Time-lapse seismic cross-equalization using temporal convolutional networks

Abdullah Alali[1], Robert Smith[2], Philippe Nivlet[2], Andrey Bakulin[2], Tariq Alkhalifah[1]
1 – King Abdullah University of Science and Technology,
2 – Geophysics Technology, EXPEC Advanced Research Center, Saudi Aramco

Summary

Monitoring of geological reservoirs using 4D seismic faces many challenges. The repeatability between different surveys needs to be optimal in which changes are only present in the target zone. Ideal conditions require having the same acquisition parameters for each survey and no near-surface variations, like those caused by seasonal changes. In practice, data processing and matching techniques are required to improve the repeatability of the data. This study proposes a deep learning approach for post-stack trace-by-trace matching to reduce the remaining 4D noise. We utilize the sequential nature of seismic data to train a temporal convolutional network (TCN), which learns to map the monitor traces to the base data in the overburden region. The goal is to suppress 4D noise while maintaining time-lapse signal caused by the reservoir changes we wish to monitor. We validate the method on synthetic time-lapse zero-offset data and show improvements in repeatability. We also perform an initial test on 4D land data to show the potential for application to real datasets.

Introduction

Time-lapse seismic, also known as 4D seismic, is often used to monitor fluid and pressure changes resulting from producing fields or injection of CO_2 into geological formations. Monitoring fluid behavior can improve reservoir simulation history matching and management, which play a significant role in enhancing oil recovery (Jervis et al., 2018).

In seismic monitoring, a survey is repeated at different points in time to observe changes in the target caused by changing reservoir conditions. The repeatability of the experiment determines whether the time-lapse study is a success. Perfect repeatability is obtained when: 1) the acquisition parameters (e.g., sources and receivers positions) for all the surveys are identical; and 2) the elasticity of overburden layers, especially the near-surface, are unchanged. Even with advanced technology, such as permanently buried receivers (Bakulin et al., 2018), nonrepeatable signals (4D noise) are unavoidable due to factors such as ambient noise and seasonal variations in the near-surface (Shulakova et al., 2014). These factors can lead to timeshifts, amplitude changes, and waveform distortions. As a result, significant post-processing is usually required to suppress 4D noise between surveys. The cross-equalization process (Ross et al., 1996; Rickett and Lumley, 2001), which aims to match the different vintages by applying matching filters, is commonly used to improve repeatability. The filters are typically designed in a temporal window where reservoir changes are not expected and can be computed on a global or trace-by-trace basis from the baseline to the monitor survey (Ross et al., 1996). Unfortunately, matching filters are unstable and depend primarily on the filter window design (Lumley et al., 2003). For example, choosing a short window will make the filter sensitive to noise, while a large window can reduce the resolution.

Recently, many geophysical processes have incorporated deep learning techniques, which can provide more robust solutions than conventional approaches. For example, in seismic data processing, Ovcharenko et al. (2019) used deep learning to extrapolate low frequencies from high frequencies, while Slang et al. (2019) applied it to denoising and deblending. For time-lapse processing, Alali et al. (2020a) corrected for timeshifts in the data using a fully-connected layer in the latent space of an autoencoder, Duan et al. (2020) showed that a trained network could outperform a conventional cross-correlation method for estimating timeshifts, while Alali et al. (2020b) suggested using recurrent neural networks to better account for time dependency in the data.

Ideally, a network is trained on diverse datasets, which hopefully generalize to perform well on unseen data. In 4D seismic, it is challenging to generate diverse labels that accommodate various timeshifts, source functions, statics, and so many other factors. Therefore, we suggest training the network on windows from the same data, where 4D reservoir changes are not expected, and then apply it on the target formation to isolate the 4D signal. This approach is similar to the conventional matching filters, which can be regarded as a single one-channel convolutional layer. Compared to the matching filters, the neural network can learn more complex features from the data by using multi-channels, nonlinear activation functions, and many other utilities.

In this study we train a temporal convolutional network (TCN) to match the data from the monitor to the base. A TCN is a convolutional network that utilizes a causal dilation filter to account for the time dependency and has been shown to outperform recurrent neural networks on many tasks (Bai et al., 2018). The architecture has been used in various geophysical applications, such as well-to-seismic tie (Nivlet et al., 2020) and acoustic impedance estimation (Mustafa et al., 2019). We verify the method on synthetic zero-offset time-lapse data and then test it on real 4D post-stack images.

10.1190/segam2021-3583439.1

Temporal convolutional networks (TCN)

A regular convolutional neural network (CNN) uses kernels to extract valuable information from the input images. For time series, we often need to capture information related to the time dependency, which can be long and requires a large kernel size; hence, larger cost. In many cases, such as in real-time applications, we only have information about the past and want to predict the future. Regular CNN kernels include future samples in the training, as shown in the top architecture in Figure 1. The TCN overcomes these limitations by utilizing dilated causal convolutions. In Figure 1, we illustrate the difference between regular, causal and dilated convolution. In causal convolution, the kernels only look back at the past samples to predict the output. Dilated convolution, as introduced by Oord et al. (2016), enables an exponential increase in the receptive field (i.e. time-samples history) at a reduced cost. Comparing the causal and dilation in Figure 1, we can see both have the same number of layers, but the dilated convolution has a larger receptive field. Mathematically, for a sequence x ∈ Rn and a kernel f : 0, .., k , the dilated causal convolution (d) applied on a time sample t is written as:

$$(\mathbf{x} *_d f)(t) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{t-i\cdot d}, \tag{1}$$

where d is the dilation factor. Also, note that (t-i·d) accounts for the direction of the past samples. If d = 1, we have the regular causal convolution. d increases exponentially with the depth of the network to allow for a large receptive field. In practice, the TCN is often combined with weight normalization, residual and skip connections (Bai et al., 2018; Oord et al., 2016).

Method

We replace the conventional matching filters with a TCN network to match the data. Like the matching filters, the model is trained to map the monitor to the baseline in regions with no expected 4D changes. We choose to optimize the network using the mean squared error that can be expressed as:

$$\theta_{glob} = \arg\min \|\hat{B} - B\|_2^2, \tag{2}$$

where B is the baseline and $\hat{B}$ is the predicted baseline. Training a TCN is very cost-efficient compared to other time-series networks, such as RNNs. Therefore, after training on the whole data as a global matching, we fine-tune the network for each trace individually, which will result in a local network per trace. This is written as:

$$\theta_{loc}^{i} = \arg\min \|\hat{B}_i - B_i\|_2^2, \tag{3}$$

where i represents the trace number (i.e., B_i is the i-th trace from B and θ_i is the network trained locally to map the ith trace). The goal for the first global training is to reduce the large data required for training a neural network, while the local training aims to learn the specific equalization needed for that particular trace. For all the networks, we divide the data into overlapping windows in time to generate sufficient training data. After training, the inference is applied on the whole data to match the monitor with the baseline using:

$$\hat{B}_{loc}^{i} = \theta_{loc}(M_i). \tag{4}$$

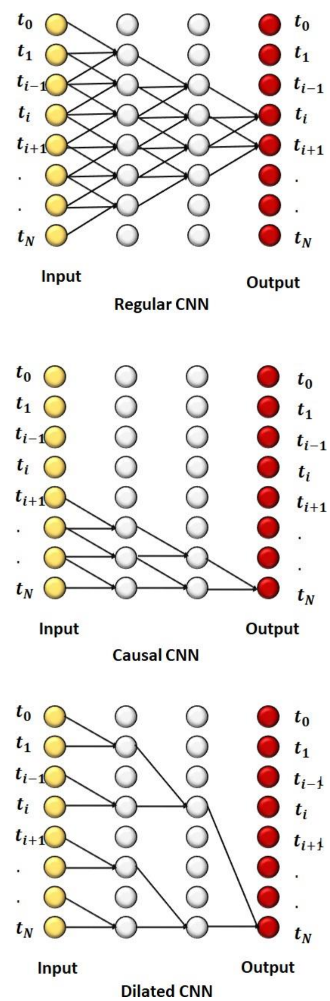

Figure 1: A CNN network and its variants. From top to bottom: regular CNN; causal CNN; dilated causal CNN.

The difference between the actual base B and the predicted $\hat{B}$ should eliminate the overburden effect and keep the 4D changes intact, as long as we do not include any 4D signal in the training.

10.1190/segam2021-3583439.1

Synthetic SEAM time-lapse data

We first test the approach on a synthetic zero-offset time-lapse dataset. We use the SEAM time-lapse model, given in Figure 2a to synthesize the base data. A 4D change, shown in Figure 2b, is added to simulate the monitor data. In the shallow part of the model, we include random Gaussian noise as near-surface variations. We generate 60 shots spaced by 17.5 m and only consider the zero-offset data. Figure 3a shows the zero-offset data from the baseline. The target 4D signal (Figure 3b) is obtained by subtracting the baseline and the monitor data before adding the random near-surface seasonal variations. Figure 3c depicts the difference after including the overburden changes. We can see that the 4D noise distorts some of the weak 4D signal as indicated by the arrows in Figures 3(b,c).

The TCN network contains five causal dilated convolutional layers starting with a kernel size of 2. We trained the global network on a time window from 0.1 s to 2 s as no reservoir signal is recorded in this range. To generate training samples, we divided the training window into overlapping sub-windows 100 time-samples in length, which is about 0.3 s. We used the last two sub-windows as a validation set as this ensures that the network learns to match the data even for subsequent samples. After that, we used the global network as an initial model for training the local networks for each trace. We increased the overlap between the traces to increase the number of samples. We tested different hyper-parameters and chose the one that resulted in the smallest loss.

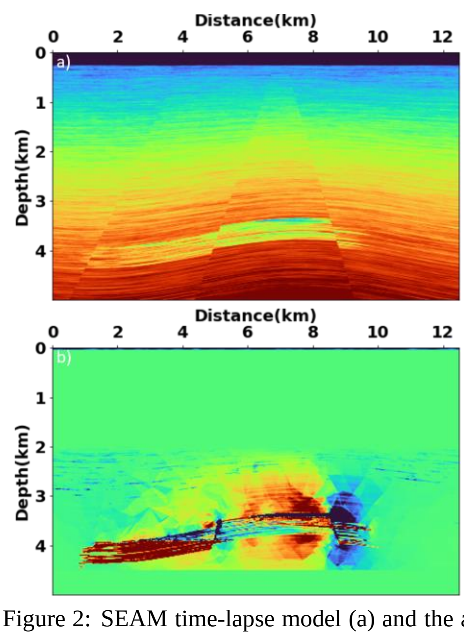

Figure 2: SEAM time-lapse model (a) and the added 4D signal(b).

Finally, we apply the network to the monitor data, including the reservoir. We show the difference between the predicted base and the actual base in Figure 3. Most of the 4D noise was suppressed and we successfully recover some of the weak 4D signal as indicated by the arrows. We compute the normalized root mean square as a repeatability measure (Kragh and Christie, 2002) and found that it decreased from about 15% to 6%.

Real post-stack data

We also tested the method on a 2D line from a 4D post-stack dataset recorded on land. Although high repeatability was achieved between surveys acquired during the same season, significant increases in 4D noise remained between data collected under different climatic conditions (Smith et al., 2019). Two baseline surveys (acquired in different seasons) were used for this test, where we do not expect any 4D signal. Figure 4(a) shows a section through the first baseline survey, with the difference between the two surveys (amplified by a factor of five) plotted in Figure 4(b). We can see that the 4D noise is stronger in the deeper part of the section. This noise is not stationary with time, which means that training the network on a window in the overburden may not be sufficient to predict the noise in the reservoir.

Here, we used a larger receptive field than the first experiment to capture a longer portion of the input traces. This was achieved by increasing the number of dilated convolutional layers to seven. We also used weight normalization and skip-connections, which increase the robustness of the TCN (Bai et al., 2018). We trained the model on a window of 700 time-samples, which is about 1.5 s, with overlapping sub-windows of size 200 time-samples.

The result of applying the network is shown in Figure 4(c). We can see that the network manages to equalize the data well in the training window but failed to correct the deeper part. This could be because the noise is not stationary and behaves differently in the deeper part. Another possibility for this behavior is that we fall into an over-fitting issue. This is an area for further investigation and research.

Conclusion

We have demonstrated matching 4D seismic data using a TCN network instead of conventional matching filters to reduce 4D noise. The method was tested on zero-offset data obtained from the SEAM time-lapse model, which showed the network could successfully match the data in the overburden region and enhanced the 4D signal. We then applied the method on real 4D post-stack data. Although the method suppresses the overburden 4D noise, it did not perform well in the deeper reservoir region. We think this is either because the noise is not stationary with time or possibly due to network over-fitting, which is the subject of further research.

10.1190/segam2021-3583439.1

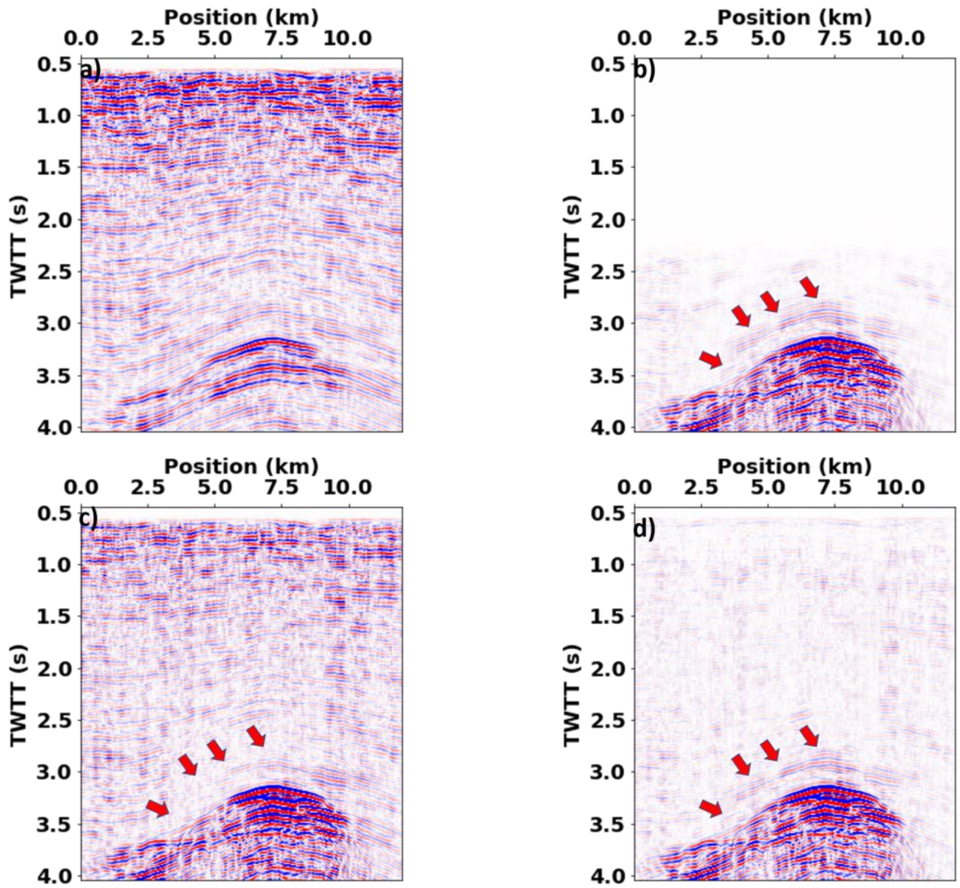

Figure 3: Zero-offset data for the SEAM model (a), the target 4D signal (b), the difference between the base and monitor before processing (c), and the difference after processing with the proposed method (d). The red arrows highlight the improvements in the4D signal. (b), (c) and (d) are plotted at the same scale.

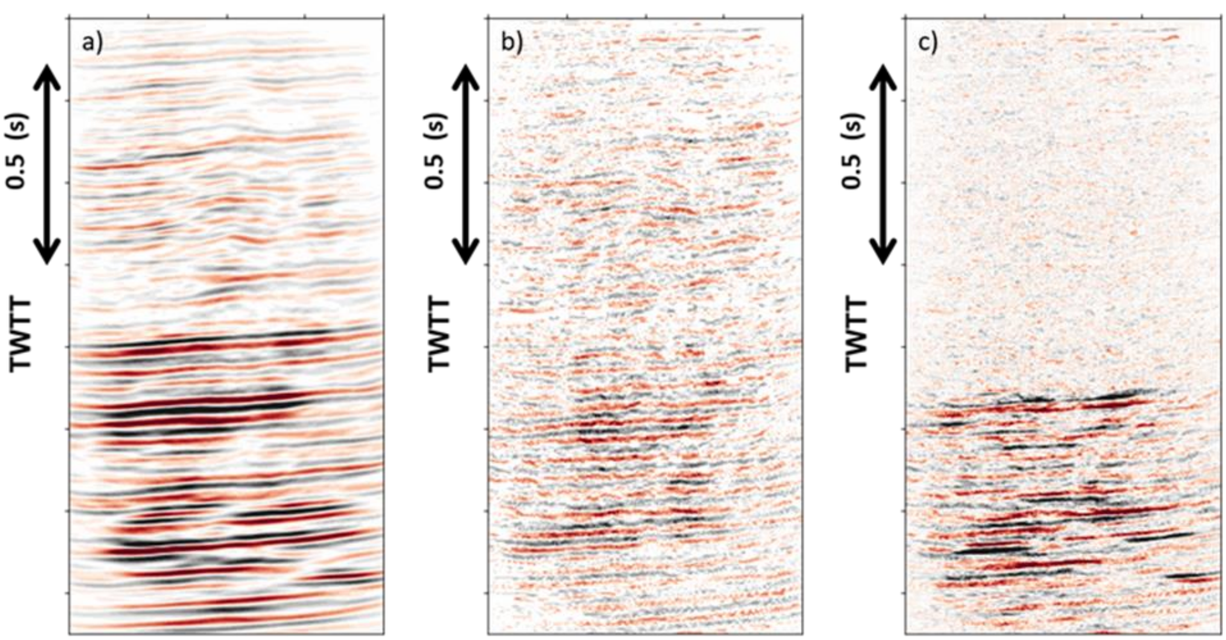

Figure 4: Baseline data (a), the difference between the base and the monitor (b), and the difference after equalizing with the proposed method (c). All the figures are plotted at the same scale but (b) and (c) are amplified by a factor of 5.

REFERENCES

Alali, A., V. Kazei, B. Altaf, X. Zhang, and T. Alkhalifah, 2020a, Time-lapse cross-equalization by deep learning: 82nd EAGE Conference and Exhibition 2020, 1720, doi: https://doi.org/10.3997/2214-4609.202011720.

Alali, A., V. Kazei, B. Sun, R. Smith, P. Nivlet, A. Bakulin, and T. Alkalifah, 2020b, Cross-equalization of time-lapse seismic data using recurrent neural networks: SEG Technical Program, Expanded Abstracts, 1506–1510, doi: https://doi.org/10.1190/segam2020-3424773.1.

Bai, S., J. Z. Kolter, and V. Koltun, 2018, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling: arXiv preprint arXiv:1803.01271.

Bakulin, A., R. Smith, and M. Jervis, 2018, Permanent buried receiver monitoring of a carbonate reservoir in a desert environment: 80th EAGE Conference and Exhibition 2018, 1–5, doi: https://doi.org/10.3997/2214-4609.201801471.

Duan, Y., S. Yuan, P. Hatchell, J. Vila, and K. Wang, 2020, Estimation of time-lapse timeshifts using machine learning: SEG Technical Program, Expanded Abstracts, 3724–3728, doi: https://doi.org/10.1190/segam2020-3423186.1.

Jervis, M., A. Bakulin, and R. Smith, 2018, Making time-lapse seismic work in a complex desert environment for CO2 EOR monitoring—Design and acquisition: The Leading Edge, **37**, 598–606, doi: https://doi.org/10.1190/tle37080598.1.

Kragh, E., and P. Christie, 2002, Seismic repeatability, normalized RMS, and predictability: The Leading Edge, **21**, 640–647, doi: https://doi.org/10.1190/1.1497316.

Lumley, D., D. C. Adams, M. Meadows, S. Cole, and R. Wright, 2003, 4d seismic data processing issues and examples: SEG Technical Program, Expanded Abstracts, 1394–1397, doi: https://doi.org/10.1190/1.1817550.

Mustafa, A., M. Alfarraj, and G. AlRegib, 2019, Estimation of acoustic impedance from seismic data using temporal convolutional network: SEG Technical Program, Expanded Abstracts, 2554–2558, doi: https://doi.org/10.1190/segam2019-3216840.1.

Nivlet, P., R. Smith, and N. AlBinHassan, 2020, Automated well-to-seismic tie using deep neural networks: SEG Technical Program, Expanded Abstracts, 2156–2160, doi: https://doi.org/10.1190/segam2020-3422495.1.

Oord, A. V. D., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, 2016, Wavenet: A generative model for raw audio: arXiv preprint arXiv:1609.03499.

Ovcharenko, O., V. Kazei, M. Kalita, D. Peter, and T. Alkhalifah, 2019, Deep learning for low-frequency extrapolation from multi-offset seismic data: Geophysics, **84**, no. 6, R989– R1001, doi: https://doi.org/10.1190/geo2018-0884.1.

Rickett, J., and D. Lumley, 2001, Cross-equalization data processing for time-lapse seismic reservoir monitoring: A case study from the Gulf of Mexico: Geophysics, **66**, 1015–1025, doi: https://doi.org/10.1190/1.1487049.

Ross, C., G. Cunningham, and D. Weber, 1996, Inside the cross-equalization black box: The Leading Edge, **15**, 1233–1240, doi: https://doi.org/10.1190/1.1437231.

Shulakova, V., R. Pevzner, J. C. Dupuis, M. Urosevic, K. Tertyshnikov, D. E. Lumley, and B. Gurevich, 2014, Burying receivers for improved time-lapse seismic repeatability: Co2crc Otway field experiment: Geophysical Prospecting, **63**, 55–69, doi: https://doi.org/10.1111/1365-2478.12174.

Slang, S., J. Sun, T. Elboth, S. McDonald, and L. Gelius, 2019, Using convolutional neural networks for denoising and deblending of marine seismic data: 81st EAGE Conference and Exhibition 2019, 1–5, doi: https://doi.org/10.3997/2214-4609.201900844.

Smith, R., E. Hemyari, A. Bakulin, and A. Alramadhan, 2019, Making seismic monitoring work in a complex desert environment—4D processing: The Leading Edge **38**, 637–645, doi: https://doi.org/10.1190/tle38080637.1.

10.1190/segam2021-3583439.1