TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Master's Thesis in Informatics

Vehicle Localization and Tracking for Collision Avoidance System

Behtarin Ferdousi

TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Master's Thesis in Informatics

Vehicle Localization and Tracking for Collision Avoidance System

Fahrzeuglokalisierung und -verfolgung für das Kollisionsvermeidungssystem

Author:	Behtarin Ferdousi
Supervisor:	Prof. Dr.-Ing. Matthias Althoff
Advisor:	Jagat Rath , Ph.D
Submission Date:	01.01.2020

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Ich versichere, dass ich diese Master's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Munich, 01.01.2020 Behtarin Ferdousi

Acknowledgments

Yet to be written

Abstract

With the current pace of development in autonomous vehicles, the demand for high-intelligent collision avoidance system is increasing. Due to noise in measurements from Lidar, GPS and radar sensors, researchers have utilized state estimation methods to converge measurements to the true state of the system. The purpose of this thesis is to review and implement different algorithms of state estimation using zonotopes as domain representation on existing dataset of real traffic participants. The algorithms are compared in terms of error, time to converge and tightness of bound.

Contents

1 Introduction

Recent progress in the autonomous driving extrapolates to launch of such vehicles in the very near future. The race to the top of automobile industry, participated by companies like BMW, Tesla, Waymo/Google, requires fast devlopment and vigorous testing of the novel vehicles. One of the many challenges of this fields is the collision avoidance system. With no human behind wheels for level 5 cars, the vehicle must keep track of roads, surrounding vehicles, safety of the passenger along with the pedestrians in different environment including rain and fog. Current collision systems based on sensors, radar and camera will be overwhelmed with high computation for this purpose. Tolerating error in such system can cause accidents; such error in vehicles have already caused real-life accidents, including one resulting in death.

On the other hand, there is parallel development in state estimation for control theory. There are set to represent the domain of the state of the system. There are development in linear and non-linear systems. Comparing different shapes to represent state like polytopes, ellipsoinds and zonotopes, zonotopes have gained much fame due to its balance between accuracy and computation cost relative to the other representations. Furthermore, zonotopes take care of Wrapping Effect and Minkowski Sum. There are library in Matlab that implemented the functionalities in zonotope required for state estimation.

In order to utilize the state estimation algorithms, the foremost necessary step is to define the tracked vehicle in a linear model. There have been various research on identifying the models balancing between computation cost and accuracy. The models used in this paper are Constant Velocity, Constant Acceleration and the Singer Acceleration Model.

- TODO: Structure of paper
- TODO: References

2 Problem Formulation

The state of the vehicle to be tracked is x_k at time k. The measured state is y_k at the time k. The equations to predict x_k from previous step x_{k-1} and the mapping from measurement is shown in equation 2.1, where A,E,C and F are known matrices, w_k and v_k are process noise and measurement noise at time k respectively.

$$\begin{aligned} x_{k+1} &= Ax_k + Ew_k \\ y_k &= Cx_k + Fv_k \end{aligned} \tag{2.1}$$

The state of the tracked vehicle can be represented using position, velocity and acceleration in x and y-axis. Different state can be estimated using different models, whereas, the measured state of the vehicle is assumed to be position in x and y-axis for all models discussed below.

$$y = [s_x \quad s_y]^T$$

Three linear systems are implemented to compare the different algorithms for tracked vehicles. Although there exists highly precise vehicle models, simplest models are used here to represent the tracked vehicle partly because the tracked vehicle physical dimensions like wheelbase or side-slip, cannot directly be measured. Another reason is adding steering angle and yaw rate makes the system non-linear and hence does not suit all the algorithms presented. Hence, the models are:

- **Constant Velocity Model**: The vehicle is assumed to travel in constant velocity
- **Constant Acceleration Model**: The vehicle is assumed to have constant acceleration
- **Singer Acceleration Model**: The acceleration of the tracked vehicle is assumed

to be first-order Markov process of the form:

$$
\begin{aligned}
a_{k+1} &= \rho_m a_k + \sqrt{1-\rho_m^2}\sigma_m r_k \\
&\text{where} \\
\rho_m &= e^{-\beta T}, \beta = 1/\tau_m \\
\tau_m &= \text{target maneuver time constant} \\
\sigma_m &= \text{target maneuver standard deviation} \\
r_k &= \text{zero-mean unit-standard deviation Gaussian distributed random variable} \\
T &= \text{time step}
\end{aligned}
\tag{2.2}
$$

The state transition matrix, A, measurement matrix, C for each model are tabulated in Table 2.1, where for Singer Acceleration Model,

$$
\begin{aligned}
f(\Delta T) &= \frac{1}{\beta^2}(-1+\beta\Delta T+\rho_m) \\
g(\Delta T) &= \frac{1}{\beta}(1-\rho_m)
\end{aligned}
\tag{2.3}
$$

Model	$\mathbf{x}$	$\mathbf{A}$	$\mathbf{C}$
Constant Velocity	$\begin{bmatrix} s_x \\ s_y \\ v_x \\ v_y \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
Constant Acceleration	$\begin{bmatrix} s_x \\ s_y \\ v_x \\ v_y \\ a_x \\ a_y \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 \\ 0 & 0 & 1 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
Singer Acceleration	$\begin{bmatrix} s_x \\ s_y \\ v_x \\ v_y \\ a_x \\ a_y \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & \Delta T & 0 & f(\Delta T)^{[2.3]} & 0 \\ 0 & 1 & 0 & \Delta T & 0 & f(\Delta T)^{[2.3]} \\ 0 & 0 & 1 & 0 & g^{[2.3]} & 0 \\ 0 & 0 & 0 & 1 & 0 & g^{[2.3]} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$

Table 2.1: Comparing state transition matrix and measurement matrix for different vehicle models

3 State Estimation

State Estimation algorithms can be braodly classified into two types: Stochastic and Set-based algorithms. Stochastic state estimation algorithms assume that the uncertainties in the state of the system follow a known probability distributions. It is difficult to fulfill the assumption for such algorithms, however, Zorzi [1] proposed a family of Kalman Filter that solves the minimax problem with an iterative probability distribution of the uncertainties. Set-based algorithms, on the other hand, utilize geometrical sets as domain representation, like elipsoid or zonotope, to bound the possible sets of state of the system. Zonotopes are better than ellipsoids due to the balance of accuracy and computational cost. Furthermore, the zonotopes can control the wrapping effect [2], which is the term referred to the growth of the estimated state due to the propagated uncertainties in each iteration. Such algortithms can be further classified into segment intersection and interval observer. The former methods focus on intersecting the set of estimated state with the set of predicted state from the measurements. These methods try to minimize the bounds of the estimated state by using different properties of the geometric set like volume and radius. The interval observer methods, on the other hand, design observer to minimize the error on each time step. The following section digs deeper on the aforementioned methods.

3.1 Segment Intersection

The predicted state of the system at a specific time and the previous state of the system are represented by zonotopes. The state estimated is the intersection of these zonotopes. Each algorithm tries to minimize the size of the intersected segment. Different properties of zonotopes, like P-radius and volume, are considered to represent the size of the segment. The following sections list and elaborates the algorithm that depends on different properties of zonotope.

3.1.1 Frobenius norm of generators

Frobenius norm of the generators of a zonotope is calculated using formula

$$||H||_F^2 = ||A + \lambda b^T||_F^2 \tag{3.1}$$

$$\lambda^* = \frac{-Ab}{b^T b} = \frac{HH^T c}{c^T HH^T c} + \sigma^2 \tag{3.2}$$

The λ that generates the minimum Frobenius norm of the generators of the intersected zonotope is calculated using the formula 3.2 for each iteration and the minimum zonotope is calculated.

3.1.2 Volume

The volume of a zonotope is calculated using the formula 3.3.

$$Vol(\hat{X}(\lambda)) = 2^n \sum_{i=1}^{N(n,r)} |1 - c^T \lambda||det(A_i)| + 2^n \sum_{i=1}^{N(n-1,r)} \sigma|det[B_i \quad v_i]||v_i^T \lambda| \tag{3.3}$$

3.1.3 P-radius

- TODO: Implement
- TODO: Write

3.2 Interval Observer

- TODO: Write

4 Result

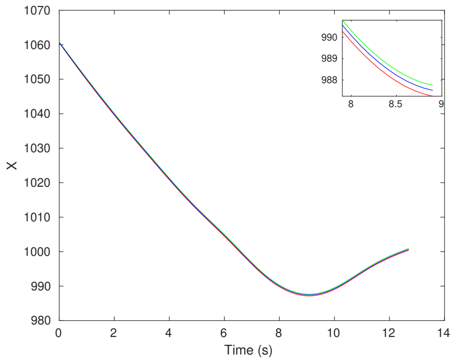
The INTERACTION Dataset [1] is used to compare the algorithms and models of traffic participant tracking. The dataset contains multiple scenario in different locations where each scenario consists of multiple participants. Each participant is identified by an id for each scenario and each frame per 0.1s has a set of vehicles and their position. The x and y position of the vehicle is noted per time step and the algorithms aforementioned are applied to compare. The initial state of the system is set using assignments 4.1.

$$\begin{aligned} x_0 &= zonotope([zeros(n), diag([1000; 1000; 10; 10; 10; 10])]) \\ w_k &= [0.1; 0.1; 0.4; 0.4; 0.1; 0.1] \\ v_k &= [0.1; 0.1] \end{aligned} \tag{4.1}$$

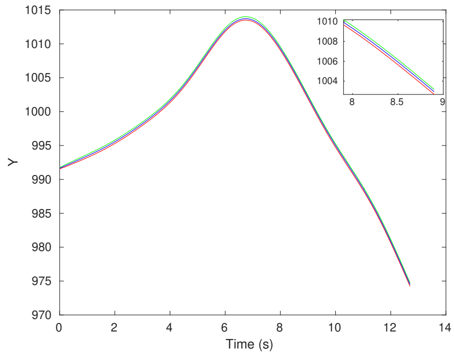
As seen from Fig. 4.1, the upper and lower bounds of the state estimation by Segment Minimization(using Frobenius norm) of the system bounds the true state. Comparing the error in different state of the system, as shown in Fig. 4.2, suggests that both the algorithm requires around 0.5s to converge to near the true state. Interesting to note, the Interval Estimation has a higher error peak compared to Segment Minimization using the same dataset.

The Fig. 4.3 shows the Histogram of RMSE of estimation from Segment Minimization. The errors are calculated after 10 seconds in order to avoid the initial peak. The histograms suggest that, the method gives little error for estimating the measured state, whereas, for unmeasured state like velocity, the error is at most twice the true state.

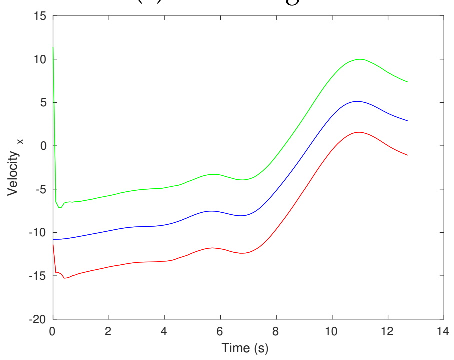
[1]https://interaction-dataset.com/



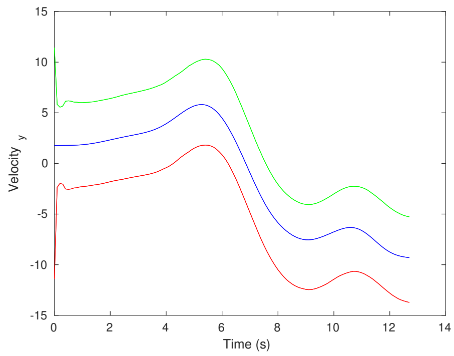
(a) Estimating x



(b) Estimating y



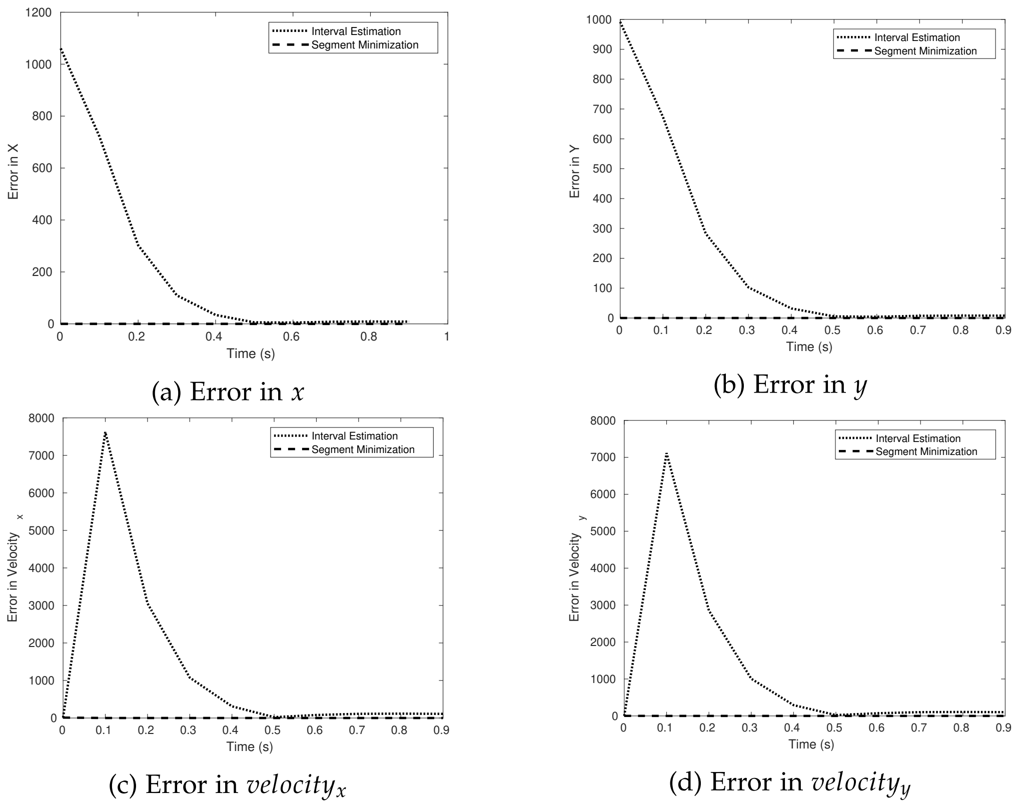
(c) Estimating $velocity_x$



(d) Estimating $velocity_y$

: Lower bound of estimation
: Upper bound of estimation
: True value

Figure 4.1: Segment Minimization using Frobenius norm



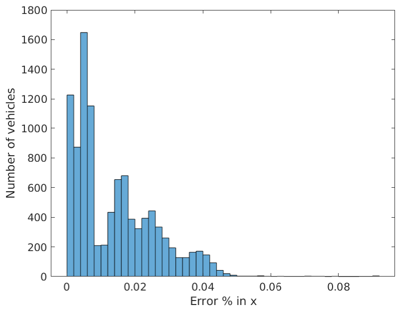
(a) Error in x

(b) Error in y

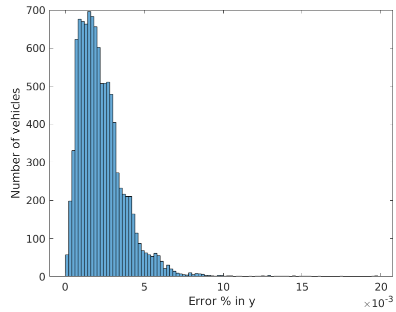
(c) Error in $velocity_x$

(d) Error in $velocity_y$

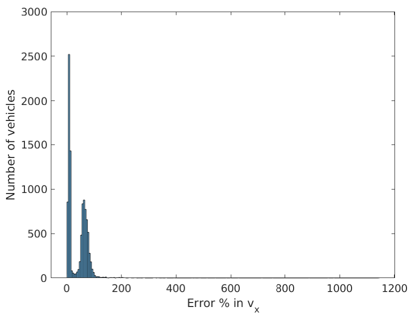
Figure 4.2: Comparing error from different algorithms on same dataset



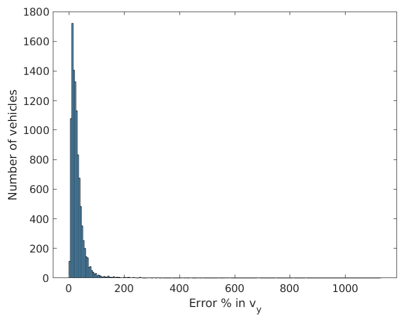
(a) RMSE x



(b) RMSE y



(c) RMSE in $velocity_x$



(d) RMSE in $velocity_y$

Figure 4.3: Histogram of errors from Segment Minimizer on 651 vehicles

List of Figures

List of Tables

Bibliography

[1] M. Zorzi. "Robust Kalman filtering under model perturbations." In: *IEEE Transactions on Automatic Control* 62.6 (2017), pp. 2902–2907. ISSN: 15582523. arXiv: `1508.01906`.

[2] W. Kühn. "Rigorously Computed Orbits of Dynamical Systems without the Wrapping Effect." In: *Computing (Vienna/New York)* 61.1 (1998), pp. 47–67. ISSN: 0010485X.