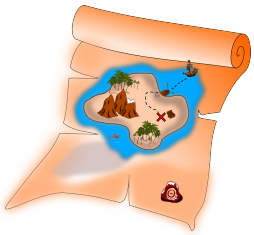
Summary of this week

- This week, you learned:
  - How to apply linear KF to linearized battery-cell model to help visualize the steps involved
  - How to simulate random systems as preparation for evaluating KF via simulation
  - How to implement Kalman filter in Octave/MATLAB and evaluate results computed by the filter
  - Some “improvements” to KF to handle real-world numeric round-off issues
  - How KF also provides helpful ways to detect sensor faults

Where from here?

- How to generalize state-state model to describe nonlinear systems
- How to apply two assumptions to derive the extended Kalman filter from the generic Gaussian sequential-probabilistic-inference solution
- How to write Octave/MATLAB code to implement EKF for a simple example and for the battery-cell state-estimation problem
- See some examples of applying EKF to estimate cell SOC using lab-test data



Credits

Credits for photos in this lesson

- “Map” on slide 2: Pixabay license (`https://pixabay.com/en/service/license/`), from `https://pixabay.com/en/map-treasure-map-treasure-pirates-162047/`