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:	05.11.2020

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, 05.11.2020 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 (Global Positioning System) 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 a point-estimate 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 any divergence from the true state. The algorithms implemented are segment intersection methods(using F-radius, P-radius and volume) and interver observer (using $H - \infty$ observer) and are compared in terms of computation time, time to converge, tightness of bound and accuracy. H-∞ interval observer has perfomed 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

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 devlopment 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 [1].

The Collision Avoidance System in a 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 gurantee measurement in low-light environment (e.g. night) [2]. On the other hand, radar gurantee robustness to weather in exchange of high cost. Similarly, GPS has disturbances too which favour 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 to 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. Such statistical data required for this method is not always deductible for all situations. Moreover, Kalman filter provides point estimation,

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

not favourable for the automobile scenario. This motivates to use set-based state estimation methods.

The set-based state estimation technique provides a set of state bounding the true state of the system considering the bounds of the noise in the measurement and process. The skeleton of every set-based state estimation method is similar to Kalman filter, consisting of a prediction step and a correction step. The prediction step comprises of computing the state estimation using previous estimation and a model to define the transition. The correction step differs across algorithms and has been solved by Single Value Decomposition, computing Gain Matrix, solving LMI etc. Three distinguishable segment minimization methods and one Luenberger observer are implemented and evaluated in this paper.

Comparing different domain representation of the sets enclosing the possible state of the system, zonotopes are chosen for this paper 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 [3], 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.

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 [4] on simulated data.

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 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.

$$y = \begin{bmatrix} s_x & s_y \end{bmatrix}^T$$

Given a model ((2.1)), the problem of state estimation is to compute an outer bound of the state(x_k) consisting of the possible values of the true possible state of the system. The dimensions 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, the following models are investigated:

- **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).

$$\begin{aligned} x &= \begin{bmatrix} s_x & s_y & v_x & v_y \end{bmatrix}^T \\ A &= \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{2.2}$$

2.2.2 Constant Acceleration Model

Although the Constant velocity Model is easy to implement, it is ineffective 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 (2.3).

$$\begin{aligned} x &= \begin{bmatrix} s_x & s_y & v_x & v_y & a_x & a_y \end{bmatrix}^T \\ A &= \begin{bmatrix} 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 1 & 0 & \Delta T & 0 & \frac{1}{2}\Delta T^2 \\ 0 & 0 & 1 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{2.3}$$

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 [3], which is similar

to the constant acceleration model, except that the acceleration can strike 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 and measurement matrices are same as the constant acceleration model. The acceleration bounds are set as $11.5 m/s^2$ in both x and y-direction for this paper.

2.3 Zonotopes: The Set Representation

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) [6], is used to construct zonotopes and apply the computation for zonotopes in the algorithms.

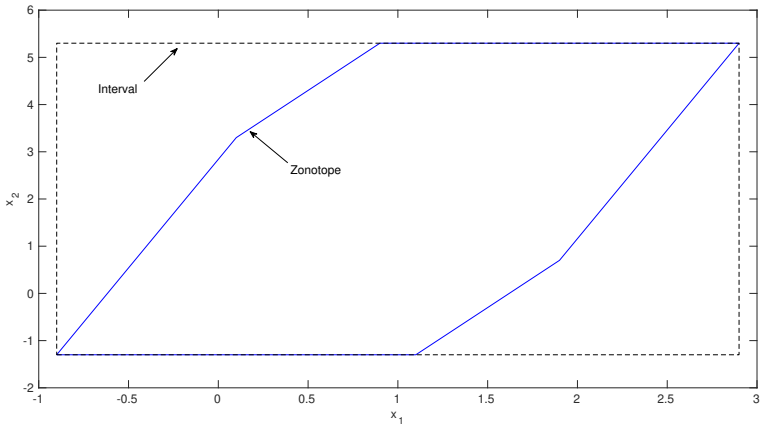

Figure 2.1: An illustration of a zonotope and its interval hull in 2-D

Zonotopes are gaining fame in the set-based state estimation techniques [7], due to its control of wrapping effect and that the Minkowski sum of the zonotope results in zonotopes. the prediction step for state estimation method can be simplified to basic matrix computation. The functionalities required in the predition method are implemented in the Matlab® toolbox CORA.

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

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

For the system in (3.1), let the set of predicted state of the system at time k be denoted by a zonotope, $\overline{\mathcal{X}_k} = p \bigoplus H\mathbf{B}^r$. The set to represent the i^{th} state in measurement($y_{k/i}$) at time k is a strip, denoted by $\mathscr{S}_i = \{x \in \mathbb{R} : |C_i x - y_{k/i}| \leq v_{k/i}\}$ [1]. The estimation at

[1] C_i is the i^{th} row of C

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 $\lambda_i \in \mathbb{R}^n$ such that (3.2).

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

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. Following sections briefly discuss three different approaches to solve the problem by focusing on three distinguishing properties of zonotope.

3.1.1 Frobenius norm of generators

This algorithm solves the problem by minimizing the F-norm of the generators of the intersected zonotope. 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)[10].

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

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

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.

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 [10]:

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

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 *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 which are not time-sensitive, e.g. fault diagnosis and fault tolerant control systems [11] .

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 [10].

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

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®.

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

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 [12].

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 [13], 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 deifinite 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).

$$\min_{\gamma_2} \; s.t. \; (3.10) \tag{3.9}$$

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

With L derived using a mosek solver in Matlab®, the estimated state of the system is found by Equation (3.8).

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 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 are 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. The initial state of the system is set using assignments (4.1).

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

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-∞ 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, 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

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

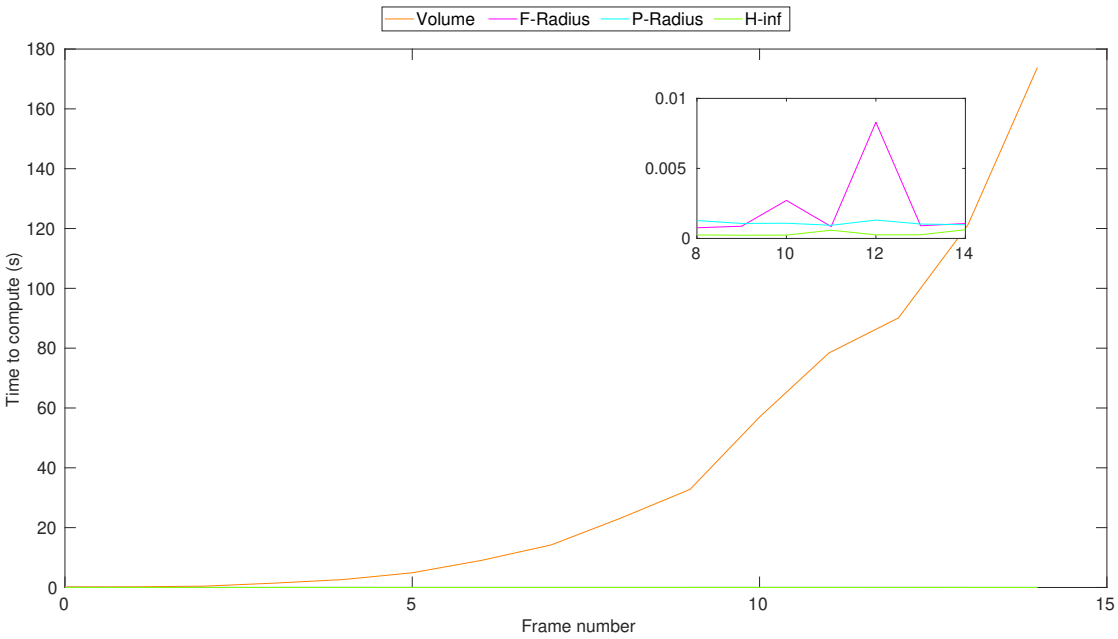

Figure 4.1: Computation time for each method to estimate using Constant Velocity Model

half the time required for the segment intersection methods, although the computation time does not consider the time for pre-computation for the techniques.

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

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

	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 Table 4.3. Segment intersection using F-radius has 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 evaluated.

The initial estimation before convergence gives extreme error which affect 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-∞ 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 Tab. 4.4. The error expectation from segment minimization from F-radius is similar to H-∞ observer, however, the error in measured state is slightly lesser in F-Radius compared to H-∞ observer.

Table 4.4: Comparison of RMSE

	Mean $\pm$ 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

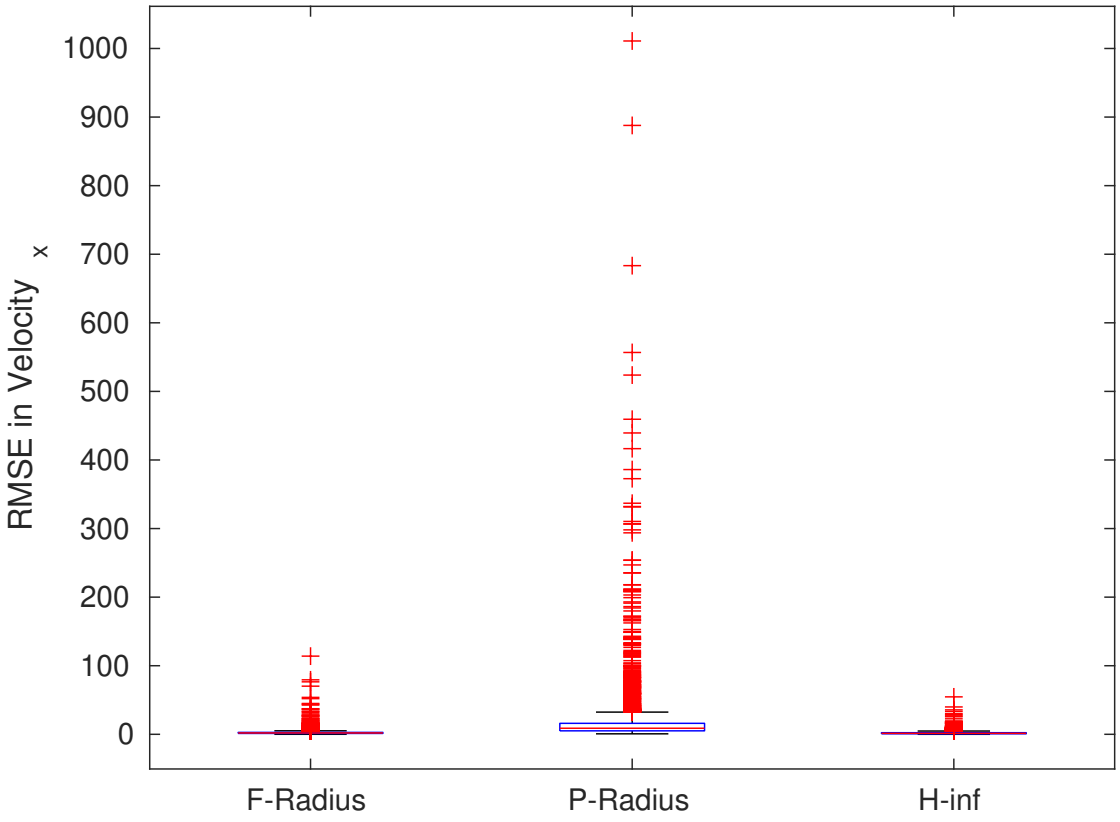

Figure 4.2: RMSE(Root Mean Squared Error)

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 ensures faster convergence to more accurate and tighter bounded estimation. H$-\infty$ 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-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 estimated state 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 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

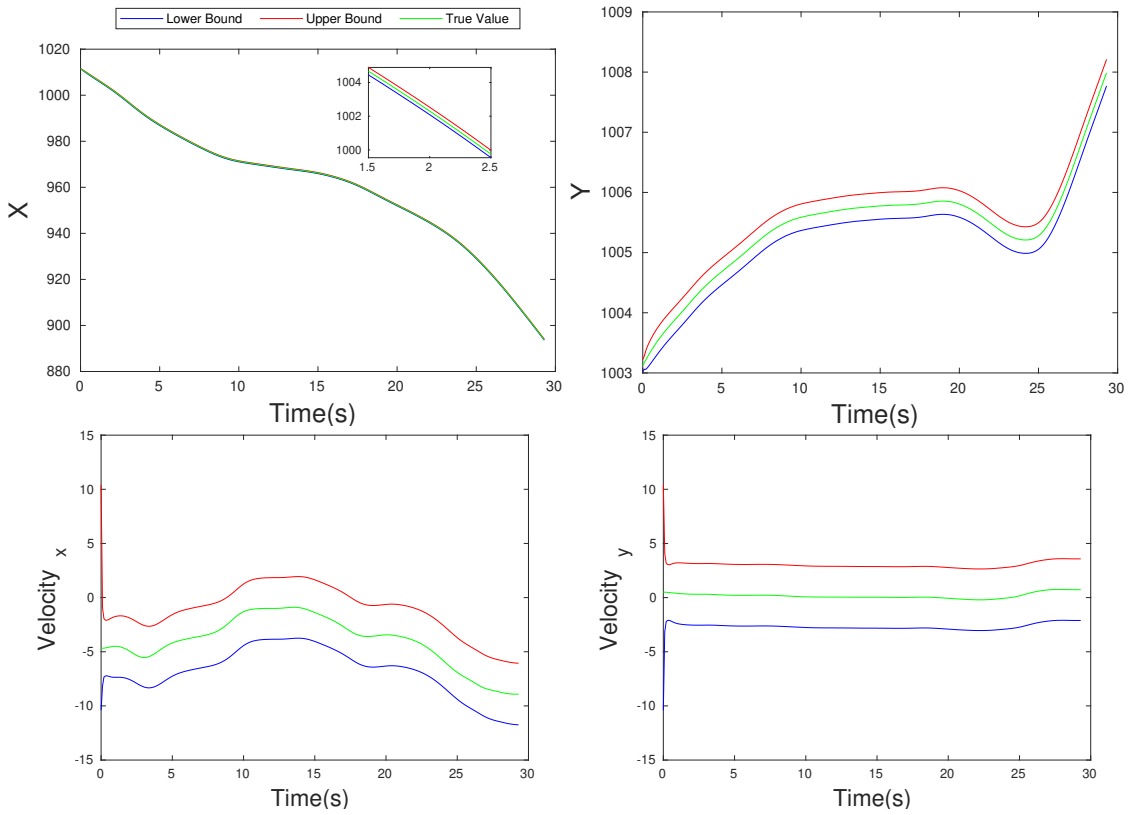

Figure 6.1: Estimation using Constant Velocity

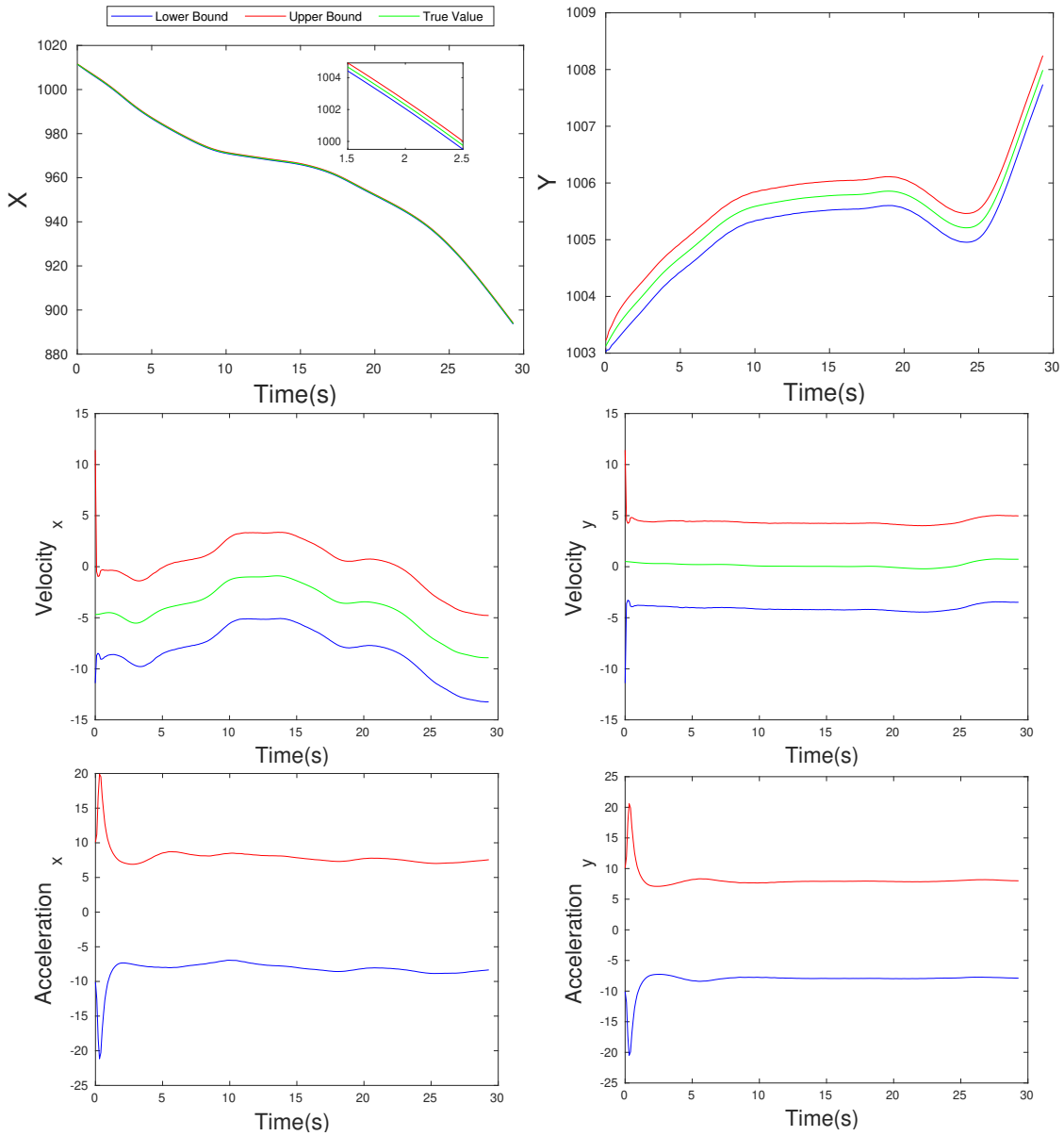

Figure 6.2: Estimation using Constant Acceleration

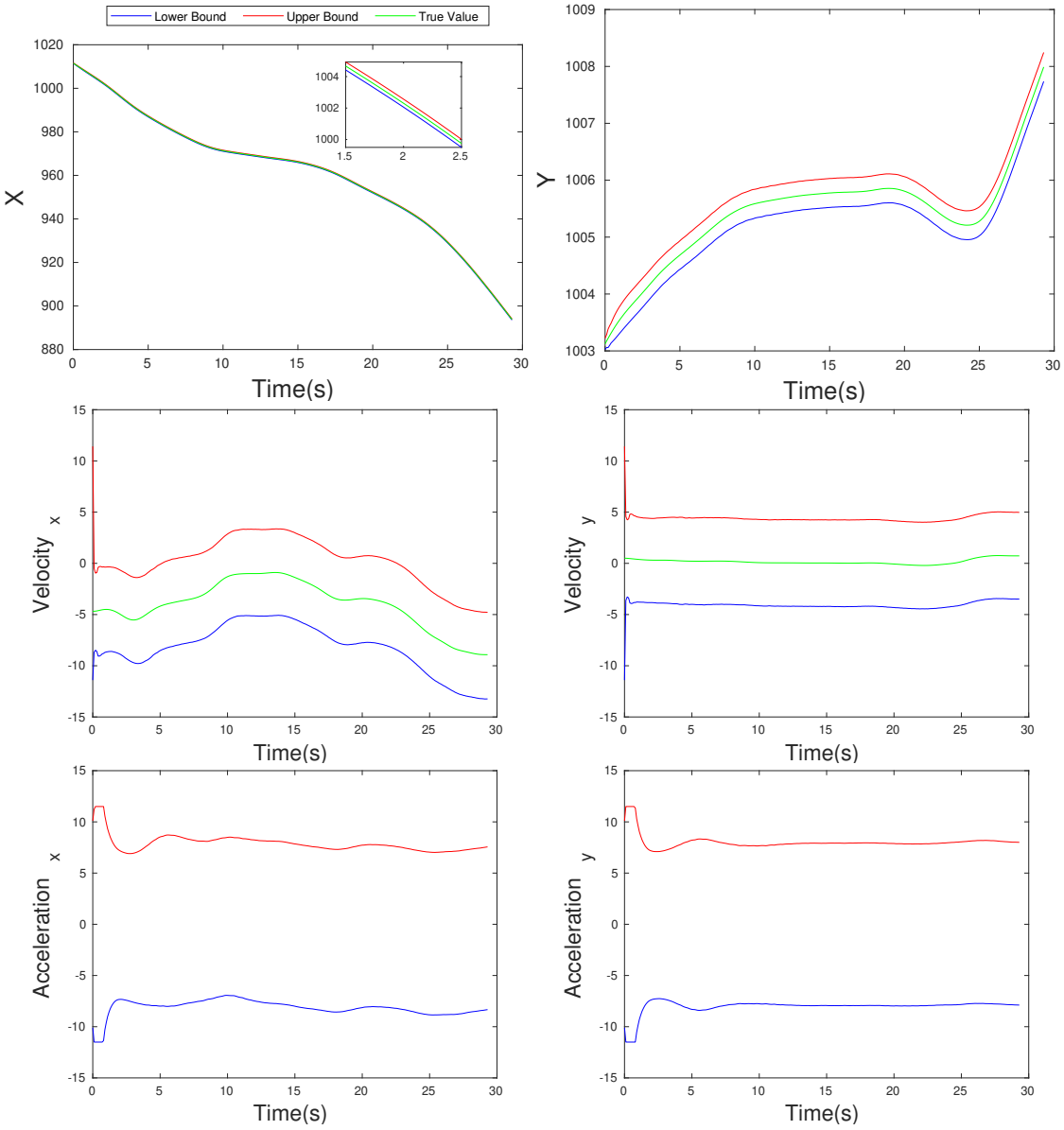

Figure 6.3: Estimation using Point Mass Model

6.1.2 Segment Minimization using P-Radius

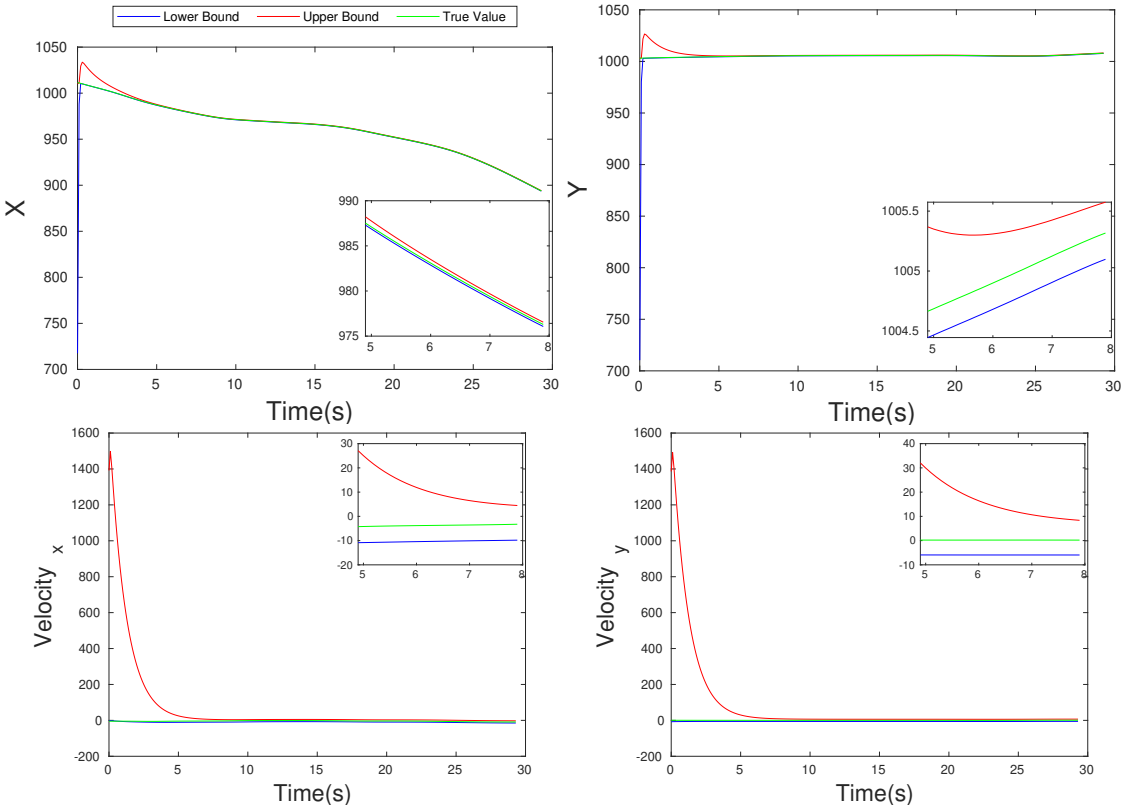

Figure 6.4: Estimation using Constant Velocity

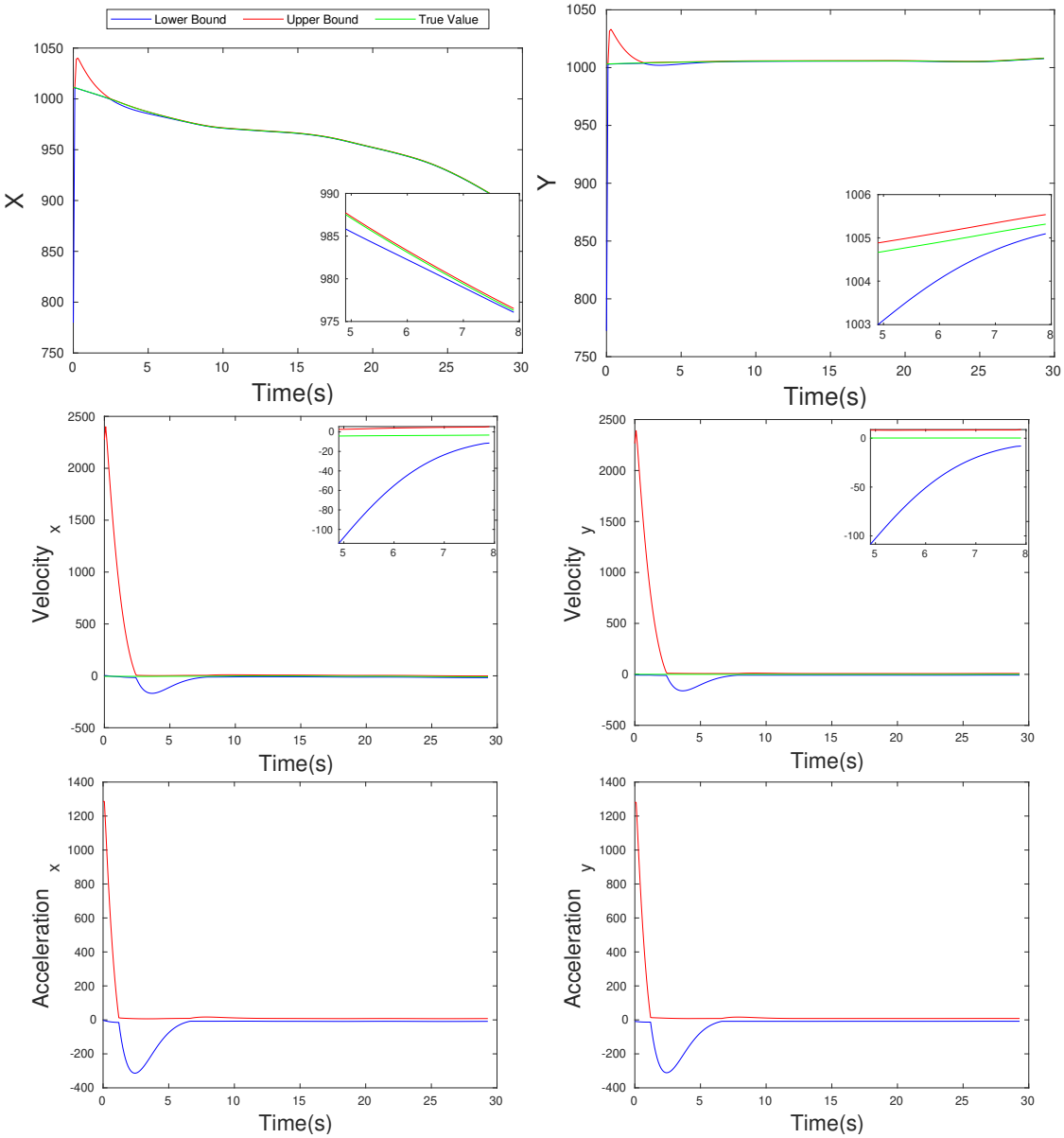

Figure 6.5: Estimation using Constant Acceleration

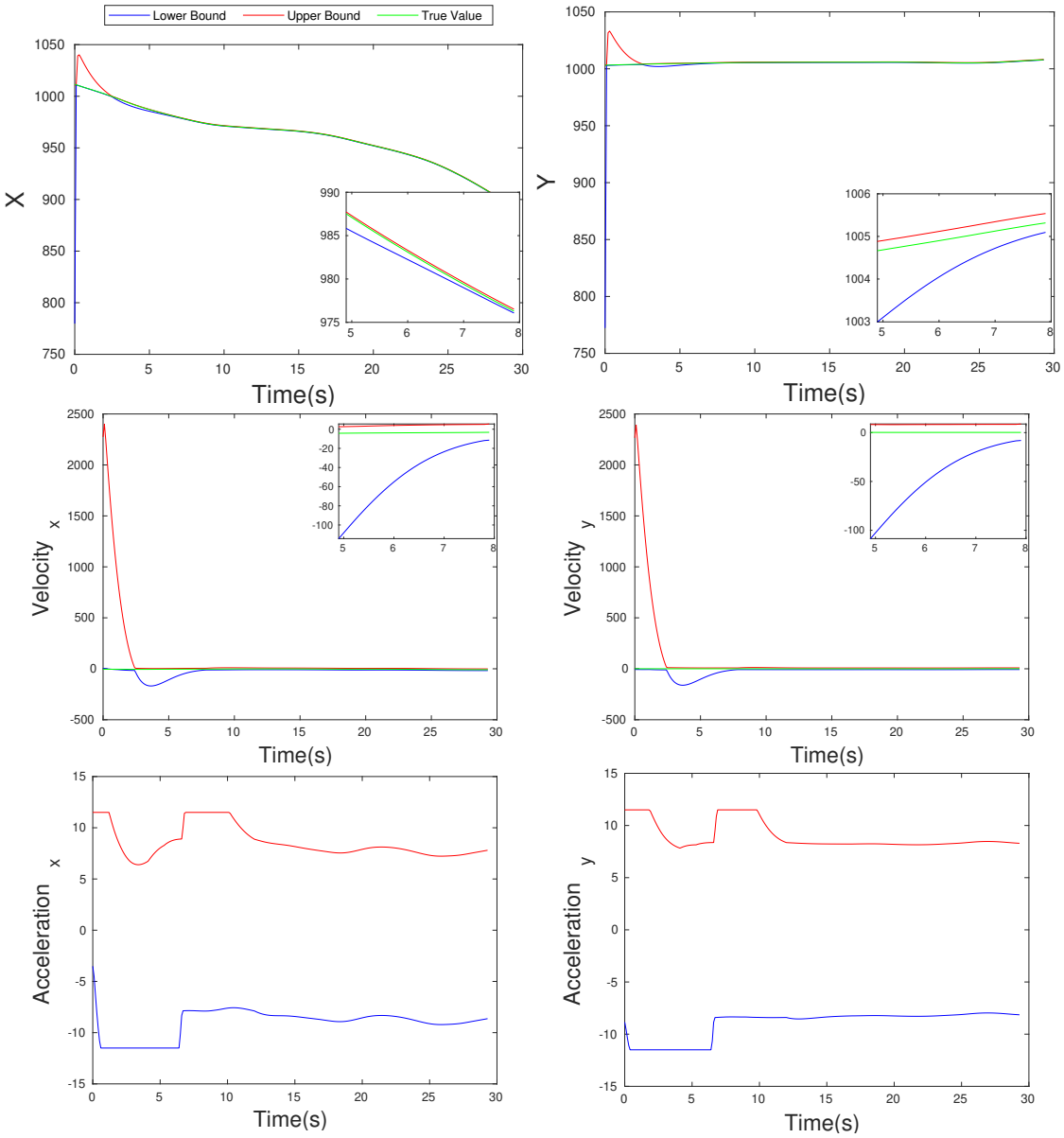

Figure 6.6: Estimation using Point Mass Model

6.1.3 Interval Observer using H-∞

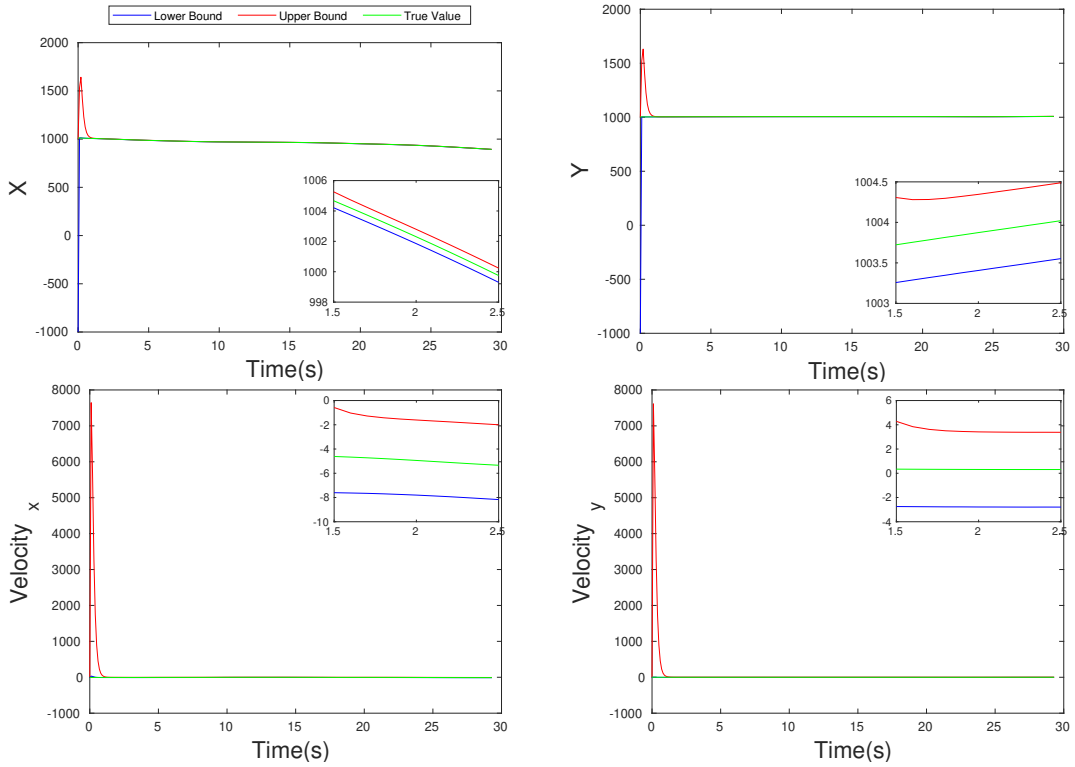

Figure 6.7: Estimation using Constant Velocity

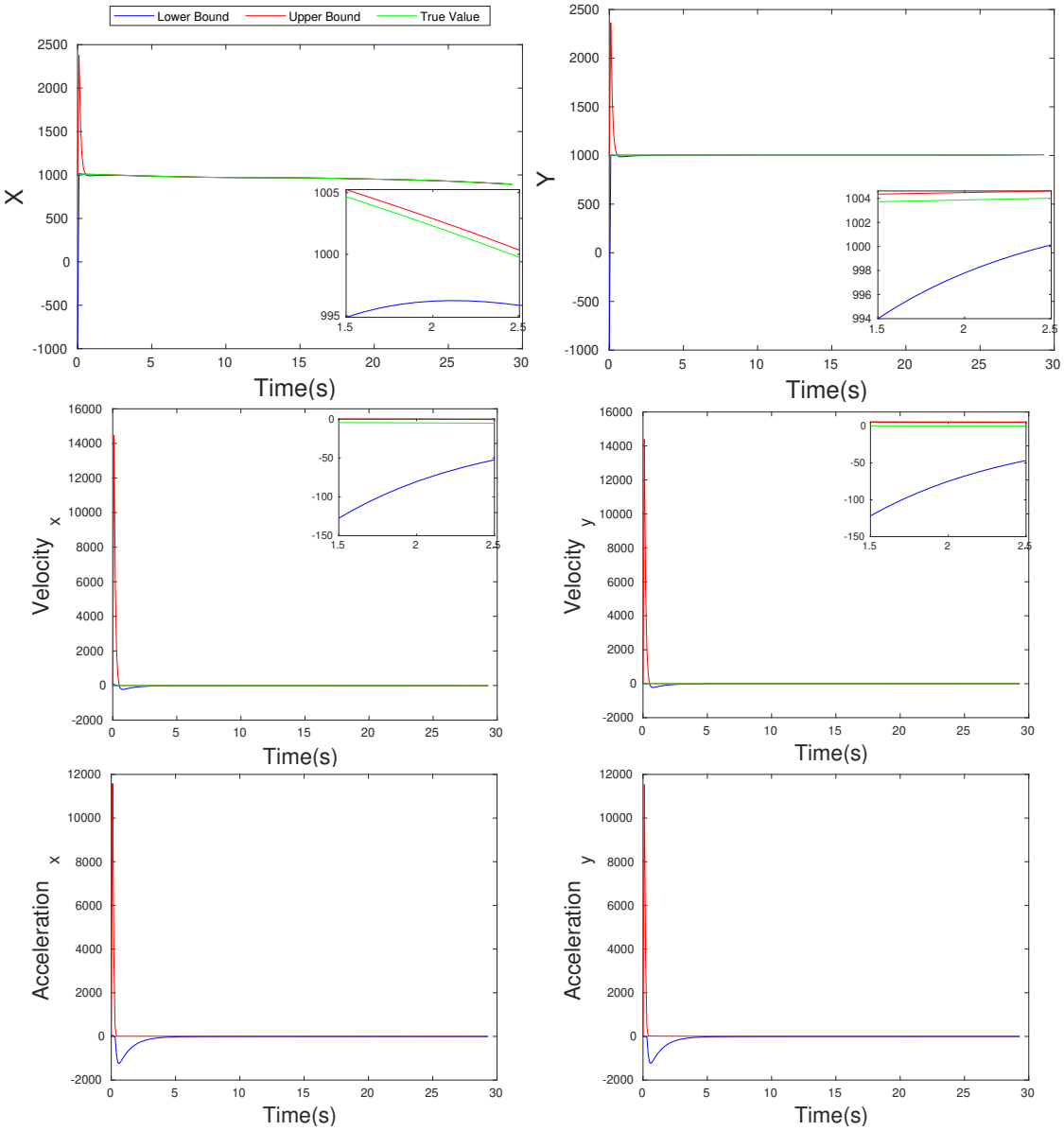

Figure 6.8: Estimation using Constant Acceleration

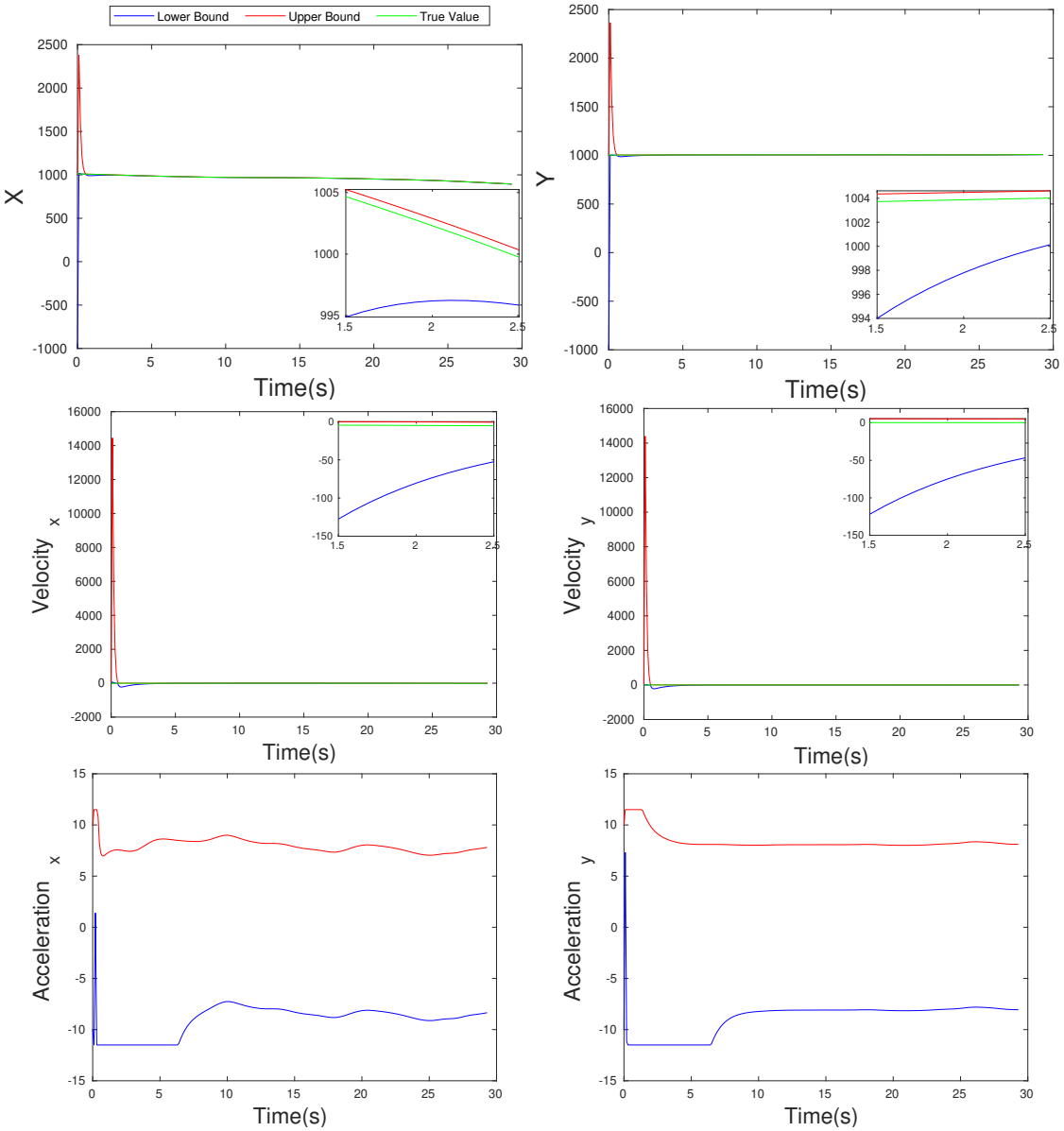

Figure 6.9: Estimation using Point Mass Model

6.2 Rate of Change of Bounds

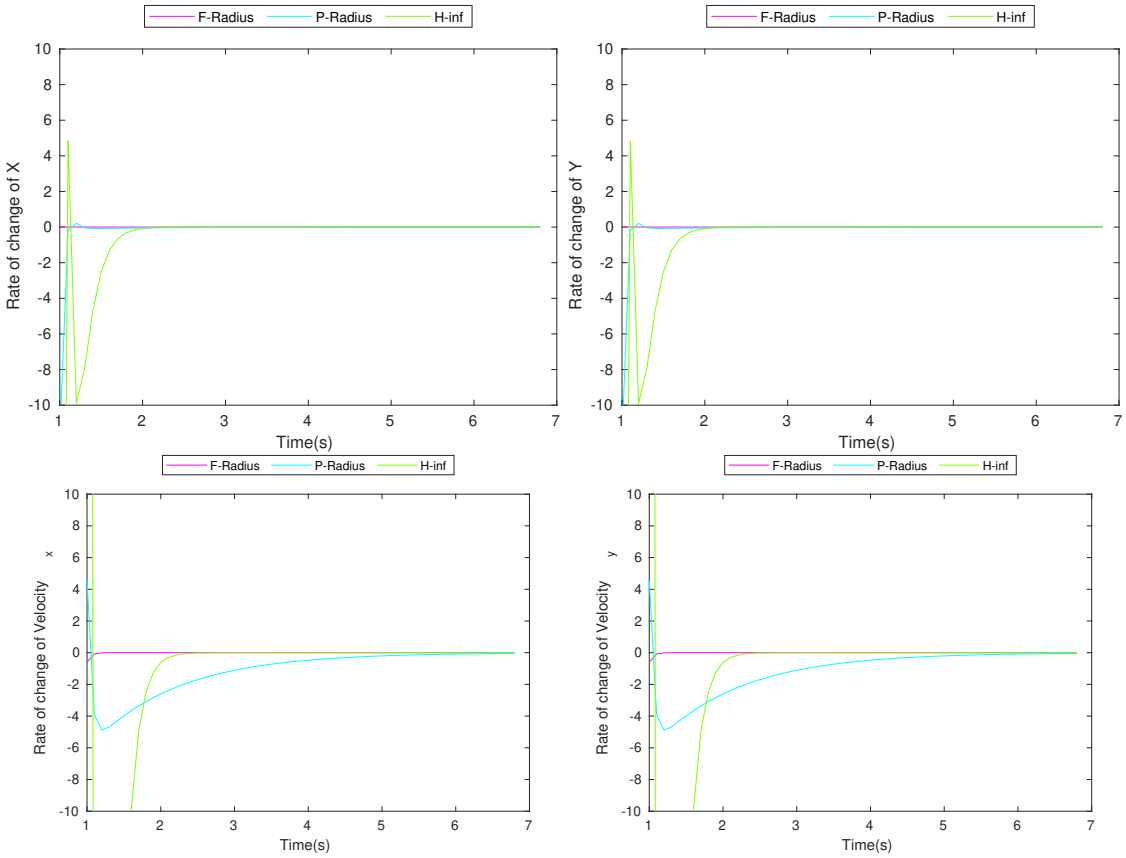

Figure 6.10: Rate of change of bounds using Constant Velocity Model

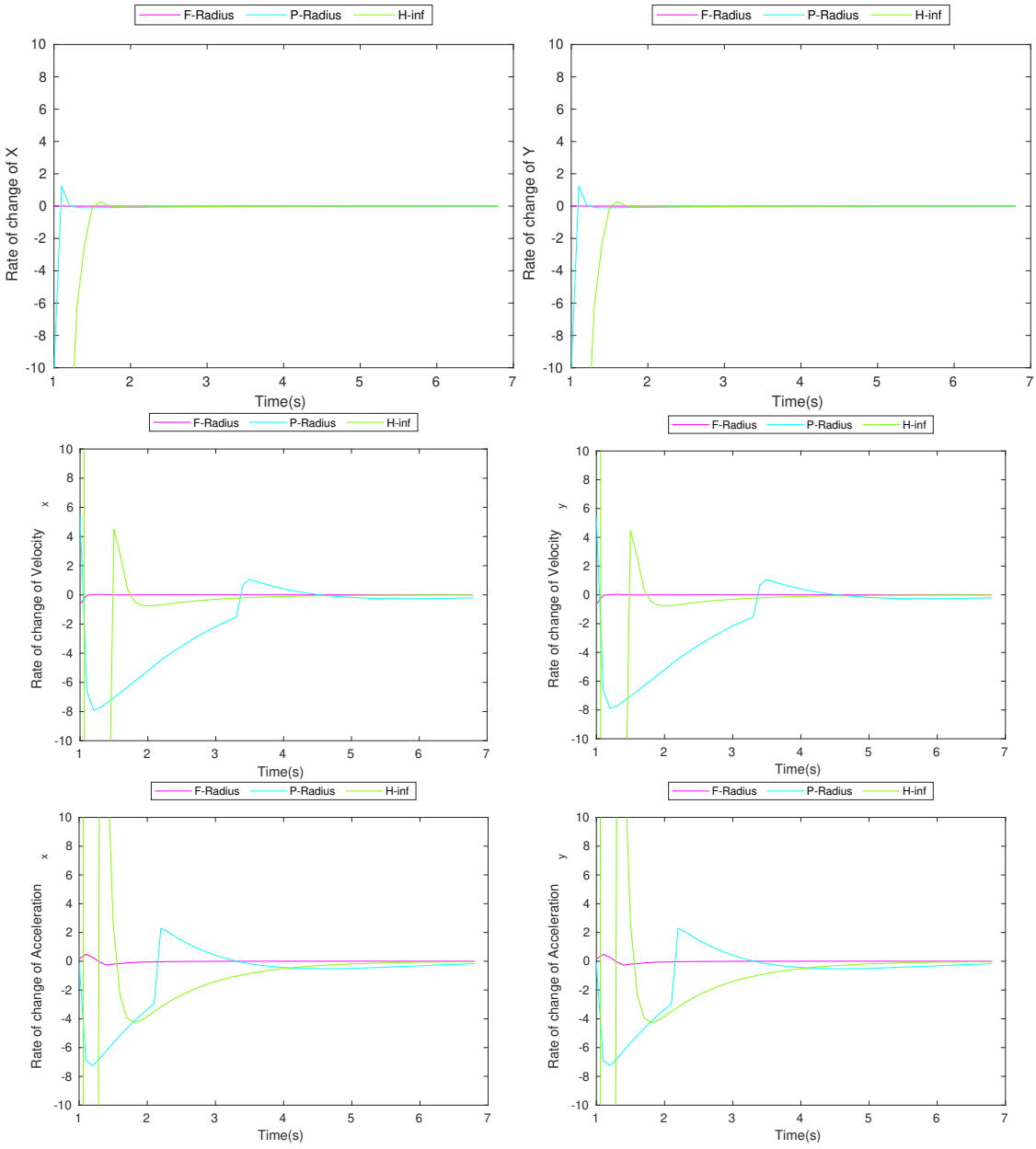

Figure 6.11: Rate of change of bounds using Constant Acceleration Model

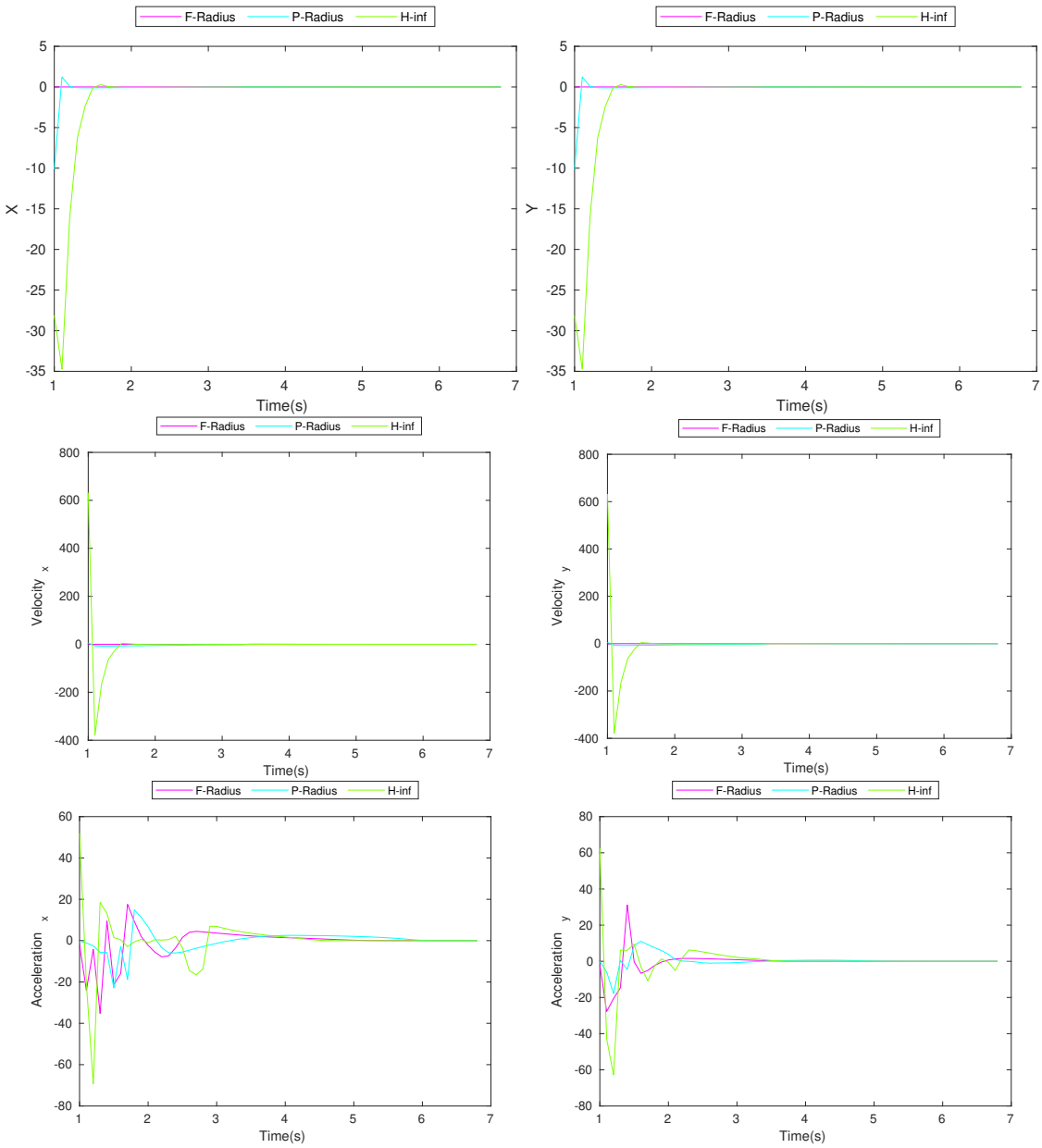

Figure 6.12: Rate of change of bounds using Point Mass Model

List of Figures

List of Tables

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] M. Althoff. "CommonRoad : Vehicle Models." In: (2016).

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

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

[7] 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.

[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] 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.

[12] 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.

[13] 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.