

Applying location estimation for reliable routing in tactical unmanned ground vehicle networks

Preetha Thulasiraman¹ · Grace A. Clark² · Timothy M. Beach³

Received: 16 December 2015 / Accepted: 15 April 2016 / Published online: 7 May 2016
© Springer Science+Business Media New York (outside the USA) 2016

Abstract An unmanned ground vehicle (UGV) network operates as an ad hoc autonomous tactical network. In this paper, we study the deployment of a UGV network and its ability to use location estimation to provide reliable data forwarding. This analysis involves (1) location estimation to obtain realistic estimates of the node's (UGV) spatial positions and (2) real-time path planning to obtain reliable routing among nodes using the estimates of the node positions. Conventional approaches commonly use the popular Random Waypoint (RWP) model (or its variants) as the mobility model. The RWP approach uses randomly selected node positions and has no way to cope with nodes moving in and out of transmission range. These deficiencies cause the route reliability to suffer. Our approach (1) uses an Extended Kalman Filter (EKF) for node mobility/location estimation and (2) uses a modified Ad Hoc On Demand Distance Vector (AODV) algorithm for route selection using a new routing parameter called contact time. We refer to this combined approach as the AODV-LocPred algorithm. We compare the AODV-LocPred algorithm with AODV using the RWP model. This is referred to as AODV-RWP. Our simulations show that the AODV-LocPred algorithm sig-

nificantly outperforms the AODV-RWP algorithm in terms of packet delivery ratio (PDR), because it provides realistic node position estimates and it copes in a reasonable way with nodes moving in and out of transmission range. However, the AODV-LocPred incurs an increase in end-to-end delay. Nonetheless, we expect that in many real-world settings, this tradeoff is acceptable.

Keywords Kalman Filter · Routing · Location estimation · Mobility models · Ground vehicles · Ad hoc network

1 Introduction

In recent years, there has been increasing interest in the design and deployment of unmanned systems. Unmanned systems are equipped with data processing units, sensors and communications systems that are used to execute tactical missions without direct human intervention. Unmanned systems include unmanned aircraft (drones), ground robots and underwater explorers. With the introduction of unmanned systems, the traditional concept of warfare has shifted to a network centric view of military systems. This involves the integration of wireless communication networking and information sharing into tactical military operations.

Communication and information distribution using unmanned aerial vehicles (UAVs) has dominated research in the unmanned systems arena because of the prevalent use of drones in combat operations. However, unmanned ground vehicles/robots (UGV) continue to be important in post-conflict regions, where humanitarian needs exists, including

✉ Preetha Thulasiraman
pthulas1@nps.edu

¹ Naval Postgraduate School, 833 Dyer Road, Monterey, CA 93943, USA

² Grace Clark Signal Sciences, Livermore, CA 94550, USA

³ Lieutenant, United States Navy, Yokosuka, Japan

detection and neutralization of landmines [1, 2]. UGVs are also commonly used for providing information surveillance in areas where combat forces can not enter.

Multiple UGVs form a mobile ad hoc network (MANET). To effectively implement communication protocols among UGVs, the nodes need to know position information. Obtaining knowledge about the UGV position requires effective estimation techniques and is an integral part of data forwarding among the network nodes.

1.1 Motivation and contributions

The problem of position estimation, also referred to as localization, is central in mobile robotics [3], and the need for accurate positional information is crucial to not only reduce the probability of collision among nodes in a UGV network but also to conduct its mission.

The most common navigational aid is the Global Positioning System (GPS). Over the years, GPS receivers have become cheaper, smaller, and can be found in everything from commercial aircraft to cell phones. However, most GPS receivers operate using the Standard Positioning Service (SPS) where it is only possible to reach a 15-meter accuracy in the best case [4]. Military users who have specially equipped receivers and the required cryptographic equipment and keys can use the Precise Positioning Service (PPS), where one meter accuracy or better can be achieved [4]. GPS, however, is prone to errors, and at times is unreliable due to a combination of noise and bias. Sources of error in GPS include: measurements in signal arrival time, numerical calculations, atmospheric effects, and multi-path signals. Furthermore, since GPS signals at the receivers tend to be rather weak, signals can easily be blocked by buildings and other high obstacles, and are vulnerable to jamming; unmanned systems must be able to operate in GPS denied environments [5]

A group of techniques that uses probabilistic and statistical approaches to aid in localization are very common in unmanned systems. The most common localization approaches use recursive Bayesian estimation, where an unknown probability density function is estimated recursively over time using a mathematical process model and incoming measurements of the environment. A variety of algorithms exist for recursive state estimation [6]. Because the observations (i.e., measurements) are nonlinear, we must choose an estimator that will handle nonlinearities. The most commonly-used, workhorse, baseline estimator of the industry is the Extended Kalman Filter (EKF), which exploits a linear approximation to the nonlinear system model. More advanced estimators include the Unscented Kalman Filter (UKF) and the Sequential Monte Carlo (SMC) estimator or Particle Filter (PF), which have performance advantages over the EKF in many circumstances, but

cost additional computational complexity. In addition, several variants of the Particle Filter offer improvements (e.g. the Rao-Blackwellized Particle Filter (RBPF)) [6, 7]. The Particle filter and the EKF tend to be the most commonly used approaches in autonomous systems [4].

In a cooperative multi-UGV system, reliable and effective communication is critical for enhancing the network's performance. A common assumption in research on multi-UGV systems is that a fully connected network is continuously available and that it is always possible for information from one node to reach another. This assumption, though, is not always realistic in all environments. Missed measurements are also possible in a dynamic network. In those applications where more measurements are needed for better estimates, a missed measurement could cause sub-optimal performance.

As discussed above, due to the errors in measurement that GPS can introduce, measurements can be obtained more accurately and with less time consuming computations when using received signal strength indicator (RSSI) transmitters at fixed position base stations [8, 9]. RSSI is the measure of the power level/signal strength received at the node from a given fixed base station. In this paper, we use RSSI signals from known fixed position base stations to triangulate the UGV node. The use of such fixed RSSI transmitters is appropriate for the mission setting (humanitarian applications, land mine detections, as stated earlier) in which UGV ad hoc networks are deployed.

For the application (UGV ad hoc network) discussed in this paper, we choose the EKF as the state estimator. We do so for the following reasons: (1) The main focus and contribution of this work is not the state estimator (EKF), but rather the new modified Ad Hoc On Demand Vector (AODV) Location Prediction (AODV-LocPred) algorithm that is developed in this paper. The conventional networking literature uses the popular Random Waypoint (RWP) Model (or its variants) for the node dynamics, coupled with the AODV routing protocol. We refer to this as the AODV-RWP algorithm. The RWP model uses randomly-selected node positions, which are not realistic for our operational scenario, as our UGVs move according to their dynamics. In addition, the AODV-RWP protocol has no way to cope with nodes moving in and out of transmission range. These deficiencies cause the route reliability to suffer. The contribution of our work lies in using the *combination* of the state-space dynamic model and state estimator (i.e., EKF) coupled with the new AODV-LocPred routing protocol. We show that this novel *combination* performs well against AODV-RWP in terms of packet delivery ratio (PDR), but there is a tradeoff in end to end delay for the AODV-LocPred algorithm compared with that of the standard AODV-RWP algorithm. We assert that the performance difference between the use of the EKF versus the PF is

much smaller than the performance difference between the AODV-RWP and the AODV-LocPred algorithms. In other words, the value added by using a state-space model and state estimator rather than the RWP model far exceeds the relative performance advantages of the PF over the EKF; (2) Our future planned work includes exploring the relative advantages of the UF and PF over the baseline EKF for this application.

The focus of this paper is to demonstrate the use of the EKF for UGV position estimation in the implementation of a data forwarding protocol based on the AODV routing algorithm. We develop a modified AODV algorithm that adopts the EKF derived position estimates to find data forwarding paths between nodes using contact time (defined in Section 4) as a routing metric that quantifies link reliability. We refer to this algorithm as AODV-LocPred. We provide evaluation of the algorithm using OPNET and MATLAB. To the best of our knowledge, this is the first work that integrates the use of signal processing and control techniques for location estimation in a wireless network of UGV nodes.

2 Related work

Routing using position estimation has been studied in various networks, ranging from vehicular networks to sensor networks.

Zaidi et al. [10] study ad hoc networks with intermittent connectivity. The algorithm for mobility tracking developed in [10] uses RSSI measurements from neighboring nodes modeled as a linear system driven by a discrete semi-Markov process in combination with an efficient averaging filter and an EKF. The proposed algorithm allows robust mobility tracking in ad hoc networks using RSSI measurements. Root Mean Square Error (RMSE) is used as the performance measure. The algorithm is able to follow mobile trajectories accurately over a wide range of parameter values.

Yang et al. [11] consider a SMC method for joint mobility tracking and cellular handoff in wireless communication networks. The mobility tracking is based on the measurement of RSSI signals from known fixed position base stations. The system dynamics are described by a nonlinear state space model. The mobility tracking includes estimation of the position and velocity of the mobile node. The EKF is identified as the main technique for solving online estimation in a nonlinear dynamic system. Using the SMC framework, Yang et al. jointly solve the problem of online estimation and online prediction of RSSI at some future time instance. The SMC was compared with the modified EKF and was shown to improve tracking accuracy and minimize the tradeoff between quality of service (QoS) and resource utilization [12]. However, the SMC-based

approach comes with a significantly high computational cost.

Mihaylova et al. [13] also consider a SMC technique for mobility tracking in wireless communication networks by means of RSSI signals from known fixed position base stations. The technique allows for estimation of mobile position, velocity and acceleration. A PF and RBPF are proposed and analyzed over real world and simulated data. A comparison with an EKF is performed with respect to accuracy and computational complexity in scenarios with abrupt maneuvers. Advantages of the RBPF compared with the PF are decreased computational complexity exhibiting similar accuracy and smaller peak-dynamic errors during abrupt maneuvers, which is important for practical purposes [14]. In this paper, we adapt the state space mobility model from [13].

There has been extensive study in the literature on routing and data forwarding mechanisms for MANETs. In the realm of unmanned systems, routing protocols have been primarily focused on UAVs [15–17]. Algorithms and performance metrics used in MANETs have been co-opted for use in unmanned systems. Recently, routing protocols have been developed for highly mobile aerial networks using a combination of AODV and greedy algorithms. The Reactive Greedy Reactive (RGR) algorithm [18–20] is one such algorithm that was developed for use in networks of unmanned aerial vehicles with speeds up to 60 m/s.

With mobile ground vehicles/robots, the closest network paradigm is that of vehicular ad hoc networks (VANETs). Routing protocols for VANETs have also been studied quite thoroughly in the literature. VANET based routing protocols have been adopted in networks of unmanned robots [21]. Like unmanned aerial vehicle networks, common routing protocols used in MANETs such as AODV and Dynamic Source Routing (DSR), have been modified for use in networks of ground robots [22, 23]. However, there is limited study on how routing protocols for UGV networks adapt to position information. The combination of location estimation using Bayesian methods (EKF) with known dynamic routing protocols has not been studied in the literature in the realm of mobile UGVs.

In this paper we modify AODV for use in UGV networks by incorporating position estimation using EKF to improve route accuracy and reliability. As discussed, we use RSSI signals from fixed position base stations to aid in the localization.

3 Location prediction models

The location estimation algorithm design requires several key problem specifications, or attributes. These are: (1) the operational mission setting and physical constraints, (2) the

set of available sensor measurements or observations, (3) an appropriate physics model, (4) an appropriate performance index or set of performance indices, and (5) an appropriate estimation/tracking algorithm or set of algorithms. We describe and compose an algorithm in terms of these five key attributes.

3.1 UGV network topology

The network environment considered in this paper is shown in Fig. 1. Spatially distributed UGV clusters are assumed to be connected via an unmanned aerial vehicle (UAV) which can act as a relay node to carry information among groups of UGV clusters. This is a common network topology for unmanned systems [2]. No direct communication exists among nodes of different clusters. UGVs within a cluster communicate cooperatively to forward messages from source to destination within one cluster. This paper explores position estimation within one individual UGV cluster, thus we exclude the role of the UAV when developing our algorithms. The purpose of the position estimation algorithm is to produce estimates of position, velocity and acceleration versus time for a particular node or nodes in the cluster.

3.2 UGV dynamic model

We use a discrete time nonlinear state space model for the UGV dynamics [6, 10, 24–26]. Specifically, we use a discrete-time variant of the Singer model originally proposed in [27]. Mihaylova et al. [13] has shown that the

modified Springer model used by Yang and Wang [11] performs well, is simple, and allows efficient computation of performance indices. This is a Gauss-Markov type model modified to include a discrete semi-Markov model for the input command function. The state propagation model for the mobile node is linear, but the measurement model is highly nonlinear.

3.2.1 Model for the state of the mobile node

Let the two-dimensional spatial coordinates be denoted by (x, y) . Let k denote the discrete time index, and let T denote the temporal sampling period. We let (x_k, y_k) denote the position of the mobile node at discrete time k . We then denote the speed by (\dot{x}_k, \dot{y}_k) and the acceleration by (\ddot{x}_k, \ddot{y}_k) . The state of the mobile node at discrete time k is then denoted by $\underline{x}_k = [x_k, \dot{x}_k, \ddot{x}_k, y_k, \dot{y}_k, \ddot{y}_k]^T$ where the superscript T denotes vector transpose. The linear state for the mobile node is given by:

$$\underline{x}_k = A(T, \alpha)\underline{x}_{k-1} + B_u(T)\underline{u}_k + B_w(T)\underline{w}_k \quad (1)$$

where $\underline{u}_k = [u_{x,k}, u_{y,k}]^T$ is the discrete time command process, or system input and $\underline{w}_k = [w_{x,k}, w_{y,k}]^T$ is a white Gaussian noise vector with zero mean and covariance matrix $Q = \sigma_w^2 I$ where I denotes the unit or identity matrix and σ_w^2 is the noise covariance. The parameter α depends on the duration of a maneuver, and is the reciprocal of the maneuver time constant. Note that the matrix $A(T, \alpha)$ is a function only of the sampling period and the reciprocal of

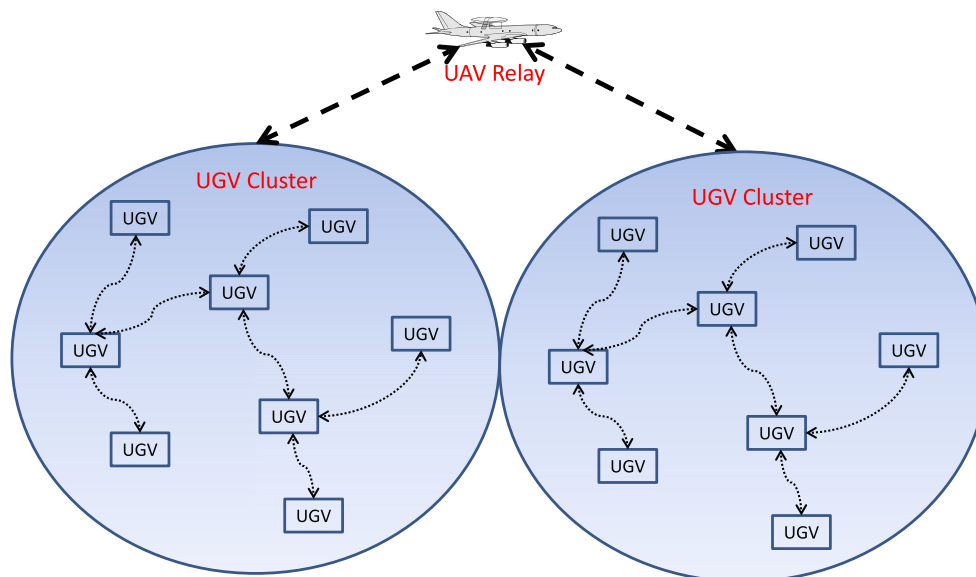


Fig. 1 The UGV network topology consists of clusters of UGVs. UGVs communicate with one another within one cluster. Inter-cluster communication is performed via the UAV relay

the maneuver constant, and the matrices $B_u(T)$ and $B_w(T)$ are functions only of the sampling period [13].

In the real world, a mobile node is likely to have both discontinuous motion and continuous motion. A mobile node is likely to have sudden and unexpected acceleration changes. Simultaneously, we must account for the fact that node acceleration is likely to be correlated over time, due to momentum. For these reasons, we model the mobile node as a dynamic system driven by a semi-Markov acceleration process $a_k = u_k + r_k$. This acceleration is the sum of a two-dimensional semi-Markov driving command $u_k = [u_{x,k}, u_{y,k}]^T$ and a two-dimensional time-correlated random acceleration vector $r_k = [r_{x,k}, r_{y,k}]^T$. Both $u_{x,k}$ and $u_{y,k}$ are independent semi-Markov processes acting in the x and y directions [10, 27, 28]. The model produces correlated random accelerations using a representative model of the autocorrelation function given by [28].

$$R_{rr}(\tau) = E\{\underline{r}(t)\underline{r}^T(t + \tau)\} = \sigma_m^2 e^{-\alpha|\tau|I} \tag{2}$$

where $\alpha \geq 0$, σ_m^2 is the variance of the random acceleration in a single dimension, and α is the reciprocal of the random acceleration time constant. The final mobile node dynamic model is summarized in Eq. 3.

$$\begin{pmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \\ y_k \\ \dot{y}_k \\ \ddot{y}_k \end{pmatrix} = \begin{pmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & \alpha \end{pmatrix} \begin{pmatrix} x_{k-1} \\ \dot{x}_{k-1} \\ \ddot{x}_{k-1} \\ y_{k-1} \\ \dot{y}_{k-1} \\ \ddot{y}_{k-1} \end{pmatrix} + \begin{pmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_{x,k} \\ u_{y,k} \end{pmatrix} + \begin{pmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 1 & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} w_{x,k} \\ w_{y,k} \end{pmatrix} \tag{3}$$

3.3 Measurement (Observation) model

The measurements consist of RSSI signals from known-location base stations. Locating a node in a two-dimensional spatial plane requires a minimum of three base stations. Increasing the number of base stations to seven will improve accuracy [13]. Let M denote the number of base stations. We are given measurements of the location $(a_{i,k}, b_{i,k})$ of each of the base stations at discrete time k , where $i = 1, \dots, M$. Let us denote the measurement model by a nonlinear vector equation of the form:

$$z_k = \underline{h}[x_k] + v_k \tag{4}$$

where z_k denotes the measurement vector, $\underline{h}[x_k]$ is a non-linear function, and v_k is the measurement noise. The RSSI signal can be modeled as a sum of two terms: path loss ($\underline{h}[x_k]$) and shadow fading (v_k). The RSSI (measured in dB) signal of a single BS is modeled by:

$$z_{k,i} = z_{0,i} - 10\eta \log_{10}(d_{k,i}[x_k]) + v_{k,i} \tag{5}$$

where $z_{0,i}$ is a constant characterizing the transmission power of the base station. It is a function of wavelength, antenna, height and gain of node i . The constant η is called the slope index, and it takes on various values, depending on the characteristics of the physical environment (i.e., $\eta=2$ for highways, or 4 for microcells in a city). The distance $d_{k,i}[x_k]$ is the distance between the mobile node and the base station i at a discrete time k . The process $v_k = [v_{k,1}, \dots, v_{k,M_{BS}}]$ is the shadowing component.

The state estimation equations for the EKF are summarized in [6, 12, 24, 26]. The EKF requires a Jacobian, or gradient matrix H for approximate linearization of our non-linear RSSI measurement equation is given in Eq. 6.

$$H \triangleq \left. \frac{\partial \underline{h}[x_k]}{\partial x_k} \right|_{x_k = \hat{x}_{k|k-1}} \tag{6}$$

The Jacobian matrix derived for our model is given in [12].

3.4 Performance measures for the state estimator

The performance of the mobility estimator is evaluated using the following performance indices.

- Zero-Mean Test on the Innovations: When the EKF is tuned, it provides a mean squared error (MSE) estimate of the state vector. When the EKF is properly tuned, the innovations sequence is zero-mean and white [25]. We therefore use hypothesis testing to ascertain that the innovations have zero mean within a 95 % confidence interval, or “two-sigma bounds” [12, 24–26].
- Whiteness Test in the Innovations: We use another hypothesis test to declare whether or not the innovations are white. The test statistic is the normalized sample auto-covariance function of an individual innovations sequence. Ideally, if the innovations are white, then the auto-covariance should be a Kronecker delta function. A 95 % confidence interval on the covariance values is computed. If fewer than 5 % of the $N - 1$ samples in the covariance sequence that follow the sample at zero lag lie within the confidence interval, then the innovations are declared to be white [12, 24–26].

- Root Mean Square Error (RMSE): The error in the states is given by $\tilde{x}(t) = x(t) - \hat{x}(t|t-1)$. We compute and plot this error for each of the states as a measure of performance. The mean is estimated by averaging over 100 realizations of the stochastic process.
- Weighted Sum Squared Residuals (WSSR): The WSSR provides a method for whiteness testing over all of the innovations by aggregating innovations vector information into a single scalar test statistic. We use a hypothesis test to ascertain that the WSSR lies below a pre-determined threshold [12, 24–26].
- Posterior Cramer-Rao Lower Bound (PCRLB): The PCRLB provides a lower bound on the state error covariance matrix. This tells us the best possible theoretical covariance that is attainable under the assumptions used to derive the estimator [7, 12, 13, 25, 26].

4 AODV using location prediction (AODV-LocPred) for UGV networks

We consider a UGV network as a mobile wireless network that uses multihop routing. There are N nodes, each with a transmission range R and L communication links in the network. Link (i, j) is connected at time k when the distance between nodes i and j at time k is less than or equal to the transmission range, R . In other words, the Euclidean distance between two nodes must be less than the transmission range. Otherwise, link (i, j) is broken/disconnected because the two nodes are out of communication range with each other. Packets from a given source, s , to a destination, d , are delivered via multiple hops. There are H possible routes at time k between s and d . To find and maintain routes, each mobile UGV node employs the AODV-LocPred routing algorithm.

4.1 AODV overview

The AODV routing algorithm consists of two operations [29]: route discovery and route maintenance. Route discovery is initiated by a source node that has data to transmit but can not find an active route to a specific destination in its routing table. The source node broadcasts a route request (RREQ) message to neighboring nodes. Each RREQ is associated with a Time to Live (TTL) that limits how many times the requests can be retransmitted. The RREQ is flooded through the entire network until the destination is found or an intermediate node is found that has a valid route to the destination. Note that every node that receives an RREQ stores a reverse route to the source node. When the RREQ is received by either the destination node or an intermediate node with a valid path to the destination, the node

sends a route reply (RREP) message to the neighboring node using the stored reverse route in a unicast manner. A node that receives a RREP sends the message to the source node via the stored reverse route and then updates or creates a forward route to the destination in its routing table.

Traditionally, AODV does not choose paths based on any metric. When RREQs are broadcast by the source node, it is possible that the destination node receives more than one RREQ from the same requestor. If this situation occurs, the destination drops duplicate RREQs immediately. Thus, the path that the destination chooses to unicast back to the source via a RREP is based on the RREQ that was received first.

Route maintenance is performed after route discovery. Route maintenance is used as a mechanism to maintain local connectivity and routes. Nodes initiate route maintenance by sending Hello messages to their neighbors to check if links are connected. If a node does not receive a Hello message within a designated timeframe, the node assumes that the link to the neighbor is currently disconnected. The time frame within which Hello messages should be received is application specific and can be tailored to be fixed or flexible time intervals.

Due to space, we refer the reader to [29] for further details on the operation of AODV.

4.2 AODV with location prediction (AODV-LocPred)

As discussed above, a link between two nodes can exist only if the spatial Euclidean distance between the two nodes is less than or equal to their transmission range. Nodes move in and out of transmission range of each other over time. As was discussed in Section 4.1, the traditional AODV algorithm [29] chooses paths based on the first RREQ that is received from the source. Thus, the traditional AODV algorithm does not make routing decisions based on the logistics or topology of the network. In addition, the mobility model used when implementing the traditional AODV algorithm is the Random Waypoint (RWP) model. The RWP model produces randomly selected node trajectories and assumes that the speed of the node is constant. The node positions are randomly generated and uncorrelated, so the trajectory cannot be reliably predicted. Our AODV-LocPred algorithm improves upon this traditional approach by 1) choosing a path based on contact time (defined in the following paragraphs) to increase route reliability and 2) using the state space model and EKF developed in Section 3.1 to model and estimate the trajectories of the nodes so that the node positions/trajectories are predictable.

To facilitate the discussion of our modified AODV algorithm, we define the following quantities: Let s_i denote the source node indexed by i and let d_j denote the destination

node indexed by j . We also refer to nodes by their indices. We define the transmission range of a node i as R_i . For simplicity, in this paper we assume that all nodes have the same transmission range. In our future work we will relax this constraint. We also define $\underline{p}_i(k) = [x_i(k) y_i(k)]^T$ to be the vector position of node i at time k . We define the route/path between s_i and d_j at time k as $h_{ij}(k)$.

The path $h_{ij}(k)$ may contain multiple hops, depending on the norm or Euclidean distance between i and j . In other words, if $\|\underline{p}_j(k) - \underline{p}_i(k)\| > R_i$, a path will contain multiple hops. Each hop/link along a path is stable as long as the two nodes are within range of each other. Otherwise, the link is broken and cannot be used. The longer two nodes stay within transmission range of each other, the more reliable the link is. Thus, our approach to improving link reliability is to allow the AODV-LocPred algorithm to choose the next hop for a path based on how long a node pair has been in contact. The node pair with the longest contact time at time k is chosen as the next hop. To this end, we define a new routing parameter $c_{uv}(k)$ which we call contact time, where (uv) denotes the node pair. First, let $c_H(k)$ be a Hello counter at a time instant k defined as follows:

$$c_H(k) = \begin{cases} 1, & \text{if a Hello message is transmitted and received} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

We define k_{in} as the time at which the two nodes are first within range and k_{out} is defined as the time at which they go out of range. At any given time k , we cannot know k_{out} ; k_{out} is not predictable, as the node trajectories are dynamic and it is not possible to predict very far into the future. Rather than try to estimate k_{out} , we assume that in general, a link that has been in contact for a few time samples is more likely to be stable than one that has not. To this end, we count the number of Hello messages over the range k_{in} to k by computing the contact time between two nodes u and v at time k , defined as follows:

$$c_{uv}(k) \triangleq c_{uv}(k-1) + c_H(k) \Big|_{k=k_{in}}^k \quad \text{where } c_H(k) = c_{uv}(k) = 0 \text{ for } k \leq 0 \quad (8)$$

At time k , the AODV-LocPred algorithm then chooses the next hop to be the one with the largest contact time $c_{uv}(k)$.

Figure 2 illustrates an example of the contact time signal. In this example, $k_{in} = 5$ msec and $k_{out} = 10$ msec. We see that at k_{out} and afterward, $c_{uv}(k)$ stays constant, implying that new Hello messages are not being received. This, in

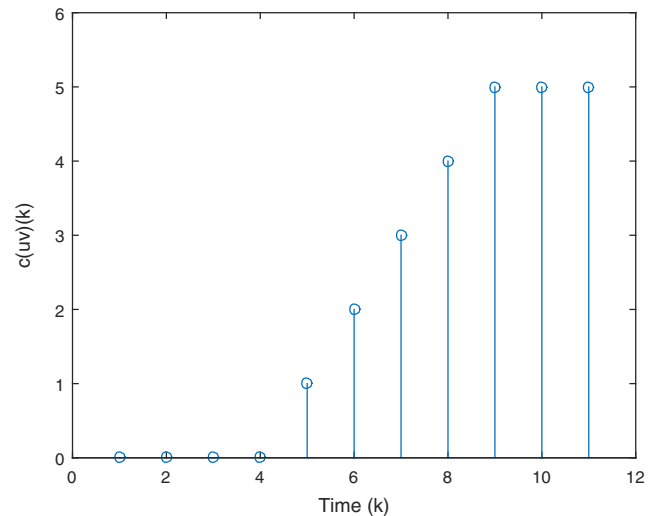


Fig. 2 Example of the contact time signal where $k_{in} = 5$ msec and $k_{out} = 10$ msec

turn, implies that the nodes have moved out of range and the link has been broken.

4.2.1 AODV-LocPred algorithm description

When data packets arrive at a node, the source node attempts to find an active route associated with the corresponding destination in its routing table, as was discussed with AODV in Section 4.1. If no active route exists, the source node initiates a route discovery process by broadcasting a RREQ message. Once a route from s_i and d_j is discovered, each node along the path begins to broadcast Hello messages to ensure local connectivity. In order to measure and retain the contact time values, we modify the routing table of each node to include a contact time entry. An example routing table and its corresponding network graph is shown in Fig. 3. In this example, i has paths to two destination nodes, j and z . The path from i to j is $i - u - v - j$ and the path from i to z is $i - x - y - z$. As shown in Fig. 3, node i 's routing table will have an entry designated to destination nodes j and z . It also contains the next hop node. The next hop node is the node that i needs to forward data to in order for the data to reach the appropriate destination. In this case, the next hops for destinations j and z are nodes u and x , respectively. The third entry of the routing table is the contact time calculated by Eq. 8. If Hello messages are not received after two time samples, node i removes the entry pertaining to destination j from its routing table. More specifically, a lack of Hello messages indicates that nodes i and u are 1) no longer in transmission range of each other or 2) the link is so weak that Hello messages are getting dropped during transmission. In either case, the reliability of

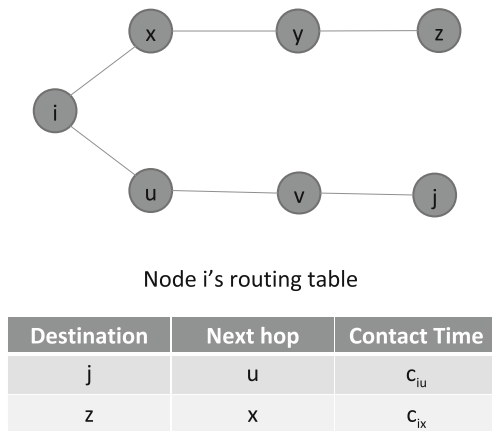


Fig. 3 Example network showing how node i 's routing table, including the new contact time entry is constructed

link is poor. As a result, node i no longer has an active route to j . The number of time samples that the node waits before expunging the obsolete entry from its routing table is a user specific/defined parameter. Due to the dynamic nature of the UGV network, we wait only two samples.

In this case, rather than node i initiating a new route discovery process, it will forward the packets to its neighbor with which it has the greatest contact time as indicated in the routing table. In our example in Fig. 3, if link (i, u) is no longer available, node i 's only option is to forward to node x . The next hop node that receives the data from i (in this case node x) will then go through the same process of looking at its routing table contact times to decide the next forwarding node. By following this procedure, we achieve two outcomes: 1) reduce the number of RREQs and RREPs sent by avoiding a new route discovery process when a link is broken and 2) increase route reliability by choosing links that are more stable.

When more than one forwarding node exists for a particular destination, the node that has the longest contact time will be chosen. For example, suppose in Fig. 3, node i had two forwarding options to node j : node u or node x (suppose a link exists between x and u). The forwarding node that is ultimately chosen will be based on which contact time is larger, $c_{iu}(k)$ or $c_{ix}(k)$. A larger contact time is indicative of a more stable link. If $c_{ix}(k) > c_{iu}(k)$, then x is recorded as the next hop node in the routing table. Even though choosing x results in a longer path from i to j (4 hops vs 3 hops), the overall reliability will be increased. It must be noted that all nodes follow the same process. For example, node x will measure the contact time between itself, u and y and maintain its routing table accordingly.

5 Performance evaluation

In this section, we present the simulation results conducted in Matlab and OPNET to demonstrate and validate 1) the EKF algorithm and 2) AODV with location prediction (AODV-LocPred).

5.1 Evaluation of state estimator performance

To evaluate the effectiveness of the state estimator, we simulate a single mobile node traveling along a trajectory that includes abrupt maneuvers. We use a Gauss-Markov state space model for the node dynamics. Process noise is assumed to be zero. The measurements are constant power RSSI signals transmitted from three fixed position base stations. We use the EKF described in Section 3.1 for state estimation, including node position coordinates in a two-dimensional spatial grid environment.

The simulation parameters used in Matlab are as follows: The discretization time step (T) is 0.5 secs, correlation coefficient (α) is 0.6, path loss index (η) is 3, base station transmission power ($z_{0,i}$) is 90, covariance (σ_w^2) of the noise w_k is $0.5^2[m/s^2]^2$, covariance (σ_v^2) of the noise $v_{i,k}$ is $4^2[db]^2$, the maximum speed (V_{max}) is 45 m/s, the transition probabilities ($p_{i,j}$) are set to 0.8, and the initial mode probabilities $\mu_{i,0}$ are $1/M, i = 1, \dots, M, M = 5$, where M is the number of base stations. The parameters are chosen such that node behavior is realistic. In other words, abrupt maneuvers are included to test the estimator's ability to adapt to rapid trajectory changes.

The EKF initial state estimate is set to:

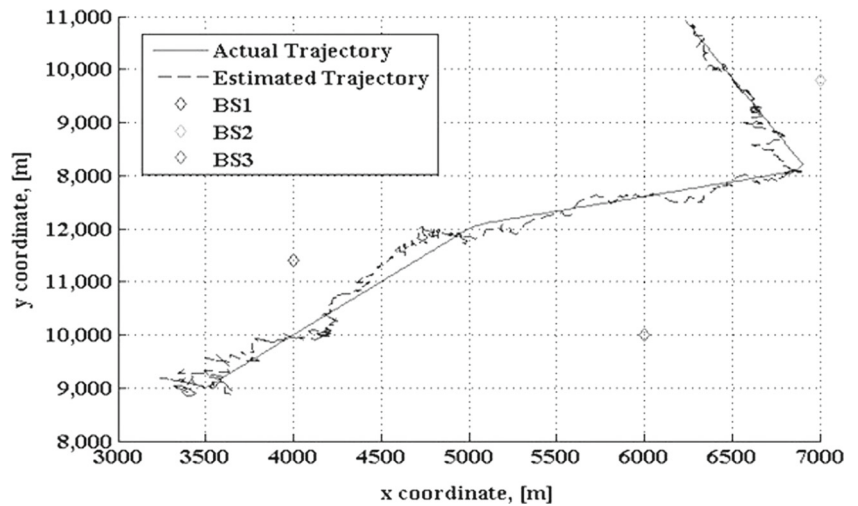
$$\hat{\underline{x}}(0 | 0) = (3400 \ 5 \ 0 \ 8700 \ 8 \ 0)^T \quad (9)$$

The EKF initial covariance estimate is set to:

$$\tilde{\underline{P}}(0 | 0) = \begin{pmatrix} 400^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 400^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5^2 \end{pmatrix} \quad (10)$$

The initial state estimate was chosen based on the assumption that we have reasonable a priori knowledge of the initial states of the UGV node because we deploy the node ourselves. The initial covariance estimate is chosen based on the estimate we make on the standard deviations of the node states based on our knowledge of the operational environment and node capabilities.

Fig. 4 Estimated track, simulated track, and locations of base stations transmitting RSSI signals used for triangulation of the UGV node



5.1.1 State Estimation Performance

The discrete-time input command process $u_{x,k}$ and $u_{y,k}$ is chosen to be a first order Markov chain that can take on discrete values -3.5, 0 and 3.5 in units of m/s^2 . The command input includes two turns, once at 150 secs and once at 200 secs. Please see [12] for further details.

A plot of the estimated track $\hat{x}_{k|k-1}$ from the EKF overlaid on a plot of the actual trajectory to include base stations used for triangulation is shown in Fig. 4. After

the initial track errors during the transient state, the estimation settles into a trajectory that tracks closely to the actual trajectory. A plot of the estimated root mean speed $\hat{x}_k = \sqrt{\hat{x}_{2,k}^2 + \hat{x}_{5,k}^2}$ overlaid on a plot of the actual root mean speed $\dot{x}_k = \sqrt{x_{2,k}^2 + x_{5,k}^2}$ is illustrated in Fig. 5. The initial velocity errors settle after about 40 seconds of transient behavior and closely track the true velocity [12].

The error between the estimated and actual states $\tilde{x}_k = x_k - \hat{x}_{k|k-1}$ over time is presented in Fig. 6. The errors \tilde{x}_k

Fig. 5 Speed plots of the UGV node. Top plot: estimated root mean speed $\hat{x}_k = \sqrt{\hat{x}_{2,k}^2 + \hat{x}_{5,k}^2}$ and actual root mean speed $\dot{x}_k = \sqrt{x_{2,k}^2 + x_{5,k}^2}$. Bottom plot: estimated x and y velocity, $\hat{x}_{2,k}$ and $\hat{x}_{5,k}$ and actual x and y velocity, $x_{2,k}$ and $x_{5,k}$ of the node

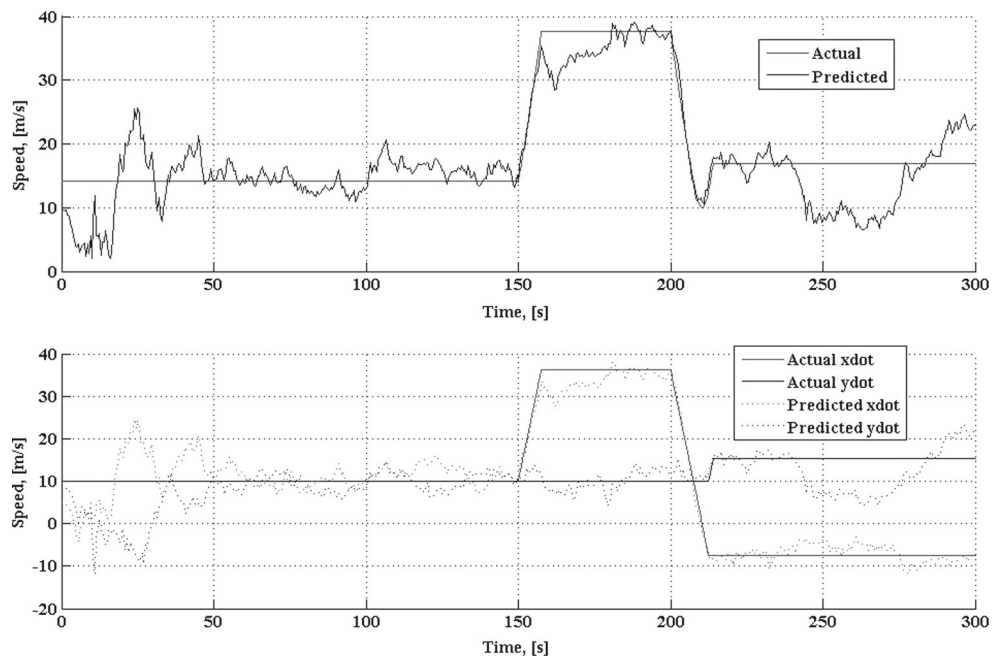
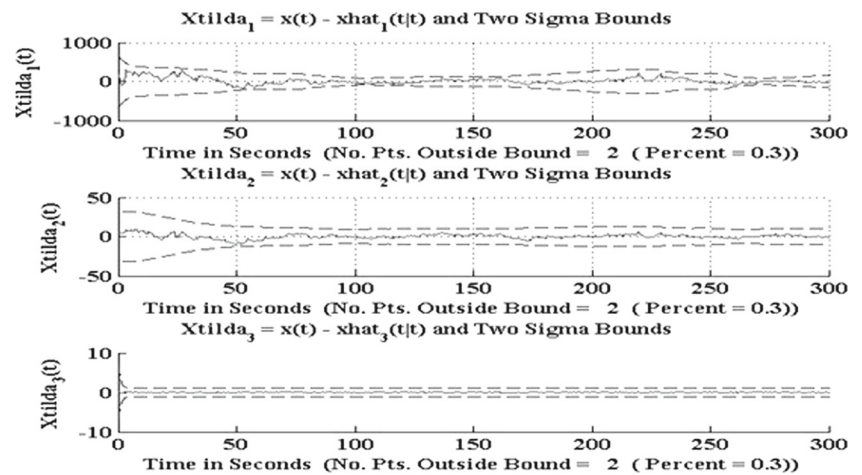


Fig. 6 Error between the estimated states and the actual states $\tilde{x}_k = \hat{x}_k - x_k$ of the UGV nodes and their respective two sigma bounds plotted over time. The top plot corresponds to position, the middle plot corresponds to velocity, and the bottom plot corresponds to acceleration of the UGV node



are shown to be acceptable in that they are zero-mean and lie within the two sigma-bounds at three tenths of a percent deviation each.

5.1.2 EKF Tuning

The performance indices described earlier were computed for this simulation. The EKF performed well, and was shown to be properly tuned [12]. Note that the convergence speed of the EKF is highly sensitive to the choice of initial conditions. The more prior knowledge one has of the operational environment and node behavior, the better choices one can make for the initial conditions.

The innovations were tested and shown to satisfy the zero-mean and whiteness tests. The position and velocity RMSE criteria were computed and compared with the position and velocity PCRLB curves. It was shown that the EKF estimates converged nicely toward the bounds with acceptable offsets from the bounds. The aggregate behavior of the vector of innovations was examined using the hypothesis test on the WSSR [12]. It was shown that the WSSR never exceeded the WSSR threshold. Because the zero-mean test on the innovations, the whiteness test on the innovations and the WSSR test were all satisfied, we declared that the EKF is tuned, and the overall performance is acceptable [12, 24, 25].

5.2 Performance Evaluation of AODV-LocPred

To evaluate the AODV-LocPred algorithm we use OPNET. OPNET, a C++ based simulation platform, can be controlled by Matlab [30]. Essentially, we integrate OPNET with MATLAB by using the MX interface provided by MATLAB which allows C programs to call functions developed in MATLAB. Thus, we can use the location estimation algorithm, developed in Matlab as input to the routing algorithm.

For further information on the integration of MATLAB and OPNET, we refer the reader to [30].

Two performance metrics are used for evaluation:

- Packet delivery ratio (PDR): the ratio of the number of packets received successfully to the number of packets sent ($\frac{\sum \text{packets received}}{\sum \text{packets sent}}$). The greater the PDR the better the performance of the routing protocol. This is used as a measure to analyze path reliability.
- End to end (E2E) delay: the average time taken by a data packet to arrive at the destination. We do not include the delay associated with the route discovery process (time to send and receive RREQ and RREP packets) as part of the delay calculation.

5.2.1 Experiment Design

In a general operational scenario, the mobile network would involve multiple nodes, each moving about the spatial setting on its own trajectory. Those trajectories could cross and nodes could possibly collide. A full simulation of this scenario would be a complex and large undertaking. The purpose of this research paper is to propose and study algorithms to measure network reliability at each time instant; and then use that knowledge to adapt the network to ensure that communication routes are maintained, even when nodes move outside their transmission range. With this in mind, we scale down and simplify the network topology in order to focus on network adaptability. This simplification is discussed in subsequent sections.

The traffic that is routed has a constant bit rate (CBR). The transport protocol used is the User Datagram Protocol (UDP) with a transmission rate of 4 packets/sec, where each packet is 512 bytes. We use the AODV-RWP and AODV-LocPred algorithms to find routes between (source node, destination node) pairs and calculate performance.

The route can include the other nodes available in the network, and in general, the number of nodes in the route can become large. Every node in each network is numbered. The size of the simulated spatial setting is 2km x 1.5km. Each node is initialized to have a transmission range of 250m. The speed of the nodes is kept constant at 45 m/s (this is the same speed used in the EKF simulations). The Rician fading model is used to model the effects of signal propagation for line of sight (LOS).

For computing the Average PDR and Average E2E delay, we design Monte Carlo style experiments in which we create ensembles of random variables over which to average. We define the network size as the number of nodes in the network, where $N_{nodes} = 50, 75, 100, 125$. For each N_{nodes} , we use an ensemble of five randomly-selected spatial node distributions, denoted $N_{dist} = 5$. For each N_{dist} , we also create an ensemble of (source node, destination node) pairs denoted by (s_i, d_j) . The number of (s, d) pairs is denoted $N_{pairs} = 10$. In all experiments, we compute PDR and E2E delay for a single (s, d) pair at a time. The way we choose the (s, d) pairs is different for the AODV-RWP algorithm and the AODV-LocPred algorithm because of the differences in their mobility models. We describe these differences as follows:

Performance Calculations for the AODV-RWP Algorithm: For (s, d) pairs indexed by $m = 1, 2, \dots, N_{pairs}$, and for spatial distributions indexed by $n = 1, 2, \dots, N_{dist}$, we (1) Randomly select a (s, d) pair and the associated node positions using the RWP model, and (2) Compute

$PDR(m, n)$ and $E2E\text{delay}(m, n)$. We then compute Average PDR and Average E2E delay by ensemble averaging $PDR(m, n)$ and $E2E\text{delay}(m, n)$ over the indices m and n for $N_{nodes} = 50, 75, 100, 125$.

Performance Calculations for the AODV-LocPred: In this case, the ensemble averaging computations are the same as they are for the AODV-RWP algorithm above, except that the two steps (1) and (2) above are replaced with the following three steps: (1) Choose the source node and its position to be the one used by the EKF to model the node dynamics and estimate the node trajectory. Note that we build only one Gauss-Markov Model and one EKF for this single node that has a single spatial trajectory and use it in all the experiments. This greatly simplifies and scales down the problem; (2) Randomly choose a destination node position from a uniform distribution. We simplify and scale down the network by fixing the position $\underline{p}_j(k)$ of this destination node d_j so the node does not move over time, $\underline{p}_j(k) = \underline{p}_j$; (3) Compute $PDR(m, n)$ and $E2E\text{delay}(m, n)$ as above.

Note that even though the EKF simulates and predicts an entire trajectory for the chosen source node, we do not use the entire trajectory in our network performance analysis for this scaled-down problem. Instead, we use the single estimated node position corresponding to a manually-chosen time sample k_c at which the EKF has converged. The use of a single source node-destination node pair as described above greatly scales down and simplifies the simulation problem. Nonetheless, it maintains the proper network characteristics that allow us to compute the performance indices.

Fig. 7 Average PDR results for networks with 50 nodes

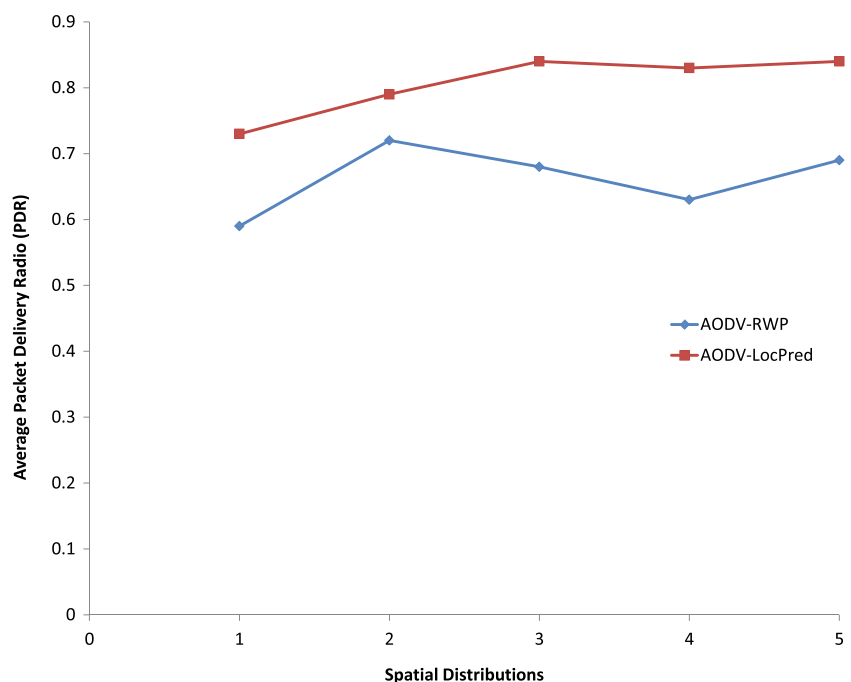
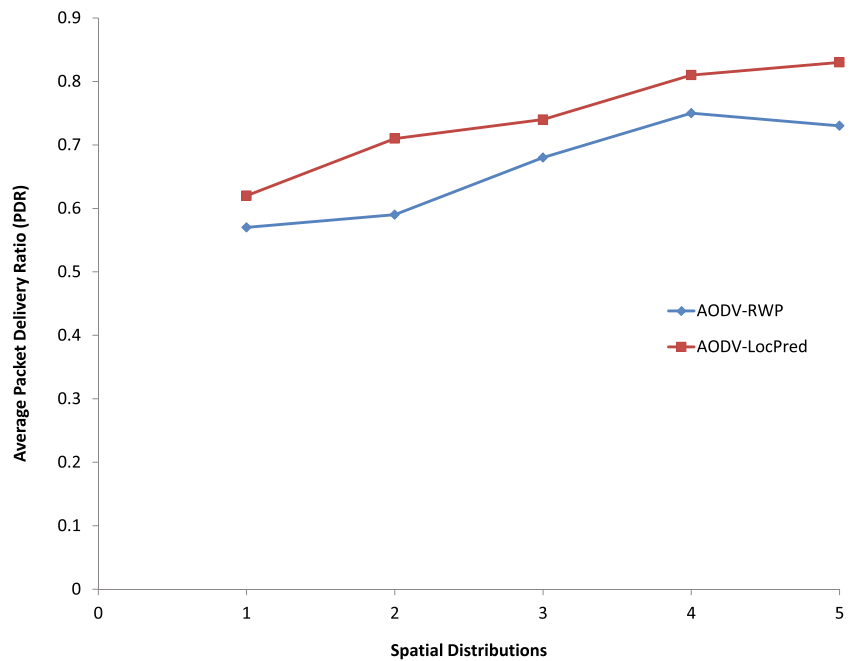


Fig. 8 Average PDR results for networks with 75 nodes

5.2.2 Experimental Performance Analysis

In Figs. 7, 8, 9, and 10, we show the average PDR results for each of 5 spatial distributions for the 4 network sizes. Note that the average is over the 10 source destination pairs per network size (N_{nodes}).

On average, the PDR increases by approximately 8 % when using AODV-LocPred in comparison to AODV-RWP. We can justify this increase in performance as follows: The

EKF used by the AODV-LocPred algorithm models a node's dynamics and provides an estimation of the node's trajectory/position. The location/position information provided by the EKF assists the node in making path (route) choices based on connectivity. The longer a node is connected to its neighbor, the more reliable that link is, thereby increasing packet delivery. We measure the link reliability using contact time based on the location/position information derived from the EKF. The routes chosen using AODV-LocPred use

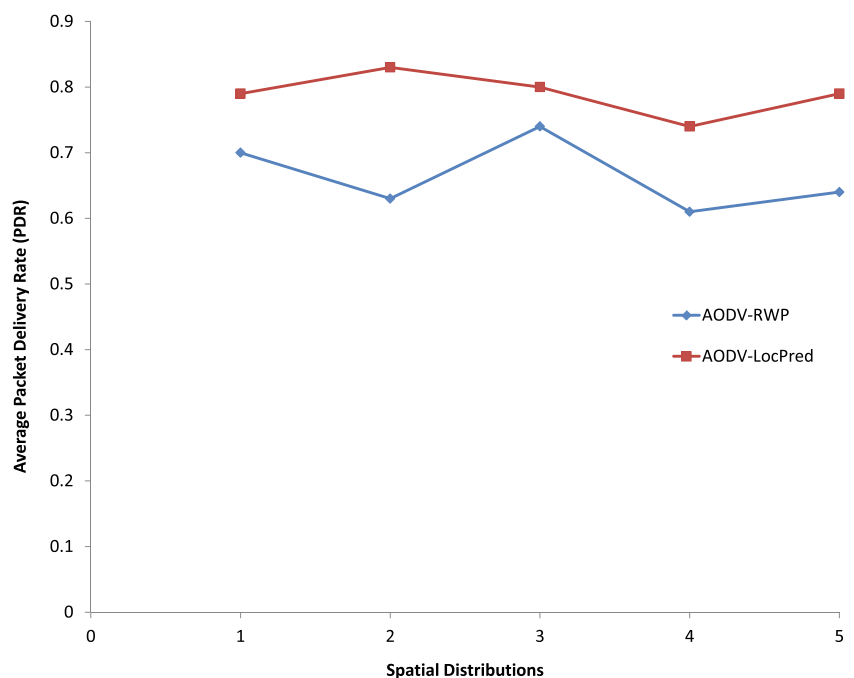
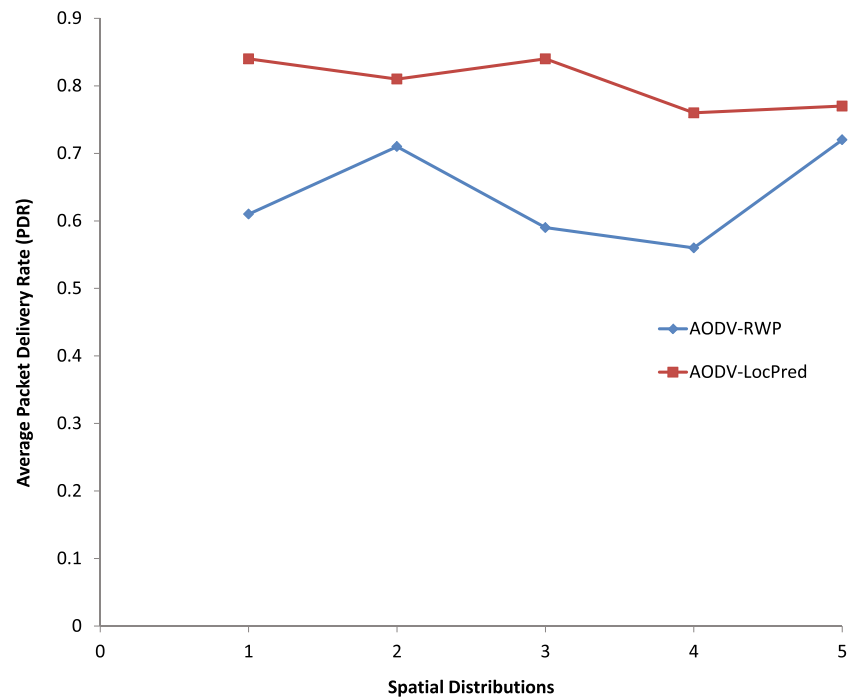
Fig. 9 Average PDR results for networks with 100 nodes

Fig. 10 Average PDR results for networks with 125 nodes



the contact time measurement. If reliable links are chosen for path selection, the PDR (a measure of data transmission reliability) will increase, as it did in our case. Note that the AODV-RWP algorithm chooses paths based on the first arriving RREQ, so the algorithm has no inherent reliability

measure. This leads to poor PDR performance as compared with that of the AODV-LocPred.

We provide a similar set of results for end to end delay for the same source-destination pairs as in the PDR results. The delay results are shown in Figs. 11, 12, 13, and 14. The

Fig. 11 Average end to end delay results for networks with 50 nodes

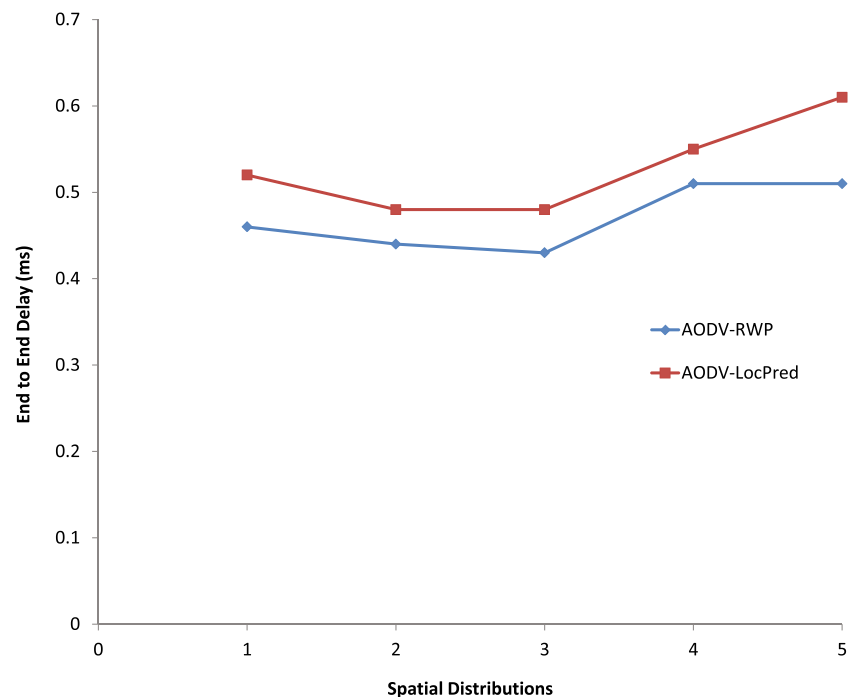
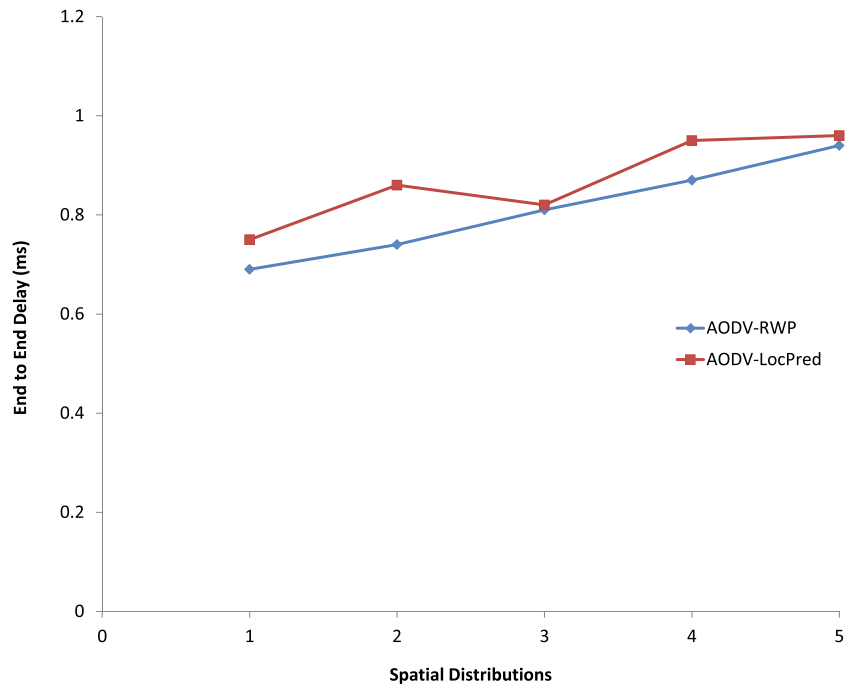


Fig. 12 Average end to end delay results for networks with 75 nodes



delay is measured in milliseconds. We can infer that as network size increases, the paths between a given source and destination will be longer, particularly if they are significantly spatially separated. Thus, if the paths, on average, are longer, then the delay will also increase.

When comparing the AODV-LocPred with the AODV-RWP algorithm, we see that the delay incurred with the

AODV-LocPred is slightly higher. On average, the AODV-RWP has lower delay costs (approximately 5 % lower). As discussed in Section 4.2.1, when using the AODV-LocPred algorithm, it is possible that paths will be longer because the contact time between specific nodes may be better but result in a longer path (more hops). However, in the AODV-RWP case, generally, the RREQ that reaches the destination

Fig. 13 Average end to end delay results for networks with 100 nodes

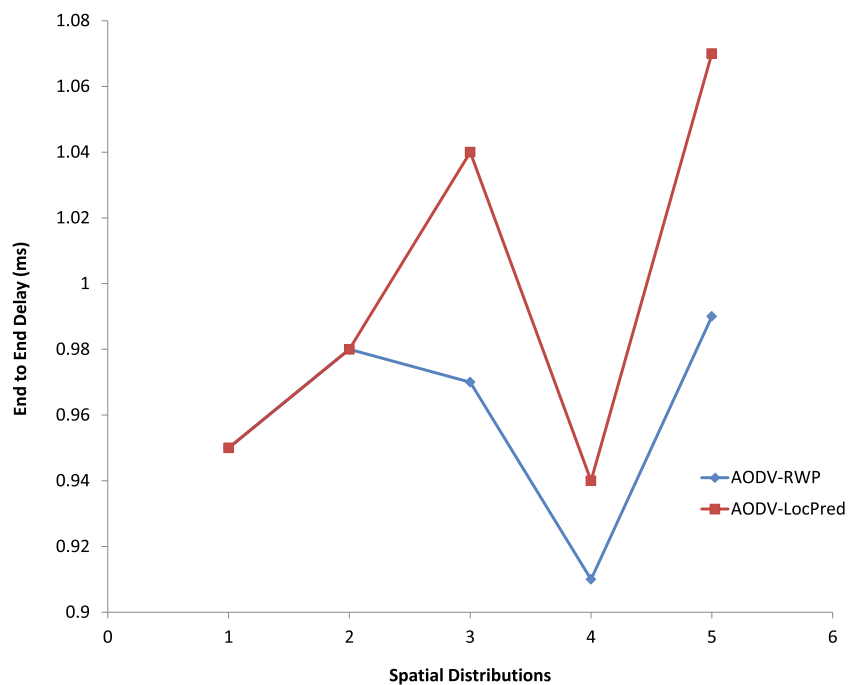
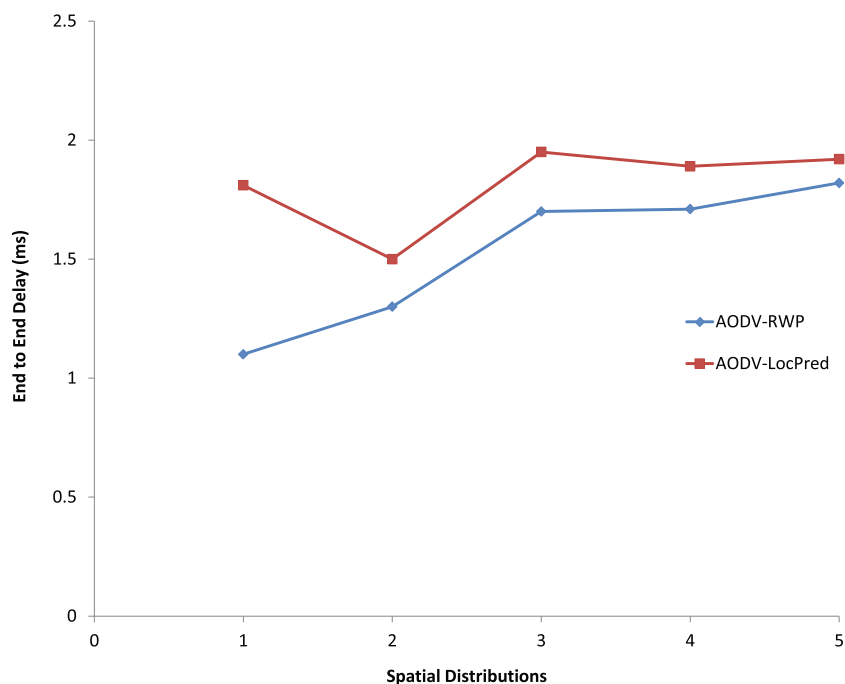


Fig. 14 Average end to end delay results for networks with 125 nodes



node will most likely take the shortest path and thereby the shortest path will be used for subsequent transmissions between the source and destination. This results in a smaller delay. Thus, there is a slight tradeoff between PDR and E2E delay when using the AODV-LocPred algorithm. In an environment that is highly dynamic, like the deployment of a UGV network, the reliability of transmission is more likely a stronger QoS requirement than delay.

6 Conclusion

In this paper we have developed a location estimation algorithm based on an EKF that has been coupled with a modified AODV algorithm, AODV-LocPred, to provide a basis for real time path planning in UGV networks. The EKF algorithm filters recursively, estimating the current state of the UGV node. The storage of additional past information is not required, so storage resource utilization for an individual UGV node is minimized. The AODV-LocPred algorithm is shown to implement efficient position tracking of UGV nodes in a wireless network while providing reliable data forwarding. We have shown that the AODV-LocPred algorithm outperforms the standard AODV algorithm that uses the random waypoint mobility model in terms of PDR but that there is a tradeoff in the end to end delay incurred. In our future work, we plan to study the problem of node collision avoidance when running the EKF and its impact on the

AODV-LocPred algorithm. We plan to relax the fixed position base station constraint and allow the base stations to be mobile. We also plan to study the use of PF and UFs as alternative state estimators and compare their performance with that of the EKF.

References

1. Cantelli L, Mangiameli M, Melita CD, Muscato G (2013) Uav and ugv cooperation for surveying operations in humanitarian demining. In: Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics, pp 1–6
2. Caska S, Gayretli A (2015) An algorithm for collaborative patrolling systems with unmanned air vehicles and unmanned ground vehicles. In: Proceedings of the IEEE International Conference on Recent Advances in Space Technologies, pp 659–663
3. Roumeliotis S, Rekleitis I (2003) Analysis of multirobot localization uncertainty propagation. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp 1763–1770
4. Davis RB (2014) Applying cooperative localization to swarm uavs using an extended kalman filter. Master's thesis, Naval Postgraduate School, CA, USA
5. Manyam SG, Rathinam S, Darbha S, Casbeer D (2014) Routing of two unmanned aerial vehicles with communications constraints. In: Proceedings of the IEEE International Conference on Unmanned Aircraft Systems, pp 140–148
6. Candy JV (2009) Bayesian Signal Processing, Classical, Modern and Particle Filtering Methods. Wiley
7. Ristic B, Arulampalam S, Gordon N (2004) Beyond the Kalman

- Filter: Particle Filters for Tracking Applications. Artech House
8. Papamantou C, Preparata FP, Tamassia R (2008) Algorithms for location estimation based on rssi sampling. In: Algorithmic Aspects of Wireless Sensor Networks. Springer
 9. Fuller R, Koutsoukos XD (eds) (2009) Mobile Entity Localization and Tracking in GPS-less Environments. Springer
 10. Zaidi Z, Mark B (2011) Mobility tracking based on autoregressive models. *IEEE Trans Mob Comput* 10(1):32–43
 11. Yang Z, Wang X (2003) Joint mobility tracking and handoff in cellular networks via sequential monte carlo filtering. *IEEE Trans Signal Process* 51(1):269–281
 12. Beach TM (2013) Mobility modeling and estimation for delay tolerant unmanned ground vehicle networks. Master's thesis, Naval Postgraduate School, CA, USA
 13. Mihaylova L, Angelova D, Honary S, Bull D, Canagarajah C, Ristic B (2007) Mobility tracking in cellular networks using particle filtering. *IEEE Trans Wirel Commun* 6(10):3589–3599
 14. Zaidi Z, Mark B (2003) A mobility tracking model for wireless ad hoc networks. In: Proceedings of IEEE WCNC, pp 1790–1795
 15. Bekmezci I, Sahingoz OK, Temel S (2013) Flying ad hoc networks (fanets): A survey. *Ad Hoc Netw (Elsevier)* 11(3):1254–1270
 16. Sahingoz OK (2014) Networking models in flying ad-hoc networks (fanets): Concepts and challenges. *J Intell Robot Syst (Springer)* 74:513–527
 17. Alshabtat A, Dong L (2011) Low latency routing algorithm for unmanned aerial vehicle ad hoc networks. *World Acad Sci Eng Technol* 5:556–562
 18. Shirani R (2011) Reactive-greedy-reactive in unmanned aeronautical ad hoc networks: A combinatorial routing mechanism. Master's thesis, Carleton University, Ontario, Canada
 19. Medjo Me Biomo J-D, Kunz T, St-Hilaire M (2014) Routing in unmanned aerial ad hoc networks: Introducing a route reliability criterion. In: Proceedings of the IEEE IFIP Wireless and Mobile Networking Conference (WMNC), pp 1–7
 20. Shirani R, St-Hilaire M, Kunz T, Zhou Y, Li J, Lamont L (2012) On the delay of reactive-greedy-reactive routing in unmanned aeronautical ad hoc networks. In: International Conference on Ambient Systems, Networks and Technologies (Elsevier), pp 535–542
 21. Vandenberghe W, Moerman I, Demeester P (2012) Adoption of vehicular ad hoc networking protocols by networked robots. *Wirel Person Commun (Springer)* 64:489–522
 22. Das SM, Hu YC, George Lee CS, Lu Y-H (2007) Mobility aware ad hoc routing protocols for netowrking mobile robot teams. *J Commun Netw* 9(3):296–311
 23. Vu TK, Kwon S (2014) Mobility assisted on demand routing algorithm for manets in the presence of location errors. *The Scientific World Journal (Hindawi)*:1–11
 24. Clark GA (2003) Angular and linear velocity estimation for a re-entry vehicle using six distributed accelerometers: Theory, simulation and feasibility. Technical report, Lawrence Livermore National Laboratory UCRLID- 153253
 25. Candy JV (2006) Model-Based Signal processing IEEE press and wiley interscience
 26. Thulasiraman P, Clark GA, Beach TM (2014) Mobility estimation using an extended kalman filter for unmanned ground vehicle networks. In: Proceedings of IEEE CoGSIMA, pp 223–229
 27. Singer R (1970) Estimating optical tracking filter performance for manned maneuvering targets. *IEEE Trans Aerosp Electron Syst* 6(4):473–483
 28. Jazwinski A (1970) Stochastic processes and filter theory academic press
 29. Perkins C, Belding-Royer E, Das S (2003) Ad hoc on demand distance vector (aodv) routing. Technical Report, RFC 3561
 30. Dham V (2003) Link establishment in ad hoc networks using smart antennas. Master's thesis. Virginia Polytechnic Institute and State University, VA, USA



Dr. Preetha Thulasiraman is an Assistant Professor and Academic Associate for International Students in the Department of Electrical and Computer Engineering at the Naval Postgraduate School (NPS), Monterey, California, USA. She has been at NPS since January 2011. She received her PhD degree from the Electrical and Computer Engineering Department at the University of Waterloo, Waterloo, Ontario, Canada in September 2010, the M.Sc.

degree in Computer Engineering from the University of Arizona, USA in 2006 and the B.Sc. degree in Electrical Engineering from the University of Illinois, Urbana-Champaign, USA in 2004. Since 2006 she has been involved in wireless networking research, studying various algorithmic and protocol design issues for wireless sensor, mesh and hybrid networks. Her current research interests include network layer design of resource allocation algorithms, interference management, wireless routing and fault tolerance, and general applications of graph theory. She is currently pursuing interests in unmanned ad hoc wireless networks with an emphasis on efficient routing, communication and security protocols, mobility management and performance evaluation. She has been a member of IEEE and ACM member since 2006. She has served on the technical program committees of several IEEE and ACM sponsored conferences. She has contributed to over 25 research papers in the area of wireless networks.



Grace A. Clark Ph.D., IEEE Fellow, currently serves as a statistical signal processing consultant via her business, Grace Clark Signal Sciences, in Livermore, CA. For more than three decades, she served as a research scientist at the Lawrence Livermore National Laboratory (LLNL), Livermore, CA. She retired from LLNL in October 2013. She took two sabbaticals from LLNL. From 2010-2012, she served as Visiting Research Professor in the Center for

Cyber Warfare at the Naval Postgraduate School (NPS), Monterey, CA. During the year 1995, she served as a Research Associate in the Center for Image Processing and Integrated Computing (CIPIC) at the U. of California Davis. Dr. Clark has served as thesis advisor for 10 graduate students: 5 MSECE students at the Naval Postgraduate School, and 3 MSECE plus 2 Ph.D. ECE students at U. of California Davis. She earned BSEE and MSEE degrees from the Purdue U. Electrical Engineering Honors Program, W. Lafayette, IN, and the Ph.D. in Electrical and Computer Engineering (ECE) from the U. of California Santa Barbara. Her technical expertise is statistical signal/image processing, estimation, detection, pattern recognition/machine learning, sensor fusion, communication and control. She has contributed more than 220 publications on signal processing in acoustics, electromagnetics and particle physics. Dr. Clark has received multiple awards for signal processing contributions and teaching. She is a Fellow of the IEEE and a member of the Acoustical Society of America (ASA) Technical Council on Signal Processing in Acoustics, the Society of Exploration Geophysicists, Eta Kappa Nu and Sigma Xi.

Lieutenant Timothy M. Beach hails from Honolulu, Hi. He was commissioned through the United States Naval Academy where he earned his undergraduate degree in Systems Engineering in 2007. He graduated as an Engineering Duty Officer Option. He qualified surface Warfare Officer onboard USS GRIDLEY (DDG 101). He executed the Engineering Duty Officer option while serving a DCA tour onboard USS BOXER (LHD 4). In 2013, he received a Master of Science degree in Electrical Engineering with an emphasis in Missile Control Systems. He is currently the Project Superintendent for USS CHANCELLORSVILLE (CG 62) at Ship Repair Facility and Japan Regional Maintenance Center in Yokosuka, Japan.