Separable Autoregressive Moving Average Graph-Temporal Filters

Elvin Isufi*, Andreas Loukas†, Andrea Simonetto* Geert Leus*
* Faculty of EEMCS, Delft University of Technology, 2826 CD Delft, The Netherlands
† Department of Telecommunication Systems, Technical University of Berlin, 10587 Berlin, Germany
e-mails: {e.isufi-1, a.simonetto, g.j.t.leus}@tudelft.nl, a.loukas@tu-berlin.de

***Abstract*—Despite their widespread use for the analysis of graph data, current graph filters are designed for graph signals that do not change over time, and thus they cannot simultaneously process time and graph frequency content in an adequate manner. This work presents ARMA^{2D}, an autoregressive moving average graph-temporal filter that captures jointly the signal variations over the graph and time. By its unique nature, this filter is able to achieve a separable 2-dimensional frequency response, making it possible to approximate the filtering specifications along both the graph and temporal frequency domains. Numerical results show that the proposed solution outperforms the state of the art graph filters when the graph signal is time-varying.**

***Index Terms*—signal processing over graphs, graph filters, separable graph-temporal filters, distributed signal processing**

I. Introduction

In the field of signal processing over graphs [1], graph filters are the basic primitives for the processing of graph signals. Graph filters originate from the extension of the Fourier transform from signals residing on regular domains like time and space, to signals that reside on the vertices of an irregular graph [1]–[5]. A graph filter is the direct analog of the classical temporal filter, but now operating on the graph signal by amplifying or attenuating part of its graph spectrum. Relevant applications of graph filters range from data classification [2], signal denoising and smoothing [6], [7] to reaching consensus [8], and anomaly detection [9].

Similar to their temporal analogs, graph filters can achieve a finite impulse response (FIR) [2], [10], [11] as well as an infinite impulse response (IIR) [12], [13] form. Nevertheless, despite approximating relatively well a desired graph frequency response when the graph signal is *time-invariant*, state-of-the-art graph filters are largely unsuitable for filtering graph signals *that change over time*. This is especially true when the filtering specifications are set w.r.t. both the graph and temporal domains. Whereas FIR graph filters completely ignore the temporal frequency content of the signal, our previously proposed autoregressive moving average (ARMA) IIR graph filters suffer from design constraints, allowing for the approximation of only a limited family of 2-dimensional frequency responses.

In this paper, we propose ARMA^{2D}, a generalization of ARMA graph filters which overcomes some of the aforementioned issues. Our design allows for the distributed computation of a wider class of frequency responses at the price of a communication and computational complexity increment (w.r.t. ARMA), which is however asymptotically negligible when the number of graph edges increases. We focus on responses which are separable w.r.t. the graph and temporal frequency, and show how to separably design the filter coefficients to meet given specifications in the graph *and* time domain. Further, the proposed ARMA^{2D} relaxes the stability constraints of the ARMA graph filters, allowing us to find the filter coefficients separably in each domain while ensuring the joint stability. These properties give the ARMA^{2D} the potential to improve the approximation accuracy of the filters. To the best of our knowledge, we are not aware of any other graph filters possessing these properties, even at a higher complexity.

We conclude this work showcasing the approximation quality of our filters and displaying the added value of our approach, allowing for different separable filter specifications in the graph and time domain. Numerical results demonstrate that ARMA^{2D} filters outperform the state of the art universal FIR graph filters (i.e., where the filter coefficients are designed for a continuous range of graph frequencies) when the signal on the graph is time varying.

II. Preliminaries

Let us consider an undirected[1] and connected graph $\mathcal{G}$ of N nodes and M edges. We indicate with $\boldsymbol{x} \in \mathbb{R}^N$ the graph signal and with $\boldsymbol{L}$ the graph Laplacian.

Graph Fourier transform (GFT). The GFT [1] $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ and its inverse are calculated as

$$\hat{x}_i = \langle \boldsymbol{x}, \boldsymbol{\phi}_i \rangle, \text{ and } x_i = \sum_{n=1}^{N} \hat{x}_n \boldsymbol{\phi}_n(i), \tag{1}$$

where $\langle \cdot \rangle$ denotes the inner product, $\{\boldsymbol{\phi}_n\}_{n=1}^N$ are the Laplacian's eigenvectors and $\boldsymbol{\phi}_n(i)$ is the ith entry of $\boldsymbol{\phi}_n$.. The corresponding eigenvalues $\{\lambda_n\}_{n=1}^N$ form the graph spectrum. To avoid any restrictions on the applicability of the proposed approach, we present the results for a general Laplacian matrix $\boldsymbol{L}$. We only require $\boldsymbol{L}$ to be symmetric and local: for all $i \neq j$, $L_{ij} = 0$ whenever the nodes u_i and u_j are not neighbors and $L_{ij} = L_{ji}$ otherwise.

Graph filters. A graph filter $\boldsymbol{H}$ is defined as a linear operator that acts on a graph signal $\boldsymbol{x}$ by amplifying or attenuating different parts of its spectrum as

$$\boldsymbol{H}\boldsymbol{x} = \sum_{n=1}^{N} H(\lambda_n)\, \hat{x}_n \phi_n. \tag{2}$$

[1]We present our results for undirected graphs, yet the core idea can be applied to directed ones using the adjacency matrix instead of the Laplacian.

Let λ_{min} and λ_{max} be the minimum and the maximum eigenvalues of $\boldsymbol{L}$. The graph frequency response $H : [\lambda_{min},\ \lambda_{max}] \to \mathbb{R}$ controls how much $\boldsymbol{F}$ amplifies the signal component of each graph frequency. Given a desired graph frequency response $H^*(\lambda_n)$, the filter coefficients can be found by solving a linear system when the underlying graph structure (i.e., the graph frequencies λ_n) is known [1], [2]. On the other hand, we can use a polynomial approximation [10] to design a universal filter, i.e., design the filter coefficients independently from the graph structure. The latter describes the graph frequency response for any graph. The filter output of an FIR_K universal filter can be calculated as

$$\boldsymbol{y} = \left(\varphi_0 \boldsymbol{I} + \sum_{k=1}^{K} \varphi_k \boldsymbol{L}^k\right) \boldsymbol{x}, \tag{3}$$

with $\varphi_0, \ldots, \varphi_K$ being the filter coefficients.

ARMA_K graph filters. In [13], we introduced the ARMA_K graph filters with the goal to implement IIR filters on graphs. This allowed us to better track a time-varying input signal, since the input enters in the computations at every iteration, and not only at the beginning, as implemented in FIR graph filters. In case of a time-varying input signal, the (parallel) ARMA_K recursion has the form

$$\boldsymbol{y}_{t+1}^{(k)} = \psi^{(k)} \boldsymbol{M} \boldsymbol{y}_t^{(k)} + \varphi^{(k)} \boldsymbol{x}_t \tag{4a}$$

$$\boldsymbol{z}_{t+1} = \sum_{k=1}^{K} \boldsymbol{y}_{t+1}^{(k)} + c\boldsymbol{x}_t, \tag{4b}$$

for arbitrary $\boldsymbol{y}_0^{(k)}$ and where the coefficients $\psi^{(k)}$, $\varphi^{(k)}$ and c are the complex-valued filter coefficients. The matrix $\boldsymbol{M}$ is a translated version of the Laplacian matrix $\boldsymbol{L}$ defined as

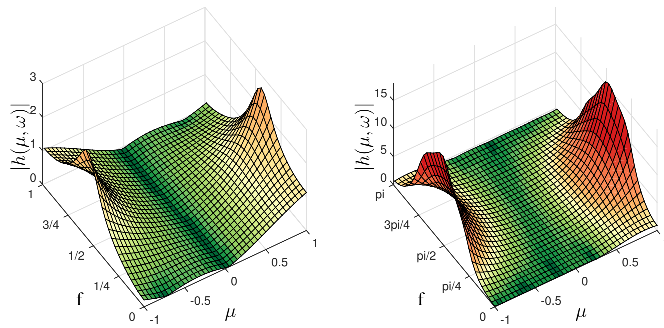
$$\boldsymbol{M} = \varrho \boldsymbol{I} - \boldsymbol{L}, \text{ with } \varrho = \frac{\lambda_{max} + \lambda_{min}}{2}, \tag{5}$$

which is chosen so as to increase the stability region of the filter (considering that $\|\boldsymbol{M}\|_2 \leq \varrho \leq \|\boldsymbol{L}\|_2$). Since a translation does not influence the eigenvectors of $\boldsymbol{L}$, we now attain the desired response $H(\lambda)$ by mapping it to the domain of $\boldsymbol{M}$'s eigenvalues (μ): $G(\mu) = H(\varrho - \lambda)$. From Sylvester's matrix theorem it is known that the eigenvalue μ_n of $\boldsymbol{M}$ is related to the eigenvalue λ_n of $\boldsymbol{L}$ as $\mu_n = \varrho - \lambda_n$. For instance, for the normalized Laplacian we obtain $G(\mu) = H(1 - \lambda)$, whereas for the standard Laplacian we have $G(\mu) = H(\lambda_{max}/2 - \lambda)$.

The graph and temporal frequency response of (4) is [13]

$$H(\mu, z) = \sum_{k=1}^{K} \frac{\varphi^{(k)} z^{-1}}{1 - \psi^{(k)} \mu z^{-1}} + cz^{-1}. \tag{6}$$

As we can see, the parallel ARMA_K recursion (4) is now an ARMA_K filter in both the graph and temporal domain. A deeper analysis of (6) reveals however a series of challenges. First of all, ARMA_K cannot approximate any desired 2-dimensional response. Indeed, the transfer function (6) presents a strong correlation between the graph (μ) and temporal frequencies (z). Another major challenge is that we have to deal with stability constraints in the design phase. The coefficients $\psi^{(k)}$ must be designed to approximate a given 2-dimensional frequency response, while ensuring the stability in the graph and temporal domain. The stable region of the coefficients $\psi^{(k)}$ is also limited by the fact that the K distinct poles of (6) obey $z = \psi^{(k)}\mu$. This reduces the stable placement area of the poles, since the poles in one domain will influence the ones in the other domain.



(a) ARMA_3 (b) ARMA_6

Fig. 1. The joint graph and temporal frequency response of two ARMA_K graph filters, both designed to approximate an ideal low pass (step) response in the graph domain with (translated) cut-off frequency $\mu_c = 0.5$ and $K = 3, 6$. A normalized Laplacian has been used (illustrated with respect to its translated version) and also the temporal frequencies are normalized ($\times\pi$ rad/sample).

In order to illustrate the latter, in Fig. 1, we have plotted the joint graph and temporal frequency response of two ARMA_K filters, for $K = 3, 6$. Both filters are designed to approximate an ideal step function in the graph domain with cut-off frequency $\lambda_c = 0.5$. Fig. 1 depicts the translated graph frequency $\mu = \varrho - \lambda$ (we adopt the normalized Laplacian for which $\varrho = 1$). The temporal axis, on the other hand, measures the normalized temporal frequency f such that, for $f = 0$, one obtains the standard graph frequency response. As we can see, both ARMAs present instability issues for higher temporal frequencies f and this becomes more evident when the order is higher. For these particular cases, the joint frequency responses are also characterized by an antisymmetry around the point $(0,\ 1/2)$, a property that has been empirically observed for different ARMA_K filters of this form.

III. ARMA^{2D} GRAPH FILTERS

In order to overcome the limitations of ARMA_K filters, in this section we propose ARMA^{2D} graph filters. Trading off computational and communication complexity, ARMA^{2D} presents the following benefits: *(i)* it improves the approximation accuracy and stability of the filters, *(ii)* it achieves a separable frequency response in the graph and time domain, *(iii)* it can approximate any prescribed separable 2-dimensional frequency response, and *(iv)* it achieves a rational frequency response with any (not necessarily the same) order in the graph and time domain.

We start by presenting the ARMA^{2D} filters and analyze their stability. We then discuss their distributed computation, and propose solutions for the 2-dimensional filter design problem.

ARMA^{2D} graph filters. Consider the enhanced ARMA recursion

$$\sum_{l=0}^{L} \sum_{p=0}^{P} \psi_l a_p \boldsymbol{M}^l \boldsymbol{y}_{t-p} = \sum_{k=0}^{K} \sum_{q=0}^{Q} \varphi_k b_q \boldsymbol{M}^k \boldsymbol{x}_{t-q}, \tag{7}$$

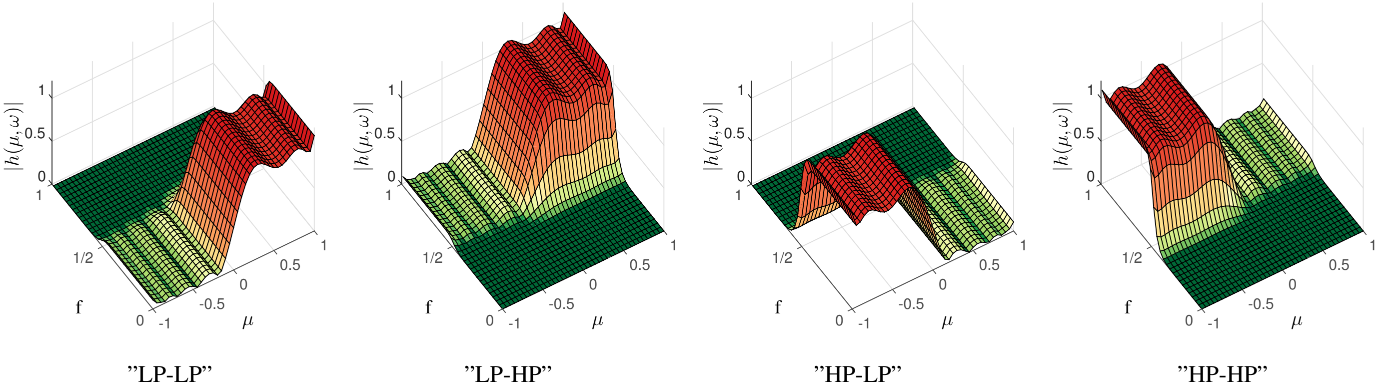


Fig. 2. Different 2-dimensional filter approximations. From left to right, we go from a low-pass (LP) filter in both graph and temporal frequency domain to a high-pass (HP) filter in both domains. The joint filter is an FIR_{10} in the graph domain and a Butterworth of order 6 in time.

where ψ_l, φ_k, a_p, b_q are coefficients to be chosen. Computing the output $\boldsymbol{y}_t$ now involves shifts of different orders in the graph domain, indicated by the powers of $\boldsymbol{M}$, and in the time domain, indicated by the temporal memory of the past output and input signal. This seems to indicate that we can capture the signal variations in both domains, and by introducing more degrees of freedom, the approximation accuracy can potentially increase. Proposition 1 asserts that the above intuition holds.

Proposition 1: Assuming that the coefficients ψ_l and a_p are chosen to ensure stability (see proof for exact conditions), the joint transfer function of recursion (7) is

$$\begin{aligned} H(\mu, z) &= \frac{\sum_{k=0}^{K}\sum_{q=0}^{Q}\varphi_k b_q \mu^k z^{-q}}{\sum_{l=0}^{L}\sum_{p=0}^{P}\psi_l a_p \mu^l z^{-p}} \qquad (8)\\ &= \left[\frac{\sum_{k=0}^{K}\varphi_k\mu^k}{\sum_{l=0}^{L}\psi_l\mu^l}\right]\left[\frac{\sum_{q=0}^{Q}b_q z^{-q}}{\sum_{p=0}^{P}a_p z^{-p}}\right] = H_g(\mu)H_t(z). \end{aligned}$$

Proof (Sketch): Assume that the filter is stable and consider a reduced dimension problem for (7). This can be done by setting $\boldsymbol{x}_t = x_t\boldsymbol{\phi}$, with x_t the scalar magnitude of $\boldsymbol{x}_t$ in the eigenspace of $\boldsymbol{\phi}$. By considering the orthogonality of the Laplacian eigenbasis, we can then rewrite (7) as

$$\sum_{l=0}^{L}\sum_{p=0}^{P}\psi_l a_p \mu^l y_{t-p} = \sum_{k=0}^{K}\sum_{q=0}^{Q}\varphi_k b_q \mu^k x_{t-q}, \qquad (9)$$

where $y_t \in \mathbb{C}$ is the magnitude of $\boldsymbol{y}_t \in \mathbb{C}^n$ in the eigenspace of $\boldsymbol{\phi}$. Taking the z-transform on both sides, we obtain the joint transfer function (8).

Let us analyze the stability condition for which recursion (7) converges. For simplicity, we focus on the dimensionality reduced equivalent problem (9), but the result also holds for (7). We start by rewriting (9) as

$$a_0\sum_{l=0}^{L}\psi_l\mu^l y_t + \sum_{l=0}^{L}\sum_{p=1}^{P}\psi_l a_p\mu^l y_{t-p} = \sum_{k=0}^{K}\sum_{q=0}^{Q}\varphi_k b_q\mu^k x_{t-q}. \qquad (10)$$

We then define the $P \times 1$ vector $\tilde{\boldsymbol{y}}_t = [y_{t-P+1}, y_{t-P+2}, \ldots, y_{t-1}, y_t]^\top$, the $(Q+1)\times 1$ vector $\tilde{\boldsymbol{x}}_t = [x_{t-Q}, x_{t-Q+1}, \ldots, x_{t-1}, x_t]^\top$, $\Psi = \sum_{l=0}^{L}\psi_l\mu^l$ and $\Phi = \sum_{k=0}^{K}\varphi_k\mu^k$. With these definitions in place, we rewrite (9) as

$$a_0\Psi\tilde{\boldsymbol{y}}_t = \boldsymbol{A}\tilde{\boldsymbol{y}}_{t-1} + \boldsymbol{B}\tilde{\boldsymbol{x}}_t, \qquad (11)$$

with $\boldsymbol{A}$ and $\boldsymbol{B}$ the $P \times P$ and $(Q+1)\times(Q+1)$ matrices, respectively, defined as

$$\boldsymbol{A} = \begin{bmatrix} 0 & a_0\Psi & \ldots & 0\\ 0 & 0 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & a_0\Psi\\ -a_P\Psi & -a_{P-1}\Psi & \ldots & -a_1\Psi \end{bmatrix} \boldsymbol{B} = \begin{bmatrix} 0 & \ldots & 0\\ 0 & \ldots & 0\\ \vdots & \ddots & \vdots\\ 0 & \ldots & 0\\ b_Q\Phi & \ldots & b_0\Phi \end{bmatrix}.$$

We can rewrite (11) as

$$\tilde{\boldsymbol{y}}_t = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{y}}_{t-1} + \tilde{\boldsymbol{B}}\tilde{\boldsymbol{x}}_t, \qquad (12)$$

with $\tilde{\boldsymbol{A}} = (a_0\Psi)^{-1}\boldsymbol{A}$ and $\tilde{\boldsymbol{B}} = (a_0\Psi)^{-1}\boldsymbol{B}$. It is then clear that (12) and thus (7) converges if the eigenvalues of $\tilde{\boldsymbol{A}}$ are lower than one in magnitude, i.e., $\|\boldsymbol{A}\| < |a_0\Psi|$, and if $(a_0\Psi)^{-1}$ is different from zero. ■

It follows that recursion (7) implements an ARMA(L, K) in the graph domain and an ARMA(P, Q) in the time domain. The ARMA^{2D} filters thus have the potential to achieve a better approximation quality of a 2-dimensional desired response, as compared to (4). Notice that (7) is a special form of implementing 2-dimensional filters, thus it will not collapse to a pure ARMA graph filter when the graph signal is time-invariant. This is because we will have $\boldsymbol{y}_t$ on both sides of (7).

The following Corollary presents a sufficient condition to obtain stable ARMA^{2D} filters and proposes how to obtain joint stability while designing the filter coefficients separably in each domain.

Corollary 1: Assuming $a_0 = 1$ and defining $\boldsymbol{A}_0 = \sum_{l=0}^{L}\psi_l\boldsymbol{M}^l$, the ARMA^{2D} recursion (7) can be expressed as

$$\boldsymbol{A}_0\boldsymbol{y}_t + \sum_{l=0}^{L}\sum_{p=1}^{P}\psi_l a_p\boldsymbol{M}^l\boldsymbol{y}_{t-p} = \sum_{k=0}^{K}\sum_{q=0}^{Q}\varphi_k b_q\boldsymbol{M}^k\boldsymbol{x}_{t-q}. \qquad (13)$$

It can be shown that (13) converges if $\boldsymbol{A}_0$ is non singular and the temporal ARMA(P, Q) has poles inside the unit circle.

Proof (Sketch): Let us start by noticing that the requirement $a_0 = 1$ can be easily satisfied considering that it is just a normalization factor. The first claim that $\boldsymbol{A}_0$ must not be singular comes from the expression (12), where we now need to take the inverse of $\boldsymbol{A}_0$. This means that only the graph filter coefficients ψ_l and φ_k must be tuned to satisfy this condition. Then, for the ARMA^{2D} to be stable we will require $\tilde{\boldsymbol{A}}$ to have eigenvalues smaller than one in magnitude. Considering that now the structure of $\tilde{\boldsymbol{A}}$ will depend only on the temporal coefficients a_p, this is equivalent to say that the poles of the temporal ARMA(P, Q) filter lie inside the unit circle. ■

The main message that Corollary 1 delivers is that now we can separably find the filter coefficients in the graph and temporal domain and still achieve stable ARMA^{2D} filters. Even-though one can find the condition that $\boldsymbol{A}_0$ is not singular for a general graph matrix $\boldsymbol{M}$, we must notice that in order to be implemented (and thus obtain the output $\boldsymbol{y}_t$) the graph structure must be known. This means that, under the stability condition, we need to invert the matrix $\boldsymbol{A}_0$ which prohibits an efficient distributed implementation. The following corollary[2] provides an alternative way to achieve separability without requiring knowledge of the graph spectrum. Note that this way of designing the filter coefficients avoids computing the eigendecomposition of the graph Laplacian, which has complexity $O(N^3)$, at the price of a decrease in approximation accuracy of the desired response. This phenomenon is present also for the universal FIR$_K$ and ARMA$_K$ graph filters.

Corollary 2: Setting $L = 0$ and $\psi_0 = 1$ in (7), its joint transfer function becomes

$$H(\mu, z) = \left[\sum_{k=0}^{K} \varphi_k \mu^k\right] \left[\frac{\sum_{q=0}^{Q} b_q z^{-q}}{\sum_{p=0}^{P} a_p z^{-p}}\right]. \qquad (14)$$

By restricting L and ψ_0, which means that now we have an ARMA$(0, K)$ = FIR$_K$ in the graph domain and an ARMA(P, Q) in time, we lift the requirement of knowing the graph spectrum during the design phase of the filter. Moreover, except for specific cases, setting $L = 0$ does not decrease the approximation accuracy[3] w.r.t. the ARMA^{2D}. On the contrary, for $L = 0$, the communication complexity decreases, and the stability condition simplifies. Indeed, in this case, the 2-dimensional filter is stable as long as the temporal ARMA(P, Q) is stable, i.e., the poles lie inside the unit circle. Due to its benefits, we will showcase this approach in the numerical results.

Distributed computation. We consider the standard message-passing model [14] where the input graph is the same as the network over which the computation is performed. We will assume that any message exchange takes much shorter than the sampling period of the signal. We quantify the efficiency of the filtering in terms of the *per-timestep communication complexity*, defined as the number of bits the network needs to exchange to compute $\boldsymbol{y}_t$ given $\boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_0$. Let us focus on the special case that $\boldsymbol{A}_0 = \boldsymbol{I}$ (see Corollary 1), for which recursion (7) is efficiently computable. The recursion involves the use of the terms $\boldsymbol{M}^l \boldsymbol{y}_{t-p}$ and $\boldsymbol{M}^k \boldsymbol{x}_{t-q}$ for all $k, l, p, q > 0$, however, only the terms $\boldsymbol{M}^1 \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{M}^l \boldsymbol{y}_{t-1}$ and $\boldsymbol{M}^1 \boldsymbol{x}_t, \ldots, \boldsymbol{M}^k \boldsymbol{x}_t$ have not been computed during a previous timestep. Since in both cases we are dealing with successive powers of $\boldsymbol{M}$, we can reduce the computation effort by obtaining $\boldsymbol{M}^l \boldsymbol{y}_{t-1}$ from $\boldsymbol{M}^{l-1} \boldsymbol{y}_{t-1}$ and so on (the same holds for $\boldsymbol{M}^k \boldsymbol{x}_t$). Thus, we will need a total of $K + L$ multiplications with matrix $\boldsymbol{M}$. Since $\boldsymbol{M}$ is a local matrix, the network can perform the multiplication of $\boldsymbol{M}$ with any graph signal distributedly, by exchanging $2M$ values. It follows that the per-timestep communication complexity is $2M(K+L) \times c_r = O(MK)$ bits, where c_r is the architecture-dependent representation length of each scalar and, w.l.o.g., we assume that $K \geq L$. The per-timestep computational complexity of ARMA^{2D} is therefore less than twice that of ARMA$_K$ (which is $2MK \times c_r$ bits per timestep) in the general case, and equivalent when $L = 0$.

The filter design problem. We now present the design problem of a 2-dimensional IIR graph-temporal filter in order to approximate any prescribed frequency response $H^*(\mu, z)$. We focus our attention on approximating $H^*(\mu, z)$ with the filter (14) since it suits better for a continuous range of frequencies in both the graph and the temporal frequency domain. This design approach is mostly desirable when the graph structure is unknown to the designer or when it is time-varying either deterministically [13] or stochastically [15]. Further, as previously mentioned, computationally more efficient in large graphs. In this way, we can design the filter coefficients only once and they will be the same for all graph realizations.

Given a 2-dimensional frequency response $H^* : [\mu_{min}, \mu_{max}] \times [\omega = 0, \omega = 2\pi] \to \mathbb{R}$, the filter coefficients are found by minimizing

$$\int_\mu \int_\omega \left| H(\mu, e^{j\omega}) - H^*(\mu, e^{j\omega}) \right|^2 \mathrm{d}\mu \, \mathrm{d}\omega, \qquad (15)$$

where $H(\mu, e^{j\omega})$ may have the form (6) or (14) (for $z = e^{j\omega}$). In case the recursion (14) is used, we can use the fact that its joint transfer function can be expressed as $H(\mu, e^{j\omega}) = H_g(\mu) H_t(e^{j\omega})$, and thus we can design the 2-dimensional filter separably in each domain. This also allows us to use different filter specifications in the graph and temporal domain. Let us for instance consider a desired filter frequency response of the form $H^*(\mu, e^{j\omega}) = H_g^*(\mu) H_t^*(e^{j\omega})$. The design problem (15) can then be reformulated as finding the respective filters taps to approximate each frequency response independently, minimizing

$$\int_\mu |H_g(\mu) - H_g^*(\mu)|^2 \mathrm{d}\mu \ \text{ and } \ \int_\omega |H_t(e^{j\omega}) - H_t^*(e^{j\omega})|^2 \mathrm{d}\omega \qquad (16)$$

for the graph and temporal filter approximations, respectively. In this way, we can employ any technique to approximate a desired graph frequency response (such as Chebyshev polynomial approximation [10]), and any of the well-established techniques for temporal filter design [16].

IV. Numerical evaluation

We illustrate our approach by first showing that recursion (7) can approximate any type of filter with given specifications

[2]The proof is omitted since it follows from Proposition 1.

[3]Current design methods for universal ARMA graph filters have only been shown to improve the approximation accuracy of universal FIR$_K$ graph filters for specific response functions and in general feature a similar performance [13].

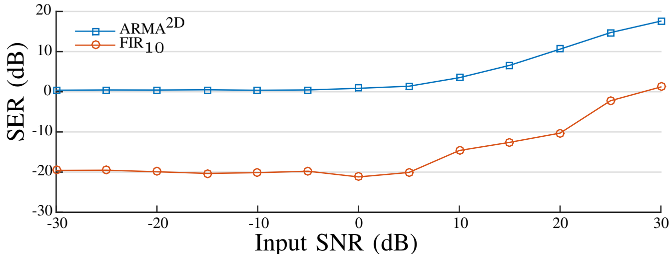


Fig. 3. SER (dB) vs. input SNR (dB) for our presented ARMA^{2D} approach (FIR_{10} in graph and Butterworth 6 in time) and for the classical universal FIR_{10} graph filters. The results are averaged over 10 iterations.

in the graph and temporal frequency domain. Then, we use our proposed filter to recover a band-limited time-varying graph signal which is affected by noise and interference in both the graph and time domain.

Filter approximation. With reference to Fig. 2, we can see that the proposed approach can approximate different desired separable 2-dimensional frequency responses. For this particular case, the cut-off frequencies in both domains are chosen as the half of the respective bands. We have further considered an FIR_{10} in the graph domain and a 6th order Butterworth filter for the temporal domain. The approximation accuracy in each domain is improved by increasing the respective filter orders.

Signal recovery. Let us now consider that the time-varying graph signal has the form $\boldsymbol{x}_t = \boldsymbol{u}_t + \boldsymbol{n}_t$, where $\boldsymbol{u}_t$ is characterized by $\langle \boldsymbol{u}_t, \boldsymbol{\phi}_n \rangle = e^{i\pi t/4}$ if $\lambda_n < \lambda_{max}/2$ and zero otherwise. We consider zero-mean white Gaussian noise with different noise powers. To recover our signal $\boldsymbol{u}_t$ we adopt the double LP filter of Fig. 2 with graph cut-off graph frequency $\lambda_{max}/2$ and temporal cut-off frequency $f_c = 1/2$. The filtering is performed over a graph of 100 randomly placed nodes, and where two nodes are considered neighbors if they are closer than 15% of the maximum distance in the area. Considering that not only the signal but also the noise is time-varying, we expect that the 2-dimensional filter will cancel out the noise spectral component outside of the band of interest not only in the graph domain but also in the temporal domain. To quantify the performance, we define the signal-to-error ratio (SER) as

$$\text{SER}_t = \frac{\|\hat{\boldsymbol{u}}_t\|^2}{\|\hat{\boldsymbol{y}}_t - \hat{\boldsymbol{u}}_t\|^2}, \tag{17}$$

which quantifies how well we cancel the out-of band noise and approximate the filter (notice that our desired response in the graph frequency domain is $\hat{\boldsymbol{u}}_0$).

In Fig. 3 we show the SER (in dB) for different input signal-to-noise ratios (SNRs), for both our presented hybrid approach and the universal FIR filter which operates only on the graph spectral domain. The pure FIR graph filter (3) implicitly assumes that the signal $\boldsymbol{u}_t$ does not oscillates in time (i.e., $\boldsymbol{u}_t = \boldsymbol{u}_0$) and that only the noise is considered time-varying. For a time varying $\boldsymbol{u}_t$, the FIR performance degrades drastically since it does not take the temporal spectrum of the input signal into account. As seen in the figure, ARMA^{2D} outperforms the FIR graph filter for all noise levels. When the noise level decreases, i.e., the input SNR is higher, the SER is lower than the input SNR due to the filter approximation accuracy. We conclude that the proposed 2-dimensional filters, as expected, better suit time-varying environments. Future research will investigate the performance in the case of time-varying graphs.

V. Conclusions

In this work, we have proposed autoregressive moving average graph-temporal filters, which achieved 2-dimensional separable frequency response. We have shows that the proposed graph filters improve the approximation accuracy and the stability w.r.t. the state of the art ARMA graph filters. We also characterize under which conditions the proposed ARMA^{2D} converges. The distributed computation and the 2-dimensional filter design problem have been addressed with a main focus on separable 2-dimensional filters. We have concluded the paper by showing that the proposed 2-dimensional filter can approximate well separable prescribed frequency responses and we show that it outperforms the state of the art graph filters when the graph signal is time-varying.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[2] A. Sandryhaila and J. M. Moura, "Discrete Signal Processing on Graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[3] ——, "Discrete Signal Processing on Graphs: Frequency Analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[4] A. Agaskar and Y. M. Lu, "A Spectral Graph Uncertainty Principle," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.

[5] M. G. Rabbat and V. Gripon, "Towards a Spectral Characterization of Signal Supported on Small-World Networks," in *Proceedings of the 2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 4826 – 4830.

[6] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovacevic, "Signal Denoising on Graphs via Graph Filtering," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Atlanta, GA, USA, December 2014, pp. 872–876.

[7] F. Zhang and E. R. Hancock, "Graph Spectral Image Smoothing Using the Heat Kernel," *Pattern Recognition*, vol. 41, no. 11, pp. 3328–3342, 2008.

[8] A. Sandryhaila, S. Kar, and J. M. Moura, "Finite-Time Distributed Consensus Through Graph Filters," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 1080–1084.

[9] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 1085–1089.

[10] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev Polynomial Approximation for Distributed Signal Processing," in *IEEE International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, Barcelona, Spain, 2011, pp. 1–8.

[11] S. Safavi and U. A. Khan, "Revisiting Finite-Time Distributed Algorithms via Successive Nulling of Eigenvalues," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 54–57, 2015.

[12] A. Loukas, A. Simonetto, and G. Leus, "Distributed Autoregressive Moving Average Graph Filters," *Signal Processing Letters*, vol. 22, no. 11, pp. 1931–1935, 2015.

[13] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Distributed time-varying graph filtering," *arXiv preprint arXiv:1602.04436*, 2016.

[14] E. Ghadimi, "Accelerating Convergence of Large-Scale Optimization Algorithms," Ph.D. dissertation, KTH, Stockholm, Sweden, 2015.

[15] E. Isufi, A. Simonetto, A. Loukas, and G. Leus, "Stochastic Graph Filtering on Time-Varying Graphs," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Cancun, Mexico, 2015, pp. 88–92.

[16] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Prentice–Hall, 1975.