University of West Bohemia in Pilsen
Faculty of Applied Sciences
Department of Cybernetics

MASTER'S THESIS

Map-based Train Positioning

PILSEN, 2020 Simona Fricová

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Simona FRICOVÁ**
Osobní číslo:	**A18N0055P**
Studijní program:	**N3918 Aplikované vědy a informatika**
Studijní obor:	**Kybernetika a řídicí technika**
Téma práce:	**Odhad polohy vlaku s využitím mapy**
Zadávající katedra:	**Katedra kybernetiky**

Zásady pro vypracování

- Seznamte se s úlohou dynamického odhadu polohy vlaku s využitím mapy.
- Seznamte se s modely senzorů, které se pro tuto úlohu využívají.
- Seznamte se s estimačními metodami pro tuto úlohu.
- Vybrané metody implementujte a analyzujte jejich chování na simulovaných či reálných datech.

Rozsah diplomové práce: **40-50**
Rozsah grafických prací: **dle potřeby**
Forma zpracování diplomové práce: **tištěná**
Jazyk zpracování: **Angličtina**

Seznam doporučené literatury:

Dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Doc. Ing. Ondřej Straka, Ph.D.**
Katedra kybernetiky

Datum zadání diplomové práce: **1. října 2019**
Termín odevzdání diplomové práce: **25. května 2020**

Západočeská univerzita v Plzni
L.S.
Fakulta aplikovaných věd

Doc. Dr. Ing. Vlasta Radová
děkanka

Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 1. října 2019

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 29. května 2020

. . . . . .

vlastnoruční podpis

Declaration

I declare that I carried out this master's thesis independently, only with the cited sources and literature.

Acknowledgements

To my supervisor doc. Ing. Ondřej Straka, Ph.D.
for his advise and patience
belongs my gratitude.
For without him it would not be
possible to meet any expectations
that this thesis should fulfill
to prove the ability and skill
of the author...

Without the consultations
that were kindly given
by my supervisor and Ing. Ivo Punčochář, Ph.D.,
who showed me, what until then remained hidden,
that I would not find without thee.

Abstract

The thesis topic is the analysis and testing of three chosen methods for the train position dynamic estimation with the usage of track database and sensors, that are usually a part of the standard train equipment. The result of these methods is matching of the measured position to the track database according to a chosen criterion. The chosen methods have the potential utilization in the automation of rail transportation for achieving independence on a local positioning system. The analyzed methods are implemented and tested on simulated data in *MATLAB*. The simulation results are discussed together with the proposed improvements and delimitation of the usage area.

Keywords: state estimation, filtration, localization, map matching

Abstrakt

Tématem této práce je analýza a testování tří vybraných estimačních metod pro dynamický odhad polohy vlaku za použití databáze tratě a senzorů, které jsou již většinou součástí vybavení vlaku. Výsledkem těchto metod je určení bodu či úseku na mapě trati, jenž je podle definovaného kritéria optimálním obrazem naměřené polohy. Zvoleny byly metody, které mají potenciální využití v automatizaci železniční dopravy pro dosažení nezávislosti na lokálním systému určování polohy. Analyzované metody jsou implementovány a testovány na simulovaných datech v prostředí *MATLAB*. Výsledky simulací jsou diskutovány spolu s návrhy na vylepšení nebo vymezením podmínek pro jejich využití.

Klíčová slova: odhad stavu, filtrace, lokalizace, přiřazení na mapu

Contents

List of Figures

Nomenclature

Abbreviations

CKF	Cubature Kalman Filter
DR	Dead Reckoning
EKF	Extended Kalman Filter
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
INS	Inertial Navigation System
MEMS	Microelectromechanical Systems
MSE	Means square error

Defined functions

$d^{odo}(t_s, t_e)$	distance measured by the odometer between the time instants t_s t_e
$dif(arg_1, arg_2)$	evaluation of difference or distance of arg_1, arg_2

Variables

δ	roll angle
Δ_{Ψ}	yaw rate offset
$\Delta_{\dot{\theta}}$	pitch rate offset
$\dot{\Psi}$	yaw rate
$\dot{\Psi}_{gyro}$	yaw rate measurement from gyroscope

$\dot{\theta}$	pitch rate
$\dot{\theta}_{gyro}$	pitch rate measurement from gyroscope
ϵ_i	sum of all random errors for the i-th satellite
$\hat{\omega}$	measured yaw rate
$\hat{\rho}$	similarity function between fusion position and candidate model
$\hat{x}$	filtration state estimate
$\hat{x}'$	prediction state estimate
λ	longitude
$\overline{\omega}$	map curvature
$\overline{v}_{odo}$	averaged speed from odometer
ϕ	latitude
Ψ	yaw (heading) angle
Ψ_{GNSS}	heading computed from GNSS
ρ	pseudorange
a	acceleration
d	distance
f	state equation
h	measurement equation
P	point of a line
P′	predictive density function covariance matrix
$\mathbf{P}(n)$	set of points for map matching defined by $[x_p, y_p]$
$\mathbf{P}_{map_s}, \mathbf{P}_{map_e}$	starting and ending point of a dataset for map matching
$\mathbf{P}_{match}$	matched point of map abscissa
Q	covariance matrix of state noise

$\mathbf{R}$	covariance matrix of measurement noise
$\mathbf{R}_f$	train position defined by $[x_f, y_f, l_l]$
$\mathbf{R}_i$	position of the i^{th} satellite
$\mathbf{R}_u$	user's position
$\mathbf{R}_{um}$	measurement of $\mathbf{R}_u$
$\mathbf{v}$	measurement noise
$\mathbf{v}$	velocity
$\mathbf{x}$	state vector
$\mathbf{z}$	measurement vector
θ	pitch angle
$\tilde{\omega}$	measured yaw rate transformed into curvature
$\tilde{\omega}^{avg}$	$\tilde{\omega}$ averaged within the segment borders
b_u	receiver clock bias
c	speed of light in vacuum
F	Jacobian matrix of state equations
H	Jacobian matrix of measurement equations
h	height (altitude)
h, h_0	bandwidth
S	scale error
t	time
$w_1 ... w_{11}$	state noise
$[E, N, U]$	[East, North, Up] coordinates
$[E_{GNSS}, N_{GNSS}, U_{GNSS}]$	East, North, Up position from the GNSS

$\mathcal{C} = \{\mathbf{C}^1_{map}, \mathbf{C}^2_{map}, ..., \mathbf{C}^K_{map}\}$ set of map curves

$\mathcal{L} = \{\mathbf{L}^1_{map}, \mathbf{L}^2_{map}, ..., \mathbf{L}^M_{map}\}$ set of map segments

$\mathcal{R} = \{\mathbf{R}^1_{map}, \mathbf{R}^2_{map}, ..., \mathbf{R}^N_{map}\}$ set of map points

Chapter 1

Introduction

In recent years automatic train positioning has become more and more required. The most relevant factors are safety and efficiency. However other, for example financial, reasons like potential lower personnel expenses, can decide in favor of automatic positioning. Possible facilitation of the management of arrivals/departures can save a lot of money and avoid possibly dangerous mistakes. Railroad modernization also brings higher comfort to passengers including, besides other things, applications that inform travelers about the train location. As for the safety reasons, the goal is to create a system that can send information to railroad level crossing or detect multiple vehicles on the railroad and thus prevent train collisions.

The current train positioning system uses so-called on track balizes, which are beacons with a known position [19]. The train, when running over a balize, reads the position information (as a telegram).

When considering other train positioning possibilities, the train position, also referred to as user's (receiver) position, can be estimated using various sensors for example the GNSS [8], with the absolute position on the output, or sensors of relative position which are called dead reckoning (DR) sensors and contain the accelerometer [8, 16], gyroscope [7], odometer [16] and Doppler radar [11].

All these mentioned sensors, no matter whether absolute or relative, suffer from various errors. The GNSS position estimates can be affected by the propagation through the atmosphere which has variable properties, or by other factors such as the surrounding of the receiver by high buildings, trees, etc., that can as well seriously influence the quality of estimates [8]. The relative sensors are not affected by the vehicle surrounding,

but suffer from errors as well. Increasing inaccuracies emerge in the DR estimates over time due to the accumulating errors, therefore these sensors cannot work for a long time without recalibration.

Due to limited reliability, the stand-alone GNSS cannot be used in critical applications. Similarily due to limited reliability DR sensors cannot be used as the sole source for critical decisions. One possible solution is to support the GNSS estimates by DR calculations especially during the time when the GNSS signal is not available. However, the need for frequent recalibration of the DR sensors, which is mostly done by the GNSS, can be a problem when the GNSS signal is shadowed or of low quality.

The method of matching the position estimate provided by the GNSS or DR to a map of the road/railroad may solve the problem. In this promising approach maps are used to help with determining the position of the receiver [20]. The principle of this method is the projection of the estimate onto a road/railroad on a map. In the case of train positioning, the location of the vehicle is constrained to the railroad. Estimates that are far from any railroad or do not make sense in the context of the previous estimates can be removed.

The vehicle dynamics can be used to constrain the train state estimate as well, and thus produce more accurate information about its position.

The thesis aim is to find, analyze and test algorithms using low-cost on-board equipment and a map database, that have the potential to replace the current system based on on-track balizes. Methods described in [17, 6, 12] will be examined and tested on simulated data, provided by Ing. Ivo Punčochář, Ph.D. This data includes simulated GNSS, odometer, accelerometer, and gyroscope measurements. The main advantage of using these sensors would be independence from specific railroad equipment e.g. balizes.

The thesis is structured as follows. The sensors used for estimating the train position are described in Chapter 2 together with a brief description of the used coordinate systems and transformation between them. Map matching, as well as the mathematical representation of the track and criteria for the algorithm, are presented in Chapter 3. In Chapter 4 different methods of map matching are mentioned with the necessary mathematical background. Three of these methods are tested and evaluated in
Chapter 5.

Chapter 2

Sensors

In this chapter, the train position in different coordinate systems, train velocity and attitude in three-dimensional space will be specified, followed by the description of sensors available on board, possibly used for determining train position, velocity and attitude in three-dimensional space.

The train position can be described in a coordinate system (described further in the next section) using the *absolute* or *relative* position. The *absolute* user's position $\mathbf{R}_u$ in three-dimensional Euclidean space is determined by three coordinates, depending on the used coordinate system. The *relative* position can be characterized by the distance **d**, velocity **v**, acceleration **a** or the yaw (heading) Ψ, roll δ, and pitch θ angles representing an attitude (Figure 2.1).

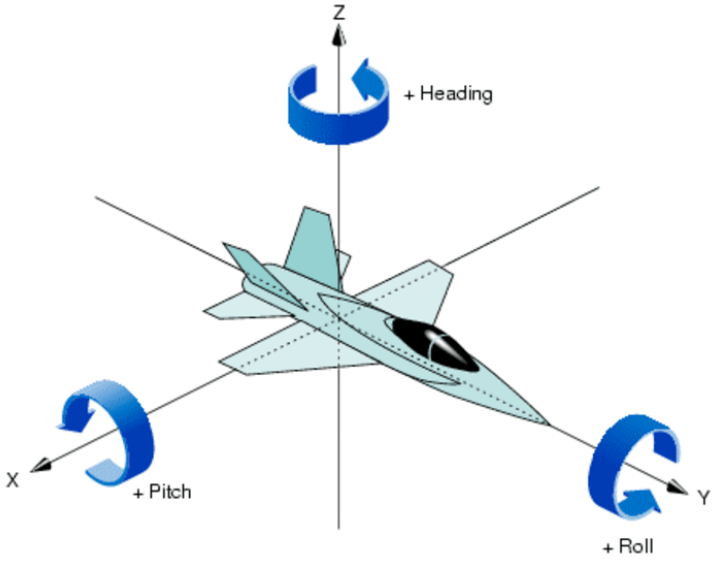


Figure 2.1: Gyroscope measurement axis [15]

For determining the position of a user we need to specify the coordinate system. The systems used for vehicle localization in this thesis are the Earth-Centred, Earth-Fixed (ECEF) and the local East, North, Up (ENU) coordinate systems [22].

Geodetic Coordinate System

The Geodetic coordinate system is the system mostly used for orientation in a map. The Earth's surface (or close above) is described by latitude, longitude, and height [22]. The **latitude** (ϕ) specifies the angle between the equatorial plane and the line to the measured point. The **longitude** (λ) measures the angle between the prime meridian and the point. The **height**/altitude (h) is the distance of the measured point above the reference ellipsoid. The user's position is described as $\mathbf{R}_u^{Geod} = [\phi, \lambda, h]$. Latitude and longitude are shown in Fig. 2.2.

ECEF Coordinate System

The ECEF system has its origin in the Earth's center and rotates around its spin axis. The x-axis intersects the sphere in the crossing of the equator and the prime meridian [22], [18]. The y-axis is orthogonal to the x-axis and also intersects the equator. The z-axis is along the spin axis, orthogonal to the x and y-axis. The user's position in ECEF is $\mathbf{R}_u^{ECEF} = [x, y, z]$. The relation between x, y, z-axis, and the ϕ, λ coordinates are shown in Figure 2.2.

ENU Coordinate System

The ENU system is a Cartesian coordinates system on the Earth's surface. The North and East axis lay in a plane that is normal to the ellipsoid model [22]. The up-direction is orthogonal to this plane. The user's position in ENU is described as $\mathbf{R}_u^{ENU} = [E, N, U]$. The East, North and up vectors are shown in Fig. 2.2.

Henceforth $\mathbf{R}_u$ will be used for the user position $\mathbf{R}_u^{ENU}$.

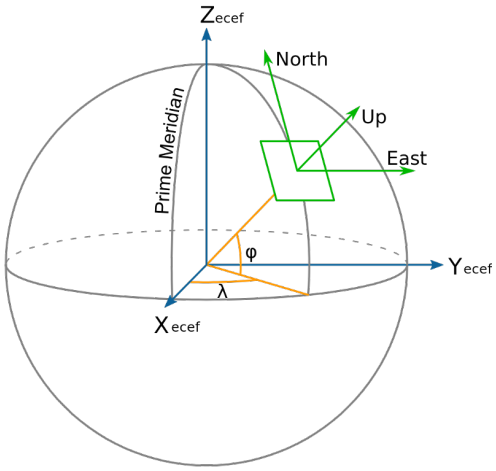


Figure 2.2: Coordinate Systems [5]

Transformation from ECEF to ENU

The transformation of a point from the ECEF to ENU coordinate system is done using the Geodetic Coordinate System by the following equation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -sin(\lambda) & cos(\lambda) & 0 \\ -sin(\phi)cos(\lambda) & -sin(\phi)sin(\lambda) & cos(\phi) \\ -cos(\phi)cos(\lambda) & cos(\phi)sin(\lambda) & sin(\phi) \end{bmatrix} \begin{bmatrix} x\text{–}N \cdot cos(\phi)cos(\lambda) \\ y\text{–}N \cdot cos(\phi)sin(\lambda) \\ z\text{–}N(1-e^2)sin(\phi) \end{bmatrix}, \tag{2.1}$$

where $N = \frac{a}{1-e^2 sin^2(\phi)}$ stands for the radius of curvature in the vertical plane normal to the astronomical meridian, a is the semi-major axis and e is the eccentricity [22], [18].

2.1 GNSS

Global Navigation Satellite System (GNSS) is a service for determining the three-dimensional position of a receiver [1]. It is a well-known technology used for positioning and navigation.

2.1.1 GNSS Constellations

There are currently five GNSS constellations. The best-known GNSS constellation is certainly the Global Positioning System (GPS) owned by the United States of America. This system, built and operated by the U.S. Air Force was originally designed for the military but was later made available for public use [8]. The specified accuracy of civil GPS is 9 meters in the horizontal and 15 meters in the vertical direction. The European satellite system (GALILEO) even claims to provide its users by position estimations with greater precision than offered by any other available systems [3]. Other global powers like China or the Russian Federation have their own satellite system (BeiDou, GLONASS) which grants at least partial independence on the American GPS [8].

2.1.2 GNSS structure

GNSS is composed of three segments. The *space segment* consists of a constellation of 20 to 30 satellites that orbit around the Earth in three to six different orbital planes [8]. The *control center* listens to the broadcast signals in several different locations and uses this information to compute the exact orbits and clock drift (further discussed later in this section) corrections for each satellite. This information is then sent back to the satellites and broadcast as a part of the navigation message. The *user segment* includes all receivers of the GNSS signal.

The GNSS calculates the user position from the time it takes the signal from the satellites to reach the receiver. This time interval is computed as $\Delta t_i = t_i - t_u^{ECEF}$, where t_i is the time when the i^{th} satellite broadcast the signal, $i \in \{1, 2, ..., N\}$ is the index of a specific satellite and N is the total number of the available satellites.

The time t_u^{ECEF}, when the signal reaches the receiver, is then used to compute the so-called pseudorange, which is the diameter of a sphere from the satellite running through R_u^{ECEF} [18]. The pseudorange of the i^{th} satellite $\boldsymbol{\rho}_i$ is counted as

$$\rho_i = c\Delta t_i, \tag{2.2}$$

where $c \approx 3 \cdot 10^8$ is the speed of light in vacuum.

However, in the real world, various errors have a nonnegligible effect of the pseudorange estimate. That is why the equation 2.3 is used for the real pseudorange characterization.

$$\boldsymbol{\rho}_i = |\mathbf{R}_u^{ECEF} - \mathbf{R}_i| + cb_u + \boldsymbol{\epsilon_i}, \tag{2.3}$$

where $\mathbf{R}_i$ is the i^{th} satellite position, b_u is the receiver clock bias, that affects all the computed pseudoranges equally, and $\boldsymbol{\epsilon_i}$ is the sum of all random errors for the i-th satellite [8].

The pseudorange between each satellite and the receiver determines the radius of a sphere around each of the satellites. The intersection of two spheres defines a circular region and adding the third sphere gives us two points in one of which the receiver is located. These points are marked as yellow dots in Fig. 2.3. The point closer to the Earth's surface is chosen as the estimated receiver position.

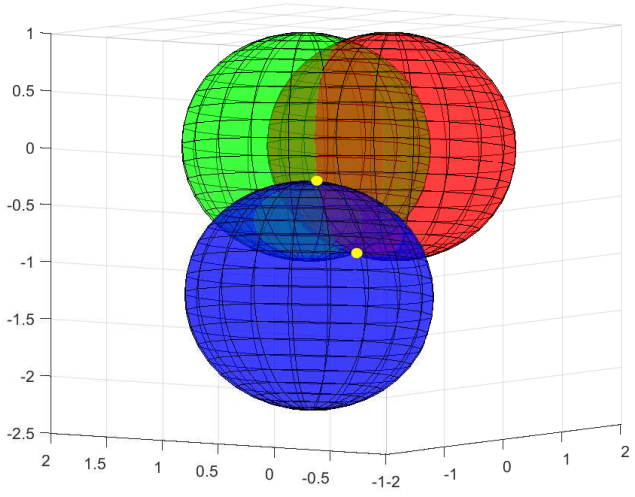


Figure 2.3: Intersection of three spheres

Despite the fact that only three coordinates (x, y, z) are estimated, at least four satellites are needed for this task. The reason for this is the absence of an extremely precise atomic clock in the receivers, thus the *local time* of a receiver is considered as a fourth unknown.

Although in theory the position can be counted with as many as four satellites, in reality there are many more satellites in a constellation, which increases the probability of at least four satellites being visible at every time all over the globe and gives us more satellites to choose from. For example GPS operates with a 27 satellite constellation [2].

The redundancy of satellites in the GNSS provides the possibility of choosing those, which are trusted the most. Also greater number of satellites can be used for achieving higher precision of estimates.

2.1.3 GNSS drawbacks

There are many factors that can limit the GNSS accuracy and availability. One of the most important factors is the error due to signal passing through the Earth atmosphere, especially the Troposphere and Ionosphere. This inhomogeneity of the environment, that the signal travels through, may add error to the time delay between the transmission and receiving of the signal. No less serious situation arises due to the absence of the signal when the receiver is passing through a tunnel or other physical barrier. The interference by multipath, when, due to the reflection from surrounding objects, multiple signals from one satellite arrive with different time delays can as well cause distorted estimates [21].

In most cases, choosing satellites above some elevation mask (usually of 5°) is used to eliminate the firstly mentioned error. The elevation mask is the minimum elevation angle above the horizon that the satellites must have to be trusted to give an undistorted estimate. In case of the receiver riding through a tunnel, the GNSS signal is shadowed and the position cannot be updated until the receiver comes out of the barrier.

Some of the drawbacks of GNSS can be corrected by using the space-based augmentation system (SBAS). This geosynchronous satellite system uses data from several reference points over the continent, transfers all of the measurement errors to a computing center, where the corrections of the atmospheric delays are calculated and broadcast by a network to all receivers which can use them to compute their own differential corrections.

2.2 Inertial Navigation System (INS)

The Inertial Navigation System uses sensors of the Inertial Measurement Unit (IMU), consisting of accelerometers and gyroscopes for determining the user's position by dead reckoning. These sensors do not receive any external broadcast signal and thus are available regardless of their surroundings.

The main technology used in accelerometers and gyroscopes is the microelectromechanical systems (MEMS), which uses the principle of mass on a spring. The specific application is further described in the following subsections. A great advantage of MEMS technology is its low-cost. The disadvantages are biased estimates, noise, and the need to calibrate often.

2.2.1 Accelerometers

Accelerometers are sensors that measure the acceleration of the vehicle in a specific axis [8]. Acceleration describes the force interactions, due to the change of velocity, between objects with mass (Figure 2.4). When the measured object moves (along the x-axis in the Figure) with an unsteady velocity, the mass in the accelerometer tends to remain stationary due to its inertia and therefore the spring is stretched or compressed. This deformation of the spring induces capacitance that corresponds to the object's acceleration.

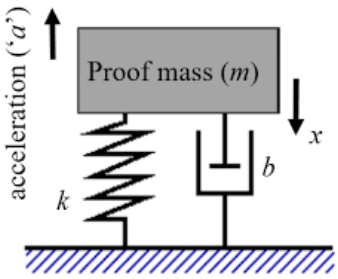


Figure 2.4: The principle of accelerometer [14]

For the purpose of object positioning, acceleration can be used to determine the change of velocity, the current velocity or position

$$\Delta \mathbf{v} = \int_{t_1}^{t_2} \mathbf{a}(t)\, dt, \tag{2.4}$$

$\Delta \mathbf{v}$ is the change in velocity between time steps t_1 and t_2 and $\mathbf{a}$ is the acceleration measured by the accelerometer.

By integrating the change in velocity $\Delta \mathbf{v}$ over the time period $< t_1, t_2 >$ we obtain the distance ($\mathbf{d}$) traveled over this time interval.

$$\mathbf{d} = \int_{t_1}^{t_2} \mathbf{v}(t)\, dt. \tag{2.5}$$

2.2.2 Gyroscopes

Gyroscopes are sensors that measure the rotation rate about an axis [7]. It is possible to measure the roll, pitch, and yaw (heading) rate (shown in Figure 2.1). Gyroscopes are rotating physical devices, that tend to maintain the orientation of their rotation axis due to the law of angular momentum conservation and measure the angular velocity of a body around a generic axis with respect to the initial reference [9].

Similarly to accelerometers, gyroscopes are mostly constructed by MEMS technology and use the Coriolis force that causes mass on a spring to displace from its original position [9]. Figure 2.5 shows the gyroscope principle. The two masses of the system are joined by means of an elastic element. If one of the masses is moving in the direction of the X-axis with velocity V_X, then applying the rotation Ω_z about the Z-axis results in the mass experiencing the Coriolis force $F_{Coriolis}$ in the direction of the blue arrow. The other mass will experience force of the same magnitude but opposite direction. This force causes changes in the piezoresistive material structure that can be transduced to an electrical signal.

For train positioning the main focus is on the heading angle Ψ and the pitch angle θ.

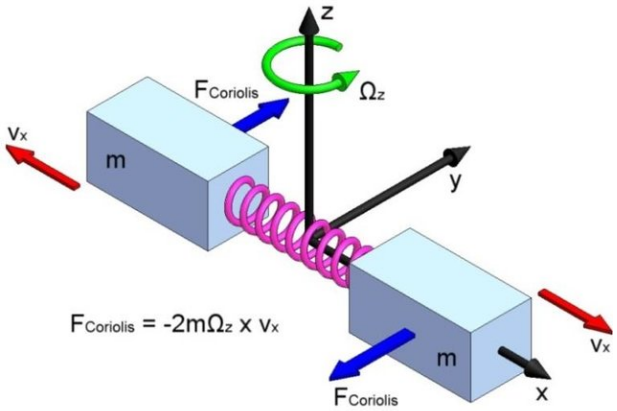


Figure 2.5: The principle of gyroscope [9]

2.3 Other sensors

2.3.1 Odometers

Odometers are a part of the dead reckoning (DR) system. Odometers calculate the traveled distance by counting the turns of the wheels presuming a constant wheel radius (and thus circumference). Because it would not be sufficient to only count the whole

360° wheel turns, there are multiple marks around the wheel placed over a constant angle so that it is possible to record smaller traveled distance. A pulse is generated whenever the wheel turns over the mark. The speed is calculated by dividing distance traveled during the sampling time interval T by the duration of T. This is shown in the following equations

$$d_i = p_i \cdot \alpha \cdot r \tag{2.6}$$

$$v_i = \frac{d_i}{T}, \tag{2.7}$$

where d_i is the distance traveled during the i^{th} sampling period, p_i is the number of marks that were read by the odometer, α is the angle in radians between two marks, r is the wheel radius and v_i is the average speed during T.

The main error affecting the odometer estimate is when the wheel slips, meaning it travels some distance without turning. This happens mainly when the vehicle starts rolling or is breaking. This error gets larger with the traveled distance which is the reason why odometers need to be calibrated. In the current railroad system, calibration is done when running over a balize with a known position [4].

A balize is an electronic transmission device (beacon) placed between the rails of a railroad. Balizes can send telegrams to an on-board subsystem passing over it. The odometer is recalibrated when passing over a balize. This system could possibly be substituted by the GNSS position estimates for achieving the independence on a system, that is different in some other countries, and avoid the risk of physical balizes being damaged or stolen.

2.3.2 Doppler radar

The Doppler radar, mounted on the bottom of the locomotive, measures the speed over ground [11]. The output of the radar is a rectangular signal with frequency proportional to the vehicle speed. Because the sensor calculates the speed by counting leading edges of the signal over the sampling period, the measurement is available only if the train speed exceeds a certain limit.

Chapter 3

Map Matching

3.1 Motivation

Due to the fact that the GNSS signal can be distorted by various factors and the DR sensors accumulate errors when running for a long time without recalibration, a possible solution is to couple these two systems in order to improve accuracy and availability, or match the estimated position to a map database. This matched position can provide calibration data for DR sensors.

3.2 Map description

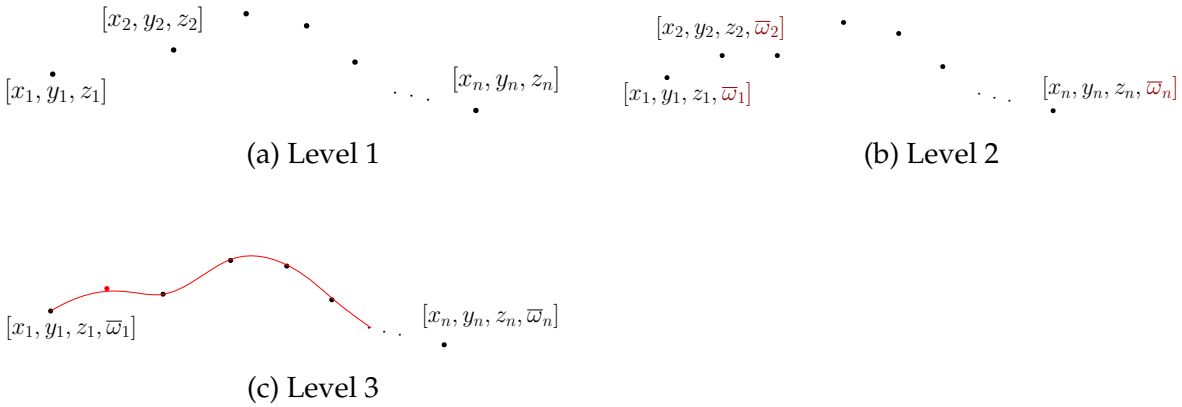
A railroad track database, containing the reference information for the train positions, is needed for the train map matching. The matching techniques differ by map models and the optimization criterion, as will be described further in this section.

There are several levels on which the railroad map database can be modeled. On **level 1**, it is operated only with the discrete measured points [12]. **Level 2** takes the points from level 1 and supplements them by important features (such as curvatures from GNSS measurement or speed limits). **Level 3** uses the points from level 2 and interpolates them with e.g, Cubic spline function, defining a continuous track map. These three approaches are illustrated in Figure 3.1, where each point is represented in 3D space by the $[x, y, z]$ coordinates and in levels 2 and 3 also with the curvature $\overline{\omega}$.

The database is a set of points $\mathcal{R} = \{\mathbf{R}^1_{map}, \mathbf{R}^2_{map}, ..., \mathbf{R}^N_{map}\}$, segments $\mathcal{L} = \{\mathbf{L}^1_{map}, \mathbf{L}^2_{map}, ..., \mathbf{L}^M_{map}\}$ or curves $\mathcal{C} = \{\mathbf{C}^1_{map}, \mathbf{C}^2_{map}, ..., \mathbf{C}^K_{map}\}$ describing the real

world environment. Letters N, M, K represent the total number of points, segments, curves. The points can be characterized by three or two coordinates (depending on if the 2D projection is used), the segment is the track between two points. These can be characterized by an abscissa with the map points as endpoints or a spline.

Curves are segments (specified by end points) supplemented by the segment curvature described by the yaw and pitch angle.



(a) Level 1

(b) Level 2

(c) Level 3

Figure 3.1: Different levels

3.3 Map Matching

Map matching is the process of taking the measured position and matching it to a road/track database minimizing some criterion [6]. There are several possible approaches to this task. The **point-to-point** algorithm chooses the point on the road/railroad (level 1 or level 2) with the smallest distance from the measured point, as shown in the following equation

$$\mathbf{R}^*_{map}(\mathbf{R}_{um}) = \underset{\mathbf{R}_{map} \in \mathcal{R}}{\operatorname{argmin}} ||\mathbf{R}_{um} - \mathbf{R}_{map}||_2, \tag{3.1}$$

where $\mathbf{R}^*_{map}$ is the selected map point, $\mathbf{R}_{um}$ is the estimate of $\mathbf{R}_u$ from the sensor measurement and $\mathbf{R}_{map}$ are points in the map database. Point-to-point matching is illustrated in Figure 3.2, where the map points $\mathbf{R}^n_{map}$ are marked by green circles, the red circle shows the position of the measured point $\mathbf{R}_{um}$, the gray lines, connecting $\mathbf{R}_{um}$ with $\mathbf{R}^n_{map}$ symbolize the measuring the distance and the orange line connects the measured point with the closest map point.

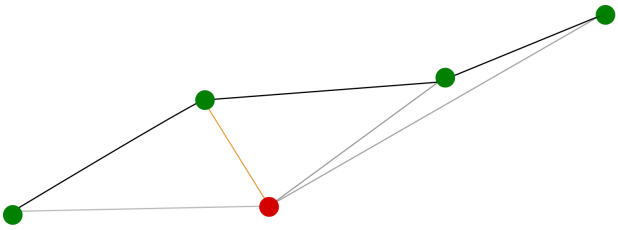


Figure 3.2: Point-to-point matching

Because these points ($\mathbf{R}_{map}$) are just a chosen subset of points on the track, it is natural to somehow connect them into a continuous line and find the segment that is closest to the measured point. This is the **point-to-curve** approach. For example the orthogonal distance from the measured point to the segment of the map DB is counted and the projected point **P** from the closest railroad segment is selected. The nearest segment is calculated according to equation 3.2

$$\mathbf{L}^*_{map}(\mathbf{R}_{um}) = \underset{\mathbf{L}_{map}(i)\in\mathcal{L}}{\operatorname{argmin}} \left(\min_{\mathbf{P}\in\mathbf{L}_{map}(i)} ||\mathbf{R}_{um} - \mathbf{P}||_2 \right) \tag{3.2}$$

where $\mathbf{L}_{map}(i)$ is a specific curve segment from the database, **P** are points in this segment and $\mathbf{L}^*_{map}(\mathbf{R}_{um})$ is the selected segment. Point-to-curve matching is shown in Figure 3.3, where the orthogonal lines from $\mathbf{R}_{um}$ to the abscissas symbolize the distance between the point and abscissa, the line to the closest segment is marked yellow and **P** is marked by a cross.

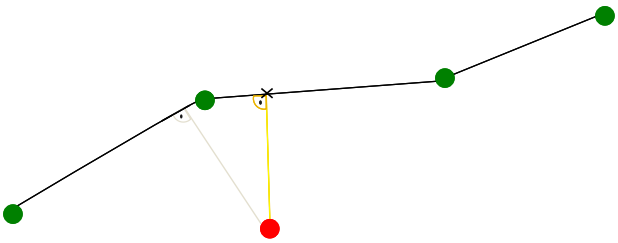


Figure 3.3: Point-to-curve matching

A method for calculating the distance from the measured position $\mathbf{R}_{um}$ to an abscissa for the point-to-curve matching is presented in section 3.3.1.

The third approach in map matching is the **curve-to-curve** matching. This technique takes a set of several measured data and matches them (the points position or the curve

shape, angles...) to the map database looking for the segment with minimal distance (or difference in case of comparing other than position parameters), as shown in the following equation

$$j^* = \underset{j}{\operatorname{argmin}}(dif(\tilde{\omega}^{k:k+N}, \overline{\omega}^{j:j+N})), \tag{3.3}$$

where $dif(arg_1, arg_2)$ is a function evaluating the difference or distance of the arguments, here the difference between the two vectors of ω in some sense. The measured yaw rate $\hat{\omega}[rad/s]$ is transformed from time-referenced into distance-referenced data $\tilde{\omega}[rad/m]$ (further described in 4.1) so that it can be compared to the track curvature. The track curvature $\overline{\omega}$ is extracted from the map database, N is the number of compared transformed yaw rates, j is the starting index of the compared subset of transformed yaw rates and j^* is the index of j with the best match.

Now let us introduce two methods for finding the index j^* from the equation (3.3). Let $I_{\tilde{\omega}}(n) = [x(n), y(n), \tilde{\omega}(n)]^T = [\mathbf{R}_{um}(n)^T, \tilde{\omega}(n)]^T$ be the n^{th} measured point specified by the x, y coordinate and yaw rate transformed to distance-referenced curvatures $\tilde{\omega}$ (section 4.1). This example is from a 2D projection. Let $\{\tilde{\omega}^k, \tilde{\omega}^{k+1}, ..., \tilde{\omega}^{k+n}\}$ be a subset of $n+1$ measured $\hat{\omega}$ transformed into $\tilde{\omega}$, where $\tilde{\omega}^{k+n}$ is the actual measured yaw rate transformed into a distance-varying variable and $\tilde{\omega}^k$ is the transformed yaw rate form n steps ago. The task is to find the corresponding subset $\{\overline{\omega}^i, \overline{\omega}^{i+1}, ..., \overline{\omega}^{i+n}\}$ from the set $\Omega = \{\overline{\omega}^j\}_{j=1}^{J}$, where $J \geq n$ and Ω is a set of curvatures from the map database. Two possibilities of finding the optimal $\overline{\omega}$ for each $\tilde{\omega}$ are illustrated by the following equations

$$j^* = \underset{j}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{l=k}^{k+n} \frac{\overline{\omega}^{j+l-k} \tilde{\omega}^l}{||\overline{\omega}^{j+l-k} \tilde{\omega}^l||_2} \right), \tag{3.4}$$

$$j^* = \underset{j}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{l=k}^{k+n} |\overline{\omega}^{j+l-k} - \tilde{\omega}^l| \right), \tag{3.5}$$

where k is the starting index of a compared set of n points and j is the starting point of the map subset of compared n points. Then j^* is the index of the best suiting starting point. The reason for 2 norm in the denominator in equation (3.4) is to suppress the role of large $\overline{\omega}$ values and thus avoid matching every measured subset $\{\tilde{\omega}^l\}_{l=k}^{k+N}$ to the subset

of map segments with the largest values. The equation (3.5) shows the computation of minimum absolute averaged error (MAAE). The curve-to-curve matching using the index j^* than becomes

$$\mathbf{C}^*_{map}(\mathbf{I}_{\tilde{\omega}}) = \mathbf{C}_{map}(j^*). \tag{3.6}$$

Note that the curvature of the map can be determined as $\kappa(p) = \left\| \left(\mathbf{I} - \frac{\dot{\mathbf{c}}(p)\dot{\mathbf{c}}(p)^T}{||\dot{\mathbf{c}}(p)||_2^2} \right) \cdot \frac{\ddot{\mathbf{c}}(p)}{||\dot{\mathbf{c}}(p)||_2^2} \right\|$, where $\mathbf{c}$ is the parametrization of a regular curve $\mathcal{P} \mapsto \mathbb{R}^3$, where $\mathcal{P} = [p_s, p_e] \subset \mathbb{R}$.

The three approaches (point-to-point, point-to-curve, curve-to-curve) differ by accuracy, computational complexity, and also fields of application.

Not surprisingly, the point-to-point method has the lowest computational complexity as the only task is to calculate distances from $\mathbf{R}_{um}$ to the map points and pick the smallest. The drawbacks of this method are straightforward. In case of the track consisting of abscissas of length L, than the average difference between the result from point-to-point and point-to-curve approach is $\frac{L}{4}$. An illustration can be seen in Figure 3.4. The train is moving between $\mathbf{R}_{map}(j)$ and $\mathbf{R}_{map}(j+1)$ with an almost constant speed, therefore the probability of $\mathbf{R}_{um}$ being matched (point-to-curve according to minimal distance) to any point of the abscissa is equal for all the points. This means that the train position has a uniform probability distribution. The biggest difference between the position matched by these methods is $\frac{L}{2}$. The places that have the average distance from $\mathbf{R}_{map}(j)$ and $\mathbf{R}_{map}(j+1)$ are marked by green points.

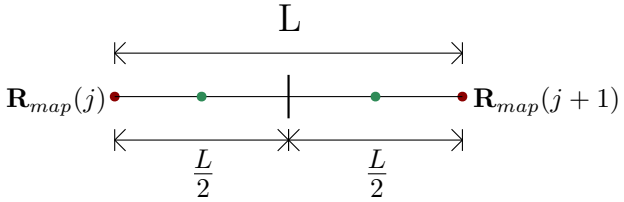


Figure 3.4: The average error

A problem can occur in case that there are multiple railroads close to each other. The solution to this problem could be taking the train dynamics into account.

The point-to-curve approach, in its simplest application of matching $\mathbf{R}_{um}$ to the nearest point of the abscissa, has higher complexity than the previously analyzed method. Finding the nearest point out of all abscissas would demand high computational power

and would be very ineffective. Instead, it is possible to restrict the number of segments for projection by calculating the point-to-point method first and trying to project $\mathbf{R}_{um}$ to several closest segments. Another way to reduce the computational complexity of the algorithm is to only consider segments among a defined area around the lastly matched segment.

The advantage of this method, compared to the point-to-point approach, is the assumption of a continuous map (instead of discrete points) and thus higher accuracy. A problematic situation can occur where multiple routes meet, e.g. on railroad crossings, as can be seen in figure 3.5.

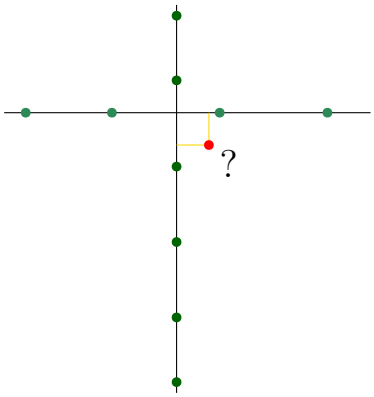


Figure 3.5: Problem on railroad crossings

The curve-to-curve approach is more complex than the two mentioned previously. The definition of segments difference itself can be a challenge. Multiple segments are compared at once, thus the number of compared pairs is the number of map segments J minus the number of compared transformed yaw rates minus 1 i.e. $(n+1-1=n)$ multiplied by the number of $\tilde{\omega}$ compared at each iteration $n+1$, that is $(J-n)\cdot(n+1)$. It is possible and perhaps even necessary to take the train dynamics into account for the sake of maintaining the computational complexity within reasonable limits. When the history of the train position and orientation is used, more accurate results can be expected. The problem illustrated in Figure 3.5 is solved by this method as soon as the train reads the first data after the crossing.

3.3.1 Point to line distance calculation

The point-to-curve map matching is mostly done by selecting the segment with the least orthogonal distance from the filtered point. Let $\mathbf{R}_{um} = [R_{um_x}, R_{um_y}]$ be the receiver position, $\mathbf{R}^i_{map} = [R^i_{map_x}, R^i_{map_y}]$ and $\mathbf{R}^{i+1}_{map} = [R^{i+1}_{map_x}, R^{i+1}_{map_y}]$ the end points of the i^{th}

abscissa, and $\mathbf{P} = [P_x, P_y]$ the point of the orthogonal intersection from $\mathbf{R}_{um}$ to the line t containing the abscissa defined by points $\mathbf{R}^i_{map}$ and $\mathbf{R}^{i+1}_{map}$. This situation is illustrated in Figure 3.6. The general equation of a straight line can be written as

$$y = kx + q, \tag{3.7}$$

where x and y are the coordinates of any point of the line, k is the slope and q determines the value when the line intersects the y axis. Then the directional vector $\mathbf{u}$ from $\mathbf{R}^i_{map}$ to $\mathbf{R}^{i+1}_{map}$ is defined as

$$\mathbf{u} = [R^{i+1}_{map_x} - R^i_{map_x}, R^{i+1}_{map_y} - R^i_{map_y}]^T \tag{3.8}$$

An auxiliary point $\overline{\mathbf{R}}_{um}$ is defined as $[R_{um_x}, R_{um_y} - q]^T$.

The orthogonal projection of the point $\mathbf{R}_{um}$ onto a line defined by a projection matrix as shown in the following equation

$$\mathbf{P} = \mathbf{u}(\mathbf{u}^T\mathbf{u})^{-1}\mathbf{u}^T\overline{\mathbf{R}}_{um} + [0,\ q]^T. \tag{3.9}$$

The orthogonal distance between $\mathbf{R}_{um}$ and the i^{th} abscissa is equivalent to the distance between $\mathbf{R}_{um}$ and $\mathbf{P}$. If the point $\mathbf{P}$ does not lie between the points $\mathbf{R}^i_{map}$ and $\mathbf{R}^{i+1}_{map}$, the distance from these two points is counted and the smaller is selected as the distance between $\mathbf{P}$ and the i^{th} map segment. Both these cases are shown in Figures 3.6 and 3.7.

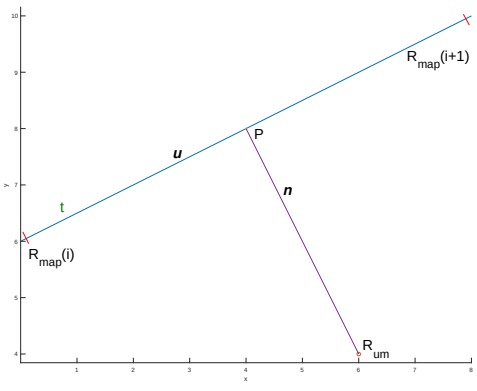


Figure 3.6: Orthogonal projection

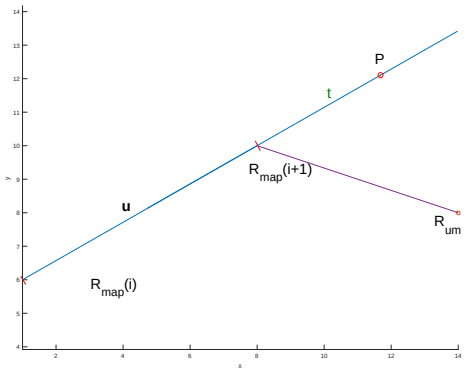


Figure 3.7: Finding the nearest end point

Chapter 4

Methods used for train positioning via Map Matching

In this chapter, several methods of map aided train positioning will be presented together with mathematical models of the train movement and the railroad. Since it is assumed that the train is not going through a very soaring landscape, it is possible to approximate the 3D position with its 2D projection. In that case, the train position $\mathbf{R}_u$ in the ENU coordinate system becomes $[E, N]$ and no information from the accelerometer z-axis or gyroscope roll angle is used. The map database of the railroad can accordingly provide the 2D projection and a database of level 2 (described in section 3.2) will be used for this purpose.

The train dynamics and the measurement can be described by the equations (4.1)

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \end{aligned} \tag{4.1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector and $\mathbf{z}_k \in \mathbb{R}^m$ is the measured system's output, $\mathbf{f}_k : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $\mathbf{h}_k : \mathbb{R}^m \mapsto \mathbb{R}^m$ are differentiable vector-valued nonlinear functions and $\mathbf{w}_k$ and $\mathbf{v}_k$ are unknown white noises [10]. The initial condition for random variable $\mathbf{x}_k$ and the probability density functions of white noises $\mathbf{w}_k$ and $\mathbf{v}_k$ are

$$\begin{aligned} \mathbf{x}_0 &\sim N(\mathbf{x}_0 : \hat{\boldsymbol{x}}'_0, \mathbf{P}_0) \\ \mathbf{w}_k &\sim N(\mathbf{w}_k : \mathbf{0}, \mathbf{Q}_k) \\ \mathbf{v}_k &\sim N(\mathbf{v}_k : \mathbf{0}, \mathbf{R}_k). \end{aligned} \tag{4.2}$$

4.1 Train positioning using DR and Map Matching

The main idea of this technique is to use the gyroscope and odometer for calculating the track curvature and finding the segment with the corresponding curvature on the map. This process is illustrated by the diagram in Figure 4.1. The algorithm is tested on simulated data in section 5.1.

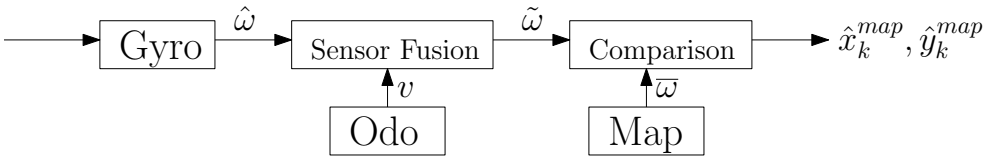


Figure 4.1: Diagram of position estimation with a map DB

As mentioned in the introduction, a dead reckoning system cannot be used as a position estimator alone but may support the GNSS during outages as it does not depend on any external signal that may be blocked. Although the error between the true and measured value gets larger over time and needs to be calibrated, for short time intervals the DR system error is small enough to give a sufficiently precise estimate. In this section, a method, mentioned in the article [17], for calculating the vehicle position using the odometer, gyroscope, and a map, that could be used to support GNSS, will be introduced.

One possibility is to measure the yaw rate over time and use this data to estimate the North, East, and up velocities $[V_N(t), V_E(t), V_U(t)]$ as shown in equation (4.4) [17]. The vehicle speed needs to be computed first and the odometer measurements can be used for this purpose. The yaw (heading) angle $\Psi(t)$ is calculated from the yaw rate by the following equation

$$\Psi(t) = \Psi(t_k) + \int_{t_k}^{t} \hat{\omega}(\tau) d\tau, \tag{4.3}$$

where $\hat{\omega}$ is the measured yaw rate and t_k is the last point where the yaw angle was determined.

Now that the vehicle speed is known, the mathematical relationship for calculating the velocities in the North, East and up directions are shown in the following equation

$$\begin{bmatrix} V_N(t) \\ V_E(t) \\ V_U(t) \end{bmatrix} \approx \begin{bmatrix} cos(\Psi(t)) & -sin(\Psi(t)) & 0 \\ sin(\Psi(t)) & cos(\Psi(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v(t)cos(\Psi(t)) \\ v(t)sin(\Psi(t)) \\ 0 \end{bmatrix}, \tag{4.4}$$

where $v(t)$ is the speed.

The North, East and up position $[R_N(t), R_E(t), R_U(t)]$ can be obtained by integrating the velocity as shown in equation 4.5

$$\begin{bmatrix} R_N(t) \\ R_E(t) \\ R_U(t) \end{bmatrix} \approx \begin{bmatrix} R_N(t_k) \\ R_E(t_k) \\ R_U(t_k) \end{bmatrix} + \begin{bmatrix} \int_{t_k}^{t} v(\tau)cos(\Psi(\tau))d\tau \\ \int_{t_k}^{t} v(\tau)sin(\Psi(\tau))d\tau \\ 0 \end{bmatrix}. \tag{4.5}$$

This algorithm for computing the N and E coordinate was tested on simulated data and the results are illustrated in Figure 5.1.

To obtain the latitude ($\phi(t)$) and longitude ($\lambda(t)$), we integrate the North and East position as follows:

$$\begin{aligned} \phi(t) &\approx \phi(t_k) + \int_{t_k}^{t} \frac{R_N(\tau)}{R_{EB}} d\tau \\ \lambda(t) &\approx \lambda(t_k) + \int_{t_k}^{t} \frac{R_E(\tau)}{R_{EB}cos(\phi(t))} d\tau \\ \phi(t) &\neq \frac{\pi}{2} \pm n\pi, n = 0, 1, ..., \end{aligned} \tag{4.6}$$

where R_{EB} is the distance between Earth center and the vehicle (approximately the Earth radius).

Matching an INS (and odometer) measurement to a map can be done by comparing the several last yaw rate measurements with the track curvature from a map database [17]. To be able to compare the measured yaw rate with the curvature from a map, two main modifications have to be done.

The first step is to convert the measured yaw rate $\hat{\omega}$ which is time-varying $[rad/s]$ to a variable that is distance-varying $[rad/m]$, so that both these variables, describing the track curvature can be compared and matched according to the criteria (3.4) and (3.5). This transformation, assuming that the speed is greater than zero, is expressed by the following equation

$$\tilde{\omega}(kT) \equiv \frac{\hat{\omega}(kT)}{v(kT)}, k \in \mathbb{N}/\{n, v(nT) = 0\} \tag{4.7}$$

where $\tilde{\omega}(kT)$ is the track curvature computed from the measured yaw rate at the time step kT, T is the sampling period, $v(Tk)$ is the speed measured by the odometer in the k^{th} sampling period. Then, $\overline{\omega}(d_l)$ can be computed as the curvature related to the traveled distance d_l (distance traveled in t_l) that is chosen from $\tilde{\omega}(kT)$ so that $d(kT)$ would be as close to d_l as possible.

The next step is to find the transformed measured yaw rates $\tilde{\omega}$ corresponding to the map segments with curvature $\overline{\omega}$ so that they can be compared. Assuming that we have more gyro measurements than there are map segments, the task is to find the index j of $\tilde{\omega}_j$ when the train crosses from the i^{th} segment to the $i+1^{th}$. This can be done using the odometer. Let the map segments be L meters long. (The algorithm works equivalently for variable spacing, this is for the facilitation of the example). The problem is illustrated in Figure 4.2.

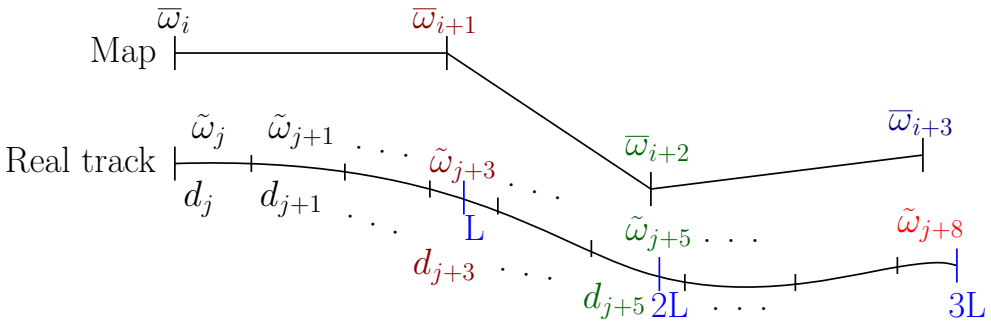


Figure 4.2: Finding the segment borders

One specific color unites the transformed yaw rate $\tilde{\omega}_j$ and distance d_j, traveled during the ${T_j}^{th}$ sampling period, belonging to one map segment with the curvature $\overline{\omega}_i$. This curvature is signed to the segment starting point.

The counter of the odometer adds up the traveled distance during the sampling periods. Let us define $d^{odo}(t_s, t_e)$ as the distance measured by the odometer between the time instants t_s to t_e, where s and e are the starting and ending indices of the time steps. The traveled distance is then counted as

$$d^{odo}(t_s, t_e) = \sum_{j=s}^{e-1} d_j,$$

where d_j is the distance traveled during the j^{th} sampling period.

The operator $div(d^{odo}(t_s, j), L)$[1] is used and the difference between $div(d^{odo}(t_s, j+1), L)$ and $div(d^{odo}(t_s, j), L)$ is calculated in order to find the index j when the train crosses segment borders. This difference gives the number of map segments that were crossed from the last odometer measurement. Since we presume INS data denser than the map points, $div(d^{odo}(t_s, j+1), L) - div(d^{odo}(t_s, j), L)$ can be either 0 or 1.

When the difference is 1, the index $j+1$ is stored as the delimiter of the gyroscope measurements corresponding to the i^{th} and $i+1^{th}$ map curve segment border. The curvature $\tilde{\omega}^{avg}$ is the averaged $\tilde{\omega}$ within the detected segment borders.

Now the map matching algorithm, described in section 3.3 - curve-to-curve matching is used. For this method to be successful, the differences in track curvature must be large enough so that the segments are distinguishable.

4.2 Sensor Integration via EKF and Point-to-Curve matching

Another approach to train positioning and map matching is mentioned in [6]. It is a method for calculating the vehicle position using the GNSS, odometer, gyroscope and a map database by the *point-to-curve* approach (discussed in section 3.3). The algorithm will be tested in section 5.2.

Sensor data are fused using the Extended Kalman Filter (EKF), further discussed in section 4.2.1. The vehicle state equations that describe the vehicle through time are shown in equations (4.8) - (4.18)

[1] $div(d^{odo}(t_s, j), L) = \lfloor \frac{d^{odo}(t_s, j)}{L} \rfloor$

$$N(k+1) = N(k) + v(k)Tcos(\theta(k))cos(\Psi(k)) + w_1 \tag{4.8}$$
$$E(k+1) = E(k) + v(k)Tcos(\theta(k))sin(\Psi(k)) + w_2 \tag{4.9}$$
$$U(k+1) = U(k) + v(k)Tsin(\theta(k)) + w_3 \tag{4.10}$$
$$\Psi(k+1) = \Psi(k) + T\dot{\Psi}(k) + w_4 \tag{4.11}$$
$$\theta(k+1) = \theta(k) + T\dot{\theta}(k) + w_5 \tag{4.12}$$
$$\dot{\Psi}(k+1) = \dot{\Psi}(k) + w_6 \tag{4.13}$$
$$\dot{\theta}(k+1) = \dot{\theta}(k) + w_7 \tag{4.14}$$
$$\Delta_{\dot{\Psi}}(k+1) = \Delta_{\dot{\Psi}}(k) + w_8 \tag{4.15}$$
$$\Delta_{\dot{\theta}}(k+1) = \Delta_{\dot{\theta}}(k) + w_8 \tag{4.16}$$
$$v(k+1) = v(k) + w_{10} \tag{4.17}$$
$$S(k+1) = S(k) + w_{11}, \tag{4.18}$$

where $\dot{\Psi}$ is the yaw rate, $\dot{\theta}$ is the pitch rate, $\Delta_{\dot{\Psi}}$ is the yaw rate offset, $\Delta_{\dot{\theta}}$ is the pitch rate offset, v is the speed, S is the scale error and $[w_1, w_2, ..., w_{11}]^T$ is a vector of white noise.

The railroad is modeled by points connected by straight lines. The position data measured by GNSS and INS are filtered by the EKF.

The measurement of the INS is usually available more often than that of the GNSS. A possibility is to wait for the GNSS signals and average the velocity measured by the odometer between the two GNSS positions. Let t_j and t_{j+1} be two consecutive time steps when the GNSS data is received and N_{j+1} be the number of INS measurements between t_j and t_{j+1}. The averaged velocity that can be paired with this position is $\overline{v}_{j+1} = \frac{1}{N_{j+1}} \sum_{i=N_j}^{N_{j+1}} v_i$, where $N_0 = 1$.

The measurement dynamics is described by the following equations

$$N_{GNSS} = N(K) + e_1 \tag{4.19}$$

$$E_{GNSS} = E(K) + e_2 \tag{4.20}$$

$$U_{GNSS} = U(K) + e_3 \tag{4.21}$$

$$\dot{\Psi}_{gyro}(k) = \dot{\Psi}(k) + \Delta_{\dot{\Psi}} + e_4 \tag{4.22}$$

$$\dot{\theta}_{gyro}(k) = \dot{\theta}(k) + \Delta_{\dot{\theta}} + e_5 \tag{4.23}$$

$$\Psi_{GNSS}(k) = \Psi(k) + e_6 \tag{4.24}$$

$$\overline{v}_{odo}(k) = \overline{v}(k)S(k) + e_7, \tag{4.25}$$

where (4.19) - (4.21) represent the N, E, U position given by the GNSS, $\dot{\Psi}_{gyro}$ and $\dot{\theta}_{gyro}$ are the measured gyroscope values, Ψ_{GNSS} is the value computed from two consecutive GNSS measurements using the relationship from equation

$$\Psi_{GNSS} = arctan\left(\frac{N_{GNSS}(j+1) - N_{GNSS}(j)}{E_{GNSS}(j+1) - E_{GNSS}(j)}\right). \tag{4.26}$$

After filtering the measurements by EKF, the task is to match the estimated position in every time step to a map segment. The matching (in the 2D projection after eliminating the z-axis) is done according to the method described in section 3.3.1.

4.2.1 Extended Kalman Filter (EKF)

EKF is a filter for systems with state or measurement equations that might not be linear, represented by the equations (4.1).

In order to use the EKF, the Jacobian matrices are computed from the state and measurement equations ((4.8) - (4.18) and (4.19) - (4.25)) getting $\mathbf{F}_k(\hat{\mathbf{x}}'_k)$ and $\mathbf{H}_k(\hat{\mathbf{x}}'_k)$.

$\mathbf{H}_k(\hat{\mathbf{x}}'_k)$ is the partial derivation of $\mathbf{h}_k(\mathbf{x}_k)$ at the point $\hat{\mathbf{x}}'_k$.

$$\mathbf{H}_k(\hat{\mathbf{x}}'_k) = \frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} | \mathbf{x}_k = \hat{\mathbf{x}}'_k. \tag{4.27}$$

Similarly the linearized version of $\mathbf{f}_k(\mathbf{x}_k)$ in the proximity of $\hat{\mathbf{x}}_k$ is achieved as

$$\mathbf{F}_k(\hat{\mathbf{x}}_k) = \frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} | \mathbf{x}_k = \hat{\mathbf{x}}_k. \tag{4.28}$$

The matrices $\mathbf{F}_k(\hat{\mathbf{x}}_k)$ and $\mathbf{H}_k(\hat{\mathbf{x}}'_k)$ for system and measurement equations (4.8) - (4.18) and (4.19) - (4.25) are

$$\mathbf{F}_k(\hat{\mathbf{x}}_k) = \begin{pmatrix} 1 & 0 & 0 & a_{14} & a_{15} & 0 & 0 & 0 & 0 & a_{110} & 0 \\ 0 & 1 & 0 & a_{24} & a_{25} & 0 & 0 & 0 & 0 & a_{210} & 0 \\ 0 & 0 & 1 & 0 & a_{35} & 0 & 0 & 0 & 0 & a_{310} & 0 \\ 0 & 0 & 0 & 1 & 0 & T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4.29}$$

$$\begin{aligned} a_{14} &= -vTcos(\theta(k))sin(\Psi(k)), \\ a_{15} &= -vTsin(\theta(k))cos(\Psi(k)), \\ a_{24} &= vTcos(\theta(k))cos(\Psi(k)), \\ a_{25} &= -vTsin(\theta(k))sin(\Psi(k)), \\ a_{35} &= vTcos(\theta(k)), \\ a_{110} &= Tcos(\theta(k))cos(\Psi(k)), \\ a_{210} &= Tcos(\theta(k))sin(\Psi(k)), \\ a_{310} &= Tsin(\theta(k)), \end{aligned}$$

$$\mathbf{H}_k(\hat{\mathbf{x}}'_k) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & S & v \end{pmatrix}. \tag{4.30}$$

The form of filtration estimate $\hat{\mathbf{x}}_k$ and approximated filtration density function covariance matrix $\mathbf{P}_k$ are shown in the following equations

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}'_k + \mathbf{P}'_k\mathbf{H}_k^T(\hat{\mathbf{x}}'_k)[\mathbf{H}_k(\hat{\mathbf{x}}'_k)\mathbf{P}'_k\mathbf{H}_k^T(\hat{\mathbf{x}}'_k) + \mathbf{R}_k]^{-1}[\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}'_k)] \\ \mathbf{P}_k &= \mathbf{P}'_k - \mathbf{P}'_k\mathbf{H}_k^T(\hat{\mathbf{x}}'_k)[\mathbf{H}_k(\hat{x}'_k)\mathbf{P}'_k\mathbf{H}_k^T(\hat{\mathbf{x}}'_k) + \mathbf{R}_k]^{-1}\mathbf{H}_k(\hat{\mathbf{x}}'_k)\mathbf{P}'_k. \end{aligned} \tag{4.31}$$

The prediction estimate $\hat{\mathbf{x}}'_k$ and approximated predictive density function covariance matrix $\mathbf{P}'_k$ is computed according to equations

$$\begin{aligned} \hat{\mathbf{x}}'_{k+1} &= \mathbf{f}_k(\mathbf{x}_k) \\ \mathbf{P}'_{k+1} &= \mathbf{F}_k(\hat{\mathbf{x}}_k)\mathbf{P}_k\mathbf{F}_k^T(\hat{\mathbf{x}}_k) + \mathbf{Q}_k. \end{aligned} \tag{4.32}$$

4.3 Positioning via GNSS, INS and similarity maximization principle

The article [12] suggests a method that integrates GNSS and INS data for train positioning using the Cubature Kalman filter - CKF that will be discussed further in section 4.3.1. The algorithm is tested in section 5.3. Odometer, accelerometer, and Doppler radar are used for distance calculation and validation, gyroscope (used in gyro-odometry) for routing detection on a switch. The track model (on level 2) from section 3.2 is used. There are two possibilities of integrating the sensors:

1. **GNSS/INS enhances the odometer** - In this approach the actual system with balizes, used for odometer recalibration, is enhanced by GNSS/INS thus has more information to improve the accuracy of the estimates. An advantage of the implementation of this approach would definitively be integration without disrupting the current system.

2. **GNSS/INS substitutes the odometer** - Here the GNSS/INS based system replaces the whole current system so that higher independence on a specific control architecture is achieved.

Data from all of the sensors are fused using the CKF. The final position is matched to a track map that is taken as an absolute reference.

Let the estimated (filtered) train position in time k be described as $\mathbf{I}_\ell(k) = [x(k)\ y(k)\ l(k)]^T = [\mathbf{R}_{um}(k)^T\ l(k)]^T$, where $l(k)$ is the traveled distance. Let $\mathbf{P}_{map} = [x_{map}\ y_{map}\ l_{map}]$ be the map points from level 2. The **point-to-curve** map matching technique (described in section 3.3) is adopted and the task of the map matching algorithm is to find the position $\mathbf{P}_{match}(k)$ in the map database, that corresponds to $\mathbf{R}_{um}(k)$ with **maximal probability and similarity**. The track map consists of abscissas with the points from the map database as endpoints. The pattern describing the similarity of the x, y coordinates of the potential matching points on the abscissa with respect to the distance from the abscissa endpoints is compared with the pattern describing the similarity of the x, y coordinates of the filtered position with respect to the distance from the endpoints of the abscissa as shown in equation (4.38). The matched point $\mathbf{P}_{match}$ is chosen as the point with maximal similarity according to the equation (4.39). The algorithm consists of 4 steps:

1. **Obtaining the data fusion position** - The data from the sensors are fused and filtered by the CKF. The state equations (4.8) - (4.18) and the measurement equations

(4.19) - (4.25) are used for data fusion.

2. **Acquiring the candidate map segment** - (A candidate map segment is a segment that is somehow considered to be matched to $\mathbf{R}_{um}$.) The probability of a candidate segment is computed by using the distance between fusion output $\mathbf{R}_{um}(k)$ and the map points $\{\mathbf{P}_{map}(j)\}\big|_{j=i}^{i+N}$ with a fixed length window of N points. The Gaussian function base probability is then

$$p_{ca}(k,j) = \frac{1}{\sqrt{2\pi}\sigma_c} \cdot exp\left(- \left\| \frac{\mathbf{R}_{um}(k) - \mathbf{P}_{map}}{h} \right\|^2 \right), \tag{4.33}$$

where σ_c is a parameter and $h = \sqrt{2}\sigma_c$ is the bandwidth[2].

The most probable starting point of a segment can be computed by the following equation

$$\mathbf{P}_{map_s}(k) = \underset{j}{\text{argmax}}(p_{ca}(k,j)). \tag{4.34}$$

The endpoint $\mathbf{P}_{map_e}(k))$ of the chosen segment is selected as the second most probable point from the set of map points $\{\mathbf{P}_{map}(j)\}\big|_{j=i}^{i+N}$ excluding $\mathbf{P}_{map_s}(k)$ using the same equation as for $\mathbf{P}_{map_s}(k))$.

The curve segment between $\mathbf{P}_{map_s}(k)$ and $\mathbf{P}_{map_e}(k)$ is chosen as a data set for map matching.

3. **Similarity calculation and identification** - Now that the segment borders $\mathbf{P}_{map_s}(k)$ and $\mathbf{P}_{map_e}(k)$ have been obtained, the matching position $\mathbf{P}_{match}(k)$ can be found from this segment. The distances between $\mathbf{R}_{um}$, $\mathbf{P}_{map_s}(k)$ and $\mathbf{P}_{map_e}(k)$ are computed in the North (y) and East (x) direction and the initial target model, a vector characterizing the position of $\mathbf{R}_{um}$ relative to the abscissa endpoints, can be given as

$$\begin{aligned} \hat{q}_1 &= C \cdot exp\left(\left\| -\frac{x(k) - x_{map_s}}{h_0} \right\|^2 \right), & \hat{q}_2 &= C \cdot exp\left(\left\| -\frac{x(k) - x_{map_e}}{h_0} \right\|^2 \right) \\ \hat{q}_3 &= C \cdot exp\left(\left\| -\frac{y(k) - y_{map_s}}{h_0} \right\|^2 \right), & \hat{q}_4 &= C \cdot exp\left(\left\| -\frac{y(k) - y_{map_e}}{h_0} \right\|^2 \right) \end{aligned} \tag{4.35}$$

where $C = \frac{1}{\sqrt{2\pi}\sigma}$ is the normalization factor[2], h_0 is the bandwidth chosen as $\sqrt{2}\sigma$ and $x_{map_s}, y_{map_s}, x_{map_e}, y_{map_e}$ are the x and y coordinates of $\mathbf{P}_{map_s}(k)$ and $\mathbf{P}_{map_e}(k)$.

[2]This relationship does not occur in the original text

The track between the points $\mathbf{P}_{map_s}(k)$ and $\mathbf{P}_{map_e}(k)$ can be defined by a set of points $\mathbf{P}(n)$, writing $\mathbf{P}_{map_s}$ as $[P_{map_s x}, P_{map_s y}]$ and using the directional vector $\mathbf{u} = [P_{map_s x}(k+1) - P_{map_s x}(k), P_{map_s y}(k+1) - P_{map_s y}(k)] \triangleq [u_x, u_y]$, as

$$\mathbf{P}(n) \in \begin{bmatrix} P_{map_s x}(k) + u_x \cdot \mathfrak{p} \\ P_{map_s y}(k) + u_y \cdot \mathfrak{p} \end{bmatrix}, \tag{4.36}$$

where $\mathfrak{p} \in < 0, 1 >$ is a parameter. The candidate target model, a vector characterizing the position of each point $\mathbf{P}(n)$ relative to the abscissa endpoints, can be described by the same parameters as the initial target model. The model for a candidate element $\mathbf{P}(n)$ is defined as

$$\begin{aligned} \tilde{p}_1 &= C_p \cdot exp\left(\left\| -\frac{x_P(n) - x_{map_s}}{h_0} \right\|^2 \right), \qquad \tilde{p}_2 = C_p \cdot exp\left(\left\| -\frac{x_P(n) - x_{map_e}}{h_0} \right\|^2 \right) \\ \tilde{p}_3 &= C_p \cdot exp\left(\left\| -\frac{y_P(n) - y_{map_s}}{h_0} \right\|^2 \right), \qquad \tilde{p}_4 = C_p \cdot exp\left(\left\| -\frac{y_P(n) - y_{map_e}}{h_0} \right\|^2 \right), \end{aligned} \tag{4.37}$$

where $C_p = \frac{1}{\sqrt{2\pi\sigma}}$ is the normalization factor[2] and $[x_P, y_P]$ are the x and y coordinates of $\mathbf{P}(n)$. Now the similarity function can be computed as

$$\hat{\rho}(n) = \rho[\tilde{p}(n), \hat{q}] = \sum_{i=1}^{4} \sqrt{\tilde{p}_i(n) \cdot \hat{q}_i}, \tag{4.38}$$

where $\hat{\rho} \in [0, 1]$ is the similarity between the fusion position and the candidate model $\mathbf{P}(n)$. The filtered position is then matched to the point $\mathbf{P}_{match}(k)$ which represents a point from all the $\mathbf{P}(n)$ points meeting the following criterion

$$\mathbf{P}_{match}(k) = \underset{n}{\operatorname{argmax}}(\hat{\rho}(n)). \tag{4.39}$$

4. **Heading validation** - Once the matched point $\mathbf{P}_{match}(k)$ is obtained, the measured track curvature can be validated by the curvature from the map database. Given a certain threshold ξ, the validation has the following form

$$|\Psi_m(k) - \Psi_{map}(k)| \leq \xi, \tag{4.40}$$

where $\Psi_{gyro}(k)$ and $\Psi_{map}(k)$ are the heading at measured position and the map matched heading.

4.3.1 Cubature Kalman Filter (CKF)

CKF is a filter for systems that can have a nonlinear state or/and measurement equation described as (4.1). Unlike EKF, the CKF does not use local linearization. Instead, a set of sampling (cubature) points is generated every time step and used to approximate the distribution function of the random variable [13]. The dimension of the state parameter is n. Let us define the variables used in the algorithm.

$$\xi^i = \sqrt{n}\{1\}_i, i = 1, ..., 2n \tag{4.41}$$

$$\{1\} = \left\{ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix} \right\}$$

$\{1\}$ represents a complete holohedral point set with $\{1\}_i$ being the i^{th} point of $\{1\}$.

There are two main steps in the CKF algorithm; the *time update*, where the state prediction x' is computed from the transformed cubature points and the *measurement update*, where the current state estimation $\hat{x}$, using new cubature points and the measurement z, is calculated.
The CKF algorithm process is as follows:

- **Time update**

  1. Generating cubature points

$$\chi_{k-1}^{(i)} = \sqrt{P_{k-1}}\xi^{(i)} + \hat{x}_{k-1} \tag{4.42}$$

  2. Transformation of cubature points

$$\overline{\chi}_{k-1}^{(i)} = f\left(\chi_{k-1}^{(i)}\right) \tag{4.43}$$

  3. State prediction

$$\hat{x}_k' = \frac{1}{2n}\sum_{i=1}^{2n}\overline{\chi}_{k-1}^{(i)} \tag{4.44}$$

  4. Covariance matrix of state prediction

$$P_k' = \frac{1}{2n}\sum_{i=1}^{2n}\overline{\chi}_{k-1}^{(i)}\left(\overline{\chi}_{k-1}^{(i)}\right)^T - \hat{x}_k'\hat{x}_k'^T + Q_{k-1} \tag{4.45}$$

- **Measurement update**

  1. Calculating cubature points

$$\overline{\chi}_k^{(i)} = \sqrt{P_k'}\xi^{(i)} + \hat{x}_k' \tag{4.46}$$

  2. Transformation of cubature points

$$\overline{z}_k^{(i)} = h\left(\overline{\chi}_k^{(i)}\right) \tag{4.47}$$

  3. Measurement prediction

$$z_k' = \frac{1}{2n}\sum_{i=1}^{2n}\overline{z}_k^{(i)} \tag{4.48}$$

  4. Covariance matrix of innovation

$$P_{z_k} = \frac{1}{2n}\sum_{i=1}^{2n}\overline{z}_k^{(i)}\left(\overline{z}_k^{(i)}\right)^T - z_k'z_k'^T + R_k \tag{4.49}$$

  5. Cross-correlation variance matrix

$$P_{x_kz_k} = \frac{1}{2n}\sum_{i=1}^{2n}\overline{\chi}_k^{(i)}\left(\overline{z}_k^{(i)}\right)^T - \hat{x}_k'z_k'^T \tag{4.50}$$

6. Calculation of gain matrix of filter

$$K_k = P_{x_k z_k} P_{z_k}^{-1} \tag{4.51}$$

7. Estimation of current state

$$\hat{x}_k = \hat{x}'_k + K_k(z_k - z'_k) \tag{4.52}$$

8. Calculating covariance matrix of error

$$P_k = P'_k + K_k P_{z_k} K_k^T \tag{4.53}$$

Chapter 5

Simulations

In this chapter the methods from sections 4.1 and 4.2 were tested in MATLAB, using simulated data, and evaluated. The data were received as a part of the assignment.

The MATLAB function $ECEF2ENU(refPoint, ECEFpoint)$ was used for ECEF to ENU transformation of the coordinates, where $refPoint$ is the origin of the local ENU system with the geodetic coordinates ϕ, λ, h and $ECEFpoint$ are the coordinates of the point $\mathbf{R}_u^{ECEF}$ that is to be transformed (discussed in chapter 2).

5.1 Numerical illustration of the method Train positioning using DR and Map Matching

The method from section 4.1, that was originally mentioned in [17] was tested on simulated data that were generated without any additional noise. From the gyroscope only the yaw rate was considered. The map database consists of North and East points spaced at 10m intervals with the corresponding curvature.

First, the odometer and gyroscope were used for track reconstruction without the map to show how precise these sensors are. The yaw (heading) angle Ψ from the gyroscope was used to determine the train turning. By using Ψ and the speed v, given by the odometer, it is possible to reconstruct the vehicle trajectory as shown in Figure 5.1. The North and East coordinates are obtained from the equation (4.5).

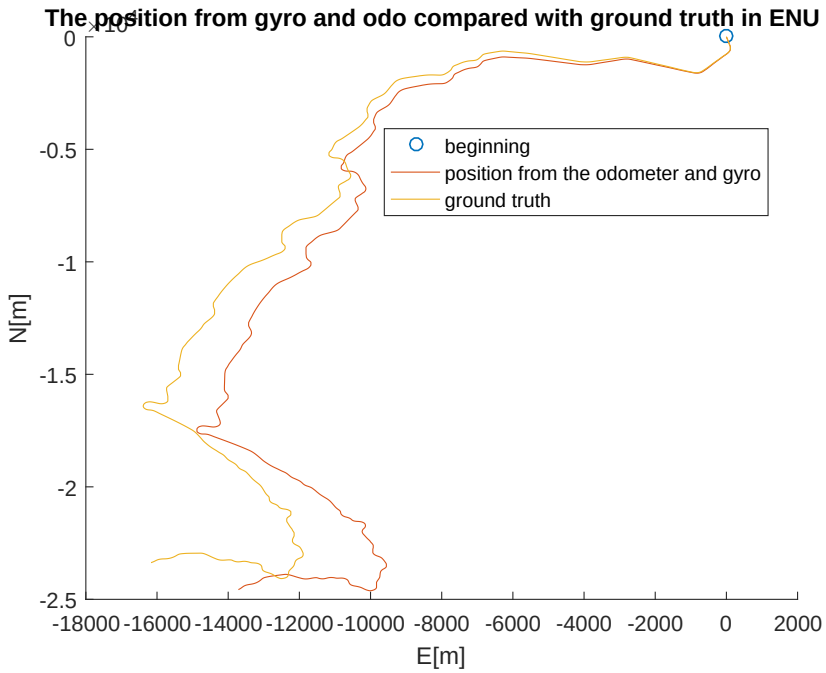


Figure 5.1: Position by DR sensors

The squared error of the real (ground truth) position and the position from DR is plotted in Figure 5.2.

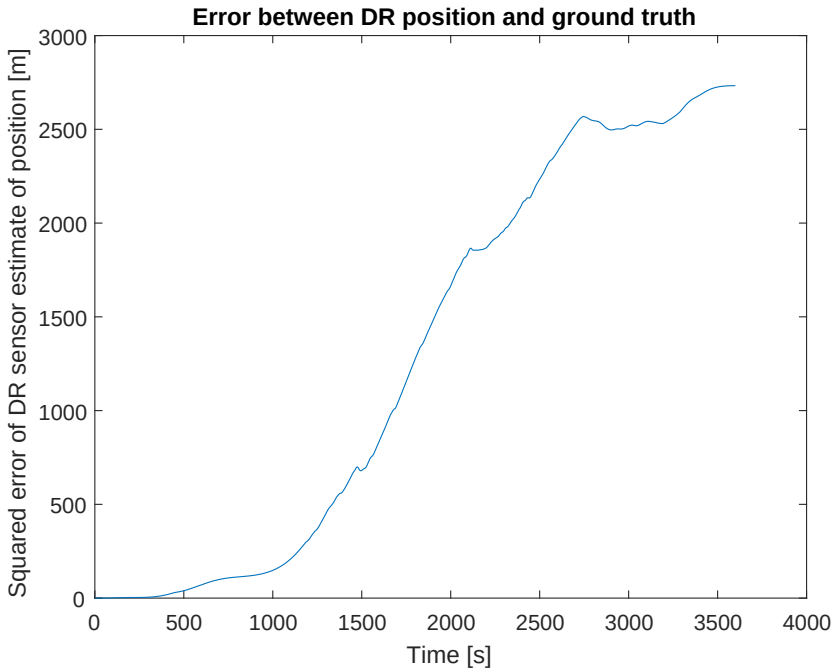


Figure 5.2: Squared error of dead reckoning sensor estimate

It is clear that dead reckoning does not provide accurate position in long terms (due to the Earth rotation) and needs to be corrected by a reference positions if used as a source of the North and East position.

In Figure 5.3 an example of the trajectory given by the odometer and gyroscope, corrected by the GNSS ENU position every 1500 samples, is plotted with the ground truth segments.

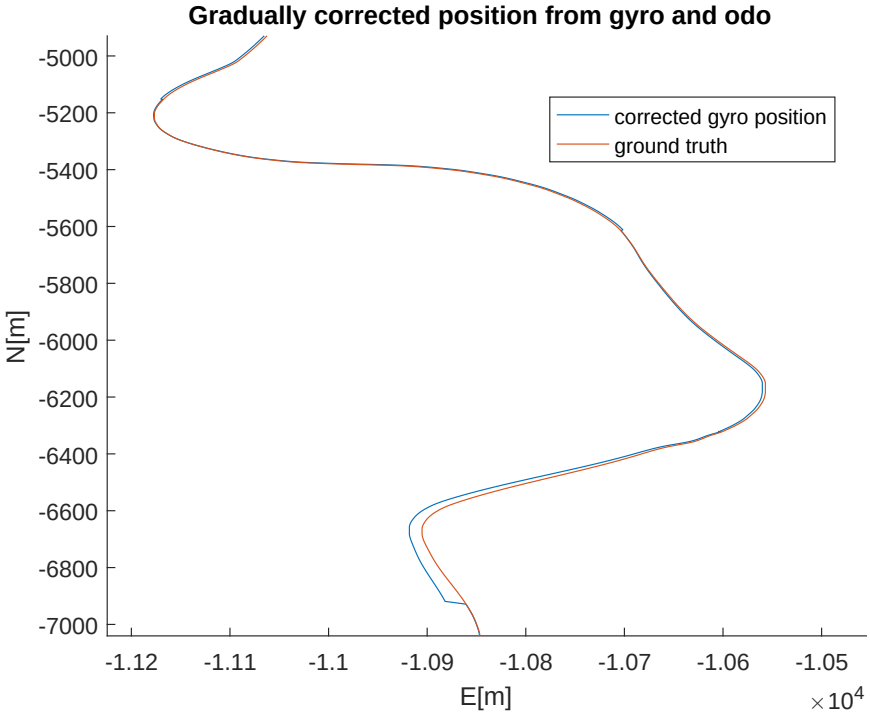


Figure 5.3: Zoomed in corrected DR position

As the gyroscope provides more frequent data than the map database, it is necessary to reduce (average) the number of measured yaw rates to match the map database. This was done as mentioned in 4.1. Now the averaged yaw rate $\tilde{\omega}^{avg}$ can be matched to a segment curvature $\overline{\omega}$ minimizing the criteria (3.4) and (3.5).

The number of compared map segments in each iteration is $(J-n)\cdot(n+1)$, where J is the number of compared map segments and $n+1$ is the number of transformed yaw rates to be compared.

Results

The percentage of correctly computed map curve segments using the equation (3.4) is 41.43%. The results of MAAE algorithm are better, for our simulated track the percent-

age of MAAE correctly determining the segments is 55.78% for the tested data when comparing the curvature of 20 consecutive map and gyro segments. This percentage drops to 29.14% when the curvature of only 10 segments is compared. If the options of the train position are constrained to the surrounding 200 segments (100 behind and 100 in front of the actual train position) the percentage of the correctly estimated position rises to 75.06% for comparing 20 segments and 59.99% for 10 segments. The time needed for the estimation also drops dramatically (by 96%).

Certain inaccuracies resulting from the conversion form $\hat{\omega}$ to $\tilde{\omega}$ and choosing the curvatures $\tilde{\omega}_j$ to be averaged into $\tilde{\omega}^{avg}$ have to be considered, thus the map curve segment was declared as matched correctly if matched to the map segment with the same index j or $j \pm 1$.

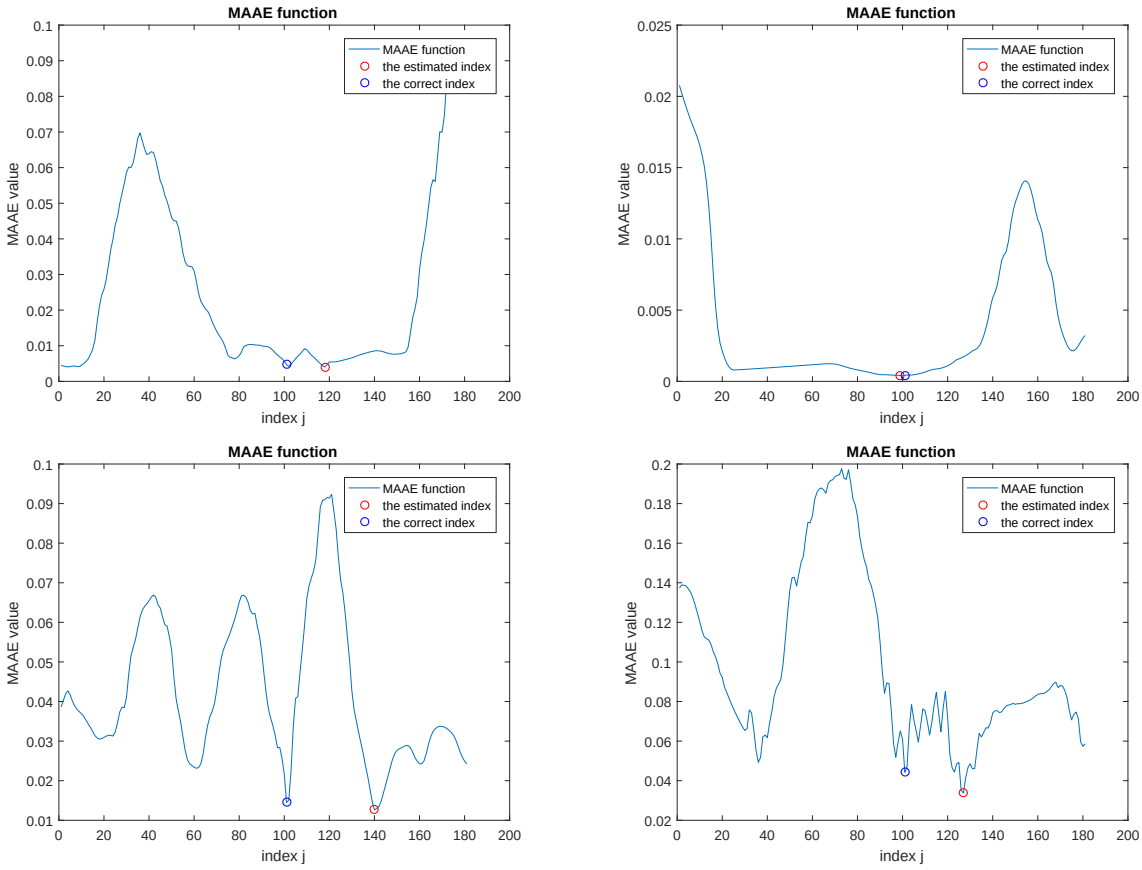


Figure 5.4: Illustration of wrong segment estimation

Figure 5.4 shows four cases when the correct segment was not found by the MAAE algorithm restricted to 200 map segments. It is clear that the function may have more

low values that relate to other similarly curved segments. One possible solution to this problem is either to compare more segments or to restrict the area where the corresponding map DB segments are searched from. Another possibility would be to integrate more sensors. The simulations results confirm the presumption that this method gives satisfactory results only for largely varying track curvatures.

5.2 Numerical illustration of the method Sensor Integration via EKF and Point-to-Curve matching

The method from section 4.2, that was originally mentioned in [6] was tested. The data from GNSS, yaw, and pitch gyroscope and odometer were used and filtered by the EKF to obtain the filtered estimate of the receiver position. Simulated data of the same trajectory as for simulation described in the previous section were used together with GNSS data, to which zero mean Gaussian white noise was added so that the efficiency of the filter could be tested. Ψ_{GNSS} was computed according to equation (4.26).

The following matrices

$$Q = \begin{pmatrix} 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-2} & 0 & T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-2} & 0 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-7} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-5} \end{pmatrix} \tag{5.1}$$

(5.2)

$$R = \begin{pmatrix} 50 \cdot DOP & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 50 \cdot DOP & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5200 \cdot DOP & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-2} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 10^{-2} & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-2} \end{pmatrix} \tag{5.3}$$

(5.4)

were taken from [6] and P_0 was initialized as $zeros(11, 11)$ as the starting position is known in this simulation. *DOP* is an acronym for *Dilution of Precision*, the number that characterizes the geometrical distribution of satellites and affects the level of trust that can be placed in each satellite. The *DOP* values were part of the simulated data.

The state equation was initialized as

$$[GNSS_{ENU}(1)^T \, , \Psi_{GNSS}(1) \, , \theta_{GNSS}(1) \, , \dot{\Psi}(1) \, , \dot{\theta}(1) \, , 0 \, , 0 \, , \overline{v}_{mean}(1) \, , 0]^T. \tag{5.5}$$

All these values are taken from the first time step the GNSS was available. $\Psi_{GNSS}(1)$ was computed according to equation (4.26) and $\theta_{GNSS}(1)$ was computed using the following equation

$$\theta = atan\left(\frac{GNSS_U(2) - GNSS_U(1)}{||(GNSS_E(2) - GNSS_E(1), GNSS_N(2) - GNSS_N(1))||_2}\right), \tag{5.6}$$

$\dot{\Psi}(1)$ and $\dot{\theta}(1)$ are the first gyroscope measurements and $\overline{v}_{mean}(1)$ is the first measured odometer speed.

Results

The filtered position was matched to the closest point of the closest map segment according to the algorithm described in section 3.3.1. The map matching using the minimal distance between measured point and abscissa is shown in Figure 5.5 for several points of the filtered position estimate.

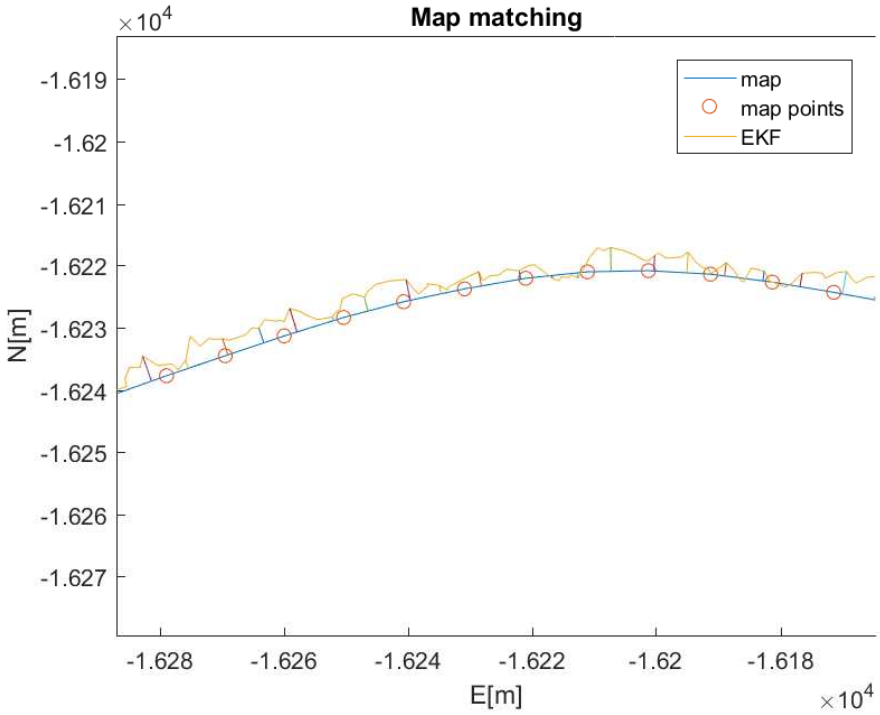


Figure 5.5: Map matching

The time-average of squared errors between the estimated and ground-truth position of the $[N, E]^T$ coordinates is $[2.99m^2, 4.70m^2]^T$ and $[1.85m^2, 1.38m^2]^T$ for the EKF-filtered and the map matched position. The time-average of squared errors of the distance between the filtered and the ground-truth position is $7.70m^2$ and $3.23m^2$ for the matched and the ground-truth position.

An experiment was conducted in order to find out how the EKF can cope with missing or senseless data. When the odometer measurements were set to zero from the 100^{th}

time step further, the time-average of squared errors between the estimated and ground-truth position of the $[N, E]^T$ coordinates became $[3.09m^2, 4.58m^2]^T$ and the time-average of squared errors of the distance between the filtered and the ground-truth position was $8.02m^2$. The speed variance (in the filtration covariance matrix $P(10, 10)$) became $14.70m^2$ at the end of the simulation instead of $1.35m^2$ with the odometer.

When the same experiment was conducted with the elimination of the yaw gyroscope measurement, almost no difference from the original error was measured. Time-average of squared errors between the estimated and ground-truth position of the $[N, E]^T$ coordinates is $[2.91m^2, 4.74m^2]^T$. The time-average of squared errors of the distance between the filtered and the ground-truth position was $7.66m^2$. Also the yaw gyro variance did not change.

On the other hand, when the GNSS North or East measurements were faulty, the EKF was not capable of getting somewhat close to the real position with the estimates. When the GNSS Up measurement was eliminated, the averaged square error in the $[N, E]^T$ coordinates was $[4.16m^2, 4.26m^2]$ and the averaged distance was $8.42m^2$. The Up variance did not change significantly.

5.3 Numerical illustration of the method Positioning via GNSS, INS and similarity maximization principle

The simulation of the method from section 4.3, originally mentioned in [12], is described and evaluated in this section. The same data as for simulation in the previous section (GNSS, odometer, gyroscope) were used including the added white noise to the GNSS measurement. Unlike the original article, nor the accelerometer nor Doppler radar were used in the simulation in order to gain data for comparing this method with the method mentioned in 4.2.

The Cubature Kalman filter was adopted to process the measured data. The state equations (4.8) - (4.18) were utilized and initialized as (5.6) as in the previous method. The state noise covariance matrix Q was also taken from the previous method - (5.1). For accomplishing satisfactory results, the measurement noise covariance matrix has to be chosen "small", otherwise the filter tends to straighten the curves in the trajectory. Therefore the covariance matrix R was chosen as $0.001 \cdot \mathbf{I}_{7\times 7}$ meaning that the measurement can be trusted and that less weight should be placed in the model dynamics, describing the movement along a straight line.

Results

The map matching technique, mentioned in [12] was applied. Several values of σ were tested, as shown in Table 5.1, and $\sigma = 4$ was chosen as the best parameter. In Figure 5.6 it can be seen that various σ values result in choosing different matched point $\mathbf{P}_{match}$.

σ	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
error in [E,N]m^2	[6.62, 6.61]	[6.57, 5.42]	[6.42, 4.4]	[6.07, 3.55]	[5.38, 2.94]	[4.27, 2.62]	[2.98, 2.57]	[2.26, 2.69]	[2.23, 2.88]	[2.59, 3.09]
distance from GT m^2	13.24	11.99	10.82	9.63	8.32	6.89	5.55	4.95	5.11	5.68

Table 5.1: The errors depending on σ

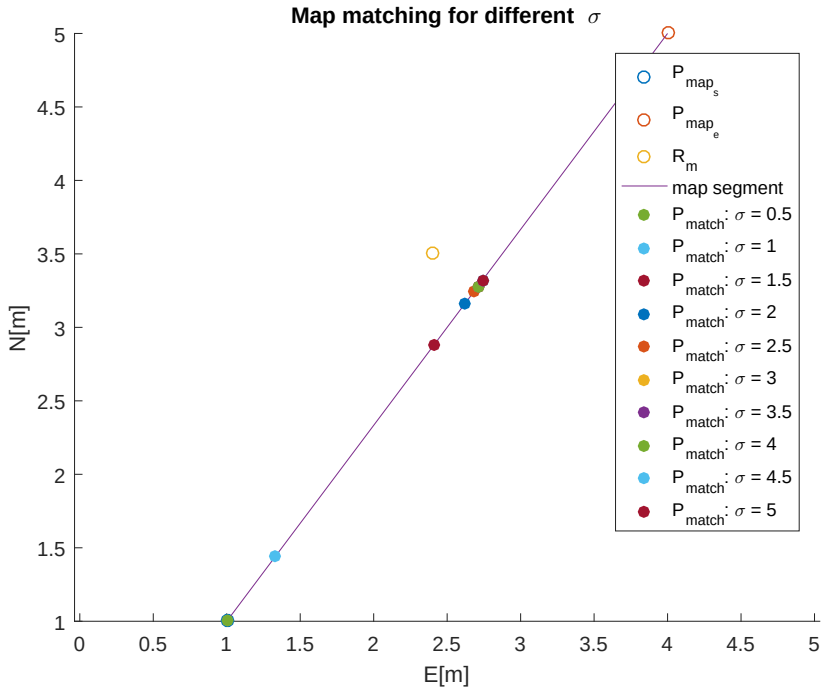


Figure 5.6: Finding the best σ

The matching of several filtered points positions to a map according to this technique with the parameter $\sigma = 4$ is shown in Figure 5.7.

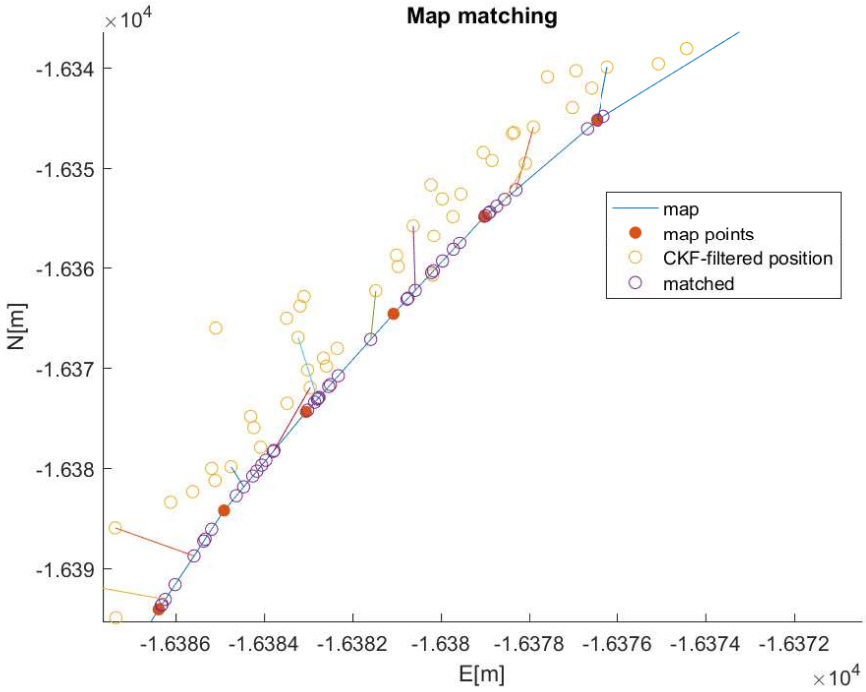


Figure 5.7: Map matching

The time-average of squared errors of the position from CKF in E, N coordinates is $[2.12m^2, 6.63m^2]$ and the average distance between the true and the estimated position is $8.74m^2$. After map matching the time-average of squared errors of the position E, N coordinates is $[2.26m^2, 2.69m^2]$ and the average distance between the true and the matched position is $4.95m^2$.

The heading validation, mentioned in the original article, was done by comparing the yaw angle calculated from the yaw rate measured by the gyroscope from equation (4.3) with the heading angle computed from the map points as

$$\psi_{map} = atan\left(\frac{|\mathbf{R}^{i+1}_{map_y} - \mathbf{R}^{i}_{map_y}|}{|\mathbf{R}^{i+1}_{map_x} - \mathbf{R}^{i}_{map_x}|}\right), \tag{5.7}$$

with the variables $\mathbf{R}_{map_x}$ and $\mathbf{R}_{map_y}$ explained in 3.3.1.

Unfortunately, this validation was not successful probably due to great inaccuracies in the computation of heading angle Ψ_{gyro}.

Instead, the transformed yaw rate $\tilde{\omega}$ (the transformation was done using equation (4.7)) was compared to the track curvature $\overline{\omega}$ of the segment to which $\mathbf{R}_{um}$ was matched.

This validation is shown in Figure 5.8 together with the compared map curvature and the speed of the train, which has a big impact on the accuracy of yaw rate measure-

ment. The estimate is marked as "validated" if $|\tilde{\omega}(k) - \overline{\omega}(k)| \leq 0.3|\overline{\omega}(k)|$.

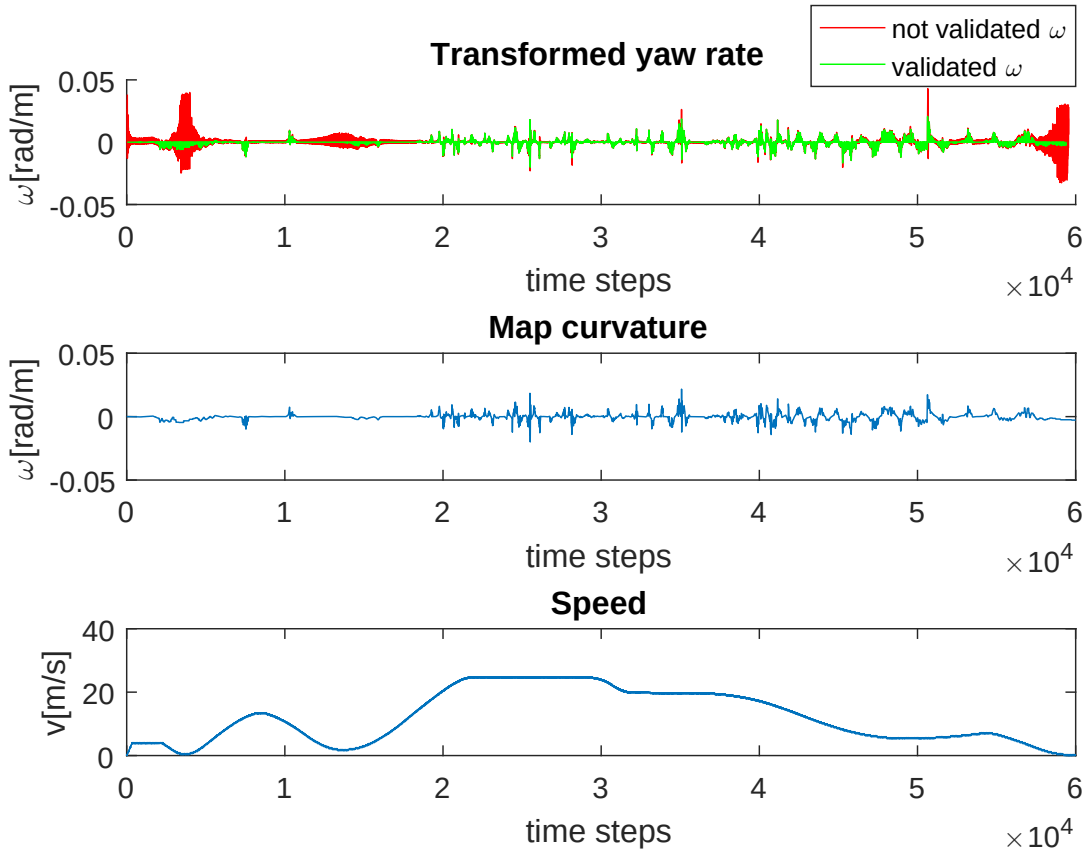


Figure 5.8: Validation of map matching

5.4 Method evaluation

The first method, **Train positioning using DR and Map Matching**, that uses only gyroscope, odometer data, and a level 2 map database (section 3.2), can be used for substitution of methods that use the GNSS in places, where the GNSS signal is blocked or poor quality. The possible map segments, where the train may be localized, must be very constrained both to cut down the time needed for the computation and to reduce the chance of the position being incorrectly matched to a segment with a curvature similar to the correct one. The time needed for the matching of one set of transformed measured yaw rates is almost proportional to the number of map segments that are considered for matching.

When the train position was constrained to the 200 surrounding segments (which gives 2km in case of segment length 10m), the percentage of correctly estimated segments reached over 75%. The main advantage of this method is the independence from the GNSS.

For the other two methods the GNSS, yaw and pitch gyroscope, and odometer data together with a map database were used. There is no need to constrain the area where the train may be situated. From these two methods, the faster and more accurate (in the sense of minimal distance from ground truth train position) was the method **Sensor Integration via EKF and Point-to-Curve matching**. The system dynamics and measurement equations need to be known as well as the covariance matrices of the state and measurement noises, that have to be Gaussian independent random processes with zero mean.

The EKF seems to be a good solution for the nonlinear state and measurement equations, presuming that the initial position of the train is known. Compared to that, the CKF, used in the method **Positioning via GNSS, INS and similarity maximization principle** seems like an unnecessarily complicated and slower solution with worse results. Also the map matching technique used in this method does not achieve as good results as matching according to minimal distance. A recapitulation of the results (time-average of squared errors) of these two methods is in Table 5.2.

	EKF	EKF - map matching	CKF	CKF - map matching
error in [E,N] m^2	[2.99, 4.70]	[1.85, 1.38]	[2.12, 6.63]	[2.26, 2.69]
distance from GT m^2	7.70	3.23	8.74	4.95

Table 5.2: Errors of methods in sections 4.2, 4.3

Conclusion

The thesis is aimed to analyze, and test methods for train positioning using map matching. First, the sensors that are used in these methods were described together with their models. The coordinate systems in which the sensors measurements are given are presented. Subsequently, the train map database and map matching were discussed.

Next, three methods of train positioning via map matching were described with the necessary filtering and map matching mathematical background. The firstly described method, **Train positioning using DR and Map Matching**, uses data sensors other than GNSS receivers and can be used to aid the GNSS or for train positioning in an environment, where the GNSS signal is unavailable or corrupted. The other two methods, **Sensor Integration via EKF and Point-to-Curve matching** and **Train positioning using the GNSS and INS data and similarity maximization principle**, take advantage of all the mounted sensors and fuse them by a nonlinear filter. The advantage of the last-mentioned method is the usage of the Cubature Kalman filter that does not require local linearization at the price of higher computational complexity. Each method uses a different map matching technique.

All these methods were implemented and tested in *MATLAB*. The method **Train positioning using DR and Map Matching** does not provide sufficiently accurate estimates under standard conditions. The condition for the application of this method is a greatly varying train track curvature or/and an additional positioning technique. The methods **Sensor Integration via EKF and Point-to-Curve matching** and **Train positioning using the GNSS and INS data and similarity maximization principle** provide more accurate estimates and could be considered for direct use in rail automation.

The future work could include finding and testing more methods of train positioning, for example using the particle filter to replace the EKF or CKF or a different method for map matching. A detailed analysis of sensor errors could be carried out. Implementation in a programming language that would allow faster computation would be a necessary step before testing during operation. Testing the algorithms on real data with diverse noises would be necessary in order to develop a method that could be used in safety critical applications.

Bibliography

[1] GNSS - Global navigation satellite system. https://www.czechspaceportal.cz/3-sekce/gnss-systemy, Last accessed on 2019-11-25.

[2] Official U.S. government information about the global positioning system (GPS) and related topics. https://www.gps.gov/systems/gps/space/, Last accessed on 2019-04-06.

[3] European global navigation satellite systems agency - Galileo is the European global satellite-based navigation system. https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system, Last accessed on 2019-04-06.

[4] Balise. http://www.railsystem.net/balise/, Last accessed on 2020-03-25.

[5] free cliparts, photos, design assets download. https://www.uihere.com/free\-cliparts/search\?q=Geodetic+datum, Last accessed on 2020-01-06.

[6] David Andersson and Johan Fjellström. Vehicle positioning with map matching using integration of a dead reckoning system and GPS, 2004. Thesis.

[7] M.N. Armenise, C. Ciminelli, F. Dell'Olio, and V.M.N. Passaro. *Advances in Gyroscope Technologies*. Springer Berlin Heidelberg, 2010. ISBN 9783642154942.

[8] D.M. Bevly and S. Cobb. *GNSS for Vehicle Control*. GNSS technology and applications series. Artech House, 2010. ISBN 9781596933026.

[9] Dwaipayan Biswas. *Recognition of elementary upper limb movements in nomadic environment*. PhD thesis, University of Southampton, 2015.

[10] Jindřich Duník. *Identifikace systémů a filtrace*. Západočeská univerzita v Plzni, 2018. ISBN 978-80-261-0775-0.

[11] J. Taufer I. Punčochář, O. Straka. Locomotive position and speed estimation in post processing mode. *In Proceedings of the 14th European Workshop on Advanced Control and Diagnosis, Bucharest, Romanias*, 2017.

[12] J Liu, B Cai, T Tang, J Wang, and Wei ShangGuan. Research on a hybrid map matching algorithm for global navigation satellite system based train positioning. In *12th International Conference on Computer System Design and Operation in the Railways and other Transit Systems, Beijing, China*, pages 59–70, 2010.

[13] Min Liu, Jizhou Lai, Zhimin Li, and Jianye Liu. An adaptive cubature kalman filter algorithm for inertial and land-based navigation system. *Aerospace Science and Technology*, 51:52–60, 2016.

[14] Zakriya Mohammed, Ibrahim Abe M Elfadel, and Mahmoud Rasras. Monolithic multi degree of freedom (MDoF) capacitive mems accelerometers. *Micromachines*, 9(11):602, 2018.

[15] Tsouknidas Nikolaos and Kiyoshi Tomimatsu. QR-code calibration for mobile augmented reality applications: linking a unique physical location to the digital world. 01 2010. doi: 10.1145/1836845.1836999.

[16] Wei Quan, Jianli Li, Xiaolin Gong, and Jiancheng Fang. *INS/CNS/GNSS integrated navigation technology*. Springer, 2015. ISBN 9783662451595.

[17] S. S. Saab. A map matching approach for train positioning. i. development and analysis. *IEEE Transactions on Vehicular Technology*, 49(2):467–475, March 2000. ISSN 1939-9359.

[18] Jan Škach. Stanovení polohy pohybujícího se objektu pomocí satelitního systému. 2013. Master's thesis.

[19] Tomáš Vajdiak. Návrh FSK modulátoru pro eurobalízu na hradlovém poli. 2012. Master's thesis.

[20] Christopher E White, David Bernstein, and Alain L Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation research part c: emerging technologies*, 8(1-6):91–108, 2000.

[21] Peng Xie and Mark Petovello. Measuring GNSS multipath distributions in urban canyon environments. *IEEE Transactions on Instrumentation and Measurement*, 64, 08 2014. doi: 10.1109/TIM.2014.2342452.

[22] Guoliang Zhu. *Trajectory-aided GNSS land navigation: application to train positioning*. PhD thesis, Universite De Technologie De Troyes, 2014.