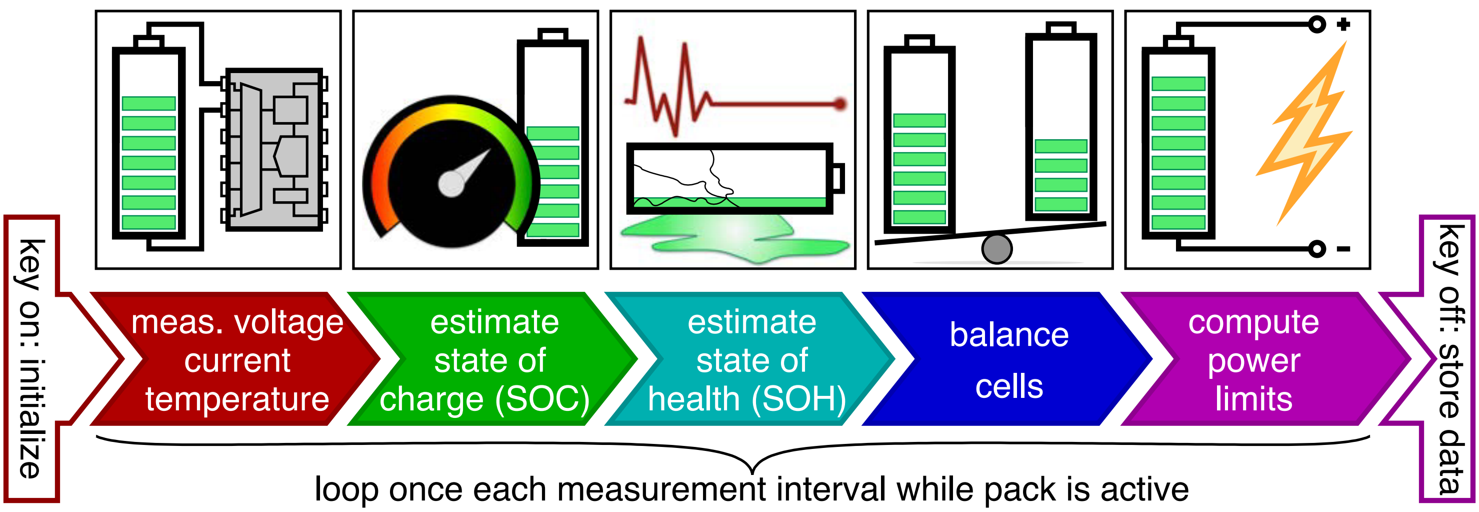
Voltage-Based Power-Limit Estimation

6.1: Problem definition

- We have now seen various methods to perform state estimation and health estimation for battery cells and packs.
- Using information on state and parameters, we have seen how to estimate the energy available in each cell and in the pack as well.
- The final major variables that we must estimate are cell power limits and battery-pack power limits.

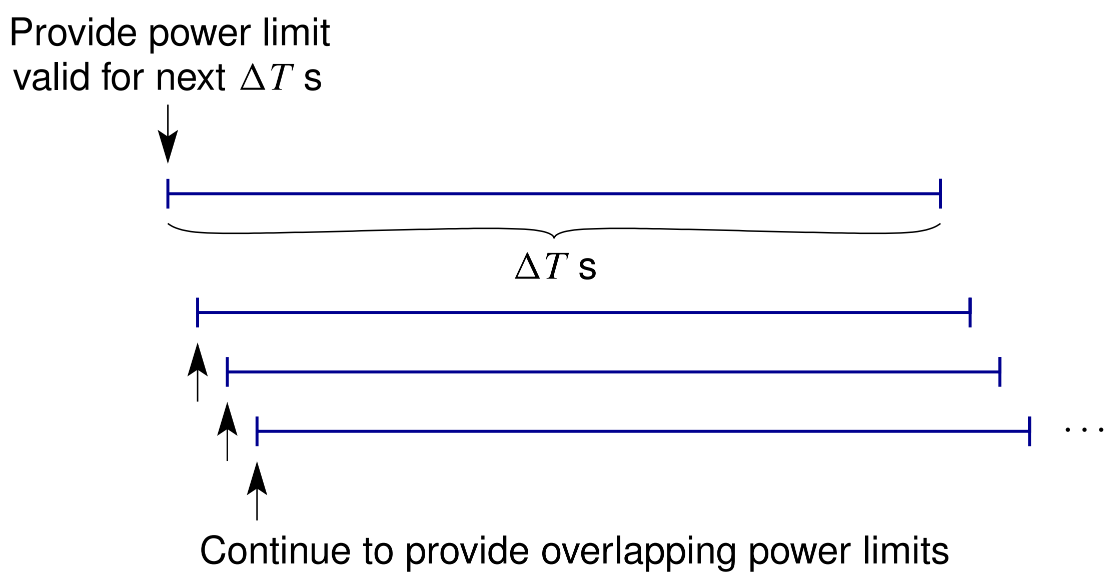


- A power limit tells us how quickly we may add or remove energy from the pack without violating a set of design constraints.
- In this chapter, we assume that the principal design constraints are on cell terminal voltage. This is the common practice today.
- The real issue, however, is not cell voltage, but the incremental damage that is experienced by the cell if it is operated at high rates.

- In the final chapter, we look at some ideas for computing power limits based on incremental damage rather than on terminal voltage.

Traditional, terminal-voltage-based power limits

- The power-limit calculations must be predictive:
  - They must specify limits on constant dis/charge power that are guaranteed to be "safe" over some future time horizon of ΔT s.



- Specifically, the problem we address in this chapter may be described in the following way:

a) Discharge power: Based on present battery-pack conditions, estimate the maximum discharge power that may be maintained constant for ΔT seconds without violating pre-set design limits on cell voltage, SOC, maximum design power, or current.

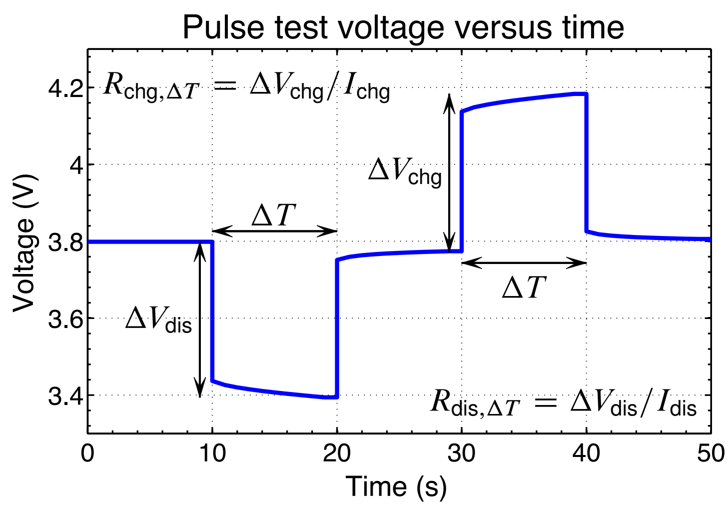
b) Charge power: Based on present battery-pack conditions, estimate the maximum battery charge power that may be maintained constant for ΔT seconds without violating pre-set design limits on cell voltage, SOC, maximum design power or current.

c) Both discharge and charge power: Any combination of (a) and (b), where ΔT may have different values for charge and discharge.

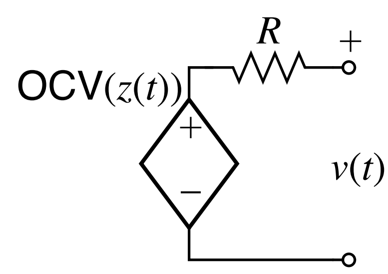
- The notation and assumptions we employ are as follows:
  - We denote the number of cells in the battery pack by N;
  - Cell voltage for cell number n in the pack by $v_n(t)$; where design limits $v_{\min} \le v_n(t) \le v_{\max}$ must be enforced for all cells;
  - State-of-charge by $z_n(t)$; where we enforce $z_{\min} \le z_n(t) \le z_{\max}$;
  - Cell power by $p_n(t)$; where we enforce $p_{\min} \le p_n(t) \le p_{\max}$; and,
  - Cell current by $i_n(t)$; where we enforce $i_{\min} \le i_n(t) \le i_{\max}$.
- Any particular limit ($v_{\max}$, $v_{\min}$, $z_{\max}$, $z_{\min}$, $i_{\max}$, $i_{\min}$, $p_{\max}$, $p_{\min}$) may be removed if desired by replacing its value by $\pm\infty$, as appropriate.
- Any limit may furthermore be a function of temperature and other factors pertaining to the present battery pack operating condition.
- Different cells may have different limits should it be desirable.
- Here, we assume that discharge current and power have positive sign and charge current and power have negative sign.
  - Other conventions are accommodated by minor math changes.
- The battery pack is assumed to comprise N_s cell modules connected in series, where each cell module comprises N_p individual cells connected in parallel, with $N_s \ge 1$, $N_p \ge 1$, and $N = N_s N_p$.

6.2: Voltage-based rate limits, using simple cell model

- As previewed in chapter 1, a standard method is one we will refer to as the Hybrid Pulse Power Characterization (HPPC) method specified by the Partnership for New Generation Vehicles (PNGV).
- Power is calculated to enforce limits on cell terminal voltage, predictive over the next ΔT s , updating at a faster rate than once every ΔT s.



- Must run cell tests: compute, store resistances at different SOCs and temperatures.
- We assume a simplified cell model



$$v(t) = \text{OCV}(z(t)) - i(t)R,$$

or

$$i(t) = \frac{\text{OCV}(z(t)) - v(t)}{R}.$$

- To compute a power estimate, we first assume we are concerned only with keeping the terminal voltage between $v_{\min}$ and $v_{\max}$.
- For discharge power, set $R = R_{\text{dis},\Delta T}$ and clamp $v(t) = v_{\min}$. Then, we may calculate the maximum discharge current as constrained by voltage as

$$i_{\max,n}^{\text{dis,volt}} = \frac{\text{OCV}\,(z_n(t)) - v_{\min}}{R_{\text{dis},\Delta T}}.$$

- Pack discharge power is then calculated as

$$P_{\max}^{\text{dis}} = N_s N_p v_{\min} \min_n \left(i_{\max,n}^{\text{dis,volt}}\right).$$

- For charge power, set $R = R_{\text{chg},\Delta T}$ and clamp $v(t) = v_{\max}$.
- Note, however, that charge current is assumed negative in sign by convention, so that maximum-magnitude current is a minimum in the signed sense. It is

$$i_{\min,n}^{\text{chg,volt}} = \frac{\text{OCV}\,(z_n(t)) - v_{\max}}{R_{\text{chg},\Delta T}}.$$

- Pack charge power is then calculated as

$$P_{\min}^{\text{chg}} = N_s N_p v_{\max} \max_n \left(i_{\min,n}^{\text{chg,volt}} \right).$$

Rate limits based on SOC, maximum current, and power

- We can quite easily extend the basic HPPC method to also include SOC-based limits with a time horizon ΔT.
- This may be done as follows. First, for a constant current i_n, the SOC recurrent relationship is:

$$z_n(t + \Delta T) = z_n(t) - (\eta_n \Delta T/Q) i_n.$$

- Assume $\eta_n = 1$ for discharge, and $\eta_n = \eta \le 1$ for charge currents.
- If we have design limits such that $z_{\min} \le z_n(t) \le z_{\max}$ for all cells in the pack, we can compute current i_n to enforce these limits.
- Simple algebra gives current limits based on the SOC of each cell

$$i_{\max,n}^{\text{dis,soc}} = \frac{z_n(t) - z_{\min}}{\Delta T/Q}$$

$$i_{\min,n}^{\text{chg,soc}} = \frac{z_n(t) - z_{\max}}{\eta \Delta T/Q}.$$

- Side information on SOC-estimate uncertainty (*e.g.*, from a Kalman filter) can be used to make power estimates more conservative.

- This is done as (assuming here that we desire to use a $3\sigma_z$ confidence interval)

$$i_{\max,n}^{\text{dis,soc}} = \frac{(z_n(t) - 3\sigma_{z,n}) - z_{\min}}{\Delta T/Q}$$
$$i_{\min,n}^{\text{chg,soc}} = \frac{(z_n(t) + 3\sigma_{z,n}) - z_{\max}}{\eta\Delta T/Q}.$$

- Once all cell current limits have been calculated, the pack discharge and charge currents with all limits enforced are computed as

$$i_{\max}^{\text{dis}} = \min\left(i_{\max}, \min_n i_{\max,n}^{\text{dis,soc}}, \min_n i_{\max,n}^{\text{dis,volt}}\right)$$
$$i_{\min}^{\text{chg}} = \max\left(i_{\min}, \max_n i_{\min,n}^{\text{chg,soc}}, \max_n i_{\min,n}^{\text{chg,volt}}\right).$$

- Power may be calculated using the sum of all cell powers, using the maximum allowed current and the predicted future voltage.

$$P_{\min}^{\text{chg}} = N_p \max\left(N_s p_{\min}, \sum_{n=1}^{N_s} i_{\min}^{\text{chg}} v_n(t+\Delta T)\right)$$
$$\approx N_p \max\left(N_s p_{\min}, \sum_{n=1}^{N_s} i_{\min}^{\text{chg}}\left(\text{OCV}\left(z_n(t) - i_{\min}^{\text{chg}}\frac{\eta\Delta T}{Q}\right) - i_{\min}^{\text{chg}} R_{\text{chg},\Delta T}\right)\right);$$
$$P_{\max}^{\text{dis}} = N_p \min\left(N_s p_{\max}, \sum_{n=1}^{N_s} i_{\max}^{\text{dis}} v_n(t+\Delta T)\right)$$
$$\approx N_p \min\left(N_s p_{\max}, \sum_{n=1}^{N_s} i_{\max}^{\text{dis}}\left(\text{OCV}\left(z_n(t) - i_{\max}^{\text{dis}}\frac{\Delta T}{Q}\right) - i_{\max}^{\text{dis}} R_{\text{dis},\Delta T}\right)\right).$$

6.3: Voltage-based rate limits, using comprehensive cell model

- This enhanced version of the HPPC method is still limited:
  - The cell model used is too primitive to give precise results. Overly optimistic or pessimistic values could be generated, either posing a safety or battery-health hazard or being inefficient in battery use.
  - Further, the equations assume initial equilibrium condition, which is not true in general.
- Hence, we usually de-rate the HPPC estimates by some “trust factor.”
- A better cell model, combined with a maximum-power algorithm that uses the cell model, can give better power prediction.
- We now assume a more accurate model of cell dynamics in a discrete-time state-space form

$$x_n[k+1] = f(x_n[k], u_n[k])$$

$$v_n[k] = h(x_n[k], u_n[k]),$$

where k is the discrete time sample index.

  - Either the physics-based model from ECE5710 or the ESC model focused on in this course may be used within this framework.
- Also assume that ΔT seconds may be represented in discrete time as exactly $k_{\Delta T}$ sample intervals.
- Then, we can use this model to predict cell voltage ΔT seconds into the future by

$$v_n[k+k_{\Delta T}] = h(x_n[k+k_{\Delta T}], u_n[k+k_{\Delta T}]),$$

where $x_n[k+k_{\Delta T}]$ may be found by simulating the state equation for $k_{\Delta T}$ time samples.

- We assume that the input to each cell remains constant from time index m to $m + k_{\Delta T}$, and denote it simply as u_n.
- The method then uses a bisection search algorithm—to be elaborated on later—to find $i_{\max,n}^{\text{dis,volt}}$ and $i_{\min,n}^{\text{chg,volt}}$ by looking for the i_n (as a member of the u_n vector) that causes equality in

$$v_{\min} = h(x_n[k + k_{\Delta T}], u_n), \quad \text{or}$$

$$0 = h(x_n[k + k_{\Delta T}], u_n) - v_{\min}$$

to find $i_{\max,n}^{\text{dis,volt}}$, and by looking for the i_n that causes equality in

$$v_{\max} = h(x_n[k + k_{\Delta T}], u_n), \quad \text{or}$$

$$0 = h(x_n[k + k_{\Delta T}], u_n) - v_{\max}$$

to find $i_{\min,n}^{\text{chg,volt}}$.

- A special case is when the state equation is linear—that is, when

$$x_n[k+1] = Ax_n[k] + Bu_n[k],$$

where A and B are constant matrices.

- Then, for input u_n constant over the entire prediction horizon, we have

$$x_n[k + k_{\Delta T}] = A^{k_{\Delta T}} x_n[k] + \left(\sum_{j=0}^{k_{\Delta T}-1} A^{k_{\Delta T}-1-j} B \right) u_n.$$

- Most of these terms may be pre-computed without knowledge of u_n in order to speed calculation using the bisection algorithm.
- Once again, SOC-based current limits $i_{\max,k}^{\text{dis,soc}}$ and $i_{\min,k}^{\text{chg,soc}}$ are computed as before.

- Power is then computed as

$$
\begin{aligned}
P_{\min}^{\text{chg}} &= N_p \sum_{n=1}^{N_s} i_{\min}^{\text{chg}} v_n(t+\Delta T) \\
&= N_p \sum_{n=1}^{N_s} i_{\min}^{\text{chg}} h(x_n[k+k_{\Delta T}], u_n),
\end{aligned}
$$

with u_n containing $i_{\min}^{\text{chg}}$ as its value for current, and

$$
\begin{aligned}
P_{\max}^{\text{dis}} &= N_p \sum_{n=1}^{N_s} i_{\max}^{\text{dis}} v_n(t+\Delta T) \\
&= N_p \sum_{n=1}^{N_s} i_{\max}^{\text{dis}} h(x_n[k+k_{\Delta T}], u_n),
\end{aligned}
$$

with u_n containing $i_{\max}^{\text{dis}}$ as its value for current.

- All that remains is to see how to determine u_n to meet the cell voltage limits.
  - We look at this in the next topic.

6.4: Bisection search

- To solve

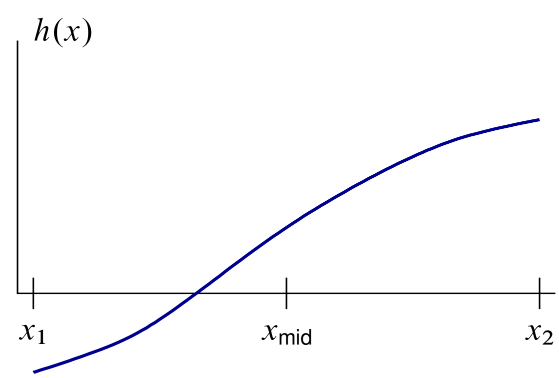
$$0 = h(x_n[k + k_{\Delta T}], u_n) - v_{\min}$$

for u_n leading to $i_{\max,n}^{\text{dis,volt}}$, or to solve

$$0 = h(x_n[k + k_{\Delta T}], u_n) - v_{\max}$$

for u_n leading to $i_{\min,n}^{\text{chg,volt}}$, we require a method to solve for a root of a nonlinear equation.

- Here, we use the bisection search algorithm to do so.
- The bisection search algorithm looks for a root of $h(x)$ (*i.e,* a value of x such that $h(x) = 0$) where it is known *a priori* that at least one root lies between values $x_1 <$ root $< x_2$.
    - One way of knowing that a root lies in this interval is that the sign of $h(x_1)$ is different from the sign of $h(x_2)$.
- Each iteration of the bisection algorithm evaluates the function at the midpoint $x_{\text{mid}} = (x_1 + x_2)/2$.
- Based on the sign of the evaluation, either x_1 or x_2 is replaced by x_{mid} to retain different signs on $h(x_1)$ and $h(x_2)$.



- The root-location uncertainty is halved by this algorithmic step.
- This bisection iteration is repeated until the interval between x_1 and x_2, (i.e., the resolution of the root of $h(x)$) is as small as desired.
- If ε is the desired root resolution, the algorithm will require at most $\lceil \log_2 (|x_2 - x_1|/\varepsilon) \rceil$ iterations. The bisection method is listed below.

```
% Search interval x1...x2 in fn h(.) for root, with tolerance tol
function x = bisect(h,x1,x2,tol)
  jmax = ceil(log2(abs(x2-x1)/tol));

  dx = x2 - x1; % set the search interval dx = x2 - x1
  if( h(x1) >= 0 )
    dx = -dx; x1 = x2;     % root now b/w (x1,x1 + dx), and h(x1) < 0
  end

  for jj = 1:jmax
    dx = 0.5 * dx; xmid = x1 + dx;

    if h(xmid) <= 0,
      x1 = xmid;
    elseif abs(dx) <= tol,
      break
    end
  end
  x = x1 + 0.5*dx;
end
```

- An example of how to run this algorithm is (returns `-9.5367e-07`):

```
h = @(x) x^3;
bisect(h,-1,2,1e-5)
```

- Bisection is incorporated in the overall algorithm as follows.
  - First, three simulations are performed to determine cell voltages $k_{\Delta T}$ samples into the future for cell current $i_n = 0$, $i_{\min}$, and $i_{\max}$.
  - If cell voltages are predicted to be between $v_{\min}$ and $v_{\max}$ for the maximum rates, then the maximum rates may be used.
  - If the cell voltages, even during rest, are outside of bounds, then set the maximum rates to zero.
  - Otherwise, we know that the true maximum rate may be found by bisecting between rate equal to zero and its maximum value.
  - Bisection is performed between current limits $(i_{\min}, 0)$ or $(0, i_{\max})$.

- To estimate power limits using bisection and an ESC model, we need to define a bisection cost function, which will involve a $k_{\Delta T}$-second prediction routine.
- The ESC-model state equation is linear, with

$$x_n[k+1] = Ax_n[k] + Bu_n[k].$$

- We first define matrix functions to compute state-space A and B based on input current:

```
A = @(ik) diag([1 exp(-1/(RC)) exp(-abs(ik*Gamma/(3600*Q)))]);
B = @(ik) [-1/(3600*Q) 0; (1-exp(-1/RC)) 0; ...
           0 (1-exp(-abs(ik*Gamma/(3600*Q))))];
```

- Then, for input u_n constant over the entire prediction horizon, we have

$$x_n[k+k_{\Delta T}] = A^{k_{\Delta T}} x_n[k] + \left(\sum_{j=0}^{k_{\Delta T}-1} A^{k_{\Delta T}-1-j} B\right) u_n.$$

- Because A is diagonal, the matrix power $A^{k_{\Delta T}}$ is simply the diagonal matrix comprising the scalar power of the diagonal elements.
- Similarly, the summation can be written as

$$\begin{aligned}
\sum_{j=0}^{k_{\Delta T}-1} A^{k_{\Delta T}-1-j} &= \left(\sum_{j=0}^{k_{\Delta T}-1} A^{-j}\right) A^{k_{\Delta T}-1} = \left(\sum_{j=0}^{k_{\Delta T}-1} \left(A^{-1}\right)^j\right) A^{k_{\Delta T}-1} \\
&= \left(I - A^{-1}\right)^{-1}\left(I - A^{-k_{\Delta T}}\right) A^{k_{\Delta T}-1} \\
&= \left(I - A^{-1}\right)^{-1}\left(A^{k_{\Delta T}-1} - A^{-1}\right) \\
&= (A - I)^{-1}\left(A^{k_{\Delta T}} - I\right).
\end{aligned}$$

- This allows us to write very efficient code to simulate a cell $k_{\Delta T}$ samples into the future.

```
% Simulate cell for KDT samples, with input current equal to ik, initial
% state = x0, A and B functions, temperature = T, with model parameters
% R0, R, M and the model structure "model".
function [vDT,xDT] = simCellKDT(ik,x0,A,B,KDT,T,model,R0,R,M)
  Amat = A(ik); Bmat = B(ik); dA = diag(Amat);
  if ik == 0,
    ADT = diag([KDT, (1-dA(2)^KDT)/(1-dA(2)), KDT]);
  else
    ADT = diag([KDT, (1-dA(2)^KDT)/(1-dA(2)), (1-dA(3)^KDT)/(1-dA(3))]);
  end
  xDT = (dA).^KDT.*x0 + ADT*Bmat*[ik; sign(ik)];
  vDT = OCVfromSOCtemp(xDT(1),T,model) - R*xDT(2) + M*xDT(3) - ik*R0;
end
```

- Can now write a bisection "cost" functions. For example, if we consider only terminal voltage and SOC limits, we have for discharge:

```
function h = bisectDischarge(ik,x0,A,B,KDT,T,model,R0,R,M)
  [vDT,xDT] = simCellKDT(ik,x0,A,B,KDT,T,model,R0,R,M);
  h = max(minV - vDT,zmin - xDT(1)); % max must be less than zero
end
```

  and charge:

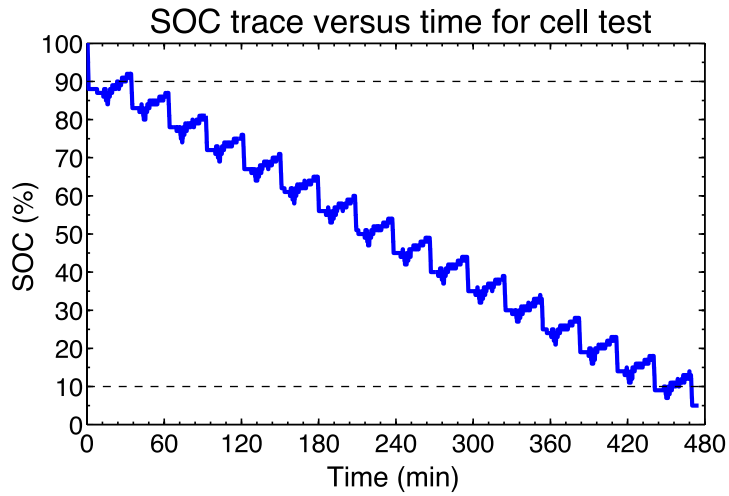
```
function h = bisectCharge(ik,x0,A,B,KDT,T,model,R0,R,M)
  [vDT,xDT] = simCellKDT(ik,x0,A,B,KDT,T,model,R0,R,M);
  h  = min(maxV - vDT,zmax - xDT(1)); % min must be greater than zero
end
```

- To use one of these functions, we use code like:

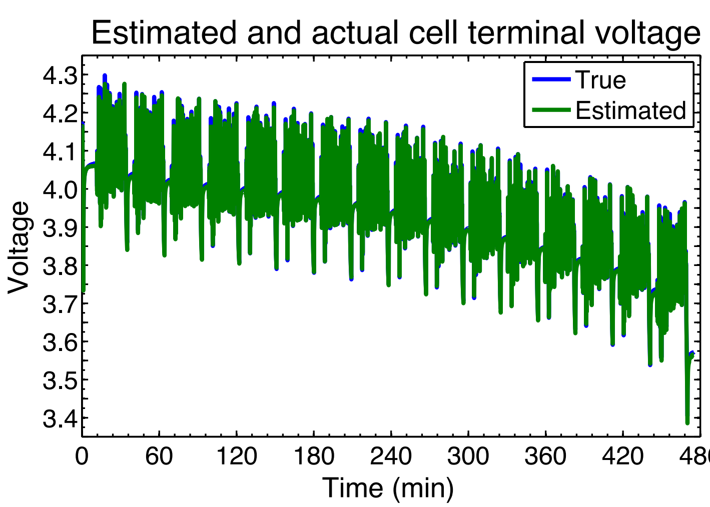
```
h = @(x) bisectDischarge(x,x0,A,B,KDT,T,model,R0,R,M)
ilimit = bisect(h,imin,imax,itol);
```

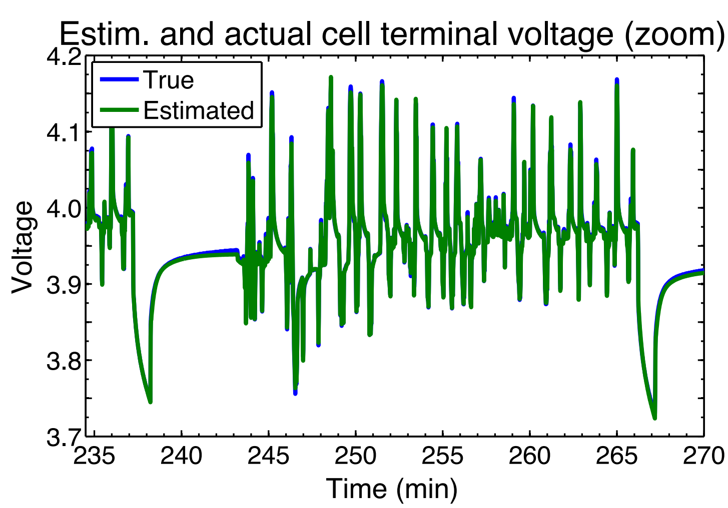
6.5: Power-limits estimation example

- We close this chapter with an example showing the similarities and differences between the two methods discussed herein.
- A cell is subjected to a sequence of sixteen UDDS cycles, separated by discharge pulses and five-minute rests.
- SOC increases by about 5 % during each UDDS cycle, but is brought down about 10 % during each discharge between cycles.



- The entire operating range for these cells (10 % SOC to 90 % SOC, delineated on the figure as the region between the thin dashed lines) is excited during the cell test.
- An ESC cell model is fit to the results, and the difference between true cell terminal voltage and estimated cell terminal voltage is very small (RMS voltage estimation error of less than 5 mV).

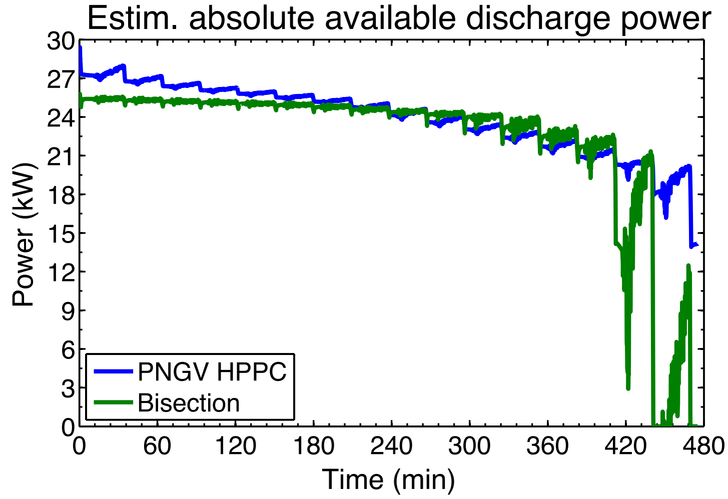




- For the following results, we assume a pack of LiPB cells with $N_s = 40$ and $N_p = 1$.
- Cells have nominal capacity of 7.5 Ah, and $\Delta T = 10$ s for both charge and discharge.
- Operational limits for the power calculations are listed.

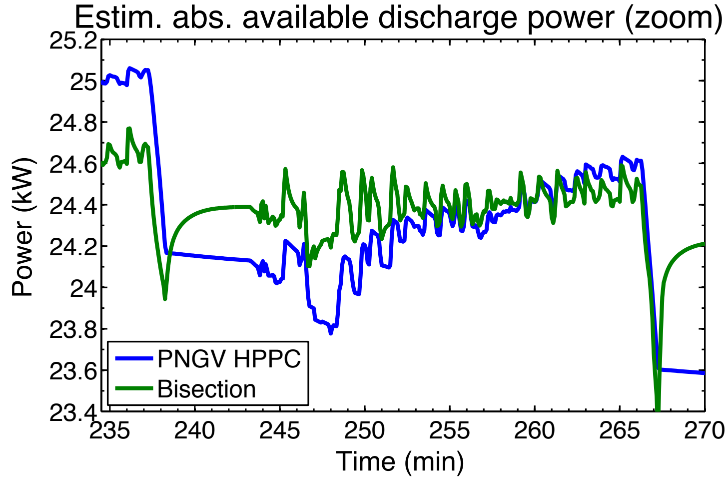
Parameter	Minimum	Maximum
$v_n(t)$	3.0 V	4.35 V
$i_n(t)$	−200 A	200 A
$z_n(t)$	0.1	0.9
$p_n(t)$	$-\infty$	∞

- Discharge power estimates are plotted to the right.
- In the discussion that follows, we consider the results of bisection method to be the “true” capability of the cell.

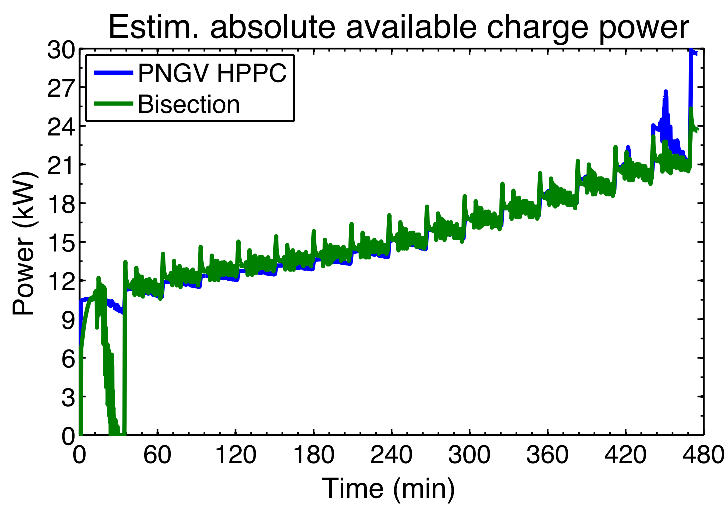


- We justify this assumption by the fidelity of the cell model’s voltage estimates, as supported by the plots on the prior page.
- First, we see that the two methods produce similar estimates.
- At high SOCs, the PNGV HPPC method predicts higher power than is actually available (by as much as 9.8 %), and at mid-to-low SOCs the PNGV HPPC method under-predicts the available power.
- Only the bisection method included SOC bounds, which explain why the predictions are so different at low SOC.
- If the vehicle controller were to discharge at the rates predicted by the PNGV HPPC method, the cell would be over-discharged in some cases (lowering its lifetime), and under-utilized in other cases.

- The figure to the right zooms in on a mid-SOC region to show greater detail.
- In this region, the methods produce nearly identical predictions.

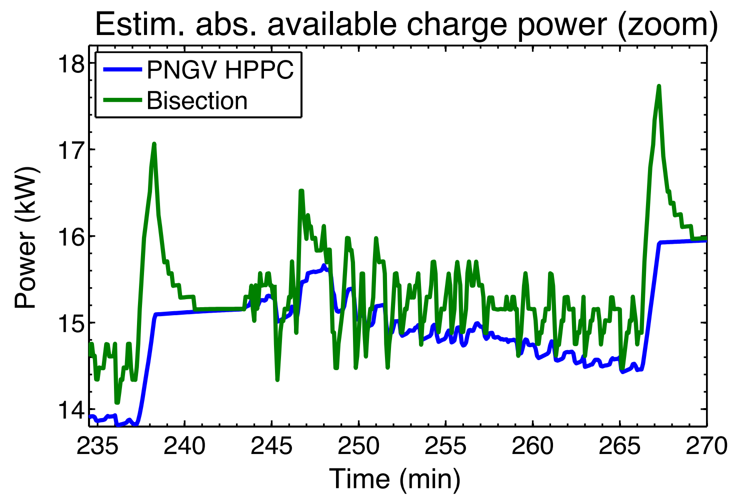


- A notable feature of the bisection method, however, is that it takes into account the entire dynamics of the cell when making a prediction.
- Therefore, the strong discharges at around time 237 and 267 minutes draw the cell voltage down, and allows less discharge power than the HPPC method, which only consider SOC when making its estimate.
- The three methods are also compared with respect to charge power, shown to the right.
- Absolute power is shown (charge power is computed as a negative value).



- Again, at this scale, the estimates appear nearly identical.
- Again, the PNGV HPPC method does not consider SOC limits, so over-predicts charge power at high SOCs.
- It also over-predicts power at low SOCs as it ignores the increase to charge resistance at low SOC.

- A zoom of this plot is shown to the right, which accentuates the differences between the predictions.



- Here, we see that the strong discharges at around time 237 and 267 minutes allow for greater charging power, as the voltage will not quickly change.

Conclusions

- In this chapter we have presented two methods to predict battery discharge and charge power that incorporate voltage, state-of-charge, power and current design constraints, and work for a user-specified prediction horizon ΔT.
- The results indicate that the two methods produce very similar results to each other.
- The bisection method requires significantly more computation, and a good cell model.
- But, if a Kalman filter is being used to estimate SOC, then the cell model will already be present and the state will be available for use.
- The bisection method produces dynamic power estimates, and is able to take advantage of recent strong discharges to increase the temporary available charge power, and is able to take advantage of

recent strong charges to increase temporary available discharge power.

Where from here?

- The implicit assumption in this chapter is that power should be computed to enforce voltage limits on a cell.
- This is not the true issue, however. Really, we are trying to minimize the incremental degradation that is being experienced by the cell.
  - For example, even in present BMS we usually derate the power limits at warm temperatures to slow down temperature rise.
  - But, this hardly begins to address a detailed optimized strategy based on aging.
- To understand this better, we need to discuss how cells age, and then some more advanced power-estimation algorithms that take advantage of these aging models.