Working Group 3

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Author : | | Christian GIRAUD | | Date : | 20/05/2014 | | | | | |
|  | | | | | | | | | | |
| **Subject :** | | | | | | | Document Review : | | 🗹 | |
| **OpenETCS WP3 Task Force** | | | | | | | Design Review : | | ❒ | |
| WG 3 Architecture - provisional - | | | | | | | Other : | |  | |
|  | |  | | | | | | | | |
|  |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **Company / Department** |
| Valerio Raimondi |  | Alstom |
| Stephane Besure |  | Alstom |
| Nicolas Boverie |  | Alstom |
| Christian Giraud |  | Alstom |
| Yoann Guyot |  | Cetic |
| Baseliyos Jacob |  | DB |
| Bernd Hekele |  | DB |
| Niklas Schaffrath |  | Siemens |
| Uwe Steinke |  | Siemens |
| Jos Holtzer |  | NS |
| Jan Welvaarts |  | L’loyds Register Rail |
| Sylvain Baro |  | SNCF |
| Marielle Petit-Doche |  | Systerel |
| Jan Welte |  | TU Braunschweig |
|  |  |  |
|  |  |  |

**Distribution to:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **Company / Department** |
|  |  | Alstom TIS |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Content

1. Objectives and strategy 4

2. Context Of Open-ETCS 5

2.1 references 5

2.2 High Level Architecture 5

2.3 Concept Of Open ETCS 6

2.3.1 Unisig Specification 7

2.3.2 Signalling Application Specification 7

2.3.3 Formal Specification 7

2.3.4 Model 7

2.3.5 Kernel Code 7

2.3.6 Virtual Machine 7

2.3.7 Host Machine 7

2.3.8 Real Time Machine 7

2.3.9 Validation 7

3. Data for Safe Movement 8

3.1 Linking 8

3.2 Basic Track Description 8

3.2.1 Movement Authority 8

3.2.2 Standard Speed Profile 8

3.2.3 Gradients Profile 9

3.3 Optional Track Description 9

3.4 System Data 10

3.4.1 Fixed Values 10

3.4.2 National Values 10

3.4.3 Train Data 10

3.4.4 Additional Data 10

3.4.5 Date and Time 11

3.4.5.1 UTC 11

3.4.6 Data view 11

4. Processes Over Data 12

4.1 Data Preliminary Operation 12

4.2 Storage of Data 13

4.3 Update of Data 14

4.4 Train Positioning and linking 14

4.5 Emergency Stop 15

4.6 Speed and Distance Monitoring 15

4.6.1 Computation of Deceleration and Brake Build up Time 15

4.6.2 MRSP computation 15

4.6.3 Computation of Target and Curves 15

4.6.4 Supervision Limits Computation 15

4.6.4.1 Ceiling Speed Monitoring 15

4.6.4.2 Target Speed Monitoring 15

4.6.4.3 Release Speed Monitoring 15

4.7 Protection against undesirable movement 15

4.7.1 Roll Away 15

4.7.2 Reverse 15

4.7.3 Standstill 15

4.8 Mode Management 15

4.9 Brake command handling 15

4.10 Special Function 15

4.11 Version Management 15

5. SysML First Draft 16

5.1 First Level IBD 16

5.2 Second Level 17

5.3 To Position Train 19

5.4 To Achieve Processes 21

6. Speed Supervision limits ( exemple) 22

6.1 references 22

6.2 overview 22

6.3 Context of Speed Limit Supervision 22

6.4 Example of Balise Group in level 1 24

6.5 EB Supervision Curve 25

6.6 EB Supervision Computation 25

6.7 EB Supervision Excel File 27

6.8 EB Supervision per Quantum 30

# Objectives and strategy

The main objective is to propose a technical plan for Open-ETCS in relation with UNISIG specification.

The aim of Open-ETCS is to insure that “Subset\_026 version\_3.3.0” shall be implemented and interpreted following the same understanding by every supplier of ETCS devices.

Given the difficulties, it has been defined that consists in working on application software of EVC. Trackside application software is anticipated as enough transparent from EVC.

Three phases are involved :

* **First Phase** : The Subset-026 chapter3 being basically a system requirement definition, it needs to be analysed and structured through SysML methodology or equivalent. That permits to build a breakdown structure as follows :
  + Basic Software and hardware, is proprietary software of each ETCS supplier,
  + Application Software (so-called “kernel software”),
    - Balise Data storage and update,
    - Positioning,
    - Speed Control Monitoring,
    - etc..
  + Application Program Interface : Alstom solution is acting as reference.
* **Second Phase** : The Breakdown Structure of Application Software is refined up to reach a level of detail that permits to proceed to a formal (non ambiguous) specification and encoding.
  + This phase leads to include most of the modules within the Open-ETCS library.
  + On a first step, it has been agreed to work on “Balise Data Storage” and “Positionning”.
  + It may appear the need to define a database structure when study of other functions.
  + At the end of this phase, “Kernel Software” is ready for integration on “Host Machine” or on “Real Time Machine”.
* **Third Phase** : In order to consolidate the specification, a model is developed for the most complex functions that permits to get a reference for each signalling implementation. This is the concept of Virtual Machine.

# Context Of Open-ETCS

## references

UNISIG Subset\_026 version\_3.3.0

Chapter 3 : ERTMS / ETCS Principes

Chapter 5 : ERTMS / ETCS Procedures

Chapter 7 : ERTMS / ETCS Language

## High Level Architecture

The following Block Diagram Definition (BDD) gives an abstract of global ETCS from the on-board (EVC) point of view :

Balise

Odometry

DMI

TIU

CORE

Radio

(RBC)

EVC

**Block Diagram Definition**

EVC is composed of :

* Balise reception
* Odometry
* DMI : Display Man Interface
* TIU : Train Interface Unit
* CORE : Computer
* Radio sub-system (GSM) and RBC (Radio Block Center)
* Euroloop (optional)
* Recorder Unit

Communication are insured through a triple Bus (CAN bus for instance).

Specification of communication are defined by API (Application Program Interface) from functional point of view.

## Concept Of Open ETCS

Open ETCS aims to provide a link between the UNISIG Specification (Mainly Subset\_026) and any implementation of On-board provider.

For reaching this challenge, a double activity diagram is proposed as follows :

Unisig Specification V3.3.0

Formal Specification

(SysML & Software Spec)

Model

Kernel Code

Host Machine

(Soft ADA C++)

Real Time Machine

(hard +soft + simu)

Virtual Machine

( Excel )

Signalling Implementation Specification

Validation

Hardware + basic soft + simulator

**Block “Actigram”**

This actigram shows the 3 branches of activity that are involved in order to validate any Signaling Application.

### Unisig Specification

The main document of reference is the subset 026 in version V3.3.0.

All chapters are applicable and all subset that are referenced in it.

### Signalling Application Specification

The Signalling Application Specification should include :

* track layout with balise group (BG) and signals,
* definition of each BG coverage, in normal and reverse direction if needing,
* definition of mandatory data for each BG,
* definition of optional data for each BG,
* definition of system data.

### Formal Specification

The Formal Specification shall include two steps :

* SysML defines the architecture & Software Spec,
* Formal language specifies the software or can be used as retro engineering.

### Model

The use of “Excel” is a solution to study braking curves of each signal and for a run of a virtual train.

Other solution ?

### Kernel Code

Programing language : ADA vs C++.

Formal language : B language.

Bridge to pass from B to ADA.

### Virtual Machine

Excel, solution linked to Model.

### Host Machine

Solution to compare result with virtual machine.

### Real Time Machine

Idem Host.

### Validation

Automatic validation is interesting for checking in case of software updating.

# Data for Safe Movement

## Linking

Linking data are playing a major role in management of track description, that is :

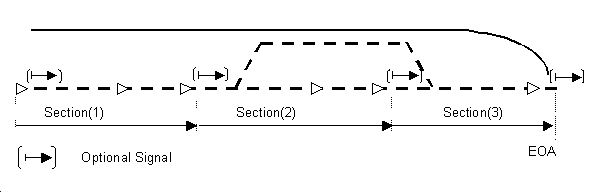
* To know in advance which BG will be met from any current BG (several BG can be defined),
* To check the train position regarding the window related to the last BG position,
* To check the train position regarding the route provided by the previous BG,
* To up-date the track description provided by the previous BG,
* To reduce the train position error to the position inaccuracy of the last BG,
* To permit or not an immediate reaction in case of BG missing.

## Basic Track Description

“Movement Authority”, “Standard Speed Profile” and “Gradient Profile” are the basic element for a Track Description.

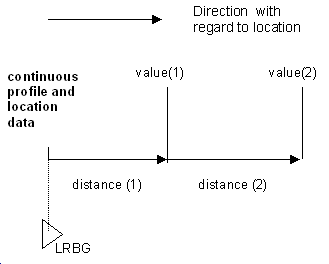
### Movement Authority

* Section : the MA is composed of several section with optional timer to enter,
* LOA : is the length of MA starting from LRBG, and is terminated by EOA,
* DP : Danger Point is beyond the EOA and permits to approach EOA,
* OL : Overlap is beyond the EOA and requires a locking from some other routes.
* Veoa : is the limit speed related to EOA.



### Standard Speed Profile

SSP is a continuous profile defined by a set of : {distance, value}.



### Gradients Profile

SSP is a continuous profile defined by a set of : {distance, value}.

## Optional Track Description

See Subset-026, Chapter 3.11.

* Axle Speed Profile
* Temporary Speed Restriction
* Mode Profile and Related Speed Restriction
* Train Related Speed Restriction
* LX Speed Restriction
* Override Function Related Speed Restriction
* Speed Restriction to ensure Permitted Braking Distance

See Subset-026, Chapter 3.12.

* Track Conditions
* Route Suitability
* Text Transmission

## System Data

See subset-026 chap3.18.

### Fixed Values

Data are defined in basic software before a compilation.

### National Values

Data are transmitted with “Country Number”.

### Train Data

Data must be defined at standstill, before a mission :

1. Train category(ies)
2. Train length
3. Traction / brake parameters
4. Maximum train speed
5. Loading gauge
6. Axle load category
7. Traction system(s) accepted by the engine
8. Train fitted with airtight system
9. List of National Systems available on-board
10. Intentionally deleted
11. Axle number

### Additional Data

Data are provided through DMI :

1. Driver\_ Id
2. ETCS level
3. Radio Network identification
4. RBC Identification
5. Telephone Number
6. ETCS Identity
7. Train Running Number
8. Adhesion factor

### Date and Time

#### UTC

### Data view

TBD.

# Processes Over Data

## Data Preliminary Operation

In level 1, data are received by telegrams that are transmitted by balises which are grouped by BG of 1 up to 8 balises.

Each balise telegram is composed of 1023 bits, only 830 bits are used for data, the 193 other bits are used for safety and for finding the beginning of the telegram.

In case of simple application, telegrams of 341 bits could be involved.

The 830 bits of each balise are composed of :

* one header of 50 bits that are used to manage the data of BG and is composed at least of :
  + one balise group identity,
  + the number of balises included in the group,
  + the ranking of each balise,
  + one redundancy indicator.
* “n” packets of variable size to provide mandatory and optional data, each of them including :
  + one packet identity,
  + the length of each packet,
  + the orientation of validity (normal or reverse).
* one “end” packet.

The direction of orientation is given by the order of the balise included in BG :

* Increasing order is “Norminal” orientation,
* Decreasing order is “Reverse” orientation.

Basically, a double track orientation is designed as follows :

Track 1

Nominal

Reverse

LRBG for packets in nominal orientation, Track 1

Track 2

Reverse

Nominal

LRBG for packets in nominal

orientation, Track 2

Reverse

Nominal

LRBG for packets in reverse

orientation, Track 1

**LRBG Orientation**

## Storage of Data

A Database Architecture needs to be defined in order to classify and to compute data.

All packets must be stored in the database with the following consideration :

* All packets available in nominal orientation (Q\_DIR = nominal) need to be ordered if :
  + Train orientation is nominal (Q\_DIRLRBG = nominal) when passing over LRBG or,
  + If radio track-to-train message refers to a LRBG whose directionality is nominal, and
  + If train is moving ahead (Q\_DIRTRAIN = nominal).
* All packets available in reverse orientation (Q\_DIR = reverse) need to be ordered if :
  + Train orientation is reverse (Q\_DIRLRBG = reverse) when passing over LRBG or,
  + If radio track-to-train message refers to a LRBG whose directionality is reverse, and
  + If train is moving ahead (Q\_DIRTRAIN = nominal).
* If train is moving in reverse (Q\_DIRTRAIN = reverse), no data is stored.
* A maximum of 8 balises must be stored in memory.
* Linking is involved to insure continuity between LRBG and to permit switching between nominal and reverse orientation, in case of double direction of traffic.
* In case of repeat, packet needs to be repeated as much as necessary.
* A Data Ranking is necessary before computation of other functions (MRSP, Integration of brake energy and gravity, etc…).
* A Completeness Checking consists in verifying that no mandatory data is missing : gradient and SSP are defined up to EOA.

## Update of Data

Purge of Data : is automatic by saving the last 8 LRBG.

Save of Data : same as purge, thanks to saving the last 8 LRBG, taking in consideration that a very long train may need 8 LRBG when crossing over a complex station.

## Train Positioning and linking

Definition of related variables:

* Q\_DIR :
  + **qualifier of direction**, attached to most of track-to-train data packet ,
  + must be in compliance with Q\_DIRLRBG to take into account the data,
  + does not exist for train-to-track data packet,
* Q\_DIRLRBG : **qualifier of orientation**,
  + is defined by the order in which balise group is read :
    - 1, 2, 3.. is nominal,
    - 3, 2, 1 is reverse.
  + decoded when overpassing the BG providing the data packets (level 1),
  + or qualifier decoded when taking into account LRBG (level 2/3),
* Q\_DIRTRAIN : **qualifier of running**,
  + determined on-board by cabine number and odometry counter,
  + for instance, nominal is :
    - Front End is cabine1 + counter increasing,
    - or Front End is cabine 2 + counter decreasing,
  + and reverse is :
    - Front End is cabine1 + counter decreasing,
    - or Front End is cabine 2 + counter increasing,
* Q\_DLRBG : qualifier of position,
  + indicate on which side of LRBG is the Front End :
    - should be equal to DIR\_LRBG if no reverse running,
    - but can be changed after a reverse running.
* Q\_LINKORIENTATION : **advance qualifier of orientation,**
  + is given by linking packet.

See IBD of Process “To Position Train”.

## Emergency Stop

See subset-026 chap3.10. Only in level 2.

## Speed and Distance Monitoring

### Computation of Deceleration and Brake Build up Time

See subset-026 chap3.13.6.

### MRSP computation

See subset-026 chap3.13.7.

### Computation of Target and Curves

See Excel Model on EB curves.

### Supervision Limits Computation

#### Ceiling Speed Monitoring

#### Target Speed Monitoring

#### Release Speed Monitoring

## Protection against undesirable movement

### Roll Away

### Reverse

### Standstill

## Mode Management

See State Machine Diagram of Intercity project.

## Brake command handling

## Special Function

## Version Management

# SysML First Draft

## First Level IBD

Hereafter is a SysML “IBD”first draft of EVC :

Messages

Command / Control

To Position Train

Balise Content

Odometry

Train Data

**EVC**

Fixed Values

To Store System Data

To check and Store Balise Content

To Store Track Description in Database

National Values

To Achieve Processes

DMI Data

To Manage DMI

Radio Message

To check and Store Radio Message

TIU Data

To Manage TIU

Additional Data (changeable on-board)

Orders / Display

Train Position & Speed

Synchro Odo / Balise

**Database**

Linking and

re-positionning

System Data

Packets

**IBD “EVC” of First Level**

## Second Level

Both boxes ***“To Check and Store Balise Content”*** and ***“To Store Track Description in Database”*** are using the schematic database matrix as follows :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BG\_n | Position | Linking | orientation | inaccuracy | other |
|  |  |  |  |  |  |
| TrackCond | Pos. | Param. | Asafe | 9,81\*grd | MRSP |
| Others | “ | “ | “ | “ | “ |
| Grade | “ | Value | “ | “ | “ |
| SSP | “ | “ | “ | “ | “ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BG\_n+1 | Position | Linking | orientation | inaccuracy | other |
|  |  |  |  |  |  |
| SSP | Pos. | Value | Asafe | 9,81\*grd | MRSP |
| Others | “ | “ | “ | “ | “ |
| Grade | “ | “ | “ | “ | “ |
| SSP | “ | “ | “ | “ | “ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BG\_n+2 | Position | Linking | orientation | inaccuracy | other |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EOA | Pos. | Veoa | Asafe | 9,81\*grd | MRSP |
| DP | “ | 0 | “ | “ | “ |
| OL | “ | 0 | “ | “ | “ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BG\_n+3 | Position | Linking | orientation | inaccuracy | other |
|  |  |  |  |  |  |
| SSP | Pos. | Value | Asafe | 9,81\*grd | MRSP |
| Others | “ | “ | “ | “ | “ |
| Grade | “ | “ | “ | “ | “ |
| SSP | “ | “ | “ | “ | “ |

xxx

-1%

60 km/h

80 km/h

2%

45 km/h

**Track Layout**

**Type Position Value Asafe Grd Mrsp**

60 km/h

-0,5%

100 km/h

LOA

LRBG

**Database of “EVC”**

Comments :

* Assuming the hereunder matrix, the convention is :
  + Direction of travel is top to down, whatever is the direction of travel or orientation,
  + Dimension is enough to keep in memory :
    - * the data related to the last 8 passed balises,
      * and the data related to 8 linked balises as anticipation,
      * only packets in correct orientation are stored.
  + The 3 first columns are filled by packet data :
    - * Type is packet identity,
      * Position is distance from related LRBG,
      * Value is data related to the type,
      * In case of iteration, as much lines as necessary are filled up,
      * Line are classified by increasing position in relation with orientation.
  + The 3 other columns are computed once classification is achieved :
    - * Asafe : emergency brake deceleration, can be a function of speed (which speed ?) and other data (adhesion factor),
      * Grd : gravitation acceleration or deceleration, depend on related packet (gradient),
      * Mrsp : depends on numerous packet.
  + Other columns can be added for energy computation.
  + Assuming that message received through BG\_n is acting as LRBG :
    - Linking packet is covering 3 BG (BG\_n+1, BG\_n+2, BG\_n+3),
    - MA packet defines an EOA close to BG\_n+2
    - BG\_n+3 is in reverse orientation, meaning a possible change of track,
    - SSP and Gradient packets are enough to cover the track over the EOA (consistency criteria),
    - EOA indicates that route signal is at stop.
    - Assuming later on, when overpassing the BG\_n+1 acting as new LRBG :
  + Linking is verified by Identity and Position regarding the permitted window,

## To Position Train

To see what is done by other partners…

A proposed draft is given hereafter :

To Position Train

Train Position & Speed

Synchro Odo / Balise

Linking and

re-positionning

**Linking (input) :**

Id of next BG

Distance to next BG

Orientation of next BG

(Q\_LINKORIENTATION)

Inaccuracy of next BG

**Positioning (output) :**

Id of last BG (NID\_LRBG)

Distance since LRBG (D\_LRBG)

Orientation of LR BG (Q\_DIRLRBG)

Train Position related LRBG (Q\_DLRBG)

L\_DOUBTOVER

L\_DOUBTUNDER

V\_TRAIN

Train Running related LRBG

(Q\_DIRTRAIN)

To Up-date when overpassing linked balise

To Up-date when overpassing re-positioning balise

To Up-date at every real time cycle

To Position when overpassing balise

**Re-Positioning (input) :**

Id of BG

Distance to next BG

Orientation of next BG

**Odometry (Input) :**

Counter Increasing

Counter decreasing

Counter standstill

Counter value

Speed (V\_TRAIN)

**Positioning (input) :**

Id of last BG

Distance since last BG

Orientation of last BG

**“To Position Train”**

Comments :

* To Up-date at every **real time cycle** :
  + Input are:
    - Counter of odometry including :
      * Value, Increasing, Decreasing, Standstill,
      * Speed.
  + Output are up-dated by :
    - D\_LRBG = ( Counter – Counter0 ) \* Qualibration
      * Counter is the current counter of odometry,
      * Counter0 is the counter value when over-passing LRBG.
    - NID\_LRBG = unchanged,
    - Q\_DIRLRBG = unchanged,
    - Q\_DLRBG = inverted if D\_LRBG changes of sign,
    - Q\_DIRTRAIN = inverted if D\_LRBG changes of variation sign,
    - L\_DOUBTOVER = LRBG\_Inaccuracy + ( | D\_LRBG | \* 5% ),
    - L\_DOUBTUNDER = LRBG\_Inaccuracy - ( | D\_LRBG | \* 5% ),
    - V\_TRAIN = Speed + 5%
* To Position when overpassing balise :
  + If balise is **“Re-positioning”,** only D\_LRBG is updated by :
    - D\_LRBG = D\_LINK – L\_SECTION
  + If balise is **linked** with previous one, checking with parameters in red must be true :
    - | ( D\_LRBG – L\_DOUBTOVER < D\_LINK ,
    - | ( D\_LRBG + L\_DOUBTUNDER > D\_LINK ),
    - NID\_LRBG = NID\_BG,
    - Q\_DIRLRBG = Q\_LINKORIENTATION.
  + Output are estimated by :
    - Counter0 = Counter,
    - D\_LRBG = 0,
    - NID\_LRBG = new,
    - Q\_DIRLRBG = new,
    - Q\_DLRBG = inverted if D\_LRBG changes of sign,
    - Q\_DIRTRAIN = Q\_DIRLRBG,
    - L\_DOUBTOVER = LRBG\_Inaccuracy,
    - L\_DOUBTUNDER = LRBG\_Inaccuracy ,
    - V\_TRAIN = Speed + 5%

## To Achieve Processes

See SyML IBD draft hereafter.

Command / Control

To Achieve Processes

Orders / Display

Messages

Train Position

& Speed

System Data

MRSP

Computation

Asafe

Computation

Supervision Limits Computation

Target and Curves Computation

**Database**

Mode and Level

Protection

Brake Handling

Version Management

Special Function

Curves & Target

MRSP

Orders / Display

Asafe

SvL

& over speed

& overide

Mode request

& EB request

Mode & Level

TBD

TBD

**“To Achieve EVC Process”**

Each box needs to be analysed through SysML IBD.

# Speed Supervision limits ( exemple)

## references

UNISIG Subset\_026 version\_3.3.0

Chapter 3 : ERTMS / ETCS Principes

Chapter 5 : ERTMS / ETCS Procedures

Chapter 7 : ERTMS / ETCS Language

## overview

[SRS-026-chapter : 3.13.9.3]

Braking to target supervision limits consists in computation of EBD, EBI, SBD and SBI curves. GUI curve is not addressed.

The use of an Excel database is involved in this exercise.

The technics to fill up the database starting from balise contents is not approached.

The structure of the database is defined and Excel capabilities such as classification of data lists following increasing or decreasing abscissa are used.

We shall start from a typical BG of 2 or 3 balises that are crossed by one train in level 1.

This BG will provide basic data packets such as :

1. National values,
2. Movement Authorization,
3. Balise linking,
4. Speed Profile,
5. Gradient Profile,
6. Level Order,
7. Temporary Speed Restriction.
8. etc…

## Context of Speed Limit Supervision

The context is defined with SysML diagram through one BDD (Block Diagram Definition) and several IBD (Internal Block Diagram) that will need to be linked with other parts of the OpenETCS definition.

The Block Diagram Definition should be composed of the 18 functions of the subset-026 chapter 3.

The Speed Limit Supervision is one IBD of BDD which is composed of the following blocks IBD :

Speed Limit

Supervision

**(SRS-026 chap3-13)**

Train Length

National Values

SSP Profile

Gradient Profile

Temporary Speed Restriction

Track Condition Request

Speed & Position & LRBG

LOA, Veoa, DP, OL

Mode Profile

Emergency Brake Curves

Service Brake Curves, S, W, P, I

Overspeeding

Fixed Values

Overiding

Target

Curent Mode

**Block “Speed Limit Supervision”**

Inputs :

* From “Balise Decoding” :
  + National Values,
  + SSP Profile,
  + Gradient Profile,
  + Temporary Speed Restriction,
  + LOA, Veoa, Danger Point, Overlap.
* From “Internal” :
  + Fixed Values,
  + Train Length,
  + Current Mode,
  + Track Condition Request.

Outputs :

* Emergency Brake Curves,
* Service Brake Curves (Service, Warning, Permitted, Indication),
* Overspeeding (cell and target),
* Overiding,
* Target,
* Release Speed.

Speed & Distance

Monitoring

(SRS-026-3 13-10)

MRSP Computation

(SRS-026-3 13-7)

Database

computed

(SRS-026-3

13-8, 13-9)

Database

classified

Database

not classified

Traction & Brake

Models

(SRS-026-3 13-2, 13-3, 13-6)

Fixed Values

National Values

SSP Profiles

Gradient Profile

Temporary Speed Restriction

Track Condition Request

Speed & Position & LRBG & LRBGDIR

LOA, Veoa, DP, OL

Mode Profile

Emergency Brake Curves

Service Brake Curves, ( S, W, P, I )

Overspeeding

Train Length

MRSP

Overiding

Target

Curent Mode

## Example of Balise Group in level 1

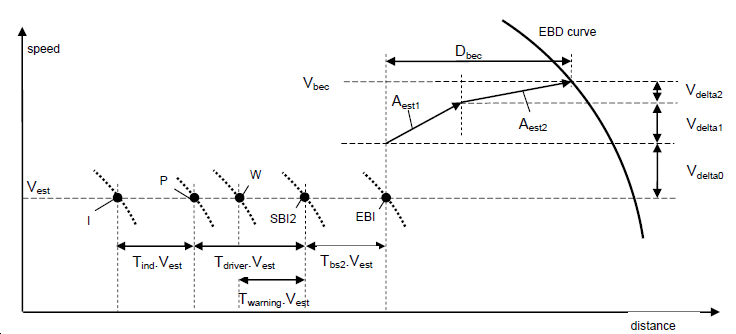
We shall show an example of database configuration for a Balise Group in level 1.

* In first, we shall deploy the database following the singular points that are involved within the BG. We shall see that is not fully meeting the requirement of an EB deceleration variable with the speed.
* In second, we shall deploy the database by quantum of 10 meters.

Definition file of such BG is given within excel file, chapter 5.7.

## EB Supervision Curve

All parameters for braking to target supervision limits for EBD curve are defined within the drawing hereafter :



Braking :

xb, vb, hb

Origin :

,x0, v0, h0

Target :

x1, v1, h1

## EB Supervision Computation

It is defined 3 particular locations :

* Origin :
  + x0 : location where the EB intervenes,
  + v0 : speed at location x0, so-called “Vest”, can be increased by inaccuracy,
  + h0 : hight of gravity centre at x0.
* Braking :
  + xb : location where the EB is active after two phases of transition,
  + vb : speed at location xb, is v0 increased after 2 phases of transition,
  + hb : hight of gravity centre at xb.
* Target :
  + x1 : target location,
  + v1 : target speedn
  + h1 : hight of gravity centre at target.

The next computation will be done through physical mechanic theory in “Energy” by mass unit (m²/s²) between origin and target. To define the EB supervision limits, we put the following equation :

(Brake\_Energy) + (Potential\_Variation\_Energy) > (Kinetic Variation Energy) + (Response\_Correction)

**Cumul\_EB > Cumul\_K**

With the following development :

* + Kinetic Variation Energy = ½ (v0² - v1²)
    - this value concerns the total mass : M \* (1+alfa)
    - alfa can be a function of position “x”.
  + Brake Energy = (x1 – x0) \* Aeb
    - this value concerns the total mass : M \* (1+alfa)
    - Aeb is the average value of the function Asafe(x,v)
    - Asafe(x,v) needs to be integrated between x0 and x1
    - what about v ? (speed at origin x0 or something else )
  + Response\_Correction = ½ [ (Aest1\* T1)² + (Aest2\*T2)²] +(Debc \* Aeb)
    - response is in 2 phases, lasting T1 and T2,
    - the running distance is so-called Debc,
    - brake energy over Debc must be added to kinetic energy
    - power energy (Aest1 and Aest2) must be added too.

* + Potential \_Variation\_Energy = (h1 – h0) \* g / Alpha
    - this value concerns only the mass M,
    - g = 9,81 m/s² must be taken into account,
    - potential energy must be compensated by division with Alpha = 1+alpha,
    - h1 – h0 is the integration of function Grade(x) between x0 and x1.

## EB Supervision Excel File

The use of an Excel database is involved in this exercise in 3 steps :

* + Step 1 : to acquire data from balise content,
  + Step 2 : to classify acquired data into another file,
  + Step 3 : to simul.

Step 1 Excel File :

\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**Position** is the distance of singular point from the reference balise B1,

**Type**  is the nature of singular point,

**DOT** is the Direction of Travel.

**Pos. Inc.** is the incremental position to the next singular point.

**Asafe** is the safe deceleration value of the Emergency Braking, related to increment.

**Grade** is the slope value which permits hight computation, related toincrement.

**Target** is the singular point of MRSP or a DP which can be a potential target.

Step 2 : singular point of database are classified following increasing position.

Step 3 : computation of final parameters.



**EB\_Ener** is the Emergency Brake Incremental Energy.

**G\_Ener** is the Gravitation Incremental Energy.

**Cumul\_EB** is the sum of EB and G energy variation from target DP to current position.

**K\_Ener** is the Kinetic Energy variation from target DP to current position.

**Debc** is the running distance during both transition.

**Corr\_E** is the increased transition energy.

**Cumul\_K** is the sum of Kinetic and Transition Increased Energy.

E\_Maxi is the minimum of both “Cumul”.

SSP\_1 60 km/h is intermediate target and DP is the final target.

## EB Supervision per Quantum

We come back on previous equations :

(Brake\_Energy) + (Potential\_Variation\_Energy) > (Kinetic Variation Energy) + (Response\_Correction)

In order to eliminate the response correction, we shall take in consideration the energy between target ”x1” and current location “xb”. That gives :

(Brake\_Energy\_B) + (Potential\_Variation\_Energy\_B) >= (Kinetic Variation Energy\_B)

and vb <= MRSP(xb)

With the following development :

* + Kinetic Variation Energy B = **½ (vb² - v1²)** ( v1 is target, vb is current v )
  + Brake Energy B = **(x1 – xb) \* Aeb** (integration from xb to x1)
  + Potential \_Variation\_Energy\_B =**(h1 – hb) \* g / Alpha** (integration xb – x1)

The process is to define “vb” from the target location (basically the Danger Point DP) up to the balise location. Then we have at the begining :

* + ½ (vb² - v1²) = (x1 – xb) \* Aeb + (h1 – hb) \* g / Alpha or
  + vb² = v1² + 2 \* [(x1 – xb) \* Aeb + (h1 – hb) \* g / Alpha] and
  + vb <= MRSP(xb)

Starting from DP location (or EOA), to define “x1”,

v1 = 0 or Vsig,

h1 =0

Then to compute vb for each “xb” iteration, with a limitation by MRSP.

When previous MRST is higher, we have a new target.

Then the EBI curve can be computed through Debc and V\_delta computation:

V\_delta = Aest1\*T1 + Aest2\*T2

v0 = vb – V\_delta

Debc = ½ ( Aest1 \* T1² + Aest2 \* T2² ) + v0 \* (T1 + T2)

x0 = xb – Debc

*Variables of speed limit*

Pos. xb is the position of control at speed vb, incremental quantum=10 m. **{ m }**

Type is the type of singular point ( “ \* ”means no type).

DOT is up or down.

MRSP is the Most Restrictive Speed Profile ( compile SSP, TSR, Vmax..). **{ km/h }**

Asafe is safe deceleration value, is a function of x and v0. **{ m/s² }**

Formula : **Asafe(n) = SI((v0 < 60); 0,80; 0,85)**

Grade is slope grade, positive in uphill.

Target is the closest target speed, it can be : MRSP reduction, EOA, DP. **{ km/h }**

EB\_Ener is the Emergency Brake Incremental Energy. **{ m²/s² }**

Formula : **EB\_Ener(n) = Asafe(n) \* Quantum**

G\_Ener is the Grade Incremental Energy. **{ m²/s² }**

Formula : **G\_Ener(n) = Grade \* 9,81 \* Quantum**

Cumul\_EB is cumul “EB\_Ener +G\_Ener”, since smallest target energy. **{ m²/s² }**

Formula : **Cumul\_EB(n) = MIN( (Cumul\_EB(n+1) + EB\_Ener + G\_Ener) ; ½ \* (MRSP/3,6)² )**

v1\_carré is 2 times Kinetic energy of smallest target. **{ m²/s² }**

Formula : **v1\_carré = ½ Target²**

vb is max speed to respect smallest target without response time. **{ km/h }**

Formula **: vb = ( (RACINE( 2 \* Cumul\_EB )) \* 3,6 )**

v\_delta is speed variation during the transition phase T1+ T2. **{ km/h }**

Formula : **v\_delta = ((Aest1\*T1) + (Aest2\*T2)) \* 3,6 )**

v0 is max speed to respect smallest target with response time. **{ km/h }**

Formula : **v0 = vb – v\_delta**

Debc is running distance during response time. **{ m }**

Formula : d = ½ ( (Aest1\*T1²) + (Aest2\*T2²) ) + ( Aest1\* T1 \* T2 )

Formula : Debc = d + (v0 \* (T1+T2) / 3,6)

x0 is max position to respect smallest target with response time. **{ m }**

Formula : x0 = xb - Debc

**Track configuration**

Simple Track Configuration is:

Balise : B1, B2, B3.

SSP : 160 km/h at position 0, 60km/h at position 160.

Grade changes at position 0, 200, 500, 610, 710, 820, 920, 1030, 1040, 1050,1080.

EOA at position 1060 with 30 km/h.

DP at position 1070.

**Train Simulation (for model only)**

Pos.xb is geographical milestone quantum per quantum of 10m = Pos(n)

Pos(0) is initiale position = 0 => Pos(n) = Pos(n-1) + 10

Pos(n+1) = Pos( n) +10

Aest is train acceleration => Aest( n ) = f ( v, n )

Vest is train speed at Pos(n) =>

Vest(n+1) = RACINE( (20 \* Aest( n) ) + Vest(n)² )

Tdelta is time to reach Pos(n+1) and Vest(n+1)

Tdelta = (Vest(n+1) – Vest(n) ) / Aest(n)

Tdelta = { RACINE [ (20 \* Aest( n) ) + Vest(n)² ] – Vest(n) } / Aest(n)

END OF DOCUMENT