Computing battery-pack available power

- Recall that HPPC cell discharge power is computed

$$P_{\text{dis}}^{\text{cell}\,j} = v_j(t)i_j(t) = v_{\min}\frac{\text{OCV}(z_j(t)) - v_{\min}}{R_{\text{dis},\Delta T,j}}$$

- Computing power in series multi-cell battery pack, still clamp $v(t) = v_{\min}$
- But, must use minimum limiting current, multiply by number of cells in series

$$P_{\text{dis}}^{\text{pack}} = \sum_{j=1}^{N_s} v_j(t)i_j(t) = N_s v_{\min} \min_j \left(\frac{\text{OCV}(z_j(t)) - v_{\min}}{R_{\text{dis},\Delta T,j}} \right)$$

- Similarly, for charge (remembering charge power is negative)

$$P_{\text{chg}}^{\text{pack}} = \sum_{j=1}^{N_s} v_j(t)i_j(t) = N_s v_{\max} \max_j \left(\frac{\text{OCV}(z_j(t)) - v_{\max}}{R_{\text{chg},\Delta T,j}} \right)$$

Computing battery-pack total energy

- Recall that cell total energy is computed as

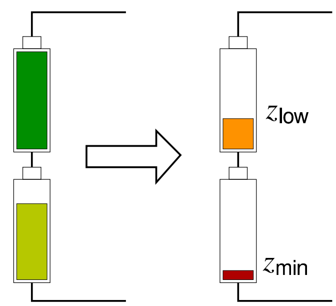
$$E(t) = Q\int_{z_{\min}}^{z(t)} \text{OCV}(\xi)\,\mathrm{d}\xi \approx QV_{\text{nom}}\Delta z$$

- In a battery, each cell may have different Q_j and $z_j(t)$
- Computing battery-pack energy is a three-step process
  1. Determine minimum Ah to discharge *any* cell to $z_{\min}$
     - Start with generic SOC equation

$$z(t) = z(0) - \text{Ah discharged}/Q$$

     - Set $z(0) = z_j(t)$, $z(t) = z_{\min}$, $Q = Q_j$, rearrange

$$\text{Ah discharged} = \min_j Q_j\left(z_j(t) - z_{\min}\right)$$



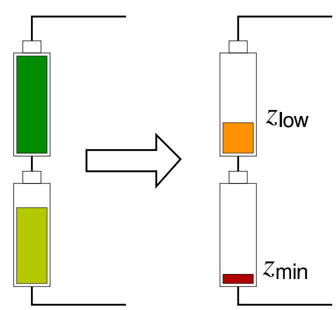
Computing battery-pack total energy (cont.)

2. For this many Ah discharged, compute resulting SOC of all cells:

$$z_{\text{low},j} = z_j(t) - \frac{\text{Ah discharged}}{Q_j}$$

3. Compute total battery discharge energy

$$E_{\text{pack}}(t) = \sum_{j=1}^{N_s} Q_j \int_{z_{\text{low},j}}^{z_j(t)} \text{OCV}(\xi)\,\mathrm{d}\xi$$



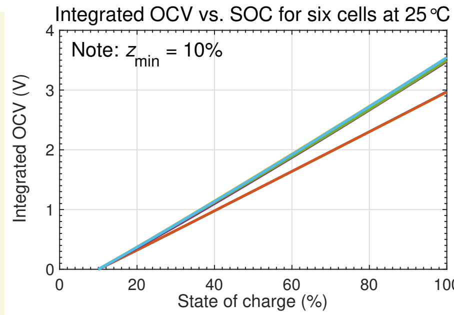
- Reminder: Can't extract all energy at high rates and cold temperatures, still need power-limit estimates

Using lookup table for efficiency

- Integrated OCV can be stored in look-up table (LUT) for “instant” computation

```
% zRef is vector of SOC points, e.g.,
%   zRef = zmin:0.01:1;
% ocvVec is a vector of OCV values
% corresponding to each SOC point.
% ivzRef is integrated OCV function:
ivzRef = cumtrapz(zRef,ocvVec);

% Table lookup uses "interp1.m" to find
% OCV integrated between zmin and "z".
ivz = interp1(zRef,ivzRef,max(zmin,z));
```



Summary

- Can now compute estimates of battery-pack power over future time horizon and battery pack total energy by expanding on how we computed cell power and energy
- Have also seen first example Octave/MATLAB code to implement a look-up table
- This completes our introduction to requirement 4 for BMS (estimation)
  - Will spend much more time examining (better) methods to do so in courses 3–5
- Will spend the remainder of this week considering BMS requirement 5, diagnostics