A Variant of The Unscented Kalman Filter for Highly Non-Linear State Estimation

Russell Schwartz
Carnegie Mellon University
rschwar2@andrew.cmu.edu

Ian Krause
Carnegie Mellon University
ikrause@andrew.cmu.edu

***Abstract*—State estimation for non-linear systems is challenging, even when we restrict our belief model to a Gaussian distribution, as in the case of the Kalman Filter. The EKF and UKF both use approximation schemes to make the problem tractable. The UKF often achieves better performance over the EKF. In part, it does this by consulting the dynamics model on a larger portion of the state space through its selection of so-called sigma-points. In this work we explore a novel variant of the usual sigma-point selection scheme that selects more than the usual $2n + 1$ points from more than one distance about the mean. These modifications help to resolve some of the error experienced by the original UKF (especially in systems with strong 3rd order behavior) at the cost of modestly increased computational cost. We compare this method to baseline EKF and UKF implementations in a number of simulated environments, including the tracking of a double-inverted-pendulum and a SLAM task. Our proposed "UKF2" exhibits increased state-estimation accuracy and a lower rate of divergence over both the EKF and UKF in all of these systems.**

I. Introduction

In the fields of robotic control, target tracking, autonomous driving, robotics at large, and beyond, accurate state estimation is a fundamental first step towards achieving correct behavior of the overall system. The Bayesian probabilistic perspective of gives rise to the Bayes Filter algorithm, which maintains a probability distribution over the state space [9]. This distribution, called the "belief", can be updated by the filter upon execution of an action of receipt of a measurement, accounting for noise in these processes. This paradigm allows for a state estimator to make fuller use of the information available, and the availability of uncertainty information enables greater robustness in downstream systems.

Out of the many different implementations of the Bayes Filter (the Histogram Filter, Particle Filter, Kalman Filter, ...) few are more popular than the Kalman Filter (KF), which has the important property of being able to operate efficiently even in high dimensional spaces [1]. The KF achieves this performance by making two concessions. First, it restricts the space of possible belief distributions to multivariate Gaussian distributions, which are analytically manageable, but cannot represent multi-modality. Second, it is restricted to purely linear systems which have Gaussian noise models. This restriction leads to a compact analytic solution to the optimal Bayes Filter equations. However, many systems encountered in practice are non-linear. Luckily, methods exist which extend the KF algorithm to these non-linear systems, at the cost of optimality.

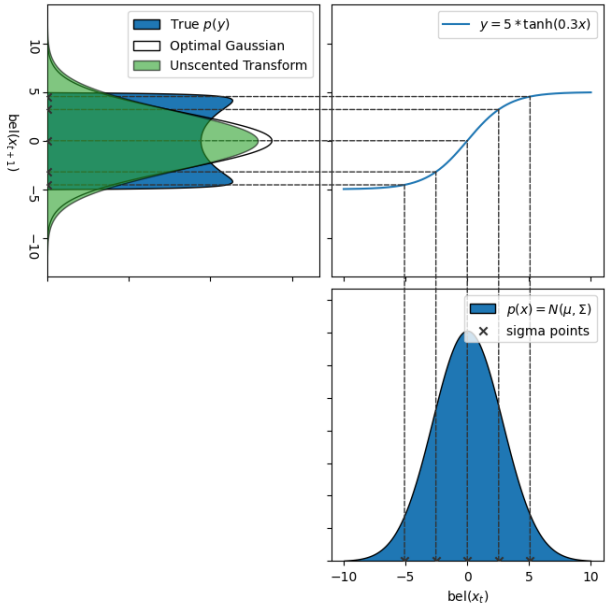


Fig. 1: Visualization of the proposed modified unscented transform. Sigma-points are selected from $k = 2$ symmetric shells about the mean $(\pm 2.5, \pm 5)$. These points are then transformed by the non-linear dynamics, and a Gaussian is fit to their image. The result is a close approximation to the optimal Gaussian. The additional outer sigma-points help capture the otherwise-unobserved 3rd-order behavior.

The Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are the two most-popular such methods, and we describe them in details in Section III. The EKF, which operates on the principle of Taylor approximation, is generally less accurate then the UKF, which samples the state-space at a small number of points. Intuitively, this performance increase is in part because the UKF samples more of the state space and thus consults more of the dynamics model [1]. In a sense, the UKF is an interpolation between single-sample methods (like the EKF) and many-sample methods (like the Histogram Filter).

In this work we introduce a novel variant of the UKF algorithm that samples additional sigma-points (beyond the

conventional $2n + 1$) from multiple distinct "shells" about the mean before performing the Unscented Transform. For highly non-linear systems, this modification enables the filter to better track higher-order behavior of the dynamics, while being less sensitive to regime changes. We demonstrate the superior performance of this variant, dubbed UKF2, on a number of simulated dynamic systems including a 2D SLAM problem and an inverted double-pendulum tracking problem.

II. Related Work

Rudolf Kálmán originally solved the Bayes Filter optimal update rules for a Gaussian parametrization in [6], branching an entire field of research centered around efficient high-dimensional state estimation [1].

McGee and and Schmidt, members of the NASA Ames Research Center, developed the Extended Kalman Filter (EKF) for use in the Apollo program [8]. Their method addresses state estimation for nonlinear series using first-order Taylor-series expansions. See III-C.

Julier and Uhlmann later introduced the Unscented Kalman Filter (UKF) in [11]. Much like the Extended Kalman filter, their method addresses nonlinear systems but instead uses the sampling-based Unscented Transform to propagate uncertainty. They elaborate on their method in [5], in which they discuss a general sigma-point selection framework. However, they focus on minimal sigma-point sets, which contain the fewest number of points in order to uniquely define the desired Gaussian. See III-D.

A number of other non-linear extensions have been proposed, including the the Second-Order EKF (EKF2) and Polynomial EKF (PEKF) [3], [10]. Both of these methods either directly compute higher order information (in the form of an explicit Hessian) or approximate these terms via forward-difference methods computed at sigma-points. The resulting methods perform only marginally better than the UKF, are complex to implement, and scale poorly in higher dimension.

III. Background

A. Bayes Filter

The Bayes Filter is a general probabilistic framework for state estimation in which the current belief of the state is represented by a probability distribution over the state space. This belief is updated upon the execution of an action or the receipt of a measurement. These computations are often called the Predict Step and the Update Step respectively. The belief can be represented non-parametrically, as in the Histogram Filter or Particle Filter, or parametrically, as in the Kalman Filter. We use the following notation to describe a general time-invariant non-deterministic dynamic system model.

State Vector:	x
Control Input:	u
Measurement:	z
Dynamics Noise:	w
Measurement Noise:	v
Dynamics Model:	$f(x, u, w)$
Measurement Model:	$h(x, v)$

A system definition consists of a definition for f, a definition for h, and probability distributions for w and v. Note that all vector quantities may have arbitrary dimensionality. We also require our system to have the property that measurements are independent of one another when conditioned on the state. This is called the Markov property.

Writing our belief as $Bel(x_t)$, the general Bayes filter Predict and Update are as follows. A full derivation is available in Appendix VII-A. The literature often presents these steps as a single computation computed sequentially, but in reality they may be decoupled, and can indeed be invoked an any order independent of one another to accommodate a mixed stream of actions and measurements. The computations is summarized in pseudocode as Algorithms 1 and 2:

Algorithm 1 Bayes Filter Predict Step

Input: state belief $Bel(x)$, control input u
Output: updated state belief $Bel'(x)$

for x' in the state space **do**
    $Bel(x') \leftarrow \int p(x' \mid u, x) Bel(x) dx$
return $Bel'(x)$

Algorithm 2 Bayes Filter Update Step

Input: state belief $Bel(x)$, measurement z
Output: updated state belief $Bel'(x)$

for x in the state space **do**
    $Bel'(x) \leftarrow p(z \mid x) Bel(x)$
$\eta \leftarrow 1 / \int Bel'(x) dx$ `# normalization`
$Bel'(x) \leftarrow \eta \cdot Bel'(x)$
return $Bel'(x)$

The KF, EKF, and UKF represent the state belief parametrically as a single unimodal Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. They also restrict to systems with Gaussian noise models. The properties of Gaussians allow us to write the Predict and Update steps explicitly in analytic form.

B. Kalman Filter

The Kalman Filter is applicable to linear systems of the form:

$$\begin{aligned} \text{Dynamics:} \quad & f(x,u,w) = Ax + Bu + D + Lw \\ \text{Measurement:} \quad & h(x,v) = Cx + E + Mv \\ \text{Dynamics Noise:} \quad & w \sim \mathcal{N}(0,R) \\ \text{Measurement Noise:} \quad & v \sim \mathcal{N}(0,Q) \end{aligned} \tag{1}$$

The Bayesian Update and Predict steps are computed as follows. The full derivation can be found in [4].

Algorithm 3 Kalman Filter Predict Step

Input: state estimate (μ, Σ), control input u
Output: updated state estimate (μ', Σ')

$\mu' \leftarrow A\mu + Bu + D$
$\Sigma' \leftarrow A\Sigma A^\top + LRL^\top$
return (μ', Σ')

Algorithm 4 Kalman Filter Update Step

Input: state estimate (μ, Σ), measurement z
Output: updated state estimate (μ', Σ')

$S_{xz} \leftarrow \Sigma C^\top$
$S_{zz} \leftarrow C\Sigma C^\top + MQM^\top$
$K \leftarrow S_{xz} S_{zz}^{-1}$ `# Kalman gain`
$\hat{z} \leftarrow C\mu + E$
$\mu' \leftarrow \mu + K(z - \hat{z})$
$\Sigma' \leftarrow \Sigma - KC\Sigma$
return (μ', Σ')

For linear systems with Gaussian state belief, the KF is optimal. It is also very efficient, even in higher dimensions. However, in the real world purely linear systems are few and far between. The EKF and UKF provide near-optimal extensions of the KF for non-linear systems.

C. Extended Kalman Filter

The EKF extends the KF to non-linear systems by computing a linear approximation of the system at each timestep, and then applying the KF update rules. This approximation takes the form of a first-order Taylor expansion:

$$\begin{aligned} f(x,u,w) \approx f(x_0,u_0,w_0) + F_x(x - x_0) + F_u(u - u_0) \\ + F_w(w - w_0) \end{aligned}$$

where

$$F_x = \frac{\partial f}{\partial x} \quad \text{and} \quad F_u = \frac{\partial f}{\partial u} \quad \text{and} \quad F_w = \frac{\partial f}{\partial w}$$

evaluated at a linearization point (x_0, u_0, w_0) chosen to be the current state mean $x_0 = \mu_t$, the given control $u_0 = u_t$, and the mean noise value $w_0 = \mathbb{E}[w]$. Similarly, we approximate the measurement model:

$$h(x,v) \approx h(x_0,v_0) + H_x(x - x_0) + H_v(v - v_0)$$

where

$$H_x = \frac{\partial h}{\partial x} \quad \text{and} \quad H_v = \frac{\partial h}{\partial v}$$

The full procedure is as follows. To make the relationship to the Kalman Filter clear, we use the same coefficient naming scheme for the linearized system as in (1).

Algorithm 5 Extended Kalman Filter Predict Step

Input: state estimate (μ, Σ), control input u
Output: updated state estimate (μ', Σ')

`# Linear Approximation`
$x_0 \leftarrow \mu, u_0 \leftarrow u, w_0 \leftarrow \mathbb{E}[w]$
$A \leftarrow \frac{\partial f}{\partial x}, B \leftarrow \frac{\partial f}{\partial u}, L \leftarrow \frac{\partial f}{\partial w}$
`# KF Predict Step`
$\mu' \leftarrow f(x_0, u_0, w_0)$
$\Sigma' \leftarrow A\Sigma A^\top + LRL^\top$
return (μ', Σ')

Algorithm 6 Extended Kalman Filter Update Step

Input: state estimate (μ, Σ), measurement z
Output: updated state estimate (μ', Σ')

`# Linear Approximation`
$x_0 \leftarrow \mu, v_0 \leftarrow \mathbb{E}[v]$
$C \leftarrow \frac{\partial h}{\partial x}, M \leftarrow \frac{\partial h}{\partial v}$
`# KF Update Step`
$S_{zz} \leftarrow C\Sigma C^\top + MQM^\top$
$S_{xz} \leftarrow \Sigma C^\top$
$K \leftarrow S_{xz} S_{zz}^{-1}$ `# Kalman gain`
$\hat{z} \leftarrow h(x_0)$
$\mu' \leftarrow \mu + K(z - \hat{z})$
$\Sigma' \leftarrow \Sigma - KC\Sigma$
return (μ', Σ')

The EKF algorithm is very popular, in part for its efficiency. Each iteration requires only 1 query to the dynamics model, and 1 call to the jacobians. In practice, the required jacobians may be computed by evaluating a user-provided symbolical derivative, or by auto-differentiation of the system model. This method of re-linearizing at every timestep can be very accurate, especially when the filter is run at a high rate or when uncertainty is low relative to the C^2-smoothness of the dynamics. However, for highly non-linear systems, greater accuracy can be obtained by using a higher-order approximation scheme, such as the UKF.

D. Unscented Kalman Filter

The UKF works by sampling the state space at a set $\mathcal{X}$ of so-called "sigma-points" which are selected near the mean of the state estimate. These points are then passed through the system dynamics, and a Gaussian is fitted to the transformed points. These sigma-points each have an associated weight w that scales their contribution to the final distribution. We require that these weights sum to 1, however it is actually possible for some of them to be negative. The conventional implementation of the Unscented Transform selects these points to include:

- 1 point at the mean of $Bel(x_t)$:

$$\mathcal{X}_0 = \mu_t \quad \text{with} \quad w_0 = 1 - \frac{1}{\alpha^2}$$

- $2n$ points distributed uniformly on a level surface of $Bel(x_t)$:

$$\mathcal{X}_i = \mu_t \pm \alpha(\sqrt{n\Sigma_t})_i \quad \text{with} \quad w_i = \frac{1}{2n\alpha^2}$$

where n is the dimension, $\sqrt{M}$ is a positive semi-definite matrix square root, and $(M)_i$ is the ith column of M. We then propagate the points through the non-linear function of interest $\mathcal{Y} = f(\mathcal{X})$ and recover the mean and covariance:

$$\mu_y = \sum_i w_i \mathcal{Y}_i$$
$$\Sigma_{yy} = \sum_i \tilde{w}_i (\mathcal{Y}_i - \mu_y)(\mathcal{Y}_i - \mu_y)^\top$$

where $\tilde{w}_i = w_i$ for all points except for the mean, which has its weight adjusted to better capture higher-order behavior of the system:

$$\tilde{w}_0 = w_0 + (1 - \alpha^2 + \beta)$$

For Gaussians, the optimal choice of β is 2 [5]. When performing the Update Step, we will also need to compute the cross-covariance:

$$\Sigma_{xy} = \sum_i \tilde{w}_i (\mathcal{X}_i - \mu_t)(\mathcal{Y}_i - \mu_y)^\top$$

When the noise model is non-additive, we perform the unscented transform on the joint spaces $\begin{bmatrix} x \mid w \end{bmatrix}$ and $\begin{bmatrix} x \mid v \end{bmatrix}$ formed as the product of the state space and respective noise spaces. This has no affect on μ_y or Σ_{yy}, but introduces some unwanted elements into Σ_{xy}. Luckily, the noise can be marginalized out by simply extracting the appropriate block. This entire process is summarized in Algorithms 7 and 8.

Algorithm 7 Unscented Kalman Filter Predict Step

Input: state estimate (μ, Σ), control input u
Output: updated state estimate (μ', Σ')

```
# Form Joint State/Noise Space
```
$\mu_{[x|w]} \leftarrow \begin{bmatrix} \mu \\ \mathbb{E}[w] \end{bmatrix}$
$\Sigma_{[x|w]} \leftarrow \begin{bmatrix} \Sigma & 0 \\ 0 & \text{var}[w] \end{bmatrix}$
```
# Unscented Transform
```
$\mathcal{X}, w \leftarrow \text{select\_sigma\_points}(\mu_{[x|w]}, \Sigma_{[x|w]})$
$\mathcal{X}' \leftarrow f(\mathcal{X}, u)$
$\mu' \leftarrow \sum_i w_i \mathcal{X}'_i$
$\Sigma' \leftarrow \sum_i \tilde{w}_i (\mathcal{X}'_i - \mu_y)(\mathcal{X}'_i - \mu_y)^\top$
return (μ', Σ')

Algorithm 8 Unscented Kalman Filter Update Step

Input: state estimate (μ, Σ), measurement z
Output: updated state estimate (μ', Σ')

```
# Form Joint State/Noise Space
```
$\mu_{xv} \leftarrow \begin{bmatrix} \mu \\ \mathbb{E}[v] \end{bmatrix}$
$\Sigma_{xv} \leftarrow \begin{bmatrix} \Sigma & 0 \\ 0 & \text{var}[v] \end{bmatrix}$
```
# Unscented Transform
```
$\mathcal{X}, w \leftarrow \text{select\_sigma\_points}(\mu, \Sigma)$
$\mathcal{Z} \leftarrow h(\mathcal{X}, u)$
$\hat{z} \leftarrow \sum_i w_i \mathcal{Z}_i$
$S_{zz} \leftarrow \sum_i \tilde{w}_i (\mathcal{Z}_i - \mu_y)(\mathcal{Z}_i - \mu_y)^\top$
$S_{xz} \leftarrow \sum_i \tilde{w}_i (\mathcal{X}_i - \mu)(\mathcal{Z}_i - \mu_y)^\top$
```
# KF Update Step
```
$K \leftarrow S_{xz} S_{zz}^{-1}$ `# Kalman gain`
$\mu' \leftarrow \mu + K(z - \hat{z})$
$\Sigma' \leftarrow \Sigma - KC\Sigma$
return (μ', Σ')

The UKF has some distinct advantages over the EKF. It requires no differentiation, and it is able to better capture 2nd and 3rd order behavior of the system. However, it is less numerically stable due to the required matrix square root, and it is more computationally expensive, requiring $|\mathcal{X}| = 2n + 1$ queries to the dynamics function. For these reasons, combined with the increased complexity to implement, the UKF is not nearly as popular as the EKF in practice [1].

IV. Methodology

We present our proposed modification to the UKF, comment on implementation details, explain our testing simulation framework, and then introduce the dynamic systems that will be used to evaluate filter performance.

A. Modified Unscented Kalman Filter

Our primary modification to the UKF consists of sampling additional sigma-points by taking multiple shells of points

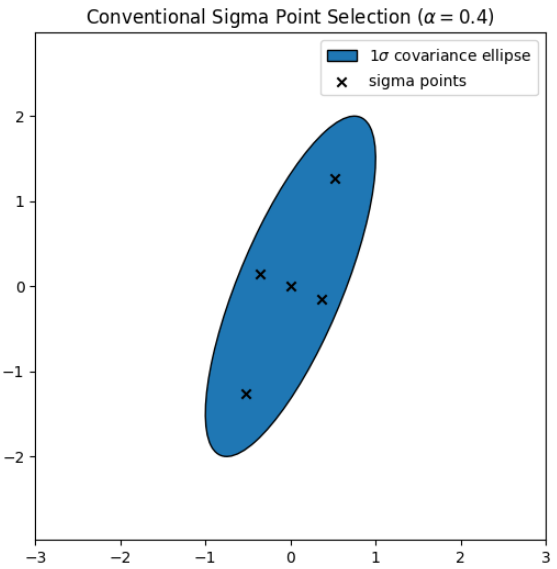


Fig. 2: Example of conventional scaled sigma point selection in 2D ($\alpha = 0.4$)

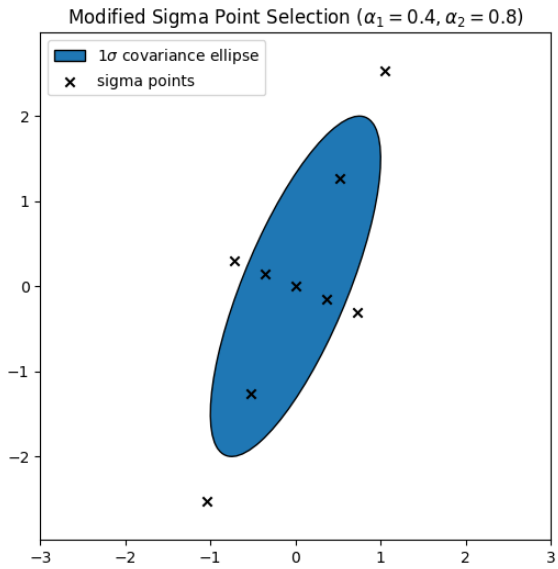


Fig. 3: Example of modified scaled sigma point selection in 2D ($\alpha_1 = 0.4, \alpha_2 = 0.8$)

with different scaling factors $\alpha_1, \cdots, \alpha_k$. The weights are selected in a similar manner to the conventional UKF, but are then normalized across the shells. This is consistent with the general weight assignment scheme introduced in [5]. The modified points and weights consist of:

- 1 point at the mean of $Bel(x_t)$:

$$\mathcal{X}_0 = \mu_t \quad \text{with} \quad w_0 = 1 - \frac{1}{k} \cdot \sum_{i=1}^{k} \frac{1}{\alpha_i^2}$$

- $2n \times k$ points distributed uniformly on k distinct level surfaces of $Bel(x_t)$:

$$\mathcal{X}_{1,i} = \mu_t \pm \alpha_1 (\sqrt{n\Sigma_t})_i \quad \text{with} \quad w_{1,i} = \frac{1}{k} \cdot \frac{1}{2n\alpha_1^2}$$
$$\vdots$$
$$\mathcal{X}_{k,i} = \mu_t \pm \alpha_k (\sqrt{n\Sigma_t})_i \quad \text{with} \quad w_{k,i} = \frac{1}{k} \cdot \frac{1}{2n\alpha_k^2}$$

The higher-order weight adjustment for the center point is also similar to the conventional UKF:

$$\tilde{w}_0 = w_0 + \frac{1}{k} \cdot \sum_{i=1}^{k} (1 - \alpha^2) + \beta$$

After the sigma-points and weights are selected, the filtering algorithm proceeds exactly as in the conventional UKF. We refer to this modified filter algorithm as UKF2. With these additional sigma-points, we are theoretically better able to capture the higher-order behavior of the system. This is demonstrated visually in 4.

We choose $\alpha = 0.2, 0.4, 0.8$ in many of our experiments, but in practice should be tuned to the system at hand. Generally speaking, a value of $k = 2$ or $k = 3$ is the point of diminishing return.

B. System Models Framework

We define a family of Python classes for representing (very generically) dynamic systems which may be candidates for filtering. The base `SystemModel` class is a high-level class representing a dynamic system, including a dynamics model, measurement model, and corresponding noise models. The constructor requires the dimensions of the following spaces:

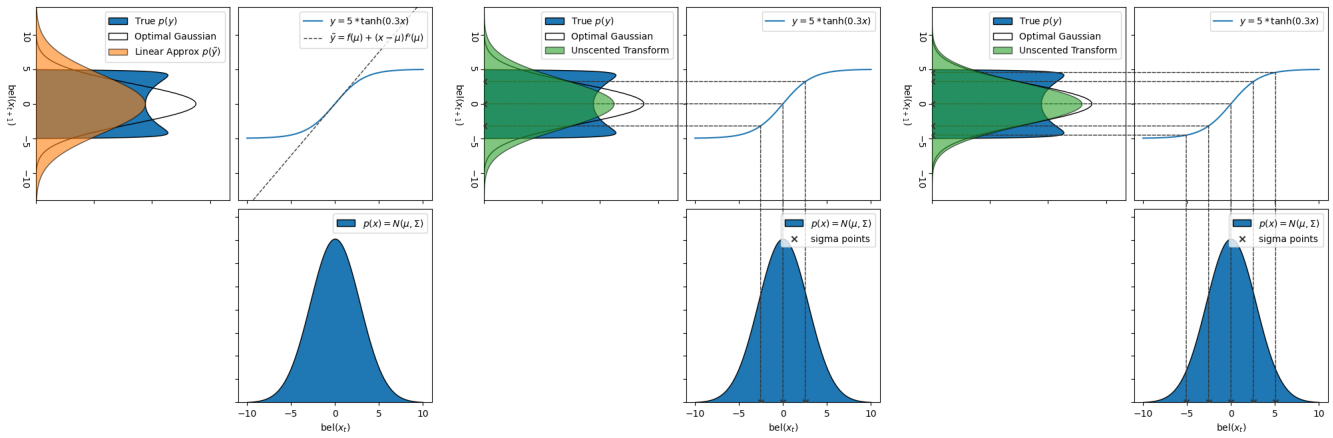
- `state_dim`: dimension of the state space (x)
- `control_dim`: dimension of the control input space (u)
- `measurement_dim`: dimension of the measurement space (z)
- `dynamics_noise_dim`: dimension of the dynamics noise vector space (w)
- `measurement_noise_dim`: dimension of the measurement noise vector space (v)

in addition to four callable functions:

- `dynamics_func`: $(x, u, w) \to x$
- `measurement_func`: $(x, v) \to z$
- `dynamics_noise_func`: $() \to w$
- `measurement_noise_func`: $() \to v$

which together define the system and its noises. In addition, we define four subclasses

- `GaussianSystemModel(SystemModel)`: both process noise and observation noise are (potentially non-additive) zero-mean gaussians
- `DifferentiableSystemModel(SystemModel)`: differentiable dynamics and measurement models (requires explicit jacobians)
- `AutoDiffSystemModel(GaussianSystemModel, DifferentiableSystemModel)`: automatically differentiable dynamics and measurement models (amenable to EKF, no need to provide explicit jacobians)



(a) Linear Approximation (EKF) (b) Unscented Transform (UKF) (c) Modified Unscented Transform (UKF2)

Fig. 4: Visualization of different methods for approximating non-linear uncertainty propagation: linear approximation, unscented transform, and modified unscented transform. The modified unscented transform produces a distribution which most closely matches the optimal Gaussian.

- `LinearSystemModel(GaussianSystemModel, DifferentiableSystemModel)`: linear dynamics, linear measurement model, and additive gaussian noises (amenable to KF)

Automatic differentiation is accomplished by the `JAX` library.

C. Filter Implementations

We implement the Kalman Filter, Extended Kalman Filter, and Unscented Kalman Filter as baselines for comparison. These implementations are built to operate on the generic system models defined above, and are effectively very modular. They each implement a `predict_step(x, u)` and `update_step(z)` method for filtering control inputs and measurements respectively. When instantiating a filter, the user specifies the system model that will be operated on.

The `ExtendedKalmanFilter` implementation requires a system which is both a `GaussianSystemModel` and a `DifferentiableSystemModel` as its target system. This allows it to call to the system's jacobian getter methods, which may be automatic.

The `UnscentedKalmanFilter` implementation is a bit more general as it only requires a `GaussianSystemModel`. The user has the opportunity to specify a custom `SigmaPointSelector`, whose job it is to select sigma-points given a prior mean and covariance. The proposed UKF2 sigma-point selection is implemented as `MultiShellSigmaPointSelector`, and supports both cholesky- and eigen-decomposition based matrix-square-root algorithms.

D. Evaluation Systems

To verify and evaluate our filter implementations, we define some example systems with varying levels of complexity and non-linearity.

- 1D Kinematic Car: a classic linear system (the double-integrator) describing the motion of a car in one dimensional space under controlled acceleration. We also implement a nonlinear variant that adds a drag term with a quadratic dependence on velocity.
- 2D SLAM: a simultaneous localization and mapping problem. This system estimates the position and orientation of a robot while simultaneously estimating a number of landmark locations via (non-linear) bearing and range measurements.
- Mackey-Glass Sequence: a sequence autoregression problem on the Mackey-Glass system, a set of differential equations that exhibit chaotic behavior. This system is known for being particularly challenging to filter [7].
- Double Inverted Pendulum: a double pendulum affixed to a movable cart, mounted on a linear rail. A strict subset of the joint positions are observed. This system is chaotic and exhibits regime changes.

V. Results

A. Kinematic Car

The 1D Kinematic car model is a classic linear system (the double-integrator) describing the motion of a car in one dimensional space under controlled acceleration. The (idealized, noiseless) system is defined:

$$\text{state: } x = \begin{bmatrix}\text{pos} & | & \text{vel}\end{bmatrix} \quad \text{control: } u = \begin{bmatrix}\text{acc}\end{bmatrix}$$

$$A = \begin{bmatrix}1 & \Delta t \\ 0 & 1\end{bmatrix} \quad B = \begin{bmatrix}(\Delta t)^2/2 \\ \Delta t\end{bmatrix} \quad C = \begin{bmatrix}1 & 0\end{bmatrix}$$

$$f(x, u, w) = Ax + Bu + w$$

$$h(x, v) = Cx + v$$

We also define a non-linear system that adds a drag term with a quadratic dependence on velocity.

$$\tilde{f}(x, u, w) = Ax + Bu + \begin{bmatrix}0 \\ -\text{sgn}(x_2)(\nu_1|x_2| + \nu_2|x_2|^2)\end{bmatrix} + w$$

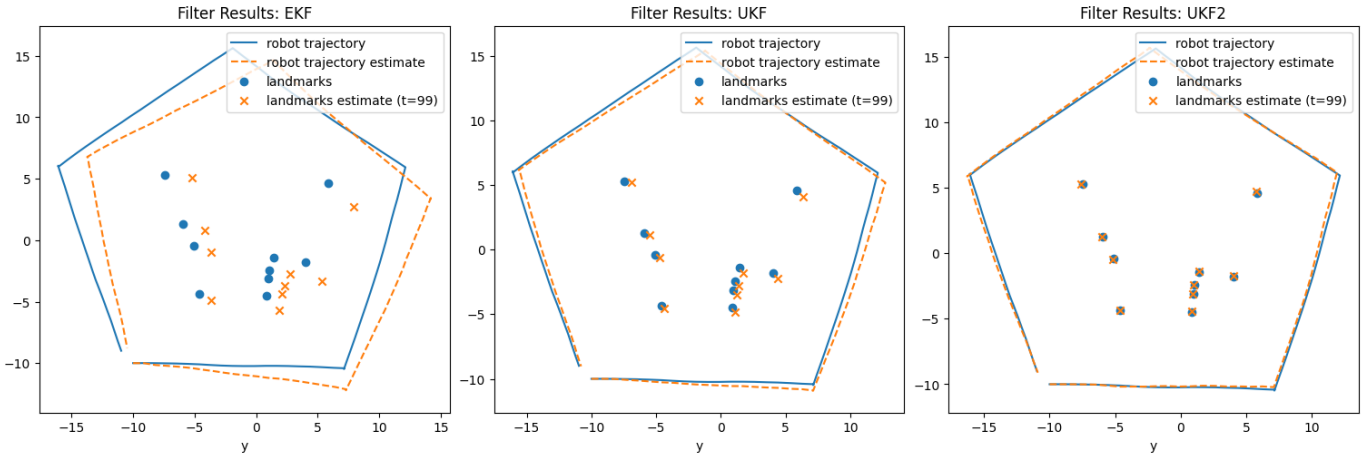


Fig. 5: Filter results on a representative run of the 2D nonlinear SLAM system. The trajectory and map produced by UKF2 is visually better than the baselines. This is confirmed by the numerical errors. See Appendix VII-D for both raw L2 and Mahalanobis errors.

These systems mainly verify that the filter implementations are correct. The UKF and UKF2 both slightly out-perform the EKF. Indeed, the (unmodified) UKF is optimal for this quadratic system since it is able to capture all 2nd order behavior. See the Appendix VII-C for figures.

B. 2D SLAM

The 2D non-linear SLAM system is significantly more complex. This system estimates the position and orientation of a robot while simultaneously estimating a number of landmark locations. Inputs to the system include a sequence of directional and rotational drive commands, and landmark measurements consisting of bearing and range.

$$\text{state: } x = \begin{bmatrix}\text{pose} & | & \text{landmarks}\end{bmatrix}$$

$$\text{control: } u = \begin{bmatrix}\text{rot1} & | & \text{drive} & | & \text{rot2}\end{bmatrix}$$

$$f(x,u,w) = \begin{bmatrix}\text{pose.rot(rot1).trans\_forward(drive).rot(rot2)} \\ \text{landmarks}\end{bmatrix}$$

$$h(x,v) = \begin{bmatrix}\text{atan2(landmark1, pose)} - \theta \\ \|\text{landmark1 - pose}\|_2 \\ \vdots\end{bmatrix}$$

A test environment was constructed involving ~ 10 landmarks and a control sequence approximating a pentagonal drive around the landmarks. The presence of inverse-tangent in this system makes it a good candidate for UKF2, since it exhibits strong 3rd order behavior. As seen in Figure 10, the UKF2 significantly outperforms the EKF and UKF baselines in terms of both localization and mapping accuracy. The results from 100 randomized trials are shown in Table I.

Filter	Mean Pose Error	Mean Landmark Error
EKF	1.687	1.202
UKF ($\alpha = 0.4$)	0.420	0.448
UKF2 ($\alpha = 0.2, 0.4$)	0.360	0.427
UKF2 ($\alpha = 0.2, 0.4, 0.8$)	**0.294**	**0.391**

TABLE I: Average state estimation errors from 100 randomized trials of the synthetic 2D SLAM system (pentagonal drive).

C. Double Inverted Pendulum

This system features a double pendulum affixed to a movable cart, mounted on a linear rail. A full derivation of the dynamics equations can be found in [2]. The system is observed only at the cart position x and the angle of the first joint θ_1. The angle of the second join θ_2 is completely unobserved, forcing the filters to rely on the prediction step to infer its value.

The pendulum is simulated at a timestep of $\Delta t = 0.001$ using RK4 numerical integration. All the filters are run at a slower rate of $\Delta t = 0.01$ and use forward Euler integration in their dynamics model. This discrepancy is representative of real-world system/model mismatch.

A number of control sequences were tested. For the run show in Figure 6, the cart was accelerated to the right for 2.0 seconds, and then accelerated to the left for 3.0 seconds, leading to chaotic behavior. All filters eventually diverged from the ground-truth due to overwhelming noise and low sample rate, with UKF2 maintaining tracking the longest. See `https://youtu.be/tWpzPl-HMUw` for an animation of a similar test run.

D. Mackey-Glass Sequence

The Mackey-Glass Sequence is a set of differential equations that aim to imitate certain biological behaviors. Rate of change of the state is highly chaotic and is dependent on previous states tau number of time steps into the past.

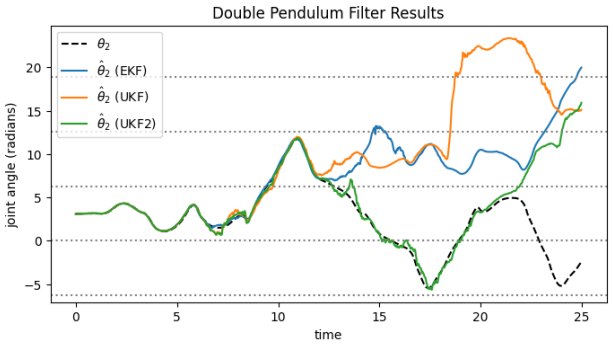


Fig. 6: Filter results on a run of the double pendulum system. The unmeasured second-joint angle θ_2 is shown. All filters diverge from the ground-truth due to overwhelming noise and low sample rate, with UKF2 maintaining tracking the longest. EKF diverges at $t \approx 12.0$, UKF at $t \approx 12.8$, and UKF2 at $t \approx 21.1$.

$$dP/dt = \frac{\beta_0 * \theta^n * P(t-\tau)}{\theta^n + P(t-\tau)}$$

The system shows decent state estimation and tracking against the ground truth results. As shown in Figure 11 the tracking is fairly accurate. The large majority of error in this case can be attributed to a sub sampling rate of 5.

The results from 100 randomized trials are shown below in Table II. We were unable to replicate the dramatic results from [11], in particular our EKF implementation did not experience a noticable phase lag (despite efforts to match experimental design), so there is not much appreciable variety in performance between filters.

Filter	Mean Error
EKF	2.9200
UKF	2.8713
UKF2 ($\alpha = 0.4$)	2.8537
UKF2 ($\alpha = 0.2, 0.4$)	2.8542
UKF2 ($\alpha = 0.2, 0.4, 0.8$)	2.8152

TABLE II: Average state estimation errors in the form of Mahalanobis Distance error from 100 randomized trials of the Mackey-Glass System for tau = 30.

E. Computational Performance

In each evaluation system, the average performance of each filter implementation was measured in iterations per second. The observed rates scaled as expected. The UKF2 runs at roughly half the rate of the UKF. In the pendulum system, for example, the UKF achieved 97Hz while the UKF2 achieved 46Hz, which corresponds to their computing $2n + 1 = 13$ and $4n + 1 = 25$ sigma-points respectively each iteration. It is worth noting that there is significant room for optimization to our current Python implementations (in particular, batch processing of sigma-points via vector operations). Table III shows the full results.

Filter	Car	SLAM	Pendulum	Mackey-Glass
EKF	850	423	488	769
UKF	388	97	409	210
UKF2 ($k = 2$)	178	46	229	113

TABLE III: Observed runtime performance of the (unoptimized) EKF, UKF, and UKF2, measured in iterations per second (Hz).

VI. Conclusion

In this work we investigate a novel modification to the conventional Unscented Kalman Filter, dubbed the UKF2. By sampling the domain at a larger (but still constant with respect to dimension) number of sigma-points, the UKF2 is better able to capture 3rd and higher order behavior of the system that would otherwise have gone ignored, leading to a more accurate propagation of uncertainty. Additionally, the wider spread of samples makes the filter less sensitive to regime changes in the dynamics. We test the UKF2 on various dynamic systems of varying levels of complexity. In all systems tested, the UKF2 achieves better state-estimation performance on average than the EKF and UKF, especially in those systems exhibiting strong 3rd order behavior (SLAM and Double-Pendulum). However, without further optimization, the UKF2 incurs a factor of $k = 2$ or $k = 3$ increase in computation cost over the UKF, which may limit its potential usage in embedded settings. Avenues for future work mainly relate to optimization. These include implementing batch sigma-point processing, as well as perhaps an adaptive scheme that selects a value for k based on knowledge of the current dynamics regime.

Our Python 3 codebase, including KF, EKF, UKF, and UKF2 implementations, is available at `https://github.com/rschwa6308/UKF-Variants`.

References

[1] Vishal Awasthi and Krishna Raj. *A Survey of Kalman Filter Algorithms and Variants in State Estimation*, pages 1–14. 08 2021.

[2] Ian Crowe-Wright. *Control theory: The double pendulum inverted on a cart*. PhD thesis, 2018.

[3] A. Germani, C. Manes, and P. Palumbo. Polynomial extended kalman filter. *IEEE Transactions on Automatic Control*, 50(12):2059–2064, 2005.

[4] Ramakrishna Gurajala, Praveen B. Choppala, James Stephen Meka, and Paul D. Teal. Derivation of the kalman filter in a bayesian filtering perspective. In *2021 2nd International Conference on Range Technology (ICORT)*, pages 1–5, 2021.

[5] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[6] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[7] Michael C. Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.

[8] Leonard A. McGee and Stanley F. Schmidt. Discovery of the Kalman filter as a practical tool for aerospace and industry. 11 1985.

[9] Jung Min Pak and Choon Ki Ahn. State estimation algorithms for localization: A survey. *International Journal of Control, Automation and Systems*, 21(9):2771–2781, Sep 2023.

[10] Michael Roth and Fredrik Gustafsson. An efficient implementation of the second order extended kalman filter. In *14th International Conference on Information Fusion*, pages 1–6, 2011.

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

VII. Appendix

A. Bayes Filter Derivation

We write $y_{1:t}$ to represent sequence of all actions and measurements up to time t.

Predict Step:

$$\begin{aligned}
\overline{Bel}(x_t) &= p(x_t \mid y_{1:t-1}, u_t) \\
&= \int p(x_t \mid y_{1:t-1}, u_t, x_{t-1}) p(x_{t-1} \mid y_{1:t-1}, u_t) dx_{t-1} && \text{Total probability} \\
&= \int p(x_t \mid u_t, x_{t-1}) p(x_{t-1} \mid y_{1:t-1}, u_t) dx_{t-1} && \text{Markov property} \\
&= \int p(x_t \mid u_t, x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) dx_{t-1} && \text{Temporal order } (u_t \text{ cannot affect } x_{t-1}) \\
&= \int p(x_t \mid u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}
\end{aligned}$$

Update Step:

$$\begin{aligned}
Bel(x_t) &= p(x_t \mid y_{1:t-1}, z_t) \\
&= \eta\, p(z_t \mid x_t, y_{1:t-1}) p(x_t \mid y_{1:t-1}) && \text{Bayes' Rule} \\
&= \eta\, p(z_t \mid x_t) p(x_t \mid y_{1:t-1}) && \text{Markov property} \\
&= \eta\, p(z_t \mid x_t) \overline{Bel}(x_t)
\end{aligned}$$

where η is a normalization constant that ensures the PDF sums to 1. The terms $p(x_t \mid u_t, x_{t-1})$ and $p(z_t \mid x_t)$ are called the Action Model and Measurement Model respectively. It is possible to define the system by explicitly specifying these conditional distributions for all possible values of x, u, and z.

B. Uncertainty Propagation

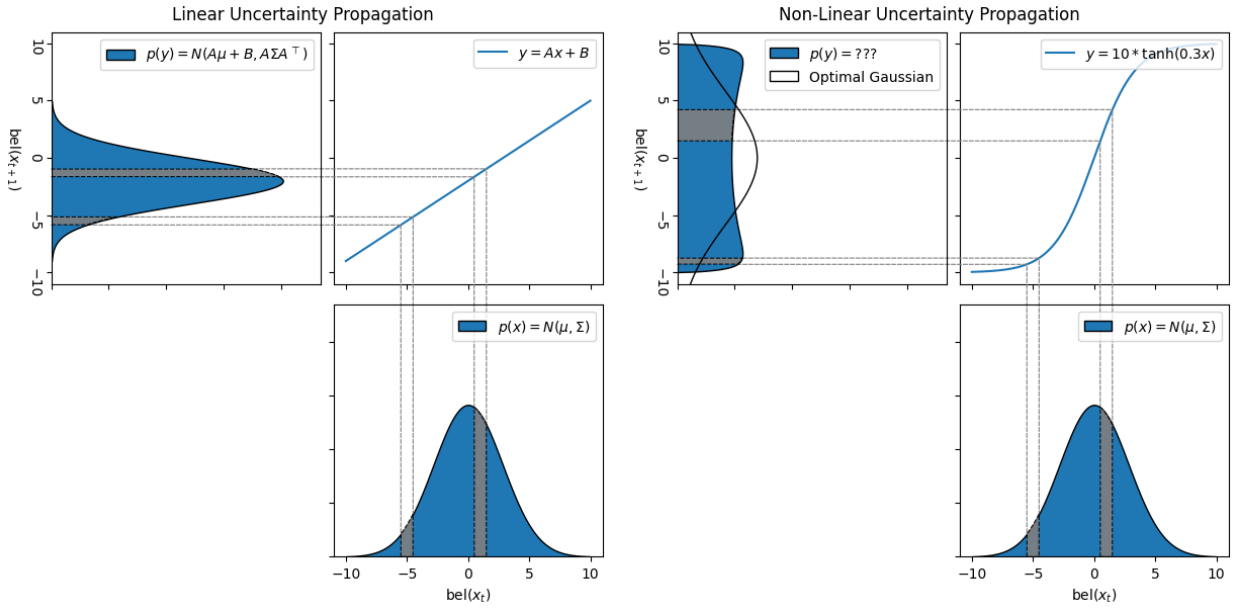


Fig. 7: Uncertainty propagation of a gaussian prior through a linear function (left) and a nonlinear function (right). Select segments of the input distribution and their images in the output distribution are highlighted to show examples of mass transport explicitly. The image under the linear function is an easily-parametrized gaussian, while the image under the nonlinear function is not. However, a gaussian can be fit to the true distribution.

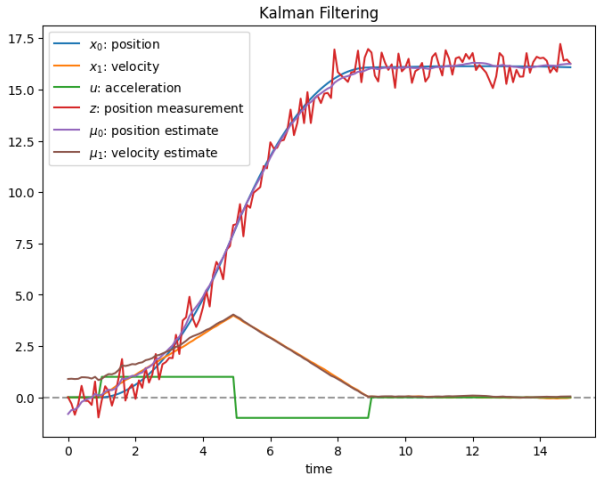


Fig. 8: Kalman Filtering on the 1D Linear Kinematic Car example system

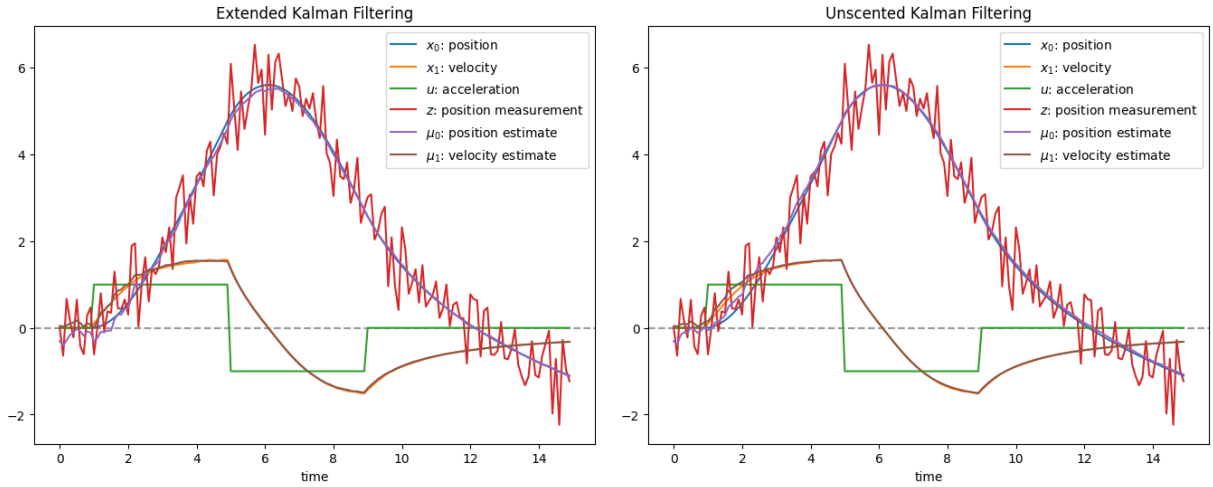


Fig. 9: Extended and Unscented Kalman Filtering on the 1D Nonlinear Kinematic Car example system. The UKF is optimal for this (quadratic) system since it is able to capture all 2nd order behavior. Indeed, UKF2 performed identically.

C. Kinematic Car Results

Filter results for both the linear and non-linear kinematic car system:

D. SLAM Additional Results

Filter results for the 2D non-linear SLAM system:

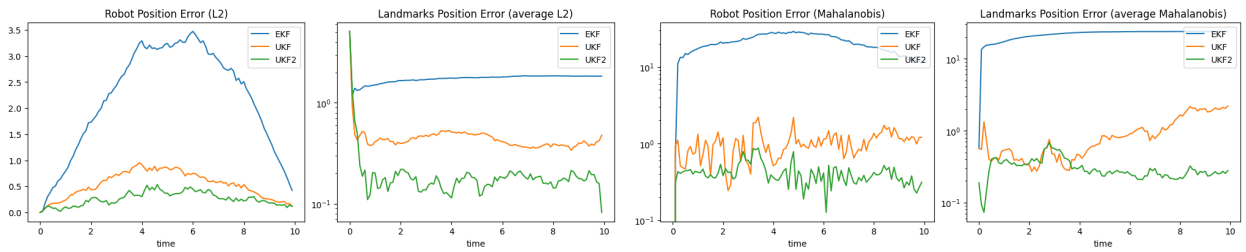


Fig. 10: Filter results on a representative run of the 2D nonlinear SLAM system. Both raw L2 and Mahalanobis errors are shown here.

E. Mackey-Glass Additional Results

Filter results for the Mackey-Glass system:

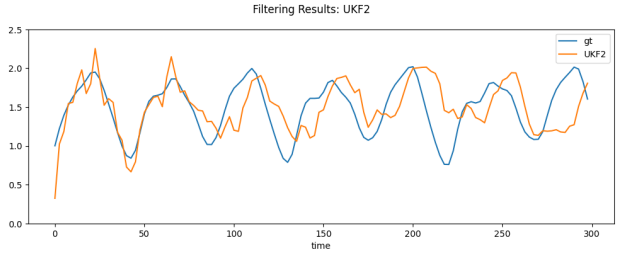


Fig. 11: Filter results on a representative run of the Mackey-Glass system. Raw system values are shown here.