

MAX-PLANCK-INSTITUTE FOR DYNAMICS AND
SELF-ORGANIZATION
GEORG-AUGUST-UNIVERSITY GÖTTINGEN

Data assimilation

Implementation and comparison of different
filtering methods on generated and real data

Participants: Anton Paul Braun
Inga Kottlarz
King Hang Mok
Niklas Weiß
Supervisor: Jan Lebert
Submitted: June 30, 2022

Contents

1. Introduction	1
2. Data Assimilation: Filter Algorithms	1
2.1. The Classical Kalman Filter	2
2.2. The Extended Kalman Filter	2
2.3. Ensembles in General	3
2.4. The Ensemble Kalman Filter	4
2.5. The Particle Filter	4
3. Models	6
3.1. The Lorenz Equations	6
3.2. Nine-dimensional Lorenz System	7
3.3. Disease Transmission Models	8
4. Materials and Methods	9
4.1. The Classical Kalman Filter	9
4.2. The Extended Kalman Filter	9
4.3. The Ensemble Kalman Filter	10
4.4. The Particle Filter	10
4.5. Parameter Estimation	10
4.6. Error Estimation of Real Data	11
4.7. Comparison of the Ensemble Kalman and Particle Filter	11
5. Results	12
5.1. Classical Kalman Filter	12
5.2. Performance of the Extended Kalman Filter on the Lorenz-9 Model	12
5.3. The Ensemble Kalman Filter on the SEIR-Model	14
5.4. Performance of the Filters on the Lorenz-9 Model	14
5.5. Infection Models	16
6. Discussion and Outlook	18
References	20
A. Equations	23
B. Figures	24

Work distribution The work distribution is the following:

Name	worked on
Anton P. Braun	Introduction, Ensembles in general, Ensemble Kalman filter, Lorenz equations, disease transmission models, performance of the filters on the Lorenz-9 model
Inga Kottlarz	Introduction to filter algorithms Particle filter Nine-dimensional Lorenz system Error estimation of real data Performance of the filters on the Lorenz-9 model Discussion and Outlook
Niklas Weiß	The Classical Kalman filter introduction Disease transmission models classical Kalman filter, particle filter from parameter estimation calssical Kalman filter infection Models Discussion and Outlook: SEIR model for SARS-CoV-2
King Hang Mok	The Extended Kalman filter introduction, Nine-dimensional Lorenz system Disease transmission models The Extended Kalman Filter

1. Introduction

A lot of dynamic real world models are governed by a set of seemingly simple state equations. I. e. Cleveland Abbe, a meteorologist, once stated that "meteorology is essentially the application of hydrodynamics and thermodynamics to the atmosphere of the earth ..." [1], since then other great minds tried to make weather forecasting an "exact science" [2] but up to this day the weather report for next week remains inaccurate.

The problem lies in the strong dependence on the initial conditions of chaotic systems [3]. To circumvent these problems clever scientists invented the subject of data assimilation [4, 5]. The common goal of this subject was and remains to optimize given data for a better guess of possible initial conditions. In this manner there is the possibility to either use smoothing algorithms, which apply to an already existent time series or filtering algorithms which work on single points of a time series to refine the next prediction step [6].

This work describes the implementation of different filtering algorithms, namely the classic, extended and ensemble Kalman filter as well as the particle filter. These filters are tested with different dynamic models i.e. the convection models proposed by Lorenz in 1963 [3] and Reiterer in 1998 [7] as well as basic compartmental epidemics models like the SEIR-model [8]. A further point is the comparison of the performance of the ensemble Kalman and the particle filter in a reduced observation space.

2. Data Assimilation: Filter Algorithms

The goal of data assimilation is to assimilate a signal \mathbf{x} , which is the true state of the system, and observed data \mathbf{y} , which depends on the signal via some observation operator \mathcal{H} [6]. The foundation of this process is Bayes formula

$$\mathbb{P}(\mathbf{x}|\mathbf{y}) = \frac{1}{\mathbb{P}(\mathbf{y})} \mathbb{P}(\mathbf{y}|\mathbf{x}) \mathbb{P}(\mathbf{x}). \quad (2.1)$$

This formula connects the Posterior probability density function (pdf) $\mathbb{P}(\mathbf{x}|\mathbf{y})$ to the Likelihood $\mathbb{P}(\mathbf{y}|\mathbf{x})$ and the Prior $\mathbb{P}(\mathbf{x})$. The Likelihood can be obtained from data space, while the Prior can be extracted from the space of unknowns as a prediction based on a dynamical model. In the filtering problems, $\mathbb{P}(\mathbf{y})$ can be treated as a normalisation factor.

The process of data assimilation depends on a model \mathcal{M} that propagates the signal \mathbf{x} in time. In the following it will be assumed that the dynamics is stochastic, and thus the statevector at time k is given by $x_k = \mathcal{M}(x_{k-1}) + \xi_{k-1}$ with $\xi_{k-1} \sim \mathcal{N}(0, \sigma^2)$. Additionally, a controller $w = \{w_k\}_{k=0}^{\infty}$ can be added to the system. Now, given the data \mathbf{y} , the goal is to find a controller so that $|\mathbf{y}_k - \mathcal{H}(\mathbf{x}_k)|$ stays small in some way. In the following, the controller will be set to be $w_k = K(\mathbf{y}_k -$

$\mathcal{H}(\mathbf{x}_k)$) so that

$$\mathbf{x}_k = \mathcal{M}(\mathbf{x}_{k-1}) + K(\mathbf{y}_{k-1} - \mathcal{H}(\mathbf{x}_{k-1})). \quad (2.2)$$

The goal of data assimilation, specifically the Kalman filter, is now to find a K to satisfy this.

2.1. The Classical Kalman Filter

A basic algorithm in data assimilation is the Kalman Filter. It consists of two steps: First the state vector is updated based on the model in a forecast or prediction step and then that new state is updated with the observation data in an analysis step [9].

The classical Kalman Filter assumes a linear Model $\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1}$ and linear transformation between the state vector \mathbf{x}_k and the observation \mathbf{y}_k . The state vector and the observation are Gaussian distributed, the state vector with a covariance \mathbf{P}_k and the observation with a covariance \mathbf{R}_k . The system can be controlled by the control matrix \mathbf{B}_k and the control vector \mathbf{u}_k , with a covariance \mathbf{Q}_k . Hence, the forecast step is given by:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}'_{k-1} + \mathbf{B}_k \mathbf{u}_k \quad (2.3)$$

$$\mathbf{P}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k. \quad (2.4)$$

For the analysis step, the matrix \mathbf{H}_k maps the state vector to the observation space. To get the weight between the forecasted state and the observation, the Kalman gain is defined:

$$\mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \quad (2.5)$$

With that, the analysis steps becomes:

$$\mathbf{x}'_k = \mathbf{x}_k + \mathbf{K}' (\mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k) \quad (2.6)$$

$$\mathbf{P}'_k = \mathbf{P}_k - \mathbf{K}' \mathbf{H}_k \mathbf{P}_k. \quad (2.7)$$

2.2. The Extended Kalman Filter

In real life, dynamical systems are mostly non-linear. The mapping between the state variables space and the observation space is also often non-linear. To perform filtering in these situations, some amendments can be made to the classical Kalman filter algorithms.

Consider the non-linear system defined by the dynamics $\mathcal{M} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ and observations $\mathcal{H} : \mathbb{R}^m \rightarrow \mathbb{R}^p$. Then the extended Kalman filter algorithm is simply the classical Kalman filter algorithm with the following changes:

In the prediction steps, the non-linear dynamics \mathcal{M} is directly used for the transitions of the state variables from the previous time step to the present time

step. Then in the prediction of the covariance matrix, the prediction matrix is the linearized dynamics, i.e. the Jacobian matrix of the dynamics. In the update steps, the measurement residual is obtained by directly using the non-linear observation \mathcal{H} . In updating the covariance matrix, the non-linear observation is linearized, i.e. the Jacobian matrix is used.

The extended Kalman filter algorithm can be concluded in a nutshell:
Prediction steps:

$$\mathbf{x}_k = \mathcal{M}(\mathbf{x}'_{k-1}, \mathbf{u}_{k-1}) \quad (2.8)$$

$$\mathbf{P}_k = \mathbf{F}_{k-1} \mathbf{P}'_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q} \quad (2.9)$$

$$\mathbf{F}_{k-1} = \left. \frac{\partial \mathcal{M}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}'_{k-1}, \mathbf{u}=\mathbf{u}_{k-1}} \quad (2.10)$$

Update steps:

$$\mathbf{K}' = \mathbf{P}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2.11)$$

$$\mathbf{x}'_k = \mathbf{x}_k + \mathbf{K}' (\mathbf{y}_k - \mathcal{H}(\mathbf{x}_k)) \quad (2.12)$$

$$\mathbf{P}'_k = \mathbf{P}_k - \mathbf{K}' \mathbf{H}_k \mathbf{P}_k \quad (2.13)$$

2.3. Ensembles in General

The calculation of the covariance matrices needs the linearization of the model and observation functions. To avoid these calculations a Monte-Carlo-approach to the problem is made [10, 11]. Instead of a single state point an ensemble of points is scattered randomly around some initial condition in state space

$$\mathbf{E} = [\mathbf{x}_1, \dots, \mathbf{x}_n, \dots, \mathbf{x}_N] \in \mathbb{R}^{M \times N}. \quad (2.14)$$

Here N denotes the dimension of the ensemble and M is the dimension of the statevectors $\mathbf{x}_n \in \mathbb{R}^M$, P is the dimension of the observation space. The mean $\bar{\mathbf{x}}$ and covariance $\bar{\mathbf{P}}$ of this ensemble can then be computed by

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n, \quad \bar{\mathbf{P}} = \frac{1}{N-1} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T. \quad (2.15)$$

Then with the anomalies defined as:

$$\mathbf{X} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \dots, \mathbf{x}_n - \bar{\mathbf{x}}, \dots, \mathbf{x}_N - \bar{\mathbf{x}}], \quad (2.16)$$

all of the three variables can be computed in terms of the ensemble and the anomalies:

$$\bar{\mathbf{x}} = \frac{\mathbf{1}^T \mathbf{E}}{N}, \quad \bar{\mathbf{P}} = \frac{\mathbf{X} \mathbf{X}^T}{N-1}, \quad \mathbf{X} = \mathbf{E} \left[\mathbf{I}_N - \frac{\mathbf{1} \mathbf{1}^T}{N} \right], \quad (2.17)$$

where $\mathbf{1} \in \mathbb{R}^N$ corresponds to the 1 vector and \mathbf{I}_N is the N -dimensional unity matrix.

2.4. The Ensemble Kalman Filter

In terms of the ensemble, the ensemble Kalman filter still consists of the forecast and the analysis step. The equations in terms of the ensemble are

$$\mathbf{E}_k = \mathcal{M}(\mathbf{E}'_{k-1}) + \mathbf{D}_{\text{mod}}, \quad (2.18)$$

$$\mathbf{E}_k = \mathbf{E}_k + \bar{\mathbf{K}}(\mathbf{y}_k \mathbf{1}^T - \mathbf{D}_{\text{obs}} - \mathcal{H}(\mathbf{E}_k)). \quad (2.19)$$

\mathbf{E}_k is the forecast and \mathbf{E}'_k the analysis of the ensemble points, $\mathbf{y}_k \in \mathbb{R}^P$ is the observation at the respective timestep, \mathcal{M} and \mathcal{H} represent the model and observation function which are applied column-wise to the ensemble, \mathbf{D}_{mod} and \mathbf{D}_{obs} are the noise matrices of the process whose columns are sampled from $\mathcal{N}(0, \mathbf{Q})$ and $\mathcal{N}(0, \mathbf{R})$, \mathbf{Q} and \mathbf{R} are the covariance matrices of model and observation.

If a linearization \mathbf{H} of the observation exists, the Kalman gain $\bar{\mathbf{K}}$ is given by

$$\bar{\mathbf{K}} = \bar{\mathbf{P}} \mathbf{H}^T [\bar{\mathbf{P}} \mathbf{H}^T + \mathbf{R}]^{-1}. \quad (2.20)$$

If the calculation of the linearization should be too inefficient or not possible, the Kalman gain can still be calculated by the following observations. $\bar{\mathbf{P}} \mathbf{H}^T$ is the cross-covariance between the forecasted ensemble \mathbf{E}_k and the measurement function of the forecasted ensemble $\mathcal{H}(\mathbf{E}_k)$, $\bar{\mathbf{P}} \mathbf{H}^T$ is the covariance of the observation function of the ensemble. If the new variable \mathbf{Y} is introduced as $\mathbf{Y} = \mathcal{H}(\mathbf{E}^f) [\mathbf{I}_N - \mathbf{1}\mathbf{1}^T/N]$, the cross-covariance can be calculated as $\mathbf{X}\mathbf{Y}^T/(N-1)$, the covariance of the observation function is just $\mathbf{Y}\mathbf{Y}^T/(N-1)$. For the inverse the factor $1/(N-1)$ gets reversed as well and the Kalman gain computes as

$$\bar{\mathbf{K}} = \mathbf{X}\mathbf{Y}^T [\mathbf{Y}\mathbf{Y}^T + (N-1)\mathbf{R}]^{-1}. \quad (2.21)$$

With the consideration of an ensemble of points instead of a single one the calculation of the covariance becomes unnecessary for the calculation of a Kalman step.

2.5. The Particle Filter

All previously mentioned Kalman Filters rely on the assumption of a Gaussian distribution of noise. This is not always given in reality, which is why filters with the ability to handle non-Gaussian noise as well as nonlinear dynamical models have been developed.

The example considered here is the particle filter [6, 12]. The particle filter algorithm consists of three distinct steps: prediction, weight update (sequential importance sampling) and resampling.

Prediction Similar to the ensemble Kalman filter, the particle filter works with an ensemble of states (particles) \mathbf{E} that is propagated through time by a model describing the dynamics. Thus, the prediction step is the same as in the ensemble Kalman filter: Each particle is propagated in time by a given model \mathcal{M} .

Weight update After each prediction step, the ensemble is compared to a measurement with the observation operator \mathcal{H} .

In contrast to the ensemble Kalman filter, each of the particles \mathbf{x}_n is assigned a weight w_k that represents how well the particle fits the measurement \mathbf{y} . In the very first step, all particles are assigned an equal weight. This weight is changed due to the congruence between the particle and the measurement during the weight update according to [6, Chapter 4] for the weight of particle \mathbf{x}_n

$$w'_n = w_n \cdot g(\|\mathbf{y} - \mathcal{H}(\mathbf{x}_n)\|),$$

where $g(z) \sim \mathcal{N}(0, \sigma_i)$ is the probability distribution function of a Gaussian distribution with standard deviation σ_i and σ_i a parameter of the filter. Afterwards, the weights are re-normalized by their sum.

In this implementation, σ_i is chosen depending on the mean distance between the ensemble and the measurement $m = \mathbb{E}(\|\mathbf{y} - \mathcal{H}(\mathbf{x})\|_2)$ by a step-function

$$\sigma_i = \begin{cases} 0.05 & m < 0.01 \\ 0.2 & m < 0.05 \\ 0.4 & \text{else} \end{cases},$$

where $\mathbb{E}(x)$ is the weighted mean. This prevents all particle weights being set to zero and thus deviding by zero in the re-normalization step.

Resampling A problem that can occur with this naive weight update method is that if most particles differ significantly from the measurement, possibly almost all of the particles weights are set to zero, resulting in a so-called weight degeneracy [13].

To prevent this, a resampling step is implemented. It is executed if the effective number of particles

$$N_{\text{eff}} = \frac{1}{\sum_{k=1}^N (w_k)^2}$$

is smaller than some threshold Number N_{thresh} . In this implementation, this threshold is chosen to be $N_{\text{thresh}} = N/4$, as suggested by [14].

The resampling step is then implemented according to [15]: A probability distribution of the particles is created according to the weights. From this distribution, N new particles that make up a resampled ensemble are drawn. To ensure that there are not multiple particles with the exact same coordinates in state space, a Gaussian replacement noise $r \sim \mathcal{N}(0, \sigma_M)$ is added to each drawn particle. After the resampling, the weights are re-normalized.

If the standard deviation of the replacement noise, σ_M , is chosen to be constant, either of the following two problems will occur: 1) If σ_M is set to a small value (relative to the scale of the model), and the ensemble deviates largely from the

measurement, it will still deviate largely from the measurement after the resampling step because the new particles are very close to where the old particles were. In this case this is not desired because even those particles with relatively high weights deviate largely from the measurement. 2) If σ_M is set to a large value (relative to the scale of the model), and the ensemble is close to the measurement, it will be spread out during the resampling step, resulting in an unnecessarily large variance that does not reflect proximity between measurement and ensemble. To prevent these problems, an adaptive resampling has been introduced, where σ_M depends on the variance σ^2 of the ensemble and the weighted mean m by

$$\sigma_M = \begin{cases} 0.01 + 0.0001 \cdot m & \sigma^2 < 0.5 \\ 0.05 + 0.0001 \cdot m & \sigma^2 < 1 \\ 0.1 + 0.001 \cdot m & \text{else} \end{cases}$$

3. Models

3.1. The Lorenz Equations

In 1963, Edward Lorenz [3] developed a simplified mathematical model for the Saltzman approximation [16] of the Rayleigh-Bénard convection, which is expressed as three coupled ordinary differential equations.

$$\frac{dx}{dt} = \sigma(y - x) \tag{3.1a}$$

$$\frac{dy}{dt} = x(\rho - z) - y \tag{3.1b}$$

$$\frac{dz}{dt} = xy - \beta z \tag{3.1c}$$

These equations are one of the best known ways to model chaotic systems. The parameters of the equations are b , a parameter referring to the size of the system, the Prandtl-number σ and the reduced Rayleigh-number r . The system is symmetric for $(x, y) \rightarrow -(x, y)$ [17, Chapter 14].

In this report the values used for σ , b and r are set to 10, 8/3 and 28 similar to the original values used by Lorenz [3]. For these parameters the dynamic converges onto a strange attractor whose shape somewhat resembles that of a butterfly. The behaviour of the trajectories on this attractor is chaotic.

For small deviations in initial conditions, these deviations grow exponentially leading to different trajectories. The question whether a butterfly in Brazil could set off a tornado in Texas has coined the term butterfly-effect for the spread of errors in a chaotic system [18]. These errors are not predictable and seem suited for the application of filters.

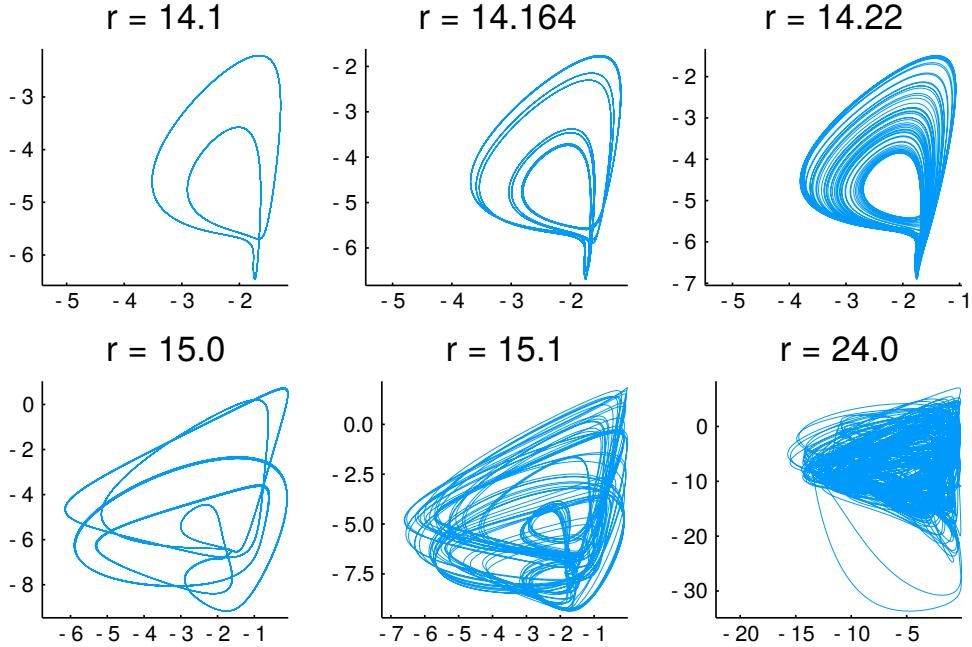


Figure 1: Phase space projections of the 9D attractor on the C_6 - C_7 plane for $a = 0.5$, $\sigma = 0.5$, initial condition $\mathbf{C}^{(0)} = (0.01, 0, 0.01, 0, 0, 0, 0, 0, 0.01)$ with increasing r .

3.2. Nine-dimensional Lorenz System

A high-dimensional version of the Lorenz system in nine-dimensional space was proposed in 1998 by Reiterer [7]. This nine-dimensional model is derived by applying a triple Fourier expansion to the Boussinesq-Oberbeck equations governing thermal convection in a 3D spatial domain by using an approach similar to Lorenz's.

The model can be represented in a system of nine ordinary differential equations (ODEs) given in Equation (A.1) which generate a period-doubling cascade as a route to chaos. The constant parameter b_i , a measure for the geometry of the square cell, is defined by Equation (A.2) depending on a constant a , which is set to $a = 1/2$ in this report.

Parameters of the model are the Prandtl number [19, 20] σ and the reduced Rayleigh number [21, 20] r that serves a control parameter. As r is varied, the attractor shows period-doubling property and becomes chaotic at $r \approx 15.1$. As an illustration, phase space projections of the C_6 - C_7 -plane for different values of r are given in Figure 1.

3.3. Disease Transmission Models

In this project, mainly compartmental disease transmission models are concerned [8], i.e. the concerned population is divided into compartments, and it is assumed that the population in a compartment is homogeneous. The population in a compartment is divided into different classes according to the specific model, and the number of population in a class is a function of time only.

A classic disease transmission model is the SEIR model:

$$\frac{dS}{dt} = -\frac{\beta IS}{N}, \quad (3.2a)$$

$$\frac{dE}{dt} = \frac{\beta IS}{N} - \sigma E, \quad (3.2b)$$

$$\frac{dI}{dt} = \sigma E - \gamma I, \quad (3.2c)$$

$$\frac{dR}{dt} = \gamma I, \quad (3.2d)$$

$S(t)$ is the stock of susceptible, people how can get the virus. $E(t)$ is the stock of exposed, which are people who had contact to the virus but don't show symptoms yet. $I(t)$ is the stock of infected and $R(t)$ is the stock of either recovered or dead population, more generally those who are removed from the disease spreading process. N is the total population. It is assumed in this model that an individual is immunized from the disease after being recovered. The parameters of this model can be directly connected to the real world, β is the infectious rate of the virus, σ the inverse incubation time and γ the inverse recovery time.

Advancements to this model are made in form of addition of vital dynamics, i.e. a birth and a death rate. Other models capture the possibility of reinfection after recovery. Even more complex models consider the possibilities of vaccination, quarantine or an asymptomatic course of disease. These models won't be considered in this work.

For the case of the COVID-19 outbreak the following model is used:

$$\frac{dS}{dt} = -\frac{\beta ES}{N}, \quad (3.3a)$$

$$\frac{dE}{dt} = \frac{\beta ES}{N} - \sigma E, \quad (3.3b)$$

$$\frac{dI}{dt} = \sigma E - \gamma I, \quad (3.3c)$$

$$\frac{dR}{dt} = \gamma I, \quad (3.3d)$$

where the conversion rate between the susceptible part of the population and the exposed part of the population is determined by the exposed instead of the infected themselves. This is, because the assumption was made, that infected people are quarantined once showing symptoms.

4. Materials and Methods

4.1. The Classical Kalman Filter

The first implementation of the Classic Kalman Filter was on a very simple robot model. The robot on position x moves through the one-dimensional room with a constant velocity v :

$$\dot{x} = v, \quad \dot{v} = 0.$$

Therefore the prediction matrix is:

$$F = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}. \quad (4.1)$$

The position of the robot is being measured. To simulate this, the model is solved by an ODE-solver and Gaussian noise with mean zero and standard derivation 0.5 is added to that result.

4.2. The Extended Kalman Filter

The extended Kalman filter (ExKF) is implemented to both the Lorenz-3 system (3.1) and the Lorenz-9 system (A.1). The dimension of observed variables is initially 3-dimensional, then the dimension is gradually reduced to check the limitations of the ExKF.

For the Lorenz-3 system, the dataset was generated by using an integrator of stochastic differential equation with white noise of $\sigma = 0.5$ and $r = 28$, which lies in the chaotic regime of Lorenz-3 system. The observed variables inputted into the ExKF algorithm are 3-, 2- and 1-dimensional respectively. The lower dimensional data points are chosen by extracting the respective dimensions of the generated datasets.

Similar procedures are repeated for the Lorenz-9 system with both periodic and chaotic regimes ($r \in \{15, 15.1\}$) modeled, with the same white noise in generated datasets. The inputted observed variables are started with 9-dimensional, then the dimension is reduced by 1 each time, until the algorithm no longer works at all (returning errors, etc.).

If the set of observed variables is given by $\{y_1, y_2, \dots, y_p\}$, then in both cases, the observation $\mathcal{H} : \mathbb{R}^m \rightarrow \mathbb{R}^p$ is a linear map given by the representation matrix H_k with $(H_k)_{i,a_i} = 1$ and other entries being zero. The covariance matrices are chosen as $P = I$, $R = \sigma I$, $Q = 0.01I$.

The mean L2-norm of the difference between the observed data and the Kalman-filtered data $\sum_{k=1}^T \|x'_k - y_k\|$ for each case is then computed and compared for each dimension.

4.3. The Ensemble Kalman Filter

The ensemble Kalman filter is implemented according to the equations of Section 2.4. To test the filter, observation data is created by integration of the Lorenz-9 model introduced in Section 3.2 as a system of stochastic differential equations (SDE), resulting in dynamical noise. The parameters used are $r \in \{15.0, 15.1\}$, with diagonal noise in the integration of $\sigma = 0.01$. The resulting data is fed into the filter to test the performance of the filter in the chaotic and periodic regime. For the infection models the adapted SEIR-model introduced in Section 3.3 together with observed data from the recent outbreak of SARS-cov2 taken from the data of the Johns Hopkins University [22, 23] was used to see whether the SEIR-model captures the dynamics.

4.4. The Particle Filter

The particle filter is implemented according to the description of the generic algorithm described in Section 2.5. It is tested with generated noisy data from the Lorenz-9 model integrated as an SDE, thus introducing dynamical noise. As for the other filters, it is chosen that $r \in \{15.0, 15.1\}$. The diagonal noise in the integration of the SDE is $\sigma_i = \{0.001, 0.005, 0.01\}$.

Additionally, the Lorenz-9 model is integrated as an ODE and random noise is added after integration. With this generated data, the performance of the filter is tested both in the periodic and chaotic regime with standard deviations $\sigma_i = \{0.01, 0.1, 1\}$.

4.5. Parameter Estimation

The particle filter is also tested with the real data of the SARS-CoV-2 pandemic from the Johns Hopkins University [22, 23]. Therefore the SEIR-model in Section 3.3 is used. Because the parameters of this model are not known and expected to change in time, parameter estimation is done: Two of the three parameters, the infectious rate β and the recovery rate σ , were added to the differential equations, with derivatives equaling zero respectively. These parameters can not change in the forecast step, but in the resampling step. The third parameter, the incubation rate γ , was set to $1/5.1$, because 5.1 day is the assumed incubation time [24].

One problem is the estimation of the initial conditions. For the recovered and the infected just the first observed values are used, the number of exposed is ten times the number of infected. The susceptible number is assumed to be some percentage of the total population in the region or country, in the filter the percentage was set to 0.1%. As a initial condition of the recovery rate $\gamma = 1/11.5$ was used, because 11.5 days is supposed to be a standard recovery time [24]. The difficulty is the estimation of the infectious rate β , because it depends on several factors like how much tests are done. After some tries, 0.4 happened to be a good initial condition. Another problem is the covariance of the states. Especially the covariances of the

parameters β and γ were difficult, because the result is heavily dependent on this choice. If it is too small, the parameters do not vary much but the stocks of population, which lead to a negative exposed stock in some cases. But if it is too big, the parameters change too much and sometimes become negative. Both of these cases were unrealistic in the real world. At the end, the covariances were set to $\sigma_\beta = 0.02$ and $\sigma_\gamma = 0.03$. They are not correlated, because this is not implemented in the algorithm. The errors of the susceptible, exposed, infected and recovered stock are set to 4000, 100, 100 and 100, respectively.

4.6. Error Estimation of Real Data

The data used for the filtering of the COVID-19 development is given by [22, 23] in the form of a time series per province in China and per country worldwide. There are no errors given, and since this time series is collected from various sources, an estimation of errors through comparison with other time series is not possible here.

For this reason, the estimation of errors is made on the following grounds: It is assumed that, for the detection of COVID-19 cases, there is a certain acceptance a , which is a percentage of people who actually gets tested. Thus, a percentage of $1 - a$ does not get tested; either because the symptoms are not severe enough or because of lack of access to/ avoidance of medical care. In [25], the avoidance of medical care among the rural US population was estimated to 34 %. Since urban population is not considered here and it can be assumed that the health care avoidance among urban population is lower due to easier access the mean health care avoidance is estimated to $(20 \pm 20) \%$. The variance is chosen to be this large satisfy the cultural differences in the various countries considered here. On these grounds, the acceptance is estimated to $a = (80 \pm 20) \%$.

Another factor that has to be taken into account here is the efficiency e of the COVID-19 lab test. Here, we estimate $e = (99 \pm 1) \%$.

4.7. Comparison of the Ensemble Kalman and Particle Filter

For the comparison of the ensemble Kalman and particle filter the Lorenz-9 model as introduced in Section 3.2 is used. This model is integrated as stochastic differential equation using a noise of $\sigma = 0.01$ over a timespan of 0 to 1000. The filters start from a random initial condition and get data consisting of one of the nine variables of the integrated Lorenz-9 system starting from $t = 500$ and continuing with time steps of 0.1 until $t = 1000$.

The results computed by the filter are then compared to the actual observed data in terms of their euclidean distance. Since the Lorenz-9 model seems to show some symmetries [26], the absolute of the filtered results and observed data is calculated and the distance between those computed as well. The mean of those distances

over the time series is calculated. This procedure is repeated 20 times to calculate a mean over the means and a respective standard deviation.

5. Results

5.1. Classical Kalman Filter

In figure 2 Kalman-filtered data is illustrated. For the position, it is compared to the observed data and for the velocity with the model data. On the right side of figure 2 the difference between the Kalman-filtered and the observed or modeled data is shown.

The difference of the position is smaller than the difference of the velocity. A reason for that is, certainly, that the observed position is input and the velocity data not. But also the generation of the observed data causes that: Because of the summation with a random number it is more likely, that the difference of two observed data points is large. The filter has problems to adopt to this as can be seen in this example between time 5 and 6. Because of that, a SDE is used to generate the observed data for future models.

In the deterministic model, the observed data always stays on the path of the deterministic solution. In contrast, in the stochastic model this is not true. After some time, the model could have an entirely different state as a deterministic solution. Especially in a chaotic system, this is important, because slight changes to the state vector result to huge changes after some time.

5.2. Performance of the Extended Kalman Filter on the Lorenz-9 Model

In the implementation of ExKF on the Lorenz-9 system, the algorithm shows larger mean deviation (defined as $\sum_{k=1}^T \|x'_k - y_k\|$) as observed variables of lower dimension are inputted in the algorithm, shown in Figure 3. When the dimension is smaller than 6, the code returns a singular matrix error. Unlike the ensemble Kalman filter and particle filter, which work well even when the observed variable is only 1-dimensional, the ExKF is less powerful.

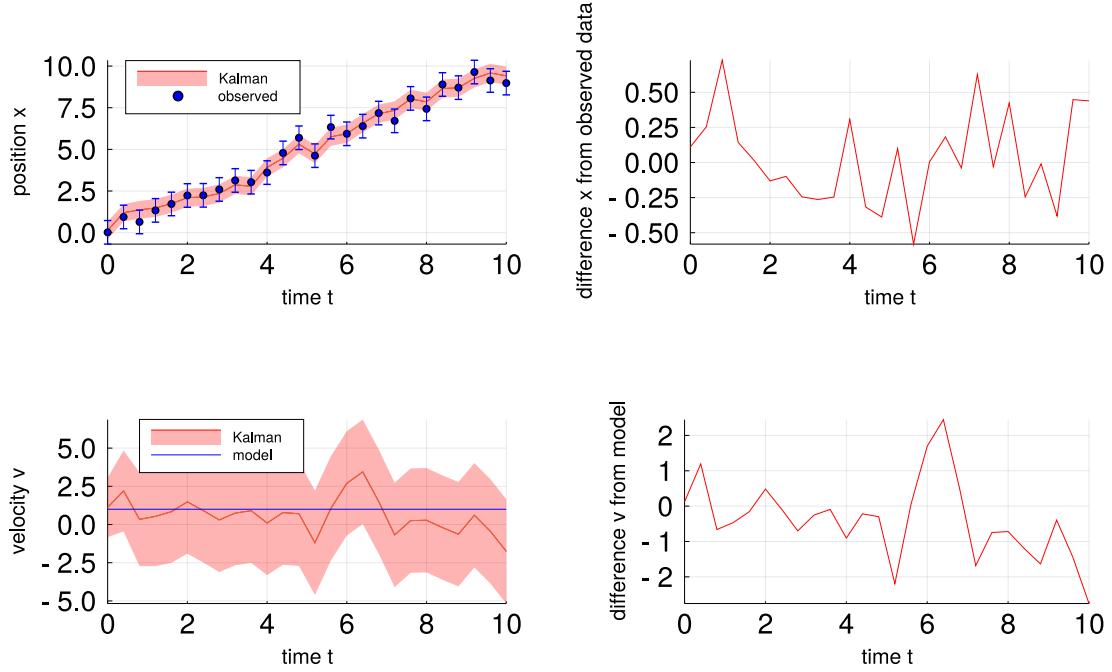


Figure 2: Observed and Kalman-filtered data of the robot model. At the top the Kalman-filtered and observed position of the robot is illustrated and at the bottom the Kalman-filtered velocity compared to the model data. On the left side the absolute values are shown and on the right side the difference from the observed and modeled data.

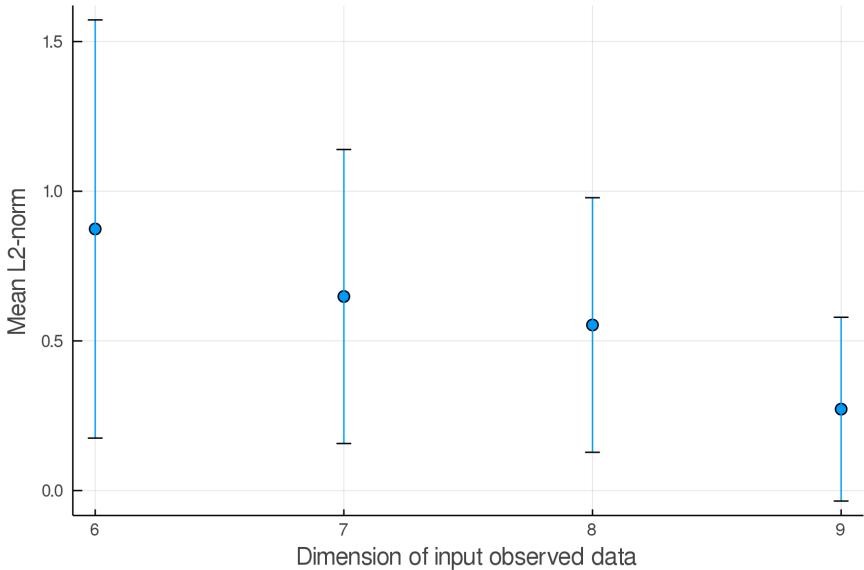


Figure 3: The time mean deviation (L2-norm) between the observed and (extended) Kalman-filtered data for number of dimensions of input variables 6, 7, 8 and 9. ($r = 15.1$, $t \in [0, 200]$)

5.3. The Ensemble Kalman Filter on the SEIR-Model

The performance of the ensemble Kalman filter together with the adapted SEIR-Model as described in Section 4.3 is shown in Figure 10. The filter uses the data for Hubei which displays a jump, because the counting method for the infected changed. The filter is relatively close to the dynamics until the counting method changes, afterwards it is not able to catch the dynamics again.

5.4. Performance of the Filters on the Lorenz-9 Model

The ensemble Kalman filter and the particle filter are tested for their capabilities of cross-prediction, that is predicting the course of the variables if only given one. Every variable out of the nine Lorenz-9 dimensions is given to the filters as an input. The results of the comparison described in Section 4.7 are shown in Figures 4, 5, 13 and 14.

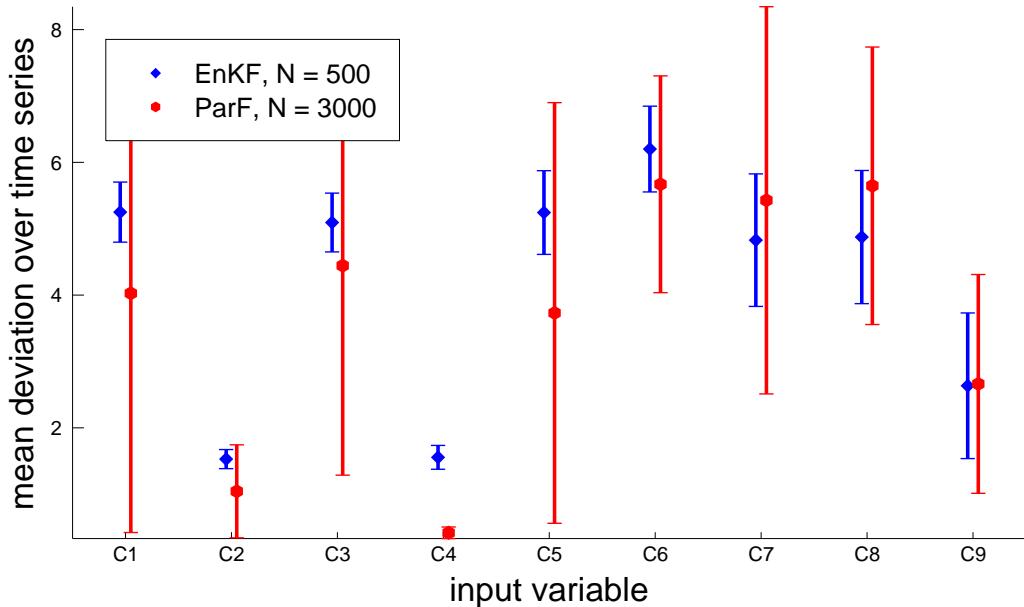


Figure 4: The mean deviation between the solution of the Lorenz-9 system with $r = 15.0$ and the results of the Particle and Ensemble Kalman filter are displayed for each of the nine variables as a single input variable. The Lorenz System was integrated in the periodic case ($r = 15.0$) as a stochastic differential equation with a diagonal noise of $\sigma = 0.01$ and the data is distanced by a timestep of $dt = 0.1$. The displayed standard deviation was calculated from the standard deviation of the mean over 20 simulations. The filter starts from random initial conditions.

Ensemble Kalman filter For the deviation displayed in Figures 4 and 13 the Ensemble Kalman filter displays lower deviation for the input variables $C2$ and $C4$. The deviation for input variable $C9$ is also somewhat lower. The filter exhibits no significant dependence whether the regime of the system is periodic or chaotic. For the absolute values shown in Figures 5 and 14 these observations differ slightly. While $C2$ and $C4$ still show the smallest deviations, the deviation of $C9$ is joined by $C6$, $C7$ and $C8$, while $C1$, $C3$ and $C5$ display the largest deviations. The filter is still independent of the used regime (periodic or chaotic).

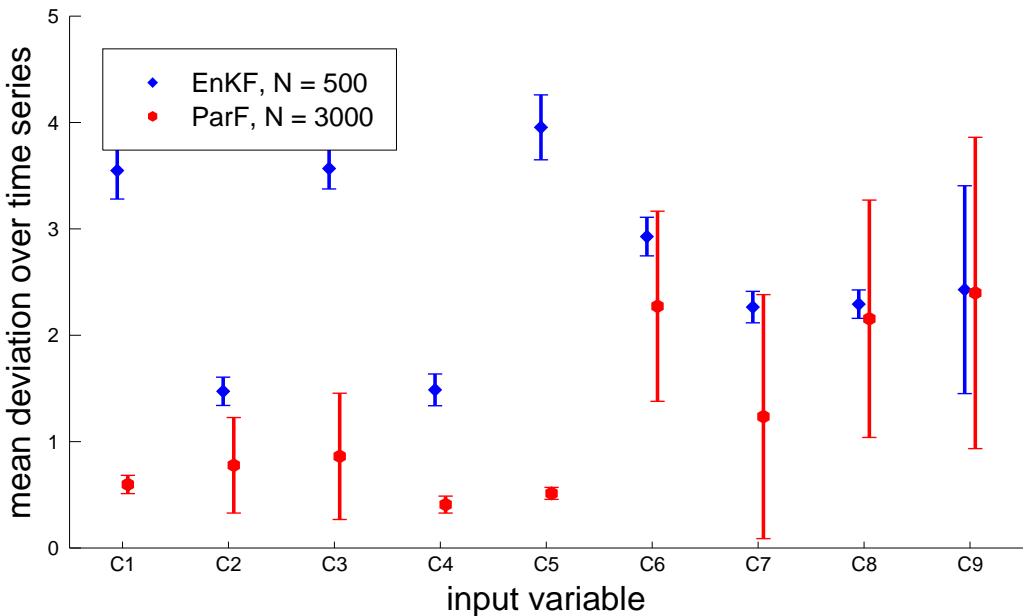


Figure 5: The mean deviation between the absolute of the solution of the Lorenz-9 system and the absolute of the results of the Particle and Ensemble Kalman filter for single input variables. The Lorenz System was integrated for $r = 15.0$ as SDE with a diagonal noise of $\sigma = 0.01$ and the filter updated with $dt = 0.1$, starting from random initial conditions. The standard deviation was calculated in terms of the mean over 20 simulations.

Particle filter In Figures 4 and 13, the mean discrepancy between simulated observed and predicted filtered data is displayed in the periodic and chaotic regime. It becomes clear that there is no significant difference between the periodic and chaotic regime for either of the input variables. But there is a significant difference in the discrepancy for different input variables: For the input variables $C2$ and $C4$, the mean discrepancy of the time series is significantly lower than for any other input variable.

Looking thoroughly at Equation (A.1), it becomes clear that the system is invari-

ant under a change of sign of the variables $C2, C4, C9$: If $(C1, C2, C3, C4, C5, C6, C7, C8, C9)$ is a solution, so is $(C1, -C2, C3, -C4, C5, C6, C7, C8, -C9)$. This explains the significant difference in discrepancy for those three input variables. Thus, additionally to the mean discrepancy, also the mean discrepancy between the absolute values was calculated. This is shown in Figures 5 and 14.

It becomes clear that for the rest of the input variables, the discrepancy between the absolute values is significantly lower than the general discrepancy, suggesting that from the input, all other variables have in fact been correctly predicted except for the sign. As an example, the filtered data and signal of $C9(t)$, given $C3$ as input, are compared in Figures 11 and 12.

Comparison Particle and Ensemble Kalman filter In the case without absolutes as shown in Figures 4 and 13 the standard deviation of the Particle filter is mostly higher than the standard deviation of the Ensemble Kalman filter but the values themselves are within one or two σ -margins of each other. The two filters seem to perform equally under the same conditions.

Taking the absolute as in Figures 5 and 14 this looks different. While variables $C6$ to $C8$ are within one or two σ -margins of each other, the variables $C1, C3$ and $C5$ have lower deviations for the Particle filter than for the Ensemble Kalman filter. Variables $C2$ and $C4$ are also lower. $C9$ is within one σ -margin of the other for both cases. The Particle filter performs better than the Ensemble Kalman filter for variables $C1$ to $C5$.

5.5. Infection Models

Initially, the models are tested with observed data of the Chinese province Anhui, which is adjacent to the city Wuhan, where the virus broke out. This is, because the virus has reached the maximum number of infected people and there are currently no infected people. Hence the model and the filter can be tested at its best. The observed and filtered data of the populations is shown in figure 6, the parameters are illustrated in figure 7.

As shown in the figure of the populations, the filter and the model seem to work with the observed data. This is just partially true, because the data is not reproducible. Future runs of the filter would lead to other results. Especially the number of exposed and the infectious rate β vary much in different runs of the filter, while both are not observed. The trend is similar in every run, but in more detail they are not equal.

The parameters can be compared to developments in the real world: The infectious rate β shrinks, which could be connected to the fact that people care more about the virus or governmental regulations like quarantine. However, the rate is only convincing before March, because in March the number of exposed and infected people goes to zero. This is different for the recovery rate γ : It is expected, that the rate stays constant or increases slightly, which is true in the beginning,

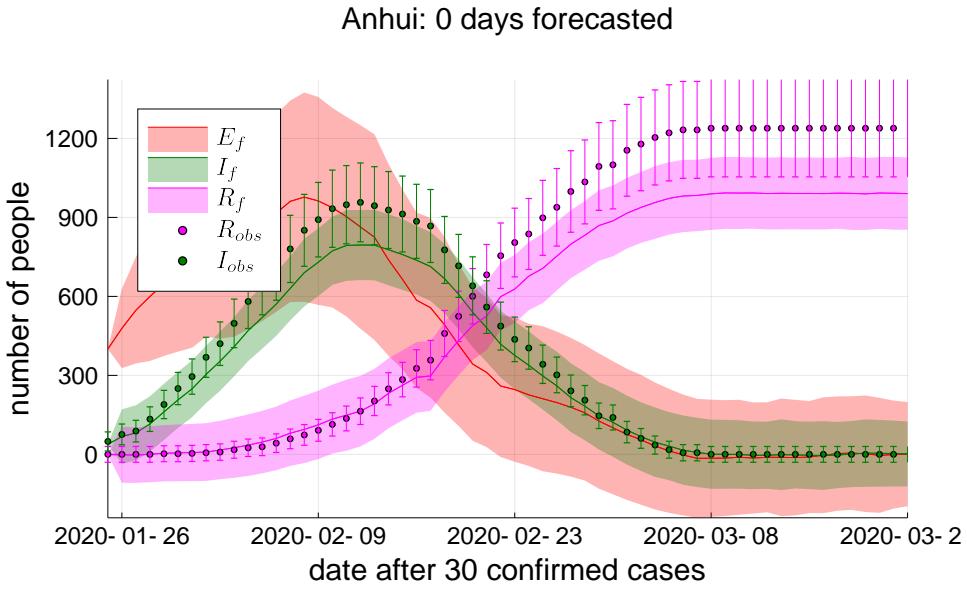


Figure 6: SARS-CoV-2 data of Anhui. E_f are the exposed, I_f the infected and R_f the recovered people, that are calculated by the Particle filter. For the observed recovered R_{obs} and infected I_{obs} stocks, data from the Johns Hopkins University [22, 23] were used.

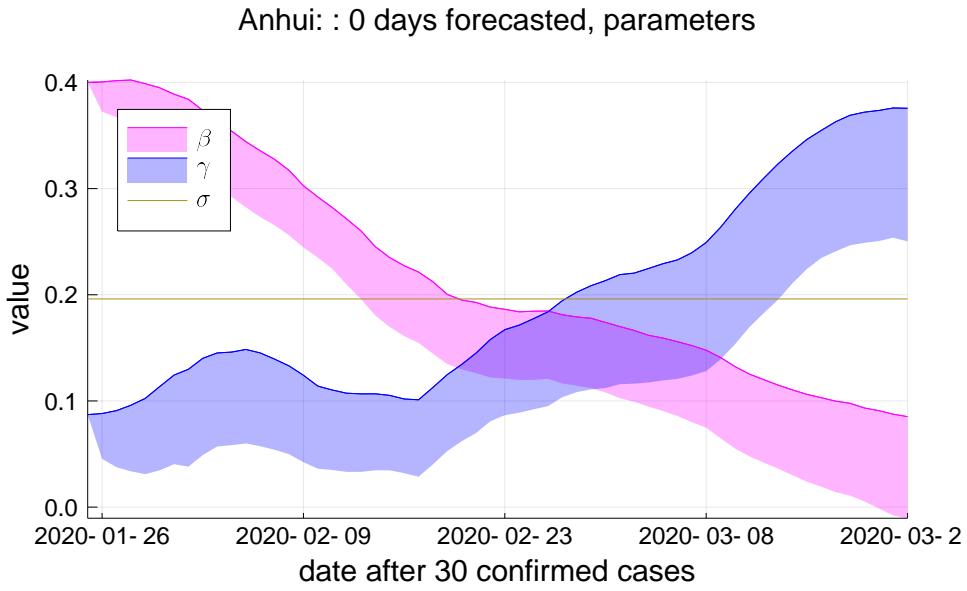


Figure 7: SARS-CoV-2 parameters of Anhui. β is the infectious rate and γ the recovery rate. The incubation rate σ is assumed to be constant. The upper area of error is missing.

but at the end of February γ increases more and reaches a value of around 0.2, which means a recovery time of five days. This is far less than studies of the virus indicate [24]. The data in March is not convincing again.

Nevertheless, with the particle filter a forecast for the next ten days is made: The update and resample step are excluded from the algorithm, the parameters stay constant. The population is just updated by the model. A figure of the country Germany with forecast is given in the appendix, Figure 15. In Figure 16 there are also the parameters of this model given. In the trend it seems to be a normal course of an infection wave, but the error of the variables increases fast.

6. Discussion and Outlook

Problems in working with real world data In this seminar, different filtering algorithms have been implemented and tested on both generated and real data. The main difficulty when working with the real data from [22, 23] has been the estimation of the covariance matrices of the measurements.

The measurement error of each individual observable can be estimated as described in Section 4.6, but the correlation between the observables cannot be determined as it changes over time. Thus, it is not constructive to calculate the correlation of the observables over the whole time series; at the same time it is not mathematically possible to calculate the correlation of two single measurements.

Because of this, the true performance of the filters cannot fully be determined due to missing or only partly available, roughly estimated input. On the other hand, this does provide insight into the feasibility of the Kalman filters with this kind of problems, where the covariance of the observables is time-dependent and unknown.

Lessons from the Lorenz-9 model During testing of the particle and ensemble Kalman filter on the Lorenz-9 model, it became clear that the quality of the cross-prediction (measured by the deviation between generated data and filtered data) heavily depends on the variables given as an input to the filter. As displayed in Figures 4, 5, 13 and 14, in the case of the Lorenz-9 model the results with $C2$ and $C4$ are in general closer to the true observation, and especially have a smaller standard deviation than for other input variables.

In Equation (A.1), it can be seen that the derivative of all variables directly depends on $C2$ and $C4$, whereas there are very few variables whose derivatives depend directly on $C6$, for example. At this point, this is only an observation that would have to be thoroughly investigated.

The results in Figures 4, 5, 13 and 14 agree with [26] in the sense that if given only one input variable, the cross prediction of the filters is not perfect in any case. In most cases, though, at least the absolute values of the time courses are within the standard deviation estimated by the filter (see for example Figure 12).

All in all, the implemented filters work as expected with generated data in the sense that the ensemble Kalman filter and the particle filter perform significantly better with the Lorenz-9 model than the extended Kalman filter.

Regarding the particle filter, other resampling methods [13] can be explored to see how this effects the performance of the filter. Difficulties in working with real data have been elaborated above; a solution for this has not been found at this point. Here, the particle filter has a clear advantage over the Kalman filters as it does not necessarily need the covariance of the observation noise as an input.

SEIR model for SARS-CoV-2 The solution of the SEIR model is unstable, different runs of the algorithm lead to different solutions. Particularly the exposed stock and the infectious rate β change. On the one hand these variables are just by one term connected to the observed values in the differential equations 3.3. On the other hand the number of susceptible people is very large. Because it is much bigger than the number of exposed, infected or recovered people and the total number of population stays constant in the model, the number of susceptible relatively does not vary much; $S \approx N$. With this the second differential equation in the modified SEIR-model (3.3b) becomes: $\frac{dE}{dt} = (\beta - \sigma)E$. Due to the constant incubation rate σ , only β determines, whether E is increasing or decreasing in the model. Overall this leads to a strong uncertainty of these two variables.

This is not a problem of the filter, but of the model which is used. Overall it might not be possible, to describe a disease, like the SARS-CoV-2, in such a simple model.

References

- [1] Edmund P. Willis and William H. Hooke. Cleveland abbe and american meteorology, 1871–1901. *Bulletin of the American Meteorological Society*, 87(3):315–326, 2006.
- [2] Peter Lynch. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, 227(7):3431 – 3444, 2008. Predicting weather, climate and extreme events.
- [3] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.
- [4] 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.
- [5] Y. Sasaki. An objective analysis based on the variational method. *Journal of the Japanese Meteorological Society. Series 2*, 36(3):77–88, 1958.
- [6] Kody Law, Andrew Stuart, and Konstantinos Zygalakis. *Data assimilation: A mathematical introduction*, volume 62 of *Texts in Applied Mathematics*. Springer, Cham, 2015. A mathematical introduction.
- [7] Peter Reiterer, Claudia Lainscsek, Ferdinand Schürrer, Christophe Letellier, and Jean Maquet. A nine-dimensional lorenz system to study high-dimensional chaos. *Journal of Physics A: Mathematical and General*, 31(34):7121–7139, aug 1998.
- [8] Fred Brauer. *Compartmental Models in Epidemiology*, pages 19–79. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [9] Ramsey Faragher. Understanding the basis of the kalman filter. *IEEE Signal Processing Magazine*, (12):128–132, Sep 2012.
- [10] Patrick Nima Raanes. Introduction to data assimilation and the ensemble kalman filter. 2016.
- [11] Jan Mandel. A brief tutorial on the ensemble kalman filter, 2009.
- [12] Guoliang Liu. *Bayes Filters with Improved Measurements for Visual Object Tracking*. Dissertation, Georg-August-Universität Göttingen, 2012. <https://ediss.uni-goettingen.de/bitstream/handle/11858/00-1735-0000-0006-B3F9-2/liu.pdf?sequence=1>.
- [13] R. Douc and O. Cappe. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69, Sep. 2005.

- [14] Frank Nielsen and Frederic Barbaresco. *Geometric Science of Information: Third International Conference, GSI 2017, Paris, France, November 7-9, 2017, Proceedings*. Springer, 01 2017.
- [15] Ali Hirsch. *Computational methods in finance*. CRC Press, Boca Raton, FL, 2013.
- [16] Barry Saltzman. Finite amplitude free convection as an initial value problem—i. *Journal of the Atmospheric Sciences*, 19(4):329–341, 1962.
- [17] Stephen Smale, Morris W Hirsch, and Robert L Devaney. *Differential equations, dynamical systems, and an introduction to chaos; 2nd ed.* Academic Press, New York, NY, 2003.
- [18] Edward N Lorenz. *The essence of chaos*. University of Washington press, 1995.
- [19] Md Rahman, Ming LV, and Zefei Zhu. Impact of prandtl numbers on turbulence modeling. *Journal of Physics: Conference Series*, 1324:012060, 10 2019.
- [20] P.-E Roche, Bernard Castaing, Benoît Chabaud, and Bernard Hébral. Prandtl and rayleigh numbers dependences in rayleigh-bénard convection. *EPL (Europhysics Letters)*, 58:693, 01 2007.
- [21] Yan Guo and Yongqian Han. Critical rayleigh number in rayleigh-bénard convection. *Quarterly of Applied Mathematics*, 68, 03 2010.
- [22] Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track covid-19 in real time. *The Lancet Infectious Diseases*, 2020.
- [23] Ensheng Dong, Hongru Du, and Lauren Gardner. Johns Hopkins CSSE git repository of sars-cov-2 cases. https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series. last accessed: June 30, 2022.
- [24] Stephen A. Lauer, Kyra H. Grantz, Qifang Bi, Forrest K. Jones, Qulu Zheng, Hannah R. Meredith, Andrew S. Azman, Nicholas G. Reich, and Justin Lessler. The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application. *Annals of Internal Medicine*, 03 2020.
- [25] Angela M. Spleen, Eugene J. Lengerich, Fabian T. Camacho, and Robin C. Vanderpool. Health care avoidance among rural populations: results from a nationally representative survey. *The Journal of rural health : official journal of the American Rural Health Association and the National Rural Health Care Association*, 30(1):79–88, 2014. 24383487[pmid], PMC3882339[pmcid].

- [26] Christophe Letellier, Irene Sendiña-Nadal, Ezequiel Bianco-Martinez, and Murilo S. Baptista. A symbolic network-based nonlinear theory for dynamical systems observability. *Scientific Reports*, 8(1):3785, 2018.

A. Equations

$$\frac{dC_1}{dt} = -\sigma b_1 C_1 - C_2 C_4 + b_4 C_4^2 + b_3 C_3 C_5 - \sigma b_2 C_7 \quad (\text{A.1a})$$

$$\frac{dC_2}{dt} = -\sigma C_2 + C_1 C_4 - C_2 C_5 + C_4 C_5 - \sigma C_9/2 \quad (\text{A.1b})$$

$$\frac{dC_3}{dt} = -\sigma b_1 C_3 + C_2 C_4 - b_4 C_2^2 - b_3 C_1 C_5 + \sigma b_2 C_8 \quad (\text{A.1c})$$

$$\frac{dC_4}{dt} = -\sigma C_4 - C_2 C_3 - C_2 C_5 + C_4 C_5 + \sigma C_9/2 \quad (\text{A.1d})$$

$$\frac{dC_5}{dt} = -\sigma b_5 C_5 + C_2^2/2 - C_4^2/2 \quad (\text{A.1e})$$

$$\frac{dC_6}{dt} = -b_6 C_6 + C_2 C_9 - C_4 C_9 \quad (\text{A.1f})$$

$$\frac{dC_7}{dt} = -b_1 C_7 - r C_1 + 2 C_5 C_8 - C_4 C_9 \quad (\text{A.1g})$$

$$\frac{dC_8}{dt} = -b_1 C_8 + r C_3 - 2 C_5 C_7 + C_2 C_9 \quad (\text{A.1h})$$

$$\frac{dC_9}{dt} = -C_9 - r C_2 + r C_4 - 2 C_2 C_6 + 2 C_4 C_6 + C_4 C_7 - C_2 C_8 \quad (\text{A.1i})$$

$$b_1 := 4 \frac{1+a^2}{1+2a^2} \quad (\text{A.2a})$$

$$b_2 := \frac{1+2a^2}{2(1+a^2)} \quad (\text{A.2b})$$

$$b_3 := 2 \frac{1-a^2}{1+a^2} \quad (\text{A.2c})$$

$$b_4 := \frac{a^2}{1+a^2} \quad (\text{A.2d})$$

$$b_5 := \frac{8a^2}{1+2a^2} \quad (\text{A.2e})$$

$$b_6 := \frac{4}{a+2a^2} \quad (\text{A.2f})$$

$$(\text{A.2g})$$

B. Figures

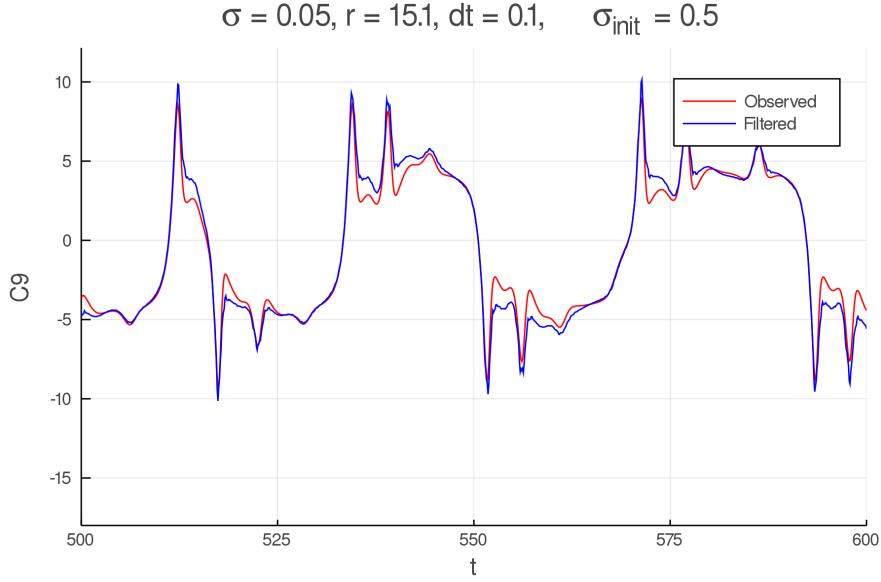


Figure 8: The comparison between the generated data from SDE with filtered data of C9 using ExKF, with variables C1, C2, C3, C4, C5, C6 as input, $t \in [500, 600]$.

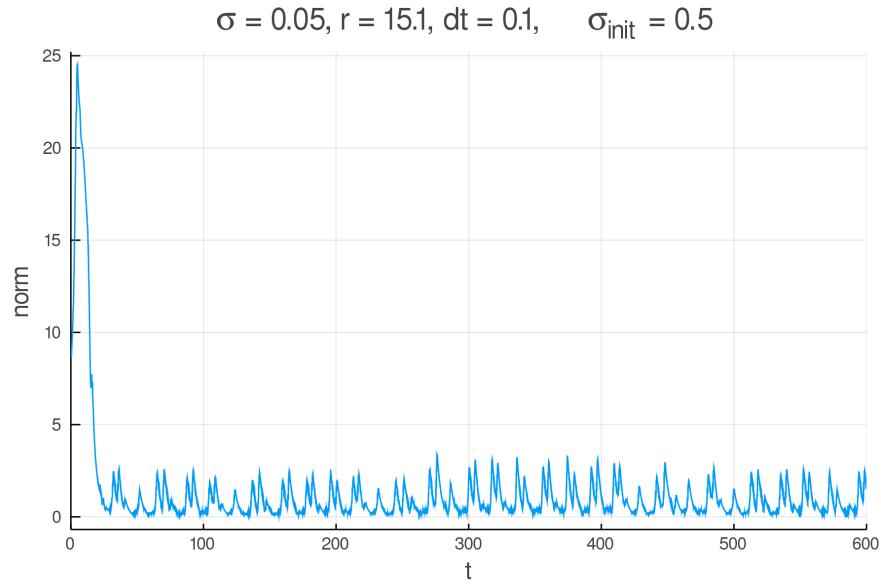


Figure 9: The norm of discrepancy between the 9-dimensional observed and filtered variables, $t \in [0, 600]$, with the same setup as in Figure 8.

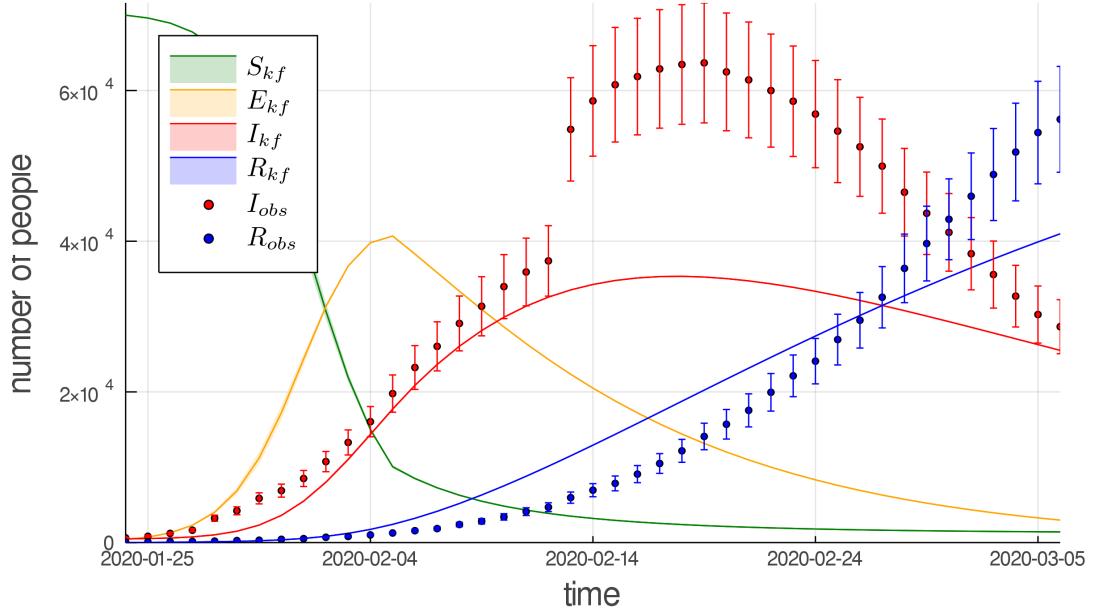


Figure 10: The prediction for the SEIR-model based on the data of the Chinese province Hubei with parameters $\beta = 0.7$ for $t < 14$ and $\beta = 0.3$ afterwards, $\sigma = 1/10$ and $\gamma = 1/24$

$\sigma = 0.01$, $r = 15.1$, input = C3, $dt = 0.1$, $N = 3000$, r.i.c.

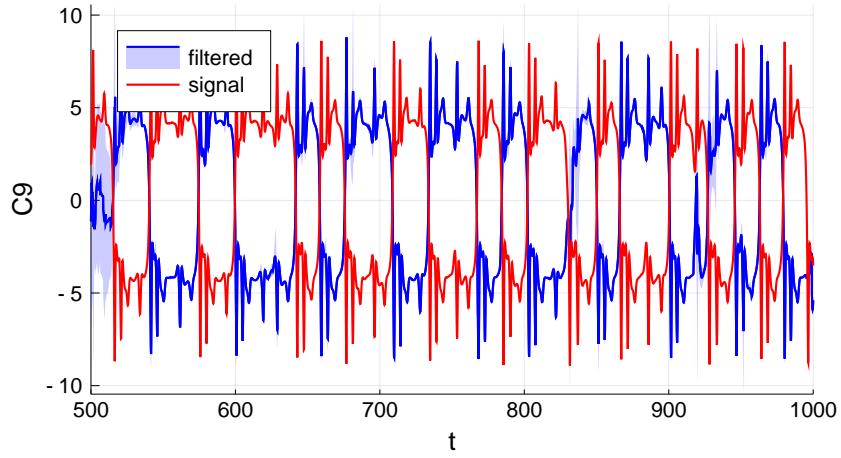


Figure 11: The true solution for $C9(t)$ of the Lorenz-9 system (integrated as an SDE) is displayed together with the output of the particle filter, given $C3$ as an input variable. The filtered data and the signal look mirrored on the x -axis. This is visually confirmed by Figure 12, where the absolute value of both signal and filtered data are displayed together.

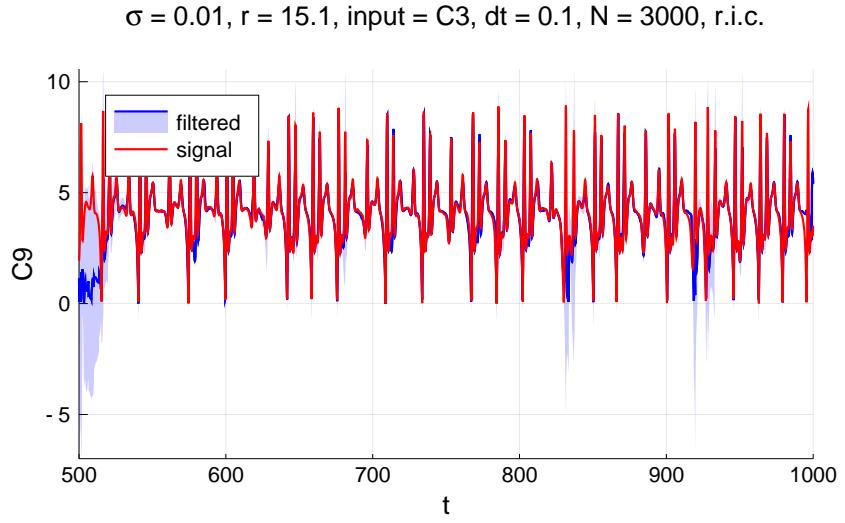


Figure 12: The true solution for $|C9(t)|$ of the Lorenz-9 system (integrated as an SDE) is displayed together with the output of the particle filter, given $C3$ as an input variable. Compare to Figure 11.

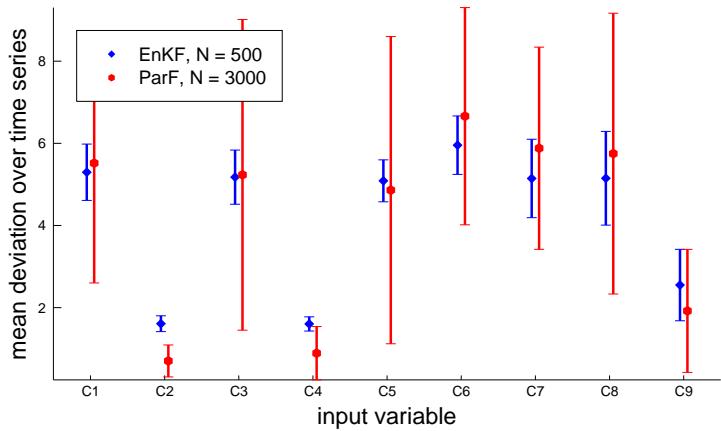


Figure 13: The mean deviation between the solution of the Lorenz-9 system and the results of the particle and ensemble Kalman filter for single input variables. The Lorenz System was integrated for $r = 15.1$ as SDE with a diagonal noise of $\sigma = 0.01$ and the filter updated with $dt = 0.1$, starting from random initial conditions. The standard deviation was calculated in terms of the mean over 20 simulations.

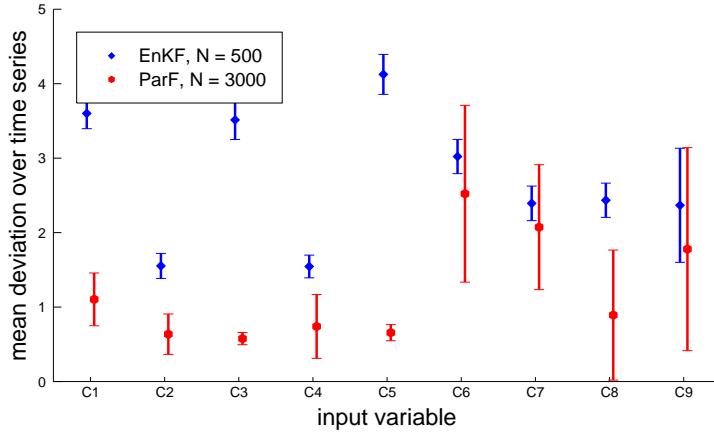


Figure 14: The mean deviation between the absolute of the solution of the Lorenz-9 system and the absolute of the results of the particle and ensemble Kalman filter for single input variables. The Lorenz System was integrated for $r = 15.1$ as SDE with a diagonal noise of $\sigma = 0.01$ and the filter updated with $dt = 0.1$, starting from random initial conditions. The standard deviation was calculated in terms of the mean over 20 simulations.

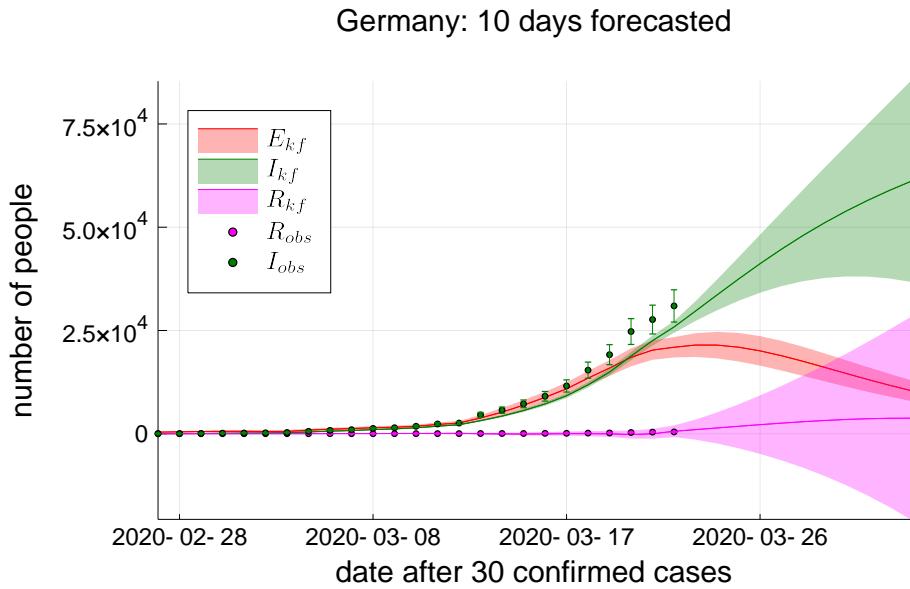


Figure 15: SARS-CoV-2 data of Anhui. E_f are the exposed, I_f the infected and R_f the recovered people, that are calculated by the Particle filter. For the observed recovered R_{obs} and infected I_{obs} stocks, data from the Johns Hopkins University [22, 23] were used.

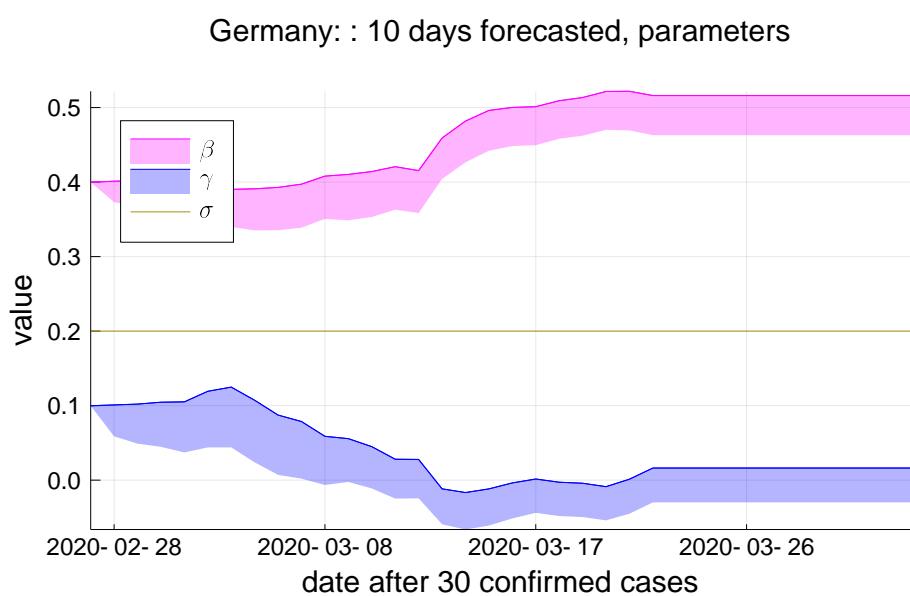


Figure 16: SARS-CoV-2 parameters of Germany. β is the infectious rate and γ the recovery rate. The incubation rate σ is assumed to be constant. The upper area of error is missing.