MULTI-VARIABLE PREDICTION OF PHYSICAL DATA

Dan ŞTEFĂNOIU[1], Janetta CULITĂ[2]

Lucrarea prezintă un studiu comparativ între diferite abordari adoptate pentru predicția datelor fizice multi-variabile. Datele provin de la fenomene naturale distribuite geografic (în special ecologice şi meteorologice) şi sunt modelate ca o colecție de serii de timp. Simularile demonstrează ca acuratetea predicției seriilor de timp distribuite creşte odata cu gradul lor de corelare. Sunt studiate trei tipuri de modele multi-variabile ale sistemului furnizor de date: MIMO-ARMA, model cu reprezentare pe stare (de tip Kalman-Bucy) şi MIMO-ARMAX. Strategiile de modelare şi predicție propuse au fost implementate pe un studiu de caz preluat din meteorologie.

The paper aims to present a comparative study related to different approaches regarding prediction of multi-variable physical data. Data are provided by natural phenomena with geographical distribution (especially ecological and meteorological) and are stored as blocks of time series. Simulations have shown that the prediction accuracy of data increased with their correlation. Three modeling approaches of data are considered: MIMO-ARMA type, a state representation of Kalman-Bucy type and MIMO-ARMAX type. The performance of modeling and prediction algorithms is demonstrated on a meteorological case study.

Keywords: distributed time series, prediction, Kalman-Bucy filter, ARMAX.

1. Introduction

The paper focuses on the problem of multi-variable physical data prediction. The physical data could be acquired from geographical distributed natural phenomena (such as ecological or meteorological) by means of a sensors network that provide a collection of time series (ts). Such data coming from different channels are in general correlated each other.

Take for example the monitoring of minimum and maximum temperatures in two cities at about 60 km far from each other, like in Fig.1. The problem is to build and estimate multi-variable identification models, in view of physical data prediction like those temperatures. Instead of using data fusion techniques [1], which lead to a unique time series, in this paper, three different approaches are introduced, in order to increase the prediction accuracy. First, the time series are

[1] Professor, Faculty of Automatic Control and Computer Science, POLITEHNICA University of Bucharest, Romania, danny@indinf.pub.ro

[2] Reader, Faculty of Automatic Control and Computer Science, POLITEHNICA University of Bucharest, Romania, jculita@yahoo.com

independently processed through MIMO-ARMA models [2]. Here, no correlations between data are accounted. Second, the models are upgraded to MIMO-ARMAX type. Third, the distributed time series is seen as a collection of measurable states of a quasi-ubiquitous open system and can be predicted via Kalman filtering [3,4]. For the last two approaches, correlation between data is crucial. The paper is structured as follows: section 2-3 are designed to describe the construction of the three predictors; the prediction performance of models is tested within the application described in section 4; section 5 concludes the paper.

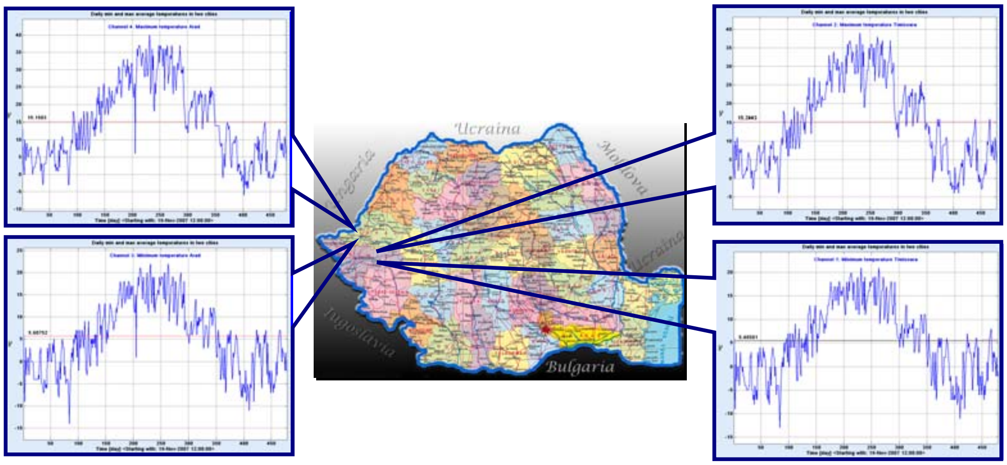

Fig. 1. Minimum (bottom) and maximum (top) temperatures of two cities.

2. ARMA and ARMAX Prediction Models

Consider $\mathcal{Y} = \left\{ y_j \right\}_{j\in\overline{1,ny}}$ the set of ny ts and denote the maximum number of acquired data by N_y. Then the vector $\mathbf{y} = \left[y_1 \cdots y_{ny} \right]^T$ could be viewed as the output of a MIMO system. Assume that each ts y_j ($j \in \overline{1,ny}$) of length $N_y \in \mathbb{N}^*$ is a colored noise produced by the corresponding white noise (wn) e_j ($j \in \overline{1,ny}$). Any such ts can be roughly modeled by a (SISO)-ARMA filter, like below:

$$\mathrm{A}_j\left(\mathrm{q}^{-1}\right) y_j \equiv \mathrm{C}_j\left(\mathrm{q}^{-1}\right) e_j, \quad \forall\, j \in \overline{1,ny}, \tag{1}$$

where A_j and C_j are polynomials and q^{-1} is the delay operator (do). Thus, (1) is actually a collection of difference equations. The white noises corrupting the data are supposed to be Gaussian and uncorrelated. In case of distributed ts, the model (1) can be extended to the rough MIMO-ARMA model:

$$\mathbf{A}\left(\mathrm{q}^{-1}\right)\mathbf{y} \equiv \mathbf{C}\left(\mathrm{q}^{-1}\right)\mathbf{e}, \tag{2}$$

by simply considering that the output channels are not correlated each other. In equation (2), $\mathbf{A} \in \mathbb{R}^{ny \times ny}\left(\mathrm{q}^{-1}\right)$ and $\mathbf{C} \in \mathbb{R}^{ny \times ny}\left(\mathrm{q}^{-1}\right)$ are diagonal polynomial matrices. Identification of model (2) is simply performed via *Minimum Prediction Error Method* (MPEM) [2,4].

After identification, the estimated polynomials $\hat{\mathrm{A}}_j$ and $\hat{\mathrm{C}}_j$ (for $j \in \overline{1,ny}$) yield the evaluation of prediction error:

$$\hat{e}_j \equiv \hat{\mathrm{A}}_j\left(\mathrm{q}^{-1}\right) y_j + \left[1 - \hat{\mathrm{C}}_j\left(\mathrm{q}^{-1}\right)\right] \hat{e}_j , \quad \forall\, j \in \overline{1,ny} . \tag{3}$$

In order to increase the precision accuracy and to incorporate the correlation between various ts, the model (2) is upgraded to MIMO-ARMAX model:

$$\mathbf{A}\left(\mathrm{q}^{-1}\right)\mathbf{y} \equiv \mathbf{B}\left(\mathrm{q}^{-1}\right)\mathbf{u} + \mathbf{C}\left(\mathrm{q}^{-1}\right)\mathbf{e} , \tag{4}$$

where $\mathbf{B} \in \mathbb{R}^{ny \times nu}\left(\mathrm{q}^{-1}\right)$ is a polynomial matrix as well and $\mathbf{u} \in \mathbb{R}^{nu}$ is the input vector (on nu *channels*, i.e. data sources) of overall stochastic process. Each input is estimated like in (3) (i.e. $\hat{u}_j \equiv \hat{e}_j$). Thus, the refined model is stimulated with the estimations of white noises corrupting the data. The model (4) is rather difficult to identify, because of great number of parameters. Therefore, two simplifying hypothesis are considered: the matrix $\mathbf{A}$ is diagonal (each output is independent on the other outputs) and the matrix $\mathbf{C}$ is diagonal (since the white noise signals are uncorrelated each other). Then the MIMO-ARMAX model can be split into ny MISO-ARMAX models:

$$\mathrm{A}_j\left(\mathrm{q}^{-1}\right) y_j \equiv \mathbf{B}_j\left(\mathrm{q}^{-1}\right)\mathbf{u} + \mathrm{C}_j\left(\mathrm{q}^{-1}\right) e_j , \quad \forall\, j \in \overline{1,ny} , \tag{5}$$

where $\mathbf{B}_j \in \mathbb{R}^{1 \times nu}\left(\mathrm{q}^{-1}\right)$ is a row vector of polynomials. Obviously, the identification of MIMO-ARMAX is solved now by identifying each MISO-ARMAX model as a result of channel isolation. The prediction can be performed when solving all equations (5) at each step. Thus, correlations between outputs are indirectly encoded by the collection of MISO-ARMAX models, through equation (3). Actually, noises of $nu = ny - 1$ output channels are affecting the data of some input channel, which means that matrix $\mathbf{B}$ (of (5)) has null diagonal.

Identification of parameters in model (5) relies on MPEM as well, whereas the estimated values of white noises are more accurately computed with approximating ARX models:

$$\hat{e}_j \equiv \hat{\mathrm{A}}_j^{n\alpha}\left(\mathrm{q}^{-1}\right) y_j - \sum_{i=1, i \neq j}^{ny} \hat{\mathrm{B}}_{j,i}^{n\beta}\left(\mathrm{q}^{-1}\right) \hat{u}_i , \quad \forall\, j \in \overline{1,ny} , \tag{6}$$

where $\mathrm{A}_j^{n\alpha}$ and $\mathrm{B}_{j,i}^{n\beta}$ are polynomials with degrees $n\alpha$ and $n\beta$, respectively. After estimating all the parameters, the optimal predictor is recursively computed, on a prediction horizon of length $K \geq 1$, as follows (for any $k \in \overline{N_y + 1, N_y + K}$):

$$\begin{aligned}\hat{y}_j\left[k \mid N_y\right] = -\hat{a}_{j,1}\hat{y}_j\left[k-1 \mid N_y\right] - \cdots - \hat{a}_{j,na_j}\hat{y}_j\left[k-na_j \mid N_y\right] + \\ + \sum_{i=1, i\neq j}^{ny}\left[\hat{b}_{j,i,1}\hat{u}_i[k-1] + \cdots + \hat{b}_{j,i,nb_{j,i}}\hat{u}_i\left[k-nb_{j,i}\right]\right] + \\ + \hat{c}_{j,1}\hat{e}_j[k-1] + \cdots + \hat{c}_{j,nc_j}\hat{e}_j\left[k-nc_j\right];\end{aligned} \tag{7}$$

$$\begin{aligned}\hat{e}_j[k] = \hat{y}_j\left[k \mid N_y\right] + \hat{\alpha}_{j,1}\hat{y}_j\left[k-1 \mid N_y\right] + \cdots + \hat{\alpha}_{j,k-N_y-1}\hat{y}_j\left[N_y+1 \mid N_y\right] + \\ + \hat{\alpha}_{j,k-N_y}y_j\left[N_y\right] + \cdots + \hat{\alpha}_{j,n\alpha}y_j[k-n\alpha] - \\ - \sum_{i=1, i\neq j}^{ny}\left[\hat{\beta}_{j,i,1}\hat{u}_i[k-1] + \cdots + \hat{\beta}_{j,i,n\beta}\hat{u}_i[k-n\beta]\right] = \hat{u}_j[k].\end{aligned} \tag{8}$$

The prediction performance on channel j is assessed by means of the *prediction quality* (PQ) cost function below:

$$\mathrm{PQ}_j \overset{def}{=} 100 \Bigg/ \left(1 + \frac{\sqrt{\sum_{k=1}^{K}\hat{\sigma}_{j,k}^2}}{\hat{\lambda}_{e_j}\sqrt{\mathrm{SNR}_j}\sqrt{\mathrm{SNR}_j^K}}\right) [\%] \quad \forall j \in \overline{1,ny}, \quad \forall j \in \overline{1,ny} \tag{9}$$

where: $\left\{\hat{\sigma}_{j,k}^2\right\}_{j\in\overline{1,ny}}$ are the prediction error variances on channel j; $\mathrm{SNR}_j \overset{def}{=} \sigma_{y_j}^2 / \hat{\sigma}_{j,1}^2$ & $\mathrm{SNR}_j^K \overset{def}{=} \left(\sigma_{y_j}^K\right)^2 / \left(\sigma_{y_j-\hat{y}_j}^K\right)^2$ are signal-to-noise ratios with σ_{y_j}, $\sigma_{y_j}^K$ as standard deviations of data on measuring and prediction horizons, respectively; $\sigma_{y_j-\hat{y}_j}^K$ is the standard deviation of prediction error. The bigger the norm of $\mathbf{PQ} = \left[\mathrm{PQ}_1 \cdots \mathrm{PQ}_{ny}\right]^T$, the better the prediction performance. The best predictors in terms of PQ are found by means of an adapted version of *Particle Swarm Optimization Algorithm* (PSO) [5].

3. Prediction models based on Kalman filtering theory

The third prediction model of distributed ts is developed in context of state representation identification models [2], [4], [6]. More specifically, a minimum state representation is obtained by a technique introduced in [6]. The resulted model is represented by the following discrete time state equations:

$$\begin{cases}\mathbf{x}[k+1] = \mathbf{A}_k\mathbf{x}[k] + \mathbf{B}_k^{\mathbf{x}}\mathbf{u}[k] + \mathbf{F}_k\mathbf{w}[k] \\ \mathbf{y}[k] \quad = \mathbf{C}_k\mathbf{x}[k] + \mathbf{B}_k^{\mathbf{y}}\mathbf{u}[k] + \mathbf{D}_k\mathbf{v}[k]\end{cases}, \quad \forall k \in \mathbb{N}, \tag{10}$$

where: $\mathbf{A}_k \in \mathbb{R}^{nx \times nx}$, $\mathbf{B}_k^{\mathbf{x}} \in \mathbb{R}^{nx \times nu}$, $\mathbf{B}_k^{\mathbf{y}} \in \mathbb{R}^{ny \times nu}$, $\mathbf{C}_k \in \mathbb{R}^{ny \times nx}$, $\mathbf{D}_k \in \mathbb{R}^{ny \times ny}$ and $\mathbf{F}_k \in \mathbb{R}^{nx \times nx}$ are matrices including all variable (but already estimated) parameters of some stochastic process; (usually, $\mathbf{D}_k = \mathbf{I}_{ny}$); $\mathbf{y} \in \mathbb{R}^{ny}$ is the vector of (measurable) output signals; $\mathbf{u} \in \mathbb{R}^{nu}$ is the vector of input signals; $\mathbf{x} \in \mathbb{R}^{nx}$ is the unknown state vector that encodes the invisible correlations between data sets; $\mathbf{w} \in \mathbb{R}^{nx}$ is the (unknown) endogenous non auto-correlated system noise, with a sparse covariance matrix $\mathbf{\Psi}_{\mathbf{w}}[k] = E\{\mathbf{w}[k]\mathbf{w}^T[k]\}$; $\mathbf{v} \in \mathbb{R}^{ny}$ is the (unknown) non auto-correlated exogenous noise, which is usually corrupting the measured data; its covariance matrix, $\mathbf{\Psi}_{\mathbf{v}}[k] = E\{\mathbf{v}[k]\mathbf{v}^T[k]\}$, is usually diagonal. The two noises are moreover uncorrelated each other.

The problem of state prediction (and, consequently, of outputs data prediction), is solved in the context of Kalman-Bucy Filtering Theory [3,4]. We succeeded to design the particular algorithm below, based on some mathematical properties of model (10).

➢ *<u>Input data</u>: a small collection of time series values (the training set $\mathcal{D}_0 = \{\mathbf{y}[n]\}_{n \in \overline{1,N_0}}$) yielding initialization.*

1. Initialization. Produce the first state representation (10). Then complete the initialization by setting: an arbitrary state vector $\hat{\mathbf{x}}_0$, the covariance matrices $\hat{\mathbf{P}}_0 = \alpha\mathbf{I}_{nx}$ (with $\alpha \in \mathbb{R}_+^*$), $\mathbf{F}_{-1}\mathbf{\Psi}_{\hat{\mathbf{w}}}[-1]\mathbf{F}_{-1}^T = \mathbf{0}_{nx}$ and $\mathbf{D}_{-1}\mathbf{\Psi}_{\hat{\mathbf{v}}}[-1]\mathbf{D}_{-1}^T = \mathbf{0}_{ny}$.

2. For $k \geq 0$:

2.1. Estimate the exogenous mixed noise: $\mathbf{D}_k\hat{\mathbf{v}}[k] = \mathbf{y}[k] - \mathbf{C}_k\hat{\mathbf{x}}[k] - \mathbf{B}_k^{\mathbf{y}}\mathbf{u}[k]$.

2.2. Update the covariance matrix of exogenous noise:

$$\mathbf{D}_k\mathbf{\Psi}_{\hat{\mathbf{v}}}[k]\mathbf{D}_k^T = \frac{1}{k+1}\left(k\mathbf{D}_{k-1}\mathbf{\Psi}_{\hat{\mathbf{v}}}[k-1]\mathbf{D}_{k-1}^T + \mathbf{D}_k\hat{\mathbf{v}}[k]\hat{\mathbf{v}}^T[k]\mathbf{D}_k^T\right).$$

2.3. Compute the auxiliary matrix: $\mathbf{Q}_k = \mathbf{C}_k\hat{\mathbf{P}}_k$.

2.4. Invert the matrix: $\mathbf{R}_k = \mathbf{D}_k\mathbf{\Psi}_{\hat{\mathbf{v}}}[k]\mathbf{D}_k^T + \mathbf{Q}_k\mathbf{C}_k^T \in \mathbb{R}^{ny \times ny}$.

2.5. Evaluate the sensitivity gain: $\mathbf{\Gamma}_k = \mathbf{Q}_k^T\mathbf{R}_k^{-1}$.

2.6. Compute the auxiliary matrix $\mathbf{S}_k = \mathbf{A}_k\mathbf{\Gamma}_k$.

2.7. Update the covariance matrix of endogenous noise:

$$\mathbf{F}_k\mathbf{\Psi}_{\hat{\mathbf{w}}}[k]\mathbf{F}_k^T = \frac{1}{k+1}\left(k\mathbf{F}_{k-1}\mathbf{\Psi}_{\hat{\mathbf{w}}}[k-1]\mathbf{F}_{k-1}^T + \mathbf{S}_k\mathbf{D}_k\hat{\mathbf{v}}[k]\hat{\mathbf{v}}^T[k]\mathbf{D}_k^T\mathbf{S}_k^T\right).$$

2.8. Update the covariance matrix of estimation error:

$$\hat{\mathbf{P}}_{k+1} = \mathbf{F}_k \boldsymbol{\Psi}_{\hat{\mathbf{w}}}[k]\mathbf{F}_k^T + \mathbf{A}_k\left(\hat{\mathbf{P}}_k - \boldsymbol{\Gamma}_k\mathbf{Q}_k\right)\mathbf{A}_k^T .$$

2.9. Predict the state: $\hat{\mathbf{x}}[k+1] = \mathbf{A}_k\hat{\mathbf{x}}[k] + \mathbf{B}_k^{\mathbf{x}}\mathbf{u}[k] + \mathbf{S}_k\mathbf{D}_k\hat{\mathbf{v}}[k]$.

2.10. Predict the output: $\hat{\mathbf{y}}[k+1] = \mathbf{C}_k\hat{\mathbf{x}}[k+1] + \mathbf{B}_k^{\mathbf{y}}\mathbf{u}[k+1]$.

2.11. Acquire new data: $\mathcal{D}_{k+1} = \mathcal{D}_k \cup \left\{\mathbf{y}[k+1]\right\}$.

2.12. Update the state model.

➢ *<u>Output data</u>:*

- *predicted time series values:* $\left\{\hat{\mathbf{y}}[k]\right\}_{k\in\mathbb{N}^*}$ *;*
- *estimated covariance matrices:* $\left\{\mathbf{D}_k\boldsymbol{\Psi}_{\hat{\mathbf{v}}}[k]\mathbf{D}_k^T\right\}_{k\in\mathbb{N}^*}$.

The predictor performance is assessed by means of PQ as well. This time, $\left\{\hat{\sigma}_{j,k}^2\right\}_{j\in\overline{1,ny}}$ is the diagonal of each matrix $\mathbf{D}_{N_y+k}\boldsymbol{\Psi}_{\hat{\mathbf{v}}}[N_y+k]\mathbf{D}_{N_y+k}^T$. The optimum structural indices (i.e. the polynomial degrees and the states number) are selected through PSO technique, like in case of previous prediction models.

4. Simulation results

Predictors performances are tested on a case study coming from Meteorology. Daily minimum and maximum temperatures of the two neighboring cities from Fig.1 have been monitored and predicted. Three designed predictors, namely PARMA, PARMAX and KARMA, have been implemented within MATLAB environment. Their performances in terms of PQ have been compared over a data block of 479 samples on 4 channels. The resulted values of PQ (after comparing the predicted data to the measured data) are listed in Tab. 1.

Tab. 1. Prediction Quality [%] for different predictors and channels.

Channel index	Predictor		
	PARMA	PARMAX	KARMA
1	54.38	66.25	**69.46**
2	53.61	**68.50**	60.19
3	68.37	**75.92**	70.78
4	55.79	**76.86**	59.79

As one can see, the first predictor (PARMA) is the worst, because no correlation between measuring channels are accounted. On the contrary, the last two predictors (PARMAX and KARMA) produce better predicted values. Although none of them could be declared as “the best” on all channels, PARMAX is seemingly the most suitable choice (with 3 best predicted values over 4). Note that not only the predicted data are considered when computing PQ, but the aperture of corresponding confidence tube too. This tube is configured around predicted data, by accounting the 3σ-rule on Gaussian stochastic processes [2,4].

The results in Tab. 1 reveal that the temperatures from the two cities are rather correlated than uncorrelated, as expected (given their geographical neighboring). This is another useful insight regarding the use of the three predictors. Whenever the multi-channel data are weakly or not correlated, PARMA should be preferred as a simpler and faster predictor. Whenever the cross-correlation between different channels is quite obvious, either PARMAX or KARMA should be used as more accurate predictor.

5. Conclusion

This paper succinctly described an approach to multi-variable physical data prediction. Three different predictors have been designed in this aim, by using stationary models. Although simulations lead to several useful insights, the most important is that PARMAX and KARMA predictors perform better than PARMA as long as data are cross-correlated. However, the prediction accuracy has increased at the expense of computational complexity. If the data are quite uncorrelated across channels, PARMA should be employed as first option. As future developments, predictors based on non stationary models are under consideration.

Acronyms list

ARMA – Auto-Regressive with Moving Average (identification) model [2,4]

ARMAX – Auto-Regressive with Moving Average and eXogenous control (identification) model [2,4]

do – one step delay operator: $\left(\mathrm{q}^{-1} y\right)[n] = y[n-1]$, $\forall n \in \mathbb{Z}$ [4]

MIMO – Multiple-Input Multiple-Output (system) [7]

MISO – Multiple-Input Single-Output (system) [7]

MPEM – Mimimum Prediction Error Method (an identification procedure for models corrupted by stochastic perturbations) [2], [4]

SISO – Single-Input Single-Output (system) [7]

PQ – Prediction Quality (cost function that has to be maximized)

PSO – Particle Swarm Optimization (algorithm) [5]

ts – time series (acquired data strings)

wn – white noise (the prototype of totally uncorrelated/unpredictable stochastic processes) [2,4]

REFERENCES

[1] *Z. Wang, D. Ziou, et al.,* A Comparative Analysis of Image Fusion Methods”, IEEE Trans. on Geoscience and Remote Sensing, Vol. 43, No. 6, pp. 1391-1402, (2005).

[2].*T. Söderström, P. Stoica,* System Identification, Prentice Hall, London, UK, (1989).

[3]. *R.E. Kalman, R.S. Bucy*, New Results in Linear Filtering and Prediction Theory”, Transactions of ASME, Journal of Basic Engineering, Series D, Vol. 83, pp. 95-108. (1961).

[4]. D. *Stefanoiu, J. Culita, P. Stoica* A Foundation to System Identification and Modeling. PRINTECH Press, Bucharest, Romania, (2005).

[5]. *J. Kennedy, R. Eberhart* (1995). „Particle Swarm Optimization”, IEEE International Conference on Neural Networks, Piscataway – N.J., USA, pp. 1942-1948.

[6]. *P. van Overschee., B. de Moor*, Subspace Identification of Linear Systems: Theory, Implementation, Applications. Kluwer Academic Publishers, Holland. (1996).

[7]. *L. von Bertalanffy* General System Theory: Foundations, Development, Applications. George Braziller Editions, New York, USA, (1968).