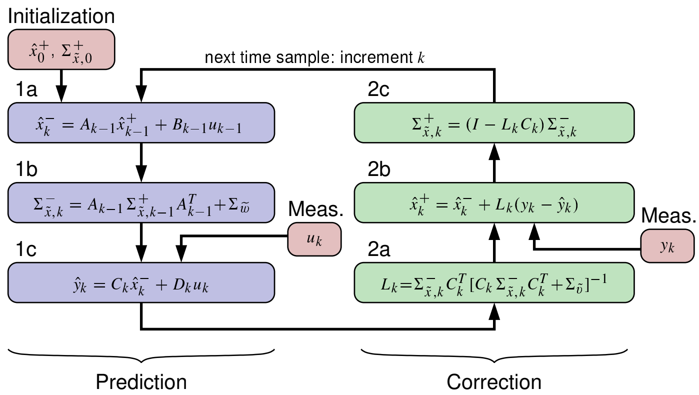
Visualizing the linear Kalman filter

- Linear Kalman-filter equations naturally form a recursion:



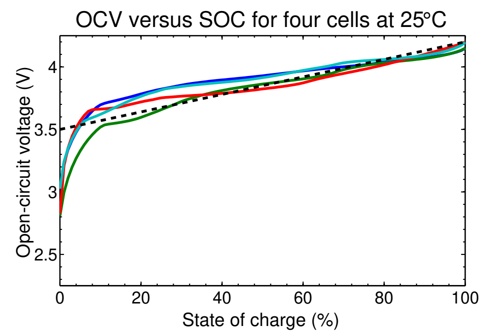
- "Simple" to implement on a digital computer
- However, note that our cell models are nonlinear, so we cannot apply (linear) Kalman filter to them directly

Setting up a demonstration (1)

- To demonstrate the KF steps, we'll develop and use a crude cell model

$$z_{k+1} = 1 \cdot z_k - \frac{1}{3600 \cdot Q} i_k$$
$$\text{volt}_k = 3.5 + 0.7 \times z_k - R_0 i_k$$

- Notice that we have:
  - Linearized the OCV relationship
  - Omitted diffusion voltages
  - Omitted hysteresis voltages

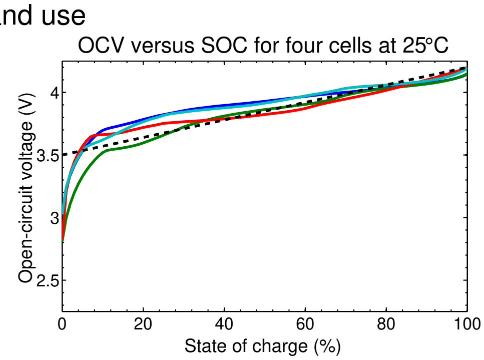


Setting up a demonstration (2)

- Model still isn't linear because of "3.5" in output equation
- Debias measurement via $y_k = \text{volt}_k - 3.5$ and use

$$z_{k+1} = 1 \cdot z_k - \frac{1}{3600 \cdot Q} i_k$$
$$y_k = 0.7 \times z_k - R_0 i_k$$

- Define state $x_k \equiv z_k$ and input $u_k \equiv i_k$
- For the sake of example, we will use $Q = 10000/3600$ and $R_0 = 0.01$
- Yields state-space description with $A = 1$, $B = -1 \times 10^{-4}$, $C = 0.7$, and $D = -0.01$
- We also model $\Sigma_{\widetilde{w}} = 10^{-5}$, and $\Sigma_{\tilde{v}} = 0.1$
- We assume no initial uncertainty so $\hat{x}_0^+ = 0.5$ and $\Sigma_{\tilde{x},0}^+ = 0$



Iteration 1

- We look at the first iteration of the linear KF, assuming $i_0 = 1$, $i_1 = 0.5$ and $v_1 = 3.85$

$\hat{x}_k^- = A_{k-1}\hat{x}_{k-1}^+ + B_{k-1}u_{k-1}$	$\hat{x}_1^- = 1 \times 0.5 - 10^{-4} \times 1 = 0.4999$
$\Sigma_{\tilde{x},k}^- = A_{k-1}\Sigma_{\tilde{x},k-1}^+ A_{k-1}^T + \Sigma_{\tilde{w}}$	$\Sigma_{\tilde{x},1}^- = 1 \times 0 \times 1 + 10^{-5} = 10^{-5}$
$\hat{y}_k = C_k\hat{x}_k^- + D_k u_k$	$\hat{y}_1 = 0.7 \times 0.4999 - 0.01 \times 0.5 = 0.34493$
$L_k = \Sigma_{\tilde{x},k}^- C_k^T [C_k \Sigma_{\tilde{x},k}^- C_k^T + \Sigma_{\tilde{v}}]^{-1}$	$L_1 = 10^{-5} \times 0.7[0.7^2 \times 10^{-5} + 0.1]^{-1}$ $= 6.99966 \times 10^{-5}$
$\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$ (where $y_k = 3.85 - 3.5$)	$\hat{x}_1^+ = 0.4999 + 6.99966 \times 10^{-5}(0.35 - 0.34493)$ $= 0.4999004$
$\Sigma_{\tilde{x},k}^+ = (I - L_k C_k)\Sigma_{\tilde{x},k}^-$	$\Sigma_{\tilde{x},1}^+ = (1 - 6.99966 \times 10^{-5} \times 0.7) \times 10^{-5}$ $= 9.9995 \times 10^{-6}$

- Output: $\hat{z} = 0.4999 \pm 3\sqrt{9.9995 \times 10^{-6}} = 0.4999 \pm 0.0094866$

Iteration 2

- For the second iteration of the linear KF, let $i_1 = 0.5$, $i_2 = 0.25$, and $v_2 = 3.84$

$\hat{x}_k^- = A_{k-1}\hat{x}_{k-1}^+ + B_{k-1}u_{k-1}$	$\hat{x}_1^- = 0.4999004 - 10^{-4} \times 0.5 = 0.49985$
$\Sigma_{\tilde{x},k}^- = A_{k-1}\Sigma_{\tilde{x},k-1}^+ A_{k-1}^T + \Sigma_{\tilde{w}}$	$\Sigma_{\tilde{x},2}^- = 9.9995 \times 10^{-6} + 10^{-5} = 1.99995 \times 10^{-5}$
$\hat{y}_k = C_k\hat{x}_k^- + D_k u_k$	$\hat{y}_2 = 0.7 \times 0.49985 - 0.01 \times 0.25 = 0.347395$
$L_k = \Sigma_{\tilde{x},k}^- C_k^T [C_k \Sigma_{\tilde{x},k}^- C_k^T + \Sigma_{\tilde{v}}]^{-1}$	$L_2 = 1.99995 \times 10^{-5} \times 0.7[1.99995 \times 10^{-5} \times 0.7^2 + 0.1]^{-1}$ $= 0.00013998$
$\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$ (where $y_k = 3.84 - 3.5$)	$\hat{x}_2^+ = 0.49985 + 0.00013998(0.34 - 0.347395)$ $= 0.499849$
$\Sigma_{\tilde{x},k}^+ = (I - L_k C_k)\Sigma_{\tilde{x},k}^-$	$\Sigma_{\tilde{x},2}^+ = (1 - 0.00013998 \times 0.7) \times 1.99995 \times 10^{-5}$ $= 1.99976 \times 10^{-5}$

- Output: $\hat{z} = 0.4998 \pm 3\sqrt{1.99976 \times 10^{-5}} = 0.4998 \pm 0.013416$

Covariance

- Note that covariance (uncertainty) converges over time

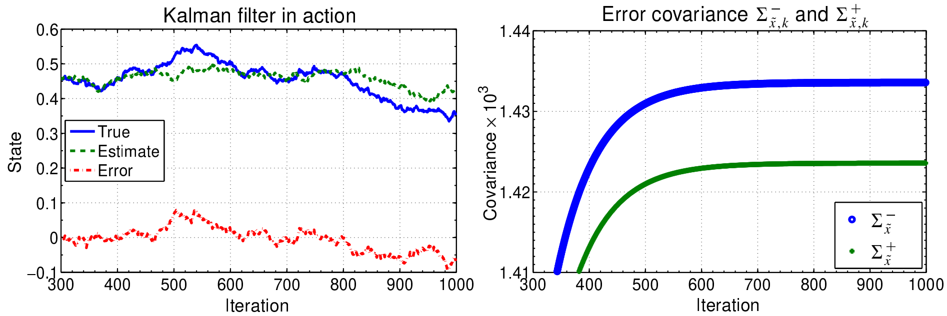
$\Sigma_{\tilde{x},1}^- = 1 \times 10^{-5}$ $\quad \Sigma_{\tilde{x},2}^- = 1.99995 \times 10^{-5}$ $\quad \Sigma_{\tilde{x},3}^- = 2.99976 \times 10^{-5}$

$\Sigma_{\tilde{x},0}^+ = 0$ $\quad \Sigma_{\tilde{x},1}^+ = 9.99951 \times 10^{-6}$ $\quad \Sigma_{\tilde{x},2}^+ = 1.99976 \times 10^{-5}$ $\quad \Sigma_{\tilde{x},3}^+ = 2.99931 \times 10^{-5}$

- Covariance increases during propagation, is then reduced by each measurement
- Covariance still oscillates in steady state between $\Sigma_{\tilde{x},ss}^-$ and $\Sigma_{\tilde{x},ss}^+$
- Estimation error bounds are $\pm 3\sqrt{\Sigma_{\tilde{x},k}^+}$ for 99$^+$ % assurance of estimate's accuracy

Results

- The plots below show a sample of the Kalman filter operating
- We'll soon look at how to write code to evaluate this example



- Note that Kalman filter does not perform especially well since $\Sigma_{\tilde{v}}$ is quite large
- However, these are best-possible results, since KF is the optimum MMSE estimator

Summary

- KF implements an MSEE-optimal state estimator for linear systems if assumptions regarding system noises are met
- KF equations naturally form a recursive algorithm for estimating the state
- While the KF works only for linear systems, we can linearize system dynamics for an approximate result (will learn better ways later)
- Example with a simplified battery model demonstrates the kinds of results you can expect from a KF
  - The KF will provide a state estimate...
  - And uncertainty bounds (error bounds) on that estimate
- We will find this very helpful for SOC estimation