Example of bar-delta, using desktop validation

- Three basic approaches to algorithm validation

PHYSICAL BMS:

- Final hardware and software, connected to physical battery pack; sometimes hard/impossible to determine “truth” of all cell states

HARDWARE IN THE LOOP (HIL):

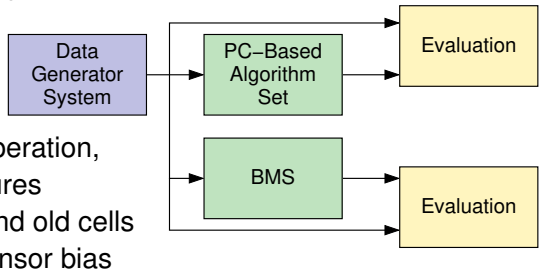
- Testing BMS algorithms on pre-recorded data, to compare to “truth” data

DESKTOP VALIDATION:

- Use model of cell to create synthetic test data
  - Allows access to “truth” of all cell and algorithm states
  - Very useful for tuning algorithms
  - Validity of results limited by the accuracy of cell model

Desktop validation

- Need “data generation” component:
  - Creates scenario-specific synthetic BMS data
- Need a “BMS algorithm simulation” component:
  - Simulates xKF algorithms using synthetic data as input, based on various initialization parameters



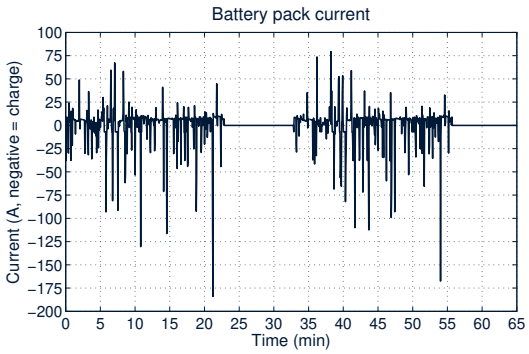
- Validation scenarios include: Normal operation, improper SOC initialization, sensor failures (fault + noise), temperature drift, new and old cells mixed, different drive cycles, current-sensor bias
- Insight and validation via analysis and display of outputs

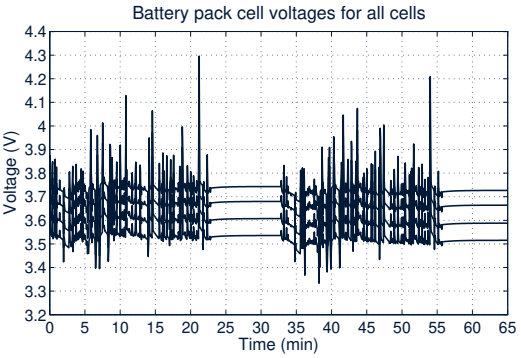
Scenarios to demonstrate bar delta

- Simulated four-cell pack with UDDS/rest/UDDS/rest
- Cells had true capacities of 6.5, 7.0, 7.5, and 8.0 Ah, resistances of 2.0, 2.25, 2.5, and 2.75 mΩ, and initial SOC values of 40, 45, 50, and 55 %. The current-sensor bias was 0.5 A
- Algorithms were initialized with all cells having estimated capacity of 6.2 Ah, estimated resistances of 2.25 mΩ, estimated current-sensor bias of 0 A, initial SOC estimates based on initial voltages
- SPKF was used for the bar filter and the SOC delta filters, and EKF was used for the resistance and capacity-inverse delta filters

Pack current, voltage response

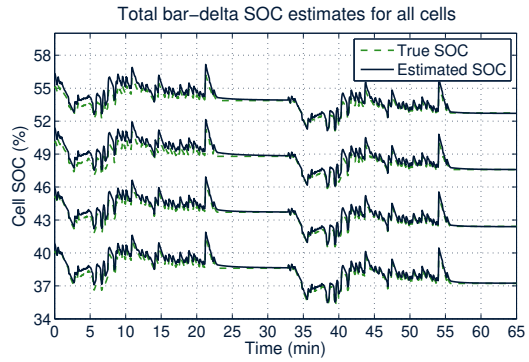
- Battery-pack electrical input current, individual cell voltage profiles for algorithm testing:

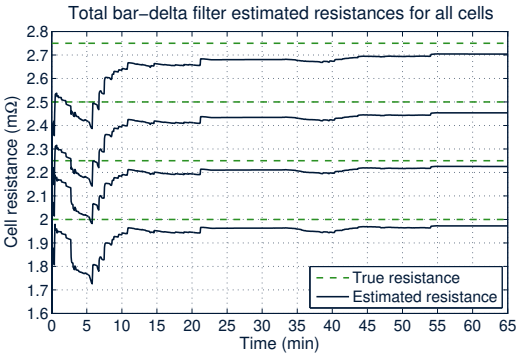




SOC and resistance results

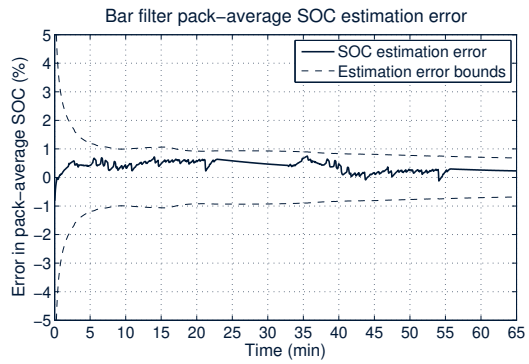
- Some results showing accuracy of the method:
  - Good convergence of both SOC and resistance estimates

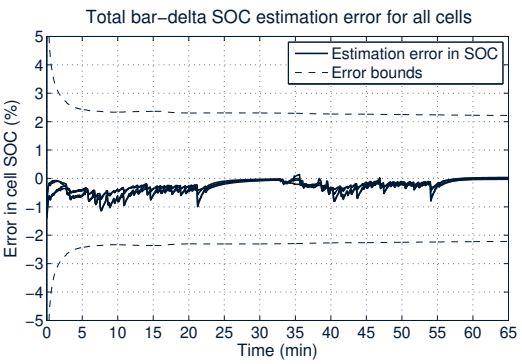




SOC estimation errors

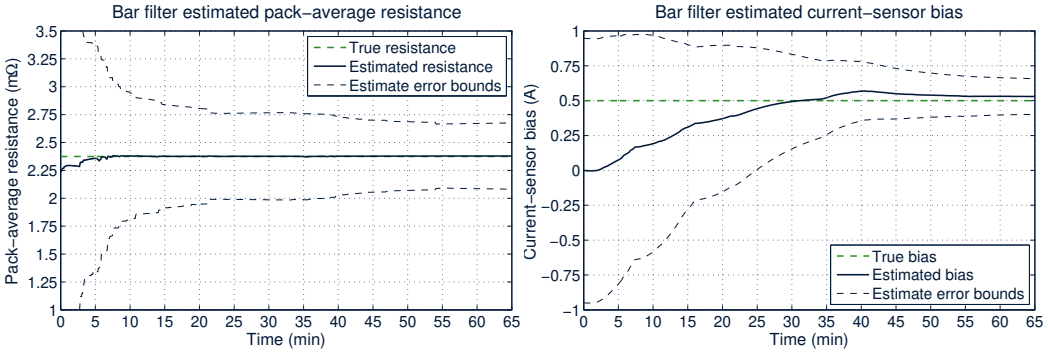
- A different point of view, with SOC estimation errors:
  - Good error bounds on both pack-average and cell SOCs





Other estimation accuracies

- Some other estimation accuracies:
  - Good error bounds on resistance, current-sensor bias



Is it faster?

- Capacity estimates evolve similarly to resistance estimates
  - But, adaptation time scale is much longer, as capacity is very weakly linked to output measurement
  - Abrupt changes in capacity won’t be tracked very quickly; but, capacity fade due to normal aging will be tracked very well
- Speedup of the method (hand-coded C code, run on G4 processor)

Description of test (for pack comprising 100 cells)	CPU time per iteration	Speedup
One SPKF per cell	5.272 ms	1.0
One pack bar filter only, no delta filters	0.067 ms	78.7
One pack bar filter, 100 delta filters updated per iter.	0.190 ms	27.7
One pack bar filter, 50 delta filters updated per iter.	0.123 ms	42.9

Summary

- Desktop validation is good preliminary approach to testing BMS algorithms
- Need to generate input data for all cases of interest, including typical operation and “corner cases”
- Example demonstrated results of bar-delta method using desktop validation
- Estimates, error bounds, speedup all as hoped