Received 13 December 2022, accepted 5 January 2023, date of publication 13 January 2023, date of current version 19 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3236784

RESEARCH ARTICLE

A Novel Approach for Reducing Train Localization Errors by Inertial Measurements

MAURO MARINONI[1], (Member, IEEE), PIERLUIGI AMATO[1,2], CARMELO DI FRANCO[1], SALVATORE SABINA[1,3], AND GIORGIO BUTTAZZO[1], (Fellow, IEEE)

[1]TeCIP Institute, Scuola Superiore Sant'Anna, 56127 Pisa, Italy
[2]Scuola di Ingegneria, Università di Pisa, 56122 Pisa, Italy
[3]Hitachi Rail STS, 16151 Genoa, Italy

Corresponding author: Mauro Marinoni (mauro.marinoni@santannapisa.it)

ABSTRACT The effectiveness of current railway signaling systems heavily depends on the accuracy in the localization of trains on the track, which is currently enforced using static markers mounted along the line, called balises. However, balises are expensive, sparse, prone to tampering, and subject to maintenance, hence they need to be integrated by additional onboard sensors. Also, the harsh operational environment and the strict safety regulation do not allow a straightforward adoption of solutions from other application domains, like the GNSS in automotive systems. Hence, finding novel cost-effective solutions to improve the localization accuracy is an emerging research topic in modern railway systems. Low-cost inertial sensors provide the required level of availability and thus represent a viable approach to integrate physical balises, but they need to be coupled with a rigorous methodology for integrating the inertial data with the track geometry features extracted from a digital railway map. This paper presents a novel methodology for exploiting inertial data in train localization. It is based on a two-phase approach: in an offline phase, track data from the digital map are analyzed to extract a number of features (curves, switches, slopes, etc.) that are stored in the map as position markers; at run time, such markers are detected by analyzing the inertial data and matched with those stored in the map. Such a marker matching allows reducing odometry errors by a significant extent. Preliminary experiments on synthetic and real data show the effectiveness of the proposed approach in enhancing train odometry.

INDEX TERMS Train localisation, inertial navigation, map matching, curvature reconstruction, railway odometry.

I. INTRODUCTION

An accurate train localization is of paramount importance for ensuring safety and efficiency of the railway network. Today, the absolute position of a train along the track is determined by using radio-frequency transponders, called balises, installed along the track at predefined locations with a given accuracy (for instance, in Italy the balise location accuracy has been set to be no higher than ±5 meters) [1]. Between consecutive balises, the position of the train is estimated by an odometry system, which typically integrates the speed measured by a set of encoders installed on the train wheels. Due to the integration operation, the relative position (with respect to the last passed balise) estimated by odometry can be affected by a significant drift. Such a drift depends on several reasons, among which wheel wearing and slip/slide phenomena caused by different environmental conditions; for example, the adhesion factor between a wheel and the rail depends on the quantity and type of debris, as snow, ice, and various slippery materials.

To account for the drift introduced by the odometry, the European Train Control System (ETCS) performance requirements for interoperability [2] state that, for every measured distance s from the last passed balise, the accuracy on the measured distance shall be better or equal to $\pm(5\text{m}+5\%s)$. Figure 1 shows an example in which a train with a distance s from the last passed balise accumulates a worst-case relative position error $\Delta = (5 + 0.05s)$, meaning that its length L

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez.

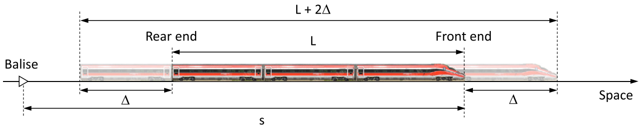


FIGURE 1. **Effect of the position error on the train length.**

has to be increased by 2Δ, considering both the effect on the front end and the one on the rear end and an equal contribution from under-reading and over-reading errors in the measured distance *s*.

The accumulated position error has the effect of increasing the length of track segment occupied by the train, thus reducing the capacity of the railway system in terms of number of trains that can occupy the line simultaneously, that is, the efficiency of the whole signalling system. It is worth observing that increasing the number of balises to reduce the position error would significantly affect the overall cost of the line, also considering that balises are typically installed in groups for safety and availability reasons and are subject to frequent maintenance, due to wearing or tampering.

For the reasons discussed above, in the recent years, especially with the need of improving the capacity of railway lines, accurate and precise train position functions became an emerging research topic, and many research projects [3], [4] were launched to investigate enhancements of the odometry function.

However, most of the solutions proposed in the literature focused on improving the localization accuracy by integrating wheel odometry with other sensors, as Global Navigation Satellite System (GNSS) [5] and Inertial Measuring Unit (IMU). However, both of these sensors have their shortcomings, since the GNSS signal could be absent or not reliable due to multipath and non-line-of-sight phenomena and radio frequency interferences, while inertial navigation, which derives the position by integrating inertial data, is affected by noise and drift.

Differently from previous works, the solution proposed in this paper exploits inertial data not for directly computing the odometry, but for detecting specific features on the track (e.g., curvatures, slopes, and switches) to be matched with the corresponding ones stored on a digital map. As a consequence, the reconstructed positions are not affected by an error drift due to integration, making this method suitable for being used as an absolute input reference in localization solutions based on data fusion. In particular, when a feature is matched with the corresponding one present in the digital map, the confidence interval of the odometry error can be reset with a mechanism logically equivalent to the one performed when encountering a balise.

Although, at present, a digital map is not regulated by the European standard, a significant effort is being carried out for taking a digital map into account within the European Rail Traffic Management System User Group [6] and the European Community under specific research projects [3], [7].

A. PAPER CONTRIBUTIONS

This paper presents a new methodology for reducing the odometry error by exploiting inertial data to recognize a set of features on the track (e.g., curves, switches, and slopes), which are matched with the corresponding ones extracted from a digital map in an offline phase and stored as position markers. More specifically, the following novel contributions are provided in the paper:

- A new method is proposed to reconstruct the curvature profile of a railway track based on the lateral accelerations sensed by an IMU.
- A new map matching algorithm is presented to align the curvature features detected by the IMU with those extracted from a digital map and stored as position markers. The alignment with the map also exploits the rough train position estimated by the odometry.
- An approach to employ the matched markers to reset the confidence interval of the odometry error is introduced.
- Several experiments are reported to validate the proposed approach both on simulated and real data.

The rest of the paper is organized as follows: Section II discusses the state of the art in the field, Section III presents the proposed solution, Section IV reports the experimental results, and Section V states our conclusions.

II. STATE OF THE ART

The efficiency of a railway transportation system is highly related to the number of trains that can travel on the tracks, which, in turns, depends on the position error of the localization system. Given the constant demand for higher efficiency and high-speed trains, accurate train localization became an emerging topic in computing systems posing great research challenges [4]. Hutchinson et al. [8] provided an overview of the research activities on this topic within the Shift2Rail Joint Undertaking, showing the strong commitment to bringing such solutions into operation in the near future. For example, the GRAIL-2 [9] European project aims at integrating GNSS data in a demonstrator to enhance odometry capabilities and verify its adherence to current standards on safety and security.

Nowadays, the train position refers to the position of train front-end with respect to the position of a reference marker. Such a relative position is obtained by measuring the traveled distance from the last position marker located on the track. The traveled distance is safely measured by the odometry function implemented in the onboard signaling platform using wheel encoders or a combination of sensors such as wheel encoders, monoaxial accelerometers, and radars. The types of sensors and how they are used depend upon the suppliers of the onboard platforms. An onboard signaling platform implements many other functions such as those responsible for (a) reading the information sent by the beacons (balises) installed on the track sleepers, (b) associating the origin of the measurement reference system to the location of the last balise, and (c) carrying out the continuous supervision of specific locations on the line and at the stations to safely guarantee that the train reaches each specific location with the allowed speed.

Many different odometry algorithms have been developed during the last decades with the primary objective of improving both the accuracy and the availability of the measured distances under all the various operational conditions (e.g., snow, water, leaves on the rails). Typically, the error models associated with the accuracy of the measured traveled distance foresee an error contribution equal to a percentage of the traveled distance (e.g., 5% for the European Rail Traffic Management System (ERTMS) system [2]).

Around 2004, the Italian Infrastructure Manager and Italian Railway Undertaking started the deployment of the national Automatic Train Protection (ATP) system. In particular, due to the vital role of the odometry function in terms of both safety and performance, they have provided detailed specifications for the Odometry Algorithm [10] and the Train Brake Model [11]. The odometry function specification comes from many research activities and experimental analyses performed on different types of trains. For example, Allotta et al. [12] proposed an odometry algorithm, compliant with this specification, which can detect under not severe environmental conditions when the train is slipping or sliding by comparing different measured speeds acquired employing multiple encoders placed on separate wheel axes. The same group performed further research activities to extend the proposed algorithm by including an IMU [13]. However, this extended version has not been included in the ATP specification; therefore, no industrial implementation has been developed and put into service.

Regarding the ERTMS Standard, in the last decade, the evolution of the specifications has improved the functional specification of many functions related to the determination of brake deceleration curves and the Traction/Braking models. In this context, the ERTMS User Group has recently activated a working group to define and deliver new user requirements [14] for a train localization system. In particular, they state that lifecycle length and cost are currently weak points of the safe part of the Command and Control System (CCS), making railways often less sustainable and competitive than other means of transport. Furthermore, updates of ERTMS specifications or ERTMS instantiations sometimes make complete system re-certification compulsory. Moreover, high-density populated areas are currently demanding higher capacity and cost-effective solutions capable of improved exploitation of existing infrastructures.

In 2015, the European Union Agency for Railways (ERA) also recognized the need to carry out a detailed ERTMS Longer Term Perspective analysis [15]. In this analysis, they reported the main potential 'game changers', which can significantly impact the ERTMS business case (due to a significant increase in operational performance and/or substantial cost reduction for the overall ERTMS system). This analysis's strategic challenges are mainly linked to the need for further capacity increase and a decrease in the overall life cycle costs of the ERTMS implementations.

In the last decade, the railway stakeholder community has recognized the benefits attainable from new technologies for innovating the railway sector [16]. Thus, many Infrastructure Managers and Joint Undertakings have started R&D programs to define new generations of control, command, and signaling systems. One of such systems is the GNSS satellite positioning. However, the achievable performances and, especially, the reachable integrity depend on factors like the surrounding environment (e.g., tunnels, mountains, buildings).

Otegui et al. [4] provided a survey on train localization, specifying that the three critical issues that need to be accounted for in this type of applications are: reducing the maintenance costs of the infrastructure, ensure the interoperability of the system on different types of trains and structures, and guaranteeing the safety-critical constraints of the applications.

Denng et al. [17] proposed an approach based on Kalman filters to fix the IMU drift and the odometry with GNSS and velocity measurements. Other works [18], [19], [20] combined the GNSS with a Doppler sensor to reconstruct the curvature of the train trajectory and used a map-matching algorithm to refine the position of the train. Lauer and Stein [21] proposed a stochastic approach to fuse GNSS data with the velocity to locate a train on a digital map represented as a graph. Heirich [22] presented a solution employing a particle filter to merge inertial and GNSS data, also exploiting a digital map to refine the computed location and detect the traveled track. Note, however, that all these works use the digital map only to reduce the localization error of an initial solution obtained exploiting the GNSS signal, which is not always available and reliable. Moreover, they use the track as a geometric constraint and are not exploiting its peculiar features.

Cai et al. [23] designed a GNSS-free algorithm based on a set of high-precision lasers mounted in suitable positions of the train to measure the distance to the buildings along the path with an accuracy up to few millimeters. Assuming that the environment's topology remains constant, sensors measurements are stored and re-used for comparing future train positions. Although this approach can be effective in a urban context, its accuracy degrades in rural environments. Tschopp et al. [24] performed a comparison between visual and visual-inertial odometry algorithms applied to railway datasets, showing that visual-aided odometry can achieve higher accuracy for rail vehicles, but it is not reliable enough to be used in isolation for safety-critical applications.

Trehag et al. [25] estimated the railroad plane geometry through onboard IMU measurements, integrating the GNSS with an extended Kalman filter for removing the bias from the measurements in the yaw rate estimate, used to create a curvature's profile. Heirich et al. [26] derived a set of geometric characteristics of the track, as curvature and inclination, from inertial data and speculated that such features could eventually be exploited to localize the train, but no method is proposed to reduce the localization error. The problem of map-matching using IMU has been analyzed by Belabbas et al. [27], who pointed out that different curvature estimation methods perform differently depending on the sensor characteristics, proposing to combine them to overcome distinctive sensors limitations.

The solution proposed in this paper does not rely on GNSS, and it is only based on inertial data and the current odometry estimation. The key idea of the proposed method is to perform offline processing of a digital route map to extract a number of features (e.g., curvatures, switches, and slopes) and save them in the map file as virtual position markers. Then, inertial data are used at run time to recognize such features and match them with those stored in the map. The position of the matched virtual marker can be used to reduce the error accumulated by the wheel odometry system, as presented in Section III, but it can be integrated as a new input in more complex data-fusion approaches, exploiting its independence from GNSS data and balises.

III. PROPOSED SOLUTION

As IMU sensors are affected by error sources that are totally independent of those concerning the GNSS positioning, the proposed work describes an approach for extracting cues from the track layout (e.g., curvatures, slopes, cross switches) using inertial data, matching their location with the corresponding ones stored in an enriched version of the Digital Map. The navigation of the Digital Map is done by using the train odometry. Once the locations of a cue is detected, an onboard signaling subsystem can potentially use this location as an a-priori known position marker.

Such position markers can be used to reset the odometry confidence intervals, independently of the equivalent train confidence interval reset done by the ETCS kernel[1] by using linked balise groups and the linking information [1]. The detection of position errors greater than a specified threshold (a configuration parameter) can be used as a fault detection mechanism whose design has to be carefully done considering the recommendations on the Detection and Negation Times provided in [28]. The detected position can be exploited for other uses, such as one of the inputs for 3D localization algorithms or a monitoring technique for detecting unbounded GNSS position errors, but the research on these benefits goes beyond the scope of the paper.

The employment of inertial sensors has also the main advantage of not relying on any external signals, which could be either tampered or unavailable in some sections of the railway paths or, in the worst case, disturbed by intentional or unintentional radio frequency interferences.

Among the track features, the path curvature ρ, defined as the inverse of the curve radius r, has a significant importance and has been selected as the key index for this paper. Nevertheless, the proposed approach can be extended to other track parameters, as discussed in the final remarks. On the other hand, the proposed approach requires a precise Digital Map to achieve train positions with a desired accuracy and availability.

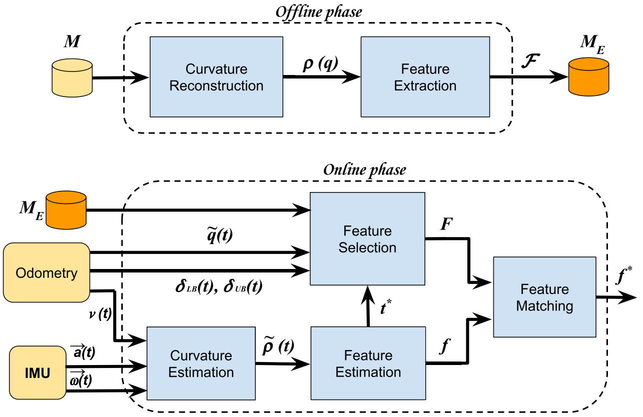


FIGURE 2. Block diagrams of the offline and online processes used for extracting features from the digital map (top) and matching them with those estimated by inertial data (bottom), respectively.

A. SYSTEM ARCHITECTURE

The overall architecture used to implement the proposed method is illustrated in Figure 2. It consists of two macro blocks, one executed offline and one executed online.

The off− =line macroblock extends the row Digital Map (M) generating a new enriched map (M_E), containing a set of features along the path. A path in the map is represented as a sequence of 3D points, each characterized by its coordinates expressed in the North, East, Down (NED) reference system [29].[2] In particular, the Curvature Reconstruction block receives the Digital Map M as input and computes the curvature $\rho(q)$ as a function of the traveled distance q, expressed in the odometry coordinate system. Then, the curvature data are sent to the Feature Extraction block, which extracts a set of features along the path, corresponding to the railway sections with the highest absolute curvature derivative. In particular, each feature f_i is represented by a tuple (q_i, ρ_i, ρ_i') storing its position, its curvature, and the curvature derivative. The set $\mathcal{F}$ of all the features selected on the Digital Map is then stored in the enriched map M_E.

The online macro block estimates the features from inertial data, matching them with the ones stored in the enriched map M_E. Every time a match is found, the odometry error is reduced. In particular, the Curvature Estimation block reconstructs the curvature $\tilde{\rho}(t)$ from the inertial data (acceleration vector $\vec{a}(t)$ and angular speed vector $\vec{\omega}(t)$) and the train velocity $v(t)$ provided by the odometry (e.g., based on wheel sensors). The Feature Estimation block extracts the features

[1] The core of the ERTMS/ETCS onboard equipment.

[2] Coordinates are stored in the map in the Earth-centered Earth-fixed (ECEF) reference system and converted to the NED reference system to be used in the algorithm.

from the curvature $\widetilde{\rho}(t)$ at runtime, applying an algorithm similar to the one used offline.

In the following, the actual train position is denoted as $q(t)$, the position estimated by odometry is denoted by $\widetilde{q}(t)$, and the odometric error is denoted by $\widetilde{\epsilon}(t) = \widetilde{q}(t) - q(t)$.

As soon as a new feature is detected by the Feature Estimation block at time t^*, it is directly sent to the Feature Matching block, which performs the matching with the features stored in the digital map M_E. The matched feature is denoted by $f^*(q^*, \rho^*, \rho'^*)$. To restrict the set of features $\mathcal{F}$ to be compared, a subset F is selected by the Feature Selection block within a moving window on the route centered on the current estimated train position $\widetilde{q}(t)$ and having a width $[\delta_{lb}(t), \delta_{ub}(t)]$ established by the maximum current error bounds of the odometry. Such bounds can be computed as follows. The odometry system is built to guarantee a maximum error specified as a function of the actual position q. In particular, the regulation requires the estimated position $\widetilde{q}$ to be in the range

$$q - a - b(q - q_L) \ < \ \widetilde{q} \ < \ q + a + b(q - q_L) \qquad (1)$$

where q_L denotes the position of the last detected feature, while the constants a and b depend on the specific country (e.g., ETCS performance requirements for interoperability [2] specify $a = 5$ m and $b = 0.05$). The value of q_L is initialized by $\widetilde{q}(t_0)$. Inverting Equation (1), it is possible to state that, given the position $\widetilde{q}$ estimated by the odometry, the actual train position is within the range

$$\delta_{lb} < q < \delta_{ub} \qquad (2)$$

where

$$\delta_{lb} = \frac{\widetilde{q} + b\, q_L - a}{1 + b}, \qquad \delta_{ub} = \frac{\widetilde{q} - b\, q_L + a}{1 - b}. \qquad (3)$$

The Feature Matching block produces as output the matched feature f^*, whose position q^* is used to reduce the odometry error by the amount $\lambda = \widetilde{q}(t^*) - q_L$.

If the detected feature is used for odometry error correction, when a matching occurs, the value of q_L is set to q^* and the correction is performed. The fixed error is computed as $\epsilon_f = \widetilde{\epsilon} - \lambda$.

A more detailed explanation of the functional blocks illustrated in Figure 2 is reported in the following sections. Table 1 summarizes the most common symbols used throughout the paper.

TABLE 1. List of Symbols used in the paper.

Symbol	Description
q	Ground truth position
$\widetilde{q}$	Estimated position
ρ	Ground truth curvature
$\widetilde{\rho}$	Estimated curvature
δ_{lb}	Lower bound error of the estimated position
δ_{ub}	Upper bound error of the estimated position
M	Digital map
M_E	Digital map enriched with track features
$\mathcal{F}$	Full set of features stored in the digital map
F	Subset of features in the digital map selected for matching
f	Generic feature detected on the track
f^*	Last detect feature
t^*	Time at which f^* is detected
q^*	Position at which f^* is located
q_L	Position of the last matched feature
η_M	Map feature detection threshold
η_I	Inertial feature detection threshold
λ	Odometric error correction
$\widetilde{\epsilon}$	Uncorrected odometric error
ϵ_f	Fixed odometric error

B. OFFLINE CURVATURE RECONSTRUCTION

At the moment, the digital map under investigation in Hitachi Rail STS [30] is an oriented graph of interconnected nodes with variable distances between them, not including information on curves, gradients, and superelevations. Therefore, an offline curvature reconstruction process has to be performed taking into account the characteristics of the current Digital Map.

To understand the curvature reconstruction process considered in this paper, we recall some basic geometric criteria adopted to design rail transit, aimed at providing cost-effective, efficient, and comfortable transportation, while maintaining adequate safety factors [31], [32]. For example, the Italian track design specification [32] states that the minimum radii of the horizontal and vertical curvatures of a railway track in line are 275 m and 500 m, respectively; the horizontal value is further decreases to 170 m for some tracks located in train stations. Moreover, transitions between curves have to follow a clothoid or a cubical parabolic function. For instance, a curve subsequent to a straight section starts with a curvature ρ equal to zero that is increased linearly until the desired plateau value and is kept constant for the central part of the curve. The end of the curve follows the same profile, where the curvature linearly reduces to zero. In addition, the track design criteria state that between a curve with a positive radius and another one with a negative radius there must be a straight section with a minimum length of 50 m.

A convenient representation of a track used in several industrial solutions makes use of a graph where the nodes are geolocalized points of the track, and a link represents the connection between two consecutive nodes [30], [33]. The basic rule used to develop a Digital Map is to sample the nodes' positions following the convention that the link length should be inversely proportional to the curvature and less than a maximum sampling distance.

The curvature reconstruction algorithm receives the sequence of geolocalized nodes from the map M and estimates the curvature at each node, using Taubin's method [34]. The reconstruction algorithm must also deal with errors in nodes coordinates due to measurement inaccuracies. Such errors in node positions can cause large errors in the estimated curvature when nodes are separated by a small distance. To reduce such errors consecutive points closer than a minimum distance d_{min} are discarded. Since the maximum location error affecting the node positions is a specification

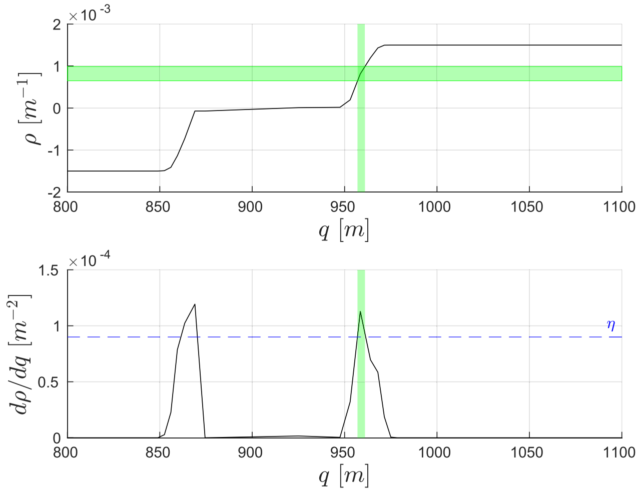


FIGURE 3. Example of a feature selection on the track curvature and its absolute derivative.

requirement for the Digital Map, the best value for d_{min} is a function of the map features. Section IV-E will present some preliminary results on an actual track while describing the map used to evaluate the proposed approach on real data.

C. OFFLINE FEATURES EXTRACTION

Curvature features are selected offline by the Feature Extraction block from the curvature profile ρ previously reconstructed from the Digital Map M. To increase the accuracy of the feature-matching algorithm, features are selected by choosing those points on the track whose curvature variation has a local maximum above a certain threshold η_M (a configurable parameter).

Figure 3 illustrates how the curvature error affects the position error of the feature, when this is selected based on the maximum absolute curvature derivative.

The threshold is selected as a trade-off between the radius range admitted by the construction rules and the error introduced by inaccuracies in the geolocalization of points composing the map. A low threshold value allows detecting curves with a large radius, while a higher value provides robustness against computation errors due to inaccuracies in the Digital Map locations. In this paper, a value $\eta_M = 5 \times 10^{-6}\text{m}^{-2}$ has been used to define the intervals in which local maxima of the absolute curvature derivative are computed.

Algorithm 1 presents the pseudocode of the procedure used by the Feature Extraction block to select the features on the digital map M from the curvature profile ρ and its derivative ρ'.

In particular, the proposed algorithm sequentially analyzes the K_M points describing the curvature profile of the track and selects each feature by choosing the point with the maximum absolute curvature derivative within a search interval. The beginning of the search interval is set to the point in which the absolute curvature derivative becomes higher than the threshold η_M (lines 4-7), while the end of the interval is set as the first successive point where the absolute curvature derivative returns below the threshold (line 14). Within each search interval, the algorithm keeps track of the point (q_{max}, ρ_{max}) with the maximum absolute curvature derivative ρ'_{max} (lines 8-13), which becomes a new feature. The condition $(q_{max} > 0)$ at line 9 is necessary to avoid computing a feature in the case in which the track starts with an absolute curvature derivative above the threshold. The new feature is then added to the set $\mathcal{F}$ of detected features that will be embedded in the enhanced Digital Map M^E (line 16).

Algorithm 1 Select Features on the Curvature From the Digital Map

```
    Input: $\rho(q)$, $\rho'(q)$
    Output: $\mathcal{F}$
 1: procedure Features_Selection
 2:     $q_{max} \leftarrow 0$
 3:     for $(k \leftarrow 0$ to $K - 1)$ do
 4:         if $(|\ \rho'(q_{k-1})\ | < \eta_M) \wedge (|\ \rho'(q_k)\ | \geq \eta_M)$ then
 5:             $q_{max} \leftarrow q_k$
 6:             $\rho_{max} \leftarrow \rho(q_k)$
 7:             $\rho'_{max} \leftarrow \rho'(q_k)$
 8:         end if
 9:         if $(q_{max} > 0)$ then
10:             if $\rho'(q_k) > \rho'(q_{max})$ then
11:                 $q_{max} \leftarrow q_k$
12:                 $\rho_{max} \leftarrow \rho(q_k)$
13:                 $\rho'_{max} \leftarrow \rho'(q_k)$
14:             end if
15:             if $(|\ \rho'(q_{k-1})\ | \geq \eta_M) \wedge (|\ \rho'(q_k)\ | < \eta_M)$
    then
16:                 $\mathcal{F} \leftarrow \mathcal{F} \cup (q_{max}, \rho_{max}, \rho'_{max})$
17:             end if
18:         end if
19:     end for
20: end procedure
```

Figure 4 shows a curvature profile reconstructed from a synthetically generated track. Red crosses denote the selected features corresponding to the points in which the absolute curvature derivative has a local maximum.

D. ONLINE CURVATURE ESTIMATION

The online curvature estimation process exploits the inertial data to reconstruct the curvature profile of the route followed by the train.

The train velocity vector can be defined as $\vec{v} = v_x\hat{u}_X + v_y\hat{u}_Y$, where u_X and u_Y represent the unit vector of longitudinal and lateral axes in the body frame, respectively. Assuming that the velocity vector is directed along the longitudinal axis of the track, the lateral speed is null $(v_y = 0)$ and the train acceleration can be expressed as

$$\vec{a} = \frac{d(v_x\hat{u}_X)}{dt} = \frac{d(\dot{s}\hat{u}_X)}{dt} = \ddot{s}\hat{u}_X + \dot{s}\dot{\hat{u}}_X, \qquad (4)$$

where s represents the traveled space.

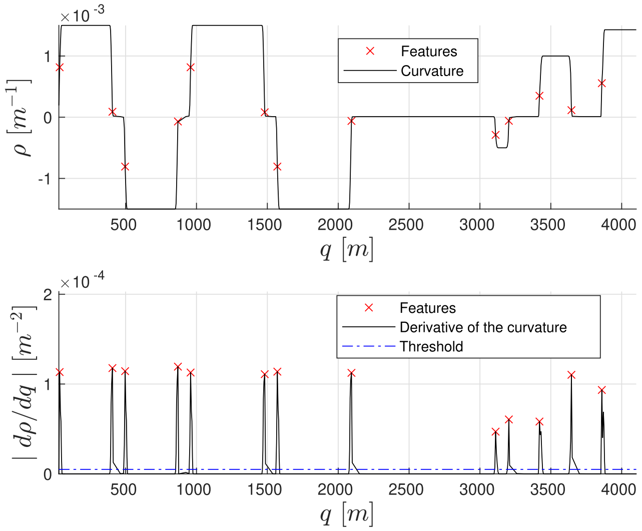


FIGURE 4. **Example of a curvature profile reconstructed from a synthetically generated track (top). Red crosses denote the selected features corresponding to the points in which the absolute curvature derivative has a local maximum (bottom).**

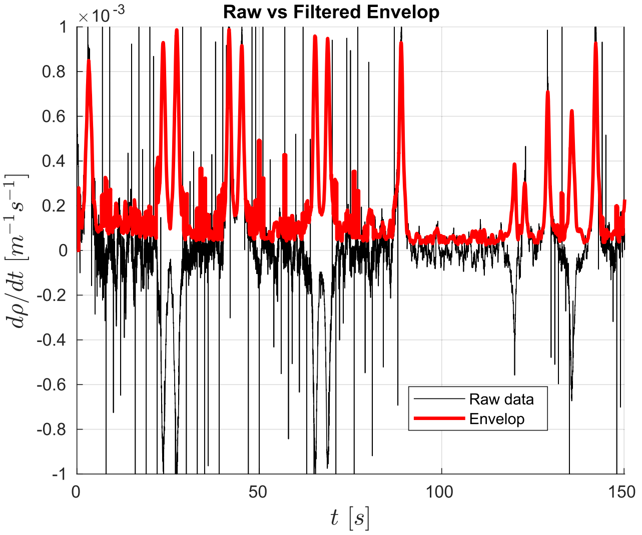


FIGURE 5. **Example of raw curvature derivative (black curve) and filtered one (thick red curve).**

Since the acceleration vector lies on the plan defined by (u_X, u_Y), it can be decomposed into two components exploiting the relation $\dot{\hat{u}}_X = \dot{\Psi}\hat{u}_Y$, where $\dot{\Psi}$ is the yaw angular rate obtained from the IMU in the NED frame. Therefore the acceleration can be expressed as $\overrightarrow{a} = a_x\hat{u}_X + a_y\hat{u}_Y$, where $a_x = \ddot{s}$ and $a_y = \dot{s}\dot{\Psi}$. It is possible to write s and its derivative in the form:

$$s = r\Psi \qquad \frac{ds}{dt} = \frac{dr}{dt}\Psi + r\frac{d\Psi}{dt}. \tag{5}$$

Under the hypothesis that the local radius r is constant, the derivative can be represented as $\dot{s} = r\dot{\Psi}$, and the normal acceleration a_y can be expressed as:

$$a_y = r\dot{\Psi}^2 = r\left(\frac{\dot{s}}{r}\right)^2 = \frac{\dot{s}^2}{r} = \frac{||v||^2}{r}. \tag{6}$$

Equation (6) describes the relationship between the longitudinal speed $||v||$, the curvature radius r, and the cross track acceleration a_y. Since the curvature ρ is the inverse of the radius, this formula allows defining three methods for deriving it:

$$\begin{aligned} \rho^{(1)}(t) &= \frac{\dot{\Psi}^2}{a_y} = \frac{w_z^2}{a_y}, \\ \rho^{(2)}(t) &= \frac{\dot{\Psi}}{||v||} = \frac{w_z}{||v||}, \\ \rho^{(3)}(t) &= \frac{a_y}{||v||^2}. \end{aligned} \tag{7}$$

The first method ($\rho^{(1)}$) does not require integration, because the IMU directly senses the dynamic variables. However, when the train is covering a straight segment of the track, both the numerator and the denominator are zero and the ratio becomes indeterminate, making the result only depending on the noise in the sensory data. For this reason, this method cannot be exploited alone.

The second and the third methods ($\rho^{(2)}$ and $\rho^{(3)}$) are similar, with the speed appearing at the denominator. However, in the third method, the speed is quadratic, meaning that $\rho^{(3)}$ is more sensitive to errors in the speed value provided by the odometry function. This effect is also more significant at low speeds, because the error component in the measurement becomes more prominent and is no longer negligible.

Bellabas et al. [27] compared the three methods and showed that no one dominates the others, suggesting that a robust solution should merge them. Although this goal is outside the scope of this work, Section IV-A presents some remarks on the performance of some integrated methods applied to data acquired in a real-world scenario.

E. ONLINE FEATURES ESTIMATION AND MATCHING

While the train is moving, the online Feature Estimation block detects a feature f using the curvature $\widetilde{\rho}$ computed from the inertial data by the Curvature Estimation block and provides it to the Feature Matching block, which compares it against the set $\mathcal{F}$ stored in the enhanced map M^E and provides the current positions q^* corresponding to the matched feature.

As inertial data are typically very noisy, the curvature derivative is first processed by a low-pass filter before being analyzed by the feature extraction algorithm. Figure 5 shows the raw curvature derivative (thin black plot) and the corresponding envelop (thick red plot) filtered using a root mean square function with a window of 50 samples.

The curvature $\widetilde{\rho}$ computed from the inertial data and the corresponding filtered derivative $\widetilde{\rho}'$ are analyzed online by the Feature Estimation block with an approach similar to the one applied offline to the curvature obtained from the map by the Feature Extraction block. The pseudocode of Feature Estimation is detailed in Algorithm 2.

When a new curvature is computed from inertial data at time t, the algorithm evaluates whether its absolute derivative crosses the threshold η_I (line 6). The crossing is detected if the value $\widetilde{\rho}'(t)$ is above η_I and ρ_{prev} is below it, where ρ_{prev} is the value of $\widetilde{\rho}'$ stored at the previous iteration (line 24). If a crossing is detected, the feature f is associated with the curvature $\widetilde{\rho}(t)$ and derivative $\widetilde{\rho}'(t)$ values of the current time t. (lines 6-11). The algorithm performs two activities when the new data arrives within a search interval (line 12). If the new absolute curvature derivative is greater than the current maximum, it is selected as the new maximum (lines 13-18). Then, if the absolute curvature derivative has decreased below the threshold η_I (line 19) the selected maximum is passed to the Feature Matching block (line 20). Note that the Curvature Estimation and the Feature Estimation blocks are executed at the rate imposed by the inertial data, while the matching is invoked only when a new feature is detected.

Algorithm 2 Detect a New Feature on the Curvature Computed From the Inertial Data

```
    Input: $\rho(t)$, $\rho'(t)$
 1: procedure Feature_Estimation
 2:     if ($first\_iteration$) then
 3:         $q_{max} \leftarrow 0$
 4:         $\rho'_{prev} \leftarrow \eta_I$
 5:     end if
 6:     if (| $\rho'_{prev}$ |< $\eta_I$) $\wedge$ (| $\widetilde{\rho}'(t)$ |≥ $\eta_I$) then
 7:         $t^* \leftarrow t$
 8:         $q_{max} \leftarrow \tilde{q}(t)$
 9:         $\rho^* \leftarrow \widetilde{\rho}(t)$
10:         $\rho'^* \leftarrow \widetilde{\rho}'(t)$
11:     end if
12:     if ($q_{max} > 0$) then
13:         if $\widetilde{\rho}'(t) > \widetilde{\rho}'^*$ then
14:             $t^* \leftarrow t$
15:             $q_{max} \leftarrow \tilde{q}(t)$
16:             $\rho^* \leftarrow \widetilde{\rho}(t)$
17:             $\rho'^* \leftarrow \widetilde{\rho}'(t)$
18:         end if
19:         if (| $\rho'_{prev}$ |≥ $\eta_I$) $\wedge$ (| $\widetilde{\rho}'(t)$ |< $\eta_I$) then
20:             $f \leftarrow (t^*, \rho^*, \rho'^*)$
21:             $F \leftarrow$ Feature_Selection($t^*$)
22:             Feature_Matching($f$, $F$)
23:             $q_{max} \leftarrow 0$
24:         end if
25:     end if
26:     $\rho'_{prev} \leftarrow \widetilde{\rho}'(t)$
27: end procedure
```

Figure 6 presents the features detected on the curvature ρ estimated from the inertial data while traveling the same synthetic track analyzed in Figure 4. In particular, red crosses denote the points with the maximum absolute curvature derivative (i.e., the selected features).

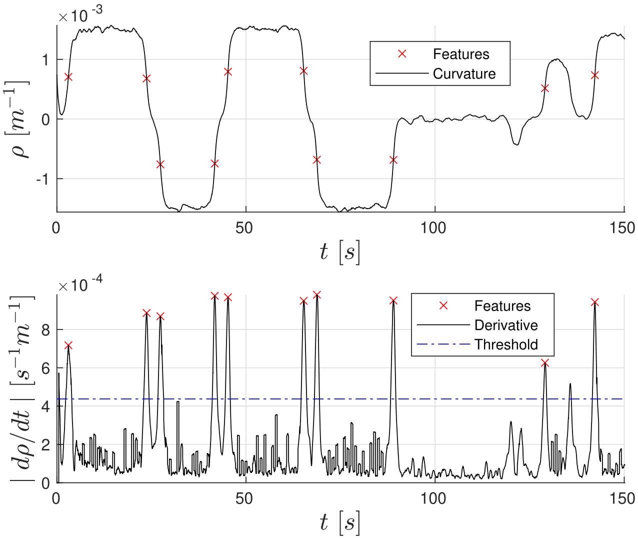


FIGURE 6. Feature estimation with inertial data on the synthetic track layout.

The Feature Selection block chooses the possible matching features for the estimated one f, and its pseudocode is detailed in Algorithm 3.

Given the time t^* at which feature f is detected, the algorithm defines the search interval W as a moving window shifted on the track with the motion of the train (line 2). The window is centered on the position estimated at time t^* by the odometry $\tilde{q}(t^*)$, whereas its width is determined by the odometry error bounds, as illustrated in Figure 7. A subset of features F is extracted from the set of features $\mathcal{F}$, computed offline by the Features Extraction block and stored in the enriched Digital Map M_E, and including all features f_i whose position q_i is within the search interval W (line 3). The subset F is provided to the Feature Matching block for the actual matching.

Algorithm 3 Identifies the Set F of Eligible Features in the Map to Match With the Estimated One f

```
    Input: $t^*$
    Output: $F$
 1: procedure Feature_Selection
 2:     Gets $\mathcal{F}$, $\delta_{LB}(t^*)$, $\delta_{UB}(t^*)$
 3:     $W \leftarrow [\delta_{LB}(t^*), \delta_{UB}(t^*)]$
 4:     $F \leftarrow \{f_i \in \mathcal{F} \mid q_i \in W \wedge q_i > q_{LM}\}$
 5: end procedure
```

The Feature Matching block performs the final step of the proposed approach. Among the features in the subset F extracted from the enriched Digital Map M_E, it computes the one matching the feature f estimated from the inertial data.

This pairing allows determining the error accumulated by the odometry estimation $\widetilde{q}(t)$ at time t^*. Algorithm 4 presents the pseudocode of the Features Matching block.

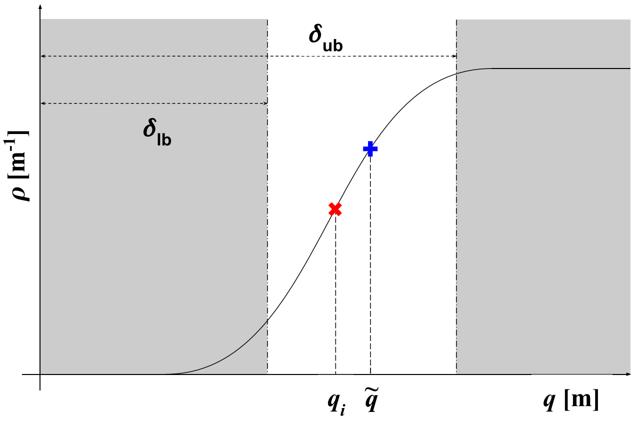


FIGURE 7. Example of the moving window centered on the current train position $\tilde{q}$ estimated by wheel odometry, and the selected feature q_i.

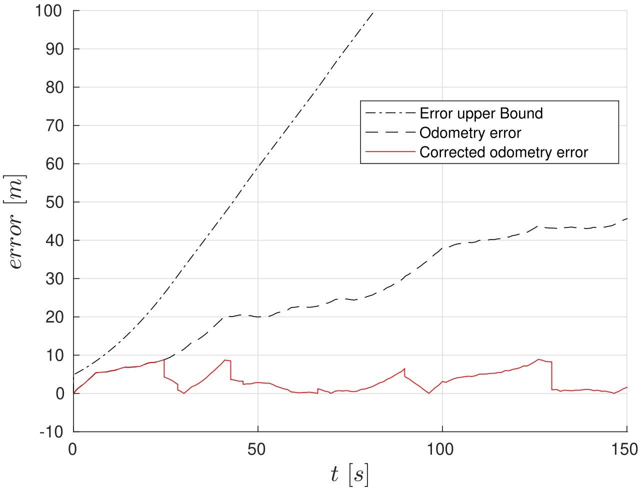


FIGURE 8. Comparison between the odometry error with and without the proposed correction.

The algorithm starts creating a new set F^N of features including those belonging to F and following the last match f_L (i.e., those features with a position q_i greater than the position of the last detected feature q_L), as performed in line 5.

A feature can be characterized by considering the signs of the curvature and its derivative. Hence, by evaluating them is possible to know whether the position is relative to the beginning or the ending of a curve and if the turn is right or left. The approach exploits this information to create a new set F^C selecting the features in F^N whose curvature and related derivative have a concording sign with those of f (line 6). Note that, given the geometric criteria in track design and the constraints in the maximum tolerance in the odometric error, the subset F^C usually contains only one feature. If the set includes more than one feature, the algorithm selects the first one along the track based on the consideration that features are engaged sequentially (i.e., with an increasing value of q), as performed in line 8.

Algorithm 4 Match the Estimated Feature f With the Ones in F

```
    Input: $f$, $F$
    Output: $q^*$
 1: procedure Features_Matching
 2:     if ($first\_iteration$) then
 3:         $q_L \leftarrow 0$
 4:     end if
 5:     $F^N \leftarrow \{f_i \in F \mid q_i > q_L\}$
 6:     $F^C \leftarrow \{f_i \in F^N \mid \rho(q_i)\rho^* > 0 \wedge \rho'(q_i)\rho'^* > 0\}$
 7:     if ($F^C \neq \emptyset$) then
 8:         $q^* \leftarrow \min\{q_i \mid f_i \in F^C\}$
 9:         $q_L = q^*$
10:     else
11:         $q^* \leftarrow$ MATCH_NOT_FOUND
12:     end if
13: end procedure
```

The feature-matching algorithm produces as output a tuple of values $(t^*, \tilde{q}(t^*), q^*)$, where t^* is the time at which the feature is detected, $\tilde{q}(t^*)$ is the odometry estimation at time t^*, and q^* is its position of the matched feature on the enhanced Digital Map.

F. ODOMETRIC ERROR CORRECTION

If the detected feature is used for correcting the odometry error, the position correction $\lambda(t)$, performed at detection time, is computed as

$$\lambda(t) = \tilde{q}(t^*) - q_L. \tag{8}$$

Figure 8 compares the error of the odometry without correction $\tilde{\epsilon}(t) = \tilde{q}(t) - q(t)$ against the one fixed with the proposed method $\epsilon_f(t) = \tilde{\epsilon}(t) - \lambda(t)$.

Note that applying the correction is not enough to reset the error. In fact, the residual error is only partially ascribable to the map-matching algorithm, due to the errors in the geolocalization of the points in the map and those in the curvature reconstruction. Also observe that the feature correction is performed at time t, when the curvature derivative goes below the threshold (line 19 in Algorithm 2), after time t^*. In the meantime, the odometry has accumulated an additional error that will be handled when the next feature is matched.

Finally, it is worth noting that, although this paper only focuses on features coming from curvature variations, other types of track features can be considered, like switches, as well as altitude and roll gradients. For example, the proposed algorithm can detect pitches due to slopes and roll variations related to track inclinations present on the train route. Each feature can potentially act as a *position marker* used for resetting the odometric error, detecting unbounded GNSS position data, and performing 3D localization.

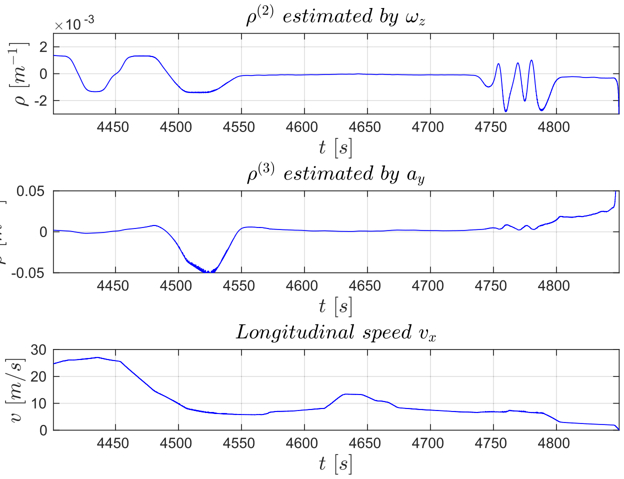


FIGURE 9. Curvature ρ estimated using $\rho^{(2)}$ (a) and $\rho^{(3)}$ (b), shown along with the longitudinal train speed (c).

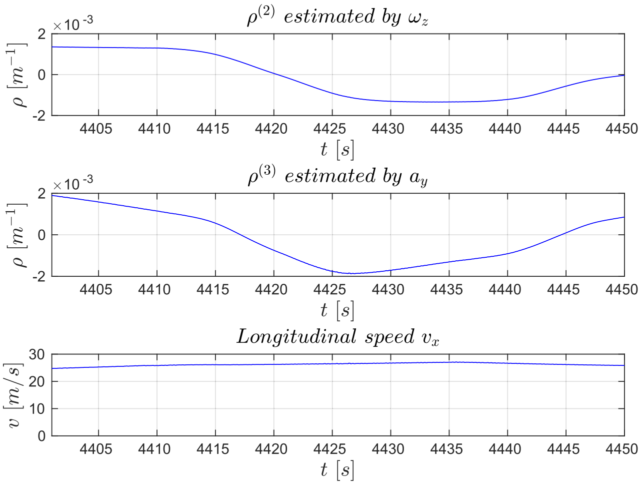


FIGURE 10. Curvature ρ estimated using $\rho^{(2)}$ (a) and $\rho^{(3)}$ (b), shown along with the longitudinal speed (c) in high speed conditions.

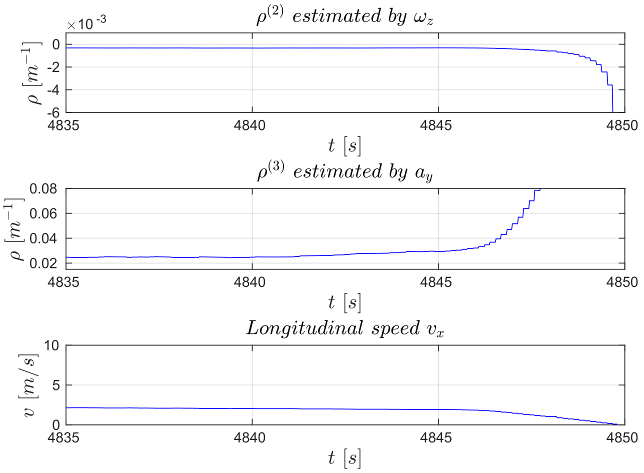


FIGURE 11. Curvature ρ estimated using $\rho^{(2)}$ (a) and $\rho^{(3)}$ (b), shown along with the longitudinal speed (c) in low speed conditions.

IV. EXPERIMENTAL RESULTS

This section presents a set of experiments carried out to test the proposed approach, both on synthetic and real-world data.

A. CURVATURE ESTIMATION

As described in Section III-D, the curvature of the actual train route can be estimated at runtime using different methods applied to inertial data, as presented in Equation (7). This subsection compares the performance of two approaches (i.e., $\rho^{(2)}$ and $\rho^{(3)}$) using real data acquired from the Hitachi Rail STS trial site. The achieved results show that the effectiveness of the two approaches depends on the quality of the inertial data (i.e., noise) and the operating conditions (mainly the train speed).

Figure 9 compares the curvature estimated by using the two approaches for a given speed profile.

In particular, the first two plots report the curvature estimated with the second ($\rho^{(2)}$) and third ($\rho^{(3)}$) approaches, while the third plot shows the profile of the longitudinal train speed during the arrival to a station, where the speed decreases to zero. As it can be observed, the difference between the two approaches depends on the train speed. When the speed is higher than 20 m/s, the curvature profiles obtained by the two approaches are comparable, as illustrated in Figure 10, which compares the two curvature profiles when the speed is between 20 and 30 m/s. Conversely, the third approach ($\rho^{(3)}$) shows a significant degradation for low speeds, due to the quadratic term at the denominator, which amplifies the odometry error. This is also confirmed by Belabbas et al. [27], who reported that the approach using the yaw angular rate ($\rho^{(2)}$) is better than the one based on the lateral acceleration ($\rho^{(3)}$) for low-medium speed ranges, up to 30 m/s.

In particular, Figure 11 compares the two approaches for low speeds, showing that both results diverge when the speed is close to zero. However, the difference between them is significant, as the curvature estimated by the acceleration ($\rho^{(3)}$) is affected by an error that is more than ten times higher than the one achieved using the angular velocity ($\rho^{(2)}$).

This comparison confirms that the performance of both approaches depends on the quality of the inertial data and the train speed, and their optimal integration depends on the characteristics of the specific inertial sensors and the train and line operational conditions. Integrating such methods, however, is outside the scope of this paper and will be carried out in a future work. In the following experiments, the curvature has been estimated by exploiting the angular speed of the yaw angle ($\rho^{(2)}$) due to its robustness.

B. EXPERIMENTAL SETUP

To evaluate the performance of the proposed approach in a large variety of situations, a generator of synthetic railway paths has been developed. To obtain realistic paths, the generator follows the specifications provided by the Italian

Infrastructure Manager [32]. Initially, the path is generated as a continuous line composed of straight and curved segments (straight segments are generated as a curve with a radius equal to 10000 meters). Each curve consists of three elements: an initial spiral curve, a circular curve with a constant radius, and a final spiral curve. Spiral curves are generated using a linearly increasing or decreasing radius to ensure a smooth transition between two consecutive track segments. The continuous path is then sampled with a fixed step of 5 m in order to have a fine-grained discrete path, used as ground truth. A synthetic path is generated by sub-sampling the fine-grained path, so obtaining a sequence of points with a structure similar to the one adopted in a real Digital Map, where the distance between consecutive points is proportional to the radius.

The map generator takes as input a number of parameters, as the maximum length of the path, the number of curves, and, for each curve, its radius and degree of curvature. Each parameter value is randomly generated within a given range to obtain a number of pseudo-random paths. Another input parameter is the maximum allowed train speed, which defines the maximum allowed curvature and characterizes the guidance law during the simulation. Once a synthetic path is generated, the simulator directly computes the train dynamics and uses the train state variables to provide the output of the inertial sensor, also adding a Gaussian noise with zero mean and variance compatible with the sensor used in the real environment.

C. SENSITIVITY TO THE IMU NOISE

A set of three experiments was carried out to verify how the noise present on inertial data affects the corrected odometric error $\epsilon_f(t)$. The experiments have been carried out on a synthetic path consisting of two curves illustrated in Figure 12. Figure 13 reports the corresponding ground truth curvature (top graph) and the curvature reconstructed by the proposed algorithm (bottom graph).

In such a path, the proposed approach detects a feature at the entrance and the exit of each curve, for a total number of four detected features.

Considering the approach used to estimate the curvature online, which employs only the yaw angle from the inertial data, a white noise with a variable standard deviation has been injected into the yaw angular data of the virtual IMU. To generate realistic inertial data, a standard deviation σ in the range [0.08, 0.28] deg/$\sqrt{\text{s}}$ has been utilized. The value of the uncorrected odometry $\widetilde{q}(t)$ was generated from the ground truth $q(t)$ by adding an error such that $\widetilde{q}(t)$ is kept within the bound of 5 m plus 5% of the traveled distance [2].

The first test was carried out for curves with radius of 500 m and degree of curvature equal to 20 degrees. Figure 14 shows the simulation results obtained over ten runs for a single value of σ equal to 0.01 deg/$\sqrt{\text{s}}$. In particular, a profile of the residual error $\epsilon_f(t)$ is presented, together with the uncorrected odometry error $\widetilde{\epsilon}(t)$. Even if remaining significantly below the bound, the uncorrected odometry error grows monotonically, arriving at a value of 18.92 m in less than 60 seconds. Instead, the proposed approach is able to detect all four features, thus limiting the residual error below a maximum value of 4.85 m, with an average error of 2.52 m.

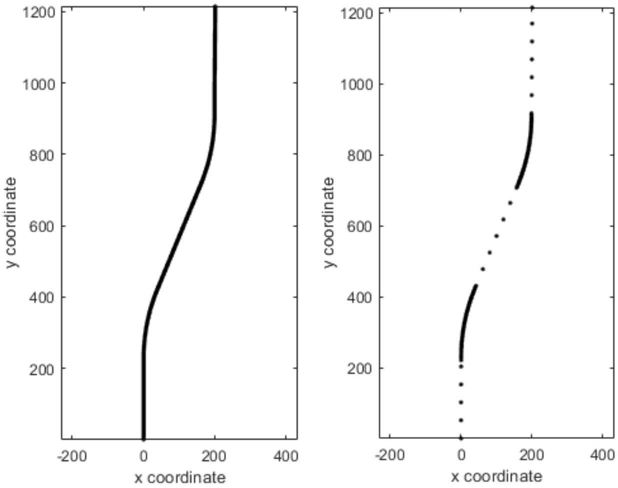


FIGURE 12. Track used to study the effect of angular random walk standard deviation (left) and the corresponding subsampled Digital Map (right).

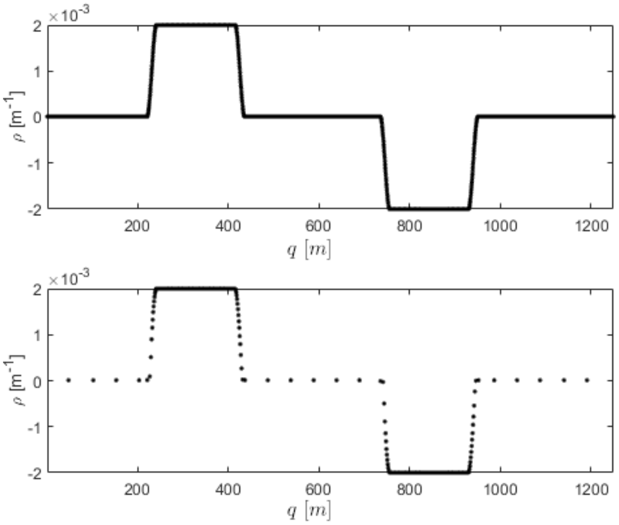


FIGURE 13. Reconstructed ρ (top graph) and the corresponding ground truth (bottom graph) in the study on the effect of angular random walk standard deviation.

The second experiment evaluated the impact of noise on the residual error by varying the value of σ in the whole range [0.08, 0.28] deg/$\sqrt{\text{s}}$ with a step of 0.02 deg/$\sqrt{\text{s}}$. The ratio between the residual error ϵ_f after the correction (t^{*+}) and the one before applying the correction (t^{*-}) is used as a metric to evaluate the performance of the proposed solution. It can be computed as $e_{\%} = 100 \cdot \widetilde{\epsilon}(t^{*+})/\widetilde{\epsilon}(t^{*-})$.

The obtained results are not shown because the variation of the residual error $e_{\%}$ resulted to be below 5% in the whole range of values used for σ. This confirms that the proposed solution is robust against a wide range of noise present in the inertial data. This behavior is due to the filtering phase of the signal performed by the Features Detection block,

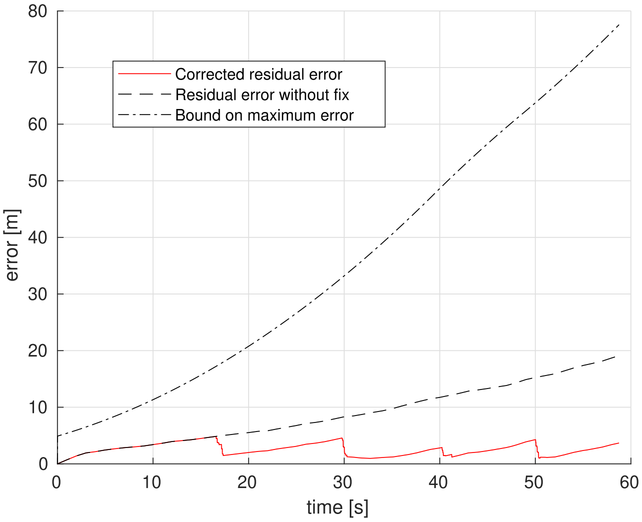


FIGURE 14. Example of the residual error as a function of time on the reference track with an AWR standard deviation $\sigma = 0.01$.

which makes the system robust even to significant changes in the standard deviation. Since an optimal filter calibration depends on the knowledge of the IMU characteristics, the selection of the IMU device and the filter design are two crucial phases that need to be carefully addressed.

A third experiment was carried out to verify whether the effect of the noise depends on the geometry of the curves. The test was performed by generating a set of ten paths formed by two curves with a degree of curve equal to 20 degrees and a radius varying in the range [150, 2500] meters. Ten repetitions were generated on each route with the same parameters used in the previous experiment, computing the average and the maximum error. The resulting trend as a function of the standard deviation σ of the noise is similar to the one observed in the previous experiment and is not reported for lack of space. Overall, the results confirm that the system is robust to noise independently of the curve parameters.

D. SENSITIVITY TO THE CURVATURE

The experiments reported here were carried out to evaluate the effects of the parameters used for filtering the inertial data and the influence of the threshold used for feature detection. Other tests were done to evaluate the effect of the curvature on the odometric error estimation for fixed filter parameters and threshold. The path shape used in these tests is the same as the one used in the previous experiments (see Figure 12 and Figure 13). The degree of curvature was kept fixed, while the maximum curvature radius of the section was varied from 150 m to 4500 m with a step of 50 m to respect the minimum and maximum values allowed by the paths construction constraints. Ten runs were performed for each value of the radius, computing the average and the maximum errors over all repetitions.

Figure 15 shows the route features detected on one of the paths, where crosses denote the detected features and circles show the point where the odometry fixes were performed. Since fixes allow reducing the cumulative error due to the drift in odometry data, precisely detecting all the features is crucial for containing the position error between balises.

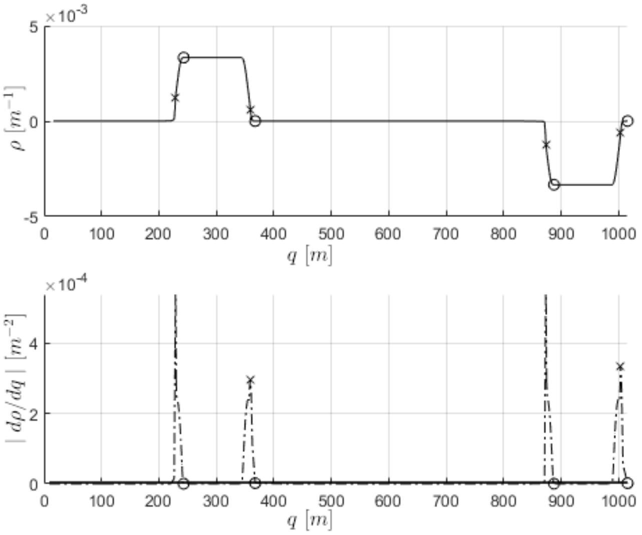


FIGURE 15. Features detected on the track curvature (denoted by crosses) and its derivative. The circles show the points where the odometry fixes were performed.

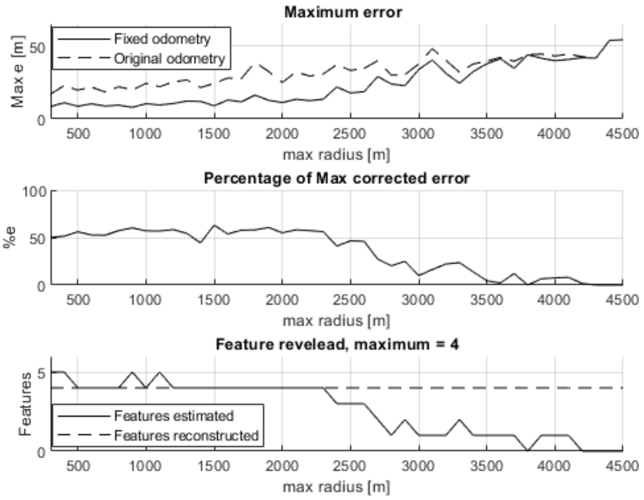


FIGURE 16. Maximum error as a function of the radius (Top), Percentage of the odometry error ($e_{\%}$) corrected as a function of the radius (Middle), Number of detected features (Bottom).

Figure 16 (top) shows the average of the maximum error of the ten repetitions as a function of the radius, starting from a minimum of 150 m to a maximum of 4500 m. As expected, as the curve radius increases, the detection capability of the approach decreases, since the curvature variation is smaller. As a result, the maximum corrected odometer error decreases, as shown in Figure 16 (middle), and the Features Estimation block cannot detect all the features, as reported in Figure 16 (bottom). This effect can be ascribed to a high threshold, set to increase the robustness against the noise of the inertial sensor after filtering. Finding the right trade-off between robustness and the residual odometry error after the correction is crucial, because inadequate filtering or thresholds limits the benefit of reducing the odometry error. In fact, the number of odometry corrections decreases as the number of detected features

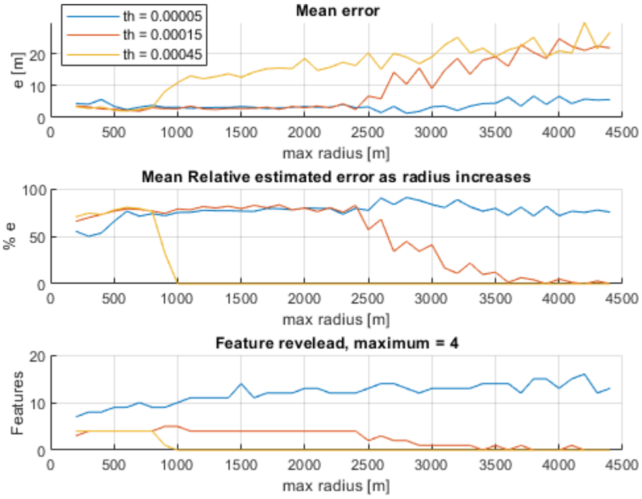


FIGURE 17. Mean error as a function of the radius (top), corrected relative error (middle), and detected features (bottom), obtained with different thresholds.

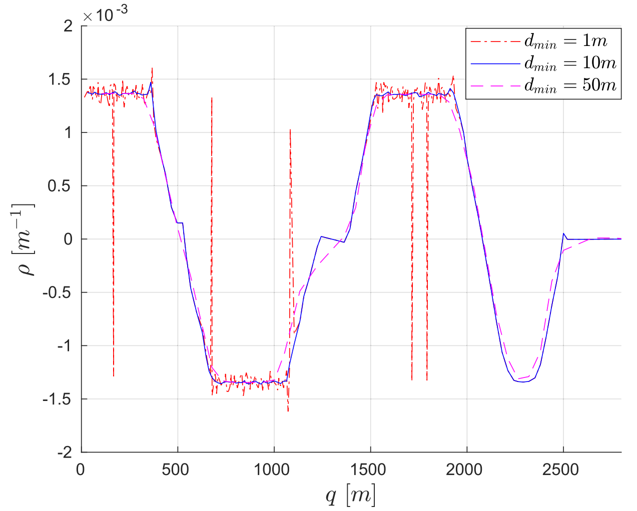


FIGURE 18. Curvature profiles reconstructed with different values of d_{min}.

decreases. Since the filtering is specific for the selected sensor characteristics, the next test focuses only on the threshold. It uses the same setup of the previous test and evaluates the effect of the threshold on the detection quality.

This test was performed to verify whether there exists an optimal threshold value for detecting the correct number of features for any radius within the constraints allowed by tracks construction rules. As expected, the threshold value strongly depends on the radius of curvature of the path. However, if the curve radius is reported in the Digital Map, the onboard platform can read the radius of the next approaching curve and select the best threshold associated with it.

Figure 17 shows the results of the previous experiment repeated with three different threshold values. With a threshold equal to 0.00045 (yellow curve), no feature is detected for most of the allowed range of curvatures and no odometry correction is performed during the train path. A threshold

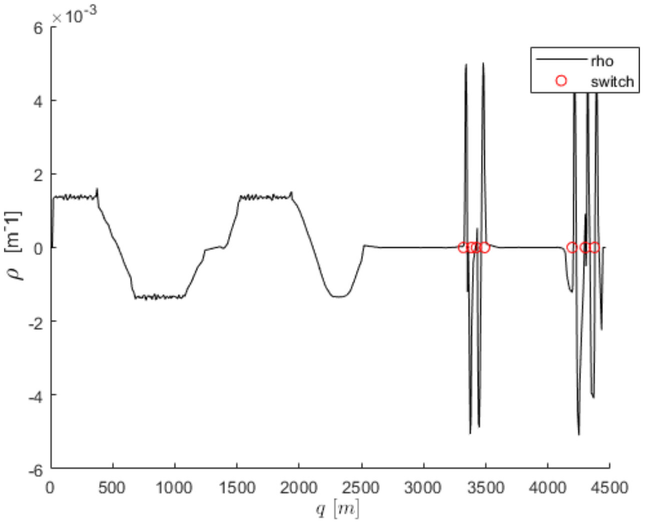


FIGURE 19. Curvature reconstructed on the reference real track.

equal to 0.00015 (orange curve) represents an excellent alternative, since all the features are detected up to a radius of 2500 m, and some fixes are performed even for values over 4000 m. A threshold value of 0.00005 (blue curve) produces inconsistent results, because the number of estimated features is much higher with respect to the real ones. This generates a wrong correction in some points detected as features, because they were associated with some reconstructed features on the moving window in the map.

E. VALIDATION WITH REAL DATA

To evaluate the correctness of the proposed approach, some preliminary validation tests have been performed on real data. The reference track is a short path of around 5000 m concerning the arrival of the train at a station.

Figure 18 shows a portion of three different reconstructed curvature profiles obtained with different values of the minimum distance d_{min} between consecutive points, as described in Section III-B. As evident from the plots, values of d_{min} smaller than 10 m generate curvature errors visible as spikes (dot-dashed red plot), while values higher than 10 m lead to a degradation of the reconstructed curvature profile (dashed pink plot). Hence, in this work, d_{min} = 10 m has been used to reconstruct the curvature.

Figure 19 shows the track curvature reconstructed on the whole traveled path extracted from the map with d_{min} = 10 m. Note that the spikes present in the intervals 3200-3500 m and 4000-4500 m are generated by the switches located on the railway line.

Applying the proposed approach to detect the features on the track produces the results presented in Figure 20, where the curvature (top graph) is shown together with its derivative (bottom graph). Note that the features (indicated by a cross) correctly match the local maxima of the derivative function, and the end of the corresponding window (indicated by a circle) is selected as the first point were the derivative decreases below the chosen threshold. The final part of the available

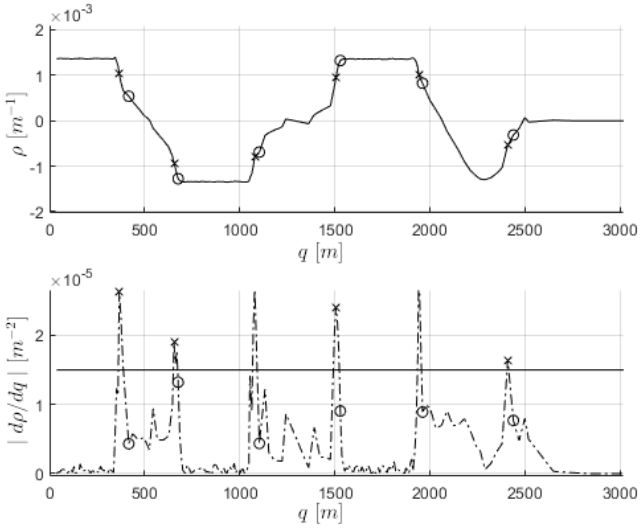


FIGURE 20. Features revealed on the reconstructed curvature of the reference real track.

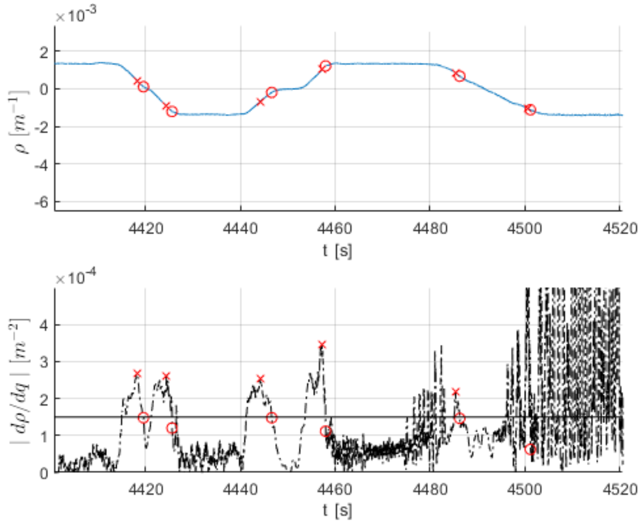


FIGURE 21. Features estimated on the real IMU signal acquired while traveling the reference track.

data was not considered due to the very low speed, requiring a precise knowledge of the sensor noise model, which was not available for the inertial unit used during the measurement campaign.

Finally, Figure 21 shows the curvature derived from the IMU signal following the presented approach, together with its derivative and the revealed features.

Note that the derivative of the estimated curvature is significantly more noisy than the one obtained for the reconstructed one, implying the use of a proper threshold value. However, the number of matched features is bounded by the number of features in the reconstructed track. The choice of the features to be reconstructed is a task that can be performed offline and needs to be repeated only if the railway is subjected to modifications, embedding the results in the Digital Map.

V. CONCLUSION

This paper presented a novel solution to improve the odometry precision in train localization. The proposed method exploits inertial data (i.e., lateral accelerations and yaw angular rotations) to estimate the track curvature and detect a number of track features, which are then matched with those extracted offline from a digital map of the train path, used as virtual position markers. The map features considered for the matching are those found in a moving window centered in the position estimated by the odometry and with a length proportional to the uncertainty bounds. Every time a feature reconstructed from the inertial data matches the corresponding feature stored in the digital map, its position is provided as output.

The experiments carried out on both synthetic and real data showed the effectiveness of the proposed approach when applied to reduce the odometry error, which can be decreased up to 74% over 4500 m stretch by exploiting the auxiliary markers. The results also showed the robustness of the method to noise in the inertial data. The proposed method can be exploited to contain the train position confidence interval without increasing the number of physical balises along the line, thus saving both infrastructure and maintenance costs.

In a future work, we plan to extend the proposed approach to extract additional features, such as switches and slopes, enriching the digital map with a higher number of virtual position markers. Finally, we envisage a more in-depth analysis of real-world data obtained from fully-characterized sensors, to be integrated in a comprehensive localization solution making use of additional sensors, as GNSS and Lidar.

REFERENCES

[1] *Subset 026, Issue 3.6.0, ERTMS/ETCS System Requirements Specification*, UNISIG, Menomonee Falls, WI, USA, 2016.

[2] *Subset 041, Issue 3.2.0, ERTMS/ETCS Performance Requirements for Interoperability*, UNISIG/UNIFE, Brussels, Belgium, 2015.

[3] *The CLUG Project*. Accessed: Feb. 22, 2022. [Online]. Available: http://www.clugproject.eu/en

[4] J. Otegui, A. Bahillo, I. Lopetegi, and L. E. Diez, "A survey of train positioning solutions," *IEEE Sensors J.*, vol. 17, no. 20, pp. 6788–6797, Oct. 2017.

[5] *GNSS User Technology Report–Issue 3*, Eur. Gobal Navigat. Satell. Syst. Agency, GSA, Washington, DC, USA, 2020.

[6] *21E109—Vehicle Locator Concept Architecture*, ERTMS Localisation Working Group, ERTMS Users Group, Brussels, Belgium, 2021.

[7] *Shift2Rail Joint Undertaking*. Accessed: Feb. 22, 2022. [Online]. Available: https://shift2rail.org/

[8] M. Hutchinson, J. Marais, E. Masson, J. Mendizabal, and M. M. Zu Horste, "Precise and reliable localization as a core of railway automation (rail 4.0)," in *Proc. Int. Congr. High-Speed Rail, Technol. Long Term Impacts*, Madrid, Spain, Oct. 2017, p. 10.

[9] E. González, C. Prados, V. Antón, and B. Kennes, "GRAIL-2: Enhanced Odometry based on GNSS," *Proc. Social Behav. Sci.*, vol. 48, pp. 880–887, Jan. 2012.

[10] *Odometria SCMT—Specifica Requisiti Funzionali*, Trenitalia, Rome, Italy, 2002.

[11] *Modello di Frenatura Per SCMT*, Rete Ferroviaria Italiana (RFI), Rome, Italy, 2001.

[12] B. Allotta, V. Colla, and M. Malvezzi, "Train position and speed estimation using wheel velocity measurements," *Proc. School Mech. Eng. F, J. Rail Rapid Transit*, vol. 216, no. 3, pp. 207–225, 2002.

[13] B. Allotta, P. D'Adamio, M. Malvezzi, L. Pugi, A. Ridolfi, and G. Vettori, ''A localization algorithm for railway vehicles,'' in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (IMTC)*, May 2015, pp. 681–686.

[14] *Localisation Working Group (LWG)—Railways Localisation System High Level Users' Requirements*, ERTMS User Group, Brussels, Belgium, 2019.

[15] *Final Report–ERTMS Longer Term Perspective*, European Railway Agency (ERA), Valenciennes, France, 2015.

[16] J. Doppelbauer, ''Command and control 4.0,'' IRSE News, London, U.K., no. 246, Tech. Rep., Jul./Aug. 2018. [Online]. Available: https://webinfo.uk/webdocssl/irse-kbase/ref-viewer.aspx?refno=1530769449&_height=1048&_width=1920 and https://webinfo.uk/webdocssl/irse-kbase/ref-viewer.aspx?RefNo=1558775503&GroupMembers=02%20IRSE%20Presidential%20Prog.04%20IRSE%20NEWS%20-%20all%20papers11%20Cab%20sig%20(inc.%20ETCS/CBTC)12%20Traffic%20ManagementIRSE%20NEWS%20-%202018%20Papers%20*_&document=IRSE%20News%20246%20(Jul-Aug%202018).PDF

[17] Z. Deng, H. Song, H. Huang, Y. Li, and H. Dong, ''Multi-sensor based train localization and data fusion in autonomous train control system,'' in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 5702–5707.

[18] K. Gerlach and C. Rahmig, ''Multi-hypothesis based map-matching algorithm for precise train positioning,'' in *Proc. 12th Int. Conf. Inf. Fusion*, Jul. 2009, pp. 1363–1369.

[19] S. S. Saab, ''A map matching approach for train positioning part I: Development and analysis,'' *IEEE Trans. Veh. Technol.*, vol. 49, no. 2, pp. 467–475, Mar. 2000.

[20] S. S. Saab, ''A map matching approach for train positioning. II. Application and experimentation,'' *IEEE Trans. Veh. Technol.*, vol. 49, no. 2, pp. 476–484, Mar. 2000.

[21] M. Lauer and D. Stein, ''A train localization algorithm for train protection systems of the future,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 970–979, Apr. 2015.

[22] O. Heirich, ''Bayesian train localization with particle filter, loosely coupled GNSS, IMU, and a track map,'' *J. Sensors*, vol. 2016, pp. 1–15, May 2016.

[23] B. Cai and X. Wang, ''Train positioning via integration and fusion of GPS and inertial sensors,'' *WIT Trans. Built Environ.*, vol. 50, pp. 1217–1226, Aug. 2000.

[24] F. Tschopp, T. Schneider, A. W. Palmer, N. Nourani-Vatani, C. Cadena, R. Siegwart, and J. Nieto, ''Experimental comparison of visual-aided odometry methods for rail vehicles,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1815–1822, Apr. 2019.

[25] J. Trehag, P. Handel, and M. Ogren, ''Onboard estimation and classification of a railroad curvature,'' *IEEE Trans. Instrum. Meas.*, vol. 59, no. 3, pp. 653–660, Mar. 2010.

[26] O. Heirich, A. Lehner, P. Robertson, and T. Strang, ''Measurement and analysis of train motion and railway track characteristics with inertial sensors,'' in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 1995–2000.

[27] B. Belabbas, A. Grosch, O. Heirich, A. Lehner, and T. Strang, ''Curvature classification for trains using along-track and cross-track accelerometer and a heading rate gyroscope,'' in *Proc. Eur. Navigat. Conf. (ENC)*, 2013, pp. 1–10.

[28] *EN 50129—Railway Applications—Communication, Signalling and Processing Systems—Safety Related Electronic Systems for Signalling*, CENELEC, Brussels, Belgium, 2019.

[29] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Norwood, MA, USA: Artech House, 2008.

[30] *P62M.0100008.S04.00EN—Track DB Algorithms—Software Requirement Specification*, Ansaldo STS, Genoa, Italy, 2021.

[31] *TCRP Report 155—Track Design Handbook for Light Rail Transit Second Edition*, Transportation Research Board 2012 Executive Committee, Washington, DC, USA, 2012.

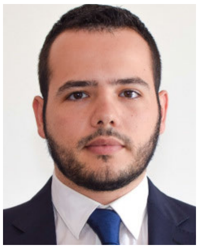
[32] *Manuale di Progettazione d'Armamento*, Rete Ferroviaria Italiana, Rome, Italy, 2019.

[33] F. Böhringer and A. Geistler, ''Location in railway traffic: Generation of a digital map for secure applications,'' in *Proc. WIT Trans. Built Environ.*, vol. 88, Jun. 2006.

[34] G. Taubin, ''Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 11, pp. 1115–1138, Nov. 1991.

MAURO MARINONI (Member, IEEE) received the M.S. and Ph.D. degrees in computer engineering from the University of Pavia, Italy, in 2003 and 2007, respectively. Since 2007, he has been working with the Real-Time Systems Laboratory (ReTiS), where he has been an Assistant Professor, from 2009 to 2020. He is currently a Research Affiliate with Scuola Superiore Sant'Anna, Pisa. He has been a Local Coordinator of the FP7 JUNIPER Project, the Eurostars RETINA Project, and several industrial projects exploiting the ReTiS Laboratory research outcomes in different application fields, from e-Health devices to autonomous and distributed systems.

PIERLUIGI AMATO received the M.S. degree in robotics and automation engineering from Pisa University. His master's thesis titled ''Improving GNSS-Free Train Localization by Integrating Odometry with Inertial Data and Track Information'' was developed at the TeCIP Institute Laboratory, Scuola Superiore Sant'Anna, in collaboration and with the support of Prof. Giorgio Buttazzo and Mauro Marinoni. He is currently working as a Software Engineer in an Italian military defense company.



CARMELO DI FRANCO received the M.S. degree in computer engineering from the University of Pisa, and the Ph.D. degree in embedded systems from Scuola Superiore Sant'Anna, Pisa. He was a Postdoctoral Researcher at the University of Virginia, working on assured autonomy, localization, navigation, and path planning of autonomous systems. He is currently a Research Affiliate with Scuola Superiore Sant'Anna. He is a Robotic Software Engineer at Aitronik, an Italian innovative start-up company that helps companies integrate artificial intelligence systems into mechatronic devices.

SALVATORE SABINA received the M.S. degree in electronic engineering from the University of Pisa, in 1983. From 1985 to 1998, he worked on error codes, satellite TDMA techniques, spread spectrum satellite applications, hardware designs, and HRT operating systems. Since 1999, he has been with Ansaldo Segnalamento Ferroviario (now Hitachi Rail STS) initially as the responsible of the Product Development Department, where ERTMS constituents and components were developed. He is currently the Vice President of the Innovation Department, Hitachi Rail STS. He is also responsible for setting up railway innovative projects and collaborations with research and development organizations.

GIORGIO BUTTAZZO (Fellow, IEEE) received the M.E. degree in electronic engineering from the University of Pisa, the M.S. degree in computer science from the University of Pennsylvania, and the Ph.D. degree in robotics from Scuola Superiore Sant'Anna, Pisa. He is currently a Full Professor of computer engineering with Scuola Superiore Sant'Anna. He has authored seven books on real-time systems and more than 300 articles in the field of real-time systems, robotics, and neural networks. He received 11 best paper awards. In 2013, he received the Outstanding Technical Contributions and Leadership Award from the IEEE Technical Committee on Real-Time Systems. He is the Editor-in-Chief of *Real-Time Systems* and an Associate Editor of the *ACM Transactions on Cyber-Physical Systems*.

Open Access funding provided by 'Scuola Superiore ''S.Anna'' di Studi Universitari e di Perfezionamento' within the CRUI CARE Agreement