KALMAN FILTER IS ALL YOU NEED: OPTIMIZATION WORKS WHEN NOISE ESTIMATION FAILS

Anonymous authors
Paper under double-blind review

ABSTRACT

Determining the noise parameters of a Kalman Filter (KF) has been studied for decades. A huge body of research focuses on the task of noise estimation under various conditions, since precise noise estimation is considered equivalent to minimization of the filtering errors. However, we show that even a small violation of the KF assumptions can significantly modify the *effective* noise, breaking the equivalence between the tasks and making noise estimation an inferior strategy. We show that such violations are common, and are often not trivial to handle or even notice. Consequentially, we argue that a robust solution is needed – rather than choosing a dedicated model per problem. To that end, we apply gradient-based optimization to the filtering errors directly, with relation to an efficient parameterization of the symmetric and positive-definite parameters of the KF. In a variety of state-estimation and tracking problems, we show that the optimization improves both the accuracy of the KF and its robustness to design decisions. In addition, we demonstrate how an optimized neural network model can seem to reduce the errors significantly compared to a KF – and how this reduction vanishes once the KF is optimized similarly. This indicates how complicated models can be wrongly identified as superior to the KF, while in fact they were merely more optimized.

1 INTRODUCTION

The Kalman Filter (KF) (Kalman, 1960) is a celebrated method for linear filtering and prediction, with applications in many fields including tracking, guidance, navigation and control (Zarchan and Musoff, 2000; Kirubarajan, 2002). Due to its simplicity and robustness, it remains highly popular – with over 10,000 citations in the last 5 years alone (Google Scholar, 2021) – despite the rise of many non-linear sequential prediction models (e.g., recurrent neural networks). The KF relies on the following model for a dynamic system:

$$\begin{aligned} X_{t+1} &= FX_t + \omega_t \qquad (\omega_t \sim N(0, Q)) \\ Z_t &= HX_t + \nu_t \qquad (\nu_t \sim N(0, R)) \end{aligned} \tag{1}$$

where X_t is the state of the system at time t (whose estimation is usually the goal), and its dynamics are modeled by the linear operator F up to the random noise ω_t with covariance Q; and Z_t is the observation, which is modeled by the linear operator H up to the noise ν_t with covariance R.

To use the KF, one must determine the noise parameters Q, R. The filtering errors (i.e., estimation errors of the states $\{X_t\}$) are minimized when Q and R correspond to the true covariance matrices of the noise (Humpherys et al., 2012). Thus, these parameters are usually determined by noise estimation. In absence of system state data $\{x_t\}$ (the "ground truth"), many methods have been suggested to determine Q and R from observations data $\{z_t\}$ alone (Abbeel et al., 2005; Odelson et al., 2006; Zanni et al., 2017; Park et al., 2019). When ground-truth data *is* available, however, noise estimation trivially reduces to calculation of the sample covariance matrices (Lacey, 1998):

$$\hat{R} \coloneqq Cov(\{z_t - Hx_t\}_t), \qquad \hat{Q} \coloneqq Cov(\{x_{t+1} - Fx_t\}_t). \tag{2}$$

Indeed, as stated by Odelson et al. (2006), "*the more systematic and preferable approach to determine the filter gain is to estimate the covariances from data*". Our work focuses on such problems with ground-truth available for learning (but not for inference after the learning, of course), which was motivated by a real-world Doppler radar estimation problem.

Noise estimation is often not optimal: The equivalence between noise estimation and errors minimization can be proved under the standard KF assumptions – that is, known and linear dynamics and observation models (F, H), with i.i.d and normally-distributed noises $(\{\omega_t\}, \{\nu_t\})$ (Humpherys et al., 2012). However, as put by Thomson (1994), "*experience with real-world data soon convinces one that stationarity and Gaussianity are fairy tales invented for the amusement of undergraduates*" – and linearity and independence can be safely added to this list. Therefore, under realistic assumptions, the covariance of the noise does not necessarily correspond to optimal filtering.

We introduce a case study in the context of radar tracking, where we demonstrate that even using the true covariance of the noise ("oracle" noise-estimation) is sub-optimal in a variety of scenarios – including very simple scenarios with relatively minor violation of the KF assumptions. In Appendices E and F, we also analyze this phenomenon analytically for two private cases (non-linearity in Doppler radar and non-i.i.d noise in lidar), where the violation of a single KF assumption is shown to modify the effective noise. By providing this extensive evidence for the sub-optimality of noise estimation in practical applications of the KF, **we re-open a problem that was considered solved for decades** (Kalman, 1960).

We also show that seemingly small changes in the properties of the scenario may lead to major changes in the desired design of the KF, e.g., whether to use a KF or an Extended KF (Sorenson, 1985). In certain cases, the design choices are easy to overlook (e.g., Cartesian vs. polar coordinates), and are not trivial to make even if noticed. As a result, it is impractical to manually choose or develop a variant of the KF for every problem. Rather, we should assume that our model is sub-optimal, and leverage data to deal with the sub-optimality as robustly as possible.

Optimization *is* optimal: We consider Q and R as model parameters that should be optimized with respect to the filtering errors (i.e., system-state estimation errors) – rather than estimating the noise. While both noise estimation and errors optimization rely on exploitation of data, only the latter explicitly addresses the actual goal of solving the filtering problem.

Gradient-based optimization methods are usually effective in the field of machine learning, but applying them naively to the entries of Q and R may violate the symmetry and positive-definiteness (SPD) constraints of the covariance matrices. Indeed, even works that come as far as optimizing Q and R (instead of estimating the noise) usually apply limited optimization methods, e.g., grid-search (Coskun et al., 2017) or diagonal restriction of the covariance matrices (Formentin and Bittanti, 2014; Li et al., 2019). To address this issue, we use a parameterization based on Cholesky decomposition (Horn and Johnson, 1985), which allows us to apply gradient-based optimization to SPD matrices. This method is computationally efficient compared to other general gradient-based methods for SPD optimization (Tsuda et al., 2005; Tibshirani, 2015).

We demonstrate that the optimization reduces the errors of the KF consistently: over different variants of the KF, over different violations of KF assumptions, over different domains (tracking from radar, video or lidar), over small and large training datasets, and even under distributional shifts between train and test datasets. Furthermore, we show that optimization improves the robustness to design decisions, by shrinking the performance gaps between different variants of the KF.

As explained above, we extensively justify the need to optimize the KF in many practical problems, and suggest a simple solution which is effective, robust, computationally efficient, and relies on standard tools in supervised machine learning. As a result, we believe that in the scope of filtering problems with available ground-truth, whenever the KF assumptions are not strictly-guaranteed, **the suggested optimization method should become the new standard procedure for the KF tuning**.

Unfair comparison: Many learning algorithms have been suggested to address non-linearity in filtering problems, e.g., based on Recurrent Neural Networks (RNN). Such works often use a linear tool such as the KF as a baseline for comparison – with tuning parameters being sometimes ignored (Gao et al., 2019), sometimes based on noise estimation (fa Dai et al., 2020), and sometimes optimized in a limited manner using trial-and-error (Jamil et al., 2020) or grid-search (Coskun et al., 2017). Our findings imply that such a methodology yields over-optimistic conclusions, since the baseline is not optimized to the same level as the learning model. This may result in adoption of over-complicated algorithms with no actual added value. Instead, any learning algorithm should be compared to a baseline that is optimized using a similar method (e.g., gradient-descent with respect to the errors).

Indeed, we consider an extension of the KF based on LSTM, which is the key component in many SOTA algorithms for non-linear sequential prediction in recent years (Neu et al., 2021). For radar tracking with non-linear motion, we demonstrate how the LSTM seems to provide a significant improvement over the KF. Then, we show that the whole improvement comes from optimization of parameters, and *not* from the expressive non-linear architecture. In particular, this result demonstrates the competitiveness of our suggested method versus SOTA sequential prediction models.

Recent works in the area of machine learning have already shown that advanced algorithms often obtain most of their improvement from implementation nuances (Engstrom et al., 2019; Andrychowicz et al., 2020; Henderson et al., 2017). Our work continues this line of thinking and raises awareness to this issue in the domain of filtering problems.

Contribution: We show that the KF is used sub-optimally in a variety of problems; demonstrate that advanced filtering algorithms are often tested unfairly; suggest a simple method to solve both issues, using gradient-based optimization along with Cholesky decomposition; and provide a detailed case study which analyzes the differences between noise estimation and parameters optimization empirically and (for certain private cases) analytically.

Limitations: The main assumption of this work is availability of the ground-truth system-states in the training data, which allows supervised learning. Ground-truth data is often available from simulations, controlled experiments or manual labeling. Within this scope, we show that although noise estimation is straight-forward (Equation 2), it is often not the right task to address.

The paper is organized as follows: Section 2 reviews the KF. Section 3 introduces our method for efficient KF optimization. Section 4 justifies the necessity of KF optimization through a detailed case study. Section 5 presents a neural version of the KF which reduces the errors compared to a standard KF – but not when compared to an optimized KF. Section 6 discusses related works.

2 Preliminaries: the Kalman Filter Algorithm

The KF algorithm (Kalman, 1960; Humpherys et al., 2012) relies on Equation 1 for a dynamic system model. It keeps an estimate of the state X_t, represented as the mean x_t and covariance P_t of a normal distribution. As shown in Figure 1, it alternately predicts the next state using the dynamics model (*prediction* step), and processes new information from incoming observations (*update* or *filtering* step).

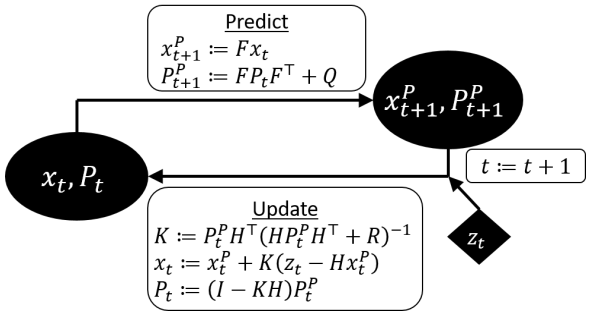


Figure 1: The KF algorithm. The prediction step is based on the motion model F with noise Q, whereas the update step is based on the observation model H with noise R.

The KF yields optimal state estimations – but only under a restrictive set of assumptions (Kalman, 1960), as specified in Definition 2.1. Note that normality of the noise is excluded since it is not necessary for optimality (Humpherys et al., 2012), although it is also often assumed.

Definition 2.1 (KF assumptions)**.** F, H of Equation 1 are constant and known matrices (i.e., linear models of motion and observation); ω_t, ν_t are i.i.d random variables with zero-mean and constant, known covariance matrices Q, R, respectively; and the initial state distribution is known.

Certain assumptions violating in Definition 2.1 can be partially handled by variations of the KF, such as the Extended KF (EKF) (Sorenson, 1985) which replaces the linear models F, H with local linear approximations, and the Unscented KF (UKF) (Wan and Van Der Merwe, 2000) which applies the filtering through sigma-points sampled from the estimated distribution. The use of multiple tracking models alternately is also possible using switching mechanisms (Mazor et al., 1998).

While F and H are usually determined based on domain knowledge, Q and R are often estimated from data as the covariance of the noise. As mentioned in Section 1, this can be done using Equation 2

if ground-truth data is available, or using more sophisticated methods otherwise (Odelson et al., 2006; Feng et al., 2014; Park et al., 2019).

See Appendix A for a detailed introduction of the KF and recurrent neural networks (RNN, LSTM).

3 KF Optimization Using Cholesky Parameterization

The performance of a KF strongly depends on its parameters Q and R (Formentin and Bittanti, 2014). These parameters are usually regarded as estimators of the noise covariance in motion and observation, respectively (Lacey, 1998), and are estimated accordingly. Although optimization has been suggested for the KF in the past (Abbeel et al., 2005), it was often viewed as a fallback solution (Odelson et al., 2006), for cases where direct estimation is not possible (e.g., the true states are unavailable in the data (Feng et al., 2014)). Accordingly, we define our baseline method for this work:

Method 1 (Estimated KF)**.** KF whose parameters Q, R were determined from data using Equation 2.

The preference of noise estimation relies on the fact that the KF – with parameters corresponding to the noise covariances – minimizes the square filtering errors (MSE) of the estimates of the system-states $\{X_t\}$. This holds under Assumptions 2.1 (Humpherys et al., 2012). Hence, equivalently, we could explicitly look for the parameters that minimize the MSE, e.g., using the Adam algorithm (Diederik P. Kingma, 2014). Adam is a popular variant of the well-known gradient-descent algorithm, and has achieved remarkable results in many optimization problems in the field of machine learning in recent years, including in non-convex problems where local-minima exist (Zhong et al., 2020).

Numeric optimization is more complicated than computing two covariance matrices, but is shown to be beneficial in Sections 4 and 5. Stable optimization algorithms are available in several open-source packages, e.g., PyTorch (Paszke et al., 2019), which supports gradient-propagation through matrix-inversion (as needed for the KF computations). As demonstrated in Sections 4,5, the PyTorch implementation of Adam provides stable results on MSE-optimization of a KF. This is conceivable, as the KF is arguably a simple model in terms of mathematical sophistication and number of parameters.

One major challenge in the KF parameters optimization is that both Q and R correspond to covariance matrices, which are symmetric and positive definite (SPD): a naive numeric optimization of their entries may ruin the SPD structure. This difficulty often motivates optimization methods that avoid gradients (Abbeel et al., 2005), or even the restriction of Q and R to be diagonal (Li et al., 2019). Indeed, Formentin and Bittanti (2014) pointed out that "*since both the covariance matrices must be constrained to be positive semi-definite, Q and R are often parameterized as diagonal matrices*".

To allow Adam to optimize the non-diagonal Q and R we use the Cholesky decomposition (Horn and Johnson, 1985), which states that any SPD matrix $A \in \mathbb{R}^{n\times n}$ can be written as $A = LL^\top$, where L is lower-triangular with positive entries along its diagonal. The reversed claim is also true: for any lower-triangular L with positive diagonal, $LL^\top$ is SPD. Now consider the parameterization of A using the following $\binom{n}{2} + n = \frac{n(n+1)}{2}$ parameters: $\binom{n}{2}$ parameters correspond to $\{L_{ij}\}_{1\le j<i\le n}$, and n parameters to $\{\log L_{ii}\}_{1\le i\le n}$. Clearly, the transformation from the parameters to the SPD matrix $A = LL^\top$ is differentiable and outputs a SPD matrix for any realization of the parameters in $\mathbb{R}^{n(n+1)/2}$. Thus, we can apply Adam (or any other gradient-based method) to optimize these parameters without worrying about the SPD constraints. This approach, which is presented more formally in Appendix C and implemented in our code, was used for all the optimizations in this work.

Cholesky decomposition for covariance matrix parameterization was already suggested by Pinheiro and Bates (1996). However, despite its simplicity, it is not commonly used for gradient-based optimization of SPD matrices in general, and to the best of our knowledge is not used at all for KF optimization in particular. Other methods exist for SPD optimization, e.g., matrix-exponent (Tsuda et al., 2005) and projected gradient-descent with respect to the SPD cone (Tibshirani, 2015). These methods require SVD-decomposition in every iteration, hence are computationally heavy, which may explain why they are not commonly used for KF tuning. The parameterization derived from Cholesky decomposition only requires a single matrix multiplication, and thus is both efficient and easy to implement. This results in our method for KF optimization:

Method 2 (Optimized KF)**.** A KF whose parameters Q, R were determined from data by optimizing the MSE of the state-estimates, using Adam algorithm with relation to a Cholesky parameterization of Q and R, as described in detail in Appendix B.1.

4 KALMAN FILTER CONFIGURATION AND OPTIMIZATION: A CASE STUDY

In this section, we introduce a detailed case study to compare noise estimation and errors optimization in the KF. As mentioned in Section 2, the two are equivalent only under Assumptions 2.1 (Humpherys et al., 2012). Some of these assumptions are clearly violated in realistic scenarios, while other violations may be less obvious. For example, even if a radar's noise is i.i.d in polar coordinates, it is not so in Cartesian coordinates (see Appendix A.4).

The need to rely on many assumptions might explain why there are several extensions and design decisions in a KF configuration. This includes the choice between KF and EKF; the choice between educated state initialization and a simple uniform prior (Linderoth et al., 2011); and certain choices that may be made without even noticing, such as the coordinates of the state representation.

The case study below justifies the following claims:

1. Design decisions in a KF are often nontrivial to make and are potentially significant.
2. Tuning a KF by noise estimation is often highly sub-optimal – even in very simple scenarios.
3. Tuning a KF by optimization improves both accuracy and robustness to design decisions.
4. KF optimization using Method 2 is robust to distributional shifts (Appendix G) and to small training datasets (Appendix H).

These claims imply that the popular KF algorithm may not be exploited to its full potential. Furthermore, many works that compare learning algorithms to a KF baseline conduct an "unfair" comparison, as the learning algorithms are optimized and the KF is not. This may lead to adoption of unnecessarily complicated algorithms, as demonstrated in Section 5. Indeed, in many works the tuning of the baseline KF is either ignored in the report (Gao et al., 2019) or relies on estimation (or knowledge) of the noise (fa Dai et al., 2020; Jamil et al., 2020), as demonstrated extensively in Section 6.

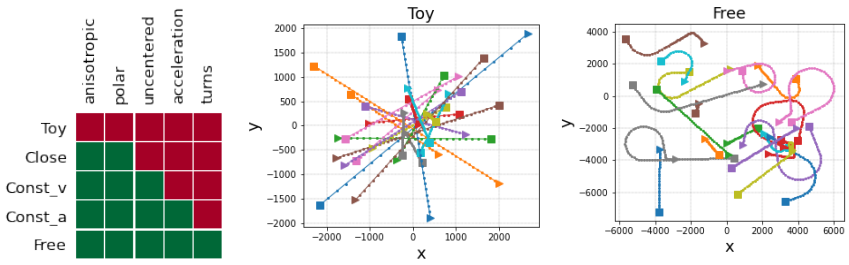
4.1 SETUP AND METHODOLOGY

In the case study of **radar tracking**, each target is represented by a sequence of (unknown) states in consecutive time-steps, and a corresponding sequence of radar measurements. A state $x_{full} = (x_x, x_y, x_z, x_{vx}, x_{vy}, x_{vz})^\top \in \mathbb{R}^6$ consists of 3D location and velocity. We also denote $x = (x_x, x_y, x_z)^\top \in \mathbb{R}^3$ for the location alone (or $x_{\text{target},t} \in \mathbb{R}^3$ for the location of a certain target at time t). An observation $z \in \mathbb{R}^4$ consists of noisy measurements of range, azimuth, elevation and Doppler signal. Note that the former three correspond to a noisy measurement of x in polar coordinates, and the latter one measures the projection of velocity onto the radial direction x. The goal is to minimize the error of the point-estimate of the state: $MSE = \sum_{\text{target}} \sum_t (\tilde{x}_{\text{target},t} - x_{\text{target},t})^2$.

The case study considers 5 types of tracking scenarios (*benchmarks*) and 4 variants of the KF (*baselines*). For each benchmark and each baseline, we use the benchmark training data to produce one estimated **KF** (Method 1) and one optimized KF (**OKF**, Method 2). We then evaluate the errors of both models on the test data of the benchmark (generated using different seeds than the training data). All experiments were run on eight i9-10900X CPU cores on a single Ubuntu machine.

Figures 2b,2c display a sample of trajectories in the simplest benchmark (*Toy*), which satisfies all KF assumptions except for a linear observation model H; and in the complex *Free Motion* benchmark, which violates several assumptions. The other benchmarks are demonstrated in Appendix L. Figure 2a defines each of the 5 benchmarks more formally as a certain subset of the following properties:

- *anisotropic*: horizontal motion is more likely than vertical motion (otherwise motion direction is distributed uniformly).
- *polar*: radar noise is generated i.i.d in polar coordinates (otherwise noise is Cartesian i.i.d, which violates the physics of the system).
- *uncentered*: targets are not forced to concentrate close to the radar.
- *acceleration*: speed change is allowed (through intervals of constant acceleration).
- *turns*: non-straight motion is allowed.



(a) (b) (c)

Figure 2: (a) Benchmarks names (rows) and the properties that define them (columns). Green means that the benchmark satisfies the property. (b,c) Targets in Toy and Free Motion benchmarks (projected onto XY plane).

The 4 baselines differ from each other by using either KF or EKF, with either Cartesian or polar coordinates for representation of R (the rest of the system is always represented in Cartesian coordinates). As mentioned above, each baseline is tuned once by Method 1 and once by Method 2. For the Toy benchmark, the optimal parameters are also derived **analytically** in Appendix E. Appendix G demonstrates the robustness of OKF to distributional shifts by training on one benchmark and testing on another. Appendix H repeats the tests for different sizes of training datasets.

We also repeat the experiment for the problem of **tracking from video**, using MOT20 dataset (Dendorfer et al., 2020) of real-world pedestrians, with train and test datasets taken from separated videos. For this problem we only consider a KF with Cartesian coordinates, since there is no polar component in the problem. See Appendix I for the detailed setup of the video tracking experiments. In addition, we test the OKF in the problem of **self-driving state-estimation from lidar measurements** (Appendix J). These two domains extend the scope of the experimented KF assumptions violations, since they correspond to linear observation models (unlike the radar benchmarks), with both noisy (lidar) and noiseless (MOT20) observations. For a simplified lidar problem, we also derive the optimal parameters **analytically** in Appendix F.

4.2 Results

Design decisions are not trivial: Table 1 summarizes the tracking errors. The left column in each cell corresponds to Method 1 (standard KF), and shows that in each benchmark, the errors strongly depend on the design decisions (R's coordinates and whether to use EKF). In the Toy benchmark, for example, EKF is the best design, since the observation model H is non-linear.

In other benchmarks, however, the winning designs of the non-optimized KF are arguably surprising:

1. Under non-isotropic motion direction (all benchmarks except Toy), EKF is worse than KF despite the non-linear motion. It is possible that since the horizontal prior reduces the stochasticity of H, the advantage of EKF no longer justifies the instability of the derivative-based approximation.

Table 1: Summary of the errors of the various models over the various benchmarks (on out-of-sample test data). Corresponding confidence intervals are available in Figure 14a in the appendix. In the model names, "O" denotes optimized, "E" denotes extended, and "p" denotes polar (e.g., OEKFp is an *extended* KF with *polar* representation of R and *optimized* parameters). For KFp, we also consider an oracle-realization of R according to the true noise of the simulated radar in polar coordinates (available only in polar benchmarks). Note that (1) for any benchmark and any baseline, optimization yields lower errors than estimation; and (2) this remains true even in the oracle variant, where the noise "estimation" is perfect.

Benchmark	KF	OKF	KFp	KFp (oracle)	OKFp	EKF	OEKF	EKFp	OEKFp
Toy	151.7	84.2	269.6	–	116.4	92.8	**79.4**	123.0	109.1
Close	25.0	24.8	22.6	22.5	**22.5**	26.4	26.1	24.5	24.1
Const_v	90.2	90.0	102.3	102.3	**89.2**	102.5	99.7	112.7	102.1
Const_a	107.5	101.6	118.4	118.3	**100.3**	110.0	107.0	126.0	108.7
Free	125.9	118.8	145.6	139.3	**117.9**	135.8	121.9	149.3	120.0

2. Even when the observation noise is polar i.i.d, polar representation of R is not beneficial unless the targets are forced to concentrate close to the radar (last 3 benchmarks in Table 1). It is possible that when the targets are distant, Cartesian coordinates have a more important role in expressing the horizontal prior of the motion.

Since the best variant of the KF for each benchmark seems difficult to predict in advance, a practical system cannot rely on choosing the KF variant optimally, and rather has to be robust to this choice.

Optimization is more accurate and robust: Table 1 shows that for *every* benchmark and *every* baseline in the **radar tracking** problem (20 experiments in total), optimization yielded smaller errors than noise estimation (over an out-of-sample test dataset). Note that OKF wins even in the Toy scenario, under the slightest violation of KF assumptions. In addition, the variance between the baselines reduces under optimization, i.e., the optimization makes the KF more robust to design decisions (which is also evident in Figure 14a in Appendix L).

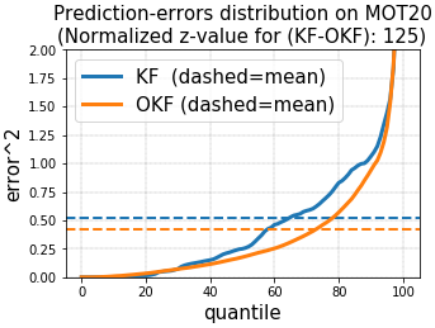
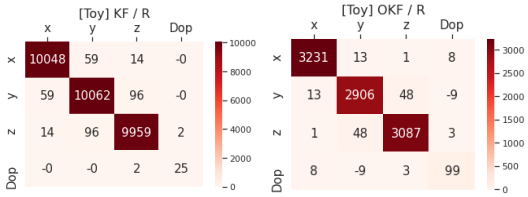


Figure 3: Prediction errors of KF and OKF on 1208 targets of the test data of MOT20 videos dataset. The MSE of OKF is smaller by 18%, with statistical significance of p-value $< 10^{-6}$ over the test samples.

We also studied the performance of a KF with a perfect knowledge of the noise covariance matrix R. Note that in the constant-speed benchmarks, the estimation of $Q = 0$ is already very accurate, hence in these benchmarks the oracle-KF has a practically perfect knowledge of both noise covariances. Nonetheless, Table 1 shows that the oracle yields very similar results to a KF with estimated parameters. This indicates that the limitation of noise estimation in the KF indeed comes from choosing the wrong goal, and not from estimation inaccuracy.

Figure 3 shows that OKF provides significantly better predictions also in **video tracking**, where the MSE is reduced by 18% (see Appendix I for more details). In addition, in **lidar-based state estimation**, Appendix J shows a similar error reduction of 15%, and Lemma F.3 explains the MSE-advantage of OKF **analytically**.



(a) Estimated R (b) Optimized R

Figure 4: The covariance matrix R of the observation noise obtained in a (Cartesian) KF by noise estimation and by optimization, based on the dataset of the Toy benchmark. The axes correspond to the observation variables associated with the matrix entries. Note that the noise estimation is quite accurate, as the true variance of the noise is 100^2 for the positional dimensions and 5^2 for Doppler. The optimization increases the variance associated with the Doppler signal, as predicted by Lemma E.1. The decrease in the other diagonal components is discussed in Appendix E.

OKF is further demonstrated to be robust to major **distributional shifts** in Appendix G, and to **small training datasets** in Appendix H.

Diagnosis of the KF sub-optimality in Toy scenario: The source of the gap between estimated and optimized noise parameters can be studied through the simplest Toy benchmark, where the only violation of KF assumptions is the non-linear observation model H. Since the non-constant entries of H correspond to the Doppler observation, the non-linearity inserts uncertainty to the Doppler observation (in addition to the inherent observation noise). This increases Doppler's effective noise in comparison to the location observation, as shown **analytically** by Lemma E.1 in the appendix. This explanation is consistent with Figure 4: the noise associated with Doppler is indeed increased by the optimization. Note that the non-linearity modifies the effective noise in a delicate way, which would *not* be compensated by a naive trial and error of noise inflation or deflation.

5 Neural Kalman Filter: Is Non-linear Prediction Helpful?

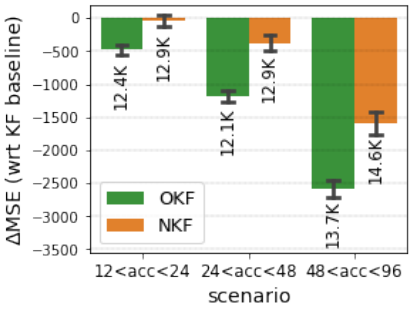
A standard Kalman Filter for a tracking task assumes linear motion, as discussed in Section 2. In this section we introduce the ***Neural Kalman Filter*** tracking model (NKF), which uses LSTM networks to model non-linear motion, while keeping the framework of the KF – namely, the probabilistic representation of the target's state, and the separation between prediction and update steps.

Every prediction step, NKF uses a neural network model to predict the target's acceleration a_t and the motion uncertainty Q_t. Every update step, another network predicts the observation uncertainty R_t. The predicted quantities are incorporated into the standard update equations of the KF, as shown in detail in Figure 7 in Appendix B. For example, the prediction step of NKF is:

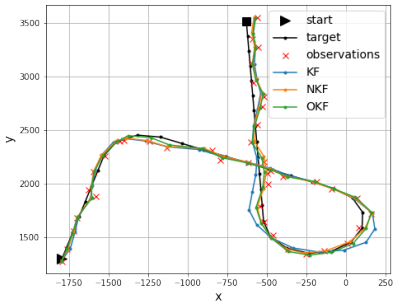
$$x^P_{t+1} = Fx_t + 0.5a_t(\Delta t)^2, \qquad P^P_{t+1} = FP_tF^\top + Q_t$$

where $F = F(\Delta t)$ is the constant-velocity motion operator, and a_t, Q_t are predicted by the LSTM (whose input includes the recent observation and the current estimated target's state). Other predictive features were also attempted but failed to provide significant value, as described in Appendix B. Note that the neural network predicts the acceleration rather than directly predicting the location. This is intended to regularize the predictive model and to exploit the domain knowledge of kinematics.

After training NKF over a dataset of simulated targets with random turns and accelerations (see Appendix B.1 for more details), we tested it on a test dataset. The test dataset is similar to the training dataset, with different seeds and an extended range of permitted accelerations. As shown in Figure 5, NKF significantly reduces the tracking errors compared to a standard KF.



(a)



(b)

Figure 5: (a) Relative tracking errors (lower is better) with relation to a standard KF, over targets with different ranges of acceleration. The error-bars represent confidence intervals of 95%. The label of each bar represents the corresponding *absolute* MSE ($\times 10^3$). In the training data, the acceleration was limited to 24-48, hence the other ranges measure generalization. While the Neural KF (NKF) is significantly better than the standard KF, its advantage is entirely eliminated once we optimize the KF (OKF). (b) A sample target and the corresponding models outputs (projected onto XY plane). The standard KF has a difficulty to handle some of the turns.

At this point, it seems that the non-linear architecture of NKF provides better accuracy in the non-linear problem of radar tracking. However, Figure 5 shows that by shifting the baseline from a naive KF to an optimized one, we *completely eliminate* the advantage of NKF, and in fact reduce the errors even further. In other words, in this experiment the benefits of NKF come *only* from optimization and *not at all* from the expressive architecture. By overlooking the sub-optimality of noise estimation in the KF, we would wrongly adopt the over-complicated NKF.

Note that the optimized KF (OKF) also generalizes well to targets with different accelerations than observed in the training, which indicates a certain robustness to distributional shifts. Appendix K extends the experiment to additional variants of NKF, to another tracking benchmark, and to the evaluation metric of likelihood (NLL) in addition to estimation error (MSE). Note that high likelihood score is important for the matching task in a multi-target tracking problem.

Of course, our results do not imply that neural-networks in general cannot be superior to a KF: only that when comparing the two, if the KF is not optimized similarly to the neural model, the experimental results may be very misleading. As discussed in Section 6, this wrong *methodology* is not uncommon in the literature.

6 Related Work

Noise estimation: When tuning a KF, the system-states are often unavailable in the data (Formentin and Bittanti, 2014). Thus, estimation of the KF noise parameters from observations alone has been studied for decades, addressed using various methods such as EM (Shumway and Stoffer, 2005) and others (Odelson et al., 2006; Feng et al., 2014; Park et al., 2019). However, if the states *are* available, noise estimation reduces to Equation 2 and is considered a solved problem (Odelson et al., 2006). In this case, we show that although noise estimation is easy, it is often not the right task to address.

Many works addressed the problem of non-stationary noise estimation (Zanni et al., 2017; Akhlaghi et al., 2017). However, as demonstrated in Sections 4,5, in certain cases stationary methods are highly competitive if tuned correctly – even in problems with complicated dynamics.

Optimization: In this work we apply gradient-based optimization to the KF with respect to its errors. Optimization without gradients computation was already suggested in Abbeel et al. (2005). In practice, "optimization" of the KF is often handled manually using trial-and-error (Jamil et al., 2020) or a grid-search over possible values of Q and R (Formentin and Bittanti, 2014; Coskun et al., 2017). In other cases, Q and R are restricted to be diagonal (Li et al., 2019; Formentin and Bittanti, 2014).

Gradient-based optimization of SPD matrices in general was suggested in Tsuda et al. (2005) using matrix-exponents, and is also possible using projected gradient-descent (Tibshirani, 2015) – both rely on SVD-decomposition. In this work, we apply gradient-based optimization using the parameterization that was suggested in Pinheiro and Bates (1996), which requires a mere matrix multiplication, and thus is both efficient and easy to implement.

Neural Networks (NNs) in filtering problems: Section 5 presents a RNN-based extension of the KF, and demonstrates how its advantage over the linear KF vanishes once the KF is optimized. The use of NNs for non-linear filtering problems is very common in the literature, e.g., in online tracking prediction (Gao et al., 2019; Dan Iter, 2016; Coskun et al., 2017; fa Dai et al., 2020; Ullah et al., 2019), near-online prediction (Kim et al., 2018), and offline prediction (Liu et al., 2019b). In addition, while Bewley et al. (2016) apply a KF for video tracking from mere object detections, Wojke et al. (2017) add to the same system a NN that generates visual features as well. NNs were also considered for related problems such as data association (Liu et al., 2019a), model-switching (Deng et al., 2020), and sensors fusion (Sengupta et al., 2019).

In many works that consider NNs for filtering problems, a KF is used as a baseline for comparison. However, while the NN parameters are typically optimized with respect to the filtering errors, the KF parameters tuning is sometimes ignored (Gao et al., 2019; Bai et al., 2020; Zheng et al., 2019), sometimes based on estimation (or knowledge) of the noise (fa Dai et al., 2020; Aydogmus and Aydogmus, 2015; Revach et al., 2021), and sometimes optimized in a limited manner as mentioned above (Jamil et al., 2020; Coskun et al., 2017; Ullah et al., 2019). Our findings imply that this methodology is wrong, since the baseline is not optimized to the same level as the learning model. Hussein (2014) explicitly discusses the sensitivity of EKF performance to the noise model accuracy, and suggests the solution of a NN with supervised learning – without considering the same supervised learning for the EKF.

7 Summary

Through a detailed case study, we demonstrated both analytically and empirically the fragility of the KF assumptions, and how the slightest violation of them may change the effective noise in the problem – leading to significant and non-trivial changes in the optimal noise parameters. We addressed this problem using optimization tools from supervised machine learning, and suggested how to apply them efficiently to the SPD parameters of the KF.

We demonstrated the accuracy of our method over different violations of KF assumptions, in different domains (radar tracking, video tracking and lidar-based state estimation), with relation to different variants of the KF, over small and large training datasets, and even under distributional shifts between train and test datasets. Indeed, once we acknowledged the need for optimization and applied the Cholesky parameterization, the optimization itself performed robustly over all these scenarios. In light of this evidence, we recommend to use our method as the default procedure for the KF tuning in presence of ground-truth data, whenever the KF assumptions are not strictly-guaranteed.

We also demonstrated one of the consequences of using a sub-optimal KF: the common methodology of comparing learning filtering algorithms to classic variants of the KF is misleading, as it essentially compares an optimized model to a non-optimized one. We argued that the baseline method should be optimized similarly to the researched one, e.g., using optimization rather than noise estimation.

REFERENCES

Pieter Abbeel, Adam Coates, Michael Montemerlo, Andrew Ng, and Sebastian Thrun. Discriminative training of kalman filters. *Robotics: Science and systems*, pages 289–296, 06 2005. doi: 10.15607/RSS.2005.I.038.

S. Akhlaghi, N. Zhou, and Z. Huang. Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation. In *2017 IEEE Power Energy Society General Meeting*, pages 1–5, 2017. doi: 10.1109/PESGM.2017.8273755.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters in on-policy reinforcement learning? a large-scale empirical study, 2020.

Zafer Aydogmus and Omur Aydogmus. A comparison of artificial neural network and extended kalman filter based sensorless speed estimation. *Measurement*, 63:152–158, 2015. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2014.12.010. URL https://www.sciencedirect.com/science/article/pii/S0263224114006071.

Yu-ting Bai, Xiao-yi Wang, Xue-bo Jin, Zhi-yao Zhao, and Bai-hai Zhang. A neuron-based kalman filter with nonlinear autoregressive model. *Sensors*, 20(1):299, 2020.

S. T. Barratt and S. P. Boyd. Fitting a kalman smoother to data. In *2020 American Control Conference (ACC)*, pages 1526–1531, 2020. doi: 10.23919/ACC45564.2020.9147485.

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. doi: 10.1109/ICIP.2016.7533003.

S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House Radar Library, Boston, 1999.

W. R. Blanding, P. K. Willett, and Y. Bar-Shalom. Multiple target tracking using maximum likelihood probabilistic data association. In *2007 IEEE Aerospace Conference*, pages 1–12, 2007. doi: 10.1109/AERO.2007.353035.

Chaw-Bing Chang and Keh-Ping Dunn. Radar tracking using state estimation and association: Estimation and association in a multiple radar system, 04 2019.

Zhaozhong Chen et al. Kalman filter tuning with bayesian optimization, 2019.

S. Kumar Chenna, Yogesh Kr. Jain, Himanshu Kapoor, Raju S. Bapi, N. Yadaiah, Atul Negi, V. Seshagiri Rao, and B. L. Deekshatulu. State estimation and tracking problems: A comparison between kalman filter and recurrent neural networks. *ICONIP*, 2004.

Huseyin Coskun, Felix Achilles, Robert DiPietro, Nassir Navab, and Federico Tombari. Long short-term memory kalman filters:recurrent neural estimators for pose regularization. *ICCV*, 2017. URL https://github.com/Seleucia/lstmkf_ICCV2017.

Philip Zhuang Dan Iter, Jonathan Kuck. Target tracking with kalman filtering, knn and lstms, 2016. URL http://cs229.stanford.edu/proj2016/report/IterKuckZhuang-TargetTrackingwithKalmanFilteringKNNandLSTMs-report.pdf.

JP DeCruyenaere and HM Hafez. A comparison between kalman filters and recurrent neural networks. *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, 4:247–251, 1992.

Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes, 2020. URL https://motchallenge.net/data/MOT20/.

Lichuan Deng, Da Li, and Ruifang Li. Improved IMM Algorithm based on RNNs. *Journal of Physics Conference Series*, 1518:012055, April 2020. doi: 10.1088/1742-6596/1518/1/012055.

Jimmy Ba Diederik P. Kingma. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo. *ICLR*, 2019. URL https://openreview.net/pdf?id=r1etN1rtPB.

Hai fa Dai, Hong wei Bian, Rong ying Wang, and Heng Ma. An ins/gnss integrated navigation in gnss denied environment using recurrent neural network. *Defence Technology*, 16(2):334–340, 2020. ISSN 2214-9147. doi: https://doi.org/10.1016/j.dt.2019.08.011. URL https://www.sciencedirect.com/science/article/pii/S2214914719303058.

B. Feng, M. Fu, H. Ma, Y. Xia, and B. Wang. Kalman filter with recursive covariance estimation—sequentially estimating process noise covariance. *IEEE Transactions on Industrial Electronics*, 61(11):6253–6263, 2014. doi: 10.1109/TIE.2014.2301756.

Simone Formentin and Sergio Bittanti. An insight into noise covariance estimation for kalman filter design. *IFAC Proceedings Volumes*, 47(3):2358–2363, 2014. ISSN 1474-6670. doi: https://doi.org/10.3182/20140824-6-ZA-1003.01611. URL https://www.sciencedirect.com/science/article/pii/S1474667016419646. 19th IFAC World Congress.

C. Gao, H. Liu, S. Zhou, H. Su, B. Chen, J. Yan, and K. Yin. Maneuvering target tracking with recurrent neural networks for radar application. *2018 International Conference on Radar (RADAR)*, pages 1–5, 2018.

Chang Gao, Junkun Yan, Shenghua Zhou, Bo Chen, and Hongwei Liu. Long short-term memory-based recurrent neural networks for nonlinear target tracking. *Signal Processing*, 164, 05 2019. doi: 10.1016/j.sigpro.2019.05.027.

Matthieu Geist and Olivier Pietquin. Kalman filtering colored noises: the (autoregressive) moving-average case. In *MLASA 2011*, pages 1–4, Honolulu, United States, December 2011. URL https://hal-supelec.archives-ouvertes.fr/hal-00660607.

Google Scholar. Citations since 2017: "a new approach to linear filtering and prediction problems", 2021. URL https://scholar.google.com/scholar?as_ylo=2017&hl=en&as_sdt=2005&sciodt=0,5&cites=5225957811069312144&scipsc=.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep Reinforcement Learning that Matters. *arXiv preprint arXiv:1709.06560*, 2017. URL https://arxiv.org/pdf/1709.06560.pdf.

Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997. URL https://direct.mit.edu/neco/article/9/8/1735/6109/Long-Short-Term-Memory.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

Jeffrey Humpherys, Preston Redd, and Jeremy West. A fresh look at the kalman filter. *SIAM Review*, 54(4):801–823, 2012. doi: 10.1137/100799666.

Ala A. Hussein. Kalman filters versus neural networks in battery state-of-charge estimation: A comparative study. *International Journal of Modern Nonlinear Theory and Application*, 2014.

Faisal Jamil et al. Toward accurate position estimation using learning to prediction algorithm in indoor navigation. *Sensors*, 2020.

S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004. doi: 10.1109/JPROC.2003.823141.

R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. ISSN 0021-9223. doi: 10.1115/1.3662552. URL https://doi.org/10.1115/1.3662552.

Chanho Kim, Fuxin Li, and James M. Rehg. Multi-object tracking with neural gating using bilinear lstm. *ECCV*, September 2018.

Yaakov Bar-Shalom X.-Rong Li Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley and Sons, Inc., 2002. doi: 10.1002/0471221279.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Tony Lacey. Tutorial: The kalman filter, 1998. URL "http://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalmanfilter.pdf".

S. Li, C. De Wagter, and G. C. H. E. de Croon. Unsupervised tuning of filter parameters without ground-truth applied to aerial robots. *IEEE Robotics and Automation Letters*, 4(4):4102–4107, 2019. doi: 10.1109/LRA.2019.2930480.

M. Linderoth, K. Soltesz, A. Robertsson, and R. Johansson. Initialization of the kalman filter without assumptions on the initial state. In *2011 IEEE International Conference on Robotics and Automation*, pages 4992–4997, 2011. doi: 10.1109/ICRA.2011.5979684.

Hu Liu et al. Kalman filtering attention for user behavior modeling in ctr prediction. *NeurIPS*, 2020.

Huajun Liu, Hui Zhang, and Christoph Mertz. Deepda: Lstm-based deep data association network for multi-targets tracking in clutter. *CoRR*, abs/1907.09915, 2019a. URL http://arxiv.org/abs/1907.09915.

Jingxian Liu, Zulin Wang, and Mai Xu. Deepmtt: A deep learning maneuvering target-tracking algorithm based on bidirectional lstm network. *Information Fusion*, 53, 06 2019b. doi: 10.1016/j.inffus.2019.06.012.

E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, 1998.

A. Paulo Moreira, Paulo Costa, and José Lima. New approach for beacons based mobile robot localization using kalman filters. *Procedia Manufacturing*, 51:512–519, 2020. URL https://www.sciencedirect.com/science/article/pii/S2351978920319296. 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021).

Dominic A. Neu, Johannes Lahann, and Peter Fettke. A systematic literature review on state-of-the-art deep learning methods for process prediction. *CoRR*, abs/2101.09320, 2021. URL https://arxiv.org/abs/2101.09320.

Brian Odelson, Alexander Lutz, and James Rawlings. The autocovariance least-squares method for estimating covariances: Application to model-based control of chemical reactors. *Control Systems Technology, IEEE Transactions on*, 14:532 – 540, 06 2006. doi: 10.1109/TCST.2005.860519.

Sebin Park et al. Measurement noise recommendation for efficient kalman filtering over a large amount of sensor data. *Sensors*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Dong-liang Peng and Yu Gu. Imm algorithm for a 3d high maneuvering target tracking. *International Conference in Swarm Intelligence*, pages 529–536, 06 2011. doi: 10.1007/978-3-642-21524-7_65.

José C. Pinheiro and Douglas M. Bates. Unconstrained parameterizations for variance-covariance matrices. *Statistics and Computing*, 6:289–296, 1996.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics, 2021.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech house Boston, 2004.

David E. Rumelhart et al. Learning representations by back-propagating errors. *Nature*, 1986. URL https://www.nature.com/articles/323533a0.

A. Sengupta, F. Jin, and S. Cao. A dnn-lstm based target tracking approach using mmwave radar and camera sensor fusion. *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, pages 688–693, 2019. doi: 10.1109/NAECON46414.2019.9058168.

Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2005. ISBN 0387989501.

Harold Wayne Sorenson. *Kalman Filtering: Theory and Application*. IEEE Press, 1985.

D. J. Thomson. Jackknifing multiple-window spectra. In *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume vi, pages VI/73–VI/76 vol.6, 1994. doi: 10.1109/ICASSP.1994.389899.

Ryan Tibshirani. Proximal gradient descent and acceleration, 2015. URL https://www.stat.cmu.edu/~ryantibs/convexopt-F15/lectures/08-prox-grad.pdf.

Koji Tsuda, Gunnar Rätsch, and Manfred K. Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projection. *Journal of Machine Learning Research*, 6(34):995–1018, 2005. URL http://jmlr.org/papers/v6/tsuda05a.html.

Israr Ullah, Muhammad Fayaz, and DoHyeun Kim. Improving accuracy of the kalman filter algorithm in dynamic conditions using ann-based learning module. *Symmetry*, 11(1), 2019. ISSN 2073-8994. doi: 10.3390/sym11010094. URL https://www.mdpi.com/2073-8994/11/1/94.

Ashish Vaswani et al. Attention is all you need. *NeurIPS*, 2017.

E. Wan. Sigma-point filters: An overview with applications to integrated navigation and vision assisted control. *IEEE*, pages 201–202, 2006. doi: 10.1109/NSSPW.2006.4378854.

E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158, 2000. doi: 10.1109/ASSPCC.2000.882463.

Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017. doi: 10.1109/ICIP.2017.8296962.

L. Zanni, J. Le Boudec, R. Cherkaoui, and M. Paolone. A prediction-error covariance estimator for adaptive kalman filtering in step-varying processes: Application to power-system state estimation. *IEEE Transactions on Control Systems Technology*, 25(5):1683–1697, 2017. doi: 10.1109/TCST.2016.2628716.

Paul Zarchan and Howard Musoff. *Fundamentals of Kalman Filtering: A Practical Approach*. American Institute of Aeronautics and Astronautics, 2000.

Tianyu Zheng, Yu Yao, Fenghua He, and Xinran Zhang. An rnn-based learnable extended kalman filter design and application. In *2019 18th European Control Conference (ECC)*, pages 3304–3309, 2019. doi: 10.23919/ECC.2019.8796088.

Hui Zhong, Zaiyi Chen, Chuan Qin, Zai Huang, Vincent W. Zheng, Tong Xu, and Enhong Chen. Adam revisited: a weighted past gradients perspective. *Frontiers of Computer Science*, 2020.