Real-Time, Anchor-Free Node Tracking Using Ultrawideband Range and Odometry Data

Brian Beck
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta GA 30332
email: bbeck6@gatech.edu

Robert Baxley, Joseph Kim
Information and Communications Laboratory
Georgia Tech Research Institute
Atlanta GA 30318
email: bob.baxley@gtri.gatech.edu, joseph.kim@gtri.gatech.edu

***Abstract*—This paper extends our prior work in node localization and tracking when using both range measurements and odometry data. In our approach both types of distance data are combined using multidimensional scaling, a well known method of producing sets of estimated coordinates from measured 'dissimilarity' values. In this extension, knowledge of the entire data set is no longer assumed, making node tracking possible in real time as computing resources allow. The new algorithm's performance is demonstrated through simulations for various parameter values. We also present tracking results from real ultrawideband range and encoder data gathered from our CSOT mobile robot testbed. The results show compelling performance when benchmarked against the traditional extended Kalman filter.**

I. Introduction

The localization and tracking of mobile nodes is of broad interest in the sensor network and robotics fields. Real-time location data is an essential prerequisite for assigning meaning to other measured data [1], [2]. Existing solutions such as GPS may not be available, or of insufficient accuracy for many applications. In such situations, the use of ultrawideband (UWB) radio techniques has attracted interest [3], [4]. UWB ranging has the advantage of very high temporal resolution and bandwidth, making it resistant to multipath propagation and interference common in many environments. This leads to more accurate ranging, a key input to most localization techniques.

Multidimensional scaling (MDS) is a data visualization technique that has also gained attention in the localization literature. MDS has historically been used in fields such as econometrics and psychology to characterize the most relevant information present in a data set. The process maps a higher dimensional set of 'dissimilarity' values into a lower dimensional set for easier visualization [5]. MDS has gained attention in the localization literature, e.g. [6]–[8], since measurements of range between nodes naturally fit the definition of dissimilarity as used in MDS, and the physical locations of the nodes is the lower dimensional set of points sought. MDS localization algorithms have been adapted to include features such as weighted measurements [9], distributed computation [10], [11], and dealing with missing data in partially connected networks [12]–[14].

This material is based in part upon work supported by the National Science Foundation under grant no. ECCS-1343256.

In our previous work [15], we presented a novel technique for anchor-free localization and tracking by combining both range and odometry data in the framework of MDS. An advantage of our approach is the ability to estimate the positions of a set of mobile nodes over time, without the need for pre-surveyed anchor nodes or stationary references. Inter-node range measurements (e.g., UWB ranges) are combined with intra-node odometry measurements (e.g., encoder wheel distances) such that the dissimilarity data is connected across time. This allows the relative paths of the nodes to be recovered jointly using all available data. Other advantages include few parametric assumptions, fusion of commonly available and accurate odometry data, and the ability to account for missing data and differences in accuracy.

In this paper we extend our previous algorithm to allow for a real-time tracking solution. In [15], it was assumed that all data was available, that is, localization of node paths is done after the fact. Here we will estimate the positions of nodes over time as the data becomes available at each time step, a very useful improvement. A system on the order of dozens of nodes can be estimated in a few seconds with typical hardware. The only additional assumption made is that at least three nodes remain stationary during the tracking, though in principle this may not always be necessary. In Section II, we detail our proposed tracking method and summarize in algorithmic form. Section III displays results of simulations performed to demonstrate performance over a wide range of parameter values. In Section IV, we provide the results of an extensive measurement campaign to gather real UWB range and encoder odometry data for tracking. Finally, we provide our conclusions in Section V.

II. Tracking with Range and Odometry

The mathematical development closely follows the notation and conventions we established in [15], with unchanged elements quoted directly. We again consider the problem of estimating the positions of N nodes in each of K time steps. The position of node n in the xy plane at time k is given by $\mathbf{s}_n^{(k)} = \left[s_{n,x}^{(k)}\ s_{n,y}^{(k)}\right]^T$. The full tracking solution is then given by compiling the NK position vectors into the matrix

$$\mathbf{S} = \left[\mathbf{s}_1^{(0)}\mathbf{s}_2^{(0)}\ldots\mathbf{s}_N^{(K-1)}\right]^T \in \mathbb{R}^{NK\times 2}. \quad (1)$$

A matrix of position coordinates is used instead of a vector by convention in the MDS literature. We assume that a set of $N(N-1)$ pairwise range measurements become available at each time step, arranged in matrix $\mathbf{R}^{(k)} \in \mathbb{R}^{N\times N}$ such that $\left[\mathbf{R}^{(k)}\right]_{i,j} = r_{ij}^{(k)}$ is the range estimate between nodes i and j at time step k. The main diagonal of $\mathbf{R}^{(k)}$ is equal to zero by definition. The range values are assumed to be already estimated from some method such as time-of-flight, RSSI, etc. Similarly, we assume a set of odometry distance measurements are available for each node, which represent the distance traveled by that node between time steps $k-1$ and k. These are arranged into the diagonal matrix $\mathbf{D}^{(k)} \in \mathbb{R}^{N\times N}$, such that $\left[\mathbf{D}^{(k)}\right]_{i,i} = d_i^{(k)}$ is the distance traveled by node i between times $k-1$ and k. The $d_i^{(k)}$ values are estimated from encoder wheel pulses, inertial navigation, etc. In this work we assume that the off-diagonal elements of $\mathbf{D}^{(k)}$ are unknown and set to zero.

The component matrices $\mathbf{R}^{(k)}$ and $\mathbf{D}^{(k)}$ represent inter-node range values at time k and the intra-node ranges between times $k-1$ and k, respectively. The available data is compiled into a block-symmetric matrix

$$\mathbf{\Delta}^{(k)} = \begin{bmatrix}\mathbf{R}^{(k-1)} & \mathbf{D}^{(k)} \\ \mathbf{D}^{(k)} & \mathbf{R}^{(k)}\end{bmatrix} \in \mathbb{R}^{2N\times 2N}\ , \quad (2)$$

which contains the measured range values related to the unknown node positions

$$\mathbf{S}^{(k)} = \left[\mathbf{s}_1^{(k-1)}\mathbf{s}_2^{(k-1)}\ldots\mathbf{s}_N^{(k)}\right]^T \in \mathbb{R}^{2N\times 2} \quad (3)$$

for two time steps. Constructing $\mathbf{\Delta}^{(k)}$ in this manner fits the definition of a *dissimilarity matrix* from the MDS literature. That is, elements of $\mathbf{\Delta}^{(k)}$ represent the dissimilarity, or distance between nodes in the plane for both the current and previous time step.

To solve for positions $\mathbf{S}^{(k)}$, we take the weighted stress function approach of [9]. We seek the set of positions $\hat{\mathbf{S}}^{(k)}$ that minimize the following stress function:

$$\mathcal{J}(\hat{\mathbf{S}}^{(k)}) = \sum_{i=1}^{2N}\sum_{j=1}^{2N}\left[\mathbf{W}^{(k)}\right]_{i,j}\left(\left[\mathbf{\Delta}^{(k)}\right]_{i,j} - \left[d(\hat{\mathbf{S}}^{(k)})\right]_{i,j}\right)^2. \quad (4)$$

In (4), the matrix $\mathbf{W}^{(k)} \in \mathbb{R}^{2N\times 2N}$ is a matrix of non-negative scalar weights, and the function $d\colon \mathbb{R}^{2N\times 2} \to \mathbb{R}^{2N\times 2N}$ maps the set of coordinates $\hat{\mathbf{S}}^{(k)}$ into a matrix of pairwise distances with the same structure as $\mathbf{\Delta}^{(k)}$. That is,

$$\left[d(\hat{\mathbf{S}}^{(k)})\right]_{i,j} = \sqrt{\left([\hat{\mathbf{S}}^{(k)}]_{i,:} - [\hat{\mathbf{S}}^{(k)}]_{j,:}\right)^T\left([\hat{\mathbf{S}}^{(k)}]_{i,:} - [\hat{\mathbf{S}}^{(k)}]_{j,:}\right)}\ . \quad (5)$$

The stress function (4) is both nonlinear and non-convex in the variable $\hat{\mathbf{S}}^{(k)}$. The most common technique for minimizing this function is known as the SMACOF algorithm [5], and involves transformation of the stress via majorizing functions. The algorithm is iterative, and usually offers quick convergence. We use the SMACOF algorithm in this paper to minimize (4), with the addition of trying multiple random initial start configurations to help avoid local minima. It should also be noted that taking the weighted approach to MDS avoids the noise amplification caused by squaring the distance values in the classical non-weighted MDS solution.

The values in the weighting matrix $\mathbf{W}^{(k)}$ represent our relative confidence in the data values. It conveniently allows accounting for any missing data by simply setting the corresponding weight to zero, and thus will have no influence on the stress function (4). For the remaining measurements, traditional linear weighted least squares sets $\left[\mathbf{W}^{(k)}\right]_{i,j} = \left(\sigma_{ij}^2\right)^{-1}$, where σ_{ij}^2 is the assumed variance of the corresponding measurement. However, the stress function (4) is not linear due to the distance mapping $d(\cdot)$. In our simulations and experimental results we have obtained best results by assigning weights based on the relative standard deviations. In this paper we assign weights by $\left[\mathbf{W}^{(k)}\right]_{i,j} = c\left(\sigma_{ij}\right)^{-1}$, with c set so that the smallest weight value equals 1 to avoid numerical instability. In the case where no a-priori information is known about measurement quality, then all measurement weights could be set to 1.

We now apply the MDS and measurement framework outlined above to the tracking of node positions over time. Unlike many localization and tracking methods, our procedure does not require any *anchor nodes*, nodes for which exact coordinates are known a-priori and do not move during the session. In this work we will only assume the existence of at least 3 *stationary nodes*, nodes which are known not to move during the tracking. The stationary nodes' positions need not be known and are estimated. This is a stronger assumption not made in [15], though we will point out how it might be avoided.

To begin tracking, an initial reference frame must be established that will be maintained for the entire session, because in general MDS-based algorithms produce relative maps that are only equivalent up to a rotation, reflection, and translation. Assume without loss of generality that the first $m \geq 3$ nodes are the stationary references. The first set of inter-node range measurements are taken and assembled into matrix $\mathbf{R}^{(0)}$. Applying the SMACOF algorithm will produce an initial set of N position estimates $\hat{\mathbf{S}}_{ref}^{(0)} \in \mathbb{R}^{N\times 2}$. The $N-m$ mobile nodes each then proceed to move independently; the measured intra-node travel distances are compiled into matrix $\mathbf{D}^{(1)}$. At the next time step matrix $\mathbf{R}^{(1)}$ is filled with the new set of inter-node ranges. Matrix $\mathbf{\Delta}^{(1)}$ can now be assembled as in (2). Again applying SMACOF will produce estimated node coordinates $\hat{\mathbf{S}}^{(1)}$.

Since MDS-based algorithms such as SMACOF produce relative maps, node coordinates are preserved only up to an orthogonal transformation (rotation/reflection/translation). Thus maps produced by successive runs of SMACOF are not generally comparable. However, note that $\hat{\mathbf{S}}^{(1)}$ contains a position estimate for each node at both time steps $k = 0$ and $k = 1$, creating an overlap. We also know that there are at least m node positions in common between $\hat{\mathbf{S}}^{(0)}_{ref}$ and $\hat{\mathbf{S}}^{(1)}$. We use this information to derive an orthogonal transformation which cancels out the ambiguity produced by successive SMACOF operations. That is, we seek a transformation that will optimally "map back" each new set of position estimates onto our original reference frame established by $\hat{\mathbf{S}}^{(0)}_{ref}$:

$$\hat{\mathbf{S}}'^{(k)} = s\hat{\mathbf{S}}^{(k)}\mathbf{T} + \mathbf{1}_{2N}\mathbf{t}^T \ , \tag{6}$$

for scale factor s, rotation/reflection matrix operator $\mathbf{T}$, and translation vector $\mathbf{t}$. Finding these values is known as a Procrustean similarity transformation [5]. It can be shown that if $\mathbf{Y}$ is the $m \times 2$ set of stationary node positions from $\hat{\mathbf{S}}^{(k)}$, and $\mathbf{X}$ is the $m \times 2$ set of stationary nodes from $\hat{\mathbf{S}}^{(0)}_{ref}$, then

$$\begin{aligned}
\mathbf{C} &= \mathbf{X}^T \left(\mathbf{I}_m - m^{-1}\mathbf{1}_m\mathbf{1}_m^T\right)\mathbf{Y} \\
\mathrm{svd}(\mathbf{C}) &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\
\mathbf{T} &= \mathbf{V}\mathbf{U}^T \in \mathbb{R}^{2\times 2} && (7) \\
s &= \frac{\mathrm{Tr}\big(\mathbf{X}^T\left(\mathbf{I}_m - m^{-1}\mathbf{1}_m\mathbf{1}_m^T\right)\mathbf{Y}\mathbf{T}\big)}{\mathrm{Tr}\big(\mathbf{Y}^T\left(\mathbf{I}_m - m^{-1}\mathbf{1}_m\mathbf{1}_m^T\right)\mathbf{Y}\big)} \in \mathbb{R} && (8) \\
\mathbf{t} &= m^{-1}\left(\mathbf{X} - s\mathbf{Y}\mathbf{T}\right)^T \mathbf{1}_m \in \mathbb{R}^{2\times 1} && (9)
\end{aligned}$$

will minimize the squared error after applying the transformation. By applying the similarity transformations derived from the stationary nodes, the original reference frame can be maintained during tracking for all nodes. This process of gathering range and odometry data, performing SMACOF, then aligning the new map continues for all time steps k. The procedure is summarized in Algorithm 1.

As stated before, the algorithm assumes the existence of at least $m \geq 3$ stationary nodes which are used to align successive maps. However, it is clear from the definition of $\mathbf{\Delta}^{(k)}$ in (2) that two sets of position estimates are being produced at each iteration: one for time $k-1$ and one for time k. Thus all N position estimates from time $k-1$ could be used to produce the similarity transformation vectors (Algorithm 1 lines 11 and 12) for time step k. If this were done, then all nodes could be mobile. However, we would be applying the transformation using successively estimated sets of points. Thus, position errors would accumulate over time, making this analogous to a type of dead reckoning localization.

III. Simulation Results

Here we present comprehensive simulations to demonstrate the algorithm's performance. We assume that nodes move within an approximately 20×20 meter area. The movements at each time step occur according to a constant velocity model, with velocity perturbation $\sim N(0, \sigma_v^2)$. The standard deviation of velocity perturbations is $\sigma_v = 200$ mm/s at each step. The error variance in both the range and odometry measurements

Algorithm 1: MDS Tracking with Range and Odometry

Data: Range and odometry measurements $\mathbf{R}^{(k)}$, $\mathbf{D}^{(k)}$
Result: Coordinates $\hat{\mathbf{S}}$ estimating true node positions $\mathbf{S}$

1 **begin**
2 *Initialize* $k = 0$: observe ranges $\mathbf{R}^{(0)}$;
3 $\hat{\mathbf{S}}^{(0)}_{ref} \leftarrow \left[\hat{\mathbf{S}}\right]_{1:N,:} \leftarrow \mathrm{SMACOF}\big(\mathbf{R}^{(0)}, \mathbf{W}^{(0)}\big)$;
4 $\mathbf{X} \leftarrow \left[\hat{\mathbf{S}}^{(0)}_{ref}\right]_{1:m,:}$;
5 **for** $k = 1$ **to** $K - 1$ **do**
6 observe ranges $\mathbf{R}^{(k)}$;
7 observe odometry $\mathbf{D}^{(k)}$;
8 assemble matrix $\mathbf{\Delta}^{(k)}$ in (2) ;
9 $\hat{\mathbf{S}}^{(k)} \leftarrow \mathrm{SMACOF}\big(\mathbf{\Delta}^{(k)}, \mathbf{W}^{(k)}\big)$;
10 $\mathbf{Y} \leftarrow \left[\hat{\mathbf{S}}^{(k)}\right]_{1:m,:}$;
11 compute s, $\mathbf{T}$, $\mathbf{t}$ by (7), (8), (9) ;
12 $\hat{\mathbf{S}}'^{(k)} \leftarrow s\hat{\mathbf{S}}^{(k)}\mathbf{T} + \mathbf{1}_{2N}\mathbf{t}^T$ by (6) ;
13 store current solution: $\left[\hat{\mathbf{S}}\right]_{Nk+1:Nk+N,:} \leftarrow \hat{\mathbf{S}}'^{(k)}$;
14 **end**
15 **end**

is assumed proportional to the actual distance, which is consistent with both measurement types. This proportionality for range and odometry are assumed known and denoted σ_r^2, σ_d^2 respectively. That is, σ_r^2, σ_d^2 are the variances in the range measurements *per meter*. In all cases there were $m = 3$ stationary nodes, and 240 simulations were run for each data point.

From the simulated movements and measurement data, we perform 3 separate estimation algorithms for comparison. To benchmark our results, we first perform tracking using the well-known Extended Kalman Filter (EKF). The EKF assumes that the *exact* coordinates of the stationary nodes are known. Thus, they become traditional anchor nodes for EKF tracking. The EKF makes use of range data only, but also assumes the same motion model described above, tracking position and velocity. That is, there is no mismatch between the EKF's state-space model and the model generating the true node trajectories. The EKF also assumes that the initial positions of the nodes are known to within 1 meter. We also perform tracking using our proposed tracking algorithm, both with and without odometry data.

For the first simulation, we measure the average RMSE of position estimates for different values of node count N over $K = 20$ time steps, with noise levels $\sigma_r^2 = (75\mathrm{mm})^2$, $\sigma_d^2 = (10\mathrm{mm})^2$. The results are shown in Fig. 1. From these results, we see that estimation accuracy generally improves with increasing N. As N increases, more range data is available at each time index, and more odometry values are available between time steps. The EKF shows a much stronger dependence on node count, actually overtaking our approach for $N \geq 14$ if odometry data is not used.

Our second simulation fixes $N = 8$, $K = 20$ time steps, and

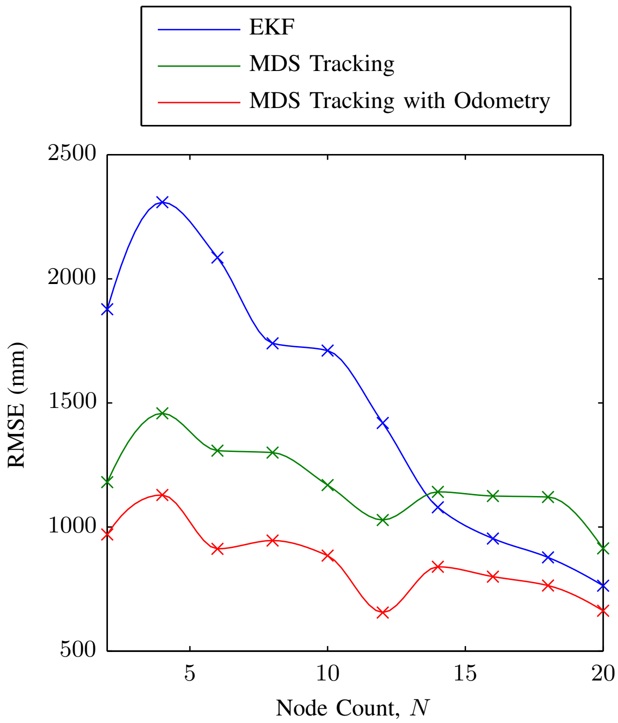

Fig. 1. First comprehensive simulation results, position RMSE vs. number of nodes N over 20 time steps. Noise levels: $\sigma_r^2 = (75\text{mm})^2$, $\sigma_d^2 = (10\text{mm})^2$.

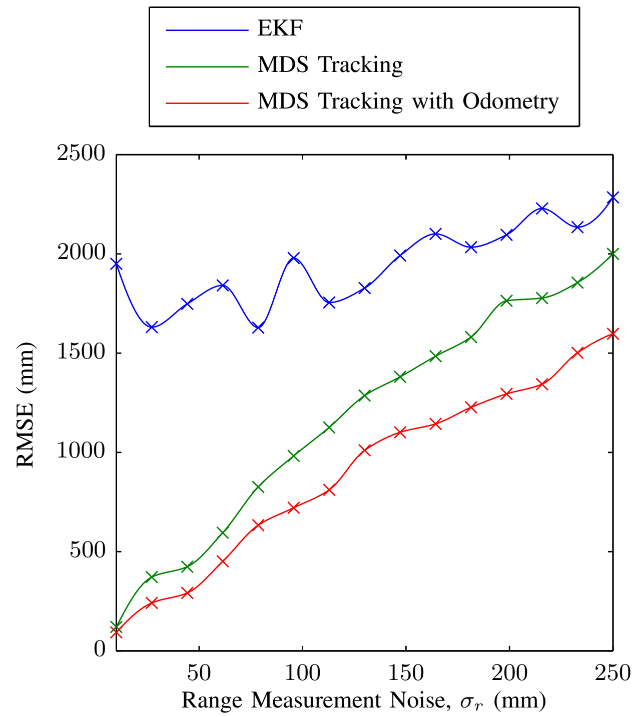

Fig. 2. Second comprehensive simulation results, position RMSE vs. range measurement noise for 8 nodes over $K = 20$ time steps, with $\sigma_d = 20$mm.

$\sigma_d = 20$ mm, with the noise standard deviation σ_r allowed to vary. The results are plotted in Fig. 2. The performance of course generally decreases as measurement uncertainty increases. Our algorithm consistently outperforms the EKF, though the RMSE increases more slowly for the latter. The EKF has an internal motion model, which is favored more as the measurement error becomes large. Of course, if the true motion of the nodes does not match the internal assumptions, then this will not help. Also, the EKF must "lock on" to the trajectories of the nodes, which may take a few samples. We note that our proposed approach does not have these problems, and measurement accuracy is almost solely a function of the measurement error. The performance difference between the two MDS approaches increases as the range measurement error increases. If the algorithm has access to the more reliable odometry data, it will be weighted more strongly, resulting in the observed better performance.

IV. Tracking Results on Real CSOT Data

We have tested the same three tracking methods using real ultrawideband range and encoder wheel odometry data gathered from our Cognitive Spectrum Operations Testbed (CSOT). The testbed capabilities were originally introduced in [16] for producing area-wide spectrum maps. The system features eight mobile robots and central processing server. Each mobile node is equipped with an ultrawideband ranging radio, capable of ~ 4.7 cm RMS range accuracy. The ranging radio separately utilizes 2 UWB antennas mounted at the top of the robot, spaced 315mm apart. Odometry data is gathered by the wheel encoders built into the iRobot mobile base, with comparable accuracy to the UWB radios over short distances. A photo of the system setup is shown in Fig. 3.

Fig. 3. CSOT initial setup prior to Test I.

Fig. 4 displays the results of three measurement campaigns performed in a laboratory hallway to measure real-world tracking performance. The same EKF of Section III is used again for comparison, and assumes that the standard deviation of velocity change is $\sigma_v = 200$ mm/s at each step. The tests

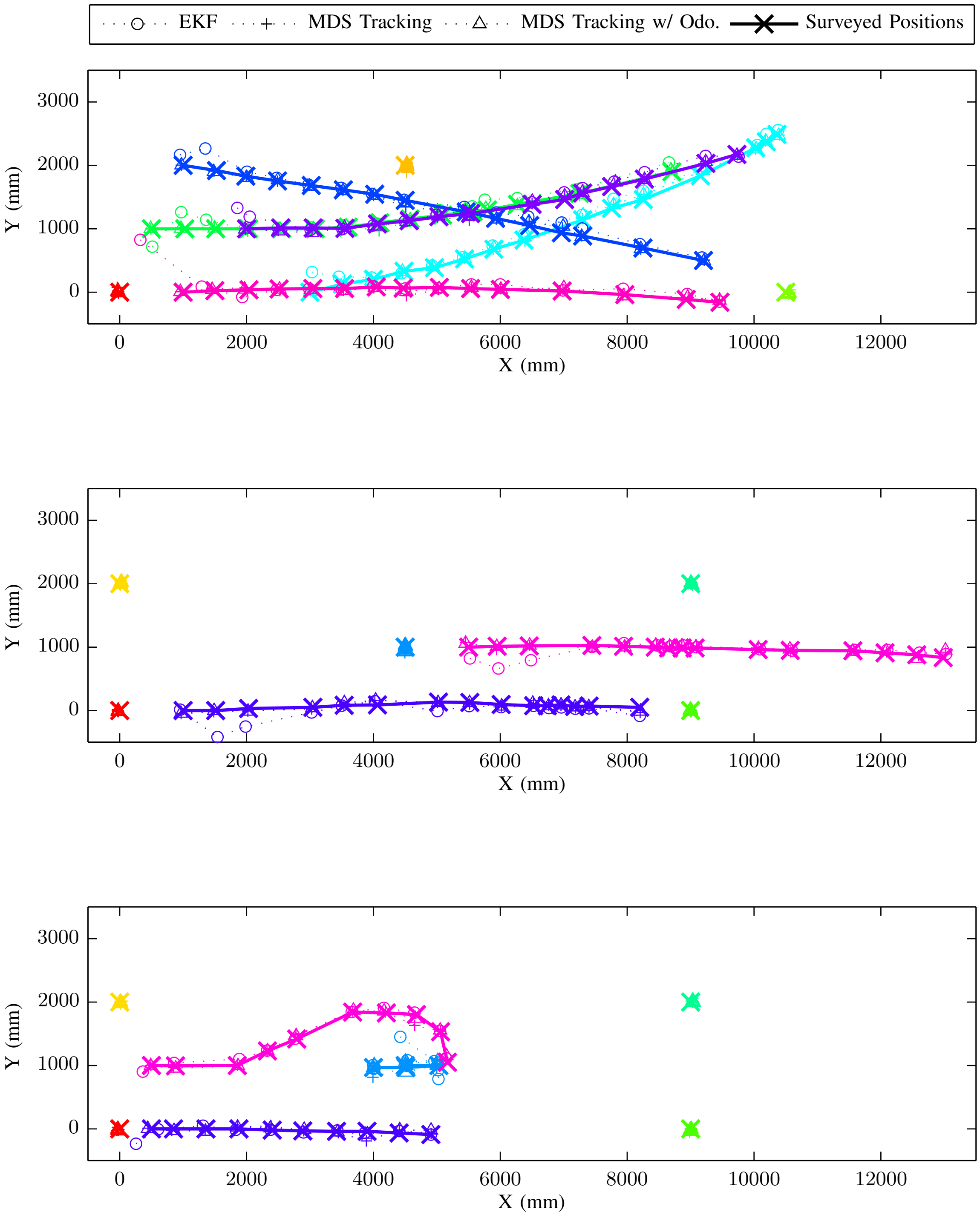

Fig. 4. CSOT testbed tracking results for Test I (top), Test II (middle) and Test III (bottom).

	Test I	Test II	Test III
Stationary nodes	3	5	4
Mobile nodes	5	2	3
Total time steps K	15	15	10
RMSE: MDS (w/ odo.)	20 mm	15 mm	19 mm
RMSE: MDS (no odo.)	23 mm	16 mm	24 mm
RMSE: EKF	91 mm	68 mm	67 mm
Avg. range error stdev., σ_r	30 mm		
Avg. odo. error stdev., σ_d	27 mm (per meter)		

TABLE I
CSOT MEASUREMENT CAMPAIGN SUMMARY

represent a range of values for stationary and mobile nodes, movement paths, and time steps. While the first three stationary node positions were known to the EKF, we stress that our MDS algorithm assumes only that they were stationary. The true position of each node was surveyed at every time step, providing a full set of truth data for comparison. A summary of the test parameters is shown in Table I.

Each test was designed vary the number of moving nodes as well as movement patterns. Test I had the most mobile nodes, and the most consistent movement between time steps. Each robot moved approximately 0.5 to 1 meters at each time step. All three algorithms performed very well here, with RMSE on the order of centimeters as shown in Table I.

For Test II, only two nodes were mobile, leaving the rest stationary. However, the distance traveled by the mobile nodes was more variable vs. Test I. Movement distances in this case varied from 0.2 to 1 meters. The results show a corresponding decrease in RMSE for all three tracking solutions, indicating that more stationary nodes can improve performance. The relative performance between tracking methods is almost identical to Test I.

Test III utilized 3 moving nodes, with distances ranging from 0 to 1 meters. One node took a curved path, one a very straight path, and the third only moved half the time. These differences had little effect on the tracking errors. Overall, our MDS-based tracking using range and odometry is able to consistently outperform the EKF on actual measured UWB and encoder data. We do note that utilizing the odometry data in our tests provides only marginal benefit. In our tests the UWB range and odometry data errors are very close in magnitude, making these test results consistent with the far left region of Fig. 2.

V. CONCLUSIONS

In this paper we have extended our previous research into employing an anchor-free, MDS-based approach for localization and tracking. The extension enables tracking to be performed as new data is gathered. Our algorithm assumes only the existence of stationary nodes with which to relate position estimates gathered over time. We have presented simulation and actual testbed results to demonstrate performance exceeding that of the canonical extended Kalman filter. Our algorithm also has the substantial advantages of few assumptions, joint estimation, and eliminating the surveying of anchor node positions.

Future research could extend the algorithm further to drop the assumption of stationary nodes. As previously discussed, the tracking would become analogous to a dead reckoning approach. The performance of such a scheme could be investigated. A non-centralized approach to estimation would also be helpful for use in distributed or power restricted networks.

REFERENCES

[1] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[2] H. Wymeersch, J. Lien, and M. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[3] D. Schneider, "New indoor navigation technologies work where GPS can't," *IEEE Spectrum*, vol. 50, no. 12, pp. 34–39, Dec 2013.

[4] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, Jul 2005.

[5] I. Borg and P. Groenen, *Modern Multidimensional Scaling*, ser. Springer Series in Statistics. Springer-Verlag New York, 1997.

[6] K. Cheung and H. So, "A multidimensional scaling framework for mobile location using time-of-arrival measurements," *Signal Processing, IEEE Transactions on*, vol. 53, no. 2, pp. 460–470, 2005.

[7] D. Macagnano and G. De Abreu, "Tracking multiple dynamic targets with multidimensional scaling," in *Personal, Indoor and Mobile Radio Communications (PIMRC), IEEE 18th International Symposium on*, 2007, pp. 1–5.

[8] F. Chan and H. So, "Efficient weighted multidimensional scaling for wireless sensor network localization," *Signal Processing, IEEE Transactions on*, vol. 57, no. 11, pp. 4548–4553, 2009.

[9] J. Costa, N. Patwari, and A. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *Sensor Networks, ACM Transactions on*, vol. 2, no. 1, pp. 39–64, Feb 2006.

[10] O. Kwon and H. Song, "Localization through map stitching in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 1, pp. 93–105, 2008.

[11] O. Kwon, H. Song, and S. Park, "Anchor-free localization through flip-error-resistant map stitching in wireless sensor network," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 11, pp. 1644–1657, 2010.

[12] A. Amar, Y. Wang, and G. Leus, "Extending the classical multidimensional scaling algorithm given partial pairwise distance measurements," *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 473–476, 2010.

[13] P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangan, R. Virrankoski, and A. Savvides, "Distance matrix reconstruction from incomplete distance information for sensor network localization," in *Sensor and Ad Hoc Communications and Networks (SECON), 3rd Annual IEEE Communications Society on*, vol. 2, 2006, pp. 536–544.

[14] Y. Shang and W. Ruml, "Improved mds-based localization," in *Computer and Communications Societies (INFOCOM), IEEE Joint Conference of*, vol. 4, 2004, pp. 2640–2651.

[15] B. Beck and R. Baxley, "Anchor free node tracking using ranges, odometry, and multidimensional scaling," in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on*, 2014, (accepted for publication).

[16] B. Beck, J. Kim, R. Baxley, and B. Walkenhorst, "RadioBOT: A spatial cognitive radio testbed," in *IEEE Aerospace Conference*, 2013, pp. 1–9.