Wang, Dieter; Schaumburg, Julia

Working Paper

Smooth marginalized particle filters for dynamic network effect models

Tinbergen Institute Discussion Paper, No. TI 2020-023/III

Provided in Cooperation with:
Tinbergen Institute, Amsterdam and Rotterdam

Suggested Citation: Wang, Dieter; Schaumburg, Julia (2020) : Smooth marginalized particle filters for dynamic network effect models, Tinbergen Institute Discussion Paper, No. TI 2020-023/III, Tinbergen Institute, Amsterdam and Rotterdam

This Version is available at:
https://hdl.handle.net/10419/220060

TI 2020-023/III
Tinbergen Institute Discussion Paper

Smooth marginalized particle filters for dynamic network effect models

Dieter Wang[1]
Julia Schaumburg[1]

[1] Vrije Universiteit Amsterdam and Tinbergen Institute

Smooth marginalized particle filters for dynamic network effect models

Dieter Wang*[1] and Julia Schaumburg†[1]

[1]Vrije Universiteit Amsterdam and Tinbergen Institute

May 4, 2020

Abstract

We propose the dynamic network effect (DNE) model for the study of high-dimensional multivariate time series data. Cross-sectional dependencies between units are captured via one or multiple observed networks and a low-dimensional vector of latent stochastic network effects. The parameter-driven, nonlinear state-space model requires simulation-based filtering and estimation, for which we suggest to use the smooth marginalized particle filter (SMPF). In a Monte Carlo simulation study, we demonstrate the SMPF's good performance relative to benchmarks, particularly when the cross-section dimension is large and the network is dense. An empirical application on the spread of the COVID-19 pandemic through international travel networks illustrates the usefulness of our method.

Keywords: Dynamic network effects; Multiple networks; Nonlinear state-space model; Smooth marginalized particle filter; COVID-19

*E-mail: d.wang@vu.nl. Corresponding author.

†E-mail: j.schaumburg@vu.nl. Schaumburg thanks the Dutch Science Foundation (NWO, grant VENI451-15-022) for financial support.

1 Introduction

Multivariate time series models are important tools for studying and predicting the dynamic interactions between key variables and/or investigation units. Depending on the dimensionality of the data, simplifying assumptions often need to be imposed to make estimation feasible. For example, if the cross-section is large while the time series is short, dynamic panel-type approaches are typically employed, in which outcomes of units at time t are functions of their own lags, but independent from contemporaneous and lagged observations of other units. This simplification, however, can be too restrictive in many empirical settings. On the other end of the spectrum, we have (structural) vector autoregression (VAR) models, which take into account the dynamic dependence structure between the constituents, but rely on identification restrictions in order to capture cross-sectional shock spillovers, and are only feasible for a small number of units. Classic dynamic factor models (DFM) strike a balance by decomposing the multivariate dynamics into a constant cross-sectional part, the factor loadings, and a small number of time-varying factors. However, they have the drawback that interpretation of both loadings and factors is often ambiguous.

This paper discusses a class of dynamic network effect (DNE) models, which allow us to incorporate contemporaneous network dependence between units even in large cross sections, while capturing dynamics in the data at the same time. The method utilizes an underlying network that is constant or slowly varying over time. The approach is closely related to the spatial literature, but is more general as (1) network linkages are not subject to constraints of geographic distances such as symmetry or non-negativity (2) the intensity parameters are not constant but follow stochastic processes and (3) we allow for different transmission channels by incorporating more than one network. Being able to incorporate these features makes our model relevant for applications in which multivariate, cross-sectionally dependent data are observed over longer time periods, and where networks can be observed or inferred from theory. Examples include financial contagion and systemic risk (Aït-Sahalia et al., 2014; Forbes & Rigobon, 2002), comovements of business cycles (Böhm et al., 2020), and spreading of contagious diseases via travel or social networks.

The paper contributes to the recent literature on time-varying spatial dependence, see Blasques et al. (2016) and Catania and Billé (2017). However, instead of assuming score-driven dynamics, we consider an alternative specification for the intensity parameters, in which they have their own disturbances. Allowing for this more general parameter-driven formulation comes at the price of having to deal with a nonlinear state-space model, for which no closed-form likelihood is available. The built-in stochastic volatilities imposes additional demands on the estimation procedure. We propose to carry out estimation and filtering using a smooth marginalized particle filter (SMPF), which combines the smooth particle filter of Malik and Pitt (2011) and Doucet et al. (2001) with the marginalized particle filter of Casella and Robert (1996) and Andrieu and Doucet (2002). This filter type is an attractive choice due to its ability to incorporate complex nonlinearities as well as non-Gaussianities without relying on Taylor-type approximations. We can obtain arbitrarily close approximations to the nonlinearity using sequential Monte Carlo simulations, the so-called particles, that undergo the actual nonlinear transformation. Furthermore, the marginalization allows us to accurately and efficiently estimate the linear parameters of the model, such as regression coefficients with their standard errors. We illustrate the good performance of the SMPF in

Monte Carlo simulations in terms of prediction accuracy, likelihood evaluation/estimation, signal extraction and coefficient estimation. We show that in our setting, the SMPF clearly outperforms the widely used extended Kalman filter (EKF), which relies on local Taylor approximations and has limited ability to handle stochastic volatility. Within the simulations, we also investigate the impact of different network structures on data features and filtering. In particular, we investigate network asymmetries, different degrees of network sparsity, and incorporating both positive and negative spillovers.

We apply the new model and filtering method to analyze the spreading of the new SARS-CoV2 virus across borders. Infected people either travel via airplane (for longer distances) or via railways and roads (for shorter distances). Therefore, our model features two candidate networks, one of which is constructed from air travel data, while the other is an adjacency matrix where countries are connected if they share a common border. We introduce the notion of a *contagion faucet*, which is a helpful concept to understand contagion dynamics between both networks. The faucet controls the overall contagion flow through either network and also regulates flows between networks. We find strong time-variation in the filtered intensity parameters, indicating several phases of international transmission of the disease. Initally, air travel was mainly responsible for elevating the disease from an epidemic to a pandemic. Subsequently, short-distance travel became more relevant. Our results also suggest that towards the end of the sample, contagion occurs mainly within countries.

The paper is structured as follows. In Section 2 we describe the different dynamic network effects models with a single network or multiple networks. Section 3 outlines the smooth marginalized particle filter as well as the extended Kalman filter, which we use as benchmark in the simulations. Section 4 provides an analyisis of the model's behavior subject to different network types. Section 5 investigates the filtering and estimation performance of our method using simulated data. In Section 6, we illustrate our results using the current COVID-19 disease where we use airline routes and common borders as the underlying networks for long-distance and short-distance travels. Section 7 concludes.

2 Dynamic network effects (DNE) model

The general form of the DNE model is most clearly expressed as a nonlinear state-space model (Durbin & Koopman, 2012). The set of model equations is given by

$$y_t = \mathbf{\Phi}_t X_t \beta + \mathbf{\Phi}_t \mathbf{\Theta}_t e_t, \qquad e_t \sim N(0, \Sigma) \tag{1}$$

$$\mathbf{\Phi}_t = F(\phi_t; \mathbf{W}_t) \qquad \phi_t = g(x_t^1) \tag{2}$$

$$\mathbf{\Theta}_t = F(\theta_t; \mathbf{W}_t) \qquad \theta_t = g(x_t^2) \tag{3}$$

$$x_t = c + A x_{t-1} + u_t, \qquad u_t \sim N(0, \Omega). \tag{4}$$

The measurement equation for vector $y_t \in \mathbb{R}^N$ features a regression component $X_t\beta$ and an additive error term e_t, as well as two nonlinear network components $\mathbf{\Phi}_t, \mathbf{\Theta}_t$.

The functional specifications of the network transformation function F and the network effect function g depend on the type of network model and number of networks considered. Nonethe-

less, we can establish a few general properties for both functions. F is a matric-valued function that applies a vector of bounded network effects ϕ_t, θ_t to a set of network matrices $\mathbf{W}_t$. In other words, $F : U \to \mathbb{R}^{N\times N}$ with $U \subseteq (-1, 1)^m$ for $m \in \{M_1, M_2\}$ and the vector-valued transformation function $g : \mathbb{R}^m \to U$ ensures boundedness of the network effects ϕ_t, θ_t. The network matrices themselves $\mathbf{W}_t = \{W_t^k\}_{k=1,\dots,K}$ with $W_t^k \in \mathbb{R}^{N\times N}$ are normalized such that their largest, absolute eigenvalues are equal to 1 at all time points (Anselin, 1988; LeSage & Pace, 2009). We emphasize that the time-variation of the networks W_t^k should always be negligible compared to the dynamics of the network effects ϕ_t, θ_t.

The latent state vector $x_t \in \mathbb{R}^M$ with $M = M_1 + M_2$ contains both x_t^1, x_t^2 and supplies the dynamics for both network effects ϕ_t, θ_t. It follows a first-order stationary autoregressive process with constant vector c, autocorrelation matrix A and state innovation vector u_t with constant covariance matrix Ω. To ensure stability of the model, x_t undergoes a transformation $g(x_t)$ before it enters the measurement equation through the network components. Depending on the network effect model, the transformation g either represents a logistic function, a softmax function or a combination thereof.

It is important to note that while the covariance matrix Σ of e_t is constant and diagonal, the effective covariance of the measurement error $\mathbf{\Phi}_t\mathbf{\Theta}_t e_t$ is not. In fact, its covariance $\Sigma_t^* := \mathbf{\Phi}_t\mathbf{\Theta}_t\Sigma\mathbf{\Theta}_t^\top\mathbf{\Phi}_t^\top$ is full and allows for multivariate stochastic volatility. Its off-diagonal structure is informed by the network matrices W_t^k and its temporal dynamics are driven by the network effects ϕ_t, θ_t, in addition to the slow dynamics of the matrices W_t^k themselves. This architecture keeps the model parsimonious even when N is large. In the following, we describe F and g for different types of network effect models.

2.1 Structural dynamic network effects model

The first model we consider is labeled structural dynamic network effects model, as it may be interpreted as a particular version of a structural vector autoregressive (VAR) model with regressors and time-varying coefficients. In order to identify contemporaneous dependencies, structural VARs requires parameter restrictions, that are, however, subject to debate in the literature, see Lütkepohl (2005) and Kilian and Lütkepohl (2017). In the structural DNE model, simultaneous dependencies are explicitly taken care of by the observed network.

We begin the derivation of the network effects component $\mathbf{\Phi}_t = F(\phi_t; \mathbf{W}_t)$ in (2) with an explicit formulation where the dependent variable y_t appears on both sides of the equation. This reflects the fundamental idea of network effects, namely that observed outcomes are not only functions of their own shocks and determinants but also of their neighbors. Who and how relevant these neighbors are is determined by the network. We assume that contemporaneous spillovers are captured by a single network matrix W_t.[1] We have

$$y_t = \phi_t W y_t + X_t\beta + e_t$$

with $\phi_t = g(x_t^1)$, where is the logistic transformation. The univariate latent state x_t^1 follows an autoregressive process of order one as in (4). Furthermore, $\mathbb{E}[e_t] = 0$, and $\text{Var}(e_t) = \Sigma$, which

[1]For clarity of notation, we suppress the time subscript t.

is diagonal with $\Sigma = \text{diag}(\sigma_1^2, ..., \sigma_N^2)$. Similar to autoregressive time-series models, repeated substitution reveals that under the stability condition on ϕ_t and W_t, i.e. shocks to the errors and regressors die out eventually and we can obtain a steady state/reduced form version of the model as

$$\begin{aligned} y_t &= \phi_t W y_t + X_t\beta + e_t \\ &= \phi_t W[\phi_t W y_t + X_t\beta + e_t] + X_t\beta + e_t \\ &= \phi_t W[\phi_t W[\phi_t W y_t + X_t\beta + e_t] + X_t\beta + e_t] + X_t\beta + e_t \\ &= \cdots \\ &= (I_N - \phi_t W)^{-1} X_t\beta + (I_N - \phi_t W)^{-1} e_t. \end{aligned} \tag{5}$$

Therefore, the nonlinear state-space formulation of the model is

$$y_t = \Phi_t X_t\beta + \Phi_t e_t, \qquad e_t \sim N(0, \Sigma) \tag{6}$$

$$\Phi_t = (I_N - \phi_t W)^{-1} \qquad \phi_t = g(x_t^1) \tag{7}$$

We refer to the parameter ϕ_t as *structural network effect*. Note that already in this simple formulation, the model contains multivariate stochastic volatility (MSV). However, the time-varying covariance is coupled with the transformation of the regression component. For many applications, it is difficult to justify why the amplification effect in the regression component should follow the same dynamics as the error covariance. To disentangle both effects, we model the error network effects separately.

2.2 Error network effects

In this specification, contemporaneous network spillovers only occur among the disturbances. Therefore, we label this model error dynamic network effects model. The conditional mean equation remains linear, corresponding to model (1) with $\mathbf{\Phi}_t = I_N$. However, the covariance matrix has a dynamic network effect θ_t, which introduces stochastic volatility. Again, we assume one network matrix W and a scalar network effect. We have

$$y_t = X_t\beta + \eta_t \tag{8}$$

$$\eta_t = \theta_t W \eta_t + e_t, \qquad e_t \sim N(0, \Sigma), \tag{9}$$

where Σ is diagonal as before. Equivalently, we express the model in reduced form as

$$y_t = X_t\beta + \Theta_t e_t \tag{10}$$

$$\Theta_t = (I_N - \theta_t W)^{-1}, \qquad \theta_t = g(x_t^2). \tag{11}$$

We refer to the parameter θ_t as *error network effect*. This linear model with multivariate stochastic volatility is a restricted version of the model in Harvey et al. (1994). Instead of a multiplicative factor h_{it} for each error variable e_i, the temporal dynamics are contained in θ_t in our case. Furthermore, Harvey et al. (1994) models the variances as stochastic process, but restricts the off-diagonal covariances to zero. In our model, the covariance structure is informed by the cross-sectional

network structure in the matrix W and, ultimately, the transformation Θ_t.

2.3 Generalized network effects

The generalised network effects model presented here is simply a synthesis of the structural and error network effects models introduced before. As before, we assume that cross-sectional dependence is captured by a single network matrix W, but that spillovers from shocks to the regressors can differ from shock spillovers among the disturbances. In explicit form, the model is given by

$$y_t = \phi_t W y_t + X_t \beta + \eta_t \tag{12}$$

$$\eta_t = \theta_t W \eta_t + e_t \tag{13}$$

with ϕ_t, θ_t as above. In the nonlinear state-space form, the measurement equation corresponds to

$$y_t = \Phi_t X_t \beta + \Phi_t \Theta_t e_t, \qquad e_t \sim N(0, \Sigma) \tag{14}$$

The model contains both structural and error components which allows us to isolate them from each other. We note that since both network effects rely on the same matrix W, the two corresponding effects ϕ_t, θ_t are only identified if the exogenous variables contribute sufficient variation, i.e. $\beta \neq 0$ (Kelejian & Prucha, 2010). It is certainly possible to introduce different networks for structural and error network effect components.

Offsetting effects in the errors As both components interact in the disturbances, it may be tempting to assume that whenever $\phi_t = -\theta_t$, both effects offset each other, for the same network W. However, this is not the case since it does not imply that $\Phi_t = \Theta_t^{-1}$, which would cancel each other out and eliminate any network effects in the error term. Instead, for $\phi_t = -\theta_t$, we observe attenuated effects of the network effect with larger absolute magnitude (see Figure 2c).[2] To achieve an offsetting effect, we require a relationship $\theta_t = -\phi_t/(1-\phi_t)$, provided no invertibility constraints are violated.

2.4 Generalized network effects with multiple networks

In some empirical applications, several networks may be relevant and the model should allow to take them into account. In fact, the interplay between multiple networks may be crucial to disentangle spillover effects and dynamics in the data. For instance, when a virus spreads through transportation networks, the spread through railway transportation are likely to be different from air travel dynamics. Restricting the model to capture only one of the networks would lead to misleading results. The generalized DNE model that allows for K networks can be expressed as

$$y_t = \phi_t \sum_{i=1}^{K} \omega_t^k W^k y_t + X_t \beta + \eta_t, \qquad \eta_t = \theta_t \sum_{i=1}^{K} \nu_t^k W^k \eta_t + e_t \tag{15}$$

[2] To illustrate, if $W = I$ then the largest eigenvalue of Φ is $\lambda(\phi) = 1/(1-\phi)$ and similarly for Θ. For $\phi = -\theta$ and $\phi > 0$ it follows that $\lambda(\phi) > 1 > \lambda(\theta) > 0$.

or in reduced form

$$y_t = \widetilde{\Phi}_t X_t \beta + \widetilde{\Phi}_t \widetilde{\Theta}_t e_t, \qquad e_t \sim N(0, \Sigma) \tag{16}$$

$$\widetilde{\Phi}_t = \left(I_N - \phi_t \sum_k \omega_t^k W^k \right)^{-1} \tag{17}$$

$$\widetilde{\Theta}_t = \left(I_N - \theta_t \sum_k \nu_t^k W^k \right)^{-1} \tag{18}$$

In this model, the state vectors x_t^1, x_t^2 have a distinct structure. For the structural network component, the first element of x_t^1 is a scalar $\overline{x}_t^1$ which determines ϕ_t through a logistic transformation, as before. The remaining $m-1$ elements $\underline{x}_t^1$ determine the weighting parameters $\omega_t = G(\underline{x}_t^1)$ through a softmax transformation $G : \mathbb{R}^{m-1} \to \Delta^{m-1}$, a unit simplex with $m-1$ dimensions. This ensures invertibility, since $\sum_k \omega_t^k = 1$ and $\omega_t^k \in (0,1)$ for all ω_t^k in ω_t. This holds analogously for x_t^2 and the error network effect components θ_t, ν_t^k.

To facilitate interpretation and comparison, we define $\phi_t^k = \phi_t \omega_t^k$ which are the *effective structural network effects* associated with network W^k. Similarly, $\theta_t^k = \theta_t \nu_t^k$ are the *effective error network effects* associated with network W^k.

Spillover faucet In case of $M = 2$, equation (15) can be intuitively understood by picturing a water faucet, specifically, either a single-level faucet (Parkison, 1973) or a ball faucet (Tang, 1998), which we often find in kitchens. Assuming only two networks, the measurement equation becomes

$$y_t = \phi_t \left[\omega_t W_1 + (1 - \omega_t) W_2 \right] y_t + X_t \beta + u_t \tag{19}$$

In this case, ϕ_t measures the overall water flow (faucet turns upwards or downwards) and ω_t regulates the shares of hot and cold water (faucet turns sideways). While water faucets are hard to generalize to more than two temperatures, the concept can be generalized for K networks as described in (15). The same concept can be analogously applied to the error network component.

Multiple networks are a useful extension for applied cases, where the observed variables are often the result of an interplay between several network effects. In our illustration, we apply this idea to a "contagion faucet" that diverts overall contagion flow through either long-distance or short-distance travel networks.

3 Signal extraction

As laid out in the previous section, the DNE model can be represented as a nonlinear state-space model. Due to the nonlinearity, which affects both the conditional mean and the variance, we cannot use the Kalman filter but have to turn to either approximation methods or simulation-based approaches (see Part II of Durbin and Koopman (2012)). In the following, we describe the smooth marginalized particle filter (SMPF), which combines the particle filter's ability to tackle any complex nonlinearity, with accurate and efficient estimation of all static parameters, even in

high dimensions. In our simulations, we benchmark the SMPF's performance against the most widely used approximation method, the extended Kalman filter (EKF). Therefore, this section also contains the derivation of the relevant model-specific derivatives for the EKF.

3.1 Smooth marginalized particle filter

The primary goal is to filter out the unobserved state vector $\{x_t\}_{t=1,\dots,T}$, see equation (4), which we assume to follow a first-order Markov process $x_t \sim p(x_t|x_{t-1})$. We infer the value of the latent state based on the observed variables $\{y_t\}_{t=1,\dots,T}$ that are generated from the state according to $y_t \sim p(y_t|x_t)$. The particle filter proceeds by simulating a set $s =, 1\dots, S$ of potential states, so-called particles, $x_t^{(s)} \sim h(x_t|x_{t-1})$. Here h is a proposal density that approximates the actual density $p(x_t|x_{t-1})$ and is easy to sample from. Then, for each particle we assess how likely it occurs for the observed y_t, i.e. we calculate the likelihood of observing y_t for a given particle. In the bootstrap particle filter, this likelihood is also called importance weight.

Our estimate of the state x_t is then simply an average of all particles, weighted by their importance weights, i.e. $\hat{x}_t = \sum_s x_t^{(s)} w_t^{(s)}$. Before we can repeat this procedure for $t+1$, we need to resample our particles. This resampling combats particle degeneracy, where only few particles determine the trajectory of the entire particle swarm (Kantas et al., 2015). Different resampling strategies have been proposed by the literature. We opt for the stratified strategy (Hol et al., 2006).

Smooth particle filter Beside the latent state x_t, the DNE model in state-space form has several static parameters, which we collect in the vector ψ.[3] Maximum likelihood estimation of ψ using particle filters is known to be challenging. The likelihood estimates are not continuous in ψ, which can be traced back to the fact that particles $\{x_t^{(s)}\}_{s=1,\dots,S}$ are path dependent (Kantas et al., 2015). Small changes $\psi + d$ may lead to proposal particles $x_{t+1}^{(s),d} \sim p(x_{t+1}|x_t; \psi + d)$ that are slightly different than without the change $x_{t+1}^{(s)} \sim p(x_{t+1}|x_t; \psi)$. However, the associated importance weights $w_s = p(y_{t+1}|x_{t+1}^{(s)})$ and $w_s^d = p(y_{t+1}|x_{t+1}^{(s),d})$ will then be different as well. In the resampling step, we draw particles $x_t^{(s)}$ from the empirical distribution function (EDF) of existing particles, yielding a step function. In the case of simple resampling, we simulate $u \sim U[0,1]$ and evaluate $x = F^{-1}(u)$ for each particle. Hence, it is conceivable that we draw a u which is close to the edge of a step in the EDF $\hat{F}(x)$ (see Figure 10). In such a case, the resulting particle could belong to $x_t^{(s)}$, under ψ and to $x_t^{(s+1),d}$, under $\psi + d$. Such a bifurcation results in all future particles to follow entirely different paths, even if d is small. The resulting discontinuous, discrete likelihood functions in ψ constitute a major challenge for gradient-based optimization methods.

Malik and Pitt (2011) propose the *smooth particle filter* as a simple and elegant way to deal with this problem. Instead of resampling from a discrete step EDF $\hat{F}(x)$, the authors propose a smooth EDF $\tilde{F}(x)$ which is simply a linear interpolation of $\hat{F}(x)$. This ensures that small changes $\theta + d$ will indeed lead to small changes between resampled $\tilde{x}_{t+1}^{(s)}$ and $\tilde{x}_{t+1}^{(s),d}$, see Figure 10 for a visual representation. The resulting likelihood estimate loses the undesirable discontinuities, as shown in Figure 11, and can be used in gradient-based optimization routines. For the models studied below, we find that only $S = 50$ particles are sufficient to obtain good convergence results.

[3] Depending on the model version, ψ contains the elements in the autoregressive parameter matrix A in the state equation, the state covariance matrix Ω and the measurement variances $\sigma_1^2, \dots, \sigma_N^2$. The regression coefficient vector β could be part of ψ but we advocate treating them as linear state variables, see below.

Marginalized particle filter To distinguish between nonlinear (a_t) and linear states b_t, we partition the state vector into $x_t = [a_t, b_t]^\top$. A conventional (smooth) particle filter carries out the estimation of regression coefficients β by treating b_t as part of the state vector, and filtering them in the same way as the nonlinear states a_t. That is, the updating relies on the likelihood density $p(a_t, b_t|y_t)$. However, this is computationally expensive and, more importantly, statistically inefficient.[4] Separating the linear from the nonlinear state variables yields significant computational gains. This process of marginalizing out the linear states is also known as Rao-Blackwellization. The likelihood density is evaluated in two parts, $p(a_t, b_t|y_t) = p(b_t|a_t, y_t)p(a_t|y_t)$. Using the particle filter steps only for the nonlinear states and the Kalman filter steps for linear states drastically reduces the computational complexity. The resulting model is known as the *marginalized particle filter* or Rao-Blackwellized particle filter (Andrieu & Doucet, 2002; Doucet et al., 2000; Schon et al., 2005; Schon et al., 2006).

3.2 Benchmark: Extended Kalman filter

For approximate filtering, the EKF requires the first-order derivative of the measurement function $\mu(x) = \mathbb{E}[y|x]$. For clarity, we suppress the time subscript t. We derive the Jacobian for the general case with regression effects, such that the state vector x is partitioned into a for the nonlinear and b for the linear state variables.

$$D_x\mu = [D_a\mu, D_b\mu] = \left[\frac{\partial\mu(x)}{\partial a}, \frac{\partial\mu(x)}{\partial b}\right]$$

The nonlinear state variable a is further subdivided into a_1, a_2, which correspond to the structural and error network effects ϕ, θ. The first part of the derivative depends on the derivative of the network transformation with respect to a, in fact only a_1, through the logistic function g

$$D_a\mu = \frac{\partial\mu(x)}{\partial(a_1, a_2)} = \left[\frac{\partial\Phi(a_1)}{\partial a_1}Xb, 0\right]$$

The derivative of the structural network component Φ with respect to a_1 is

$$\frac{\partial\Phi(a)}{\partial a_1} = \frac{\partial A^{-1}}{\partial a_1} = -A^{-1}\frac{\partial A}{\partial a_1}A^{-1}$$

where $A = I_N - g(a_1)W$ and its derivative $\partial A/\partial a_1 = -g'(a_1)W$. We use $g(x) = e^x(e^x+1)^{-1}$ with the first derivative $g'(x) = e^{-x}(e^{-x}+1)^{-2}$. The second part of the derivative then follows immediately

$$D_b\mu = \frac{\partial\mu(x)}{\partial b} = \frac{\partial\Phi(a_1)Xb}{\partial b} = \Phi(a_1)X$$

Hence, the Jacobian matrix has dimensions $N \times (M_a + M_b)$

$$D_x\mu = \left[g'(a_1)\Phi W\Phi Xb,\ 0,\ \Phi X\right]$$

[4]Instead of treating them as state variables, $b_t \sim p(b_t|b_{t-1})$, one could estimate them as additional hyperparameters in ψ. However, this approach may become infeasible if the number of regression coefficients increases.

With these derivatives we can implement the EKF using adjusted Kalman filter recursions (Durbin & Koopman, 2012) and evaluate the likelihood straightforwardly using prediction error decomposition (Harvey, 1990).

4 Model behavior

In this section we analyze properties of the DNE model by looking at different combinations of network effects ϕ_t, θ_t for the following two network types. We use simplified DNE models that abstract from regression components but still retain network components of interest. The cross-section is set to $N = 50$.

Circular network W_C The circular network (Figure 1a) is a useful benchmark, already employed by Ord (1975) in the study of spatial autoregressive models, due to its simplicity, irreducibility and scalability. The graph is strongly connected and is defined as $W_C = [w_{ij}] \in \mathbb{R}^{N \times N}$ where the typical element $w_{ij} = 1$ if $i = j - 1$ or $i = j + 1 - N$, which ensures a closed circle. Its simplicity makes the effects of ϕ_t, θ_t clearly visible and yields interesting parallels to the autoregressive time-series models.

Random network W_{ER} We also consider directed Erdős–Rényi networks, W_{ER} (Figure 1b) which are closer to observed networks. Such networks have been widely studied due to their stochastic nature and known limiting properties. The underlying graph $\mathbb{D}(N, p)$ is often referred to as a binomial random directed graph and is closely related to the Erdős–Rényi model (Graham & Pike, 2008). The N^2 possible edges of the random graph $\mathbb{D}(N, p)$ are each drawn with probability p. Similar to the Erdős–Rényi model, the graph is almost surely strongly connected, if $p > \ln(n)/n$. Figure 1c represents a special case where the network is partitioned into two subnetworks, that will become relevant for the study of common factors.

4.1 Measurement behavior

In this part, the object of interest is the outcome or measurement variable y_t. For these purposes, it is sufficient to abstract from the time dimension, such that $y_t = y$, and use a reduced version of the generalized network effect model that only contains a constant

$$y = \phi W y + c + e \qquad e = \theta W e + u$$

To keep the simulation results tractable, we restrict our attention to the circular network W_C and linearly increasing or decreasing network effects ϕ, θ.[5]

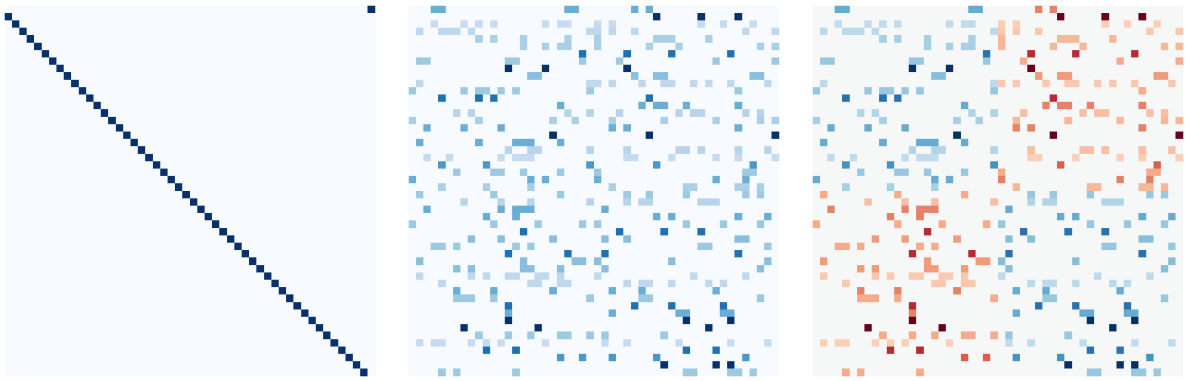
Behavior for $y = \Phi c$**, constant only** In the deterministic case in Figure 2a, we consider a model that only contains a constant with $c_i = (i - N/2)/N$. A positive structural network effect $\phi > 0$ leads to a shift of the distribution of y, according to the direction of the circle in W_C. Furthermore, the distribution of constants fans out and the values are amplified through the transformation. A negative $\phi < 0$, in contrast, reduces the distance between the measurements but does not entirely

[5]The results for the random network W_{ER} are less insightful in the static case here.

Figure 1: Network types

This figure visually represents the three network types we study here. From left to right, the circular network, the random network (or Erdős–Rényi network), and a split random network. The latter has positive network linkages (blue) on the main block diagonal and negative linkages on the off-diagonal (red). N is set to 50.

(a) Circular network W_C **(b) Random network W_{ER}** **(c) Random network W_{ER}^{split}**



collapse them. Both effects lead to asymmetric measurement distributions, but preserve the mean at zero.

Behavior for $y = \Theta e$, error only, single network effect In the error only model, Figure 2b, we can isolate the effect the network transformation has on Gaussian noise processes. In the case of positive effects, $\theta > 0$, increasing the network effect parameter leads to higher concentration and a lashing out of y. Hence, the average $\bar{y}$ oscillates unstably. In the opposite case, $\theta < 0$, decreasing the network effect parameter leads to lower concentration and y are fanning out symmetrically. Hence, the average $\bar{y}$ is centered around zero. The variance increases in both cases, differing by whether or not they are skewed.[6]

Behavior for $y = \Phi\Theta e$, error only, both network effects When both effects are present, we observe compounding and offsetting effects. For $\phi > 0, \theta > 0$, we can see the combined effect of $\theta > 0$ in the single network case, with the same asymmetry and lashing out, but stronger. Similarly, $\phi < 0, \theta < 0$ has the combined effect of $\theta < 0$ in the single network case, which leads to even more dispersion but maintaining the symmetry. The cases with opposite signs are identical to each other by construction, $\phi > 0, \theta < 0$ and $\phi < 0, \theta > 0$. They represent the attenuated combinations of the two isolated cases, $\theta < 0$ and $\theta > 0$ in the single network case.

4.2 Common factors behavior

In this part we reintroduce the time dimension and shift our attention to common factors contained in y_t. We restrict our attention to $\phi_t \neq 0$ and set $\theta_t = 0$. We demonstrate how networks yield interesting patterns in the principal components of y_t.

[6]It is instructive to draw the parallel to AR(1) processes over time. The value of a particular y_{it} we observe for a given θ resembles the outcome of an AR(1) process, with autocorrelation coefficient θ over N periods. Strong, positive autocorrelation values lead to persistent deviations from the mean. Strong, negative autocorrelation values lead to oscillations around the mean.

For a circular network, Figure 3a, we observe a curious behavior in the residuals. This unmistakable wavelet character is the result of the spatial autoregressive dynamics we impose on the Gaussian noise process e_t. Essentially, what we observe over the cross-section over different components is the spectral decomposition of the $\Theta_t e_t$ process over different frequencies.

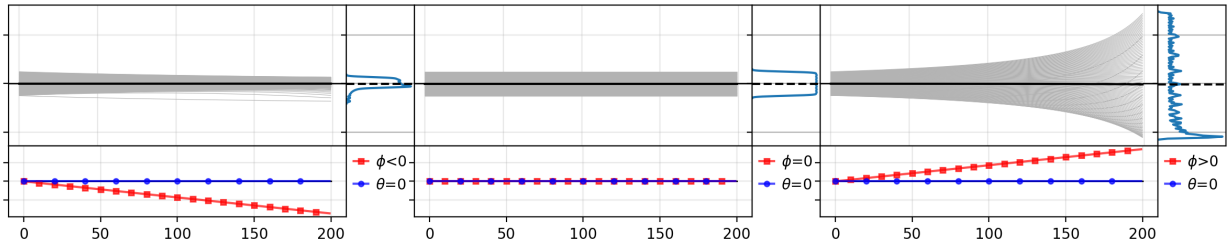
Given the random network, we can observe a common factor that affects all outcome variables y_t (Figure 3b). More interesting is the random network with positive and negative quadrants (see Figure 1c), which can be constructed using (15) with the $W_{ER} = W^1 = W^2$ and two positive and negative network effects ϕ_t^1, ϕ_t^2. The positive and negative network linkages translate into two groups of the first principal component, designated by the positive and negative coefficients (Figure 3c).[7] More involved patterns can be constructed in a similar fashion using the DNE model with multiple networks (15).

[7]Such a pattern can be seen in the case of credit contagion among European banks studied in Wang et al. (2019).

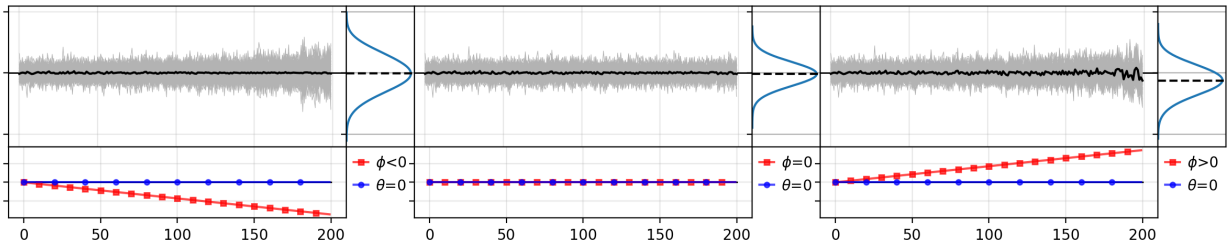
Figure 2: Behavior of the network effect model under a circular network W_C

We examine the resulting y for the reduced generalized network effects model $y = \phi W_C y + c + \eta$ and $\eta = \theta W \eta + e$ by plotting the y_i for each combination of (ϕ, θ). The black line represents the average $\overline{y}$. The density plot at the right side of each panel is a kernel density estimate of the y_i for the largest values of ϕ, θ

(a) Behavior for $y = \Phi c$, constant only



(b) Behavior for $y = \Theta e$, error only and single network effect



(c) Behavior for $y = \Phi\Theta e$, error only and both network effects

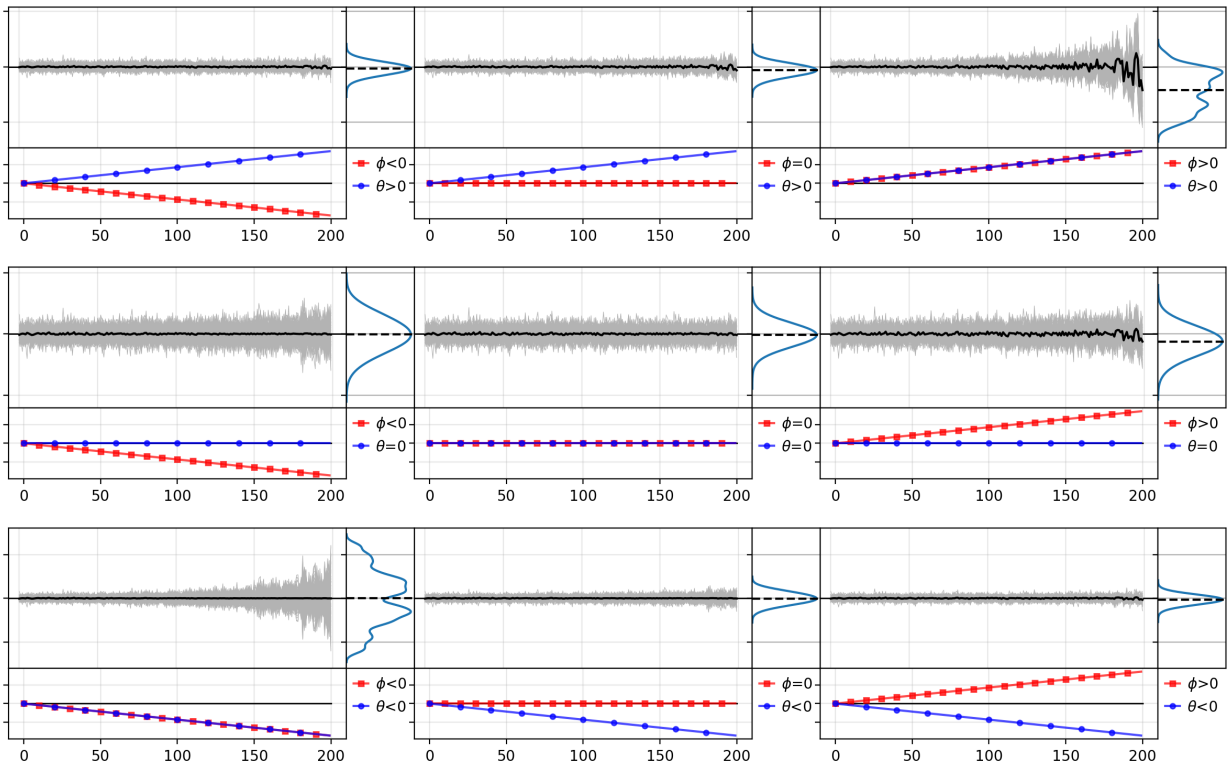
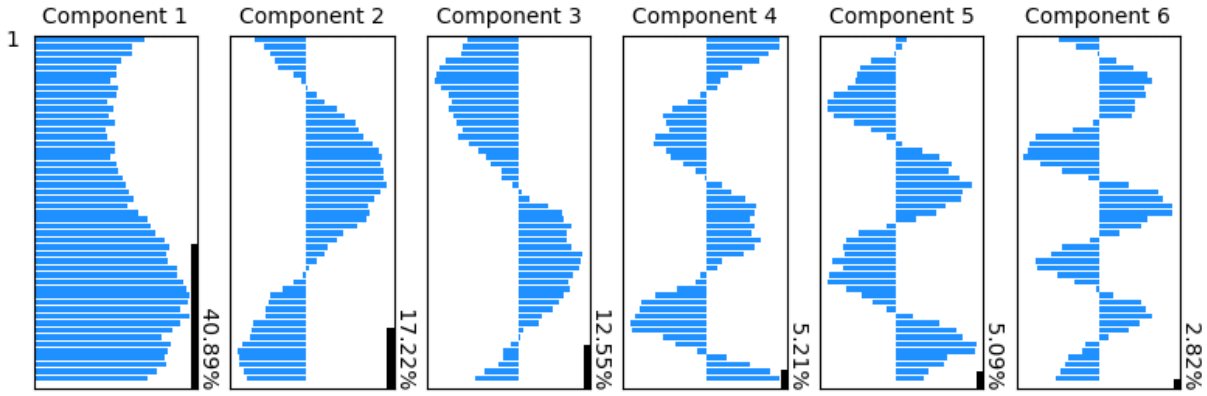


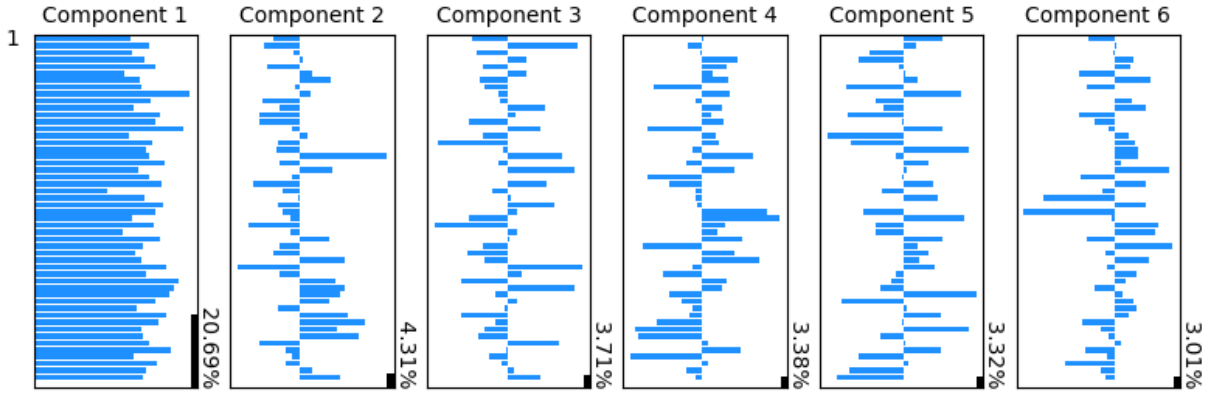
Figure 3: Principal componenents of y_t for different network types

We examine the first principal components of y_t for the reduced generalized network effects model. The vertical bars depict the share of explained variances in descending order. In the last panel, the colors correspond to the negative and positive groups in Figure 1c.

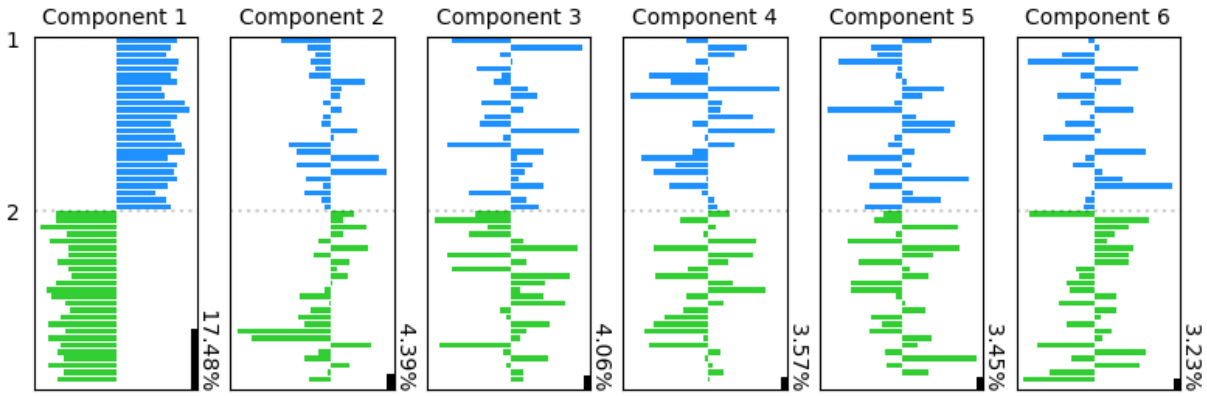
(a) Circular network W_C



(b) Random network W_{ER}



(c) Random network W_{ER}^{split}



5 Monte Carlo study

In this section we investigate the performance of the smooth marginalized particle filter (SMPF) in filtering unobserved time-varying network intensities and estimating the vector of static coefficients. We consider all network effects introduced in Section 2, namely structural, error and generalised network effects. We analyze four different network types for each network effect, circular as well as sparse, normal and dense random networks as it is conceivable that the connectedness of a network will influence an estimator's performance.

5.1 Data-generating process and network inputs

The simulation study uses equation (12) as the data-generating process (DGP). We simulate data for $T = 200$ and $N = 50, 100, 200$. The measurement variance is set to $\Sigma = 0.1\, I_N$. The regressor matrix X_t consists of a constant $X_{it}^1 = 1$ and a Gaussian noise process $X_{it}^2 \sim N(0,1)$ for all i, t. The associated coefficients are individual-specific and defined as $\alpha_i = \beta_i = i/N$ for $i = 1, ..., N$. The unrestricted state variables x_t^1, x_t^2 follow the stationary process described in equation (4) with no intercepts, $c_1 = c_2 = 0$, autocorrelation matrix $A = 0.8 I_2$ and covariance matrix variances $\Omega = 0.4 I_2$. To obtain the structural network effects model, we set $A_{22} = 0$ and $\Omega_{22} = 0$. For the error network effects model, we set $A_{11} = 0$ and $\Omega_{11} = 0$, accordingly.

All networks we have the property that they can be generated for different numbers of cross-sectional units.[8] As explained in the previous section, we control the density of the random networks through the edge probability parameter p. The network W_{ER}^{τ} represents the threshold case where the probability of two nodes being connected is $p = \tau = ln(N)/N$. Erdős and Rényi (1960) show that the network is almost surely connected for values above τ and disconnected for values below it. We choose $p = 2\tau$ for W_{ER}^{dense} and $p = \tau/2$ for W_{ER}^{sparse} as edge probabilities. Table 1 summarizes the network properties.

5.2 Computer specifications and programming language

The computations for the Monte Carlo study and the subsequent illustration were carried out on a Linux 64-bit server with Intel(R) Xeon(R) E5-2690 32-core processsor @ 2.90GHz and 128 GB memory. The methods described here were implemented in Python 3.7 and for random number generation we used the `numpy.random.RandomState` module (version 1.14) which employs the Mersenne Twister pseudo-random number generator. See Figure 12 (Appendix) for an overview of the computational time for different filters.

5.3 Results

Tables 2-4 show the results of our simulations. The number of replications is 1,000. In all cases, we use the median as a robust measure for the average. Due to the persistent auto-correlation in the network states it can happen that network effects approach 1, which results in diverging amplification effects (see equation (5)). For the same reason, we use the median absolute deviation to measure dispersion in Table 5.

[8] This is not possible or other network types, such as the core periphery network.

Table 1: Network properties

This table displays the properties of the four networks considered in the Monte Carlo study. The circular network W_C is deterministic, hence the edge probability is simply $1/N$ and the number of edges is N. For the Erdős–Rényi networks, the edge probability p is a function of $\tau = ln(N)/N$. We set $p = \tau/2$ to generate W_{ER}^{sparse}, $p = \tau$ for W_{ER}^{τ} and $p = 2\tau$ for W_{ER}^{dense}. In every random network, the expected edge number is $\binom{N}{2}p$.

	N=50				N=100				N=200			
Network property	W_C	W_{ER}^{sparse}	W_{ER}^{τ}	W_{ER}^{dense}	W_C	W_{ER}^{sparse}	W_{ER}^{τ}	W_{ER}^{dense}	W_C	W_{ER}^{sparse}	W_{ER}^{τ}	W_{ER}^{dense}
Edge probability	0.0200	0.0391	0.0782	0.1565	0.0100	0.0230	0.0461	0.0921	0.0050	0.0132	0.0265	0.0530
(Expected) number of edges	50	47.9	95.8	191.7	100	114.0	228.0	455.9	200	263.6	527.2	1054.4

The smooth marginalized particle filter (SMPF) outperforms the Extended Kalman Filter (EKF) in almost every case. However, there are exceptions where the EKF is better or is it least on par. In terms of RMSE of the predicted variable, the EKF is more accurate for sparse networks like W_C or W_{ER}^{sparse}. The improvement over the SMPFs decreases as the network becomes more dense. This is because the first order approximation used in the EKF is more accurate for sparse networks than for dense networks. At the same time, the approximation performed in the SMPF leads to more accurate results as the network density increases.

When considering the average log-likelihood, the SMPF outperforms the EKF except for the cases where $N = 50$.[9,10] We observe that models with the lowest RMSE do not always have the highest log-likelihoods. This is because the RMSE only considers the prediction error while the log-likelihood also depends on the prediction error variance. Linear approximation errors that occur in the EKF enter the likelihood in a quadratic form. The SMPF, in contrast, does not depend on Taylor approximation errors to the nonlinearity as it evaluates the network transformation exactly.

In Table 3, we consider the ability of the filters to accurately estimate the underlying network effects. Different from RMSE of y_t, the SMPF strictly outperforms the EKF.[11] On average the SMPF with 100 particles reduces the RMSE by more than half compared to the EKF. While more particles generally lead to more accurate results for both x_t^1, x_t^2, the improvement is most striking in the structural network effects. For the error network effects, we only find slight improvements. As expected, the denser the networks are, the more challenging it is for the filters to accurately estimate x_t^1, x_t^2.

Table 4 also demonstrates the superior estimation performance of the SMPF. The superiority is most notable for the structural network effects with a circular network, where the SMPF(100) reduce the RMSE of the EKF by almost 90 percentage points. As before, the estimation performance deteriorates with increasing network density. For the error network effects, we do not expect any differences since the regression coefficients are not affected by the stochastic volatility. Both filters can estimate the constant more accurately. $\hat{\beta}$ estimates are almost indistinguishable across models and filters.

[9] Note that the log-likelihood values calculated via the EKF are exact while particle filters only provide estimates. They are directly comparable only asymptotically for the number of particles tending to infinity.

[10] In Table 2, the EKF log-likelihoods for the error network effects cases are almost identical because the EKF cannot identify x_t^2, which drives the error network effects.

[11] Since the structural and error network effects only contain one of the two network effects we remove the corresponding segments in the table. Furthermore, since the EKF is unable to identify the error network effect we remove these cases in the Table accordingly.

Finally, Table 5 compares the dispersion of the log-likelihood values of each filter across all simulations. We find that the log-likelihood values estimated with the SMPF are less dispersed across simulations than the EKF. The error network effect model is the exception, where for large, dense cross-sections, the EKF has lower dispersion. However, in conjunction with the low average log-likelihoods and high RMSE for the same cases (Table 2), this implies that the EKF is inaccurate in the same way across all simulations. In contrast, the SMPF values are less dispersed than the EKF in all cases and more particles reduce the amount of dispersion. This is relevant for parameter estimation, particularly for gradient-based optimization procedures.

To summarize our findings, the SMPF strictly outperforms the EKF in terms of signal extraction (x_t^1, x_t^2) and regression coefficient estimation (α_i, β_i). For the in-sample prediction performance of the outcome variable y_{it} the results are not as clear cut. For sparse networks and small cross-sections the linear approximation in the EKF is it sufficiently accurate. For denser networks and larger cross-sections, the prediction errors the approximation error in the EKF weigh down its performance, while the SMPF yields higher likelihoods and more precise estimates. A relatively small number of particles is sufficient to obtain good estimates of the measurement variable y_t. For signal extraction and coefficient estimation, increasing the number of particles leads to major improvements. However, this comes at a cost of computational power. On average, for $N = 200$ the SMPF with 100 particles required about 316.4 seconds per run, 88.4 seconds for 25 particles, while the EKF only needed 11.2 seconds. For more details, see Figure 12, Appendix.

Table 2: RMSE of $\hat{y}_t$ and average log-likelihoods

Monte Carlo results (1 of 4): For each filter we compare their performance in predicting the outcome variable y_t and evaluate their log-likelihoods for three network effect models (structural, error and generalized), four networks (circular network, and sparse, normal, dense random networks) and three different cross-sections $N = 50, 100, 200$.
We find that the SMPF performs better in most cases. Except for sparse matrices (W_C, W_{ER}^{sparse}) where the EKF produces lower RMSE. For more dense matrices, the SMPFs perform better. The differences between SMPFs with 25 or 50 particles are negligible. In terms of average log-likelihood, the SMPFs achieve higher values for larger cross-sections ($N = 100, 200$). This is likely due to the error component, that the EKF only models indirectly.

Network			Average log-likelihood				RMSE of y			
Effect	Type	N	EKF	SMPF(25)	SMPF(50)	SMPF(100)	EKF	SMPF(25)	SMPF(50)	SMPF(100)
Structural	W_C	50	-1.0982	1.2177	2.0230	• 2.3181	• 1.1496	1.2532	1.2608	1.2521
		100	0.7539	13.7004	14.4925	• 14.8684	• 1.1653	1.2793	1.2715	1.2753
		200	2.3980	14.7431	15.6215	• 15.9880	• 1.1875	1.2847	1.2818	1.2934
	W_{ER}^{sparse}	50	• 6.1191	2.4978	2.8364	2.9115	• 0.9964	1.0594	1.0583	1.0601
		100	7.2617	14.7781	15.1337	• 15.2702	• 1.0106	1.0722	1.0726	1.0739
		200	8.1912	15.6293	16.0408	• 16.2209	• 1.0190	1.0796	1.0876	1.0853
	W_{ER}^{τ}	50	• 5.2338	2.0917	2.5890	2.7175	1.0986	• 1.0758	1.0843	1.0800
		100	7.7987	14.6840	15.0934	• 15.2461	1.0641	1.0587	1.0569	• 1.0546
		200	9.1578	15.6919	16.1224	• 16.3105	1.1147	1.0515	• 1.0488	1.0525
	W_{ER}^{dense}	50	• 6.5806	2.4181	2.8813	2.9729	1.1363	• 1.0409	1.0460	1.0446
		100	8.6650	14.7826	15.2443	• 15.3699	1.1408	1.0374	1.0355	• 1.0335
		200	9.9112	15.8676	16.2587	• 16.4108	1.2015	1.0313	• 1.0272	1.0328
Error	W_C	50	• 10.4451	5.6663	5.6655	5.6624	0.9323	0.9285	• 0.9284	0.9286
		100	10.4726	16.5399	16.5420	• 16.5448	0.9386	0.9360	0.9358	• 0.9357
		200	10.4187	• 16.6546	16.6510	16.6511	0.9415	• 0.9372	0.9374	0.9373
	W_{ER}^{sparse}	50	• 11.4386	6.3445	6.3437	6.3365	0.9321	0.9315	• 0.9312	0.9314
		100	11.4340	17.1833	• 17.1851	17.1816	0.9361	0.9351	0.9351	• 0.9350
		200	11.5054	• 17.2967	17.2939	17.2919	0.9400	0.9394	• 0.9394	0.9394
	W_{ER}^{τ}	50	• 11.5417	6.4715	6.4658	6.4603	0.9319	0.9314	0.9313	• 0.9313
		100	11.6842	17.4079	• 17.4087	17.4043	0.9363	• 0.9359	0.9359	0.9359
		200	11.7523	• 17.5658	17.5634	17.5609	0.9398	0.9396	• 0.9395	0.9396
	W_{ER}^{dense}	50	• 11.7733	6.7173	6.7159	6.7143	0.9315	0.9312	0.9311	• 0.9311
		100	11.8499	• 17.6145	17.6123	17.6142	0.9363	• 0.9361	0.9361	0.9361
		200	11.8827	• 17.7190	17.7185	17.7183	0.9396	0.9395	• 0.9395	0.9395
Generalized	W_C	50	-3.5417	0.2288	1.1335	• 1.3067	• 1.1596	1.2789	1.2722	1.2761
		100	-1.5715	12.7468	13.4887	• 13.8819	• 1.1774	1.2935	1.2953	1.2993
		200	-0.2021	13.7391	14.7108	• 15.0418	• 1.1964	1.3068	1.3091	1.3156
	W_{ER}^{sparse}	50	• 5.5617	2.0334	2.3491	2.4049	• 0.9989	1.0646	1.0609	1.0655
		100	6.6910	14.3462	14.6835	• 14.8093	• 1.0125	1.0716	1.0722	1.0746
		200	7.6382	15.2622	15.6415	• 15.7899	• 1.0217	1.0884	1.0918	1.0910
	W_{ER}^{τ}	50	• 4.7172	1.9549	2.2833	2.4358	1.1003	• 1.0784	1.0806	1.0798
		100	7.4379	14.5752	14.8370	• 14.9940	1.0687	1.0575	1.0553	• 1.0513
		200	8.9163	15.7847	16.0455	• 16.1758	1.1187	1.0502	• 1.0445	1.0504
	W_{ER}^{dense}	50	• 6.3349	2.5819	2.8499	2.9435	1.1456	1.0450	1.0402	• 1.0368
		100	8.5084	15.0967	15.2827	• 15.3841	1.1652	1.0382	1.0332	• 1.0327
		200	9.8226	16.1800	16.3543	• 16.4462	1.2104	1.0340	• 1.0264	1.0289

Notes. EKF stands for extended Kalman filter and SMPF(X) for smoothed marginalized particle filter with X particles. The three random networks W_{ER}^{i} with $i \in \{sparse, \tau, dense\}$ stand for Erdős–Rényi networks with three corresponding threshold parameter $\tau \in \{0.5, 1, 2\}$ that determine the networks' degree of connectedness. The average log-likelihoods are simple averages over cross-sectional size N of the evaluated log-likelihoods using prediction error decomposition for the EKF and log-likelihood estimates using the SMPF's importance weights. The bullets (•) demarcate the best values (smallest for RMSE and largest for log-likelihoods).

Table 3: RMSE of network effect states $\hat{x}_t^1, \hat{x}_t^2$

Monte Carlo results (2 of 4): For each filter we compare their estimation performance of the latent network effect states $\hat{x}_t^1, \hat{x}_t^2$ for three network effect models (structural, error and generalized), four networks (circular network, and sparse, normal, dense random networks) and three different cross-sections $N = 50, 100, 200$. We find that the SMPF performs better in all cases. Furthermore, more particles generally produce more accurate state estimates. See Figure 8 for a visual depiction of this result.

Network			RMSE of x_1 (for ϕ)				RMSE of x_2 (for θ)			
Effect	Type	N	EKF	SMPF(25)	SMPF(50)	SMPF(100)	EKF	SMPF(25)	SMPF(50)	SMPF(100)
Structural	W_C	50	0.1742	0.1362	0.0949	• 0.0712	—	—	—	—
		100	0.1757	0.1379	0.0952	• 0.0753	—	—	—	—
		200	0.1736	0.1386	0.0948	• 0.0740	—	—	—	—
	W_{ER}^{sparse}	50	0.1510	0.1515	0.1163	• 0.1021	—	—	—	—
		100	0.1600	0.1448	0.1052	• 0.0935	—	—	—	—
		200	0.1734	0.1458	0.1039	• 0.0885	—	—	—	—
	W_{ER}^{τ}	50	0.2229	0.1580	0.1194	• 0.1156	—	—	—	—
		100	0.2255	0.1572	0.1249	• 0.1098	—	—	—	—
		200	0.2545	0.1699	0.1325	• 0.1158	—	—	—	—
	W_{ER}^{dense}	50	0.2714	0.2004	0.1632	• 0.1557	—	—	—	—
		100	0.2858	0.1946	0.1707	• 0.1551	—	—	—	—
		200	0.2942	0.1950	0.1652	• 0.1613	—	—	—	—
Error	W_C	50	—	—	—	—	∘	0.6797	0.6733	• 0.6724
		100	—	—	—	—	∘	0.6104	0.6064	• 0.6015
		200	—	—	—	—	∘	0.5585	0.5484	• 0.5452
	W_{ER}^{sparse}	50	—	—	—	—	∘	0.9061	0.8981	• 0.8972
		100	—	—	—	—	∘	0.8406	• 0.8315	0.8328
		200	—	—	—	—	∘	0.8243	0.8174	• 0.8148
	W_{ER}^{τ}	50	—	—	—	—	∘	0.9101	• 0.9030	0.9036
		100	—	—	—	—	∘	0.9279	0.9188	• 0.9138
		200	—	—	—	—	∘	0.9022	0.9027	• 0.8953
	W_{ER}^{dense}	50	—	—	—	—	∘	0.9556	0.9501	• 0.9479
		100	—	—	—	—	∘	0.9548	0.9482	• 0.9435
		200	—	—	—	—	∘	0.9416	0.9391	• 0.9348
Generalized	W_C	50	0.1937	0.1692	0.1331	• 0.1202	∘	1.0156	0.9043	• 0.8667
		100	0.1976	0.1716	0.1370	• 0.1202	∘	1.0019	0.8852	• 0.8084
		200	0.1928	0.1742	0.1365	• 0.1236	∘	1.0178	0.8531	• 0.7627
	W_{ER}^{sparse}	50	0.1697	0.1749	0.1441	• 0.1322	∘	1.0774	1.0169	• 1.0033
		100	0.1771	0.1710	0.1408	• 0.1227	∘	1.0970	1.0017	• 0.9836
		200	0.1935	0.1720	0.1417	• 0.1213	∘	1.1038	1.0100	• 0.9579
	W_{ER}^{τ}	50	0.2409	0.1804	0.1504	• 0.1368	∘	1.1373	1.0688	• 1.0456
		100	0.2388	0.1787	0.1511	• 0.1316	∘	1.1546	1.0735	• 1.0653
		200	0.2648	0.1817	0.1515	• 0.1322	∘	1.2333	1.1518	• 1.0850
	W_{ER}^{dense}	50	0.2860	0.2100	0.1746	• 0.1676	∘	1.1709	1.1006	• 1.0974
		100	0.2976	0.2066	0.1786	• 0.1547	∘	1.2456	1.1385	• 1.1207
		200	0.3020	0.2025	0.1736	• 0.1523	∘	1.3132	1.2177	• 1.1927

Notes. EKF stands for extended Kalman filter and SMPF(X) for smooth marginalized particle filter with X particles. The three random networks W_{ER}^i with $i \in \{sparse, \tau, dense\}$ stand for Erdős–Rényi networks with three corresponding threshold parameter $\tau \in \{0.5, 1, 2\}$ that determine the networks' degree of connectedness. The bullets (•) demarcate the smallest for RMSE. Cases where the EKF does not produce estimates are signed with circles (

∘

). We use a dash (—) to label the cases where the respective state variable is absent from the model considered (e.g. x_t^2 is not part of the structural NE model).

Table 4: RMSE of regression coefficients $\hat{\alpha}_i, \hat{\beta}_i$

Monte Carlo results (3 of 4): For each filter we compare their estimation performance of the individual regression coefficients $\hat{\alpha}_i, \hat{\beta}_i$ for three network effect models (structural, error and generalized), four networks (circular network, and sparse, normal, dense random networks) and three different cross-sections $N = 50, 100, 200$.
We find that the SMPFs perform better than the EKF in all cases. Furthermore, more particles generally produce more accurate state estimates. The differences between the SMPFs are negligible compared to the EKF results. All filters estimate the constant α_i more accurately than the coefficient β_i. While all filters estimate the latter equally well, the SMPF increases the accuracy significantly for the constant.

Network			RMSE of α				RMSE of β			
Effect	Type	N	EKF	SMPF(25)	SMPF(50)	SMPF(100)	EKF	SMPF(25)	SMPF(50)	SMPF(100)
Structural	W_C	50	0.0581	0.0106	0.0074	• 0.0065	• 0.0996	0.1001	0.1000	0.1000
		100	0.0573	0.0114	0.0080	• 0.0068	0.1002	0.1002	0.1001	• 0.1001
		200	0.0571	0.0108	• 0.0072	0.0073	0.1004	0.1001	0.1000	• 0.1000
	W_{ER}^{sparse}	50	0.0354	0.0099	0.0078	• 0.0071	0.1011	0.1000	0.1000	• 0.1000
		100	0.0400	0.0111	0.0093	• 0.0090	0.1005	0.1002	0.1001	• 0.1000
		200	0.0441	0.0120	• 0.0102	0.0106	0.1007	0.1002	0.1001	• 0.1000
	W_{ER}^{τ}	50	0.0466	0.0153	• 0.0117	0.0135	0.1009	0.1001	0.1000	• 0.1000
		100	0.0438	0.0152	• 0.0118	0.0134	0.1007	0.1001	0.1001	• 0.1000
		200	0.0447	0.0184	0.0174	• 0.0174	0.1003	0.1001	0.1000	• 0.1000
	W_{ER}^{dense}	50	0.0447	0.0236	• 0.0169	0.0218	0.1006	0.1000	0.1000	• 0.1000
		100	0.0415	0.0238	• 0.0219	0.0227	0.1004	0.1001	0.1001	• 0.1000
		200	0.0405	0.0263	• 0.0238	0.0263	0.1001	0.1001	0.1001	• 0.1000
Error	W_C	50	0.0054	0.0041	0.0041	• 0.0041	0.1003	0.0932	0.0933	• 0.0932
		100	0.0054	0.0039	0.0039	• 0.0039	0.1000	0.0922	0.0922	• 0.0921
		200	0.0056	0.0039	0.0039	• 0.0039	0.1001	0.0913	• 0.0913	0.0914
	W_{ER}^{sparse}	50	0.0031	0.0030	0.0030	• 0.0030	0.1006	0.0993	0.0993	• 0.0993
		100	0.0033	0.0030	0.0030	• 0.0030	0.1003	0.0984	0.0984	• 0.0984
		200	0.0032	0.0029	0.0029	• 0.0029	0.1003	0.0986	• 0.0985	0.0986
	W_{ER}^{τ}	50	0.0028	0.0027	0.0027	• 0.0027	0.1006	0.0996	0.0996	• 0.0996
		100	0.0024	0.0023	0.0023	• 0.0023	0.1005	0.0999	0.0999	• 0.0999
		200	0.0021	0.0020	0.0021	• 0.0020	0.1002	• 0.0996	0.0997	0.0997
	W_{ER}^{dense}	50	0.0019	0.0019	0.0018	• 0.0018	0.1006	0.1002	• 0.1002	0.1003
		100	0.0016	0.0016	0.0016	• 0.0016	0.1004	• 0.1001	0.1001	0.1002
		200	0.0014	0.0014	0.0014	• 0.0014	0.1001	• 0.0999	0.0999	0.0999
Generalized	W_C	50	0.0601	0.0120	• 0.0103	0.0104	0.0994	0.0967	0.0968	• 0.0964
		100	0.0572	0.0145	0.0124	• 0.0110	0.1002	0.0966	0.0962	• 0.0957
		200	0.0581	0.0146	• 0.0117	0.0119	0.1004	0.0962	0.0961	• 0.0954
	W_{ER}^{sparse}	50	0.0366	0.0105	0.0103	• 0.0095	0.1015	• 0.0995	0.0996	0.0996
		100	0.0403	0.0113	0.0104	• 0.0104	0.1006	• 0.0990	0.0991	0.0991
		200	0.0446	• 0.0113	0.0126	0.0121	0.1009	0.0993	0.0993	• 0.0991
	W_{ER}^{τ}	50	0.0477	0.0145	0.0126	• 0.0121	0.1011	0.0996	0.0995	• 0.0994
		100	0.0448	0.0148	0.0144	• 0.0126	0.1010	0.1000	0.0999	• 0.0998
		200	0.0457	0.0166	0.0160	• 0.0149	0.1004	0.0997	0.0997	• 0.0995
	W_{ER}^{dense}	50	0.0443	0.0176	0.0170	• 0.0169	0.1010	0.1000	0.1000	• 0.0999
		100	0.0423	0.0215	0.0202	• 0.0187	0.1006	0.1000	0.1000	• 0.1000
		200	0.0395	0.0252	0.0228	• 0.0218	0.1001	0.0998	0.0999	• 0.0997

Notes. EKF stands for extended Kalman filter and SMPF(X) for smooth marginalized particle filter with X particles. The three random networks W_{ER}^{i} with $i \in \{sparse, \tau, dense\}$ stand for Erdős–Rényi networks with three corresponding threshold parameter $\tau \in \{0.5, 1, 2\}$ that determine the networks' degree of connectedness. The bullets (•) demarcate the smallest for RMSE.

Table 5: Dispersion of log-likelihood values

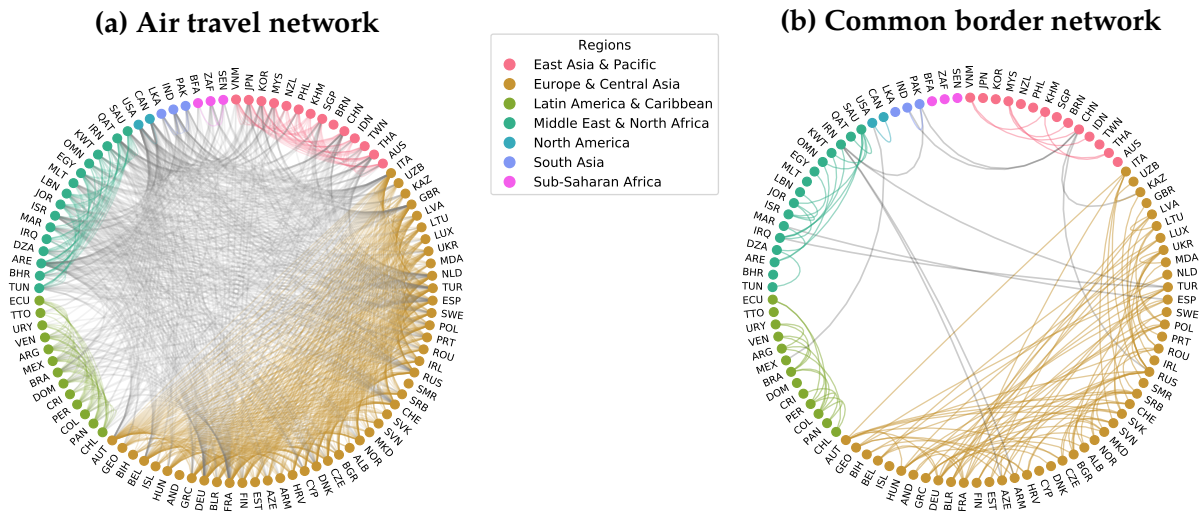
Monte Carlo results (4 of 4): For each filter we compare the precision of log-likelihood values, using median absolute deviation, for three network effect models (structural, error and generalized), four networks (circular network, and sparse, normal, dense random networks) and three different cross-sections $N = 50, 100, 200$. Note that the log-likelihood of the EKF are precise evaulations while the SMPF only provides estimates.
We find that the log-likelihood values estimated with the SMPF are less dispersed across all simulations than the EKF. Furthermore, more particles reduce the dispersion. The error network effect model is the exception, where for large, dense cross-sections, the EKF has lower dispersion. However, in conjunction with the low average log-likelihoods and high RMSE for the same cases, this implies that the EKF is inaccurate in the same way across all simulations.

Network			Median absolute deviation of log-likelihoods			
Effect	Type	N	EKF	SMPF(25)	SMPF(50)	SMPF(100)
Structural	W_C	50	117.7157	41.8602	26.0843	• 18.3775
		100	242.1026	81.2481	55.9454	• 37.4607
		200	410.0242	159.6669	108.4760	• 68.4864
	W_{ER}^{sparse}	50	31.5782	20.4561	13.8994	• 13.6018
		100	70.8985	41.0193	31.5474	• 24.5061
		200	139.6400	89.4826	61.4464	• 47.4991
	W_{ER}^{τ}	50	54.5740	34.9102	21.3749	• 17.8309
		100	69.9725	52.4510	31.9474	• 25.7047
		200	101.9624	105.6580	64.1798	• 43.1675
	W_{ER}^{dense}	50	38.9950	34.3572	19.9837	• 18.4380
		100	53.6085	60.5676	33.8521	• 27.1196
		200	69.7629	99.1985	53.5456	• 39.5332
Error	W_C	50	17.7775	8.8232	8.3023	• 8.2848
		100	35.7197	17.5047	17.5319	• 17.0523
		200	72.1706	• 28.1864	28.4456	29.2141
	W_{ER}^{sparse}	50	4.8205	4.7610	4.5879	• 4.5730
		100	10.1016	• 10.0249	10.1905	10.3226
		200	16.1719	16.3256	16.5890	• 16.1567
	W_{ER}^{τ}	50	5.2059	4.3597	4.2906	• 4.2186
		100	• 6.3872	8.6065	8.3596	8.2775
		200	• 8.6109	12.5694	12.7289	12.5731
	W_{ER}^{dense}	50	3.5753	2.9601	2.9842	• 2.8636
		100	• 4.4752	7.8064	7.9373	7.6320
		200	• 5.2964	10.8140	10.9273	10.7518
Generalized	W_C	50	134.5517	42.8931	25.3617	• 21.7171
		100	267.0145	72.4297	56.1628	• 46.7697
		200	457.3822	148.2512	105.4596	• 85.8157
	W_{ER}^{sparse}	50	33.9229	17.7477	13.6346	• 12.5149
		100	75.3883	32.5214	30.3554	• 24.1527
		200	156.7100	71.1266	54.3688	• 42.1840
	W_{ER}^{τ}	50	54.6410	23.5044	17.1538	• 14.7294
		100	74.1757	36.3376	29.7439	• 22.1599
		200	111.7836	67.9336	49.1194	• 34.3181
	W_{ER}^{dense}	50	37.9592	21.9724	15.9517	• 14.4682
		100	55.2080	30.2732	24.2585	• 19.6290
		200	68.0529	47.6886	36.8616	• 28.9590

Notes. EKF stands for extended Kalman filter and SMPF(X) for smooth marginalized particle filter with X particles. The three random networks W_{ER}^{i} with $i \in \{sparse, \tau, dense\}$ stand for Erdős–Rényi networks with three corresponding threshold parameter $\tau \in \{0.5, 1, 2\}$ that determine the networks' degree of connectedness. The bullets (•) demarcate the smallest for RMSE.

Figure 4: Long-distance and short-distance travel networks

In panel (a) we show the air routes between all countries considered. In panel (b) we connect countries if they share a common border. Both networks jointly capture 95% of international modes of transport (OECD, 2016).



6 Empirical Illustration: COVID-19 and travel networks

The SARS-CoV2 virus is spreading rapidly around the world. As of April 20, 211 countries reported a total of 2,520,000+ confirmed cases. Due to the long incubation and even longer asymptomatic period (Lauer et al., 2020), research has shown that airport screenings are of limited effectiveness (Quilty et al., 2020). It is therefore highly probable that long-distance travel, such as commercial air travel, or short-distance travel, such as railways or road travel, serve as propagation channels for the virus' global reach.

In this empirical illustration we analyze the importance of networks for the spread of COVID-19 between countries. For the number of confirmed cases we rely on the database of Johns Hopkins University, Center For Systems Science and Engineering (Dong et al., 2020). To capture long distance travel, we construct an airline network using historical flight data for more than 10,000 airports and 5,888 airlines.[12] This network reflects air travel under normal conditions and does not consider disruptions and cancellations, therefore reflecting persistent, long-term relationships between populations of different countries. For short distance travel, we construct an adjacency matrix where countries are connected if they share a common border.[13] This network also reflects permanent relationships, abstracting from temporal border closures or increased border screenings. Both networks capture more than 95% of international modes of transport, excluding naval travel (OECD, 2016). We find 128 countries with sufficient numbers of confirmed cases and network information. Since both networks reflect historical conditions, we expect the result of international containment measures to be reflected in abated dynamic network effects.

[12]Data source: http://openflights.org
[13]Data source: https://www.geodatasource.com

6.1 Phases of contagion

Figure 5 compares the number of days until the first case was reported in a country with its network centrality in the global air travel network (upper panel) and with its shortest path length to China (lower panel). The latter is measured using the inverse number of flight routes to China, that is the more flight routes a country has with China, the closer it is. The shortest path is computed using the Dijkstra algorithm. We observe that the more central a country is in the air travel network, the earlier its first case was reported. Its centrality is also correlated with the total number of infected cases. Furthermore, the lower panel shows that the closer a country is to China in terms of how easy it is to travel to China from a certain country, the earlier a first case is reported. Both pictures show a clear distinction between two phases around 20 days in mid-February (red vertical line).

Phase 1 Countries were likely directly affected by China due to their proximity or international travel hub status.

Phase 2 The spread occurs globally among all countries and is less likely fueled by direct exposure to China.

The subsequent analysis will show that as the effects of Phase 2 abate in early April, the world enters a **Phase 3** where the growing numbers are not due to international spread but domestic contagion dynamics.

6.2 Contagion models

In the analysis we consider DNE models with a single network, W_{long} or W_{short}, and with both networks combined. The single-network DNE models in (21) treat the respective networks as the only channels of contagion, which may result in overestimating the corresponding network effects. In contrast, the contagion faucet model (22) estimates the effects of both networks together, and allows us to identify which network dominates in different time periods.[14]

We first consider models without regression components. When regression components are absent, structural and error network effects coincide. We opt for error network effects θ_t to retain consistency with the subsequent extension.

$$y_t = \alpha + \eta_t \tag{20}$$

$$\text{Single network} \qquad \eta_t = \theta_t W \eta_t + e_t \qquad \text{with} \quad W \in \{W_{long}, W_{short}\} \tag{21}$$

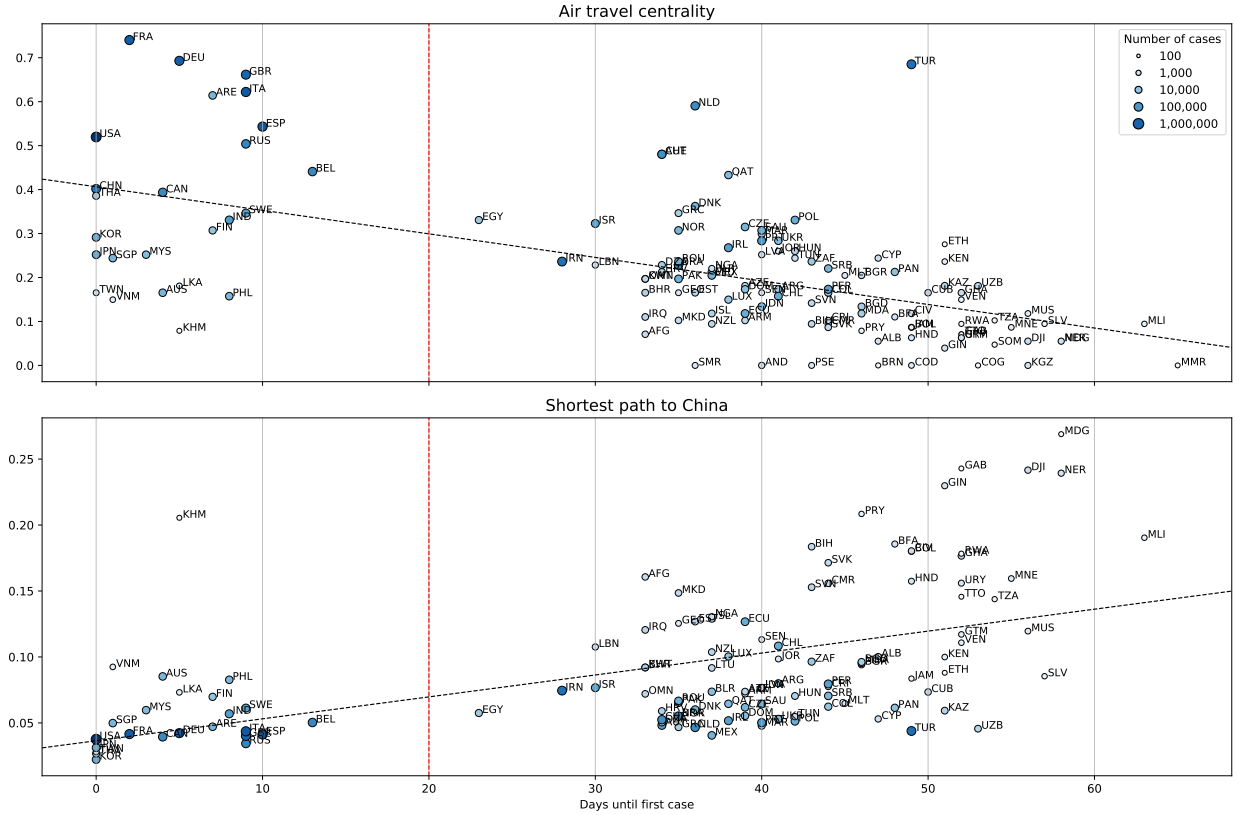
$$\text{Contagion faucet} \qquad \eta_t = \theta_t \left[\nu_t W_{long} + (1-\nu_t) W_{short}\right] \eta_t + e_t \tag{22}$$

Due to the exponentially growing case numbers, the observed variable y_t is a vector containing the log-differences of all reported cases at time t. The constant is country-specific, $\alpha = [\alpha_1, ..., \alpha_N]^\top$. The error e_t has a Gaussian distribtuion $N(0, \sigma^2 I_N)$. The state variable x_t follows a stationary process and enters the network transformation F through the logistic transformation g (see Section 2 for details of both functions). Since the models do not contain regression coefficients that could be marginalized out, it suffices to use a smooth particle filter.

[14] See Section 2.4 for the explanation of the faucet.

Figure 5: Occurrence of first case and network position

The upper panel shows the negative relationship between the centrality of a country in the global air travel network versus how many days had passed until the first COVID-19 was reported. The more central a country, the earlier the first case was reported. We also note a correlation between centrality and (maximum) number of reported cases, reflected by color and size of the dots.
The lower panel identifies a positive relationship between the shortest path length between a country and China and the days until the first COVID-19 case. The closer a country is to China in terms of air travel connections, the earlier a case was reported. Proximity to China is also correlated with the (maximum) number of reported cases.
Both panels suggest the existence of two phases, separated by the 20-days marker (red vertical line). In Phase 1 countries were likely directly affected by China, due to their proximity or international travel hub status. Phase 2 is the global spread that is less likely fueled by direct exposure to China.



Next, we extend the contagion faucet with a regression component consisting of a constant and Flu_t, the three-year average global influenza cases between 2017 and 2019. Flu_t is reported weekly and we calculate the percentage changes before using a cubic spline to obtain daily values. Controlling for the seasonal common flu cases may be relevant due to its similarity with the SARS-CoV-2 virus in terms of disease presentation, transmission process and occurrence period.[15] We obtain the data from the World Health Organization's Global Influenza Surveillance and Response System (GISRS). The model (22) is extended as follows

$$y_t = \alpha + Flu_t\beta + \eta_t \tag{23}$$

$$\eta_t = \theta_t \left[\nu_t W_{long} + (1-\nu_t)W_{short}\right]\eta_t + e_t \tag{24}$$

Similar to the Monte Carlo study, we consider heterogeneous coefficients $\alpha = [\alpha_1, ..., \alpha_N]^\top$ and $\beta =$

[15] https://www.who.int/news-room/q-a-detail/q-a-similarities-and-differences-covid-19-and-influenza, accessed on April 10, 2020.

$[\beta_1, ..., \beta_N]^\top$ for all countries $i = 1, ..., N$. We estimate this model using the smooth marginalized particle filter.

6.3 Air travel or common borders?

The estimated network effects $\hat{\theta}_t^1$ and $\hat{\theta}_t^2$ are shown in Figure 6. We can see that the long-distance network intensity $\hat{\theta}_t$ picks up two phases of the virus outbreak: Phase 1, which are likely due to the initial spread from China, and Phase 2, when cases that already entered other countries in Phase 1 begin to spread globally. Short-distance travel, $\hat{\theta}_t^2$, in comparison, plays a minor role in Phase 1 and becomes more prominent in Phase 2. For both networks it seems that air travel was responsible for early, intensive spreads while common border effects lag behind with more sustained effects. Since both models are estimated independently and people can use either of them to travel, we cannot distinguish which network effect dominates.

This problem is resolved in the combined contagion faucet model where both network effects are jointly estimated (Figure 7a). Different from Figure 6, we now introduce the overall error network importance $\hat{\theta}_t$ and the weighting parameter $\hat{v}_t$ divides this overall effect among both networks. The effective network effect for each network is then $\hat{\theta}_t^1 = \hat{\theta}_t \hat{v}_t$ and $\hat{\theta}_t^2 = \hat{\theta}_t(1 - \hat{v}_t)$. We can see that the initial spread is driven by both networks, a finding which is corroborated by media reports. In subsequent, calm period in February, the contagion faucet is turned off. When it is turned on again at the end of February, the faucet lever is turned all the way to the W_{long} side, meaning that contagion is almost entirely due to long-distance travel $\hat{\theta}_t^1$. Towards the end, the handle gradually moves towards the middle, allowing both W_{long} and W_{short} channels to contribute almost equal parts. Fortunately, the overall contagion flow slowly dries up as $\hat{\theta}_t$ drops. Note that total numbers are still increasing as of the date of writing. Taken together with the decreasing overall network importance, we can tentatively ascribe the ongoing rise to domestic spreads rather than imported foreign cases.

6.4 Controlling for global influenza cases

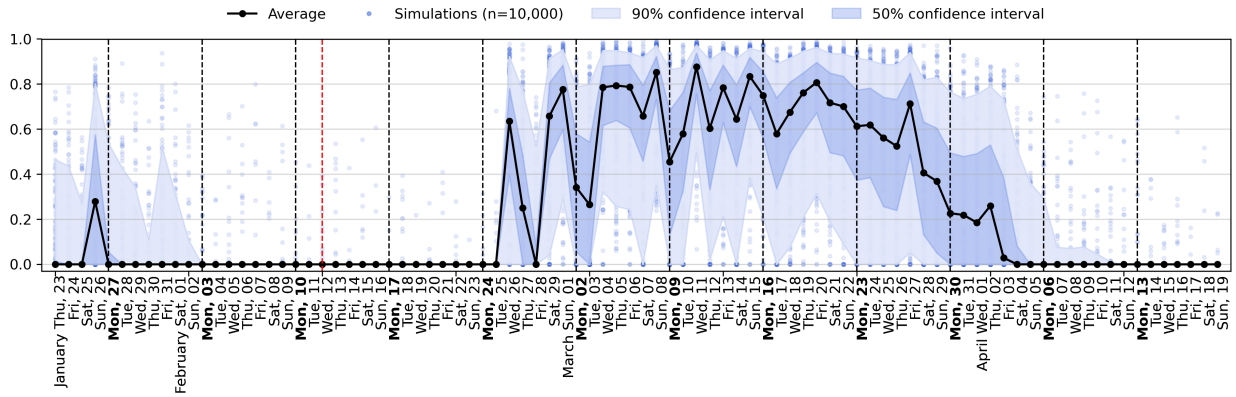
In Figure 7b we display the results from the contagion faucet model while controlling for global influenza as exogenous variable (24). We first find that the initial spike that was picked up in the pure contagion faucet model vanishes. This is likely due to the fact that the cases at the end of January where reported almost simultaenously. The results show that this inital, common occurrence is better captured by the observed common factor Flu_t. The second phase starting towards the end of February remains clearly visible. The overall contagion flow is slightly lower than in the pure contagion faucet and diminishes faster. After controlling for historical influenza cases, the overall hump shape remains the same and the relative importances between air travel and common borders strongly favors the former. The final peak in early April also vanished. Afterwards, it appears that the world has entered a **Phase 3** where the staggering numbers are not driven by international travel, but by domestic contagion. Table 6 shows the regression coefficients estimated for the constant and Flu_t.

The data is certainly subject to measurement and reporting errors not captured in e_t. Also, the long incubation period, cancelled flights and changes in reporting standards limit the insights we can gleam from this illustration. We acknowledge these data short-comings but addressing them

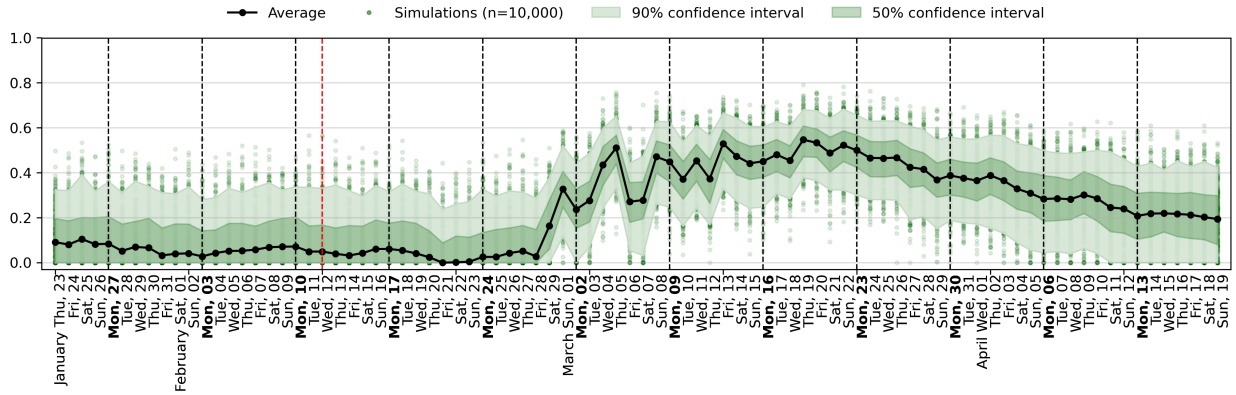
Figure 6: Spread of COVID-19 cases through airtravel or common borders

We present the estimated $\hat{\theta}_t$ for the models in (21). We find that the initial spike in Phase 1 is explained by airtravel, $\hat{\theta}^1$ (Air travel network, W_{long}), as well as the global spread during Phase 2. In contrast, common borders ($\hat{\theta}^2$) only gain in importance in Phase 2 but their effect is more sustained until the end of the observation period. Note that both network models are estimated independently from each other. The red line roughly distinguishes between Phase 1 and 2 (see Figure 5).

(a) Single network: Air travel



(b) Single network: Common border



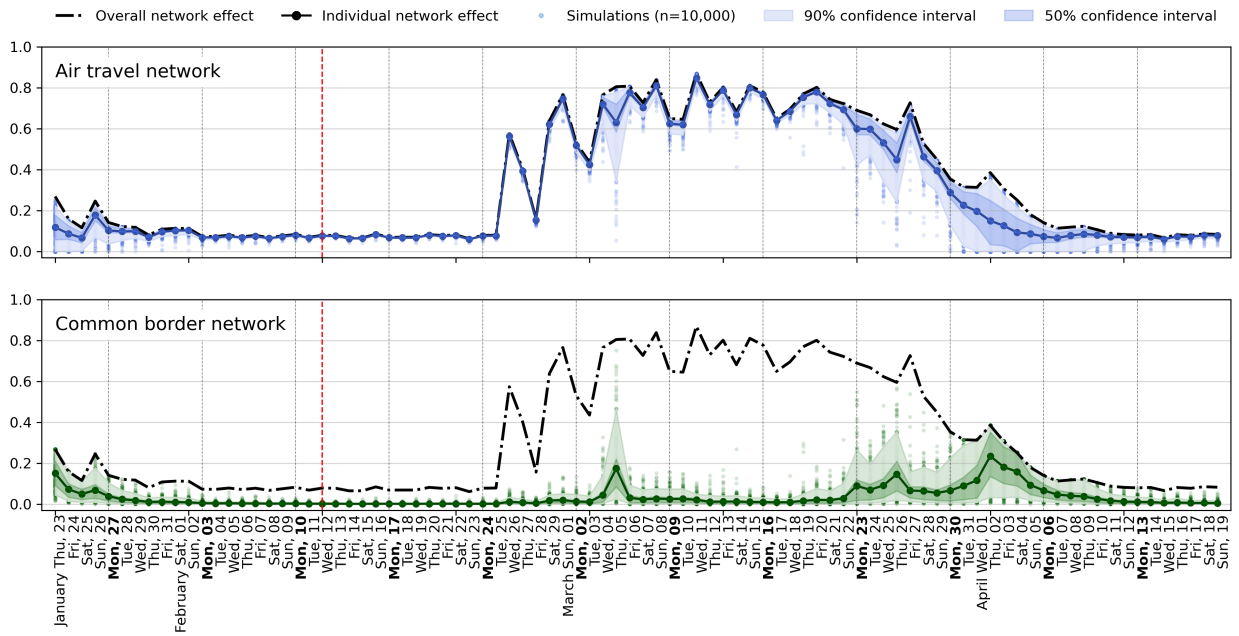
is outside the scope of this study. Nevertheless, this exercise demonstrates the ability of the SMPF to extract the time-varying network effects for multiple networks.

Figure 7: Spread of COVID-19 cases through the contagion faucet

We show the effective error network effects $\hat{\theta}_t^1 = \hat{\theta}_t \nu_t$ (Air travel network, W_{long}) and $\hat{\theta}_t^2 = \hat{\theta}_t(1 - \nu_t)$ (Common border network, W_{short}) for the model in (22). The dashed lines represents the overall importance of both networks, $\hat{\theta}_t$. The red line roughly distinguishes between Phase 1 and 2 (see Figure 5).

(a) Pure contagion faucet

As with the single network model (Figure 6) it seems that the initial spike was driven by the air travel. After the contagion faucet had been turned off for the most part of Feburary (see Section 2.4 for an explanation of the faucet), we observe stronger overall network effects $\hat{\theta}_t$ since the end of February, which was predominately driven by long-distance travel (W_{long}). In contrast, short-distance travel (W_{short}) grew in importance towards the end of the observation period. This may reflect the result of global containment measures.



(b) Contagion faucet, controlling for Flu_t

After controlling for global influenza levels, we find that while the overall hump shape in Phase 2 persists, its values are lower. The predominance of the air travel network also remains. However, the initial spike around the end of January and the spike at the beginning of April vanished, indicating that they are better explained by a global common factor Flu_t.

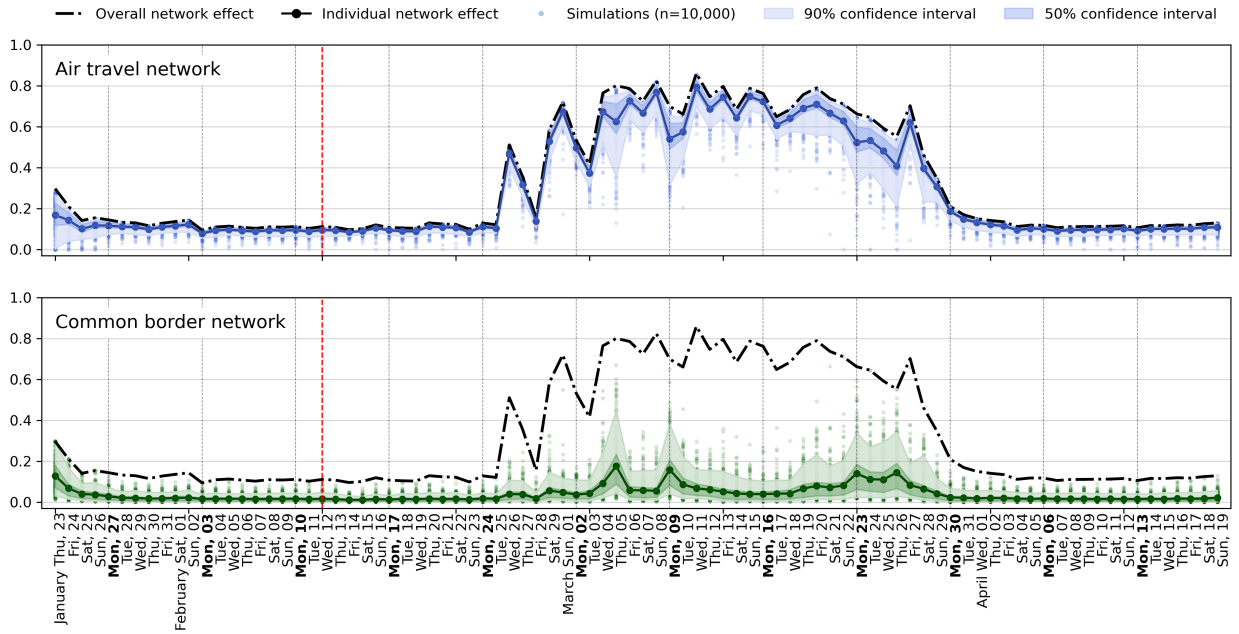


Table 6: Regression coefficients of the contagion faucet

This table presents the regression coefficients that were estimated using the marginalized particle filter as linear state variables. The global influenza variable refers to the average global levels of influenza cases between 2016 and 2019. The model covers 128 countries for which sufficient COVID-19 cases had been reported. Coefficients significant on a 5% significance level are highlighted in bold.

	Constant			Global influenza				Constant			Global influenza		
	Coef.	t-stat	p-val.	Coef.	t-stat	p-val.		Coef.	t-stat	p-val.	Coef.	t-stat	p-val.
AFG	**-0.791**	-26.336	0.000	-0.279	-1.416	0.157	JPN	**-0.690**	-23.161	0.000	0.320	1.626	0.104
ALB	**-0.770**	-25.881	0.000	-0.112	-0.570	0.569	KAZ	**-0.792**	-26.237	0.000	-0.361	-1.829	0.067
AND	**-0.813**	-27.360	0.000	-0.267	-1.358	0.175	KEN	**-0.803**	-26.556	0.000	-0.319	-1.616	0.106
ARE	**0.066**	2.295	0.022	0.022	0.114	0.909	KGZ	**-0.795**	-26.611	0.000	**-0.391**	-1.985	0.047
ARG	**-0.780**	-25.804	0.000	-0.197	-0.996	0.319	KHM	**-0.790**	-26.552	0.000	-0.063	-0.321	0.748
ARM	**-0.784**	-26.338	0.000	-0.223	-1.132	0.258	KOR	**-0.664**	-22.100	0.000	**0.497**	2.519	0.012
AUS	**-0.679**	-22.361	0.000	**0.404**	2.042	0.041	KWT	**-0.801**	-26.922	0.000	**-0.636**	-3.237	0.001
AUT	**-0.766**	-25.728	0.000	-0.317	-1.610	0.107	LBN	**-0.800**	-26.883	0.000	-0.229	-1.167	0.243
AZE	**-0.796**	-26.719	0.000	-0.344	-1.749	0.080	LKA	**-0.790**	-26.274	0.000	-0.147	-0.745	0.456
BEL	**-0.755**	-25.334	0.000	-0.346	-1.757	0.079	LTU	**-0.777**	-26.122	0.000	-0.244	-1.242	0.214
BFA	**-0.774**	-26.010	0.000	-0.271	-1.379	0.168	LUX	**-0.758**	-25.409	0.000	-0.176	-0.892	0.372
BGD	**-0.817**	-27.451	0.000	**-0.533**	-2.711	0.007	LVA	**-0.769**	-25.729	0.000	-0.134	-0.683	0.495
BGR	**-0.799**	-26.316	0.000	-0.125	-0.634	0.526	MAR	**-0.793**	-26.681	0.000	-0.344	-1.750	0.080
BHR	**-0.807**	-27.145	0.000	**-0.480**	-2.440	0.015	MDA	**-0.792**	-26.181	0.000	-0.338	-1.713	0.087
BIH	**-0.783**	-26.215	0.000	-0.244	-1.241	0.215	MDG	**-0.808**	-27.162	0.000	-0.247	-1.257	0.209
BLR	**-0.805**	-26.788	0.000	**-0.530**	-2.690	0.007	MEX	**-0.782**	-25.671	0.000	**-0.486**	-2.458	0.014
BOL	**-0.795**	-26.717	0.000	-0.302	-1.534	0.125	MKD	**-0.791**	-26.313	0.000	-0.227	-1.150	0.250
BRA	**-0.763**	-25.633	0.000	**-0.419**	-2.131	0.033	MLI	**-0.824**	-27.711	0.000	**-0.392**	-1.992	0.046
BRN	**-0.795**	-26.162	0.000	0.001	0.006	0.995	MLT	**-0.782**	-26.219	0.000	-0.140	-0.709	0.478
CAN	**-0.729**	-23.981	0.000	0.026	0.132	0.895	MMR	**-0.816**	-26.974	0.000	-0.344	-1.741	0.082
CHE	**-0.766**	-25.646	0.000	**-0.442**	-2.244	0.025	MNE	**-0.805**	-27.056	0.000	-0.293	-1.492	0.136
CHL	**-0.760**	-25.570	0.000	-0.284	-1.442	0.149	MUS	**-0.795**	-26.716	0.000	-0.279	-1.419	0.156
CHN	**-0.639**	-21.270	0.000	**0.970**	4.918	0.000	MYS	**-0.681**	-22.889	0.000	**0.431**	2.192	0.028
CIV	**-0.786**	-26.241	0.000	-0.369	-1.872	0.061	NER	**-0.828**	-27.838	0.000	**-0.566**	-2.879	0.004
CMR	**-0.807**	-26.624	0.000	-0.375	-1.896	0.058	NGA	**-0.793**	-26.628	0.000	-0.369	-1.874	0.061
COD	**-0.813**	-26.854	0.000	-0.252	-1.274	0.203	NLD	**-0.754**	-25.338	0.000	**-0.419**	-2.132	0.033
COG	**-0.806**	-27.116	0.000	**-0.474**	-2.409	0.016	NOR	**-0.767**	-25.785	0.000	-0.329	-1.673	0.094
COL	**-0.752**	-25.284	0.000	-0.296	-1.508	0.132	NZL	**-0.768**	-25.846	0.000	-0.307	-1.564	0.118
CRI	**-0.787**	-26.044	0.000	-0.126	-0.638	0.524	OMN	**-0.796**	-26.356	0.000	**-0.497**	-2.515	0.012
CUB	**-0.801**	-26.904	0.000	-0.360	-1.833	0.067	PAK	**-0.782**	-26.295	0.000	**-0.434**	-2.209	0.027
CYP	**-0.808**	-27.261	0.000	-0.140	-0.712	0.476	PAN	**-0.782**	-25.950	0.000	-0.266	-1.346	0.178
CZE	**-0.758**	-25.444	0.000	-0.239	-1.217	0.224	PER	**-0.781**	-26.209	0.000	-0.384	-1.951	0.051
DEU	**-0.675**	-22.660	0.000	0.220	1.117	0.264	PHL	**-0.750**	-24.924	0.000	-0.043	-0.219	0.826
DJI	**-0.815**	-27.345	0.000	**-0.560**	-2.847	0.004	POL	**-0.765**	-25.676	0.000	-0.231	-1.175	0.240
DNK	**-0.762**	-25.554	0.000	-0.233	-1.184	0.236	PRT	**-0.768**	-25.431	0.000	-0.248	-1.257	0.209
DOM	**-0.789**	-26.262	0.000	-0.328	-1.666	0.096	PRY	**-0.788**	-26.078	0.000	-0.220	-1.116	0.265
DZA	**-0.768**	-25.830	0.000	-0.383	-1.947	0.052	PSE	0.023	0.800	0.424	-0.223	-1.165	0.244
ECU	**-0.787**	-26.004	0.000	-0.368	-1.861	0.063	QAT	**-0.777**	-25.625	0.000	-0.376	-1.903	0.057
EGY	**-0.778**	-26.026	0.000	-0.257	-1.305	0.192	ROU	**-0.776**	-26.042	0.000	-0.354	-1.801	0.072
ESP	**-0.725**	-24.377	0.000	**-0.462**	-2.347	0.019	RUS	**-0.758**	-25.306	0.000	**-0.387**	-1.967	0.049
EST	**-0.766**	-25.717	0.000	-0.158	-0.804	0.421	RWA	**-0.807**	-26.961	0.000	-0.208	-1.056	0.291
ETH	**-0.816**	-27.249	0.000	-0.276	-1.399	0.162	SAU	**-0.774**	-25.981	0.000	-0.283	-1.440	0.150
FIN	**-0.777**	-25.916	0.000	-0.296	-1.503	0.133	SEN	**-0.792**	-26.504	0.000	-0.185	-0.941	0.347
FRA	**-0.710**	-23.760	0.000	0.100	0.509	0.611	SGP	**-0.616**	-17.440	0.000	0.569	1.185	0.236
GAB	**-0.805**	-27.017	0.000	**-0.411**	-2.092	0.036	SLV	**-0.833**	-27.927	0.000	-0.333	-1.690	0.091
GBR	**0.083**	2.880	0.004	-0.117	-0.615	0.538	SMR	**-0.770**	-25.890	0.000	-0.216	-1.097	0.273
GEO	**-0.801**	-26.945	0.000	-0.201	-1.023	0.306	SOM	**-0.830**	-27.893	0.000	**-0.522**	-2.655	0.008
GHA	**-0.797**	-26.560	0.000	-0.339	-1.722	0.085	SRB	**-0.798**	-26.605	0.000	-0.243	-1.234	0.217
GIN	**-0.835**	-28.081	0.000	**-0.555**	-2.821	0.005	SVK	**-0.658**	-23.627	0.000	0.842	1.522	0.128
GRC	**-0.796**	-26.738	0.000	-0.299	-1.520	0.128	SVN	**-0.764**	-25.670	0.000	-0.056	-0.285	0.776
GTM	**-0.819**	-27.282	0.000	-0.316	-1.605	0.109	SWE	**-0.780**	-26.239	0.000	**-0.416**	-2.117	0.034
HND	**-0.798**	-26.506	0.000	-0.286	-1.451	0.147	THA	**-0.700**	-23.301	0.000	**0.463**	2.348	0.019
HRV	**-0.785**	-26.384	0.000	-0.382	-1.944	0.052	TTO	0.014	0.485	0.628	-0.142	-0.748	0.455
HUN	**-0.779**	-26.102	0.000	-0.249	-1.264	0.206	TUN	0.021	0.721	0.471	-0.182	-0.958	0.338
IDN	**-0.801**	-26.432	0.000	-0.239	-1.210	0.226	TUR	0.017	0.584	0.559	**-0.476**	-2.499	0.012
IND	**-0.748**	-24.586	0.000	-0.264	-1.335	0.182	TWN	**-0.705**	-23.692	0.000	**0.522**	2.653	0.008
IRL	**-0.766**	-25.764	0.000	-0.265	-1.347	0.178	TZA	**-0.824**	-27.430	0.000	-0.383	-1.944	0.052
IRN	**-0.758**	-25.472	0.000	**-0.454**	-2.307	0.021	UKR	0.021	0.734	0.463	**-0.465**	-2.434	0.015
IRQ	**-0.800**	-26.343	0.000	**-0.412**	-2.082	0.037	URY	-0.002	-0.084	0.933	-0.149	-0.779	0.436
ISL	**-0.775**	-26.073	0.000	-0.237	-1.205	0.228	USA	**0.143**	4.955	0.000	0.068	0.355	0.722
ISR	**-0.769**	-25.750	0.000	**-0.448**	-2.274	0.023	UZB	0.001	0.052	0.959	**-0.475**	-2.493	0.013
ITA	**-0.729**	-24.486	0.000	-0.227	-1.155	0.248	VEN	0.007	0.251	0.802	-0.152	-0.793	0.428
JAM	**-0.827**	-27.780	0.000	-0.220	-1.117	0.264	VNM	**0.098**	3.371	0.001	**0.493**	2.582	0.010
JOR	**-0.794**	-26.685	0.000	-0.186	-0.944	0.345	ZAF	**-0.758**	-25.475	0.000	-0.131	-0.666	0.506

7 Conclusion

We proposed the dynamic network effects (DNE) model and put forward the smooth marginalized particle filter (SMPF) as an appropriate estimation method. The DNE model incorporates cross-sectional network spillovers, the intensity of which are captured by a vector of latent time-varying parameters. Similarly to dynamic factor models, DNE models allow us to decompose panel data into a constant cross-sectional component, informed by the networks, and a lower-dimensional time-varying component. Different from the extracted factors from a dynamic factor model, the interpretation of estimated network effects is always unambiguous as it is derived from the associated network(s).

Depending on the application, different DNE models can be formulated. The *structural network effect model* assumes that the observed outcome of a unit this is not only a function of its own regressors and idiosyncratic shocks, but also of regressors and disturbances of its neighbors. The *error network effects model* follows the same rationale but restricts its effect to the error only, allowing us to parsimoniously incorporate multivariate stochastic volatilities into the model. Combining the two yields the *generalized network effects model*.

For the purpose of estimation and filtering, we cast the DNE model into a nonlinear state-space framework, and advocate the use of the SMPF, as it can handle complicated nonlinearities as well as large numbers of regressors. We demonstrate its superior performance compared to a widely-used alternative, the extended Kalman filter (EKF), in a Monte Carlo study. We highlight the performance differences in terms of prediction, signal extraction, coefficient estimation and likelihood evaluation/estimation. We also compare the filtering behavior for different network types, such as the circular network and Erdős–Rényi networks with different degrees of connectedness. We find that the EKF performs better only for small cross sections with sparse networks and when error network effects are absent.

To illustrate our framework, we apply it to model the contagion process of the current COVID-19 outbreak. In this case, the model features a "contagion faucet" that measures overall contagion flow between 128 countries through either long-distance travel, proxied by commercial air travel routes, or short-distance travel, proxied by a network of common borders. We find that the pandemic has spread around the globe in two phases, at first primarily through air travel, and later through shorter-distance travel between neighboring countries. As of the time of writing the overall importance of networks has abated, in particular that of air travel. However, the number of cases is still rising exponentially. This leads us to conclude that we have a new phase where the growth of infection numbers seems to be primarily due to domestic contagion. This is definitely an interesting subject of further research.

References

Aït-Sahalia, Y., Laeven, R. J., & Pelizzon, L. (2014). Mutual excitation in eurozone sovereign CDS. *Journal of Econometrics*, *183*(2), 151–167.

Andrieu, C., & Doucet, A. (2002). Particle filtering for partially observed gaussian state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *64*(4), 827–836.

Anselin, L. (1988). *Spatial econometrics: Methods and models*. Springer, Dordrecht.

Blasques, F., Koopman, S. J., Lucas, A., & Schaumburg, J. (2016). Spillover dynamics for systemic risk measurement using spatial financial time series models. *Journal of Econometrics*, *195*(2), 211–223.

Böhm, H., Schaumburg, J., & Tonzer, L. (2020). Financial linkages and sectoral business cycle synchronization: Evidence from europe. *Tinbergen Institute Discussion Paper 2020-008/III*.

Casella, B. Y. G., & Robert, C. P. (1996). Rao-blackwellisation of sampling schemes, 81–94.

Catania, L., & Billé, A. G. (2017). Dynamic spatial autoregressive models with autoregressive and heteroskedastic disturbances. *Journal of Applied Econometrics*, *32*(6), 1178–1196.

Dong, E., Du, H., & Gardner, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases*, S1473309920301201.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential monte carlo methods in practice*. Springer-Verlag New York.

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, *10*(3), 197–208.

Durbin, J., & Koopman, S. J. (2012). *Time series analysis by state space methods* (2nd ed). Oxford, Oxford University Press.

Erdős, P., & Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, *5*(1), 17–60.

Forbes, K. J., & Rigobon, R. (2002). No contagion, only interdependence: Measuring stock market comovements. *The Journal of Finance*, *57*(5), 2223–2261.

Graham, A. J., & Pike, D. A. (2008). A note on thresholds and connectivity in random directed graphs. *Atlantic Electronic Journal of Mathematics*, *3*(1), 1–5.

Harvey, A. C. (1990). *Forecasting, structural time series models and the kalman filter*. Cambridge University Press.

Harvey, A., Ruiz, E., & Shephard, N. (1994). Multivariate stochastic variance models. *The Review of Economic Studies*, *61*(2), 247–264.

Hol, J. D., Schon, T. B., & Gustafsson, F. (2006, September). On resampling algorithms for particle filters, In *2006 IEEE nonlinear statistical signal processing workshop*. 2006 IEEE Nonlinear Statistical Signal Processing Workshop.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., & Chopin, N. (2015). On particle methods for parameter estimation in state-space models. *Statistical Science*, *30*(3), 328–351.

Kelejian, H. H., & Prucha, I. R. (2010). Specification and estimation of spatial autoregressive models with autoregressive and heteroskedastic disturbances. *Journal of Econometrics*, *157*(1), 53–67.

Kilian, L., & Lütkepohl, H. (2017). *Structural vector autoregressive analysis*. Cambridge University Press.

Lauer, S. A., Grantz, K. H., Bi, Q., Jones, F. K., Zheng, Q., Meredith, H. R., Azman, A. S., Reich, N. G., & Lessler, J. (2020). The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: Estimation and application. *Annals of Internal Medicine*.

LeSage, J., & Pace, R. K. (2009). *Introduction to spatial econometrics*. Boca Raton, Chapman; Hall/CRC Press.

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Berlin Heidelberg, Springer-Verlag.

Malik, S., & Pitt, M. K. (2011). Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics*, *165*(2), 190–209.

OECD. (2016). *OECD tourism trends and policies 2016*. Paris, OECD Publishing.

Ord, K. (1975). Estimation methods for models of spatial interaction. *Journal of the American Statistical Association*, *70*(349), 120–126.

Parkison, R. (1973, June 5). *Single lever faucet* (U.S. pat. 3736959A).

Quilty, B., Clifford, S., Flasche, S., & Eggo, R. M. (2020, February 2). *Effectiveness of airport screening at detecting travellers infected with 2019-nCoV* (preprint). Epidemiology.

Schon, T., Gustafsson, F., & Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, *53*(7), 2279–2289.

Schon, T., Karlsson, R., & Gustafsson, F. (2006). The marginalized particle filter in practice, In *2006 IEEE aerospace conference*. 2006 IEEE Aerospace Conference, Big Sky, MT, USA, IEEE.

Tang, T. (1998, April 21). *Ball valve faucet* (U.S. pat. 5740836A).

Wang, D., Schaumburg, J., & van Lelyveld, I. (2019, May). *Do information contagion and business model similarities explain bank credit risk commonalities?* (ESRB Working Paper No. 94). ESRB.

8 Appendix

8.1 Illustrative simulation: Filtering and estimation differences between SMPF, EKF and simple Kalman Filter

We illustrate the performance of different candidate filters filters in terms of signal extraction and regression coefficient estiamtion. The simulation study is set up as follows. We simulate from a generalized DNE model with a regression component $\alpha_i + X_{it}\beta_i$ and a circular network W_C. We simulate $T = 200$ periods and a cross-section of $N = 50$. The measurement error variance is set to $\sigma^2 = 1$ for all i. The state equation for $x_t^1 = g^{-1}(\phi_t)$ has no intercept, i.e., $c_1 = 0$, the autocorrelation coefficient is $A_1 = 0.8$ and the state variance is set to $\Omega_1 = 0.5$. The state equation for $x_t^2 = g^{-1}(\theta_t)$ also has $c_2 = 0$, autocorrelation coefficient $A_2 = 0.4$ and the state variance $\Omega_2 = 0.7$.

We first analyze the performance of the KF, EKF and the SMPF with respect to filtering the true state x_t and thereby correctly estimating the network effects ϕ_t, θ_t. We then illustrate the bias introduced to the regression coefficient estimates α_i, β_i when using a linear filter like the KF and how the EKF and SMPF remove this bias.

8.1.1 Results: Signal extraction

We compare the results of the EKF and SMPF, as the KF does not have the capacity to handle nonlinear latent states. The EKF is able to accurately estimate the signal ϕ_t but has no tractability for the θ_t signal. This is by construction of the EKF. In the spirit of Kalman filtering, the EKF updates the state variable based on the prediction error v_t, which relies on the derivative of the measurement function $D_x\mu(x_t)$. However, this derivative is zero for x_t^2, since it does not play a role in $\mu(x_t)$. The derivative does play a role in the Kalman filtering steps through the prediction error variance and the measurement error covariance $\Sigma_t^* := \mathbf{\Phi}_t\mathbf{\Theta}_t\Sigma\mathbf{\Theta}_t^\top\mathbf{\Phi}_t^\top$. In comparison, the SMPF is able to estimate the signal ϕ_t with a lower MSE than the EKF and, most importantly, also provides an accurate measure for the signal θ_t.

8.1.2 Results: Regression coefficient estimation

Regression intercept α_i We see that for increasing $|\phi_t|$, the KF introduces significant biases into the individual $\hat{\alpha}_i$ as well as their average, The EKF largely contains the bias and is not strongly affected by varying A_1 values. The SMPF has the most efficient estimates with only small individual biases.

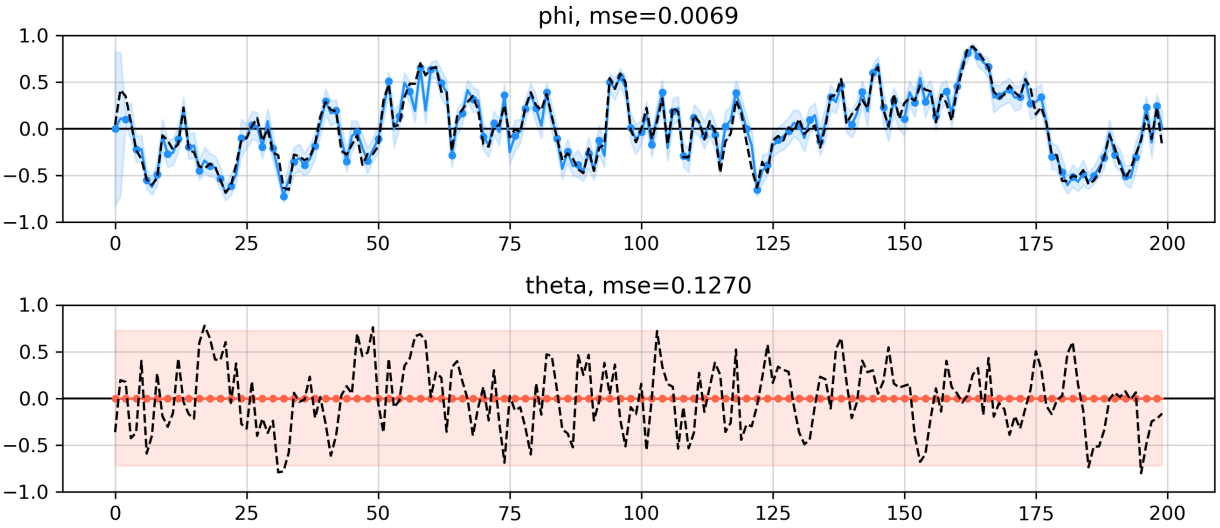
Regression coefficient β_i A different picture emerges for the $\hat{\beta}$. We see that for increasing $|\phi_t|$, the KF introduces significant biases into the individual $\hat{\alpha}_i$ but on average, the estimated intercept stays around zero. As before, the EKF largely contains the bias and is not strongly affected by varying A_1 values. The SMPF again yields the most efficient estimates with negligible individual biases.

8.2 Additional figures

Figure 8: Signal extraction with EKF and SMPF

The figures below demonstrate the performance of the EKF and SMPF in terms of extracting the true signals or network effects (black, dashed). The shaded areas around the extracted signals (blue and red) represent the 95% asymmetric confidence intervals.

(a) Extended Kalman Filter



(b) Particle Filter

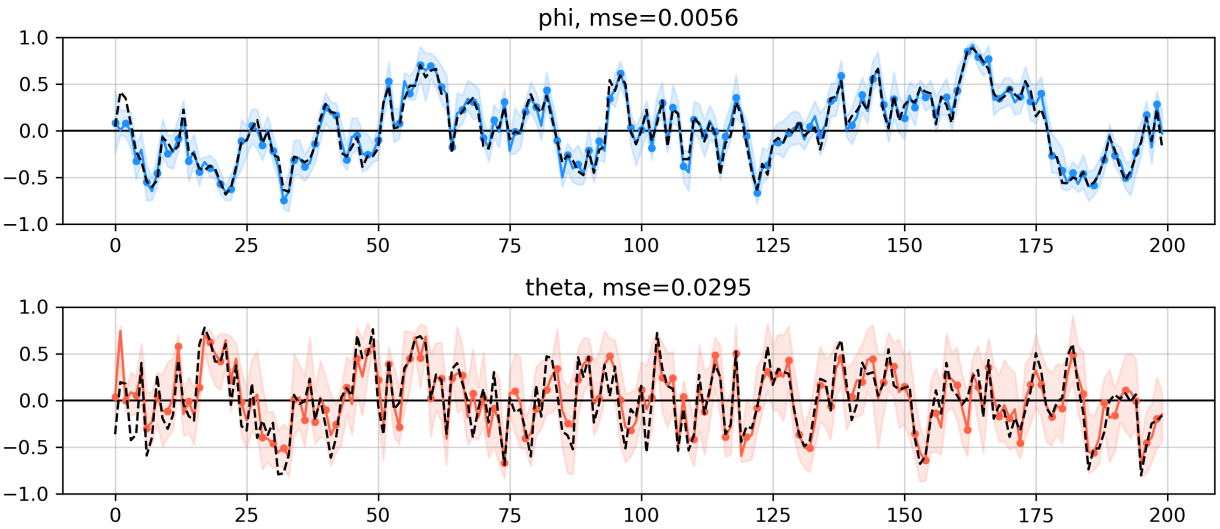
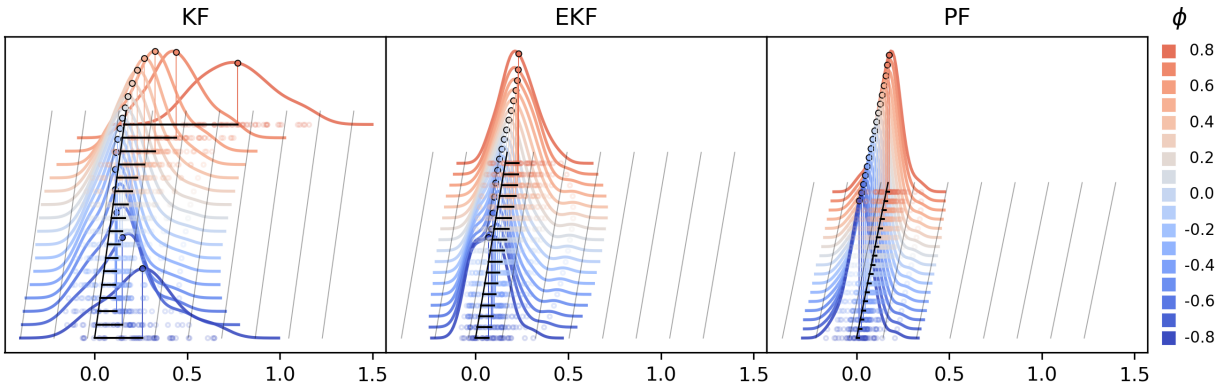


Figure 9: Regression coefficient estimation with KF, EKF and SMPF

The figures summarize the behavior of the estimated $\hat{\alpha}, \hat{\beta}$ under different ϕ_t intensities. Each density represents a kernel density over the estimated coefficients for all $i = 1, .., 50$. The vertical lines mark the average estimate coefficient and the dots on the floow depict the individual coefficient estimates. The true coefficients are $\alpha_i = 0$ and $\beta_i = 1$.

(a) Intercept α_i



(b) Regression coefficient β_i

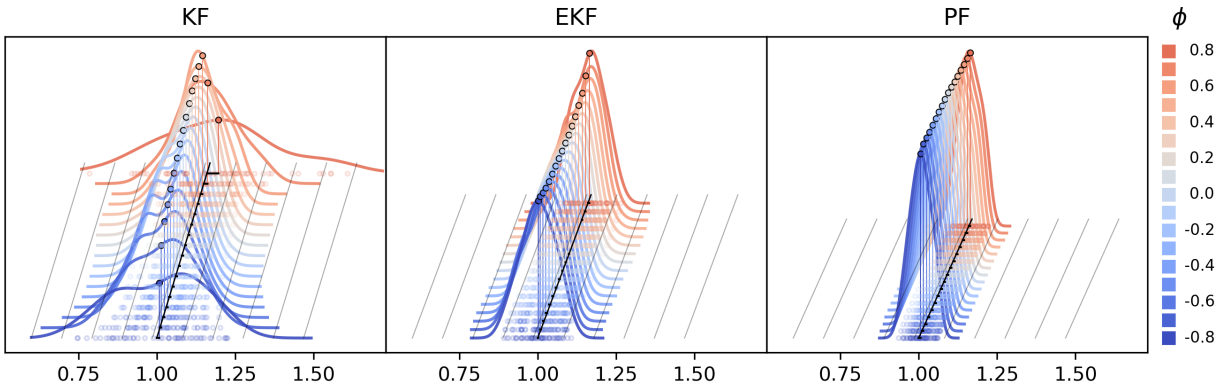


Figure 10: Empirical distribution function of particles

In this figure we compare the empirical distribution function of a stratified and continuous resampling procedures. The latter allows the resampled particles to be distributed more evenly over the domain, whereas the former subjects the resampling to the abrupt changes in the step function.

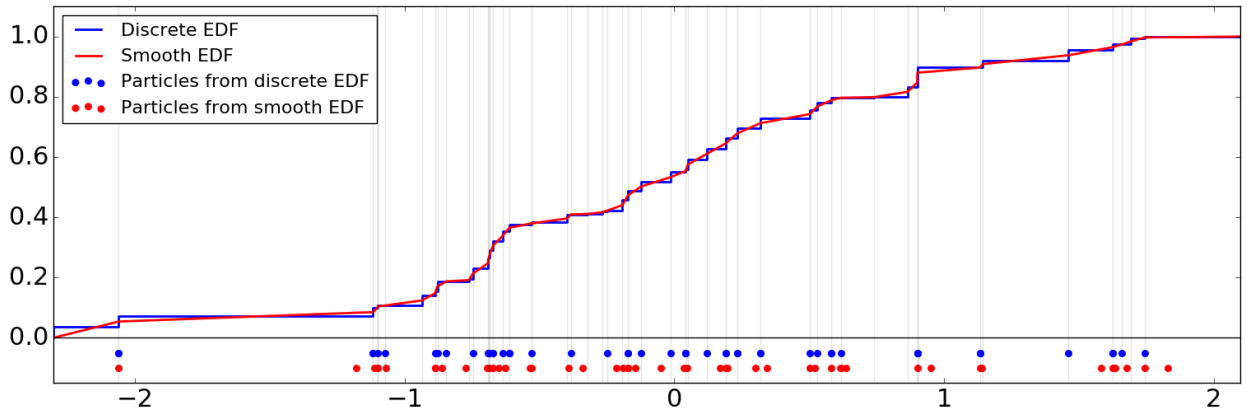


Figure 11: Discontinuous and smooth likelihood estimates

Discrete and smooth log-likelihoods from corresponding empirical distribution functions by varying the autocorrelation parameter T in the state equation. The true value is $T_0 = 0.8$. We can see that the smooth likelihood has unique maximum while the likelihood estimate using stratified resampling may result in local minima.

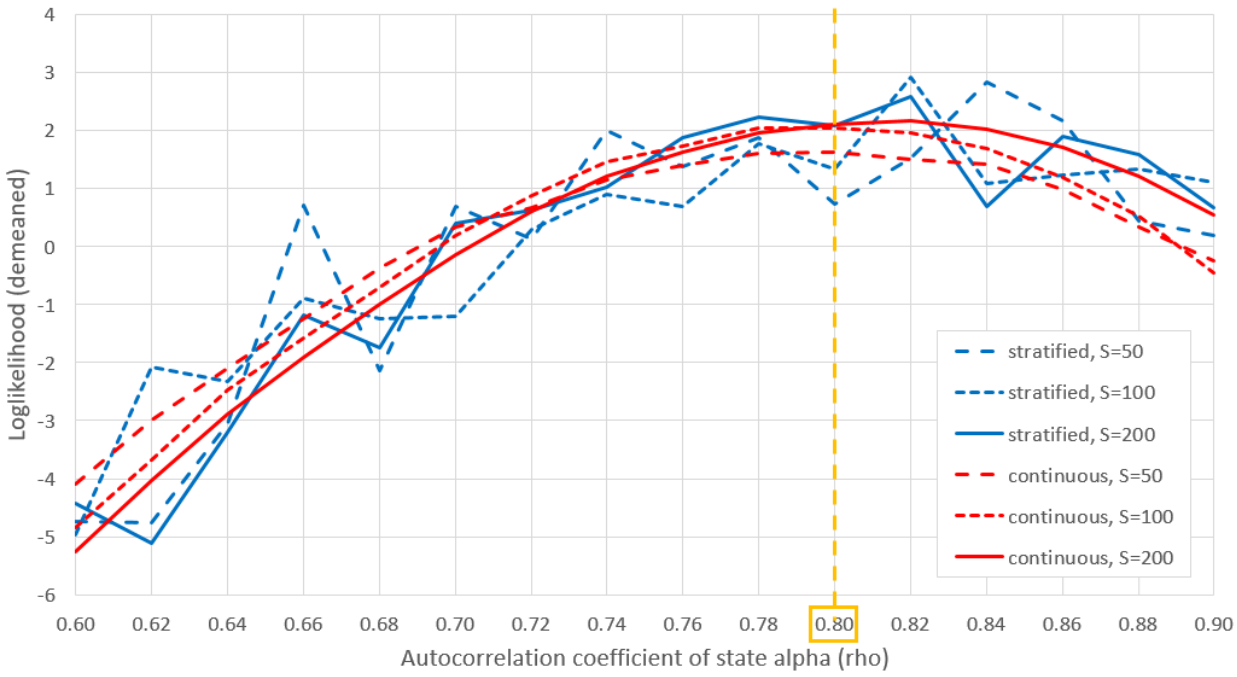


Figure 12: Computational time of EKF versus SMPF with different number of particles

This figure shows the average computational time for the extended Kalman filter (EKF) and the smooth marginalized particle filter (SMPF) considered in the Monte Carlo study. S denotes the number of particles used in the SMPF. The vertical bars show the average time plus/minus one standard deviation. See Section 5.2 for a description for the computational environment used for these results.

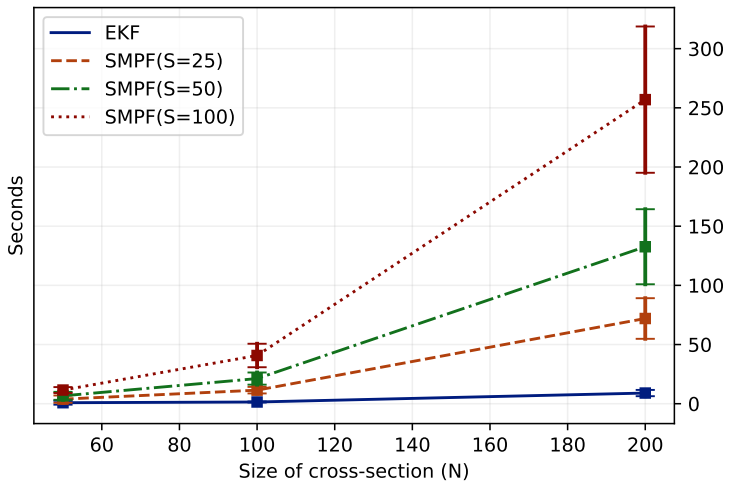


Figure 13: Overview of global COVID-19 cases

This figure shows the number of COVID-19 cases in 128 countries between Jan 22 and April 19, 2020. For improved visualization we take the square root. The black lines and dots demarcate the beginning of a week. Behind the country codes we list the number of active cases at the end of the observation period. We can see two phases of the pandemic: A first phase starts at the end of January, where earliest cases are reported in 33 countries outside of China. After a few calm weeks, a second phase begins in the last week of February. The virus spreads around the globe until the second week of March, before the World Health Organization recognizes the disease as a pandemic on March 11.

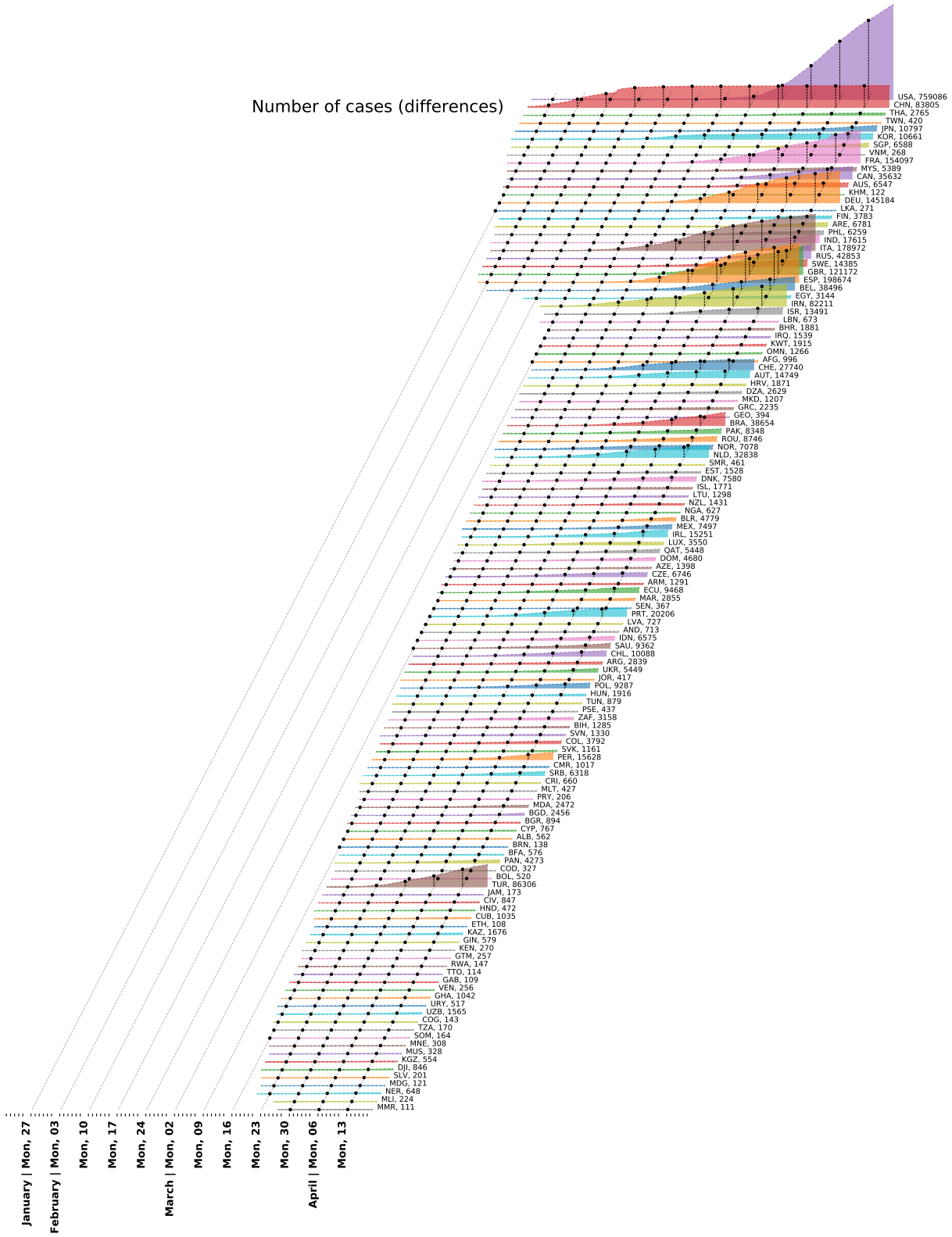


Figure 14: Overview of global COVID-19 cases (differences)

This figure shows the changes in numbers of COVID-19 cases in 128 countries between Jan 22 and April 19, 2020. For improved visualization we take the square root. The black lines and dots demarcate the beginning of a week. Behind the country codes we list the number of active cases at the end of the observation period. We can see two phases of the pandemic: A first phase starts at the end of January, where earliest cases are reported in 33 countries outside of China. After a few calm weeks, a second phase begins in the last week of February. The virus spreads around the globe until the second week of March, before the World Health Organization recognizes the disease as a pandemic on March 11.