Example simulations, HEV cases

- This week, we examine usage scenarios to exercise total-capacity estimation methods, compare their performance
- Use fading-memory version of all methods, omitting prefix "FM" for brevity
- Unless otherwise stated, fading-memory forgetting factor $\gamma = 1.0$
- We assume that the individual SOC estimates that are input to these methods can be determined to an accuracy of $\sigma_z = 0.01$
  - This is generous, as SPKF achieves only around $\sigma_z = 0.01$ for LMO and $\sigma_z = 0.03$ for LFP in practice, when Q_{nom} is used instead of Q in the estimator
  - EKF achieves only around $\sigma_z = 0.02$ or higher for LMO cells in practice
  - A nice advantage of both EKF and SPKF is that they give dynamic estimates of σ_z that ensure that the values of σ_{x_i} used in total capacity estimation are accurate

Why simulations?

- We use computer simulation to validate the algorithms as it allows constraining factors that would be difficult to control in a real-time embedded system, including:
  - The efficacy and accuracy of the SOC estimation algorithms used to provide input to the total-capacity estimation algorithms
  - Accuracy and precision of the raw sensor measurements used as input (including challenges of bias, nonlinear, and random errors, for example)
  - Repeatability of the experiment
  - Total capacity of a cell fades over time and difficult/impossible to know "true" value of total capacity with which to compare results
- Using synthetic data isolates performance of total-capacity estimation algorithms themselves, when all other factors are in some sense idealized

HEV scenario 1, generating x_i

- From perspective of total-capacity estimation, HEV characterized by narrow window of SOC used
  - We assume that the vehicle uses a SOC range of 40 $\%$ to 60 $\%$.
  - Each time total-capacity estimate updated, true SOC change ranges between ± 0.2, simulated by choosing of x_i to be uniform RV selected between these limits
- In HEV, battery pack is never fully charged to precisely known SOC; so, each time total-capacity estimate updated, two estimates of SOC are required to compute $x = z(t_2) - z(t_1)$, giving overall $\sigma_x^2 = 2\sigma_z^2 = 2(0.01)^2$
  - Simulate this by computing "measured" value of x_i to be equal to the true value of x_i added to a zero-mean Gaussian random number having variance σ_x^2

HEV scenario 1, generating y_i

- Compute true y_i as Q_{nom} of cell multiplied by true value of x_i
- Accumulated quantization errors comprise noise on y_i
- For y_i computed by summing m_i measurements, taken at a 1 Hz rate, from a sensor having quantizer resolution q, the total noise is $\sigma_{y_i}^2 = q^2 m_i/(12 \times 3600^2)$
- For HEV scenario 1, we assumed that the maximum range of the current sensor is $\pm 30 Q_{\text{nom}}$ and that a 10 bit A2D is used to measure current
- This leads to $q = 60 Q_{\text{nom}}/1024$
- We chose $m_i = 300$ s for every measurement and a nominal capacity of $Q_{\text{nom}} = 10$ Ah

Setting up simulation

- Each scenario we look at has similar setup code
- Initializes simulation parameters, then executes simulation

```
Q0 = 10;             % actual new-cell capacity of cell
maxI = 30*Q0;        % must be able to measure current up to +/- maxI
precisionI = 1024; % 10-bit precision on current sensor
slope = 0;
Qnom = 0;            % nominal capacity, possibly used for initializaiton
xmax = 0.2; xmin = -xmax; % range of the x(i) variables
m = 300;             % number of samples between updates
theCase = 1;         % fixed interval between updates
socnoise = sqrt(2)*0.01; % standard deviation of x(i)
Gamma = 1;           % forgetting factor
plotTitle = 'HEV Scenario 1';
runScenario
```


Running the simulation

- This second code segment, `runScenario.m`, runs an individual scenario, and plots results

```
n = 1000;                           % number of data points collected
Q = (Q0+slope*(1:n))';              % evolution of true capacity over time
x = ((xmax-xmin)*rand(n,1)+xmin);   % true x(i), without noise
y = Q.*x;                           % true y(i), without noise

binsize = 2*maxI/precisionI;        % resolution of current sensor
rn1 = ones(n,1);                    % init std. dev. for each measurement
sx = socnoise*rn1;                  % scale Gaussian std. dev.
if theCase == 1,                    % the typical case
  rn2 = rn1;                        % same scale on y(i) as x(i) noise
  sy = binsize*sqrt(m/12)/3600*rn2; % std. dev. for y(i)
else % this case will be discussed for BEV scenario 3
  mu = log(mode)+sigma^2; m  = 3600*lognrnd(mu,sigma,n,1);
  sy = binsize*sqrt(m/12)/3600;     % std. dev. for y(i)
end
x = x + sx.*randn(n,1);             % measured x(i) data, including noise
y = y + sy.*randn(n,1);             % measured y(i) data, including noise
```

Invoking algorithm, plotting results

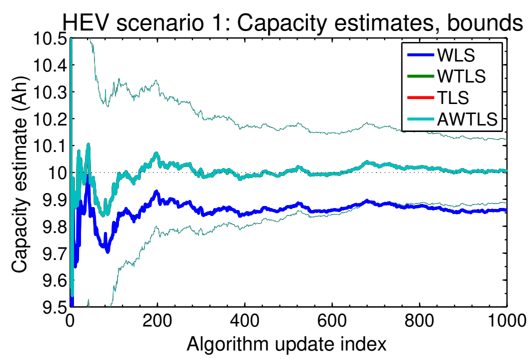
- Function `runScenario.m` continues by invoking algorithms and plotting results

```
[Qhat,SigmaQ,Fit] = xLSalgos(x,y,sx.^2,sy.^2,Gamma,Qnom);

figure; plot(Qhat); hold on % baseline plot of all estimates
xlabel('Algorithm update index');
ylabel('Capacity estimate (Ah)');
title(sprintf('%s: Capacity Estimates with Error Bounds',...
              plotTitle));
legend('WLS','WTLS','TLS','AWTLS','location','northeast');
plot(Qhat+3*sqrt(SigmaQ),'linewidth',0.5);
plot(Qhat-3*sqrt(SigmaQ),'linewidth',0.5);
plot(Qhat); % make sure estimate is plotted on top
plot(1:n,Q,'k:','linewidth',1);
```


Results for HEV scenario 1

- Recursive estimates not initialized prior to first data pair (equivalently, recursive parameters were initialized to zero)
- Estimates plotted as thick lines; three-sigma error bounds, computed using Hessian method, as thin lines
- WTLS, TLS, AWTLS estimates and error bounds identical under this scenario, converge to neighborhood of true total capacity
- WLS estimate is biased, error bounds are (incorrectly) so tight that they are indistinguishable from estimate itself



Summary

- Range of x_i for HEV from max delta SOC
- Noises on x_i assumed to be SOC-estimation errors
- Range of y_i corresponds to true x_i and true total capacity
- Noises on y_i assumed to be accumulated current-sensor quantization errors
- WTLS, TLS, AWTLS all give identical estimates and bounds for HEV scenario 1, all of which are very reasonable
- WLS clearly biased away from true total capacity, unrealistic confidence