St. Petersburg State Univeristy

Manuscript copyright

Pankov Vikentii Dmitrievich

ESTIMATION OF SPARSE MODELS PARAMETERS

1.2.3. Theoretical informatics, cybernetics

Thesis for the degree of Candidate of Physical and Mathematical Sciences

Translation from Russian

Scientific supervisor:
Doctor of Physical and Mathematical Sciences, Professor
Oleg Nikolayevich Granichin

St. Petersburg
2024

Contents

Introduction **4**

Chapter 1. Problem Statement and Method Overview **11**

1.0.1 Inverse Problems in Mathematical Modeling 11

1.0.2 Compressed Sensing Methodology 16

1.0.3 Overview of Sparse Signal Reconstruction Methods from Compressed Observations 17

1.0.4 Conclusions 22

Chapter 2. Randomized Stochastic Approximation Algorithm for Geological Model Parameter Adaptation to Development History **24**

2.0.1 Mathematical Representation and Parameterization of Geological Models 26

2.0.2 SPSA Algorithm and Theorem on the Properties of Estimates in the Context of the Given Problem 34

2.0.3 Conclusions 40

Chapter 3. Compression-Based Recognition Methods in Distributed Clustering and Ultrasonic Tomography Tasks **41**

3.0.1 Distributed Clustering in Multi-Agent Systems Based on Compression-Based Recognition Methodology . . . 41

3.0.2 Compression and Reconstruction of Ultrasonic Tomography Data 44

3.0.3 Full-Waveform Inversion with Compressed Measurements 50

3.0.4 Discussion 55

Chapter 4. Software Simulation and Experimental Research Results **56**

4.0.1 Geological Model Parameter Adaptation System to Development History 56

4.0.2 Simulation of the Distributed Clustering Method . . . 64

4.0.3 Experiments on Full-Waveform Inversion and Compression of Ultrasonic Tomography Data 66

4.0.4 The ℓ_1 Control Problem for Discrete Non-Minimum Phase Systems with Unknown Bounded Disturbances . 70

4.0.5 A Voice Cloning System with Noise-Robust Speech Synthesis Based on Latent Space Regularization 73

4.0.6 Discussion 77

Conclusion **78**

Bibliography **79**

Introduction

Inverse problems in mathematical modeling play a crucial role in numerous scientific and engineering applications, including medical tomography and geological modeling. These problems involve reconstructing hidden model parameters based on observed data, which are often incomplete or noisy.

In many cases, model parameters are highly sparse, meaning that the solution depends on only a small number of parameters. Sparsity is a key concept that unifies various practical approaches in this study. It enables high-precision solutions, reduces dimensionality and computational costs, enhances solution robustness to noise, and promotes the creation of more interpretable and manageable models.

ℓ_1-regularization is a key tool in solving inverse problems with sparse structures. Research conducted at Saint Petersburg State University has been instrumental in advancing methods for ℓ_1 regularization. For example, work by V.F. Demyanov and V.N. Malozemov in the 1960s-1970s on minimax methods significantly influenced the development of ℓ_1 regularization theory in control problems. Later, the compressed sensing methodology introduced by E. Candes, J. Romberg, and T. Tao in 2006 [17] gained wide recognition. This approach allows the accurate reconstruction of sparse signals using a small number of measurements, well below the Nyquist limit.

Most recent research on sparse models has primarily focused on reconstructing the original signal from compressed observations using generative deep learning methods [20, 21, 51, 53, 54, 65] or iterative learning-based methods [5, 68, 76, 78]. However, often the original signal is not used directly but rather serves as an initial step for solving another modeling problem. In such cases, it is advantageous to avoid direct reconstruction of the complete signal, solving the desired problem in the compressed domain and obtaining the full (target) model output. This significantly reduces the complexity of the problem, eliminating the need for storing and processing large volumes of data. Thus, an urgent task is to investigate methods that allow solving sparse inverse modeling problems directly in the compressed space.

Goal of the study — the development of mathematical methods for solving inverse problems with sparse structures directly in compressed latent spaces, without reconstructing the full signal, using stochastic optimization algorithms and deep learning methods. To achieve this goal, the following tasks have been formulated and addressed:

- Develop a method for adapting sparse geological model parameters based on dynamic observational data, utilizing model parameterization to obtain a compressed representation and applying a stochastic optimization algorithm to find the solution in the compressed space.
- Implement a distributed clustering algorithm for multi-agent systems using compressed sensing methodology to predict cluster parameters based on compressed data; develop an efficient method for data collection and storage from a three-dimensional ultrasound tomograph, as well as a hybrid full-wave inversion method for direct image reconstruction based on compressed ultrasound data.
- Develop a noise removal method for robust speech synthesis using regularization in the compressed voice representation space.
- Investigate the effectiveness of sparse controller synthesis using ℓ_1 optimization methods for controlling non-minimum phase systems under unknown bounded disturbances.
- Conduct software modeling and experimentally evaluate the performance of the proposed methods.

Research methods. This dissertation utilizes optimization methods, deep learning, compressed sensing theory, and simulation modeling.

Scientific novelty. All primary scientific results of the dissertation are original.

Theoretical value and practical significance. The theoretical value of this work lies in the development of stochastic optimization methods and compressed sensing for solving inverse problems with sparse structures directly in compressed or latent spaces without reconstructing the full signal. The practical significance of the study is reflected in the potential application of the developed methods in real industrial and medical settings, greatly enhancing the efficiency of respective modeling, data collection, and processing systems.

Validation of the work. The results of this research were presented at the Department of System Programming, Faculty of Mathematics and Mechanics, SPbU, at the 14th All-Russian Conference on Control Problems (Moscow, Russia, June 17-20, 2024), and at the InterSpeech 2024 Conference (Kos Island, Greece, September 1-5, 2024).

The findings were used in studies supported by the RSF Grant 21-19-00516 "Multi-agent adaptive control in networked dynamic systems with application to groups of robotic devices under uncertainties."

Publication of results. The primary results are published in the works [6, 25, 30, 31, 61]. The author has published five scientific papers, three of which are in journals indexed in Scopus, and two in conference proceedings indexed in Scopus. All works are co-authored. In [31], V.D. Pankov contributed the algorithm for determining the angle of reflection and intersection point of the ray with the object under reconstruction, while the co-authors developed the rest of the method for identifying special regions based on ultrasound tomography data and the general problem statement. In [61], V.D. Pankov contributed to the method development and simulation modeling, while the co-authors provided the general problem statement. In [6], V.D. Pankov contributed to method development and simulation modeling in Sections 3.1-3.2 (distributed reconstruction of cluster structure from compressed observations), with the co-authors providing the general problem statement, methods development, and simulation modeling for other parts of the system. In [30], V.D. Pankov contributed part of the simulation modeling, with the co-authors responsible for the general problem statement, theorem formulation and proof, and methods development. In [25], V.D. Pankov contributed to the general problem statement, formulation, and development of noise regularization methods, with the co-authors developing the training method for noisy data.

Structure and volume of the dissertation. The dissertation consists of an introduction, four chapters, a conclusion, and a bibliography containing 84 sources. The text comprises 87 pages, includes 17 figures, and 4 tables.

Summary of the dissertation content.

In the **Introduction**, the relevance of the study, the goals and objectives, as well as the main results are outlined.

In **Chapter 1**, the general formulation of inverse problems in mathematical modeling is presented, along with an overview of primary approaches in the compressed sensing methodology for solving problems with sparse structures.

Chapter 2 describes the task of reconstructing sparse geological models based on dynamic observational data. To reduce the dimensionality of the problem, parameterization methods based on deep learning are applied. The primary focus is on the Stochastic Approximation (SPSA) algorithm, adapted for the task of geological model adaptation. A theorem on the upper bound of mean-square error for the considered problem is provided.

Chapter 3 outlines methods based on compressed sensing theory, enabling the solution of various inverse modeling problems directly in the compressed space. The first method, based on deep learning, allows each agent to determine its cluster parameters based on local interactions with other agents. The proposed algorithm uses compressed measurements for information exchange and is capable of operating in real-time, reducing computational and communication costs. In the context of ultrasound computed tomography, a method for gathering and processing large volumes of data required for object image reconstruction is discussed. Additionally, a method for data compression and reconstruction is proposed, allowing a substantial reduction in the volume of information transmitted from tomography sensors. Finally, an adapted full-wave inversion method using compressed data is presented, allowing direct image construction based on the compressed signal representation without reconstructing the full data volume. A theorem estimating image reconstruction accuracy when using compressed measurements is provided.

Chapter 4 presents the results of software modeling and experiments, confirming the effectiveness of the proposed methods. Experiments demonstrate that using stochastic algorithms for geological model adaptation in the compressed space, as well as compressed sensing theory approaches for clustering and ultrasound tomography data recovery, allows significant reduction in computational costs and data processing volumes while improving noise robustness and maintaining high solution accuracy. Additionally, the effectiveness of the ℓ_1-synthesis method for sparse controllers for discrete systems with arbitrary bounded disturbances is analyzed. Furthermore, the high noise robustness of the proposed speech synthesis method based on regularization in the compressed space is experimentally demonstrated, preserving high-quality voice cloning.

In the **Conclusion**, the main research results are summarized.

Findings and arguments of the dissertation to be defended

- A method for adapting sparse geological model parameters to reservoir development history, based on the Stochastic Approximation (SPSA) algorithm combined with a neural network-based dimensionality reduction algorithm.

- A method for predicting clusters from compressed observations for multi-agent systems with a clustered state space structure.

- A method for data reconstruction and image recovery from compressed measurements in ultrasound tomography tasks.

- A noise-robust speech synthesis method based on voice representation regularization in the compressed space.
- The specifics of software implementations and numerical demonstration of the proposed methods' effectiveness.

Main scientific results

1. A method for adapting sparse geological model parameters to reservoir development history, based on the Stochastic Approximation (SPSA) algorithm combined with a neural network-based dimensionality reduction algorithm (see [61], with the author's contribution in this result being no less than 80%).

2. A clustering method for multi-agent systems with a clustered state space structure, based on compressed sensing theory and a neural network algorithm for cluster prediction from compressed state representations (see Sections 3.1, 3.2 in [6], with the author's contribution in this result being no less than 80%).

3. A method for object reconstruction in ultrasound computed tomography based on image processing algorithms for determining signal reflection location and angle (see Section 5.2 in [31], with the author's contribution in this result being 100%).

4. Research on ℓ_1 optimization in control problems for discrete systems with non-minimum phase and under unknown bounded disturbances (see [30], analytical computations in Section 4 completed by the author, overall contribution at least 20%).

5. A noise reduction method for robust speech synthesis using regularization in the compressed speaker representation space, employing an

approach that encourages the generation of noise-invariant representations without loss of synthesis quality (see [25], the author's personal contribution to this result is 100

Chapter 1

Problem Statement and Method Overview

1.0.1 Inverse Problems in Mathematical Modeling

In many scientific fields and problems, there is a need to identify hidden properties or parameters of the corresponding model. An example is reconstructing an image from tomography or MRI data or evaluating reservoir properties from seismic observations.

An inverse problem involves finding the vector $x \in X$ of model parameters based on observed data $y \in Y$:

$$y = F(x) + e, \tag{1.1}$$

where X is the parameter space of the model, and Y is the observation space. In most cases, X and Y are Banach or Hilbert spaces. $F : X \to Y$ is a continuous operator that defines how observed data y is generated from the model's internal parameters x in the absence of noise.

The problem can be formulated as obtaining information about x based on available data y and knowledge provided by the operator F.

The solution methods depend on the properties of the operator F. The most important of these are:

- Injectivity of the operator, which determines the sufficiency of data y for uniquely reconstructing x.
- Stability of the operator: $||x_1 - x_2|| \leq \omega(||F(x_1) - F(x_2)||)$, for a given function $\omega : \mathbb{R}^+ \to \mathbb{R}^+$ and all $x_1, x_2 \in X$. This defines the extent to which model parameters change with slight modifications in observations (e.g., due to noise).

Well-posed problems, according to Hadamard (1902, 1923), should satisfy these properties. If a solution can be uniquely found from observations and does not drastically change with minor disturbances in observations due to noise, the problem is considered well-posed.

When noise levels are low, and observations are sufficient, the problem reduces to the inversion of the operator F, which has a well-studied theory, especially for linear operators.

Most often, inverse problems are ill-posed. It is not always possible to gather enough measurements, and/or they may be affected by noise with an unknown distribution. Solving these problems requires some a priori knowledge. This chapter reviews regularization theory, which restricts the parameter space, for example, by assuming that they are sufficiently smooth. Separately, we consider an approach based on finding sparse solutions, used as the basis for methods in the following chapters.

Next, we present examples of various inverse problems [67].

Compressive Sensing Problem

The compressive sensing problem involves data collection at a sampling rate lower than required by the Nyquist-Shannon theorem. The theorem states that a signal with a limited spectrum can be reconstructed without loss from measurements taken at a rate higher than twice the maximum frequency of the spectrum. Compressive sensing methodology allows for lossless signal reconstruction with fewer observations if it is sufficiently "sparse" (i.e., has a small number of non-zero elements in a certain basis).

The forward operator is the measurement matrix Φ with specific properties. The most critical of these is the Restricted Isometric Property (RIP), which guarantees that the signal is not distorted when reducing the dimensionality from N observations to m, provided it is sufficiently sparse. A matrix Φ satisfies RIP if [58]:

$$\sqrt{1-\delta} \leq \frac{||\Phi z||_2}{||z||_2} \leq \sqrt{1+\delta},$$

for some δ and s-sparse vectors z.

The problem is formulated as follows:

$$y = A\Psi x = \Phi x + e,$$

where $y \in \mathbb{R}^m$ represents compressed measurements, $x \in \mathbb{R}^n$ is the original signal, e is additive noise, $\Phi \in \mathbb{R}^{m \times n}$ is the measurement matrix, and Ψ is the basis matrix in which the signal is sparse.

Super-resolution

The super-resolution problem aims to enhance data quality by increasing the sampling frequency. For example, it may involve increasing the resolution of an image by a given factor or increasing the sampling rate of an audio file. The problem can be formulated as follows:

$$y = \Theta x + e = DBx + e,$$

where $y \in \mathbb{R}^m$ are low-resolution observed data, $x \in \mathbb{R}^n$ are the high-resolution desired data, and e is additive noise. $D \in \mathbb{R}^{m \times n}$ is the down-sampling matrix, and $B \in \mathbb{R}^{n \times n}$ is the degradation matrix (e.g., blurring).

Noise Removal

The noise removal problem can be formulated as follows.

$$y = x + e.$$

Here, the operator F is the identity: $F = I$. The observed data are the noisy desired parameter vector x.

In the simplest case, the noise distribution is known, allowing the parameter vector to be determined by statistical methods. However, in real-world problems, this is often not the case, and only bounded noise is assumed.

Image Sharpening

$$y = K * x + e.$$

In this case, the forward operator is a convolution with a blurring mask K, and the observed data are the convolution of the desired parameter vector x in the presence of noise.

In the absence of noise and with an available convolution mask K, this problem can be solved by transforming to Fourier space and performing deconvolution.

Inpainting (Filling in Missing Data)

An example of an inpainting problem is completing missing parts of an image, such as parts that were lost or not measured.

In this problem, the operator is a restriction of the vector x to a certain subset Ω.

$$y = x|_{\Omega} + e.$$

Computed Tomography

The simplest model of computed tomography (CT) assumes a monoenergetic beam of X-rays without scattering effects. Under these conditions, the desired parameter is a function from the spatial domain $\Omega \subset \mathbb{R}^3$ to the set of real numbers $\mathbb{R}$.

According to the Beer-Lambert law, the forward operator is defined as follows:

$$A = e^{-\nu \int_{-\infty}^{\infty} f(x+sw)ds},$$

where $w \in S^{d-1}$, and $x \in w^{\perp}$ defines the line $x + sw$ along which the X-rays propagate. To linearize the problem, the logarithm of the data is often taken [67].

Regularization Methods

Regularization refers to the addition of certain constraints on the desired vector of model parameters. The purpose of regularization is to solve an ill-posed problem and to demonstrate that the obtained solution is stable with respect to noise and close to the true solution or converges to it.

Analytical Operator Inversion. These methods are based on finding an analytical expression for a stable inversion of the forward operator. Their effectiveness and the approach used depend heavily on the specifics of the problem. For example, in CT image reconstruction, the Radon transform is a standard inversion algorithm frequently applied in practice.

Variational Methods are based on minimizing an objective function that includes an additional term that defines the desired properties of the solution.

$$\mathcal{R}_{\theta} : \arg\min_{x \in X}(\mathcal{L}(F(x), y) + S_{\theta}(x)). \tag{1.2}$$

This is a general method where the regularizing term $S_{\theta}(x)$ and the functional $L : Y \times Y \to \mathbb{R}$ are chosen depending on the problem.

Formally, the regularization functional $\mathcal{S}$ can be defined as a mapping: $\mathcal{S} : x \to \mathbb{R}_{+}$, which maps the solution to a positive number characterizing how well x satisfies the desired properties (it is generally assumed that smaller values correspond to "better" solutions).

The type and interpretation of the functionals L and S depend on the chosen approach to solving the problem. For example, in the Bayesian approach, S can be viewed as the negative logarithm of the prior probability distribution, and L as the negative logarithm of the likelihood function. In this case, minimization aims to find a solution that corresponds to the maximum likelihood and accounts for the prior distribution of the observation y.

The reconstruction operator (here and throughout) is denoted as $\mathcal{R}_\theta : Y \to X$, and its result is a point estimate $\hat{x}$ of the solution to the inverse problem. θ represents the vector of parameters involved in the functionals L and S.

A well-known regularization method of the form (1.2) is Tikhonov regularization, with the operator $\mathcal{L}$ in the form of an ℓ_2 norm and a regularizer of the form $\theta S(x), \theta \in \mathbb{R}$. For $S(x) = \frac{1}{2}||x||^2$ and a linear operator F, the inverse problem solution is represented as $\mathcal{R}_\lambda = (F^* \circ F + \lambda id)^{-1} \circ F^*$, where F^* is the adjoint operator and id the identity operator.

1.0.2 Compressed Sensing Methodology

Sparsity is a key concept in compressed sensing. This characteristic allows for efficient data compression and subsequent recovery with minimal information loss. Compressed sensing (CS) methodology consists of two parts: first, obtaining compressed measurements of the signal, and second, reconstructing the original signal from the compressed observations.

In the data acquisition stage of CS methodology, there is a measurement matrix Φ with certain properties. The most important of these is the Restricted Isometric Property (RIP), which guarantees that the signal is not distorted when reducing the dimensionality from N observations to m, provided the signal is sufficiently sparse. A matrix Φ satisfies RIP if [58]:

$$\sqrt{1-\delta} \leq \frac{||\Phi z||_2}{||z||_2} \leq \sqrt{1+\delta},$$

for some δ and s-sparse vectors z.

The problem is formulated as follows:

$$y = A\Psi x = \Phi x + e,$$

where $y \in \mathbb{R}^m$ are compressed measurements, $x \in \mathbb{R}^n$ is the original signal, e is additive noise, $\Phi \in \mathbb{R}^{m \times n}$ is the measurement matrix, and Ψ is the basis matrix in which the signal is sparse.

In the tasks considered in the following chapters, the measurement matrix is a random Gaussian matrix, which satisfies the required properties with a high probability.

Signal reconstruction involves minimizing the ℓ_0 norm, which allows finding the most appropriate (sparse) solution among an infinite number of possibilities. Direct minimization of the ℓ_0 norm is an NP-hard problem. However, there are numerous approximate numerical methods for solving this problem. We will examine the main ones below.

1.0.3 Overview of Sparse Signal Reconstruction Methods from Compressed Observations

Convex Optimization

The ℓ_0 optimization problem is often replaced with a convex optimization problem (typically ℓ_1 optimization). Examples of algorithms based on this approach include Basis Pursuit [11] and Basis Pursuit De-Noising (BPDN). These methods solve the following optimization problem:

$$\min_x \left(\frac{1}{2} ||y - \Phi x||_2 + ||x||_1 \right). \tag{1.3}$$

There are numerous other algorithms, including the Least Absolute Shrinkage and Selection Operator (LASSO), the Iterative Shrinkage/Thresholding Algorithm (ISTA) [12], the Alternating Direction Method of Multipliers

(ADMM), the Gradient Projection for Sparse Representation [37], and Total Variation Denoising.

If the measurement matrix satisfies the RIP property, the following theorem (Candès, Romberg, and Tao 2006) [18] guarantees that if:

$$y = \Phi x + e,$$

with a noise level $||e|| \leq \delta$, then the solution:

$$\hat{x} = \arg\min ||x||_1 \quad \text{subject to} \quad ||\Phi x - y||_2 \leq \delta \tag{1.4}$$

satisfies the following inequality:

$$||\hat{x} - x||_2 \leq C \left(\delta + \frac{||x - x_s||_2}{\sqrt{s}} \right), \tag{1.5}$$

where x_s is the vector consisting of the s largest (in absolute value) components of the true solution x (with zeros elsewhere).

Greedy Algorithms

The greedy approach involves iterative optimization on an element-by-element basis, with a locally optimal choice at each step. The solution is constructed iteratively by sequentially adding non-zero components and solving a least squares optimization problem with constraints at each iteration. Widely used algorithms in this class include Matching Pursuit (MP) and Orthogonal Matching Pursuit (OMP) [52]. OMP starts the reconstruction process by identifying the column of the measurement matrix most correlated with the observations y in the first step, then finds the column with the maximum correlation to the current residual at each iteration. The signal vector estimate is updated on each iteration, accounting for the selected column. There are several enhancements to this algorithm, such as Compressive Sampling Matching Pursuit [56], Stagewise Orthogonal Matching Pursuit (StOMP) [69], and Generalized Orthogonal Matching Pursuit [74].

Thresholding algorithms, such as Iterative Hard Thresholding (IHT) [13] and Iterative Soft Thresholding [15], are also often used. These algorithms alternate optimization steps (gradient descent) with thresholding functions to ensure sparsity constraints.

Greedy algorithms tend to be more computationally efficient than convex optimization methods. However, they may not provide satisfactory reconstruction quality in the presence of noise or insufficient measurements.

Iterative Learning-Based Methods

Let us consider two approaches to solving the reconstruction problem using deep learning. The first is based on classical iterative optimization algorithms for CS, where each iteration's operations are replaced or supplemented by neural networks. Network parameters and classical algorithm hyperparameters are tuned based on training data, allowing the algorithm to adapt effectively to specific tasks and enhancing signal reconstruction quality [50].

ISTA [12] is a popular iterative algorithm used for reconstruction. The ISTA algorithm and its extensions rely on a gradient-based approach, where each iteration projects the gradient and then thresholds (zeros out) the solution components based on a threshold value. An adaptive variant is LISTA (Learned ISTA) [68], which has an autoencoder architecture designed to find a sparse representation of the signal in a given dictionary. Another adaptation, ISTA-Net [76], directly solves the reconstruction problem compared to LISTA, which tackles a sparse coding auxiliary task. TISTA further improves LISTA by using error variance estimation to enhance convergence speed.

Approximate Message Passing (AMP) is another iterative algorithm based on message passing on graphs. Its adaptive counterparts include Learned AMP (LAMP) and LDAMP.

The ADMM algorithm, used for CS reconstruction, divides the optimization problem into smaller parts, each of which is straightforward to solve. For MRI reconstruction from CS measurements, an adaptive version called

ADMM-NET [28], based on ADMM and convolutional neural networks, has been proposed. ADMM-CSNet [5] represents a further improvement.

An analog of the Total Variation (TV) algorithm is TVINet [78], which addresses regularization by combining the iterative TV method with a convolutional neural network.

The main advantage of replacing parts of iterative algorithms with learning-based counterparts is the significant reduction in computational requirements. For example, after training, LISTA is 20 times faster than ISTA, and LDAMP is 10 times faster than BM3D-AMP. Moreover, most learning-based variants automatically tune hyperparameters, eliminating the need for fine-tuning to ensure fast convergence and satisfactory reconstruction quality. As a result, adaptive algorithms significantly improve reconstruction quality compared to their classical iterative versions.

Direct Reconstruction Methods Based on Learned Models

The second approach involves direct reconstruction from compressed measurements using a trained model. Unlike the first approach, this method is not bound to traditional iterative algorithms. Additionally, it allows for modifying the sampling (compression) process of the data; for instance, it can implement a measurement matrix selection algorithm based on data. Numerous direct reconstruction methods exist, often categorized by the neural network architectures underpinning these methods.

Autoencoders are widely applied in CS because they consist of two parts: an encoder, which maps input data to a lower-dimensional latent space, and a decoder, which reconstructs the original data from the encoded representation. Naturally, the encoder can be seen as implementing the CS sampling process, while the decoder serves as the reconstruction algorithm. In [51], a stacked denoising autoencoder (SDAE) based on a fully convolutional neural network was proposed. Its drawback is the increasing number of parameters as the signal dimension grows, leading to high resource consumption and the risk of overfitting for large signals. To address this issue, [**?**, 65] proposed dividing the signal into blocks, each of which is independently compressed and reconstructed. To reduce the risk of overfitting, [83] proposed the sparse autoencoder compressed sensing (SAECS) method, with added regularization that zeros out part of the parameters. Furthermore, [80] introduces a method based on SDAE and SAECS, robust to noise.

Fully connected and convolutional neural networks are also often applied for implementing reconstruction algorithms. ReconNet [65] is one of the earliest works utilizing convolutional networks in compressive sensing. In addition to CNN, ReconNet uses a fully connected layer to map the compressed signal to the original dimension, showing improved reconstruction quality over traditional algorithms. CombNet [20] achieved further reconstruction quality improvement with a deeper architecture and smaller convolution sizes, reducing the likelihood of overfitting.

A notable work is DeepInverse [53], which is also based on CNN. Instead of a fully connected layer, a conjugate operator (the transposed measurement matrix) is used for the initial reconstruction. This significantly reduces the model parameters, allowing for efficient recovery of high-dimensional signals without splitting them into blocks. A comparison with the D-AMP algorithm showed an increase in reconstruction speed by several orders of magnitude and improved quality at compression ratios below 5-7%. At higher compression ratios, D-AMP demonstrated better reconstruction quality. Nevertheless, DeepInverse has potential for further improvements due to its lightweight model and block-free application of compressive sensing. Based on DeepInverse, DeepCodec [54] was proposed, which replaces the random measurement matrix with a sampling algorithm based on training data. An experiment comparing this algorithm to DeepInverse and LASSO with varying compression and sparsity levels demonstrated substantial improvements in model parameter count, reconstruction quality, and computational complexity. ConvCSNet [21] also implements a learning-based measurement acquisition algorithm. Unlike DeepInverse, ConvCSNet uses a linear layer for compressed measurements instead of a convolutional layer. It also includes a reconstruction algorithm composed of two branches of convolutional layers. ConvCSNet is implemented with the option of applying it to full-size signals (without splitting into blocks).

Residual neural networks (ResNets) are primarily designed to address the vanishing gradient problem when a large number of layers are present. ResNet introduces shortcut connections between layers, such as summing a layer's output with its input, helping the network find the shortest paths during training. In [26], a Deep Residual Reconstruction Network model with a fully connected layer for adaptive sampling and several fully connected layers followed by ResNet blocks for reconstruction is presented. A similar method is proposed in [39], with the primary difference being the use of a convolutional network for sampling instead of a fully connected layer. In [45], the sampling algorithm is designed to learn measurement matrices suitable for physical implementation (e.g., binary and bipolar matrices with elements -1 and +1). Experiments showed comparable reconstruction accuracy with other deep learning methods when using such matrices.

Generative adversarial networks (GANs) are also widely used in solving inverse problems. In [22], a modification of ReconNet with an added discriminator was proposed, where ReconNet is implemented as the generator. The discriminator is trained to differentiate between real and ReconNet-generated data, used in the generator's loss function during training. Experiments showed higher reconstruction quality compared to the original method. GANs are also used in MRI data reconstruction [29], where the generator is based on a ResNet architecture, and the discriminator on a CNN. Simultaneously training two neural networks (generator and discriminator) is challenging, requiring careful tuning of model parameters and architecture to ensure training stability and convergence. This issue is partially addressed in [24], where the generator is only used to construct missing image details, which reduced network complexity and led to faster convergence.

1.0.4 Conclusions

This section provides an overview of inverse problems in mathematical modeling, including the compressive sensing problem, which forms the basis of the methods presented in Chapter 3. A significant characteristic of many problems is that they are often ill-posed, necessitating regularization methods to stabilize solutions. Solutions to inverse problems rely on the properties of the operator that describes the relationship between model parameters and observed data, as well as prior knowledge about the desired solutions.

Convex optimization methods are effective tools for solving compressive sensing and signal reconstruction problems. However, despite their theoretical justification, traditional iterative methods often require significant computational resources and may not provide satisfactory quality with limited data or high noise levels. The application of deep learning methods has opened new possibilities for improving the efficiency of inverse problem-solving. Neural networks enable significant reductions in computation time and improved data reconstruction quality. Adaptive versions of classical algorithms, such as LISTA and ADMM-NET, as well as approaches based on autoencoders and convolutional neural networks, demonstrate high efficiency in solving reconstruction problems from compressed observations.

However, for each specific problem, careful model tuning, selection of appropriate neural network architectures, and mathematical methods are required. In the following chapters, we will examine various methods for solving inverse problems, with a primary focus on leveraging sparsity properties and developing methods to solve them within compressed latent spaces.

Chapter 2

Randomized Stochastic Approximation Algorithm for Geological Model Parameter Adaptation to Development History

Geological models are used in many tasks related to mineral resource development. They enable the assessment of the cost and potential success of activities such as seismic surveys, the placement of new wells, and forecasting mineral extraction. For a geological model to support sound decision-making, it must be highly accurate and incorporate the maximum amount of available data about the deposit.

Data used in constructing a geological model can include such characteristics as the geological concept of the reservoir, measurements at drilling sites, and geophysical measurements. This data is used in the initial stages of geological model creation. Additionally, there are dynamic data, continually received during field operation, such as historical oil and fluid production data. These data allow for continuous model updates. The process of updating the geological model based on dynamic data is called history matching.

History matching is an inverse problem where the model's input parameters are adjusted to minimize the difference between simulated and observed data. The simulated data are obtained by modeling the hydrodynamic behavior of the reservoir through simulation of the geological model with a given set of parameters. The model's input parameters can be various properties of the simulated environment, such as porosity and permeability at each point in the reservoir.

Geological models include a large number of parameters that need to be adjusted during history matching. For example, a model with 60 by 60 grid cells requires 3600 values. Since history matching involves minimizing the deviation of simulated data from observed data, finding an optimal set of parameters becomes highly challenging in high-dimensional space. To address this, parameterization methods for geological models are used, allowing the models to be described by a significantly smaller set of new parameters due to their sparsity. A frequently used parameterization method is PCA [35,60]. Its drawback, however, is its inapplicability to non-Gaussian geological models. To solve this issue, the CNN-PCA method was developed [49], based on PCA and using a neural network to build a parameterizing model.

Various approaches are used to minimize the objective function that defines the deviation between simulated and observed data. Multiple Kalman filters (EnKF) are frequently used [59, 71]. EnKF is an iterative model adaptation method. It involves calculating the covariance between the analyzed model parameters and simulated data, which is then used to update model parameters based on observed data. A disadvantage of this method is the loss of accuracy when adapting binary geological models [72], as well as the requirement to generate a relatively large number of geological realizations. Another widely used method is the Particle Swarm Optimization (PSO) [9,34,49,81], which is based on swarm intelligence principles. It does not require gradient calculation with respect to model parameters, which is often impossible due to the complexity and lack of implementation in many hydrodynamic simulators.

This work uses the Stochastic Optimization Algorithm SPSA [14]. Compared to the aforementioned methods EnKF and PSO, SPSA has the advantages of a minimal number of calls to the objective function for gradient estimation and consistency under nearly arbitrary noise in the observed data [73]. The application of the SPSA algorithm to geological model recovery was previously examined in [40]. Comparisons with L-BFGS and gradient descent methods showed greater efficiency in solving the problem. However, SPSA was applied without prior parameterization, and its study was limited to a simple, small-scale geological model. Additionally, only the case of normal noise in the observed data was considered.

Among existing systems that allow history matching of geological models, Schlumberger Petrel [1] can be noted. The advantages of this system include ease of use and integration with other widely used Schlumberger systems, such as the Eclipse hydrodynamic simulator. The main disadvantages of Schlumberger's tools are the lack of free software versions, closed-source code, and lack of descriptions of the algorithms used. Currently, there are no open, extensible systems for history matching of geological models, making the creation of a system based on modern parameterization methods and open geological tools a relevant problem that can be formulated as an inverse problem with a sparse structure.

This chapter proposes an algorithm for the history matching of sparse geological models based on historical production data, using the gradient-free stochastic optimization method SPSA. A theorem providing an upper bound on the mean square error of the algorithm is formulated.

2.0.1 Mathematical Representation and Parameterization of Geological Models

Geological models can be deterministic or stochastic. For stochastic models, model parameters at each point in the medium are defined by a probability distribution. Stochastic modeling can generate not a single model but a family of different models, each consistent with the available data. This approach allows for the uncertainty's impact on the modeling outcome to be considered [33].

Geostatistical methods are used in stochastic modeling. Sequential indicator simulation, object-based modeling, and multipoint statistics methods are widely applied. Sequential indicator simulation uses variograms (correlations between two points in the medium) to model spatial correlations. In object-based modeling, the geological model is generated by randomly generating objects according to specified rules, such as channel sinuosity and thickness. Multipoint statistics methods, used in this work, generate geological models based on a training image that serves as a conceptual representation of the spatial structure of the modeled reservoir.

Using a training image is one way to set a priori information about the geological features of the studied reservoir. The training image includes essential structures, such as sinuous channels of a specific orientation and thicknesses. An example of a training image is shown in Figure 2.1.

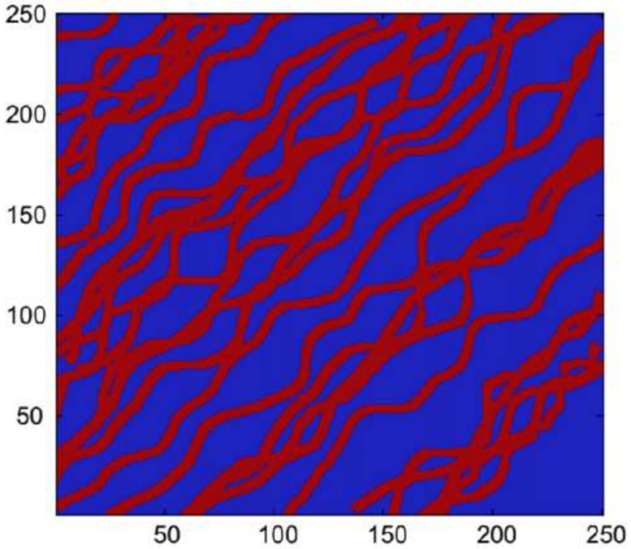

Figure 2.1: An example of a training image from [49]. The image provides prior information on the number, thickness, and orientation of channels at 45°. This training image is used in subsequent experiments.

The geological model is represented as a random field. Based on the training image, realizations are generated that include the features defined by this image and satisfy additional conditions (hard data), such as values at specific points (e.g., including geological properties measurable at drilling sites).

In practice, grid models of specific sizes are used, and realizations are represented as a discrete set of grid blocks.

SNESIM Algorithm for Geological Model Distribution Simulation

The SNESIM algorithm [70] is a widely used method for generating geological realizations based on a training image and additional data. In the first step of the algorithm, the frequencies of various patterns in the training image are calculated. A random path through each block of the generated realization is then constructed. According to this path, blocks are sequentially traversed. For the current block, a pattern is randomly selected from the training image with a probability proportional to the pattern's frequency. It is also checked that the selected pattern for each block matches the previously constructed blocks and the additional data.

A more detailed description of the algorithm is provided below. Let $S = \{s(k), k = 1, \dots, K\}$ represent possible values that the blocks in the realizations can take (in the case of binary models, these are 0 and 1). Define a data template d_n of size n as a set of vectors $\{n_\alpha, \alpha = 1, \dots, n\}$ specifying values $\{s(u+h_\alpha), \alpha = 1, \dots, n\}$ in the blocks neighboring u. Based on these definitions, the algorithm proceeds as follows:

1. The training image is scanned with the given template. Possible value sets corresponding to the template and the values of central blocks are stored in a search tree. For each set, a probability distribution $P(S(u) = s_k|d_n) = \frac{c_k(d_n)}{c(d_n)}, k = 1, \dots, K$ is saved. The value $c(d_n)$ is calculated as the frequency of the data set d_n in the training image, while $c_k(d_n)$ is the frequency of the set d_n given that the central block value u equals $s(k)$;

2. Values in specific blocks, set by additional data, are fixed. The remaining blocks are added to a queue according to a random path that determines the block traversal order;

3. For the current block, a value set is found in the search tree, taking into account values in neighboring blocks specified by the template;

4. If an appropriate set is found, a value is generated from the corresponding distribution and assigned to the current block. If no set is found, a retry with a smaller template size is performed;

5. Steps 2-4 are repeated until all blocks in the queue are processed. The result is a realization with the correlation structure of the training image and that satisfies the additional data.

Dimensionality Reduction Methods

Numerous methods have been developed for parameterizing and reducing the dimensionality of geological models. A commonly used method is Principal Component Analysis (PCA), based on the singular decomposition of a matrix obtained from a set of realizations. The drawback of PCA is that it only considers correlations between two points in the medium, making it inapplicable to non-Gaussian geological models. There are methods based on the discrete wavelet transform (DWT) and discrete cosine transform (DCT). Their primary limitation is that the parameterized realizations may no longer match the additional data, as these methods, unlike PCA, do not use the covariance matrix of prior realizations.

Deep learning-based methods are currently gaining popularity. Generative adversarial networks (GANs) [36] are employed; they consist of two parts: a generative model that generates geological realizations from a low-dimensional parameter vector and a discriminative model that evaluates the generated realizations' adherence to additional data and the training image. Variational autoencoders [46] are also used, comprising two interconnected neural networks: an encoder and a decoder. The encoder compresses the initial realizations by mapping them into latent space. The decoder seeks to reconstruct the initial realizations from latent variables with minimal information loss. The paper [46] demonstrates a substantial improvement in parameterization results using variational autoencoders compared to PCA, DCT, and DWT parameterization methods.

This work employs the CNN-PCA parameterization method, which combines PCA and deep neural networks. Unlike GAN and autoencoder-based methods, CNN-PCA requires significantly fewer training resources, as it only enhances realizations generated by PCA without needing to generate them from scratch.

Principal Component Analysis (PCA) Parameterization

The PCA-based parameterization method includes the following steps:

1. Using the SNESIM algorithm, N_R realizations satisfying the additional data are generated based on the training image;
2. A matrix $X_C = [m_1 - \overline{m}, \cdots, m_{N_R} - \overline{m}]$ is constructed, where m_i is a realization vector of size N_C, $\overline{m}$ is the mean vector over all realizations, and N_c is the number of blocks in the geological model;
3. An SVD decomposition of the matrix $Y = \frac{X_C}{\sqrt{N_R - 1}}$ is performed: $Y = U \Sigma V^T$;
4. New realizations are obtained using the formula $m_{pca}(\xi) = U_l \Sigma_l \xi + \overline{m}$, where U_l is the matrix of l singular vectors, Σ_l is the diagonal matrix of the l largest singular values, and ξ is a vector of size l, which can be a sample from a standard normal distribution for generating a random realization or a vector that best matches historical data for history matching.

CNN-PCA Parameterization

For binary geological models, the O-PCA method [72] is proposed, which solves the following optimization problem:

$$m_{opca}(\xi) = \arg\min_{x} \|U_l \Sigma_l \xi + \overline{x} - x\|_2^2 + \gamma x^T(\vec{1} - x) \tag{2.1}$$

where x is a vector of size N_C. The component x_i describes the value at block i, $\vec{1}$ is a vector of ones of size N_C, the first term ensures similarity to the PCA model, and the second term regularizes the solution to be as close as possible to binary (minimum values at 0 and 1). λ is the regularization parameter.

For binary models, this optimization problem has a simple analytical solution.

However, O-PCA represents a histogram transformation of the PCA solution. The solutions obtained through it do not always fully reproduce what is contained in the training image, especially if only a small amount of additional data is available.

The generalization of the O-PCA method proposed in [49] involves solving the following optimization problem:

$$m_(\xi) = \arg\min_{x} \left(L_C(x, m_{pca}(\xi)) + \gamma_s L_S(x, M_{ref})\right). \tag{2.2}$$

The function L_C ensures that x is similar to m_{pca}. In a more general form, unlike O-PCA, where L_C is represented by the Euclidean norm, L_S represents the style of realizations, with "style" meaning the similarity of the desired realization x to the training image M_{ref}, preserving channel continuity, width, and the sharp contrast between facies (high-permeability and low-permeability media). In O-PCA, this is analogous to the regularization term $X^T(1 - X)$, ensuring binary outputs.

Representation of L_C

The function L_C, which ensures similarity between realization x and the

PCA realization, is defined as:

$$L_C(x, m_{pca}(\xi)) = \frac{1}{N_F}|F(x) - F(m_{pca}(\xi))|^2_{Fr}, \tag{2.3}$$

where $F(x)$ is the chosen representation of model x. If $F(m_1) = F(m_2)$, then models m_1 and m_2 have similar content (e.g., channels in m_1 are approximately in the same places as in m_2), and N_F is the number of elements in the representation (matrix).

The advantage of using a more complex representation $F(x)$ over directly comparing model x to $m_{pca}(\xi)$ ($||x - m_{pca}||^2_2$) is the reduced likelihood of models being mistakenly compared blockwise (with direct comparison, one region of model x could mistakenly be matched with a similar region in $m_{pca}(\xi)$, but located elsewhere).

For obtaining $F(x)$ functions and those in L_S, the VGG-16 convolutional neural network pre-trained on the ImageNet database is used. The fourth layer output (activation function output from convolution) — $F_4(m)$ — serves as the $F(x)$ function. Consequently, the L_C function is expressed as:

$$L_C(x, m_{pca}(\xi)) = \frac{1}{N_{z,4}N_{c,4}}|F_4(m) - F_4(m_{pca}(\xi))|^2_{Fr}, \tag{2.4}$$

where $N_{z,4}$ is the number of filters (convolutions) in the fourth layer, and $N_{c,4}$ is the number of elements in each convolution result for the fourth layer.

Representation of L_S

To describe the correlation structure of the space specified by the training image, a set of statistical metrics G_k is used. These metrics are obtained from the convolutional network as the covariance matrices of the nonlinear responses of filters at the k-th layer output:

$$\begin{gathered} G_k(m) = \frac{1}{N_{c,k}N_{z,k}} F_k(m)F_k(m)^T, \\ L_S(x, M_{ref}) = \sum_{k=(2,4,7,10)} \frac{1}{N^2_{z,k}} ||G_k(x) - G_k(M_{ref})||^2_{Fr}, \end{gathered} \tag{2.5}$$

where $N_{z,k}$ is the number of filters for layer k.

Thus, L_S captures the average similarity of realization x to the training image M_{ref} at various levels of image detail.

Model for Realization Generation

Generating realizations by minimizing $m(\xi)$ is too resource-intensive, as optimization would need to be performed in each iteration during history matching.

The model for generating realizations based on vectors ξ in explicit form is presented in [49]. A second CNN network f_W (model transform net) is used, trained on random data ξ generated from a standard normal distribution. The loss function is defined as follows (in accordance with (2.3) and representations (2.4) for L_C and (2.5) for L_S):

$$Loss(W, \xi) = L_C(f_W(\xi), m_{pca}(\xi)) + \gamma_s L_s(f_W(\xi), \xi) + L_h, \tag{2.6}$$

where W represents the parameters of network f_W.

The term L_h in $Loss(x)$ ensures that the generated realizations satisfy additional data. It is calculated by:

$$L_h = \frac{1}{N_h}(h^T m_{pca}^i - f_W(m_{pca}^i))^2, \tag{2.7}$$

where h is a vector of ones at positions where additional data is available, and zeros elsewhere, and N_h is the number of points where additional data is provided.

2.0.2 SPSA Algorithm and Theorem on the Properties of Estimates in the Context of the Given Problem

We formulate the task of adapting the geological model to production history as minimizing the following objective function with respect to ξ:

$$f_t(\xi) = \frac{||d_{\text{fopr}}(m) - d_{t_{\text{obs}}}||_2}{\max(d_{t_{\text{obs}}})} + \sum_i \frac{||d_i(m) - d_{t_{\text{obs},i}}||_2}{\max(d_{t_{\text{obs},i}})} + \frac{||\xi - \xi_0||_2}{\max(\xi_0)}, \quad (2.8)$$

where $d_{t_{\text{obs}}}$ in the first term represents the observed production data available at time t, such as oil and water production rates. $d_{\text{fopr}}(m)$ denotes the simulated data obtained by forecasting oil and water production using a hydrodynamic simulator with variable permeability parameters m. These parameters are derived from ξ using the CNN-PCA parameterization model, mapping parameterized realizations to physical permeability values.

The second term is the difference between the simulated $d_i(m)$ and observed $d_{t_{\text{obs},i}}$ production rates, specified individually for each well i.

The third term is a regularization term, requiring ξ to be close to the parameters ξ_0. This is necessary as CNN-PCA was trained using ξ values chosen from this distribution.

Each term of the objective function is normalized by the maximum of the corresponding observation, ensuring that data from the entire field and from each well carry equal weight, regardless of the absolute values of the observations.

SPSA

Stochastic approximation algorithms can be applied to optimization problems when the objective function is noisy or its gradients are unavailable.

These algorithms have the general form:

$$\hat{x}_{n+1} = \hat{x}_n - \alpha_n \hat{g}_n(x_n),$$

where $\hat{x}_n$ is the sequence of parameter estimates, $\hat{g}_n(x_n)$ is a pseudo-gradient with an expected value equal to the true gradient, and α_n is the convergence rate parameter.

SPSA is a stochastic approximation algorithm that requires only two measurements of the objective function per iteration to approximate the gradient. Moreover, parameter estimates obtained via this algorithm can converge to their true values under arbitrary bounded additive noise, while many other optimization algorithms require the noise to be normally distributed and/or have a zero mean. If α is constant and sufficiently small, SPSA guarantees convergence of the mean-square estimates to a small bounded region around the true parameter value [41].

Consider an observation model where historical production data are available at time $t \in \mathbb{R}$:

$$y_t = f_t(x_t) + \nu_t, \tag{2.9}$$

where ν_t is the additive noise due to parameterization and measurement errors in the production data.

Let $\mathcal{F}_{t-1}$ denote the σ-algebra of all probabilistic events occurring up to time t, where $\mathbb{E}_{\mathcal{F}_{t-1}}$ represents the symbol of conditional expectation with respect to the σ-algebra $\mathcal{F}_{t-1}$.

Using observations $y_1, y_2, \ldots, y_t$, we need to construct an estimate $\hat{\xi}$ of the unknown parameter vector ξ that minimizes the mean-risk functional F_t:

$$F_t(\xi) = \mathbb{E}_{\mathcal{F}_{t-1}}[f_t(x_t)].$$

These estimates can be constructed using the SPSA algorithm.

The SPSA algorithm's input parameters are sequences of positive numbers α_n, β_n that approach zero, and an initial solution $\hat{\xi}_0$. The parameter estimates ξ_n can then be obtained using the following formulas:

$$\begin{cases} u_{2n} = \hat{\xi}_{n-1} + \beta_n \Delta_n, \\ u_{2n-1} = \hat{\xi}_{n-1} - \beta_n \Delta_n, \\ \hat{\xi}_n = \hat{\xi}_{n-1} - \dfrac{\alpha_n}{2\beta_n} \Delta_n (y_{2n} - y_{2n-1}), \end{cases}$$

where y represents the noisy measurements (2.9) of the objective function, and Δ_n is an observable sequence of independent Bernoulli random vectors from $\mathbb{R}^d$, with each component independently taking values $\pm\frac{1}{\sqrt{d}}$.

Theorem 1. *Assume the following conditions hold:*

1. *The functions $F_t(\cdot)$ have unique minima ξ_t and, for all $x \in \mathbb{R}^d$:*

$$\langle x - \xi_t,\ \mathbb{E}_{F_{t-1}}[\nabla f_{\xi_t}(x)] \rangle \geq \mu \|x - \xi_t\|^2,$$

where $\mu > 0$ is a constant.

2. *The gradient at the minimum point is bounded:*

$$\mathbb{E}\|\nabla f_{\xi_t}(\xi_t)\|^2 \leq g^2.$$

3. *For all $\xi \in \Xi$, the gradient $\nabla f_\xi(x)$ satisfies the Lipschitz condition:*

$$\forall x', x'' \in \mathbb{R}^d,\ \|\nabla f_\xi(x') - \nabla f_\xi(x'')\| \leq M\|x' - x''\|,$$

where $M \geq \mu$ is a constant.

4. *For $n = 1, 2, \dots$, the observation noise difference is bounded:*

$$|\bar{v}_n| = |v_{2n} - v_{2n-1}| \leq c_v,$$

where $c_v > 0$ is a constant.

5. *The algorithm's steps $\alpha_n = \alpha$ and $\beta_n = \beta$ are constant and satisfy:*

$$0 < \alpha < \frac{2\mu}{M^2}, \quad \beta > 0.$$

Then the sequence of estimates $\hat{\xi}_{2n}$ generated by the SPSA algorithm satisfies the inequality:

$$\limsup_{n\to\infty} \mathbb{E}\left[\|\hat{\xi}_{2n} - \xi_{2n}\|^2\right] \leq L,$$

where

$$L = h + \sqrt{h^2 + \frac{\bar{l}}{k}},$$

with constants h, $\bar{l}$, and k.

Proof.

We will prove the theorem by showing that under these assumptions, all conditions of Theorem 1 in [41] hold, and therefore, the result follows from this theorem. Consider assumptions 1–8 in [41] and justify their fulfillment in the context of the geological model adaptation problem (2.8).

1. In this problem, observations are subject to noise ν_t, which models measurement and simulation errors. We assume that the noise difference between consecutive observations is bounded:

   $$|\bar{v}_n| = |v_{2n} - v_{2n-1}| = |\nu_{2n} - \nu_{2n-1}| \leq c_v.$$

   This corresponds to Assumption 1 in [41]. The constant c_v can be estimated through the known measurement errors of dynamic production data and the degree to which simulated data changes relative to measurement error.

2. The true parameter vector ξ_t is either constant over time or changes very slowly:

   $$\|\xi_t - \xi_{t-1}\| \leq \delta_\xi,$$

where $\delta_\xi \geq 0$ is a small constant. This assumption holds in real-world applications because the geological model describes intrinsic rock properties — such as porosity and permeability values, or channel locations, which hardly change over time within the planning horizon.

3. The arrival of new data leads to changes in the function $f_t(\xi)$ and its gradient. We assume that the gradient change rate is bounded:

$$\|\mathbb{E}_{F_{2n-2}}[\nabla \phi_n(x)]\| \leq a_1\|x - \xi_{2n-2}\| + a_0,$$

where $\phi_n(x) = f_{\xi_{2n}}(x) - f_{\xi_{2n-1}}(x)$, and constants a_1 and a_0 are small. This corresponds to Assumption 3 in [41].

It can be assumed that constants $\delta_\xi, \alpha_0, \alpha_1$ are small in practice because model adaptation at each time point is based on a small number of new, and a larger amount of accumulated prior observed data.

4. The assumption in the theorem condition corresponds to Assumption 4 in [41]:

$$\langle x - \xi_t,\ \mathbb{E}_{F_{t-1}}[\nabla f_{\xi_t}(x)]\rangle \geq \mu\|x - \xi_t\|^2.$$

In practical problems, this condition is challenging to verify as the relationship between geological model parameters and dynamic production data is complex and nonlinear, and it largely depends on the specifics of the hydrodynamic simulator. This requirement is partially met by the regularization term in (2.8), which keeps the solution within a region where the function behavior is more convex.

5. The theorem condition assumes that the gradient of the function at the minimum is bounded:

$$\mathbb{E}\|\nabla f_{\xi_t}(\xi_t)\|^2 \leq g^2,$$

where $g \geq 0$ is a small constant. This corresponds to Assumption 5 in [41]. In practice, the true model parameter (e.g., reservoir permeability) changes minimally over time, and the error function itself remains near its minimum close to the true parameter, considering minor parameterization model errors and data noise. Under physical constraints and parameterization model smoothness, the gradient near the true parameter remains bounded by a small constant.

6. The Lipschitz assumption corresponds to Assumption 6 in [41]. Like assumption 4, Lipschitz continuity is challenging to prove due to the hydrodynamic simulator's presence in the optimization problem. It can be assumed that the constant M is relatively small for most x', x'', as hydrodynamic simulators, while not strictly ensuring Lipschitz continuity, use numerical methods for solving partial differential equations that provide moderate solution changes relative to model parameters. Additionally, the neural network-based parameterization model with PCA also exhibits relatively small value changes in response to input parameter changes, resulting in moderate M values.

7. Perturbation vectors Δ_n and functions $K_n(\Delta_n)$ satisfy Assumptions 7 and 8 from [41]. In this problem formulation, independent Bernoulli random vectors Δ_n with components taking values of $\pm\frac{1}{\sqrt{d}}$ with a probability of 0.5 are used. This corresponds to the following properties:

   - $\|\Delta_n\| = 1 \leq c_\Delta$;
   - $\|K_n(\Delta_n)\| = 1$, where $K_n(x) \equiv x$;
   - Functions $K_n(\cdot)$ and perturbation distributions $P_n(\cdot)$ meet the required conditions, including independence and symmetry.

   These assumptions hold by the construction of the SPSA algorithm and the properties of the random perturbations used.

Since all assumptions of Theorem 1 in [41] hold, applying it yields that the sequence of estimates $\hat{\xi}_n$ satisfies the inequality:

$$\limsup_{n\to\infty} \mathbb{E}\left[\|\hat{\xi}_n - \xi_n\|^2\right] \leq L,$$

where

$$L = h + \sqrt{h^2 + \frac{\bar{l}}{k}}.$$

In our problem, given the small values of δ_ξ, a_0, a_1, and g, the constants h, $\bar{l}$, and k can be estimated, yielding in the limit:

$$L = \frac{\alpha^2 c_v^2}{4\beta^2\gamma},$$

where

$$\gamma = 2\alpha\sigma_\Delta^2\mu - \alpha^2\sigma_\Delta^2 M^2 - \alpha a_1 > 0,$$

and $\sigma_\Delta^2 = \frac{1}{d}$.

□

2.0.3 Conclusions

This chapter discusses methods for adapting sparse geological models to historical field production data. It is shown that the primary challenge in model adaptation lies in the high dimensionality of model parameter space, making the search for optimal solutions computationally difficult. To address this, parameterization methods such as PCA and its improved versions for non-Gaussian models, including CNN-PCA, are used.

Various optimization algorithms have been considered for minimizing discrepancies between simulated and observed data. It is noted that commonly used methods have limitations, such as difficulties with binary models and the need to generate numerous realizations. The stochastic optimization method SPSA is proposed as an alternative, which has the advantage of minimal function evaluations to estimate the gradient and robustness to noisy observed data. Furthermore, a theorem on the upper bound of the mean-square error of the SPSA algorithm for the given problem has been formulated, showing that even under uncertainty and noise, the geological model parameter estimate will converge to the true value with controllable accuracy.

Chapter 3

Compression-Based Recognition Methods in Distributed Clustering and Ultrasonic Tomography Tasks

This chapter proposes methods for solving sparse problems in compressed space based on a compressed sensing methodology. The first section describes a distributed clustering algorithm for multi-agent systems, assuming the global state of the system is sparse due to its clustered structure. The second section presents two methods: a method for compression and reconstruction of ultrasonic data based on a trainable neural network, and a method for solving the modeling problem (ultrasound imaging from data) in compressed space, without preliminary data restoration.

3.0.1 Distributed Clustering in Multi-Agent Systems Based on Compression-Based Recognition Methodology

Multi-agent systems model a wide range of real-world problems, such as the Internet of Things and distributed sensor systems. Consequently, many challenges arise in controlling such systems, traditionally involving mesoscale (individual agent control synthesis) and macroscale control levels (unified control synthesis for the entire system). Macroscopic control is often insufficiently precise, especially when there are significant data transmission delays. Meanwhile, microscopic control becomes overly complex as the number of agents increases.

An alternative solution in many cases can be based on the property that many systems exhibit cluster synchronization [6]—agent states cluster into a small number of clusters, within which they exhibit similar behavior. In this scenario, agent control can be simplified significantly by synthesizing control for each cluster instead of each individual agent.

For distributed cluster control, each agent must be aware of the parameters of the cluster to which it belongs. This necessitates an effective distributed clustering method that enables each agent, based on local interactions with other agents, to obtain parameters for all system clusters. This chapter proposes a distributed clustering method for multi-agent systems that reduces data transmission volumes and reconstructs clusters in real-time.

Problem Formulation

A multi-agent system can be represented as a set $\mathcal{N} = \{1, 2, \ldots, N\}$, where each agent i in this set has its own state x_i, a vector in the space $\mathbb{R}^d$. The system consists of N agents, and the state space has dimension d. Communication among agents is modeled using a directed graph $G = (\mathcal{N}, E)$. The adjacency matrix $B = [b_{ij}]$ describes the directed communication paths between agents.

Cluster parameters, denoted as Θ, include the centroids of k clusters, defined as $\overline{x}_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$ for each cluster C_k. It is assumed that the agent states are clustered at any given time, ensuring system sparsity, and allowing the use of compressive sensing theory for efficient data transmission. Note that communication topology is not necessarily linked to agent clustering—two agents within the same cluster may not be connected, and agents from different clusters may exchange information.

We propose a decentralized algorithm to estimate the parameters Θ based on the Local Voting Protocol (LVP) and a deep learning method for rapid centroid recovery. The algorithm should be scalable with respect to the number of agents N and adaptable to changes in cluster structure. To avoid direct transmission of large volumes of data about multi-agent states, only compressed measurements are used in communication, represented as $\overline{y} = A\mathbf{x}$. Here, $\mathbf{x} = [x_1^{\mathrm{T}}, \dots, x_N^{\mathrm{T}}]^{\mathrm{T}}$ is the vector of all agent states in the space $\mathbb{R}^{Nd}$, and $A \in \mathbb{R}^{md \times Nd}$, where $m \sim k \ll N$.

Distributed Clustering Method

We assume that each agent $i \in \mathcal{N}$ independently collects private measurements as follows:

$$y_i = A_i x_i, \tag{3.1}$$

where $y_i \in \mathbb{R}^{md}$ is the compressed observation of agent i, $A_i = A_{(\cdot,i)} \in \mathbb{R}^{md \times d}$ is the measurement operator for agent i, and $A_{(\cdot,i)}$ represents the columns of matrix A corresponding to agent i. The aggregate measurement vector is then obtained as follows:

$$\overline{\mathbf{y}} = \frac{1}{N} \sum_{i \in \mathcal{N}} y_i = \frac{1}{N} \sum_{i \in \mathcal{N}} A_i x_i, \tag{3.2}$$

where $\overline{\mathbf{y}}$ represents a set of compressed measurements, computed as weighted sums of agent states with random weights. This method of gathering measurements $\overline{\mathbf{y}}$ allows the use of the Local Voting Protocol to synchronize compressed measurements across all agents. To estimate cluster parameters, a trainable method based on a neural network—similar in architecture to PointNet [64] except for the first and last layers—is proposed:

1. **First Layer**: A learnable projection from N (number of agents) to m (128—dimension of the compressed measurements) using a linear layer, applied independently to each of two dimensions (compression factor—$N//m$).

2. **Convolutional Layers**: Five convolutional layers with Rectified Linear Unit (ReLU) activation and Batch Normalization. Channel dimensions: $2 \to 64 \to 64 \to 128 \to 1024$, transforming the two-dimensional input into a feature vector of dimension 1024.

3. **Average Pooling**: Applied along one of the dimensions, reducing the size from m to 1. The result is then concatenated with the output of the third layer (to account for local and global features of the compressed point cloud).

4. **Convolutional Layers**: Five convolutional layers reduce the dimension from $1024 + 64$ to the number of output channels corresponding to the number of clusters K.

5. **Final Projection**: The second dimension is projected to 2 in the final step, yielding the output dimensions: $[B, 2, K]$.

As a loss function (3.3), a function based on the Hungarian method for solving the linear assignment problem is used, following the method in [42]. This choice accommodates any arbitrary ordering in the predicted cluster set, simplifying the training process (it is important only to predict the set of centroids, not their order).

$$L_{\text{hun}}(\hat{Y}, Y) = \min_{\pi \in \Pi} \|\hat{y}_i - y_{\pi(i)}\|_2^2, \tag{3.3}$$

where $\hat{Y}$ are the predicted centroids, Y are the true centroids, and Π is the permutation space. The assignment problem is solved using the Hungarian algorithm in $O(n^3)$ time.

3.0.2 Compression and Reconstruction of Ultrasonic Tomography Data

Ultrasound Computed Tomography (UCT) has garnered significant interest due to the absence of radiation exposure for the human body, which distinguishes it favorably from X-ray Computed Tomography (CT). Moreover, UCT does not require the injection of contrast agents, essential for high-quality diagnostics in Magnetic Resonance Imaging (MRI). Unlike traditional ultrasound methods, UCT offers substantially higher reconstruction quality of the examined object by utilizing a large number of ultrasound sources and sensors. However, this leads to a considerable increase in data volume and heightened computational complexity of the image reconstruction process, necessitating the development of more efficient data acquisition and processing methods.

This section proposes and describes a method for efficient data acquisition from UCT sensors. Such a tomograph is composed of a ring of N elements, uniformly distributed around the circumference, capable of simultaneously emitting and receiving ultrasound signals. The examined object is placed inside this ring. Each sensor sequentially generates an ultrasound signal, while the others remain in receiving mode for t time intervals. After collecting data from all sensors, the ring shifts along the vertical axis, and the process repeats to obtain data for a new slice of the object.

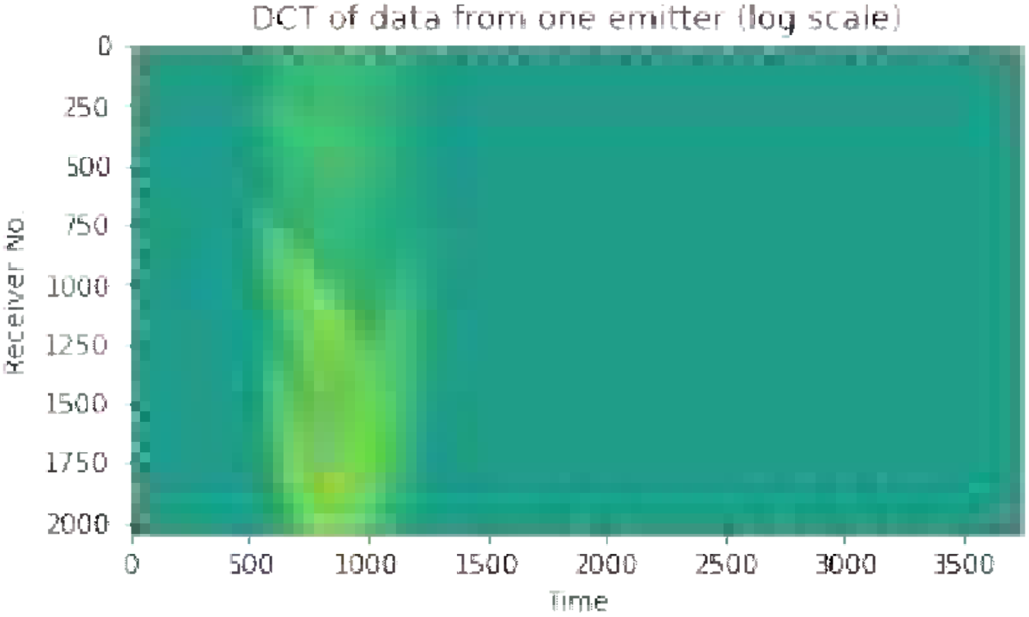

Figure 3.1: Example of ultrasound signal data obtained from a single source. Data displayed in the frequency domain (Discrete Cosine Transform) on a logarithmic scale.

As a result, the tomograph generates a data volume of N^2t. In the conducted experiment, with $N = 2048$ and $t = 3750$, this results in the need to collect over 50 GB of information for each slice. The application of a compressed sensing approach enables a significant reduction in the amount of data collected by using a sampling rate lower than required by the Nyquist-Shannon theorem. According to this theorem, a signal with a limited spectrum can be fully reconstructed from measurements sampled at a rate more than twice the maximum frequency of the spectrum. However, ultrasound data is inherently sparse (as illustrated in Fig. 3.1), and the useful signal can be represented in a more compact form.

Full-Waveform Inversion (FWI) Problem

Full-Waveform Inversion (FWI) is a numerical optimization method aimed at accurately recovering acoustic properties of the medium, such as sound speed and density, based on wavefield data. In the context of Ultrasound Computed Tomography, FWI is used to obtain an image of the object based on a large volume of wave data from sensors.

FWI is based on solving the inverse problem for the wave equation. In acoustic problems, the main wave equation can be written as:

$$\frac{1}{c^2(\mathbf{r})}\frac{\partial^2 p(\mathbf{r},t)}{\partial t^2} - \Delta p(\mathbf{r},t) = s(\mathbf{r},t), \quad (\mathbf{r},t) \in \mathbb{R}^d \times [0,T], \tag{3.4}$$

where $c(\mathbf{r})$ is the speed of sound, $p(\mathbf{r},t)$ is the sound pressure, $s(\mathbf{r},t)$ is the source function, and Δ is the Laplace operator.

The initial conditions for the wave equation are given by:

$$p(\mathbf{r},0) = 0, \quad \frac{\partial p(\mathbf{r},0)}{\partial t} = 0. \tag{3.5}$$

Sensors are located on a surface S around the object and collect measurements over time T:

$$d(\mathbf{r},t) = p(\mathbf{r},t)\Big|_{\mathbf{r}\in S}, \quad t \in [0,T]. \tag{3.6}$$

The goal of full-waveform inversion is to recover the sound speed matrix $c(\mathbf{r})$ from the measured data $d(\mathbf{r},t)$. This is formulated as an optimization problem for a loss function between the simulated and measured data:

$$\hat{c} = \arg\min_{c} \frac{1}{2}\sum_{i=1}^{I}\int_0^T \int_S \left|d_i(\mathbf{r},t) - p_i^{\text{sim}}(\mathbf{r},t;c)\right|^2 \,\mathrm{d}S\,\mathrm{d}t, \tag{3.7}$$

where $p_i^{\text{sim}}(\mathbf{r},t;c)$ represents the simulated data for the i-th source at the current estimate of the sound speed $c(\mathbf{r})$. The problem is solved iteratively, computing simulated data (applying the forward wave equation operator) and solving the adjoint problem to compute the gradient.

Image Reconstruction Method Based on Time-of-Flight Analysis

An alternative simplified image reconstruction method, based on Time-of-Flight (TOF) analysis of ultrasound signals, was proposed in [31]. This method aims to detect specific regions and estimate their density using ultrasound tomography data.

The primary steps of this method include data acquisition and preprocessing, TOF calculation, and subsequent image reconstruction. During data acquisition, each element of the ring array sequentially emits an ultrasound signal, while the other elements in the array receive the signal, forming a complete dataset for each source.

To compute the TOF, the Akaike Information Criterion (AIC) is employed, enabling accurate detection of the arrival time of the signal at each receiver. The AIC method is based on dividing the signal into two parts, before and after the wave arrival, and then computing variances for each part. The modified AIC formula is given as:

$$\mathrm{AIC}(k) = k \cdot \ln\left(\sigma_1^2(k)\right) + (N - k - 1) \cdot \ln\left(\sigma_2^2(k)\right),$$

where k is the current time index in the analysis window, N is the total number of points in the window, and $\sigma_1^2(k)$ and $\sigma_2^2(k)$ are the variances of the signals before and after point k.

After TOF calculation, an inverse problem is solved to estimate the distribution of sound speeds in the area under study. A ray-tracing method is applied, which relates TOF to integrals over the signal propagation paths. The time-of-flight between transmitter i and receiver j is expressed as:

$$\mathrm{TOF}_{ij} = \int_{L_{ij}} \frac{1}{c(\mathbf{r})}\, dl,$$

where TOF_{ij} is the time-of-flight between transmitter i and receiver j, $c(\mathbf{r})$ is the sound speed at point $\mathbf{r}$, and L_{ij} is the signal propagation path between these elements. An image is obtained by solving a system of linear equations, which can be solved using least squares or regularization to ensure stable solutions.

The advantages of this method include its simplicity, high computational speed, and potential for parallelization. The method does not require complex calculations related to solving the full-wave equation, making it suitable for rapid preliminary image generation. TOF calculations for different transmitter-receiver pairs are independent, enabling parallel computation.

Ultrasound Data Compression and Reconstruction Method

The following describes the proposed method for compressing and reconstructing data obtained from the sensors of the ultrasound tomograph.

Data Preprocessing

In the initial preprocessing step, the median value of the signal data from each receiver is subtracted. This eliminates noise resulting from varying signal levels among sensors in the idle state (example shown in Fig. 3.2).

Next, high-amplitude data portions are isolated and transmitted separately without compression. This step significantly improves signal recovery

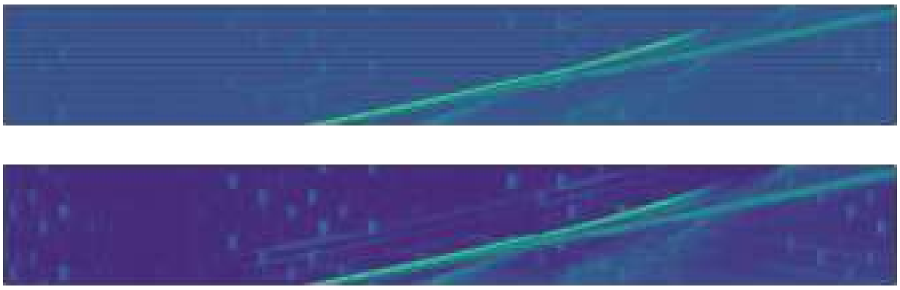

Figure 3.2: Example of preprocessing results. Top - original image with noise in the form of horizontal lines.

accuracy, while adding minimal data since the high-amplitude portion occupies less than 5

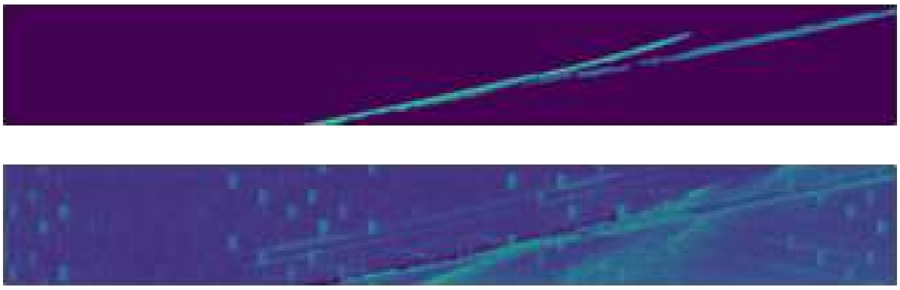

Figure 3.3: Data separation example. Top - isolated high-amplitude portion, bottom - main data portion.

Finally, data from each source are divided into small fragments (64x64), enabling the use of algorithms with low memory requirements.

Implementation Scheme

The overall reconstruction method scheme is illustrated in Fig. 3.4. The algorithm input is a vector of size $64 \cdot 64/CR$, where CR is the desired compression ratio (5 in the experiment). Initial reconstruction is performed using a fully connected neural network layer, outputting a $64 \cdot 64$ matrix. Subsequently, a convolutional neural network RIDNet [8] is used to restore local structures. Additionally, a skip connection is added to the network output, which improves training by allowing partial signal recovery from the fully connected layer.

The RIDNet convolutional network architecture is depicted in Fig. 3.5. EAM represents an attention mechanism-based module, which enables the network to detect patterns between input and output data. EAM begins with two branches of convolutional layers with different parameters, which are then summed and fed into further convolutional layers.

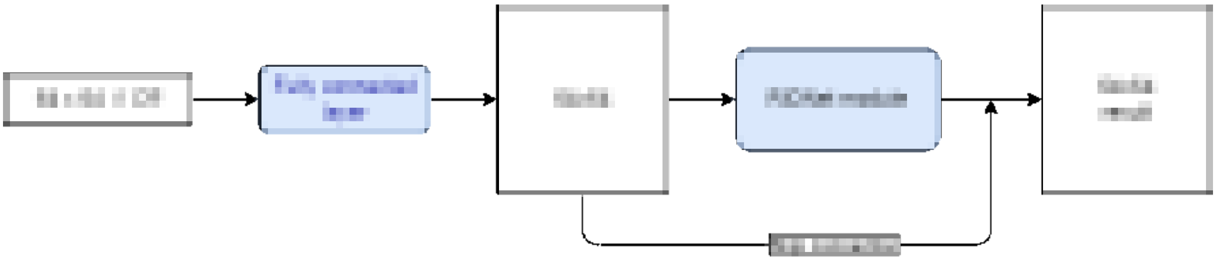

Figure 3.4: Reconstruction method architecture. CR - signal compression ratio, Fully connected layer - fully connected layer (linear transformation with learnable weights), RIDNet module - convolutional neural network.

The final three blocks implement a channel attention mechanism. The first of these blocks, Global Pooling, computes the mean and variance of the input data. The last block, a sigmoid function, outputs weights that indicate the importance of each data channel. The resulting vector is the input vector multiplied by these weights.

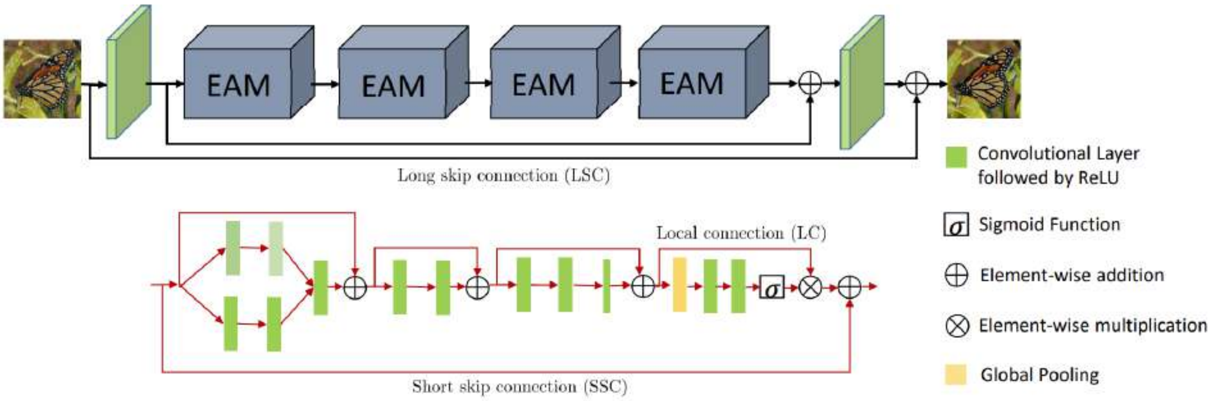
EAM
EAM
EAM
EAM
Long skip connection (LSC)
Local connection (LC)
Short skip connection (SSC)
σ
🟩 Convolutional Layer followed by ReLU
$\boxed{\sigma}$ Sigmoid Function
⊕ Element-wise addition
⊗ Element-wise multiplication
🟨 Global Pooling

Figure 3.5: RIDNet architecture [8]. EAM - attention mechanism module, structure shown below. Convolutional layers are in green. Arrow lines - skip connections (adding results from one layer to another)

The loss function for training the proposed network on ultrasound tomography data is defined as follows:

$$\frac{||x - f(x)||_1}{||x||_1} + \frac{||Af(y) - y||_2}{||y||_2},$$

where x represents the desired data, $f(y)$ is the function defined by the neural network with architecture shown in Fig. 3.4 producing reconstructed data, y is the compressed observation vector, and A is the designed measurement matrix. Thus, the network is trained to minimize the ℓ_1 distance between the original and reconstructed signal while also striving to minimize the distance between $Af(y)$ and y.

3.0.3 Full-Waveform Inversion with Compressed Measurements

The previous section examined the problem of efficient data acquisition in ultrasound computed tomography. A method based on compressive sensing was proposed to significantly reduce data collection volume without compromising result quality. Next, we describe a hybrid full-waveform inversion method for image reconstruction from compressed ultrasound data, enabling direct image reconstruction from the compressed signal representation without requiring full data reconstruction beforehand.

The problem is formulated as in [7]. Let us consider an inhomogeneous medium with a distribution of inverse velocities $\mathbf{s}(\mathbf{r})$, where $\mathbf{r} = (x, z)$ are spatial coordinates. The goal is to reconstruct the distribution $\mathbf{s}(\mathbf{r})$ based on ultrasound wave measurements passing through this medium. Observed data are collected by ultrasound sources and receivers rotating around the object at angles θ_i, $i = 1, \dots, N_{\text{views}}$. At each angle, waves are emitted and signals recorded on the opposite side of the object.

For ultrasound wave propagation modeling, the angular spectrum method, detailed in [7], is employed.

We proceed with a method modification for operation in compressed space.

Compression is applied via a random matrix $A \in \mathbb{R}^{M\times N}$, where $M \ll N$. The observed data for angle θ_i are recorded as:

$$\mathbf{y}_i = A\mathbf{p}_{\text{obs},i},$$

where $\mathbf{p}_{\text{obs},i}$ represents the full-space data vector.

The simulated signal is expressed as:

$$\mathbf{y}_{\text{sim},i}(\mathbf{s}) = A\mathbf{p}_{N,i}(\mathbf{s}),$$

where $\mathbf{p}_{N,i}(\mathbf{s})$ is the simulated data for viewing angle i at the current estimate of the medium parameters $\mathbf{s}$.

The objective function for full-waveform inversion based on compressed data is formulated as:

$$J(\mathbf{s}) = \frac{1}{2}\sum_{i=1}^{N_{\text{views}}} \left\|\mathbf{y}_{\text{sim},i}(\mathbf{s}) - \mathbf{y}_i\right\|^2.$$

To minimize this function, a conjugate state method is used, which allows efficient gradient computation $J(\mathbf{s})$ with respect to the medium parameters $\mathbf{s}$. This method involves solving both forward and adjoint problems:

The forward problem for wavefield modeling is described by the equation:

$$\mathbf{A}(\mathbf{s}, f)\mathbf{p}(\mathbf{s}, f) = \mathbf{b}(f),$$

where $\mathbf{A}(\mathbf{s}, f)$ is the wave propagation operator, and $\mathbf{b}(f)$ is the wave source model for frequency f.

The adjoint problem is formulated as:

$$\mathbf{A}^{\dagger}(\mathbf{s}, f)\boldsymbol{\lambda}(\mathbf{s}, f) = \mathbf{K}^{\top} A^{\top} \left(A\mathbf{p}_{N,i}(\mathbf{s}) - \mathbf{y}_i\right),$$

where $\mathbf{A}^{\dagger}$ is the adjoint operator, $\boldsymbol{\lambda}(\mathbf{s}, f)$ is the adjoint state, and $\mathbf{K}$ is the sampling operator for data recorded by sensors.

The gradient of the objective function with respect to the medium parameters $\mathbf{s}$ is given by:

$$\frac{\partial J}{\partial s_m} = -\sum_{i=1}^{N_{\text{views}}} \sum_{f} \operatorname{Re}\left\{\boldsymbol{\lambda}^{\dagger}(\mathbf{s}, f)\frac{\partial \mathbf{A}(\mathbf{s}, f)}{\partial s_m}\mathbf{p}(\mathbf{s}, f)\right\},$$

where s_m is the inverse velocity at grid point m.

Data compression affects the right-hand side of the adjoint problem. The error residual is computed in the compressed space:

$$\delta\mathbf{y}_i = \mathbf{y}_{\text{sim},i}(\mathbf{s}) - \mathbf{y}_i.$$

Then, this residual is "uncompressed" using the transpose of the compression matrix:

$$\delta\mathbf{p}_{N,i} = A^{\top}\delta\mathbf{y}_i.$$

This ensures accurate computation of the adjoint state $\boldsymbol{\lambda}(\mathbf{s}, f)$ and, consequently, the gradient $\frac{\partial J}{\partial \mathbf{s}}$.

To improve solution stability, ℓ_1-regularization is introduced, encouraging sparsity in the modeled data:

$$J_{\text{reg}}(\mathbf{s}) = \lambda \sum_{i=1}^{N_{\text{views}}} \left\|A\mathbf{p}_{N,i}(\mathbf{s})\right\|_1,$$

where λ is the regularization coefficient. The full objective function becomes:

$$J_{\text{total}}(\mathbf{s}) = J(\mathbf{s}) + J_{\text{reg}}(\mathbf{s}).$$

Minimizing $J_{\text{total}}(\mathbf{s})$ enables recovery of parameters $\mathbf{s}$ that fit the observed data while also exhibiting sparsity properties. The optimization problem is solved using a wave-equation migration velocity analysis method, as described in [7].

Thus, a method for effectively solving the full-waveform inversion problem with compressed data is proposed. Next, we derive an upper error bound for the solution based on the number of compressed measurements.

Theorem 2. *Let the goal be to recover a model $f^* \in \mathbb{R}^N$ from data $g = F(f^*)$, where $F : \mathbb{R}^N \to \mathbb{R}^N$ is a nonlinear forward wave operator.*

Assume that:

1. For any $f \in \mathbb{R}^N$, the vector $F(f)$ is s-sparse, containing no more than s non-zero components.

2. The operator F is Lipschitz continuous with constant $L > 0$, meaning that for any $f_1, f_2 \in \mathbb{R}^N$,

$$\|F(f_1) - F(f_2)\|_2 \leq L\|f_1 - f_2\|_2.$$

3. An approximate solution $\tilde{f} \in \mathbb{R}^N$ exists in the full space such that

$$\|\tilde{f} - f^*\|_2 \leq \varepsilon_0.$$

Let $A \in \mathbb{R}^{m\times N}$ be a random matrix with independent, identically distributed entries, e.g., normally distributed $\mathcal{N}(0, 1/m)$.

Suppose the number of measurements m satisfies:

$$m \geq C \cdot s \cdot \log\left(\frac{N}{s}\right),$$

where $C > 0$ is a positive constant dependent on the Restricted Isometry Property (RIP) constant $\delta \in (0, 1)$.

Then with probability at least $1 - 2e^{-cm}$, where $c > 0$ is a constant dependent on δ, matrix A satisfies the RIP of order $2s$ with parameter δ, i.e., for all $2s$-sparse vectors $h \in \mathbb{R}^N$:

$$(1 - \delta)\|h\|_2^2 \leq \|Ah\|_2^2 \leq (1 + \delta)\|h\|_2^2.$$

Denote the compressed measurements y as:

$$y = AF(f^*).$$

Then the solution $\hat{f}$ obtained by minimizing the loss function in the compressed space:

$$\hat{f} = \arg\min_{f \in \mathbb{R}^N} \|AF(f) - y\|_2^2,$$

satisfies the following bound with probability at least $1 - 2e^{-cm}$:

$$\|F(\hat{f}) - F(f^*)\|_2^2 \leq \frac{1+\delta}{1-\delta} L\varepsilon_0.$$

Proof.

According to [82], for a random matrix A with elements $a_{ij} \sim \mathcal{N}(0, 1/m)$, if the number of measurements:

$$m \geq C \cdot s \cdot \log\left(\frac{N}{s}\right),$$

then with probability at least $1 - 2e^{-cm}$, matrix A satisfies the RIP of order $2s$ with parameter δ, i.e.,

$$(1-\delta)\|h\|_2^2 \leq \|Ah\|_2^2 \leq (1+\delta)\|h\|_2^2$$

for all $2s$-sparse vectors h.

Considering the loss function:

$$\tilde{\mathcal{L}}(f) = \|AF(f) - y\|_2^2 = \|A(F(f) - F(f^*))\|_2^2.$$

Since $\hat{f}$ minimizes $\tilde{\mathcal{L}}(f)$, for any f,

$$\tilde{\mathcal{L}}(\hat{f}) \leq \tilde{\mathcal{L}}(f).$$

In particular, for the full-space solution $\tilde{f}$,

$$\tilde{\mathcal{L}}(\hat{f}) \leq \tilde{\mathcal{L}}(\tilde{f}) = \|A(F(\tilde{f}) - F(f^*))\|_2^2.$$

Using the RIP for vectors $h = F(\hat{f}) - F(f^*)$ and $h = F(\tilde{f}) - F(f^*)$, which are $2s$-sparse, we obtain:

$$(1-\delta)\|F(\hat{f}) - F(f^*)\|_2^2 \leq \|A(F(\hat{f}) - F(f^*))\|_2^2 \leq (1+\delta)\|F(\hat{f}) - F(f^*)\|_2^2,$$

$$(1-\delta)\|F(\tilde{f}) - F(f^*)\|_2^2 \leq \|A(F(\tilde{f}) - F(f^*))\|_2^2 \leq (1+\delta)\|F(\tilde{f}) - F(f^*)\|_2^2.$$

Therefore, from $\tilde{\mathcal{L}}(\hat{f}) \leq \tilde{\mathcal{L}}(\tilde{f})$, we have:

$$(1-\delta)\|F(\hat{f}) - F(f^*)\|_2^2 \leq (1+\delta)\|F(\tilde{f}) - F(f^*)\|_2^2.$$

Hence:

$$\|F(\hat{f}) - F(f^*)\|_2^2 \leq \frac{1+\delta}{1-\delta}\|F(\tilde{f}) - F(f^*)\|_2^2.$$

Given the Lipschitz continuity of F,

$$\|F(\hat{f}) - F(f^*)\|_2^2 \leq \frac{1+\delta}{1-\delta} L\varepsilon_0.$$

□

Note: The theorem provides a bound on reconstruction accuracy in the data space. Under more stringent assumptions (e.g., local invertibility and Lipschitz continuity of the inverse operator), an upper error bound in the image space can also be derived.

Remark:
Further, by applying Theorem 9.25 from [38], we can derive an upper bound on the RIP constant δ and, consequently, on $\|F(\hat{f}) - F(f^*)\|_2^2$ in terms of the number of measurements and a desired probability ϵ:

If the number of measurements m satisfies the inequality:

$$m \geq 2\eta^{-2}\left(s\ln\left(\frac{eN}{s}\right) + \ln\left(\frac{2}{\varepsilon}\right)\right), \tag{3.8}$$

then with probability at least $1-\varepsilon$, the RIP constant δ_s of the matrix $\frac{1}{\sqrt{m}}A$ satisfies:

$$\delta_s \leq 2\left(1 + \frac{1}{\sqrt{2\ln\left(\frac{eN}{s}\right)}}\right)\eta + \left(1 + \frac{1}{\sqrt{2\ln\left(\frac{eN}{s}\right)}}\right)^2 \eta^2. \tag{3.9}$$

Thus, we obtain an accuracy bound (in the data space) when solving the full-waveform inversion problem in the compressed space.

The proposed method is based on an optimization approach to solve the full-waveform inversion problem. In the future, it could be combined with the compression method described in Section (3.0.2) and adapted for learning-based full-waveform inversion approaches, which generate seismic images directly from data using generative models, such as those proposed in [79].

3.0.4 Discussion

In this chapter, we proposed methods based on compressive sensing theory to address problems of distributed clustering and image reconstruction from ultrasound data, both in the original and directly in the compressed space, eliminating the need for full data reconstruction. A theorem was formulated, providing an upper bound on the error of the solution.

In distributed clustering, data compression allows efficient information exchange between nodes in multi-agent systems with a cluster structure. This significantly simplifies interaction between nodes, reducing the volume of transmitted information and enabling efficient computation of the global state of the system by each agent. In ultrasound tomography tasks, applying compressed measurements reduces data volume, speeds up processing and image reconstruction, which is especially relevant for three-dimensional ultrasound tomographs that may consist of a vast number of sensors and receivers.

Practical implementation of such diagnostic systems faces challenges, such as the complexity of creating a physical bus for data transmission from numerous sensors and the need for efficient processing of large data volumes. Potentially, applying a multi-agent approach to ultrasound tomographs allows grouping sensor data according to similar states, which opens up the possibility of sparse interconnections between system components and direct application of the proposed compression and image reconstruction methods without intermediate data reconstruction or transmitting the full data volume to a central node.

Chapter 4

Software Simulation and Experimental Research Results

This chapter provides implementation details and simulation results for the proposed methods.

4.0.1 Geological Model Parameter Adaptation System to Development History

System Architecture

The system's purpose is to search for an optimal set of parameters for geological models that best describe the available field development history data. The system is designed for two-dimensional stochastic geological models, defined using a training pattern and supplementary data.

The main components of the system are illustrated in Figure 4.1:

The realization generation module encompasses an algorithm for producing realizations based on a stochastic geological model. This model is defined through a training pattern. The module input consists of a training image in the form of a picture and supplementary data at specific points (measured data from drilling sites). The module output is a matrix where each column represents a realization vectorized into a one-dimensional format.

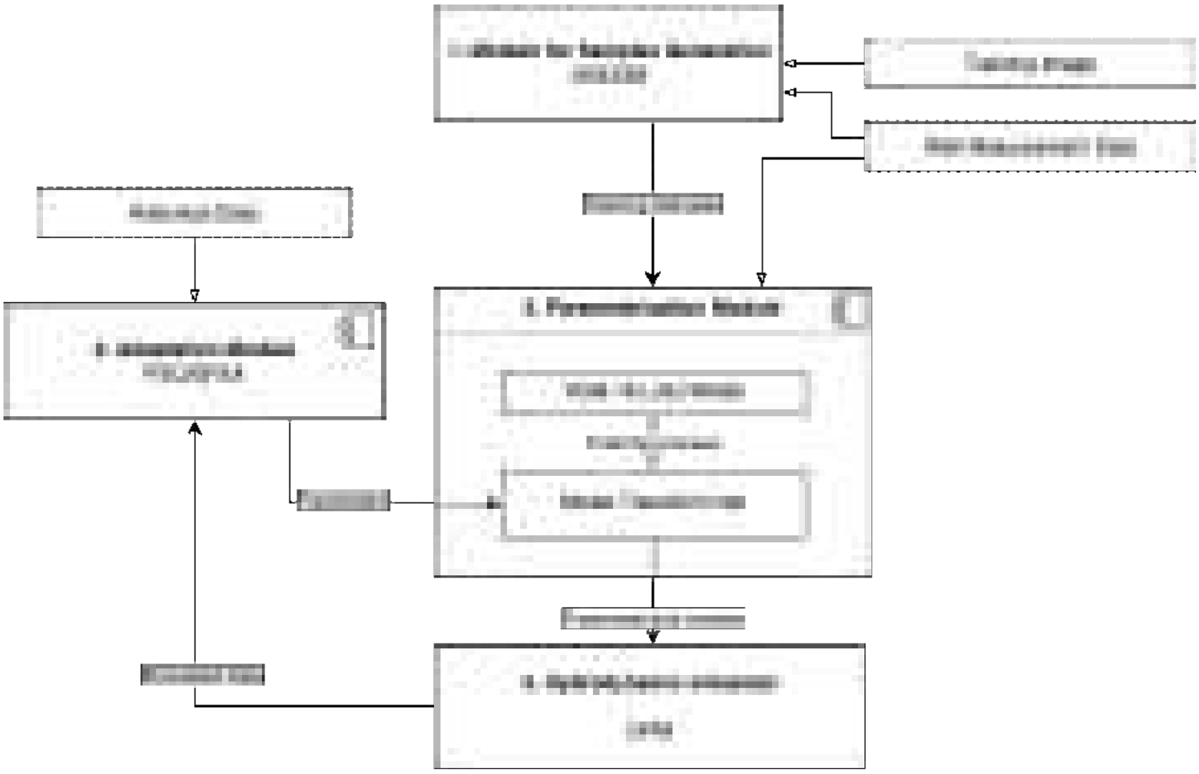

Figure 4.1: System Architecture

The parameterization module trains a neural network, Model Transform Net, to parameterize the realizations. The input to this module includes the matrix of realizations and hyperparameters utilized in the training process. The outcome is a trained model that, given a low-dimensional parameter vector ξ, generates realizations of the geological model. During training, an auxiliary neural network, VGG-16, is used in constructing the loss function. A detailed description of the CNN-PCA parameterization algorithm used is provided in Chapter 2.

The hydrodynamic simulator enables fluid flow simulation within an environment defined by variable porosity and permeability values, alongside static parameters. Within the system, the simulator is regarded as a function, where the input consists of permeability values for each grid block, and the output includes the oil or fluid production rates over a specified time interval for each well.

The model adaptation module addresses the primary task of the system. It is based on an optimization algorithm (SPSA or PSO), which minimizes the objective function, representing the deviation of simulated data from observed data.

The main use case of the system is depicted in Figure 4.2.

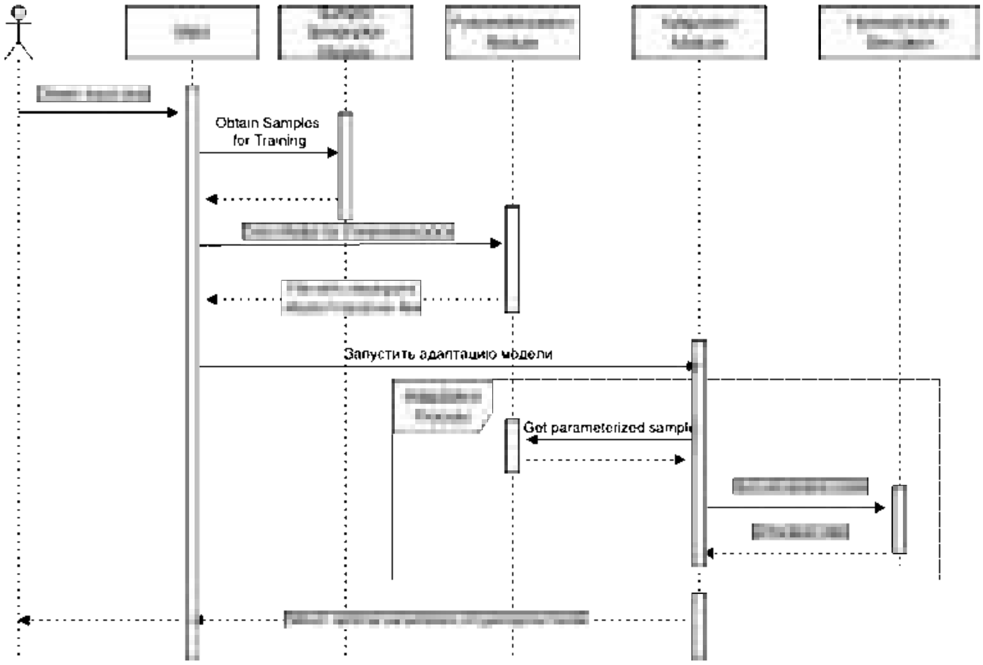

Figure 4.2: Primary System Use Case. Visualization of system interaction with the user and its behavior in solving the main task.

The user interacts with the system through a command line interface (Main), or, for example, via an interactive environment like Jupyter Notebook, by calling relevant classes. In the first step, input data is provided, including the training pattern, supplementary drilling data, historical data, and a set of static data for the hydrodynamic simulator (such as well locations, bottomhole pressures, and fluid properties). Subsequently, a set of geological realizations is generated, which is then used to train the parameterizing model. Once the trained model is obtained, the adaptation process begins. In each iteration, the parameterization module is called, the parameterized realization is passed to the hydrodynamic simulator, which returns simulated data. The geological model parameters are then adjusted to minimize the deviation of the simulated data from the historical data. The adaptation terminates upon meeting a specified condition (e.g., iteration count, accuracy), after which the user receives a geological realization that optimally describes the historical data. The Python programming language was selected for implementing the system, simplifying the core system module—the parameterization module—due to the availability of suitable libraries and ease of neural network construction and training.

The realization generation module is based on the MPSLib library [43]. MPSLib implements several multiple-point statistics algorithms, including the SNESIM method used in this work [70]. The training image and supplementary data are loaded in GSLIB format [2].

The parameterization module is implemented using the PyTorch library [62], which employs a computational graph representation and applies autograd algorithms for computing parameter gradients and performing backpropagation. The loss function is constructed using a pre-trained VGG-16 neural network, available in the PyTorch Hub repository [63]. The Model Transform Net neural network is constructed according to the architecture described in [49]. The Adam optimizer was used for training. The overall parameterization model training process includes the steps listed below.

- Construct a PCA model by calling the realization generation module and performing a singular value decomposition on the resulting matrix of geological realizations. The singular value decomposition algorithm from the Numpy library is used.
- Generate a specified number of PCA realizations with parameters ξ drawn from a standard normal distribution.
- Train the Model Transform Net neural network on the generated realizations using the ADAM algorithm, with a specified batch size and learning rate.
- Save the trained model to a file for subsequent use in the adaptation module.

The Open Porous Media Flow (OPM) hydrodynamic simulator [10] is used. For interaction with it, automatic generation of Eclipse-format files is implemented, forming them based on a template and input parameters (Numpy matrices of porosity and permeability). The system enables parallel execution of the simulator on different parameter sets. The Ecl library [3] is used for reading simulation results and obtaining them in Numpy vector format.

The model adaptation module is based on two alternative optimization algorithms—PSO and SPSA. Non-gradient optimization methods were chosen, as many simulators, including OPM, do not provide derivative calculations with respect to parameters, and implementing this feature is a labor-intensive task.

The particle swarm optimization (PSO) algorithm uses the pyswarms library [4].

The SPSA algorithm implementation involves selecting sequences of coefficients α_n and β_n. The sequences $\alpha_n = \alpha n^{1/5}, \alpha = 0.5$ and $\beta_n = 0.2n^{1/10}$ were selected based on experiments and theoretical results from [73]. Additionally, a condition was added so that the coefficient α is halved, and the best parameter vector found during optimization is used if, at any optimization step, the current function value exceeds the initial value. Without this condition, the algorithm was sensitive to initial data, requiring different α values for each case.

The objective function used is as follows:

$$f(\xi) = \frac{||d_{fopr}(\xi) - d_{obs}||}{\max(d_{obs})} + \sum_i \frac{||d_i(\xi) - d_{obs_i}||}{\max(d_{obs_i})} + \frac{||\xi - \xi_0||}{\max(\xi_0)},$$

where $d_{fopr}(\xi)$ and $d_i(\xi)$ are the simulated production rates obtained for the porosity and permeability values corresponding to the parameter vector ξ. d_{obs} and d_{obs_i} represent the observed production rates (historical field development data).

The first term of the objective function accounts for the deviation of the observed total oil production rate from the simulated rate. The sum over i corresponds to the deviation of production rates for each individual well. The last term in the function acts as a regularizer, ensuring that the target parameter vector does not deviate excessively from the prior ξ_0. Each term is normalized by the maximum value of the corresponding observation vector. This normalization ensures that data for each well is weighted equally, and the minimization algorithm seeks to achieve similarity for each set of observations, regardless of absolute values.

Experiments and Conclusions

Synthetic Geological Model

A binary two-dimensional geological model, consisting of a 92 by 92 grid with a block size of 50x50x10 m, was constructed for system testing and adaptation algorithm comparison. Using a training image from [49], a true realization was generated, shown in Figure 4.3. This realization was not used during the training of the parameterization model.

Five injection wells and four production wells were specified. The well locations were chosen to ensure each production well connects to an injection well via high-permeability channels. The bottomhole pressures of the injection wells were set to allow water from the injection wells to reach the production wells within the simulation period, ensuring a non-trivial dependence between model parameters (porosity/permeability values) and oil production rates.

A simulation period of 7 years with 3-month intervals was specified, resulting in a total of 28 observation points for each production well.

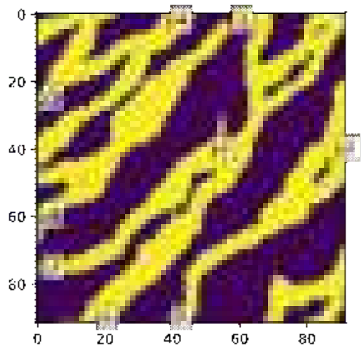

Figure 4.3: Permeability map of the simulated model. High-permeability channels are marked in yellow. Blue markers denote injection wells, and red markers denote production wells.

Figure 4.4 shows the true and observed data for each production well. Observed data were derived from the true data with added noise. Four types of noise were considered: random Gaussian noise $\mathcal{N}(0, 0.02m)$, constant noise $\nu_t^0 = 0.05m$, plus-minus noise $\nu_t^1 = 0.05\,\mathrm{sign}(\sin t) \cdot m$, and irregular noise $\nu_t^2 = 0.03m\,(0.1 \sin t + 2\,\mathrm{sign}(3 - t \mod 5))$, where $t = 1 \ldots 28$, and m is the maximum value of the corresponding observation.

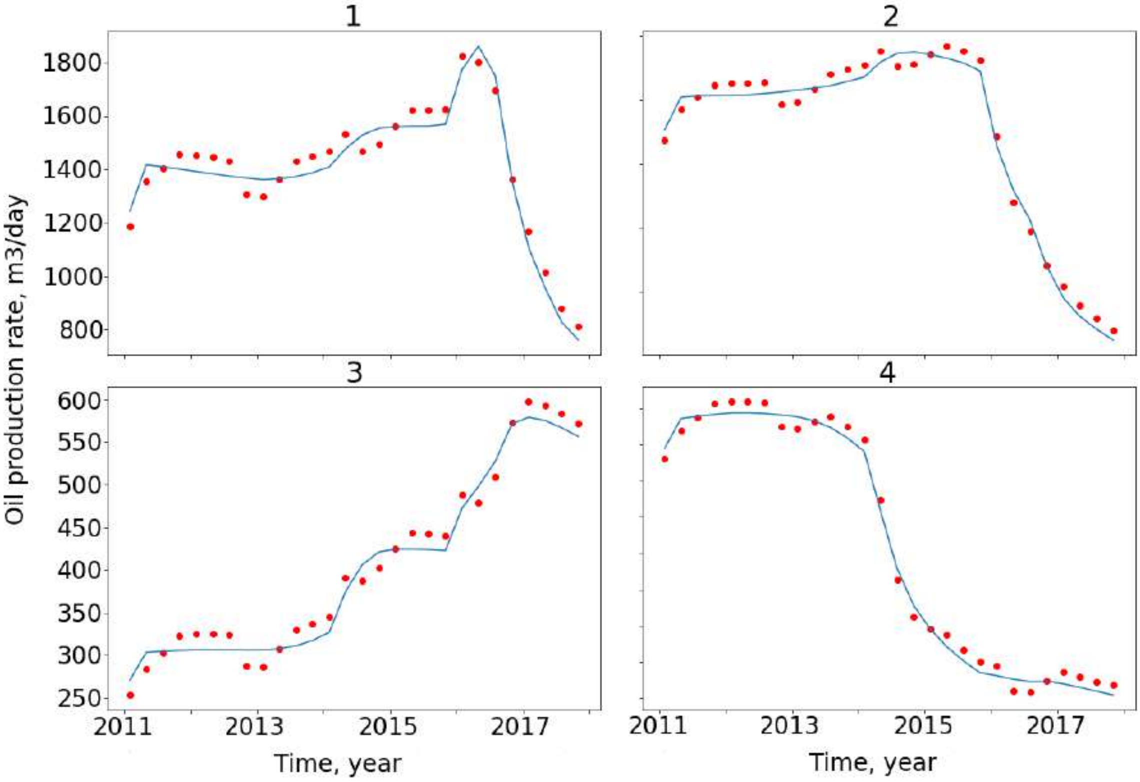

Figure 4.4: Simulated data: oil production rate curves for each production well. Observed data are shown as red dots.

Comparison of Adaptation Methods for Various Types of Noise in Observed Data

A total of 6000 realizations were generated using the training image, and 20,000 PCA realizations were obtained, with a parameter vector size of $l = 170$ (a 50-fold dimensionality reduction). The parameterization model was trained for 50 epochs with a batch size of 32.

Threshold transformation O-PCA with a parameter $\gamma = 0.9$ was applied. Using O-PCA in place of threshold truncation enabled a continuous dependence of realizations on the parameter vector ξ, preventing abrupt changes in the objective function during optimization.

The optimization algorithms used a stopping criterion with an accuracy threshold of 1.2, and a maximum of 2000 objective function evaluations was specified.

A total of 7 posterior realizations were obtained for various prior realizations (initial data). Prior realizations were drawn randomly from a standard normal distribution.

Figure 4.5 shows the model adaptation results, with total oil production rates for all wells across prior and posterior realizations obtained via the PSO and SPSA algorithms. A two-year forecast period, whose data were not used during adaptation, is displayed. The results show a significant reduction in uncertainty, with production rate curves for all realizations approaching the true curve. It can be observed that realizations obtained by the SPSA algorithm exhibit less variance and are closer to the true curve, especially in the forecast period.

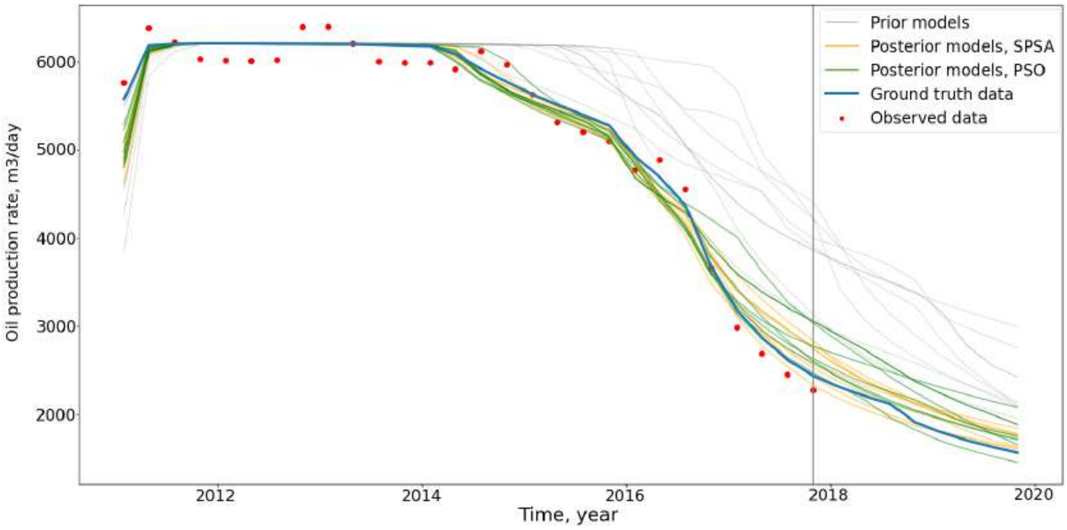

Figure 4.5: Model adaptation results. Observations are marked with red dots, and the true data are represented by the blue curve. Gray curves correspond to data from prior realizations, yellow curves show data for posterior realizations obtained by the SPSA algorithm, and green curves show data obtained by the PSO algorithm. The vertical gray line separates the historical period from the forecast period.

Table 4.1 presents the comparison results for the different types of noise:

Noise Type	Algorithm	mean	stddev	fail
Gaussian Noise	PSO	502	254	2
	SPSA	**439**	332	**2**
Constant Noise	PSO	657	364	4
	SPSA	**473**	317	**2**
Plus-Minus Noise	PSO	908	291	7
	SPSA	**448**	402	**3**
Irregular Noise	PSO	708	415	6
	SPSA	**514**	358	**3**

Table 4.1: Comparison of algorithms for four types of noise. The mean column shows the average number of iterations, the stddev column shows the standard deviation in iteration count, and the fail column shows the number of attempts for which the accuracy threshold was not reached within the maximum number of iterations.

As shown in Table 4.1, the SPSA algorithm achieves the required accuracy level with significantly fewer iterations than PSO on average for all types of noise. SPSA is also equally effective across all noise types. The standard deviation in iteration count is high in all cases due to the use of random realizations as initial data.

Thus, the experiments demonstrate that the SPSA method can yield good and stable results even with non-Gaussian noise in the observed data. It has been shown that SPSA is well-suited for adapting non-Gaussian geological models with various types of observation noise and outperforms particle swarm optimization in terms of convergence speed.

4.0.2 Simulation of the Distributed Clustering Method

We present the results of a software simulation for the method of recovering clusters of states in a multi-agent system based on compressed observations, as described in Section 3.0.1. A synthetic dataset was used for training and testing, based on randomly generated states of 3000 agents divided into either 3 or 10 random clusters. Experiments were conducted with a batch size of 128, using the Adam optimizer with an initial learning rate of 0.003, linearly decreasing to 0.0003. Experiments were conducted to assess the effectiveness of the proposed method, using 10 and 3 clusters. For comparison, clustering results using k-means and randomly assigned cluster centers are also provided. The evaluation metric used was the minimum absolute deviation.

Table 4.2: Comparison of Clustering Methods. Metric calculated on 128,000 randomly generated sets of agent states

Method	10 Clusters	3 Clusters
k-means	0.012 ± 0.01	0.00 ± 0.00
proposed	0.11 ± 0.01	0.12 ± 0.01
random	0.17 ± 0.03	0.30 ± 0.10

Visualizations of the true and predicted centroids, as well as the agent positions for experiments with 3 and 10 clusters, are presented in the figures below.

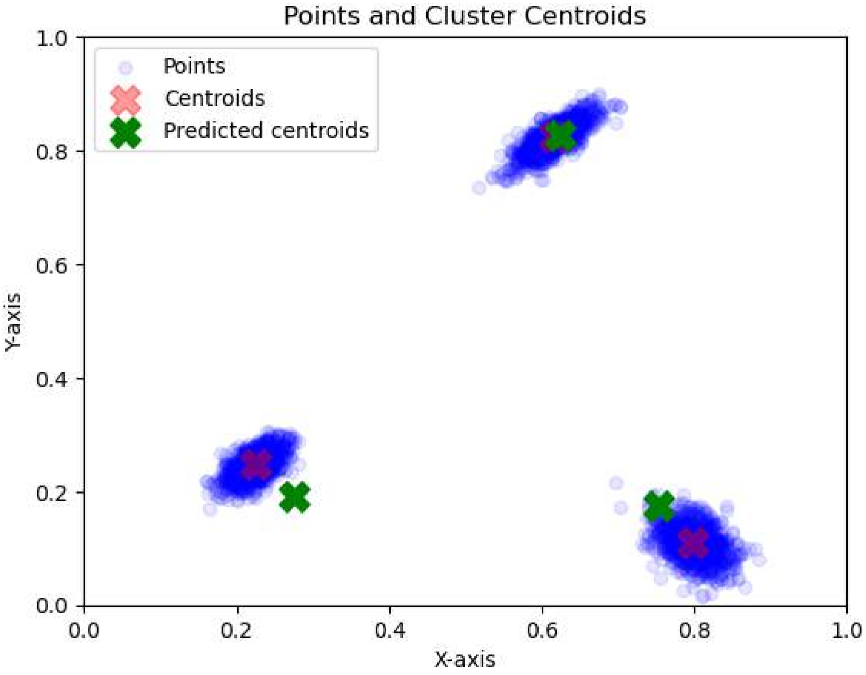

Figure 4.6: Example for 3 Clusters

Conclusions

Currently, the proposed method is significantly less accurate than classical non-distributed clustering methods, such as k-means. Nevertheless, the

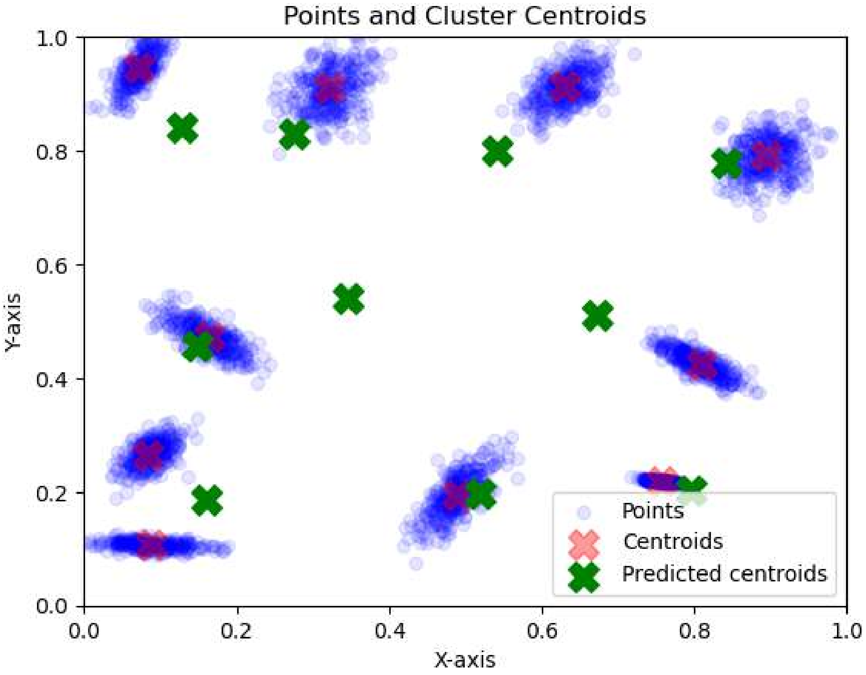

Figure 4.7: Example for 10 Clusters

proposed method enables distributed clustering and can predict cluster parameters in real time, allowing rapid adaptation to changes in the cluster structure of multi-agent systems.

4.0.3 Experiments on Full-Waveform Inversion and Compression of Ultrasonic Tomography Data

We present experimental results for the method outlined in Section 3.0.2. The software implementation was carried out in Python, using the PyTorch library [62]. The reconstruction model was trained for 50 epochs with a batch size of 32. The training dataset consisted of ultrasonic tomography data for four different objects. Experiments were conducted with a compression ratio of 0.2.

Figure 4.8 shows a comparison of ultrasonic data after reconstruction. Visually, all signal features were preserved.

Figure 4.9 compares images obtained through full-waveform inversion across the entire dataset for a single slice. Visually, all essential features, including fine details, were preserved. The Structural Similarity (SSIM) metric value was approximately 0.88. This value may be lower than optimal, as the

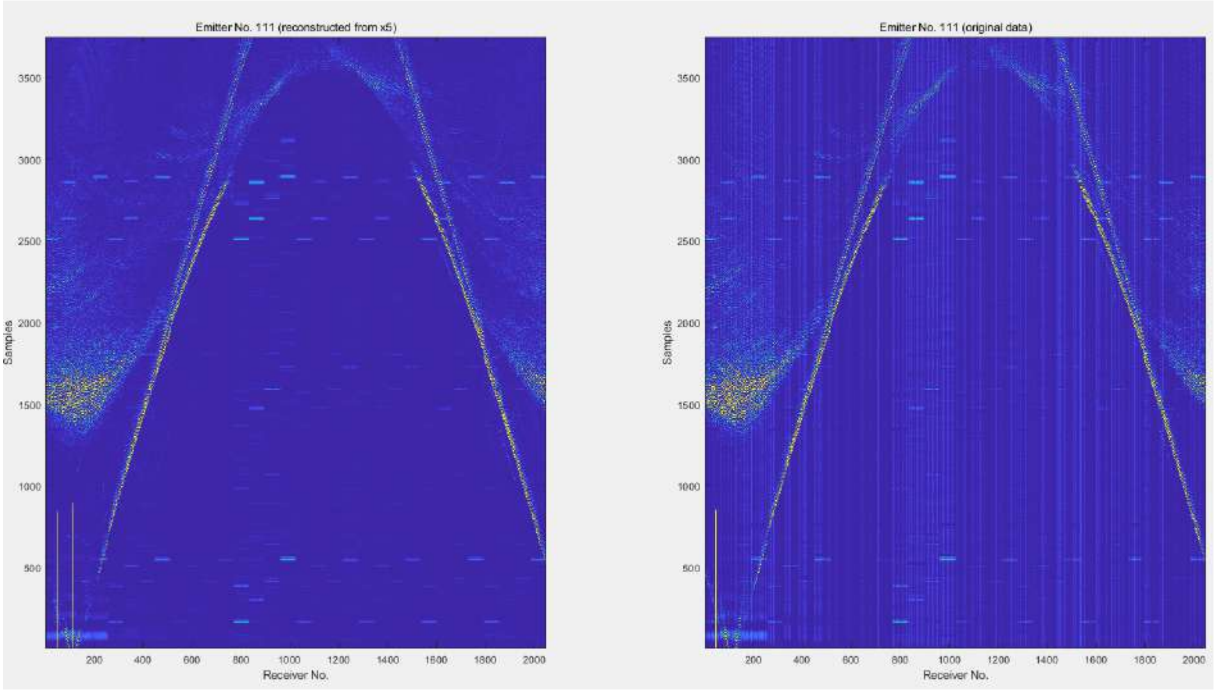

Figure 4.8: Comparison of original data obtained for a single source. Left: reconstructed data after compression. Right: original data

final images differ in intensity in some areas, potentially because the original image was reconstructed from noisy data (vertical stripes), while the reconstructed one was based on noise-free data.

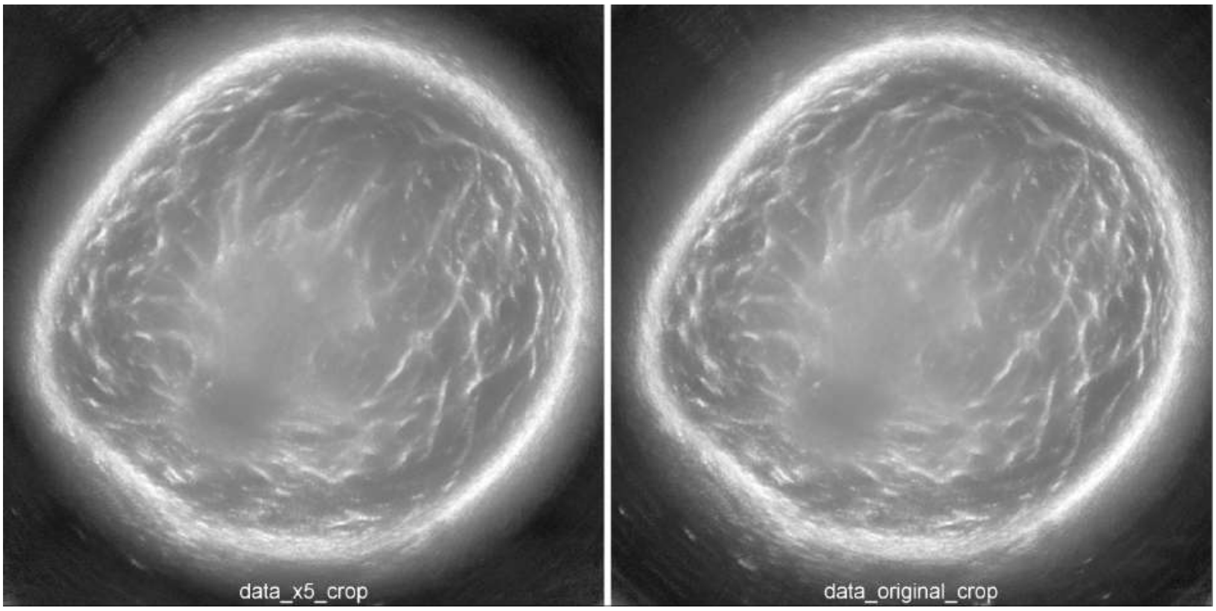

Figure 4.9: Comparison of images reconstructed from original data for one object slice. Left: image from reconstructed data. Right: from original data

Image Reconstruction from Compressed Ultrasonic Data

Further results are presented for direct image reconstruction from compressed ultrasonic data, without preliminary reconstruction of the original data, using the method described in Section 3.0.3. The following metrics were used to evaluate reconstruction quality:

- **PSNR (Peak Signal-to-Noise Ratio)** — measures the ratio between the maximum signal power and noise level, expressed in decibels.
- **SSIM (Structural Similarity Index)** — assesses structural similarity between the reconstructed and original images. Unlike PSNR and MSE, SSIM considers such image aspects as brightness, contrast, and structural changes. This metric better reflects human visual perception of image quality by focusing on structural resemblance.
- **MSE (Mean Squared Error)** — measures the mean squared deviation between pixels of two images (sound velocities).

Compression Ratio	**PSNR (dB)** $\uparrow$	**SSIM** $\uparrow$	**MSE** $\downarrow$
Uncompressed	54.8	0.866	8.12
2x	54.9	0.862	8.04
5x	53.9	0.835	10.03
10x	53.4	0.810	11.47
15x	52.4	0.772	14.37
20x	52.5	0.769	14.12

Table 4.3: Comparison of the quality of reconstructed images from compressed data with varying compression ratios, relative to the original image.

As seen in Table 4.3, reconstruction quality declines only slightly as the compression ratio increases, remaining high for compression ratios of 2-10 times. The object images in Figure 4.10 show that all features of the original image are preserved. Notably, artifacts present in the original full-waveform inversion method (concentric circles) are intensified with compression, but all image details are retained (including fine features). Artifact intensification is minor, as they do not obscure key image elements and are predictable and regular distortions. The primary metric reduction in the table can also be explained by the increased presence of concentric circles rather than by the loss of important details.

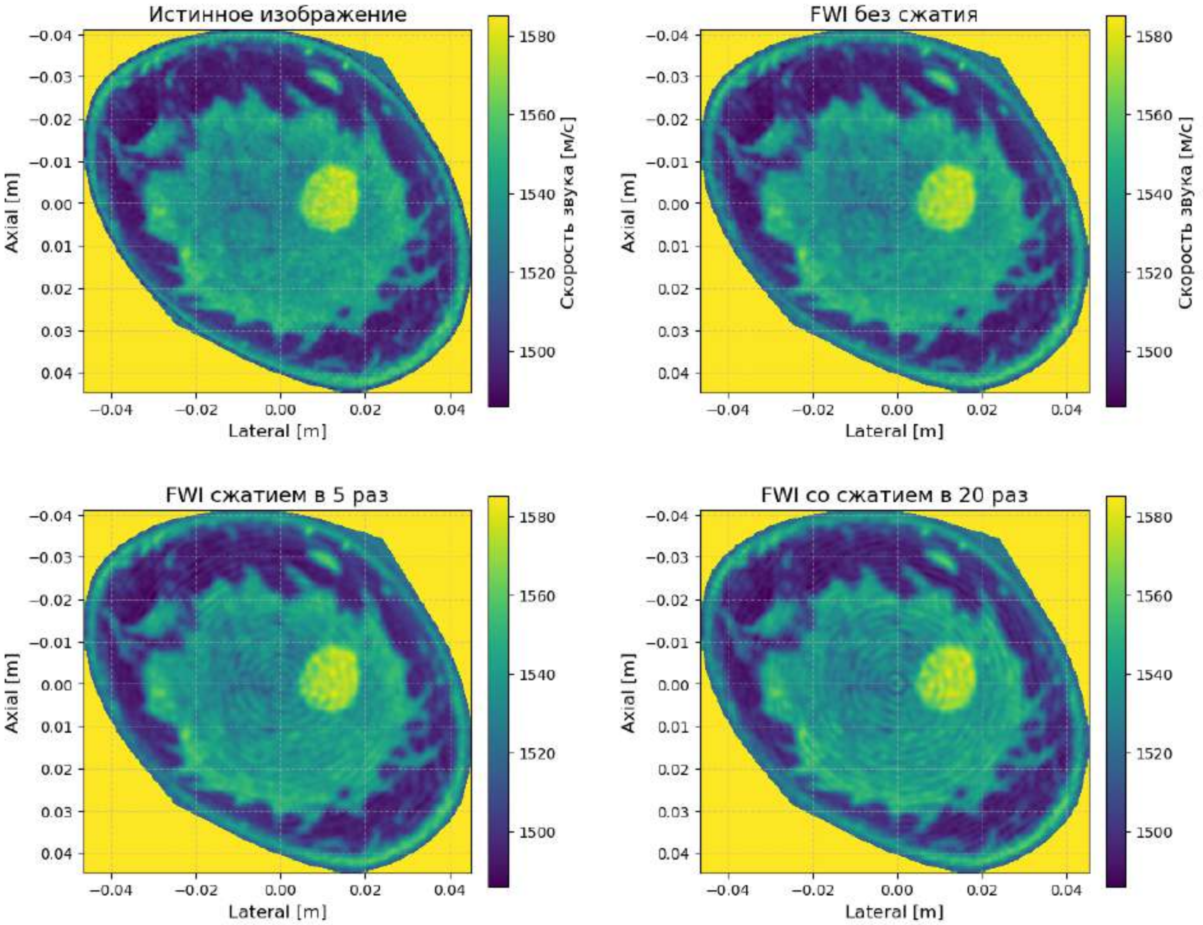

Figure 4.10: Comparison of images reconstructed by full-waveform inversion from compressed data. Top left: original image used to obtain simulated data. Top right: image reconstructed by full-waveform inversion from uncompressed data. Bottom: images reconstructed from data compressed 5 and 20 times (without preliminary recovery)

Conclusions

Thus, the potential for using full-waveform inversion to reconstruct images from compressed ultrasonic data without preliminary recovery, transmission, and storage of data in the original volume has been demonstrated. This is especially important for ultrasonic tomographs, as it enables significant reduction in the amount of transmitted and stored data, allowing for an increase in the number of sources and sensors without requiring higher data transmission bandwidth and without substantial computational cost increase. Additionally, the method can be further adapted for use in learned image reconstruction methods, such as [44, 48, 77]. In this case, it could significantly accelerate the training and execution of such image reconstruction models.

4.0.4 The ℓ_1 Control Problem for Discrete Non-Minimum Phase Systems with Unknown Bounded Disturbances

In control tasks for systems with unknown but bounded disturbances, regularization of parameters is often necessary to stabilize the system, similarly to ℓ_1-regularization methods in compressed sensing. This section examines the problem of synthesizing a sparse controller for a discrete dynamic non-minimum phase system with unknown but bounded disturbances, based on the method presented in [30].

Consider a discrete dynamic system of the form:

$$A(q^{-1})y_t = B(q^{-1})x_t + v_t,$$

where y_t is the system's output signal, x_t is the input signal (control action), and v_t is a disturbance bounded in norm by $|v|_\infty \leq C_v$, with shift operator q^{-1} such that $q^{-1}y_t = y_{t-1}$. The polynomials $A(q^{-1})$ and $B(q^{-1})$ define the system dynamics.

The stabilizing controller is given by:

$$\alpha(q^{-1})u_t = \beta(q^{-1})y_t,$$

where $\alpha(q^{-1})$ and $\beta(q^{-1})$ are polynomials in the shift operator q^{-1}. The goal is to determine coefficients α and β that minimize the maximum deviation of the output signal in the worst-case disturbance scenario. This deviation is characterized by the following control performance metric:

$$J(\alpha(q^{-1}), \beta(q^{-1}), C_v) = \inf_{\alpha(q^{-1}), \beta(q^{-1})} \sup_{\|v\|_\infty \leq C_v} \lim_{t \to \infty} |y_t - \overline{y}_t|,$$

where $\overline{y}_t$ represents the desired system behavior.

Next, we present a theorem from [30], forming the basis of the fractional delay method.

Theorem 3. *Let the polynomial $B(q^{-1})$ have m unstable zeros $\lambda_1, \dots, \lambda_m$, and contain no unit roots. Then, the minimum value of the control performance metric $J(\alpha(q^{-1}), \beta(q^{-1}), C_v)$, corresponding to the minimization of the output signal deviation under worst-case disturbances, is estimated as follows:*

$$\min_{X \in \mathbb{R}_+^m} C_v \sum_{j=1}^{m+r} |F_j(X)| \le J(\alpha(q^{-1}), \beta(q^{-1}), C_v),$$

where $F(X) = W^{-1}(X)G$, and the matrix $W(X)$ and vector G depend on system parameters and unstable zeros $\lambda_1, \ldots, \lambda_m$. Components of matrix $W(X)$ depend on the powers $x_1, x_2, \ldots, x_m$, which represent delay coefficients for the unstable zeros. The problem reduces to minimizing the sum over $X = (x_1, x_2, \ldots, x_m)$, where x_i may take both integer and fractional values.

If the minimum value is achieved at point X_0 with integer components, the polynomials $\alpha(q^{-1})$ and $\beta(q^{-1})$ can be derived as in [30]:

$$\alpha(q^{-1}) = \frac{F(q^{-1})B(q^{-1})}{q^{-r}\prod_{j=1}^{m}(q^{-1} - \lambda_j)},$$

$$\beta(q^{-1}) = \frac{(A(q^{-1})F(q^{-1}) - 1)}{q^{-r}\prod_{j=1}^{m}(q^{-1} - \lambda_j)},$$

where

$$F(q^{-1}) = \sum_{j=0}^{r-1} F_j(X_0)q^{-j} + \sum_{j=r}^{m+r-1} F_j(X_0)q^{-\sum_{k=1}^{i-r+1} x_k}.$$

In cases where X_0 includes fractional values, the fractional delay method can be applied [30]. Here, fractional delays x_i are divided into integer and fractional parts:

$$x_i = \tilde{x}_i + d_i,$$

where $\tilde{x}_i = \text{floor}(x_i)$ is the integer part, and $d_i = x_i - \tilde{x}_i$ is the fractional part. Fractional delay filters, such as:

$$\hat{D}(q^{-1}) = (1 - d) + dq^{-1},$$

are used to implement fractional delays, where d represents the fractional delay. In this case, fractional delays can be approximated as follows:

$$q^{-x_i} \approx q^{-\tilde{x}_i}\hat{D}(q^{-1}),$$

allowing fractional delays to be incorporated into the calculation of coefficients $\alpha(q^{-1})$ and $\beta(q^{-1})$.

Below, we present results of regulator synthesis for a non-minimum phase system described by the following equation:

$$y_t - 1.91y_{t-1} + 5.2y_{t-2} = \lambda_1\lambda_2 u_{t-1} - (\lambda_1 + \lambda_2)u_{t-2} + u_{t-3} + \upsilon_t,$$

where $\lambda_1 = 0.5$, $\lambda_2 = 0.7$.

We then perform a comparative analysis of suboptimality levels for two controller design methods: rounding and fractional delay.

1. Calculate matrix $A(X)$ and vector B:

$$A\left(X\right) = \left\| \begin{matrix} 1 & 0 & 0 \\ 1 & 0.5^{x_1} & 0.5^{x_1+x_2} \\ 1 & 0.7^{x_1} & 0.7^{x_1+x_2} \end{matrix} \right\|, \quad B = \begin{pmatrix} 1 \\ 0.743 \\ 0.452 \end{pmatrix}$$

2. Find $F(X) = A^{-1}(X)B$.

3. Calculate the functional minimum:

$$J_{min}(x_1, x_2) = 1 + |f_1(x_1, x_2)| + |f_2(x_1, x_2)| .$$

4. The minimum value $J_{min} = 2.224$ is achieved at $X^{opt} = (2.255, 2.409)$.

5. Determine polynomial α coefficients:

$$\begin{matrix} 1 + \alpha_1\lambda_1^{\lfloor x_1^{opt}\rceil} + \alpha_2\lambda_1^{\lfloor x_1^{opt}+x_2^{opt}\rceil} = a(\lambda_1) \\ 1 + \alpha_1\lambda_2^{\lfloor x_1^{opt}\rceil} + \alpha_2\lambda_2^{\lfloor x_1^{opt}+x_2^{opt}\rceil} = a(\lambda_2) \end{matrix},$$

   where $\lfloor x\rceil$ denotes rounding to the nearest integer. We obtain $\alpha_1 \approx 0.754$ and $\alpha_2 \approx 5.006$.

6. The suboptimality (error) level is calculated as:

$$1 + |\alpha_1| + |\alpha_2| - J_{min} = 2.536.$$

Next, we examine the fractional delay method.

The fractional delays are extracted as follows:

$$d_1 = 0.745 + 0.255q^{-1},\ d_2 = 0.336 + 0.664q^{-1}.$$

The polynomial $\alpha = (\alpha_{11} + \alpha_{12}, \alpha_{21} + \alpha_{22})$ coefficients are found via:

$$\alpha(q^{-1}) = \frac{F(q^{-1})B(q^{-1})}{q^{-r}\prod_{j=1}^{m}(q^{-1} - \lambda_j)},$$

where $\alpha_{11} = -0.855$, $\alpha_{21} = -0.049$ are integer part coefficients, and $\alpha_{12} = -0.293$, $\alpha_{22} = -0.098$ are fractional part coefficients. The suboptimality level in this case is calculated as:

$$1 + |\alpha_{11}| + |\alpha_{12}| + |\alpha_{21}| + |\alpha_{22}| - J_{min} = 0.071.$$

Conclusions

The synthesis method for a suboptimal ℓ_1 controller with fractional delays for discrete non-minimum phase systems with unknown but bounded disturbances significantly improves control accuracy compared to methods based on delay rounding. The use of fractional delays reduces suboptimality and enhances control quality.

The connection of this approach with ℓ_1-regularization methods, traditionally used in compressed sensing, lies in minimizing the ℓ_1 norm to obtain sparse solutions. In compressed sensing, ℓ_1-regularization promotes sparsity in signals recovered from a limited number of measurements. Similarly, in the control problem, ℓ_1-regularization is applied to find an optimal controller with sparse coefficients, minimizing the deviation of the system output from the desired behavior in the presence of unknown but bounded disturbances. Sparsity in this context reduces controller complexity and enhances robustness, ensuring minimal worst-case error levels.

4.0.5 A Voice Cloning System with Noise-Robust Speech Synthesis Based on Latent Space Regularization

This section presents a noise reduction method within the compressed latent representation of audio, utilizing self-supervised learning via the DINO (Distillation with No Labels) approach [19]. Unlike traditional noise reduction methods applied in the original audio signal space, such as those in [23, 27, 57], the proposed approach performs denoising in a compressed space, significantly reducing computational costs, handling noise more effectively, and preserving key voice characteristics.

The noise reduction task in the context of voice cloning from noisy audio, where the goal is to copy the speaker's voice to synthesize speech based on a given text, can be formulated as follows. Let y denote the observed noisy audio recording of the speaker (reference), x be the desired clean voice representation, and e the noise present in the reference. The noisy reference audio can then be described by the equation:

$$y = F(x) + \epsilon, \tag{4.1}$$

where x represents the voice characteristics, such as timbre, pitch, and speaking style, to be extracted for voice cloning, and ϵ is arbitrary noise.

Traditional methods assume that the operator F is identical ($F = I$), with the task being to remove the noise e from the audio signal y, after which the preprocessed signal is passed to the encoder to obtain a compressed voice representation. These methods aim to estimate x from y using pre-trained models trained on explicitly labeled noisy data [23,27,57]. Such methods are computationally intensive, as the reference audio must be processed in the original space. Moreover, in real-world conditions, such as those involving various types of background noise, the noise distribution may be unknown, and the noise itself may be complex and change depending on the recording environment. This makes traditional denoising methods, which process audio in the original space, less effective and risks the loss of important voice characteristics required for accurate cloning.

Instead of noise removal in the original audio space, the work in [25] proposes a method for denoising in the compressed space. The Speaker Encoder model, based on the Context Aware Masking (CAM)++ architecture [16], is used to obtain latent representations. Regularization is implemented through joint training of the encoder with a speech synthesis model (a decoder trained via a generative adversarial approach), with additional regularization of the compressed voice representation based on the DINO approach [19]. DINO enables model training without labeled data, encouraging the generation of embeddings with certain properties that facilitate noise removal and extraction of relevant information necessary for effectively capturing the voice style and identity.

The DINO approach uses a teacher-student method, where the student is trained to predict the teacher's outputs for different versions of the same input (with two randomly chosen and augmented audio segments as input). The DINO loss function is defined as the cross-entropy between the teacher output distribution P_T and the student output distribution P_S:

$$\mathcal{L}_{\text{DINO}} = -\sum_{i=1}^{K} \sigma\left(\frac{P_T(x_{a1})_i - C}{\tau}\right) \log \sigma\left(\frac{P_S(x_{a2})_i}{\tau}\right), \tag{4.2}$$

where σ is the softmax function, C is a vector of means calculated as the exponentially weighted average of the previous teacher outputs, and τ is a temperature constant for stabilizing model training. Here, x_{a1} and x_{a2} are two different segments of the input signal x, augmented with random noise.

With DINO self-supervised learning, the speaker encoder model is trained to generate compressed representations that are invariant to noise and other non-essential variations while preserving key voice and speech style characteristics. This allows for efficient noise removal in the latent representation, enabling the decoder to reproduce a clean voice. Similar to how ℓ_1-regularization in compressed sensing promotes sparse solutions, DINO encourages the model to create embeddings with specific useful properties based on unlabeled data, facilitating noise reduction while retaining critical information.

System Architecture

The implementation was carried out in Python using the PyTorch machine learning framework [62].

The implemented system consists of four modules (see Fig. 4.11):

S2U (Speech-to-Unit): A pre-trained module based on the HuBERT model [32] that converts input speech into a sequence of discrete representations.

T2U (Text-to-Unit): A module based on the mBART model [55], which converts text into the same representation space as the S2U module, learning to predict HuBERT outputs from text.

Speaker Encoder: A model based on the CAM++ architecture, trained jointly with the speech synthesis model using DINO regularization to produce voice representations in a compressed space that are noise-robust and maintain speech style.

U2S (Unit-to-Speech): A speech synthesis model based on the VITS architecture [47], which takes discrete representations from the S2U module

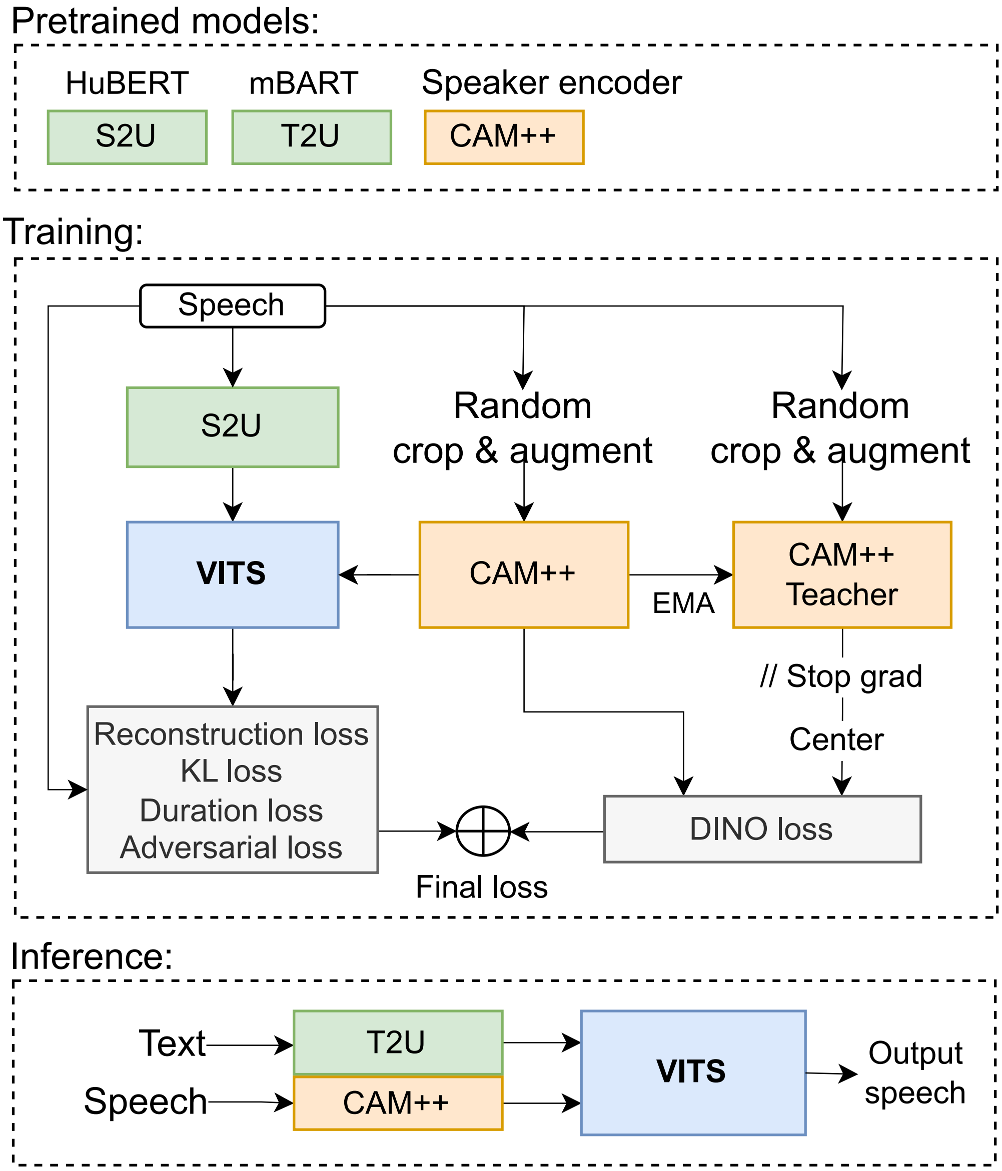

Figure 4.11: Architecture of the proposed DINO-VITS method.

(during training) or the T2U module (to obtain an encoded text representation during inference) and the compressed voice representation to synthesize speech with a voice similar to that in the reference audio.

Experiments and Results

Experiments were conducted to evaluate synthesis quality and voice similarity retention in the presence of noise in the reference audio. The proposed DINO-VITS method showed significant improvements compared to baseline methods such as YourTTS [75] and BYOL-A [66].

To empirically verify the hypothesis that the proposed approach enhances the ability to encode style in reference embeddings, an additional emotion recognition experiment was performed based on the compressed voice representation. Results showed a 9% increase in emotion recognition accuracy, indicating that joint training with DINO regularization preserves speech style information in the latent space more effectively than baseline methods.

Subjective evaluation results of naturalness and voice similarity are presented in Table 4.4.

Table 4.4: Comparison of the proposed DINO-VITS system with YourTTS, DEMUCS denoiser (preliminary denoising in the original space) + YourTTS, and BYOL-A for clean and noisy reference audio. Metrics are scored from 1 to 5. For each model, 210 audio samples were used, each evaluated by 10 annotators from the Yandex Toloka crowdsourcing platform.

	Speech Naturalness		Speech Similarity	
	Clean Data	Noisy Data	Clean Data	Noisy Data
Original Data	4.68 ± 0.03	-	3.94 ± 0.07	-
Proposed Method	**4.00 ± 0.05**	**3.55 ± 0.10**	**3.85 ± 0.08**	**3.52 ± 0.08**
YourTTS	3.96 ± 0.05	3.11 ± 0.11	3.33 ± 0.08	3.20 ± 0.08
YourTTS+Denoiser	-	3.28 ± 0.10	-	3.35 ± 0.08
BYOL-A	-	1.85 ± 0.09	-	1.89 ± 0.07

As shown in the table, the implemented system outperforms baseline methods on all metrics, particularly in noisy conditions, even compared to the YourTTS+Denoiser variant based on preliminary noise removal in the original space (audio spectrograms).

4.0.6 Discussion

Chapter Four presents experimental results confirming the effectiveness of the proposed methods for solving inverse modeling problems in compressed space. The experiments validated the practical significance of the proposed methods, demonstrating that using compressed data representations allows for a substantial reduction in data volume, avoids the need for preliminary recovery in task solutions, enhances noise robustness, and maintains high solution accuracy.

Conclusion

The primary scientific results obtained from the completion of the research objectives are as follows:

- An adaptation algorithm for sparse geological models to historical observed data was developed based on the stochastic optimization method SPSA. The method operates effectively under nearly arbitrary bounded disturbances and is applicable to non-Gaussian geological models (Chapter 2).

- Based on the theory of compressed sensing, methods were developed for efficient data acquisition and reconstruction using deep learning for ultrasonic computed tomography, full-waveform inversion in the compressed space of ultrasonic data, and a decentralized clustering method in multi-agent systems with direct cluster prediction from compressed observations, employing a local voting protocol for the exchange of compressed information. These methods significantly reduce the volume of data collected, stored, processed, and transmitted while maintaining high accuracy for the respective tasks (Chapter 3).

- A noise-robust speech synthesis method was developed based on regularization in the compressed voice representation space. Experiments demonstrated high resilience to various types of noise and preservation of essential voice characteristics (Section 4.0.5). Additionally, the effectiveness of the ℓ_1 control method for synthesizing sparse controls in discrete non-minimum phase systems under unknown but bounded disturbances was investigated (Section 4.0.4).

- The proposed methods were implemented in software, and experiments were conducted to confirm their effectiveness (Chapter 4).

Bibliography

[1] https://www.software.slb.com/products/petrel/petrel-reservoir-engineering/history-matching-production. — [Online; accessed 22-April-2021].

[2] http://www.gslib.com/gslib$_{h}elp/format.html.^{-} [Online; accessed 22-April-2021]$.

[3] https://github.com/equinor/ecl. — [Online; accessed 22-April-2021].

[4] https://github.com/ljvmiranda921/pyswarms. — [Online; accessed 22-April-2021].

[5] ADMM-CSNet: A deep learning approach for image compressive sensing / Yan Yang, Jian Sun, Huibin Li, Zongben Xu // IEEE transactions on pattern analysis and machine intelligence. — 2018. — Vol. 42, no. 3. — P. 521–538.

[6] Adaptive Distributed Cluster Flow Control for a Group of Autonomous Robots / V Erofeeva, V Ershov, O Granichin et al. // IFAC-PapersOnLine. — 2023. — Vol. 56, no. 2. — P. 8690–8695.

[7] Ali Rehman. Open-source full-waveform ultrasound computed tomography based on the angular spectrum method using linear arrays // Medical Imaging 2022: Ultrasonic Imaging and Tomography / Ed. by Nick Bottenus, Nicole V. Ruiter ; International Society for Optics and Photonics. — Vol. 12038. — SPIE, 2022. — P. 120380R. — URL: https://doi.org/10.1117/12.2601257.

[8] Anwar Saeed, Barnes Nick. Real Image Denoising With Feature Attention // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). — 2019. — P. 3155–3164.

[9] Application of Particle Swarms for History Matching in the Brugge Reservoir / Linah Mohamed, Michael Christie, Vasily Demyanov et al. — Vol. 6. — 2010. — 09.

[10] Atgeirr Flø Rasmussen, Tor Harald Sandve, Kai. The Open Porous Media Flow reservoir simulator // Computers and Mathematics with Applications. — 2021. — Vol. 81. — P. 159 – 185. — Development and Application of Open-source Software for Problems with Numerical PDEs.

[11] Basis Pursuit / Shaobing Chen, D. Donoho, Iain Johnstone, Michael Saunders. — 1996. — 03.

[12] Beck Amir, Teboulle Marc. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems // SIAM Journal on Imaging Sciences. — 2009. — Vol. 2, no. 1. — P. 183–202.

[13] Blumensath Thomas, Davies Mike E. Iterative hard thresholding for compressed sensing // Applied and computational harmonic analysis. — 2009. — Vol. 27, no. 3. — P. 265–274.

[14] Boiarov A. Granichin O. Granichina O. Simultaneous perturbation stochastic approximation for few-shot learning // Proc. of the 2020 European Control Conference, Saint Petersburg, Russia. — 2020. — 05. — P. 350–355.

[15] Bredies Kristian, Lorenz Dirk A. Linear convergence of iterative soft-thresholding // Journal of Fourier Analysis and Applications. — 2008. — Vol. 14, no. 5. — P. 813–837.

[16] CAM++: A Fast and Efficient Network for Speaker Verification Using Context-Aware Masking / H. Wang, S. Zheng, Y. Chen et al. // Proc. INTERSPEECH 2023. — 2023. — P. 5301–5305.

[17] Candes Emmanuel J, Romberg Justin K, Tao Terence. Stable signal recovery from incomplete and inaccurate measurements // Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences. — 2006. — Vol. 59, no. 8. — P. 1207–1223.

[18] Candès Emmanuel, Romberg Justin, Tao Terence. Stable Signal Recovery from Incomplete and Inaccurate Measurements // Communications on Pure and Applied Mathematics. — 2006. — 08. — Vol. 59.

[19] A Comprehensive Study on Self-Supervised Distillation for Speaker Representation Learning / Zh. Chen, Y. Qian, B. Han et al. // SLT Workshop. — 2023. — P. 599–604.

[20] Compressed Sensing Image Reconstruction Based on Convolutional Neural Network / Yuhong Liu, Shuying Liu, Cuiran Li, Danfeng Yang // International Journal of Computational Intelligence Systems. — 2019. — 01. — Vol. 12. — P. 873.

[21] ConvCSNet: A Convolutional Compressive Sensing Framework Based on Deep Learning / Xiaotong Lu, Weisheng Dong, Peiyao Wang et al. — 2018. — 01.

[22] Convolutional Neural Networks for Noniterative Reconstruction of Compressively Sensed Images / Suhas Lohit, Kuldeep Kulkarni, Ronan Kerviche et al. // IEEE Transactions on Computational Imaging. — 2018. — Vol. 4, no. 3. — P. 326–340.

[23] Cross-lingual Prosody Transfer for Expressive Machine Dubbing / J. Swiatkowski, D. Wang, M. Babianski et al. // Proc. Interspeech 2023. — 2023. — P. 4838–4842.

[24] DAGAN: Deep De-Aliasing Generative Adversarial Networks for Fast Compressed Sensing MRI Reconstruction / Guang Yang, Simiao Yu, Hao Dong et al. // IEEE Transactions on Medical Imaging. — 2018. — Vol. 37, no. 6. — P. 1310–1321.

[25] DINO-VITS: Data-Efficient Zero-Shot TTS with Self-Supervised Speaker Verification Loss for Noise Robustness / Vikentii Pankov, Valeria Pronina, Alexander Kuzmin et al. // Interspeech 2024. — 2024. — P. 697–701.

[26] DR2-Net: Deep Residual Reconstruction Network for Image Compressive Sensing / Hantao Yao, Feng Dai, Shiliang Zhang et al. // Neurocomputing. — 2019. — 05. — Vol. 359.

[27] Data Efficient Voice Cloning from Noisy Samples with Domain Adversarial Training / J. Cong, S. Yang, L. Xie et al. // Proc. Interspeech 2020. — 2020. — P. 811–815.

[28] Deep ADMM-Net for compressive sensing MRI / Jian Sun, Huibin Li, Zongben Xu et al. // Advances in neural information processing systems. — 2016. — Vol. 29.

[29] Deep Generative Adversarial Networks for Compressed Sensing Automates MRI / Morteza Mardani, Enhao Gong, Joseph Cheng et al. — 2017. — 05.

[30] Design of ℓ_1 New Suboptimal Fractional Delays Controller for Discrete Non-Minimum Phase System under Unknown-but-Bounded Disturbance / Dmitrii Ivanov, Oleg Granichin, Vikentii Pankov, Zeev Volkovich // Mathematics. — 2022. — Vol. 10, no. 1. — P. 69.

[31] Detection of specific areas and densities for ultrasound tomography / Victoria Erofeeva Oleg Granichin Anna Leonova, Vasilisa Galyamina Kseniya Gonta Vikentiy Pankov, Munira Tursunova Mingyue Ding Ming Yuchi, Mingyue Ding Ming Yuchi // CYBERNETICS AND PHYSICS. — 2019.

[32] De'hubert: Disentangling Noise in a Self-Supervised Model for Robust Speech Recognition / D. Ng, R. Zhang, J.Q. Yip et al. // ICASSP. — 2023. — P. 1–5.

[33] Dubrule Olivier. Introducing More Geology in Stochastic Reservoir Modelling // Geostatistics Tróia '92: Volume 1 / Ed. by Amilcar Soares. — Dordrecht : Springer Netherlands, 1993. — P. 351–369. — ISBN: 978-94-011-1739-5. — URL: https://doi.org/10.1007/978-94-011-1739-5$_2$9.

[34] Robust Scheme For Inversion of Seismic And Production Data For Reservoir Facies Modeling. — Vol. All Days of SEG International Exposition and Annual Meeting, 2009. — 10. — SEG-2009-2432. https://onepetro.org/SEGAM/proceedings-pdf/SEG09/All-SEG09/SEG-2009-2432/1783086/seg-2009-2432.pdf.

[35] Efficient real-time reservoir management using adjoint-based optimal control and model updating / P. Sarma, L. Durlofsky, K. Aziz, W. H. Chen // Computational Geosciences. — 2006. — Vol. 10. — P. 3–36.

[36] Efficient training-image based geostatistical simulation and inversion using a spatial generative adversarial neural network / E. Laloy, R. Hérault, D. Jacques, N. Linde // ArXiv. — 2017. — Vol. abs/1708.04975.

[37] Figueiredo Mário, Nowak Robert, Wright Stephen. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems // Selected Topics in Signal Processing, IEEE Journal of. — 2008. — 01. — Vol. 1. — P. 586 – 597.

[38] Foucart S., Rauhut H. A Mathematical Introduction to Compressive Sensing. Applied and Numerical Harmonic Analy-

sis. — Springer New York, 2013. — ISBN: 9780817649487. — URL: https://books.google.am/books?id=zb28BAAAQBAJ.

[39] Fully-Convolutional Measurement Network for Compressive Sensing Image Reconstruction / Xuemei Xie, Jiang Du, Chenye Wang et al. // Neurocomputing. — 2017. — 11. — Vol. 328.

[40] A Stochastic Optimization Algorithm for Automatic History Matching. — Vol. All Days of SPE Annual Technical Conference and Exhibition, 2004. — 09. — SPE-90065-MS. URL: https://doi.org/10.2118/90065-MS.

[41] Granichin Oleg, Amelina Natalia. Simultaneous Perturbation Stochastic Approximation for Tracking Under Unknown but Bounded Disturbances // IEEE Transactions on Automatic Control. — 2015. — Vol. 60, no. 6. — P. 1653–1658.

[42] H. Rezatofighi S., N. Tsoi, J. Gwak. DeepSetNet: Predicting sets with deep neural networks // 2017 IEEE International Conference on Computer Vision (ICCV) / IEEE. — 2017. — P. 5257–5266.

[43] Hansen Thomas Mejer, Vu Le Thanh, Bach Torben. MPSLIB: A C++ class for sequential simulation of multiple-point statistical models // SoftwareX. — 2016. — Vol. 5. — P. 127–133. — URL: https://www.sciencedirect.com/science/article/pii/S2352711016300164.

[44] He Qinglong, Wang Yanfei. Reparameterized full-waveform inversion using deep neural networks // Geophysics. — 2021. — Vol. 86, no. 1. — P. V1–V13.

[45] Image Compressed Sensing Using Convolutional Neural Network / Wuzhen Shi, Feng Jiang, Shaohui Liu, Debin Zhao // IEEE Transactions on Image Processing. — 2020. — Vol. 29. — P. 375–388.

[46] Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network / E. Laloy, Romain H'erault, J. Lee et al. // Advances in Water Resources. — 2017. — Vol. 110. — P. 387–405.

[47] Kim Jaehyeon, Kong Jungil, Son Juhee. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech // Proceedings of the 38th International Conference on Machine Learning. — Vol. 139 of Proceedings of Machine Learning Research. — PMLR, 2021. — 18–24 Jul. — P. 5530–5540.

[48] Lewis Winston, Vigh Denes. Deep learning prior models from seismic images for full-waveform inversion // SEG International Exposition and Annual Meeting / SEG. — 2017. — P. SEG–2017.

[49] Liu Yimin, Sun Wenyue, Durlofsky Louis. A Deep-Learning-Based Geological Parameterization for History Matching Complex Models // Mathematical Geosciences. — 2019. — 03. — Vol. 51.

[50] Machidon Alina, Pejovic Veljko. Deep Learning Techniques for Compressive Sensing-Based Reconstruction and Inference – A Ubiquitous Systems Perspective. — 2021. — 05.

[51] Majumdar Angshul. Real-time Dynamic MRI Reconstruction using Stacked Denoising Autoencoder // ArXiv. — 2015. — Vol. abs/1503.06383.

[52] Mallat S.G., Zhang Zhifeng. Matching pursuits with time-frequency dictionaries // IEEE Transactions on Signal Processing. — 1993. — Vol. 41, no. 12. — P. 3397–3415.

[53] Mousavi Ali, Baraniuk Richard. Learning to invert: Signal recovery via Deep Convolutional Networks. — 2017. — 03. — P. 2272–2276.

[54] Mousavi Ali, Dasarathy Gautam, Baraniuk Richard. DeepCodec: Adaptive Sensing and Recovery via Deep Convolutional Neural Networks. — 2017. — 07.

[55] Multilingual Denoising Pre-training for Neural Machine Translation / Y. Liu, J. Gu, N. Goyal et al. // Transactions of the Association for Computational Linguistics. — 2020. — 11. — Vol. 8. — P. 726–742.

[56] Needell Deanna, Tropp Joel A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples // Applied and computational harmonic analysis. — 2009. — Vol. 26, no. 3. — P. 301–321.

[57] NoreSpeech: Knowledge Distillation based Conditional Diffusion Model for Noise-robust Expressive TTS / Dongchao Yang, Songxiang Liu, Jianwei Yu et al. // ArXiv. — 2022. — Vol. abs/2211.02448. — URL: https://api.semanticscholar.org/CorpusID:253370556.

[58] O.N. Granichin, Pavlenko D.V. Randomization of data acquisition and ℓ_1-optimization (recognition with compression) // Autom Remote Control 71. — 2010.

[59] Oliver Dean, Chen Yan, Naevdal Geir. Updating Markov chain models using the ensemble Kalman filter // Computational Geosciences. — 2011. — 03. — Vol. 15. — P. 325–344.

[60] Oliver Dean S. Multiple Realizations of the Permeability Field From Well Test Data // SPE Journal. — 1996. — 06. — Vol. 1, no. 02. — P. 145–154. — URL: https://doi.org/10.2118/27970-PA.

[61] Pankov Vikentii, Granichin Oleg. SPSA ALGORITHM FOR HISTORY DATA MATCHING OF COMPLEX NON-GAUSSIAN GEOLOGICAL MODELS // CYBERNETICS AND PHYSICS. — 2022. — Vol. 11, no. 1. — P. 18–24.

[62] PyTorch: An Imperative Style, High-Performance Deep Learning Library / Adam Paszke, Sam Gross, Francisco Massa et al. — 2019. — 12.

[63] PyTorch Hub. — https://pytorch.org/hub/. — 2021. — [Online; accessed 22-April-2021].

[64] R. Qi C., H. Su, K. Mo. PointNet: Deep learning on point sets for 3D classification and segmentation // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2017. — P. 652–660.

[65] ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Random Measurements / Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga et al. — 2016. — 01.

[66] Self supervised learning for robust voice cloning / K. Klapsas, N. Ellinas, K. Nikitaras et al. // Proc. Interspeech 2022. — 2022. — P. 4935–4939.

[67] Solving inverse problems using data-driven models / Simon Arridge, Peter Maass, Öktem Ozan, Carola-Bibiane Schönlieb // Acta Numerica. — 2019. — 05. — Vol. 28. — P. 1–174.

[68] Sparse coding with gated learned ISTA / Kailun Wu, Yiwen Guo, Ziang Li, Changshui Zhang // International Conference on Learning Representations. — 2019.

[69] Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit / David L Donoho, Yaakov Tsaig, Iddo Drori, Jean-Luc Starck // IEEE transactions on Information Theory. — 2012. — Vol. 58, no. 2. — P. 1094–1121.

[70] Strebelle Sebastien. Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics // Mathematical Geology. — 2002. — 01. — Vol. 34. — P. 1–21.

[71] The Ensemble Kalman Filter in Reservoir Engineering a Review / Sigurd I. Aanonsen, Geir Naevdal, Dean S. Oliver et al. // SPE Journal. — 2009. — 08. — Vol. 14, no. 03. — P. 393–412.

[72] Vo Hai, Durlofsky Louis. A New Differentiable Parameterization Based on Principal Component Analysis for the Low-Dimensional Representation of Complex Geological Models // Mathematical Geosciences. — 2014. — 10. — Vol. 46. — P. 775–813.

[73] Volkovich O. Granichin Z., Toledano-Kitai D. Randomized Algorithms in Automatic Control and Data Mining // Springer. — 2015.

[74] Wang Jian, Kwon Seokbeop, Shim Byonghyo. Generalized orthogonal matching pursuit // IEEE Transactions on signal processing. — 2012. — Vol. 60, no. 12. — P. 6202–6216.

[75] YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for Everyone / E. Casanova, J. Weber, C.D. Shulby et al. // ICML. — 2022. — P. 2709–2720.

[76] Zhang Jian, Ghanem Bernard. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — P. 1828–1837.

[77] Zhang Zhendong, Alkhalifah Tariq. Regularized elastic full-waveform inversion using deep learning // Advances in subsurface data analytics. — Elsevier, 2022. — P. 219–250.

[78] A deep unrolling network inspired by total variation for compressed sensing MRI / Xiaohua Zhang, Qiusheng Lian, Yuchi Yang, Yueming Su // Digital Signal Processing. — 2020. — Vol. 107. — P. 102856.

[79] An empirical study of large-scale data-driven full waveform inversion / Peng Jin, Yinan Feng, Shihang Feng et al. // Scientific Reports. — 2024. — Vol. 14, no. 1. — P. 20034.

[80] The optimally designed autoencoder network for compressed sensing / Zufan Zhang, Yunfeng Wu, Chenquan Gan, Qingyi Zhu // EURASIP Journal on Image and Video Processing. — 2019. — 04. — Vol. 2019.

[81] A parallel BOA-PSO hybrid algorithm for history matching / Alan Reynolds, Asaad Abdollahzadeh, David Corne et al. — 2011. — 06. — P. 894–901.

[82] A simple proof of the restricted isometry property for random matrices / Richard Baraniuk, Mark Davenport, Ronald DeVore, Michael Wakin // Constructive approximation. — 2008. — Vol. 28. — P. 253–263.

[83] A sparse autoencoder compressed sensing method for acquiring the pressure array information of clothing / Tao Han, Kuangrong Hao, Yongsheng Ding, Xue-Song Tang // Neurocomputing. — 2017. — 10. — Vol. 275.