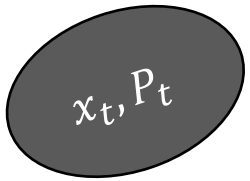
Optimization or Architecture:
How to Hack Kalman Filtering

Ido Greenberg Netanel Yannay Shie Mannor

NeurIPS 2023

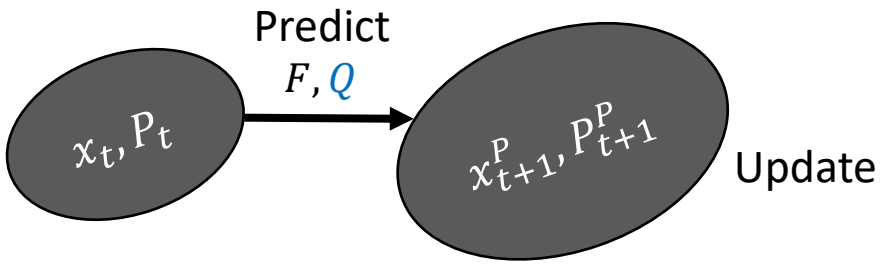
Background: Kalman Filter (KF)

- Goal:
  - Estimate x_{t+1} from measurements $\{z_\tau\}_{\tau=1}^{t}$



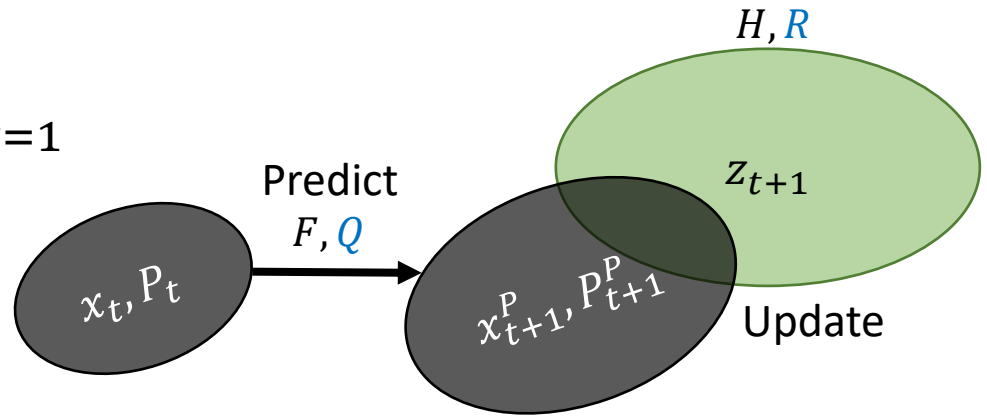
Background: Kalman Filter (KF)

- Goal:
  - Estimate x_{t+1} from measurements $\{z_\tau\}_{\tau=1}^{t}$



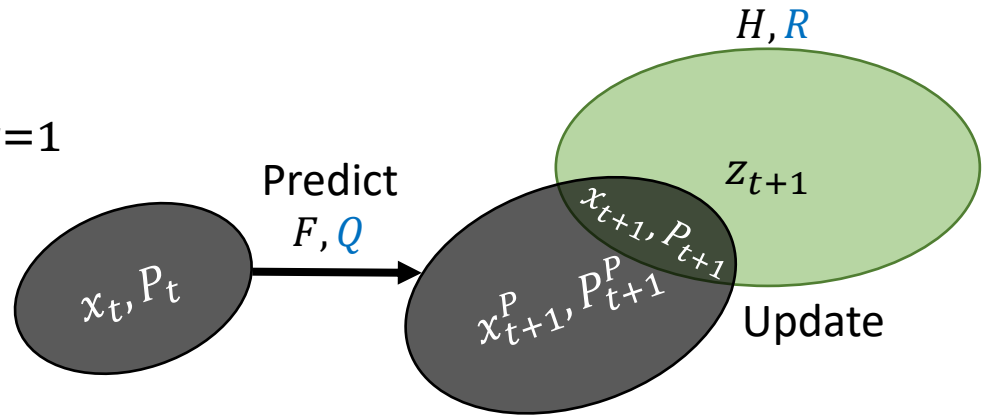
Background: Kalman Filter (KF)

- Goal:
  - Estimate x_{t+1} from measurements $\{z_\tau\}_{\tau=1}^{t}$



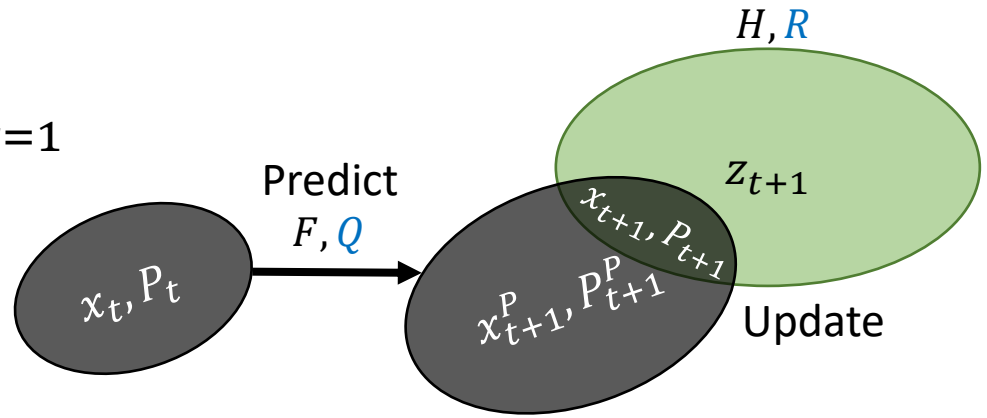
Background: Kalman Filter (KF)

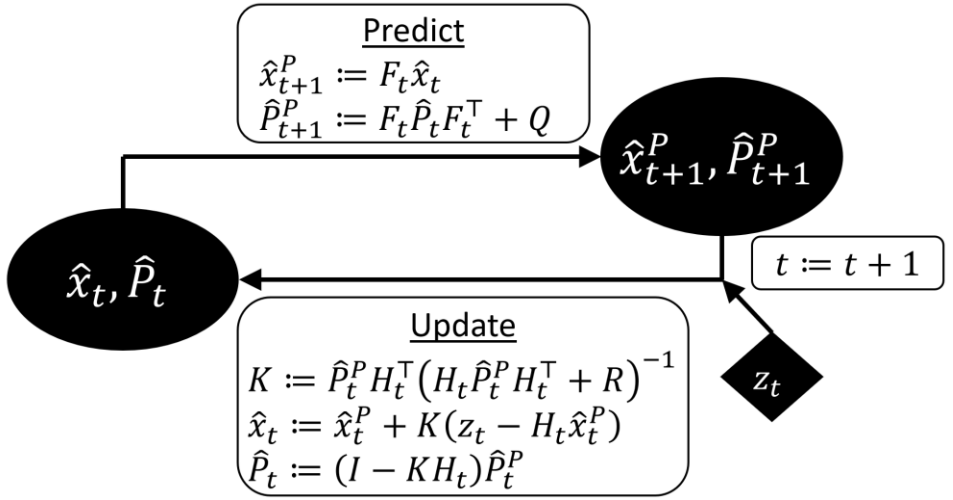
- Goal:
  - Estimate x_{t+1} from measurements $\{z_\tau\}_{\tau=1}^{t}$



Background: Kalman Filter (KF)

- Goal:
  - Estimate x_{t+1} from measurements $\{z_\tau\}_{\tau=1}^{t}$





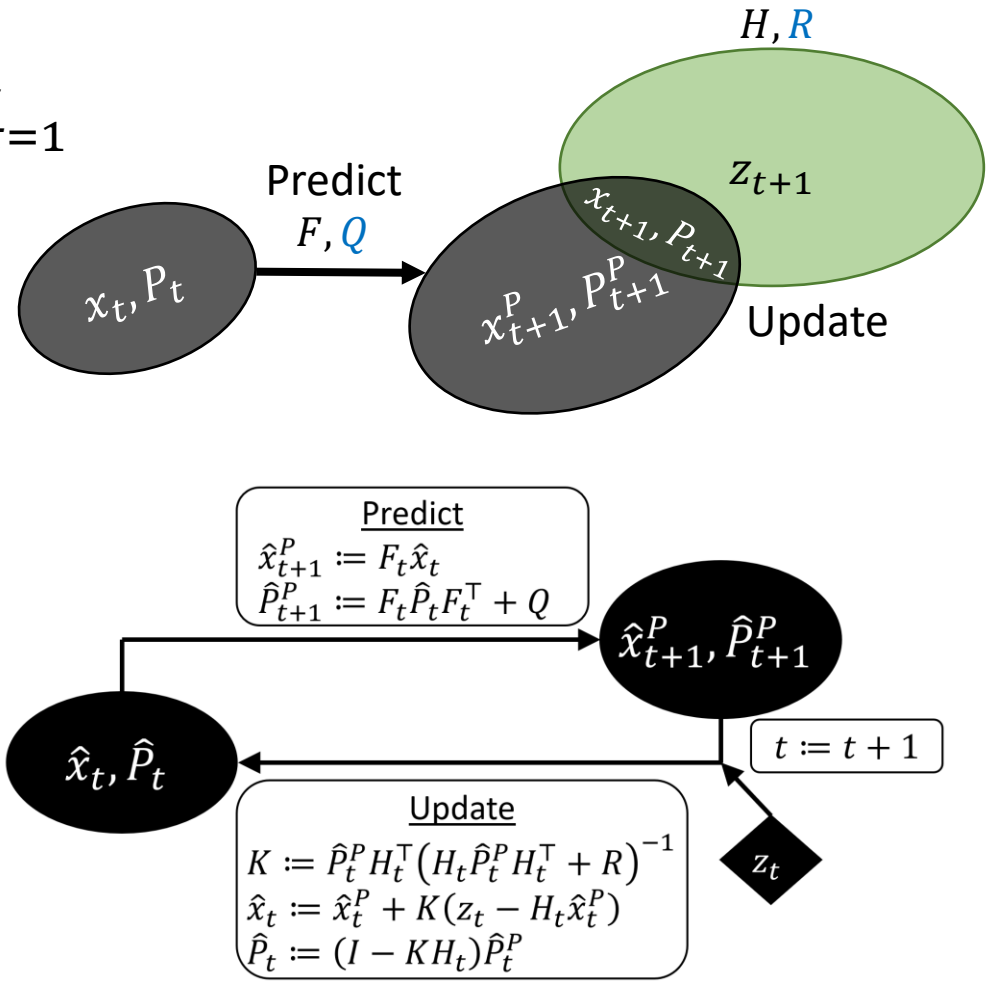
Background: Kalman Filter (KF)

- Goal:
  - Estimate x_{t+1} from measurements $\{z_\tau\}_{\tau=1}^{t}$

- How to know the noise Q, R?

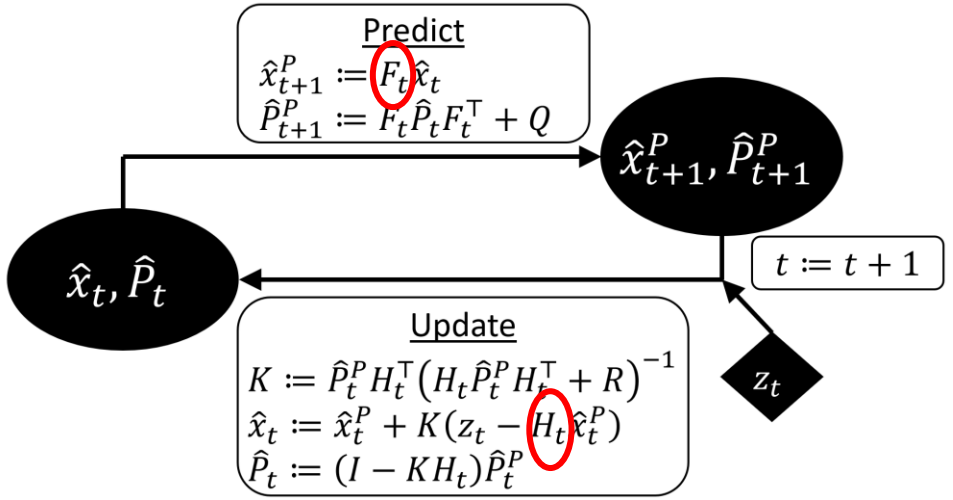
$$Q \coloneqq Cov(\{x_{t+1} - Fx_t\}_t)$$

$$R \coloneqq Cov(\{z_t - Hx_t\}_t)$$



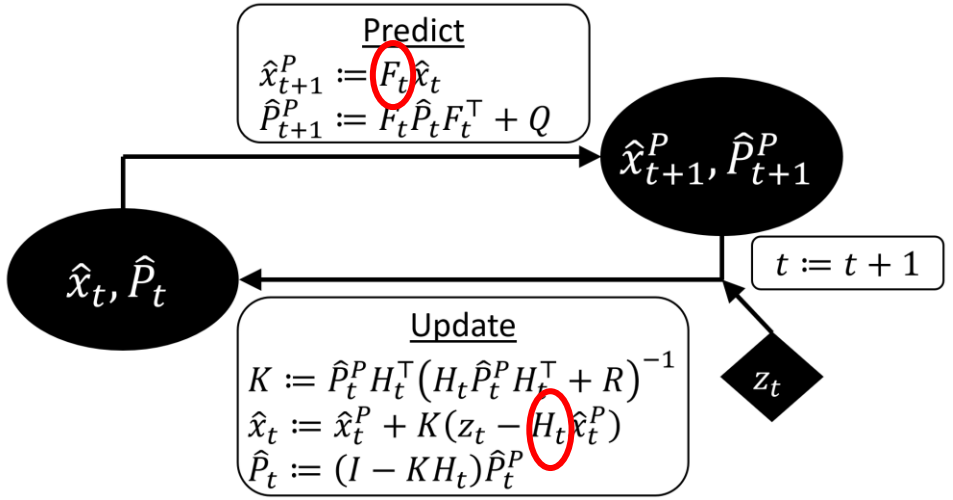
KF vs. Neural KF (NKF)

- Neural KF:
  - Replace F, H with LSTMs $F_{\theta_1}, H_{\theta_2}$



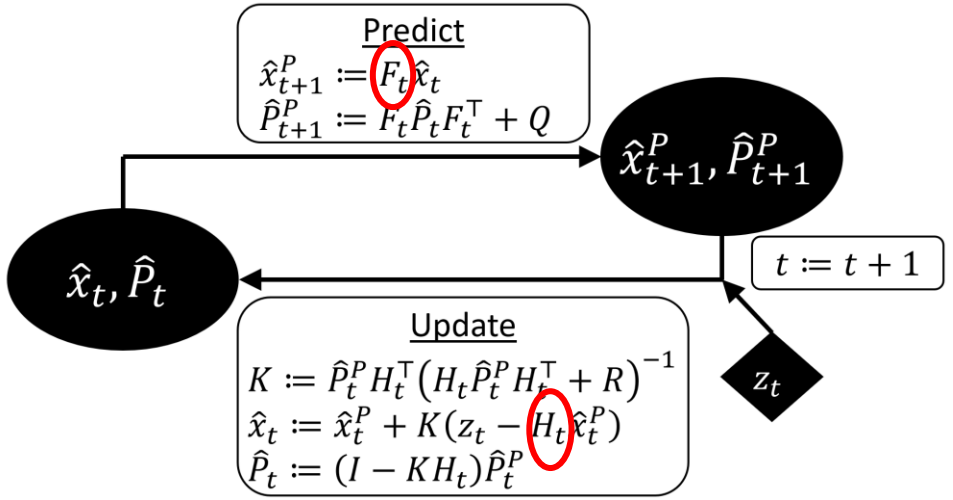
KF vs. Neural KF (NKF)

- Neural KF:
  - Replace F, H with LSTMs $F_{\theta_1}, H_{\theta_2}$
  - Optimize θ and Q, R



KF vs. Neural KF (NKF)

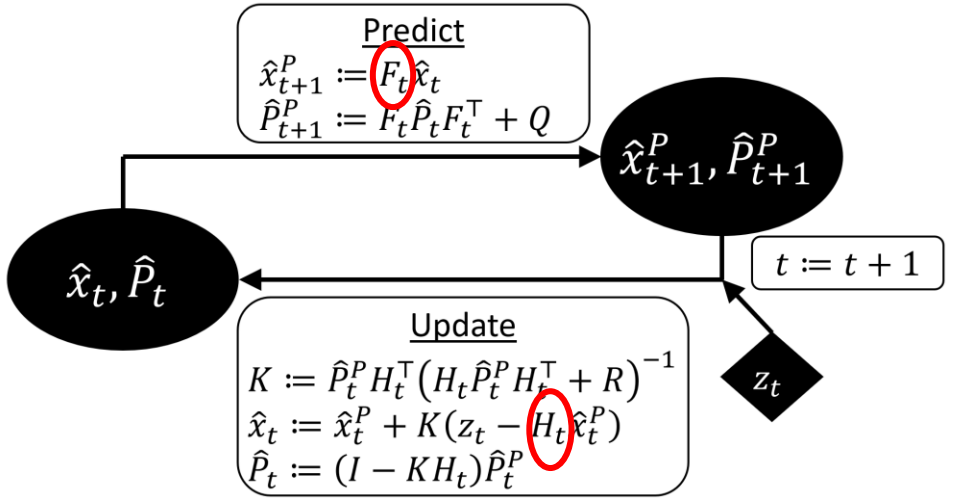
- Neural KF:
  - Replace F, H with LSTMs $F_{\theta_1}, H_{\theta_2}$
  - Optimize θ and Q, R
- NKF beats KF!



KF vs. Neural KF (NKF)

- Neural KF:
  - Replace F, H with LSTMs $F_{\theta_1}, H_{\theta_2}$
  - Optimize θ and Q, R
- NKF beats KF!

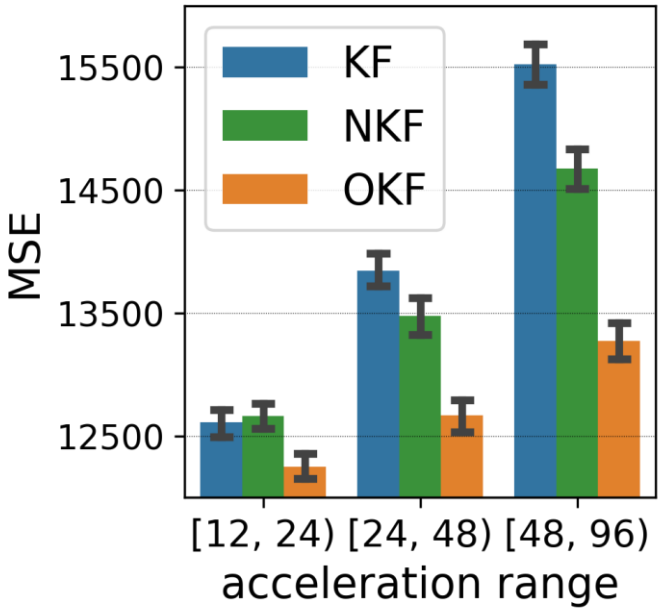
- Ablation test:
  - Standard F, H
  - Optimize Q, R

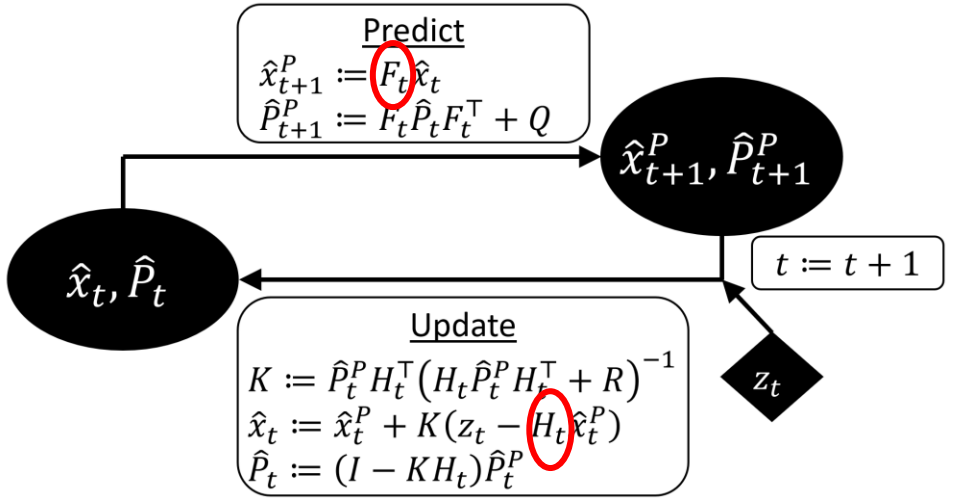


KF vs. Neural KF (NKF)

- Neural KF:
  - Replace F, H with LSTMs $F_{\theta_1}, H_{\theta_2}$
  - Optimize θ and Q, R
- NKF beats KF!

- Ablation test:
  - Standard F, H
  - Optimize Q, R
- Beats KF *and* NKF!





Optimized KF (OKF)

- Improve KF just by learning Q, R

Optimized KF (OKF)

- Improve KF just by learning Q, R

- How can it be?
  - $Q, R = Cov(\dots)$ is already MSE-optimal…

Optimized KF (OKF)

- Improve KF just by learning Q, R

- How can it be?
  - $Q, R = Cov(\dots)$ is already MSE-optimal…
    under unrealistic assumptions

Optimized KF (OKF)

- Improve KF just by learning Q, R

- How can it be?
  - $Q, R = Cov(\dots)$ is already MSE-optimal…
    under unrealistic assumptions

- Non-linear filtering:
  - Comparing **optimized NN** vs. **non-optimized KF** is common…
    and WRONG

Optimized KF (OKF)

- Improve KF just by learning Q, R

PyPI
`Optimized-Kalman-Filter`

- How can it be?
  - $Q, R = Cov(\dots)$ is already MSE-optimal…
    under unrealistic assumptions

- Non-linear filtering:
  - Comparing **optimized NN** vs. **non-optimized KF** is common…
    and WRONG