

The Kalman Filter algorithm (KF)

[KF](#) is highly popular for filtering problems (e.g., tracking, navigation and control). It provides optimal predictions under the following assumptions:

- Known linear models for motion (F) & observation (H)
- I.i.d Gaussian noise with known covariance matrix in motion (Q) & observation (R)
- Known initial-state distribution (X_0)

KF parameters tuning

Given training data of states $\{x_t\}_t$ & observations $\{z_t\}_t$, the noise parameters R, Q can be estimated by $\hat{R} := Cov(\{z_t - Hx_t\}_t)$, $\hat{Q} := Cov(\{x_{t+1} - Fx_t\}_t)$. These \hat{R}, \hat{Q} are optimal up to the estimation error.

So what is wrong?

The KF assumptions practically rarely hold – even in very simplistic scenarios. For example, in the standard problem of Doppler radar tracking:

- Motion model (F) is not linear (nor known)
- Observation model (H) is not linear
- Observation noise (R) is i.i.d in polar coordinates – but not in Cartesian ones
- Initial-state distribution is unknown

Once the KF assumptions are violated, determining R, Q by estimation of noise is no longer equivalent to optimizing the predictions, i.e., the wrong problem is addressed. **This observation re-opens a problem considered solved for decades.**

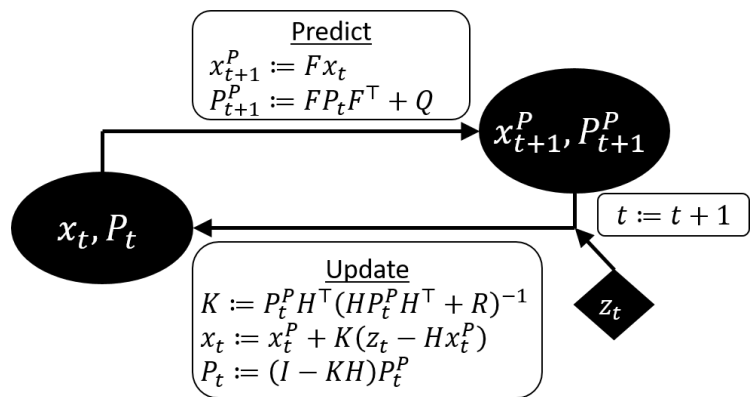
How can we solve it?

Given states ground-truth $\{x_t\}_t$ in the training data, the KF can be run on the data and optimized by standard gradient-based methods (e.g., [Adam](#)) wrt the prediction errors.

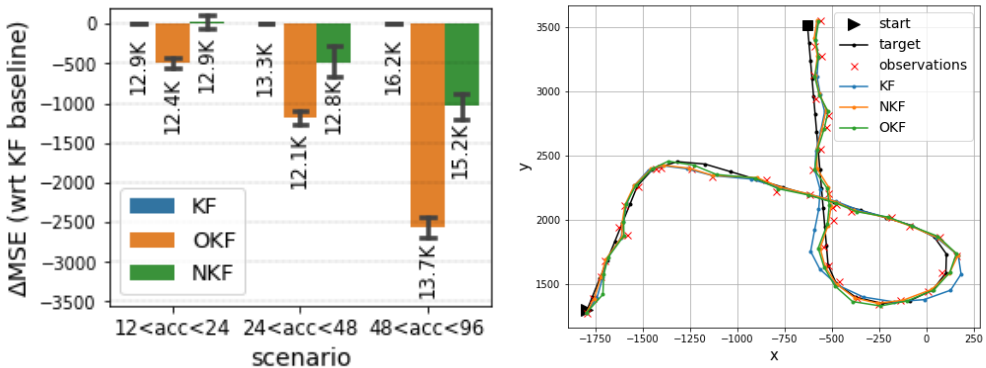
The optimized parameters (R, Q) represent covariance matrices, and thus should remain symmetric & positive-definite. Standard methods for such constrained optimization (e.g., projected GD and matrix-exponent parameterization) require SVD-decomposition and hence are computationally heavy. Thus, we use the [Cholesky-parameterization](#), which only costs a single matrix multiplication.

Does it really matter?

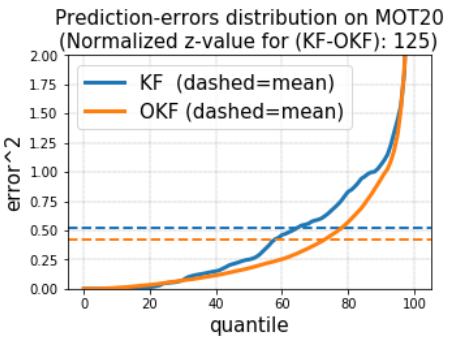
- In experiments, optimization reduced prediction errors under any subset of assumptions violations.
- Even when the only violation was non-linear H – optimization reduced the errors by 15%-45%.
 - For this scenario, we also show analytically how the violation modifies the effective noise.
- Optimization also compensates for “wrong” configuration (e.g., KF or EKF, Cartesian or polar representation, etc.).
- Without optimization of KF, learning models (e.g. LSTM) can wrongly seem to improve the prediction – leading to adoption of over-complicated algorithms.



The KF algorithm



Relative tracking errors of an Optimized KF (OKF) and a KF with LSTM predictor (NKF) – compared to a standard KF. The label of each bar corresponds to the absolute MSE. The right figure shows a sample target (projected onto XY plane). All models were learned over targets with acceleration range of 24-48, then tested on targets with acceleration ranges 12-24, 24-48 and 48-96. While the LSTM seems to beat the linear KF – its advantage is entirely eliminated once the KF is optimized.



Pedestrians tracking (MOT20 dataset): optimization reduces KF's errors by 18%.