MACIN: Map-Aided Cooperative Inertial Navigation for Ground Vehicles

by

Robert Grissom Cofield

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 2 2020

Keywords: inertial navigation, differential GPS, Rao-Blackwellized particle filter, map, road network, connected vehicles, V2V, cooperative navigation, localization

Approved by

David M. Bevly, Chair, Professor of Mechanical Engineering
John Y. Hung, Professor of Electrical and Computer Engineering
Scott M. Martin, Assistant Research Professor of Mechanical Engineering

Abstract

This thesis presents an inertial navigation system (INS) that leverages the global positioning system (GPS) and a sparse road network database to cooperatively localize ground vehicles within close proximity to one another. The algorithm that constitutes the core contribution is named MACIN, an acronym for **m**ap **a**ided **c**ooperative **i**nertial **n**avigation.

Increasing demand for driver assistance features in consumer ground vehicles has spurred demand for ubiquitous high-accuracy absolute positioning. Accuracy at the lane level (under 1 meter) is required to execute complex operations such as maneuver planning. At the same time, standard automotive sensors such as cameras, inertial measurement units (IMUs), and GPS receivers do not provide this accuracy. Furthermore, common navigation techniques for fusing these sensor measurements, such as loose GPS/INS coupling in an extended Kalman filter (EKF), produce position performance that is consistently on the order of several meters in benign conditions.

MACIN comprises several improvements upon the loosely coupled GPS/INS EKF approach to acheive sub-meter accuracy and accurate lane determination. It uses sparse lane geometry information and lane sensing capability to apply position constraints along the earth tangent plane. The states of neighboring vehicles are estimated concurrently, and differential GPS is used to relate their states to one another. Lastly, Rao-Blackwellized particle filtering (RBPF) is used to estimate position with particles, while all other variables within the state are estimated with standard linearized filtering.

The success of these improvements is measured by reduction of positional error along the earth tangent plane. MACIN's performance is compared to that of a loosely coupled GPS/INS EKF in both highway and suburban conditions. This thesis shows that the proposed novel filter consistently reduces error from 1-3 meters to the submeter level.

Acknowledgments

To Prof. Leslie Whatley, your mentorship and guidance through my graduate school years were invaluable, and you taught me how to be a student outside the classroom.

To Dan Pierce and Grant Apperson, we started in the GAVLab at the same time and stuck together through many sleepless nights and impossible problems. This thesis wouldn't be possible without your help, and I wouldn't be an engineer without the lessons we learned together.

To my family, I hope only to one day grant others the level of grace you have shown me. I love you and cannot begin to thank you.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ECEF	Earth-centered, earth-fixed coordinate frame
ECI	Earth-centered inertial coordinate frame
EKF	Extended Kalman Filter
GMTI	Ground Moving Target Indicator
GNSS	Global navigation satellite system
GPS	Global positioning system
INS	Inertial navigation system
LC	Loosely coupled
OOSM	Out of Sequence Measurement
PVA	Position, velocity, and attitude
RBPF	Rao-Blackwellized Particle Filter
RTK	Real time kinematic
UKF	Unscented Kalman Filter
WGS84	World Geodetic System, 1984
$\mathbf{r}^C_{A/B}$	Three dimensional position vector, describing Frame A with respect to Frame B, expressed in Frame C

$\mathbf{v}_{A/B}^{C}$ Three dimensional velocity vector, describing Frame A with respect to Frame B, expressed in Frame C

$\mathbf{a}_{A/B}^{C}$ Three dimensional acceleration vector, describing Frame A with respect to Frame B, expressed in Frame C

Chapter 1

Introduction

The idea of automating personal vehicles so they could drive themselves has been around almost as long as the automobile itself. Engineering efforts to accomplish this began as early as 1926 [1], and work to that end has proceeded ever since. The modern approach to ground vehicular autonomy requires ubiquitous location services at the sub-meter level, and while the first part is satisfied, work still remains for the second.

Anyone with a smartphone can know their location within 100m, but sub-meter accuracy still requires an expert to create a custom solution with custom infrastructure. Generally, this entails equipping a vehicle with costly laser ranging sensors, and comparing their readings to a sophisticated map. The creation and maintenance of said map is a technical and logistical feat unto itself.

At the same time, the technology requisite to bring ubiquitous accuracy to sub-meter levels already exists without these labor- and capital-intensive special techniques. Anyone on earth with line of sight to the sky can use some form of Global Navigation Satellite System ("GNSS"), be it GPS (GNSS operated by the United States), GLONASS (the Russian counterpart), or some other constellation. Much of the earth's roadways have already been mapped (albeit sparsely) and are available for free use as part of the Open Street Map project [2]. And lastly, means of transmitting and receiving data between two vehicles electronically are available to just about anyone. Many vehicles come pre-equipped with cellular modems which enable vehicle-to-infrastructure (V2I) communication, and dedicated short range communications (DSRC) technology [3] has enabled a standardized method of direct vehicle-to-vehicle communication. This thesis contends that intelligent application of those 3 things can make ubiquitous sub-meter localization possible for ground vehicles.

A common approach for attempting high-accuracy localization is to use Extended Kalman Filtering (EKF) to combine measurements from a GNSS receiver and an inertial measurement unit (IMU), thus creating an inertial navigation system (INS). This baseline strategy is discussed in Section 3.5.2. *The goal of this thesis is to improve upon the loosely coupled GNSS/INS EKF approach in two ways: allow multiple vehicles to work cooperatively to improve their individual absolute positioning performance, and further refine the PVA estimates using a sparse a priori map.* The resultant approach is referred to as **Map Aided Cooperative Inertial Navigation, or MACIN**. In order to further define the problem space prior to attempting a solution, the following problem criteria are put in place to guide design:

- Assume that a high-accuracy, lane level map exists for any road upon which a vehicle will drive. Many such databases exist today, both public and proprietary, with varying levels of information. For the sake of maximizing applicability, the information in this map should be as sparse as possible.

- Assume some system exists that is capable of determining whether or not the host vehicle is within the marked boundaries of a lane (i.e., whether or not the vehicle is not drifting between neighboring lanes is known). Many commercial off-the-shelf computer vision systems exist which are capable of this, and Lane Departure Warning (LDW) is a standard feature in many consumer vehicles on the road today.

- Full 6 degree of freedom ("DoF") pose estimation is required, as well as 3D velocity estimates. Pose is defined here as 3 dimensional position in conjunction with 3 dimensional orientation.

- No physical infrastructure is available to aid in positioning, so solutions such as real time kinematic (RTK) GNSS with a base-station are not available. Physical infrastructure outside of a host vehicle is a common barrier to scalability and accessibility.

- The algorithm should be able to run in real-time on a modern computing system to supply PVA estimates to an online system. As such, post-processing techniques are disallowed.

- Only commercially available sensors should be employed.

Following these assumptions, MACIN comprises the following enhancements upon the baseline filter:

1. Rao-Blackwellize the position portions of the state vectors of all vehicles to allow non-Gaussian distributions.

2. Couple location estimation with the map as a hard position constraint by assigning zero probability to any position that is not on the mapped roadway.

3. Enable cooperative localization using high-accuracy estimates of the relative position vectors (RPVs) between adjacent vehicles in the update step. These RPVs are generated with dynamic base real time kinematic (DRTK) GPS, which is discussed in Section 5.5.

The baseline filter commonly experiences position errors between 1 and 2 meters, as shown in Chapter 6. For roads with lane widths as small as 2.7 meters [4], this means that it is not sufficiently accurate to for lane determination. Figure 1.1 below depicts this position uncertainty as a red ellipse, spanning lane boundaries. The application of steps 1 and 2 above are intended to reduce position error in the direction perpendicular to the roadway, as show below in Figure 1.2. And lastly, use of high-accuracy DRTK relative positioning in Step 3 above, is expected to further reduce errors so that they are consistently below 1 meter, accurate enough for lane determination. This final step is shown in Figure 1.3 below. These hypotheses are tested and proven to be accurate in this thesis.

MACIN combines several subfields within the localization and navigation discipline. This thesis is divided into several chapters, all focusing on a particular subfield. Each begins

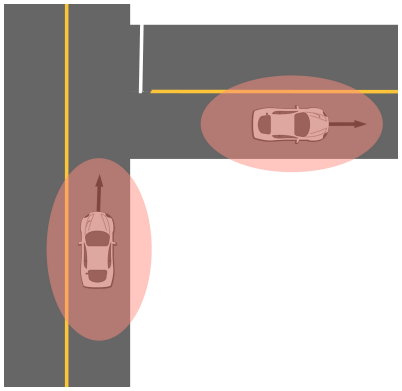

Figure 1.1: Typical position errors for the baseline filter, shown as red ellipses, reach well over standard lane widths.

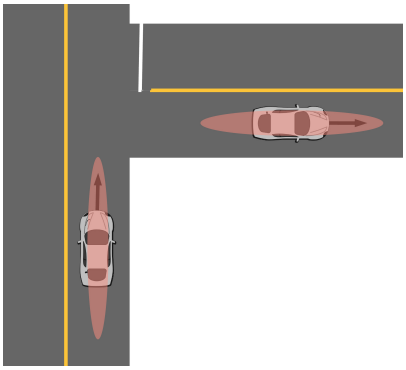

Figure 1.2: Application of map constraints will reduce position error lateral to the roadway.

with existing research for the relevant field, explaining approaches at a general level and giving greater detail where relevant. They then conclude with the specific strategy employed by MACIN, such that readers may replicate results on their own. Chapter 2 reviews the basics of inertial sensors, and gives the IMU error model used herein. Chapter 3 reviews estimation strategies for inertial navigation and presents MACIN's filtering approach, known as a Rao-Blackwellized Particle Filter (RBPF). Chapter 4 reviews existing approaches for leveraging map information for navigation, and gives the MACIN approach for using sparse maps as probabilistic constraints. Chapter 5 reviews approaches for multi-vehicle cooperative navigation, and shows how differential GPS is used to this end in MACIN. Chapter 6 presents the core work of this thesis as a unified algorithm, with pseudocode that will allow the reader to implement MACIN and replicate the results presented later in Appendix A. This chapter

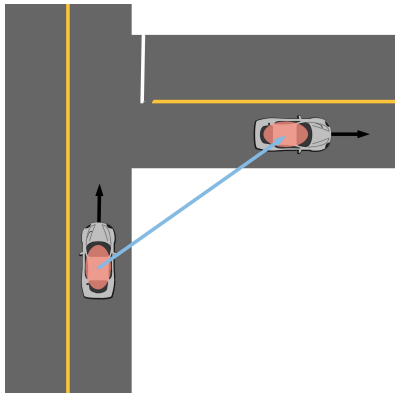

Figure 1.3: Use of DRTK as a final improvement will constrain MACIN's position errors such that they are consistently less than 1 meter.

also covers the implementation of MACIN in a software package and examines real-world data collection scenarios which prove the effectiveness of MACIN. Chapter 7 summarizes the work and empirical findings, and it also gives future extensions. Appendix A contains tabulated data from field testing described in Chapter 6.

Chapter 2

Inertial Measurement Units

Inertial measurement units are perhaps the most common navigation sensor for most ground vehicles. The most common general purpose configuration comprises 6 indivual sensors: 3 accelerometers mounted in orthogonal Cartesian axes, and 3 gyroscopes that are mounted in axes coincident with the aforementioned accelerometers. This allows full 6 degree of freedom (DoF) pose estimation. The term "pose" refers to both position and orientation. Six DoF refers to the ability to observe both position and orientation in all 3 Cartesian axes. In order to do this, the IMU measurements must be integrated over time, and a method for doing this is detailed in Section 3.5.2.

Other configurations include sensors in extra axes to over-observe pose [5], or possibly fewer sensors. A configuration common in automotive applications is to include 1 accelerometer in the vehicle's longitudinal axis and 1 gyroscope in the vehicle's vertical axis. A secondary accelerometer parallel to the vehicle's lateral axis is sometimes used as well. Using this reduced sensor set along with a dimensionally reduced vehicle model is one of several popular definitions of the phrase "dead reckoning". Another, more broad, usage of the term refers to any strategy that uses mathematical integration over time to compute pose without absolute correction.

An IMU typically provides measurements in one of two ways: continuous and discrete. Discrete sensor measurements report a change from one epoch to the next in the first integral of the quantity that they truly measure with respect to time. That is, discrete accelerometers report change in velocity and discrete gyroscopes report change in attitude. Han et al. [6] examines characterization of errors in discrete IMUs, as the process differs from that of continuous sensors. Discrete measurement sensors have the drawback of requiring that no

measurements be missed. Continuous sensors report the value that they are measuring directly. Thus the sampling rate is extremely flexible, and may even be varied while running, if needed, to accommodate changing system dynamics or computational concerns. This thesis will focus on continuous sampling sensors, and all formulation is done in the continuous time domain.

2.1 Error Models

All inertial sensors have errors, and time integration causes their effect to be exponentially magnified. *The fundamental problem of inertial navigation is to infer these IMU errors and remove them such that the resultant errors in position, velocity, and attitude are as low as possible.* A detailed analysis of error growth over time in position, velocity, and attitude is provided in [7].

The error sources for inertial sensors are well investigated in the literature. In order to infer their contribution to the sensor's output at any instant in time, they must be modeled. When selecting a model to use, the necessary complexity is dependent upon the required accuracy and quality of the sensor. In general, the most commonly modeled sources of error which are present in both gyroscope and accelerometer measurements are:

- Additive noise: η_ν.
- Markovian walking bias: b
- Output scaling: k
- Constant (turn on) bias: c
- Sensor misalignment: m

2.1.1 Additive Noise

Additive noise is typically modeled as having zero mean with a Gaussian probability density function (PDF): $\eta_\nu \sim \mathcal{N}(0, \sigma_\nu)$, where σ_ν is the standard deviation. The simplest and least accurate model includes only this term, where measurement errors are then formulated as:

$$\tilde{\omega}^B_{B/I} = \omega^B_{B/I} + \eta_{g,\nu} \tag{2.1a}$$

$$\tilde{\mathbf{a}}^B_{B/I} = \mathbf{a}^B_{B/I} + \eta_{a,\nu} \tag{2.1b}$$

The measurements received from the IMU are $\tilde{\omega}^B_{B/I}, \tilde{\mathbf{a}}^B_{B/I}$ denoting that they represent the angular velocity and linear acceleration, respectively, of the IMU body (B) relative to the inertial frame (I), and are expressed in the IMU body frame. They are functions of the true values, $\omega^B_{B/I}$ and $\mathbf{a}^B_{B/I}$. Note that the portion of the acceleration measurement attributable to gravity is not separated from true acceleration here, but is dealt with later on during estimation (3.60). For strapdown sensors (most typical), each axis represents a measurement from a separate piece of hardware, so the covariance matrix is diagonal to show that the errors are not correlated across axes:

$$E[\eta_{g,\nu}\eta^T_{g,\nu}] = \sigma^2_{g,\nu} I_{3x3} \tag{2.2a}$$

$$E[\eta_{a,\nu}\eta^T_{a,\nu}] = \sigma^2_{a,\nu} I_{3x3} \tag{2.2b}$$

2.1.2 Markov Bias

The error model used by Groves [8] also includes a first order Markovian bias:

$$\tilde{\omega}^B_{B/I} = \omega^B_{B/I} + \mathbf{b}_g + \eta_{g,\nu} \tag{2.3a}$$

$$\tilde{\mathbf{a}}^B_{B/I} = \mathbf{a}^B_{B/I} + \mathbf{b}_a + \eta_{a,\nu} \tag{2.3b}$$

The Markov process has a time evolution with its own white noise process:

$$\dot{\mathbf{b}}_g = \frac{-1}{\tau_g}\mathbf{b}_g + \eta_{g,u} \tag{2.4a}$$

$$\dot{\mathbf{b}}_a = \frac{-1}{\tau_a}\mathbf{b}_a + \eta_{a,u} \tag{2.4b}$$

$$\eta_{g,u} \sim \mathcal{N}(\mathbf{0}, \sigma_{g,u}) \tag{2.4c}$$

$$\eta_{a,u} \sim \mathcal{N}(\mathbf{0}, \sigma_{a,u}) \tag{2.4d}$$

The vectors $\eta_{g,u}$ and $\eta_{a,u}$ are additive zero-mean Gaussian noise processes as well, which are assumed to have no correlation. As such, they are characterized by diagonal covariance matrices as shown in Equations (2.5):

$$E[\eta_{g,u}\eta_{g,u}^T] = \sigma_{g,u}^2 I_{3x3} \tag{2.5a}$$

$$E[\eta_{a,u}\eta_{a,u}^T] = \sigma_{a,u}^2 I_{3x3} \tag{2.5b}$$

The rows of the column vectors $\mathbf{b}_g$, $\mathbf{b}_a$ correspond to axes of the IMU. Some choose to ignore the terms containing the time constants τ_g and τ_a, despite the ease with which they may be characterized (as discussed in Section 2.2). The bias then becomes a random walk process instead of a Markov process.

Once the sensor's time constant is identified, one must give special attention to how it will affect estimator behavior during aiding sensor outages when the bias cannot be directly inferred. A small time constant will cause the bias estimate to converge to zero as time approaches infinity in open loop operation. Neglecting the time constant (removing the term) or using a model with a large value will result in bias estimates that are static or very slow moving during outages of aiding information. Regardless of sensor grade, a properly characterized time constant will produce behavior that is closest to optimal.

2.1.3 Scaling

The error model used by Crassidis [9] builds on this by including output scaling on the measurements:

$$\tilde{\omega}^B_{B/I} = (\mathbf{I}_{3x3} + K_g)\ \omega^B_{B/I} + \mathbf{b}_g + \eta_{g,\nu} \tag{2.6a}$$

$$\tilde{\mathbf{a}}^B_{B/I} = (\mathbf{I}_{3x3} + K_a)\ \mathbf{a}^B_{B/I} + \mathbf{b}_a + \eta_{a,\nu} \tag{2.6b}$$

The scaling factor matrices are simply a reshaping of the respective vectors, where $K_g = \mathbf{k}_g I_{3x3}$, $K_a = \mathbf{k}_a I_{3x3}$ and the rows of the columns vectors $\mathbf{k}_g$, $\mathbf{k}_a$ correspond to axes of the IMU. Output scaling errors are most typically the result of the sensor's operating temperature. The effects of temperature are primarily mechanical in nature, due to expansion or contraction of the sensor body or housing. Most IMUs are calibrated such that output error is minimized at an operating point around 25C. Modelling output scaling is beneficial for automotive applications where the IMU is mounted near a vehicle's engine, transmission, or some other hot environment. Flenniken [10, 11] performs an in-depth analysis of the contribution scaling has to overall errors.

2.1.4 Constant Error

Many IMUs have a constant offset in each measurement, sometimes referred to as "turn-on bias". The authors of [7,10,12] include this in their formulations. The measurement model when including this bias becomes:

$$\tilde{\omega}^B_{B/I} = (\mathbf{I}_{3x3} + K_g)\ \omega^B_{B/I} + \mathbf{b}_g + \mathbf{c}_g + \eta_{g,\nu} \tag{2.7a}$$

$$\tilde{\mathbf{a}}^B_{B/I} = (\mathbf{I}_{3x3} + K_a)\ \mathbf{a}^B_{B/I} + \mathbf{b}_a + \mathbf{c}_a + \eta_{a,\nu} \tag{2.7b}$$

Extra care must be given when modeling uncertainty of the turn-on bias. If noise is added to the constant term, it becomes a random walk process. [13] The Markov bias term

also begins to approach the behavior of a random walk process as its time constant grows very large, so these two must be considered together.

2.1.5 Misalignment

Lastly, a strapdown IMU typically attempts to mount each of the three accelerometers and gyroscopes orthogonally, such that they are rotated 90 degrees from one another. Mechanical shortcomings may cause mismounting (non-orthogonality) errors to become non-negligible. When this is the case the raw readings must be rotated, so the measurement equation becomes:

$$\tilde{\omega}^B_{B/I} = (\mathbf{I}_{3x3} + K_g + M_g)\ \omega^B_{B/I} + \mathbf{b}_g + \mathbf{c}_g + \eta_{g,\nu} \tag{2.8a}$$

$$\tilde{\mathbf{a}}^B_{B/I} = (\mathbf{I}_{3x3} + K_a + M_a)\ \mathbf{a}^B_{B/I} + \mathbf{b}_a + \mathbf{c}_a + \eta_{a,\nu} \tag{2.8b}$$

$$M = \begin{bmatrix} 0 & m_{xy} & m_{xz} \\ m_{yx} & 0 & m_{yz} \\ m_{zx} & m_{zy} & 0 \end{bmatrix} \tag{2.8c}$$

The value $I + K + M$ is a complete rotation matrix, but it is not a direction cosine matrix (DCM). A DCM maintains orthogonality as a special case, wheras the purpose of this quantity is to rotate the individual axes relative to one another. The diagonals of this matrix correspond to the alignment of each axis relative to its "ideal" orientation, while the addition of off-diagonals in the misalignment matrix M allows the skew of each axis relative to the others the be represented and corrected. Many authors will analytically compute the values of each scalar m trigonometrically, so that a more intuitive Euler representation can be used [14, 15]. However, this step is unnecessary and can be omitted for simplicity with the values of M estimated directly. Since misalignment refers to a permanent hardware configuration, it need only be estimated once, and can thereafter be treated as constant.

Special care must be given to ensure the misalignment is observable in a given system, which is why it is omitted from the MACIN prototype.

2.1.6 Other Error Sources

The above sections outlined the most commonly modeled error sources. As one pursues higher and higher accuracy, modeling and removing errors sources becomes progressively more difficult. Some of these include nonlinearity, g-Sensitivity, coning and sculling, and thermal stress.

Increasingly, the pursuit for lower drift will require the navigation engineer to consider the physics behind each gyroscope and accelerometer. Furthermore, the mounting of the IMU unit will likely contribute unit-specific errors, particularly in the absence of shock mounting or vibration dampening. For instance, any aiding sensors whose mounting pose relative to the coordinate frame of the IMU likely undergoes stresses which violate the fixed extrinsic assumption. Calibration of these smaller values is outside the scope of this thesis and left to future research.

2.2 Error Source Characterization

This thesis will utilize the error model of Equations (2.6). Determining the statistical properties of the error sources is a relatively straightforward matter and is referred to as "intrinsic calibration". Specifically, Allan deviation and autocorrelation analyses are used to estimate the variance of white noise processes $(\eta_{g,\nu}, \eta_{a,\nu}, \eta_{g,u}, \eta_{a,u})$ and the Markov time constants (τ_g, τ_a). This thesis follows the strategies laid forth by [10, 16], which are briefly summarized here.

To characterize the sensor both the gyroscope and the accelerometer each have three quantities that must be estimated, all [3x1] vectors:

1. The standard deviation of white measurement noise, σ_ν

2. The standard deviation of the Markov bias noise, σ_u

3. The Markov time constant, τ

This makes for a total of 12 scalar values, individually enumerated later in Equation (3.54).

A long static data set is collected, during which time the sensor is not moving. A recommendation commonly found throughout literature is that 2 hours is a minimum length of time required. For this thesis, a 2 hour data set and a 36 hour data set were used, the shorter for algorithm development and the longer for characterization.

After data recording is finished, the Allan deviation and autocorrelation of those sensor measurements is then computed. The Allan deviation value at the time window point $\Delta t = 1.0$ is then σ_ν. The time and value at which the autocorrelation function decays to a fraction $1/e$ of its original value then corresponds to τ and σ_u, respectively. Flenniken describes this procedure in greater detail in the appendices of [10].

In areas where earthquakes are common, one should ensure that no seismic events occur during this recording, lest they introduce error into the data. The United States Geological Survey catalogs known seismic events and they can be readily searched for this purpose [17]. Other common sources of contamination from unexpected movement include passing vehicles, construction, and human foot traffic. Furthermore, one must be sure that the sensor reaches a steady state temperature very close to its operating environment so that nominal temperature effects will be characterized. The best way to do this is to power on sensor and leave it running for several hours prior to recording data, to allow internal heat generation effects to reach equilibrium. Leaving the sensor in its target vehicle mounting configuration, if possible, will allow the data recording temperature to be the same as the operating temperature.

Chapter 3

Estimation Strategies for Inertial Navigation

The IMU is a relative navigation sensor, rather than an absolute navigation sensor, meaning that the measurements cannot provide pose in a global earth frame. Rather, standalone IMUs can only provide change in pose over time with respect to an initial pose. In order to use inertial readings to navigate in a global frame, an absolute positioning sensor or algorithm is required. This chapter will present strategies for general fusion of IMU data with absolute positioning data, disregarding the positioning data source. Following that, Section 3.5 will discuss the specific sensing solution used for the remainder of the work.

3.1 System Modeling

In light of the errors described in Section 2.1, some strategy must be employed to discern the true values underlying the raw measurements. For a model that includes only additive noise, a least squares approach may be used in some cases, but its usefulness is very limited. More complex models necessitate online calibration to estimate and remove error sources. Here, "online" means that the error terms are estimated concurrently with the pose in real time. Specifically, there is some vector of *states* $\mathbf{x}_k$ which should be estimated at a given time epoch k using information from sensor measurements at the same epoch $\mathbf{y}_k$. Given the basic goal of navigation, one must assume that the three-dimensional quantities position, orientation, and velocity will comprise some subset of variables within the state. Additionally, the state vector evolves over time according to its own value as well as some system input $\mathbf{u}_k$ and some input (process) noise η_k. This is part of the time evolution model $\mathbf{f}$. The measurement at the same time $\mathbf{y}_k$ is assumed to be a function of the state as well as

some other noise source ϵ:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}, \eta\right) \tag{3.1}$$

$$\mathbf{y} = \mathbf{h}\left(\mathbf{x}, \epsilon\right) \tag{3.2}$$

Bayesian techniques are commonly used to estimate the state $\mathbf{x}_k$ as a function of measurements $\mathbf{y}_{0:k}$. They fundamentally rely upon Bayes' rule:

$$p(\mathbf{x}_k|\mathbf{y}_{0:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{y}_{0:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{0:k-1})} \tag{3.3}$$

The function $p(\mathbf{x}_k|\mathbf{y}_{0:k})$ is referred to as the "posterior", and is the PDF of the state after the latest measurement has been assimilated. This is the function that must be estimated. It also accounts for all previous measurements $\mathbf{y}_{0:k}$ by the very nature of recursive estimation. The function $p(\mathbf{x}_k|\mathbf{y}_{0:k-1})$ is referred to as the "prior" and is the state PDF after propagation using the model f and PDF of process noise η_k. The prior does not consider the latest measurement $\mathbf{y}_k$. The function $p(\mathbf{y}_k|\mathbf{x}_k)$ is referred to as the "proposal" distribution and is the *expected* distribution of the measurement based only on the prior distribution. The proposal is calculable using the measurement model h and knowledge of the PDF of measurement noise ϵ_k. The denominator simply comprises a normalizing function since the analytical sum of any PDF must be equal to 1:

$$p(\mathbf{y}_k|\mathbf{y}_{0:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{y}_{0:k-1})\ d\mathbf{x}_k \tag{3.4}$$

Further discussion of Bayes' rule can be found in [18].

3.2 Extended Kalman Filter

The basic Kalman filter (KF) assumes that the system f is linear time-invariant (LTI), an assumption which is rarely valid or applicable in real-world scenarios. As such, the EKF is

often used instead, since it accounts for nonlinear time-varying systems. An excellent introduction to both the KF and EKF is provided by [19], which is summarized here. The EKF simplifies recursive estimation of the posterior using Bayes' rule by making four assumptions:

1. The process and measurement noises are normally distributed: $\eta_k \sim \mathcal{N}(0, Q(t_k))$
2. The process and measurement noises are independent of one another.
3. Neither the process noise nor measurement noise is serially correlated (i.e., unrelated to themselves over time).
4. The posterior distribution is Gaussian, and thus the prior is Gaussian as well.

Thus, the system estimate PDF can be completely defined by its first and second moments: a mean state $\mathbf{x}$ and state covariance P. The process and measurement noises can be characterized by covariance matrices:

$$\eta_k \sim \mathcal{N}(\mathbf{0}, Q_k) \tag{3.5}$$

$$\epsilon_k \sim \mathcal{N}(\mathbf{0}, R_k) \tag{3.6}$$

The core mechanism of this is defining a way to linearize the system at any point:

$$\dot{\mathbf{x}} = A\mathbf{x} + B_u\mathbf{u} + B_\eta\eta \tag{3.7}$$

$$\mathbf{y} = H\mathbf{x} + \epsilon \tag{3.8}$$

$$A = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \tag{3.9}$$

$$B_u = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \tag{3.10}$$

$$B_\eta = \frac{\partial \mathbf{f}}{\partial \eta} \tag{3.11}$$

$$H = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \tag{3.12}$$

$$\dot{P} = AP + PA^T + B_\eta Q B_\eta^T \tag{3.13}$$

Here, A is often referred to as the *dynamic matrix*, and describes how the system evolves as a function of itself. B_u is the *input sensitivity matrix* and desribes the effect of the *input vector* $\mathbf{u}$ on the state. B_η is the *noise sensitivity matrix* and desribes the effect of the *process noise vector* η on the state. H is the *measurement sensitivity matrix* and describes how the state and measurement vector are related to one another. The time derivative of the state covariance matrix $\dot{P}$ has an evolution that is described above using the matrix Ricatti equation.

In practice, the EKF is operated at discrete time epochs. There are two approaches to formulating the EKF in order to account for this: deriving the system in the continuous domain then discretizing it, and formulating the system in the discrete domain from the start. Formulating in the continuous domain is sometimes more intuitive for the designer, and allows easily implementing the system with filters other than the EKF, since most filters make use of a formulation of the continuous time derivative vector f at some point.

The system's time evolution from epoch t_{k-1} to epoch t_k can be discretized so that the system formulation becomes:

$$\mathbf{x}_k^- = \Phi_{k-1}\mathbf{x}_{k-1}^+ + B_{u,k-1}\mathbf{u}_{k-1} \tag{3.14}$$

$$P_k^- = \Phi_{k-1}P_{k-1}^+\Phi_{k-1}^T + Q_{d,k-1} \tag{3.15}$$

where the Φ and Q_d matrices are derived at every epoch from the system dynamic Jacobians using the following step from [9] as shown in Equations 3.19:

$$J^* = \left[\begin{array}{c|c} -A & B_\eta Q B_\eta^T \\ \hline 0 & A^T \end{array}\right] \tag{3.16}$$

$$J = e^{J^*\Delta t} = \begin{bmatrix} J_{11} & J_{12} \\ 0 & J_{22} \end{bmatrix} \tag{3.17}$$

$$\Phi = J_{22}^T \tag{3.18}$$

$$Q_d = \Phi J_{12} \tag{3.19}$$

Formulating in the discrete domain requires calculating Φ directly, and circumvents the need for the potentially costly discretization at every estimation epoch. On the other hand, it disallows some measure of extensibility to other filters, since Φ is rarely used outside of EKF applications. Since this thesis compares two approaches, and Φ isn't applicable to both of them, a continuous formulation is used.

The measurement sensitivity matrix H does not require discretization, so the Kalman update step is simply:

$$S_k = H_k P_k H_k^T + R_k \tag{3.20}$$

$$K_k = P_k H_k^T S_k^{-1} \tag{3.21}$$

$$x_k^+ = x_k^- + K_K z_k \tag{3.22}$$

$$z_k = y_k - H_k x_k^- \tag{3.23}$$

$$P_k^+ = P_k^- - K_k H_k P_k \tag{3.24}$$

where z is the measurement innovation, K is the Kalman gain matrix, and S is the innovation covariance.

For highly nonlinear systems, or systems whose dynamics are much faster than the EKF update rate, using the nonlinear equations wherever possible will yield better results. The

prior is obtained by directly integrating the dynamic model and the matrix Ricatti equation

$$x_{k+1}^{-} = \int_{t_k}^{t_{k+1}} f\left(\mathbf{x}_k^{+}, \mathbf{u}_k, \eta = 0\right) dt \tag{3.25}$$

$$P_{k+1}^{-} = \int_{t_k}^{t_{k+1}} \left[A_k P_k^{+} + P_k^{+} A_k^T + B_{\eta,k} Q_k B_{\eta,k}^T\right] dt \tag{3.26}$$

Numerically, Euler integration may be sufficient for the state propagation. However, as detailed in [20], this will not preserve positive definiteness when integrating $\dot{P}$. Using mid-point integration, as presented by [21], presents a good compromise between accuracy and numerical efficiency. This approach is referred to as *Taylor-Heun* integration for the state derivative f, and *Gauss-Legendre* integration for the state covariance derivative. Note that the introduction of a new matrix inverse necessitates useage of a matrix decomposition in order to prevent compute times from increasing unnecessarily, particularly for large state dimensions.

3.3 Particle Filter

The EKF makes several assumptions about the target system to allow "approximately" optimal state estimation to be performed by linearizing the system dynamics at every epoch in time to compute a single solution for that epoch. In practice, these assumptions are not often completely met. There is another class of Bayesian filters that circumvent the highly restrictive assumptions of the Kalman filter by *multiple hypothesis testing*. These filters determine the PDF of the state by sampling it at various places to approximate the shape and characteristics of the state's PDF, rather than creating a mathematical formulation of the PDF and characterizing it by its moments. These sample points are variously referred to as "sigma points" or "particles", depending upon the filter in which they are used. Determining how to sample the PDF is one of the key distinguishing features between these filters. Popular multiple hypothesis methods include the unscented Kalman filter (UKF), ensemble Kalman filter (EnKF), filter banks, and particle filter (PF).

The UKF provides better calculation of the state error covariance P by deterministically creating a state-dependent number of sigma points and calculating the covariance of this minimally sufficent set. This prevents having to freely integrate the derivative $\dot{P}$ from the Riccati equation and has been shown to be more accurate while retaining computational efficiency in many implementations. For more on the UKF, see [22].

The EnKF is typically used in data assimilation problems where state and/or measurement vectors are extremely large and highly nonlinear, such as climatology. The key advantage is that an EnKF requires fewer samples relative to the number of estimated state variables than other filters. It samples the state PDF stochastically, propagating each member of the ensemble in time, but performs measurement updates in a batch. Unlike the UKF, the EnKF does not track the state mean and covariance estimates, as it is unnecessary. As such when mean and covariance information are desired, they must be computed as an extra step, but in general the EnKF is often more computationally efficient than the UKF as state vector lengths increase. Additional resources for implementing EnKFs may be found in [23–25].

Kalman filter banks seem similar to the EnKF on a superficial level, but perform separate measurement updates on all particles, treating each sample of the PDF as a completely separate filter. This works because they are typically used to estimate discrete states (one possible state per filter) or estimate the likelihood of different dynamic models (one model per filter). Each entry still gets a separate propagation and measurement update using the same data, but what differentiates them is the model [26]. The filter bank approach is not limited only to Kalman or pure-Kalman techniques. The Rao-Blackwellized Particle Filter (RBPF), which is discussed next in Section 3.4, can be used in a bank approach [27]. Estimating discrete states and maintaining multiple concurrent models is highly relevant to map matching, and will be covered in detail in Chapter 4.

The particle filter (PF) can be considered the most "pure" form of multiple hypothesis Bayesian estimation, as no restrictions are placed on the state or measurement PDFs. Arulampalam [18] is most heavily cited in the literature as an excellent introduction to particle filtering. The PF relies on random sampling to add noise to each particle, which is a full state vector on its own. It is then assumed that as the number of particles approaches infinity, the accuracy of randomly sampling the noise PDF will approach 100%. As such, this is referred to as a "Monte-Carlo" method. In order to produce a single "answer" from a PF at time t_k, one must combine all the particles by averaging them with a different weight w_i assigned to each particle i according to the proposal distribution.

$$\mathbf{x}_k = \sum_{i=1}^{N_p} w_{i,k} \mathbf{x}_{i,k} \tag{3.27}$$

$$w_{i,k} = w_{i,k-1}\ p(\mathbf{y}_k|\mathbf{x}_{i,k})\ p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1}) \tag{3.28}$$

Here, N_p is the number of particles used. Afterward, a normalization step is required to ensure the particle weights sum to 1.0:

$$w_{i,k} = \frac{w_{i,k}}{\sum_{i=1}^{N_p} w_{i,k}} \tag{3.29}$$

Updating the particle weights (not the estimate) forms the measurement update step, and computing $\mathbf{x}$ afterward is how a posterior estimate is produced.

As time goes on, repeatedly adding noise to all of the particles will cause the vast majority of them to be nowhere near the solution. This is referred to as the *degeneracy problem*, and can be quantified by computing the number of *effective particles* N_{eff} as a function of the number of actual particles and their respective weights:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_p} w_i^2} \tag{3.30}$$

The procedure for mitigating degeneracy is known as "resampling", wherein particles with low weight are replaced with copies of particles that have higher weight. The copies subsequently diverge from one another after having sampled noise added. This ensures that the effective number of particles stays high, and only needs to be performed when the ratio N_{eff}/N_p drops below a certain value which is generally system-specific. There are several different variations of resampling, and Douc [28] does a good job of comparing the more common methods. This thesis employs the most common algorithm as used by [18], "systematic resampling", which is given below in Algorithm 1. Figure 3.1 graphically depicts how systematic resampling selects which particles are to be removed and which particles are to be cloned using a scenario where the proposal and prior distributions are significantly exaggerated.

Algorithm 1 Systematic Resampling from Arulampalam [18].

$c_1 \leftarrow 0$ ▷ Initialize CDF
for $i \in 2, ..., N_p$ **do** ▷ Construct CDF
    $c_i \leftarrow c_{i-1} + w_i$
$i \leftarrow 1$
$u_1 \leftarrow \mathcal{U}[0, N_p^{-1}]$ ▷ Pick random start point
for $j \in 1, ..., N_p$ **do**
    $u_j \leftarrow u_1 + (j-1)/N_s$
    while $u_j > c_i$ **do**
        $i = i + 1$
    $x_j \leftarrow x_i$ ▷ Resample particle i
    $w_j \leftarrow N_p^{-1}$ ▷ Reset weight

3.4 Rao-Blackwellized Particle Filter

The principle drawback of particle filtering is that the requisite number of particles to estimate a given state vector may cause the filter to be prohibitively slow when implemented on a real world computing system. Often-repeated conventional wisdom in the estimation fields is that one should use 10^N particles, where N is the state dimension [1]. Larger state

[1] This research was unable to locate the originating source of the hypothesis that 10^{N_p} is the ideal number of particles. It is used without explanation or citation in many publications. [29]

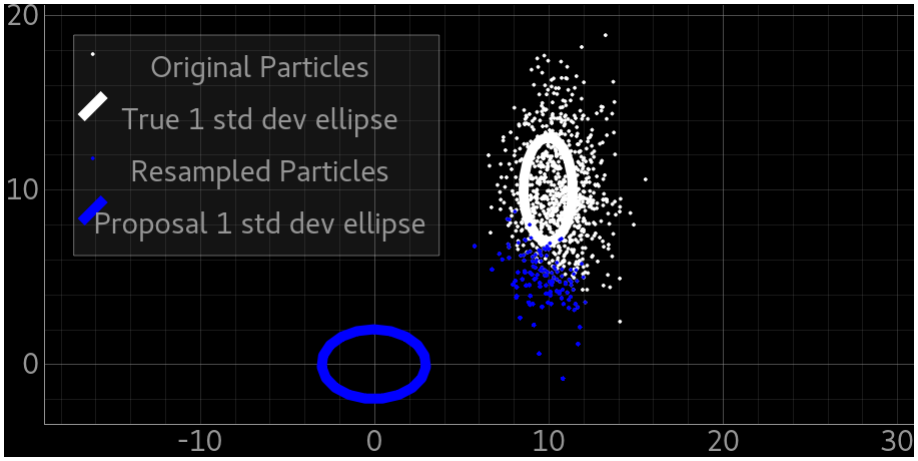

Figure 3.1: Example of systematic resampling, with exaggerated model differences

vectors require higher numbers of particles and using too few will result in performance degradation or divergence. As such, reducing state vector size is of critical importance for computational practicality. A more detailed investigation of the number of particles required to meet various performance goals is found in [30], which sets forth an approach for dynamically varying the number of particles used. Kotecha et al. [31] examines the effect of the number of particles used on performance.

The Rao-Blackwellized (or marginalized) particle filter addresses this issue by operating on the premise that only a subset of the state vector needs to be estimated with multiple hypothesis techniques. The remaining state variables, then, can be estimated using a typical extended Kalman filter. This thesis follows the notation and formulation set forth by Gustafsson [32], Schon [33], and Ryan [12].

The state vector $\mathbf{x}_k$ is divided into a linear (Kalman) partition $\mathbf{x}_k^l$ and a nonlinear (particle) partition $\mathbf{x}_k^n$:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^l \\ \mathbf{x}_k^n \end{bmatrix} \tag{3.31}$$

The transient dynamics of the system in the general case are:

$$\mathbf{x}_{k+1}^l = \mathbf{f}_k^l(\mathbf{x}_k^n) + A^l(\mathbf{x}_k^n)\mathbf{x}_k^l + G^l(\mathbf{x}_k^n)\eta_k^l \tag{3.32}$$

$$\mathbf{x}_{k+1}^n = \mathbf{f}_k^n(\mathbf{x}_k^n) + A^n(\mathbf{x}_k^n)\mathbf{x}_k^l + G^n(\mathbf{x}_k^n)\eta_k^n \tag{3.33}$$

It is worth noting that several terms can often be neglected, and the trade-offs of doing so are examined in [33]. The system is formulated in the discrete domain, so choices of discretization and numerical integration strategies are very important. The process noise is assumed to be zero mean Gaussian, just as in typical Kalman filtering:

$$\eta_k = \begin{bmatrix} \eta_k^l \\ \eta_k^n \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, Q_k) \tag{3.34}$$

The covariance matrix associated with the process noise is partitioned similarly to the state vector:

$$Q_k = \begin{bmatrix} Q_k^l & Q_k^{ln} \\ (Q_k^{ln})^T & Q_k^n \end{bmatrix} \tag{3.35}$$

The process noises for each partition are then

$$\eta_k^l \sim \mathcal{N}(\mathbf{0}, Q_k^l) \tag{3.36}$$

$$\eta_k^n \sim \mathcal{N}(\mathbf{0}, Q_k^n) \tag{3.37}$$

The cross-covariance matrix Q_k^{ln} relates the process noises on the linear and nonlinear partitions. The resulting measurement model is:

$$\mathbf{y}_{k+1} = \mathbf{h}_{k+1}(\mathbf{x}_{k+1}^n) + C_k(\mathbf{x}_{k+1}^n)\mathbf{x}_{k+1}^l + \epsilon_{k+1} \tag{3.38}$$

$$\epsilon_{k+1} \sim \mathcal{N}(\mathbf{0}, R_{k+1}) \tag{3.39}$$

Additive measurement noise ϵ_{k+1} is assumed zero mean Gaussian just as in typical Kalman formulations.

The time update occurs in two steps, one for each partition. First, the particle states are updated using Equation (3.33), then the linear portion of the particle states is subtracted:

$$\mathbf{z}_k = \mathbf{x}_{k+1}^n - \mathbf{f}_k^n \tag{3.40}$$

The state dynamics of each partition are affected by the cross-covariance between the process noise on each partition. This is first accounted for by computing a new linear state dynamic matrix and linear process noise covariance matrix:

$$\bar{A}_k^l = A_k^l - G_k^l \left(Q_k^{ln}\right)^T \left(G_k^n Q_k^n\right)^{-1} A_k^n \tag{3.41}$$

$$\bar{Q}_k^l = Q_k^l - \left(Q_k^{ln}\right)^T \left(Q_k^n\right)^{-1} Q_k^{ln} \tag{3.42}$$

It is worth noting that in cases where the process noise vectors affecting the two partitions are uncorrelated, $Q^{ln} = 0$. As such, $\bar{A}_k^l = A_k^l$ and $\bar{Q}_k^l = Q_k^l$ will be true in this case.

Following this, a modified Kalman time update is performed. The equivalent particle state error covariance matrix is propagated forward, and a quantity similar to the typical Kalman measurement gain is computed:

$$N_k = A_k^n P_k \left(A_k^n\right)^T + G_k^n Q_k^n \left(G_k^n\right)^T \tag{3.43}$$

$$L_k = \bar{A}_k^l P_k \left(A_k^n\right)^T N_k^{-1} \tag{3.44}$$

The difference here is that discrepancies arising from interdependence within the nonlinear and linear states must be weighted, rather than a measurement and a prior state. The linear state and its error covariance estimate can now be propagated forward. Just as the Kalman gain is multiplied by the measurement innovation in the EKF, here the gain L must be

multiplied by a "propagation innovation" of sorts, $z_k - A_k^n x_k^l$

$$\mathbf{x}_{k+1}^l = \bar{A}_k^l \mathbf{x}_k^l + G_k^l \left(Q_k^{ln}\right)^T \left(G_k^n Q_k^n\right)^{-1} z_k + f_k^l + L_k \left(\mathbf{z}_k - A_k^n \mathbf{x}_k^l\right) \tag{3.45}$$

$$P_{k+1} = \bar{A}_k^l P_k \left(\bar{A}_k^l\right)^T G_k^l \bar{Q}_k^l \left(G_k^l\right)^T - L_k N_k L_k^T \tag{3.46}$$

After propagation, a measurement update is performed when one is available. This follows the same order as the time update step (i.e., particle partition first, then the Kalman partition). The particle weights are updated by evaluating the proposal density function at the every point in N_n space that is represented by a particle. More precisely, every particle $x_{n,(i)}$ is assigned a weight w_i. Since the totality of particles is expected to represent its entire probability space, the weights must necessarily sum to one. As such, a normalization step must be performed:

$$w_i = \frac{w_i}{\sum_{i=1}^{N_p} w_i} \tag{3.47}$$

The measurement update for the linear partition is again very similar to EKF operation, with the exception that there is now an extra quantity $\mathbf{h}_{k+1}$ in the measurement innovation:

$$M_{k+1} = C_{k+1} P_{k+1|k} C_{k+1}^T + R_{k+1} \tag{3.48}$$

$$K_{k+1} = P_{k+1|k} C_{k+1}^T M_{k+1}^{-1} \tag{3.49}$$

$$\mathbf{x}_{k+1|k+1}^l = \mathbf{x}_{k+1|k}^l + K_{k+1} \left(\mathbf{y}_{k+1} - \mathbf{h}_{k+1} - C_{k+1} \mathbf{x}_{k+1|k}^l\right) \tag{3.50}$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} M_{k+1} K_{k+1}^T \tag{3.51}$$

Here, M_{k+1} is the innovation covariance and K_{k+1} is the Kalman gain. The value h_{k+1} accounts for particle partition contributions to the measurement innovation, since it is a function purely of the particle state, as shown in Equation (3.38). The measurement and

prior nonlinear particle state vectors are assumed to have PDFs described below:

$$p\left(\mathbf{y}_{k+1}|X_{0:k}, Y_{0:k+1}\right) = \mathcal{N}\left(\mathbf{h}_{k+1} + C_{k+1}\mathbf{x}^{l}_{k+1|k}\ ,\ M_{k+1}\right) \tag{3.52}$$

$$p\left(\mathbf{x}^{n}_{k+1}|X_{0:k}, Y_{0:k}\right) = \mathcal{N}\left(\mathbf{f}^{n}_{k|k} + A^{n}_{k}\mathbf{x}^{n}_{k|k}\ ,\ N_{k}\right) \tag{3.53}$$

3.5 GNSS/INS Coupling

GNSS is commonly used to augment an IMU, as their error behaviors are highly complementary. IMUs are extremely robust to operating environments, and have low drift over short time periods. However, the pose solution from an IMU alone will diverge over long time periods. On the other hand, GNSS receivers are very stable over long time periods, but are vulnerable to environmental factors like multipath and satellite occlusions that may cause large "jumps" in the position solution from one epoch to the next. As such, coupling the two sensors together yields excellent benefits. While many other sensing solutions exist for the land vehicle navigation problem, this thesis focuses exclusively on GNSS/INS fusion. Coupling strategies for GNSS/INS are covered here in the following section.

3.5.1 Overview of Existing Strategies

Loose coupling is the simplest form of GNSS/INS integration, whereby the position and velocity solutions computed by the receiver are used directly as measurement updates, typically in an EKF. The IMU is used as the time update. The loosely coupled formulation is generally considered to be one of the integration strategies with the lowest fidelity, as it naively breaks the Kalman filter assumptions. Namely, the measurements are serially correlated since they come from the receiver's internal filter and contain errors other than additive white noise. As such, coupling strategies of increasing complexity generally provide greater levels of accuracy, precision, and resilience to environmental errors. Nomenclatures for these more complex coupling strategies often varies between authors.

A *close coupling* still keeps the two sensors relatively separate, with the IMU driving the time update and the GNSS receiver driving the measurement update. The on-board solution from the GNSS receiver is ignored in favor of using its raw measurements to aid the INS. *Tight coupling* strategies go a step further and use kinematic motion estimates from the INS to aid the GNSS tracking loops. This is the point at which difference between various GNSS constellations begin to have a significant impact on algorithmic structure. A more detailed overview of various coupling strategies is provided by [34].

3.5.2 Loosely Coupled GNSS/INS Formulation

Reasoning

This section lays out the the most basic possible navigation filter with GNSS/INS coupling, in order to establish a baseline for improvement attributable to MACIN. It will hereafter be referred to as the "baseline filter". Later, this section will discuss cooperative localization with multiple vehicles. GNSS/INS with multiple vehicles is treated here as an extension of the single vehicle case, and as such this section will lay out the formulation for a single vehicle with one IMU and one GNSS receiver/antenna.

The following is an itemized list of key design choices, with the reasoning for each:

- A *loosely coupled* GNSS/INS integration scheme is chosen, as it is the simplest and most easily realizable formulation (one which can be improved upon later with higher levels of integration).
- The *GPS constellation* is used exclusively for GNSS data in order to further limit confounding factors.
- An *EKF* is used as it is generally regarded as the standard against which potential improvements must be measured.
- The *direct* (or "total state") estimation strategy is used instead of a typical indirect ("error state") filtering strategy. The drawback of this is that linearization is performed

about larger, more nonlinear, and faster varying quantities. However, the direct approach is simpler to understand and implement, and it also provides a cleaner view of the contribution provided by MACIN.

Formulation

The notation here most closely follows the work of Groves and Crassidis [8, 9], except that an EKF is used to compute direct estimates of the state described in Eq (3.54) with no error forumlation. The state to be estimated is composed of:

- $\mathbf{q}_{B/E} = [q_w, q_x, q_y, q_z]^T$, A quaternion describing a rotation from the ECEF frame into the body frame. The IMU sensor frame is assumed to be coincident with the vehicle body frame.
- $\mathbf{r}^E_{B/E} = [r_x, r_y, r_z]^T$, The 3D position vector from the ECEF frame origin to the body frame origin, expressed in the ECEF frame.
- $\mathbf{v}^E_{B/E} = [v_x, v_y, v_z]^T$, The 3D vector describing the velocity of the body frame relative to the ECEF frame, expressed in the ECEF frame.
- $\mathbf{b}_g = [b_{g,x}, b_{g,y}, b_{g,z}]^T$, The gyroscope bias vector modeled as a first order Markov process (see Equations (2.4)). This is expressed in the body (IMU) frame.
- $\mathbf{b}_a = [b_{a,x}, b_{a,y}, b_{a,z}]^T$, The accelerometer bias vector modeled as a first order Markov process (see Equations (2.4)). This is expressed in the body (IMU) frame.
- $\mathbf{k}_g = [k_{g,x}, k_{g,y}, k_{g,z}]^T$, The gyroscope scale factor vector (see Equation (2.6a)). This is expressed in the body (IMU) frame.
- $\mathbf{k}_a = [k_{a,x}, k_{a,y}, k_{a,z}]^T$, The accelerometer scale factor vector (see Equation (2.6b)). This is expressed in the body (IMU) frame.

These variables are then assembled in the state vector in the following order:

$$\mathbf{x} = \left[(\mathbf{q}_{B/E})^T, (\mathbf{r}^E_{B/E})^T, (\mathbf{v}^E_{B/E})^T, (\mathbf{b_g})^T, (\mathbf{b_a})^T, (\mathbf{k_g})^T, (\mathbf{k_a})^T\right]^T \tag{3.54}$$

Since the state vector contains a quaternion, the first four elements must be normalized after each propagation and update step. Furthermore, the orientation of the IMU relative to the ECEF frame is estimated. When the target IMU frame is aligned with the frame of the host platform, no further transformations are needed to compute the vehicle position, velocity, and attitude (PVA). In this case, that means placing the IMU at the coordinate frame origin of the sprung mass (often either the center of mass or halfway between the rear wheels) and aligning its x, y, and z axes with the vehicle forward, left, and down axes respectively. This is common practice in automotive applications because it reduces the complexity of both the measurement and propagation models.

The vector of process noises contains four [3x1] sub-vectors mentioned previously in Chapter 2:

$$\eta = \left[(\eta_{g,\nu})^T, (\eta_{a,\nu})^T, (\eta_{g,u})^T, (\eta_{a,u})^T\right]^T \tag{3.55}$$

The process noise covariance matrix is then:

$$Q = \begin{bmatrix} \sigma^2_{g,\nu} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma^2_{a,\nu} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma^2_{g,u} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma^2_{a,u} \end{bmatrix} \tag{3.56}$$

The direction cosine matrix ("DCM" or "rotation matrix") $C_{B/E}$ describes a three dimensional rotation from the ECEF frame to the Body frame of the IMU (or, equivalently, the attitude of the body relative to the earth). It can be calculated using the attitude

quaterion $\mathbf{q}_{B/E}$ as follows:

$$C_{B/E} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y + q_z q_w) & 2(q_x q_z - q_y q_w) \\ 2(q_x q_y - q_z q_w) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z + q_x q_w) \\ 2(q_x q_z + q_y q_w) & 2(q_y q_z - q_x q_w) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{3.57}$$

The accelerometer measurements, corrected according to Equation (2.6b), are rotated into the ECEF frame:

$$\mathbf{a}_{B/I}^{E} = C_{E/B}\mathbf{a}_{B/I}^{B} \tag{3.58}$$

This DCM can also then be used to rotate the earth's angular velocity vector into the body frame. That is used to determine the angular velocity of the body relative to ECEF from the corrected gyroscope measurement from Equation (2.6a):

$$\omega_{B/E}^{B} = \omega_{B/I}^{B} - C_{B/E}\omega_{E/I}^{E} \tag{3.59}$$

The first time derivative of the hidden Markov state can then be calculated so that the propagation model is defined as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}}_{B/E} \\ \dot{\mathbf{r}}_{B/E}^{E} \\ \dot{\mathbf{v}}_{B/E}^{E} \\ \dot{\mathbf{b}}_{g} \\ \dot{\mathbf{b}}_{a} \\ \dot{\mathbf{k}}_{g} \\ \dot{\mathbf{k}}_{a} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Xi(\omega_{\mathbf{B/E}}^{\mathbf{B}})\mathbf{q}_{B/E} \\ \mathbf{v}_{B/E}^{E} \\ \mathbf{a}_{B/I}^{E} + \mathbf{g}_{B}^{E} - 2\Omega_{E/I}^{E}\mathbf{v}_{B/E}^{E} \\ \frac{-1}{\tau_g}\mathbf{b_g} \\ \frac{-1}{\tau_a}\mathbf{b_a} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{3.60}$$

The skew symmetric matrix representing the angular velocity vector of the earth in ECEF frame is a function of the rotation magnitude of earth with respect to the inertial frame

$\omega_{E/I} = 7.292115\,\mathrm{e}\,{-5}\ rad/s$, and is described by:

$$\Omega^E_{E/I} = \begin{bmatrix} 0 & -\omega_{E/I} & 0 \\ \omega_{E/I} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.61}$$

The mechanization matrix $\Xi(\mathbf{w})$ for relating the time derivative of the quaternion to Cartesian angular velocity ω is:

$$\Xi = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{3.62}$$

The acceleration on the IMU due to gravity $\mathbf{g}^E_B$ is expressed in ECEF as:

$$\mathbf{g}^E_B = \gamma^E_{B/I} + \omega^2_{E/I} I_{3x3} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \mathbf{r}^E_{B/E} \tag{3.63}$$

$$\gamma^E_{B/I} = -\frac{\mu}{r^3}\left(\mathbf{r}^E_{B/E} + J_2 \frac{3R_0^2}{2r^2}\left(I_{3x3}\begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} - z_{scale}\mathbf{r}^E_{B/E}\right)\right) \tag{3.64}$$

$$z_{scale} = 5\frac{\left(\mathbf{r}^E_{B/E,z}\right)^2}{r^2} \tag{3.65}$$

$$r = ||\mathbf{r}^E_{B/E}|| \tag{3.66}$$

where $R_0 = 6378137$ is the WGS84 equatorial radius in meters, $\mu = 3.986004418\,\mathrm{e}\,14$ is the WGS84 earth gravitational constant, $J_2 = 1.082627\,\mathrm{e}\,{-3}$ is the WGS84 earth second gravitational constant.

A standalone GNSS receiver outputs measurements of the antenna's kinematics. More specifically, the GNSS receiver outputs the 3D position of the antenna relative to the ECEF frame and expressed in the ECEF frame ($\mathbf{r}^E_{A/E}$), and the 3D velocity of the antenna relative to the ECEF frame and expressed in the ECEF frame ($\mathbf{v}^E_{A/E}$) As such, the frame of the

antenna must be related to the vehicle body frame. The transformation from the vehicle body frame into the GNSS antenna frame, for the purposes of the present work, is completely parameterized by the 3D position vector from the vehicle body to the antenna, expressed in the body frame ($\mathbf{r}^B_{A/B}$). This is referred to as the "lever arm", and no orientation information is required.

$$\mathbf{y} = \begin{bmatrix} \mathbf{r}^E_{A/E} \\ \mathbf{v}^E_{A/E} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^E_{B/E} + \mathbf{r}^E_{A/B} \\ \mathbf{v}^E_{B/E} + C_{E/B}\left[\omega^B_{B/E} \times \mathbf{r}^B_{A/B}\right] \end{bmatrix} \tag{3.67}$$

Calculation of $\mathbf{r}^B_{A/B}$ may be done either as part of a separate calibration process, which produces static constant calibration values, or online in an autocalibration approach where the value of $\mathbf{r}^B_{A/B}$ is placed into the filter's state vector and varies over time. Static values were used for the lever arm in this thesis.

3.5.3 MACIN State Formulation

The core dilemma motivating this thesis is that the true PDF of position on a road network is both mutli-modal and discontinuous (discussed in the next chapter). As such, it cannot be adequately modeled with a mean and standard deviation alone. Therefore position is chosen for the particle partition. The state vector formulated in Equation (3.54) is then re-arranged following the RBPF partition scheme:

$$\mathbf{x} = \left[\begin{array}{c} \mathbf{x}^l \\ \hline \mathbf{x}^n \end{array}\right] = \left[\begin{array}{c} \mathbf{q}_{B/E} \\ \mathbf{v}^E_{B/E} \\ \mathbf{b_g} \\ \mathbf{b_a} \\ \mathbf{k_g} \\ \mathbf{k_a} \\ \hline \mathbf{r}^E_{B/E} \end{array}\right] \tag{3.68}$$

Note that all remaining state variables stay inside of the linear (Kalman) partition. In the literature, it is fairly common to see the attitude vector chosen for the nonlinear partition when authors implement an RBPF for the inertial navigation problem. As Vernaza, et al. note [35], not only are all 3D attitude representations nonlinear, they are all nonlinear *within nonlinear spaces*. However, the navigation problem becomes linear when attitude is no longer a consideration. In other words, if the body frame was modeled as a point mass rather than a Cartesian coordinate frame, then strapdown inertial navigation would be a linear problem. It is for this reason that Gustafsson, et al., ignore the attitude component in [32] for their investigation of particle filter applications to both navigation and tracking problems for various platforms. If this were the case, a traditional Kalman filter would suffice as well. So it is easy to see that since the nonlinearities arise from the attitude partition, moving *attitude*–and not position–into the nonlinear partition makes sense.

For the purposes of this thesis, MACIN leaves attitude in the linear partition of the RBPF and estimates the 3 position states variables in the particle partition. This is similar to the approach Ryan took in [12], where the two earth-tangent position dimensions are Rao-Blackwellized because they are measured by ranging beacons. There are multiple reasons MACIN does this. Investigating the effects of imposing a map-based constraint on the position PDF is a core focus, and MACIN has nothing new to offer the attitude PDF which further departs from the Gaussian assumption. Additionally, the well-known computational burden of particle filtering makes minimizing the size of the nonlinear partition a priority, and the addition of attitude alongside position would require increasing the number of particles from 10^3 to 10^6. Lastly, no 3D attitude truthing system was available to concurrently run on multiple participant vehicles. As such, no validation would be possible to quantify the effect of Rao-Blackwellizing the attitude. For these reasons, only the 3 position state variables are moved to the particle partition.

Chapter 4

Map Usage in Navigation, Tracking, and Localization

Use of map or survey data to refine navigation solutions varies in both the type of road information which is used, as well as how the information gets used. For this thesis, only a priori map techniques are considered, and online mapping or simultaneous localization and mapping (SLAM) strategies are neglected.

4.1 Map/Road Representations

Basic strategies for representing road survey information can be consolidated into 4 categories:

1. **Line Graph**: A series of locations corresponding to road or lane centers, with connectivity information relating points to one another to represent driveable routes.

2. **Nonlinear Curves**: A bank of nonlinear equations describing road surfaces, possibly including lane markings, lane centers, or road centers.

3. **High-Density (HD) Databases**: LiDAR data that has been aligned to a local navigation frame. A post-processing procedure to abstract some of the features may or may not be performed.

4. **Visual Map**: A collection of camera images that have been projected into some local mapping frame with various post-processing techniques to allow localization using visual features.

HD map matching is relatively new, and has become ubiquitous in the field of autonomous vehicles, where GPS availability is often limited at best. Rather than taking GPS

survey points or aligning a series of arcs to aerial imagery, a LiDAR- and/or camera-enabled vehicle is driven along the target road network. The key distinction here is that this form of map building and matching need not necessarily be global. That is, localization alone is sometimes sufficient, particularly in the robotics field. However, when the survey vehicle(s) are equipped with some absolute positioning solution, though, global navigation based solely on the map is of course possible. During the survey procedure, sensor data is transformed into the vehicle body frame using mounting calibration, and then transformed into the map frame using the vehicle's localization solution. Often post-processing is performed to extract visual semantics or feature abstractions which are used to aid in the map matching process.

Once an HD map is built, localization typically involves estimating a translation and rotation to align perceived LiDAR pointclouds and/or camera images with the map. The transformation required for this alignment is then the pose of the vehicle within the map frame. It is worth noting that HD maps features often contain road contour and routing graph information similar to more primitive maps, which allows leveraging the techniques for those data types as well. Line graph and nonlinear curve methods are discussed further in Section 4.3. For the purposes of this thesis, LiDAR and visual maps are not considered.

4.2 On-Road Determination

Inferring whether a vehicle is actually traveling on the mapped road network is a difficult problem. An errant position estimate may show the vehicle 10 meters removed from the road laterally when it is actually traveling well within lane boundaries. Alternatively, the same position estimate may be perfectly correct, and the ego vehicle truly is 10 meters away from the road in question, and traveling on some unmapped portion of roadway. Determining which of these is the case is of utmost importance before the map can be used to refine any navigation solution. A common solution is to assume that the host vehicle is traveling on the nearest roadway until the uncorrected position estimate exceeds some type of confidence region, either probabilistic or heuristic, and thereafter to classify it as "off-roadway" [36].

Many methods simply make the assumption that the vehicle is indeed confined to known lanes.

Increasing availability of computer vision and perception in automotive applications has made this task considerably easier to observe directly. Finding lane markings, asphalt, or other road features within the view of either a LiDAR device or camera allows probabilistic determination of whether a vehicle platform is traveling on *some* roadway. The question then becomes, "on *which* roadway is the vehicle traveling?" The uncertainty of whether a host vehicle is traveling outside of the road network is neglected here after, and it is always assumed to be traveling in one of the mapped lanes. Given this assumption, the use of maps as *constraints* becomes the core focus.

A natural weakness is immediately obvious in that intersections do not have lane boundaries in the traditional sense, and that a vehicle traveling inside of an intersection is not inside of any one lane. In autonomous platforms, which must be controlled through the intersection, a trajectory is often already mapped from each possible starting lane through the intersection into each possible ending lane. The solution cannot be constrained while traversing the intersection with this approach, but the lane may be continuously tracked such that once back on the nominal roadway, the correct lane is already known and need not be re-determined before constraints can once again be applied. The drawback is that the shape of lane connector curves must be heavily scrutinized to ensure they are realistic. An easier approach is to simply ignore the period of time and allow the uncertainty to grow through the intersection and then re-converge afterward [37]. Since the time and distance spent in intersections is typically quite small, the drift is low. This is what MACIN does, as discussed later in Section 4.5.

The remainder of this chapter will examine two architectures for map usage in navigation, tracking, and localization:

- "*Map Matching*", in which map information is used to modify an existing solution. This is discussed in Section 4.3 and depicted as a simplified flow chart in Figure 4.1.

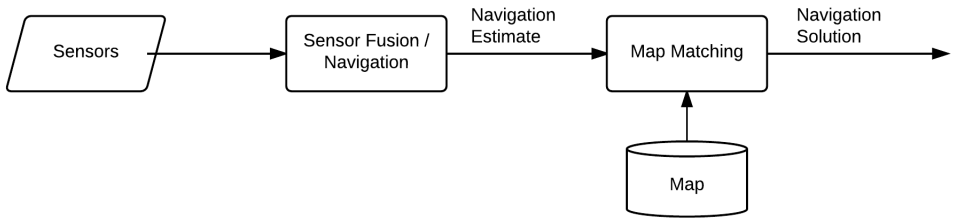

Figure 4.1: Flow chart depicting the general operation of open loop map matching algorithms

- "*Map Aiding*", in which map information is integrated into the navigation filter prior to the navigation solution. This is discussed in Section 4.4

4.3 Map Matching

The simplest method of incorporating map information is by placing a map module downstream from the navigation algorithm in an "open loop" configuration as depicted in Figure 4.1. This is referred to as "map matching" and may be performed either on- or off-line. Once sensor information has been fused, the resultant position/velocity/attitude estimate is then projected onto a known roadway. The principle problem is then to determine the correct portion of the map (road) on which the vehicle is traveling.

4.3.1 Line Graph Map Matching

Early work on map matching was primarily geometric in nature, and focused on using line graphs (nonlinear curves were used later as well). These geometric methods can be divided into three categories:

1. *Point-to-Point*: Every position solution is matched to a survey point.

2. *Point-to-Curve*: Every position is matched to a line between neighboring survey points (or "link").

3. *Curve-to-Curve*: A history of prior positions estimates is matched to possible successive positions which all lie on known roads

The nature of graphs enables leveraging topological (connectivity) information to significant benefit, and this is a key distinguishing factor between the above 3 categories. For example, consider the case of an urban area in which the roads take a grid shape. From a purely intuitive standpoint, it is nonsensical to say that a vehicle "jumps" between parallel, unconnected roads between any two adjacent time epochs. However, a naive algorithm can easily compute this as a solution if it does not consider that no connection between the parallel roads exists. As such, the existence of possible routes between successive map matches is often considered in order to eliminate implausible solutions. This also has the potential to greatly reduce processing time. The drawback is that one wrong match could potentially eliminate the possibility of future correct matches. Curve-to-curve algorithms necessitate using this information, whereas point-to-point and point-to-curve algorithms may or may not ignore the topology. A brief review of the literature relevant to geometric line graph methods is presented below.

A great body of work on creating open-loop geometric map-matching methods has been built by Quddus, et al [38–43]. In [38] they perform an extensive literature survey and present a new method for link selection which heavily relies on matching vehicle heading to the direction of the roadway. This adds robustness to large position errors during intersection navigation. In [39] they build on this by including confidence weighting for link selection using position uncertainty ellipses. Then, in [40] they present an integrity analysis with specific consideration for the roles multiple candidate lanes play in matching error. An in-depth survey of limitations of the field was presented in [41] with attention to the need for confidence indicators when used in intelligent transportation systems. They expand their work to use the map topology more heavily in [42] so that complex features such as parallel roadways and turn restrictions imposed by traffic law could be considered. And lastly, in [43] they retroactively match historical paths to optimize the matched trajectory using classical path-planning techniques, and show its effectiveness with sparse data from low-frequency GNSS.

A comparison of 4 graph-based methods was performed by White et al., [44]: 3 point-to-curve algorithms and 1 curve-to-curve algorithm. The first and simplest is point-to-curve matching wherein the position estimate is "snapped" to the closest graph edge using minimum distance alone. The second builds on this by considering that the vehicle heading must be close to that of the matched roadway. The third expands further by leveraging topological information to narrow the set of candidate nodes and edges by only considering neighbors from previous matches. The fourth, a curve-to-curve algorithm, is the batch processing adaptation of the previous method with a tree search over the history of the navigation solutions. None of the algorithms presented are probabilistic in nature. The author comes to the conclusion that the second approach performs most reliably.

A good way of preventing correct roadways from being discarded in curve-to-curve matching is by performing shape comparisons alone, typically through the use of curve distance metrics. A very common metric for comparing two curves is the *Fréchet distance*. Alt et al. [45] provide mathematical proof of a generalized planar map matching strategy for curves composed of series of 2D positions using this metric and give the following illustrative definition:

> Suppose a person is walking a dog, the person is walking on the one curve and the dog on the other, and the person is holding the dog at a leash. Both are allowed to control their speeds but they are not allowed to go backwards. Then the Fréchet distance of the curves is the minimal length of a leash that is necessary for both to walk the curves from beginning to end.

This metric is tolerant to small irregularities, but is still sensitive to constant offsets (such as a curve formed from GNSS-only position estimates) as well as drift (such as a curve formed from a dead-reckoning system). A simpler curve-to-curve algorithm is the one used by White [44] (discussed previously), and simply takes each point from the curves and projects it onto the other curve (using a point-to-curve technique) in sequence. However, White's approach is more sensitive to outliers.

Davidson et al. [46, 47] use point-to-curve matching atop a particle filter which fuses GNSS with dead-reckoning from a reduced-axis IMU. Their link selection strategy of picking the link by orthogonally projecting the prior estimate onto candidate links and selecting the one whose projection is shortest is very common. This strategy is one of several steps in MACIN's map fusion approach, which will be detailed in Section 4.5.

Also relevant to this thesis is the work of Chu et al. [48], who employ a loosely coupled GPS/INS filter with IMU bias and scale factor correction. They use the trajectory of the INS over time in a curve-to-curve map matching strategy, with the addition of an extra dimension for yaw angle in the matching algorithm. They lose accuracy, though, by only matching points to roadway segments instead of considering positions in the coordinate frame of the roadway.

4.3.2 Nonlinear Curve Map Matching

The most popular types of nonlinear roadway representations are:

1. **Clothoids and/or clothoidal splines**: These are used heavily in computer vision roadway detection algorithms. Clothoids (also called "Euler spirals") very closely approximate the underlying design procedure used to build roads in the first place [49,50].

2. **Polynomial splines**: Generally this is the best method of extending a line graph beyond the piecewise linear assumption, and typically sees use as an "add-on" feature. Cubic splines are the most popular [51, 52].

3. **Arc splines**: Splines made of circular segments are very easily scaled and translated to represent lane markings and adjacent lanes which are parallel to a particular lane center [53, 54].

One key advantage of representing roadways as a series of nonlinear arcs is that camera information is more naturally integrated into the map matching algorithm. The contour of

the road surface or lane markings can be fitted to a curve in the camera or vehicle frame, then matched to existing contours using a distance and orientation minimization.

In [51], Pink and Hummel use GPS as the sole measurement source in an EKF. They then estimate 2D pose by modeling velocity and yaw rate as random walk processes for the time update. They utilize a topological line graph approach and a hidden Markov model to determine the probability of being on each possible link within an area of interest, with particular focus on robustness to intersection traversal. The authors then use "shape points" to interpolate the vehicle position between graph nodes using cubic splines. A technique for handling U-turn maneuvers is also presented. The nonlinear curve matching step then becomes a sort of add-on to the the underlying approach which uses probabilistic point-to-curve matching.

4.4 Map Aiding

When a map database is used within the navigation algorithm to refine or fuse sensor information, the task is more complex. The level of coupling can be thought of as "closed loop", to borrow a term from classical controls, and will be referred to herein as "map aiding".

One of the earliest usages of map databases was in the Ground Moving Target Indicator (GMTI) approach, which was used primarily for military applications in tracking ground vehicle movements using ground-based RADAR installations and generally worked with accuracies on the order of meters to tens of meters. The GMTI approach generally neglects vertical information and assumes that map information is available in the form of a line graph [55–62]. If one is able to determine with complete certainty the link of the line graph on which a target vehicle is traveling, then the vehicle motion could be modeled as occurring in the frame of the roadway, that is, either lateral or longitudinal to the roadway. With the goal of determining a likely link, multiple potential links are considered at once, and motion along each is modeled in the same EKF state vector. In this way, *each link of the line graph*

is a separate motion model, and the technique of maintaining multiple hypotheses simultaneously (with each hypothesis following a different model) in order to pick one that is most likely is referred to as the Multiple Model ("MM") approach. The RADAR measurement forms the measurement update for each model's substate, and either velocity or acceleration become relegated to representation via process noise, depending on the author. All that is left is picking the most likely link/model. The general case where a discrete state (also "mode", or Markov chain) sets the dynamical model for a continuous state is also known as a jump-Markov system, which is examined thoroughly by [63].

When one considers that each link is connected to at least one other link, and that with enough time a moving vehicle will transition from one to another, modeling transitions between links probabilistically is a natural next step. The probability density function for a given state becomes dependent upon the transition probability function of the discrete mode state. Using topological information to aid in probabilistic selection of the correct model is referred to as an Interacting Multiple Model approach ("IMM") approach. Bar Shalom et al. worked on the IMM approach extensively [64, 65].

For a single target, maintaining a fixed set of possible models like IMM does is no longer necessary with the use of topological connectivity information. To illustrate, if a vehicle is located at one end of a map, then maintaining hypotheses on the other end of the map is a waste of resources. The solution is to dynamically add and remove models from consideration as the vehicle progresses, using the topology to intelligently decide at any time which models to consider. This is referred to as the "Variable Structure Interacting Multiple Model" (VS-IMM) approach, and as the name implies, the consequence is that one must implement a dynamically varying state vector. Kirubarajan et al., introduced the VS-IMM in [61] and estimated planar position and velocity in the state vector, with acceleration as process noise.

Arulampalam et al., in [56], took the basic VS-IMM further by employing a particle filter instead of an EKF. Perhaps most importantly, they demonstrated a probabilistic method of

determining whether a vehicle was traveling on or off the road network. Other approaches to GMTI typically stem from [56,61]. For instance, Payne and Marrs [55] subsequently extended Arulampalam's work by using a novel Unscented Particle filter for the VS-IMM solution instead of a standard particle filter. Particlesr were used to represent uncertainty about the correct model (map section), while a traditional UKF is used to estimate 2D position and velocity with noise modeled along the coordinate system of the roadway. Jabbour [66] takes the VS-IMM approach, but uses on-board GNSS and dead reckoning for propagation instead of external RADAR. They contribute an excellent analysis of the assumption that the position probability density function is two dimensional Gaussian within the coordinate system of the roadway, along with a novel method of assigning probabilities to each candidate link.

Outside GMTI, probabilistic approaches to closely coupling map constraints with inertial navigation vary widely. Hall [36] compares two multiple hypothesis filters, a grid filter and a particle filter, for fusing wheel encoders, GPS, and a single gyroscope orientated along the vertical body axis. The author notes that when used in map constraint systems, particle filters are abnormally vulnerable to clustering. Clustering is a phenomenon wherein the particle population collapses to a very small portion of the state space. Map constraints exacerbate this because when a population collapses to the wrong location, subsequent turns are deemed impossible by the map, and particle weights all approach zero. Proper tuning of process noise can prevent this, but grid-based filters, on the other hand, are naturally robust to clustering because they always maintain a fixed search area.

Smaili et al. [67, 68] focus on fusing GPS measurements with wheel encoder odometry using jump-Markov map representations. The addition of wheel encoder information allows them to add heading to the state vector, since differencing left and right encoder readings produces a rotational measurement when combined with vehicle geometry. It also allows for continuous estimation, even during GNSS outages, since propagation of odometry may be performed along the roadway to curtail drift.

Fouque et al. [37] use the map as a heading pseudomeasurement inside an EKF which relies primarily on dead reckoning with closely coupled aiding from GNSS. They present a sophisticated measurement noise computation model as well as an innovative method of using both linear and angular distance in the same cost function to compute map link likelihoods. An interesting facet of this work is the possibility of performing map error detection.

Ray [29] uses a particle filtering map constraint model very similar to that in MACIN, but for indoor pedestrian navigation applications. The state variables contained within this particle filter are simply two-dimensional pose: north and east position as well as heading. Whenever a particle violates a map constraint, in this case by crossing an impassable barrier such as a wall, it is assigned zero weight and resampled. The strength of this is that it achieves the desired performance by only applying the binary constraint, and does not impose any false information on the proposal distribution. An example of imposing false information is how GMTI assumes all vehicles travel exactly along the line of the roadway. MACIN incorporates Ray's approach to applying map constraints, as it naturally integrates into the RBPF. Furthermore, it implicitly captures mutli-modality in the position PDF, whereas IMM requires explicit handling of mutli-modality, which is explained in detail in the next section.

4.5 Map Usage in MACIN

One of the requirements specified at the outset of this thesis was that MACIN must use the sparsest possible lane-level map. Here, "sparse" refers to using as little information as possible about each lane. This is done to maximize the number of use cases. Requiring specialized information narrows the number of potential applications and the possible coverage areas, and decreases the likelihood that pre-existing maps can be used.

The most widely available data generally takes the form of a collection of points which lie on lane divider markings, or perhaps lane centers. These points may be used to construct line strings. Whether the data represents lane dividers or lane centers is not particularly

important, since one may be used to construct the other. The only caveat is that lane center information must be accompanied by lane width at each point, whereas lane width can be computed directly from the distance between lane dividers. For this thesis the roadway is represented using lane center points and width measurements. Once a map is built accordingly, MACIN's approach to using map data is as follows:

- For each position particle, find the nearest map link.
- Assume the particle represents a hypothesis that the vehicle is travelling on the nearest link.
- If the particle lies in a position such that the vehicle would not be inside the link's lane boundaries, set the particle weight to zero.

4.5.1 Per-particle Lane Determination

One of the assumptions made earlier was that the system already has some way of knowing whether it is within lane boundaries. It is important to note the system is not required to know *which* lane it is in, only that it is completely inside of *one of them* and not straddling a lane boundary. The easiest method to accomplish this is by using computer vision to identify lane lines and then determine whether or not the vehicle is straddling them. It is worth noting that once such a system identifies the lane boundaries in the camera frame it is relatively straightforward to transform that observation into the ego vehicle body frame. One could then use this relative position measurement as a particle update. Ryan does this with fixed ranging beacons in [12]. However, building and leveraging the computer vision or LiDAR technology to produce such a relative distance measurement is outside the scope of this thesis.

With this high-level in-lane determination in place, a hard constraint on position can now be imposed in the two dimensions of the earth-tangent plane. This will take the form of a test to determine whether a given candidate position adheres to the map constraint. First,

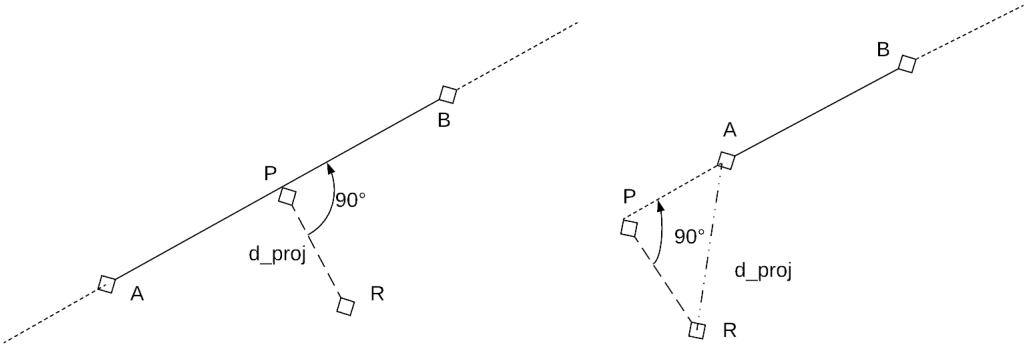

Figure 4.2: Perpendicular projection illustration for cases where the query point $\bar{R}$ lies within the link (left) and outside of the link (right).

a straightforward point-to-line map matching technique is employed to select the nearest connected pair of nodes (which form a "link"). The nearest link is that which minimizes the perpendicular projection distance d_{proj}, illustrated in Figure 4.2 and formulated in Equations (4.1). Making a correct choice of the nearest link is simply a matter of finding the value of d_{proj} for all links within a reasonable search radius, and picking the link with the lowest associated value of d_{proj}. This thesis used a 15.0 meter search radius.

$$d_{proj} = \left\| \bar{P} - \bar{R} \right\| \tag{4.1a}$$

$$\bar{P} = \bar{A} + \alpha \bar{C} \tag{4.1b}$$

$$\alpha = \frac{\left(\bar{R} - \bar{A}\right) \cdot \bar{C}}{\gamma} \tag{4.1c}$$

$$\gamma = \bar{C} \cdot \bar{C} \tag{4.1d}$$

$$\bar{C} = \bar{B} - \bar{A} \tag{4.1e}$$

Topological constraints can be added by simply omitting links which are not connected to the previously matched link. MACIN does this on a per-particle basis. When selecting links to consider for a single particle, any link is rejected if it is not adjacent to the link that was assigned to the same particle in the previous estimation epoch.

For many links on a given map, the perpendicular projection may lay outside the link's defining nodes $\bar{A}$ and $\bar{B}$, as seen in Figure 4.2. This only occurs on links for which another link's projection is smaller, meaning that the projection point can be "snapped" to the nearest node, since it is the closest projected point that lies on the node. For any case where this happens, another link will have a projection within $\bar{A}$ and $\bar{B}$, and produces a smaller d_{proj}. The snapping algorithm is given in Algorithm 2.

Algorithm 2 Perpendicular projection out of bounds handling.

```
1: procedure CHECK BOUNDS
2:     β ← (P̄ − Ā) · C̄
3:     if β < 0 then
4:         P̄ ← Ā
5:     if β > γ then
6:         P̄ ← B̄
```

The candidate position adheres to the map constraint so long as the projection distance is less than or equal to the selected link's on-network distance, $d_{ONN} = (d_{lane} - d_{veh})/2$. The on-network distance is a function of the host vehicle width (d_{veh}) and the lane width (d_{lane}), which is assigned on a per-link basis to deal with natural variations in lane width as codified later in Algorithm 6. An overhead view of the region of non-zero probability ("on-network region") is depicted in Figure 4.3.

Working off of this information alone, there would be no positional information in the vehicle's longitudinal direction and the vehicle's lateral position would have a probability density function with uniform values between two points that are fixed to the road surface. The two fixed points would be symmetric about the lane's center line (or longitudinal axis) and their distance would be a function of the lane width and the vehicle width. The remaining region of non-zero probability is illustrated in Figure 4.4.

To illustrate further, a representative particle cloud is shown in Figure 4.5, with particles that violate the constrain shown in red, and particles that satisfy the constraint shown in blue. Those that violate the constraint have their respective weights set to zero and are

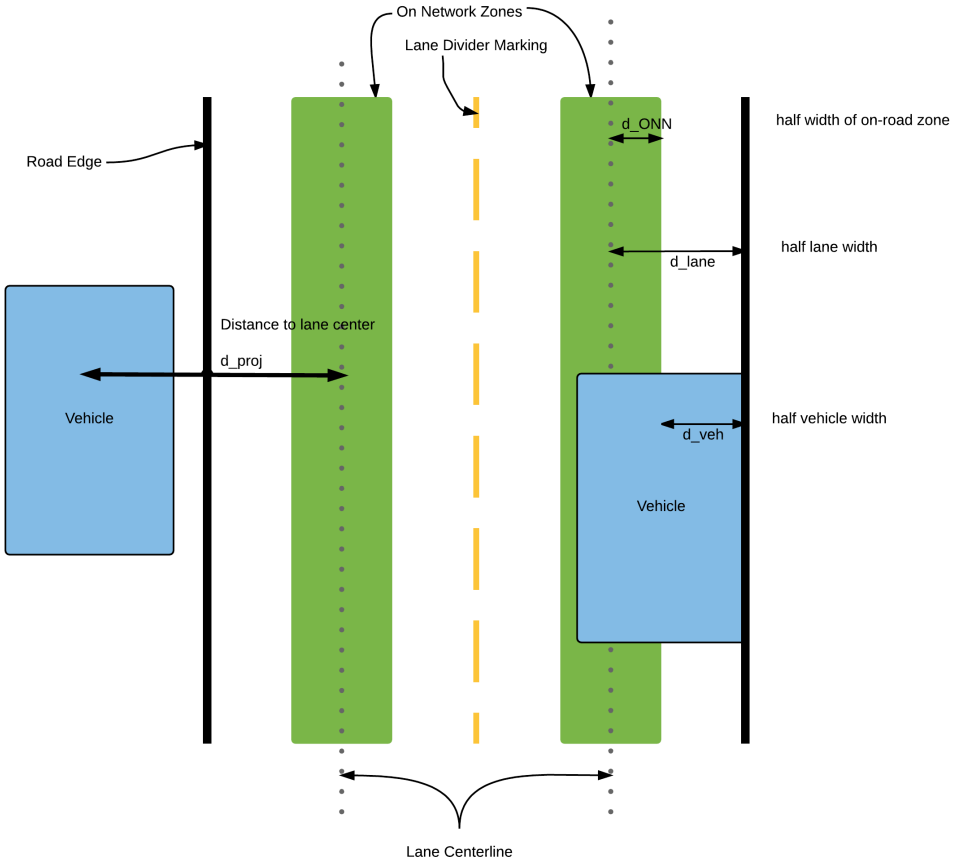

Figure 4.3: Strategy for determining whether a vehicle is within the boundaries of a given lane

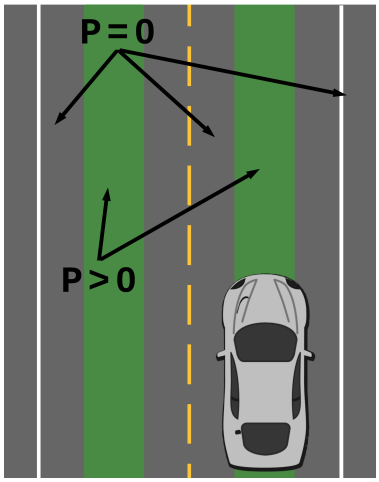

Figure 4.4: Probability areas for a vehicle position within lane

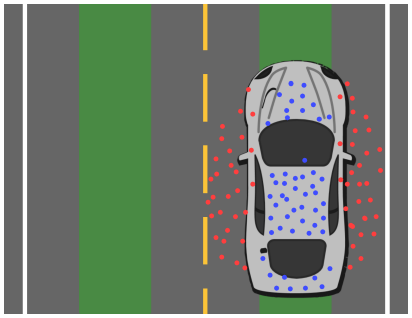

Figure 4.5: Depiction of position particles with those that violate the map constraint in red, and those that satisfy the map constrain shown in blue.

resampled on the next filter iteration, whereas no action is taken for the particles that satisfy the constraint.

The map aiding strategy used by MACIN may be consolidated to: *the probability for any state whose position violates a lane boundary constraint is zero.* MACIN makes no attempt to impose artificial probability distributions onto any state space, and does not update non-zero probability state hypotheses with false information. This means that common practices such as using the roadway's direction as a heading measurement or providing a psuedomeasurement placing the vehicle in the center of the lane are disallowed. These techniques would add constant errors in the nominal case, where a driver simply stays closer to one side of the lane than another. In the case of stop-and-go driving at low speed, heading may differ greatly from the roadway direction even when the vehicle stays within lane boundaries.

As such, the map update step of MACIN imposes a multimodal uniform distribution atop the position PDF, which would otherwise be a Gaussian distribution (as this is an operating assumption of the EKF). An illustration of this overlay is provided in Figure 4.6.

To handle the case of intersections, links are placed to connect the adjoining roadways, but all lane widths are assigned to be infinite, so erroneous constraints are not applied. This approach was discussed in the beginning of Chapter 4. As each vehicle traverses the intersection, its performance degrades due to the lack of constraining information.

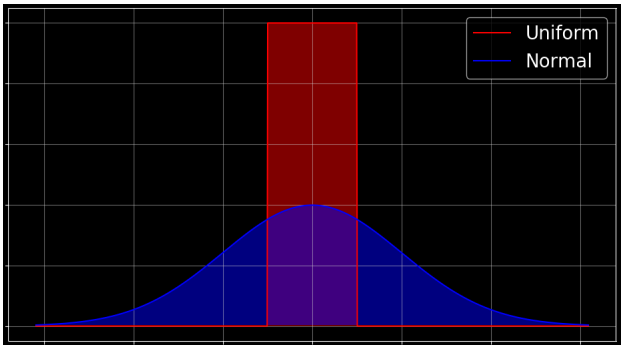

Figure 4.6: Typical shapes of unimodal Gaussian and uniform densities overlaid.

4.5.2 Final Lane Determination

Since the map update removes all particles that do not lie within lane boundaries, each remaining particle corresponds to a single lane. Since the particle cloud is a representative sampling of the position PDF, all possible lanes are represented by at least one particle, and some lanes are represented by multiple particles. Lanes with no particles lying in them have zero probability of being the correct lane. Since each particle has an importance weight attached, the weights of each particle in a given lane can be summed and the resultant value will be the probability that the ego vehicle is driving in that lane. The lane with the highest probability is then the estimated ego lane, and the sum of the probabilities of all candidate lanes is, of course, one. After selecting a lane, the particle population should be constrained only to particles which lie in that lane when computing a value for $\mathbf{x}_{k+1}^n$ in Equation 3.40. Otherwise, a mean solution would often end up lying between lanes and defeating the purpose of map constraints.

This is illustrated in Figure 4.7. Some particles (shown in red) may satisfy map constraints for the incorrect lane. However, the final position solution is only computed over particles whose lane receives the highest summed weight (shown in blue). The other particles (in read) are excluded from the output solution, but they are maintained within the overal particle population in order to maintain multiple concurrent lane hypotheses.

Defined above is a method of probabilistically determining the lane the ego vehicle lies in, a method which naturally arises from the choice to model map information as a pure

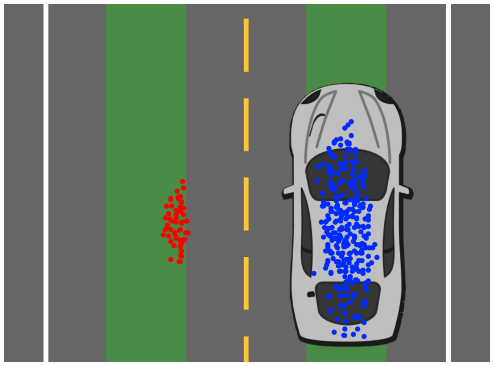

Figure 4.7: Selecting particles that only lie in the chosen lane.

constraint at the particle level. This intrinsically produces not only a instantaneous solution that lies in the correct lane, but a solution which historically has always been in the correct lane due to the recursive nature of the filter. No explicit step needs to be taken to optimize a buffer of previous poses, as each particle already represents a time evolution where all errant positions violating map constraints have been eliminated. One key caveat is that when any of the particles lie on an intersection lane (one whose width is infinite), the solution is undefined.

4.5.3 Discussion

It is worth noting that only two of the six degrees of vehicle freedom are constrained under this map aiding. In the literature, yaw is commonly constrained alongside the two earth tangent dimensions. However, this requires either imposing an artificial probability function, or using another sensor to create an orientation measurement. It is common to assume that all vehicles face the direction of the roadway, and set the standard deviations of the heading pseudomeasurement using some heuristic [43, 68]. If this approach is not used, some external sensing system that produces estimates of the heading of the roadway relative to the ego vehicle would be required in order to constrain yaw with the mapping. Since none of the assumed sensing capabilities include this, and artificial probability functions are forbidden, map constraining of the yaw direction is not used.

For roll, pitch, and vertical position, it is very tempting to constrain the vehicle's pose in these dimensions so that the vehicle is forced to always lie on the roadway (or some offset from it) and be oriented parallel to the roadway. When this is done, the motion of the vehicle is always along a sparsely modeled manifold in 3 dimensional space, similar to an ant traveling the length of a ribbon. In the nominal case, this would theoretically allow for massive reduction of errors. However, it would also require producing a dynamic model for suspension motion between the sprung and unsprung masses in order to compute offsets between the roadway and the sensor frame. The technique of "clamping" the vehicle to the roadway is not used here in order to avoid dependence on a vehicle dynamic model.

Lastly, it is important for the reader to understand that this thesis presumes the availability of perfect map information and neglects mapping errors. This is made possible since all map survey points were collected using a real time kinematic (RTK) GPS survey station, which has errors on the centimeter level, one order of magnitude smaller than the performance of MACIN. In practice, if one were to generate map data using some estimation technique with errors on the same order of magnitude as the filter that used the map data, this uncertainty must be accounted for. Furthermore, if survey points were spaced too far apart, the piecewise linear model would introduce more errors by failing to adequately approximate the lane geometry. This thesis leaves consideration of such mapping errors to future work.

Chapter 5

Cooperative Navigation Techniques

5.1 Introduction

Generally speaking, cooperative navigation involves using measurements or information from one or more secondary vehicles to aid the navigation solution for the ego vehicle. In order for information sharing to improve positioning accuracy, there must be some way to express the location of one participating vehicle in terms of the location of another, such that they become probabilistically linked. The most common method of doing this uses directly cooperative measurements, where two vehicles that are sharing information directly observe each other and share these measurements. Consider as an example the case where the positions of two vehicles, i and j, are being estimated independently in a single standard EKF. The state vector and covariance matrix would be:

$$x = \begin{bmatrix} x_i \\ x_j \end{bmatrix} \tag{5.1}$$

$$P = \begin{bmatrix} P_i & \mathbf{0} \\ \mathbf{0} & P_j \end{bmatrix} \tag{5.2}$$

The states and covariances evolve independently. When vehicle j receives a measurement $y_j = h(x_j)$, it only improves the information about state x_j, and only covariance P_j is decreased, such that $P_i^+ = P_i^-$. That is, the posterior value of P_i is equivalent to the prior value of P_i, meaning no new information is gained.

Now consider the use of a relative measurement of the form $y_{ij} = h(x_i, x_j)$. Since the measurement is a function of both states, the posterior covariance would then be:

$$\begin{bmatrix} P_i & P_{ij} \\ P_{ij}^T & P_j \end{bmatrix} \tag{5.3}$$

where $P_{ij} \neq \mathbf{0}$. Now, when the measurement y_j for vehicle j is applied, the covariance P_i for vehicle i is decreased as well, because the two vehicles are statistically "tethered". This principle scales as more vehicles participate in the cooperative localization network.

To mentally conceptualize cooperative localization as a physical system, Patwari et al. [69] offer a fitting illustration which depicts the entire network as a mass-spring system from classical mechanics. Each participating vehicle may be thought of as a mass, and each cooperative relationship (inter-vehicle ranging, for instance) between two vehicles may be thought of as a spring with an unknown equilibrium point which is connected between the two masses. Each spring's coefficient of force is comparable to the relative measurement confidence, and infrastructure waypoints such as base stations or road side units may be thought of as masses that have been fixed in place and are immovable. The entire system then has a natural settling point, where all the masses reach steady states which are a function of their connection to neighboring masses. While this is a rather simplistic illustration (for instance, it does not account for participant dynamics), it serves a useful purpose in showing how each vehicle state becomes inextricably linked to the swarm state.

5.2 Architectures

Broadly speaking, cooperative system architectures can be divided into two categories, differentiated by the location where navigation estimation is performed. When a single estimation algorithm takes responsibility for computation and uses two-way communication to exchange measurements and estimates with each particpant, the architecture is "centralized". A common use for this is air traffic control. When each participant performs

computations onboard and receives measurements directly from its peers without a master actor, this is a decentralized architecture. Research into systems for consumer mass market automobiles often uses decentralized architectures.

One critical drawback of decentralized approaches is the "double counting problem", in which Bayesian independence assumptions are violated when each filter uses other filter states as measurements. Mokhtarzadeh [70] shows that a numerical optimization known as Covariance Intersection can be applied to remove unknown correlation between state and measurement. This technique is often used to convert a centralized system into a decentralized system. As such, whether or not a cooperative localization system is centralized becomes more a concern of implementation than of filter design. Mu et al. [71] and Soatti et al. [72] show examples of how one can go about creating equivalent versions of cooperative estimation systems using either centralized or decentralized techniques.

The decentralized approach allows each vehicle to dynamically add and remove cooperative relationships as participant vehicles gain and lose the ability to communicate directly with one another. Loss of communication is typically a result of limits of the range of vehicle-to-vehicle (V2V) technology such as dedicated short range communications (DSRC). When participating vehicles dynamically add and remove cooperate relationships with one another, this is referred to as a Vehicular Ad-hoc Network (VANET), and is generally offered as a practical way to implement cooperative localization using direct V2V instead of a centralized communication structure such as cell modems. With a VANET, any one participant may only occasionally be cooperating with another vehicle, and several disparate networks may exist simultaneously. Research into VANETs is abundant and the topic is well-covered, but it is of only tangential relevance to the work of this thesis. For further reading, see [73, 74]

5.3 Existing Approaches

A thorough survey of sensing techniques used in cooperative localization is presented by de Ponte Müller in [75]. This section will briefly discuss a few individual techniques which are relevant.

Soatti et al. [72] introduce a new class of cooperative approaches termed to be "implicitly" cooperative, in that no direct vehicle-to-vehicle (V2V) measurement (such as relative ranging) is used. Instead, all participant vehicles are assumed to observe a set of common features using perception sensors (such as cameras or RADAR). It is by having each vehicle observe the same feature that relative information is inferred, but never directly produced. The implicit technique does not necessarily require a priori mapping of observable features, either, as the formulation accounts for feature location uncertainty by adding it to the estimation state. As such, simultaneous mapping is possible, though it is not treated as a traditional SLAM algorithm. They present the implicit technique in both centralized and decentralized forms.

Wang et al. [76], closely couple GPS, IMU and Ultra-Wide Band (UWB) radio ranging measurements in a Kalman filter with classical gain scheduling. They communicate ego positions and ranges to common infrastructure points using DSRC radios. UWB ranging and Doppler measurements between each participant vehicle and an infrastructural UWB are treated as pseudolite measurements, so that integration into a more traditional closely coupled GPS/INS is relatively seamless. It is worth noting that they do not use the UWB radio's ability to perform inter-vehicle ranging (demonstrated by [77]), which represents a missed opportunity for increased accuracy. Furthermore, no attention is given to the double counting problem.

Perhaps the most relevant work is that of Rohani et al. [78–81], which focuses on using GNSS positioning signals to perform cooperative map matching. The work begins in [78], by assuming the existence of a ranging sensor and using scalar distance as an input to a novel probabilistic framework which produces an independent position estimate. The position

is then used as a pseudomeasurement to a standard Kalman filter. This serves only as a starting point, as it is not particularly relevant to the problem at hand (since it requires a ranging sensor), and rarely achieves sub-meter accuracies.

Then, in [79], they begin using differential GPS instead of a separate ranging sensor. However, the differential step is performed only to get pseudorange corrections which can then be sent to other vehicles to aid in standard single-vehicle positioning. No carrier-phase information is considered, which limits the accuracy considerably, but it still forms the basis of a cooperative system. For multiple copies of pseudoranges from multiple vehicles, they use simple least squares to produce the value of the pseudorange that they use.

In [80,81] they use this relative positioning in conjunction with map aiding. This starts by assuming all vehicles are on the roadway and treat the entire roadway (not individual lanes) as constraints. They then use relative positioning from the same pseudorange-only differential GPS to "translate" one vehicle's road constraints to other participating vehicles, so that for a group of vehicles which all observe the same satellites, they all use the entire group's map constraints. Their approach to applying map constraints is similar to MACIN's, in that they use a particle filter for position and eliminate any particle violating the constraints. MACIN, on the other hand, offers 3D estimates of both velocity and orientation as well as position.

Like Rohani, Mahmoud et al. [74] use a GNSS-only approach operating at the pseudorange level, but cooperative information is used to fill in gaps in satellite information due to occlusion, rather than constrain the solution space. Once non-line-of-sight (NLOS) satellites are detected ("hindered SVs" in their terminology), the measured pseudorange is discarded and an artifical pseudorange is generated using satellite geometry and pseudorange measurements from another vehicle. This method is highly sensitive to geometries of participant vehicles relative to occluding objects, and the accuracy degrades as distance increases.

Alam et al. [82], use a unique approach tailored for vehicles in opposing traffic lanes to share information over a short time window as they drive past one another. They rely on

the doubling effect of opposing velocity vectors to create a more pronounced Doppler shift in GNSS signals when differenced, noting that typically ground vehicles have dynamics so slow that the Doppler shift is negligible. They model the behavior in the time domain as an impulse response, thus underscoring the fleeting nature of the cooperative relationship in this arrangement.

Obst et al. [83, 84] model vehicle networks as multi-node graphs in which only adjacent vehicles share information. They estimate position using a particle filter, but use wheel speed and steering sensors to propagate with a piecewise constant yaw rate and velocity model. For measurement update and cooperative information sharing, they utilize relative position vectors from differential GPS between vehicles and present simulated results.

5.4 Timing

Cooperative approaches must by definition rely upon data that is transmitted over some communication network. This will unavoidably introduce a time delay. For a given vehicle, on-board sensor measurements should generally arrive within a few milliseconds after the true time. If the time delay is well-known, this is not a problem. In the worst case, the posterior estimate for time t_k may simply be updated at time t_{k+1}, after it has arrived, but before the propagation step. For high-rate IMUs, a GPS measurement can simply be "snapped" to the closest IMU epoch if the dynamics are sufficiently sedate. A $\geq$500 Hz IMU in mass-market passenger vehicles generally matches these conditions, but at low speeds IMU frequencies as low as 100Hz may suffice. However, cooperative navigation dicates that information from off-board sensors and filters can be fused as well, which may be delayed by up to several seconds due to transmission overhead.

This is referred to in the literature as an out of sequence measurement (OOSM) problem. Dealing with the OOSM problem is out of scope for this thesis, and researching the techniques to deal with it is left to the reader, with recommended starting points being [59, 85–87].

This thesis relies on a simple post-processing method to remove time delays and synchronize measurements, which is given later in Section 6.5.

5.5 MACIN Approach

MACIN uses direct vehicle-to-vehicle 3-dimensional Cartesian relative position vectors (RPVs) for cooperation. This has a distinct advantage over indirect methods which require shared observation of environmental objects, since the two vehicles can cooperatively localize in a vacuum (i.e., with no supporting infrastructure or features). Cooperation does not even require map constraints. For instance, this type of cooperation continues even while traversing intersections, when map constraints are disabled. Each RPV is produced using GPS-only dynamic base real time kinematic (DRTK) estimation from [88]. Observing all 3 positional axes has a clear advantage over 2 dimensional sensors such as RADAR (and many LiDARs) which only produce a range and azimuth measurement. Producing estimates in the exact same Cartesian coordinate system (ECEF) as the position estimation frame of the INS allows direct measurement integration, versus something like high-resolution LiDAR, which produces estimates in a spherical coordinate system that is fixed to the ego vehicle body frame, and whose covariance must be transformed.

DRTK is different from RTK in that the first receiver is not statically affixed to a base station position. However, both RTK and DRTK rely on the assumption that gross errors, such as those arising from ionospheric and tropospheric disturbances, are shared among GPS receivers within a few kilometers of one another. Differencing measurements from two receivers removes these common-mode errors, resulting in high-precision relative positioning and standard precision absolute positioning. De Ponte Muller [89] shows how differencing the code-based pseudoranges produces superior results to differencing final position solutions from the same two receivers. RTK and DRTK are based on a similar principle, but go a step further and difference the carrier wave measurements, allowing even greater relative accuracy.

In order to obtain an absolute solution, RTK assumes that one of the receivers is affixed to a base station whose position is extremely certain, with the obvious drawback being the required infrastructure. This thesis uses RTK only for truth measurements, as requiring physical infrastructure makes a solution significantly less scalable. While DRTK use only contributes relative accuracy, it indirectly improves global accuracy when combined with map constraints, as shown empirically by the results in Appendix A. This approach to cooperation can be used anywhere across the globe where two or more vehicles are within a few kilometers of one another. The obvious drawback is that DRTK is limited by line of sight to satellite vehicles, whereas perception sensors are impervious to urban canyons or other degraded sky conditions.

In order to reduce complexity in development of the core approach, a centralized cooperative strategy is used in this thesis, so the results presented are for a single federated RBPF. Once a centralized filter is shown to work, the same approach may be easily extended in later work to a decentralized strategy using the covariance intersection method [70].

Since GPS receivers output measurements aligned to GPS time epochs, synchronizing between the vehicles is very easy and all vehicles' GNSS measurements in general (both SPS position and velocity as well as the DRTK RPVs) already align with one another to within nanosecond level precision [90]. It is possible for the time epoch for all vehicles' individual GNSS measurements to align with one another in time as well, so they can all be used in theF same epoch. For a system with N_j participating vehicles, the full measurement vector can be written as:

$$\mathbf{y} = \left[\begin{array}{c} \mathbf{y}^{GNSS} \\ \hline \mathbf{y}^{DRTK} \end{array}\right] \tag{5.4}$$

$$\mathbf{y}^{GNSS} = \left[\left(\mathbf{y}_1^{GNSS}\right)^T, \ \cdots \ , \left(\mathbf{y}_{N_j}^{GNSS}\right)^T\right]^T \tag{5.5}$$

$$\mathbf{y}^{DRTK} = \left[\left(\mathbf{y}_{1\to 2}^{DRTK}\right)^T, \ \cdots \ , \left(\mathbf{y}_{(N_j-1)\to N_j}^{DRTK}\right)^T\right]^T \tag{5.6}$$

The DRTK measurement subvector $\mathbf{y}^{DRTK}$ will have $N_j(N_j-1)/2$ separate RPVs. The top portion of the full measurement vector, $\mathbf{y}^{GNSS}$, contains measurements typically used for loosely coupled filters: position and velocity of the antenna with respect to the ECEF frame, and expressed in the ECEF frame. For vehicle j, the measurement vector is equivalent to Eq (3.67) and is notated with the vehicle index as:

$$\mathbf{y}_j^{GNSS} = \begin{bmatrix} \mathbf{r}_{A/E,j}^E \\ \mathbf{v}_{A/E,j}^E \end{bmatrix} + \epsilon_j^{GNSS} \tag{5.7}$$

These GNSS measurements are simply obtained directly from the receiver's internal navigation processor along with the covariance of each vector, which is used to construct the measurement covariance matrix:

$$R_j^{GNSS} = E\left[(\epsilon_j^{GNSS})(\epsilon_j^{GNSS})^T\right] \simeq \begin{bmatrix} R_{r,j}^{GNSS} & 0 \\ 0 & R_{v,j}^{GNSS} \end{bmatrix} \tag{5.8}$$

The second portion of the measurement vector contains RPVs for all possible vehicle pairs within the cooperative network, as computed using the DRTK algorithm:

$$\mathbf{y}_{1\to 2}^{DRTK} = \mathbf{r}_{A/E,2}^E - \mathbf{r}_{A/E,1}^E \tag{5.9}$$

While it is possible to derive an analytical method for computing the RPV covariance matrix, this thesis found that the values were always very small (on the order of 100 μ m - 1 mm), and resulted in filter overconfidence. Additionally, DRTK is known to have relative position errors on the centimeter level. As such, the DRTK measurement covariance matrix is set using constant tuning values which are defined in a local navigation frame, with a standard deviation in the earth-tangent directions set to 2cm, and standard deviation is set to 4cm in

the vertical direction:

$$R_j^{DRTK} = C_{NED \rightarrow ECEF} \begin{bmatrix} \sigma_{north}^2 & 0 & 0 \\ 0 & \sigma_{east}^2 & 0 \\ 0 & 0 & \sigma_{down}^2 \end{bmatrix} C_{NED \rightarrow ECEF}^T \tag{5.10}$$

5.6 Summary

This chapter has discussed the field of cooperative navigation and how MACIN leverages cooperative techniques to improve position accuracy. Basic architectural approaches for estimator design were discussed in Section 5.2, as well as considerations thereof that are germaine to this thesis. Section 5.3 gave the reader an overview of the current state of the art within the field of cooperative navigation, and Section 5.5 discussed achieving cooperation using DRTK relative positioning.

This concludes the discussion of background information to support the design decisions that this thesis made. Chapter 3 discussed using an RBPF as an improvement upon the EKF, Chapter 4 discussed the use of map information, and this chapter has discussed the cooperative aspect. The next chapter will discuss how to unify these technologies into a cohesive estimation framework and present experimental results that prove its efficacy.

Chapter 6

MACIN Implementation & Results

The previous chapters have focused on individual pieces of the MACIN algorithm and the reasoning behind them. This chapter gives all remaining information necessary for the readers to implement MACIN for themselves and replicate findings. Section 6.1 discusses setting the remaining tuning parameters that have not already been mentioned. Section 6.2 gives algorithmic pseudocode to aid in building necessary software Sections 6.3 through 6.5 cover data collection, post processing procedures, and map representations used for experimental validation. This chapter concludes with Section 6.6, which presents experimental findings.

In order to establish a benchmark for comparison, the most common and simple approach to GNSS/INS navigation was formulated and implemented as described in Chapter 3 This was a loosely coupled EKF with no cooperative or map-based aiding, as formalized in Section 3.5.2. Use of the baseline filter allows isolating the benefits that MACIN provides.

6.1 Noise Terms

Selection and tuning of noise terms requires special attention, so this section lays out key information for readers to understand in that regard. For the baseline filter, process noise is unchanged from Equation (3.55). Since all the independent baseline filters are assembled into a single augmented filter, this means the augmented process noise covariance matrix is created by assembling all consituent process noise covariance matrices into a single matrix

along the diagonal:

$$Q = \begin{bmatrix} Q_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & Q_2 & \mathbf{0} & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & Q_{N_j} \end{bmatrix} \tag{6.1}$$

Here, each submatrix Q_j for vehicle j is unchanged from Equation (3.56).

For MACIN, however, partitioning position as a particle state requires determining a new, separate process noise η_n which creates a spreading effect on the particles. The most obvious choice is to model position process noise as zero mean Gaussian and use the velocity submatrix from the Kalman estimate covariance, as shown below.

$$\dot{\mathbf{x}}_n = \dot{\mathbf{r}}^E_{B/E} = \mathbf{v}^E_{B/E} + \eta_n \tag{6.2}$$

$$\eta_n \sim \mathcal{N}(0, Q_n) \tag{6.3}$$

$$Q_n = E\left[\eta_n \eta_n^T\right] \simeq P_l \mid_v \tag{6.4}$$

For MACIN's linear partition, the process noise covariance matrix is the same as for the baseline filter, $Q^l = Q$.

It is important to note that the noise formulation used for MACIN is suboptimal in five ways, all regarding covariance (off-diagonal) terms that have been set to zero:

- The true covariance terms in R^{GNSS} between position and velocity are not zero. This is a general shortcoming of the loosely coupled formulation, which can be addressed with tighter integration.
- The true value of R_j^{DRTK} is not a diagonal matrix.
- The covariance terms in R^{DRTK} between RPV pairs that share a common vehicle is non-zero. For vehicle 1, 2, and 3, this would mean $E\left[(\epsilon^{DRTK}_{1\to2})(\epsilon^{DRTK}_{2\to3})\right] \neq 0$. This thesis only tests 2 vehicles, so this suboptimality never manifests itself.

- The true values of the covariance terms relating y^{GNSS} and y^{DRTK} are non-zero, since measurements are computed use the same set of satellite observations, meaning that the measurements are correlated.

The covariance between the particle and Kalman partition process noises Q^{ln} is set to zero, just as in Ryan's work [12]. However, due to the choice of Q_n in Equation (6.4), more work is needed to verify that this is optimal, and the existing body of literature is silent on the matter of selecting and/or deriving Q^{ln}. To compensate for over-confidence, though, the particle process noise matrix may be scaled as a tuning parameter.

6.2 MACIN Algorithm

This section presents the algorithm in detail in the form of pseudocode so that the reader may be fully equipped to implement MACIN on their own. The high-level MACIN algorithm (Alg. 3) can be broken down into four steps:

1. Time update, or "propagation" (Alg. 4)
2. Particle update using GNSS measurements (Alg. 5)
3. Map constraint update (Alg. 6)
4. Kalman update (Alg. 7)

Of these, only Step 3 is different from the traditional RBPF procedure. Note that the map constraint update and resampling may be applied at every propagation epoch, regardless of the presence of a GNSS measurement. This feature curtails the position drift inherent in open-loop integration of IMU data.

6.3 Test Scenarios

Map-aiding is most beneficial in situations where the route network is highly circuitous and complex, causing the particle cloud to be reduced more significantly and quickly. As

Algorithm 3 MACIN Top-Level Algorithm

Initialize:
  $x_n, x_l \leftarrow$ `starting state`
  $P_n, P_l \leftarrow$ `starting covariance`
  $Q_l \leftarrow \sigma_{g,\nu}, \sigma_{a,\nu}, \sigma_{g,u}, \sigma_{a,u}$ ▷ Set Kalman process noise
  $Q_{ln} \leftarrow 0$
  $N_t \leftarrow$ `# of time epochs`
  $N_X \leftarrow$ `# of particles`
  $w^i \leftarrow 0, i = 1, ..., N_X$
for $k \in [2, ..., N_t]$ **do** ▷ Iterate over IMU samples
  TIME UPDATE ▷ Alg. 4
  if GNSS measurement available at t_k **then**
    Run DRTK on available SVs
    $y^{GNSS} \leftarrow$ Receiver position/velocity solution
    $y^{DRTK} \leftarrow$ RPV pairs from DRTK
    PARTICLE UPDATE ▷ Alg. 5
  MAP UPDATE ▷ Alg. 6
  Normalize weights w
  Resample ▷ Alg. 1
  Normalize weights w
  KALMAN UPDATE ▷ Alg. 7

such, interstate driving is a scenario in which map-aiding is typically the least useful. GNSS coverage generally already provides excellent quality with low occlusion rates (meaning that map input is of minimal benefit), and uncertainty along the length of the lane cannot be curtailed by geometric diversity in the routing network. This makes interstate-like conditions ideal to find the true merit of MACIN. Data collection was performed at the National Center for Asphalt Technology (NCAT) test track, which is a 1.7 mile oval with two lanes built to U.S. standards for interstate highways. It is pictured in satellite imagery in Figure 6.1, and the lane network representation of the NCAT test track is depicted in Figure 6.2. For all scenarios, two vehicles were used with identical equipment. While introducing additional vehicles to the cooperative swarm should yield increased performance, testing that hypothesis was outside the scope of this thesis.

At the NCAT location, six scenarios were enacted, which are given in Table A.1 and described here. Scenarios A, B, and C involve a large curve, whereas Scenarios D, E, and

Algorithm 4 MACIN IMU Time Update

procedure TIME UPDATE
  $\Delta t \leftarrow t_k - t_{k-1}$
  Calculate $f_n(x_n)$
  Calculate $f_l(x_n, x_l, \tilde{\omega}_{k-1}, \tilde{a}_{k-1}, \Delta t)$
  Calculate $A_n(\Delta t)$
  Calculate $A_l(x_l, \tilde{\omega}_{k-1}, \tilde{a}_{k-1}, \tau_g \tau_a, \Delta t)$
  Calculate $G_n(\Delta t)$
  Calculate $G_l(x_l, \Delta t)$
  $Q_n \leftarrow$ `velocity submatrices of` P_l
  $\eta \leftarrow \mathcal{N}(\mathbf{0}, Q_n)$ ▷ Draw process noise set
  for $i \in [1, ..., N_X]$ **do**
    $x_n^{(i)} += \int_{t_{k-1}}^{t_k} \dot{x}_n dt$ ▷ Propagate each particle
  $x_n \leftarrow mean(x_n^{(1,...,N_X)})$
  $P_n \leftarrow cov(x_n^{(1,...,N_X)})$
  $\bar{A}^l \leftarrow A^l$
  $\bar{Q}^l \leftarrow Q^l$
  Calculate N using Equation (3.43)
  Calculate L using Equation (3.44)
  $z \leftarrow x_n - f_n$ ▷ Use propagated particle state for innovation
  Propagate x^l
  for $j \in [1, ..., N_v]$ **do** ▷ iterate over vehicles
    $q_j \leftarrow q_j \ / \ ||q_j||$ ▷ Normalize attitude quaternion
  Normalize attitude quaternions q_1 and q_2
  Propagate P^l

F are straight the entire time. In Scenario A, both vehicles begin traveling in straight lines parallel to one another, then enter opposite sides of a curve and pass each other in the middle before traveling in straight parallel lines again. In Scenario B, the vehicles travel the same course (straight, curve, straight), but travel side-by-side and adjacent lanes. Scenario C is a slight modification of Scenario B, with the cars traveling single file in the same lane. Scenario D is similar to A, in that both cars begin and end on opposite side of the track and travel in opposing lanes, but instead of entering a curve they maintain a straight path the entire time and pass each other in the middle. In Scenario E, they travel straight, side-by-side, in adjacent lanes. In Scenario F, they travel straight, single file, in the same lane.

Algorithm 5 MACIN Particle Measurement Update

procedure PARTICLE UPDATE
  Calculate $C(x_l, \tilde{\omega}_k, r^B_{A/B})$
  Calculate M using Equation (3.48)
  Calculate K using Equation (3.49)
  for $i \in [1, ..., N_X]$ **do**
    Calculate $h^{(i)}(x_n^{(i)}, x_l, \tilde{\omega}_k, r^B_{B/A})$
    Calculate $y^{(i)}$ using Equation (3.38) ▷ Particle measurement prediction
    $w^{(i)} \leftarrow p\left(y^{(i)} \mid y, M\right)$ ▷ Evaluate particle on measurement PDF

Algorithm 6 MACIN Map Update

procedure MAP UPDATE
  for $i \in [1, ..., N_X]$ **do** ▷ iterate over particles
    for $j \in [1, ..., N_v]$ **do** ▷ iterate over vehicles
      Pick lane that minimizes d_{proj} for $r_j^{(i)}$ ▷ Closest lane to vehicle j
      $d_{ONN} \leftarrow d_{lane} - d_{veh,j}$ ▷ See Figure 4.3
      if $d_{proj} > d_{ONN}$ **then**
        $w^{(i)} \leftarrow 0$

Each of the NCAT scenarios is enumerated in Table A.1. Drive results for NCAT are given later in Tables 6.2 and 6.3, and drives in which each vehicle was piloted at the very edge of the lane boundary are denoted in the "Near Bound" column. Figures A.1 through A.6 in Appendix A give graphical illustrations of vehicle motion for all NCAT drives.

A second environment in which map aiding becomes less useful is the center of intersections. Intersections have no defined lanes or visible lane boundaries, and nearly the entire area is a plausible driving surface, so little can be inferred in the way of positional constraints.

Figure 6.1: Satellite image of the NCAT test track

Algorithm 7 MACIN Kalman Measurement Update

procedure KALMAN UPDATE
  $\Gamma \leftarrow$most likely lane ▷ Section 4.5.2
  $I \leftarrow [\,]$ ▷ List of in-lane particles
  for $i \in [1, ..., N_X]$ **do**
    if $l_i = \Gamma$ **then** ▷ Particle is in-lane
      $I \leftarrow i$ ▷ Add particle to list
  $x_n \leftarrow mean(x^{(i)} \text{ for } i \in I)$ ▷ Particle state mean
  $P_n \leftarrow cov(x^{(i)} \text{ for } i \in I)$ ▷ Particle state covariance
  Calculate $h(x_n, x_l, \tilde{\omega}, r^B_{A/B})$
  $P_l \mathrel{-}= KMK^T$ ▷ Linear covariance update
  $x_l \mathrel{+}= K\,(y - h - Cx_l)$ ▷ Linear state update

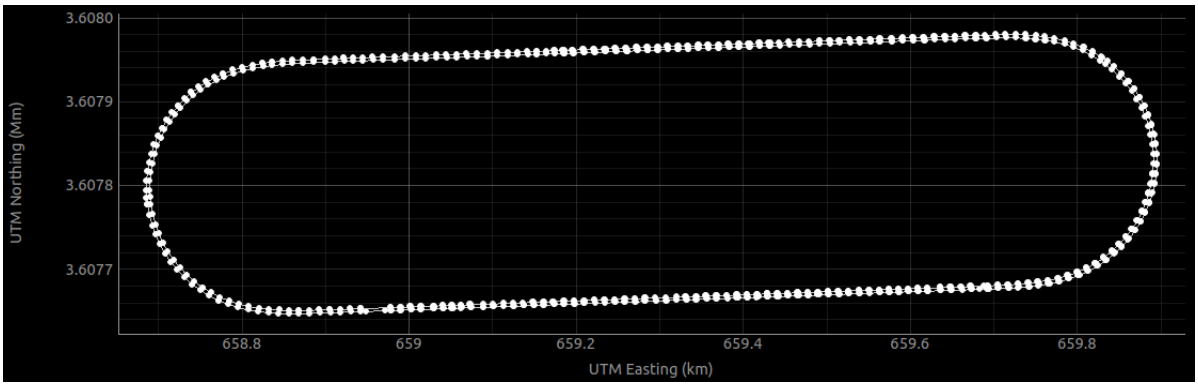

Figure 6.2: NCAT test track lane network representation

This makes intersections ideal for stress-testing the MACIN algorithm. The intersection used for testing is pictured in satellite imagery in Figure 6.3, and its representation as a lane network graph is depicted in Figure 6.4. Two vehicles with identical equipment were used in the intersection scenarios as well. Drive results for the intersection are given later in Tables 6.4 and 6.5.

For the intersection location, three scenarios were enacted, which are given in Table A.2 and described here. In Runs 1 and 2, vehicle 1 travels west to east, going straight through the intersection without stopping. Vehicle 2 travels south to north, stopping before traveling straight through the intersection. In Run 3, vehicle 1 travels west to south, turning right at the intersection. Vehicle 2 travels south to west, turning left at the intersection. In Run 4, vehicle 1 travels west to north, turning left at the intersection. Vehicle 2 travels south to

Figure 6.3: Intersection used for data collection

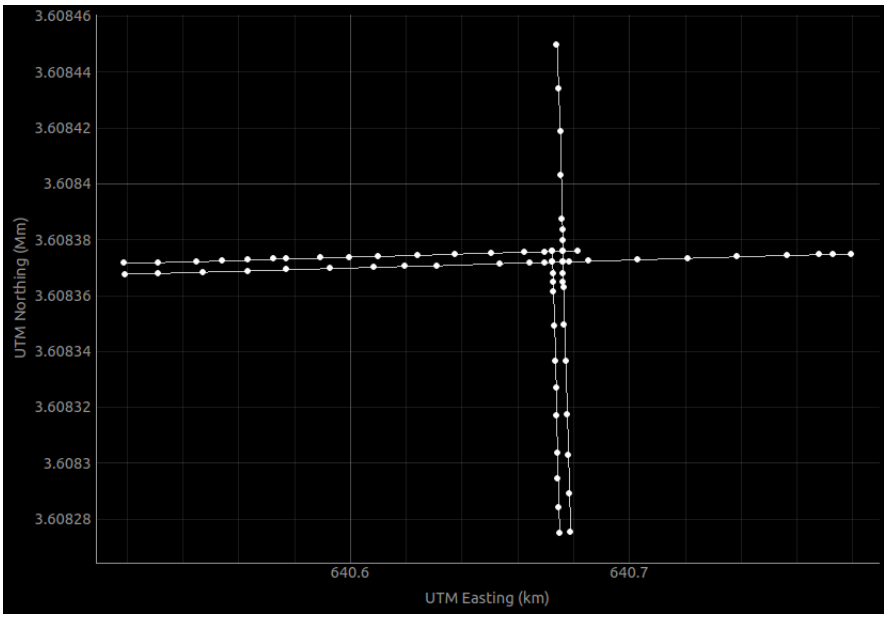

Figure 6.4: Intersection lane network representation

east, turning right at the intersection. Each of the intersection scenarios is enumerated in Table A.2, and Figures A.7 through A.9 in Appendix A give graphical illustrations of vehicle motion for all NCAT drives.

The map representations for both cases are stored using the OpenStreetMap format [91], which is an extension of the XML markup language. For an example of representing lane networks in the OSM XML format, see Appendix B. Truth data for absolute position was obtained using real-time kinematic (RTK) GPS corrections from a fixed base station. RTK provides positioning with errors at the centimeter level [92], which is an order of magnitude more accurate than MACIN, which is providing accuracy on the order of decimeters. The primary metric for MACIN's performance will be root-mean-square error (RMSE) of the position solution in the 2-dimensional earth-tangent plane. The RMSE value for an arbitrary quantity is calculated by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{estimate} - x_{true})^2} \tag{6.5}$$

This is appropriate given the primary goal of improving the lane-level position estimates to sub-meter level.

6.4 Experimental Hardware

A single set of sensor hardware is used here to isolate the effects of MACIN on navigation performance relative to a baseline EKF. Each vehicle is outfitted with a NovAtel OEMV GPS receiver (Figure 6.5), and a Crossbow IMU440CA200 inertial measurement unit (Figure 6.6). The fixed RTK base station that was used as a source of position truth also employed the same NovAtel OEMV receiver. Removing differences in sensing equipment between the two vehicles makes comparing the performance between them more meaningful. All sensor measurements were collected over a universal serial bus (USB) connection to a Linux laptop, and recorded using the Mission Oriented Operating Suite (MOOS) [93] as a middleware.

Figure 6.5: NovAtel OEM-V Propak V3 GNSS receiver and antenna used for data collection

Figure 6.6: Crossbow 440-CA200 IMU used for data collection

6.5 Time Synchronization

All middleware solutions have inherent jitter, that is, time-varying lags in message delivery and time stamping which are difficult to predict and correct between successive epochs. The effect that jitter has on PVA estimation is largely dependent on the system dynamics relative to the magnitude of the timing errors. For instance, in a system using position updates, position accuracy becomes more sensitive to timing errors as velocity increases. Accordingly, in a system using orientation updates, orientation accuracy becomes more sensitive to timing errors as angular velocity increases.

The middleware and sensor interfaces used for data collection in this thesis were subject to considerable timing errors, particularly in the first minutes of data collection. However, GPS data is aligned to GPS time epochs to an accuracy of less than 100 ns for SPS [90], and the GPS time is packaged into the measurement. For the purposes of this thesis, it is assumed that GPS time stamps contain negligible errors. Each GPS measurement also provides a middleware time stamp, expressed as seconds removed from the Unix time epoch

and subject to timing error. Timing errors for the GPS receiver and IMU are modeled as:

$$\tilde{t}_g^{MW} = t_g^{GPS} + \delta t + c_g + \eta_g \tag{6.6}$$

$$\tilde{t}_i^{MW} = t_i^{GPS} + \delta t + c_i + \eta_i \tag{6.7}$$

where $\tilde{t}_g^{MW}$ and $\tilde{t}_i^{MW}$ are the middleware time stamps of the GNSS and IMU measurements, respectively. t_g^{GPS} is the GNSS measurement's true GPS time stamp (which is known), and t_i^{GPS} is the IMU measurement's true GPS time stamp (which is not known). δt is the true offset between GPS time and Unix time, and is known since it can computed analytically. c_g and c_i are constant offsets that exist primarily due to repeatable computation time for processing each measurement. η_g and η_i are unknown and unmodeled additional errors.

In this thesis it is assumed that both the GNSS receiver and the IMU experience the same constant time offset, and that the additive noise on each is zero mean with the same standard deviation, as stated in Equations (6.8) below.

$$c_g \approx c_i \tag{6.8}$$

$$\eta_g \approx \eta_i \sim \mathcal{N}(\mathbf{0}, \sigma_t) \tag{6.9}$$

$$\tilde{t}_g^{MW} = t_g^{GPS} + \Delta t + \eta_g \tag{6.10}$$

$$\tilde{t}_i^{MW} = t_i^{GPS} + \Delta t + \eta_i \tag{6.11}$$

$$\Delta t \approx mean(\tilde{\mathbf{t}}_g^{MW} - \mathbf{t}_g^{GPS}) \tag{6.12}$$

It is also assumed that the sampling period of the IMU remains constant. GNSS and IMU measurements can be matched by estimating an constant offset from the GNSS measurements alone.

For further work on network time delay analysis and compensation, see [94]. Since this thesis deals with post-processing only, time stamping inaccuracies are the sole concern and delivery delays are irrelevant. However, on an online system the PVA estimator is expected

Table 6.1: RMS value of position error for one run with and without time synchronization.

Position RMSE	3D	2D
Without time synchronization	3.328 m	1.262 m
With time synchronization	1.607 m	1.073 m

to provide estimates for states at epochs that are very near to the current time. Under such requirements, receiving measurements that correspond to estimation epochs far in the past or which are out of sequence is problematic, and a new class of filters and methods exists solely for dealing with this. Recommended reading includes [59, 85–87, 95–100].

In order to examine the effect of this time synchronization scheme, results before and after its application must be compared. This is done below using a representative 5 minute data set in benign GPS conditions, traveling 10-15 m/s. Standard single-vehicle loosely coupled EKF estimates of position are compared to fixed-base RTK measurements, and the results are tabulated in Table 6.1. In this case, "2D" indicates the RMS value for position error magnitude in north and east earth-tangent directions, but not elevation.

6.6 MACIN Experimental Results

This section will show results from experimental data collected according to the procedures described in the previous sections. Section 6.6.1 presents all of the results in aggregate, with a high-level discussion. Section 6.6.2 will show best case behavior for MACIN, whereas Section 6.6.3 will show worst case behavior. Section 6.6.4 looks at lane selection accuracies, and finally, Section 6.6.5 will discuss MACIN's behavior when aiding sources are removed.

6.6.1 Aggregated Results

All results for MACIN and the baseline are all consolidated into a single plot in Figure 6.7. For this figure, each point represents a separate drive for a single vehicle, with the x-axis value being the earth tangent position RMSE of the baseline filter, and the y-axis value being the earth tangent position RMSE of MACIN. A "parity" line with a slope of 1.0

indicates possible locations where the baseline filter and MACIN had equivalent performance. Points above the line indicate test runs where the baseline had lower errors, whereas points below the line represent drives where MACIN performed better than the baseline. One can see that MACIN always performed better than the baseline, as expected. In all but two outlying scenarios, the earth tangent position RMSE is below 1.0m, meaning that MACIN yields nominally sub-meter accurate positioning for the scenarios that were tested.

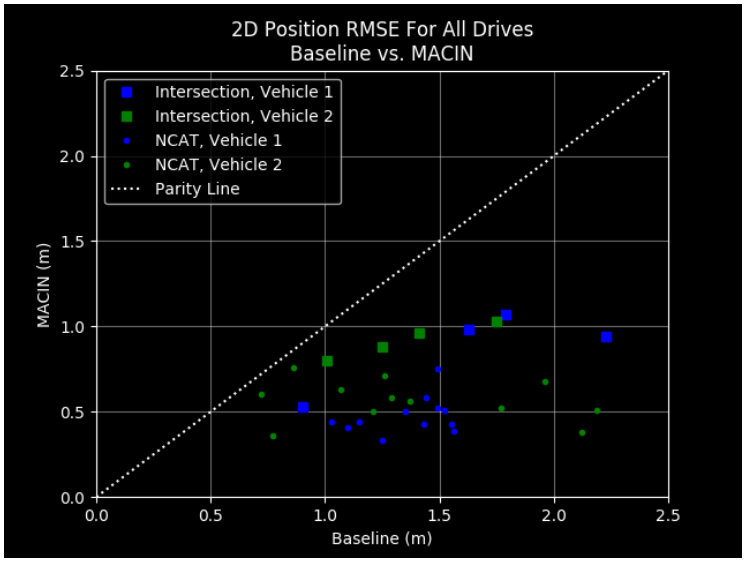

Figure 6.7: Scatter plot of results for all data collection drives. The further a point is below the white line, the better MACIN performed relative to the baseline.

Results for each individual drive at the NCAT test track are tabulated in Tables 6.2 and 6.3 for vehicles 1 and 2, respectively. For a description of each test run collected at NCAT, see Section 6.3 and Table A.1. Results for each individual drive at the intersection location are tabulated in Tables 6.4 and 6.5 for vehicles 1 and 2, respectively. For a description of each data set collected at the intersection location, see Section 6.3 and Table A.2. Each of the numberical results tables contains both the baseline and MACIN 2D position RMSE values for each test run, along with the error reduction MACIN delivered.

Looking a bit further, Vehicle 2 has much less consistent performance compared to Vehicle 1. Vehicle 1 had its IMU permanently mounted, whereas Vehicle 2 utilized a temporary IMU mounting solution due to hardware availability constraints. This mounting resulted in

Table 6.2: 2D Position RMSE for data collected at NCAT, Baseline vs MACIN, vehicle 1

Run #	Run Type	Near Bound	Baseline	MACIN	Improvement
2	E	Y	1.55 m	0.43 m	72.30 %
3	F	N	1.49 m	0.52 m	65.00 %
4	D	N	1.15 m	0.44 m	61.55 %
5	D	Y	1.43 m	0.43 m	69.97 %
6	D	N	1.10 m	0.41 m	63.30 %
7	A	N	1.35 m	0.50 m	63.23 %
8	D	Y	1.03 m	0.44 m	56.91 %
9	B	N	1.44 m	0.58 m	60.05 %
10	C	N	1.25 m	0.33 m	73.99 %
11	C	Y	1.52 m	0.51 m	66.33 %
12	C	N	1.49 m	0.75 m	50.11 %
13	A	N	1.56 m	0.39 m	75.15 %

Table 6.3: 2D Position RMSE for data collected at NCAT, Baseline vs MACIN, vehicle 2

Run #	Run Type	Near Bound	Baseline	MACIN	Improvement
2	E	Y	1.96 m	0.68 m	65.44 %
3	F	Y	1.29 m	0.58 m	55.36 %
4	D	Y	0.86 m	0.76 m	11.29 %
5	D	Y	0.77 m	0.36 m	53.05 %
6	D	N	1.07 m	0.63 m	41.11 %
7	A	Y	1.37 m	0.56 m	59.47 %
8	D	Y	0.72 m	0.60 m	17.75 %
9	B	N	1.21 m	0.50 m	58.74 %
10	C	N	1.77 m	0.52 m	70.51 %
11	C	N	2.19 m	0.51 m	76.84 %
12	C	Y	1.26 m	0.71 m	44.00 %
13	A	Y	2.12 m	0.38 m	82.11 %

Table 6.4: 2D Position RMSE for intersection data, Baseline vs MACIN, for vehicle 1

Run #	Baseline	MACIN	Improvement
1	1.63 m	0.98 m	39.88 %
2	2.23 m	0.94 m	57.89 %
3	0.90 m	0.53 m	40.78 %
4	1.79 m	1.07 m	40.33 %

Table 6.5: 2D Position RMSE for intersection data, Baseline vs MACIN, for vehicle 2

Run #	Baseline	MACIN	Improvement
1	1.25 m	0.88 m	29.62 %
2	1.75 m	1.03 m	40.96 %
3	1.01 m	0.80 m	20.57 %
4	1.41 m	0.96 m	31.64 %

IMU vibration and variation in the lever arm $\mathbf{r}^B_{A/B}$ between the IMU frame and the GNSS antenna. Since $\mathbf{r}^B_{A/B}$ was assumed to be rigid, this motion led to decreased performance and the process noise terms were inflated to accommodate the unmodeled dynamics. Furthermore, the internal positioning engine in the GNSS receiver on vehicle 2 reported consistently higher position uncertainties, as shown in Figure 6.8. Even with this handicap, MACIN always outperformed the baseline. In fact, while these issues cause the baseline to see its worst performances on vehicle 2 in runs 2, 3, 11, and 13, those very same test runs saw no discernable change in MACIN's nominal behavior on the same vehicle. This is an indication that use of the techniques presented in this thesis allow a vehicle with degraded sensing to maintain accurate navigation by relying on information from cooperating vehicles.

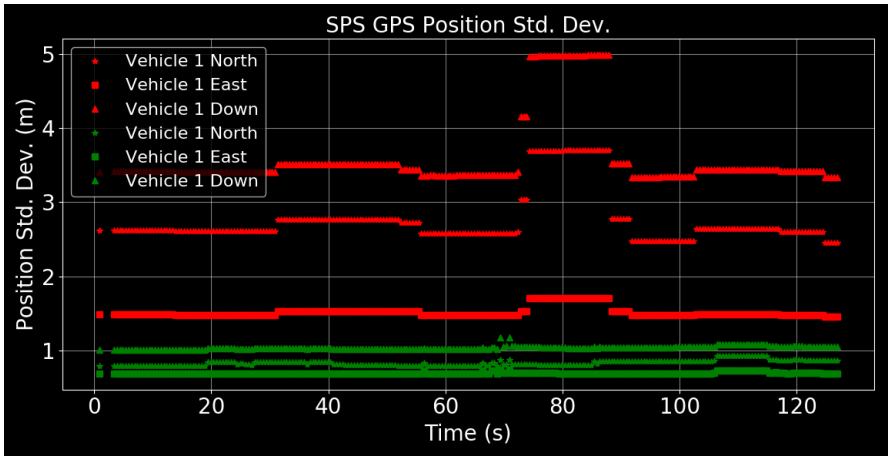

Figure 6.8: Position standard deviation reported by the GPS receiver internal positioning engine for NCAT run 13, vehicle 1 versus vehicle 2.

One can also notice that in the intersection drives, MACIN saw a slight reduction in performance. This is because those drives include non-negligible amounts of time during which both vehicles were traveling without map constraints due to being within the bounds

of an intersection. Since MACIN cannot make any reasonable probabilistic inference about the vehicle's position using the map when the vehicle is within an intersection, it briefly shuts off map constraints as discussed in Chapter 4 For this reason MACIN performs closer to the baseline during test runs which included an intersection. However, relative updates are still turned on during these brief periods, and the vehicles still spend a majority of time on regular roadway, so there is still a significant improvement.

6.6.2 MACIN Best Case Results

One instance of ideal behavior occurs during run number 13 of the NCAT data sets is shown below in Figure 6.9. The ground truth (RTK) paths traversed by both test vehicles during approximately the same time period are shown side by side, as well as the position solutions for both baseline filter and MACIN. One can see that vehicle 1 was travelling near the center of the right lane, and MACIN correctly placed the vehicle in that position. The baseline filter estimate is impinging upon the adjacent lane or straddling the lane divider the whole time. This is not unexpected, since the hard lane constraints would be expected to bias results toward the center.

Vehicle 2 was driven close to the lane divider in order to stress test this hypothesis. One can see that its true position was very near the adjacent lane, and yet MACIN defied expectations and accurately captured this. The baseline filter, on the other hand, was biased toward the lane center. This case can be considered an ideal outcome.

6.6.3 MACIN Worst Case Results

Run number 4 of the intersection data set contains a very challenging situation for any navigation filter. Ground truth paths for both vehicles are shown side by side below for this run in Figure 6.10, along with the paths estimated by the baseline and MACIN. The speeds stay under 12 m/s throughout the drive since it occurs in a parking lot, and in conjunction with the short duration, this prevents the vehicle dynamics from becoming

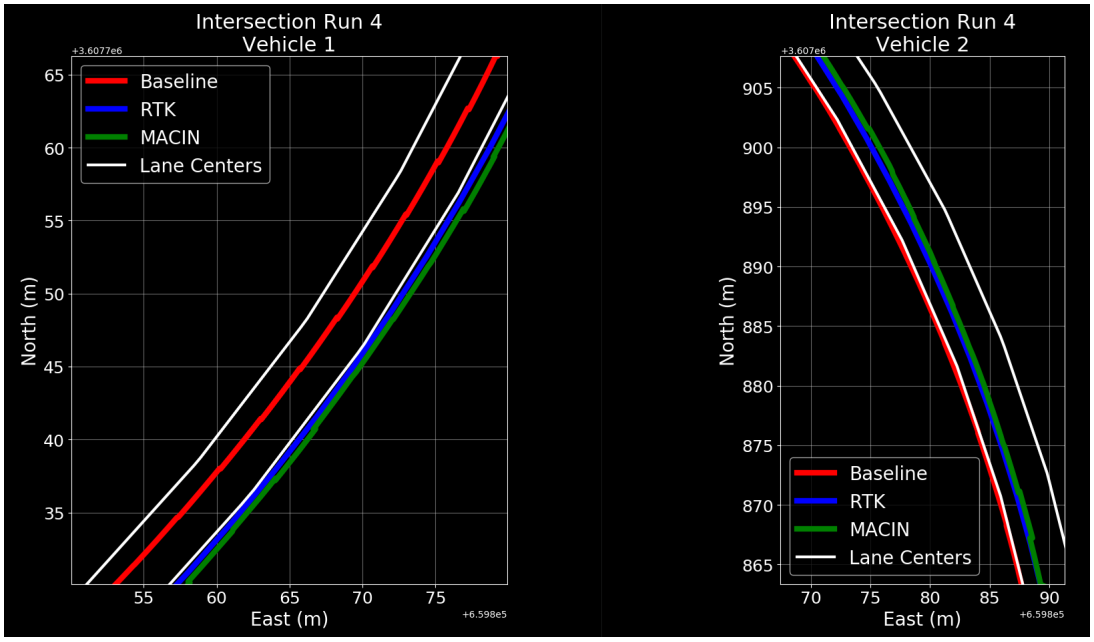

Figure 6.9: Concurrent paths estimated by MACIN, the baseline, and RTK truth for both vehicles during NCAT run 13.

excited enough for the IMU error terms to sufficiently converge to the proper values. When error terms have not converged to the proper value, it causes the position to rapidly drift away between Kalman updates, and then experience large changes when updates do occur. This "sawtooth" behavior is clearly visible in the path of the baseline filter. MACIN also suffers for the same reason, with a clear difference: large jumps in position occur within the confines of the correct lane. This shows that even in poor conditions with incorrect IMU error estimates, MACIN is able to bound the errors such that the position estimate is always lane-level accurate.

6.6.4 Lane Selection Accuracy

All test drives stayed in one lane for the duration of each test. This allows the correct lane to be determined in postprocessing at every single point in each drive, in order to evaluating lane selection accuracy. In order to collect data with lane changes, some system for annotating the exact time a test vehicle crossed over a lane boundary would be required, and no such system was available for this thesis.

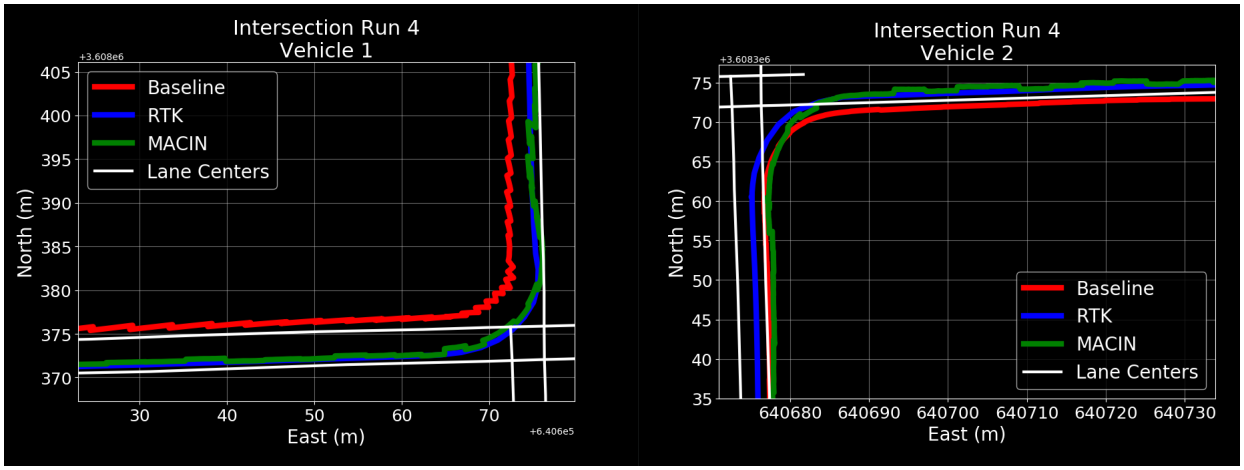

Figure 6.10: Concurrent paths estimated by MACIN, the baseline, and RTK truth for both vehicles during intersection run 4.

Every data set began with the test vehicles stopped, and this played an important role in lane selection. Experimental results for lane selection accuracy were very poor during this static period before the vehicles began moving. MACIN output an incorrect lane choice at least once during initialization for 20 of the 32 instances (2 vehicles and 16 test drives). Figure 6.11 below shows an example of this occuring for vehicle 2 during test drive number 3 at NCAT. The northernmost lane was incorrectly selected briefly prior to the moment the vehicle first moved.

However, in all test drives, *there was not a single instance of MACIN making an incorrect lane selection after the initial static period ended.* This means that MACIN was 100% accurate in selecting the correct lane while the vehicles were in motion for the data collected in this thesis. So not only are the overall RMS values of position error under the threshold for lane-level accuracy, but the actual lane assignment is accurate after the vehicle begins moving.

6.6.5 Performance with Aiding Partially Disabled

An area that requires further research is MACIN's performance when its aiding sources are removed. This thesis investigates the behavior in 2 cases: when either map constraints

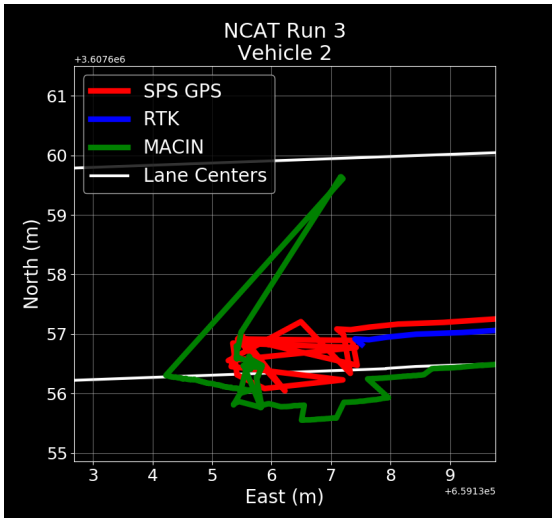

Figure 6.11: Example of MACIN selecting the wrong lane during initialization, prior to the vehicle moving.

or DRTK relative positioning are individually disabled for the duration of the test drive. In the case of no map aiding, performance degraded on a per-vehicle basis and in many cases the filter failed due to sample impoverishment (all particle weights went to zero), whereas with DRTK disabled MACIN was observed to experience sample impoverishment for all data sets.

Map Aiding Disabled

Results for the NCAT drives in which MACIN did not experience sample impoverishment with map aiding disabled and DRTK enabled are shown in Tables 6.6 and 6.7 for vehicles 1 and 2, respectively. None of the intersection data sets were successful under these conditions.

From this information, it is immediately apparent that vehicle 1 experienced degraded performance, but still outperformed the baseline. On the other hand, the positioning for vehicle 2 is even worse than the baseline. Two key differences between the two test vehicle data sets contributed to this. First, the value of the linear partition process noise η from Equation 3.55 was inflated for vehicle 2. As mentioned earlier, this was done in order to account for modeling errors that arose from the hardware setup allowing the IMU to vibrate

Table 6.6: 2D Position RMSE for data collected a NCAT, comparison of MACIN performance with map constraints disabled, for vehicle 1.

Run #	**Run Type**	**Baseline**	**MACIN No Map**	**MACIN**
2	E	1.55	1.44	0.43
3	F	1.49	1.03	0.52
6	D	1.10	1.27	0.41
7	A	1.35	1.14	0.50
8	D	1.03	1.06	0.44
9	B	1.44	0.98	0.58
12	C	1.49	1.15	0.75
13	A	1.56	0.79	0.39

Table 6.7: 2D Position RMSE for data collected a NCAT, comparison of MACIN performance with map constraints disabled, for vehicle 2.

Run #	**Run Type**	**Baseline**	**MACIN No Map**	**MACIN**
2	E	1.96	2.19	0.68
3	F	1.29	1.87	0.58
6	D	1.07	1.27	0.63
7	A	1.37	1.64	0.56
8	D	0.72	1.81	0.60
9	B	1.21	1.72	0.50
12	C	1.26	2.20	0.71
13	A	2.12	1.58	0.38

and the GNSS antenna lever arm to fluctuate. As a result of this increased linear partition process noise, the linear partition covariance P_l was larger for vehicle 2 relative to vehicle 1. Since $P_l \mid_v$ was selected for the particle process noise in Equation 6.4, the position for vehicle 2 spread at a much higher rate than that of vehicle 1. This can be seen in the standard deviation computation over the position particles, shown in Figure 6.12. Without a map constraint update every propagation epoch, though, the particles were only resampled at the GNSS receiver measurement rate of 2 Hz, leaving a much longer period of time for the particles to propagate open-loop, and thus having a much higher likelihood of diverging. The second difference is in the performance of the vehicles' respective GNSS receivers, as noted earlier (see Figure 6.8). However, it is worth noting that despite the elevated GPS position

uncertainty in vehicle 2, the presence of DRTK allowed vehicle 2 to converge to the same level of certainty as vehicle 1 at each update.

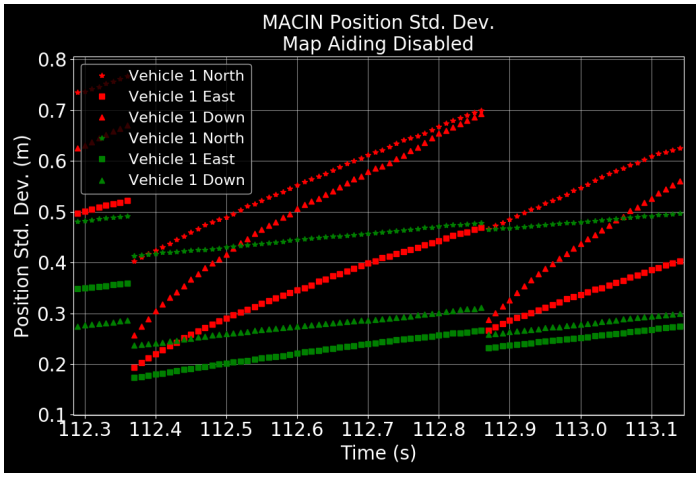

Figure 6.12: Standard deviation of MACIN position particles with map aiding completely disabled for NCAT run 13.

The failure mechanism, then, is that particles become too spread out for any single particle to evaluate to a non-zero weight, and the filter has no valid samples. The number of effective particles N_{eff} from Equation 3.30 is a direct measure of this sample impoverishment. It is computed after the weight update, and before the resampling step. Even in one of the best-performing cases with no map updates, NCAT run 13, there was significant sample impoverishment. Figure 6.13 shows that N_{eff} was at most 4% of the total number of particles for this run. For reference, in [66] the authors indicate that N_{eff} should not be allowed to go lower than 67% of the total number particles before resampling. Some authors, however, do allow this to drop as low as 10% [101]. Further research is required to adequately characterize and address the problems that arise when map aiding is completely disabled.

DRTK Disabled

This phenomenon of sample impoverishment occurred in every data set when map aiding was enabled, but DRTK was disabled. Each particle in MACIN is a concatenation of the position samples for all participating vehicles. As such, each particle's weight is a score

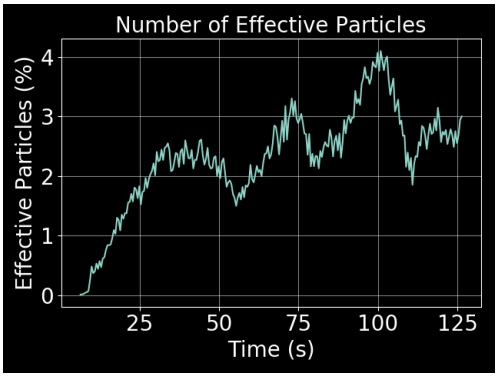

Figure 6.13: Number of effective particles during NCAT run 13 with map aiding disabled.

of the combined likelihood of both vehicle position samples. However, when cooperative aiding from a relative position vector is withheld, the two vehicle's respective state variable partitions evolve independently of one another. The only correlation between them is that they use similar sets of satellites and the same map, but these correlations were neglected in this thesis, so there is nothing to explicitly link them any longer. However, the weight update of the particle partition still evaluates them as a single state, as if they were correlated.

As an example, consider a single particle that contains a high likelihood position sample for vehicle 1, and a low likelihood position sample for vehicle 2. The process noise distribution that was sampled in order to generate the two positions is effectively two separate distributions, and yet the two particles will be evaluated as if they are linked. This means that the high likelihood sample of the position PDF for vehicle 1 will recieve the same low weight as the low likelihood sample of vehicle 2's position PDF.

So without DRTK, the particles contain uncorrelated variables whose weights are evaluated as if they were correlated. This is a clear issue, particularly when one vehicle experiences degraded measurement quality, as was the case in the experimental data collection for this thesis. Further research is required to adequately characterize and address the problems that arise when DRTK is completely disabled.

Chapter 7

Summary & Conclusions

The preceding chapters have comprehensively laid out a novel map aided cooperative inertial navigation system which produces sub-meter performance. Chapter 2 explained basics of IMUs, and the error model used in this thesis. Chapter 3 gave a basic background on estimation as it relates to inertial navigation, and gave formulations for the baseline EKF as well as the Rao-Blackwellized particle filter MACIN is built on. Chapter 4 discussed the current state of map usage in navigation applications and detailed the map constraint strategy used herein. Chapter 5 gave an overview of the burgeoning field of cooperative navigation and the relative positioning technique that MACIN employs. Chapter 6 gave the detailed MACIN algorithm for readers to replicate on their own, and discussed real world data collection along with the results MACIN produced. Following are some key areas for potential expansion of this work in the future:

- Implement MACIN for online operation in a real-time embedded computing system. Timing concerns arise from the fact that GNSS data are typically experience delays on the order of several milliseconds, which can be detrimental in high-dynamic driving. Setting up a pulse-per-second (PPS) connection or some other system to ascertain the true time to which a GNSS measurement corresponds is critical to real-time functionality. In addition to that, the algorithmic concerns with processing out of sequence measurements must be considered (as discussed in Chapter 5).

- Implement MACIN in a decentralized architecture. This is theoretically simple, using the covariance intersection method employed by [70]. The core difficulty in getting a decentralized cooperative navigation filter is in the implementation (mostly software framework) and timing (as discussed above).

- Add close- or tight-coupling between GNSS and IMU measurements. This will improve satellite tracking performance and allow more fine-grained fault detection and exclusion.

- Closer map coupling: MACIN currently uses a binary pseudomeasureent of whether each vehicle is inside or outside of lane boundaries. This information is assumed to come from a camera- or LiDAR-based vision system. Two key improvements can be made:

  - Use detected lane lines as relative position measurements (similar to [102]), either as projected onto the closest roadway, or using shape matching.

  - Use visual landmarks from a semantic map as range measurements.

As was mentiond in Section 4.5.1, applying vertical constrains to match the physical limitation that all ground vehicles must travel along the manifold of the roadway would allow extremely tight restrictions. This would exceed the lower limit on accuracy in the earth-tangent directions that arises from lane width minimums. In order to do this, accurate computation of the IMU position relative to the road surface is required, and this must come from a dynamic roll and pitch model that takes suspension dynamics into account. If one were to apply vertical constraints in this manner, not only would positional accuracy be greatly increased, but the attitude would likely be significantly more accurate as well. This would of course necessitate an attitude truthing system to verify.

This thesis has presented a novel navigation filter for use in ground vehicles that require lane-level positioning. MACIN's approach of combining Rao-Blackwellized particle filtering, map constraints, and DRTK relative positioning may be implemented by the reader with the algorithms presented in this thesis. On current computing hardware, it is capable of running in real time at 100 Hz, allowing use in applications where low latency is important. Experimental data showed highly accurate lane selection capability and consistent submeter positioning performance. This drastic reduction in errors is possible using only sensors and

information that are currently available on commercial passenger ground vehicles. As such, the objectives outlined in the introduction to this thesis have been accomplished.

Bibliography

[1] "'Phantom Auto' will tour city". *The Milwaukee Sentinel,* ***Dec. 8 1926***.

[2] OpenStreetMap Contributors. OpenStreetMap. `https://www.openstreetmap.org/copyright`. Accessed: 2019-10-01.

[3] Society of Automotive Engineers (SAE) V2X Core Technical Committee, 2016. *Dedicated Short Range Communications (DSRC) Message Set Dictionary.* J2735.

[4] Federal Highway Administration, July 2007. *Mitigation Strategies for Design Exceptions.* Table 3: Ranges for Lane Width.

[5] Jafari, M., and Roshanian, J., 2013. "Inertial Navigtion Accuracy Increasing Using Redundant Sensors". *Journal of Science and Engineering,* ***1***(1), pp. 55–66.

[6] Han, S., and Wang, J., 2011. "Quantization and Colored Noises Error Modeling for Inertial Sensors for GPS/INS Integration". *IEEE Sensors Journal,* ***11***(6), jun, pp. 1493–1503.

[7] Wall, J. H., 2007. "A Study of the Effects of Stochastic Inertial Sensor Errors in Dead-Reckoning Navigation". PhD thesis, Auburn University.

[8] Groves, P. D., 2013. *Principles of GNSS, inertial, and multisensor integrated navigation systems.* Artech house.

[9] Crassidis, J., 2005. "Sigma-Point Kalman Filtering for Integrated GPS and Inertial Navigation". In AIAA Guidance, Navigation, and Control Conference and Exhibit, American Institute of Aeronautics and Astronautics, pp. 1–24.

[10] Flenniken, W. S., 2005. "Modeling Inertial Measurement Units and Anlyzing the Effect of Their Errors in Navigation Applications". PhD thesis, Auburn University.

[11] Flenniken IV, W. S., Wall, J. H., and Bevly, D. M., 2005. "Characterization of Various IMU Error Sources and the Effect on Navigation Performance". *Ion Gnss 2005*, pp. 967–978.

[12] Ryan, J., 2016. "Classification of Ego Platform Motion for Platform Independent Plug and Play Navigation". Doctoral dissertation, Auburn University.

[13] Engelberg, S., 2006. *Random Signals and Noise.* CRC Press.

[14] Wang, B., Ren, Q., Deng, Z., and Fu, M., 2015. "A Self-Calibration Method for Nonorthogonal Angles Between Gimbals of Rotational Inertial Navigation System". *IEEE Transactions on Industrial Electronics,* ***62***(4), apr, pp. 2353–2362.

[15] Shin, E.-H., and El-Sheimy, N., 2002. "A new calibration method for strapdown inertial navigation systems.". *Z. Vermess,* ***127***(1), pp. 1–10.

[16] Wall, J. H., 2007. "A Study of the Effects of Stochastic Inertial Sensor Errors in Dead-Reckoning Navigation". PhD thesis, Auburn University.

[17] USGS Earthquake Map. `https://earthquake.usgs.gov/earthquakes/map/`. Accessed: 2019-12-28.

[18] Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T., 2002. "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking". *IEEE Transactions on Signal Processing,* ***50***(2), feb, pp. 174–188.

[19] Welch, G., and Bishop, G., 2006. An Introduction to the Kalman Filter. Tech. rep., University of North Carolina, Chapel Hill, Chapel Hill, North Carolina, jul.

[20] Dieci, L., and Eirola, T., 1994. "Positive definiteness in the numerical solution of Riccati differential equations". *Numerische Mathematik,* ***67***(3), apr, pp. 303–313.

[21] Mazzoni, T., 2007. "Computational Aspects of Continuous-Discrete Extended Kalman -Filtering". pp. 1–15.

[22] Wan, E. A., and van der Merve, R., 2000. "The Unscented Kalman Filter for Nonlinear Estimation". pp. 153–158.

[23] Mandel, J., 2009. "A Brief Tutorial on the Ensemble Kalman Filter". *SciencesNew York,* ***2007***(242), jan, p. 7.

[24] PAPADAKIS, N., MÉMIN, E., CUZOL, A., and GENGEMBRE, N., 2010. "Data assimilation with the weighted ensemble Kalman filter". *Tellus A,* ***62***(5), apr, pp. no–no.

[25] Liu, J., Chen, H.-z., Cai, B.-g., and Wang, J., 2015. "State estimation of connected vehicles using a nonlinear ensemble filter". *Journal of Central South University,* ***22***(6), jun, pp. 2406–2415.

[26] Hanlon, P., and Maybeck, P., 2000. "Multiple-model adaptive estimation using a residual correlation Kalman filter bank". *IEEE Transactions on Aerospace and Electronic Systems,* ***36***(2), apr, pp. 393–406.

[27] Hendeby, G., Karlsson, R., and Gustafsson, F., 2010. "The Rao-Blackwellized Particle Filter: A Filter Bank Implementation". *EURASIP Journal on Advances in Signal Processing,* ***2010***(1), dec, p. 724087.

[28] Douc, R., Cappé, O., and Moulines, E., 2005. "Comparison of Resampling Schemes for Particle Filtering". *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, jul, pp. 64–69.

[29] Ray, T. N., 2019. "Pedestrian Navigation using Particle Filtering and a priori Building Maps". PhD thesis, Auburn University.

[30] Elvira, V., Miguez, J., and Djurie, P. M., 2017. "Adapting the Number of Particles in Sequential Monte Carlo Methods Through an Online Scheme for Convergence Assessment". *IEEE Transactions on Signal Processing,* ***65***(7), pp. 1781–1794.

[31] Kotecha, J., and Djuric, P., 2003. "Gaussian particle filtering". *IEEE Transactions on Signal Processing,* ***51***(10), oct, pp. 2592–2601.

[32] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J., 2002. "Particle filters for positioning, navigation, and tracking". *IEEE Transactions on Signal Processing,* ***50***(2), pp. 425–437.

[33] Schon, T., Gustafsson, F., and Nordlund, P.-J., 2005. "Marginalized particle filters for mixed linear/nonlinear state-space models". *IEEE Transactions on Signal Processing,* ***53***(7), jul, pp. 2279–2289.

[34] Schmidt, G. T., and Phillips, R. E., 2011. "INS / GPS Integration Architectures". *NATO Lecture Series,* ***116***(2011), pp. 1–18.

[35] Vernaza, P., and Lee, D., 2006. "Rao-Blackwellized particle filtering for 6-DOF estimation of attitude and position via GPS and inertial sensors". In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., Vol. 2006, IEEE, pp. 1571–1578.

[36] Hall, P., 2001. "A Bayesian Approach to Map-Aided Vehicle Positioning". PhD thesis, Linkoping University, Sweden.

[37] Fouque, C., Bonnifait, P., and Betaille, D., 2008. "Enhancement of global vehicle localization using navigable road maps and dead-reckoning". *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 1286–1291.

[38] Quddus, M., Ochieng, W., Zhao, L., and Noland, R., 2003. "A general map matching algorithm for transport telematics applications". *GPS Solutions,* ***7***(3), pp. 157–167.

[39] Ochieng, W. Y., and Quddus, M. A., 2003. "Map-Matching in Complex Urban Road Networks". *Brazilian J. Catrography,* ***2***(55), dec, pp. 1–14.

[40] Quddus, M. A., Ochieng, W. Y., and Noland, R. B., 2006. "Integrity of map-matching algorithms". *Transportation Research Part C: Emerging Technologies,* ***14***(4), pp. 283–302.

[41] Quddus, M. A., Ochieng, W. Y., and Noland, R. B., 2007. "Current map-matching algorithms for transport applications: State-of-the art and future research directions". *Transp. Res. Part C Emerg. Technol.,* ***15***(5), pp. 312–328.

[42] Velaga, N. R., Quddus, M. a., and Bristow, A. L., 2009. "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems". *Transportation Research Part C: Emerging Technologies,* ***17***(6), pp. 672–683.

[43] Quddus, M., and Washington, S., 2015. "Shortest path and vehicle trajectory aided map-matching for low frequency GPS data". *Transportation Research Part C: Emerging Technologies,* ***55***, pp. 328–339.

[44] White, C. E., Bernstein, D., and Kornhauser, A. L., 2000. "Some map matching algorithms for personal navigation assistants". *Transp. Res. Part C Emerg. Technol.,* ***8***(1â6), pp. 91–108.

[45] Alt, H., Efrat, A., Rote, G., and Wenk, C., 2003. "Matching planar maps". *Journal of Algorithms,* ***49***(2), nov, pp. 262–283.

[46] Davidson, P., Collin, J., Raquet, J., and Takala, J., 2010. "Application of Particle Filters for Vehicle Positioning Using Road Maps". In Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation, no. April 2016, pp. 1653–1661.

[47] Davidson, P., Collin, J., and Takala, J., 2011. "Application of particle filters to a map-matching algorithm". *Gyroscopy and Navigation,* ***2***(4), oct, pp. 285–292.

[48] Chu, H.-J., Tsai, G.-J., Chiang, K.-W., and Duong, T.-T., 2013. "GPS/MEMS INS Data Fusion and Map Matching in Urban Areas". *Sensors,* ***13***(9), aug, pp. 11280–11288.

[49] Toledo-Moreo, R., Betaille, D., Peyret, F., and Laneurit, J., 2009. "Fusing GNSS, Dead-Reckoning, and Enhanced Maps for Road Vehicle Lane-Level Navigation". *IEEE Journal of Selected Topics in Signal Processing,* ***3***(5), oct, pp. 798–809.

[50] Toledo-Moreo, R., Betaille, D., and Peyret, F., 2010. "Lane-Level Integrity Provision for Navigation and Map Matching With GNSS, Dead Reckoning, and Enhanced Maps". *IEEE Transactions on Intelligent Transportation Systems,* ***11***(1), mar, pp. 100–112.

[51] Pink, O., and Hummel, B., 2008. "A statistical approach to map matching using road network geometry, topology and vehicular motion constraints". In Intell. Transp. Syst. 2008. ITSC 2008. 11th Int. IEEE Conf., pp. 862–867.

[52] Jo, K., Chu, K., and Sunwoo, M., 2013. "GPS-bias correction for precise localization of autonomous vehicles". *IEEE Intelligent Vehicles Symposium, Proceedings*(Iv), pp. 636–641.

[53] Schindler, A., Maier, G., and Pangerl, S., 2011. "Exploiting arc splines for digital maps". In 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 1–6.

[54] Schindler, A., 2013. "Vehicle self-localization with high-precision digital maps". In 2013 IEEE Intelligent Vehicles Symposium (IV), no. Iv, IEEE, pp. 141–146.

[55] Payne, O., and Marrs, A., 2004. "An unscented particle filter for GMTI tracking". In Aerosp. Conf. 2004. Proceedings. 2004 IEEE, Vol. 3, pp. —-1875 Vol.3.

[56] Arulampalam, M. S., Gordon, N., Orton, M., and Ristic, B., 2002. "A variable structure multiple model particle filter for GMTI tracking". In Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997), Vol. 2, Int. Soc. Inf. Fusion, pp. 927–934.

[57] Koch, W., Koller, J., and Ulmke, M., 2006. "Ground target tracking and road map extraction". *ISPRS Journal of Photogrammetry and Remote Sensing,* ***61***(34), pp. 197–208.

[58] Ulmke, M., and Koch, W., 2006. "Road-map assisted ground moving target tracking". *IEEE Transactions on Aerospace and Electronic Systems,* ***42***(4), oct, pp. 1264–1274.

[59] Mallick, M., Kirubarajan, T., and Arulampalam, S., 2002. "Out-of-sequence measurement processing for tracking ground target using particle filters". In Proceedings, IEEE Aerospace Conference, Vol. 4, IEEE, pp. 4–1809–4–1818.

[60] Pannetier, B., Benameur, K., Nimier, V., and Rombaut, M., 2005. "VS-IMM using road map information for a ground target tracking". In 2005 7th International Conference on Information Fusion, FUSION, Vol. 1, pp. 24–31.

[61] Kirubarajan, T., Bar-Shalom, Y., Pattipati, K., and Kadar, I., 2000. "Ground target tracking with variable structure IMM estimator". *IEEE Transactions on Aerospace and Electronic Systems,* ***36***(1), jan, pp. 26–46.

[62] Blackman, S. S., 2004. "Multiple Hypothesis Tracking For Multiple Target Tracking". *IEEE Aerospace and Electronic Systems Magazine,* ***19***(January), p. 14.

[63] Doucet, A., Gordon, N. J. J., Kroshnamurthy, V., and Krishnamurthy, V., 2001. "Particle filters for state estimation of jump Markov linear systems". *IEEE Trans. Signal Process.,* ***49***(3), mar, pp. 613–624.

[64] Bar-Shalom, Y., 1990. "Multitarget-multisensor tracking: advanced applications". *Norwood, MA, Artech House, 1990, 391 p.*

[65] Bar-Shalom, Y., and Li, X.-R., 1995. *Multitarget-multisensor tracking: principles and techniques*, Vol. 19. YBs Storrs, CT.

[66] Jabbour, M., Bonnifait, P., and Cherfaoui, V., 2008. "Map-Matching Integrity Using Multihypothesis Road-Tracking". *Journal of Intelligent Transportation Systems,* ***12***(4), oct, pp. 189–201.

[67] Smaili, C., Najjar, M. E. E., Charpillet, F., and Rose, C., 2005. Multi-Sensor Fusion for Mono and Multi-Vehicle Localization using Bayesian Network.

[68] Smaili, C., Najjar, M. E. B. E., and Charpillet, F., 2014. "A Hybrid Bayesian Framework for Map Matching: Formulation Using Switching Kalman Filter". *Journal of Intelligent & Robotic Systems,* ***74***(3-4), jun, pp. 725–743.

[69] Patwari, N., Ash, J. N., Kyperountas, S., H, A. O., Moses, R. L., and Correal, N. S., 2005. "Locating the Nodes: Cooperative localization in wireless sensor networks". *IEEE Signal Processing Magazine*(July), pp. 54–69.

[70] Mokhtarzadeh, H., and Gebre-Egziabher, D., 2014. "Cooperative Inertial Navigation". *Navigation,* ***61***(2), jun, pp. 77–94.

[71] Mu, H., Dai, M., Gao, M., and Pan, X., 2013. "Fully Decentralized Cooperative Localization of a Robot Team: An Efficient and Centralized Equivalent Solution". pp. 520–531.

[72] Soatti, G., Nicoli, M., Garcia, N., Denis, B., Raulefs, R., and Wymeersch, H., 2017. "Implicit Cooperative Positioning in Vehicular Networks". *IEEE Transactions on Intelligent Transportation Systems,* ***19***(12), sep, pp. 3964–3980.

[73] Hoang, G.-M., Denis, B., Harri, J., and Slock, D. T., 2016. "On communication aspects of particle-based cooperative positioning in GPS-aided VANETs". In 2016 IEEE Intelligent Vehicles Symposium (IV), Vol. 2016-Augus, IEEE, pp. 20–25.

[74] Mahmoud, A., Noureldin, A., and Hassanein, H. S., 2015. "VANETs Positioning in Urban Environments: A Novel Cooperative Approach". In 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall), IEEE, pp. 1–7.

[75] de Ponte Müller, F., 2017. "Survey on Ranging Sensors and Cooperative Techniques for Relative Positioning of Vehicles". *Sensors,* ***17***(2), jan, p. 271.

[76] Wang, J., Gao, Y., Li, Z., Meng, X., and Hancock, C., 2016. "A Tightly-Coupled GPS/INS/UWB Cooperative Positioning Sensors System Supported by V2I Communication". *Sensors,* ***16***(7), jun, p. 944.

[77] Broshears, E., Martin, S., and Bevly, D., 2011. "Ultra-wideband Radio Aided Carrier Phase Ambiguity Resolution in Real-Time Kinematic GPS Relative Positioning". pp. 1277–1284.

[78] Rohani, M., Gingras, D., Vigneron, V., and Gruyer, D., 2013. "A new decentralized Bayesian approach for cooperative vehicle localization based on fusion of GPS and inter-vehicle distance measurements". In Connect. Veh. Expo (ICCVE), 2013 Int. Conf., pp. 473–479.

[79] Rohani, M., Gingras, D., and Gruyer, D., 2014. "Dynamic base station DGPS for cooperative vehicle localization". In 2014 International Conference on Connected Vehicles and Expo (ICCVE), IEEE, pp. 781–785.

[80] Rohani, M., Gingras, D., and Gruyer, D., 2014. "Vehicular cooperative map matching". In 2014 International Conference on Connected Vehicles and Expo (ICCVE), IEEE, pp. 799–803.

[81] Rohani, M., Gingras, D., and Gruyer, D., 2016. "A Novel Approach for Improved Vehicular Positioning Using Cooperative Map Matching and Dynamic Base Station DGPS Concept". *IEEE Transactions on Intelligent Transportation Systems,* ***17***(1), jan, pp. 230–239.

[82] Alam, N., Kealy, A., and Dempster, A. G., 2013. "Cooperative Inertial Navigation for GNSS-Challenged Vehicular Environments". *IEEE Transactions on Intelligent Transportation Systems,* ***14***(3), sep, pp. 1370–1379.

[83] Richter, E., Obst, M., and Schubert, R., 2008. "Cooperative Localization Algorithms for Improved Road Navigation". In Royal Institute of Navigation Annual Conference and Exhibition 2008 and 37th ILA Annual Convention and Technical Symposium (NAV 08/ILA 37).

[84] Obst, M., Schubert, R., Mattern, N., and Liberto, C. "Gnss-Based Relative Localization for Urban Transport Applications Within the Covel Project Covel Technical Approach". pp. 1–11.

[85] Larsen, T., Andersen, N., Ravn, O., and Poulsen, N., 1998. "Incorporation of time delayed measurements in a discrete-time Kalman filter". In Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171), Vol. 4, IEEE, pp. 3972–3977.

[86] Mallick, M., and Marrs, A., 2003. "Comparison of the KF and particle filter based out-of-sequence measurement filtering algorithms". In Sixth International Conference of Information Fusion, 2003. Proceedings of the, Vol. 1, IEEE, pp. 422–429.

[87] Orton, M., and Marrs, A., 2005. "Particle filters for tracking with out-of-sequence measurements". *IEEE Transactions on Aerospace and Electronic Systems,* ***41***(2), apr, pp. 693–702.

[88] Martin, S. M., 2011. "Closely Coupled GPS / INS Relative Positioning For Automated Vehicle Convoys". PhD thesis, Auburn University.

[89] De Ponte Müller, F., Diaz, E. M., Kloiber, B., and Strang, T., 2014. "Bayesian cooperative relative vehicle positioning using pseudorange differences". *Record - IEEE PLANS, Position Location and Navigation Symposium*, may, pp. 434–444.

[90] Misra, P., and Enge, P., 2011. *Global Positioning System: Signals, Measurements, and Performance*, revised se ed. Ganga-Jamuna Press.

[91] OpenStreetMap Wiki: OSM XML. `https://wiki.openstreetmap.org/wiki/OSM_XML`. Accessed: 13 January 2017.

[92] Valente, D. S. M., Momin, A., Grift, T., and Hansen, A., 2020. "Accuracy and precision evaluation of two low-cost rtk global navigation satellite systems". *Computers and Electronics in Agriculture,* ***168***, p. 105142.

[93] Newman, P., 2006. "Moos -mission orientated operating suite". *Mass. Inst. Technol. Tech. Rep.,* ***2299***, 01.

[94] Woodall, W. J., 2012. "Low-Bandwidth Three Dimensional Mapping and Latency Reducing ModelPrediction to Improve Teleoperation of Robotic Vehicles". PhD thesis, Auburn University.

[95] Zhang, H., Xie, L., Zhang, D., and Soh, Y. C., 2004. "A reorganized innovation approach to linear estimation". *IEEE Transactions on Automatic Control,* ***49***(10), pp. 1810–1814.

[96] Lu, X., Xie, L., Zhang, H., and Wang, W., 2007. "Robust Kalman Filtering for Discrete-Time Systems With Measurement Delay". *IEEE Transactions on Circuits and Systems II: Express Briefs,* ***54***(6), jun, pp. 522–526.

[97] Gopalakrishnan, A., Kaisare, N. S., and Narasimhan, S., 2011. "Incorporating delayed and infrequent measurements in Extended Kalman Filter based nonlinear state estimation". *Journal of Process Control,* ***21***(1), jan, pp. 119–129.

[98] Challa, S., Evans, R. J., and Wang, X., 2003. "A Bayesian solution and its approximations to out-of-sequence measurement problems". *Information Fusion,* ***4***(3), sep, pp. 185–199.

[99] van der Merwe, R., Wan, E., and Julier, S., 2004. "Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion: Applications to Integrated Navigation". In AIAA Guidance, Navigation, and Control Conference and Exhibit, no. August, American Institute of Aeronautics and Astronautics.

[100] Keshu Zhang, and Li, X., 2002. "Optimal update with out-of-sequence measurements for distributed filtering". In Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997), Vol. 2, Int. Soc. Inf. Fusion, pp. 1519–1526.

[101] Särkkä, S., 2013. *Bayesian Filtering and Smoothing*. Cambridge University Press.

[102] Laneurit, J., Chapuis, R., and Chausse, F., 2005. "Accurate Vehicle Positioning on a Numerical Map". *International Journal of Control, Automation, and Systems,* ***3***(1), pp. 15–31.

Appendices

Appendix A

Supplemental Information for Experimental Data

Table A.1: Description of data collection scenarios at NCAT

Code	**Description**	**Runs Collected**
A	Both vehicles going around curve in opposite direction, pass each other in middle	7, 13
B	Go around curve in the same direction, different lanes, side by side	9
C	Go around curve in same direction, same lane, single file	10, 11, 12
D	Go down straight in opposite directions, different lanes, pass each other in middle	4, 5, 6, 8
E	Go down straight in same direction, different lanes, side by side	2
F	Go down straight in same direction, same lane, single file	3

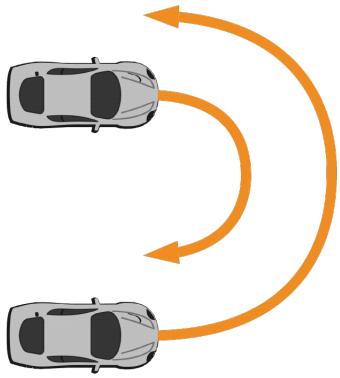

Figure A.1: Diagram of vehicle motion for NCAT run type A.

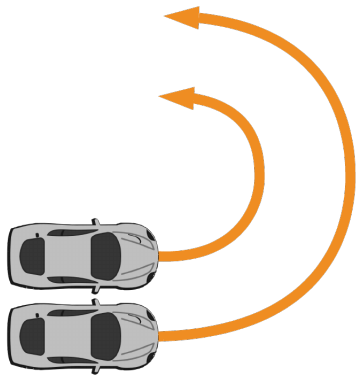

Figure A.2: Diagram of vehicle motion for NCAT run type B.

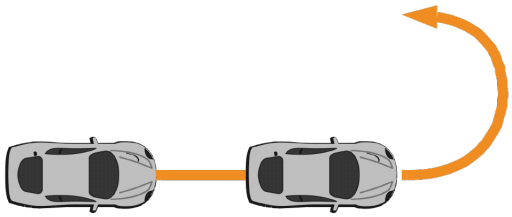

Figure A.3: Diagram of vehicle motion for NCAT run type C.

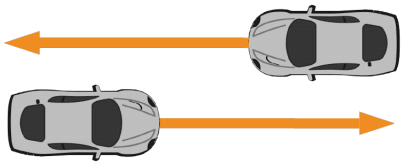

Figure A.4: Diagram of vehicle motion for NCAT run type D.

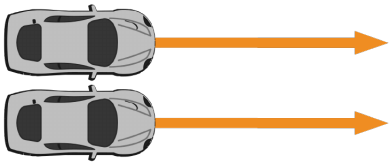

Figure A.5: Diagram of vehicle motion for NCAT run type E.

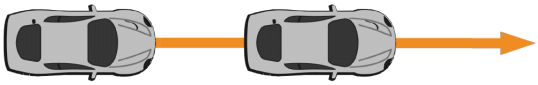

Figure A.6: Diagram of vehicle motion for NCAT run type F.

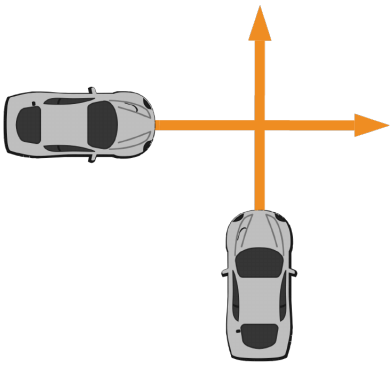

Figure A.7: Diagram of vehicle motion for intersection runs 1 and 2.

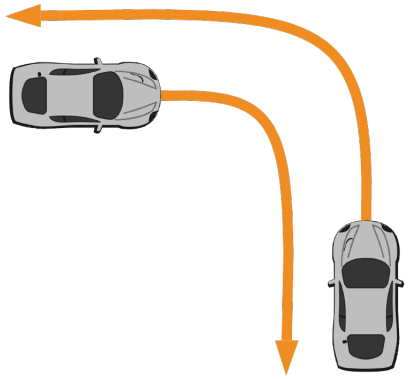

Figure A.8: Diagram of vehicle motion for intersection run 3.

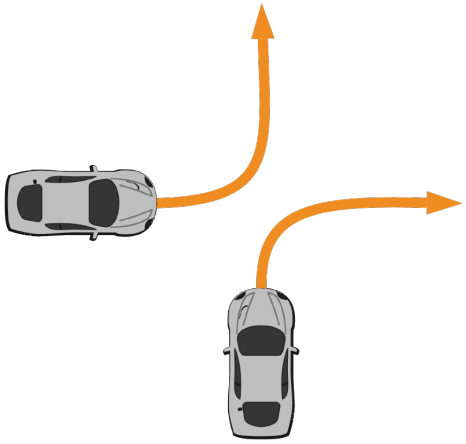

Figure A.9: Diagram of vehicle motion for intersection run 4.

Table A.2: Description of data collection runs in intersection environment

Run #	**Description**
1	Vehicle 1 travels west to east, traveling straight. Vehicle 2 travels south to north, stopping before traveling straight.
2	Vehicle 1 travels west to east, traveling straight. Vehicle 2 travels south to north, stopping before traveling straight.
3	Vehicle 1 travels west to south, turning right. Vehicle 2 travels south to west, turning left.
4	Vehicle 1 travels west to north, turning left. Vehicle 2 travels south to east, turning right.

Appendix B

Example OpenStreetMap Network

Listing B.1: XML source code for an example lane network.

```
<?xml version='1.0' encoding='UTF-8'?>
<osm version='0.6'>
    <node id='1' lat='32.0000000' lon='-85.0000000'>
        <tag k='width' v='3.6' />
    </node>
    <node id='2' lat='32.0000100' lon='-85.0000000'>
        <tag k='width' v='3.6' />
    </node>
    <node id='3' lat='32.0000100' lon='-85.0000100'>
        <tag k='width' v='3.6' />
    </node>
    <node id='4' lat='32.0000200' lon='-85.0000000'>
        <tag k='width' v='3.6' />
    </node>
    <node id='5' lat='32.0000300' lon='-85.0000100'>
        <tag k='width' v='3.6' />
    </node>
    <way id='100'>
        <nd ref='1' />
        <nd ref='2' />
        <nd ref='3' />
    </way>
    <way id='101'>
        <nd ref='2' />
        <nd ref='4' />
        <nd ref='5' />
    </way>
</osm>
```

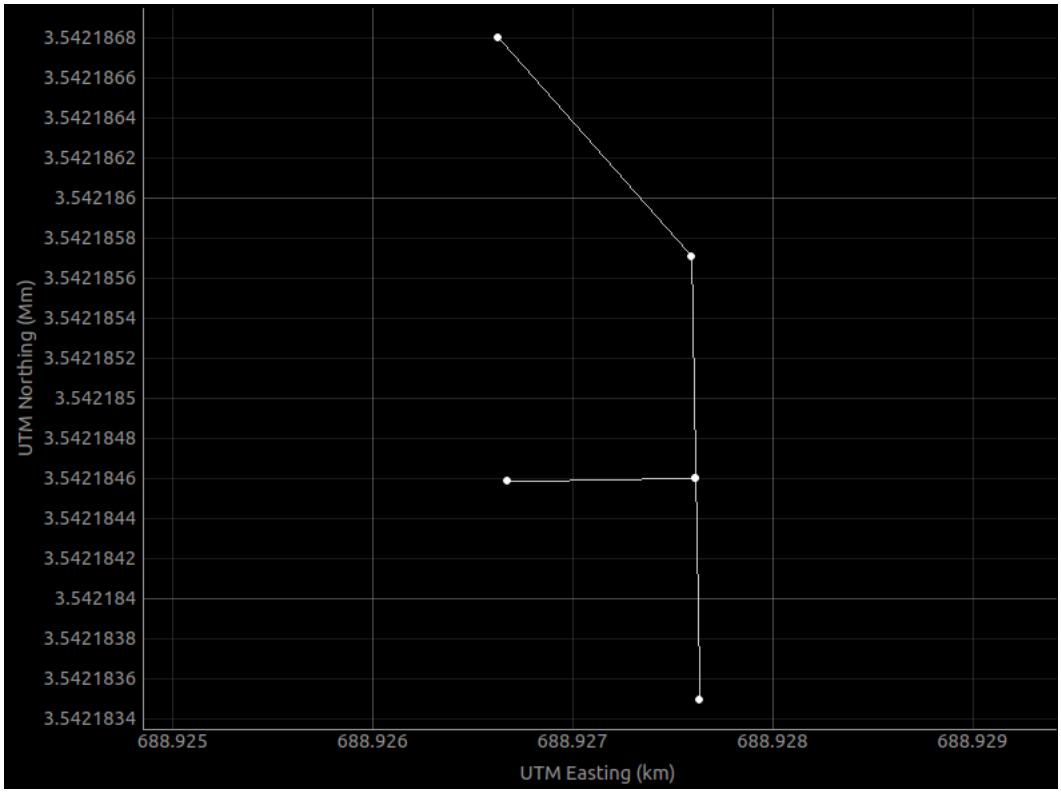

Figure B.1: UTM frame depiction of the example lane network