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:	11.05.2020

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, 11.05.2020 Behtarin Ferdousi

Acknowledgments

I want to thank Jagat Rath to help me from ground up for this project. His support and motivation have given me the confidence to work on this topic. Regardless of my countless silly questions, he was always patient and calm in discussions. I am grateful for his time and availability to me.

I am very lucky to have Professor Matthias Althoff for his guidance and advice. I want to express my gratitude for his time and feedback, in spite of his tremendously busy schedule.

My heartfelt gratitude goes to my husband, who calmed me when I panicked, my parents who constantly encouraged me and believed in me, my sister for sharing laughter and my in-laws for supporting and standing by me. I feel blessed to have such a wonderful company around me and I thank God with all of my heart for the strength and opportunity to study here in TUM.

Abstract

With the current rate of development in autonomous vehicles, the demand for a high-intelligent collision avoidance system is increasing. Due to the inability to determine the inner state of tracked vehicles from Lidar, GPS (Global Positioning System), and radar sensors, researchers have utilized state estimation methods to converge available measurements to the true state of the system. Set-based methods are used to enclose the true state of the system in a set, in contrast to stochastic methods which give a point-estimate close to the true state. Encapsulating the true state in a set is important to not allow any divergence from the true state for safety-critical tasks in autonomous vehicles. The purpose of this thesis is to review and implement different algorithms of set-based state estimation, using zonotopes as domain representation, on existing datasets of real traffic participants (approx. 10,518 entities). The algorithms implemented are segment intersection methods (using F-radius, P-radius, and volume) and an interval observer (using H-∞ observer). They are compared in terms of computation time, time to converge, tightness of bound and accuracy. The H-∞ interval observer performs better in terms of computation time but starts with a wider initial bound. Segment intersection minimization using P-radius is faster than using F-radius, but compromises on the bounds and accuracy. Of all the methods compared, segment minimization using F-radius gives the most desirable estimates for this use-case.

Contents

Acknowledgments **iii**

Abstract **iv**

1 Introduction **1**

2 Vehicle Localization : The Guaranteed Estimation Problem **4**

2.1 Preliminaries 4

2.1.1 Zonotopes: Definitions and Properties 4

2.2 Problem Formulation 6

2.3 Vehicle Model 7

2.3.1 Constant Velocity Model 7

2.3.2 Constant Acceleration Model 7

2.3.3 Point-Mass Model 8

3 Zonotope-based guaranteed state estimation **9**

3.1 Segment Intersection 9

3.1.1 F-radius 10

3.1.2 Volume 11

3.1.3 P-radius 11

3.2 Interval Observer 12

3.2.1 H-∞ Observer 12

4 Evaluations **14**

4.1 Computation Time 14

4.2 Time to Converge 16

4.3 Bounds 16

4.4 Accuracy 17

5 Conclusion **19**

6 Extended Results **20**

6.1 Set Estimation 20

6.1.1 Segment Minimization using F-Radius 20

6.1.2 Segment Minimization using P-Radius 24
6.1.3 Interval Observer using H-∞ 27
6.2 Rate of Change of Bounds 30

List of Figures **33**

List of Tables **34**

Bibliography **35**

1 Introduction

There is steep progress in the research and development of autonomous vehicles. The race to the top of the automobile industry, featuring companies like BMW (Bavarian Motor Works), Tesla, Waymo/Google, requires fast development and vigorous testing of novel technologies. One of the many challenges of this field is to ensure collision avoidance. With no human behind the wheel for Level 5 [1] cars, the vehicle must keep track of roads and surrounding traffic participants (like vehicles and pedestrians) in different circumstances, including rain and fog, to ensure the safety of its passengers. Current collision avoidance systems based on sensors, radar, and camera would be overwhelmed with high computation demands for this purpose. Tolerating error in such a system can cause accidents, including fatality [1].

The collision avoidance system in a car consists of two parts: sensing and tracking and motion planning. The sensing and tracking part is achieved by applying sophisticated algorithms on signals from sensors like radar, camera, and GPS (Global Positioning System). With decline in cost of cameras and advancement in technologies in image processing, image analysis, and object detection, sensing and tracking is developing fast. Although cameras can classify vehicles, they cannot guarantee measurement in a low-light environment (e.g. night) [2]. In contrast, radar guarantees robustness to weather in exchange of a higher cost. Similarly, there are limitations in GPS, e.g. the inability to function in the urban canyon environment. Thus, one uses sensor fusion to compensate for shortcomings of specific sensors . After detecting all relevant elements in the environment, a motion planner has to find a collision-free path. Computation of such a path requires certain parameters to predict the tracked vehicle's trajectory. The sensors cannot solely measure all these parameters, hence researchers have turned to state estimation algorithms.

One of the widely-applied state estimation techniques is the Kalman filter [3], which can estimate target dynamics for measurement with additive Gaussian noise. Despite its simplicity, the filter is not suitable for vehicle localization for two reasons. Firstly, statistical noise with known covariance is, unfortunately, not practical. Secondly, the filter provides close point-estimation, relying on which can be safety-critical. These motivate to use set-based state estimation methods.

[1] https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe

The set-based state estimation technique computes a set of state enclosing the true state of the system as long as the dynamics are accurately modeled and the noise and perturbations have known bounds. The main steps are prediction step and correction step. The prediction step extrapolates prior estimate, while the correction step improves the extrapolation. There are multiple algorithms with varying approach for the correction step. Another differentiating factor is the choice of geometric shape to represent the estimated set. Zonotope is one of the popular choice, compared to ellipsoid and pollytopes, due to higher accuracy for a lower computation cost. Furthermore, zonotopes have gained fame for state estimation because of wrapping effect(i.e. not increasing in size due to accumulated noises over time) and Minkowski sum(i.e. the sum of zonotopes is also a zonotope). Therefore, we chose zonotopes to represent state for all the algorithms in this paper.

The first zonotope-based guaranteed state estimation is developed by Puig et al. in 2001 [4], when they used gain matrix to map input measurement to a set of estimation. Following in 2003, Combastel [5] used a singular value decomposition to overapproximate the estimate consistent with the input. Although aforementioned methods are computationally light, they did not focus on the size of the estimated region. In 2005, T. Alamo et al. [6] formulated convex optimization problem to minimize some size criterion. They focused on two main size criterion: F-radius and volume. F-radius resulted in fast but convervative estimates, whereas volume computation is heavy, but gave tight bounds. In 2011, T. Alamo et al. [7] optimized P-radius to obtain good accuracy for a reasonable computation load. Initially, the algorithms were developed for single-output linear discrete systems, and were later generalised to multi-output and non-linear systems.

Another classification of set-based estimation are interval observers, where the idea is to design observers such that the error in the estimation is minimal. Despite its high efficiency, construction of such observer is not very easy. Hence, the observer design requirements are relaxed in [8], [9]. The relaxation results in conservatism, which lead to an interval observer based on H-∞ with reachability analysis.

The prequisite step before applying state estimation algorithms is to model the tracked vehicle in a well-defined mathematical model. Although there are complex models that can be used to represent a vehicle state [10], not all can be used due to the unavailability of parameters like wheelbase, velocity, etc. accessible to the ego vehicle. Hence, the models used in this paper to compare are the simplest, yet complete enough to determine the properties of the tracked vehicle for trajectory prediction: Constant Velocity, Constant Acceleration, and the Point-Mass Model.

A high degree of accuracy and guarantee is the necessity of the collision avoidance system, hence we chose to compare the set based state estimation algorithms for different scenarios involving dynamic traffic participants from a dataset collected from intersections using drones and fixed cameras. [11] has encouraged many sections in this paper and compares a superset of algorithms covered here; however, the algorithms were compared on simulated data, in contrast to this paper.

The paper is organized as follows. Chapter 2 presents the vehicle localization problem to be solved by state estimation algorithms. The following chapter 3 discusses the zonotope-based state estimation algorithms to be compared. Chapter 4 gives the evaluation of the algorithms, with extended results in chapter 6. Finally, chapter 5 concludes with a summary and a discussion of possible future works.

2 Vehicle Localization : The Guaranteed Estimation Problem

2.1 Preliminaries

The following standard notations are maintained in this paper.

- $\mathcal{R}^n$ and $\mathcal{R}^{n\times m}$ denote the n and $n \times m$ dimensional Euclidean space, respectively.
- I^n represents the n-identity matrix. If n is missing, then appropriate dimension is assumed.
- For a matrix A, A^T, A^{-1}, A_i and A^j denote its transpose, inverse, i^{th} row, and j^{th} column, respectively . $rs(A)$ is the row sum of A, and $det(A)$ the determinant.
- $|.|$ is the absolute value and $||.||_x$ is the x-norm.
- With a vector $a \in \mathcal{R}^n$, $diag(a)$ is a diagonal matrix of dimension n.
- For a real symmetric matrix, $P \in \mathcal{R}^{n\times n}, P \prec 0 (P \succ 0)$ implies P is a negative (positive) definite.

2.1.1 Zonotopes: Definitions and Properties

The following definitions and properties are essential for this paper.

Definition 1 ***Intervals*** *An interval $[a, b]$ is defined as the set $\{x : a \leq x \leq b\}$. The unitary interval, denoted by $\boldsymbol{B}$, is [-1,1]. A box($([a_1, b_1], ..[a_n, b_n])^T$) is an interval vector. A unitary box in $\mathcal{R}^n$ is denoted by $\boldsymbol{B}^n$ and is a box with n unitary intervals.*

Definition 2 *The Minkowski sum of two sets,$\mathcal{X}$ and $\mathcal{Y}$, is defined by:*

$$\mathcal{X} \bigoplus \mathcal{Y} = \{\boldsymbol{x} + \boldsymbol{y} : \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}\} \tag{2.1}$$

Definition 3 ***Zonotope****: An affine transformation of a hypercube, $\boldsymbol{B}^m$ is called m-ordered zonotope, denoted by $\mathcal{Z} \in \mathcal{R}^n$:*

$$\mathcal{Z} = \langle p, H\rangle = \{p + H\boldsymbol{z} : \boldsymbol{z} \in \boldsymbol{B}^m\} \tag{2.2}$$

where $p \in \mathcal{R}^n$ is the center of $\mathcal{Z}$ and $H \in \mathcal{R}^{n\times m}$ is called the generator of $\mathcal{Z}$.

Property 1 *For two zonotopes, $\mathcal{Z}_1 = \langle p_1, H_1\rangle$ and $\mathcal{Z}_2 = \langle p_2, H_2\rangle$, the following equations hold:*

$$
\begin{aligned}
\mathcal{Z}_1 \bigoplus \mathcal{Z}_2 &= \langle p_1 + p_2, [H_1 \quad H_2]\rangle \\
L\mathcal{Z}_1 &= \langle Lp_1, LH_1\rangle
\end{aligned}
\tag{2.3}
$$

Property 2 *[12] A box of an m-zonotope , $\mathcal{Z} \in \mathcal{R}^n$, is an n-interval vector, over-approximating the zonotope such that:*

$$
\lceil \mathcal{Z} \rceil = box(\mathcal{Z}) = [p - \Delta H, p + \Delta H], \quad \Delta H = \sum_{i=1}^{m} |H^i| \tag{2.4}
$$

where $\mathcal{Z} = \langle p, H\rangle$

Property 3 ***Zonotope Reduction*** *[6], [5]: Given an m-zonotope $\mathcal{Z} = \langle p, H\rangle \in \mathcal{R}^n$, and the integer s, with $n < s < m$, let's denote $\hat{H}$ as the resulting matrix after reordering the columns of the matrix $\hat{H} = [\hat{h}_1 ... \hat{h}_m]$ in decreasing order of Euclidean norm ($\hat{H} = [\hat{h}_1 ... \hat{h}_m]$, with $||\hat{h}_i||_2 \geq ||\hat{h}_{i+1}||_2$). Let $\hat{H}_A$ denote the matrix obtained from the first $s - n$ columns of $\hat{H}$, and $\hat{H}_B$ be the rest of the matrix $\hat{H}$. Then the following inclusion is obtained:*

$$
\mathcal{Z} \subseteq p \bigoplus [\hat{H}_A \quad rs(\hat{H}_B)]\boldsymbol{B}^s \tag{2.5}
$$

It is denoted by $\mathcal{Z}_{\downarrow s}$ in this paper.

Property 4 *[6] Given a zonotope $\mathcal{Z} = p \oplus H\boldsymbol{B}^m \in \mathcal{R}^n$, a strip $\mathscr{S} = x \in \mathcal{R}^n : |cx - y| \leq \phi$, and a vector $\lambda \in \mathcal{R}^n$, define a vector $p(\lambda) = p + \lambda(y - cp) \in \mathcal{R}^n$ and a matrix $H(\lambda) = I - \lambda c)H \quad \phi\lambda$. Then a family of zonotopes parameterized by λ that contains the intersection of a zonotope and a strip is obtained such as*

$$
\mathcal{Z} \cap \mathscr{S} \subseteq \mathcal{Z}(\lambda) = p(\lambda) \bigoplus H(\lambda)\boldsymbol{B}^{m+1} \tag{2.6}
$$

The construction, reduction and interval calculation of zonotopes are implemented in the Matlab® toolbox CORA (COntinuous Reachability Analyzer) [13].

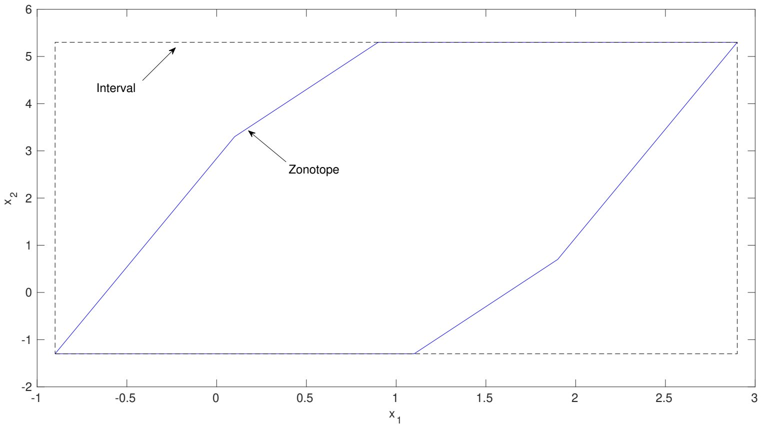

Figure 2.1: An illustration of a zonotope and its interval hull in 2-D

2.2 Problem Formulation

Let us denote the state of the vehicle to be tracked at time k as x_k and the measured state as y_k. (2.7) formuates a multi-output discrete-time linear system for the tracked vehicle, where $A, E \in \mathcal{R}^{n_x \times n_x}$, $C \in \mathcal{R}^{n_y \times n_x}$, and $F \in \mathcal{R}^{n_y \times n_y}$ are matrices defined by the vehicle system model; w_k and v_k are process noise and measurement noise at time k, respectively.

$$\begin{aligned} x_k &= Ax_{k-1} + Ew_k \\ y_k &= Cx_k + Fv_k \end{aligned} \tag{2.7}$$

As w_k and v_k have known bounds ($\overline{w_k}$ and $\overline{v_k}$), we can define $\mathcal{W}$ and $\mathcal{V}$ such that $w_k \in \mathcal{W}$ and $v_k \in \mathcal{V}$ as (2.8).

$$\mathcal{W} = \langle 0, H_w \rangle, \quad \mathcal{V} = \langle 0, H_v \rangle \tag{2.8}$$

The dimension of x_k (n_x) varies across vehicle models; however, the measurement vector (y_k) is fixed to position in x and y-direction (2.9).

$$y = [s_x \quad s_y]^T \tag{2.9}$$

Given a vehicle model (discussed in the next section), the problem of set-based state estimation is to compute an outer bound of the state (x_k) containing all the possible values of the true state of the system consistent with the uncertain vehicle model and measurements.

2.3 Vehicle Model

A major performance-influencing factor is to choose the right model for the tracked vehicle. Three linear systems are implemented in this paper to compare the different algorithms for tracked vehicles. Although there exists highly precise vehicle models for ego vehicles, the simplest models are used here to represent the tracked vehicle as complex vehicle models require parameters which are non-acquirable for tracked vehicles. In particular, physical dimensions like wheelbase or side-slip, cannot be measured directly. Another reason is that adding steering angle and yaw rate makes the system non-linear and hence does not suit all the algorithms presented. Hence, the following models are investigated:

- **Constant Velocity Model**
- **Constant Acceleration Model**
- **Point Mass Model**

2.3.1 Constant Velocity Model

The vehicle is assumed to travel in constant velocity [14]. The state of the system (x_k), state transition matrix (A), and the measurement matrix(C) is shown in (2.10).

$$
\begin{aligned}
x &= \begin{bmatrix} s_x & s_y & v_x & v_y \end{bmatrix}^T \\
A &= \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
\end{aligned}
\tag{2.10}
$$

2.3.2 Constant Acceleration Model

Although the constant velocity model is easy to implement, it is unrealistic to assume constant velocity. Acceleration model takes care of changing velocity and assumes constant acceleration [14].Hence, the estimation error for position and velocity are relatively smaller when the velocity is constantly changing. The state of the system (x_k), state transition matrix (A), and the measurement matrix(C) is shown in (2.11).

$$
\begin{aligned}
x &= \begin{bmatrix} s_x & s_y & v_x & v_y & a_x & a_y \end{bmatrix}^T \\
A &= \begin{bmatrix}
1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 & 0 \\
0 & 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 \\
0 & 0 & 1 & 0 & \Delta T & 0 \\
0 & 0 & 0 & 1 & 0 & \Delta T \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \\
C &= \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{aligned}
\tag{2.11}
$$

2.3.3 Point-Mass Model

It is trivial to note that vehicles might have varying acceleration which is not satisfied in the previous models, which brings us to the point-mass model [10], which is similar to the constant acceleration model, except that the acceleration can strike up to a certain limit. This model treats the tracked vehicle as a point mass, ignoring wheel-base, slip-angle, etc. of the tracked vehicle. The state transition and measurement matrices are the same as the constant acceleration model. The acceleration bounds are set as $11.5m/s^2$ in both x and y-direction for this paper.

3 Zonotope-based guaranteed state estimation

With the essential knowledge on zonotope and vehicle models from the previous chapter, this chapter digs deeper into the state estimation algorithms and discusses how they can be applied to track vehicle dynamics. With the vehicle dynamics represented by 3.1, we look into segment intersection techniques (minimizing F-radius, volume and P-radius) and the H-∞ based interval observer.

$$\begin{aligned} x_{k+1} &= Ax_k + Ew_k \\ y_k &= Cx_k + Fv_k \end{aligned} \tag{3.1}$$

3.1 Segment Intersection

For the system in (3.1), let the set of predicted state of the system at time k be denoted by an r-zonotope, $\overline{\mathcal{X}_k} = \langle p, H \rangle$. The set to represent the i^{th} state in measurement($y_{k/i}$) at time k is a strip, denoted by $\mathscr{S}_i = \{x \in \mathbb{R} : |C_i x - y_{k/i}| \leq v_{k/i}\}$. The estimation at time k, denoted by $\hat{\mathcal{X}}_k$, is the intersection of the strip, $\mathscr{S}$, and the zonotope $\overline{\mathcal{X}_k}$, which can be parametrized by a vector $\lambda_i \in \mathbb{R}^n$ according to Property 4 to Eq. 3.2.

$$\begin{aligned} \hat{\mathcal{X}}_{k/i} &= \hat{p}(\lambda_i) + \hat{H}(\lambda_i)\mathbf{B}^{r+1} \\ \textit{where} \quad \hat{p}(\lambda_i) &= p + \lambda_i(y_{k/i} - C_i p) \\ \textit{and} \quad \hat{H}(\lambda_i) &= [(I - \lambda_i C_i)H \quad v_{k/i}\lambda_i] \end{aligned} \tag{3.2}$$

The motive of segment intersection methods is to find the value of λ such that the intersected segment is compact. For every iteration, the order of the zonotope increases, and hence to reduce accumulating compuation burden, the estimated zonotope is reduced to maximum order of 20 for this paper using the reduction function in CORA. In the following sections, we briefly discuss three different approaches to solve λ that parametrizes the minimum intersected zonotope by focusing on three distinguishing properties of zonotope.

Alg. **??** summarizes the skeleton of segment intersection. Set of state estimated at time k is represented by a zonotope, $\hat{\mathcal{Z}}_k$. The consistent state with respect to measurement is $\overline{\mathcal{X}_y}$ and is intersected with the predicted estimation 2. The intersected segment can be parametrized by λ according to Property 4.The motive of segment intersection methods is to find the value of λ such that the intersected segment is compact. For every iteration, the order of the zonotope increases, and hence to reduce accumulating compuation burden, the estimated zonotope is reduced to maximum order of m. In the following sections, we briefly discuss three different approaches to solve λ that parametrizes the minimum intersected zonotope by focusing on three distinguishing properties of zonotope.

Algorithm 1 Segment intersection of state

Input y
Output $\overline{x}$, $\underline{x}$

1: $\overline{\mathcal{X}_k} \leftarrow$ PREDICT$(\hat{\mathcal{X}}_{k-1})$
2: $\hat{\mathcal{X}} \leftarrow \overline{\mathcal{X}_k} \cap \overline{\mathcal{X}_y}$
3: $\hat{\mathcal{X}}_k \leftarrow \hat{\mathcal{X}_{k\downarrow m}}$
4: $[\overline{x}, \underline{x}] \leftarrow$ INTERVAL$(\mathcal{X}_k)$

3.1.1 F-radius

The F-radius of a zonotope is the F-norm of its generators, calculated by Eq. 3.3. Let us rewrite $\hat{H}(\lambda)$ as $A + \lambda b^T$ such that $A = [H \quad 0]$ and $b^T = [-C_i H \quad v_{k/i}]$.

Thus, the Frobenius norm of the generators of a zonotope is calculated using the formula (3.3)[6].

$$\begin{aligned} ||H||_F^2 &= ||A + \lambda b^T||_F^2 \\ &= 2\lambda^T A b + (b^T b)\lambda^T \lambda + tr(A^T A) \end{aligned} \tag{3.3}$$

$$\lambda^* = \frac{-Ab}{b^T b} = \frac{HH^T C_i^T}{C_i HH^T C_i^T} + v_{k/i}^2 \tag{3.4}$$

The λ^* that corresponds to the minimum Frobenius norm of the generators of the intersected zonotope is calculated using the formula (3.4) for each measurement in each iteration and the minimum zonotope to represent the estimation is calculated.

Algorithm 2 Estimation by minimizing F-norm of intersected segment

Input: y_k

Output: $\overline{x_k}$, $\underline{x_k}$

$\langle p, H\rangle \leftarrow \mathcal{X}$

for $j \leftarrow 0$ to n_y **do**

  $\lambda \leftarrow$ CALCULATE_LAMBDA(H, C_j, V_j)

  $p \leftarrow p + \lambda(y_{k/j} - C_j p)$

  $H \leftarrow [(I - \lambda C_j)H \quad V_j\lambda]$

end for

$\mathcal{X} \leftarrow \langle p, H\rangle$

$[\overline{x_k}, \underline{x_k}] \leftarrow$ INTERVAL$(\mathcal{X})$

$\mathcal{X} \leftarrow \mathcal{X}_{\downarrow m}$

3.1.2 Volume

Volume is a precise metric directly proportional to the size of the zonotope. The volume of the $\hat{\mathcal{X}}_k$ for i^{th} measurement state is [6]:

$$\begin{aligned} Vol(\hat{X}(\lambda)) =& 2^n \sum_{j=1}^{N(n,r)} |[(I - \lambda C_i)det(A_j)| \\ &+ 2^n \sum_{j=1}^{N(n-1,r)} \sigma|det[(I - \lambda C_i)B_j \quad v_k/i\lambda]| \end{aligned} \tag{3.5}$$

where $N(n, r)$ denotes the number of combinations of r elements from a set of n elements, A_j and B_j denote each of the different matrices generated by choosing n and $n-1$ columns from H respectively.

For this paper, the `zonotope.volume` function provided by CORA is used along with `fmincon` solver in Matlab® to find the value of λ corresponding to the minimum volume of the intersected zonotope. Although volume minimizes the intersected zonotope significantly, the calculation of volume is extremely computationally heavy. Therefore, it works best for use-cases that are not time-sensitive, e.g. fault diagnosis and fault-tolerant control systems [15].

3.1.3 P-radius

The P-radius of a zonotope can be calculated with the formula (3.6) where P is a positive definite matrix [6].

$$\max_{z\in Z}(||z-p||_P^2) = \max_{z\in Z}((z-p)^T P(z-p)) \tag{3.6}$$

To make sure the P-radius does not increase in every iteration, λ can be computed off-line by solving the LMI(Linear Matrix Inequality) in Equation (3.7) using Mosek solver in Matlab®.

$$\begin{bmatrix} \beta & P & 0 & A^T P - A^T C_i Y^T \\ * & F^T F & 0 & F^T P - F^T C_i Y^T \\ * & * & v_k/i^2 & Y^T v_k/i \\ * & * & * & P \end{bmatrix} \succeq 0, where \quad Y = P\lambda_i \tag{3.7}$$

Due to off-line computation, this method is substantially faster and has been used in lower accuracy-prone systems like secure monitoring of cyber-physical systems against attacks [16].

3.2 Interval Observer

Interval observers need to design observers to minimize the error in the estimation. For the system in (3.1), (3.8) defines the observer, where L is the observer gain to be designed. The design of such observers is not very easy. The following section discusses about a method, which uses H-∞ observer design.

$$x_{k+1} = Ax_k + L(y_k - Cx_k) \tag{3.8}$$

3.2.1 H-∞ Observer

The interval observer, proposed in [17], designs the observer gain to minimize the estimation error in each step by using the observer gain as $L = P^{-1}Y$ with P, a positive definite matrix with dimension $n_x \times n_x$, and Y, a matrix with dimension $n_x \times n_y$, both solution to the optimization problem in (3.9).

$$\underset{\gamma_2}{min}\ s.t.\ (3.10) \tag{3.9}$$

$$\begin{bmatrix} I_{n_x} - P & * & * & * \\ 0 & -\gamma^2 I_{n_w} & * & * \\ 0 & 0 & -\gamma^2 I_{n_v} & * \\ PA - YC & PE & -YF & -P \end{bmatrix} \prec 0 \tag{3.10}$$

With L derived using a Mosek solver in Matlab®, the estimator is initialized with a *model* and the following parameters

$$\begin{aligned} \mathcal{W} = \langle 0, H_w \rangle, &\quad \mathcal{V} = \langle 0, H_v \rangle \\ D_w = E\mathcal{W}, &\quad D_v = -LF\mathcal{V} \\ S_x = \langle 0, H_0 \rangle, &\quad S_w = \emptyset, \quad S_v = \emptyset \end{aligned} \tag{3.11}$$

where $H_w = diag(\overline{w})$ and $H_v = diag(\overline{v})$

For every measurement, y, the estimator estimates using the Alg. 3.

Algorithm 3 Estimation using H-∞ interval observer

Input y

Output $\overline{x}$, $\underline{x}$

1: $[\overline{e}, \underline{e}] \leftarrow \text{INTERVAL}(S_x) \bigoplus S_w \bigoplus S_v$
2: $\overline{x} \leftarrow \hat{x} + \overline{e}$
3: $\underline{x} \leftarrow \hat{x} + \underline{e}$
4: $\hat{x} \leftarrow A\hat{x} + L(y - C\hat{x})$
5: $S_x \leftarrow (A - LC)S_x$
6: $S_w \leftarrow S_w \bigoplus \text{INTERVAL}(D_w)$
7: $S_v \leftarrow S_v \bigoplus \text{INTERVAL}(D_v)$
8: $D_w \leftarrow (A - LC)D_w$
9: $D_v \leftarrow (A - LC)D_v$

4 Evaluations

The INTERACTION Dataset [1] is used to compare the algorithms and models for tracking traffic participants. The dataset contains multiple scenarios in different locations captured using drones or fixed cameras over a variable amount of time. Each scenario consists of multiple traffic participants, identified by an ID, and each frame per 0.1s has a set of vehicles and their position and velocity in the x and y-direction. Over different videos, the location with video of maximum length, 259.43 minutes, is chosen for this paper. There are 60 recorded files in this location with a total of 10,518 vehicles. The position for the vehicles in the x and y-direction is used as a measurement input to the algorithms, whereas the velocity in the x and y-direction are used to calculate the error and evaluate the estimates. Without loss of generality, the matrices E and F are set as identity matrices with proper dimensions. The initial state of the system is set using assignments (4.1).

$$
\begin{aligned}
\mathcal{X}_0 &= \langle 0, diag([1000100010101010]^T)\rangle \\
\overline{w_k} &= [0.10.10.40.40.10.1]^T \\
\overline{v_k} &= [0.10.1]^T
\end{aligned}
\tag{4.1}
$$

All the evaluation is carried out by single-threaded scripts run on an Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz machine with MATLAB® 2019b using CORA toolbox for set computations and the Mosek solver in YALMIP toolbox for optimization problems.

4.1 Computation Time

Fig. 4.1 shows that computation time for volume minimization rises exponentially with time making it futile in the state estimation for collision avoidance system. On the contrary, Tab. 4.1 shows that the computation time for the other methods are negligible compared to the frame rate, i.e 100ms. Furthermore, the interval observer using H-∞ has almost half the time required for the segment intersection methods, although the computation time does not consider the time for pre-computation for the techniques.

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

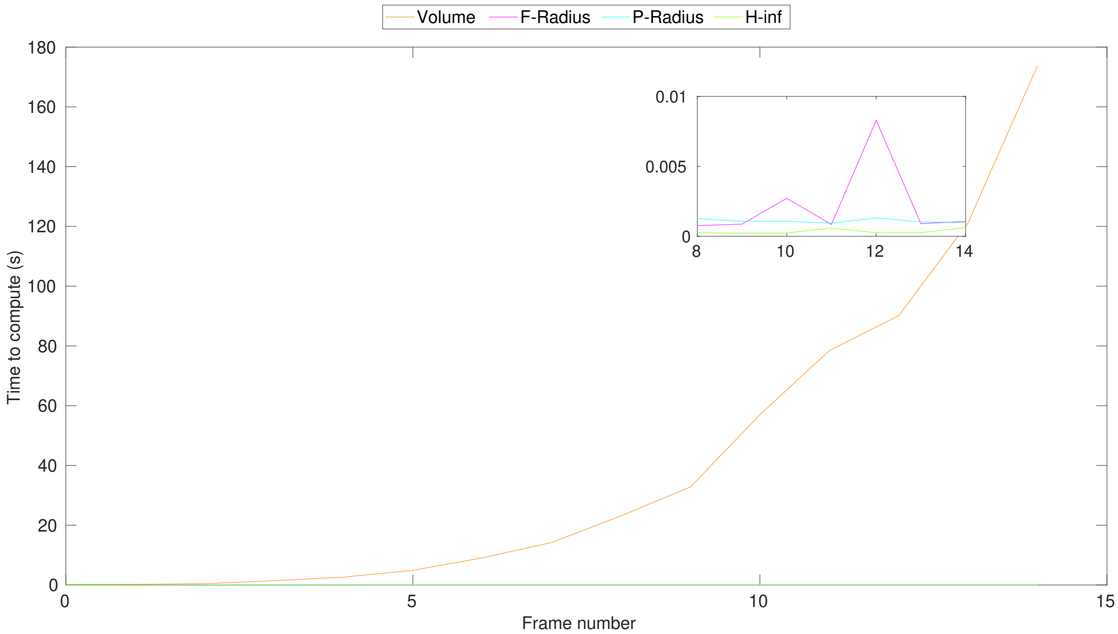

Figure 4.1: Computation time for each method to estimate using Constant Velocity Model

Table 4.1: Comparison of computation time (ms)

	Average Computation Time (ms)		
Method	CV model	CA model	PM model
F-Radius	0.396	0.375	0.621
P-Radius	0.312	0.319	0.544
H-∞ approximation	0.145	0.147	0.144

4.2 Time to Converge

Table 4.2: Comparison of average time(in ms) to converge for unmeasured state

Method	Constant Velocity	Constant Acceleration	Point-Mass Model
F-Radius	30	50	22
P-Radius	32	45	35
H-∞ approximation	24	50	35

Tab. 4.2 compares the time for each of the technique to converge unmeasured state(velocity for constant velocity, acceleration for the rest). Segment intersection using F-radius works the best using point-mass model, as it converges the fastest.

Section 6.1 in the Extended Result chapter shows the estimated bounds for all the models on a vehicle with varying acceleration. On comparing the estimation of velocity, the same bounds are obtained from constant acceleration and point-mass model, however, the bounds of acceleration from the latter are better than the former. Hence, the point-mass model is used to compare the algorithms for estimating acceleration in the following sections.

4.3 Bounds

Table 4.3: Comparison of bounds of estimation

	Constant Velocity					
Method	s_x	s_y	v_x	v_y	a_x	a_y
F-Radius	.441	.441	5.686	5.686	-	-
P-Radius	.4606	.459	13.45	13.45	-	-
H-∞ approximation	.9867	.937	6.177	6.177	-	-
	Point-Mass Model					
	s_x	s_y	v_x	v_y	a_x	a_y
F-Radius	.5713	.5075	8.461	8.461	15.79	15.78
P-Radius	.4598	.4523	16.39	16.39	16.43	16.18
H-∞ approximation	1.5	1.5	9.414	9.414	16.11	16.24

Bounds using constant velocity model is tighter compared to point-mass model as seen from Tab. 4.3. Segment intersection using F-radius has tighter bounds compared to the other techniques. Interesting to note, the H-∞ has much higher bounds in the initial time steps as can be seen from Sec. 6.1.3.

4.4 Accuracy

Accuracy is represented by the root mean square error (RMSE) of the estimation from the true state of the system. Since, the dataset does not have the measurement for acceleration, the accuracy of acceleration cannot be evaluated.

The initial estimation before convergence gives an extreme error which affects the result, hence the estimations after convergence (i.e. after 50 time-steps) are allowed in the evaluation. The RMSE is then computed as a percentage from the maximum measurement in the time frame.

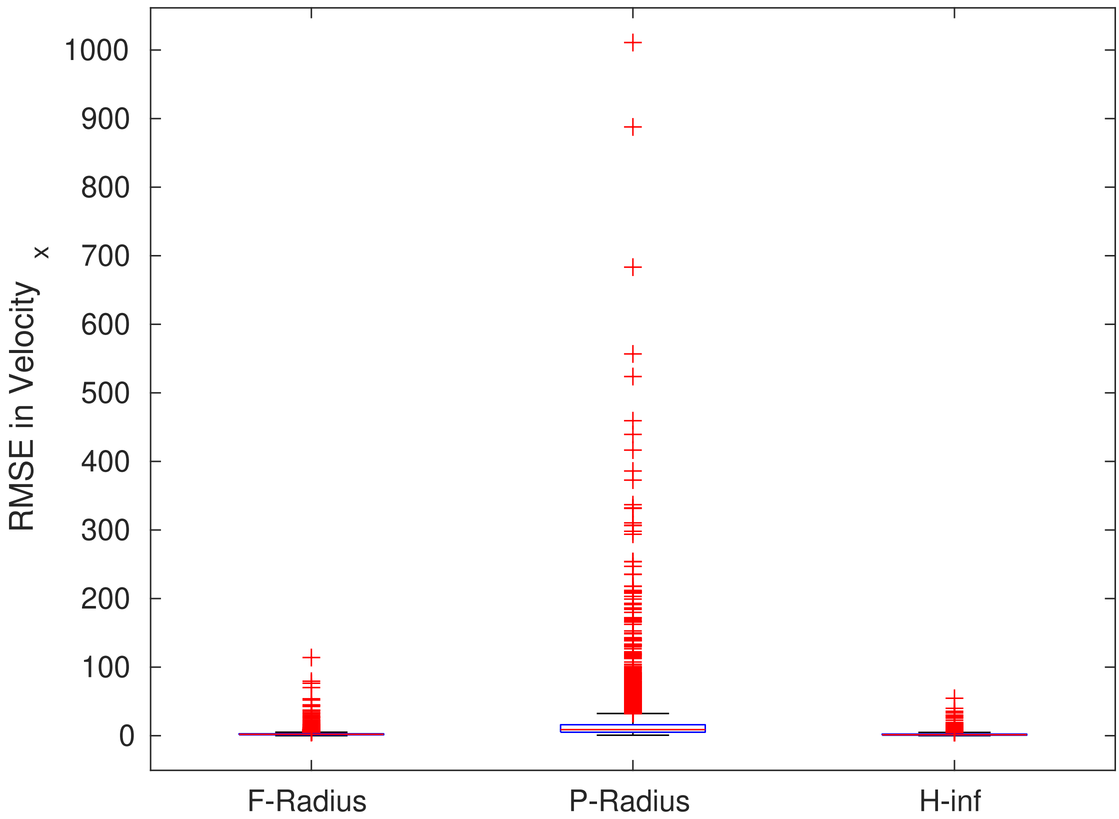

Figure 4.2: RMSE(Root Mean Squared Error)

The boxplot of RMSE using measurements in velocity in x-direction using the point-mass model for all the techniques are shown in Fig. 4.2. The segment minimization using P-radius has a high range of extremes and the mean error is also greater than the other methods. The range of error is the lowest for H-∞ observer with mean 1.9048 and standard deviation of 1.9776 for $velocity_x$. A detailed report of the mean and standard deviation of each of the methods can be found in Tab. 4.4. The error expectation from segment minimization from F-radius is similar to H-∞ observer, however, the error in the measured state is slightly lesser in F-Radius compared to H-∞ observer.

Table 4.4: Comparison of RMSE

	Mean ± SD			
Method	s_x	s_y	v_x	v_y
F-Radius	0.0007 ± 0.0004	0.0004 ± 0.0003	2.3412 ± 2.7092	2.4184 ± 1.5641
P-Radius	0.0016 ± 0.0020	0.0008 ± 0.0017	13.4470 ± 26.2465	13.7642 ± 54.6724
H-∞ approximation	0.0007 ± 0.0004	0.0006 ± 0.0005	1.9048 ± 1.9776	2.1230 ± 2.1946

5 Conclusion

A demand for intelligent collision avoidance system is timeless. To take the load off sensors and hardware of a vehicle, state estimation algorithms can be used to track vehicles and estimate properties required for collision-free path prediction. On comparing multiple techniques using different models to represent the tracked vehicle, it can be concluded that the segment intersection minimizing F-radius ensures faster convergence to more accurate and tighter bounded estimation. H-∞ and P-radius carry out off-line computation and hence are ahead in terms of run-time computation cost; nonetheless, as a consequence, these methods over-approximate and do not improve estimation significantly for each measurement. The choice of a model to represent the state of the system also has a significant effect on the performance. To estimate velocity, the constant velocity model gives better results, whereas, for acceleration, the point-mass model gives a better estimate compared to the constant acceleration model. This paper can be a starting point to implement higher-defined models of the tracked vehicle and compare the performance of state estimation methods. The state estimation methods can further be evaluated on implementing with a distinguishable set of initial starting states to determine the effect of the initial estimated state on the algorithms if any. Further developments can be to use non-linear state-estimation algorithms on complex vehicle models and compare the performance.

6 Extended Results

6.1 Set Estimation

Estimation using the techniques with different models are illustrated using data for one particular vehicle in the dataset in this chapter. Results show that the true state is always bounded by the set of estimation. For acceleration, there is no true measurement because the acceleration of the tracked vehicle is absent in the dataset.

6.1.1 Segment Minimization using F-Radius

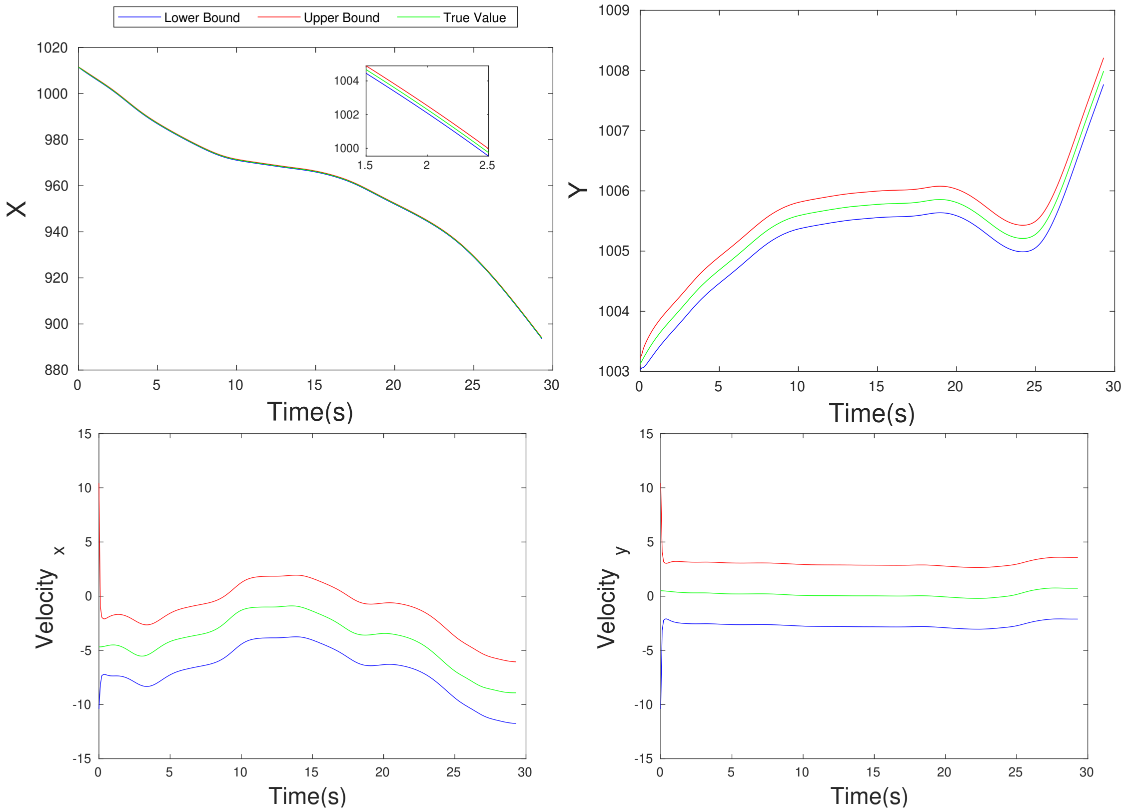

Figure 6.1: Estimation using Constant Velocity

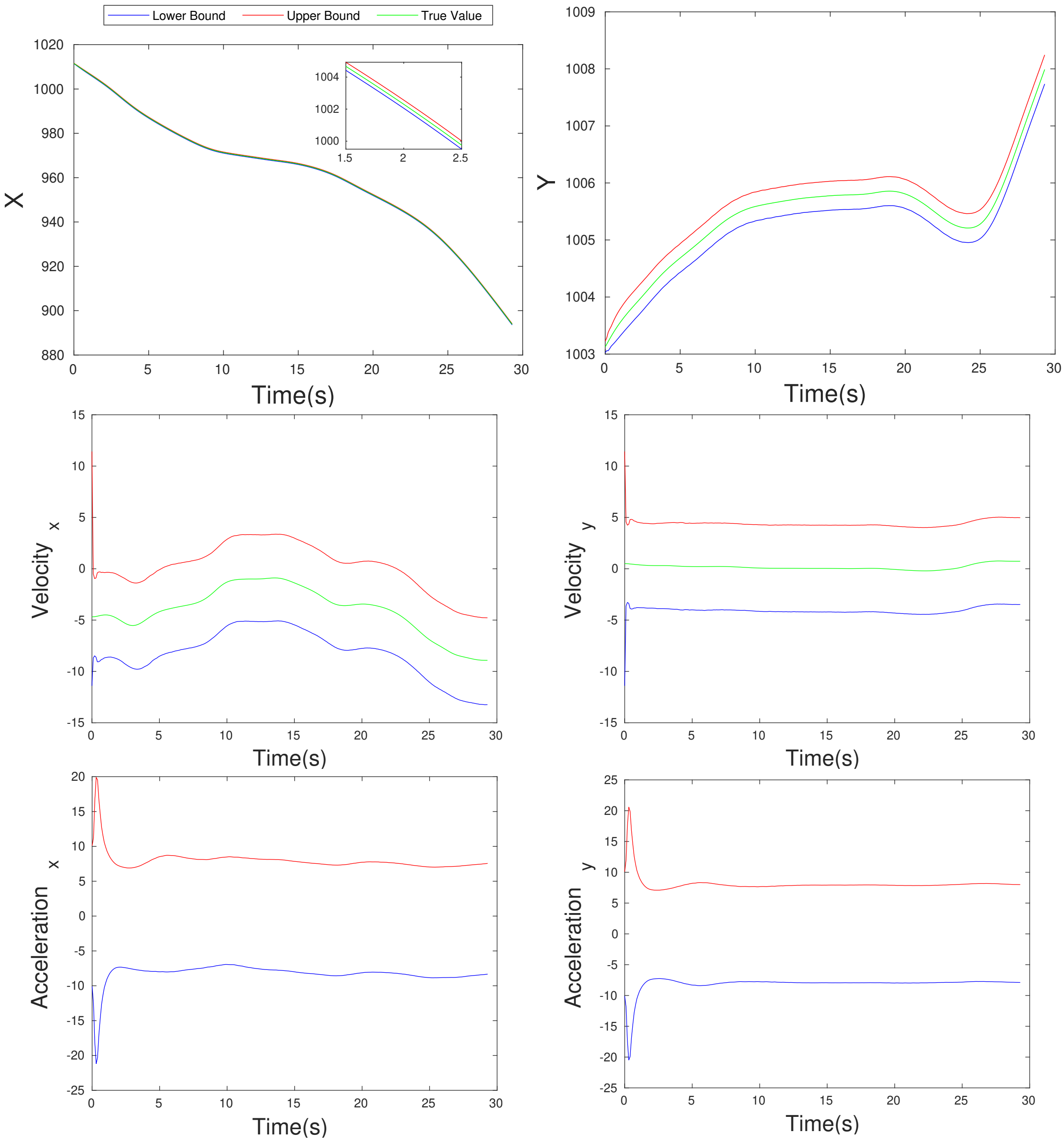

Figure 6.2: Estimation using Constant Acceleration

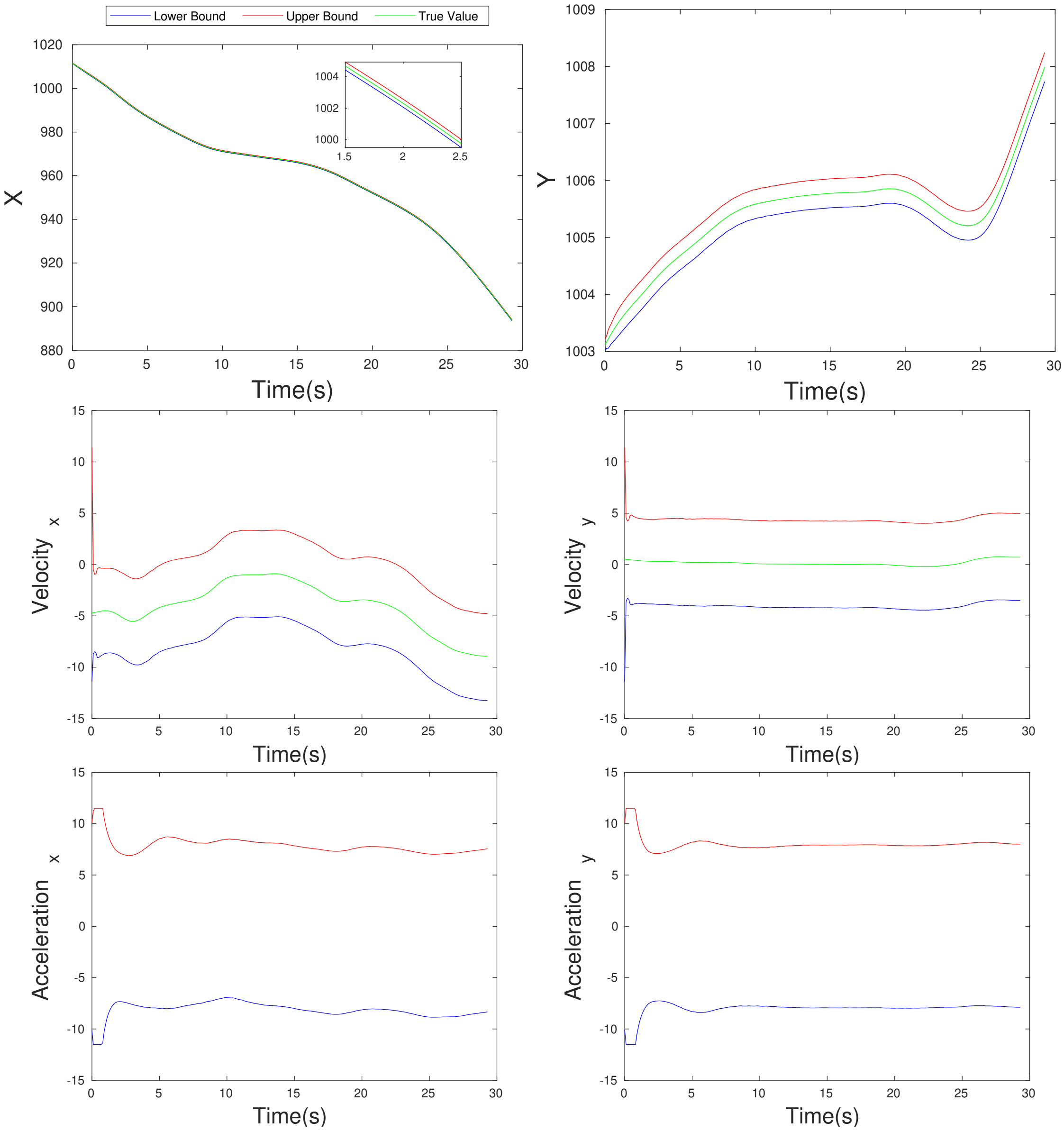

Figure 6.3: Estimation using Point Mass Model

6.1.2 Segment Minimization using P-Radius

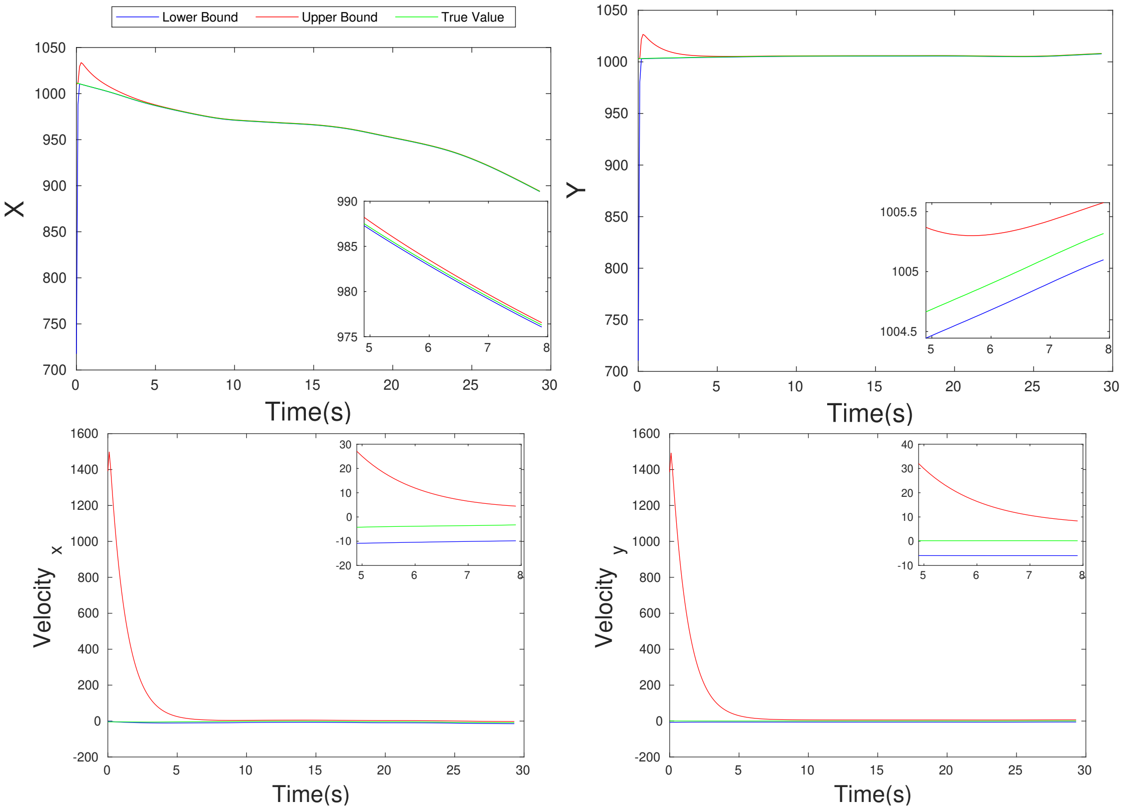

Figure 6.4: Estimation using Constant Velocity

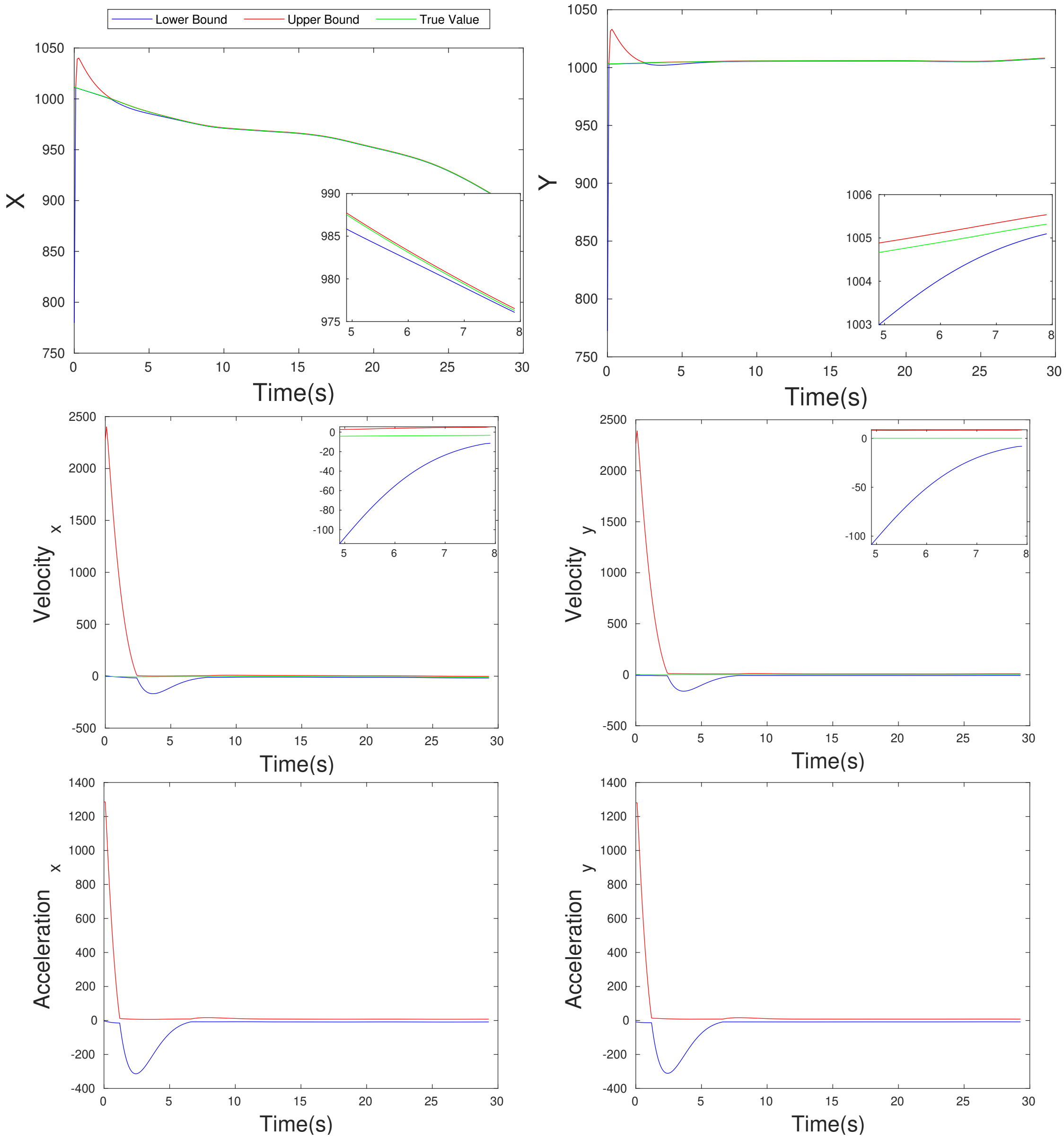

Figure 6.5: Estimation using Constant Acceleration

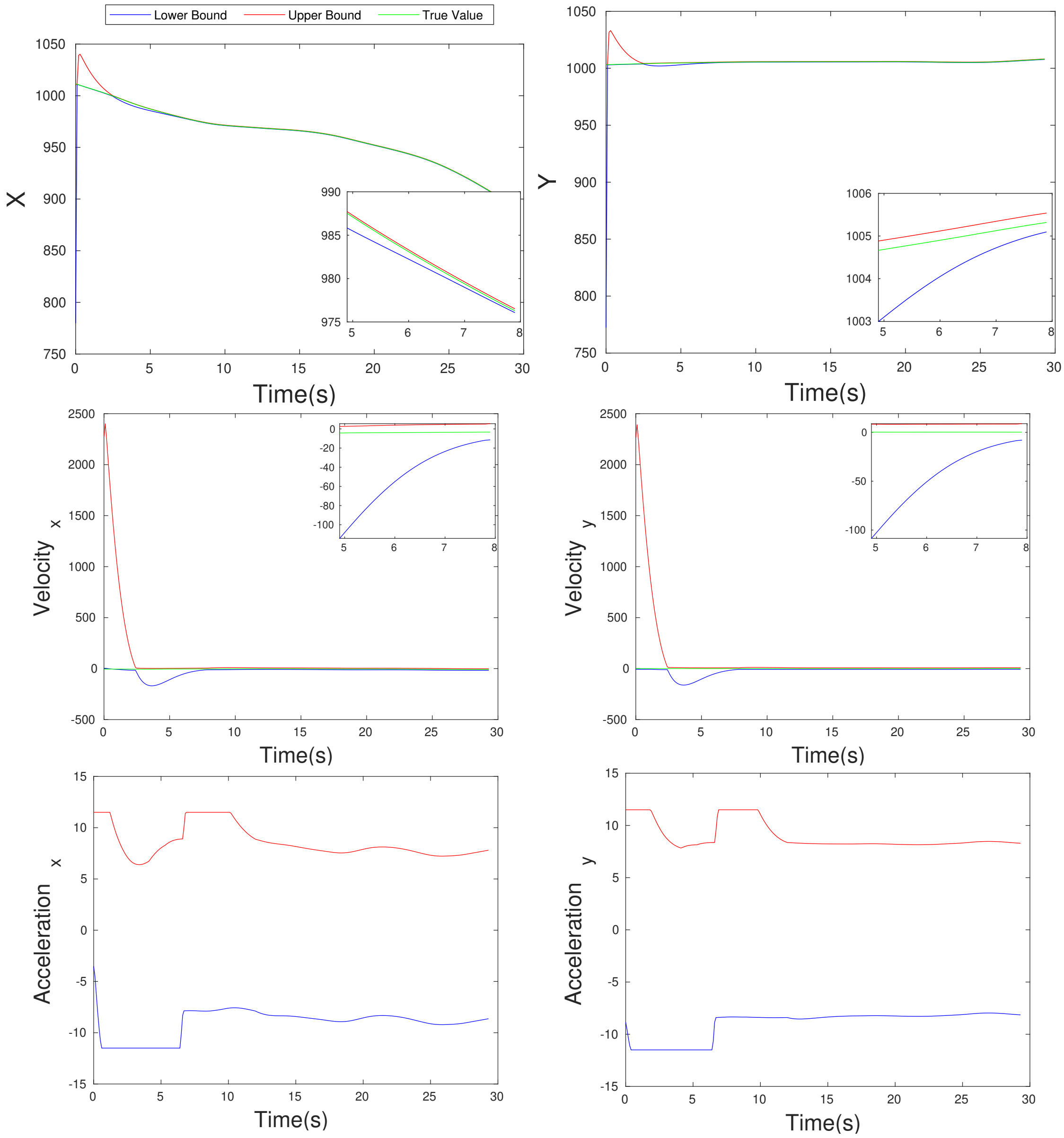

Figure 6.6: Estimation using Point Mass Model

6.1.3 Interval Observer using H-∞

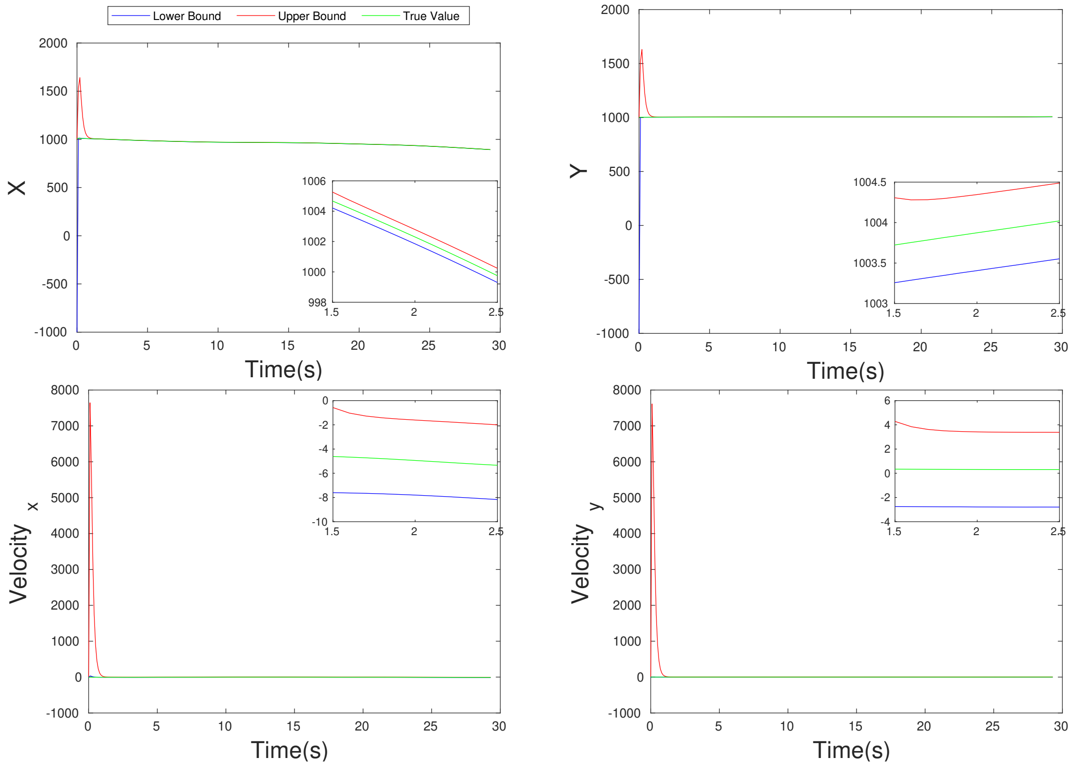

Figure 6.7: Estimation using Constant Velocity

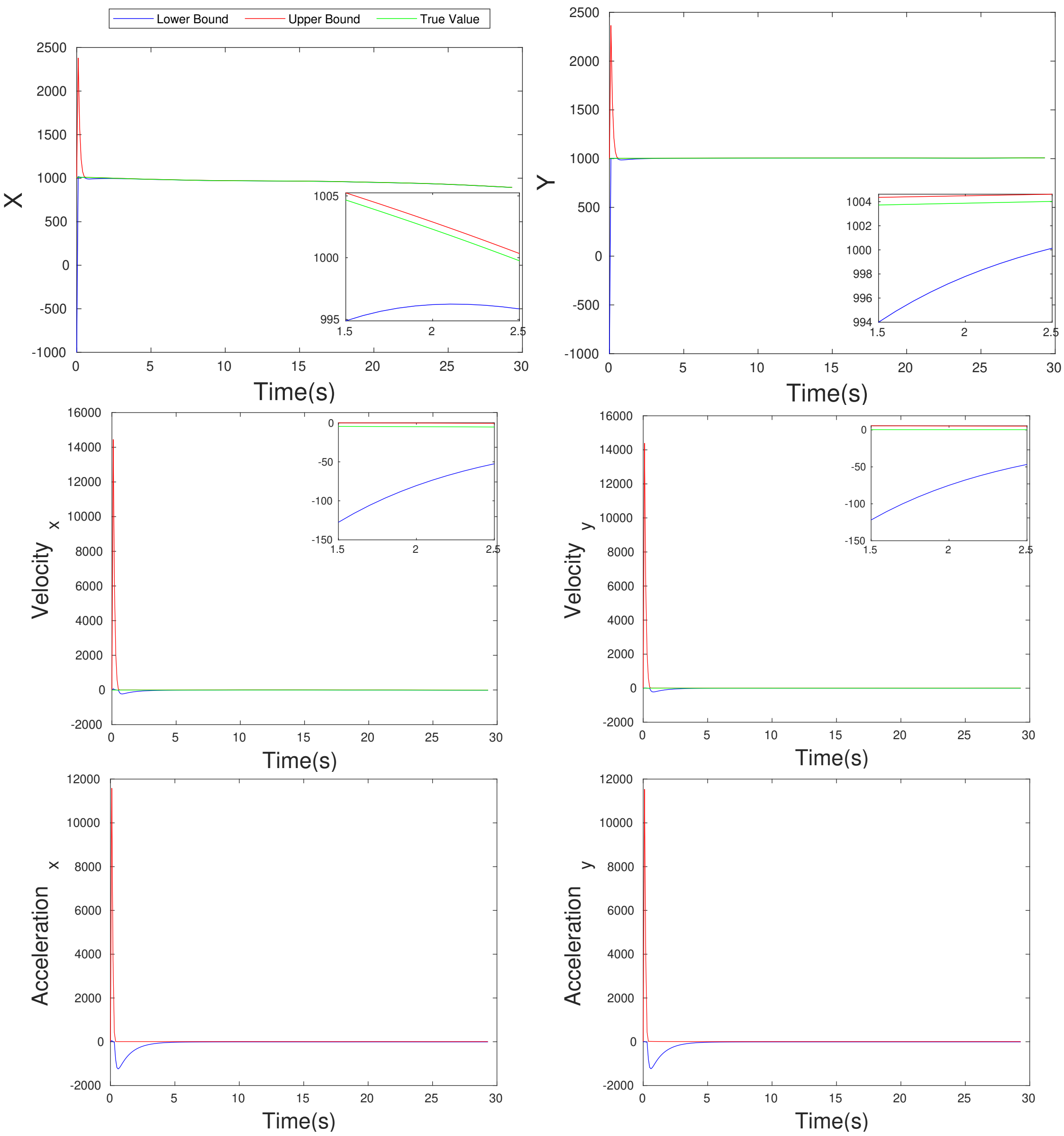

Figure 6.8: Estimation using Constant Acceleration

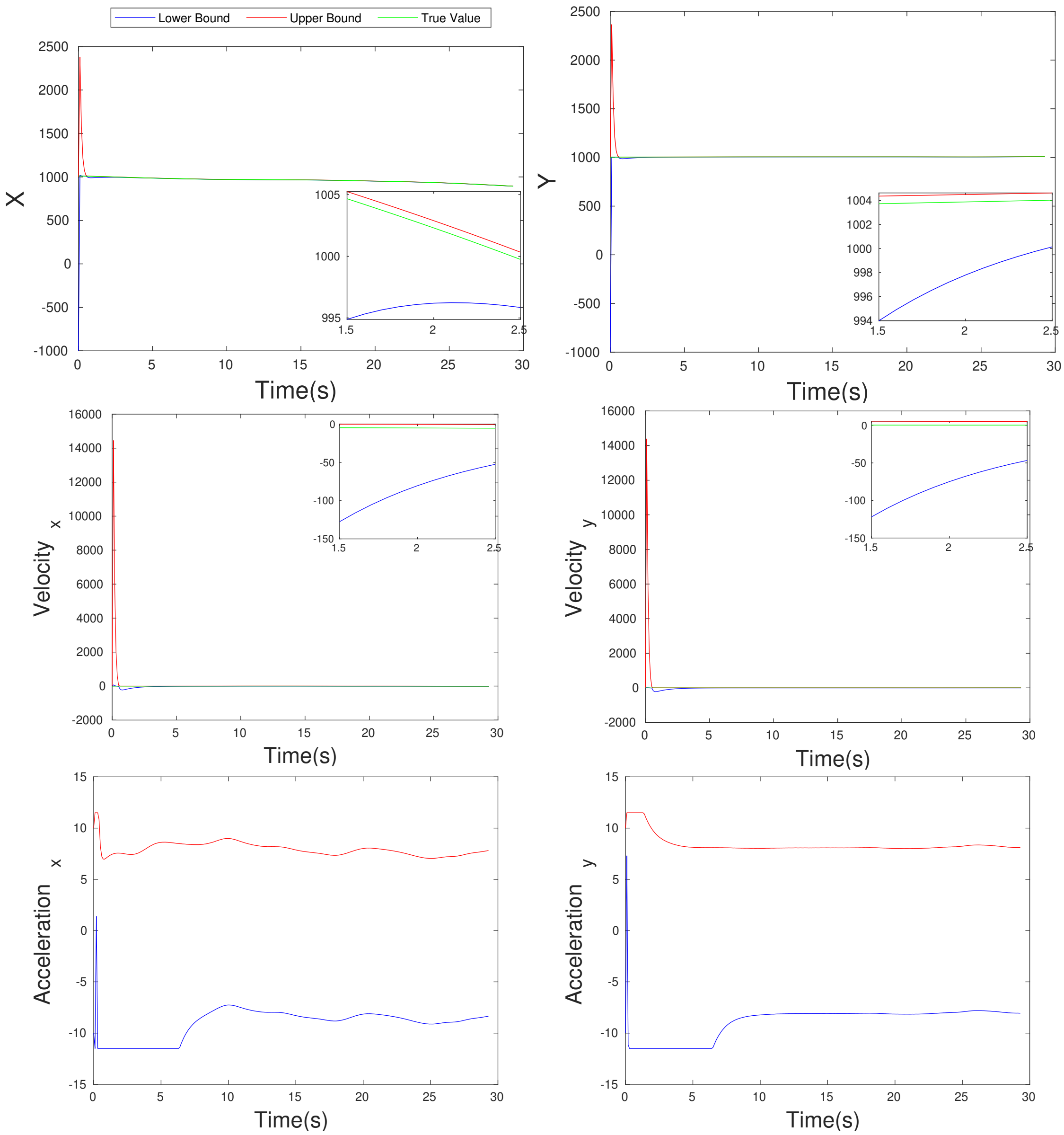

Figure 6.9: Estimation using Point Mass Model

6.2 Rate of Change of Bounds

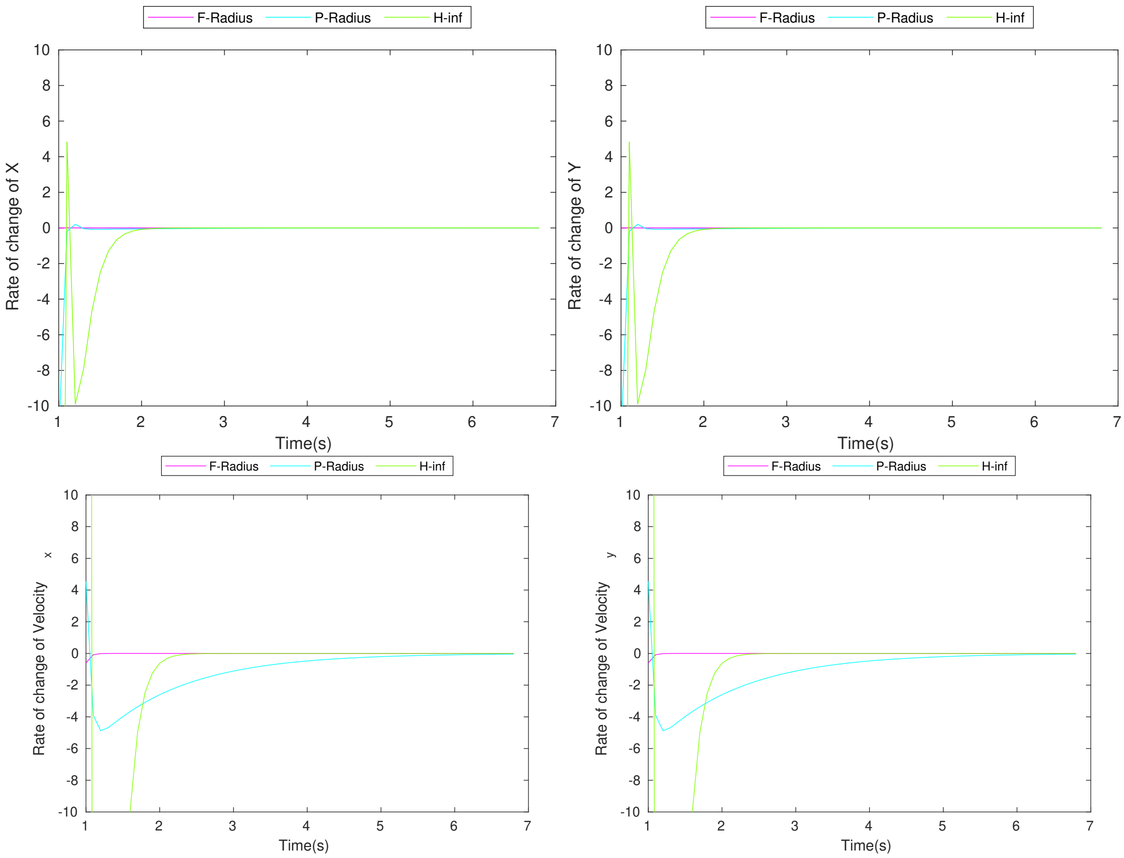

Figure 6.10: Rate of change of bounds using Constant Velocity Model

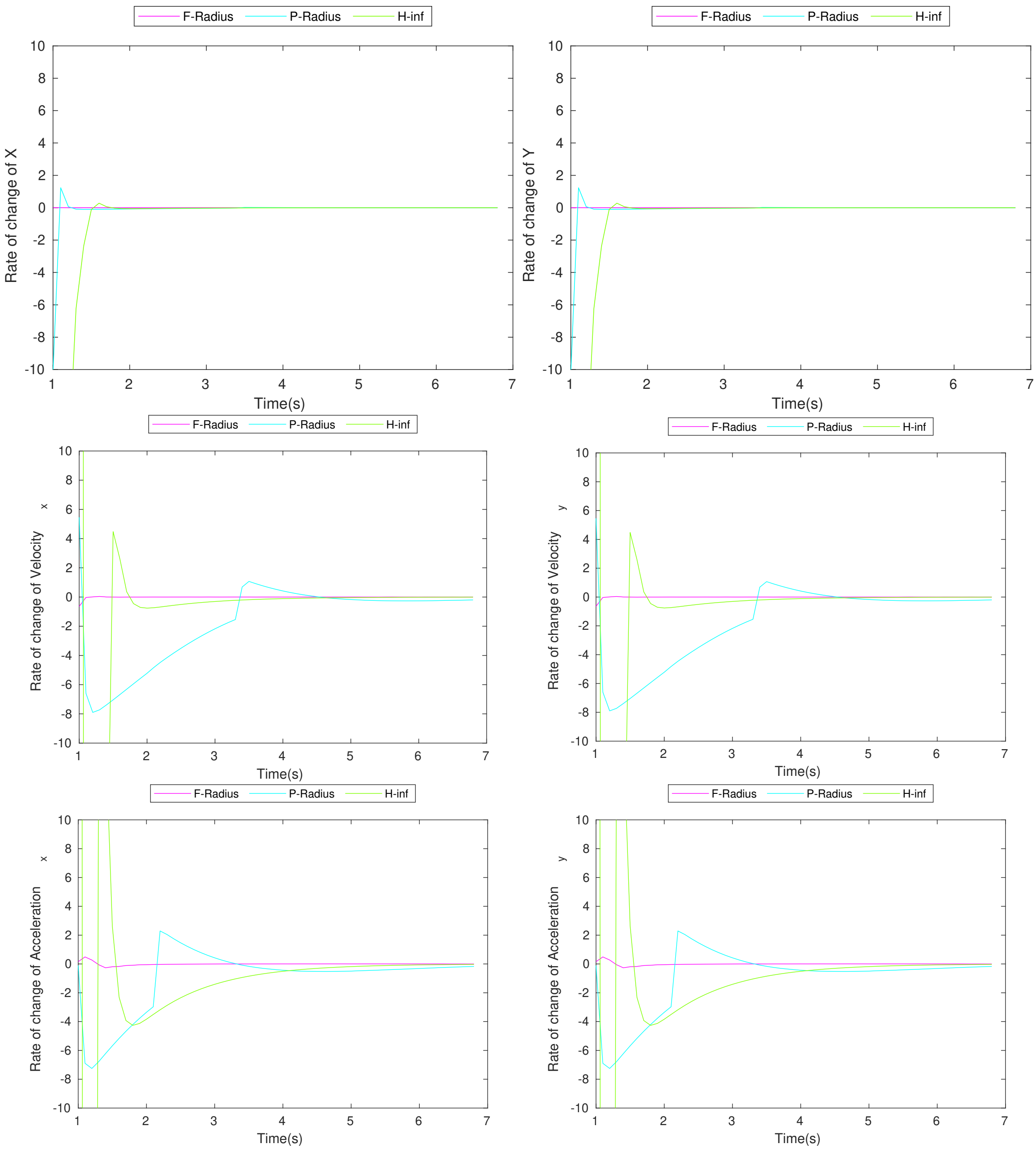

Figure 6.11: Rate of change of bounds using Constant Acceleration Model

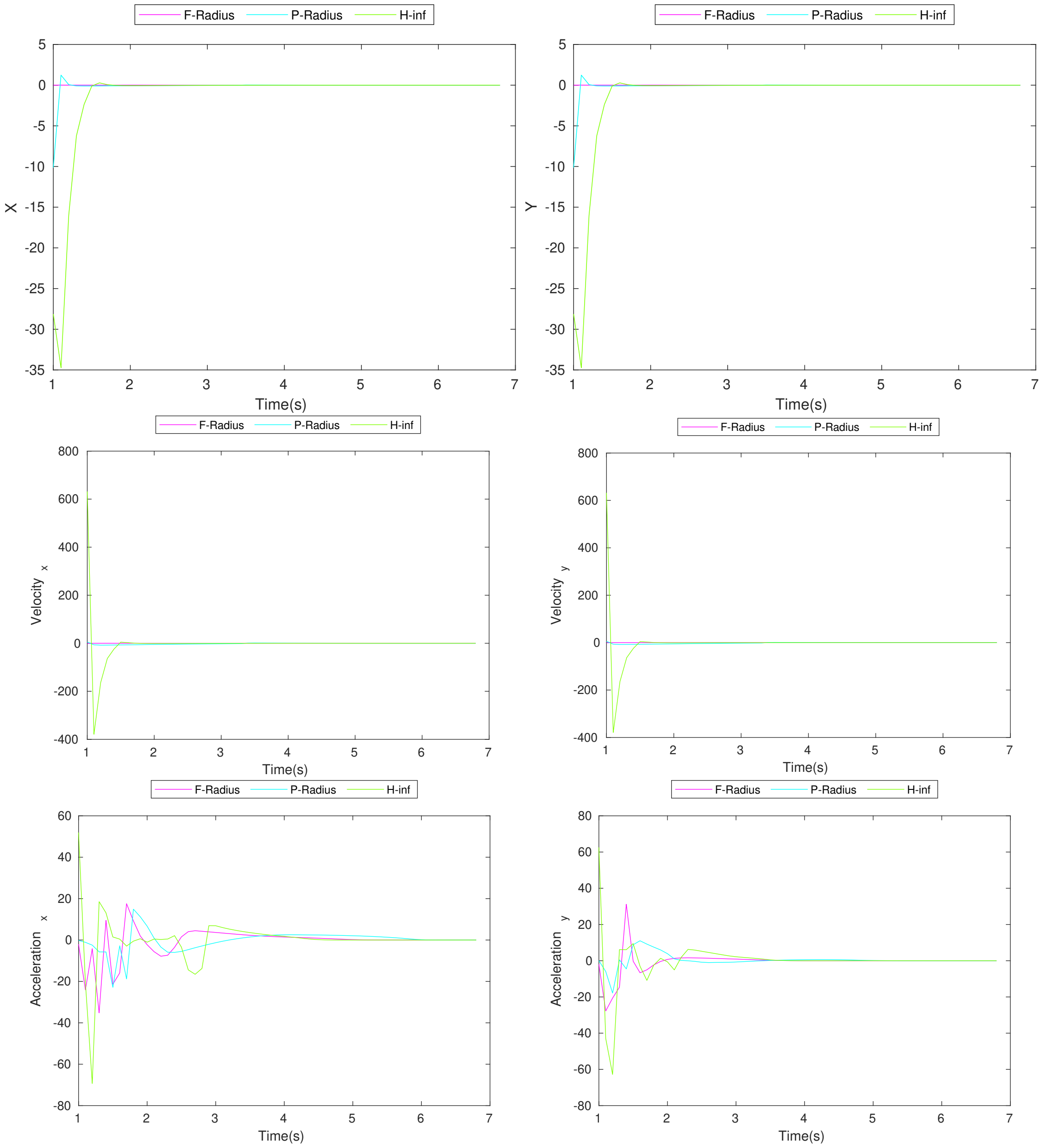

Figure 6.12: Rate of change of bounds using Point Mass Model

List of Figures

2.1 An illustration of a zonotope and its interval hull in 2-D 6

4.1 Computation time for each method to estimate using Constant Velocity Model 15
4.2 RMSE(Root Mean Squared Error) 17

6.1 Estimation using Constant Velocity 21
6.2 Estimation using Constant Acceleration 22
6.3 Estimation using Point Mass Model 23
6.4 Estimation using Constant Velocity 24
6.5 Estimation using Constant Acceleration 25
6.6 Estimation using Point Mass Model 26
6.7 Estimation using Constant Velocity 27
6.8 Estimation using Constant Acceleration 28
6.9 Estimation using Point Mass Model 29
6.10 Rate of change of bounds using Constant Velocity Model 30
6.11 Rate of change of bounds using Constant Acceleration Model 31
6.12 Rate of change of bounds using Point Mass Model 32

List of Tables

4.1 Comparison of computation time (ms) 15
4.2 Comparison of average time(in ms) to converge for unmeasured state . 16
4.3 Comparison of bounds of estimation 16
4.4 Comparison of RMSE 18

Bibliography

[1] S. (2014. “Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.” In: *SAE International* (2014).

[2] M. Hirz and B. Walzel. “Sensor and object recognition technologies for self-driving cars.” In: *Computer-Aided Design and Applications* 15.4 (2018), pp. 501–508. ISSN: 16864360.

[3] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems.” In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. ISSN: 0021-9223. eprint: `https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35\_1.pdf`.

[4] V. Puig, P. Cugueró, and J. Quevedo. “Worst-case state estimation and simulation of uncertain discrete-time systems using zonotopes.” In: *2001 European Control Conference (ECC)* (2001), pp. 1691–1697.

[5] C. Combastel. “A state bounding observer based on zonotopes.” In: *European Control Conference, ECC 2003* (2003), pp. 2589–2594.

[6] T Alamo, J. M. Bravo, and E. F. Camacho. “Guaranteed state estimation by zonotopes.” In: (2005).

[7] V. T. Le, T. Alamo, E. F. Camacho, C. Stoica, and D. Dumur. “Zonotopic set-membership estimation for interval dynamic systems.” In: *Proceedings of the American Control Conference* (2012), pp. 6787–6792. ISSN: 07431619.

[8] F. Mazenc and O. Bernard. “Interval observers for linear time-invariant systems with disturbances.” In: *Automatica* 47.1 (2011), pp. 140–147. ISSN: 0005-1098.

[9] T. Raïssi, D. Efimov, and A. Zolghadri. “Interval State Estimation for a Class of Nonlinear Systems.” In: 57.1 (2012), pp. 260–265.

[10] M. Althoff. “CommonRoad : Vehicle Models.” In: (2016).

[11] J. J. Rath and M. Althoff. “Evaluation of Set-Based Techniques for Guaranteed State Estimation of Linear Disturbed Systems.” In: (2020).

[12] M. Althoff. “ur Steuerungs- und Regelungstechnik Technische Universit ¨ at M ¨ unchen Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars.” In: (2010).

[13] M. Althoff, N. Kochdumper, and C. Arch. "CORA 2018 Manual." In: (2018).

[14] R. Schubert, E. Richter, and G. Wanielik. "Comparison and evaluation of advanced motion models for vehicle tracking." In: *Proceedings of the 11th International Conference on Information Fusion, FUSION 2008* 1 (2008).

[15] V. Puig. "Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies." In: *International Journal of Applied Mathematics and Computer Science* 20.4 (2010), pp. 619–635. ISSN: 1641876X.

[16] X. Ge, Q.-L. Han, X.-M. Zhang, D. Ding, and F. Yang. "Resilient and secure remote monitoring for a class of cyber-physical systems against attacks." In: *Information Sciences* 512 (2020), pp. 1592–1605. ISSN: 0020-0255.

[17] W. Tang, Z. Wang, Y. Wang, and T. Ra. "Interval Estimation Methods for Discrete-Time Linear Time-Invariant Systems." In: *IEEE Transactions on Automatic Control* 64.11 (2019), pp. 4717–4724.