Iteratively Saturated Kalman Filtering

Alan Yang Stephen Boyd

Abstract

The Kalman filter (KF) provides optimal recursive state estimates for linear-Gaussian systems and underpins applications in control, signal processing, and others. However, it is vulnerable to outliers in the measurements and process noise. We introduce the iteratively saturated Kalman filter (ISKF), which is derived as a scaled gradient method for solving a convex robust estimation problem. It achieves outlier robustness while preserving the KF's low per-step cost and implementation simplicity, since in practice it typically requires only one or two iterations to achieve good performance. The ISKF also admits a steady-state variant that, like the standard steady-state KF, does not require linear system solves in each time step, making it well-suited for real-time systems.

1.1 Introduction

The Kalman filter is the prevalent tool for state estimation, prized for its simplicity, low computational cost, and optimality for linear-Gaussian systems. It has found extensive use in many fields, including control, signal processing, robotics, navigation, neural interface systems, and econometrics [1], [2], [3], [4]. Despite its popularity, the KF is notoriously vulnerable to outliers in the measurements and process noise in the dynamics [5]. Measurement outliers may arise from occasional sensor malfunctions, while process noise outliers can result from sudden shocks to the system or unmodeled dynamics.

In this work, we propose the iteratively saturated Kalman filter, which is a modification of the standard KF's update (or correction) step. It iterates a modified KF update step, in which a saturating nonlinearity is applied to compensate for both measurement and process noise outliers. The method is derived as a scaled gradient method [6], [7] for solving a

particular convex robust estimation problem involving the Huber function. Since the ISKF typically requires only one or two iterations to achieve good performance, it retains the standard KF's ease of implementation and per-step cost.

A key advantage of the ISKF is its steady-state variant, which matches the computational efficiency of the steady-state KF. Whereas the full KF must update its covariance estimate at each step, incurring matrix-matrix multiplications and linear solves, the steady-state KF uses precomputable gain matrices and requires only matrix-vector multiplications and vector additions. Our steady-state ISKF inherits this low per-step cost while compensating for both measurement and process-noise outliers. In contrast, existing robust KF extensions either lack robustness to process-noise outliers or rely on full covariance estimates in each step.

The rest of the paper is organized as follows. We review prior work in §1.2. The system model is given in §1.3, and the iteratively saturated Kalman filter is introduced in §1.4. We derive the filter as a scaled gradient method in §1.5. Finally, we present numerical experiments in §1.8 and conclude in §1.9.

1.2 Prior work

There is a large body of work on modifying the KF to be robust to outliers without sacrificing its computational efficiency. Many of these heuristics involve modifying the covariance estimate in each KF step, by scaling the measurement noise covariance matrix or the process and prior covariance matrices when outliers are detected [8], [9], [10]. The idea is that if an outlier is detected, the corresponding covariance should be scaled up to account for the increased uncertainty. Some variations of this idea were derived by replacing the Gaussian distribution used by the KF with heavy-tailed distributions [11], [12]. Others are derived by Huberizing the quadratic costs used by the KF [5], [8], [13], [14], [15], [16].

Yet others have proposed combining the KF with inlier detection methods such as RANSAC [17], [18]. Several of the robust KF methods have also been extended to nonlinear systems [19], [20].

Our method generalizes the saturated KF [21] by compensating for process noise outliers in addition to measurement outliers. In the single-step case, our method is almost equivalent to the saturated KF, but uses a different saturation function.

Besides modified KF methods, we also mention particle filtering methods, which have been applied to handle outliers [22], although they tend to be computationally expensive.

1.3 System model

We consider a linear time-invariant dynamical system that evolves according to

$$x_{t+1} = Ax_t + w_t, \quad y_t = Cx_t + v_t, \quad t = 0, 1, \ldots, \tag{1.1}$$

where t denotes time or epoch, $x_t \in \mathbf{R}^n$ is the state, and $y_t \in \mathbf{R}^p$ is the output measurement. The matrix $A \in \mathbf{R}^{n\times n}$ is the state dynamics matrix, and $C \in \mathbf{R}^{p\times n}$ is the output matrix. We assume that the dynamics matrix A and the output matrix C are known. The dynamics are driven by the process noise $w_t \in \mathbf{R}^n$, and the outputs are influenced by the measurement noise $v_t \in \mathbf{R}^p$. We assume that the initial state x_0 is Gaussian, with $x_0 \sim \mathcal{N}(0, X_0)$.

Linear Gaussian model. In the classical model, the process noise $w_t \in \mathbf{R}^n$ and the measurement noise $v_t \in \mathbf{R}^p$ are Gaussian with

$$w_t \sim \mathcal{N}(0, W), \quad v_t \sim \mathcal{N}(0, V),$$

where W is the known positive semidefinite (PSD) process noise covariance (which can be degenerate, *i.e.*, singular) and V is the known positive definite (PD) measurement noise covariance. We assume that the initial state x_0, the process noise w_t, and the measurement noise v_t are independent and identically distributed (IID). In this case, the state and measurements are jointly Gaussian, and the Kalman filter [1] gives the optimal estimate of the state, both in the minimum mean squared error (MMSE) and maximum a posteriori (MAP) sense.

Outlier model. In this work, we consider a model where the process and measurement noises are typically Gaussian, but may occasionally be corrupted by outliers. We consider the model

$$w_t = F(\tilde{w}_t + s_t), \quad v_t = G(\tilde{v}_t + o_t), \tag{1.2}$$

where $F \in \mathbf{R}^{n\times m}$ and $G \in \mathbf{R}^{p\times p}$ are known matrices, and $\tilde{w}_t \in \mathbf{R}^m$ and $\tilde{v}_t \in \mathbf{R}^p$ are zero-mean whitened Gaussian noises with $\tilde{w}_t \sim \mathcal{N}(0, I)$ and

$\tilde{v}_t \sim \mathcal{N}(0, I)$. The additional terms $s_t \in \mathbf{R}^n$ and $o_t \in \mathbf{R}^p$ are sparse outlier terms which are zero for most times t.

The process noise outliers s_t can result from system shocks or unmodeled dynamics, while measurement outliers o_t can arise from sensor malfunctions or environmental disturbances. In the absence of outliers, *i.e.*, when $s_t = 0$ and $o_t = 0$ for all t, the system reduces to the linear Gaussian model, with $W = FF^T$ and $V = GG^T$.

Extensions. Several extensions of the system model are readily handled. The model can be modified to handle known control inputs, process and measurement noises w_t and v_t with nonzero mean, and correlation between w_t and v_t. We may also consider the time-varying case, where A, C, W, and V are allowed to vary with t.

1.4 Iteratively saturated Kalman filtering

Given a sequence of measurements $y_1, \ldots, y_t$, our goal is to recursively estimate both the state x_t and its covariance P_t. At each time step t, we update our previous estimates $\hat{x}_{t-1|t-1}$ and $P_{t-1|t-1}$ to obtain new estimates $\hat{x}_{t|t}$ and $P_{t|t}$. We assume in the following that (A, C) is detectable and $(A, W^{1/2})$ is stabilizable [23]. Here, $W^{1/2}$ denotes any matrix such that $(W^{1/2})^T W^{1/2} = W$. Such a matrix can be found from the eigenvalue decomposition of W, or when W is PD, via Cholesky factorization.

Our estimator starts with the standard Kalman filter prediction step

$$\begin{aligned} \hat{x}_{t|t-1} &= A\hat{x}_{t-1|t-1} && (1.3)\\ P_{t|t-1} &= AP_{t-1|t-1}A^T + W. && (1.4) \end{aligned}$$

The update step is then given by the iteration

$$\begin{aligned} \hat{x}^0 &= \hat{x}_{t|t-1} && (1.5)\\ \hat{x}^k &= \hat{x}^{k-1} + K_t\sigma(y_t - C\hat{x}^{k-1}) + (I - K_tC)\rho_t(\hat{x}^0 - \hat{x}^{k-1}), && (1.6) \end{aligned}$$

for $k = 1, \ldots, \tilde{k}$, where $\tilde{k}$ denotes the number of iterations. We then take $\hat{x}_{t|t} = \hat{x}^{\tilde{k}}$. The nonlinear functions

$$\rho_t(z) = \min\left(1, \frac{\lambda_x}{\|P_{t|t-1}^{-1/2}z\|_2}\right)z, \quad \sigma(z) = \min\left(1, \frac{\lambda_y}{\|V^{-1/2}z\|_2}\right)z,$$

saturate their inputs at threshold values λ_x and λ_y, respectively, and K_t is the Kalman gain matrix satisfying the standard Kalman filter covariance

update equations

$$K_t = P_{t|t-1}C^T(CP_{t|t-1}C^T + V)^{-1} \tag{1.7}$$
$$P_{t|t} = (I - K_tC)P_{t|t-1}. \tag{1.8}$$

Comments. Our filter compensates for outliers in the measurements by reducing the effect of outliers in the measurement noise on the state estimate by attenuating the magnitude of the *innovation* $y_t - C\hat{x}_{t|t-1}$ when

$$\|V^{-1/2}(y_t - C\hat{x}_{t|t-1})\|_2$$

is large, *i.e.*, unlikely under the Gaussian noise model. Similarly, our filter compensates for outliers in the process noise by attenuating the magnitude of the proximity to the predicted state $\hat{x}_{t|t-1}$.

Single-step case. When $\tilde{k} = 1$, the update step (1.6) is simply

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t\sigma(y_t - C\hat{x}_{t|t-1}).$$

In this case, the ISKF does not compensate for outliers in the process noise, only in the measurements. It closely resembles the standard KF, except that the innovation $y_t - C\hat{x}_{t|t-1}$ is attenuated by the function σ. The single-step ISKF is very similar to the saturated KF [21], but uses a different saturation function.

Outlier-free case. In the absence of detected outliers, our state estimate $\hat{x}_{t|t}$ satisfies

$$\|V^{-1/2}(y_t - C\hat{x}_{t|t})\|_2 \leq \lambda_y, \quad \|P_{t|t-1}^{-1/2}(\hat{x}_{t|t} - \hat{x}_{t|t-1})\|_2 \leq \lambda_x,$$

and the saturation functions ρ_t and σ reduce to the identity function. In this case, the ISKF is equivalent to the standard KF

$$\hat{x}_{t|t}^{\mathrm{kf}} = \hat{x}_{t|t-1} + K_t(y_t - C\hat{x}_{t|t-1}),$$

with $\hat{x}^k = \hat{x}_{t|t}^{\mathrm{kf}}$ for all k.

Computational cost. At each time step, the ISKF costs $O(\tilde{k}(n^3 + p^3 + np))$ floating-point operations (FLOPS) online, dominated by matrix-matrix products and solving a linear system. For $\tilde{k}$ fixed and small, this is comparable to the cost of the KF. In our experiments, we found that values between 1 and 5 were effective.

Steady-state case. A key property of the Kalman filter recursion is that the covariance update steps (1.4) and (1.8) are independent of the measurements. This means we can compute $P_{t|t}$ offline, before processing any measurements. Furthermore, as $t \to \infty$, the covariance matrices $P_{t|t}$, $P_{t|t-1}$, and gain matrix K_t converge to steady-state values given by:

$$\begin{aligned} P &= APA^T + W - APC^T(CPC^T + V)^{-1}CPA^T \\ \Sigma &= APA^T + W \\ K &= \Sigma C^T(C\Sigma C^T + V)^{-1} \end{aligned}$$

respectively. In steady-state, we may dispense with the covariance and gain update steps (1.4), (1.7), and (1.8).

This leads to the *steady-state ISKF*

$$\hat{x}^0 = \hat{x}_{t|t-1} \tag{1.9}$$

$$\hat{x}^k = \hat{x}^{k-1} + K\sigma(y_t - C\hat{x}^{k-1}) + (I - KC)\rho(\hat{x}^0 - \hat{x}^{k-1}), \tag{1.10}$$

for $k = 1, \ldots, \tilde{k}$. Like before, we take $\hat{x}_{t|t} = \hat{x}^{\tilde{k}}$. Here, we drop the subscript t on the function ρ, since it is now time-invariant.

Since K can be precomputed offline, the steady-state ISKF only requires matrix-vector multiplications and vector additions online, with has cost $O(\tilde{k}(n^2+np))$ FLOPS online. Since the convergence of P_t to P is in practice often quick, the ISKF estimate (1.10) gives an excellent approximation to the ISKF (1.6).

1.5 Derivation as a scaled gradient method

In this section, we show that the ISKF can be interpreted as a scaled gradient method for solving a convex regularized maximum a posteriori (MAP) estimation problem in which we estimate x_t, s_t, and o_t jointly. We focus on the steady-state case for simplicity, but the discussion applies to the general case by replacing P with $P_{t|t}$ and Σ with $P_{t|t-1}$, and adding the subscript t on K and ρ.

1.5.1 Model

At each time step t, we consider the optimization problem

$$\text{minimize } \frac{1}{2}\left\| \begin{bmatrix} \Sigma^{-1/2}(x - \hat{x}_{t|t-1} - s) \\ V^{-1/2}(y_t - Cx - o) \end{bmatrix} \right\|_2^2 + \lambda_x\|\Sigma^{-1/2}s\|_2 + \lambda_y\|V^{-1/2}o\|_2$$

with variables x, s, and o. This is a convex optimization problem, and it has the following interpretation. When the outlier terms s and o are known and fixed, the first term corresponds to the negative log likelihood of the prior distribution over x_t and the measurement noise. The second and third terms are sparse regularization terms on s and o. When the thresholds λ_x and λ_y are infinite, the optimal values of s and o are zero, and the optimal value of x is simply given by the (steady-state) KF estimate $\hat{x}^{\mathrm{kf}}_{t|t}$.

Comments. Our optimization problem is only an approximate MAP estimate, since in general we do not have a statistical model of the outliers s_t and o_t. Moreover, in the presence of process noise outliers, the prior distribution over x_t is no longer necessarily Gaussian with mean $\hat{x}_{t|t-1}$ and covariance P. The goal is to reduce the influence of outliers on the state estimate by regularization, where the regularization terms are chosen such that the partial minimizations over s and o have closed-form solutions which can be written in terms of the Huber function.

Circular Huber function. Let the function φ denote the partial minimization

$$\varphi(a;\lambda) = \min_b \left(\frac{1}{2}\|a-b\|_2^2 + \lambda\|b\|_2 \right).$$

The function φ has a closed-form solution given by

$$\varphi(a;\lambda) = \begin{cases} \frac{1}{2}\|a\|_2^2 & \|a\|_2 \leq \lambda, \\ \lambda(\|a\|_2 - \lambda/2) & \|a\|_2 > \lambda. \end{cases}$$

We refer to this function as the *circular Huber function* with threshold λ. It is a smooth convex function that is quadratic for $\|a\|_2 \leq \lambda$, and grows only linearly with the norm of a for $\|a\|_2 > \lambda$. It is equivalent to the standard Huber function, composed with the Euclidean norm.

The estimation problem may then be written as

$$\hat{x}_{t|t} = \operatorname*{argmin}_x f(x),$$

where

$$f(x) = \varphi(\Sigma^{-1/2}(x - \hat{x}_{t|t-1}); \lambda_x) + \varphi(V^{-1/2}(y_t - Cx); \lambda_y). \tag{1.11}$$

For future reference, we observe that the Hessian satisfies $\nabla^2\varphi(a;\delta) \leq I$, which implies that its gradient $\nabla\varphi(a;\delta)$ has Lipschitz constant one.

1.5.2 Scaled gradient method

We propose a scaled gradient method for minimizing (1.11) over $x \in \mathbf{R}^n$. The method has iterates

$$x^{k+1} = x^k - \eta M^{-1} \nabla f(x^k), \quad k = 0, 1, \ldots, \tilde{k} - 1,$$

where $\eta \in (0, 2)$ is a constant step size,

$$M = \Sigma^{-1} + C^T V^{-1} C$$

is the scaling matrix, and the initial iterate $x^0 = \hat{x}_{t|t-1}$ is the predict step given by (1.3).

The gradient of f is given by

$$\begin{aligned} \nabla f(x) &= (\Sigma^{-1/2})^T \nabla \varphi \left(\Sigma^{-1/2}(x - \hat{x}_{t|t-1}); \lambda_x \right) \\ &\quad - C^T (V^{-1/2})^T \nabla \varphi \left(V^{-1/2}(y_t - Cx); \lambda_y \right), \end{aligned} \tag{1.12}$$

where

$$\nabla \varphi(a; \lambda) = \begin{cases} a & \|a\|_2 \leq \lambda, \\ \lambda a / \|a\|_2 & \|a\|_2 > \lambda. \end{cases} \tag{1.13}$$

To simplify the scaled gradient $M^{-1} \nabla f(x)$, we can use the fact that the Kalman gain can be written as

$$K = \Sigma C^T (C \Sigma C^T + V)^{-1} = M^{-1} C^T V^{-1},$$

by applying the Woodbury matrix identity. Then, since $\nabla \varphi(a; \lambda)$ is a scalar multiple of a, the scaled gradient becomes

$$M^{-1} \nabla f(x) = -K \sigma(y_t - Cx) - (I - KC) \rho(\hat{x}_{t|t-1} - x).$$

By setting the step size $\eta = 1$, we arrive at the ISKF update (1.6). Note that in practice, it may be numerically advantageous to implement the saturation functions as

$$\rho(z) = \nabla \varphi(\Sigma^{-1/2} z; \lambda_x), \quad \sigma(z) = \nabla \varphi(V^{-1/2} z; \lambda_y)$$

where $\nabla \varphi$ has the form in (1.13), since the division by $\|a\|_2$ only occurs when $\|a\|_2 > \lambda$.

Choice of gradient method. In general a gradient method would be a very poor choice for a problem that must be solved in real-time, since its practical convergence can vary considerably depending on the input data.

An interior-point method, which typically takes around 20 or so steps independent of the data, would seem like a better choice. We propose the use of a gradient method in this specific case only because an excellent estimate of the curvature of the function f is available, which eliminates the need for a line search, and gives very good estimates in just a few iterations, independent of the data.

1.5.3 Convergence

We now show that the scaled gradient method converges to a solution, and is a strict descent method, when the iterations are not terminated. Let L be a matrix that satisfies $LL^T = M^{-1}$, *e.g.*, the Cholesky factorization of M^{-1}. Let $g(z) = f(Lz)$. The scaled gradient method is then equivalent to the iteration

$$z^{k+1} = z^k - \eta \nabla g(z^k), \quad k = 0, 1, \ldots, T-1,$$

where $x^k = Lz^k$ for all k. To establish convergence and the descent property, it is sufficient to show that the gradient ∇g is Lipschitz continuous with constant one [24]. Note that an upper bound on the Lipschitz constant of ∇g can be found by taking λ_x and λ_y to infinity, in which case g is a quadratic function. In that case, $g(z)$ is given by

$$\begin{aligned} g(z) &= \|\Sigma^{-1/2}(Lz - \hat{x}_{t|t-1})\|_2^2 + \|V^{-1/2}(y_t - CLz)\|_2^2 \\ &= \left\| \begin{bmatrix} \Sigma^{-1/2} \\ -V^{-1/2}C \end{bmatrix} Lz - \begin{bmatrix} \Sigma^{-1/2}\hat{x}_{t|t-1} \\ -V^{-1/2}y_t \end{bmatrix} \right\|_2^2. \end{aligned}$$

Since

$$L^T \begin{bmatrix} \Sigma^{-1/2} \\ -V^{-1/2}C \end{bmatrix}^T \begin{bmatrix} \Sigma^{-1/2} \\ -V^{-1/2}C \end{bmatrix} L = L^T M L = I,$$

it follows that ∇g has Lipschitz constant one.

1.6 Parameter selection

The performance of the ISKF depends on the number of iterations $\tilde{k}$ and the choice of parameters λ_x and λ_y.

Number of iterations. In our experiments, we found that the ISKF can compensate for outliers in the measurement noise even with $\tilde{k} = 1$, and outliers in both the measurement noise and process noise even with $\tilde{k} = 2$. While small improvements can be achieved for larger $\tilde{k}$, we found $\tilde{k} = 2$ to be a good default choice, with more iterations giving diminishing returns.

Threshold parameters. The parameters λ_x and λ_y balance robustness against outliers and estimate bias. Larger values of λ_x and λ_y are ideal when there are no outliers (with $\lambda_x = \lambda_y = \infty$ reducing to the standard KF), while tuned values improve performance when outliers are present.

Step size. The discussion in §1.5.2 suggests that the step size η in the scaled gradient method could be made a tunable parameter. The (steady-state) ISKF (1.10) can be modified as

$$\hat{x}^0 = \hat{x}_{t|t-1} \tag{1.14}$$

$$\hat{x}^k = \hat{x}^{k-1} + \eta K\sigma(y_t - C\hat{x}^{k-1}) + \eta(I - KC)\rho(\hat{x}^0 - \hat{x}^{k-1}). \tag{1.15}$$

However, we suggest fixing $\eta = 1$ in practice. In our experiments, we found that increasing η to be greater than one could give marginal improvements, but at the cost of reducing the performance of the filter when there are no outliers present. This is because η effectively acts as a type of gain parameter for the filter. This is most clearly seen in the single-step case. When $\tilde{k} = 1$ and there are no detected outliers (the saturation function σ is identity), the ISKF reduces to the standard KF with gain matrix ηK. Choosing $\eta = 1$ leads to a natural interpretation, since the filter is then equivalent to the standard KF when there are no detected outliers.

Grid search. The parameters may be chosen via a simple grid search, given a sequence of measurements $y_1, \ldots, y_N$ collected in the past. For each combination of parameters, the ISKF can be run on the past data containing outliers, and the parameter combination that best predicts the observed measurements is chosen. For each parameter combination, we compute the RMSE of the predicted measurements

$$\text{RMSE} = \left(\frac{1}{N} \sum_{t=1}^{N} \|y_t - CA\hat{x}_{t-1|t-1}\|_2^2 \right)^{1/2}, \tag{1.16}$$

although other metrics could be used. Note that we consider the RMSE of the residuals $y_t - CA\hat{x}_{t-1|t-1}$, rather than the innovations $y_t - C\hat{x}_{t|t}$, since the estimate $\hat{x}_{t|t}$ is computed as a function of y_t.

In practice, a strategy for choosing the parameters is to first fix the number of iterations $\tilde{k}$ based on the computational budget, and then search over λ_x and λ_y to minimize the RMSE.

1.7 Extensions and variations

1.7.1 Time-varying and non-linear systems

When the system model (1.1) is time-varying, we introduce subscripts t to the matrices A, C, W, and V. The ISKF naturally extends to this scenario, though a steady-state version typically does not exist. For nonlinear systems, we approximate the model by linearizing around the current state estimate, resulting in time-varying matrices A_t, C_t, W_t, and V_t. Similar to how the EKF and UKF generalize the KF to nonlinear cases, analogous modifications enable the ISKF to handle nonlinear systems by iteratively updating the linearization.

1.7.2 Missing measurements

We have assumed thus far that the measurements y_t are fully available at each time t. When this is not the case, the update step is replaced with conditioning on only the *known entries* of y_t. The most general way to handle this is to allow for any subset of the entries of y_t to be known or unknown. This is equivalent to the case where the measurement matrix C and measurement covariance V are time-varying, where only the rows of C and V corresponding to the available measurements are included. In the case where no measurements are available, the update step is skipped.

However, this approach leads to a time-varying system, so cannot be applied to the steady-state case without increasing the online computational cost from $O(n^2 + np)$ to $O(n^3 + p^3 + np)$. Alternatively, there are only 2^p possible patterns of missing measurements. If 2^p isn't too large, we could precompute a different steady-state Kalman gain matrix K for each pattern of known measurements.

1.8 Numerical experiments

In this section, we present numerical experiments evaluating the performance of the ISKF, in comparison with the KF and other outlier-robust filters. In our experiments, we use the steady-state form of the ISKF and the KF.

Competing methods. We compare the steady-state ISKF against the steady-state KF, and two outlier-robust Kalman filter variants: the weighted observation likelihood filter (WoLF) [10] and the Huberized KF [5], [8],

[13], [14], [15]. WoLF is a covariance scaling method, and is only designed to reject measurement outliers. Our implementation of the Huberized KF solves the Huber regression problem (1.11) using the interior-point method Clarabel [25]. Both WoLF and the Huberized KF have online computational cost approximately $O(n^3 + p^3 + np)$, comparable to the full KF. In contrast, the steady-state ISKF and steady-state KF have online cost $O(n^2 + np)$.

Evaluation. We evaluate the performance of each filter on a simulated test trajectory of, using the state estimate RMSE

$$\text{RMSE} = \left(\frac{1}{T} \sum_{t=1}^{T} \|x_t - \hat{x}_{t|t}\|_2^2 \right)^{1/2}, \tag{1.17}$$

where x_t is the true state and $\hat{x}_{t|t}$ is the state estimate produced by a filter. For the purposes of evaluation, we assume that the true state trajectory is available. Test trajectories had lengths of 1000 time steps.

Parameter selection. We tuned the parameters of each filter using a separate simulated trajectory than the test trajectory used for evaluation. Unlike the test trajectory, we assumed that the true state trajectory was not available for the tuning trajectory data. Instead, we minimized the predicted measurement RMSE (1.16) in our grid search. For all parameters, we considered 20 values between 0.1 and 10, logarithmically spaced. Like the test trajectory, the tuning trajectory had length 1000 time steps. The ISKF (for $\tilde{k} > 1$) and the Huberized KF have two tunable parameters, and WoLF has one.

1.8.1 Vehicle tracking

System model. The position and velocity of a vehicle in two dimensions are denoted $\xi_t \in \mathbf{R}^2$ and $\nu_t \in \mathbf{R}^2$. At time t, we observe a noisy measurement of the position ξ_t, and aim to estimate the state $x_t = (\xi_t, \nu_t)$. The vehicle has unit mass, and is subject to a drag force $-\gamma\nu_t$ with coefficient of friction $\gamma = 0.05$. The discrete-time system with time step $h = 0.05$ is

$$x_{t+1} = Ax_t + Bu_t, \quad y_t = Cx_t + v_t,$$

where

$$A = \begin{bmatrix} 1 & 0 & \left(1 - \frac{\gamma h}{2}\right) h & 0 \\ 0 & 1 & 0 & \left(1 - \frac{\gamma h}{2}\right) h \\ 0 & 0 & 1 - \gamma h & 0 \\ 0 & 0 & 0 & 1 - \gamma h \end{bmatrix},$$

and

$$B = \begin{bmatrix} \frac{h^2}{2} & 0 \\ 0 & \frac{h^2}{2} \\ h & 0 \\ 0 & h \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The vehicle is driven by a random applied force $u_t \in \mathbf{R}^2$, which is distributed according to

$$\begin{cases} \mathcal{N}(0, 10I) & \text{with probability } 0.9, \\ \mathcal{N}(0, 100I) & \text{with probability } 0.1, \end{cases}$$

The measurement noise $v_t \in \mathbf{R}^2$ is distributed according to

$$\begin{cases} \mathcal{N}(0, 5I) & \text{with probability } 0.9, \\ \mathcal{N}(0, 500I) & \text{with probability } 0.1. \end{cases}$$

such that 10% of the samples are large outliers. This fits the system model (1.2) with $F = \sqrt{10}B$ and $G = \sqrt{5}I$.

Performance comparison. With the number of iterations fixed at $\tilde{k} = 2$, we selected the parameters to be

$$\lambda_x = 0.10, \quad \lambda_y = 1.8$$

using the grid search procedure described in §1.6. The grid search was carried out over the predicted measurement RMSE (1.16) on a separate simulated trajectory of measurements. Figure 1.1 shows the true vehicle position over time, along with measurements and position estimates produced by the (steady-state) ISKF and KF. Figure 1.2 shows the state estimate errors (absolute values) for the KF and the two-iteration ISKF.

Table 1.1 shows the state estimate RMSE (1.17) evaluated for several filters on the same test trajectory. The ISKF with $\tilde{k} = 1$ achieves comparable performance to WoLF, as both are only designed to reject measurement outliers. The (steady-state) ISKF with $\tilde{k} = 2$ and $\tilde{k} = 3$ achieves better performance than WoLF, and performs comparably to the Huberized KF, at lower computational cost.

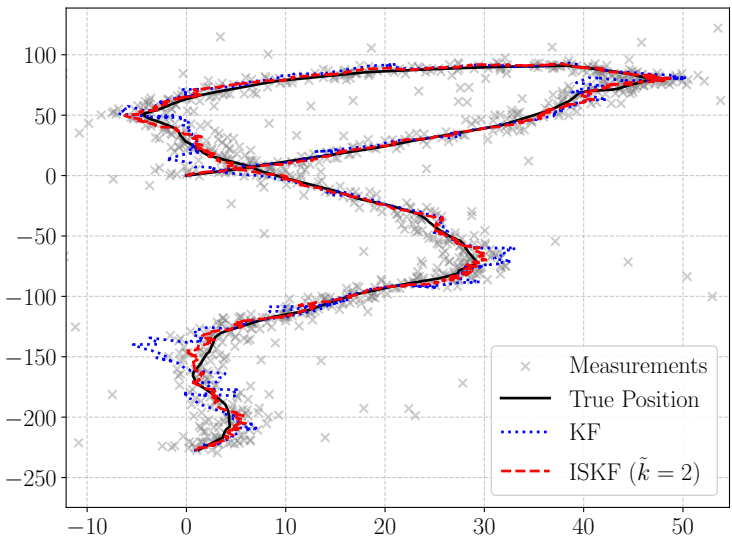


Figure 1.1: True vehicle position over time, along with measurements and position estimates produced by the (steady-state) ISKF and KF.

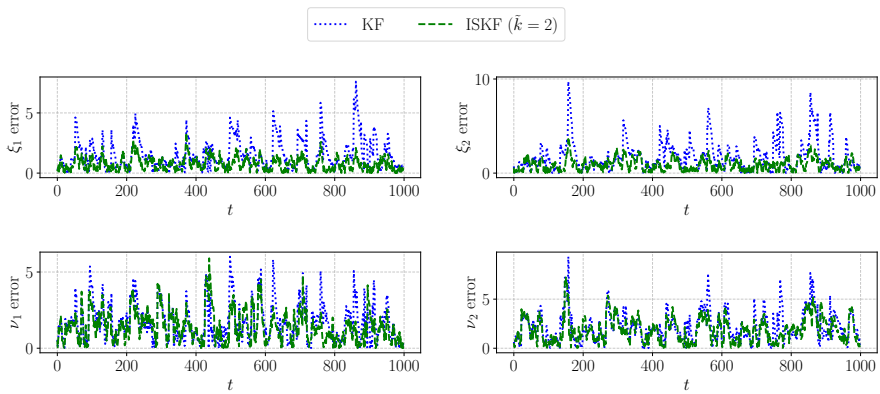
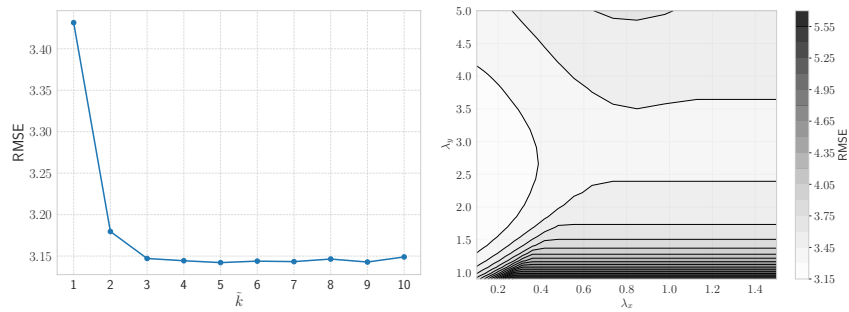


Figure 1.2: Vehicle tracking state estimate errors (absolute values) for the KF and the two-iteration ISKF.



(a) **(b)**

Figure 1.3: Parameter tuning for vehicle tracking example. Left: state estimate RMSE vs. number of iterations $\tilde{k}$. A grid search was performed to choose λ_x and λ_y for each value of $\tilde{k}$. Right: contour plot of state estimate RMSE (1.17) for different values of λ_x and λ_y.

Effect of parameters. Figure 1.3 illustrates the effect of the parameters $\tilde{k}$, λ_x, and λ_y on the performance of the ISKF on the test data. The left plot shows the best-achieved state estimate RMSE (by grid search) for each value of $\tilde{k}$. The ISKF achieves a 30% improvement over the KF after two iterations, and does not improve after three iterations. The contour plot on the right shows the state estimate RMSE for different values of λ_x and λ_y.

1.8.2 Cascaded continuously-stirred tank reactor

CSTR model. The adiabetic continuous stirred-tank reactor (CSTR) is a commonly-appearing system in the chemical process industry [26], [27]. We consider an ideal model of a single, first-order exothermic and irreversible reaction taking place in a reactor tank, which is assumed to be perfectly-mixed. The reagent enters the tank through the inlet at a constant rate,

Method	RMSE	Improvement over KF
KF	4.55	-
ISKF ($\tilde{k}=1$)	3.43	25%
ISKF ($\tilde{k}=2$)	3.19	30%
ISKF ($\tilde{k}=3$)	3.15	31%
WoLF	3.50	23%
Huber KF	3.15	31%

Table 1.1: State estimate RMSE comparison of several filtering methods for the vehicle tracking example. All filters were tuned using the same data, and evaluated on the same test data.

and the output product leaves the reactor at the same constant rate. The state consists of the concentration of the reagent c and the temperature τ in the reactor, and the dynamics are controlled by the inlet concentration c^{in} and the tank's jacket coolant temperature τ^{c}.

The continuous-time dynamics are linearized around an operating point $(c_0, \tau_0, c_0^{\text{in}}, \tau_0^{\text{c}})$. In the linearized model, the state is $\xi = (c - c_0, \tau - \tau_0)$, and the process is driven by $u = (c^{\text{in}} - c_0^{\text{in}}, \tau^{\text{c}} - \tau_0^{\text{c}})$. We observe measurements of the reactor's temperature, but not of the reagent concentration. The discrete-time dynamics with step size of h are

$$\xi_{t+1} = \tilde{A}\xi_t + \tilde{B}u_t, \quad y_t = \tilde{C}\xi_t + \nu_t,$$

where $\nu_t \in \mathbf{R}$ is the measurement noise and the matrices are [28]

$$\begin{aligned}
\tilde{A} &= \begin{bmatrix} 1 - 5h + 4.33h^2 & -0.34h + 0.38h^2 \\ 47.68h - 52.81h^2 & 1 + 2.79h - 4.29h^2 \end{bmatrix}, \\
\tilde{B} &= \begin{bmatrix} h - 2.5h^2 & -0.05h^2 \\ 23.84h^2 & 0.3h + 0.42h^2 \end{bmatrix}, \\
\tilde{C} &= \begin{bmatrix} 0 & 1 \end{bmatrix}.
\end{aligned}$$

Note that here, y_t represents a measurement of the temperature offset from the operating point τ_0, rather than the temperature itself.

System model. In this example, we consider a cascade of three such reactors, with the state of each reactor being the input to the next. Let c_i and τ_i be the reagent concentration and temperature of the i-th reactor, respectively. Then, the cascaded system has state $x \in \mathbf{R}^6$ given by $x = (\xi_1, \xi_2, \xi_3)$, where $\xi_i = (c_i - c_0, \tau_i - \tau_0)$ is the state of the i-th reactor. The cascaded system has discrete-time model

$$x_{t+1} = Ax_t + w_t, \quad y_t = Cx_t + v_t,$$

where

$$A = \begin{bmatrix} \tilde{A} & 0 & 0 \\ \tilde{B} & \tilde{A} & 0 \\ 0 & \tilde{B} & \tilde{A} \end{bmatrix}, \quad C = \begin{bmatrix} \tilde{C} & 0 & 0 \\ 0 & \tilde{C} & 0 \\ 0 & 0 & \tilde{C} \end{bmatrix},$$

and $w \in \mathbf{R}^6$ and $v \in \mathbf{R}^3$ are the process and measurement noises, with F and G matrices given by

$$F = \frac{1}{\sqrt{10}} \begin{bmatrix} \tilde{B} & 0 & 0 \\ 0 & \tilde{B} & 0 \\ 0 & 0 & \tilde{B} \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

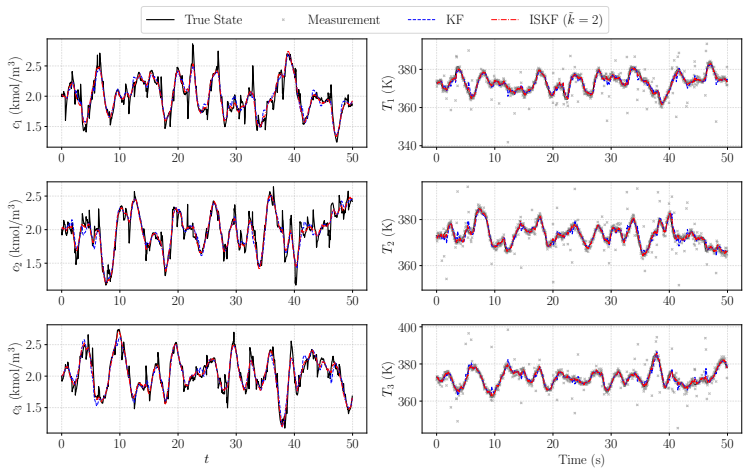


Figure 1.4: True reagent concentration and temperature values over time, along with measurements and state estimates produced by the (steady-state) ISKF and KF.

The process noise w_t and measurement noise v_t are distributed according to

$$w_t \sim \begin{cases} \mathcal{N}(0, FF^T) & \text{with probability } 0.9, \\ \mathcal{N}(0, 100FF^T) & \text{with probability } 0.1, \end{cases}$$

and

$$v_t \sim \begin{cases} \mathcal{N}(0, I) & \text{with probability } 0.9, \\ \mathcal{N}(0, 100I) & \text{with probability } 0.1, \end{cases}.$$

We discretized the continuous-time dynamics with a step size of $h = 50$ms. The operating point is $c_0 = 2\text{kmol/m}^3$, $\tau_0 = 373$K, $c^{\text{in}} = 10\text{kmol/m}^3$, and $\tau^{\text{c}} = 299$K.

Performance comparison. With the number of iterations fixed at $\tilde{k} = 2$, we selected the parameters to be

$$\lambda_x = 0.10, \quad \lambda_y = 3.3$$

using the grid search procedure described in §1.6. Like in the vehicle tracking example, the grid search was carried out over the predicted measurement RMSE (1.16) on a separate simulated trajectory of measurements. Figure 1.4 shows the true reagent concentration and temperature values over time, along with measurements and estimates produced by the (steady-state) ISKF and KF. Figure 1.5 shows the state estimate errors (absolute values) for the KF and the two-iteration ISKF.

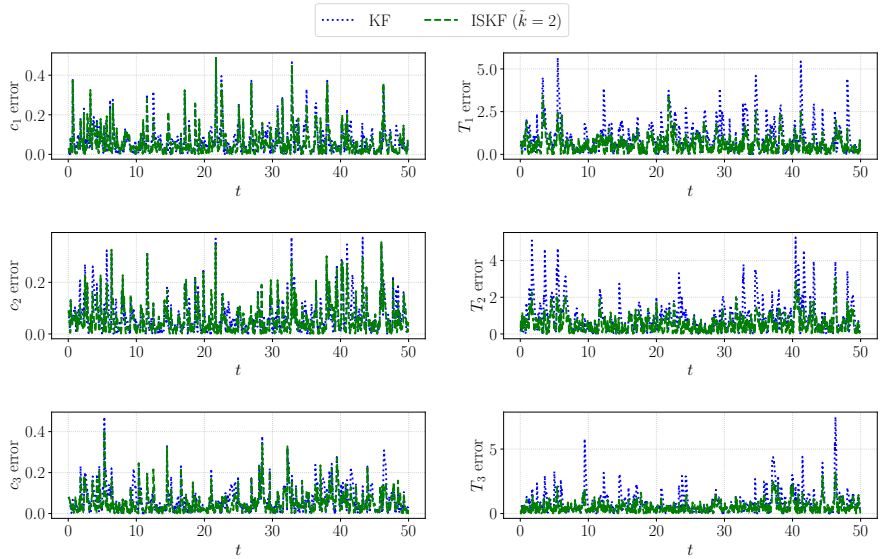


Figure 1.5: State estimate errors (absolute values) for the KF and the two-iteration ISKF.

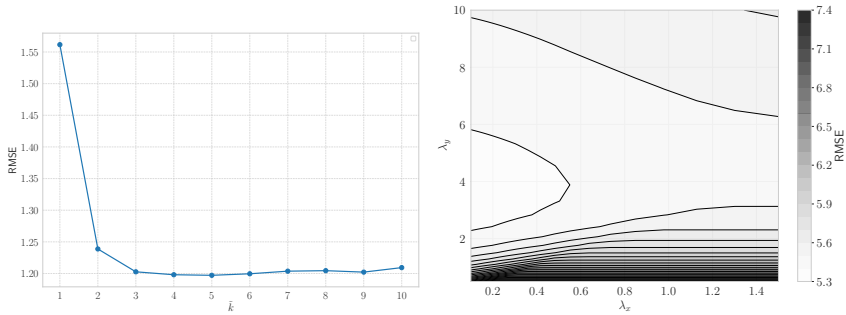
Table 1.2 shows the state estimate RMSE (1.17) evaluated for several filters on the same test trajectory. The ISKF with $\tilde{k} = 1$ achieves comparable performance to WoLF, as both are only designed to reject measurement outliers.

Effect of parameters. Figure 1.6 illustrates the effect of the parameters $\tilde{k}$, λ_x, and λ_y on the performance of the ISKF on the test data. The left plot shows the best-achieved state estimate RMSE (by grid search) for each value of $\tilde{k}$. The ISKF achieves a 49% improvement over the KF after two iterations, and, like in the vehicle tracking example, does not improve after

Table 1.2: State estimate RMSE comparison of several filtering methods for the cascaded CSTR example. All filters were tuned using the same data, and evaluated on the same test data.

Method	RMSE	Improvement over KF
KF	2.46	-
ISKF ($\tilde{k} = 1$)	1.57	36%
ISKF ($\tilde{k} = 2$)	1.25	49%
ISKF ($\tilde{k} = 3$)	1.21	51%
WoLF	1.79	27%
Huber KF	1.56	37%

Figure 1.6: Parameter tuning for CSTR example. Left: state estimate RMSE vs. number of iterations $\tilde{k}$. A grid search was performed to choose λ_x and λ_y for each value of $\tilde{k}$. Right: contour plot of state estimate RMSE (1.17) for different values of λ_x and λ_y.



(a) **(b)**

three iterations. The contour plot on the right shows the state estimate RMSE for different values of λ_x and λ_y.

1.8.3 Tuning step size

As discussed in §1.6, the step size η may be tuned jointly with λ_x and λ_y as part of the same grid search procedure. For both the vehicle tracking and CSTR examples, we found that while increasing η can provide marginal

Table 1.3: Results of tuning the step size η jointly with λ_x and λ_y. RMSE is evaluated on the same test data as in §?? and §??. The RMSE with no outliers is computed by removing the outliers from the test data, so that the process and measurement noises are Gaussian with fixed covariance.

Method	η	λ_x	λ_y	RMSE	RMSE (no outliers)
KF	-	-	-	4.55	1.40
ISKF ($\tilde{k} = 2$)	1	0.10	1.83	3.19	1.50
ISKF ($\tilde{k} = 2$)	2.64	0.10	0.89	3.15	1.67

(a) Vehicle tracking example

Method	η	λ_x	λ_y	RMSE	RMSE (no outliers)
KF	-	-	-	2.46	0.68
ISKF ($\tilde{k} = 2$)	1	0.10	3.80	1.27	0.78
ISKF ($\tilde{k} = 2$)	1.83	0.26	2.34	1.23	0.92

(b) CSTR example

improvements in performance, the resulting filter underperforms when there are no outliers in the simulation, *i.e.*, the process noise and measurement noises are Gaussian with fixed covariance. In the following, we consider a grid search over 20 values of η between 0.1 and 100, logarithmically-spaced. The results are shown in Table 1.3. In the vehicle tracking example, the (two-step) ISKF with a tuned value of $\eta = 2.64$ achieves a 1% improvement over the ISKF with $\eta = 1$ on the test data, but underperforms by 10% when the outliers are removed from the test data. In the CSTR example, the ISKF with tuned value $\eta = 1.83$ achieves a 3% improvement over the ISKF with $\eta = 1$ on the test data, but underperforms by 15% when the outliers are removed from the test data.

1.9 Conclusion

We have introduced the iteratively saturated Kalman filter, a modification of the standard KF's update step that makes it robust to outliers. The method is derived as a scaled gradient method for solving a particular convex maximum a posteriori estimation problem. The steady-state variant of the ISKF matches the computational efficiency of the steady-state KF, and is well-suited for real-time applications.

References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of Basic Engineering*, vol. 82, no. 1, 1960, pp. 35–45.

[2] W. Q. Malik, W. Truccolo, E. N. Brown, and L. R. Hochberg, "Efficient decoding with steady-state Kalman filter in neural interface systems", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 1, 2010, pp. 25–34.

[3] F. Smets and R. Wouters, "Shocks and frictions in us business cycles: A Bayesian DSGE approach", *American economic review*, vol. 97, no. 3, 2007, pp. 586–606.

[4] J. Huber, "An augmented steady-state Kalman filter to evaluate the likelihood of linear and time: Invariant state-space models", Volkswirtschaftliche Diskussionsreihe, Tech. Rep., 2022.

[5] C. Masreliez and R. Martin, "Robust Bayesian estimation for the linear model and robustifying the Kalman filter", *IEEE transactions on Automatic Control*, vol. 22, no. 3, 1977, pp. 361–371.

[6] W. C. Davidon, "Variable metric method for minimization", Tech. Rep., May 1959. DOI: 10.2172/4252678. URL: https://www.osti.gov/biblio/4252678.

[7] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization", *The Computer Journal*, vol. 6, no. 2, Aug. 1963, pp. 163–168. DOI: 10.1093/comjnl/6.2.163. eprint: https://academic.oup.com/comjnl/article-pdf/6/2/163/1041527/6-2-163.pdf. URL: https://doi.org/10.1093/comjnl/6.2.163.

[8] Ž. Đurović and B. Kovačević, "Robust estimation with unknown noise statistics", *IEEE Transactions on Automatic Control*, vol. 44, no. 6, 1999, pp. 1292–1296.

[9] G. Chang, "Robust Kalman filtering based on mahalanobis distance as outlier judging criterion", *Journal of Geodesy*, vol. 88, no. 4, 2014, pp. 391–401.

[10] G. Duran-Martin, M. Altamirano, A. Y. Shestopaloff, L. Sánchez-Betancourt, J. Knoblauch, M. Jones, F.-X. Briol, and K. Murphy, "Outlier-robust Kalman filtering through generalised bayes", *arXiv preprint arXiv:2405.05646*, 2024.

[11] J.-A. Ting, E. Theodorou, and S. Schaal, "A Kalman filter for robust outlier detection", in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 1514–1519, 2007.

[12] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "An outlier-robust Kalman filter", in *2011 IEEE international conference on robotics and automation*, IEEE, pp. 1551–1558, 2011.

[13] P. J. Huber, "Robust Estimation of a Location Parameter", *The Annals of Mathematical Statistics*, vol. 35, no. 1, 1964, pp. 73–101. DOI: 10.1214/aoms/1177703732. URL: https://doi.org/10.1214/aoms/1177703732.

[14] T. Cipra and R. Romera, "Robust Kalman filter and its application in time series analysis", *Kybernetika*, vol. 27, no. 6, 1991, pp. 481–494.

[15] B. Kovačević, Ž. Đurović, and S. Glavaški, "On robust Kalman filtering", *International Journal of Control*, vol. 56, no. 3, 1992, pp. 547–562.

[16] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization", *Optimization and Engineering*, vol. 13, 2012, pp. 1–27.

[17] H. Cantzler, "Random sample consensus (RANSAC)", *Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh*, vol. 3, 1981.

[18] A. Vedaldi, H. Jin, P. Favaro, and S. Soatto, "KALMANSAC: Robust filtering by consensus", in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, IEEE, vol. 1, pp. 633–640, 2005.

[19] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate student-t distribution", in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, IEEE, pp. 1–6, 2012.

[20] C. D. Karlgaard, "Nonlinear regression Huber–Kalman filtering and fixed-interval smoothing", *Journal of guidance, control, and dynamics*, vol. 38, no. 2, 2015, pp. 322–330.

[21] H. Fang, M. A. Haile, and Y. Wang, "Robustifying the Kalman filter against measurement outliers: An innovation saturation mechanism", in *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, pp. 6390–6395, 2018.

[22] A. Boustati, O. D. Akyildiz, T. Damoulas, and A. Johansen, "Generalised Bayesian filtering via sequential Monte Carlo", *Advances in neural information processing systems*, vol. 33, 2020, pp. 418–429.

[23] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[24] B. Polyak, *Introduction to Optimization*. Optimization Software, 1987.

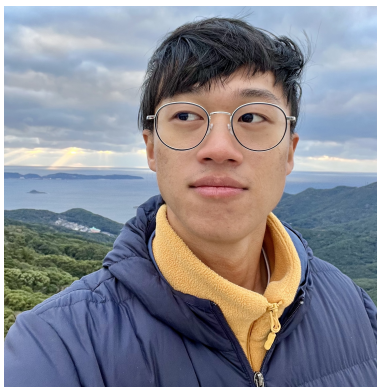
[25] P. J. Goulart and Y. Chen, *Clarabel: An interior-point solver for conic programs with quadratic objectives*, 2024. arXiv: 2405.12762 [math.OC].

[26] B. W. Bequette, *Process dynamics*. Prentice Hall Englewood Cliffs, NJ, 1998.

[27] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III, *Process dynamics and control*. John Wiley & Sons, 2016.

[28] MathWorks, Inc., *CSTR Model*, https://www.mathworks.com/help/mpc/gs/cstr-model.html, 2025.

About the Authors



Alan Yang is a Ph.D candidate in Electrical Engineering at Stanford University. He received the B.S. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign in 2018. His research interests include convex optimization and machine learning, control systems, and signal processing.

Stephen Boyd is the Samsung Professor of Engineering, and Professor of Electrical Engineering at Stanford University. He received the A.B. degree in Mathematics from Harvard University in 1980, and the Ph.D. in Electrical Engineering and Computer Science from the University of California, Berkeley, in 1985, before joining the faculty at Stanford. His current research focus is on convex optimization applications in control, signal processing, machine learning, and finance. He is a member of the US National Academy of Engineering, a foreign member of the Chinese Academy of Engineering, and a foreign member of the National Academy of Korea.