On the prediction of functional time series*

Alexander Aue† Diogo Dubart Norinho‡ Siegfried Hörmann§

Abstract

This paper addresses the prediction of functional time series. Existing contributions to this problem have largely focused on the special case of first-order functional autoregressive processes because of their technical tractability and the current lack of advanced functional time series methodology. While the linear prediction equations for any stationary functional time series can be stated explicitly, it seems in most situations infeasible to solve them in practice. Using functional principal components analysis, it is shown here how standard multivariate prediction techniques can be utilized instead. The connection between functional and multivariate predictions is made precise for the important case of vector and functional autoregressions. The proposed method is easy to implement, making use of existing statistical software packages, and may therefore be attractive to a broader, possibly non-academic, audience. Its practical applicability is demonstrated in a simulation study and an application to environmental data, namely the prediction of daily pollution curves describing the concentration of particulate matter in ambient air. It is found that the proposed prediction method, if based on the multivariate innovations algorithm, often outperforms the standard functional prediction technique.

Keywords: Dimension reduction; Forecasting, Functional autoregressions; Functional principal components, Functional time series; Particulate matter, Vector autoregressions

MSC 2010: Primary 62M10, 62M20; Secondary 62P12, 60G25

1 Introduction

Functional data are often collected in sequential form. The common situation is a continuous-time record that can be separated into natural consecutive time intervals, such as days, for which a

*This research was partially supported by NSF grant DMS 0905400 and Communauté française de Belgique—Actions de Recherche Concertées (2010–2015).

†Department of Statistics, University of California Davis, One Shields Avenue, Davis, CA 95616, USA, email: aaue@ucdavis.edu

‡Department of Computer Science, University College London, London WC1E 6BT, UK, email: ucabdub@ucl.ac.uk

§Département de Mathématique, Université Libre de Bruxelles, CP 215, Boulevard du Triomphe, B-1050 Bruxelles, Belgium, email: shormann@ulb.ac.be

reasonably similar behavior is expected. Typical examples include the daily price and return curves of financial transactions data and the daily patterns of geophysical, meteorological and environmental data. The resulting functions may be described by a time series $(Y_k\colon k \in \mathbb{Z})$, each term in the sequence being a (random) function $Y_k(t)$ defined for t taking values in some interval $[a, b]$. Here, $\mathbb{Z}$ denotes the set of integers. The object $(Y_k\colon k \in \mathbb{Z})$ will be referred to as a functional time series (see Hörmann and Kokoszka [19] for a recent survey on time series aspects, and Ferraty and Vieu [14] and Ramsay and Silverman [23] for general introductions to functional data analysis). Interest for this paper is in the functional modeling of concentration of particulate matter with an aerodynamic diameter of less than $10\mu m$ in ambient air, measured half-hourly in Graz, Austria. It is widely accepted that exposure to high concentrations can cause respiratory and related health problems. Local policy makers therefore monitor these pollutants closely. The prediction of concentration levels is then a particularly important tool for judging whether measures, such as partial traffic regulation, have to be implemented in order to meet standards set by the European Union.

Providing reliable guesses for future realizations is in fact one of the most important goals of any time series analysis. In the univariate and multivariate framework, this is often achieved by setting up general prediction equations that can be solved recursively by methods such as the Durbin-Levinson and innovations algorithms (see, for example, [10, 24]). Prediction equations may be derived explicitly also for general stationary functional time series (see Section 1.6 of the monograph Bosq [9]) but they seem difficult to solve and implement. As a consequence, much of the research in the area has focused on the first-order functional autoregressive model, shortly FAR(1). Bosq [9] has derived one-step ahead predictors that are based on a functional form of the Yule-Walker equations. Besse et al. [8] have proposed nonparametric kernel predictors and illustrated their methodology by forecasting climatological cycles caused by the El Niño phenomenon. While this paper, and also Besse and Cardot [7], have adapted classical spline smoothing techniques, Antoniadis and Sapatinas [6], see also Antoniadis et al. [4, 5], have studied FAR(1) curve prediction based on linear wavelet methods. Kargin and Onatski [20] have introduced the predictive factor method, which seeks to replace functional principal components with directions most relevant for predictions. Didericksson et al. [13] have evaluated several competing prediction models in a comparative simulation study, finding Bosq's [9] method to have the best overall performance. Other contributions to the area are Aneiros-Pérez et al. [2], and Aneiros-Pérez and Vieu [3].

In spite of its statistical relevance and its mathematical appeal, functional time series modeling has still some unpleasant limitations for the practitioner. First, to date there are not many "ready to use" statistical software packages that can be utilized directly for estimation and prediction purposes. The only available packages that the authors are aware of are the `far` package of Damon and Guillas [12] and the `ftsa` package of Hyndman and Shang [18], both implemented for the statistical software `R` . The lack of tailor-made procedures often requires manual implementation. This may be challenging and therefore restrict use of the methodology to an academic audience. Second, the methodology developed for the FAR(1) case is difficult to generalize. What can be done if an FAR(1) approach is infeasible? In addition, how can exogenous predictors and further lags beyond the first be incorporated? For these cases functional theory and estimation becomes even more complex. Research addressing these questions is scarce, Damon and Guillas [11] being an exception. These authors include exogenous covariates into an autoregressive framework for functional ozone predictions.

The goal of this paper is then to fill in this gap by promoting a simple alternative prediction algorithm which consists of three basic steps, all of which are easy to implement by means of existing software. First, use functional principal components analysis, FPCA, to transform the functional time series observations $Y_1, \ldots, Y_n$ into a vector time series of FPCA scores $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$ of low dimension p, where p is typically no more than 4. Second, fit a vector time series to the FPCA scores and obtain the predictor $\hat{\boldsymbol{Y}}_{n+1}$ for $\boldsymbol{Y}_{n+1}$. Third, utilize the Karhunen-Loève expansion to re-transform $\hat{\boldsymbol{Y}}_{n+1}$ into a curve predictor $\hat{Y}_{n+1}$. The first and the third step are simple and can be performed, for example, with the `fda` package in `R`. The second step may be tackled with standard multivariate time series methodology. This approach is developed in detail in Section 2.

It should be noted that, in this article, the data Y_k is assumed to be given in functional form. In practice one observes only vectors $Y_k(t_1), \ldots, Y_k(t_m)$, with spacings, $t_i - t_{i-1}$, and number of intraday sampling points, m, potentially varying from day to day. The problem of transforming the vector observations into (smooth) functions has been treated in many articles and will not be detailed here. As an excellent starting point for reading in this direction we refer to Chapters 3–7 of Ramsay and Silverman [23].

Functional principal components have been employed in other approaches to functional prediction, for example in Bosq's [9] FAR(1) prediction method or in Aguilera et al. [1]. These, and many other

existing approaches have in common that, roughly speaking, the lagged observation Y_{k-1} is regressed onto Y_k, by minimizing $E\int[Y_k(t)-\Psi(Y_{k-1})(t)]^2dt$ with respect to a linear operator Ψ. The solution of this problem involves an infinite series representation of Ψ along FPCs. (More details will be given in Section 3.1.) In contrast, the proposed approach first uses dimension reduction via FPCA and then fits a model to the reduced data. No a priori knowledge of the functional model is needed and instead of a single estimator, a variety of existing tools for vector processes can be entertained. In particular, further lags or exogenous covariates are easily included into the prediction algorithm (see Section 4).

Hyndman and Ullah [16] and Hyndman and Shang [17] have suggested a curve prediction approach based on modeling FPC scores by scalar time series. They argue that scores are uncorrelated and that hence individual time series can be fit. Depending on the structure of the data, this can be quick and efficient in some cases but less accurate in other cases. The fact that FPC scores have no instantaneous correlation, does, in the time series context, not imply that the ith and jth score are uncorrelated at lags different from zero. Hence univariate modeling may invoke loss of valuable information hidden in the dependence of the data. This will be demonstrated in Section 5 as part of a simulation study.

Although the focus of this paper is on the practical side, some theoretical justification of the proposed approach is given. More specifically, a comparison to Bosq's [9] classical benchmark FAR(1) prediction is made. It is shown that if in the second step of the proposed algorithm a first-order vector autoregression, VAR(1), is fit to the FPCs, then the two forecasts are asymptotically equivalent. Such a type of relationship does not appear evident and will be worked out as part of Section 3.

The remainder of the paper contains a supporting simulation study in Section 5 and an application of the new prediction methodology to the forecasting of intraday patterns of particulate matter concentrations in Section 6. Section 7 concludes and technical proofs are given in Appendix A.

2 Methodology

In what follows, let $(Y_k\colon k\in\mathbb{Z})$ be an arbitrary stationary functional time series. It is assumed that the observations Y_k are elements of the Hilbert space $H=L^2([0,1])$ equipped with the inner product $\langle x,y\rangle=\int_0^1 x(t)y(t)dt$. Each Y_k is therefore a square integrable function satisfying $\|Y_k\|^2=\int_0^1 Y_k^2(t)dt<\infty$. All random functions are defined on some common probability space $(\Omega,\mathcal{A},P)$.

The notation $Y \in L_H^p = L_H^p(\Omega, \mathcal{A}, P)$ is used to indicate that, for some $p > 0$, $E[\|Y\|^p] < \infty$. Any $Y \in L_H^1$ possesses then a mean curve $\mu = (E[Y(t)] \colon t \in [0,1])$, and any $Y \in L_H^2$ a covariance operator C, defined by $C(x) = E[\langle Y - \mu, x\rangle (Y - \mu)]$. The operator C is a kernel operator given by

$$C(x)(t) = \int_0^1 c(t,s)x(s)ds, \qquad c(t,s) = \mathrm{Cov}(X(t), X(s)).$$

As in the multivariate case, C admits the spectral decomposition

$$C(x) = \sum_{\ell=1}^{\infty} \lambda_\ell \langle v_\ell, x\rangle v_\ell,$$

where $(\lambda_\ell \colon \ell \in \mathbb{N})$ are the eigenvalues (in strictly descending order) and $(v_\ell \colon \ell \in \mathbb{N})$ the corresponding normalized eigenfunctions, so that $C(v_\ell) = \lambda_\ell v_\ell$ and $\|v_\ell\| = 1$. Here, $\mathbb{N}$ is the set of positive integers. The $(v_\ell \colon \ell \in \mathbb{N})$ form an orthonormal basis, ONB, of $L^2([0,1])$. Hence Y allows for the Karhunen-Loéve representation $Y = \sum_{\ell=1}^{\infty} \langle Y, v_\ell\rangle v_\ell$. The coefficients $\langle Y, v_\ell\rangle$ in this expansion are called the functional principal components, FPCs, of Y.

Suppose now that we have observed $Y_1, \ldots, Y_n$. In practice μ as well as C and its spectral decomposition will be unknown and need to be estimated from the sample. To estimate μ, let

$$\hat{\mu}_n(t) = \frac{1}{n}\sum_{k=1}^{n} Y_k(t), \qquad t \in [0,1],$$

be the sample mean function. Theorem 4.1 of Hörmann and Kokoszka [15] implies that for a large class of stationary sequences $E[\|\hat{\mu}_n - \mu\|^2] = O(n^{-1})$, thereby showing that $\hat{\mu}_n$ is $\sqrt{n}$-consistent for μ. For this reason estimation of the mean curve can be done in a separate step, and henceforth the more convenient assumption $E[Y_k] = 0$, the zero function, is adopted. The covariance operator and its eigenvalues and eigenfunctions can be estimated using the sample covariance estimator

$$\hat{C}_n(x) = \frac{1}{n}\sum_{k=1}^{n} \langle Y_k - \hat{\mu}_n, x\rangle (Y_k - \hat{\mu}_n),$$

respectively. An application of Theorem 2.1 in Hörmann and Kokoszka [15] yields that this estimator is $\sqrt{n}$-consistent for C. More precisely, $E[\|\hat{C}_n - C\|_{\mathcal{L}}^2] = O(n^{-1})$, where the operator norm $\|\cdot\|_{\mathcal{L}}$ is, for any operator A, defined by

$$\|A\|_{\mathcal{L}} = \sup_{\|x\| \le 1} \|A(x)\|.$$

From $\hat{C}_n$, estimated eigenvalues $\hat{\lambda}_{1,n}, \ldots, \hat{\lambda}_{p,n}$ and estimated eigenfunctions $\hat{v}_{1,n}, \ldots, \hat{v}_{p,n}$ can be computed for an arbitrary fixed, but typically small, $p < n$. These estimators inherit $\sqrt{n}$-consistency from $\hat{C}_n$. For notational convenience, $\hat{\lambda}_\ell$ and $\hat{v}_\ell$ will often be used in place of $\hat{\lambda}_{\ell,n}$ and $\hat{v}_{\ell,n}$.

Functional linear prediction equations for the general case have been derived in Section 1.6 of the monograph Bosq [9]. They appear to be infeasible in most situations. As pointed out in the introduction, the notable exception is the FAR(1) process defined by the stochastic recursion

$$Y_k - \mu = \Psi(Y_{k-1} - \mu) + \varepsilon_k, \qquad k \in \mathbb{Z}, \tag{2.1}$$

where $(\varepsilon_k \colon k \in \mathbb{Z})$ are centered, independent and identically distributed innovations in L_H^2 and Ψ a bounded linear operator satisfying $\|\Psi\|_{\mathcal{L}} < 1$. The latter condition ensures that the recurrence equations (2.1) have a strictly stationary and causal solution. Bosq [9] has in the FAR(1) case used the prediction equations to devise what is now often referred to as the common predictor. This one-step ahead prediction is based on an estimator $\tilde{\Psi}_n$ of Ψ and then given by $\tilde{Y}_{n+1} = \tilde{\Psi}_n Y_n$. Details of this method are given in Section 3, where it will be used as a benchmark to compare with the novel methodology to be introduced in the following. The new prediction technique avoids estimating operators directly and instead utilizes existing multivariate prediction methods.

The proposed prediction algorithm proceeds in three steps. First, select p, the number of principal components to be included in the analysis, for example by ensuring that a certain fraction of the data variation is explained. With the sample eigenfunctions, empirical FPC scores $y_{k,\ell}^e = \langle Y_k, \hat{v}_\ell \rangle$ can now be computed for each combination of observations Y_k, $k = 1, \ldots, n$, and sample eigenfunction $\hat{v}_\ell$, $\ell = 1, \ldots, p$. The superscript e emphasizes that empirical versions are considered. Create from the FPC scores the vectors

$$\boldsymbol{Y}_k^e = (y_{k,1}^e, \ldots, y_{k,p}^e)',$$

where $'$ signifies transposition. By nature of FPCA, the vector $\boldsymbol{Y}_k^e$ contains most of the information on the curve Y_k. In the second step, fix the prediction lag h. Then, use multivariate prediction techniques to produce the h-step ahead prediction

$$\hat{\boldsymbol{Y}}_{n+h}^e = (\hat{y}_{n+h,1}^e, \ldots, \hat{y}_{n+h,p}^e)'$$

given the vectors $\boldsymbol{Y}_1^e, \ldots, \boldsymbol{Y}_n^e$. Standard methods such as the Durbin-Levinson and innovations algorithm can be readily applied, but other options such as exponential smoothing and nonparametric prediction algorithms are available as well. In the third and last step, the multivariate predictions are re-transformed to functional objects. This conversion is achieved by defining the truncated Karhunen-Loéve representation

$$\hat{Y}_{n+h} = \hat{y}_{n+h,1}^e \, \hat{v}_1 + \cdots + \hat{y}_{n+h,p}^e \hat{v}_p \tag{2.2}$$

based on the predicted FPC scores $\hat{y}^e_{k,\ell}$ and the estimated eigenfunctions $\hat{v}_\ell$. The resulting $\hat{Y}_{n+h}$ is then used as the h-step ahead functional prediction of Y_{n+h}. The three prediction steps are summarized in Algorithm 1.

Algorithm 1 Functional Prediction

1. Fix p. For $k = 1, \ldots, n$, use the data $Y_1, \ldots, Y_n$ to compute the vectors
$$\boldsymbol{Y}^e_k = (y^e_{k,1}, \ldots, y^e_{k,p})',$$
containing the first p empirical FPC scores $y^e_{k,\ell} = \langle Y_k, \hat{v}_\ell \rangle$.
2. Fix h. Use $\boldsymbol{Y}^e_1, \ldots, \boldsymbol{Y}^e_n$ to determine the h-step ahead prediction
$$\hat{\boldsymbol{Y}}^e_{n+h} = (\hat{y}^e_{n+h,1}, \ldots, \hat{y}^e_{n+h,p})'$$
for $\boldsymbol{Y}^e_{n+h}$ with an appropriate multivariate algorithm.
3. Use the functional object
$$\hat{Y}_{n+h} = \hat{y}^e_{n+h,1}\, \hat{v}_1 + \ldots + \hat{y}^e_{n+h,p} \hat{v}_p$$
as h-step ahead prediction for Y_{n+h}.

Several remarks are in order. The proposed algorithm is conceptually simple and allows for several immediate extensions and improvements as it is not bound by an assumed FAR(1) structure and, in fact, any other particular functional time series specification. This is important because there is no well developed theory for functional versions of the the well-known linear ARMA time series models ubiquitous in univariate and multivariate settings. Moreover, if an FAR(1) structure is indeed imposed on $(Y_k \colon k \in \mathbb{Z})$, then it appears plausible that $\boldsymbol{Y}^e_1, \ldots, \boldsymbol{Y}^e_n$ should approximately follow an VAR(1) model. This statement will be made precise in Section 3.

The FAR(1) model should in practice be employed only if it provides a reasonable approximation to the unknown underlying dynamics. To allow for more flexible predictions, higher-order FAR processes could be studied. There is, however, until now no method available in the literature that would aid practitioners in determining the appropriate order of a functional autoregressive process and their application for prediction purposes appears therefore to be of little practical use. The proposed methodology avoids these difficulties. It can, in fact, be applied to any stationary functional time series. For example, by utilizing the multivariate innovations algorithm (see Section 11.4 in [10]) in the second step of Algorithm 1. How this is done in the present prediction setting is outlined in

Algorithm 2 for the case $h = 1$.

Algorithm 2 The Innovations Algorithm for Step 2 in Algorithm 1

1. Fix $m \in \{1, \ldots, n\}$. The last m observations will be used to compute the predictor.
2. For $k = 0, 1, \ldots, m$, compute

$$\hat{\Gamma}(k) = \frac{1}{k} \sum_{j=1}^{k} (\hat{\boldsymbol{Y}}_j^e - \bar{\boldsymbol{Y}}^e)(\hat{\boldsymbol{Y}}_j^e - \bar{\boldsymbol{Y}}^e)',$$

   where $\bar{\boldsymbol{Y}}^e = \frac{1}{n} \sum_{k=1}^{n} \hat{\boldsymbol{Y}}_k^e$.

3. Set

$$\hat{\boldsymbol{Y}}_{n+1}^e = \sum_{j=1}^{m} \Theta_{mj} (\boldsymbol{Y}_{n+1-j}^e - \hat{\boldsymbol{Y}}_{n+1-j}^e),$$

   where

$$\begin{aligned}
&\Theta_{00} = \hat{\Gamma}(0), \\
&\Theta_{m,m-k} = \left(\hat{\Gamma}(n-k) - \sum_{j=0}^{k-1} \Theta_{m,m-j} \Theta_{j0} \Theta'_{k,k-j} \right) \Theta_{k0}^{-1}, \qquad k = 0, \ldots, m-1, \\
&\Theta_{m0} = \hat{\Gamma}(0) - \sum_{j=0}^{m-1} \Theta_{m,m-j} \Theta_{j0} \Theta'_{m,m-j}.
\end{aligned}$$

   The recursion is solved in the order $\Theta_{00}; \Theta_{11}, \Theta_{10}; \Theta_{22}, \Theta_{21}, \Theta_{20}; \ldots$

Instead of the innovations algorithm, standard linear prediction equations can be employed. This is detailed in a more general setting allowing for the inclusion of covariates in Section 4. The advantage of the innovations algorithm is that it can be updated quickly when new observations arrive. Note, however, that updating usually means including further lags into prediction algorithm: $X_{n-m+1}, \ldots, X_{n+1}$ are used to predict X_{n+2}, then $X_{n-m+1}, \ldots, X_{n+2}$ are used to predict X_{n+3}, and so on. In order to apply Algorithm 2 this in turn requires estimation of covariances $\Gamma(k)$ for increasing lag k. Such estimates are less reliable the smaller n and the larger k. Therefore including too many lag values has a negative effect in estimation. The simulation study in Section 5 considers cases for which $m \leq 4$.

If estimated eigenfunctions and the covariance matrices $\hat{\Gamma}(k)$ are replaced by population analogues, then this algorithm gives the best linear prediction (in mean square sense) of the population FPC

scores based on the last m observations. It will be demonstrated in Sections 5 and 6 that the innovations algorithm based predictions are best among a number of competitors when the true model deviates from an FAR(1).

It should be emphasized that the numerical implementation of the new prediction methodology is convenient in `R`. For the first step, FPC score matrix $(\boldsymbol{Y}_1^e \colon \ldots \colon \boldsymbol{Y}_n^e)$ and corresponding empirical eigenfunctions can be readily obtained with the `fda` package. For the second step, forecasting for the FPC scores can be done in another routine step using the `vars` package in case VAR models are employed. The obtained quantities can be easily combined for obtaining (2.2).

To assess the forecast accuracy, a method for computing uniform prediction bands is provided next. The target is to find parameters $\xi_\alpha^{(u)}, \xi_\alpha^{(\ell)} \geq 0$, such that for a given $\alpha \in (0,1)$ and $\gamma : [a,b] \to [0,\infty)$ we have

$$P\left(\hat{Y}_{n+1}(t) - \xi_\alpha^{(\ell)}\gamma(t) \leq Y_{n+1}(t) \leq \hat{Y}_{n+1}(t) + \xi_\alpha^{(u)}\gamma(t) \quad \text{for all } t \in [a,b]\right) = \alpha.$$

There is no a priori restriction on the function γ, but clearly it should account for the structure and variation of the data. Although this problem is very interesting from a theoretical standpoint, only a practical approach for the determination of $\xi_\alpha^{(u)}, \xi_\alpha^{(\ell)}$ and γ is proposed here.

Algorithm 3 Algorithm for determining prediction bands

1. Use the entire sample to compute the p-variate score vectors $\boldsymbol{Y}_1^e, \ldots, \boldsymbol{Y}_n^e$ and the sample FPCs $\hat{v}_1, \ldots, \hat{v}_p$.
2. Fix $0 < L \leq i \leq n-1$ and compute

   $$\hat{Y}_{i+1} = \hat{y}_{i+1,1}^e\, \hat{v}_1 + \cdots + \hat{y}_{i+1,p}^e \hat{v}_p$$

   where $\hat{y}_{i+1,\ell}^e$ $(1 \leq \ell \leq p)$ are components of the one-step ahead prediction obtained from $\boldsymbol{Y}_1^e, \ldots, \boldsymbol{Y}_i^e$ by means of some multivariate algorithm.
3. Define the residuals $\hat{\epsilon}_i = Y_{i+L} - \hat{Y}_{i+L}$, $1 \leq i \leq M := n - L$.
4. Define $\gamma(t) = \mathrm{sd}(\hat{\epsilon}_i(t) \colon 1 \leq i \leq M)$, $t \in [a,b]$.
5. Determine $\xi_\alpha^{(u)}, \xi_\alpha^{(\ell)}$ such that $\alpha \times 100\%$ of the residuals satisfy

   $$-\xi_\alpha^{(\ell)}\gamma(t) \leq \hat{\epsilon}_i(t) \leq \xi_\alpha^{(u)}\gamma(t) \quad \text{for all } t \in [a,b].$$

The purpose of the parameter L is to ensure a reasonable sample size for the predictions in Step 2 of Algorithm 3. The sequence of residuals $(\hat{\epsilon}_i)$ is then expected to be approximately stationary and,

by a law of large numbers effect, to satisfy

$$\frac{1}{M}\sum_{i=1}^{M} I\Big(-\xi_\alpha^{(\ell)}\gamma(t) \le \hat{\epsilon}_i(t) \le \xi_\alpha^{(u)}\gamma(t) \quad \text{for all } t\in[a,b]\Big)$$
$$\approx P\Big(-\xi_\alpha^{(\ell)}\gamma(t) \le Y_{n+1}(t) - \hat{Y}_{n+1}(t) \le \xi_\alpha^{(u)}\gamma(t) \quad \text{for all } t\in[a,b]\Big).$$

Note that, in Step 2, the principal components $\hat{v}_\ell$ $(1 \le \ell \le p)$ have been obtained from the entire sample $Y_1, \ldots, Y_n$ and not just from the first i observations. The choice of γ in Step 4 clearly accounts for the variation of the data. For an intraday time exhibiting a higher volatility there should also be a broader prediction interval. Typically the constants $\xi_\alpha^{(\ell)}$ and $\xi_\alpha^{(u)}$ are chosen equal, but there may be situations when this is not desired.

One advantage of this method is that it does not require particular model assumptions. If two two competing prediction methods exist, then the one which is performing better on the sample will lead to narrower prediction bands. The simulation results reported in Section 5 indicate that Algorithm 3 performs well in finite samples even for moderate sample sizes.

3 Predicting first-order functional autoregression

3.1 The standard predictor

The FAR(1) is the most often applied functional time series model. It will be used here as a benchmark to compare the proposed methodology to. In order to obtain Bosq's [9] predictor, estimation of the autoregressive operator Ψ is briefly discussed. The approach is based on a functional version of the Yule-Walker equations. Let then $(Y_k \colon k \in \mathbb{Z})$ be an FAR(1) time series for which $\mu = 0$ without loss of generality. Applying $E[\langle\cdot, x\rangle Y_{k-1}]$ to (2.1) for any $x \in H$, leads to the relations

$$E[\langle Y_k, x\rangle Y_{k-1}] = E[\langle \Psi(Y_{k-1}), x\rangle Y_{k-1}] + E[\langle \varepsilon_k, x\rangle Y_{k-1}]$$
$$= E[\langle \Psi(Y_{k-1}), x\rangle Y_{k-1}].$$

Let again $C(x) = E[\langle Y_1, x\rangle Y_1]$ be the covariance operator of Y_1 and also let $D(x) = E[\langle Y_1, x\rangle Y_0]$ be the cross-covariance operator of Y_0 and Y_1. If Ψ' denotes the adjoint operator of Ψ, given by the requirement $\langle \Psi(x), y\rangle = \langle x, \Psi'(y)\rangle$, the operator equation $D(x) = C(\Psi'(x))$ is obtained. This formally gives $\Psi(x) = D'C^{-1}(x)$, where $D'(x) = E[\langle Y_0, x\rangle Y_1]$. The operator D' can be estimated by $\hat{D}'(x) = \frac{1}{n-1}\sum_{k=2}^{n}\langle Y_{k-1}, x\rangle Y_k$. A more complicated object is the unbounded operator C^{-1}. Using the spectral decomposition of $\hat{C}_n$, it can be estimated by $\hat{C}_n^{-1}(x) = \sum_{\ell=1}^{p}\hat{\lambda}_\ell^{-1}\langle\hat{v}_\ell, x\rangle\hat{v}_\ell$ for

an appropriately chosen p. Combining these results with an additional smoothing step, using the approximation $Y_k \approx \sum_{\ell=1}^{p} \langle Y_k, \hat{v}_\ell \rangle \hat{v}_\ell$, gives the estimator

$$\tilde{\Psi}_n(x) = \frac{1}{n-1} \sum_{k=2}^{n} \sum_{\ell=1}^{p} \sum_{\ell'=1}^{p} \hat{\lambda}_\ell^{-1} \langle x, \hat{v}_\ell \rangle \langle Y_{k-1}, \hat{v}_\ell \rangle \langle Y_k, \hat{v}_{\ell'} \rangle \hat{v}_{\ell'}. \tag{3.1}$$

for $\Psi(x)$. The foregoing gives rise to the functional predictor

$$\tilde{Y}_{n+1} = \tilde{\Psi}_n(Y_n) \tag{3.2}$$

for Y_{n+1}. This is the estimator of Bosq [9]. In the next section, the predictor (3.2) is compared to the proposed predictor.

3.2 Fitting vector autoregression to FPC scores

The main goals of this section are to show that the one-step predictors $\hat{Y}_{n+1}$ in (2.2), based on fitting VAR(1) models in Step 2 of Algorithm 1, and $\tilde{Y}_{n+1}$ in (3.2) are asymptotically equivalent for FAR(1) processes, and that the FPC score vectors $\boldsymbol{Y}_1^e, \ldots, \boldsymbol{Y}_n^e$ follow indeed a VAR(1) model, albeit a non-standard one. The first statement is justified in the next theorem.

Theorem 3.1. *Assume that a VAR(1) model is fit to $\boldsymbol{Y}_1^e, \ldots, \boldsymbol{Y}_n^e$ by means of ordinary least squares. The resulting predictor* (2.2) *is asymptotically equivalent to* (3.2)*. More specifically,*

$$\|\hat{Y}_{n+1} - \tilde{Y}_{n+1}\| = O_P\left(\frac{1}{\sqrt{n}}\right) \qquad (n \to \infty).$$

The proof of Theorem 3.1 is given in the Section A.1, where moreover the exact difference between the two predictors is detailed. The finite sample performance of $\hat{Y}_{n+1}$ and $\tilde{Y}_{n+1}$ were compared in a simulation study whose results are reported in Section 5.

In case of a VAR(1), Step 2. of Algorithm 1 can be performed with least squares. To explicitly calculate $\hat{\boldsymbol{Y}}_{n+1}^e$, apply $\langle \cdot, \hat{v}_\ell \rangle$ to both sides of $Y_k = \Psi(Y_{k-1}) + \varepsilon_k$ to obtain

$$\begin{aligned} \langle Y_k, \hat{v}_\ell \rangle &= \langle \Psi(Y_{k-1}), \hat{v}_\ell \rangle + \langle \varepsilon_k, \hat{v}_\ell \rangle \\ &= \sum_{\ell'=1}^{\infty} \langle Y_{k-1}, v_{\ell'} \rangle \langle \Psi(v_{\ell'}), \hat{v}_\ell \rangle + \langle \varepsilon_k, \hat{v}_\ell \rangle \\ &= \sum_{\ell'=1}^{p} \langle Y_{k-1}, \hat{v}_{\ell'} \rangle \langle \Psi(\hat{v}_{\ell'}), \hat{v}_\ell \rangle + \delta_{k,\ell}, \end{aligned} \tag{3.3}$$

with remainder terms

$$\delta_{k,\ell} = d_{k,\ell}^{(1)} + d_{k,\ell}^{(2)} + d_{k,\ell}^{(3)} + \langle \varepsilon_k, \hat{v}_\ell \rangle$$

given by

$$d_{k,\ell}^{(1)} = \sum_{\ell'=1}^{p} \langle Y_{k-1}, \hat{v}_{\ell'} \rangle \big(\langle \Psi(v_{\ell'}), \hat{v}_\ell \rangle - \langle \Psi(\hat{v}_{\ell'}, \hat{v}_\ell \rangle \big),$$
$$d_{k,\ell}^{(2)} = \sum_{\ell'=1}^{p} \big(\langle Y_{k-1}, v_{\ell'} \rangle - \langle Y_{k-1}, \hat{v}_{\ell'} \rangle \big) \langle \Psi(v_{\ell'}), \hat{v}_\ell \rangle,$$
$$d_{k,\ell}^{(3)} = \sum_{\ell'=p+1}^{\infty} \langle Y_{k-1}, v_{\ell'} \rangle \langle \Psi(v_{\ell'}), \hat{v}_\ell \rangle.$$

Some notation is needed. Set $\boldsymbol{e}_k = (\langle \varepsilon_k, v_1 \rangle, \ldots, \langle \varepsilon_k, v_p \rangle)'$ and $\boldsymbol{u}_k = (u_{k,1}, \ldots, u_{k,p})'$ where $u_{k,\ell} = \sum_{\ell'>p} \langle Y_{k-1}, v_{\ell'} \rangle \langle \Psi(v_{\ell'}), v_\ell \rangle$, and let $B_p \in \mathbb{R}^{p \times p}$ be the matrix with entry $\langle \Psi(v_\ell), v_{\ell'} \rangle$ in the ℓth row and the ℓ'th column, $\ell, \ell' = 1, \ldots, p$. Let moreover $\boldsymbol{\beta} = \mathrm{vec}(B_p')$, $\boldsymbol{Z} = (\boldsymbol{Y}_2', \ldots, \boldsymbol{Y}_n')'$, $\boldsymbol{E} = (\boldsymbol{e}_2', \ldots, \boldsymbol{e}_n')'$, $\boldsymbol{U} = (\boldsymbol{u}_2', \ldots, \boldsymbol{u}_n')'$, $X_k = I_p \otimes \boldsymbol{Y}_k'$ and $X = (X_1' \colon \ldots \colon X_{n-1}')'$. Replacing the eigenfunctions v_ℓ by their sample counterparts $\hat{v}_\ell$, empirical versions of the above variables are denoted by $\boldsymbol{Y}_k^e$, $\boldsymbol{Z}^e$, X_k^e, X^e, B_p^e and $\boldsymbol{\beta}_p^e$. For a vector $\mathbf{x} \in \mathbb{R}^{p^2}$, the operation $\mathrm{mat}(\mathbf{x})$ creates a $p \times p$ matrix, whose ℓ-th column contains the elements $v_{(1-\ell)p+1}, \ldots, v_{\ell p}$. Define now $\boldsymbol{\delta}_k = (\delta_{k,1}, \ldots, \delta_{k,p})'$ to arrive at the equations

$$\boldsymbol{Y}_k^e = B_p^e \, \boldsymbol{Y}_{k-1}^e + \boldsymbol{\delta}_k, \qquad k = 2, \ldots, n. \tag{3.4}$$

The equations in (3.4) formally resemble VAR(1) equations. Notice, however, that it is a nonstandard formulation, since the errors $\boldsymbol{\delta}_k$ are generally not centered and dependent. Furthermore, $\boldsymbol{\delta}_k$ depends in a complex way on $\boldsymbol{Y}_{k-1}^e$, so that the errors are not uncorrelated with past observations. The coefficient matrix B_p^e is also random, but fixed for fixed sample size n. In the sequel these effects are ignored. Utilizing some matrix algebra, (3.4) can be written as the linear regression

$$\boldsymbol{Z}^e = X^e \boldsymbol{\beta}_p^e + \boldsymbol{\Delta}, \tag{3.5}$$

where $\boldsymbol{\Delta} = (\boldsymbol{\delta}_2', \ldots, \boldsymbol{\delta}_n')'$. The ordinary least squares estimator is then $\hat{\boldsymbol{\beta}}_p^e = (X^{e\prime} X^e)^{-1} X^{e\prime} \boldsymbol{Z}^e$, and the prediction equation

$$\hat{\boldsymbol{Y}}_{n+1}^e = \hat{B}_p^e \boldsymbol{Y}_n^e = (\hat{y}_{n+1,1}^e, \ldots, \hat{y}_{n+1,p}^e)', \tag{3.6}$$

follows directly, defining $\hat{B}_p^e = \mathrm{mat}\big(\hat{\boldsymbol{\beta}}_p^e\big)'$.

3.3 Estimation by GLS

If the functional sequence $(Y_k \colon k \in \mathbb{Z})$ follows an FAR(1) process, then the errors $\boldsymbol{\delta}_k$ in the VAR(1) model (3.4) are correlated. This could be taken into account by applying generalized least squares

(GLS) estimation instead of ordinary least squares. The GLS estimator is

$$\hat{\boldsymbol{\beta}}_p^e(\text{GLS}) = (X^{e\prime}\Omega^{e-1}X^e)^{-1}X^{e\prime}\Omega^{e-1}\boldsymbol{Z}^e, \tag{3.7}$$

where $\Omega^e = \text{Cov}(\boldsymbol{\Delta})$. The prediction procedure leading to (2.2) can then be modified accordingly. In view of the Gauss-Markov theorem, GLS outperforms OLS in the given setting. The main difficulty in the implementation of (3.7) is to obtain an adequate estimator for Ω^e. To achieve this goal, information on the errors $\boldsymbol{\delta}_k$ of model (3.4) needs to be extracted. Since they cannot be directly observed, a two-step procedure is proposed. In the first step, the model is estimated by OLS. We denote the resulting residuals by

$$\hat{\boldsymbol{\delta}}_k = \boldsymbol{Y}_k^e - \hat{B}_p^e \boldsymbol{Y}_{k-1}^e.$$

In the second step, the residuals $\hat{\boldsymbol{\delta}}_k$ are used to obtain an estimator for Ω^e, which can then be plugged into (3.7). The estimator we propose then is obtained as follows. Let O be the zero matrix in $\mathbb{R}^{p\times p}$. For $b \in \{1,\dots,n-1\}$, define $V_h^{(b)} = O$ if $h > b$ and

$$V_h^{(b)} = \frac{1}{n}\sum_{k=1}^{n-h}(\hat{\boldsymbol{\delta}}_{k+h} - \bar{\boldsymbol{\delta}}_n)(\hat{\boldsymbol{\delta}}_k - \bar{\boldsymbol{\delta}}_n)' \qquad h = 0,1,\dots,b,$$

where $\bar{\boldsymbol{\delta}}_n = \frac{1}{n}\sum_{k=1}^n \hat{\boldsymbol{\delta}}_k$. Set then

$$\hat{\Omega}_b^e = \begin{pmatrix} V_0^{(b)} & V_1^{(b)} & \cdots & V_{n-1}^{(b)} \\ V_{-1}^{(b)} & V_0^{(b)} & \cdots & V_{n-2}^{(b)} \\ \vdots & \vdots & \ddots & \vdots \\ V_{-n+1}^{(b)} & V_{-n+2}^{(b)} & \cdots & V_0^{(b)} \end{pmatrix}.$$

The parameter b determines the number of lags which are taken into account for the estimation of the cross-covariances. Choosing b small has the advantage that the resulting banded matrix $\hat{\Omega}_b^e$ is easier to invert. Note that inversion can be a difficult problem as $\hat{\Omega}_b^e$ has np rows. It will be shown in Appendix A.2 that the correlation of the model errors $\hat{\delta}_k$ decays exponentially fast, thereby justifying the choice of small b in the estimation. Furthermore, it is evident that the estimator for the cross-covariances at higher lags are more volatile, as the number of observations which are far enough apart becomes small. On the other hand, if b is chosen too small, much of the dependence could potentially be ignored. A discussion on related issues may be found, for example, in Section 3.2.1 of Lütkepohl [22].

The simulations reported below indicate that the gains in efficiency for GLS are negligible in the settings considered. This is arguably due to the fact that possible improvements may be significant

only for small sample sizes. In this case, however, estimation of Ω_b^e is almost infeasible. GLS maybe applied if there is some preliminary estimate for Ω_b^e, for example obtained from historical data.

4 Prediction with covariates

In many practical problems, such as in the particulate matter example presented in Section 6, predictions could not only contain lagged values of the functional time series of interest, but also other exogenous covariates. These covariates might be scalar, vector-valued and functional. Formally the goal is then to obtain a predictor $\hat{Y}_{n+h}$ given observations of the curves $Y_1, \ldots, Y_n$ and a number of covariates $X_n^{(1)}, \ldots, X_n^{(r)}$. The exogenous variables need not be defined on the same space. ($X_n^{(1)}$ could be scalar, $X_n^{(2)}$ a function and $X_n^{(3)}$ could contain lagged values of $X_n^{(2)}$). The following adaptation of the methodology given in Algorithm 1 is derived under the assumption that $(Y_k \colon k \in \mathbb{Z})$ as well as the covariates $(X_n^{(i)} \colon n \in \mathbb{N})$ are stationary processes in their respective spaces. The modified procedure is summarized in Algorithm 4.

Algorithm 4 Functional Prediction with Exogenous Covariates

1. (a) Fix p. For $k = 1, \ldots, n$, use the data $Y_1, \ldots, Y_n$ to compute the vectors
$$\boldsymbol{Y}_k^e = (y_{k,1}^e, \ldots, y_{k,p}^e)',$$
containing the first p empirical FPC scores $y_{k,\ell}^e = \langle Y_k, \hat{v}_\ell \rangle$.
(b) For a functional covariate, fix q. For $k = 1, \ldots, n$, use the data $X_1, \ldots, X_n$ to compute the vectors
$$\boldsymbol{X}_k^e = (x_{k,1}^e, \ldots, x_{k,p}^e)',$$
containing the first q empirical FPC scores $x_{k,\ell}^e = \langle X_k, \hat{w}_\ell \rangle$. Repeat this step for each functional covariate.
(c) Combine all covariate vectors into one vector $\boldsymbol{R}_n^e = (R_{n1}^e, \ldots, R_{nr}^e)'$.
2. Fix h. Use $\boldsymbol{Y}_1^e, \ldots, \boldsymbol{Y}_n^e$ and $\boldsymbol{R}_n^e$ to determine the h-step ahead prediction
$$\hat{\boldsymbol{Y}}_{n+h}^e = (\hat{y}_{n+h,1}^e, \ldots, \hat{y}_{n+h,p}^e)'$$
for $\boldsymbol{Y}_{n+h}^e$ with an appropriate multivariate algorithm.
3. Use the functional object
$$\hat{Y}_{n+h} = \hat{y}_{n+h,1}^e \, \hat{v}_1 + \cdots + \hat{y}_{n+h,p}^e \hat{v}_p$$
as h-step ahead prediction for Y_{n+h}.

The first step of Algorithm 4 is expanded compared to Algorithm 1. Step 1(a) performs FPCA on the response time series curves $Y_1, \ldots, Y_n$. In Step 1(b), all functional covariates are first transformed via FPCA into empirical FPC score vectors. For each functional covariate, a different number of principal components can be selected. Vector-valued and scalar covariates can be used directly. All exogenous covariates are finally combined into one vector $\boldsymbol{R}_n^e$ in Step 1(c).

Details for Step 2 and the one-step ahead prediction case $h = 1$ could be as follows. Since stationarity is assumed for all involved processes, the resulting FPC scores form stationary time series. Define hence

$$\Gamma_{\boldsymbol{YY}}(i) = \mathrm{Cov}(\boldsymbol{Y}_k^e, \boldsymbol{Y}_{k-i}^e), \qquad \Gamma_{\boldsymbol{YR}}(i) = \mathrm{Cov}(\boldsymbol{Y}_k^e, \boldsymbol{R}_{k-i}^e), \qquad \Gamma_{\boldsymbol{RR}} = \mathrm{Cov}(\boldsymbol{R}_k^e, \boldsymbol{R}_k^e)$$

and notice that these matrices are independent of k. Fix again $m \in \{1, \ldots, n\}$. The best linear predictor $\hat{\boldsymbol{Y}}_{n+1}^e$ of $\boldsymbol{Y}_{n+1}^e$ given the vector variables $\boldsymbol{Y}_n^e, \ldots, \boldsymbol{Y}_{n-m+1}^e, \boldsymbol{R}_n^e$ can be obtained by projecting each component $y_{n+1,\ell}^e$ of $\boldsymbol{Y}_{n+1}^e$ onto $\overline{\mathrm{sp}}\{y_{k,i}^e, R_{nj}^e | \, 1 \le i \le p, \, 1 \le j \le r, \, n-m+1 \le k \le n\}$. Then there exist $p \times p$ matrices Φ_i and a $p \times r$ matrix Θ, such that

$$\hat{\boldsymbol{Y}}_{n+1}^e = \Phi_1 \boldsymbol{Y}_n^e + \Phi_2 \boldsymbol{Y}_{n-1}^e + \cdots + \Phi_m \boldsymbol{Y}_{n-m+1}^e + \Theta \boldsymbol{R}_n^e.$$

Using the projection theorem, it can be easily shown that the matrices $\Phi_1, \ldots, \Phi_m, \Theta$ are characterized by the equations

$$\Gamma_{\boldsymbol{YY}}(i+1) = \Phi_1 \Gamma_{\boldsymbol{YY}}(i) + \cdots + \Phi_m \Gamma_{\boldsymbol{YY}}(i+1-m) + \Theta \Gamma_{\boldsymbol{RY}}(i), \quad i = 0, \ldots, m-1;$$

$$\Gamma_{\boldsymbol{YR}}(1) = \Phi_1 \Gamma_{\boldsymbol{YR}}(0) + \cdots + \Phi_m \Gamma_{\boldsymbol{YR}}(1-m) + \Theta \Gamma_{\boldsymbol{RR}}.$$

Let

$$\Gamma = \begin{pmatrix} \Gamma_{\boldsymbol{YY}}(0) & \Gamma_{\boldsymbol{YY}}(1) & \cdots & \Gamma_{\boldsymbol{YY}}(m-1) & \Gamma_{\boldsymbol{YR}}(0) \\ \Gamma_{\boldsymbol{YY}}(-1) & \Gamma_{\boldsymbol{YY}}(0) & \cdots & \Gamma_{\boldsymbol{YY}}(m-2) & \Gamma_{\boldsymbol{YR}}(-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Gamma_{\boldsymbol{YY}}(1-m) & \Gamma_{\boldsymbol{YY}}(2-m) & \cdots & \Gamma_{\boldsymbol{YY}}(0) & \Gamma_{\boldsymbol{YR}}(1-m) \\ \Gamma_{\boldsymbol{RY}}(0) & \Gamma_{\boldsymbol{RY}}(1) & \cdots & \Gamma_{\boldsymbol{RY}}(m-1) & \Gamma_{\boldsymbol{RR}}(0) \end{pmatrix}.$$

Assuming that Γ has full rank, it follows that

$$(\Phi_1, \Phi_2, \ldots, \Phi_m, \Theta) = (\Gamma_{\boldsymbol{YY}}(1), \ldots, \Gamma_{\boldsymbol{YY}}(m), \Gamma_{\boldsymbol{YR}}(1))\Gamma^{-1}.$$

The matrices $\Gamma_{\boldsymbol{YY}}(i)$, $\Gamma_{\boldsymbol{YR}}(i)$ and $\Gamma_{\boldsymbol{RR}}$ have to be replaced in practice by the corresponding sample versions. This explains why predictions should not be made conditional on all data $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$. It would involve the matrices $\Gamma_{\boldsymbol{YY}}(n), \Gamma_{\boldsymbol{YY}}(n-1), \ldots$ which cannot be reasonably estimated from the sample.

5 Simulations

To analyze the finite sample properties of the various prediction methods, a comparative simulation study was conducted and the results are reported in this section. The set-up consisted of fifteen cubic B-spline functions $v_1, \ldots, v_{15}$ on the unit interval $[0, 1]$, which together determine the (finite-dimensional) space $H = \mathrm{sp}\{v_1, \ldots, v_{15}\}$. Innovations were defined by setting

$$\varepsilon_k(t) = \sum_{\ell=1}^{15} A_{k,\ell} v_\ell(t), \tag{5.1}$$

where $(A_{k,1}, \ldots, A_{k,15})'$ were i.i.d. random vectors with mean zero and independent t_4-distributed components. The four degrees of freedom were selected in order to keep the simulations relevant for the application to pollution concentrations in Section 6 for which underlying Gaussianity is an unrealistic assumption. However, simulations using normal instead of t_4 distributed errors have lead to very similar conclusions and are thus not reported here. The prediction methods were tested on on three functional time series, namely

(a) FAR(1): $X_k = \Psi(X_{k-1}) + \varepsilon_k$,

(b) FAR(2): $X_k = \Psi_1(X_{k-1}) + \Psi_2(X_{k-2}) + \varepsilon_k$,

(c) FMA(1): $X_k = \Theta(\varepsilon_{k-1}) + \varepsilon_k$.

To generate the functional autoregressive time series in (a) the starting value $X_{-9} = \sum_{\ell=1}^{10} N_\ell v_\ell$, with a normal random vector $(N_1, \ldots, N_{10})' \sim \mathcal{N}(\mathbf{0}, I_{10})$, was utilized. For (b), $X_{-10} = \sum_{\ell=1}^{10} \tilde{N}_\ell v_\ell$, with $(\tilde{N}_1, \ldots, \tilde{N}_{10})' \sim \mathcal{N}(\mathbf{0}, I_{10})$, was constructed in a similar fashion. The first ten elements $X_{-9}, \ldots, X_0$ were used for a burn-in in both cases.

Note that an arbitrary element in H has the representation $x(t) = \sum_{\ell=1}^{15} c_\ell v_\ell(t)$ with coefficients $\mathbf{c} = (c_1, \ldots, c_{15})'$. If $\Psi\colon H \to H$ is a linear operator, then

$$\begin{aligned}
\Psi(x) &= \sum_{\ell=1}^{15} c_\ell \Psi(v_\ell) \\
&= \sum_{\ell=1}^{15} \sum_{\ell'=1}^{15} c_\ell \langle \Psi(v_\ell), v_{\ell'} \rangle v_{\ell'} \\
&= (\Psi \mathbf{c})' \mathbf{v},
\end{aligned}$$

where, by a slight abuse of notation, Ψ is also the matrix whose ℓ'-th row and ℓ-th column is $\langle \Psi(v_\ell), v_{\ell'} \rangle$ and $\mathbf{v} = (v_1, \ldots, v_{15})'$ is the vector of basis functions. The linear operator can thus be

represented by a 15×15 matrix that operates on the coefficients in the basis function representation of the curves. For the FAR(1) process (a), Ψ was chosen to take the value 1/8 in the diagonal, 1/2 in first lower off-diagonal, and 1/8 in the second lower off-diagonal, and zero else. For the FAR(2) process (b), $\Psi_1 = \mathrm{diag}(1/4, \ldots, 1/4)$, Ψ_2 was chosen to take the value 1/3 in the diagonal, 1/8 in the first and second lower off-diagonals, and zero else. For the FMA(1) process (c), $\Theta = \mathrm{diag}(1, \ldots, 1)$.

The simulation results are shown in Tables 2–4. They are based on 5,000 repetitions for each setting. For each of the processes in (a), (b) and (c) the sample size n and number of FPC scores p considered for the prediction were varied. In practice one could employ cross-validation to obtain the optimal choice of p for each setting under consideration. Since the focus of this simulation study is on a comparison of different predictors rather than their calibration, the same values of p were used across all methods. For all combinations, median ($\mathrm{MED}_{\mathtt{Pr}}$), mean ($\mathrm{MSE}_{\mathtt{Pr}}$) and standard deviation ($\mathrm{SD}_{\mathrm{Pr}}$) of the squared prediction errors $\int_0^1 [\mathtt{Pr}(X_{n+1})(t) - X_{n+1}(t)]^2 dt$ for the 5,000 repetitions were computed. Here `Pr` stands for any of the prediction methods considered. For the `FAR` setting, Tables 2–4 display $\mathrm{MED}_{\mathtt{FAR}}$, $\mathrm{MSE}_{\mathtt{FAR}}$ and $\mathrm{SD}_{\mathtt{FAR}}$. For ease of comparison, results for all other methods are reported relative to the forecasts obtained from the benchmark prediction method (3.2), that is

$$\mathrm{RMED}_{\mathtt{Pr}} = \frac{\mathrm{MED}_{\mathtt{Pr}}}{\mathrm{MED}_{\mathtt{FAR}}}, \quad \mathrm{RMSE}_{\mathtt{Pr}} = \frac{\mathrm{MSE}_{\mathtt{Pr}}}{\mathrm{MSE}_{\mathtt{FAR}}}, \quad \mathrm{RSD}_{\mathtt{Pr}} = \frac{\mathrm{SD}_{\mathtt{Pr}}}{\mathrm{SD}_{\mathtt{FAR}}}.$$

It should be noted that an approximate 95% confidence interval for the expected squared prediction error $E \int_0^1 [\mathtt{Pr}(X_{n+1})(t) - X_{n+1}(t)]^2 dt$ is

$$\mathrm{MSE}_{\mathtt{Pr}} \pm \frac{1.96}{\sqrt{5000}} \mathrm{SD}_{\mathtt{Pr}}. \tag{5.2}$$

Specifically, one-step predictions $\mathtt{Pr}(X_{n+1})$ were obtained from Algorithm 1 based on the sample $X_1, \ldots, X_n$, in Step 2 using

- `VAR.ls`: fitting a VAR(1) to FPC scores by OLS;
- `VAR.gls`: fitting a VAR(1) to FPC scores by GLS and banding parameter $b = 1$, and
- `Inno`: employing the innovations algorithm (i.e. linear prediction) with different values of m.

Reported are also predictions `Mean` ($\mathtt{Pr}(X_{n+1}) = 0$) and `Naive` ($\mathtt{Pr}(X_{n+1}) = X_n$) which were clearly outperformed by the other methods. The procedures based on `VAR.ls` and `VAR.gls` performed almost identical. The latter method performed poorly for small sample sizes and $p = 3$, in which case estimation of Ω^e can be problematic, but shows a slight advantage for all sample sizes when

$p = 2$. Setting aside prediction of the FAR(1) time series, for which all methods gave roughly the same results, the application of the innovations algorithm led to improvements for virtually all constellations of n and p under consideration.

A special case of the method proposed in this paper has been pursued by Hyndman and Ullah [16] and Hyndman and Shang [17]. Motivated by the fact that PCA score vectors have uncorrelated components, these authors have proposed predicting the scores individually as univariate time series. This will be referred to as the *scalar method*, in contrast to the *vector method* promoted in this paper. The scalar method is fast and works well as long as the cross-spectra related to the score vectors are close to zero. However, in general the score vectors have non-diagonal autocorrelations. Then, scalar models are not theoretically justified. To explore the effect of neglecting cross-sectional dependence, functional time series of intermediate sample size $n = 200$ were generated as described above, focusing on the FAR(1) process described in **(a)**. Two choices for Ψ were considered. For the first, the corresponding matrix $\Psi^a = \mathrm{diag}(0.8, \ldots, 0.8)$, while the corresponding matrix Ψ^b of the second operator is chosen to have i.i.d. standard normal entries. To ensure stationary solutions, the normalization $\|\Psi^b\| = 0.8$ was applied. For each of the 5,000 repetitions, the log differences of the squared prediction errors obtained from the vector method and the scalar method were compared, see Figure 1. A value larger (smaller) than 0 indicates that the scalar method (vector method) performed better. For the vector method predictions, $p = 3$ was fixed and the innovations algorithm employed, while for the scalar method predictions were obtained by fitting AR(1) models to each score sequence. Of course, in practice one would use more sophisticated model selection procedures for both approaches, but this analysis showed that the improvement is marginal and the conclusion remains unchanged. While both methods perform almost identical in the case of the simple operator Ψ^a, a clear skewness to the left emerges in the ratios for the non-structured operator Ψ^b. It can be concluded that, depending on the complexity of the underlying functional time series, univariate modeling of FPC scores may be quite efficient, but can also lead to less accurate forecasts.

The final simulation experiment is related to Algorithm 3. Functional time series of size $n = 200$ were generated as before, using the operator Ψ^b. This time, $p = 2$ and $p = 3$ were chosen. Algorithm 3 was applied with $L = 60$ on the first 199 observations. Then observation 200 was predicted and it was checked if it is covered by the functional prediction band obtained from the algorithm. This experiment was repeated 1,000 times. Table 1 shows the comparison between the desired coverage

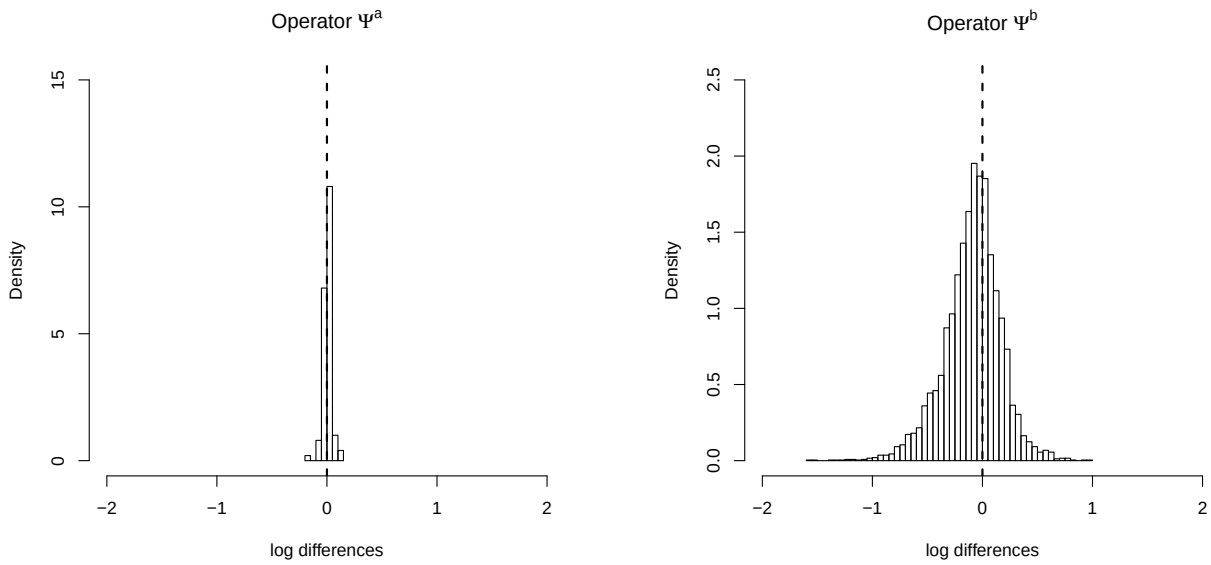

Figure 1: Log differences of squared prediction errors for operator Ψ^a (left panel) and Ψ^b (right panel).

rate α and the empirical coverage probabilities `CP`. The performance is excellent when $p = 2$ and still very good when $p = 3$. Obviously, larger p will necessitate larger sample sizes.

α	0.8	0.9	0.95
`CP` $(p = 2)$	0.791	0.895	0.951
`CP` $(p = 3)$	0.784	0.881	0.928

Table 1: Coverage rate α compared to the empirical coverage probabilities `CP`.

6 Predicting particulate matter concentrations

In order to demonstrate its practical usefulness, the new methodology has been applied to environmental data on pollution concentrations. The observations are half-hourly measurements of the concentration (measured in μgm^{-3}) of particulate matter with an aerodynamic diameter of less than $10\mu m$, abbreviated PM10, in ambient air taken in Graz, Austria from October 1, 2010 until March 31, 2011. Since epidemiological and toxicological studies have pointed to negative health effects, European Union (EU) regulation sets pollution standards for the level of the concentration. Policy makers have to ensure compliance with these EU rules and need reliable statistical tools to determine, and justify to the public, appropriate measures such as partial traffic regulation (see Stadlober et

al. [25]). Accurate predictions are therefore paramount for well informed decision making.

Functional data were obtained as follows. In a first step, very few missing intra-day data points were replaced through linear interpolation. A square-root transformation was then applied to the data in order to stabilize the variance. A visual inspection of the data revealed several extreme outliers around New Years Eve known to be caused by firework activities. The corresponding week was removed from the sample. The data was then centered and adjusted for weekly seasonality by subtracting from each observation the corresponding weekday average. This is done because PM10 concentration levels are significantly different for the weekends when traffic volume is much lower. In the next step, 48 observations for a given day were combined into vectors and transformed into functional data using ten cubic B-spline basis functions and least squares fitting. The `fda` package available for the statistical software `R` was applied here. Eventually, 175 daily functional observations, say, $Y_1, \ldots, Y_{175}$, were obtained, roughly representing one winter season for which pollution levels are known to be high. They are displayed in the upper left panel of Figure 6. Shown in this figure are also the effect of the first three FPCs on the mean curve. Following Ramsay and Silverman [23], a multiple (using the factor .5) of the ℓth empirical eigenfunction $\hat{v}_\ell$ was added to and subtracted from the overall estimated mean curve $\hat{\mu}$ to study the effect of large (small) first, second or third FPC. Notice that

$$Y_k \approx \hat{\mu} + y^e_{k1}\hat{v}_1 + y^e_{k2}\hat{v}_2 + y^e_{k3}\hat{v}_3, \qquad k = 1, \ldots, 175,$$

where $y^e_{k\ell} = \langle Y_k, \hat{v}_\ell \rangle$ are the empirical FPC scores. These combine to explain about 89% of variability in the data. The upper right panel of Figure 6 indicates that if the first FPC score y^e_{k1}, which explains about 72% of the variation, is large (small), then a positive (negative) shift of the mean occurs. The second and third FPCs are contrasts, explaining respectively 10% and 7% of variation, with the second FPC describing differences in the first and second half of the day and the third FPC indicating whether the diurnal peaks are more or less pronounced (see the lower panel of Figure 6).

For the comparison of the quality of the various prediction methods, the following was adopted. Consecutive functional observations $Y_k, \ldots, Y_{k+n-1}$ for selected values of $1 < n < 175$ were chosen and used for estimation and prediction. Then, squared prediction errors

$$\int_0^1 \big[Y_{n+k}(t) - \mathtt{Pr}(Y_{n+k})(t)\big]^2 dt, \qquad k = 1, \ldots, 175 - n =: N,$$

were computed, where $\mathtt{Pr}(Y_{k+n})$ can stand for any of the prediction methods introduced in Section 5,

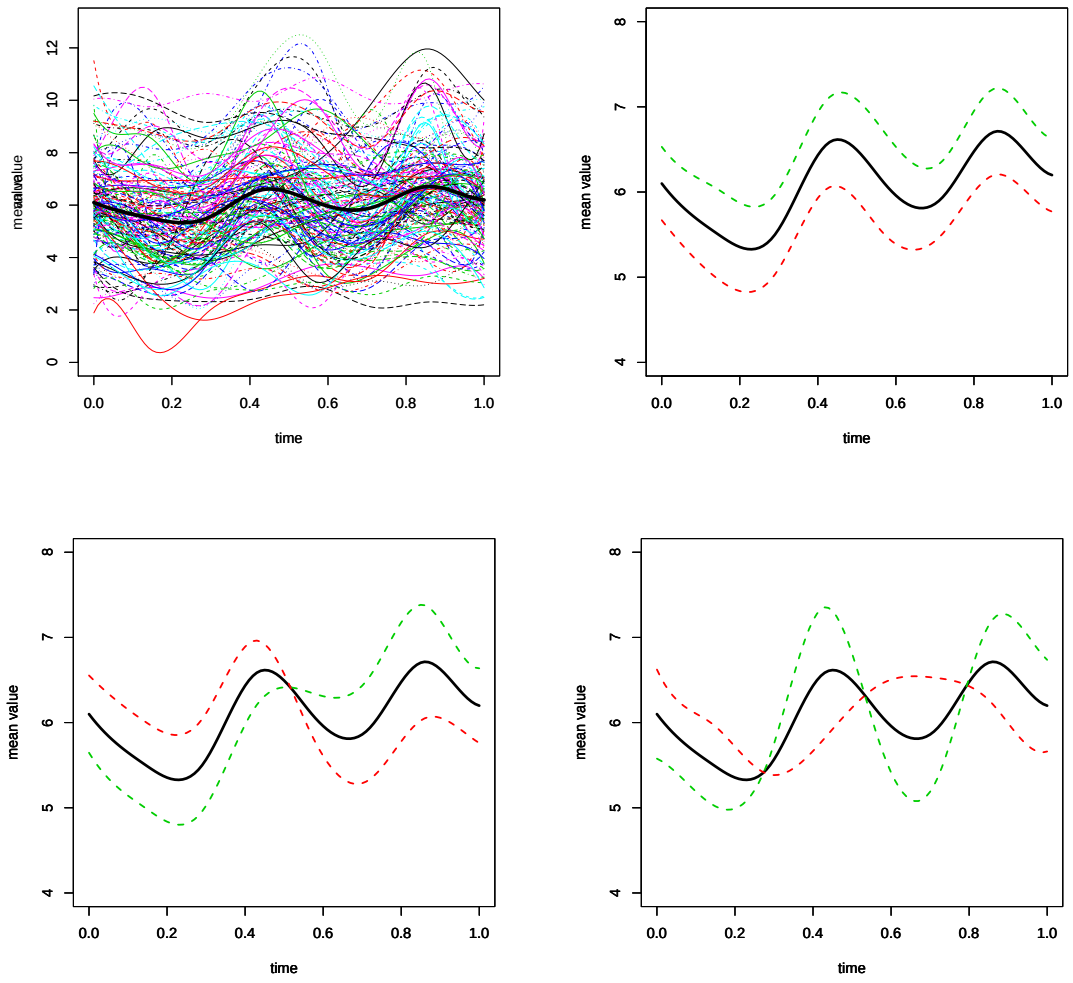

Figure 2: Square-root transformed `PM10` observations with fat overall mean curve (upper left panel), effect of the first FPC (upper right panel), effect of the second FPC (lower left panel), and effect of the third FPC (lower right panel).

noting again that $\texttt{Pr}(Y_{k+n})$ is based solely on observations $Y_k, \ldots, Y_{k+n-1}$. From the N resulting numbers median ($\text{MED}_{\texttt{Pr}}$), mean ($\text{MSE}_{\texttt{Pr}}$) and standard deviation ($\text{SD}_{\texttt{Pr}}$) were computed. This procedure was performed for the values $n = 20, 40, 60$ and 80. Results are reported in Table 5. Confidence intervals for the expected square prediction error $E \int_0^1 \left[Y_{n+k}(t) - \texttt{Pr}(Y_{n+k})(t)\right]^2 dt$ may be obtained analogously to (5.2).

It can be seen that the prediction methods `Mean` and `Naive` are not competitive. The methods `VAR.ls` and `FAR` give almost identical results for mean squared prediction error, thus corroborating the theoretical findings. In accordance with the simulation study the innovations algorithms with $m = 2$ and $m = 3$ generally provide the best predictions among the methods that do not invoke covariates. While the mean of the squared prediction errors is not much different for `VAR.ls`, `FAR` and `Inno` and $n = 20, 40$, differences become apparent for the larger choices $n = 60, 80$. The findings are similar for the medians. Since the first three principal components already describe close to 90% of the data variability, the addition of another FPC score ($p = 4$) did not lead to further improvements and prediction results for this case are hence not displayed.

PM10 concentrations are known to be high at locations suffering from severe temperature inversions such as the basin areas of the Alps. Following Stadlober et al. [25], temperature difference between Graz ($350m$ above sea level) and Kalkleiten ($710m$ above sea level) can be utilized to model this phenomenon. Temperature inversion is often seen as a key factor influencing PM10 concentrations because temperatures increasing with sea level result in a sagging exchange of air, thereby yielding a higher pollutant load at the lower elevation. This has been impressively captured in a study conducted on behalf of the ZAMG Regionalstelle Steiermark, for which balloon probes were used to analyze the diffusion of PM10 with respect to local meteorological variables. Detailed explanations of the experiment may be found in [21]. As a graphical illustration, PM10 concentrations at different altitude (vertical axis) and times (horizontal axis, between 6.10 am and 8.15 pm) are displayed in the left panel of Figure 6, while the right panel shows the corresponding temperature values. The peaks at ground level around 9 am and 6 pm coincide with rush hour traffic. When temperature inversion begins to weaken, air exchange among different atmospheric layers leads to an almost uniform vertical spread of PM10 and later to a decrease of pollution concentration at ground level.

To illustrate functional prediction with covariates, temperature difference curves of Graz and

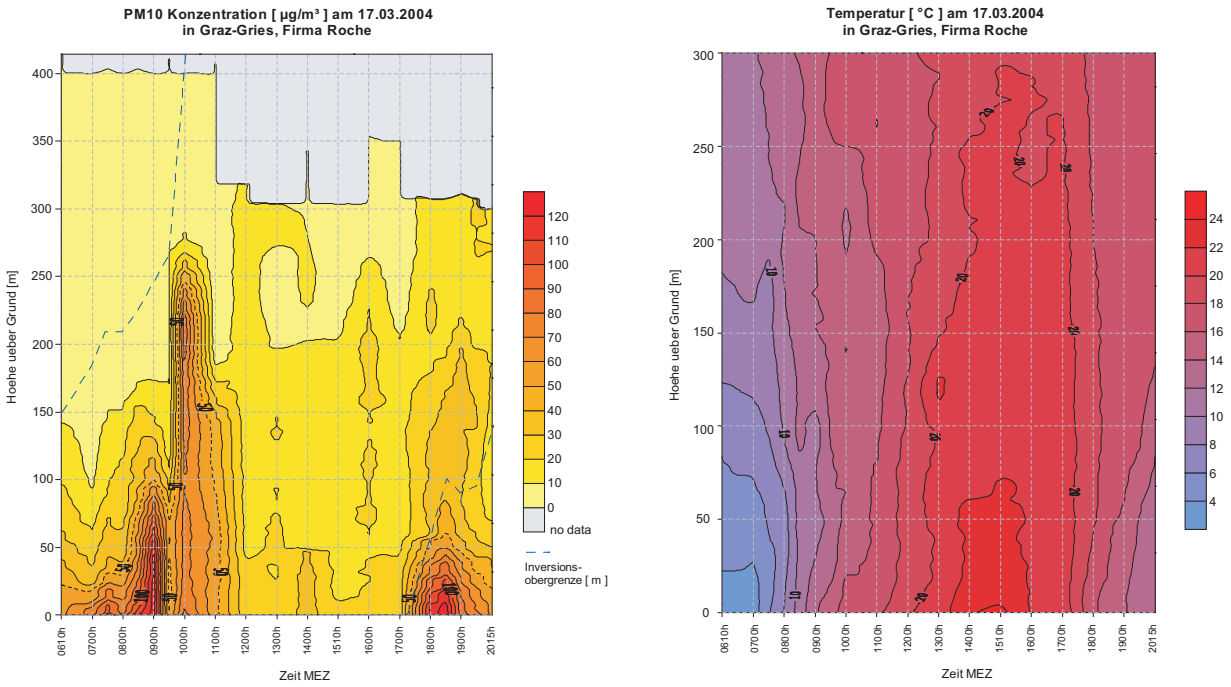

Figure 3: PM10 concentration (left) and temperature (right) for March 17, 2004. The vertical axis is altitude above ground (0–400m), the horizontal axis is time (6.10 am to 8.15 pm).

Kalkleiten have been included as a dependent variable. For the overall sample, the first three FPCs of the temperature difference curves describe about 95% of the variance. FPCA was used for covariate dimension reduction using $q = 3$, leading to the inclusion of a three-dimensional exogenous regressor (which is almost equivalent to the true regressor curve) in the second step of Algorithm 4. Results for the mean squared prediction errors are summarized under the label `CTD` (covariate temperature difference) in Table 5, performing the predictions in the same way as above. A significant improvement in the mean and median square prediction error can be observed.

7 Conclusions

This paper proposes a new prediction methodology for functional time series that appears to be widely and easily applicable. It is based on the idea that dimension reduction with functional principal components analysis should lead to a vector-valued time series of FPC scores that can be predicted with any existing multivariate methodology, parametric and nonparametric. The multivariate prediction is then transformed to a functional prediction using a truncated Karhunen-Loéve decomposition.

The proposed methodology seems to be advantageous for several reasons. Among them is its

intuitive appeal, made rigorous for the predominant FAR(1) case, but also its ease of application as existing software packages can be readily used, even by non-experts. It is in particular straightforward to extend the procedure to include exogenous covariates into the prediction algorithm. Simulations and an application to pollution data suggest that the proposed method leads to predictions that are always competitive with and often superior to the benchmark predictions in the field.

Future research could look into fine-tuning the proposed algorithms and developing automatic procedures to select the number of FPC scores p (and q if covariates are considered) as well as the number m used to run the innovations algorithm. Further theoretical developments would be welcome, but are beyond the scope of the present paper. It is hoped that the present article can spawn interest among researchers working in the active area of functional time series.

A Theoretical considerations

A.1 Proof of Theorem 3.1

Recall the notations introduced above equation (3.4). In order to prove the asymptotic equivalence between $\tilde{Y}_{n+1}$ in (2.2) and $\hat{Y}_{n+1}$ in (3.2) for the case of FAR(1) functional time series, observe first that

$$\left(\frac{1}{n-1}X^{e\prime}X^e\right)^{-1} = I_p \otimes \hat{\Gamma}_p^{-1},$$

where $\hat{\Gamma}_p$ is the $p\times p$ matrix with entries $\hat{\Gamma}_p(\ell,\ell') = \frac{1}{n-1}\sum_{k=1}^{n-1} y^e_{k,\ell}y^e_{k,\ell'}$ determined by the FPC scores $y^e_{k,\ell} = \langle Y_k, \hat{v}_\ell\rangle$, and $\otimes$ signifies the Kronecker product. With the help of (3.6), the VAR(1) based predictor (2.2) can be written in the form

$$\hat{Y}_{n+1} = \frac{1}{n-1}\left\{\left(\mathrm{mat}\left(\left[I_p\otimes\hat{\Gamma}_p^{-1}\right]X^{e\prime}\boldsymbol{Z}^e\right)\right)'\boldsymbol{Y}^e_n\right\}'\hat{\boldsymbol{v}},$$

with $\hat{\boldsymbol{v}} = (\hat{v}_1,\ldots,\hat{v}_p)'$ being the vector of the first p empirical eigenfunctions. On the other hand, defining the $p\times p$ matrix $\tilde{\Gamma}_p$ by the entries $\tilde{\Gamma}_p(\ell,\ell') = \frac{1}{n}\sum_{k=1}^{n} y^e_{k,\ell}y^e_{k,\ell'} = \mathrm{diag}(\hat{\lambda}_1,\ldots,\hat{\lambda}_p)$, direct verification shows that (3.2) takes the form

$$\tilde{Y}_{n+1} = \frac{1}{n-1}\left\{\left(\mathrm{mat}\left(\left[I_p\otimes\tilde{\Gamma}_p^{-1}\right]X^{e\prime}\boldsymbol{Z}^e\right)\right)'\boldsymbol{Y}^e_n\right\}'\hat{\boldsymbol{v}}.$$

The only formal difference between the two predictors under consideration is therefore in the matrices $\hat{\Gamma}_p$ and $\tilde{\Gamma}_p$. Now, for any $\ell,\ell' = 1,\ldots,p$,

$$\hat{\Gamma}_p(\ell,\ell') = \tilde{\Gamma}_p(\ell,\ell') + \frac{1}{n-1}\frac{1}{n}\sum_{k=1}^{n} y^e_{k,\ell}y^e_{k,\ell'} - \frac{1}{n-1}y^e_{n,\ell}y^e_{n,\ell'}$$

$$= \tilde{\Gamma}_p(\ell,\ell') + \frac{1}{n-1}\left(\hat{\lambda}_\ell I\{\ell=\ell'\} + y^e_{n,\ell}y^e_{n,\ell'}\right),$$

so that

$$\left|\hat{\Gamma}_p(\ell,\ell') - \tilde{\Gamma}_p(\ell,\ell')\right| \leq \frac{1}{n-1}\left(\frac{1}{n}\sum_{k=1}^{n}\|Y_k\|^2 + \|Y_n\|^2\right) = O_p\left(\frac{1}{n}\right).$$

In the following $\|\cdot\|$ will be used with a slight abuse of notation not only to indicate L^2 norm, but also Euclidean norm in $\mathbb{R}^p$ and matrix norm $\|A\| = \sup_{\|\mathbf{x}\|=1}\|A\mathbf{x}\|$, for a square matrix $A \in \mathbb{R}^{p\times p}$. Let

$$\Delta = \mathrm{mat}\left(\left[I_p \otimes \left(\hat{\Gamma}_p^{-1} - \tilde{\Gamma}_p^{-1}\right)\right]\frac{1}{n-1}X^{e\prime}\boldsymbol{Z}^e\right).$$

The orthogonality of the $\hat{v}_\ell$ together with Pythagoras' theorem and Bessel's inequality imply that

$$\|\hat{Y}_{n+1} - \tilde{Y}_{n+1}\|^2 = \left\|\Delta'\boldsymbol{Y}^e_n\right\|^2 \leq \|\Delta\|^2\|\boldsymbol{Y}^e_n\|^2 = \|\Delta\|^2\sum_{\ell=1}^{p}(y^e_{n,\ell})^2 \leq \|\Delta\|^2\|Y_n\|^2.$$

Define $S = \mathrm{mat}(\frac{1}{n-1}X^{e\prime}\boldsymbol{Z}^e)$ and notice that

$$\|\Delta\| = \left\|\left(\hat{\Gamma}_p^{-1} - \tilde{\Gamma}_p^{-1}\right)S\right\| \leq \left\|\hat{\Gamma}_p^{-1} - \tilde{\Gamma}_p^{-1}\right\|\|S\|.$$

Let $\boldsymbol{w} = (w_1,\ldots,w_p)'$. Since $S(\ell,\ell') = \frac{1}{n-1}\sum_{k=1}^{n-1} y^e_{k,\ell}y^e_{k+1,\ell'}$, iterative applications of the Cauchy-Schwarz inequality yield

$$\begin{aligned}
\|S\|^2 &= \sup_{\|\boldsymbol{w}\|=1}\sum_{\ell=1}^{p}\left(\sum_{\ell'=1}^{p}\frac{1}{n-1}\sum_{k=1}^{n-1}y^e_{k,\ell}y^e_{k+1,\ell'}w_{\ell'}\right)^2 \\
&\leq \sum_{\ell=1}^{p}\sum_{\ell'=1}^{p}\left(\frac{1}{n-1}\sum_{k=1}^{n-1}y^e_{k,\ell}y^e_{k+1,\ell'}\right)^2 \\
&\leq \sum_{\ell=1}^{p}\sum_{\ell'=1}^{p}\frac{1}{n-1}\sum_{k=1}^{n}(y^e_{k,\ell})^2\frac{1}{n-1}\sum_{k=1}^{n}(y^e_{k,\ell'})^2 \\
&\leq \left(\frac{1}{n-1}\sum_{k=1}^{n}\|Y_k\|^2\right)^2 \\
&= O_P(1).
\end{aligned}$$

It remains to estimate $\|\hat{\Gamma}_p^{-1} - \tilde{\Gamma}_p^{-1}\|$. The next step consists of using the fact that, for any $A, B \in \mathbb{R}^{p\times p}$, it holds that $(A+B)^{-1} = A^{-1} - A^{-1}(I+BA^{-1})^{-1}BA^{-1}$, provided all inverse matrices exist. Now choosing $A = \hat{\Gamma}_p$ and $B = \tilde{\Gamma}_p - \hat{\Gamma}_p$, it can be seen that

$$\begin{aligned}
\left\|\hat{\Gamma}_p^{-1} - \tilde{\Gamma}_p^{-1}\right\| &= \left\|\tilde{\Gamma}_p^{-1}\left[I_p + (\hat{\Gamma}_p - \tilde{\Gamma}_p)\tilde{\Gamma}_p^{-1}\right]^{-1}(\hat{\Gamma}_p - \hat{\Lambda}_p)\hat{\Lambda}_p^{-1}\right\| \\
&\leq \left\|\tilde{\Gamma}_p^{-1}\right\|^2\left\|\hat{\Gamma}_p - \tilde{\Gamma}_p\right\|\left\|\left[I_p + (\hat{\Gamma}_p - \tilde{\Gamma}_p)\tilde{\Gamma}_p^{-1}\right]^{-1}\right\|
\end{aligned}$$

$$\leq \sum_{\ell=0}^{\infty} \hat{\lambda}_p^{-\ell+2} \left\| \tilde{\Gamma}_p - \hat{\Gamma}_p \right\|^{\ell+1}$$
$$= O_P\left(\frac{1}{n}\right).$$

Putting together all results, the statement of Theorem 3.1 is established.

A.2 Analysis of theoretical model errors

The arguments in Section 3.2 show that the errors $\boldsymbol{\delta}_k$ of model (3.4) have a rather complicated form and it is generally infeasible to explicitly determine Ω^e. The main reason for this is that empirical eigenfunctions were used for the projections. On the other hand, Hörmann and Kokoszka [15] have noted that the empirical eigenfunctions $\hat{v}_\ell$ are $\sqrt{n}$-consistent for the population eigenfunctions v_ℓ (up to random signs) under mild conditions that are assumed to be satisfied in the following. This suggests that for a theoretical analysis one can work with the population eigenfunctions v_ℓ instead. Using these, (3.4) becomes

$$\boldsymbol{Y}_k = B_p \boldsymbol{Y}_{k-1} + \boldsymbol{\eta}_k, \tag{A.1}$$

where $\boldsymbol{\eta}_k = \boldsymbol{e}_k + \boldsymbol{u}_k$. Stacking the vectors in (A.1) one obtains in analogy to (3.5) the regression

$$\boldsymbol{Z} = X\boldsymbol{\beta}_p + \boldsymbol{E} + \boldsymbol{U}.$$

Defining $\Omega = \mathrm{Cov}(\boldsymbol{E} + \boldsymbol{U})$, the generalized least squares estimator for this problem becomes

$$\widehat{\boldsymbol{\beta}_p}(\mathrm{GLS}) = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\boldsymbol{Z}.$$

Note that the $\boldsymbol{\eta}_k$ are the theoretical counterparts of $\boldsymbol{\delta}_k$ and $\hat{\boldsymbol{\delta}}_k$. Their second-order structure can, however, be computed explicitly. First note that $E[\boldsymbol{\eta}_k] = \boldsymbol{0}$. Defining $W_h = \mathrm{Cov}(\boldsymbol{\eta}_{k+h}, \boldsymbol{\eta}_k)$, it holds that

$$\Omega = \begin{bmatrix} W_0 & W_1 & W_2 & \cdots \\ W_{-1} & W_0 & W_1 & \cdots \\ \vdots & \vdots & & \ddots \end{bmatrix}.$$

Stationarity implies that

$$W_h = E[\boldsymbol{e}_{1+h}\boldsymbol{e}_1'] + E[\boldsymbol{u}_{1+h}\boldsymbol{e}_1'] + E[\boldsymbol{e}_{1+h}\boldsymbol{u}_1'] + E[\boldsymbol{u}_{1+h}\boldsymbol{u}_1']$$
$$= \Sigma_h + D_{-h}^T + D_h + C_h,$$

where $\Sigma_h = E[\boldsymbol{e}_{1+h}\boldsymbol{e}_1']$, $D_h = E[\boldsymbol{e}_{1+h}\boldsymbol{u}_1']$ and $C_h = E[\boldsymbol{u}_{1+h}\boldsymbol{u}_1']$. By assumption, $\Sigma_h = O$, the $p \times p$ zero matrix, if $h \neq 0$. Also, $D_h = O$ for $h \geq 0$. Hence

$$\Omega = \begin{bmatrix} C_0 & C_1 & C_2 & \cdots \\ C_1' & C_0 & C_1 & \cdots \\ C_2' & C_1' & C_0 & \cdots \\ \vdots & \vdots & & \ddots \end{bmatrix} + \begin{bmatrix} \Sigma_0 & D_{-1}' & D_{-2}' & \cdots \\ D_{-1} & \Sigma_0 & D_{-1}' & \cdots \\ D_{-2} & D_{-1} & \Sigma_0 & \cdots \\ \vdots & \vdots & & \ddots \end{bmatrix}.$$

Using the definitions of $\boldsymbol{e}_k$ and $\boldsymbol{u}_k$, the elements of C_h and D_{-h} are for $\ell, \ell' = 1, \ldots, p$ computed as

$$C_h(\ell, \ell') = \sum_{\ell_1=p+1}^{\infty} \sum_{\ell_2=p+1}^{\infty} E\left[\langle Y_{1+h}, v_{\ell_1}\rangle \langle Y_1, v_{\ell_2}\rangle\right] \langle \Psi(v_{\ell_1}), v_\ell\rangle \langle \Psi(v_{\ell_2}), v_{\ell'}\rangle, \tag{A.2}$$

$$D_{-h}(\ell, \ell') = \sum_{\ell_1=p+1}^{\infty} E\left[\langle Y_0, v_{\ell_1}\rangle \langle \varepsilon_{1-h}, v_\ell\rangle\right] \langle \Psi(v_{\ell_1}), v_{\ell'}\rangle. \tag{A.3}$$

The following lemma shows that the matrices D_{-h} and C_h decrease rapidly to O when $h \to \infty$. This indicates that in general relatively small values of b for the banded covariance estimator $\hat{\Omega}_b^e$ may be chosen.

Lemma A.1. *Let Ψ^* denote the adjoint operator of Ψ. If $\varepsilon_0 \in L_H^2$, then there is a constant c which depends only on the distribution of ε_0 and Ψ, such that*

$$|C_h(\ell, \ell')| \leq c \, \|\Psi\|_{\mathcal{L}}^h \|\Psi^*(v_\ell)\| \|\Psi^*(v_{\ell'})\|,$$
$$|D_{-h}(\ell, \ell')| \leq c \, \|\Psi\|_{\mathcal{L}}^h \|\Psi^*(v_{\ell'})\|,$$

for any $h \geq 1$.

Proof. Using Example 2.1 in Hörmann and Kokoszka [15], one can define for any $h > 0$ a sequence $(Y_k^{(h)} \colon k \in \mathbb{Z})$ having the same marginal distribution as the FAR(1) process $(Y_k \colon k \in \mathbb{Z})$ and satisfying that $(Y_{k+m}^{(h)} \colon m \geq h)$ and $(Y_{k-\ell} \colon \ell \leq 0)$ are independent for any $k \in \mathbb{Z}$ and

$$\nu_2(Y_k - Y_k^{(h)}) = \left(E\big[\|Y_k - Y_k^{(h)}\|^2\big]\right)^{1/2} \leq c_1 \, \nu_2(\varepsilon_0) \|\Psi\|_{\mathcal{L}}^h.$$

This and the Cauchy-Schwarz inequality imply that

$$\begin{aligned} \big|E\big[\langle Y_{1+h}, v_{\ell_1}\rangle \langle Y_1, v_{\ell_2}\rangle\big]\big| &= \Big|E\big[\langle Y_{1+h} - Y_{1+h}^{(h)}, v_{\ell_1}\rangle \langle Y_1, v_{\ell_2}\rangle\big]\Big| \\ &\leq \nu_2\big(Y_k - Y_k^{(h)}\big)\nu_2(Y_1) \\ &\leq c_2 \, \nu_2(\varepsilon_0)\|\Psi\|_{\mathcal{L}}^h. \end{aligned}$$

Notice that $\nu_2(Y_1)$ depends only on the distribution of ε_0 and Ψ. Hence subsequently using (A.2), the Cauchy-Schwarz inequality and Parseval's identity, leads to

$$\begin{aligned}
|C_h(\ell,\ell')| &\leq c\,\|\Psi\|_{\mathcal{L}}^h \sum_{\ell_1=p+1}^{\infty}\sum_{\ell_2=p+1}^{\infty} |\langle\Psi(v_{\ell_1}),v_\ell\rangle\langle\Psi(v_{\ell_2}),v_{\ell'}\rangle| \\
&\leq c\,\|\Psi\|_{\mathcal{L}}^h \Big(\sum_{\ell_1=p+1}^{\infty}\langle\Psi(v_{\ell_1}),v_\ell\rangle^2\Big)^{1/2}\Big(\sum_{\ell_2=p+1}^{\infty}\langle\Psi(v_{\ell_2}),v_{\ell'}\rangle^2\Big)^{1/2} \\
&\leq c\,\|\Psi\|_{\mathcal{L}}^h \Big(\sum_{\ell_1=1}^{\infty}\langle v_{\ell_1},\Psi^*(v_\ell)\rangle^2\Big)^{1/2}\Big(\sum_{\ell_2=1}^{\infty}\langle v_{\ell_2},\Psi^*(v_{\ell'})\rangle^2\Big)^{1/2} \\
&= c\,\|\Psi\|_{\mathcal{L}}^h \|\Psi^*(v_\ell)\|\|\Psi^*(v_{\ell'})\|.
\end{aligned}$$

The second statement can be proven in a similar way and the proof is complete. □

References

[1] Aguilera, A.M., Ocaña, F.A. and Valderrama, M.J. (1999). Forecasting Time Series By Functional PCA. Discussion Of Several Weighted Approaches. *Computational Statistics* **14**, 443–467.

[2] Aneiros-Pérez, G., Cao, R. and Vilar-Fernánez, J.M. (2010). Functional methods for time series prediction: a nonparametric approach. *Journal of Forecasting* **30**, 377–392.

[3] Aneiros-Pérez, G. and Vieu, P. (2008). Nonparametric time series prediction: A semi-functional partial linear modeling. *Journal of Multivariate Analysis* **99**, 834–857.

[4] Antoniadis, A., Paparoditis, E. and Sapatinas, T. (2006). A functional wavelet-kernel approach for time series prediction. *Journal of the Royal Statistical Society, Series B* **68**, 837–857.

[5] Antoniadis, A., Paparoditis, E. and Sapatinas, T. (2009). Bandwidth selection for functional time series prediction. *Statistics & Probability Letters* **79**, 733–740.

[6] Antoniadis, A. and Sapatinas, T. (2003). Wavelet methods for continuous time prediction using Hilbert-valued autoregressive processes. *Journal of Multivariate Analysis* **87**, 133–158.

[7] Besse P. and Cardot, H. (1996). Approximation spline de la prévision dun processus fonctionnel autorégressif dordre 1. *Canadian Journal of Statistics* **24**, 467–487.

[8] Besse P., Cardot H. and Stephenson, D. (2000). Autoregressive forecasting of some functional climatic variations. *Scandinavian Journal of Statistics* **27**, 673–687.

[9] Bosq, D. (2000). *Linear Processes in Function Spaces.* Springer-Verlag, New York.

[10] Brockwell, P.J. and Davis, R.A. (1991). *Time Series Analysis: Theory and Methods (2nd ed.).* Springer-Verlag, New York.

[11] Damon J. and Guillas, S. (2002). The inclusion of exogenous variables in functional autoregressive ozone forecasting. *Environmetrics* **13**, 759–774.

[12] Damon, J. and Guillas, S. (2010). The `far` package for `R`. `http://cran.r-project.org/web/packages/far/far.pdf`

[13] Didericksen, D., Kokoszka, P. and Zhang, X. (2012). *Computational Statistics* **27**, 285–298.

[14] Ferraty, F. and Vieu, P. (2010). *Nonparametric Functional Data Analysis.* Springer-Verlag, New York.

[15] Hörmann, S. and Kokoszka, P. (2010). Weakly dependent functional data. *The Annals of Statistics* **38**, 1845–1884.

[16] Hyndman, R.J. and Ullah, M.S. (2007). Robust forecasting of mortality and fertility rates: A functional data approach. *Computational Statistics and Data Analysis* **51**, 4942–4956.

[17] Hyndman, R.J. and Shang, H.L. (2009). Forecasting functional time series (with discussion) *Journal of the Korean Statistical Society* **38**, 199–221.

[18] Hyndman, R.J. and Shang, H.L. (2012). The `ftsa` package for `R`. `http://cran.r-project.org/web/packages/ftsa/ftsa.pdf`

[19] Hörmann, S. and Kokoszka, P. (2012). Functional time series. In: *Handbook of Statistics: Time Series Analysis–Methods and Applications*, in press.

[20] Kargin V. and Onatski, A. (2008). Curve forecasting by functional autoregression. *Journal of Multivariate Analysis* **99**, 2508–2526.

[21] Podesser, A. Wölfelmaier, F. and Rieder, H. (2004). Kombinierte Feinstaub–Meteorologiesondierungen am 17.03.2004 in Graz. `http://app.luis.steiermark.at/berichte/Download/Fachberichte/Feinstaub_Ballonsondierungen.pdf`

[22] Lütkepohl, H. (2006). *New Introduction to Multiple Time Series Analysis*. Springer-Verlag, New York.

[23] Ramsay, J.O. and Silverman, B.W. (2005). *Functional Data Analysis (2nd ed.)*. Springer-Verlag, New York.

[24] Shumway, R.H. and Stoffer, D.S. (2011). *Time Series and Its Applications (3rd ed.)*. Springer-Verlag, New York.

[25] Stadlober, E. Hörmann, S. and Pfeiler, (2008). Quality and performance of a PM10 daily forecasting model. *Atmospheric Environment* **42**, 1098–1109.

n	method	$p=2$			$p=3$		
		med	mean	sd	med	mean	sd
	FAR	1.368	1.707	2.074	1.347	1.615	1.258
20	VAR.ls	1.016	1.031	1.060	1.037	1.050	1.114
	VAR.gls	1.018	1.068	1.724	1.044	1.954	68.399
	Naive	1.098	1.099	1.042	1.117	1.122	1.100
	Mean	1.373	1.328	1.129	1.404	1.376	1.200
	Inno, $m=1$	0.995	0.997	0.999	0.997	0.997	0.999
	Inno, $m=2$	1.020	1.015	1.003	1.029	1.032	1.023
	Inno, $m=3$	1.039	1.028	1.001	1.054	1.060	1.048
	FAR	1.323	1.638	1.611	1.257	1.513	1.149
40	VAR.ls	1.006	1.009	1.013	1.012	1.013	1.013
	VAR.gls	1.009	1.008	1.009	1.014	1.017	1.012
	Naive	1.155	1.135	1.041	1.190	1.185	1.126
	Mean	1.426	1.377	1.191	1.495	1.445	1.220
	Inno, $m=1$	0.999	0.999	0.999	0.995	0.999	1.000
	Inno, $m=2$	1.012	1.007	0.997	1.023	1.017	1.014
	Inno, $m=3$	1.019	1.016	1.000	1.039	1.037	1.023
	FAR	1.282	1.534	1.118	1.217	1.458	1.048
80	VAR.ls	1.003	1.002	0.999	1.002	1.003	1.004
	VAR.gls	1.004	1.002	0.999	1.005	1.005	1.005
	Naive	1.156	1.164	1.164	1.245	1.229	1.191
	Mean	1.438	1.422	1.413	1.526	1.481	1.383
	Inno, $m=1$	1.000	1.000	1.000	1.000	1.000	1.001
	Inno, $m=2$	0.999	1.002	1.007	1.009	1.009	1.007
	Inno, $m=3$	1.008	1.008	1.014	1.024	1.022	1.014
	FAR	1.280	1.560	1.262	1.187	1.437	1.086
160	VAR.ls	1.000	1.000	0.995	1.000	1.001	1.001
	VAR.gls	0.999	1.000	0.995	1.008	1.003	1.003
	Naive	1.201	1.176	1.128	1.279	1.262	1.193
	Mean	1.467	1.414	1.228	1.575	1.500	1.270
	Inno, $m=1$	1.000	1.000	1.000	1.001	1.000	1.000
	Inno, $m=2$	1.004	1.000	1.002	1.006	1.000	0.992
	Inno, $m=3$	1.005	1.003	1.002	1.011	1.005	0.994
	FAR	1.268	1.533	1.088	1.149	1.444	2.250
320	VAR.ls	1.000	1.000	1.001	0.999	1.000	1.000
	VAR.gls	1.000	1.000	1.000	1.004	1.001	0.999
	Naive	1.200	1.186	1.196	1.296	1.265	1.054
	Mean	1.490	1.421	1.362	1.612	1.529	1.069
	Inno, $m=1$	1.000	1.000	1.000	1.000	1.000	1.000
	Inno, $m=2$	0.999	0.999	1.003	1.006	1.000	0.998
	Inno, $m=3$	0.999	1.000	1.005	1.008	1.003	0.999

Table 2: $\text{MED}_{\texttt{FAR}}$, $\text{MSE}_{\texttt{FAR}}$ and $\text{SD}_{\texttt{FAR}}$ obtained from 5,000 repetitions are presented when the data generating process follows an FAR(1) model. For the competing methods `Pr` equal to `VAR.ls`, `VAR.gls`, `Naive`, `Mean` and `Inno` relative values $\text{RMED}_{\texttt{Pr}}$, $\text{RMSE}_{\texttt{Pr}}$ and $\text{RSD}_{\texttt{Pr}}$ are presented.

n	method	$p=2$			$p=3$		
		med	mean	sd	med	mean	sd
	FAR	1.347	1.632	1.245	1.348	1.606	1.122
20	VAR.ls	1.028	1.037	1.061	1.042	1.070	1.136
	VAR.gls	1.066	1.468	27.318	1.085	3.458	177.030
	Naive	1.103	1.093	1.073	1.094	1.087	1.061
	Mean	1.203	1.194	1.382	1.173	1.160	1.098
	Inno, $m=1$	1.001	0.998	1.001	0.993	0.997	0.996
	Inno, $m=2$	0.989	0.977	0.987	0.982	0.987	1.006
	Inno, $m=3$	0.993	0.990	0.993	1.010	1.017	1.024
	FAR	1.324	1.613	1.732	1.291	1.586	1.785
40	VAR.ls	1.009	1.009	1.002	1.020	1.020	1.016
	VAR.gls	1.037	1.261	10.989	1.067	346.569	21665.501
	Naive	1.126	1.111	1.065	1.135	1.124	1.043
	Mean	1.207	1.192	0.997	1.243	1.212	1.112
	Inno, $m=1$	1.000	0.999	1.001	0.999	0.999	1.000
	Inno, $m=2$	0.951	0.953	0.964	0.947	0.945	0.994
	Inno, $m=3$	0.958	0.961	0.968	0.955	0.963	1.002
	FAR	1.304	1.561	1.199	1.280	1.518	1.143
80	VAR.ls	1.001	1.005	1.009	1.007	1.006	1.012
	VAR.gls	1.039	1.071	2.694	1.045	1.129	7.914
	Naive	1.118	1.121	1.091	1.165	1.161	1.126
	Mean	1.218	1.221	1.143	1.253	1.247	1.310
	Inno, $m=1$	0.998	1.000	1.000	0.999	1.000	1.000
	Inno, $m=2$	0.938	0.938	0.949	0.901	0.913	0.955
	Inno, $m=3$	0.941	0.942	0.953	0.912	0.923	0.962
	FAR	1.303	1.569	1.168	1.274	1.556	1.607
160	VAR.ls	1.000	1.001	1.000	1.001	1.001	1.000
	VAR.gls	1.035	1.044	1.557	1.032	1.037	1.038
	Naive	1.128	1.125	1.097	1.144	1.150	1.042
	Mean	1.241	1.226	1.200	1.249	1.239	1.070
	Inno, $m=1$	1.000	1.000	1.000	1.000	1.000	1.000
	Inno, $m=2$	0.929	0.928	0.956	0.884	0.908	0.971
	Inno, $m=3$	0.930	0.931	0.957	0.894	0.914	0.970
	FAR	1.300	1.543	1.191	1.246	1.508	1.291
320	VAR.ls	1.000	1.002	1.011	1.000	1.001	1.001
	VAR.gls	1.037	1.035	1.043	1.033	1.036	1.009
	Naive	1.132	1.125	1.063	1.189	1.172	1.095
	Mean	1.233	1.228	1.299	1.261	1.260	1.177
	Inno, $m=1$	0.999	1.000	1.000	1.000	1.000	1.000
	Inno, $m=2$	0.918	0.923	0.961	0.870	0.887	0.921
	Inno, $m=3$	0.918	0.924	0.963	0.873	0.890	0.924

Table 3: $\text{MED}_{\texttt{FAR}}$, $\text{MSE}_{\texttt{FAR}}$ and $\text{SD}_{\texttt{FAR}}$ obtained from 5,000 repetitions are presented when the data generating process follows an FAR(2) model. For the competing methods `Pr` equal to `VAR.ls`, `VAR.gls`, `Naive`, `Mean` and `Inno` relative values $\text{RMED}_{\texttt{Pr}}$, $\text{RMSE}_{\texttt{Pr}}$ and $\text{RSD}_{\texttt{Pr}}$ are presented.

n	method	$p=2$			$p=3$		
		med	mean	sd	med	mean	sd
	FAR	1.759	2.102	1.811	1.753	2.049	1.430
20	VAR.ls	1.017	1.028	1.609	1.039	1.048	1.063
	VAR.gls	1.000	1.017	1.746	1.033	1.383	14.873
	Naive	1.035	1.054	1.113	1.035	1.056	1.088
	Mean	1.036	1.038	0.853	1.023	1.050	1.133
	Inno, $m=2$	0.982	0.989	0.985	0.988	0.996	1.558
	Inno, $m=3$	0.979	0.990	0.985	0.987	0.994	1.009
	Inno, $m=4$	0.976	0.989	0.999	0.991	0.994	1.006
	FAR	1.710	2.050	1.461	1.714	2.027	1.479
40	VAR.ls	1.008	1.011	0.994	1.014	1.011	0.981
	VAR.gls	1.003	1.000	0.981	1.005	1.004	0.989
	Naive	1.045	1.054	1.101	1.084	1.092	1.142
	Mean	1.065	1.079	1.114	1.061	1.059	1.024
	Inno, $m=2$	0.990	0.978	0.964	0.990	0.983	0.972
	Inno, $m=3$	0.983	0.974	0.962	0.983	0.983	0.979
	Inno, $m=4$	0.977	0.972	0.970	0.983	0.985	0.988
	FAR	1.721	2.042	1.509	1.666	1.999	1.694
80	VAR.ls	1.003	1.005	1.040	1.002	1.016	1.754
	VAR.gls	0.999	1.000	1.020	0.998	1.008	1.656
	Naive	1.050	1.064	1.121	1.077	1.082	0.946
	Mean	1.055	1.072	1.160	1.082	1.119	2.161
	Inno, $m=2$	0.975	0.979	0.986	0.974	0.972	0.856
	Inno, $m=3$	0.971	0.972	0.981	0.965	0.966	0.828
	Inno, $m=4$	0.962	0.968	0.976	0.961	0.961	0.817
	FAR	1.754	2.058	1.408	1.716	2.006	1.515
160	VAR.ls	1.001	1.001	0.999	1.001	1.002	1.000
	VAR.gls	0.996	0.999	1.002	0.999	1.000	0.996
	Naive	1.048	1.063	1.120	1.083	1.100	1.108
	Mean	1.056	1.074	1.110	1.066	1.098	1.142
	Inno, $m=2$	0.979	0.980	0.983	0.962	0.966	0.971
	Inno, $m=3$	0.968	0.970	0.978	0.947	0.954	0.968
	Inno, $m=4$	0.961	0.966	0.976	0.933	0.948	0.967
	FAR	1.724	2.082	1.773	1.650	1.979	1.414
320	VAR.ls	1.000	1.001	1.005	1.001	1.009	1.407
	VAR.gls	1.003	1.001	1.006	1.001	1.007	1.265
	Naive	1.068	1.073	1.110	1.092	1.099	1.112
	Mean	1.057	1.067	1.102	1.099	1.117	1.933
	Inno, $m=2$	0.975	0.975	0.979	0.970	0.965	0.927
	Inno, $m=3$	0.965	0.964	0.970	0.952	0.950	0.912
	Inno, $m=4$	0.961	0.960	0.966	0.945	0.940	0.908

Table 4: $\text{MED}_{\texttt{FAR}}$, $\text{MSE}_{\texttt{FAR}}$ and $\text{SD}_{\texttt{FAR}}$ obtained from 5,000 repetitions are presented when the data generating process follows an FMA(1) model. For the competing methods `Pr` equal to `VAR.ls`, `VAR.gls`, `Naive`, `Mean` and `Inno` relative values $\text{RMED}_{\texttt{Pr}}$, $\text{RMSE}_{\texttt{Pr}}$ and $\text{RSD}_{\texttt{Pr}}$ are presented.

n	method	$p=2$			$p=3$		
		med	mean	sd	med	mean	sd
	FAR	1.58	2.43	2.46	1.53	2.36	2.36
20	VAR.ls	1.64	2.45	2.43	1.58	2.41	2.34
	Mean	1.69	2.88	3.28	1.69	2.88	3.28
	Naive	1.81	2.57	2.67	1.81	2.57	2.67
	Inno, $m=2$	1.58	2.38	2.49	1.71	2.37	2.29
	Inno, $m=3$	1.55	2.45	2.50	1.78	2.47	2.32
	CTD, $m=2$, $q=3$	1.41	2.37	2.70	1.65	2.57	2.88
	FAR	1.51	2.37	2.42	1.40	2.29	2.38
40	VAR.ls	1.51	2.37	2.40	1.39	2.31	2.40
	Mean	1.69	2.87	3.28	1.69	2.87	3.28
	Naive	1.73	2.58	2.78	1.73	2.58	2.78
	Inno, $m=2$	1.38	2.37	2.52	1.43	2.29	2.39
	Inno, $m=3$	1.51	2.38	2.56	1.53	2.27	2.38
	CTD, $m=2$, $q=3$	1.40	2.22	2.32	1.34	2.16	2.25
	FAR	1.61	2.24	2.22	1.36	2.03	2.17
60	VAR.ls	1.63	2.23	2.20	1.32	2.03	2.17
	Mean	1.79	3.09	3.48	1.79	3.09	3.48
	Naive	1.69	2.63	2.96	1.69	2.63	2.96
	Inno, $m=2$	1.44	2.09	2.09	1.17	1.98	2.06
	Inno, $m=3$	1.41	2.10	2.09	1.42	2.04	2.28
	CTD, $m=2$, $q=3$	1.16	1.94	1.96	1.06	1.95	2.07
	FAR	1.59	2.20	2.20	1.30	1.95	2.17
80	VAR.ls	1.51	2.19	2.21	1.25	1.95	2.22
	Mean	1.82	3.32	3.70	1.82	3.32	3.70
	Naive	1.91	2.84	3.13	1.91	2.84	3.13
	Inno, $m=2$	1.31	2.01	2.07	1.25	1.91	2.02
	Inno, $m=3$	1.47	1.98	2.02	1.22	1.89	2.09
	CTD, $m=2$, $q=3$	1.26	1.80	1.73	1.21	1.82	1.80

Table 5: $\text{MED}_{\texttt{Pr}}$, $\text{MSE}_{\texttt{Pr}}$ and $\text{SD}_{\texttt{Pr}}$ are shown for prediction methods `Pr` equal to `FAR`, `VAR.ls`, `VAR.gls`, `Mean`, `Naive`, `Inno` and `CDT`.