concerning ([[4.1]]), if the locations of all BGs ahead and astern are recalculated relative to the passed BG location afterwards. With $nominalLocation(\overline{loc_{passedBG\,[[\,5.5\,]]}})$ as the nominal value of $\overline{loc_{passedBG}}$ in [[5.5]], the location of the just passed BG is

$$\overline{loc_{passedBG}} = nominalLocation(\overline{loc_{passedBG\,[[\,5.5\,]]}}) + \overline{qLocAcc_{passedBG}} + \overline{centerDetectionInacc_{passedBG}} \quad [[\ 5.8\ ]]$$

5.2.2 (Re-)calculation of all announced linked BG locations ahead

Based upon [[5.8]], the locations of all announced BGs ahead are recalculated. Since announcements stand synonym for linking, the distance from $\overline{loc_{passedBG}}$ is determined via the linking distances according to [[2.3]]. While the linking distances given by linking information are without any inaccuracies by definition, the first and the last BG in a chain of linking distances is mounted on the track not exactly where intended ($\overline{qLocAcc}$) and not detected by the trains balise antenna exactly where it is mounted ($\overline{centerDetectionInacc}$). The location of a BG "n" in front of the train is then

$$\overline{loc_{BG(n)}} = \overline{loc_{passedBG}} + \sum_{i=passedBG}^{n} \left(\overline{d_{link}(i)} \right) + \overline{qLocAcc_{BG(n)}} + \overline{centerDetectionInacc_{BG(n)}} \quad \text{[[5.9]]}$$

The implementation of the CalculateTrainPosition function starts with the next BG ahead of the passed BG and subsequently iterates in forward direction through all announced BGs. The sub-term $\overline{loc_{passedBG}} + \sum_{i=passedBG}^{n}\left(\overline{d_{link}(i)} \right)$ is reused in the iteration due to performance reasons and accumulates the linking distances and the location of the passed BG:

$$\overline{loc_{BG(n)}} = \left(\overline{loc_{passedBG}} + \sum_{i=passedBG}^{n-1} \left(\overline{d_{link}(i)} \right) \right) + \left(\overline{d_{link}(n)} \right) + \overline{qLocAcc_{BG(n)}} + \overline{centerDetectionInacc_{BG(n)}} \quad \text{[[5.10]]}$$

5.2.3 Recalculation of all linked BG locations with linking information available astern

Based upon [[5.8]], the locations of all linked BGs with existing linking information astern are recalculated similar to chapt. 5.2.2.

The implementation of the CalculateTrainPosition function starts with the previous BG astern of the passed BG and subsequently iterates in backward direction through all those BGs. Since [[2.3]] is independent from forward or backward direction, the linking distances must be taken with negative prefix or subtracted while preserving the calculation comparable to [[5.10]]:

$$\overline{loc_{BG(k)}} = \left(\overline{loc_{passedBG}} - \sum_{i=k+2}^{passedBG} \left(\overline{d_{link}(i)} \right) \right) - \left(\overline{d_{link}(k+1)} \right) + \overline{qLocAcc_{BG(k)}} + \overline{centerDetectionInacc_{BG(k)}} \quad \text{[[5.11]]}$$

The sub-term in parenthesis is reused in the iteration due to performance reasons and accumulates the linking distances and the location of the passed BG.

For all BGs astern, odometry information is available in addition to the linking distances. The location is calculated referring to odometry data to

$$\overline{loc_{BG(k),odo}} = \overline{loc_{BG(k+1)}} - \overline{d_{BG(k)\rightarrow BG(k+1),odo}} \quad \text{[[5.12]]}$$

Herein is $\overline{d_{BG(k)\rightarrow BG(k+1),odo}}$ the distance from BG_k to BG_{k+1} measured by odometry and calculated according to [[5.3]].

Both sources are combined for the resulting location of the astern linked BGs analogue to [[5.5]]:

$$\overline{loc_{BG(k)}} = intersectionOf\ (\overline{loc_{BG(k),linking}} \parallel \overline{loc_{BG(k),odo}}\ ) \qquad [[\ 5.13\ ]]$$

In the case of section-wise linking holes, the location calculation for the range of the linking hole is based on odometry only.

5.2.4 Recalculation of unlinked BGs and linked BGs without linking information

Unlinked BGs and linked BGs without linking information are not related to other linked BG via linking distances, but only via odometry measurement. They must have been passed by the train previously and are located astern of the currently passed BG ([[5.2]]).

After processing the steps in chapt. 5.2.1, 5.2.2 and 5.2.3, the locations of all BGs without linking information are recalculated. For simplicity, these unlinked and linked BGs without linking information are designated with “unlinked” in the following equations, the BGs with linking information as “linked”.

The locations of those “unlinked” BGs are related to the next “linked” BG astern and next “linked” BG ahead.

The location of an “unlinked” BG with reference to the previous “linked” BG astern is

$$\overline{loc_{\text{"unlinked"}BG(astern)}} = \overline{loc_{\text{"}linked\text{"}BG,astern}} + \overline{d_{(\text{"linked"}BG,astern\rightarrow\text{"}unlinked\text{"}BG),odo}} \qquad [[\ 5.14\ ]]$$

The location of the same “unlinked” BG with reference to the next “linked” BG ahead is

$$\overline{loc_{\text{"unlinked"}BG,(ahead)}} = \overline{loc_{\text{"}linked\text{"}BG,ahead}} - \overline{d_{(\text{"unlinked"}BG\rightarrow\text{linked}BG,ahead),odo}} \qquad [[\ 5.15\ ]]$$

The best known location of the “unlinked” is then

$$\overline{loc_{\text{"}unlinked\text{"}BG}} = intersectionOf\ (\overline{loc_{\text{"unlinked"}BG(astern)}} \parallel \overline{loc_{\text{"}unlinked\text{"}BG(ahead)}}\ ) \qquad [[\ 5.16\ ]]$$

5.3 TRAIN POSITION

The current train position is determined with reference to the location of the most recently passed BG ([[5.8]]) by adding the distance measured by odometry since then:

$$\overline{pos_{\text{train}}} = \overline{loc_{passedBG}} + \overline{d_{(\text{passed}BG\rightarrow currentTrainpos),odo}} \qquad [[\ 5.17\ ]]$$