



Summary of this week

- WTLS (last week) gave great results but was computationally ill-conditioned for implementation in an embedded BMS
- This week, started by looking at TLS with proportional confidence on x_i and y_i
 - Had simple closed-form recursive solution, fading memory
- Generalized TLS to AWTLS
 - Does not optimize same cost function as WTLS, so not optimal in the same sense
 - But, has closed-form solution (Ferrari method), recursive and fading memory



Where from here?

- Next week, you will learn how to implement all the xLS methods in Octave code
 - Code has main `xLSalgos.m` component that does the computations
 - Also has “driver” code to execute `xLSalgos.m` with different input conditions
- You will see how the different methods perform under three different HEV and BEV operating scenarios
- Examples will also show the importance of a good SOC estimator



Credits

Credits for photos in this lesson

- “Blank slate” on slide 2: Pixabay license
 (<https://pixabay.com/en/service/license/>), from
<https://pixabay.com/en/white-male-3d-model-isolated-3d-1992593/>