EE793 Paper Implementation Project Report

Oğul Can Yurdakul

I. Introduction

THIS report is a brief exposition of the Extended Target Tracking (ETT) algorithm proposed in [1]. The algorithm adapts the existing ETT algorithm proposed in [2], which works with measurements generated from the whole body of the target, to work with measurements that are generated only by the boundary of the target. The downside of using the algorithm in [2] directly in the case of contour measurements is due to fact that the contour measurements can only be taken from a segment of the target boundary. This shifts the filtered object centroid and extent towards the measurement mean and covariance, respectively, and hence causes a significant amount of bias in centroid, heading angle, and inferred target dimensions. The adaptation proposed in [1] corrects the algorithm in [2] by adding an iterative correction step that works by generating virtual measurements (VMs) and matching their statistics to those of the actual measurements.

The rest of this report is organized as follows: Section II gives a summary of the paper and provides some more details regarding the logic of the algorithm proposed. Section III provides more details specific to the algorithm as it is implemented for this project and experimentation setup. Section IV provides the results of the implementation and the related discussion of the algorithm structure, and concludes the report.

II. Paper Summary

A. State-Space and Extent Models

After providing an introduction to the material, the paper proceeds by illustrating the state and measurement models, together with the random matrix (RM) approach [1]. In concord with this baseline algorithm's paper, they use Singer's Object Dynamic model [3], where three state variables are defined per dimension: Position p, velocity v and acceleration a. The dynamics in each of the dimensions are assumed to be identical, therefore these variables are added for each motion dimension and the dynamic model is the same for each motion dimension, given by the linear relation

$$F = \begin{bmatrix} 1 & \Delta t & {\Delta t^2}/{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & e^{-\Delta t/\theta} \end{bmatrix}, \tag{1}$$

meaning the whole state transition matrix for the total state vector x_k is given as $F \otimes I_d$, where Δt is the sampling time, θ is a time constant that serves to keep the acceleration bounded and I_d is the $d \times d$ identity matrix. Furthermore, the measurements are taken as noise-corrupted position measurements, with a measurement matrix for each dimension given as

$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{2}$$

with noise covariance σ_x^2. With this setup, the measurement likelihood is expressed as

$$p(\mathbf{Z}_k | n_k, x_k, X_k) = \prod_{n=1}^{n_k} \mathcal{N}\left(z_k^n ;\, (H \otimes I_d) x_k,\, X_k + \text{diag}(\begin{bmatrix} \sigma_x^2 & \sigma_x^2 \end{bmatrix})\right) \tag{3}$$

where n_k is the number of measurements at time k and X_k is the object extent matrix, assumed to be distributed according to the inverse Wishart density with (posterior) positive-definite scale matrix $X_{k|k}$and degrees of freedom $\nu_{k|k}$. The prediction and measurement updates for these kinematic and extent state variables, given by the combination of the Kalman Filter (KF) and the RM approach, are given in Algorithm 1.

B. Virtual Measurement Generation

The next section is dedicated to an exposition of ellipses and how one simulated LIDAR measurements from them. The key idea is that the LIDAR sensor's rays intersect with the ellipse contour and the closest intersection point is a measurement source generating a measurement with probability p_d. This measurement source can be determined by inserting the line equation describing each measurement angle into the ellipse equation and solving the resulting quadratic equation. This procedure results in at most two real solutions. If there exists no solution, then the ray does not intersect the ellipse, generating no measurement. If there is only one solution, then the ray is tangent to the ellipse, and there is a unique measurement source. Lastly, if there are two solutions, the one that corresponds to the intersection point with the closest Euclidean distance to the LIDAR sensor is the measurement source.

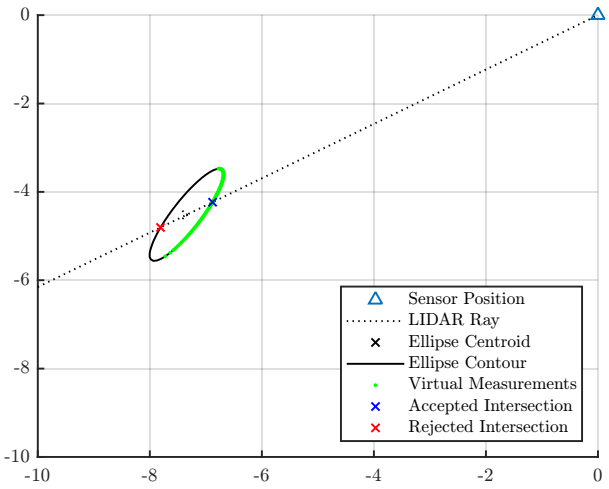


Fig. 1. Sample ellipse and corresponding virtual measurements.

C. Adaptation Algorithm

The key idea of the paper is presented in the subsequent two sections with the exposition of the reference model concept and the adaptation algorithm. The idea is to consider the measurement generation process as a function of an additional state-like variable u_k that corresponds to the objects' extent variables, namely the ellipse centroid, length and width (but not the tilt angle), and to find an estimate $\tilde{u}_k$ via an optimization routine. This optimization routine matches the statistics of the actual measurements from the LIDAR sensor, assumed to have been generated with the input u_k, to those that are simulated using $\tilde{u}_k$ using the virtual measurement model (VMM). This extent variable u_k is defined as

$$u_k = \begin{bmatrix} c_k^{(x)} & c_k^{(y)} & l_k & w_k \end{bmatrix}^T \tag{4}$$

where c_k^x and c_k^y are the object's centroid coordinates, $0 < l_k$ its length and $0 < w_k < l_k$ its width. To match the statistics of the real measurements with those of the virtual ones, the output y_k for u_k is defined as

$$y_k = \begin{bmatrix} m_{k|k}^{(x)} & m_{k|k}^{(y)} & 2\sqrt{e_k^{(1)}} & 2\sqrt{e_k^{(2)}} \end{bmatrix}^T \tag{5}$$

where $\begin{bmatrix} m_{k|k}^{(x)} & m_{k|k}^{(y)} \end{bmatrix}$ is the mean of the measurements with $e_k^{(1)}$ and $e_k^{(2)}$ being the larger and smaller eigenvalue of the measurement covariance matrix, respectively. The vector $\tilde{y}_k$ corresponding to the virtual measurements generated by $\tilde{u}_k$ has the same form. Then, the optimization rule to match $\tilde{u}_k$ to u_k is given iteratively as

$$\tilde{u}_{k,i+1} = \tilde{u}_{k,i} + K(y_k - \tilde{y}_{k,i}) \tag{6}$$

where i is the iteration index going from 1 to the total number of iterations I selected by the user, and K is the stepsize parameter. Notice that for each iteration, a new $\tilde{y}_{k,i}$ vector is used, meaning a new virtual measurement set has to be generated. The authors also provide proof of stability for this discrete-time evolution rule to find that the adaptation rule is stable for $0 < K < 2$. Although, despite this stability, notice how there is no imposition of the constraints $0 < w_k < l_k$ in (6), meaning after an iteration the constraint can be violated in any of these ways: The length can become smaller than the width, or the positivity of either the length or the width can become negative. Either of these cases would increase the error after the update and the authors avoid embedding these constraints into the adaptation rule by checking the error norm $\varepsilon_{k,i} := \|y_k - \tilde{y}_{k,i}\|$. If the error norm increases after applying (6), the previous $\tilde{u}_{k,i}$ is restored and K is decreased. For initialization, $\tilde{u}_1$ is derived from the one-step posterior of the KF and RM filters and a single-step prediction is applied when moving from $\tilde{u}_{k-1,I}$ to $\tilde{u}_{k,1}$ using the target kinematics, although the authors mention other possibilities for this prediction.

In the proceeding sections, the authors provide the results of their proposed algorithm, both on synthetic and real data, and compare their results to those of different algorithms. Importantly, they also discuss the computational load of their iterative approach and provide an analysis of the average computation time as a function of I.

III. Implementation & Results

Algorithm 1 Detailed algorithm of the Random Matrix approach with VMM adaptation, as proposed in [1] and implemented for this project.

for $k = 1 : K$ **do**
  ▷ KF + RM Approach Prediction Update
  $x_{k|k-1} \leftarrow (F \otimes I_d) x_{k-1|k-1}$
  $P_{k|k-1} \leftarrow F P_{k-1|k-1} F^T + D$
  $\nu_{k|k-1} \leftarrow e^{-\Delta t/\theta} \nu_{k-1|k-1}$
  $X_{k|k-1} \leftarrow \frac{e^{-\Delta t/\theta}\nu_{k-1|k-1} - d - 1}{\nu_{k-1|k-1} - d - 1} X_{k-1|k-1}$
  ▷ KF + RM Approach Measurement Update
  $\bar{z}_k \leftarrow \frac{1}{n_k} \sum_{n=1}^{n_k} z_k^{(n)}$
  $Z_k \leftarrow \sum_{n=1}^{n_k} (z_k^{(n)} - \bar{z}_k)(\cdot)^T$
  $S_{k|k-1} \leftarrow H P_{k|k-1} H^T + \frac{\sigma_x^2}{n_k}$
  $W_{k|k-1} \leftarrow P_{k|k-1} H^T (S_{k|k-1})^{-1}$
  $x_{k|k} \leftarrow x_{k|k-1} + (W_{k|k-1} \otimes I_d)(\bar{z}_k - (H \otimes I_d) x_{k|k-1})$
  $P_{k|k} \leftarrow P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^T$
  $N_{k|k-1} \leftarrow (\bar{z}_k - (H \otimes I_d) x_{k|k-1})(\cdot)^T (S_{k|k-1})^{-1}$
  $X_{k|k} \leftarrow X_{k|k-1} + N_{k|k-1} + Z_k$
  $\nu_{k|k} \leftarrow \nu_{k|k-1} + n_k$
  ▷ VMM Adaptation
  if $k == 1$ **then** ▷ Eqns. (46)-(49)
    Initialize $\tilde{u}_{1,1}$ with $x_{1|1}(1:2)$ and $\frac{1}{\nu_{1|1} - 6} X_{1|1}$
  else ▷ Eqns. (50)-(53)
    Predict $\tilde{u}_{k,1}$ from $\tilde{u}_{k-1,I}$ using the state dynamics
  end if
  Generate VMs using $\tilde{u}_{k,1}$ and calculate $\tilde{y}_{k,1}$
  $\varepsilon_{k,1} \leftarrow y_k - \tilde{y}_{k,1}$
  Calculate y_k using $\bar{z}_k$ and $\frac{1}{n_k} Z_k$
  Initialize $K = 0.1$
  for $i = 1 : I$ **do**
    Store VMM input $\tilde{u}_{k,i}$
    $\tilde{u}_{k,i+1} \leftarrow \tilde{u}_{k,i} + K(y_k - \tilde{y}_{k,i})$ ▷ Eqn. (33)
    Generate VMs using $\tilde{u}_{k,i+1}$ and calculate $\tilde{y}_{k,i+1}$
    $\varepsilon_{k,i+1} \leftarrow y_k - \tilde{y}_{k,i+1}$
    if $\|\varepsilon_{k,i+1}\|^2 > \|\varepsilon_{k,i}\|^2$ **then**
      $\tilde{u}_{k,i+1} \leftarrow \tilde{u}_{k,i}$
      $K \leftarrow K/2$
    end if
  end for
end for

The full algorithm as it is implemented in this project is given as in Algorithm 1. The measurements are generated artificially in MATLAB by replicating a LIDAR sensor with an

angle step size of 1° and having a non-elliptical object perform a U-turn scenario. Measurements with and without noise are used to observe the effect of noise the on the performance of the algorithms, both with and without the adaptation rule (6) to observe its contribution to extent estimation. When measurement noise is present, it is assumed to be zero-mean with covariance $\text{diag}(\begin{bmatrix}\sigma_x^2 & \sigma_x^2\end{bmatrix})$.

The parameters used in this implementation, which are mostly taken from the original paper, are given as follows:

$$\Delta t = 0.1 \text{ s} \qquad \theta = 0.5 \text{ }^1/_\text{s} \qquad \sigma_x = 0.5 \text{ m}$$
$$D = 0.01(1 - e^{-2\Delta t/\theta})\text{diag}(\begin{bmatrix}0 & 0 & 1\end{bmatrix})$$
VMs generated with 0.1° angle step size
$K = 0.1$ for each k, and $K \leftarrow K/2$ if necessary
$$x_{0|0} = 0_{6\times 1} \qquad P_{0|0} = \text{diag}(\begin{bmatrix}10 & 5 & 1\end{bmatrix})$$
$$X_{0|0} = 100 I_2 \qquad \nu_{0|0} = 4$$

IV. Results & Discussions

Figures 2 and 3 show the simulation results with and without the VMM adaptation proposed, respectively. The VMM adaptation algorithm provides a much better contour estimate in both cases of noise-free and noise-corrupted measurements. The RM approach by itself sticks to the measurement mean and covariance, which causes the extent to collapse when only one side of the object is visible with noise-free measurements essentially located on a line as seen in Figure 2a. Even in the noise-corrupted case, the RM approach by itself does nothing more than reflecting the noise covariance mostly, as seen in Figure 2b.

Furthermore, given the VM generation scheme used, it would also avoid the "which side of the car?" problem to some degree. Even if at some time instant the wrong side of the vehicle would coincide with the measurements obtained, the adaptation rule (6) would naturally push the extent estimate to match the correct side of the vehicle with the measurements. This improvement, however, only applies to the extent estimate and not the kinematics, which will be elaborated on in the next section.

A. Downsides & Improvement Strategies

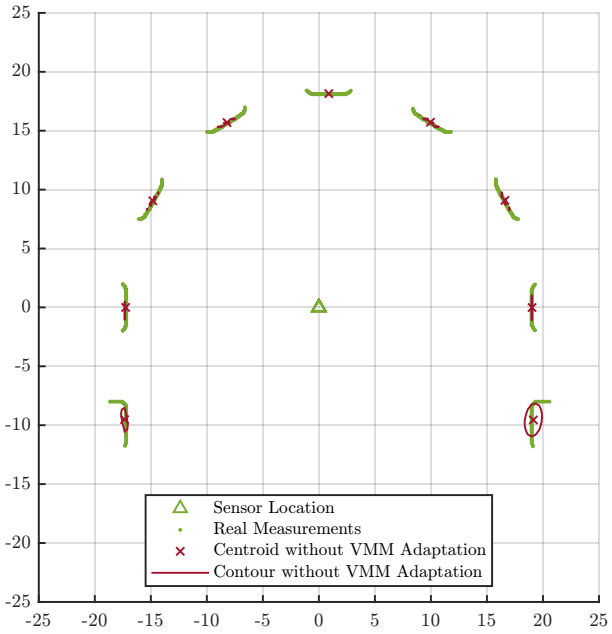
1) Not estimating the heading angle: As it is also stated by the authors, their contour estimation algorithm does not estimate the tilt angle $\varphi_{k|k}$ of the ellipse, but instead takes it directly from the KF and RM filter posterior as the heading angle

$$\varphi_{k|k} = \arctan\left(\frac{v_{k|k}^{(y)}}{v_{k|k}^{(x)}}\right), \tag{7}$$

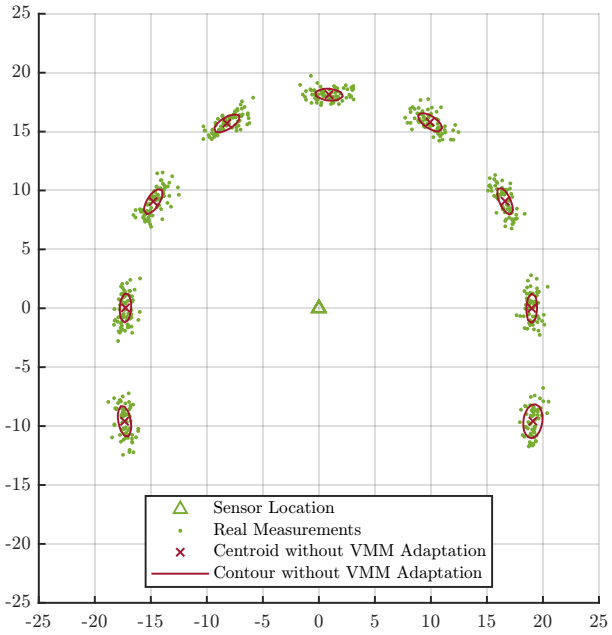
and hence their 1σ-ellipse is given by the implicit quadratic equation

$$\left(\begin{bmatrix}x\\y\end{bmatrix} - \begin{bmatrix}c_k^{(x)}\\c_k^{(y)}\end{bmatrix}\right)^T \left(R_{\varphi_{k|k}} \begin{bmatrix}(l_k/2)^2 & 0\\0 & (w_k/2)^2\end{bmatrix} R_{\varphi_{k|k}}^T\right)^{-1} (\cdot) = 1. \tag{8}$$

This means that this adaptation algorithm (and filter, overall) cannot handle objects that are drifting, i.e. objects whose extent is not aligned with their movement.



(a) Noise-free measurements



(b) Noise corrupted measurements

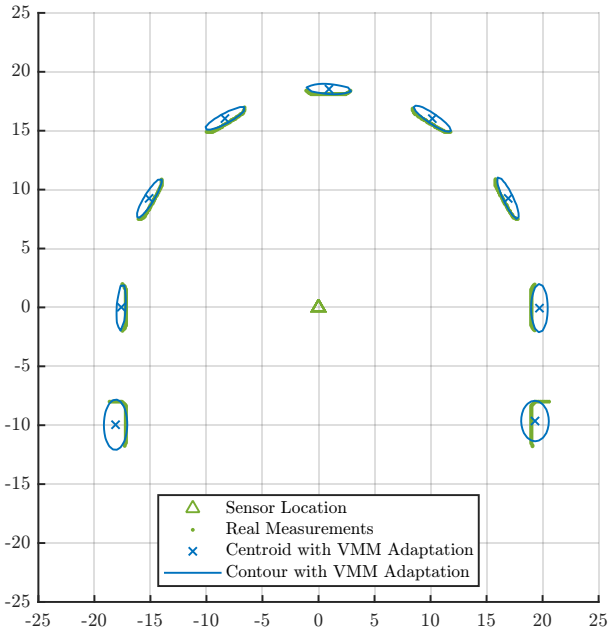
Fig. 2. Simulation results for the original algorithm proposed in [2], without the VMM adaptation. 1σ-ellipses drawn every 50 time step.

To alleviate this downside, the adaptation rule can be better adapted to learn the tilt angle. One can rewrite (8) as

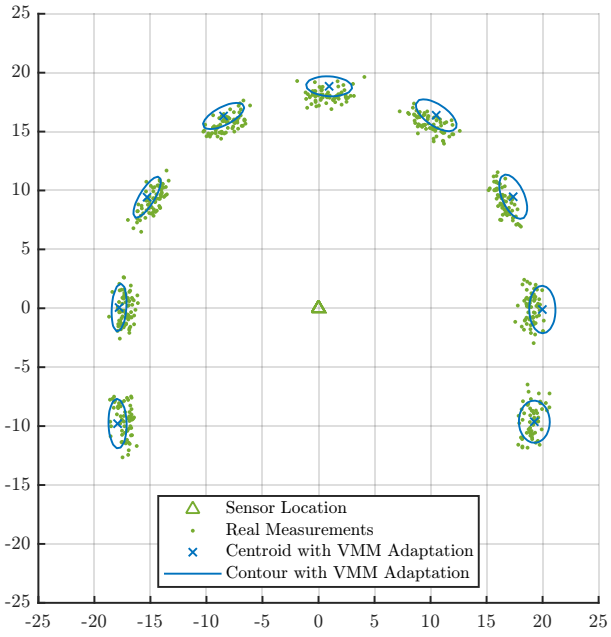
$$\left(\begin{bmatrix}x\\y\end{bmatrix} - \begin{bmatrix}c_k^{(x)}\\c_k^{(y)}\end{bmatrix}\right)^T \begin{bmatrix}a_k & b_k/2\\b_k/2 & c_k\end{bmatrix}^{-1} (\cdot) = 1 \tag{9}$$

where a_k, b_k and c_k are new variables that replace l_k and w_k in u_k to describe the extent of the ellipse. This time, the restriction is that their matrix remains positive-definite. With these new variables, one learns the tilt angle as well, and hence can express a drifting target with the same VMM approach.

2) Generation of virtual measurements: The fact that numerous VMs have to be generated I times at each time



(a) Noise-free measurements



(b) Noise corrupted measurements

Fig. 3. Simulation results for the algorithm proposed in [1], with the VMM adaptation. 1σ-ellipses drawn every 50 time step.

step increases the computational complexity. This downside, however, can be justified by noticing that taking I as low as 2 is quite enough to obtain satisfactory contour estimates.

A remedy can be changing the adaptation rule to work with the present measurements directly. The adaptation scheme is already very reminiscent of the Least Mean Squares (LMS) algorithm in adaptive filtering [4], which uses the instantaneous filter output error to the actual measured one to perform a gradient descent-like online optimization procedure. If the adaptation scheme were changed with such an algorithm to work with the actual measurements directly, the virtual measurement generation process can be discarded entirely.

3) Non-Bayesian extent estimate: The output $\tilde{u}_k$ extent variable is obtained via non-Bayesian methods, which means that neither of its components has statistical descriptions and that no confidence region around the contour can be drawn. This can be undesirable for gating purposes.

This problem is harder to remedy, as it necessitates a passage to the Bayesian framework altogether, and so I will make do with stating that elliptical regression methods within the Bayesian framework can be beneficial.

4) No use of the extent estimate in filtering: Since the algorithm does not feed the improvements in the contour estimate back into the actual estimated system state, the performance in the filtering part executed by the KF and RM approaches is not affected by the improvement in the contour estimation process.

This, I believe, is one of the most crucial downsides of this algorithm. If this VMM adaptation process could be fed back into system to modify this step's posterior (and hence the next one's prior), the filter performance regarding the other state variables can be improved drastically. However, my experiments show that it is not a trivial task, and may lead to discarding the KF and RM approaches as they are implemented here altogether. I tried very naively feeding back the contour and centroid estimates artificially into the state variables $x_{k|k}$ and $X_{k|k}$, but it did not improve the filter performance and even caused signficant errors in the tilt angle of the ellipse.

REFERENCES

[1] P. Hoher, S. Wirtensohn, T. Baur, J. Reuter, F. Govaers, and W. Koch, "Extended Target Tracking With a Lidar Sensor Using Random Matrices and a Virtual Measurement Model," *IEEE Transactions on Signal Processing*, vol. 70, pp. 228–239, 2022.

[2] J. W. Koch, "Bayesian Approach to Extended Object and Cluster Tracking Using Random Matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, 2008.

[3] R. A. Singer, "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-6, no. 4, pp. 473–483, 1970.

[4] A. H. Sayed, *Adaptive Filters*. John Wiley & Sons, 2011.