Calling Attention to Some Interesting New Results in Applying EKF's

Thomas H. Kerr III, **Zivaro** Algorithm Engineer/Sr. Systems Engineer

I have recently encountered five surprisingly good, open-literature publications that push the state-of-the-art in Kalman Filter-like estimation and target-tracking in important new ways. **They are listed below:**

1. Yulong Huang; Yonggang Zhang; Bo Xu; Zhemin Wu; Jonathon A. Chambers, "A New Adaptive Extended Kalman Filter for Cooperative Localization," **IEEE Transactions on Aerospace and Electronic Systems (AES)**, Vol. 54, No. 1, pp. 353-368, February 2018.
2. Yulong Huang; Mingming Bei; Youfu Li; Yonggang Zhang; Jonathon A. Chambers, "An Improved Variational Adaptive Kalman Filter for Cooperative Localization," **IEEE Sensors Journal**, Vol. 21, No. 9, pp. 10775-10786, May 2021.
3. Yuming Chen; Wei Li; Yuqiao Wang, On-line "Adaptive Kalman Filter for Target tracking with Unknown Noise Statistics," **IEEE Sensors Letters**, Vol. 5, No. 3, March 2021.
4. Zhifu Shi; Xiaohui Xu, "Near and Supersonic Target Tracking Algorithm Based on Adaptive Kalman Filter," **8th International Conference on Intelligent Human-Machine Systems** (**IHMSC**), 25-8 August 2016; IEEE Xplore: 15 December 2016.
5. Yuming Chen; Wei Li; Yuqiao Wang, On-line "Adaptive Kalman Filter for Target Tracking with Unknown Noise Statistics," **IEEE Sensors Letters**, Vol. 5, No. 3, March 2021.

Notable is that the above authors are predominantly Chinese and go on to discuss some relatively lucrative defense applications such as real-time radar tracking of an enemy sea-skimming missile to provide successful intercepts to protect their ships. The 4th of the above papers deals with tracking and handling of nearby **supersonic** enemy missiles to successfully intercept them. In general, the above papers initially invoke an **E-M Algorithm** (**E**xpectation-**M**aximization **Algorithm**, which usually incurs a notoriously large and slow processing burden to *eventually* get the ideal result that is sought. In the subsequent papers above, the authors identify faster alternative approximate solution techniques that speed up the associated processing burden to get a timely practical solution that works. (They include Monte-Carlo simulations first and ultimately hardware implementation and testing confirmations of the efficacy of their approach.) These papers grabbed my attention since, at the end of Sec. 4.3 on page 17 of Kerr, T.H., "Assessing and Improving the Status of Existing Angle-Only Tracking (AOT) Results," **Proceedings of the International Conference on Signal Processing and Applications & Technology**, Boston, pp. 1574-1587, 24-26 Oct. 1995, I suggested using the E-M Algorithm (instead of GLR) within certain EKF applications early on. Now some investigators have done so!

The first paper received the **M. Barry Carlton Award** for **Outstanding Paper** appearing in the ***IEEE Transactions on Aerospace and Electronic Systems*** in **2018.** The other papers are related in some important ways by explaining the approach and feasible E-M Algorithm approximations for processing speed-up.

"The EM algorithm is used **to find (local) maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly**. Typically, these models involve latent variables in addition to unknown parameters and known data observations."

"Maximum likelihood estimation is challenging on data in the presence of latent variables. **Expectation maximization provides an iterative solution to maximum likelihood estimation with latent variables**."

Accessible related Background:

- Expectation–maximization algorithm - Wikipedia
- https://www.statisticshowto.com/em-algorithm-expectation-maximization/
- https://towardsdatascience.com/expectation-maximization-explained-c82f5ed438e5
- https://personal.utdallas.edu/~prr105020/biol6385/2017/lecture/lecture_4_em_paper.pdf
- https://smartech.gatech.edu/bitstream/handle/1853/3281/02-20.pdf?sequence=1
- https://www.google.com/search?q=expectation-maximization+explained&rlz=1C1GCEJ_enUS1012US1012&oq=Expectation-Maximization&aqs=chrome.4.0i512l10.102341j0j15&sourceid=chrome&ie=UTF-8
- https://machinelearningmastery.com/expectation-maximization-em-algorithm/
- https://stats.stackexchange.com/questions/72774/numerical-example-to-understand-expectation-maximization
- https://stats.stackexchange.com/questions/550700/understanding-numerical-example-of-expectation-maximization
- https://stats.stackexchange.com/questions/62940/why-is-the-expectation-maximization-algorithm-used
- https://python-course.eu/machine-learning/expectation-maximization-and-gaussian-mixture-models-gmm.php
- http://robots.stanford.edu/papers/thrun.3d-planar-mapping.pdf **(not secure)**
- https://www.sciencedirect.com/topics/engineering/expectation-maximization-algorithm
- https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/slides/lec16-slides.pdf
- https://courses.csail.mit.edu/6.867/wiki/images/b/b5/Em_tutorial.pdf

As seen from the above brief sampling, there is no lack of material explaining the E-M algorithm in detail, with step-by-step examples to provide more of an intuitive understanding of what it can achieve. See 2022 paper: "**Deterministic Approximate EM Algorithm; Application to the Riemann Approximation EM and the Tempered EM" by Thomas Lartigue, Stanley Durrleman, and Stéphanie Allassonnière.**

There are also discussions of machine learning utilizations of the **E-M Algorithm** and of implementations in Python and in other popular modern computer languages. Recall that **The MathWorks** also has its own approach to **Machine learning** (**ML**), and so likely also has supporting results along these same lines. I know that **The MathWorks** already demonstrates how to utilize MatLab with Python (with calls both ways). One can search and see how to implement **E-M algorithm** in **R** (an open-source competitor to MatLab) and in other modern computer languages. My motivating interest here is how the **E-M Algorithm** (and its **approximate** implementations) can expedite the practical result of providing better, more accurate target tracking. This is my **only** motivating objective for bringing up this topic. However, applications differ!

In this provocative quick search, a surprising result is that some researchers have claimed a connection between one approach to applying an **E-M Algorithm** and calculating **Lower Bounds**. Since one of my own historical bailiwicks is in applying **C**ramer-**R**ao **L**ower **B**ound (**CRLB**) calculations to gauging the accuracy of Extended Kalman tracking Filters within the Missile Defense scenario, I will look further into this by viewing https://tminka.github.io/papers/minka-em-tut.pdf, while being cautious to **never** use anything from here within actual Test and **D**evelopment **software**. However, merely using recent ideas is fair game and still safe to do.