

Robust Min-Max Localization Algorithm

Ryan Parker and Shahrokh Valaee

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto, 10 King's College Road, Toronto, ON, M5S 3G4, Canada
e-mail: {rparker, valaee}@comm.utoronto.ca

Abstract—The advent of dedicated short range communication (DSRC) devices — which are designed to support vehicle-to-vehicle and vehicle-to-infrastructure communication — promises a wide set of new active safety applications (e.g. a cooperative collision warning system based on vehicle-to-vehicle communication). One of the fundamental challenges for realizing these systems will be to accurately and reliably determine a vehicle's position with respect to its neighbours. We propose a distributed algorithm using inter-vehicle distance estimates to localize a vehicle among its neighbours. Given that the inter-vehicle distance measurements contain noise, we will present a robust min-max optimization algorithm that precisely predicts the vehicle position within a cluster. Simulation studies show that our algorithm outperforms previously proposed localization schemes.

I. INTRODUCTION

The motivation for this work stems from the recent introduction of dedicated short range communication (DSRC) devices. DSRC is based on the 802.11p standard for Wireless Access in Vehicular Environments (WAVE) and is designed to support short range, low latency, high speed vehicle-to-vehicle and vehicle-to-infrastructure wireless communication. In the future, vehicles will likely be equipped with DSRC devices. DSRC will support a variety of safety applications and other ITS applications such as electronic toll collection, real-time traffic advisories, and digital map updates [1].

One of the most promising vehicular safety applications is the development of an advanced cooperative collision warning system. It is envisioned that the advanced vehicle collision warning system will use vehicle-to-vehicle radio communications to create a cooperative collision warning system, where vehicles — equipped with DSRC devices — cooperatively share information (i.e. location, speed, heading, acceleration, etc.) for collision anticipation. By sharing this information between peers, each vehicle is able to predict potential hazards. It was shown by Tatchikou et al. [2] that sending safety warning messages containing position information could substantially reduce the probability of collision within a platoon. However, accurately and reliably determining a vehicle's relative position among its neighbours still remains a fundamental challenge; as it is an essential component in estimating the likelihood of collision.

Designing a solution for accurate localization of neighbouring vehicles based on real-time exchange of position estimates, using vehicle-to-vehicle communication, is a challenging task. Given the sub-second decision latency requirement of cooperative collision warning systems, the solution must be able to establish the relative position of all neighbours in

real-time and continuously track their motion to proactively identify potential vehicle collision scenarios.

Currently, global positioning system (GPS) is used to determine a vehicle's location, which provides an accuracy of approximately 10 meters [3]. In GPS, a vehicle locates itself by comparing the signal received from four or more GPS satellites. Although, GPS can create precise location estimation where direct line-of-sight to multiple satellites is possible, GPS signals can often become blocked or degraded when vehicles enter tunnels or are in downtown areas. Typically, during short-outages of GPS, vehicles can use a dead-reckoning system to maintain an estimate for their position [3]. However, dead-reckoning systems are prone to error accumulation. Even during short-outages (e.g. 30 seconds or less) the position estimate can become inaccurate by as much as 10-20 meters, if the vehicle is traveling at 100km/h. Inaccuracies in position estimation may result in alerting a driver when there is no danger of collision. Conversely, if position information is unavailable or degraded the system may fail to alert the driver when there is danger on the road ahead.

Although, there has been GPS systems with improved accuracy, such as differential GPS (DGPS) [3] and assisted GPS (A-GPS) [4], which achieve accuracies between 3 and 7 meters, these systems still are prone to the same reliability issues as regular GPS. It has been recently argued that a combined solution using GPS — or one of its improved variants — and the use of radio based ranging techniques (such as the received signal strength indicator (RSSI)), to determine distance estimates between vehicles in a cluster, can be used to increase the reliability of the location estimator [5].

The use of radio based ranging techniques also presents a number of unique challenges. First, the distance measurements are inherently noisy as a result of a number of factors including: limitations of the measurement device, multipath fading, shadowing and non-line-of-sight errors. Second, mobility complicates the handling of noise, since outliers and noisy measurements can be mistaken as observed motion and the effects of fading becomes more prevalent. Therefore, it is critical for a radio based position estimation algorithm to attempt to mitigate these errors.

In recent years, localizing nodes using the inter-nodal radio ranging distance measurements has been an active area of research. Radio based distance measurement based algorithms have been mainly used in wireless sensor networks, where it is not feasible for all the sensor nodes to have GPS.

Unfortunately, the majority of the earlier algorithms fail to consider that the range measurements contain noise or node mobility. Moore et al. [6] propose a *robust quads* algorithm, which seems to show promise for use in vehicular networks; so we will compare our algorithm to this one.

In [7], we have proposed a new localization algorithm for vehicular networks. In that algorithm, we minimize the mean square error of the measured distance between any two vehicles and their parametric distance model in a rectangular coordinate system. For a Gaussian error, the algorithm is optimal and produces Maximum Likelihood estimates. Since the cost function in [7] reduces the average error, single terms may have large error. In the present paper, we introduce a new robust min-max estimator that minimizes the maximum error. At a high level, our localization algorithm works as follows. Each node measures distances to its neighbouring nodes and shares this information with its neighbours. Based on this one-hop information, each node solves a min-max problem to create a map of the relative position of its neighbours in the same cluster. Also, if a subset of these nodes (i.e. at least 3) has prior information about its position with respect to a global coordinate system, then all the nodes within the cluster can determine their global position.

The remainder of this paper will show how accurate position estimates for vehicles can be made using inter-vehicle distance estimates. We will first provide an overview of related radio based localization algorithms and techniques used. Then, give a detailed explanation of how our algorithm works and show how its performance compares to other localization schemes.

II. RELATED WORK

In recent years, localizing a node using inter-nodal radio range based distance measurements has been an active area of research (see [8] for survey). In general, localization schemes can be separated into two classes: course grain and fine grain. Within each of these schemes there are distributed and centralized approaches. Typically, course grain localization uses mere connectivity radio between nodes and requires a central node in the network to establish position estimates for the nodes (e.g. [9]) and are generally used when a rough estimate of a node's position is sufficient. Conversely, the goal of fine grain localization is to provide very accurate node position estimates using either centralized or distributed approach. We will mainly focus on reviewing the distributed fine grain approaches, as these are most applicable for vehicle networks and most closely related to our work.

An algorithm, which is successfully used in a vehicular network environment, must meet the following criteria: (i) be able to handle node mobility; (ii) produce accurate position estimates, given there is noise in the inter-vehicle distance measurements; (iii) operate in a distributed fashion. So, we will now examine some of the existing localization approaches and evaluate them against the fore-mentioned criteria to determine their feasibility for application in vehicle networks.

In [10] a GPS-free positioning algorithm for mobile ad-hoc networks was proposed where each node runs a self-positioning algorithm that computes the angles between the one-hop neighbours — based on the inter-node distance measurements — to establish a local coordinate system. Once the local coordinate systems are formed, the nodes orient their coordinate system to a common coordinate system where all nodes' x, y coordinates point in the same direction. Unfortunately, the GPS-free positioning algorithm [10] is expensive in terms of the number of messages that need to be exchanged between nodes. As a result, this procedure does not scale well and is not well suited for high mobility environments, where nodes are frequently entering and exiting clusters. Iyengar and Sikdar [11] derived an improved version of [10], to tackle these issues, by creating an algorithm that improves scalability and convergence times.

Kukshya et al. [5] made use of the results from [11] to create a scheme for localizing neighbouring vehicles based on radio range measurements. Their goal was to establish an accurate map of the relative positions of all neighbouring vehicles. Under the assumption that vehicle did not have access information from GPS or dead-reckoning system (e.g. operating in conditions where GPS did not have line of sight). They use trilateration for estimating a vehicles position, however noise in the range measurements can quickly cause error to propagate as the coordinate systems are aligned to a common coordinate system.

In [12] a "DV-hop" method is proposed where a node is localized based on considering a set of distances to anchor nodes. The method works as follows: each anchor node floods its location to all nodes in the network. Each node with unknown location records the position and the number of hops to at least 3 anchors. Whenever anchor a_1 infers the position of another anchor a_2 , it calculates the distance between them, divides by the number of hops and floods the network with the average hop distance. Each node next uses this average hop distance to convert hop counts to distances. Then, the node uses these distances to perform triangulation to three or more distant anchors to establish an estimate of its position. This method works well for dense topologies, but degrades significantly for sparse and "hard" network topologies (i.e. cases where whole or parts of the network can be rotated or flipped given the same set of distance measurements.) Savarese et al. [13] use a similar method to [12], but the author's propose a heuristic to detect networks with "hard" network topologies. They consider a node uniquely localizable only if a node has three disjoint paths to three distinct beacons and suggest using a least-squares optimization to reduce the effects of measurement errors. However, [14] disprove this heuristic and derive a theoretical framework for determining nodes that have a unique localization in terms of graph rigidity theory.

Moore et al. [6] use this theoretical framework to derive a *robust quads* algorithm to determine the nodes positions in the presence of noisy range measurements. They identify two scenarios which prevent a localization from being unique: (i) a flip ambiguity where all the distance measurements are the

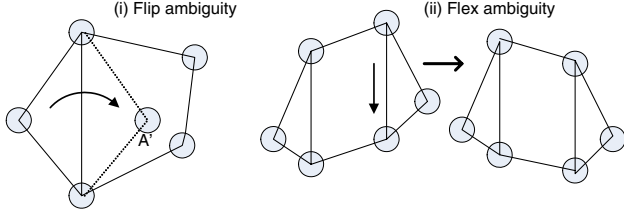


Fig. 1. (i) Flip ambiguity Node A can be either A or A' and the distance constraints are still satisfied (ii) Flex ambiguity the distance constraints are the same except the graph is sheared

same but parts of the graph can become mirror reflections; or (ii) flex ambiguities where the distance constraints remain the same, but the graph is sheared. Moore et al. [6] proved that flex ambiguities do not pose a problem for fully connected graphs. They derived a heuristic to detect nodes which have a high probability of having a unique realization (i.e. a unique position, given a set of distance constraints). The objective of their algorithm was to only localize those nodes with unique realization. With the goal that the propagation of localization error would be minimized as the coordinate systems are aligned to a common orientation. The *robust quads* algorithm [6] shows promise for use in vehicular networks. The algorithm was deployed in a sensor network and had one experiment where a single node had mobility and rest of the nodes remained stationary, showing that a mobile node could accurately be tracked within the network. Of the algorithm's surveyed the *robust quads* seems to show the most promise for use in vehicular networks, so we will compare our method to it.

III. ALGORITHM FRAMEWORK

In this section, we will give a formal definition of the problem and then present our algorithm.

A. Problem definition

Consider a network with n vehicles labeled $1, 2, \dots, n$ at unknown distinct locations in some physical region. By using a radio-based ranging technology (e.g. received signal strength indicator (RSSI)) each vehicle estimates the distance to its neighbours. We denote the distance measured between vehicle i , and vehicle j as $\hat{d}_{i,j}$. So, given the network of n vehicles, our objective is to produce a set of coordinate points, (x_i, y_i) , for each vehicle i (where $i \in \{1, 2, \dots, n\}$) such that after running our algorithm, the estimated Euclidean position of each node closely resembles (ideally identical to) the actual (or the ground truth) Euclidean position of each vehicle up to a global translation and rotation.

B. Algorithm

Our algorithm works as follows, each vehicle builds its own local coordinate system, setting itself as the center of its own coordinate system with position $(0,0)$, with the objective to estimate the (x,y) coordinates of all its neighbours. The distributed vehicle localization algorithm can be broken

down into three phases, as follows.

Phase 1: Create Initial Position Estimates

Rough initial estimates for unknown vehicle positions are obtained, using GPS or trilateration. Specifically, each vehicle measures inter-vehicle distance and exchanges this information with its neighbours to establish a matrix of inter-vehicle distance estimates for its one hop neighbours. Using this distance information, trilateration can be performed to establish initial position estimates. Note that in trilateration, vehicles position may be ambiguous. It is possible to have flip or flex ambiguities (see Figure 1). To attempt to mitigate these ambiguities, one possibility is to leverage vehicle mobility and make multiple iterations of the trilateration and use the position estimate with the maximum likelihood as an initial position estimate.

Phase 2: Refine the Position Estimates:

Each node refines the initial estimates to provide a final estimate of its neighbours location. We formulate the objective function as min-max optimization problem, let $f_{i,j}$ be defined as

$$f_{i,j}(t) = \hat{d}_{i,j}(t) - \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}. \quad (1)$$

where $(x_i(t), y_i(t))$ and $(x_j(t), y_j(t))$ are the location estimates of nodes i and j , respectively, at time t and $\hat{d}_{i,j}(t)$ is the measured distance between the vehicle i and vehicle j at time t . Using (1), each vehicle can create an $n \times n$ matrix (where n is the number of nodes in the cluster), which takes the form:

$$\begin{bmatrix} 0 & f_{1,2} & f_{1,3} & \dots & f_{1,n} \\ f_{2,1} & 0 & f_{2,3} & \dots & f_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_{n-1,1} & f_{n-1,2} & \dots & f_{n-1,n-1} & f_{n-1,n} \\ f_{n,1} & f_{n,2} & \dots & f_{n,n-1} & 0 \end{bmatrix}.$$

Then location estimates $(x_i(t), y_i(t))$ are selected according to:

$$\min \max_{i,j} w_{i,j}(t) |f_{i,j}(t)| \quad (2)$$

where $w_{i,j}(t)$ is a weight based on the magnitude of the distance measured between vehicle i and j ; because it is important that vehicles at close range be localized more accurately for cooperative collision based warning system. With knowledge that vehicles are confined to road boundaries, we can reduce the search space of (2); thus, reducing the complexity and potentially improving the accuracy of our algorithm. Also, using final position estimates from the previous iterations, the current position estimate can be compared to ensure the change in a vehicle's position is consistent with velocity constraints. Performing this check would also allow us to smooth out any irregularities that exist from one round of position estimation to the next and reduce the probability of incorrect localization. Taking these constraints into account

we can reformulate the optimization problem as:

$$\begin{aligned} \min \max_{i,j} \quad & w_{i,j}(t) |f_{ij}(t)| \\ \text{s.t.} \quad & (x_i(t), y_i(t)) \in \text{Road Constraints} \quad \forall i \\ & (x_i(t), y_i(t)) \in \text{Velocity Constraints} \quad \forall i. \end{aligned} \quad (3)$$

where Road Constraints are found from the road geometry and Velocity Constraints are found by predicting the location of the vehicles using location estimates from the previous timeslots and vehicle kinematics. Examples of Road Constraints can be suggested by enforcing road geometry. For example, each vehicle on the road should have a $x_i(t)$ position that satisfies:

$$a_x \leq x_i(t) \leq b_x$$

where a_x and b_x are found from road width. An example for Velocity Constraints may be suggested as:

$$|y_i(t) - y_i(t-1) - \Delta_y(t)| \leq \delta_y$$

where $y_i(t-1)$ is the estimated location of the vehicle at the previous timeslot, $\Delta_y(t)$ is the predicted movement of the vehicle in the current timeslot, and δ_y is an error term that presents the uncertainty in the velocity and location estimates. The optimization problem (3) can then be solved using iterative algorithms, as outlined in [15], [16], which exhibit a quadratic rate of convergence to a solution.

Phase 3: Iterate

Phases 1 and 2 are repeated every T seconds, constituting a new round of the algorithm. In general, the T seconds between inter-vehicle distance exchanges is dictated by whether or not a vehicle can make use of neighbouring vehicles velocity to update their position estimate.

IV. SIMULATION RESULTS

We study the performance of our algorithm under two scenarios. First, we compare our algorithm to the *robust quads* proposed by Moore et al. [6]. The *robust quads* algorithm is designed for sensor networks it does not take advantage of some of the unique information available in vehicle networks, therefore to make a fair comparison we have chosen to compare it to the generalized version of our algorithm (2). Second, we study the performance of our algorithm on a simulated highway using CORSIM (CORridor SIMulator), which has been developed by the US Federal Highway Administration to model vehicle movements.

The *robust quads* algorithm can be described as follows. Each node becomes the center of a cluster and estimates the relative location of its neighbours that can be unambiguously localized. Next, all the robust quadrilaterals within the cluster are identified. Then nodal position estimates within the cluster are incrementally computed by trilateration and chaining the quadrilaterals with 3 or more nodes in common. Finally, a set of transformations are performed between neighbouring clusters to align them to a common orientation.

A. Performance Metrics

To measure the accuracy of the initial position estimates, made in phase 1 of our algorithm, we compare the initial vehicle position estimate to the actual position. We define the root-mean-square error (RMSE) in the initial position estimate as:

$$\sigma_{\text{initial}} = \sqrt{\frac{\sum_{i=1}^n (x_{\text{int. est. } i} - x_{\text{actual } i})^2 + (y_{\text{int. est. } i} - y_{\text{actual } i})^2}{n}} \quad (4)$$

where $x_{\text{int. est. } i}, y_{\text{int. est. } i}$ is defined as the initial position estimate of vehicle i and $x_{\text{actual } i}, y_{\text{actual } i}$ represent the actual position of vehicle i . This metric can be thought of as the average distance the initial position estimate is from the actual position.

Similarly, we define the RMSE of the final position estimate as:

$$\sigma_{\text{final}} = \sqrt{\frac{\sum_{i=1}^n (x_{\text{final est. } i} - x_{\text{actual } i})^2 + (y_{\text{final est. } i} - y_{\text{actual } i})^2}{n}} \quad (5)$$

where $x_{\text{final est. } i}, y_{\text{final est. } i}$ is the position estimate of vehicle i after running phase 2 of our algorithm.

It is also important to consider the worst possible estimate where the distance between a vehicle's final position estimate and the actual position is a maximum. We have defined the following metric to capture this:

$$\sigma_{\text{max}} = \max_i \sqrt{(x_{\text{fin. est. } i} - x_{\text{actual } i})^2 + (y_{\text{fin. est. } i} - y_{\text{actual } i})^2} \quad (6)$$

We also examined the amount of noise present in the inter-vehicle distance measurements, defining the RMSE in the distance measurements as:

$$\sigma_d = \sqrt{\frac{\sum_{i=1}^M (\hat{d}_i - d_i)^2}{M}} \quad (7)$$

where M is the number of inter-vehicle distance measurements, \hat{d}_i is the measured distance to vehicle i , and d_i is the actual distance to vehicle i .

B. Comparison Study

In this study, we examined the effects of error in the inter-vehicle distance estimates on the final position estimate. We considered a cluster of 9 vehicles randomly distributed over a 200×200 square-meter section of road, at a single time instant. The RMSE in the distance measurements were varied leaving the other parameters constant. We set the (x,y) coordinates of the initial position estimate to deviate from the actual position according to a Gaussian distribution with standard deviation of 5 meters. The results of this experiment are shown in Figure 2. Each of the data points shown in Figure 2 is the average result of 30 runs of our algorithm; each run has different vehicle positions, to average out the effects of poor network topologies. From this figure, it can be seen that the plot of the root-mean-squared error in the

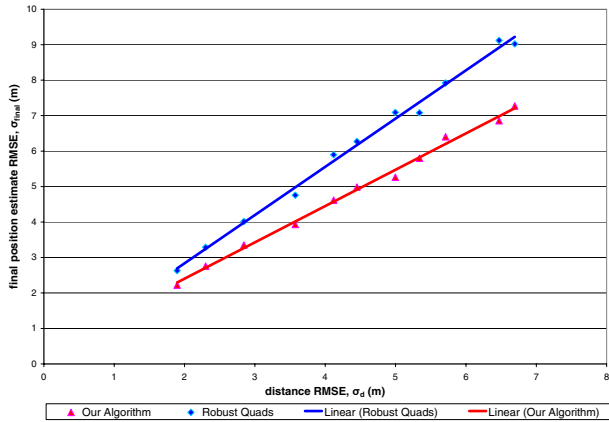


Fig. 2. Shows the average performance of our algorithm versus the robust quad algorithm

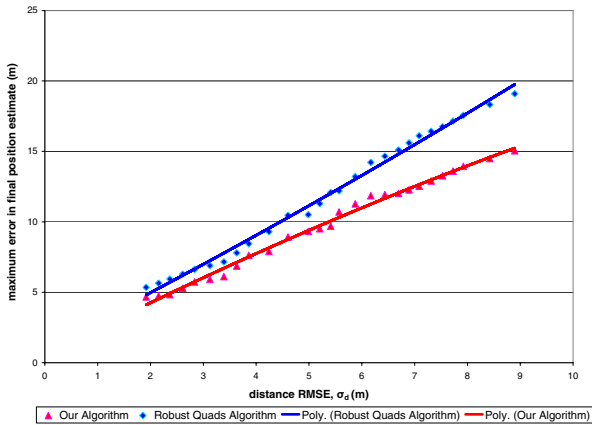


Fig. 3. Shows the worst case scenario performance of our algorithm versus the robust quad algorithm

final position versus the RMSE in the distance has a linear relationship for both [6] and our algorithm and that our algorithm yields more accurate results as the error in the distance estimates is increased.

Under the same scenario we have also examined the worst case performance of each algorithm — that is the maximum distance between the final position estimate and the actual position (see Figure 3). This can be thought of as an upper-bound on the error levels for the algorithms. So, for example if the RMSE in the distance is 3 meters on average, we can expect an accuracy of our algorithm of 3.4 meters with the worst case performance of 6 meters. Whereas, the *robust quads* algorithm yields an average and maximum error of 4.2 meters and 7 meters respectively.

C. Vehicular Environment Study

In this experiment, we study the performance of our algorithm in a simulated highway. We have used a microscopic transportation simulator CORSIM (CORridor SIMulator) developed by the US Federal Highway Administration to model vehicle movement. We modeled a 4 km road with 3 east

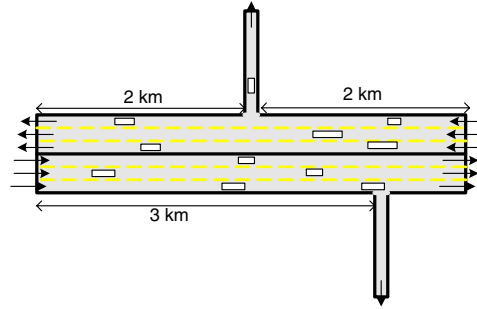


Fig. 4. Roadway for simulated vehicular environment

bound and 3 west bound lanes with vehicles entering into the east and west end of system at the rate of 1200 vehicles per hour depicted in Figure 4. The speed limit for the road was set to 60 mph (97 km/h).

For the experiment, we have used GPS to act as an initial position estimate for phase 1 of our algorithm, then in subsequent iterations the final position estimate from the previous time slot, plus a correction for the vehicle’s movement was used. The GPS position was set to differ from the true position by a Gaussian distributed random variable with standard deviation of 6 meters, for consistency with real GPS error levels of 3 to 10 meters [3].

Figure 5 shows the results of how well a single vehicle was able to localize its neighbours at each time step as it traveled through the system. We have assumed that each vehicle has a communication range of 150 meters; resulting in each vehicle having between 8 and 14 neighbours at each time step of the simulation. The inter-vehicle distance estimates were set to deviate from the actual values according to a Gaussian distribution with the standard deviation of 3 meters. Also, we have used the optimization outlined in (3) to take advantage of road and velocity constraints. In Figure 5, the median error in the final position of 2.7 meters is also shown, which is a relatively large improvement over the initial GPS estimates that were set to have an average error of 6 meters. Examining Figure 5 notice that there are a few spikes in the error levels. These spikes are mainly due to the poor estimate of a single vehicle within the cluster. The poor estimate is the result of two factors: the vehicle just entered the communication range and it had a very poor initial position estimate of its position; this problem was then compounded by the topology of the other vehicle’s within the same cluster, at that time instant, (i.e. a large set of the vehicles in the cluster were collinear making them difficult to mitigate the error in the initial position estimate). However, notice that because our algorithm makes use of the past position estimates and the velocity constraints it quickly recovers from these temporary inaccurate position estimates.

In general, the effect of poor initial estimates alone does not yield a poor final position estimates. We considered a cluster of 9 vehicles where the inter-vehicle distance estimate noise was considered to be Gaussian distributed with different

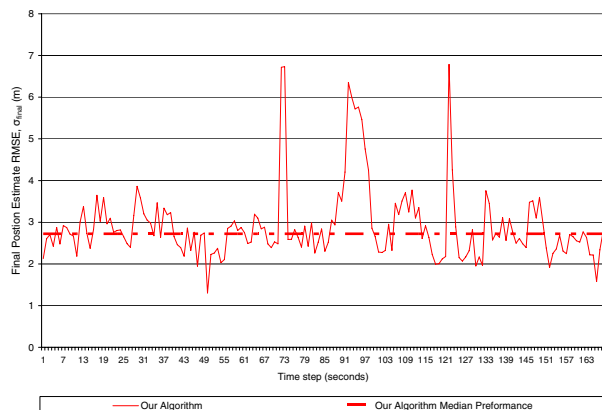


Fig. 5. The curve represent the position accuracy of our algorithm at each time step as the vehicle travels through the road network. The dash line is the median error in the position estimate

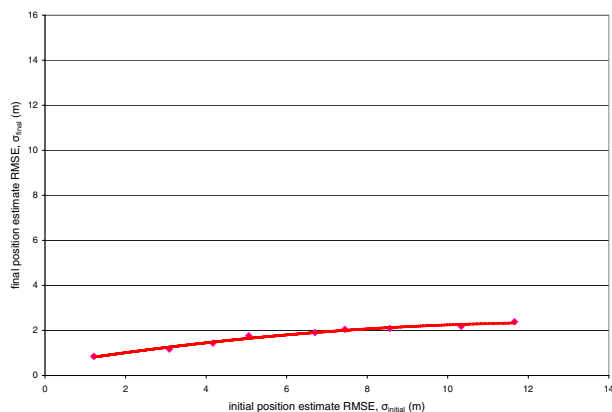


Fig. 6. The average error in the final position estimate as a function of the error in the initial estimates.

standard deviations. Figure 6, shows the results of this experiment. Each data point represents the average results of 30 runs of the experiment to remove the effects of poor network topologies. From Figure 6 note that regardless of the error in the initial position estimates, the error in the final position is still relatively low.

V. CONCLUSION

In the future, it is envisioned that vehicles will be equipped with dedicated short range communication (DSRC) devices — allowing vehicle-to-vehicle and vehicle-to-infrastructure communication. Equipping vehicles with DSRC devices will offer a wide array of new safety and ITS applications. One of the promising new safety applications will be a cooperative vehicle collision warning system; where vehicles cooperatively share information (i.e. location, speed, heading, acceleration, etc.) with their peers and distribute warnings of potential hazards that lie on the road ahead. However, the ultimate success of this system will be dependent on how accurately the relative positioning of neighbouring vehicles can be predicted.

We have presented a novel distributed solution for localization in vehicular networks. We found the majority of existing localization approaches were unfit for vehicle networks, for one or more of the following reasons: they do not account for noise that is present in the inter-node distance measurements; or the node localization cannot be performed in a distributed fashion; or fail with node mobility. However, the *robust quads* algorithm showed promise for use in vehicular networks. So, we used it as a basis for comparison. We proposed a robust min-max localization scheme and showed that our algorithm provides a high level of accuracy in predicting a vehicle position and outperforms the *robust quads* algorithm [6]. Also, that even with high levels of error in initial node position estimates our algorithm converges toward the actual position. Therefore, our algorithm can be made to complement GPS for providing a reliable and accurate vehicle localization system — as is required for future applications in vehicular networks.

REFERENCES

- [1] J. Zhu and S. Roy, "MAC for dedicated short range communications in intelligent transport system," *IEEE Communications*, pp. 60–67, December 2003.
- [2] R. Tatchikou, S. Biswas, and F. Dion, "Cooperative vehicle collision avoidance using inter-vehicle packet forwarding," in *Proceedings of Global Telecommunications Conference, 2005. GLOBECOM '05*. IEEE, November 2005.
- [3] C. Drane and C. Rizos, "Positioning systems in intelligent transportation systems," *Artech House, Boston London*, pp. 230–231, 1997.
- [4] G. M. Djuknic and R. E. Richton., "Geolocation and assisted GPS," *Computer*, vol. 34, no. 2, pp. 123–125, Feb. 2001.
- [5] V. Kukshya, H. Krishnan, and C. Kellum, "Design of a system solution for relative positioning of vehicles using vehicle-to-vehicle radio communications during GPS outages," in *Proceedings of IEEE Vehicular Technology Conference (VTC)*, September 2005.
- [6] D. Moore, J. Leonard, D. Rus, , and S. Teller, "Robust distributed network localization with noisy range measurement," in *Proceedings of The Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2004.
- [7] R. Parker and S. Valaee, "Vehicle localization in vehicular networks," *IEEE Vehicular Technology Conference*, September 2006.
- [8] J. Hightower and G. Borriella, "A survey and taxonomy of location systems for ubiquitous computing," *IEEE Computer*, pp. 57–66, August 2001.
- [9] L. Doherty, L. E. Ghaoui, and K. Pister, "Convex position estimation in wireless sensor networks," in *Proceedings of INFOCOMM*, 2001.
- [10] S. Capkun, M. Hamdi, and J. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *Proceedings of Hawaii Int. Conf. on System Sciences*, January 2001.
- [11] R. Iyengar and B. Sikdar, "Scalable and distributed GPS free positioning for sensor networks," *IEEE Intl. Conference on Communications*, May 2003.
- [12] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proceedings of GLOBECOMM*, November 2001.
- [13] J. R. C. Savarese and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proceedings of USENIX Technical Annual Conference*, June 2002.
- [14] T. Eren, D. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation and randomization of network localization," in *Proceedings of IEEE INFOCOM '04*, April 2004.
- [15] C.-I. Chand and L. Davisson, "Two iterative algorithms for finding minimax solutions," *IEEE Transactions on Information Theory*, vol. 36, 1990.
- [16] R. Brayton, S. Director, G. Hachtel, and L. M. Vidgal, "A new algorithm for statistical circuit design based on quasi-netwon methods and function splitting," *IEEE Transactions on Circuits and Systems*, vol. 9, 1979.