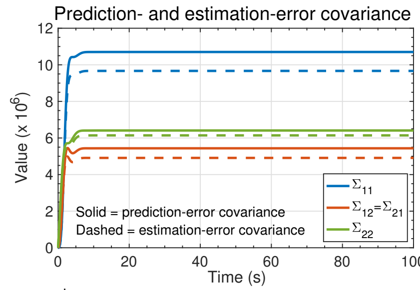
Convergence of the covariance matrices

- We sometimes observe an interesting phenomenon when we plot the prediction- and estimation-error covariance-matrix entries versus time.
- Often, they exhibit an initial transient and then converge to steady-state solutions.
- The figure to the right shows the entries of $\Sigma^-_{\tilde{x},k}$ and $\Sigma^+_{\tilde{x},k}$ for the spring-mass-damper system we have been using in our examples.
- After about 10 s (or 100 iterations), there is no longer any clear variation in these parameters.



- Since the primary purpose for computing $\Sigma^-_{\tilde{x},k}$ and $\Sigma^+_{\tilde{x},k}$ is to be able to compute L_k, this implies that if the covariances converge to a steady-state solution, L_k must also converge to a steady-state solution.

Idea

- Idea: Can we replace the time-varying L_k in the Kalman filter with a constant L_{ss} vector?
  - If yes, then we can omit all calculations of $\Sigma^-_{\tilde{x},k}$ and $\Sigma^+_{\tilde{x},k}$, which dramatically reduces the computational demand of the Kalman filter.
- Since the optimal solution uses L_k, a solution that uses L_{ss} instead will (by definition) be sub-optimal.
  - But, perhaps we are willing to accept a small reduction in estimate quality if the reduction in computation is significant.
- Conditions that guarantee the existence of a steady-state solution:
  - In addition to prior assumptions, A, B, C, and D are constant.
  - $\Sigma_{\tilde{w}} \geq 0$, $\Sigma_{\tilde{v}} > 0$, $\{C, A\}$ is "detectable" (or observable) and $\{A, \Sigma_{\tilde{w}}\}$ is "stabilizable" (or controllable).

Deriving the steady-state solution

- To derive a steady-state solution, we set:

$$\begin{aligned}\Sigma^-_{\tilde{x},k+1} &= \Sigma^-_{\tilde{x},k} = \Sigma^-_{\tilde{x},ss}\\ \Sigma^+_{\tilde{x},k+1} &= \Sigma^+_{\tilde{x},k} = \Sigma^+_{\tilde{x},ss},\end{aligned}$$

  in the Kalman-filter steps, and solve.
- Consider filter loop as $k \to \infty$:

$$\begin{aligned}\Sigma^-_{\tilde{x},ss} &= A\Sigma^+_{\tilde{x},ss}A^T + \Sigma_{\tilde{w}}\\ \Sigma^+_{\tilde{x},ss} &= \Sigma^-_{\tilde{x},ss} - \underbrace{\Sigma^-_{\tilde{x},ss}C^T\left[C\Sigma^-_{\tilde{x},ss}C^T + \Sigma_{\tilde{v}}\right]^{-1}}_{L_{ss}} C\Sigma^-_{\tilde{x},ss}.\end{aligned}$$

- Combine these two to get:

$$\Sigma^-_{\tilde{x},ss} = \Sigma_{\tilde{w}} + A\Sigma^-_{\tilde{x},ss}A^T - A\Sigma^-_{\tilde{x},ss}C^T\left[C\Sigma^-_{\tilde{x},ss}C^T + \Sigma_{\tilde{v}}\right]^{-1} C\Sigma^-_{\tilde{x},ss}A^T.$$

Solving this equation

- This equation is known as a “discrete algebraic Riccati equation” (DARE).
- For scalar systems (where the number of states $n = 1$), we can solve it using standard algebra, as we will do in an example in a few minutes.
- For systems having $n > 1$, we still have one matrix equation and one matrix unknown, so a solution is attainable.[1]
- Octave implements a solution to the DARE in `dlqe.m`, which stands for “discrete linear quadratic estimator” (Kalman filters are optimal linear quadratic estimators).
  - Once we know $\Sigma^-_{\tilde{x},ss}$, we can compute $L_{ss} = \Sigma^-_{\tilde{x},ss} C^T \left[C \Sigma^-_{\tilde{x},ss} C^T + \Sigma_{\tilde{v}} \right]^{-1}$.
  - Error bounds can be computed via $\Sigma^+_{\tilde{x},ss} = \Sigma^-_{\tilde{x},ss} - L_{ss} C \Sigma^-_{\tilde{x},ss}$.

[1] A common algorithm used to solve a DARE is discussed in the paper: A. Laub, “A Schur method for solving algebraic Riccati equations.” *IEEE Transactions on Automatic Control* , 24(6), 1979, 913–921.

Scalar-system example

- To visualize, suppose: $x_{k+1} = x_k + w_k$; $z_k = x_k + v_k$.
- Let $\mathbb{E}[w_k] = \mathbb{E}[v_k] = 0$, $\Sigma_{\tilde{w}} = 1$, $\Sigma_{\tilde{v}} = 2$.
- Notice $A = 1$, $C = 1$, so:

$$\begin{aligned}\Sigma^-_{\tilde{x},ss} &= \Sigma_{\tilde{w}} + A\Sigma^-_{\tilde{x},ss}A - \frac{\left(A\Sigma^-_{\tilde{x},ss}C\right)\left(C\Sigma^-_{\tilde{x},ss}A\right)}{C\Sigma^-_{\tilde{x},ss}C + \Sigma_{\tilde{v}}} \\ &= 1 + \Sigma^-_{\tilde{x},ss} - \frac{\left(\Sigma^-_{\tilde{x},ss}\right)^2}{\Sigma^-_{\tilde{x},ss} + 2}\end{aligned}$$

$$\Sigma^-_{\tilde{x},ss} + 2 = \left(\Sigma^-_{\tilde{x},ss}\right)^2,$$

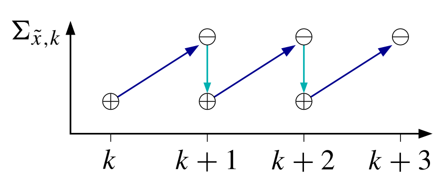
  which leads to the solutions $\Sigma^-_{\tilde{x},ss} = -1$ or $\Sigma^-_{\tilde{x},ss} = 2$.
- We must chose the positive-definite solution, so $\Sigma^-_{\tilde{x},ss} = 2$ and $\Sigma^+_{\tilde{x},ss} = 1$.

Verifying the example’s solution

- If we iterate the covariance equations from some initial condition, we examine the evolution of $\Sigma^-_{\tilde{x},k}$ and $\Sigma^+_{\tilde{x},k}$ to see whether they agree with this solution:

$$\begin{array}{ccccccccc} \Sigma^-_{\tilde{x},0} = \infty & & \Sigma^-_{\tilde{x},1} = 3 & & \Sigma^-_{\tilde{x},2} = 11/5 & & \Sigma^-_{\tilde{x},3} = 43/21 & \\ \downarrow & \nearrow & \downarrow & \nearrow & \downarrow & \nearrow & \downarrow & \nearrow \\ \Sigma^+_{\tilde{x},0} = 2 & & \Sigma^+_{\tilde{x},1} = 6/5 & & \Sigma^+_{\tilde{x},2} = 22/21 & & \Sigma^+_{\tilde{x},3} = 86/85 & \end{array}$$

- Since uncertainty always increases during prediction and decreases during the measurement update, there will be two distinct steady-state covariances, as we have seen.
- The iteration does indeed converge toward $\Sigma^-_{\tilde{x},ss} = 2$ and $\Sigma^+_{\tilde{x},ss} = 1$.



Implementing the steady-state filter in Octave

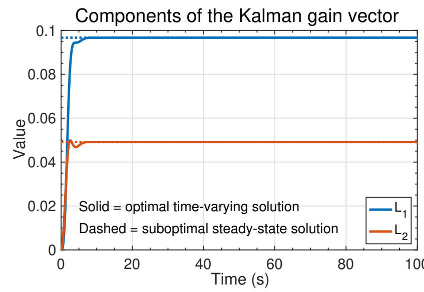
- It is simple to compute the steady-state covariances and Kalman gain in Octave:

```
[Lss,SigmaMinusss,SigmaPlusss] = dlqe(Ad,eye(nx),Cd,SigmaW,SigmaV);
```

- When we implement a steady-state Kalman filter, we can combine steps such that:

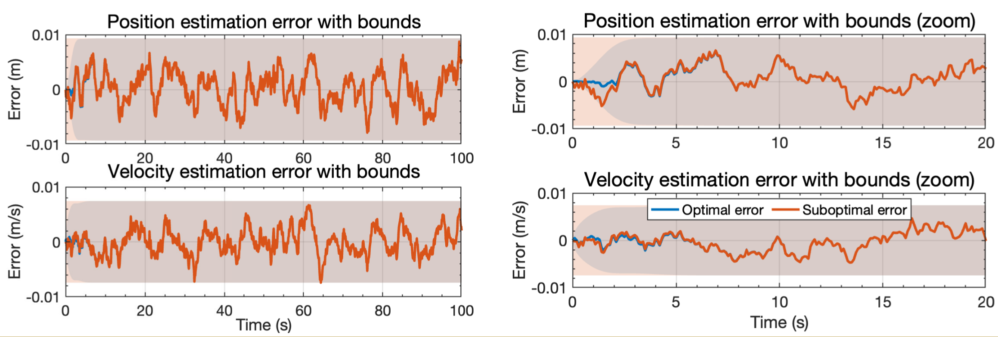
$$\hat{x}_k^+ = A\hat{x}_{k-1}^+ + Bu_{k-1} + L_{ss}(z_k - C(A\hat{x}_{k-1}^+ + Bu_{k-1}) - Du_k).$$

- We can use the precomputed steady-state $\Sigma_{\tilde{x},ss}^+$ to output confidence bounds, so no online computation is required.
- The figure compares optimal L_k versus L_{ss} for the spring-mass-damper example.
- We see very little difference after about 10 s.



Results from steady-state filtering

- The figures compare estimation error and bounds between the optimal and suboptimal steady-state Kalman filters.
- The zoom plots on the right help us see that the differences are most evident in the first seconds of operation; after that, the filters perform essentially identically.



Summary

- Under some common conditions, the prediction-error and estimation-error covariances of a Kalman filter converge to steady-state solutions.
- In these cases, the Kalman gain also converges to a steady-state solution.
- If we replace the time-varying L_k in the Kalman filter with the steady-state L_{ss}, we no longer need to compute $\Sigma_{\tilde{x},k}^-$ or $\Sigma_{\tilde{x},k}^+$.
- This can save a lot of computation, but does result in a suboptimal solution.
- An example demonstrated that a steady-state solution differs most from the optimal solution only at the beginning of the filter's execution until the transient in L_k decays.
- If this level of error is acceptable, a steady-state Kalman filter can be a good option.