



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:	05.01.2020

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, 05.01.2020

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 unavailability of 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 set-based state estimation using zonotopes as domain representation on existing dataset of real traffic participants. Set-based methods are used to bound the true state of the system to a set, in contrast to stochastic methods which gives an estimation close to the true state. Encapsulating the true state in a set is important for the use-case in autonomous vehicles to forbid tolerance in divergence from the true state. The algorithms are compared in terms of computation time, time to converge, tightness of bound and accuracy.  $H_\infty$  interval observer has performed 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, the segment minimization using F-Radius gives the most accurate estimates.

# Contents

<b>Acknowledgments</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Vehicle Localization : The Guaranteed Estimation Problem</b>	<b>3</b>
2.1 Problem Formulation . . . . .	3
2.2 Vehicle Model . . . . .	3
2.2.1 Constant Velocity Model . . . . .	4
2.2.2 Constant Acceleration Model . . . . .	4
2.2.3 Point Mass Model . . . . .	4
2.3 Zonotopes: The Set Representation . . . . .	5
<b>3 Zonotope-based guaranteed state estimation</b>	<b>6</b>
3.1 Segment Intersection . . . . .	6
3.1.1 Frobenius norm of generators . . . . .	7
3.1.2 Volume . . . . .	7
3.1.3 P-radius . . . . .	7
3.2 Interval Observer . . . . .	7
3.2.1 H- $\infty$ Observer . . . . .	8
<b>4 Evaluations</b>	<b>9</b>
4.1 Computation Time . . . . .	9
4.2 Time to Converge . . . . .	10
4.3 Bounds . . . . .	11
4.4 Accuracy . . . . .	11
<b>5 Conclusion</b>	<b>14</b>
<b>6 Extended Results</b>	<b>15</b>
6.1 Set Estimation . . . . .	15
6.1.1 Segment Minimization using F-Radius . . . . .	15
6.1.2 Segment Minimization using P-Radius . . . . .	19

*Contents*

---

6.1.3 Interval Observer using $H-\infty$ . . . . .	22
6.2 Rate of Change of Bounds . . . . .	25
<b>List of Figures</b>	<b>28</b>
<b>List of Tables</b>	<b>29</b>
<b>Bibliography</b>	<b>30</b>

# 1 Introduction

There is a steep progress in research and development of autonomous vehicles. The race to the top of automobile industry, participated by companies like BMW, Tesla, Waymo/Google, requires fast development and vigorous testing of novel technology. One of the many challenges of this field is to ensure collision avoidance. With no human behind wheels for Level 5 [1] cars, the vehicle must keep track of roads, surrounding traffic participants, like vehicles and pedestrians, in different circumstances including rain and fog, to ensure safety of its passengers. Current collision avoidance systems based on sensors, radar and camera will be overwhelmed with high computation demands 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 <sup>1</sup>.

The Collision Avoidance System in car system consists of two parts: Sensing and Tracking, and Maneuver. The sensing and tracking part is done by sensors like radar, camera and GPS (Global Positioning System). With advancement in technologies in image processing, image analysis and object detection and the decline in the cost of camera sensors, the sensing and tracking is developing fast. Although cameras can classify vehicles, it cannot guarantee measurement in low-light environment (e.g. night) [2]. On the other hand, radar guarantee robustness to weather in exchange of high cost. Similarly, GPS has disturbances too which favours methods using combination of sensors to cover each others drawbacks to provide vehicle localization. After getting the data, the maneuver is carried out in a way to avoid collision with the location found from the sensors. The probable location of the tracked vehicle, is thus also important to calculate a predicted trajectory. However, all the data to predict the vehicle's location is not measurable using just sensors. Furthermore, the sensor data are not 100% accurate, and hence solely cannot be relied on carry on maneuver to avoid collision.

Due to the lack of quality and availability of sensors, researchers have used state estimation algorithms to determine the state of the tracked vehicle. One of the widely applied technique is the Kalman Filter, that requires a probability distribution of perturbation in the measurements.

---

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



On the other hand, there exists set-based state estimation which provides a set of possible states of the system, instead of a point-wise estimation like Kalman filter. A high degree of accuracy and guarantee is 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 drone and fixed camera. A similar comparison can be found in [3] on simulated data.

Comparing different domain representation of the sets enclosing the possible state of the system, we chose zonotopes as opposed to ellipsoid and polytopes 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 in time due to accumulated noises) and Minkowski sum(i.e. sum of zonotopes is also a zonotope). We used CORA in Matlab for 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. Although there are complex models that can be used to represent a vehicle state [4], not all can be used due to unavailability of measurements like wheelbase, velocity, etc. as it is unlikely to be acquired in run-time from tracked vehicle. Hence, the models used in this paper to compare are the simplest, yet complete enough to determine the properties of tracked vehicle for trajectory prediction : Constant Velocity, Constant Acceleration and the Point Mass Model.

The paper is organized as follows. Chapter 2 presents the vehicle localization problem suitable 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 on 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 Problem Formulation

Let us denote the state of the vehicle to be tracked at time  $k$  as  $x_k$  and the measured state as  $y_k$ . The equations to predict  $x_k$  from a previous step  $x_{k-1}$  and the mapping from measurement,  $y_k$  to state,  $x_k$  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-direction. Different states can be estimated using different models, whereas the measured state of the vehicle is assumed to be position in x and y-direction for all models.

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

Given a model ((2.1)), the problem of state estimation is to compute an outer bound of the state( $x_k$ ) consisting the possible values of the true possible state of the system. The dimension of the state( $x_k$ ), the state transition matrix( $A$ ) and measurement matrix ( $C$ ) differ across different models as discussed in the next section.

### 2.2 Vehicle Model

Three linear systems are implemented 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 since no precise model of tracked vehicles are available. 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, we chose to investigate the following models:

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

### 2.2.1 Constant Velocity Model

The vehicle is assumed to travel in constant velocity [5]. The state of the system ( $x_k$ ), state transition matrix ( $A$ ), and the measurement matrix( $C$ ) is shown in (2.2.1).

$$x = [s_x \quad s_y \quad v_x \quad v_y]^T$$

$$A = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

### 2.2.2 Constant Acceleration Model

Although the Constant velocity Model is easy to implement, however it is dangerous to assume constant velocity. Acceleration model takes care of changing velocity and assumes constant acceleration [5]. The state of the system ( $x_k$ ), state transition matrix ( $A$ ), and the measurement matrix( $C$ ) is shown in (2.2.2).

$$x = [s_x \quad s_y \quad v_x \quad v_y \quad a_x \quad a_y]^T$$

$$A = \begin{bmatrix} 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 \\ 0 & 0 & 1 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

### 2.2.3 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[4] which is similar to the

constant acceleration model, except that the acceleration can change upto 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 matrix and measurement matrix are 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.

### 2.3 Zonotopes: The Set Representation

Zonotopes are gaining fame in the set-based state estimation techniques [6], due to its control of wrapping effect and that the Minkowski sum of the zonotope results in zonotopes.

Zonotopes are represented by a center, denoted by  $p$  and generators, denoted by  $H$ . An m-zonotope in  $\mathbb{R}^n$  can be defined as an affine transformation by  $H$  of an m-dimensional hypercube in  $\mathbb{R}^n$  centered at  $p$ . Minkowski sum, zonotope reduction and convex hull of zonotopes are required to be computed in the state estimation algorithms. The toolbox, CORA (COntinuous Reachability Analyzer) [7], is used to construct zonotopes and apply the computation for zonotopes in the algorithms.

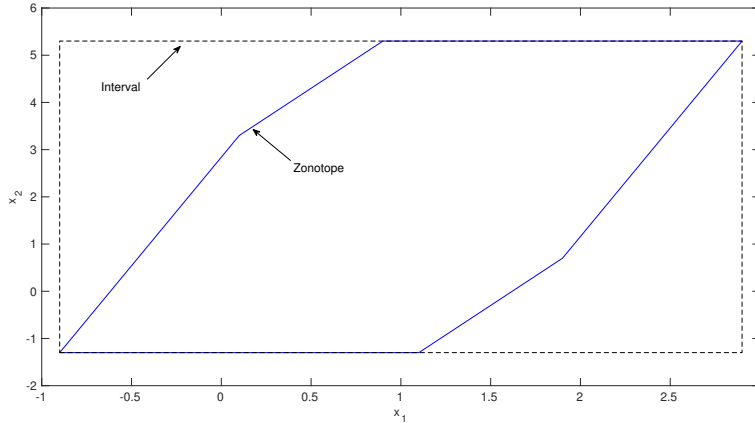


Figure 2.1: An illustration of a zonotope and its interval hull in 2-D

## 3 Zonotope-based guaranteed state estimation

State Estimation algorithms can be broadly 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 [8] proposed a family of Kalman filters 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 ellipsoid 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, zonotopes can control the wrapping effect [9], which is the term referred to the growth of the estimated state due to the propagated uncertainties in each iteration. In addition, sum of zonotopes is also a zonotope (Minkowski sum), which is a desirable property for the techniques.

Set-based algorithms 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, like volume and radius, of the geometric set. The interval observer methods, on the other hand, design observer to minimize the error on each time step. The following sections dig deeper on each of 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 F-radius, P-radius and volume, are considered to represent the size of the segment. The following sections list and elaborate the algorithm that focus and optimize different properties of zonotopes to minimize the segment.

### 3.1.1 Frobenius norm of generators

Frobenius norm of the generators of a zonotope is calculated using the formula (3.1) [10].

$$\|H\|_F^2 = \|A + \lambda b^T\|_F^2 \quad (3.1)$$

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

The  $\lambda$  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) [10].

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

### 3.1.3 P-radius

The P-radius of a zonotope can be calculated by the formula (3.4) where  $P$  is a positive definite matrix [10].

$$\max_{z \in Z} (\|z - p\|_P^2) = \max_{z \in Z} ((z - p)^T P (z - p)) \quad (3.4)$$

## 3.2 Interval Observer

Interval Observers need to design observers to minimize the error in the estimation. For the system in (2.1), (3.5) defines the observer, where  $L$  is the observer gain to be designed. Although, accuracy is highly expected from these methods, the design of such observers are not very easy. The following section discusses about a method, which uses H- $\infty$  observer design.

$$x_{k+1} = Ax_k + L(y_k - Cx_k) \quad (3.5)$$

### 3.2.1 H- $\infty$ Observer

The interval observer, proposed in [11], 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 (3.6).

$$\min_{\gamma_2} \text{ s.t. (3.7)} \quad (3.6)$$

$$\begin{bmatrix} I_{n_x} - P & * & * & * \\ 0 & -\gamma^2 I_{n_w} & * & * \\ 0 & 0 & -\gamma^2 I_{n_v} & * \\ PA - YC & PE & -YF & -P \end{bmatrix} \prec 0 \quad (3.7)$$

With  $L$  derived using a mosek solver in Matlab, the estimated state of the system is found by Equation (3.5).

## 4 Evaluations

The INTERACTION Dataset <sup>1</sup> 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 camera over 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 is used to calculate the error and evaluate the estimates. The initial state of the system is set using assignments (4.1).

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

### 4.1 Computation Time

Table 4.1: Comparison of computation time (ms) using Constant Velocity Model

Method	Average Computation Time (ms)
F-Radius	0.375
P-Radius	0.319
H- $\infty$ approximation	0.147

Figure 4.1 shows that computation time for Volume Minimization rises exponentially making it futile in the state estimation for collision avoidance system. On the contrary, Table 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- $\infty$  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.

<sup>1</sup><https://interaction-dataset.com/>



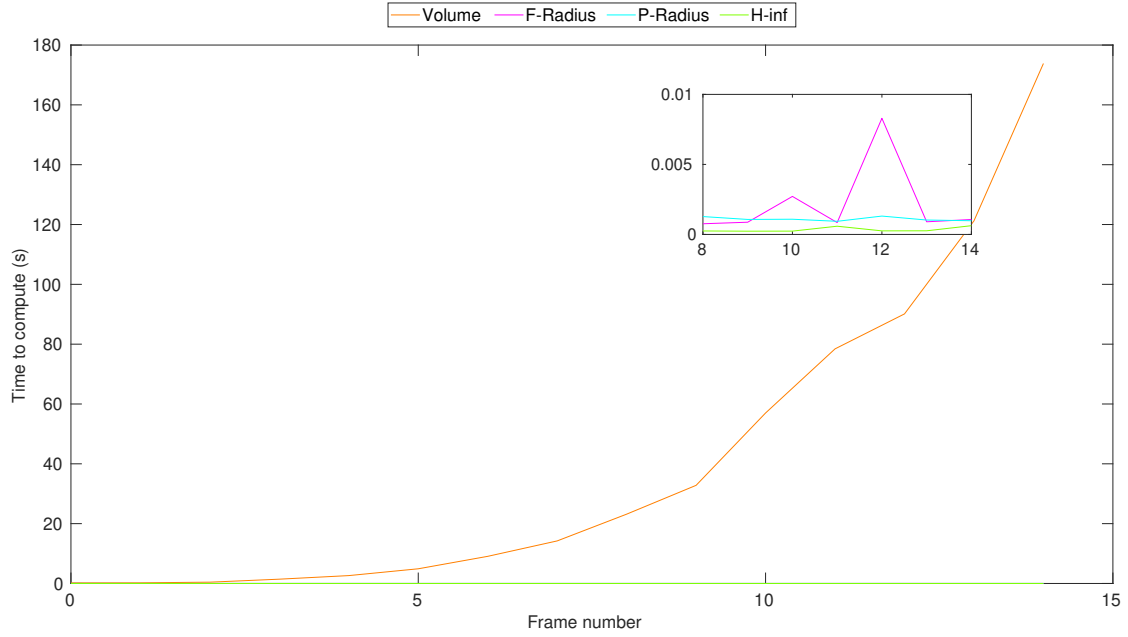


Figure 4.1: Computation time for each method to estimate using Constant Velocity Model

## 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- $\infty$ approximation	24	50	35

Table 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, same bounds are obtained from constant acceleration and point mass model, however, the bounds of acceleration from the latter is better than the former. Hence, point mass model is used to compare the algorithms for estimating acceleration on the following

sections.

### 4.3 Bounds

Table 4.3: Comparison of bounds of estimation

Method	Constant Velocity					
	$s_x$	$s_y$	$v_x$	$v_y$	$a_x$	$a_y$
F-Radius	.441	.441	5.686	5.686	-	-
P-Radius	.4606	.459	13.45	13.45	-	-
H- $\infty$ approximation	.9867	.937	6.177	6.177	-	-
Method	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- $\infty$ 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 Table 4.3. The segment intersection using F-Radius has the tighter bounds compared to the other techniques. Interesting to note, the H- $\infty$  has much higher bounds in the initial time steps.

### 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 measurement for acceleration, the accuracy of acceleration cannot be evaluated.

The initial estimation before convergence gives extreme error which affects the result, hence the estimation 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.

The boxplot of RMSE using measurements in velocity in x direction using Point Mass Model for all the techniques are shown in Figure 4.2. The segment minimization using P-Radius has high range of extremes and the mean error is also greater than the other methods. The range of error is the lowest for H- $\infty$  observer with mean 1.9048 and standard deviation of 1.9776 for  $velocity_x$ . Detailed report of mean and standard deviation of each of the method can be found in table 4.4. The error expectation from

segment minimization from F-Radius is similar to  $H-\infty$  observer, however, the error in measured state is slightly lesser in F-Radius compared to  $H-\infty$  observer.

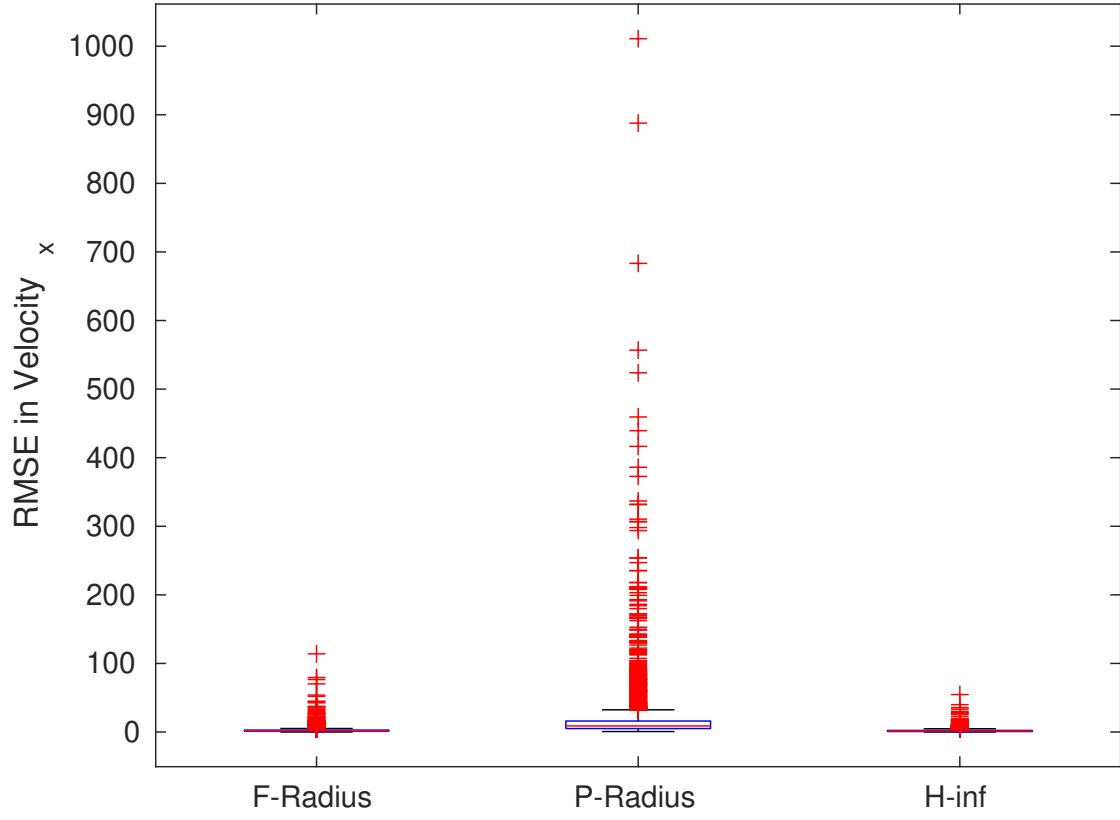


Figure 4.2: RMSE(Root Mean Squared Error)

Table 4.4: Comparison of RMSE

<b>Method</b>	<b>Mean <math>\pm</math> SD</b>			
	$s_x$	$s_y$	$v_x$	$v_y$
F-Radius	$0.0007 \pm 0.0004$	$0.0004 \pm 0.0003$	$2.3412 \pm 2.7092$	$2.4184 \pm 1.5641$
P-Radius	$0.0016 \pm 0.0020$	$0.0008 \pm 0.0017$	$13.4470 \pm 26.2465$	$13.7642 \pm 54.6724$
H- $\infty$ approximation	$0.0007 \pm 0.0004$	$0.0006 \pm 0.0005$	$1.9048 \pm 1.9776$	$2.1230 \pm 2.1946$

---

## 5 Conclusion

A demand for intelligent collision avoidance system is timeless. To take load off sensors and hardware of a vehicle, state estimation algorithms can be used to track vehicle 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 minimization using F-radius gives a good trade-off between fast convergence, accuracy and tight bound. H- $\infty$  and segment minimization using P-radius are ahead in terms of computation time because these methods carry out computation off-line before any input measurement, and hence as a consequence over-approximates and does not improvise estimation significantly for each measurement. Choice of model to represent the state of the system also has significant effect on the performance. To estimate velocity, constant velocity model gives better results, whereas for acceleration, the point mass model gives better estimate compared to the constant acceleration model. This paper can be a starting point to implement higher defined models of tracked vehicle and compare performance of state estimation methods. The state estimation methods can further be evaluated on implementing with distinguishable set of initial starting state to determine the effect of initial estimation on the algorithms, if any. Further developments can include implementing the technique in real vehicle system and using the estimation to track vehicles.

## 6 Extended Results

### 6.1 Set Estimation

Estimation using the techniques with different models are illustrated using data for one 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

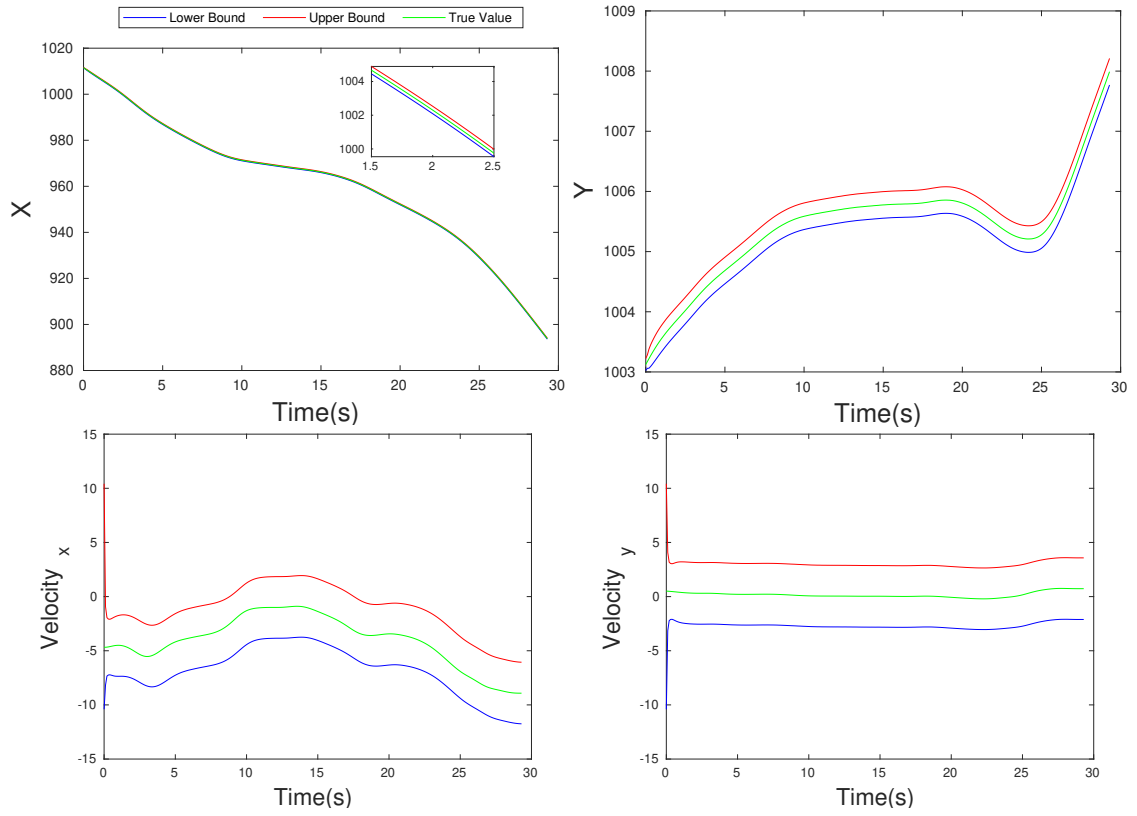


Figure 6.1: Estimation using Constant Velocity

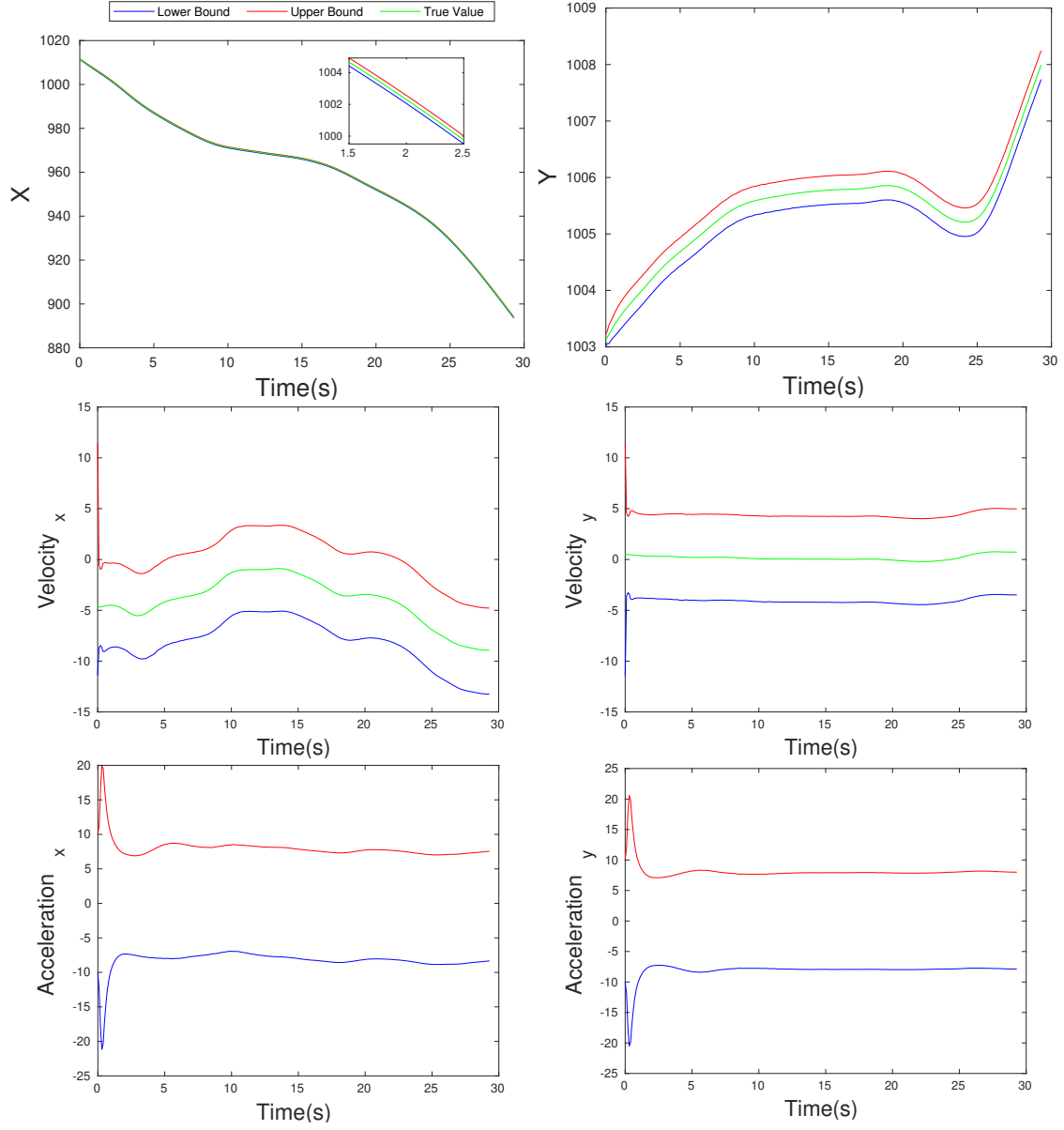


Figure 6.2: Estimation using Constant Acceleration



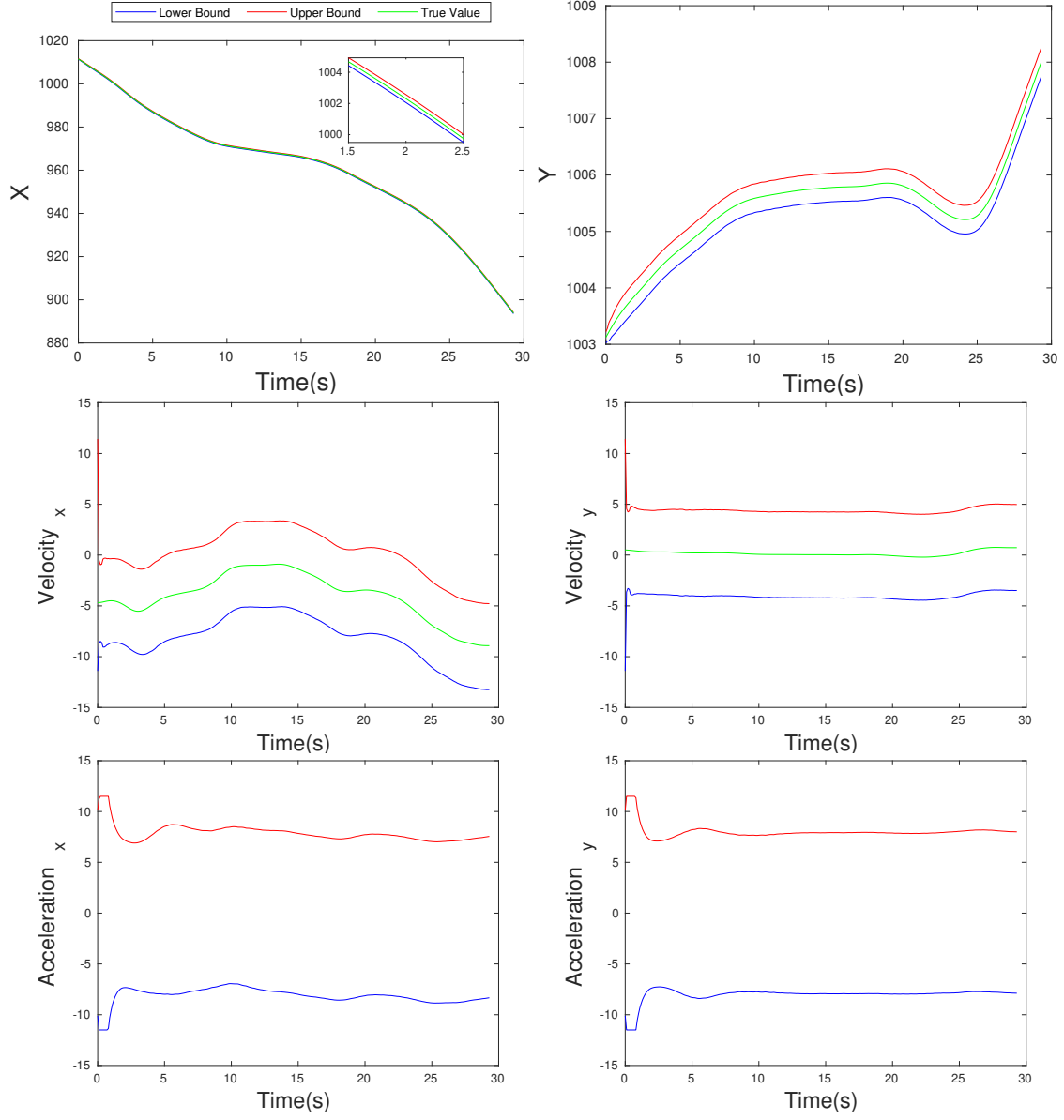


Figure 6.3: Estimation using Point Mass Model

### 6.1.2 Segment Minimization using P-Radius

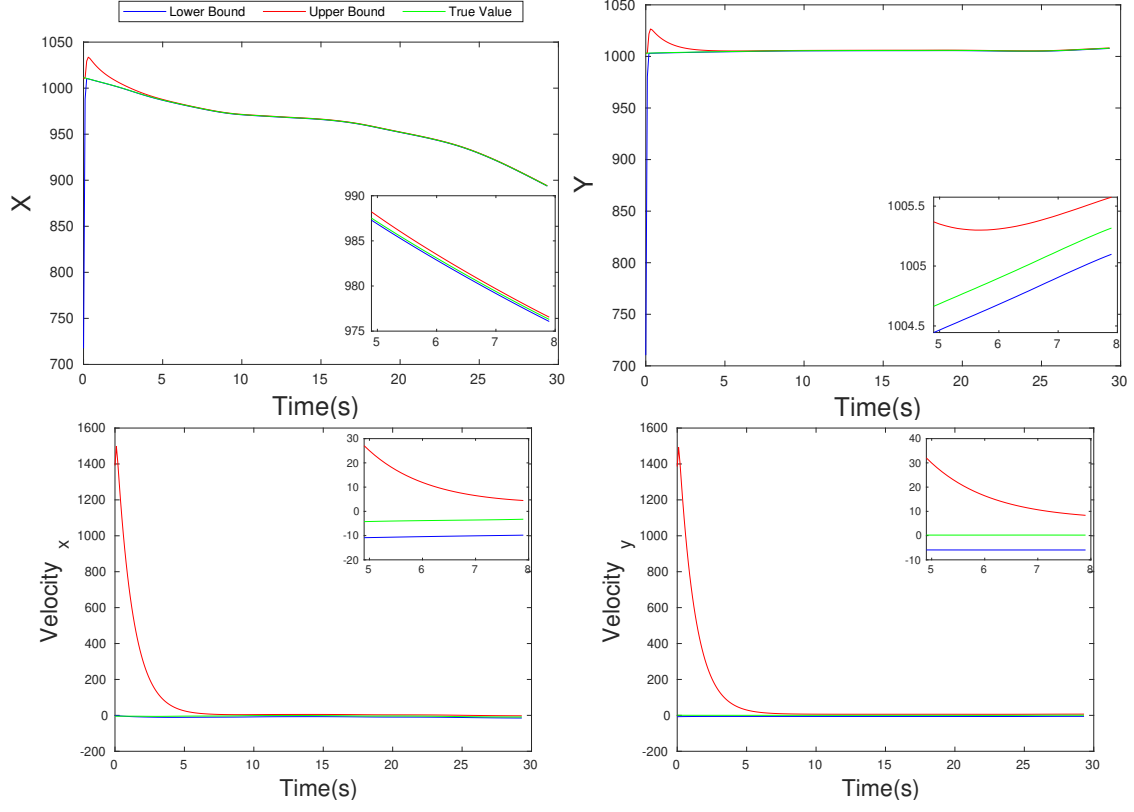


Figure 6.4: Estimation using Constant Velocity

## 6 Extended Results

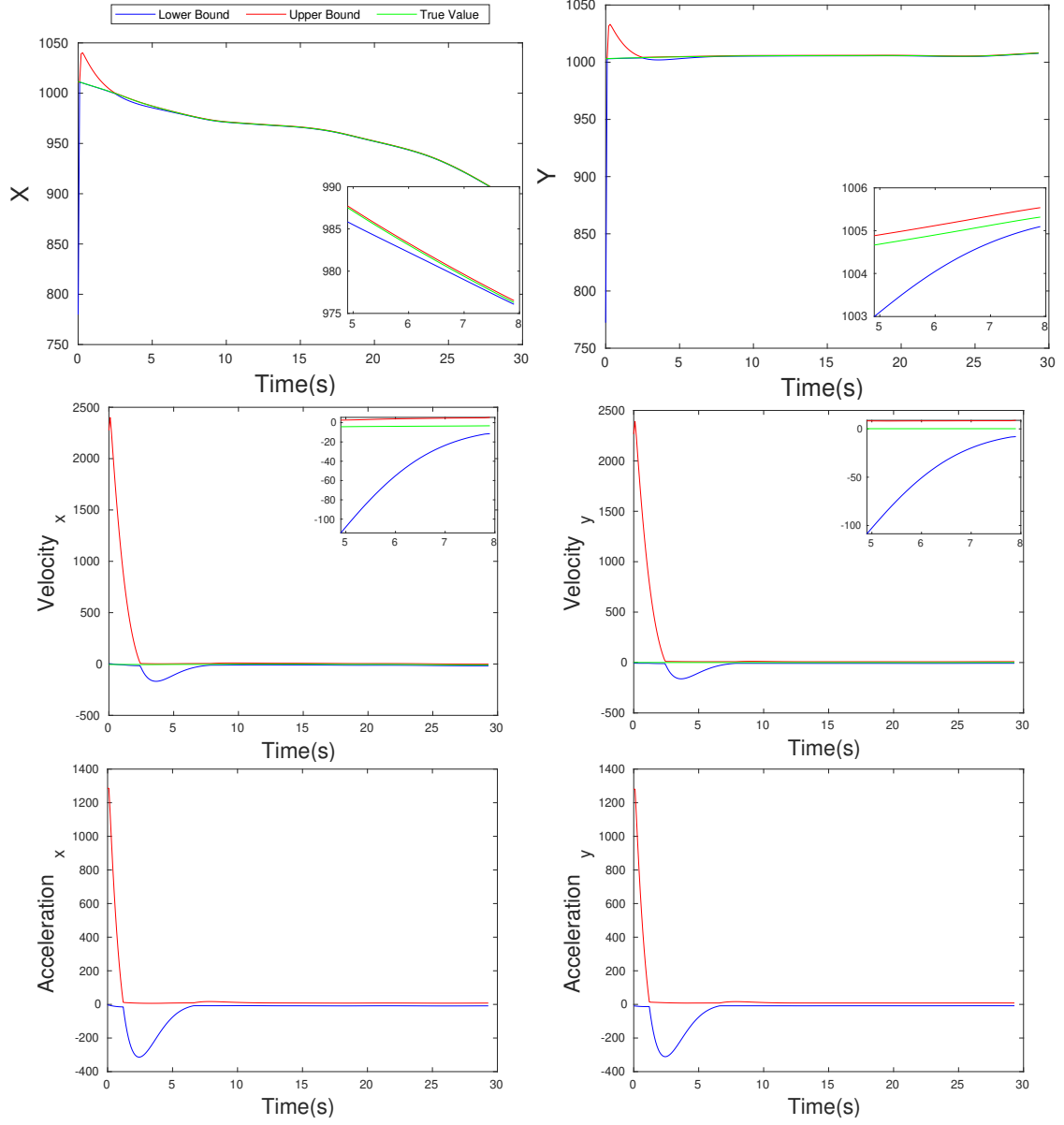


Figure 6.5: Estimation using Constant Acceleration

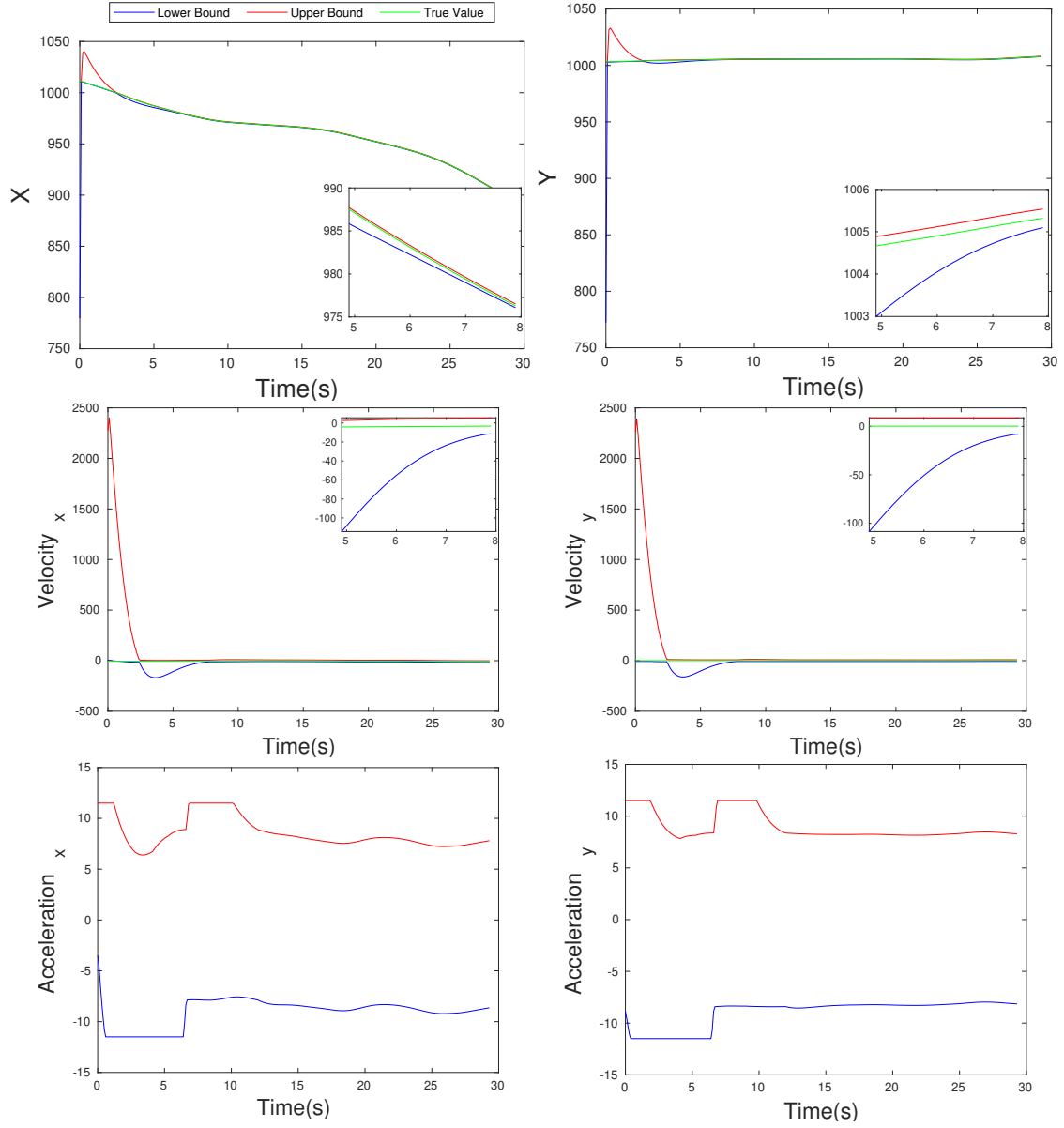


Figure 6.6: Estimation using Point Mass Model

### 6.1.3 Interval Observer using $H_\infty$

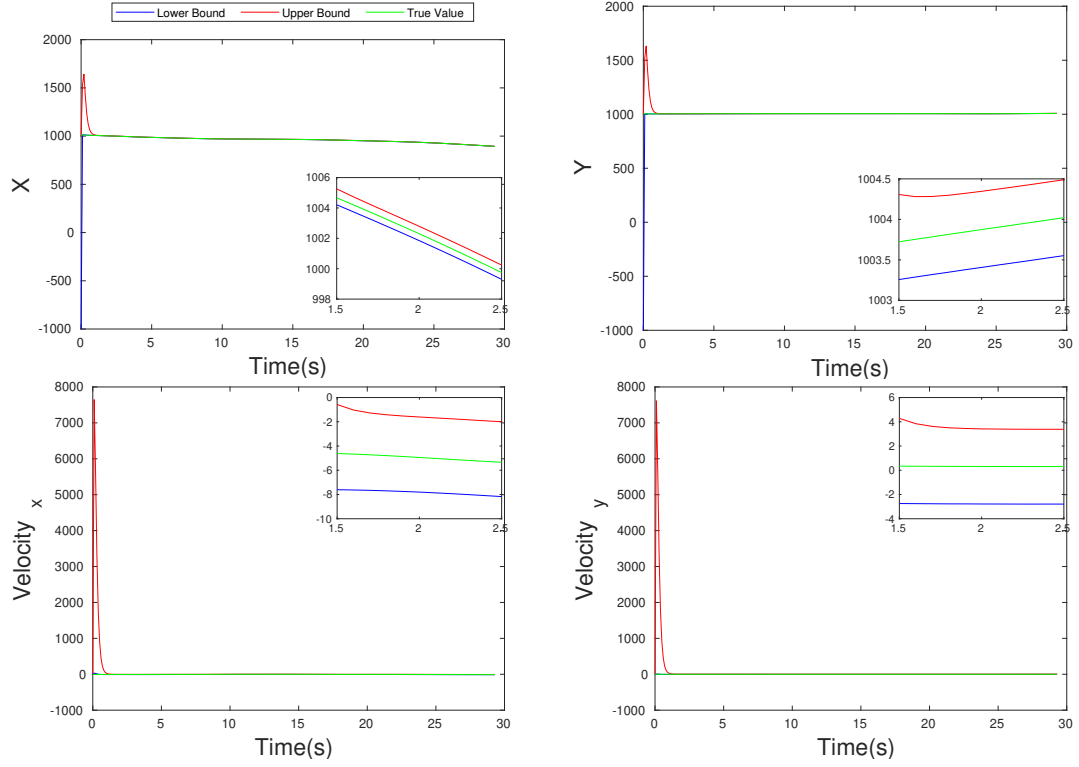


Figure 6.7: Estimation using Constant Velocity

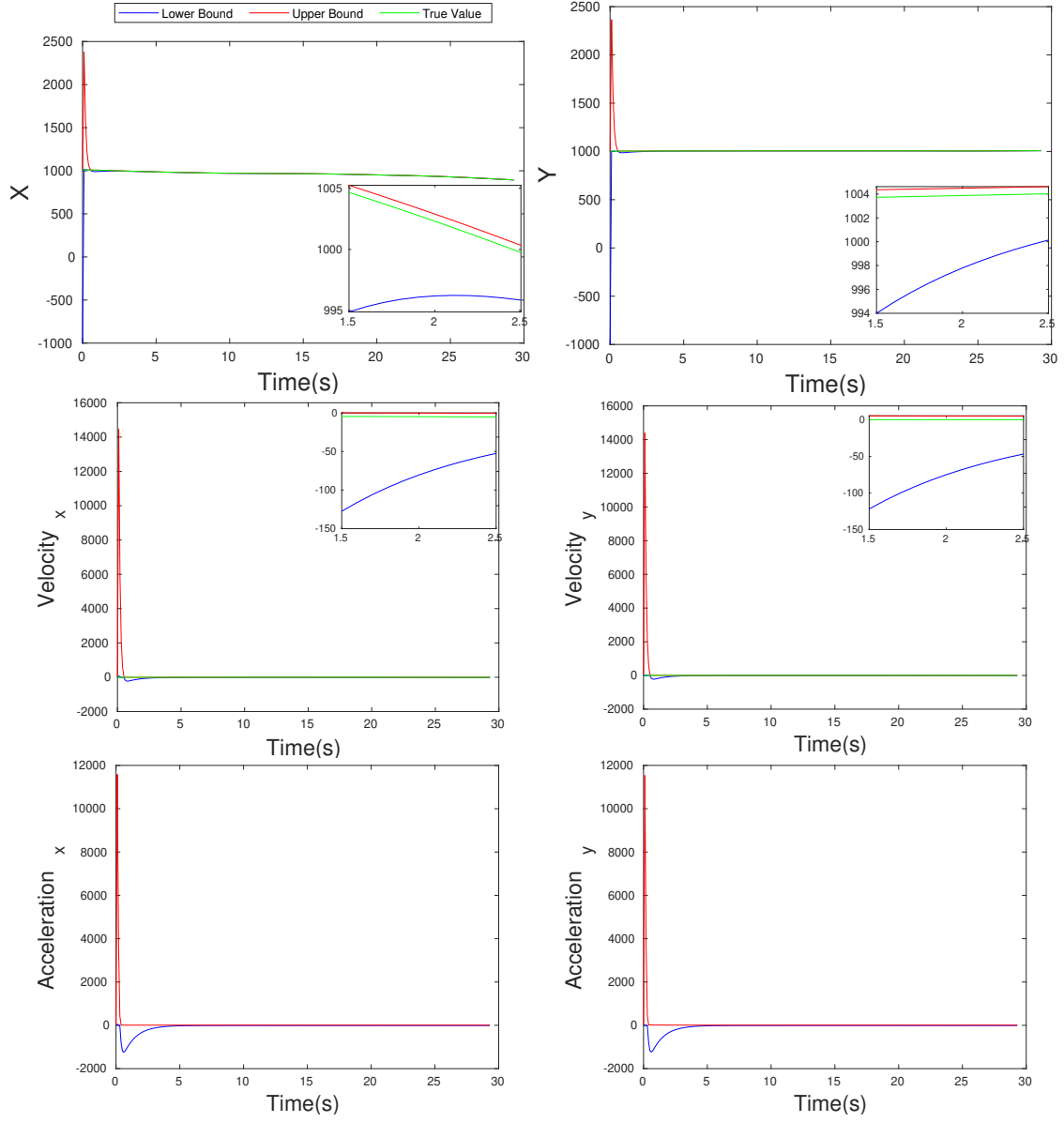


Figure 6.8: Estimation using Constant Acceleration

## 6 Extended Results

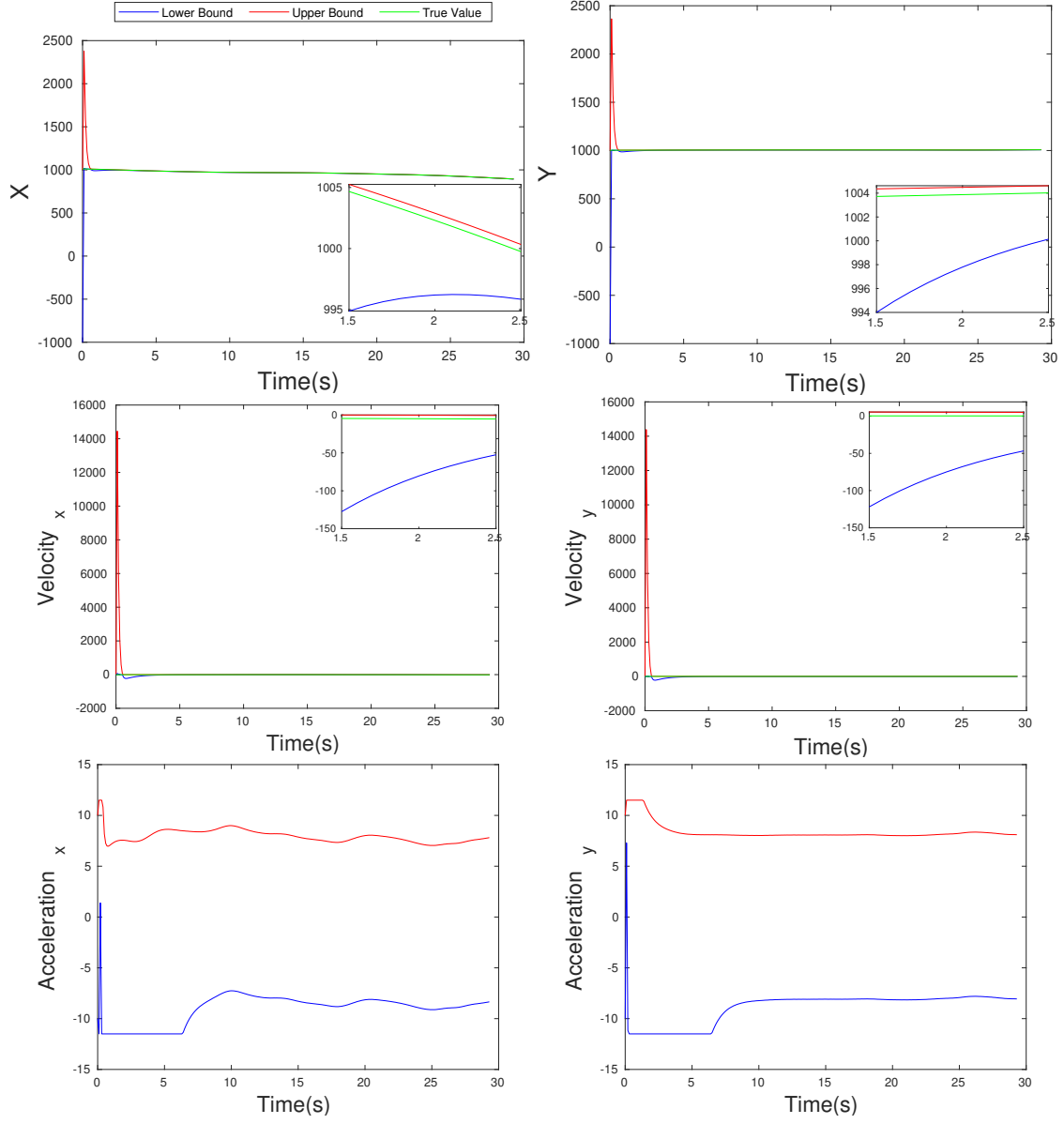


Figure 6.9: Estimation using Point Mass Model

## 6.2 Rate of Change of Bounds

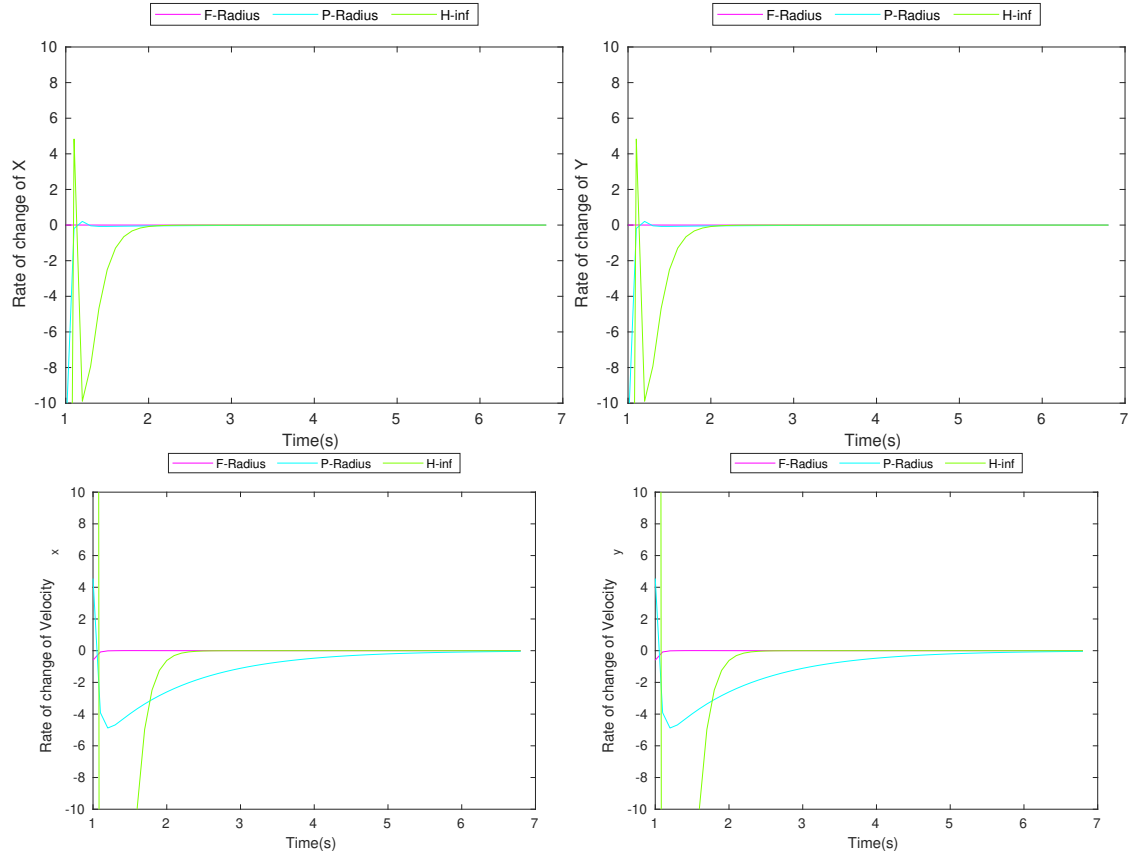


Figure 6.10: Rate of change of bounds using Constant Velocity Model



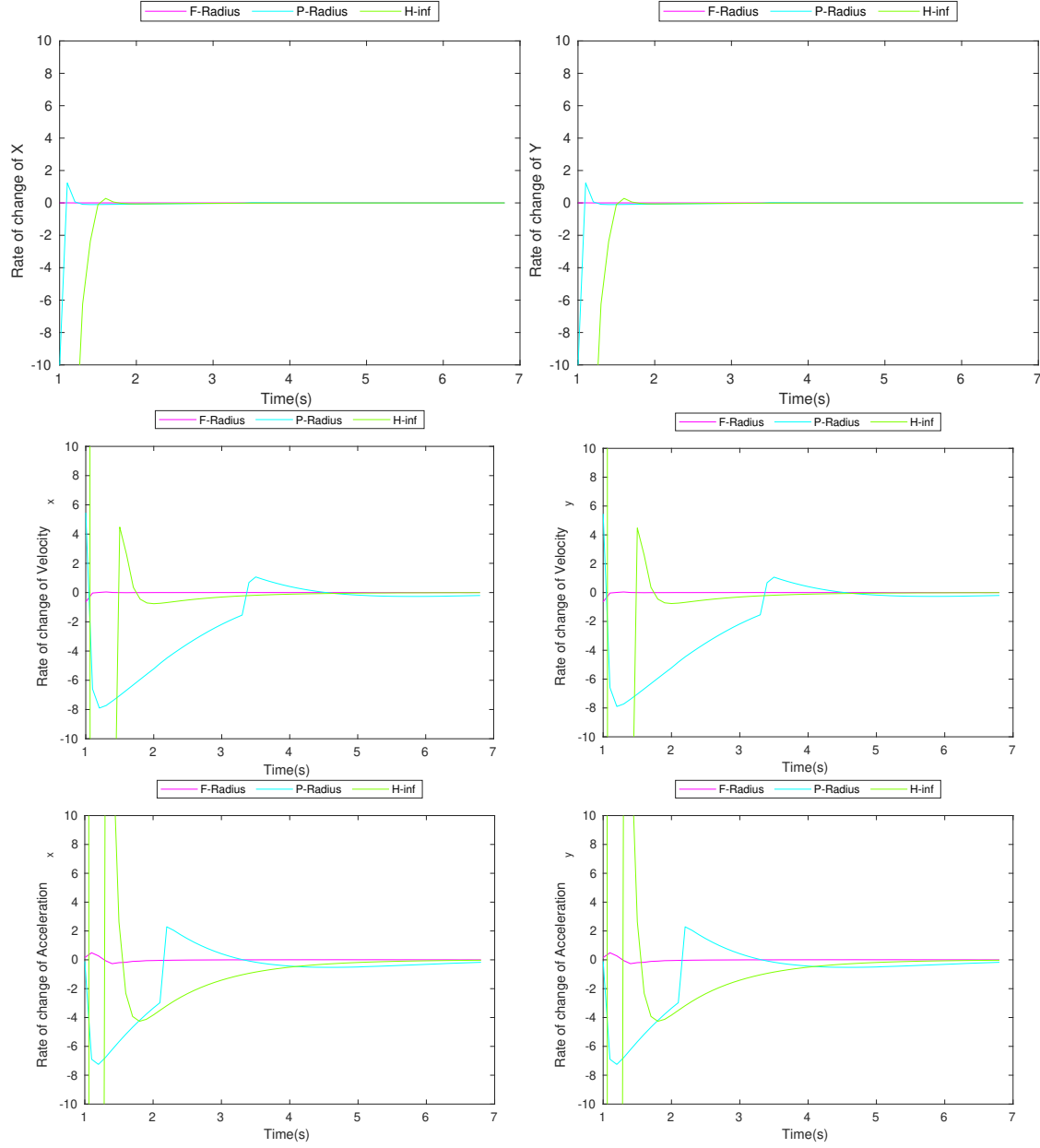


Figure 6.11: Rate of change of bounds using Constant Acceleration Model

## 6 Extended Results

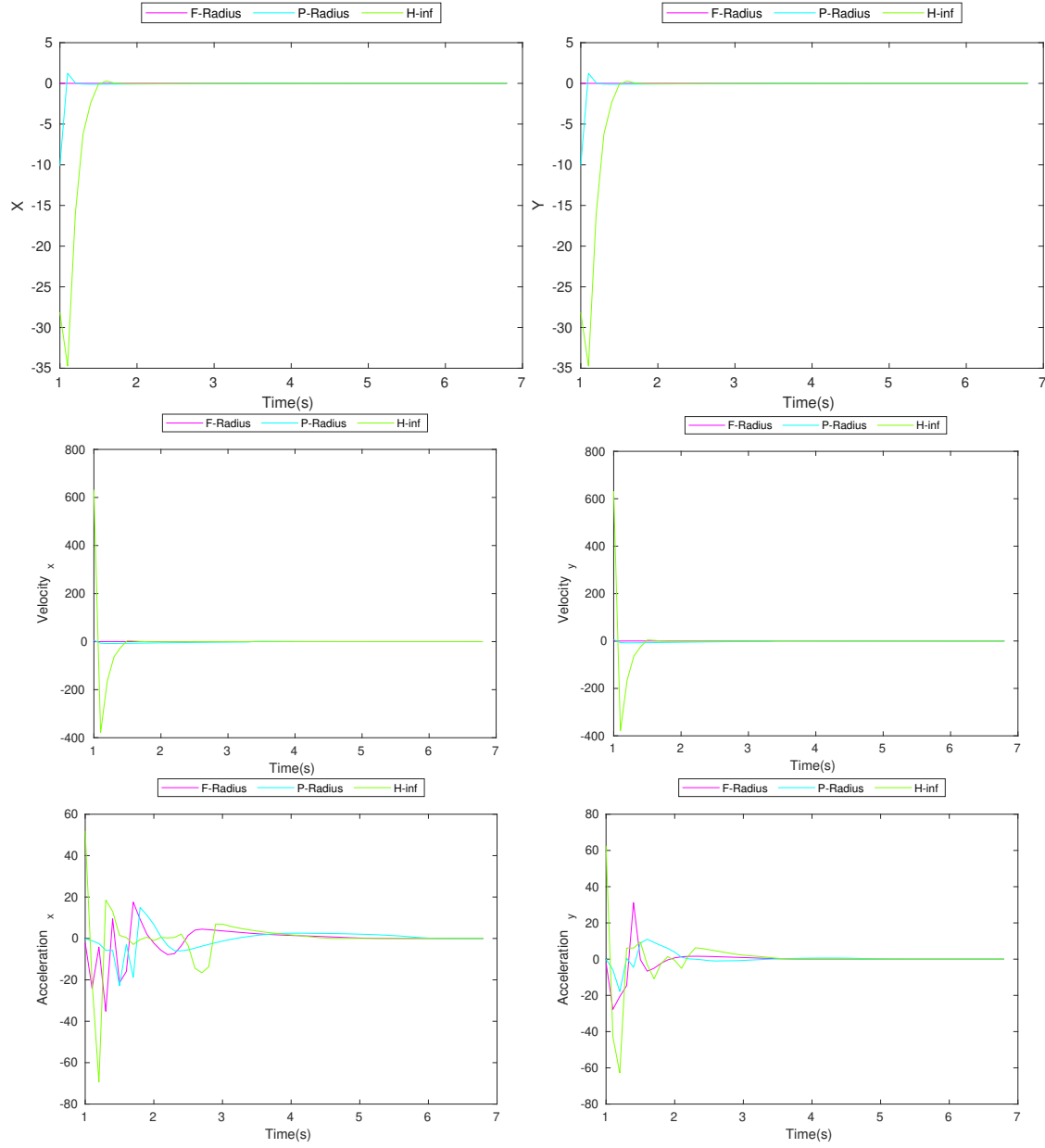


Figure 6.12: Rate of change of bounds using Point Mass Model

## List of Figures

2.1	An illustration of a zonotope and its interval hull in 2-D . . . . .	5
4.1	Computation time for each method to estimate using Constant Velocity Model . . . . .	10
4.2	RMSE(Root Mean Squared Error) . . . . .	12
6.1	Estimation using Constant Velocity . . . . .	16
6.2	Estimation using Constant Acceleration . . . . .	17
6.3	Estimation using Point Mass Model . . . . .	18
6.4	Estimation using Constant Velocity . . . . .	19
6.5	Estimation using Constant Acceleration . . . . .	20
6.6	Estimation using Point Mass Model . . . . .	21
6.7	Estimation using Constant Velocity . . . . .	22
6.8	Estimation using Constant Acceleration . . . . .	23
6.9	Estimation using Point Mass Model . . . . .	24
6.10	Rate of change of bounds using Constant Velocity Model . . . . .	25
6.11	Rate of change of bounds using Constant Acceleration Model . . . . .	26
6.12	Rate of change of bounds using Point Mass Model . . . . .	27

## List of Tables

4.1	Comparison of computation time (ms) using Constant Velocity Model .	9
4.2	Comparison of average time(in ms) to converge for unmeasured state .	10
4.3	Comparison of bounds of estimation . . . . .	11
4.4	Comparison of RMSE . . . . .	13

# 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] J. J. Rath and M. Althoff. "Evaluation of Set-Based Techniques for Guaranteed State Estimation of Linear Disturbed Systems." In: (2020).
- [4] M. Althoff. "CommonRoad : Vehicle Models." In: (2016).
- [5] 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).
- [6] V. T. H. Le, C. Stoica, T. Alamo, E. F. Camacho, and D. Dumur. "Zonotopic guaranteed state estimation for uncertain systems." In: *Automatica* 49 (2013), pp. 3418–3424.
- [7] M. Althoff, N. Kochdumper, and C. Arch. "CORA 2018 Manual." In: (2018).
- [8] 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.
- [9] 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.
- [10] T Alamo, J. M. Bravo, and E. F. Camacho. "Guaranteed state estimation by zonotopes." In: (2005).
- [11] 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.