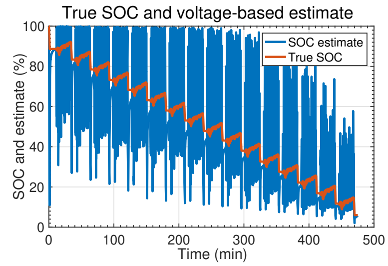
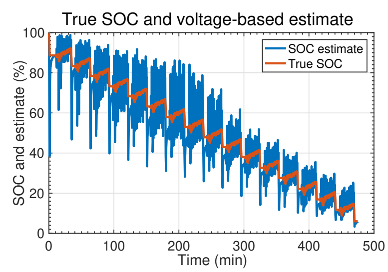
Poor, voltage-based method to estimate SOC

- We begin to consider the question, “how should we estimate battery cell state of charge?”
- One method would be to measure cell terminal voltage under load, $v(t)$, and look up on “SOC versus OCV” curve
  - `SOCest = SOCfromOCVtemp(v,T,model);`
  - Ignores effects of $i(t) \times R_0$ losses, diffusion voltages, and hysteresis on $v(t)$
  - Wide flat areas of OCV curve dilute accuracy of estimate — very noisy



Poor, voltage-based method to estimate SOC

- A modification of this method assumes a cell model $v(t) = \text{OCV}(z(t)) - i(t)R_0$ and then looks up $v(t) + i(t)R_0$ on “SOC versus OCV” curve
  - `SOCest = SOCfromOCVtemp(v+i*R0,T,model);`
  - Better, but still ignores effects diffusion voltages, hysteresis and so is still noisy
- Filtering helps but adds delay, which must be accounted for
- Hysteresis is another complicating factor
- Even though its estimates are noisy, we’ll find an application for this modified method in the next course



Poor, current-based method to estimate SOC

- Coulomb counting keeps track of charge in, out of cells via

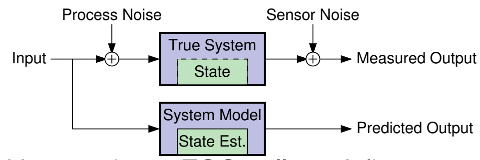
$$\hat{z}(t) = \hat{z}(0) - \frac{1}{\widehat{Q}} \int_0^t \eta(\tau) i_{\text{meas}}(\tau)\, \mathrm{d}\tau$$

$$i_{\text{meas}}(t) = i_{\text{true}}(t) + i_{\text{noise}}(t) + i_{\text{bias}}(t) + i_{\text{nonlin}}(t) + i_{\text{sd}}(t) + i_{\text{leakage}}(t)$$

  - Okay for short periods of operation when initial conditions are known or can be frequently “reset”
  - Subject to drift due to current sensor’s fluctuations, current-sensor bias, incorrect capacity estimate, other losses
- Uncertainty/error bounds grow (without limit) over time until estimate is “reset”

Model-based state estimation

- An alternative to a voltage-only method or a current-only method is somehow to combine the approaches
- Model-based estimators implement algorithms that use sensed measurements to infer internal hidden state of dynamic system



- Mathematical model of system is assumed known (e.g., ESC cell model)
- Same input propagated through true system, model, measured and predicted outputs compared; error used to update model's state estimate
  - Output error due to: state, measurement, model errors;
  - Update must be done carefully to account for all of these.

Linear Kalman filter

- Under specific conditions, Kalman filter (KF, special case of sequential probabilistic inference) gives optimal state estimate
- We study linear KF and some of its variants throughout remainder of course
- We start by assuming a general, possibly nonlinear, state-space model

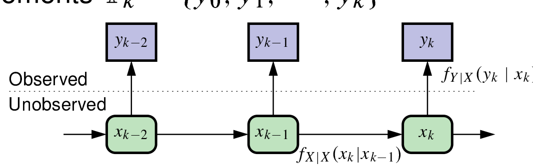
$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1})$$
$$y_k = h(x_k, u_k, v_k),$$

where u_k is known (measured) input signal, w_k is a process-noise random input, and v_k is a sensor-noise random input (ESC cell model fits this framework!)

- Functions $f(\cdot)$ and $h(\cdot)$ may be time-varying, but we generally omit time dependency from notation to avoid clutter

Sequential probabilistic inference

- KF is special case of sequential probabilistic inference (SPI):
  - Estimate present state x_k of dynamic system using all measurements $\mathbb{Y}_k = \{y_0, y_1, \cdots, y_k\}$



- The observations allow us to "peek" at what is happening in true system
  - Based on observations and model, we estimate state
- Process- and sensor-noise randomness *always* cause imperfect estimates
- So, to understand SPI solution, must study vector random processes

Summary

- Direct lookup of terminal voltage in OCV versus SOC table gives very poor estimate of cell SOC
- Modifying lookup to account for ohmic resistance helps, but not enough
- Coulomb counting also has many problems we would like to avoid
- Model-based state estimators combine voltage and current measurements, using a cell model to do so, to yield better state estimates
- Sequential probabilistic inference is the general framework that describes model-based state estimators of interest
- We will study SPI this week, as preparation for developing KF solution