ENABLING TECHNOLOGIES FOR PERVASIVE SENSING AND DEEP LEARNING IN GEOTECHNICAL ENGINEERING

BY

OMAR BALTAJI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Civil Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Youssef M. A. Hashash, Chair
Professor Andrew Singer
Professor Jamshid Ghaboussi
Professor Scott M. Olson

ABSTRACT

The observational method has been historically employed to adapt to the numerous uncertainties encountered during geotechnical construction. It is based on measuring geotechnical systems performance, updating estimates of their future performance, and implementing changes to construction activities and design if needed. Underground sensing systems embedded in soil and rock formations and inverse analysis frameworks are powerful tools that allow engineers to better realize the potential of the observational method in the age of pervasive sensing and big data. Nevertheless, these tools suffer from important limitations: below ground sensing systems often rely on wired communication that is costly, vulnerable to damage, and of limited versatility; and inverse analyses are typically performed manually by ad hoc methods or computationally by sophisticated and time-consuming methods. In this study, advances are made in wireless through-soil communication and algorithmic development of the deep-learning-based inverse analysis simulation framework, SelfSim.

A communication system, termed SoilComm, was developed, enabling communicating sensor data wirelessly through soil, and hence the deployment of underground sensors that are more economical, versatile, and robust than wired sensors. This system was tested in laboratory and field environments, and its performance was demonstrated by transmission of piezometer measurements and digital images. The latest prototype achieved a 10-m communication range with a power efficiency estimated to enable its batteries to last for around 3 years of operation.

The deep-learning-based inverse analysis simulation framework, SelfSim (Self Learning Simulation), is re-implemented in a modern computational framework and is highly optimized for speed and computational efficiency, enabling much faster learning of geotechnical systems behavior. The performance of the new SelfSim framework was demonstrated through inverse analyses of simulated and real triaxial laboratory tests. SelfSim run times dropped significantly, completing some analyses in less than 5 minutes, compared to hours or days using prior implementations.

In addition to their employment in geotechnical applications, both tools can potentially be employed in many other fields including agricultural, mining, petroleum, material, and bio engineering.

To Mama, Baba, Miysa & Ibrahim, Razan & Abdurrahman, Ali, and Mohammad

ACKNOWLEDGEMENTS

All praise is due to Allah. Without his help, I would have never made it till this point. I ask Him to accept my efforts during this study and allow it to help people and improve their lives. I ask Him to forgive my shortcomings and help me become a better person.

I would like to thank my advisor, Professor Youssef Hashash, from whom I learned a lot. His hard work, high standards, dedication, balance between development and practicality, are but some of the traits I hope I was able to learn from him. His kindness, mentorship, and patience throughout my journey at UIUC are very much appreciated. I hope him and his family the best! Thank you Professor!

I would also like to thank my Professors, from which I learned a lot: Gholamreza Mesri, Scott Olson, Timothy Stark and others. Your hours of patient and passionate instruction in addition to your friendly and kind interactions made this journey much nicer. Thank you very much!

I would also like to thank my co-advisors and thesis committee members Professor Andrew Singer, Professor Jamshid Ghaboussi and Professor Scott Olson. Your guidance and insights were very valuable in producing and completing this study!

I would also like to thank my wonderful colleagues who taught me a lot, and with whom I spent some of my best moments: Yuamar Basarah, Okan Ilhan, Ozgun Numanoglu, Karim Khatib, Eun Hyun Park, Randa Asmar, Guangchao Xing, Xiaoxue Liu, Michael Musgrove, Joseph Harmon, and others.

I would like to thank my dear dad and mom, and my siblings for their love and patience throughout those long six years. Your calls, your prayers, your kind words in my lows and my highs were so dear to me! I am very blessed to be one of you! I miss you a lot and hope to spend much more time with you!

I would like to thank my lovely wife, who was one of the biggest blessings in my life. Your support, patience, and love, despite my busy schedule and stressful moments are things I shall never forget. With you, I became a stronger person, knowing that a pure loving heart is always there for me no matter what happens.

I would finally like to praise Allah for blessing me with Ibrahim while I was working on the final edits of this thesis. Ibrahim, I pray to Allah to give you a blessed life and make you a source of good and happiness for the humanity! Indeed, He is the Most Merciful, the Most Gracious.

TABLE OF CONTENTS

CHAPTER 1. Introduction 1

CHAPTER 2. The Observational Method: A Control Theory Approach to Geotechnical Construction 6

CHAPTER 3. Through-Soil Communication: Attempts, Challenges, and Prospects 12

CHAPTER 4. Inverse Analysis Frameworks: Properties, Challenges and Prospects 42

CHAPTER 5. Development of a Wireless Through-Soil Communication System 58

CHAPTER 6. Development of a Computationally Efficient Machine-Learning-Based Material Modeling and Inverse Analysis Platform (SelfSim Gen3) 83

CHAPTER 7. Testing of the New SelfSim Code Using Synthetic Data 100

CHAPTER 8. Applying the New SelfSim Code to Ottawa Sand Triaxial Tests 153

CHAPTER 9. Conclusions, Limitations and Recommendations 203

References 206

Appendix A: Additional Photos for the SoilComm Research Project 216

Appendix B: New SelfSim Manual 219

Appendix C: Effect of LS-DYNA Element Formulations on FEM Results 235

CHAPTER 1. Introduction

1.1 Motivation

Dealing primarily with natural (i.e. soil and rock) rather than man-made materials (e.g. concrete and steel), geotechnical engineering is characterized by significant uncertainties. Important sources of geotechnical uncertainties are inherent variability, measurement error, and transformation uncertainty resulting from the idealization and simplification of soil behavior (Phoon and Kulhawy 1999a). Engineers can deal with uncertainty by ignoring it, by being conservative, by using the observational method, or by quantifying it (Christian 2004).

The observational method, used by Terzaghi and formally described by (Peck 1969), is a feedback control method that helps mitigate uncertainty in a system's response. As in other feedback control methods, the observational method relies on performance measurements, taken via sensors, to guide decision making and to update estimations of the future response of the geotechnical system which typically encompasses geomaterials in addition to supporting or supported structures. The need for these tools is greater nowadays, due to the tight performance requirements expected in urban geotechnical construction, especially in urban settings where the construction can be surrounded by towers, bridges, or other sensitive structures (Figure 1-1).

Typically deployed in aggressive underground conditions under high soil and water pressures, underground instrumentation systems suffer from congestion (Figure 1-2) and vulnerability to defects and vandalism, due to their wired connectivity. To reduce the risk of loss of underground sensors, engineers often increase the number of sensors to provide redundancy (Azzam et al. 2011, Arbanas et al. 2013), which comes at significant added costs. A wireless underground through-soil communication system would overcome those vulnerabilities hence increasing the robustness of the embedded sensors and consequently of the application of the observational method.

Updating estimations of geotechnical systems' future response, generated via numerical modeling, based on measured performance is usually done by manually modifying numerical models' parameters to match or approximate the measured performance, which is labor-intensive, ad hoc and of limited predictive value. While optimization techniques were developed to automate this process, these approaches are constrained by the limitations of the employed constitutive models, and often are unable to take advantage of the complete set of acquired measurements. The Self-Learning Simulation framework, that employs neural-network-based constitutive models and a

smart learning algorithm, the auto-progressive algorithm, proved to be a superior inverse analysis framework. The current Self-Learning Simulation framework, however, suffers from speed issues and code bugs which are limiting its usability and hindering its adoption. A fast and efficient code capable of running SelfSim-type analyses is needed. Solving the inverse problem quickly would allow pseudo real-time update of future response estimations which is a very valuable decision-assistance tool and would make site-specific constitutive models feasible and timely, and the technology more practical and accessible to engineers.

1.2 Thesis Research Objectives

Motivated by the need for better tools for underground sensing and inverse analysis, the research presented in this thesis aimed at developing a wireless through-soil communication system that can be practically employed in geotechnical applications. The system should have an adequate size, range, data rate, energy efficiency and cost. This research also aimed at developing a fast and efficient code for running SelfSim-type analyses, then testing this code using synthetic and real data.

1.3 Organization of the Thesis

Chapter 2 describes the observational method, which represents the main framework in which the developed tools play vital roles. It also points to the similarity between the observational method and the control theory methods.

Chapter 3 discusses the background information available on through-soil communication. It describes and challenges facing wired systems and shows the need for wireless systems. It then describes attempts for development of wireless-through soil communication systems and highlights the main challenges encountered during those attempts, and the channel properties causing those challenges. It also points to the use of through-soil communication by animals, by employing low frequency mechanical waves, and to the recent human success in wireless communication in water using mechanical waves as well. Based on this realization, it describes mechanical waves in soils and their treatment in different disciplines. This chapter also presents an introduction to the key terminology and techniques used in wireless communication.

Chapter 4 discusses the work done for developing a new through-soil wireless communication system, starting with defining the design requirements then selecting the communication channel, then designing the communication system and testing setups. It describes the testing and

optimization of the system then the development of a communication system prototype that can be used in real geotechnical projects. The chapter is concluded by a large-scale wireless communication demonstration of the developed system, then a summary of findings and recommendations.

Chapter 5 discusses the background information available on the conventional and machine-learning-based constitutive modeling and inverse analysis frameworks. It thoroughly describes SelfSim and its past applications and limitations.

Chapter 6 discusses the development of the new code for running SelfSim-type analyses. It starts with describing the major bottlenecks of the old code, then moves to describing and the architecture of the new code and the basic functions in each of its modules, while highlighting the choices made and the methods used for optimizing the code. It also presents a new formulation of the artificial neural network constitutive model stiffness matrix that allows faster assembly.

Chapter 7 describes the testing of the new code through a series of inverse analysis of simulated laboratory tests. Chapter 8 describes the use of the new code to learn the behavior of Ottawa sand tested by 12 laboratory tests using conventional and modified triaxial cells. It also presents the attempts at developing a unified model that learns the behavior of all 12 tests.

Chapter 9 lists the main conclusions of this study and the recommendations for future research.

1.4 Figures

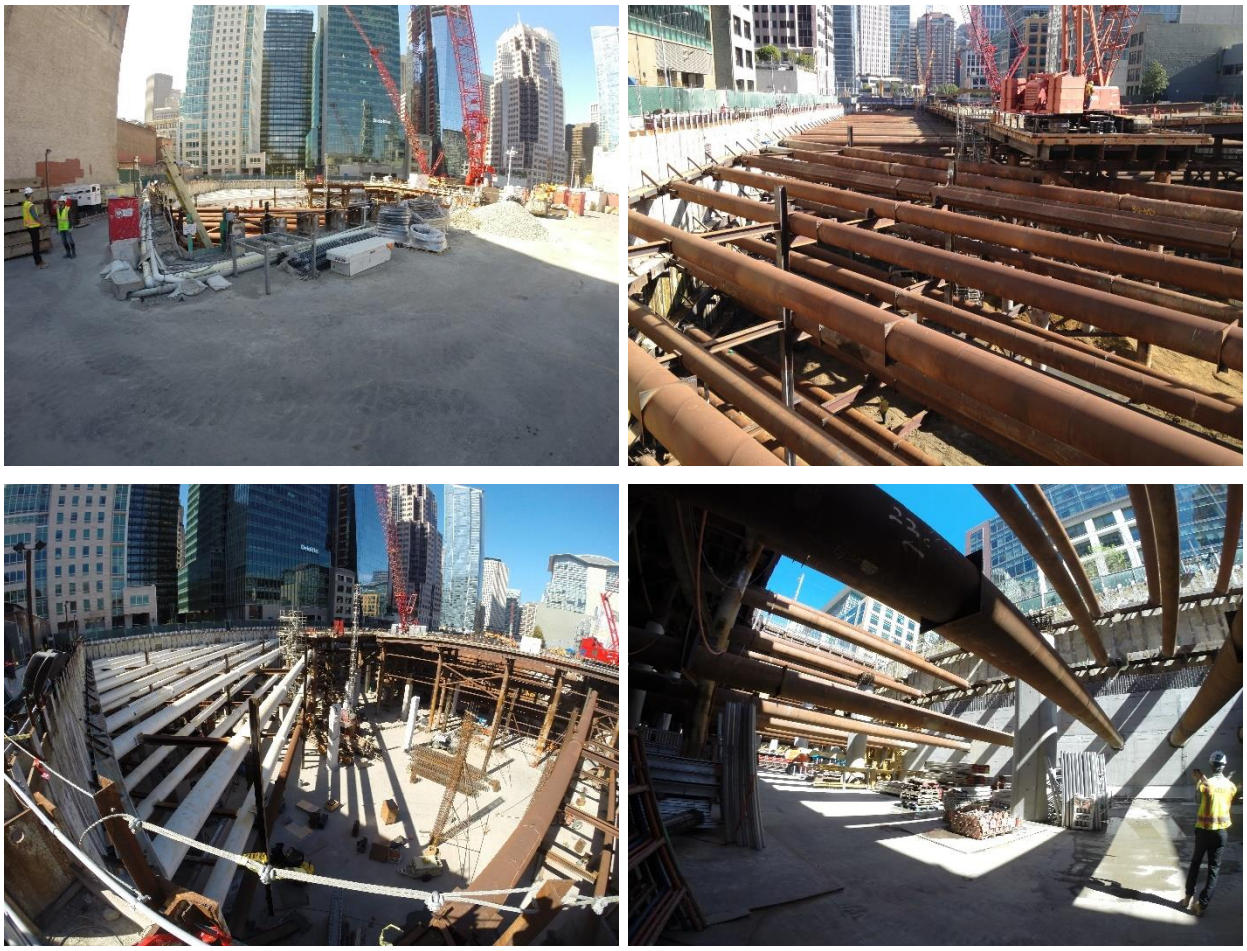


Figure 1-1: Photos of the Transbay Transit Center deep excavation, San Francisco, CA. The excavation is surrounded by towers and buildings from all directions. Source: photo courtesy of Professor Youssef Hashash.



Figure 1-2: Photo showing a large borehole congested with instruments used in the excavation for the Transbay Transit Center, San Francisco, CA. The borehole diameter had to be increased to fit all the instruments. Source: photo courtesy of Professor Youssef Hashash.

CHAPTER 2. The Observational Method: A Control Theory Approach to Geotechnical Construction

2.1 Uncertainty in Geotechnical Engineering

The primary sources of geotechnical uncertainties are inherent variability, measurement error, and transformation uncertainty: the first results primarily from the natural geologic processes that produced and continually modify the soil mass in situ; the second is caused by equipment, procedural-operator, random testing effects, statistical uncertainty or sampling errors that result from limited amounts of information; the third is introduced when field or laboratory measurements are transformed into design soil properties using empirical or other correlation models (Phoon and Kulhawy 1999a). Transformation uncertainty still would be present even for theoretical relationships because of idealizations and simplifications in the theory (Phoon and Kulhawy 1999b). The relative contribution of these three sources to the overall uncertainty in the design soil properties depends on the site conditions, degree of equipment and procedural control, and precision of the correlation model (Phoon and Kulhawy 1999a).

Engineers can deal with uncertainty by ignoring it, by being conservative or by quantifying it. As the discipline developed, it became clear that it was seldom technically or economically possible to design for the most unfavorable possibilities (Christian 2004). Quantifying uncertainty and applying probabilistic methods to them is limited to "known unknowns". The fundamental problem with those methods is the lack of data. Field exploration is expensive, so there is seldom enough data to support meaningful statistical analyses. The presence of "geologic surprises" due to scarcity of data belongs to the category of "unknown unknowns" and cannot be captured through probabilistic methods (Phoon 2017). Engineers can also deal with uncertainty during construction using the observational method.

2.2 Control Theory: The Original Framework for Dealing with Uncertainty

Uncertainty is not exclusive to geotechnical engineering. It is common in all other engineering disciplines. Those disciplines have dealt with uncertainty using control systems.

Control systems are most often based on the principle of feedback, whereby the signal to be controlled is compared to a desired reference signal and the discrepancy is used to compute corrective control action (Doyle et al. 2013). Control systems are needed because of uncertainties in inputs and uncertainties in system response; no mathematical system can precisely model a real physical system; there is always uncertainty (Doyle et al. 2013).

The main elements of a simple feedback control system are illustrated in Figure 2-1. The target performance of a system (e.g., target speed of a car) is compared to the measured performance of the system (e.g., current speed of the car) measured via sensors (e.g., speedometer), and based on the performance error which is the difference between the target and the measurement (e.g., overspeed), a corrective response, opposite to the error, is taken via controllers (e.g., car pedals), hence adjusting the system response. This control loop is repeated over and over till the current system performance matches, or approaches as much as possible, the target system performance.

The control theory, which is the theory for developing control systems, has been developing over centuries, however, automatic control systems have been developed much earlier in history. In fact, humans intuitively use control in learning, balancing themselves and other objects, adjusting their speed or power used in a certain activity, etc.

Bennett (1996) summarized the history of control theory: "*Automatic feedback control systems have been known and used for more than 2000 years; some of the earliest examples are water clocks described by Vitruvius and attributed to Ktesibios (circa 270 B.C.). Some three hundred years later, Heron of Alexandria described a range of automata which employed a variety of feedback mechanisms. Knowledge of the control systems of the Hellenic period was preserved within the Islamic culture that was rediscovered in the West toward the end of the Renaissance. New inventions and applications of old principles began to appear during the 18th century-for example, Rene-Antoine Ferchault de Reamur (1683-1757) proposed several automatic devices for controlling the temperature of incubators. The most significant control development during the 18th century was the steam engine governor. In 1868, James Clerk Maxwell (1831-1879) published his now-famous paper entitled "On Governors". The early years of the 20th century saw the rapid and widespread application of feedback controllers for voltage, current, and frequency regulation, boiler control for steam generation, electric motor speed control, ship and aircraft steering and auto stabilization, and temperature, pressure, and flow control in industries. The second world war developments focused on launching, maneuvering, guidance, and tracking of missiles and space vehicles and was influenced by the advent of the digital computer*".

2.3 The Observational Method, a Realization of the Control Theory

An adaptation of the control theory, extensively used in geotechnical engineering is the observational method. Comparing the observational method with the simple control system shown in Figure 2-1, demonstrates that the observational method is also a feedback control system which

relies on performance measurements (taken via sensors) to guide decision making and to update estimations of the system's future response.

The observational method, was used by Terzaghi and formally described by (Peck 1969):

a. Exploration sufficient to establish at least the general nature, pattern and properties of the soil deposits, but not necessarily in detail.
b. Assessment of the most probable conditions and the most unfavorable conceivable deviations from these conditions. In this assessment geology often plays a major role.
c. Establishment of the design based on a working hypothesis of behavior anticipated under the most probable conditions.
d. Selection of quantities to be observed as construction proceeds and calculation of their anticipated values on the basis of the working hypothesis.
e. Calculation of values of the same quantities under the most unfavorable conditions compatible with the available data concerning the subsurface conditions.
f. Selection in advance of a course of action or modification of design for every foreseeable significant deviation of the observational findings from those predicted on the basis of the working hypothesis.
g. Measurement of quantities to be observed and evaluation of actual conditions.
h. Modification of design to suit actual conditions.

The observational method was employed in many projects (Peck 2001) such as embankments (Wu 2011, Prästings et al. 2014), dam foundations (Spross et al. 2016), tunnels (Powderham 1994, Kovári and Lunardi 2000, Powderham 2004, Moritz and Schubert 2009), excavations (Beadman et al. 2001, Chapman and Green 2004, Yeow et al. 2014), , tailing dams (Jamiolkowski 2014), and hydraulic fill (Choa 1994).

The general philosophy of the method can be traced back through history. An antique example comes from the Phoenicians, who improved the design of a canal after observing that its sides were prone to fail when cut too steeply. The importance of making observations when there are geotechnical uncertainties was also recorded by pioneering Swedish engineers in the early 20th century (Spross 2014). In tunneling, it was quite common to construct a pilot tunnel whose observations served as an additional input for the choice of the construction method and determination of lining type and thickness (Schubert 2008). NATM is in principle an empirical

approach for tunneling, developed from practical experience. It is sometimes viewed as an application of the observational method (Spross 2014).

The comparison between the observational method and control systems, points out to broader framework of the observational method, and suggests the potential of employing some of the modern control theory methods in the modern application of the observational method. For example, controllable elements of the geotechnical systems might be developed to automatically respond to undesirable changes. The amount and rate of the change applied can also benefit from the modern control theory.

2.4 Enhancements to the Observational Method

Contrary to other automated control systems, the application of the observational method was dependent on human's work or input in most of its stages: it relied on hand-written construction updates, manual processing and analysis of instrumentation data, manual adjustment of empirical or numerical models to match observations then estimate the system's future response, which is time- and resource-consuming.

Researchers and professionals attempted to enhance and automate the application of the method. Numerical modelling tools, which benefitted from great advances of their modelling capabilities and computational power, were increasingly used to inform and assist design. Automated monitoring systems were developed and deployed in many projects (Finno automated monitoring of supported excavations). Imaging technologies (e.g., drone-mounted cameras, image processing software) and social media were employed for capturing and documenting construction updates (Fleming et al. 2016).

To automate learning from performance measurements, (Finno and Calvello 2005) introduced what he termed the "adaptive management", in which optimization techniques are used to optimize the material properties in numerical models based on a system's performance measurements collected through sensors. This optimization enabled the numerical models to better capture measurements, hence learning from them and improving estimations of the future system response. Combining laser scanning and webcams, automated and remote instrumentation systems, self-updating numerical models and integration of heterogeneous data using geographic-information system, (Hashash and Finno 2008) introduced an enhanced realization of the observational method termed the "deformation control cycle", illustrated in Figure 2-2. For enhanced learning from performance measurements, Hashash and his colleagues (Hashash et al. 2006, Hashash et al.

2010). suggested modelling materials using artificial neural networks then employing the SelfSim inverse analysis framework. They also demonstrated SelfSim's improved performance over typical optimization techniques.

2.5 Enhancements to the Observational Method in this Thesis

Despite those enhancements, the observational method is not yet fully automated or optimized. In this thesis, two new tools were developed, aiming at a more robust and more automated application of the method. The role of those tools in the application of the observational method is illustrated in Figure 2-2.

The observational method heavily relies on sensor systems. Underground sensors are typically deployed in small boreholes and connected via wires to dataloggers near the surface. This wired connectivity has many downsides, described in detail in section 3.1, with the main two being the vulnerability of those wires to damage and the high cost of: wiring and drilling larger boreholes to accommodate multiple sensors. Losing sensors due wire damage might degrade or even hinder the application of the observational method. To address this a wireless through-soil communication system, termed "SoilComm", was developed, which eliminates the need for wires in underground sensors, hence making them more robust and allowing for significant cost reductions (e.g., by saving on the cost of wiring, and enabling the deployment of multiple sensors within a single small borehole).

While major progress has been made for automating learning from performance measurements then employing this knowledge in estimating future performance, a common downside of the work was the difficulty and low speed of running the inverse analyses required for this learning. The codes for performing inverse analyses were not optimized and, in many cases, suffered from bugs that prevented consistent analysis convergence. This hindered the adoption of those codes in practice, which then had to mostly rely on manual ad hoc adjustments of numerical models to try to match measured performance. To address this, a new computationally efficient inverse analysis framework, termed "SelfSim V3" capable of running SelfSim-type analyses was developed. This framework enables very fast and efficient learning of the material behavior based on performance measurements then using the learned behavior in estimating future performance.

2.6 Figures

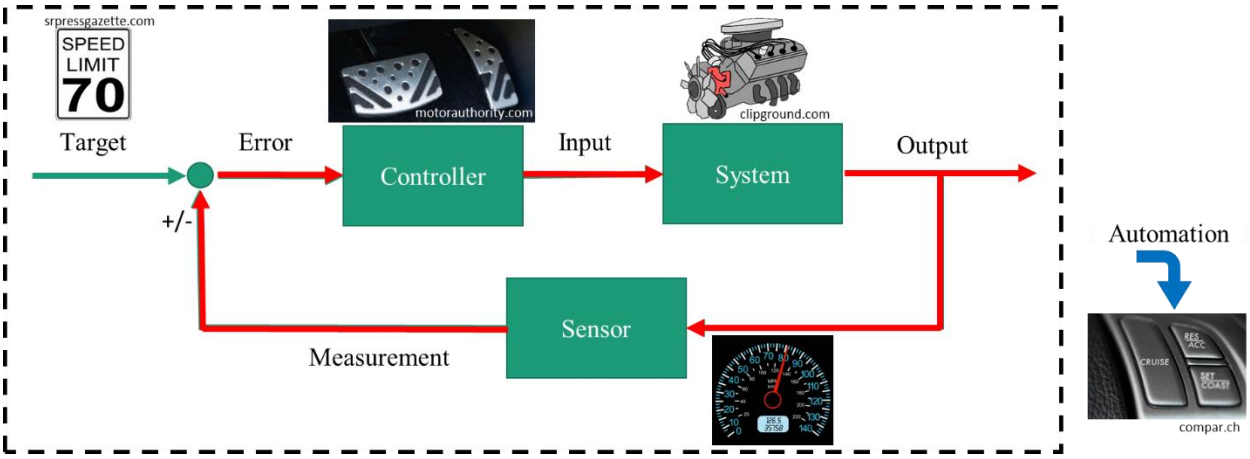


Figure 2-1: Schematic of a simple feedback control system. The figure shows how car speed control, as an example, is in fact achieved by a control loop. The control is automated using via the familiar cruise control key.

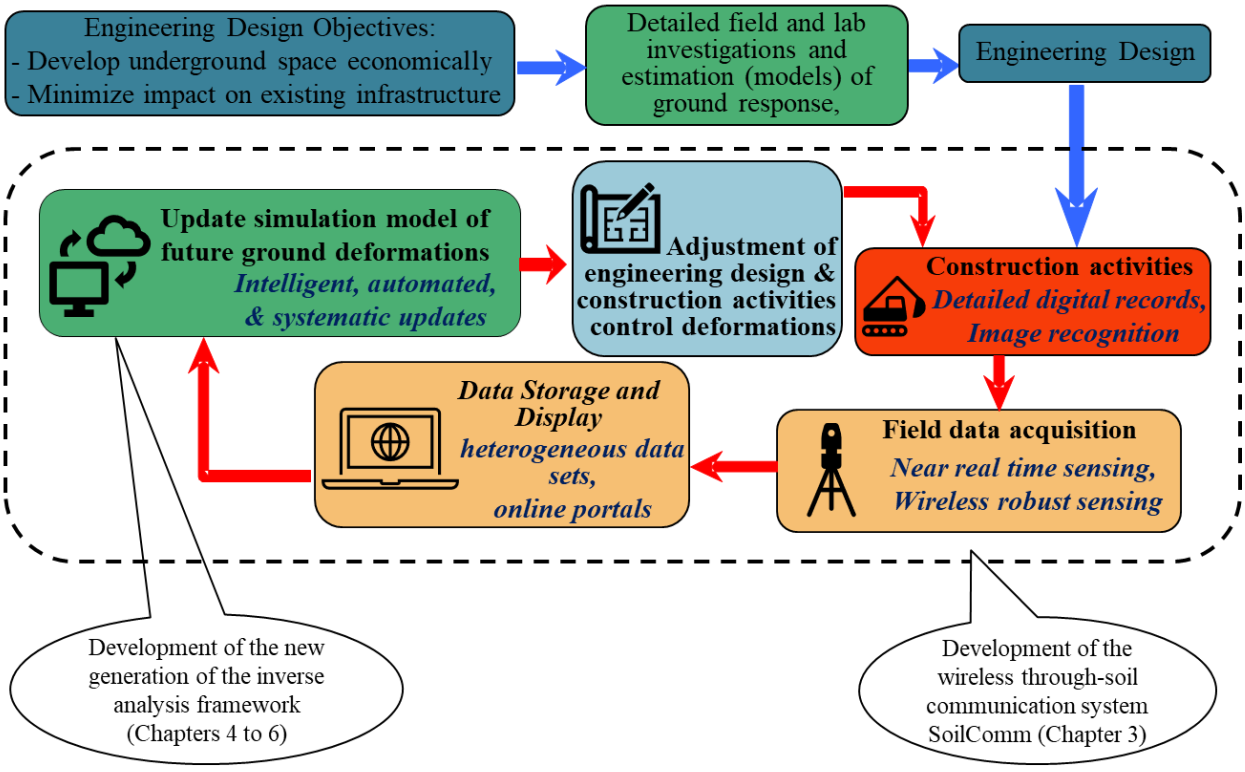


Figure 2-2: Deformation Control cycle during deep geotechnical projects. In this thesis two elements of the cycle were developed, the field data acquisition and the simulation model update. Source: modified from (Hashash and Finno 2008)

CHAPTER 3. Through-Soil Communication: Attempts, Challenges, and Prospects

In geotechnical projects, a lot of sensors are deployed below the ground. Piezometers and accelerometers are typically installed in small-diameter boreholes where they are lowered to their specified depths, then properly locked or sealed in place. Other sensors such as pressure cells and extensometers may be embedded in ground or attached to underground structural elements. This instrumentation can provide early warnings, reveal unknown underground conditions, evaluate critical design assumptions, assess contractor means and methods, minimize damage to adjacent structures, document performance, reduce litigations and advance the state of knowledge (Dunnicliff 1996, Marr 2007). Underground sensors usually communicate their readings via wires that are extended to data loggers placed next to the ground surface.

3.1 Wired Through-Soil Communication Systems

Teal (1990) highlighted the major drawbacks of traditional wired communication systems: cost of wiring, introduction of non-homogeneous sand bedding in earthworks, disruption of earthworks progress, risk of damage during construction and risk of damage after construction. Teal (1990) elaborates on some of those drawbacks: "*The cost of wires is directly dependent on the length required and it is not uncommon for the cost of wiring to exceed that of the sensors themselves. Sand bedding for wires and tubes is generally specified in order to reduce the risk of damage to them. This cannot always be tolerated, for example across the core of an embankment dam where low permeability is required or in coarse rockfill where extra filter layers or fabric would be needed to retain the sand. During fill placement and associated operations there are many occasions when previously-installed sensor wires will be vulnerable to accidental damage. Non-accidental damage can range from inquisitive tinkering by schoolboy trespassers to deliberate attempts to steal the equipment. Damage of this nature is particularly frustrating as arranging for repairs can be protracted and expensive. Where hydraulic or pneumatic sensors are involved there is the risk of de-airing or de-humidifying the system".* Additional causes of damage include lightning damage (BureauofReclamation 2014), and biological and chemical underground hostility. When many sensors are deployed in one borehole, and connected serially, the sensor cluster becomes no more robust than the weakest point in its connecting wires. Congestion of instruments in one borehole may require larger or extra boreholes which adds to the cost of instrumentation. The main causes of wire damage are illustrated in Figure 3-1.

3.2 Need for Through-Soil Wireless Communication Systems

Communication systems that can transmit underground sensors data wirelessly through soil would overcome the listed limitations and vulnerabilities caused by the wired connectivity and may reduce the cost and increase the robustness of underground instrumentation systems. For example, existing through-air electromagnetic wireless systems, when employed in structural projects, have provided cost savings and reduced disruption of operation of existing facilities due to wire failures (Kim et al. 2007).

In addition to those primary benefits, wireless systems would enable deployment of many more sensors in the same number of boreholes. Sensors would occupy their own volume without the volume of pipes needed for wires which make them significantly less intrusive. The elimination of piping is especially useful for dam or embankment cores where pipes and sand filling constituting high permeability paths are unfavorable. The deployment of sensors in challenging locations, such as excavation faces or ahead of tunnel machines cutting faces, or locations with a lot of machinery activity, posing high risks on the wires of wired systems, such as embankments, would be facilitated with the elimination of wires. Those characteristics and case uses will allow for truly robust and pervasive sensing.

Attempts for development of through-soil wireless communication systems were hence explored and assessed. Lessons learned from those attempts will be later used in developing a new communication system, as described in CHAPTER 4.

3.3 Communication Systems Terminology and Techniques

Before describing the attempts for development of wireless through-soil communication systems, it is important to introduce key concepts, terminology, and techniques and that will be the basis for understanding the communication problem, comparing the available communication systems, then designing a better communication system.

A communication system in its simplest forms is composed of a message transmitted through a channel by a transmitter then received by a receiver. The message is encoded in a signal that can travel through the channel. For example, a voice record is encoded into an electromagnetic wave before being sent via an antenna through vacuum.

3.3.1 Signal Impairments

Operating in an ideal channel, the received signal will be identical to the transmitted signal. In real channels, however, many impairments (changes) occur to the signal while propagating through the channel. The major signal impairments of importance to this study are listed below and their effects are illustrated in Figure 3-2.

a) Power Loss: The signal typically loses power due to material and geometric damping. The terms *path loss* and *attenuation* are used to describe this loss:

- Path loss (PL) is the logarithmic ratio of the amplitude of the sent signal (A_0) to the amplitude of the received signal (A):

$$\text{Path Loss } = 20\log\left(\frac{A_0}{A}\right) \text{ decibel (dB)}$$

- Attenuation is the ratio of the path loss to the path length:

$$\text{Attenuation } = \frac{\text{Path Loss}}{\text{Path Length}}$$

where the path length is the length of the shortest path between transmitter and receiver. Attenuation affects the maximum range over which the signal can be detected. Once a signal's amplitude drops below the amplitude of noise (noise level), it becomes hard or impossible to distinguish, and hence it cannot be used for communication.

b) Distortion: A transmitted signal is distorted while travelling through a channel due to the channel effects. Types of *distortion* include linear distortion, nonlinear distortion, and multipath fading (described in (c)). If the amplitude response of the channel is not constant, *amplitude distortion* is said to have occurred (different components of the signal suffer different amounts of attenuation). If the phase response of the channel is not linear (different components of the signal suffer different amounts of delay), *phase distortion* is said to have occurred. In linear distortion, either one or both of amplitude distortion and phase distortion occur. Nonlinear distortion occurs when the relationship between the transmitted signal and the received signal in not linear (i.e., superposition principle is not held).

c) Multi-path Propagation: The signal might travel along different paths due to reflections and refractions; this is termed *multi-path propagation*. This would result in receiving many signals with different delays for every single signal sent which is called *multi-path fading*. The time lag between receiving the earliest arriving signal and the latest arriving signal is called the *delay spread*. Multipath signals may interfere constructively or destructively. This multipath propagation also divides the power of the signal along many paths (Figure 3-3).

d) Interference: If the channel used to transmit signals is used by other sources, the transmitted signals will interfere with signals from other sources, and hence *interference* is said to have happened. This issue is especially significant if the other sources are transmitting at the same operation frequency of the transmitted signal.

e) Noise: *"Noise refers to unwanted, ever-present, random waves that tend to disturb the transmission and processing of signals in a communication system, thus yielding a corrupted version of the transmitted signal. There are several types of noise, such as thermal noise (due to the random motion of electrons in a conductor), shot noise (due to the discrete nature of current flow in electronic devices), and impulse noise (due to natural sources, such as lightning, and man-made sources, such as high-voltage power lines). Noise is generally assumed to be added to the signal."(Grami 2015)*

f) Dispersion: Due to the frequency-dependent attenuation, distortion, multipath propagation, interference and noise, the received signal is likely to have a different balance of frequencies (i.e., frequency content), compared to the transmitted signal, and hence *dispersion* is said to have occurred. Dispersion may lead to *inter-symbol interference* which can cause errors in the received bits.

3.3.2 Communication Techniques

To enable transmitting signals wirelessly despite all the listed impairments and while optimizing energy efficiency and data rate, many techniques were developed within the fields of signal processing and communication engineering. Some of those techniques deal with the methods of efficiently encoding the signal with the information (preprocessing of the transmitted signal), and other techniques deal with the channel effects incurred by the signal during propagation (processing of the received signal). Following is a brief description of the main techniques of interest to this study: *modulation, equalization,* and *error detection and correction.*

Modulation

Modulation is the process of modifying one or more of the parameters (e.g. amplitude, frequency, phase) of a signal such as a sinusoid (Figure 3-4) known as a carrier wave, using an information-bearing message signal to encode it with information *(Grami 2015)*. While signals travelling through wires can be digital (a signal varying over two values only; examples: zero or one, +5 Volt or -5 Volt, +5 Volt or 0 Volt, etc.) or analog (a signal varying over a continuous range of values; example: from -5 to 5 Volt; including any value in between those numbers), only analog signals can be used in wireless systems.

The three main types of modulation are frequency modulation (FM; also called frequency shift keying or FSK) in which the frequency of the carrier signal is modulated, amplitude modulation (AM; also called amplitude shift keying or ASK) in which the amplitude of the carrier signal is modulated, phase modulation (PM; also called phase shift keying or PSK) in which the phase of the carrier signal is modulated. A widely used hybrid modulation scheme is quadrature amplitude modulation (QAM) which is a hybrid of ASK and PSK. Those modulation schemes are demonstrated in Figure 3-5, Figure 3-6, and Figure 3-7. Demodulation is the inverse of modulation, and hence is the process of extracting the information signal from the modulated signal.

Some of the objectives and benefits of modulation are:

- *"Modulation for efficient transmission: Modulation helps translate the message signal with low frequency components into a signal with much higher frequency components. The resulting modulated signal can thus possess a much smaller wavelength, and that in turn allows much smaller antennas.*
- *Modulation to match channel characteristics: Modulation allows modification of the message signal to a form suited to the characteristics of the transmission channel (for example by modulating the signal with frequencies that propagate well in the channel).*
- *Modulation for frequency assignment: Modulation allows many radio and television stations to broadcast simultaneously in a given geographical area. Since each station has a different assigned carrier frequency, the desired broadcast signals can be separated from others by tuning the receiver to select different stations as required.*
- *Modulation to reduce noise and interference: The effect of noise and interference cannot be completely eliminated in a communication system. However, by significantly*

expanding the bandwidth of the transmitted signal, the noise and interference immunity in some cases can be considerably enhanced. In other words, bandwidth increase is traded for noise and interference reduction." (Grami 2015)

Equalization

Equalization refers to any signal processing or filtering technique that is designed to eliminate or reduce channel effects that were incurred by the signal while travelling through the channel (Langton 2012). The quality of equalization depends on the extent of knowledge of the channel effects or simply the channel transfer function (Langton 2012).

The following section summarized from (Langton 2012) describes the concept of equalization and lists some of its main types:

A common approach to equalization is through inverse filtering in which the equalizer transfer function needed to compensate for channel distortion is simply the inverse of the channel transfer function:

$$H_e(f) = \frac{1}{H(f)} = \frac{1}{|H(f)|} e^{-j\theta(f)}$$

Where $H_e(f)$ is the equalizer transfer function, while $H(f)$ is the system's transfer function (includes transfer functions of the channel, transmitter and receiver), as illustrated in Figure 3-8.

The concept of transfer functions is described in detail in (Kramer 1996). In simple words, the transfer function is a function that relates one signal to another. The channel transfer function is the function that relates the signal entering the channel to the same signal when exiting the channel (after being distorted by the channel). Applying the inverse of the channel transfer function to the would recover the signal that entered the channel, hence removing the channel distortions.

The general process for equalization is illustrated in Figure 3-9. A symbol x(i) is transmitted passing through an unknown channel, and the symbol y(i) is received. The symbol y(i) may contain parts of past symbols (inter-symbol interference). The symbol y(i) goes through the equalizer which filters it according to its parameters which are called taps weights, hence producing the filtered version $\hat{x}(i)$.

To calibrate the equalizer, a training sequence known to the equalizer is transmitted at the beginning of transmission. The training sequence provides the missing transmitted data, at least for a short while. The sequence used is often chosen to be a noise-like, and "rich" in spectral content, which is needed to estimate the channel frequency response.

The filtered version $\hat{x}(i)$, also called the estimation, is compared with the transmitted symbol by the algorithm. Based on the difference, the algorithm directs the filter to adjust the tap-weights and continues in this loop until the error is acceptably small. The equalizer calibration becomes a problem of estimating the function that relates the input to the output with the least error. Equalizers can be designed to eliminate all inter-symbol interference, or to minimize the mean squared error. The equalizer designed to eliminate all ISI is called a Zero-Forcing Equalizer and the one designed to minimize the squared error (or the variance) is called a Minimum Squared Error Equalizer. At sufficiently high signal to noise ratio (SNR), MMSE and Zero-Forcing equalizers coincide. The theory behind the MMSE and Zero-Forcing equalization, two of the major types of equalization, in addition to a numerical example on each is provided by (Langton 2012).
Different types of equalizers were developed, with varying performances based on the application considered. These types include in addition to the MMSE equalizer and the Zero Forcing equalizer mentioned above: Linear equalizers, Decision Feedback equalizers, Blind equalizers, Adaptive equalizers, Viterbi equalizer, and Turbo equalizers, among other types. Some of these equalizers are better equipped to deal with time variability of the channel's response (e.g., adaptive equalizers), or to employ previous symbols decisions to enhance decisions on future symbols (e.g., decision feedback equalizers). More on equalization and an in-depth explanation of its types and their properties can be found in (Sklar and Harris 1988, Rappaport 1996, Sayed 2003, Koetter et al. 2004).

Error Detection and Correction

The following section summarized from Grami (2015) discusses the concepts of error detection and correction:

Errors occur in virtually all digital transmission systems, even after they have been optimally designed. In a digital communication system, an error occurs when a transmitted bit is a 1 and its corresponding received bit becomes a 0 or a 0 becomes a 1.

There are two types of errors: random (single-bit) errors and burst (multiple-bit) errors. In random errors, the bit errors occur independently of each other, and usually when one bit has changed, its neighboring bits remain unchanged. Examples of sources of random errors include shot noise in the transmitting and receiving equipment as well as thermal noise in free-space communication channel. In burst errors, two or more bits in a row have usually changed. Examples of sources of burst errors include impulse noise due to lightening and switching transients in radio channels.

Error-control coding is employed to reduce the bit error rate. The central concept in this coding is the calculated use of redundancy in that the number of bits transmitted is intentionally more than the required information bits. The transmitter sends the information bits along with some extra (also known as parity) bits which the receiver uses to detect or correct corrupted bits, and then removes the extra bits.

In error detection, the goal is to find out if errors have occurred, not the number of errors, or which bits are in error. Upon detecting of an error in a data unit, a system with no error correction, may simply require this data unit to be retransmitted as necessary until it is correctly received. This system is called an automatic repeat request (ARQ) system.

In error correction, the goal is to find out which bits are in error, i.e., to determine their locations in the received bit stream. The initial part of error correction is inherently error detection. The techniques that introduce redundancy to allow for correction of errors are collectively known as forward error correction (FEC) coding techniques. It is important to note that in theory it is possible to correct all errors, but the price in bandwidth may be too high as too many extra (redundant) bits are then required.

Error control codes are usually divided into two broad classes: block codes and convolutional codes. In block codes, a block of information bits is followed by check bits that are exclusively derived from the block of information bits. The check bits of a block code can be used at the receiver to detect or correct erroneous bits. This type of block code is called a systematic block code. Nonsystematic block codes in which the division into data and parity bits is not explicit were also developed. In convolutional codes, check bits are continuously interleaved with information bits to verify the information bits not only in the block immediately preceding them, but also in the previous blocks. In other words, the codeword emitted by the convolutional encoding procedure is not only a function of an information message, but it is also a function of the previous messages.

Error Detection Methods

There are three distinct widely-used error-detection methods: parity-checkcodes, checksum, and cyclic redundancy check (CRC). These techniques, which can be concurrently employed, require different levels of complexities and have different performances. The first two methods are described below.

- **Parity-CheckCodes:** In a single parity-checkcode, an extra bit is added to every data unit (e.g., byte, character, block, segment, frame, cell, and packet). The objective of adding the parity bit (a 0 or a1) is to make the total number of 1s in the data unit (including the parity bit) to become even for example. If the received number of ones is odd, the occurrence of an error is detected.
- **Checksum:** Many checksum algorithms are available. One of the simple checksum algorithms is the longitudinal parity algorithm in which a data block is divided into smaller segments of equal size (n bits; for example: 010010010001 maybe divided into 0100, 1001, and 0001), and then the exclusive or (XOR) of those segments bit by bit is computed; in simple words, an XOR operation will return 1 if and only if only one of the bits provided to it is a 1 (for example, the XOR of 0100, 1001, and 0001 is calculated as follows: the first bits of the segments are 0, 1 and 0 so the XOR with be 1. Repeating the operation for the second and third bit, the XOR of all the segments will be 1100). Note that if the calculated checksum is appended to the original data block and then the checksum of the new data block is calculated, the checksum would be all 0s (for example: the checksum of 0100-1001-0001-1100 is 0000). At the transmitter, the checksum is calculated and sent with the message, not necessarily at the end of the message, as it can be inserted in the middle of the message as well. At the destination, a new checksum, based on the received message and checksum, is calculated. If the new checksum is all 0s, the message is accepted, as it indicates with a high probability the message was not altered during the transmission; otherwise, the message is discarded. The checksum detects all errors involving an odd number of bits as well as most errors involving an even number of bits. However, any time a bit inversion (i.e., a bit in error) is balanced by an opposite bit inversion (i.e., another bit in error) in the corresponding position of another data segment, the error is invisible (undetected).

Error Correction Methods

Many error correction codes have been developed including block codes, convolutional codes and compound codes (including turbo codes). A brief description of block codes is provided below. A more comprehensive description of block codes and other error correction codes can be found in the literature.

- **BLOCK CODES:** In block codes, a data block code is first partitioned into groups of k successive bits and then each k-bits group is transformed into a larger block of n-bits. The n-bits groups are transmitted rather than the k-bits groups and the extra bits are used to correct errors.

  In linear block coding, the additional bits are generated from a linear combination of the k information bits and hence the encoding and decoding operations can be described using matrices and vectors. For example, if the data block to be divided into groups of 3 bits (k=3), and a block code of n=6 is to be used: the possible 3-bits groups (000, 001, 010, 011, 100, 101, 110, 111) are put in a matrix M:

$$M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

  The actual matrix containing the 6-bits groups representation of those 3-bits groups C whose number of columns is n and number of rows is the number of possible groups of M (8). The two matrices are related by the matrix G called the generator matrix which maps M to C and hence C = MG.

  The coding matrix G is a k by n matrix and its rows should be linearly independent. An example of G is:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

  So, C is calculated as follows:

$$C = MG$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Hence, for example rather than sending the information group 010 (third row of M), 010101 (third row of C) is sent instead. This allows for error correction. A very simple example of how a block code representation is capable of correcting a single error is shown in Table 3-1.

3.3.3 Quality of Communication

The quality of communication is the basis of assessing a communication method, channel, or device, and selecting between alternatives. The following summary discussing this topic is taken from (Grami 2015):

"The quality of digital transmission can be determined by two parameters: the transmission bit rate and the bit error rate (i.e., the fraction of bits that are received in error). These two parameters can be determined by the channel bandwidth W, measured in Hz, and the signal-to-noise ratio at the receiver input, where the average signal power S and the average noise power N are both measured in watts (W). Shannon addressed the question of determining the maximum achievable bit rate at which essentially error-free transmission is possible over an ideal channel of bandwidth W Hz and of a given signal-to-noise ratio, S, by using sufficiently complex channel coding. The larger the ratio R is, the more efficient the system is.

In a communication system, whether analog or digital, there are two primary resources: transmitted power and channel bandwidth. The transmitted power is the average power of the transmitted signal. The channel bandwidth reflects the band of significant frequency components allocated for the transmission of the signal. One of these two resources may be more precious than the other; hence communication channels are generally classified as bandlimited, such as the plain old telephone system, or power-limited, such as optical fiber links and satellite channels.

The digital communication design objectives are many, but not all of equal importance; some are required and some are desirable. Some are more difficult than others to achieve, so further

resources and additional complexities are needed to accommodate them. Table 2.2 provides a list of digital communication design objectives. It is virtually impossible to achieve all digital communication design objectives simultaneously, because they are all inherently inter-related and some are clearly in conflict with other design imperatives, so difficult trade-offs must be made. For instance, high-speed transmission is in conflict with low channel bandwidth, and low bit error rate is in conflict with low transmit power. Some constraints must be satisfied, such as those dictated by theorems in communication theory or required by governments, and some are desirable, such as those required by the users. Table 2.2 also provides a list of digital communication design constraints. Design of digital communication systems and networks thus present many challenges, as it is a multidimensional, nonlinear, constrained optimization problem. It is imperative to emphasize that there are many system variables and parameters representing a multitude of constraints, some of which cannot be easily quantified and directly factored into a design, and there are a host of conflicting design objectives that all need to be fully met, and only some, but not all, can be accommodated."

3.4 Attempts for Development of Wireless Through-soil Communication Systems

Attempts for development of wireless through-soil communication systems included electromagnetic systems, magnetic induction systems and seismic systems. Those systems encountered different challenges that resulted mostly from the characteristics of their communication channels.

Electromagnetic systems employ antennas that send and receive electromagnetic waves by perturbing the electromagnetic field. Akyildiz and Stuntebeck (2006) described the concept and some of the applications of wireless through-soil communication systems and identified the major properties of the soil electromagnetic (EM) channel: extreme path loss, reflections and refractions, multi-path fading, reduced propagation velocity and noise. The paper identifies water content, particle size, density and temperature as soil properties that combine to form soil's complex dielectric constant and hence impact electromagnetic propagation. The paper predicted that antennas to be used in those systems will be necessarily large. Vuran and Akyildiz (2007) developed theoretical estimations of the electromagnetic waves path loss in the soil channel (Figure 3-10). A theoretical model for the soil electromagnetic channel was developed in (Akyildiz et al. 2009).

In (Silva and Vuran 2009), electromagnetic communication was tested in the field by deploying sensors and receivers in shallow boreholes bored in unsaturated clay loam. Communication ranges up to 20 cm were achieved at burial depths up to 40 cm while greater-range communications were not successful. Effect of saturation level was tested, and it was realized that signal attenuation significantly increases with increasing saturation levels. Other experiments achieved longer ranges but was limited to burial depths less than 10 cm (Stuntebeck et al. 2006, Ritsema et al. 2009). Silva and Vuran (2010), achieved a maximum through-soil communication of 35 cm and confirmed the sensitivity of electromagnetic communication to soil saturation level. A maximum communication range of 130 m was achieved via a system that used low-frequency electromagnetic radiation but required expensive, low-temperature superconducting quantum interference devices which would require continuous cooling with liquid nitrogen and hence impractical, permanent access to such receivers (Vasquez et al. 2004).

Magnetic induction systems employ coils that send and detect magnetic induction (MI) waves by perturbing the magnetic field. Sun and Akyildiz (2010) introduced the concept of MI through-soil communication and demonstrated its advantages over electromagnetic systems. The MI system is theoretically claimed to be less sensitive to saturation levels. The paper suggests employing waveguides to increase the communication range. This waveguide, however, is composed of coils to be embedded in the ground, closely spaced and distributed between the transmitter and receiver. The proposed MI system is also characterized by a narrow bandwidth (1 to 2 kHz) which significantly reduces the achievable data rate compared to EM systems.

The MI system was tested in laboratory settings in (Tan et al. 2015) where 10-cm wide square coils were deployed in a 2.5 m long sand trough. A waveguide made of many coils had to be used to achieve a communication range of 2.2 m in dry sand as shown in Figure 3-11. When saturation level was increased, the path losses increased (Figure 3-12).

Ikrath and Schneider (1968) employed mechanical waves in underground communications by inducing unmodulated 80-Hz seismic signals via surface-mounted thumpers shown in Figure 3-13. Signals were detected hundreds of meters away from the transmitters. Each pulse (1 bit) varied in length between 0.5 to 3 seconds. Effects of refraction and reflection were reported. The paper concluded that seismic waves can be detected over long ranges and suggested that ground can be used as a communication channel. Transforming this system into a communication system might work by employing a simple communication scheme, similar to the famous Morse code device

(Figure 3-14), but this would yield extremely low data rates (~ 1bps). The transmitters and receivers are characterized by many limitations that prevent practical deployment in underground applications:

- They are surface mounted rather than buried which makes them unsuitable for fully embedded systems.
- Their dimensions are very big (greater than 30 cm).
- Their energy consumption is very big, approximating 10 J/bit.
- Their achievable data rate is very low (~ 1bps).

Despite the limitations of this attempt, it demonstrated the capability of generated mechanical waves to travel over large ranges which might be useful in establishing a through-soil communication system. Table 3-3 summarizes the performance of the available wireless through-soil communication systems.

3.5 Animals' Through-water and Through-soil Communication Systems

Animals have preceded humans in practical and successful communication in water and soil, by employing mechanical waves. Some submarine mammals communicate through water sometimes over thousands of kilometers (Payne and Webb 1971), while many animals and insects were found to be communicating through soil (Hill 2009), yet over much shorter ranges. Elephants, well equipped to detect seismic cues through either bone conduction or somatosensory reception (Figure 3-15), produce long repeated signals that couple with earth and respond to seismic alarm signals by aggregating into denser groups and orienting in the direction of the signal (O'Connell-Rodwell 2007). The cape mole-rat (Figure 3-16), a solitary animal that is entirely fossorial, apparently communicates by alternately drumming its hind legs on the burrow floor. Males may drum continuously for up to 2 min (Narins et al. 1992) as shown in Figure 3-17. In addition, it is conservatively estimated that at least 150,000 insect species use substrate-borne, including soil-borne, vibrations for communications (Cocroft and Rodríguez 2005); for instance, pupae and larvae of a group-living Japanese rhinoceros beetle are known to generate vibrational impulses to deter larval intrusion through soil (Kojima et al. 2012).

The comparison between the audiogram of normal rodents and that of fossorial rodents, in addition to the audiogram of elephants shown in Figure 3-18 demonstrate the sensitivity of elephants and fossorial rodents to relatively lower acoustic frequencies. Since lower frequencies are

characterized by lower attenuation (as demonstrated later in section 5.2), this sensitivity to lower frequencies enable those two species to communicate over longer ranges. The large ranges that seismic waves can travel encouraged animal scientists in (O'Connell-Rodwell et al. 2001) to suggest exploring the potential of soil to house seismic communications.

3.6 Through-water Mechanical Communication

While the purpose of this research is to communicate data through soil, it is valuable to check if any communication attempt has succeeded in other media. Communication through air is very familiar to humans, and communication through vacuum (using electromagnetic waves) was achieved in the past 2 centuries. The relatively recent breakthrough is the success to efficiently communicate through water by employing mechanical waves (acoustic waves), mimicking in that whales and dolphins. Successful attempts had to overcome this channel's inherent challenges: multi-path propagation and frequency-dependent absorption yielding high levels of dispersion, and motion-induced Doppler effects. Even though frequency-dependent absorption limits km-scale acoustic communications to modest data rates of less than 10 kbps, algorithms capable of communicating at data rates in excess of 1 Mbps over 100m ranges and up to 400 Mbps over cm-scale distances have been developed (Riedl and Singer 2013, 2014).

3.7 Mechanical Waves in Other Fields

Since attempts to employ electromagnetic waves in through-soil communication were not very successful or practical as discussed in section 3.4, and since the mechanical waves seemed to work better in one communication system, in the animal world, and through water, the literature on mechanical wave propagation in soils is further studied.

Mechanical wave propagation in soils has been extensively studied in multiple fields including geotechnical earthquake engineering, seismology, acoustics, ocean bed acoustics and geophysics. Geotechnical earthquake engineering, for example, aims at assessing the seismic hazard in areas of interest then estimating the response of soil and overlying structures to possible seismic events, hence, providing designers with needed design metrics. In geotechnical earthquake engineering, the propagation of seismic motions from bedrock to the ground surface can be estimated if: 1- the properties of the channel (soil column) are well measured; 2- the input motion is measured; 3- decent models are available to represent the response of the soil and rock to the input motion. A decent overview of the topic can be found in (Kramer 1996). Testing methods aiming at measuring

dynamic properties of soil were developed for field and lab. Thousands of real seismic motions were recorded across the globe for numerous seismic events using sensor systems. Sophisticated soil models and damping formulations were developed for modeling dynamic soil behavior and tested in different platforms (Phillips and Hashash 2008, Groholski et al. 2016, Baltaji et al. 2017, Numanoglu 2019). This problem was modelled as a one-dimensional problem (Hashash et al. 2016) or using FEM as a two-dimensional problem or a 3-dimensional problem (Basarah et al. 2019).

Seismology is concerned with understanding causes and mechanisms of earthquakes, their propagation throughout the earth (including deep earthquakes), recording and locating them when they occur, and estimating their occurrence based on active measurements and earth crust modelling (Bullen et al. 1985, Sheriff and Geldart 1995, Aki and Richards 2002, Shearer 2019).

In Acoustics, the propagation of the mechanical waves in air, fluids and solids is studied. While pressure waves propagate in water and most fluids, additional types of waves propagate in solids including shear waves, and surface waves. The behavior of waves at the interfaces between materials of different properties (like in layered media such as soil or oceans whose properties vary with depth) is studied as well. The interaction of the waves propagating in water with the ocean bed was extensively studies as this problem was of an interest to underwater communication (Brekhovskikh et al. 1991, Lurton 2002, Jensen et al. 2011, Brekhovskikh and Godin 2012).

In geophysics, mechanical (acoustic/seismic) waves have been extensively employed. Seismic geophysical methods typically target measuring pressure and shear wave velocities in ground. In those methods a mechanical signal (like a hammer hit) is typically induced and then recorded at other locations. The ability of detecting the sent signals and timing their travel times allows back-calculation of wave velocities. Common seismic methods include seismic refraction, seismic reflection, spectral analysis of surface waves (SASW), multichannel analysis of surface waves (MASW), refraction microtremor (ReMi), downhole testing, and cross-hole testing (Robinson 1988, Telford et al. 1990, Lowrie and Fichtner 2020).

It is clear from the description of those different fields that there are a lot of common topics. The basic common theme across them is that they all deal with mechanical waves. The terminology might be confusing, giving the reader a wrong impression that these field discuss very different topics. For example, the term acoustic wave is usually used to refer to pressure waves which humans can hear (typically in the frequencies ranging between 20 Hz and 20 kHz), but in

Acoustics, propagation of waves whose frequencies are beyond this range has been studied. It also studied waves that cannot be “heard” such as shear waves in solids. Some authors will call waves other than pressure waves elastic waves (waves that propagate through homogeneous elastic solids) and seismic waves (waves that propagate through heterogeneous solids) (Wencai 2013). Similarly, seismic waves typically refer to mechanical waves generated during earthquake events and are usually characterized by lower frequencies (typically thought of as less than 200 Hz). However, those waves are just low frequency “acoustic waves” propagating in “layered media”. Hence, the term “mechanical waves” is a general term that encompasses acoustic and seismic waves.

While through-soil propagation is studied at different scales and depths in those fields, none of these fields employed soil as a channel for communicating information. The closest field to achieve what might be described as communication is geophysics in which no information is encoded in the induced wave, but the wave itself (specifically its arrival time) is the information that is communicated and then used to infer soil or rock dynamic properties.

In addition, it is important to realize the differences in the scales and consequent assumptions considered in each of those fields. Geotechnical earthquake engineering and seismology typically focus on strong far away motions that propagate from the earthquake epicenter to the ground surface. This assumption allows geotechnical earthquake engineers to approximate the site response problem in many cases as a one-dimensional problem with the assumption that the waves have spread enough such that bedrock at the bottom of the site’s soil column is assumed to move as a single unit. However, this is not the case in geophysics and in a lot of the cases studied in Acoustics. For example, acoustic point sources and line sources are studied in Acoustics, and hence the propagation direction is multi-dimensional, and consequently, the effects of local reflection and refractions due to local media features become much more significant.

3.8 Conclusions

In this chapter, the main challenges facing wired through-soil communication systems are presented, showing the need for wireless communication systems. Key communication terminology, concepts and techniques were described. Attempts for development of wireless through-soil communication systems were then examined, while highlighting the challenges faced by each system. Electromagnetic and magnetic induction systems suffered from huge electromagnetic wave attenuation which greatly limited the achievable communication range,

while the seismic (mechanical) system suffered mainly from the system's big size and high power consumption.

It was also found that many animals employ mechanical waves for through-soil communication. A recent success in through-water communication also employed mechanical waves. With the mechanical wave showing an increasing promise for the target application of this research, the main disciplines studying the mechanical wave propagation were briefly reviewed, while highlighting their similarities and the fields of focus.

Based on the literature review discussed in this chapter, the following plan for developing a wireless through-soil communication system is proposed:

- Identify design requirements for the target application (geotechnical instrumentation systems).
- Better characterize the soil channels properties, then select the most suitable channel for the communication system.
- Propose an initial communication system based on the identified soil channel characteristics.
- Test the proposed system and identify further challenges.
- Iteratively improve the communication system till the design requirements are achieved.

Those steps were followed in CHAPTER 5, and the new wireless through-soil communication system termed "SoilComm" was successfully developed.

3.9 Tables

Table 3-1: Table showing how redundancy helps in detecting and/or correcting communication errors. This table assumes that a maximum of one bit flip can occur.

Code	Possible Outcomes if a 0 is Transmitted				Decision
0 as "0" **1 as "1"**	0	1			a 0 can be a true 0 or a flipped, so errors can't be detected or corrected.
0 as "00" **1 as "11"**	00	01	10		a 01 can be a true 0 and a flipped 1 or a flipped 0 and a true 1, so one error can be detected but not corrected.
0 as "000" **1 as "111"**	000	001	010	100	Assuming one error occurs, then 001, 010 and 100 are all originally 000, so this scheme can detect and correct a single error.

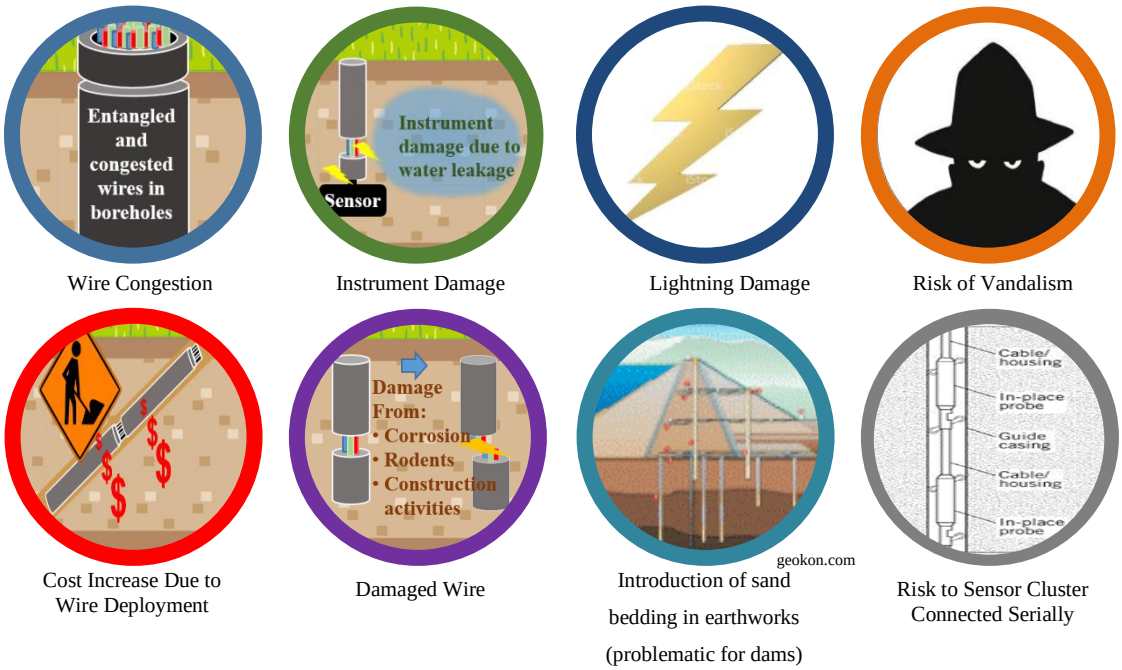
Table 3-2: Summary of design objectives and constrains. Source: (Grami 2015).

Design Objectives	Design Constraints
maximize transmission rate (bps)	the Shannon capacity theorem (and the Shannon limit)
minimize bit error rate	the Nyquist theoretical minimum bandwidth requirement
minimize signal bandwidth (Hz)	source entropy for lossless compression
minimize transmission power (W)	subjective perception for lossy compression
maximize system throughput	government regulations (e.g. frequency allocations and coordination)
minimize overhead and signaling bits	technological limitations (e.g. state-of-the-art components)
minimize noise and interference (W)	laws of nature (e.g. multipath fading, ionospheric propagation)
minimize overall delay (s)	de jure and de facio standards (e.g. IEEE, ISO, Bluetooth)
minimize jitter (s)	user device specifications (e.g. aesthetics, size, weight, cost)
maximize system security	user interface requirements (e.g. user-friendly features)
maximize system flexibility	mass, power and real estate envelopes (e.g. satellite buses)
maximize system capacity (bps)	networking requirements (e.g. mobile device coverage)
minimize computational load	radiation restrictions (e.g. acceptable health hazard in mobile devices)
minimize system complexity	service and traffic differentiation (e.g. user and application priority)
maximize system utilization	maintainability (e.g. ease of testing, monitoring and control)
maximize system reliability	upgrade requirements (e.g. forward and backward compatibility)
minimize system cost ($)	application requirements (e.g. traffic patterns and characteristics)
minimize access/usage fee ($)	market considerations (e.g. risks, economics of scale, affordability

Table 3-3: Summary of the performance of available wireless-through soil communication systems

System Type	Max Range (m)	Data Rate	Size	Energy Efficiency
Electro-magnetic	less than 2	high	small	high
Magnetic Induction	less than 2	high	large	high
Seismic	hundreds	low	large	low

3.10 Figures



Wire Congestion | Instrument Damage | Lightning Damage | Risk of Vandalism

Cost Increase Due to Wire Deployment | Damaged Wire | Introduction of sand bedding in earthworks (problematic for dams) | Risk to Sensor Cluster Connected Serially

Figure 3-1: Cartoon showing the main causes of damage of underground sensors. Source: (Baltaji et al. 2019)

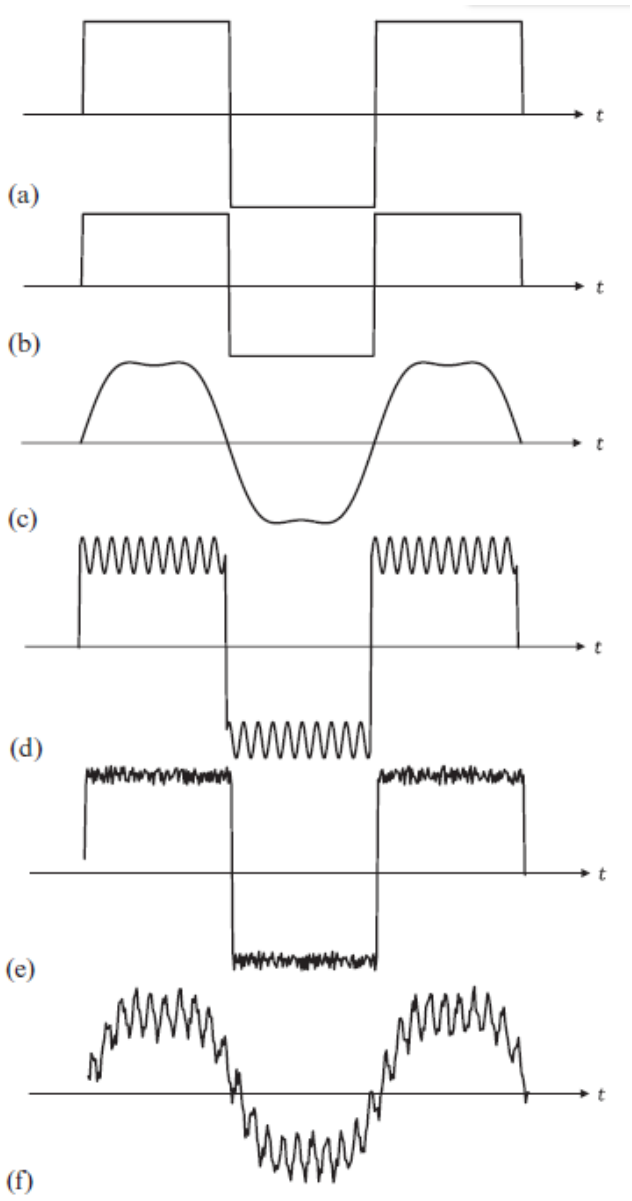


Figure 3-2: An example of transmission impairments: (a) transmitted signal; (b) effects of attenuation; (c) effects of distortion; (d) effects of interference; (e) effects of noise; and (f) aggregate effects of impairments. Source: (Grami 2015).

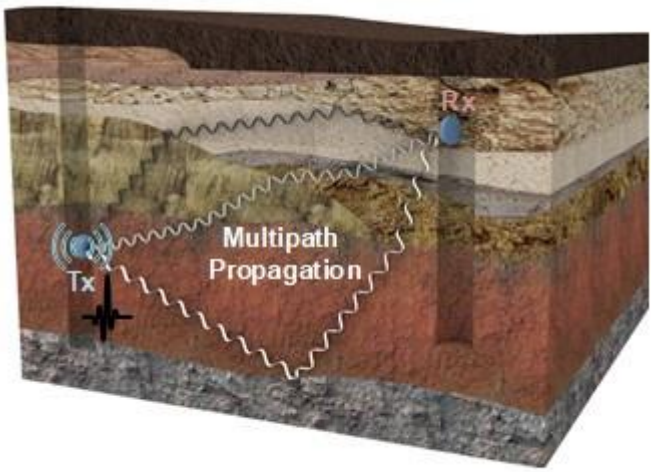


Figure 3-3: Schematic illustrating multi-path propagation. Source: (Yang et al. 2019)

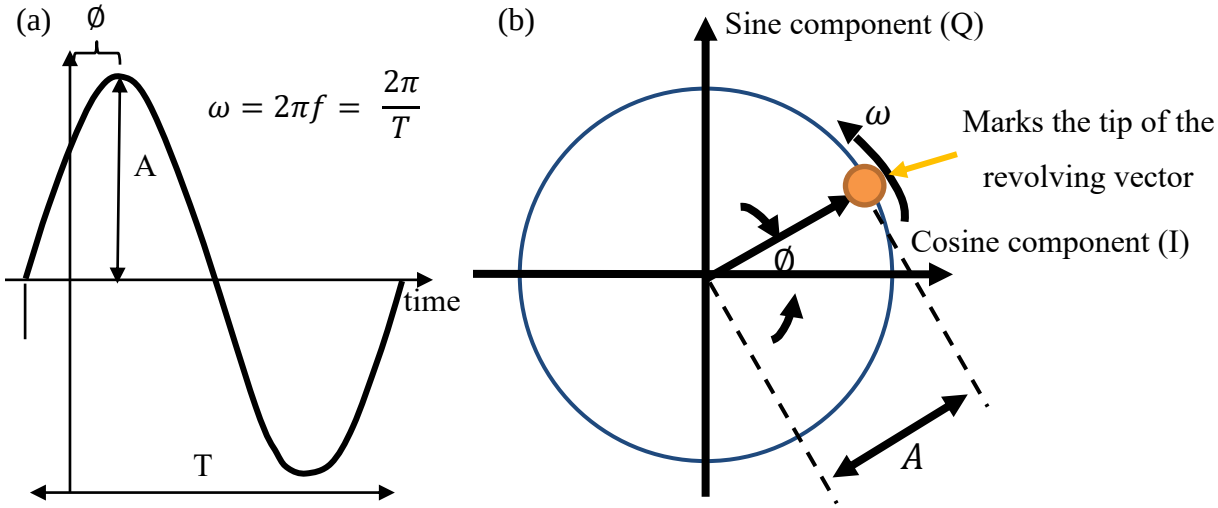


Figure 3-4: Illustration of (a) the sinusoid and its properties, and (b) its representation in the phasor diagram. Instead of plotting the amplitude of the wave versus time as in the plot in (a), the phasor diagram in (b) represents the wave by a rotating vector whose length is the maximum amplitude of the wave, whose angle with the horizontal axis is the phase angle, and whose instantaneous amplitude is the projection on the horizontal axis. The frequency of the sinusoid is represented by the angular frequency ω. The tip of the vector is typically shown instead of showing the whole vector.

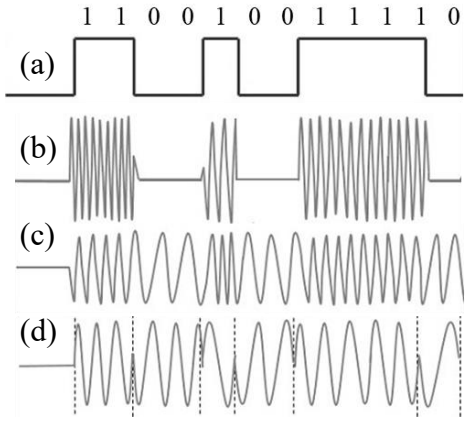


Figure 3-5: (a) sample signal composed of '1's and '0's and its modulation using: (b) amplitude shift keying (ASK), (c) frequency shift keying (FSK), or phase shift keying (PSK). In (b), one amplitude represents “1” and another amplitude represents “0”. In (c), one frequency represents “1” and another represents “0”. In (d), one phase angle represents “1” and another represents “0”.

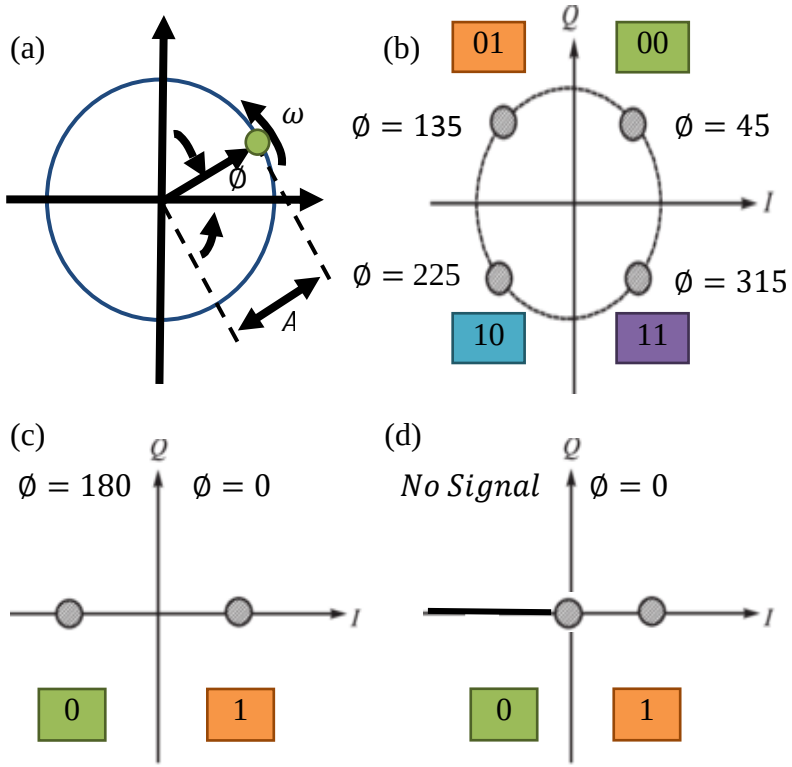


Figure 3-6: (a) a schematic showing phasor diagram of a sinusoid where the grey circle represents the tip of the phasor, and the corresponding constellation diagrams for some modulation schemes: (b) QPSK: “00”is represented by a phase of 45°, “01”is represented by a phase of 135°, “10”is represented by a phase of 225°, “11”is represented by a phase of 315° (c) BPSK (sometimes called 2-PSK): “0”is represented by a phase of 180°, “1”is represented by a phase of 0° (d) On-Off Keying (OOK): “0”is represented by no signal (amplitude equal zero), “1”is represented by a sinusoid of phase equal to 0°.

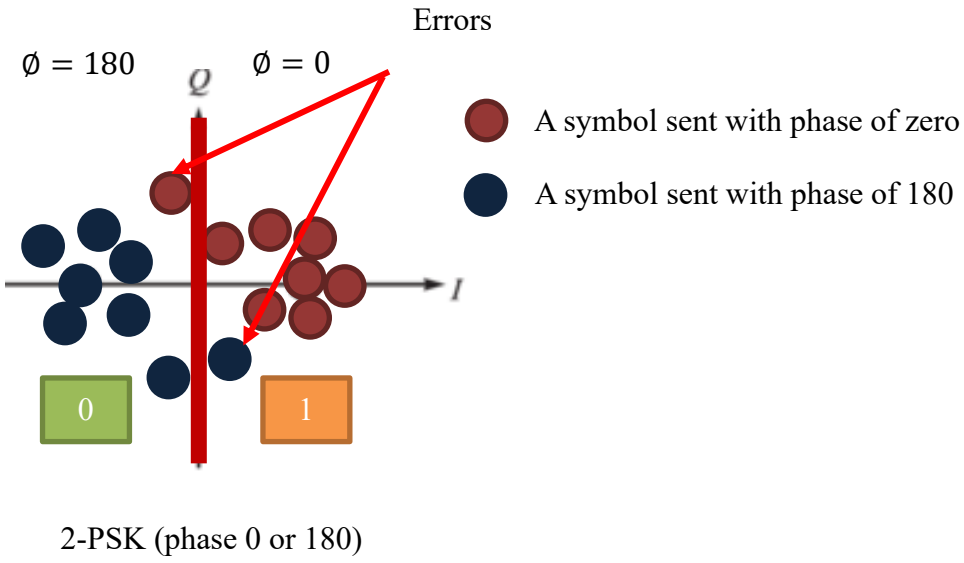


Figure 3-7: Schematic showing an example of how symbols modulated by a 2-PSK modulation scheme may be received and decided. The red symbols were sent with a phase of 0° while the blue symbols were send with a phase of 180°. Symbols received on the right are decided as symbols originally sent with a phase of 0° while those received on the left are decided as symbols originally send with a phase of 180°. Symbols received on the wrong side cause errors.

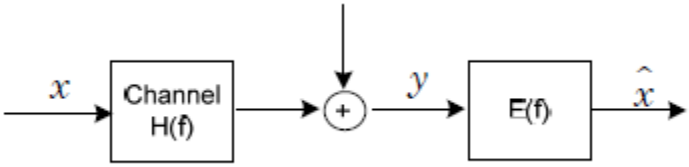


Figure 3-8: Schematic illustrating the role of an equalizer (Langton 2012). The signal "x" is altered while propagating through the channel. Noise is added to this signal as well, making up the received signal “y”. The equalizer attempts to recover the signal “x”, by reversing the effects of the channel, and hence producing the equalized signal “$\hat{x}$”. Source: (Langton 2012).

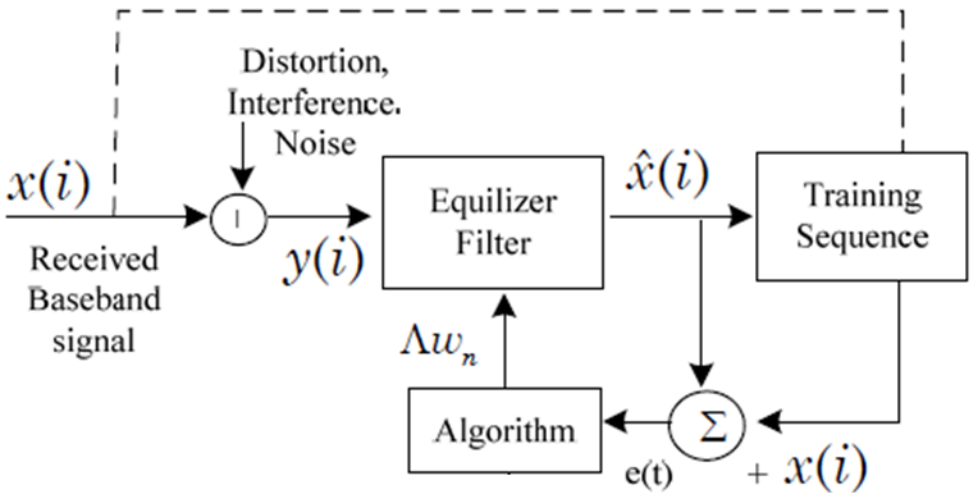


Figure 3-9: Schematic illustrating the general process of equalization. A training sequence is used to adjust the equalizer parameters such that the equalized signal approximates the transmitted signal. The calibrated equalizer can then be used to equalize signals unknown to the equalizer. Source: modified from (Langton 2012).

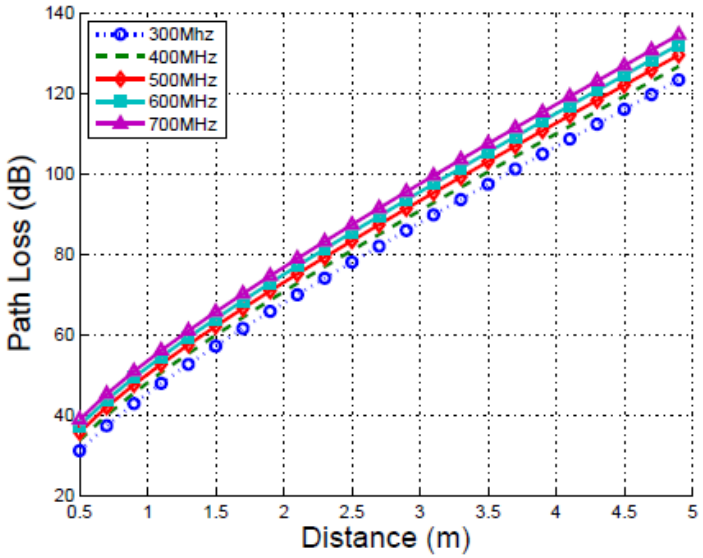


Figure 3-10: Theoretical path loss for electromagnetic waves propagating in soil at different frequencies on the order of hundreds of MHz as estimated by Vuran and Akyildiz (2007). Source: (Vuran and Akyildiz 2007)



Figure 3-11: Photo showing the magnetic induction communication system developed by Tan et al. (2015). The black coils are part of the waveguide used to extend the range of communication to 2.2 m. Source: (Tan et al. 2015)

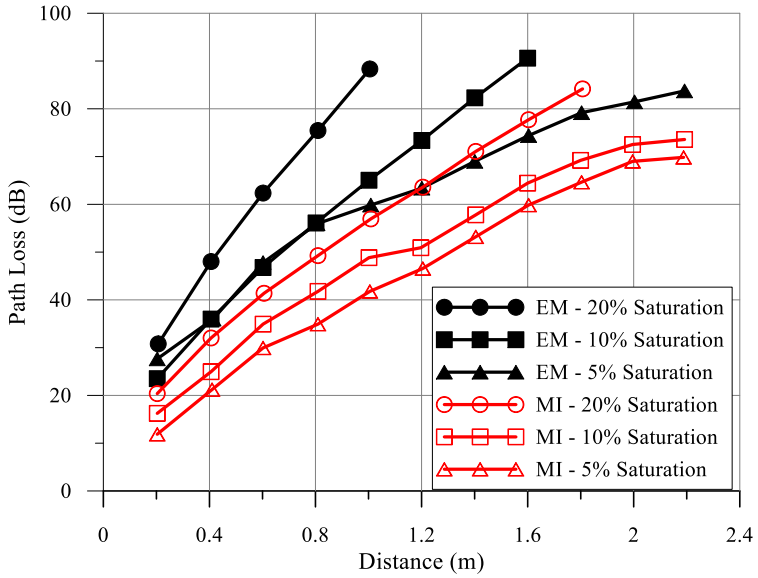


Figure 3-12: Comparison of electromagnetic path loss to magnetic induction path loss over different saturation levels as measured by Tan et al. (2015).

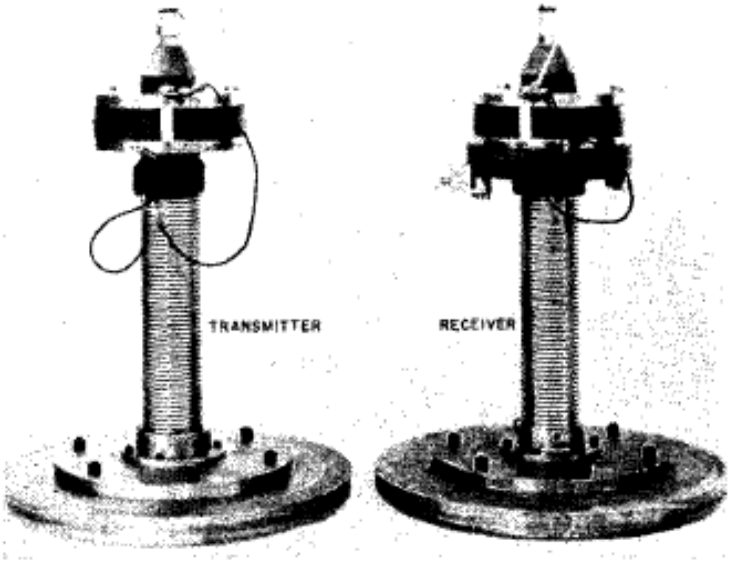


Figure 3-13: Photo of the thumpers used by Ikrath and Schneider (1968) to generate and receive seismic signals. Source: (Ikrath and Schneider 1968).

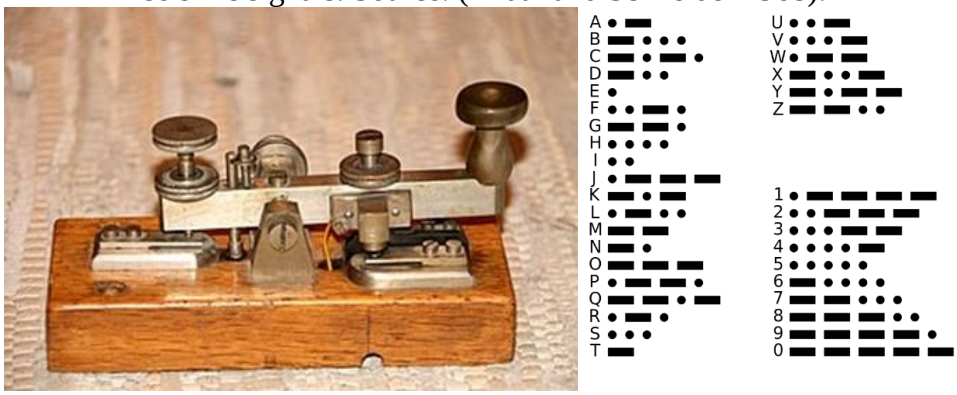


Figure 3-14: Photo of a Morse key device (left) and the international Morse code (right). Source: wikipedia.org.



Figure 3-15: Schematic of an elephant leg showing the mechanism which makes elephants senseitive to seismic waves. Source: (O'Connell-Rodwell et al. 2001).



Figure 3-16: Photo of the Cape mole-rat. Source: keywordbasket.com

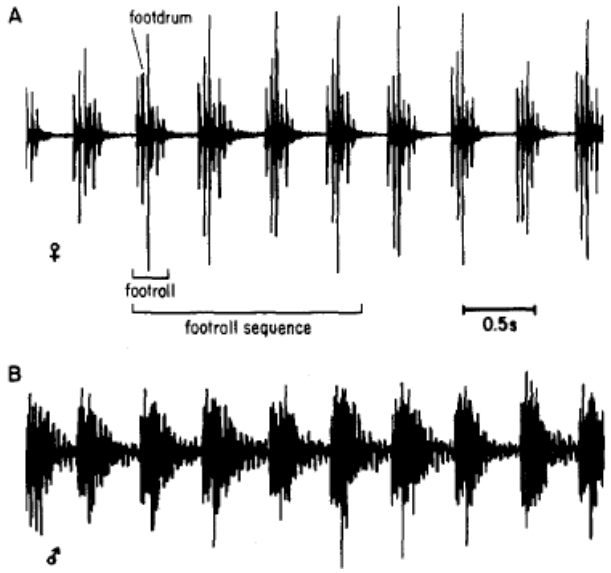


Figure 3-17: Recordings of simultaneous thumping on soil by a male and female Cape mole rats Source: (Narins et al. 1992).

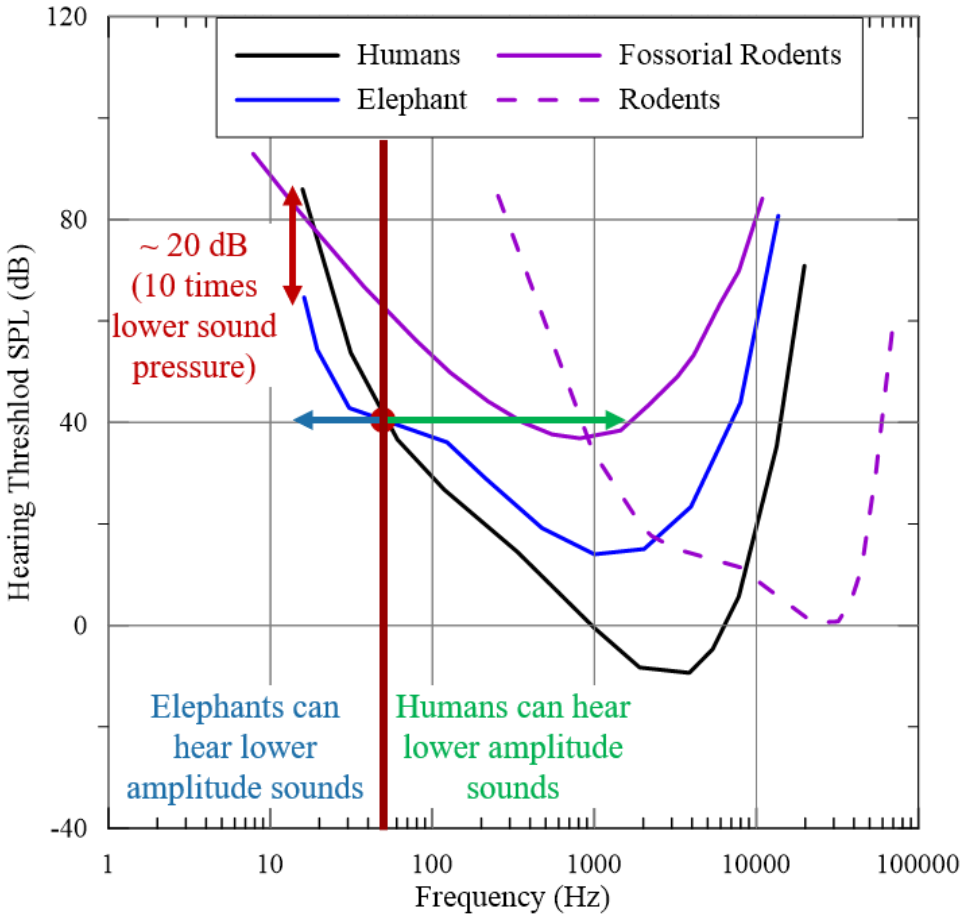


Figure 3-18: Audiograms of humans and some animals showing their sensitivities to sound at different frequencies. Source: modified from (Slabbekoorn et al. 2018).

CHAPTER 4. Inverse Analysis Frameworks: Properties, Challenges and Prospects

The major characteristic that differentiates a control method, such as the observational method, from a forward method, is its capability to incorporate the system's performance measurements to update the system's model (i.e., learn the system's behavior) which is then used to update the system's future performance estimations.

In modern applications of the observational method, the system is modeled via a numerical method such as the finite element method (FEM). The FEM solution depends on idealized representations of material behavior, known as material models or constitutive models. Hence, in FEM, the problem of learning a system's behavior is the same problem as finding a material model that results in this system's behavior.

Solving this inverse problem requires two elements: a material model capable of this learning, and a learning algorithm. This chapter summarizes the material modeling approaches and learning algorithms that were previously employed in this problem.

4.1 Conventional Material Modeling

Constitutive models are idealized representations of quite complex material behavior. They are composed of a set of relationships and rules that describe the response of materials (e.g., strain, etc.) to loading (e.g., stress, thermal variations, etc.). Exact models do not exist; for example, it may be appropriate to claim that the behavior of steel can be represented by an elastoplastic model, but not to claim that steel is elastoplastic (Runesson et al. 2006).

Development of constitutive models is guided by a set of principles according to (Runesson et al. 2006):

- "*coordinate invariance: constitutive relations should not be affected by arbitrary coordinate transformations*
- *determinism (or causality): the stress in a given body is determined entirely by the history of the motion of the body, i.e. it is not affected by future events*
- *material objectivity (or frame-indifference): constitutive relations must not be affected by an arbitrary Rigid Body Motion (RBM) that is superposed on the actual motion.*
- *second law of thermodynamics: the production of internal entropy, or rate of "material disorder", must be non-negative or in other words: dissipation of energy is never negative*"

Approaches for constitutive modeling include the fundamental micromechanics approach (in which behavior is tested at the microscopic scale then the macroscopic behavior is estimated via averaging techniques), the phenomenological approach (in which the macroscopic behavior is directly tested) and the statistical approach (in which the behavior is represented by statistical distributions that account for scatter in the observed behavior) (Runesson et al. 2006). The phenomenological approach is the most widely used approach for geotechnical engineering problems.

Conventional material models are characterized by their explicit functional form which clearly presents their behavior, however, this fixed functional form encompasses many drawbacks: the functional forms and their calibrated parameters are biased towards the data sets that were used to develop those models and as new data sets become available, recalibration or reformulation of the functional forms might become necessary (Ghaboussi 2010). Refined models, capable of capturing fine details of material behavior, encompass a lot of parameters which require testing and calibration and their functional uniqueness and precision requirement makes them more suitable for forward analyses and difficult to use in inverse problems (Ghaboussi 2010).

In geotechnical engineering a wide variety of constitutive models is used ranging from simple linear elastic models to elastic plastic models such as the Mohr-Coulomb model to more sophisticated models such as the hardening soil model with small strain stiffness (Schanz et al. 1999, Benz et al. 2009) and the MIT-E3 model (Whittle and Kavvadas 1994). As the level of sophistication of a model increases, the complexity of the behavior that it can capture usually increases, but the number of parameters required to use the model often increases as well. With increased number of required parameters, the number of laboratory and field tests required to quantify those parameters often increase.

4.2 Conventional Inverse Analysis Methods

Engineering problems are usually classified into forward and inverse (also called backward) analyses (Figure 4-1). A forward analysis is an analysis where the input and the system behavior are known, and the output is solved for. An inverse analysis is an analysis where the output and system behavior are known, and the input is solved for, or the output and input are known, and the system behavior is solved for. The latter type is the one needed in a modern application of the observational method: the input (which is the construction activities) is known and the output (which is the system performance measured via sensors) is measured, while the system's model is

solved for. As described earlier, the system's model is a consequence of the material model used, and hence it is sufficient to solve for the material model.

Solving this inverse problem is typically done either by ad hoc procedures or by using optimization algorithms. Ad hoc methods can be used to solve this inverse analysis whereby a conventional material model parameters are adjusted to match numerical estimations more-closely with field observations. In many cases the ad hoc approaches for model update are not sufficient (Osouli et al. 2009). For example, a simple linear elastic model will not be able to learn a nonlinear plastic behavior no matter how much it is optimized. Similarly, a model that is not formulated to capture soil complex behavior such as small strain nonlinearity, anisotropy, etc. will not be able to learn this behavior no matter how much its parameters are adjusted.

Optimization techniques automate conventional material model calibration by using optimization algorithms such as gradient decent and genetic algorithm. The main drawbacks of optimization algorithms are complexity, non-uniqueness, and instability: the complexity of real nonlinear systems sometimes leads to problems of insensitivity when the observations do not contain enough information to support estimation of the parameters; non-uniqueness may result when different combinations of parameter values match the observations equally well; instability can occur when slight changes in model variables radically change model results (Finno and Calvello 2005). Those methods are also time and resource extensive and become quickly impractical with increased number of constitutive model parameters. For example, (Hashash et al. 2010) reports that an excavation problem requiring identification of three soil model parameter takes about 2 days using an office computer. In addition, because they act on conventional material models, they are limited by those models' inherent limitations.

4.3 Artificial-Neural-Networks-Based Material Modeling

An artificial neural network (ANN) is composed of an organization of simple elements called neurons. A neuron is a simple bio-inspired computational element (Figure 4-2) that sums input from other neurons then applies a mathematical function, called the activation function, to the sum to produce an output as demonstrated in Figure 4-3. The neurons are organized in networks. Numerous types of networks are available including fully connected, recurrent, convolutional and many others. The type discussed in this thesis is the fully connected network. A fully connected network is composed of an input layer, one or more hidden layers and an output layer (Figure 4-4). The inputs are passed to the input layer which propagates them forward to the hidden layers then

the output layer. The network is provided with data sets called the training sets composed of inputs and target outputs. The neural network is first initialized, either completely randomly, or following a certain distribution. The network is then provided with the inputs. The network calculates outputs based on its initialized weights. The outputs will be different than the target outputs, and hence errors (i.e., difference between outputs and target outputs) can be calculated. A backpropagation algorithm is used to adjust the weights of the neural network to reduce those errors. This process, called training, is repeated until the output becomes close to the target output.

Ghaboussi et al. (1991) introduced the idea of employing ANNs in constitutive modeling. ANNs were trained with stress and strain measurements from monotonic biaxial and cyclic uniaxial tests done on plain concrete samples. The When tested after training, the ANNs yielded very good matches with other test results which were not included in the training. This indicated that ANNs were capable of learning the constitutive behavior of plain concrete. The study realized the need for large amounts of data for training an ANN. The study emphasized the benefits of using ANNs for material modeling: no a-priori assumptions about the material behavior is needed, and ease of update of model as more measurements become available. The study points to the need for a deeper understanding of neural networks and for an automatic method for selecting the architecture of the ANN.

Wu and Ghaboussi (1995) employed adaptive node generation to adjust the ANN architecture as needed. Intending to overcome the limitation of small data sets, Sidarta and Ghaboussi (1998) and Ghaboussi et al. (1998) introduced the autoprogressive algorithm, discussed later, to extract training data from nonuniform tests applied to geomaterials. (Ghaboussi and Sidarta 1998) suggested benefitting from the fact that lower dimension problems are subsets of higher dimension problems, termed as nested structure of data: 1D problems are subsets of 2D problems which are subsets of 3D problems and the drained condition is a subset of the undrained condition.

After employing ANN-based constitutive models in an excavation problem in (Hashash et al. 2003), Hashash et al. (2004) derived a stiffness matrix for an ANN-based constitutive model that employs the hyperbolic tangent activation function. This enabled conducting implicit analyses, and benefitting from its numerical and computational advantages in static problems. Butkovich and Hashash (2005) proposed methods for enhancing training data sets by benefitting from material centerline symmetry and by selective data removal. Jung and Ghaboussi (2006) employed ANN in learning behavior of rate-dependent materials. Tsai and Hashash (2008) used ANN models

in 1-D site response problems, and Groholski and Hashash (2013) Groholski et al. (2014) incorporated learning pore-water pressure generation in those problems. ANN-based constitutive models were taught the behavior of concrete, steel, soil, and even red blood cells (Song and Hashash 2009).

4.4 The SelfSim Inverse Analysis Method

SelfSim is an inverse analysis method which was developed and enhanced by Ghaboussi, Hashash and their co-authors. SelfSim integrates two core ideas: ANN-based constitutive modeling and a modified version of the autoprogressive algorithm developed by ((Sidarta and Ghaboussi 1998) and (Ghaboussi et al. 1998)). The inverse algorithm typically operates as follows: the system under study (e.g., excavation, lab test, etc.) is modeled via two finite element analysis models both employing an ANN-based constitutive model. In the first model, the loading (or unloading) boundary conditions are applied. In the second model, the measured response of the system to this loading (e.g., displacements measured by inclinometers) is applied as displacement boundary conditions. The ANN model is initially pretrained to learn a linear elastic behavior via a user-provided training dataset, which is necessary for the first run of the analyses. Both models are run for a single analysis *step*, then their results are extracted and compared. The goal of the analysis is to find an ANN model with which the results of the two analyses would match: the first analysis would yield the actual displacements and the second would yield the actual loads. However, since the ANN model was only pretrained by a linear elastic dataset which does not represent the actual behavior of the material, the results of the two analyses will not match. To find an ANN model that matches both loads and displacements, the following assumptions are made:

- Since correct boundary forces were applied in the first analysis and equilibrium was reached, the computed stress field in the first analysis provides an acceptable approximation of the actual stress field.
- Since correct boundary displacements were applied in the second analysis and equilibrium was reached, the computed strain field in the second analysis provides an acceptable approximation of the actual strain field.

The extracted stresses of the first analysis and strains of the second analysis are assumed to compose together a stress-strain training dataset. This dataset is used to retrain the ANN model, or to train a new ANN model. The new ANN model will better match the behavior of the two

analyses. However, multiple *cycles* of running and training will likely be needed, before the two analyses yield well matching results. The degree of matching is assessed by calculating the normalized average and max displacement errors between the two analyses. Once those errors drop below a user-defined threshold (e.g., 2.5% and 5% respectively), or the number of cycles reach a user-defined limit (e.g., 3) the inverse analysis is considered ready to move to the next *step*. The same process is repeated till all steps of the analyses are completed, hence completing the first *pass*. Multiple passes might be needed before the two models match well.

In (Hashash et al. 2010), a comparison between the performance of the SelfSim framework and that of the genetic algorithm, which is an optimization algorithm, showed a superior performance of the SelfSim framework.

The SelfSim framework can be thought of as a simultaneous solution of two inverse problems:

1- extracting constitutive behavior from global behavior and
2- finding the ANN material parameters that will produce this constitutive behavior.

A schematic showing the flow and elements of a SelfSim analysis is shown in Figure 4-5.

The SelfSim framework has been successfully used to learn material behavior from lab and field measurements. In (Marulanda 2005), SelfSim was used to learn the behavior of soil in the Stata Center Excavation in Boston. The 117 m by 96 m and 13 m deep excavation is described in (Hashash et al 2006). A schematic plan view showing the excavation, the adjacent buildings and the locations of the monitoring points is shown in Figure 4-6. The excavation was supported via a diaphragm wall and 3 rows of tiebacks. A schematic of the support system and the construction stages is shown in Figure 4-7. Excavation performance measurements along the west section of the excavation were used to train the ANN models. The trained models were capable of learning the behavior as shown in Figure 4-8. The trained ANN models were then used to estimate the performance along the south and east sections of the excavation. The estimations were in good agreement with the actual measurements as shown in Figure 4-9.

In (Osouli and Hashash 2010), performance measurements from the TNEC excavation in Taipei was used to train ANN constitutive models which were then used to estimate the performance of the Formoza excavation which is 2 km away from the TNEC excavation and which is characterized by similar soil deposits. The estimations were in good agreement with the actual measurements. Marulanda (2005) and Osouli et al. (2009) studied the effect of the availability of different

instruments data, their geometric distribution and their number on the SelfSim learning, and came up with recommendations for instrumentation systems.

Tsai and Hashash (2008) and Groholski et al. (2014) employed SelfSim in 1-D site response. Data from downhole array measurements were used to learn the cyclic soil behavior without porewater pressure generation (Tsai and Hashash 2008) and with porewater pressure generation (Groholski et al. 2014).

In (Moon and Hashash 2015), data from nonuniform direct simple shear tests were used to train ANNs, which were then used in simulations of an excavation problem. The results yielded good matches with the actual performance of the excavation.

In (Hashash and Asmar 2018), a new non-uniform triaxial test that benefits from 3D scanning for deformation measurements was introduced. The new device inherits all features of a conventional triaxial test, and adds lateral restraint clamps, to increase non-uniformity in specimen deformation, combined with a digital photogrammetry system to measure the 3-D deformed shapes of the specimen (Asmar (2017)). Using two high resolution digital cameras mounted in front of the cell, the system is able to capture the specimen's deformed shape synchronously with measurements of loads, axial displacement, and pore pressures/volume change during the shearing process (Asmar (2017)). The setup of the this new test is shown in Figure 4-10. This new setup was found to generate very rich data composed of multiple stress paths from each data set. SelfSim was then successfully used to learn this complex behavior. A sample of the results of the study is shown in Figure 4-11.

Despite its success in different problems, the framework suffers from many limitations. Osouli (2010) noted the difficulty in preparing SelfSim input files and interacting with the framework's interface, and claimed that the time spent because of those issues is not affordable in practice. Groholski (2012) suggested that research should be conducted to reduce the duration of the SelfSim learning process and accelerate its convergence without sacrificing the correctness of the solution.

Asmar (2017) recommended rewriting the SelfSim code benefitting from the available ANN codes, to reduce the run-time and improve the frameworks capability of handling a broad range of geotechnical problems. It also emphasized the need for a better user-interface. From a personal communication, the author of (Asmar 2017) reported run times on the order of 4 to 5 hours to learn soil behavior from a single triaxial test included in the study. Given the small number of elements of this model, larger problems such as excavation or tunneling problems, will consume days to

finish. Also, bugs in the SelfSim code issues were reported. Those bugs prevented the convergence of many analyses and caused huge additional delays. Those issues are limiting the framework's usability and hindering its adoption. For example, waiting for a SelfSim inverse analysis for days or even weeks, will limit its value as a decision-assisting tool. This could explain the fact that the industry has not yet adopted this framework.

4.5 Conclusions

Based on the literature review discussed in this chapter, the following plan for developing a new computationally efficient SelfSim framework is proposed:

- Identify the causes of slowdown of the old code.
- Develop a new code while optimizing it for computational speed and efficiency.
- Test the new code using synthetic data.
- Use the new code to learn the behavior of real soil.
- Assess the performance of the new code in terms of its speed relative to previous versions of the code.

The first two steps were explained in detail in CHAPTER 6. Extensive testing of the new framework using synthetic data was demonstrated in CHAPTER 7. The new framework was used to learn the behavior of Ottawa Sand subjected to a series of conventional and modified triaxial testing. The performance of the new code was monitored and compared to the performance of previous versions of the code. This is described in detail in 0.

4.6 Figures

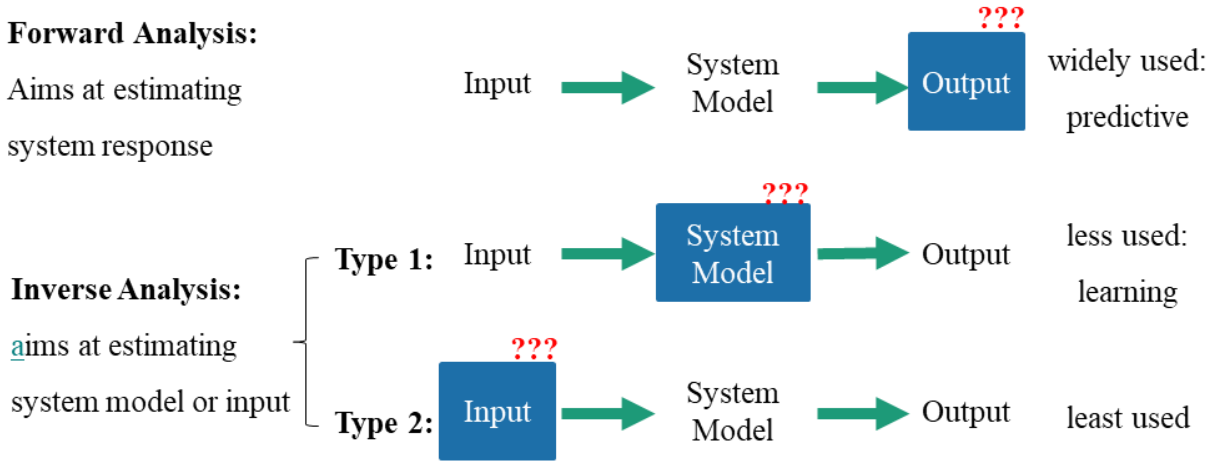


Figure 4-1: Types of analyses

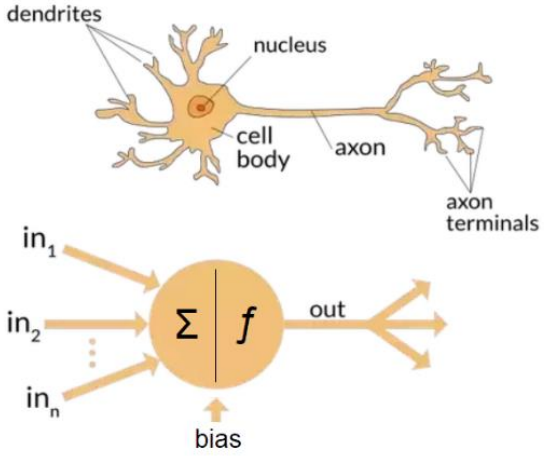


Figure 4-2: Schematic showing the function of an artificial neuron and its mimicry to natural neurons. Source: (https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7)

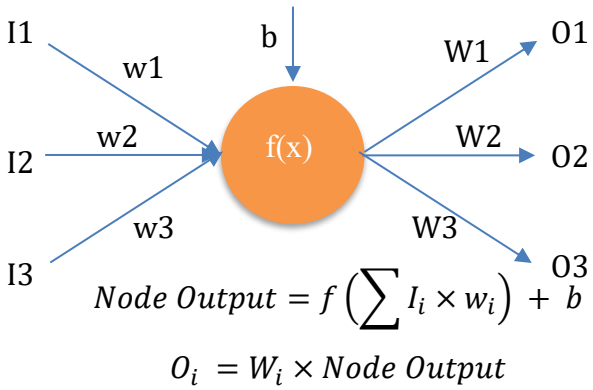


Figure 4-3: Schematic of an artificial neuron demonstrating its function

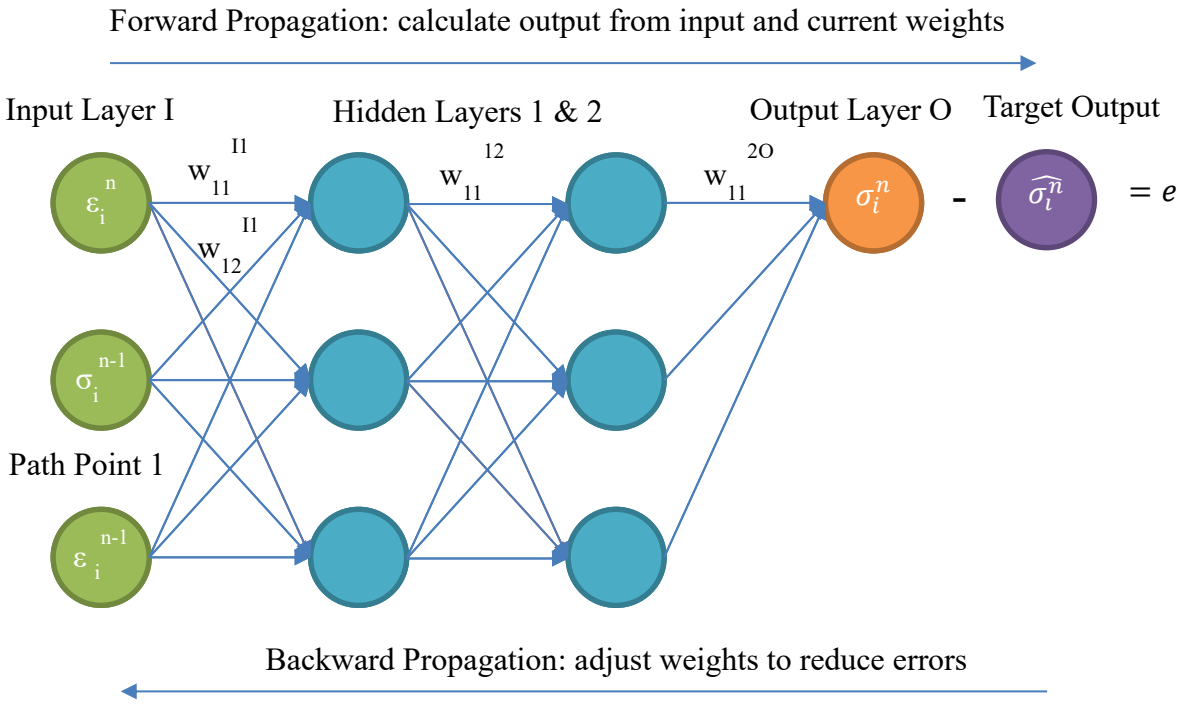


Figure 4-4: Schematic of a fully connected artificial neural network that calculates stresses from current strains and the stresses and strains of the previous point.

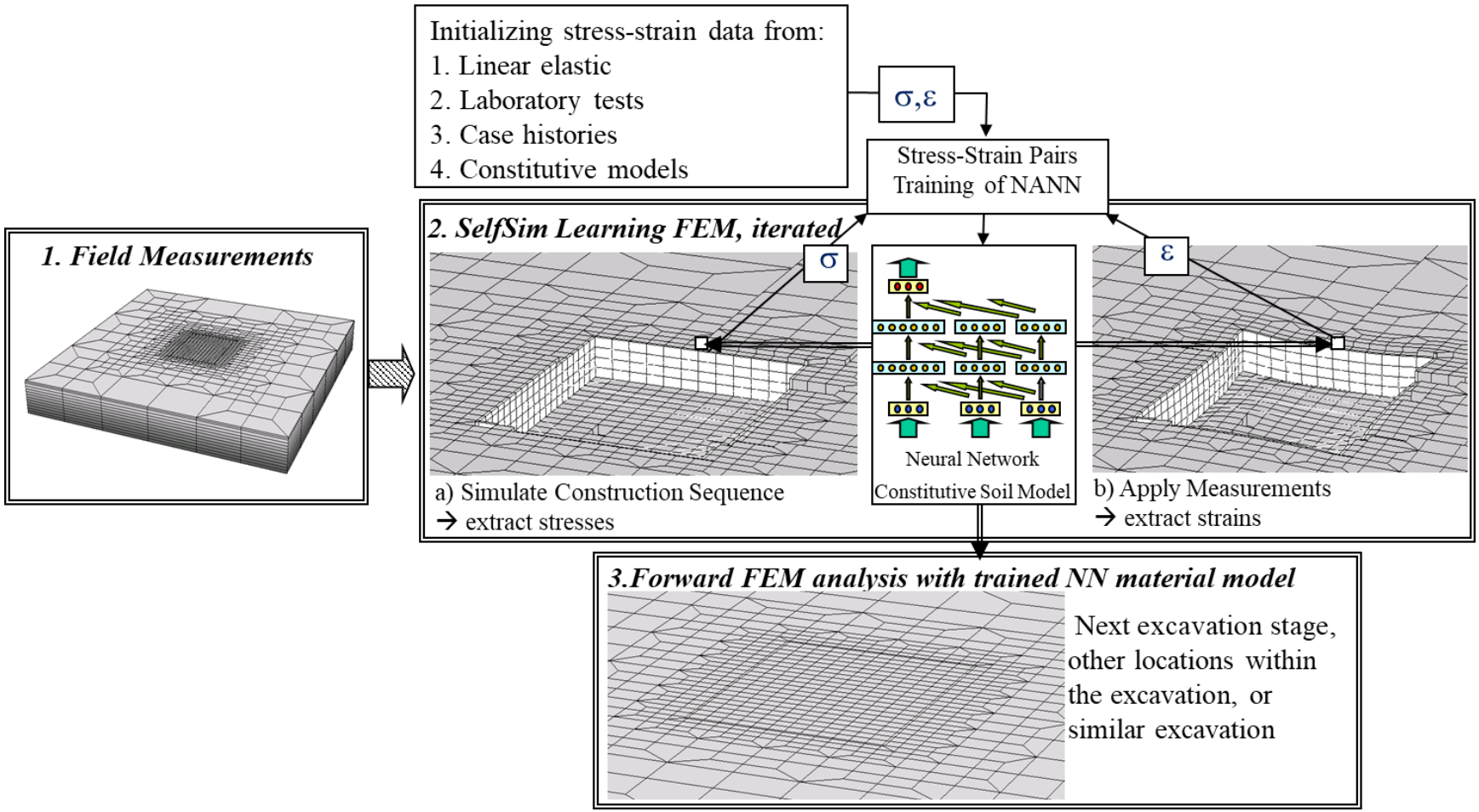


Figure 4-5: Schematic showing the steps of a SelfSim analysis performed for an excavation simulation problem (Hashash 2014)

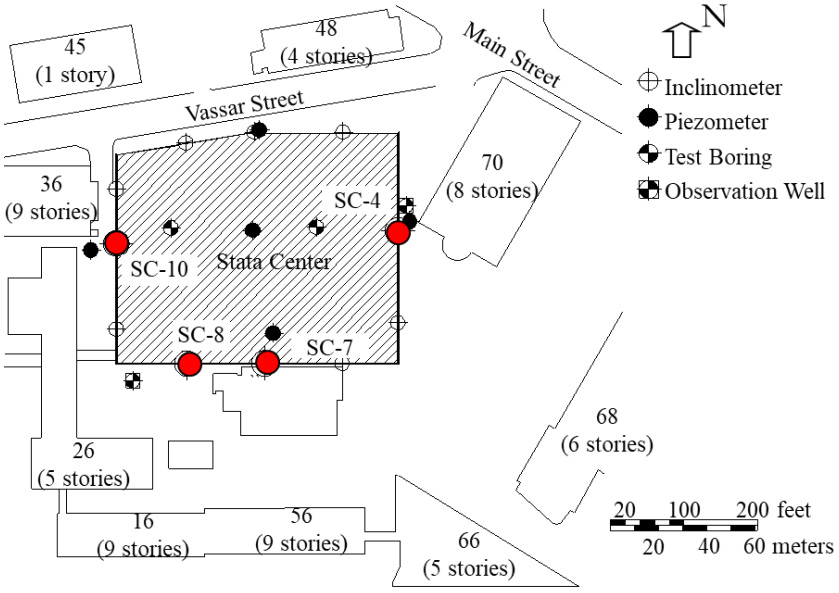


Figure 4-6: Plan view showing the excavation and the monitoring points locations. Source: (Marulanda 2005)

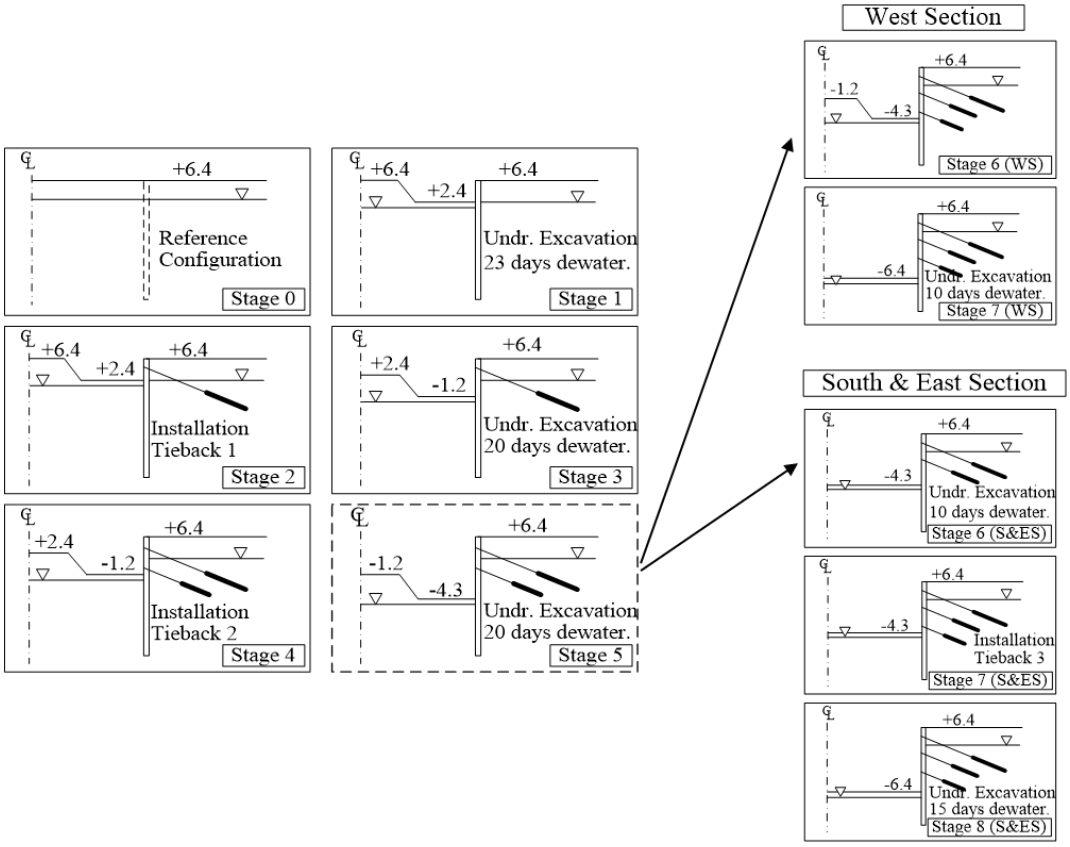


Figure 4-7: Schematic describing the construction stages used for the different sections of the excavations. Source: (Marulanda 2005)

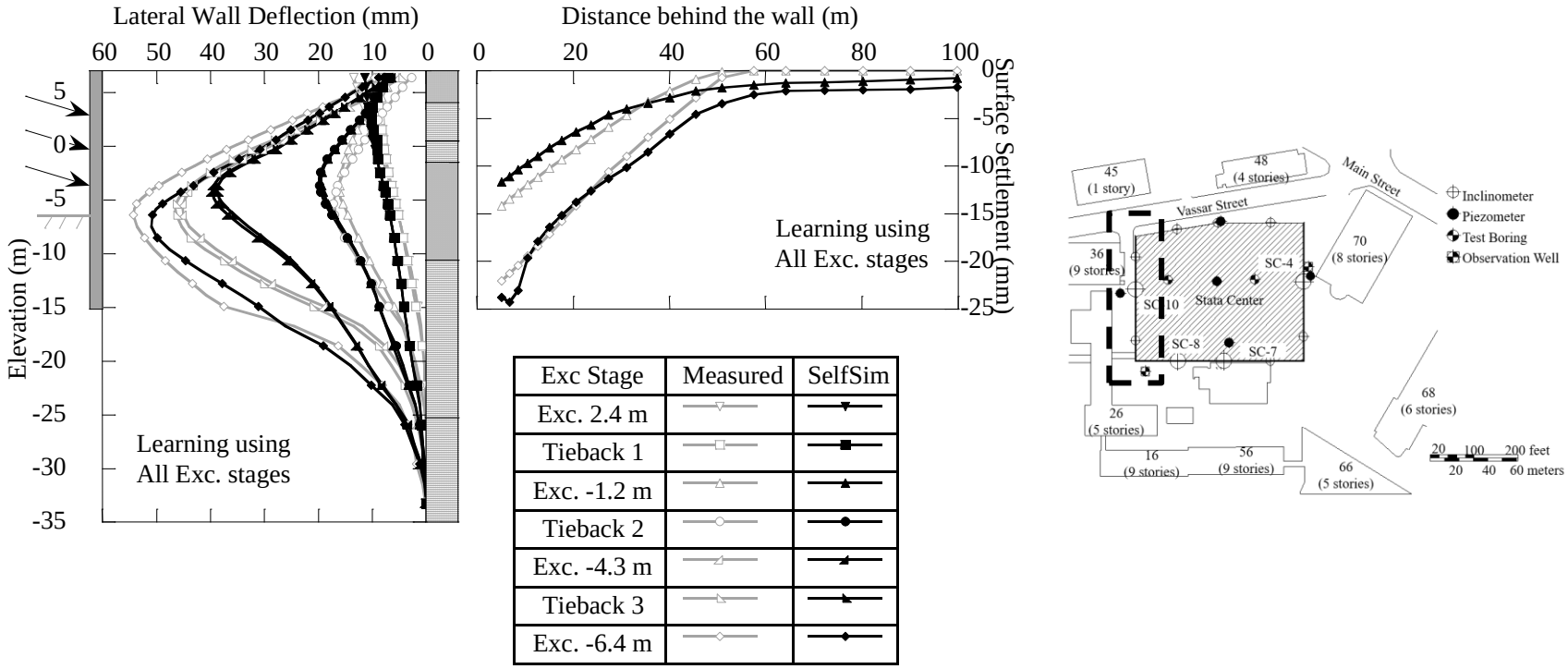


Figure 4-8: Comparison of the estimates of the FEM analysis employing the ANN trained through SelfSim using the results of all construction stages at the western section of the excavation. Source: modified from (Marulanda 2005)

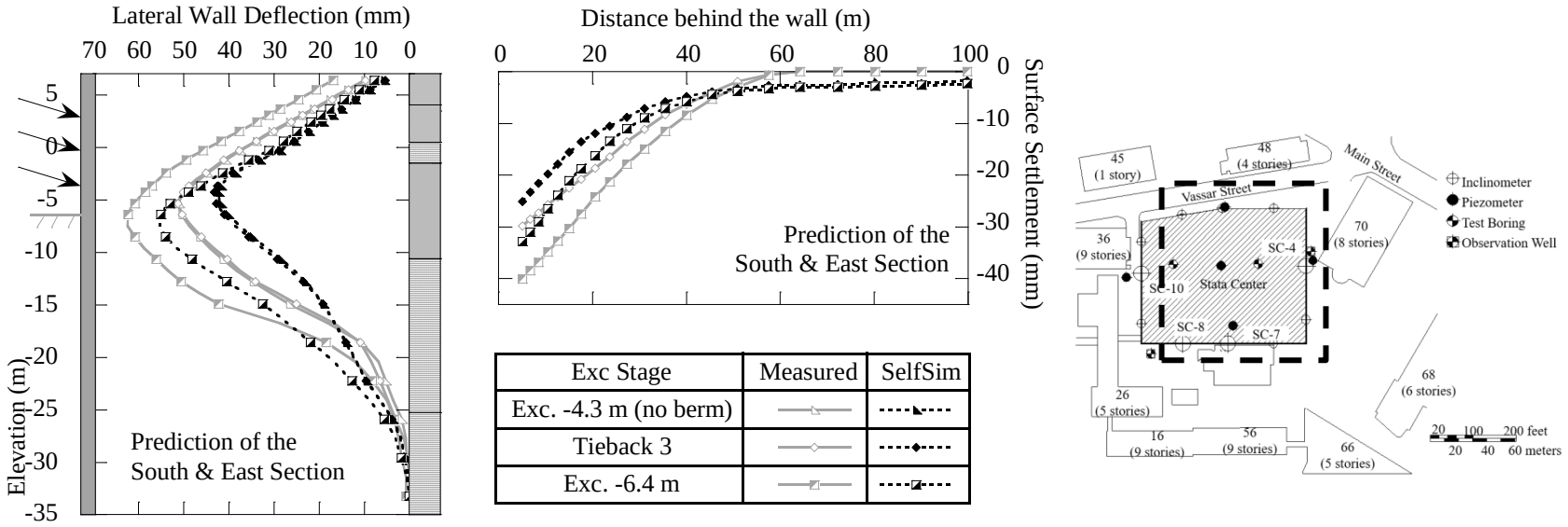


Figure 4-9: Comparison of the estimates of the FEM analysis performed for the southern and east section of the excavation, employing the previously trained ANN. Source: modified from (Marulanda 2005)

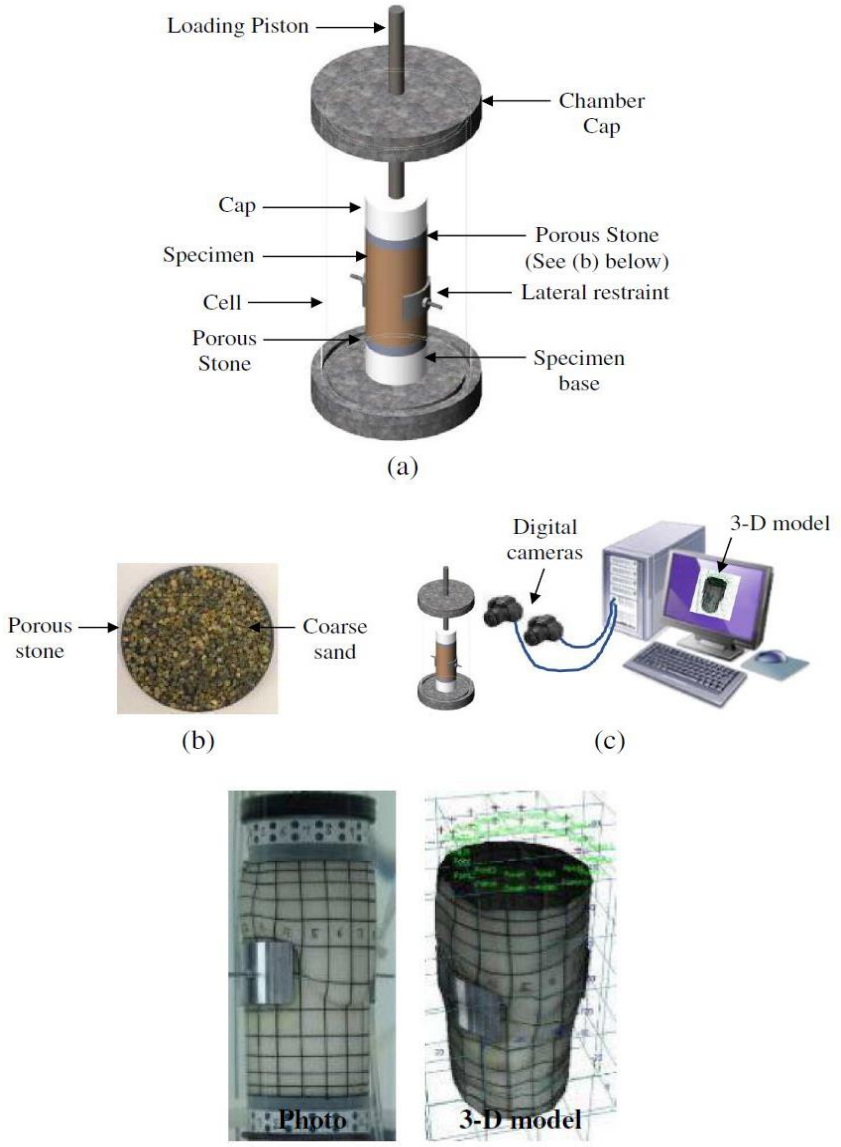


Figure 4-10: Photos of the modified triaxial test setup used by (Hashash and Asmar 2018) for testing then conducting SelfSim analyses on Ottawa sand. Source: (Hashash and Asmar 2018)

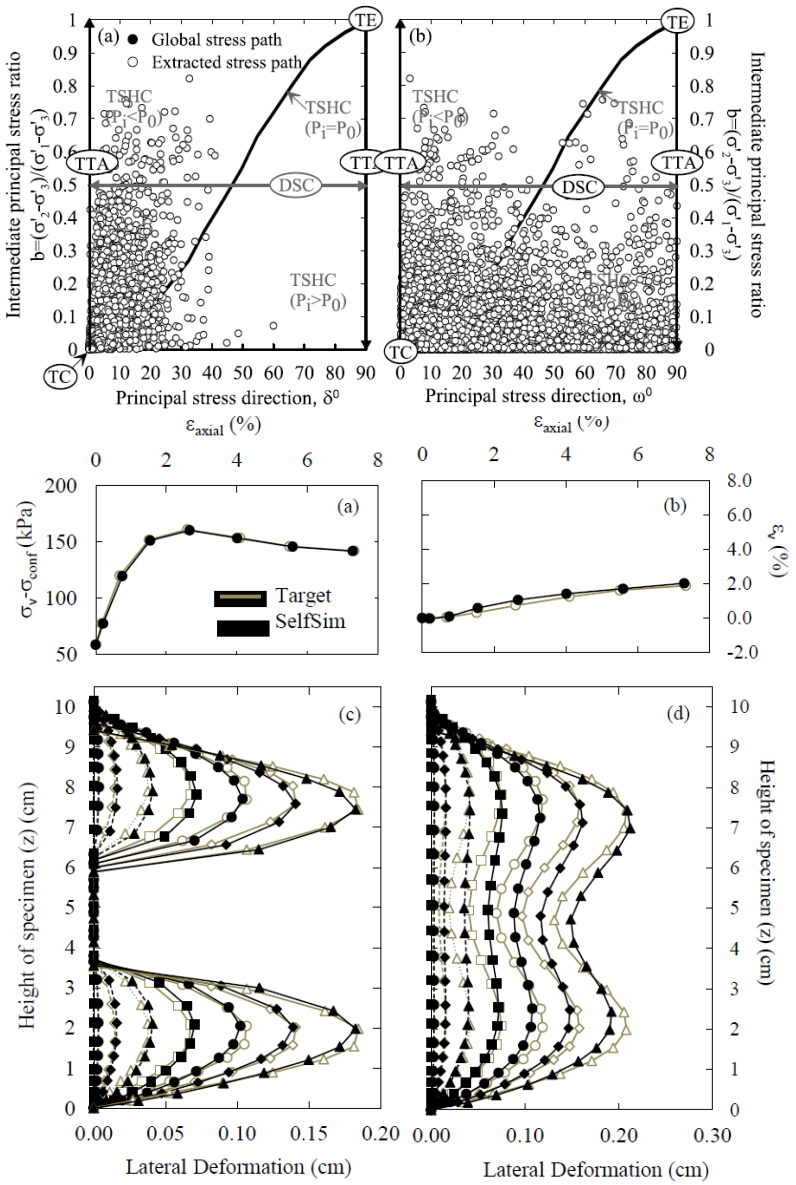


Figure 4-11: A sample of the results of the SelfSim analysis performed by (Hashash and Asmar 2018) for a modified triaxial test.

CHAPTER 5. Development of a Wireless Through-Soil Communication System

The design of the new wireless through-soil communication system went through different stages. First, the minimum requirements of the communication system were determined. Next, the channels available for communication were explored and the channel more suited to the design requirements was selected. Then, the communication system was designed in an iterative process that included quantifying the channel properties through lab and field testing, adjusting the communication system to deal with the channel's challenges then attempting to communicate data while checking the data rate and error rate. A prototype of a new transmitter, optimized for the design requirements and channel properties, was then developed. The communication system's performance was finally demonstrated through large scale data communications.

5.1 Design Requirements

While developing a communication system that works for any underground application would be a great achievement, channels usually impose constraints on design that force optimizing systems based on the application requirements. The primarily application being targeted in this design is geotechnical underground instrumentation systems. However, the methodology and findings can be employed in different designs, optimized as needed for other applications.

Geotechnical underground instrumentation systems usually include sensors, such as piezometers, inclinometers, accelerometers, etc. installed in boreholes at varying depths and horizontally spaced as needed depending on the project size and monitoring requirements. Multiple sensors might be installed in a single borehole. Geotechnical instruments are mainly used during the construction of a geotechnical project, and rarely after the completion of construction.

Considering an average project duration of 1 to 3 years, and an average sensor-to-sensor spacing of 20 to 30 ft, the minimum design requirements are selected as follows:

- **range of communication**: the range should extend at least 20 to 30 ft.
- **data rate**: the rate should be enough to send at least one reading (32 bit) a day.
- **energy efficiency**: the energy efficiency should enable the system's batteries to last for the duration of a typical geotechnical project (1 to 3 years).
- **size**: the system should be compact enough to fit in a typical borehole (2 to 4" in)

- **cost:** the system should be feasible compared to other options (redundancy, adding more boreholes …)

5.2 Characterization of Soil Channels Properties and Selection of Channel

Designing a through-soil communication system operational in a certain medium, requires the selection of an adequate communication channel. As discussed in sections 3.4, 3.5, and 3.6 it seems that previous attempts for practically employing the electromagnetic channel were not successful, while attempts employing the mechanical channel seemed more promising. However, a deeper comparison of those two channels should be executed before deciding on this critical design component. The comparison is narrowed to those two channels as they are the main long range wireless communication channels currently known and employed by humans. The chemical channel in which some bio-systems such as the human body communicate using chemical signals is another example for a channel, but its communication is generally short ranged and requires complicated design of the communication components and medium.

Identifying the sources of noise to expect during transmission plays an important role in the selection/design of system components. The major anticipated sources of noise include:

- acoustic noise:
    - natural: wind, rain, etc.
    - man-caused: traffic, machinery, sirens, etc
- electric noise: (Bennett 1954)
    - natural:
        - thermal noise (also called Johnson noise or Brownian motion of electrons): is the electronic noise generated by the thermal agitation of electrons inside an electrical conductor at equilibrium, which happens regardless of any applied voltage. Thermal noise is present in all electrical circuits.
        - shot noise: an electrical noise that is due to the discrete nature of electric charge which makes a perfectly continuous (analog) signal impossible
    - man-caused:
        - interfering signals
        - power hum: is a noise caused by the AC electromagnetic fields from nearby electric equipment and wiring, which usually operate on power-line

frequency of 50-60 Hz. This electric noise is usually picked by audio devices at approximately double the power-line frequency (100-120 Hz) and can be heard as a hum from speakers.

- impulse noise: a category of acoustic noise that includes impulse-like sharp noise. Common causes of this noise include: electromagnetic interference, synchronization issues, lightening and switching transients in channels.

Soil's electromagnetic channel was studied theoretically and empirically while attempting to develop wireless electromagnetic communication systems as demonstrated in section 3.4. In summary, the channel is characterized by a huge attenuation (which increases with increasing saturation levels), multipath propagation and dispersion due to reflections and refractions and scattering caused by its heterogeneity, in addition to noise from transmission lines, antennas and electromagnetic systems.

Soil's mechanical channel was extensively studied by geotechnical engineers, geologists and seismologists. The channel shares similar challenges with the electromagnetic channel: huge attenuation, multipath propagation, dispersion and noise, but at different levels compared to the electromagnetic channel. Mechanical waves' attenuation in soil is due to geometric and material damping. Geometric damping depends on the type and location of the source of waves while material damping depends on the material properties and the wave amplitude (Kim and Lee 2000). Scattering, frictional losses due to grain-to-grain contacts, and viscous losses due to the relative motion between the pore fluid and the sediment frame are the mechanisms that account for most of soil's material damping (Brunson 1983). Energy conversion between different waves and multiple interbed reflections can also introduce significant attenuation (Kibblewhite 1989). Material damping depends on a host of factors, including plasticity (Vucetic and Dobry 1991), strain amplitude, effective mean stress, void ratio, loading cycles, degree of saturation and thixotropic effects (Hardin and Drenevich 1972). Theoretical derivations (Kramer 1996) in addition to laboratory (Oelze et al. 2002) and field (Kibblewhite 1989) tests show that the attenuation coefficient is linearly dependent on frequency. Measurements and formulations for attenuation and damping in soil are abundant in the geophysical (Kibblewhite 1989) and the geotechnical literature (Darendeli 2001, Phillips and Hashash 2009, ASTM-International 2015).

Soil is typically inhomogeneous, non-uniform, and of variable units of irregular boundaries and varying characteristics, hence, longitudinal and shear waves induced by embedded acoustic

sources in surrounding soil follow various paths as shown in Figure 5-1a illustrating a sample stratigraphy from an urban area of San Francisco (McLandrich et al. 2013). Refractions and reflections from layers cause multi-path propagation. Multimodal and multipath propagation result in large transmission delay spreads and in dispersion. Figure 5-1b, for example, illustrates how a single transmitted wave was estimated to travel along 5 different propagation paths in one of the field measurements conducted in this research (section 5.5).

Checking the design requirements in section 5.1, and referring to the results of the available wireless communication systems discussed in section 3.4, it is realized that the primary issue hindering through-soil electromagnetic communication systems is the low range resulting from the very high electromagnetic waves attenuation in soil, therefore, attenuation in soil was further explored.

Theoretical estimates in addition to testing measurements of electromagnetic attenuation in soil (Klein and Santamarina 1997, Silva and Vuran 2010) were compiled in Figure 5-2. The figure also contains mechanical waves attenuation measurements compiled by Kibblewhite (1989). Figure 5-2 shows that electromagnetic waves can benefit from lower attenuation compared to mechanical waves operating at a certain frequency which is favorable. However, to benefit from low attenuation levels relatively low operational frequencies must be employed. The main issue with this is the difficulty of generating low frequency electromagnetic waves with available technologies since generating electromagnetic waves typically require antennas of a length varying between 0.1 to 0.5 the wavelength of the wave.

For example, based on Figure 5-2, a 10-kHz electromagnetic wave propagating in moist sand, would benefit from a relatively low attenuation of 0.1 dB/m, but would require a quarter wavelength antenna size of at least 3,750 m. Acoustic waves of frequencies below 100 Hz, would benefit from the same attenuation. However, those low frequencies can be generated using acoustic transmitters smaller than 30 cm in size (speakers for example).

The major downside of employing low frequencies, is that the communication system would necessarily be limited to low data rates, but since the targeted data rate (section 5.1) is not high, the transceiver size, ease and feasibility of deployment and robustness are more critical than high data rates.

Based on this comparison, it is found that the mechanical channel is better suited for the target application of this research and will hence be employed in the new communication system.

5.3 Communication System Outline

After selecting the communication channel and understanding its challenges, the outline of the communication system was proposed and is shown in Figure 5-3. The proposed communication system can be described as follows: the data to be sent, in addition to parity bits added for error detection and correction, are encoded into a carrier signal via modulation. The modulated signal is transmitted via a transmitter through the channel, incurring noise as it propagates through the channel, before being received by the receiver. Since the signal is impaired by the channel effects, an equalizer is used to reduce or invert those effects. The equalized signal is than demodulated to extract the data. Error correction and detection methods are applied to eliminate bit errors. While different components (devices, code, etc.) were used throughout this project, the outline of the system remained mostly the same.

Communication system components suitable for the channel were then chosen or built. This was an iterative process that included:

- characterization of the mechanical channel through lab and field testing
- adjusting the communication system to deal with the channel's challenges (dealing with noise, energy losses, etc.)
- testing individual components and overall system performance
- attempting to communicate data while checking the data rate and error rate

5.4 Communication System Testing Setups

The laboratory and field setups designed for this process are shown in Figure 5-4 and Figure 5-6 and can be described as:

- In the laboratory setup, a 2×0.5×0.5m sized metal trough equipped with sound dampening wooden plates attached to the inner walls, was filled with uniform clean fine sand (SP) whose gradation is shown in Figure 5-5. The transmitter and receiver would then be embedded in the sand at a selected spacing limited to the length of the trough. This setup enabled quick short-range testing of the communication system or its individual components (transmitters, receivers, modulation schemes, etc.), checking the limits of achievable data-rates, and other quick checks (such as testing of coupling and matching materials and testing the effect of saturation levels).

- Field tests were conducted in two agricultural sites in Champaign, IL. The first site was used for the first two field testing experiments, while the second was used for all other field testing including the large-scale communication testing later shown in section 5.7. The soil in the first site consisted of organic clay, while the soil in the second site consisted of approximately 1 m of organic clay (w_0 = 16, LL = 57, PI = 32), while the layer below it consisted of organic silt (w_0 = 26, LL = 40, PI = 17). Boreholes, 20 cm in diameter, would be drilled using hand augers or a gasoline-powered auger to a depth typically less than 2 m. The transmitter would then be deployed in one borehole while the receiver would be deployed in another borehole (Figure 5-7) at depths ranging between 30 and 150 cm. The transducers (transmitter and receivers) would then be covered with soil, sand, water or mud as described in section 5.5. The setup, which included in addition to the communication system a personal computer and multiple amplifiers, was powered via a gasoline-powered generator. Some of the equipment used in this setup are shown in Figure 5-8. Field tests were designed to target maximum achievable ranges, while experiencing natural soil conditions typically encountered in construction or agricultural sites.

Many commercially available transmitters, which varied in power, size and operational frequency, were used in the laboratory and field testing and include: 30W UW30 183mm underwater speaker (operational frequency: 100Hz – 17 kHz), 135W AQ339 Diluvio 203mm underwater loudspeaker (operational frequency: 20Hz – 17kHz), and 25W DAEX58FP 58mm exciter (operational frequency: 60Hz – 17kHz) shown in Figure 5-9. Those transmitters were used to transform electrical signals (modulated message) into mechanical signals.

Two receivers were used which are: DolphinEar underwater hydrophone (operational frequency 1Hz – 24kHz), and the ACH-01 accelerometer geophone (operational frequency: 2Hz to 20 kHz) with the IB-ACH-01 amplifier shown in Figure 5-10. Those receivers were used to record mechanical signals in the form of electric signals.

5.5 Communication System Testing and Optimization

Any laboratory or field communication testing was always proceeded by a channel frequency response measurement. This is done by sending probe harmonic signals of different frequencies through the channel then recording the signals after propagating through the channel, hence

inferring the effect of the channel on different frequencies. This step guides the selection of optimal data rates and frequency ranges.

The initial laboratory testing of the proposed system employed the UW30 underwater speaker and the DolphinEar hydrophone. The transmitter and receiver were embedded into the sand, 30 cm away from each other. The channel's response was first measured (by sending signals of different frequencies then checking the response of the channel at each frequency). This helped in identifying the frequency range over which the channel is imposing lower losses and hence is more suitable for communication.

Then, for the communication test, a QPSK modulation scheme and an adaptive decision feedback equalization (DFE) algorithm were used. The QPSK modulation allows for higher data transmission rates compared to simpler modulation schemes such as the 2-PSK scheme since each of the 4 phase values (demonstrated in Figure 3-6) represents two bits of information. Using a more data efficient modulation scheme such as the 8-PSK was not possible because of the scatter measured in the phases of the received symbols, and hence using the 8-PSK would significantly increase the likelihood of errors. The DFE equalization algorithm is a nonlinear equalization algorithm that may from previous symbols decisions to enhance decisions on future symbols, and hence its performance is expected to improve with increasing number of transmitted symbols. Since the soil mechanical channel is expected to experience small changes over short durations such as minutes or ever days, equalizers that account for the channel variability were not considered at this stage.

Operating at a frequency range of 250-750 Hz and a data rate of 1kbps data rate, the communication succeeded. The results of the channel response estimation test and the communication test are shown Figure 5-14. This initial testing demonstrated the viability of this acoustic communication system at transmitting data wirelessly through soil at a high data rate and using acceptable power consumption. However, the range was low, and the challenge was to increase the range. Later laboratory testing succeeded in increasing the range to 1 m, after which field testing was required to check for longer ranges.

The initial field testing was performed in the first field testing site. To increase the chances of longer-range communication, a more powerful transmitter (the 135W AQ339 Diluvio 203mm underwater loudspeaker) was used. The transmitter and receiver were lowered into the drilled boreholes, before being covered with he drilled local soil. Operating at a 100-200 Hz or 300-400

Hz frequency ranges, the communication up to 4 m was successful. Operating at a 500-600 Hz frequency range, communication up to 2 m was successful, while for a 4 m range, the received signal was dropping below the noise level. Operating at a 700-800 Hz frequency range, communication up to only 1 m was successful, while for a 2 m or 4 m range, the received signal was dropping below the noise level. While this communication range was higher than that achieved in the field, the signals measured were typically weak suggesting that large data losses were incurred between the transmitter and the receiver. To address this, multiple things were attempted:

- Employing lower frequencies characterized by better channel's response
- Employing lower data rates by increasing redundancy
- Understanding the noise and attempt to reduce its effects
- Enhancing the coupling between the transducers and the surrounding soil, by providing a better interface material that would allow more energy to be transmitted (and hence less energy is lost at those interfaces)

For the next two field tests, the UW30 underwater speaker was used (after realizing its shape was enabling it to better couple with the surrounding soil) and was operated at around 200 to 260 Hz frequency at a low data rate of 10 to 20 bits per second. With those changes, an error free transmission range increased to around 7 m, while error-contaminated range of 15 m was achieved. The noise issue was then explored. Although the second testing field is in a low noise area, the hydrophone was recording decent amounts of noise. It was suspected that this is an electric noise rather than acoustic noise, especially that high voltage electric transmission lines pass nearby. The electric noise could also be incurred from the other components of the system (the amplifiers, the long wires, etc.). To check this, the hydrophone was wrapped (shielded) with an aluminum foil which would work as a grounding for electric noise. The recorded noise dropped significantly. This correction would now enable weaker signals to be recorded, as they will be stronger than the noise level. In addition, the geophone was tested, but it was found that the DolphinEar hydrophone was more sensitive that this geophone, especially after it was shielded; hence, the hydrophone was employed in all following testing.

Many coupling materials including local soil, water, bentonite, and clean sand were than explored. Coupling refers to the degree to which energy is transmitted between materials. For example, an air gap between the transmitter and the adjacent soil will cause a lot of energy to be lost via air

compression, and hence air is a poor coupler between the transmitter and the soil. The quality of coupling between two materials is described through the power transmission coefficient which is the ratio of the power reaching the interface between the two materials to the power exiting it (Kinsler et al. 1999). The power transmission coefficient (T) is given by:

$$T = \frac{4z_1 z_2}{(z_1 + z_2)^2}$$

where z_1 and z_2 are the characteristic impedances of the first and second materials; z is defined as:

$$z = \rho V_p$$

where ρ is the density and V_p is the pressure wave velocity in the material.

A coupling material is usually needed between two materials if a good contact between those two materials cannot be established, or if the impedance mismatch between them is too big. In the latter case (similar to the gel material put between the ultrasound probe and the human body) the coupling material has an impedance in between the impedances of the two materials being coupled. Water and bentonite were first tried as coupling materials between the transducers and the adjacent soil. The results deteriorated compared to the previous testing whose coupling material was local soil. Clean sand was then tested in the lab as a coupling material, and it showed a superior performance. While the local soil is chunky and can leave many large gaps between the transducers and the adjacent soil, the sand better fills the gaps and conform to the surfaces of the transducers and the borehole. Figure 5-11 shows the bentonite and sand which were tested.

A further reduction of the operational frequency range and a further increase of the transmitter output power were then explored. To achieve that, the 250Wrms-rated ButtKicker LFE tactile shaker, capable of generating vibrations lower than 100 Hz, was purchased. While the transmitters used previously may operate at this frequency range, this range is far from their optimal operational frequency and hence their output power is much lower at this frequency range. A custom fiberglass casing was built for this new transmitter to enable embedding it into soil. A photo of the new transmitter is shown in Figure 5-12. Using this transmitter, clean sand coupling, hydrophone shielding and operating at 20-80Hz, the communication was tested in the field again. Multiple hydrophones were installed at varying distances from the source (transmitter) as shown in Figure 5-13. A maximum communication range of 50 m was achieved.

Figure 5-14 shows a summary of the achieved results in lab and field settings (illustrated in Figure 5-13) at this stage of the research. The estimated frequency response is shown for two channels, one in the laboratory with a range of 30 cm, and another in the field with a range of 25 m. In the laboratory, characterized by its short target range, a frequency range of 500-1500 Hz was adopted, based on its frequency response illustrated in the first row of Figure 5-14. In the field, characterized by its long target range, a frequency range of 20-80 Hz was adopted, based on its frequency response illustrated in the second row of Figure 5-14. Since the received signals were heavily distorted from propagation and contaminated by noise, several post-processing operations were applied at the receiver. First, high-frequency thermoelectric noise was filtered out through low-pass filters implemented in both digital and analog means. To mitigate effects of multipath propagation, an adaptive decision feedback equalizer, which nonlinearly compensates the effects of delayed responses of previously transmitted symbols, was used. Error correction was then applied. The constellation diagrams shown in the figure show that symbols were properly identified in both lab and field settings. No inter-symbol interference persisted after equalization and error correction.

This summary showed that both laboratory and field wireless through-soil communication were achieved. Despite this success, the developed system is not suitable for actual deployment in geotechnical projects due to many reasons primarily related to the transmitter size and power consumption. This is explained in detail in section 5.6 and motivated the development of a new compact and power efficient transmitter.

5.6 Development of New Small Transmitter

Based on the design requirements presented in section 5.1 and the lessons learned from the previous laboratory and field testing, it is found that for a wireless transmitter to be feasibly used for underground applications, it should be:

- capable of delivering frequencies suitable for the channel response and target data rate
- small enough to fit inside typical boreholes
- power-efficient
- water-tight
- have an impedance matched to the soil's impedance to maximize power transmission
- capable of delivering sufficient acoustic power over a practical range

- and compatible with the chosen modulation scheme

Attenuation measurements from the literature and from our channel characterization tests confirmed that lower-frequency waves suffer less attenuation. Based on measurements in this research, frequencies ranging between 10 and 40 Hz attained ranges and data rates that are more suitable for the long-range target applications. Common loudspeakers are unable to effectively deliver frequencies in this range, and transmitters that can deliver such frequencies are necessarily characterized by larger dimensions that exceed typical borehole diameters of 2 to 4 inches. Larger speakers also have greater contact areas with the soil over which power is distributed, hence reducing mechanical wave amplitudes. Common loudspeakers also suffer from coupling issues, since their short strokes are incapable of ensuring contact with the often poorly compacted soil, especially since excellent compaction around embedded transmitters is hard to achieve in small-sized boreholes. This coupling issue can result in severe energy losses even before the signal is transmitted into the soil. Short strokes prevent speakers from accumulating high kinematic energy that is converted to high amplitude stress waves. While common loudspeakers are designed to match air impedances and underwater speakers are designed to match those of water, there are no commercial speakers designed to match the impedance of soil, which is characterized by high spatial variability. Those limitations of the available speakers suggested the need for designing a new transmitter.

A new transmitter design, that employs a voice coil actuator (VCA) characterized by a maximum stroke of 19 mm and a maximum continuous power of 18 W, was designed. The transmitter uses two controllers, powered by 4 Li-ion batteries, and encased in plexiglass. A 12 mm x 12 mm plexiglass cap was attached to the actuator head, thus reducing the area of contact with soil and increasing the amplitude of the stress wave. The transmitter is sufficiently small to fit into a 4-in-diameter borehole. A power-efficient on-off keyed (OOK) modulation scheme was employed; a logical ‘1’ is represented by a single or multiple mechanical impulse generated by hammering the motor head against soil, while a logical ‘0’ is represented by the absence of any impulse. The modulation order can be modified by changing the number and location of impulses, while the period of each impulse is fixed by design, i.e., in our design, this period was set at 1/24s. The motor provides 18 W strokes sending ‘1’ symbols and consumes no power sending ‘0’ symbols, resulting in 9W average power consumption, assuming equivalent priors of ‘on’ and ‘off’ states. In addition to its power efficiency, this scheme is needed as fully continuous signals cannot be delivered using

the new transmitter (since the thumper travels some distance inside the casing before hitting the soil) and hence using a phase-controlled modulation scheme such as the QPSK is very hard. This modulation is An Arduino Uno 3 microcontroller board was chosen to deliver the electric control signal to the motor controller. The main components of this transmitter are shown in Figure 5-15. The size of the new transmitter relative to a 4-in diameter plastic pipe section is demonstrated in Figure 5-16.

5.7 Large Scale Communication Testing

The developed small transmitter is used for data transmission in the field. A sample transmitted signal traveling 6 m along this channel is shown in Figure 5-18. The signal, which is 3.3 seconds long, is OOK-modulated. The electrical signal is transformed into a mechanical wave: a +1 amplitude is transformed into 4 full repetitive thumper strokes. The transmitted signal is then received by a hydrophone. It is however distorted from the original transmitted signal. The equalizer adjusts the effects of the frequency-dependent scattering from multi-path propagation and attenuation, transforming it to a close approximate of the transmitted signal. The symbol decoder finally makes decisions on the symbols of the equalized outputs, mapping soft symbols having values larger than 0.5 into logical '1', and symbols having values smaller than 0.5 into logical '0'. The sample signal shown is recovered without error.

Performing a larger scale test, the system was used to transmit data of a piezometer cluster composed of two piezometers. The data included piezometric elevation readings taken every 2 hours over 275 days. The whole time-history, composed of 74,000 bits, was sent within 6 hours including resting times of the system to prevent the circuit and motor from overheating. Readings were collected via hydrophones 3, 6, and 10 m away from the transmitter. The transmitter and the receivers were embedded at a depth of 0.5 m. Results corresponding to the 6-m hydrophone are presented in Figure 5-19. The signal was successfully restored but suffered from bit-errors reflected in the observed spikes. To mitigate these residual errors, forward error correction (FEC), was adopted to detect and correct bit errors.

An 11-kb photo was also sent using the same system. This is described in Figure 5-20 where received data over 10-m distance is illustrated in the form of a histogram: before equalization, many '1' and '0' symbols were overlapped, causing errors in symbol decisions, and even after applying the equalization technique, a number of bit errors persist. Forward error correcting codes were adopted to resolve residual errors in the equalizer output. Based on this setting, a 4-bps rate

for ranges up to 10 meters was achieved. The figure shows the raw restored image where only 0.17 percent bit errors occurred, and the error correcting code was successful in correcting those errors. Figure 5-21 shows a summary of the data rates and energy efficiencies achieved in lab and field using different transmitters and configurations.

5.8 Integrating the Communication System in Geotechnical Instrumentation Systems

With the new transmitter, an underground wireless geotechnical instrumentation system similar to that illustrated in Figure 5-17 can be built. A geotechnical instrumentation system can benefit from such a system by simply equipping each embedded sensor with an acoustic transmitter, while equipping an embedded data logger with an acoustic receiver and a programmed processing unit. The acoustic transmitter may be integrated within the sensor body, connected to it via a short wire, or via electromagnetic communications which are functional over very short ranges, depending on the sensor requirements and installation conditions. The system can include as many sensors and data-loggers as convenient, as long as the spacing is limited by the maximum communication range and transmission is coordinated to avoid interference.

With the energy efficiency achieved, if the small transmitter is used to transmit 3 readings equivalent to 96 bits every day (which is sufficient for a geotechnical piezometer, for example), the current prototype can serve for 45 months before its Li-ion batteries are exhausted. A single receiver may cover a sphere 10 m in radius, acquiring information from possibly tens of transmitters installed within this radius.

All embedded geotechnical sensors may benefit from this system including piezometers, accelerometers, pressure cells, extensometers, geophones, joint meters, tiltmeters, settlement sensors.

5.9 Limitations of the Work

While this research demonstrated the viability of wireless through-soil communication, it had some limitations that should be highlighted to guide future testing and development of this system and similar systems:

- Most testing was performed along the horizontal direction. The transmitter and receivers were embedded at similar depths and directed towards one another. In such settings, a lot of the major paths of the transmitted wave can lie in one horizontal layer, which is favorable from an energy conservation point of view. That is because, when a wave propagates from

one layer to another, some of its energy is reflected or refracted, due to the impedance difference between the two layers as described in section 5.5. If most of the propagation is happening in one layer, those energy losses are reduced. On the other hand, communicating in the vertical direction, the waves will most likely encounter layer to layer interfaces, since the typical soil is closer to horizontal than vertical. Considering this added cause of energy loss, higher transmission energy might be needed. This however might be offset by the decreased attenuation in stiffer deeper soil layers as described next. Future testing would quantify the effect of layering on the energy loss and the communication system performance.

- All testing was performed in shallow depths limited to 2 m. Geotechnical sensors are typically deployed at larger depths. Despite the lack of deep depths testing due to logistic limitations, the system is expected to perform well at greater depths, benefitting from typically stiffer soils or rocks, characterized by higher propagation speeds and lower damping ratios, translating into lower attenuation coefficients. This should be verified via future testing at medium to deep depths.
- All field testing was performed in unsaturated soil layers. Some laboratory tests were performed in saturated sand, and communication succeeded in those tests. Field testing below the ground water table would show the performance of the system in fully saturated soils.
- Most tests were performed in low acoustic noise conditions. However, some tests were performed while the gas-fueled electric generator was operational and producing intense vibrations, or during windy or rainy days. The system performed well – all bits were correctly restored - in these situations, except for receivers very close to the generator (in a radius of 3m), in which case the noise saturated the receiver dynamic range masking any received signal. The shallow installation depths used in testing magnified this issue, but since such shallow installations are not typical, this issue is not anticipated to be problematic for systems deployed at medium to deep depths. Transmission timing algorithms, exploiting low-noise periods, for example when noisy equipment is not operational (as in night hours), or periodic noise-filtering algorithms can be applied to overcome this issue. Timing algorithms can be also employed to avoid interference of waves from different transmitters operating simultaneously.

- Although only point-to-point communication scenarios were considered in this study, multiple-input-multiple-output (MIMO) techniques can be applied to network scenarios.
- While the proposed small transmitter design showed superior energy efficiency, compactness, and suitability for target frequency ranges, when compared to commercial off-the-shelf speakers, the design can be further optimized in many ways. Exploiting the nonlinearity of the soil layers (Haupt and Rolt 2005) may enable design of substantially more directional transmitters avoiding most of geometric losses in propagation.
- Preliminary laboratory and field testing in this research showed the possibility of estimating the pressure wave velocity via processing the received signal and comparing its time of arrival with the time of transmission of the transmitted signal. Future studies may further explore and demonstrate this feature, hence benefitting from the communication system for geophysical testing purposes.

5.10 Findings and Recommendations

A wireless through-soil communication systems was developed. The mechanical channel was found to be more suitable for underground sensor systems and was hence employed in the communication system. The system was tested and optimized through a set of laboratory and field testing. Using a large powerful transmitter, data communication up to range of 50 meters without deploying additional relay nodes was demonstrated. For practical deployment, a new transmitter, small enough to fit in a typical 4-inch-diameter borehole, was also developed, and a data communication data up to range of 10 meters was demonstrated. With the new transmitter, the communication system is projected to have a life span exceeding 3 years, which is estimated to be enough for most geotechnical projects.

This system although developed mainly for geotechnical applications, might be used for many other applications including agriculture, mining, environmental testing and monitoring, seismology, animal studies, etc. Underground sensors that may benefit from this system include piezometers, accelerometers, pressure cells, extensometers, geophones, hydrophones, joint meters, tiltmeters, settlement sensors, soil humidity sensors, water quality sensors, soil contaminants sensors, soil pH sensors, underground thermometers, etc.

Further testing of this system is required before being adopted by the industry, including tests along the vertical direction, tests at medium to large depths, and tests in saturated soil deposits. The system might be further optimized for range, data rate and data consumption.

5.11 Figures

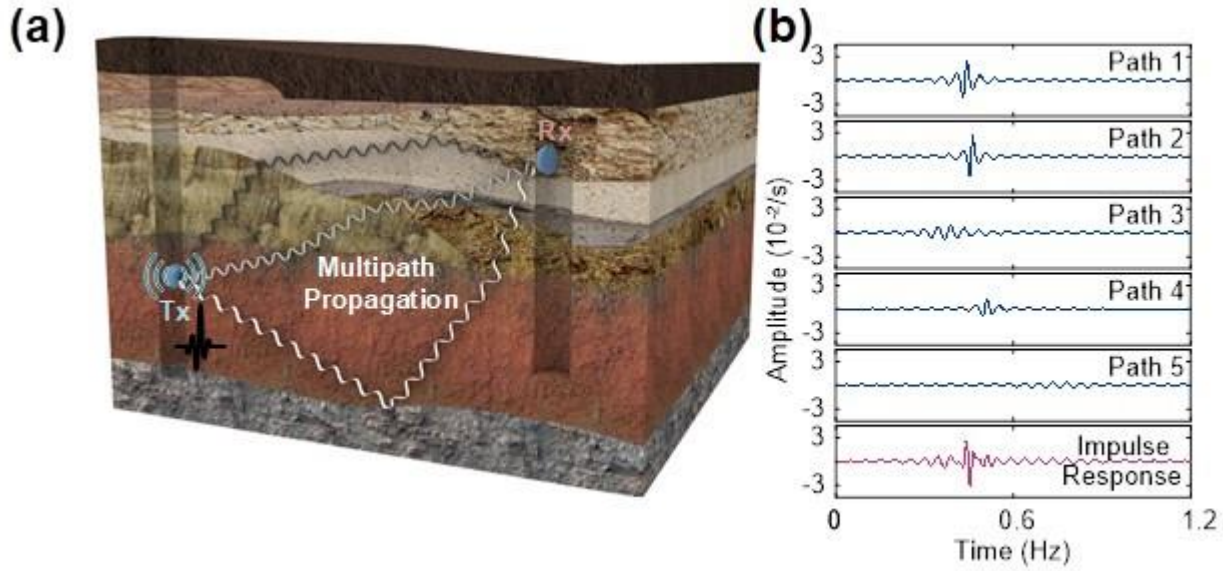


Figure 5-1: a) Schematic of multipath propagation of acoustic waves in a complex underground stratigraphy b) Sample resolved impulses of individual propagation paths. Source: (Yang et al. 2019)

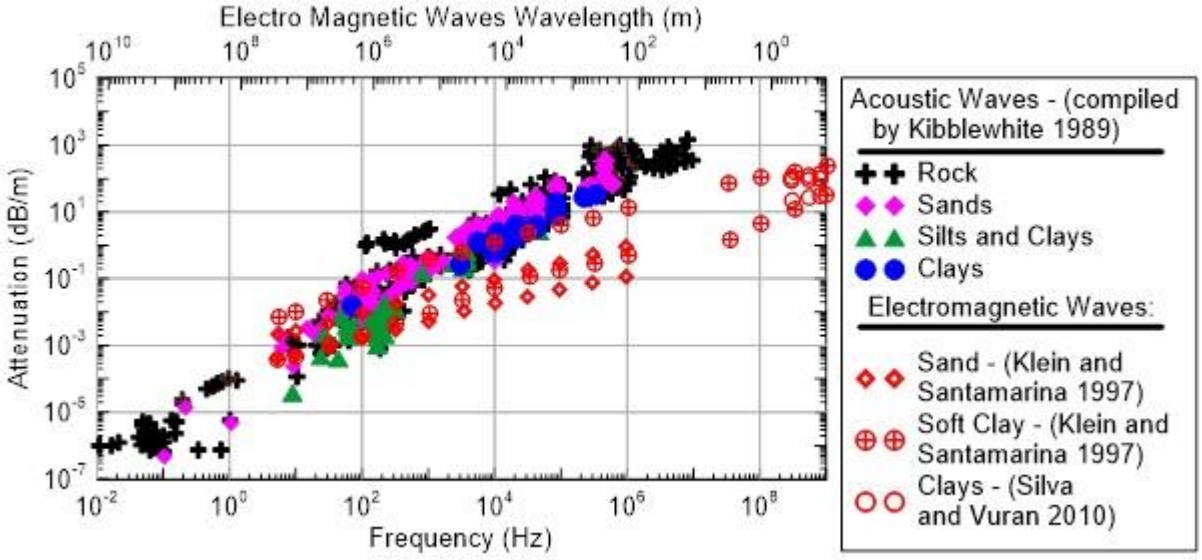


Figure 5-2: Soils attenuation measurements of acoustic and electromagnetic waves. Source: (Yang et al. 2019)

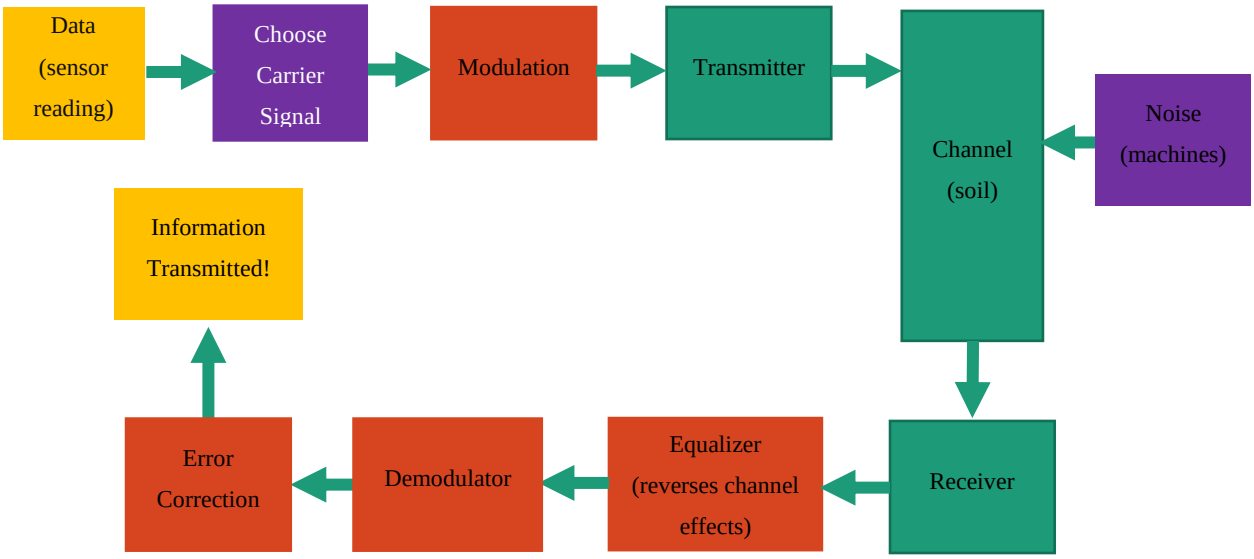


Figure 5-3: Elements of the proposed communication system



Figure 5-4: Photo of the sand trough used in the laboratory testing

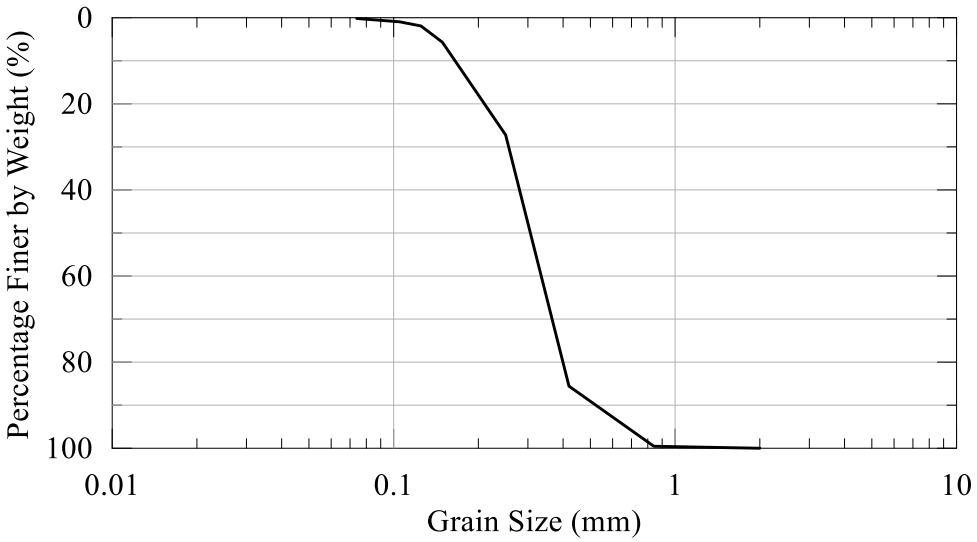


Figure 5-5: Gradation of the fine sand using in the laboratory testing.

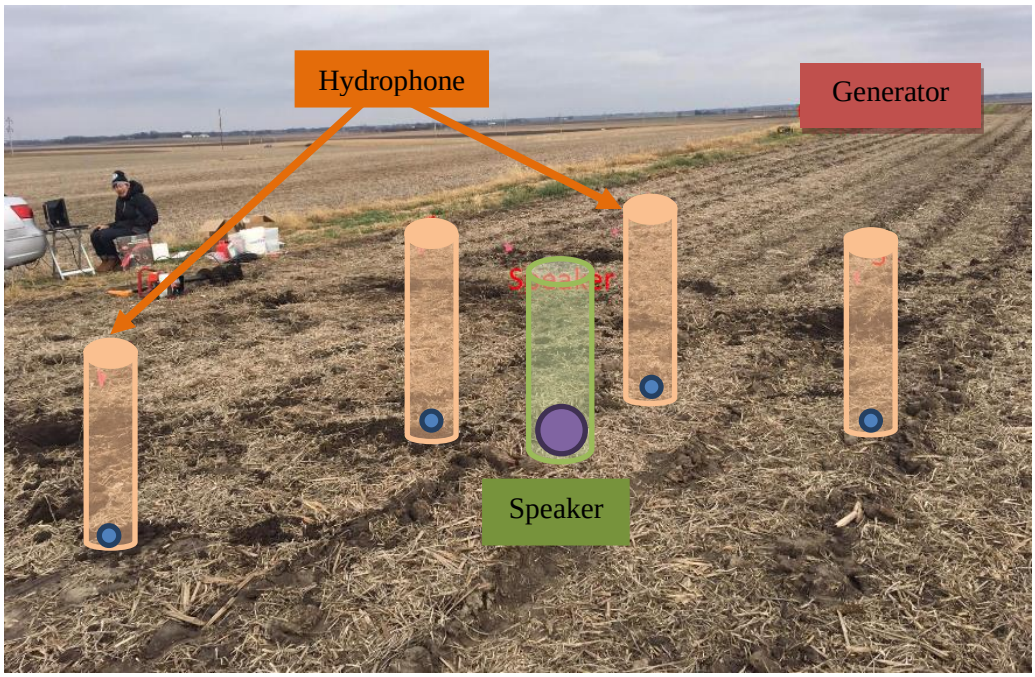


Figure 5-6: Photo showing the field testing setup



Figure 5-7: Left: Photo showing one of the boreholes drilled for the transmitters and receivers. Right: photo showing the same borehole after the speaker has been deployed inside it, and then filled with soil.



Figure 5-8: Photos showing some of the equipment used in field testing. Left: hand augers. Center: gas-powered auger. Right: gas-powered electric generator.



Figure 5-9: Photos showing some of the transmitters used in this project. Left: 30W UW30 183mm Underwater Speaker (operational frequency: 100Hz – 17 kHz). Center: 135W AQ339 Diluvio 203mm Underwater Loudspeaker (operational frequency: 20Hz – 17kHz). Right: 25W DAEX58FP 58mm Exciter (operational frequency: 60Hz – 17kHz)



Figure 5-10: Photos showing some of the receivers used in this project. Left: DolphinEar PRO hydrophone (operational frequency: 1Hz to 24 kHz. Right: ACH-01 accelerometer with IB-ACH-01 amplifier (operational frequency: 2Hz to 20 kHz).



Figure 5-11: Photos of some of the coupling materials used in field testing. Left: bentonite mud. Right: clean uniform fine sand.



Figure 5-12: Photo of the 250 Wrms low frequency generator encased with a custom fiberglass casing

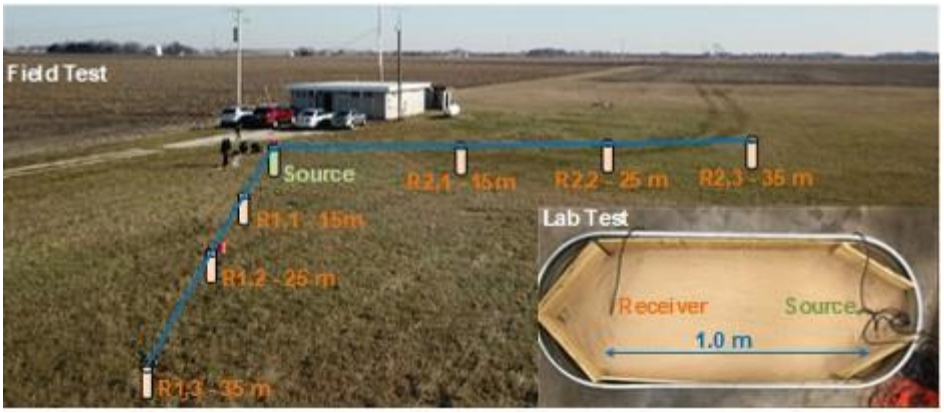


Figure 5-13: Photos showing lab and field test settings and transducer locations. The symbol “R” refers to receiver (hydrophone). Source: (Yang et al. 2019)

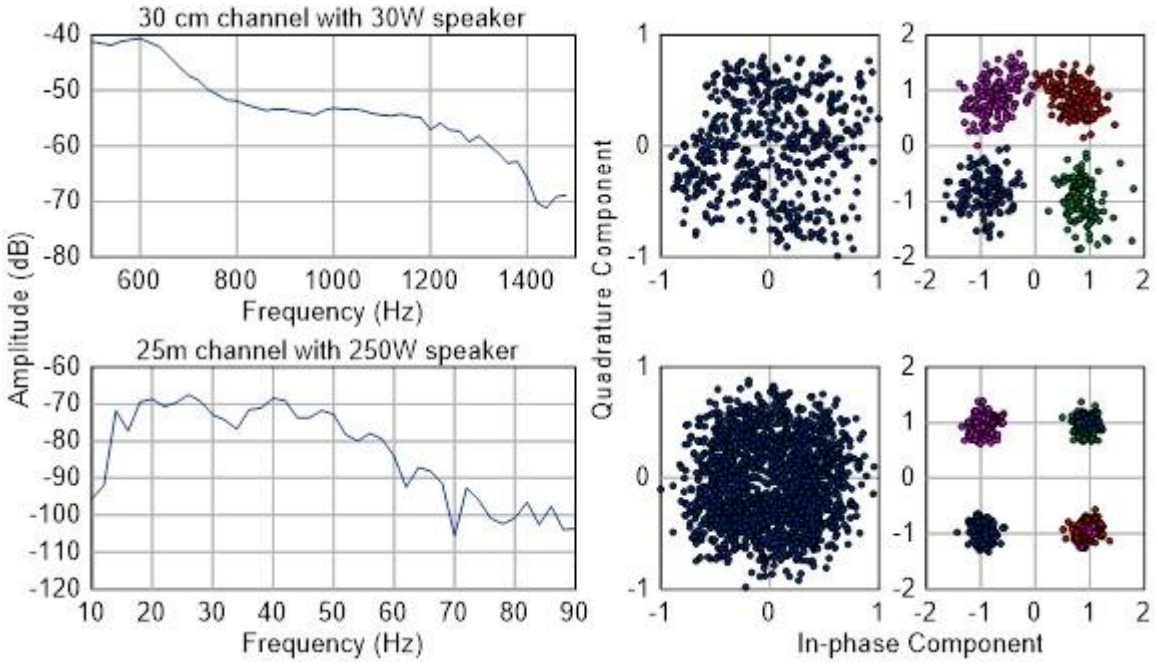


Figure 5-14: Channel response (left), constellation diagrams of received symbols before (center) and after (right) the equalization are shown for soil electroacoustic channels using the 30 W speaker tested in lab over a range of 30 cm (top) and using 250 Wrms speaker deployed at 50 cm depth at the field area over a range of 25 m (bottom). Source: (Yang et al. 2019)

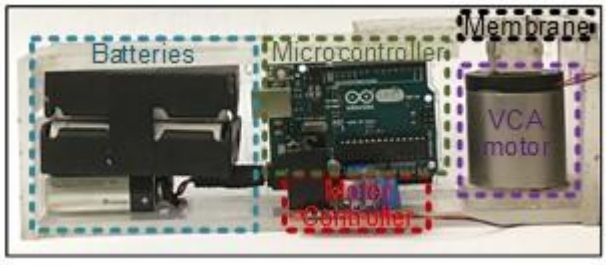


Figure 5-15: Photo of the built small transmitter. Source: (Yang et al. 2019)

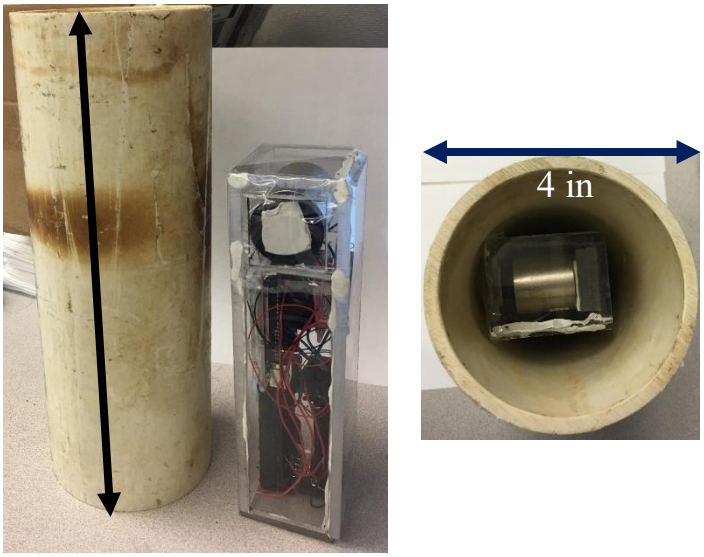


Figure 5-16: Photos showing the size of the built small transmitter relative to a 4-in diameter plastic pipe section.

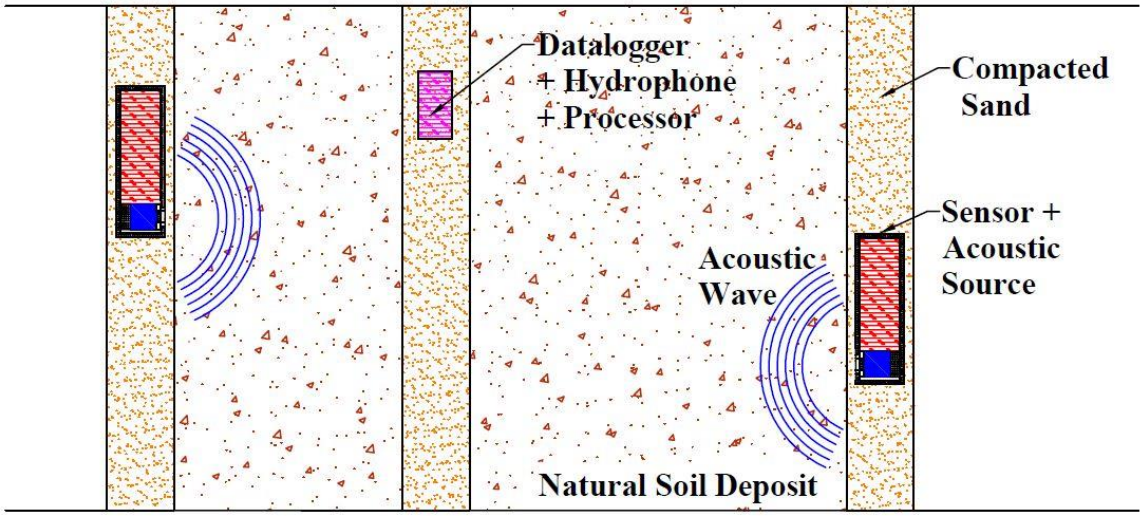


Figure 5-17: Proposed Schematic of the Underground Communication System

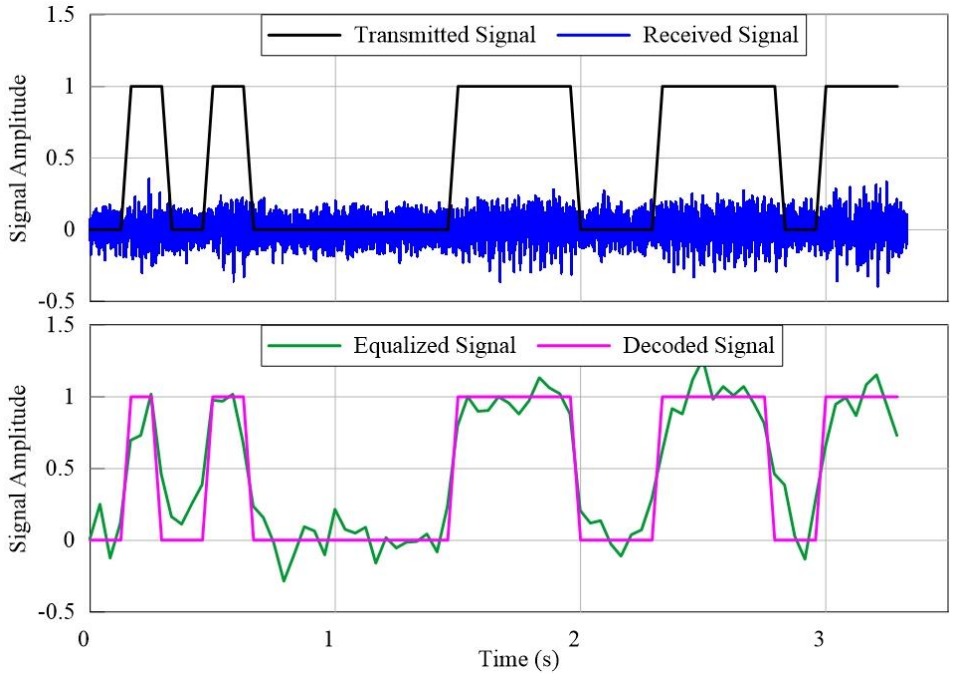


Figure 5-18: Plots of a sample transmitted signal, and its corresponding received signal (top), which was then equalized then decoded (bottom). Source: (Baltaji et al. 2019)

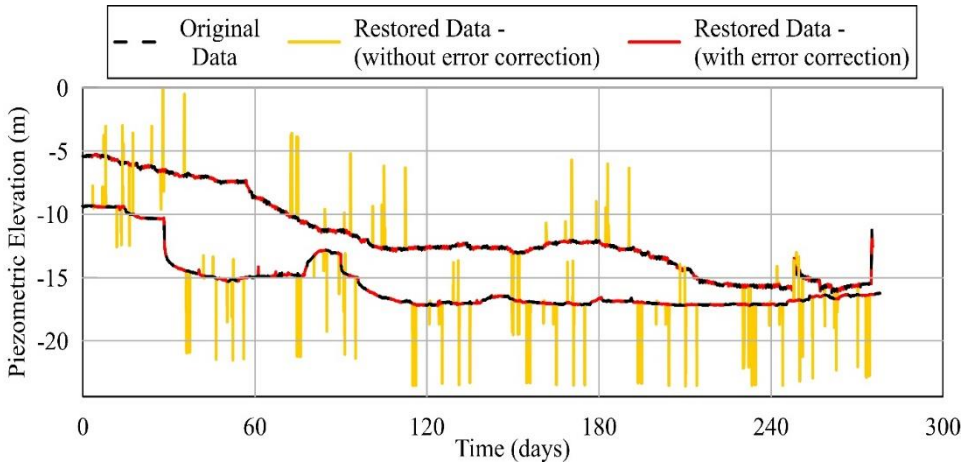


Figure 5-19: The piezometric data, sent, received and recovered. Source: (Baltaji et al. 2019)

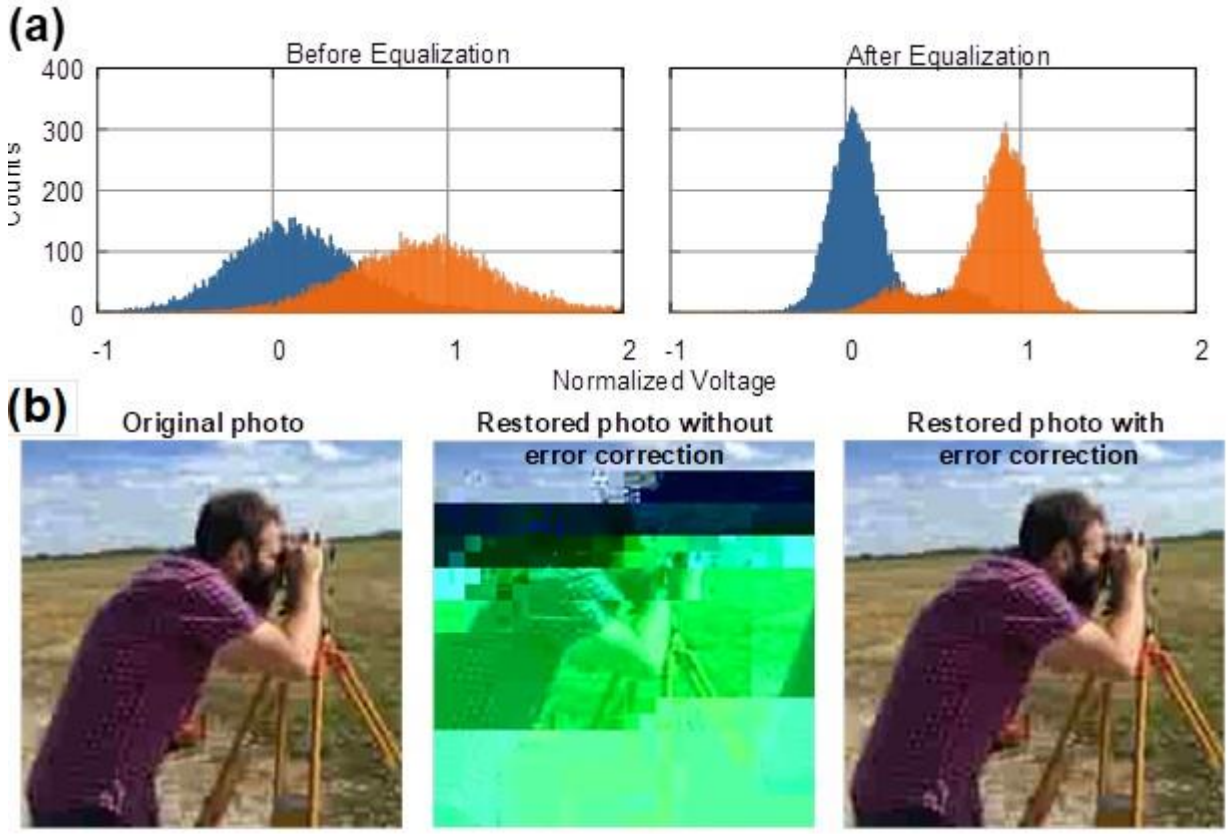


Figure 5-20: Illustration of results of the On-Off keyed system operated with the proposed small transmitter. (a) histograms of restored symbols from 10-m data transmission using the small transmitter (b) the original photo sent (left), the restored image without error correction (center), and the restored image after error correction (right). Source: (Yang et al. 2019)

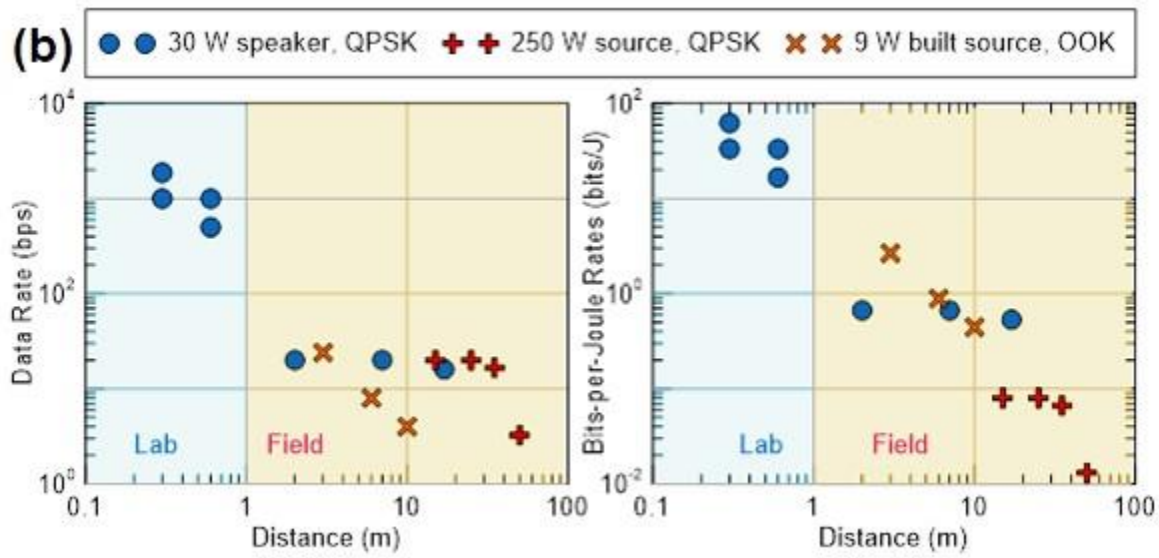


Figure 5-21: Performance of soil acoustic communication using commercial speakers and new transmitter design employing a VCA motor: Data rates vs. Distance (left), and Bits / Joule vs. Distance (right). Source: (Yang et al. 2019).

CHAPTER 6. Development of a Computationally Efficient Machine-Learning-Based Material Modeling and Inverse Analysis Platform (SelfSim Gen3)

The SelfSim framework is composed of 4 core components: a FEM software in which an ANN material model is implemented, a pre-post processor that processes the results of the 2 FEM analyses then compares them to each other and prepares the training dataset, a training module that trains the ANN model using the training dataset, a control panel that marches the inverse analysis per in the inverse algorithm described in section 4.4. Those components are illustrated in Figure 6-1. The first generation of the code written by Ghaboussi and his co-authors employed the autoprogressive algorithm and the ABAQUS. The code was then adjusted by Hashash and his co-authors to adjust the autoprogressive algorithm, define an ANN model tangent stiffness, and to incorporate porewater pressure measurements. The process of developing the third generation of the code is described in this chapter. It is worth mentioning that other generations of the code were written by Ghaboussi and his co-authors to adapt to biomedical applications (Hoerig, C., Ghaboussi, J., & Insana, M. F. (2018). Data-driven elasticity imaging using cartesian neural network constitutive models and the autoprogressive method. *IEEE transactions on medical imaging*, *38*(5), 1150-1160.).

6.1 Identification of the Causes of Slowdown of the SelfSim Gen2 Code

Assessing the SelfSim Gen2 code, many flaws and causes of slowdown can be identified. The Gen2 code was written in UIUC by a few researchers and was independently modified over years. This independent development made the code busy, very hard to read, poorly commented and having many redundancies. Since the researchers were not coding specialists, the code's data structures, algebraic operations, algorithms, and order of execution were not optimized for speed, memory efficiency and parallelization.

The training module has many features that can cause slowdown:

1. the auto-node generation, in which the ANN adaptively adjusts its architecture, adding or dropping nodes, in order to learn new behavior or to eliminate low-importance nodes. This process might consume a lot of time just to reach an acceptable architecture. This might be justified in a research context, but once an architecture is identified, it should be used without having to learn it every time. Alternatively, a suitable architecture can be simply identified by trial and error.

2. the nested structure of the ANN in which history points are added in a way that includes freezing then unfreezing of training of nodes, and controlled connectivity of nodes. While this feature is an interesting approach in training the ANN, it creates asymmetric neural networks which are harder to deal with computationally during forward propagation and back propagation.
3. the random initialization of ANN weights which might significantly slowdown the training. Many other weight initialization techniques have significantly better performance.
4. the training algorithm used (Rprop), which has been improved and even superseded by other more efficient algorithms.

The user-defined material model suffers from a critical disadvantage which is the need for recompiling the whole executable every time the ANN is retrained. This compilation alone might take minutes, which is alone a huge delay. Another major issue is the non-vectorized implementation of the forward propagation and stiffness matrix assembly.

Part of the data preprocessing and postprocessing is done manually which delays setting up the analyses and then evaluating their results. Previous users of the code reported difficulties in setting up the analysis due to the complexity of the interface.

The new code should employ optimized training libraries developed by coding professionals and must be well commented to be easy to read, and stream-lined to eliminate redundancy. The auto-node generation and nested architecture is not to be included in a typical application of the code. State-of-the-art weight initialization and training algorithms should be employed. The need for recompiling the executable after each training should be eliminated. Computationally expensive parts of the code should be vectorized when possible. Additional code for preprocessing and postprocessing will be developed to minimize the need for manual processing of data. The user interface should of the code should be simple allowing for fast set up of analyses.

6.2 Development of the Training Module

Given the potential of neural networks in different applications, many companies including Google, Facebook, and Microsoft in addition to coding communities, have made available many artificial neural network code libraries that are highly optimized by coding experts. Those code libraries are fast, memory efficient, parallelizable, and capable of working with CPUs, GPUs and

even application specific integrated circuits (ASICs), specifically designed for their purpose, called tensor processing units (TPUs).

TensorFlow (Abadi et al. 2016) is an ANN code which was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015. TensorFlow provides stable Python and C++ application programming interfaces, but its core is completely written in C++ which makes it fast. In 2018, TensorFlow had the most GitHub activity, Google searches, books on Amazon and ArXiv articles. It also has the most developers using it and is listed in the most online job descriptions. TensorFlow has been adopted by several giant companies such as Airbus, Twitter, IBM, and others. With the wide community interaction and contribution and the continuous Google support, this framework has exceled quickly since 2015, and is expected to benefit from continuous developments and enhancements. Due to those metrics, the TensorFlow ANN library was employed in training ANNs in the new SelfSim framework.

The training module contains many elements:

- A linear elastic training file generator: which takes an input the elastic moduli and uses them to generate a linear elastic training set that is used for pre-training of the ANN
- A time history generator: which takes a set of time-tagged stress and strain states and generates a time history containing a predefined number of past stress and strain states at every time. This is useful when the behavior being learned is path dependent.
- ANN model constructors: which takes user input to create and initialize the ANN. The input required include the number of nodes in each layer and the activation functions to be used. This code uses non-random weight initialization which was shown to accelerate the ANN training.
- ANN trainers: which takes a training data set then acts on the constructed ANN employing backpropagation and optimization techniques to adjust the weights of the ANN, hence training it to produce outputs closer to the target outputs.
- An ANN evaluator: which compares the estimates of an ANN with the target estimates and produces comparison graphs and metrics.
- An ANN model reader: which reads an existing ANN model. It is useful when training of an ANN is to be resumed rather than restarted when new data sets become available.

- An ANN model parameters generator: which generates keyword files that can be read by the FEM software used.

Note that most of the ANN construction and training functions were written for both current versions of Tensorflow: v1 and v2.

The neural network types available include nested and regular networks (without nesting).

Loss functions include in addition to the mean-squared error, used in previous version of SelfSim, a variety of other loss functions including mean absolute error, Huber error, mean-squared logarithmic error, etc.

A variety of optimizers are were selected from the optimizers library in Tensorflow, including the state-of-the art ADAM (Kingma and Ba 2014) and RMSProp (Hinton 2012) optimizers. Old optimizers such as gradient descent and stochastic gradient descent are also included. RProp (Riedmiller and Braun 1993), which is the optimization algorithm used in the previous version of SelfSim is not supported in Tensorflow, so a custom implementation of RProp was added to the code to support this optimizer. This might be useful for comparing efficiency of RProp to other newer optimizers when used in SelfSim.

The code also includes 2 functions for calculating the initial stiffness matrix of the generated ANN model. This is useful for checking how the stiffness matrix looks like before using the ANN in a real analysis. The first function implements the original formulation of the stiffness matrix presented in (Hashash et al. 2004). The second function implements a vectorized version of the formulation, discussed in section 6.3.1.

The weight initialization scheme used is the (Glorot and Bengio 2010) uniform weight initialization scheme (also called Xavier initialization). This allows for faster convergence of the training compared to random weight initialization.

Regularization of the neural network is also supported to resist overfitting of the neural network and prevent larger weights from dominating the training. L2 regularization as well as dropout regularization were included.

The main differences between the training modules in SelfSim Gen2 and Gen3 are summarized in Table 6-1.

6.3 Development of the LS-DYNA User-defined Material Model

The SelfSim employs an ANN-based constitutive model integrated in a FEM framework, hence a suitable FEM framework should be selected. The previous version of the SelfSim employed the Abaqus FEM software, but recent experience with the LS-DYNA software in dynamic problems showed many of its advantages. LS-DYNA is a general-purpose finite element program used by the automobile, aerospace, construction, military, manufacturing, and bioengineering industries. LS-DYNA is optimized for shared and distributed memory Unix, Linux, and Windows based, platforms. The code's origins lie in highly nonlinear, transient dynamic finite element analysis using explicit time integration. An implicit solver was also added to the software. LS-DYNA is characterized by its speed, memory efficiency, friendly user interface, and lower price (compared to Abaqus). Based on this, LS-DYNA is employed to house the SelfSim-type FEM analyses.

To be capable of running Self-Sim type analyses, an ANN-based constitutive model should be implemented in LS-DYNA. LS-DYNA provides a user-defined material modeling package which was used for this purpose. The ANN-based constitutive model should be capable of:

- Reading the structure and parameters of the trained ANN
- Conducting matrix calculations of forward propagation to estimate stresses increments from strain increments
- Calculating the tangent stiffness to enable implicit analyses

After training the ANN, two files are generated. The first file is a keyword file which should be included in the analysis keyword file, and it contains the name of the second file, and the number of nodes in the longest layer, in addition to initial shear and bulk moduli required by LS-DYNA to calculate initial timesteps. LS-DYNA reads this file and allocates memory based on it. The second file contains the definition of the neural network parameters including its structure, weights, biases, activation functions, and stress and strain scalars.

A main improvement in this code is the fact that updating neural networks parameters or even structure does not require re-compiling the whole code, as was the case in the previous version. The recompilation used to take minutes and was a main reason for the low speed.

The code currently supports neural networks with up to 3 hidden layers of any size. Allowing for more than 3 hidden layers if needed is very simple and requires changing few parameters inside the solver then compiling it, but is not used here to reduce the memory allocated by the solver.

6.3.1 Vectorization of the Code

Three different levels of computational parallelism are available to improve computational performance of a code: Instruction-Level Parallelism ILP (multiple instructions of the same thread are executed concurrently), Thread-Level Parallelism TLP (execution of instructions in multiple threads of a processor or in multiple processors), and Data-Level Parallelism DLP (the same instruction is applied concurrently to different inputs (vector), a process called single instruction multiple data – SIMD, or vectorization) (Basics of Vectorization for Fortran Applications). DLP is illustrated in Figure 6-2. Vectorization is supported by most modern processors but require data to be stored contiguously in the memory. ILP is usually done by hardware or compiler developers. TLP is well understood, and modern software is written to allow parallelizing computation over multiple threads or cores. For example, the TensorFlow library used for the training module, and the LS-DYNA FEM suite both support multi-thread and multicore computations. DLP however is sometimes overlooked in code writing, while it can provide a significant computational advantage. A main feature of the new code is the vectorization of the computationally intensive calculations. The most expensive calculations are the forward propagation and the stiffness matrix calculation; both of which were not vectorized in the old code. The vectorization of the forward propagation is simply achieved by performing matrix multiplication rather than individual weight-to-node multiplication, which is a common practice in modern ANN codes. On the other hand, the vectorization of the stiffness matrix required reformulating the stiffness matrix to support vectorization. This process is explained below.

The final equation for the stiffness matrix as presented in (Hashash et al. 2004) is:

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\varepsilon_j}=\frac{S_j^{\sigma}}{S_j^{\varepsilon}}\beta^3\sum_{k=1}^{NC}\left(\begin{array}{l}\left\{\left(1-\left({}^{n+1}\sigma_i^{NN}\right)^2\right)w_{ik}^{\sigma C}\right\}\\ \times\left[\sum_{l=1}^{NB}\left\{\left(1-\left({}^{n+1}C_k\right)^2\right)w_{kl}^{CB}\right\}\left\{\left(1-\left({}^{n+1}B_l\right)^2\right)w_{lj}^{B\varepsilon}\right\}\right]\end{array}\right)$$

where W_{ij}^{xy} represents the weight connecting node i from layer x to node j from layer y, while B and C are nodal outputs of the first and second hidden layers respectively.

The first step to vectorize this equation is to realize that the weights are matrices, and hence the term that looks like $\left(1-\left({}^{n+1}B_l\right)^2\right)w_{lj}^{B\varepsilon}$ is the weight matrix $w_{lj}^{B\varepsilon}$ whose rows have been scaled by a

scalar equals to $\left(1-\left({}^{n+1}B_l\right)^2\right)$. Let $W_{lj}^{B\varepsilon}=\left(1-\left({}^{n+1}B_l\right)^2\right)w_{lj}^{B\varepsilon}$, $W_{kl}^{CB}=\left(1-\left({}^{n+1}C_k\right)^2\right)w_{kl}^{CB}$ and $W_{ik}^{\sigma C}=\left(1-\left({}^{n+1}\sigma_i^{NN}\right)^2\right)w_{ik}^{\sigma C}$.

The equation above can then be written as:

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\varepsilon_j}=\frac{S_j^{\sigma}}{S_j^{\varepsilon}}\beta^3\sum_{k=1}^{NC}\left(\left\{W_{ik}^{\sigma C}\right\}\times\left[\sum_{l=1}^{NB}\left\{W_{kl}^{CB}\right\}\left\{W_{lj}^{B\varepsilon}\right\}\right]\right)$$

Then converting from index notation to matrix notation:

$$\left[\frac{\partial^{n+1}\Delta\sigma}{\partial^{n+1}\Delta\varepsilon}\right]=\frac{S^{\sigma}}{S^{\varepsilon}}\beta^3W^{\sigma C}W^{CB}W^{B\varepsilon}$$

Thus, the stiffness matrix is calculated simply by matrix multiplication of the scaled weight matrices.

This implementation scales very easily and efficiently when larger networks are used. For example, the stiffness matrix of a network with 4 hidden layers would simply be:

$$\left[\frac{\partial^{n+1}\Delta\sigma}{\partial^{n+1}\Delta\varepsilon}\right]=\frac{S^{\sigma}}{S^{\varepsilon}}\beta^3W^{\sigma E}W^{ED}W^{DC}W^{CB}W^{B\varepsilon}$$

With the most two computationally expensive operations being converted to matrix multiplications, using matrix multiplication functions that support vectorization is enabled. In Fortran, used in the LS-DYNA UMAT, this is done via the Matmul operation, while in Python, this is done via the NumPy.dot operation. In addition to the vectorization, those operations are additionally optimized, to allow faster communication with the memory. For example, the entire matrices will typically not fit in the local memory of a processor (Cache), so those implementations will divide the data in small blocks that fit in the local memory and swiftly execute the instructions. This is much faster than retrieving data (such as node weights) every time a data unit is needed.

To demonstrate the computational gain of the vectorization and efficient memory management of the stiffness matrix assembly, the two ways of assembling the stiffness matrices were coded in Python. The stiffness matrices for a 24x30x30x6 and a 24x50x50x6 neural network were calculated 1000 times using both methods. The timings for the new algorithm were 0.090 and 0.101 seconds respectively, while the timings of the old algorithm were 21.989 and 62.414 seconds. This sums to a speedup ratio of 244 and 617 times respectively as shown in Table 6-2. This speed up is expected to be much more significant with more layers. Since the stiffness matrix

assembly is done at every element, this speedup is expected to have a strong impact on the FEM analysis times.

Note that the first weights matrix which contains the connections between the input layer and the first hidden layer should be handled differently. Before performing the matrix multiplication for calculating the stiffness matrix, the weights connecting the state variable nodes to first hidden layer should beset to zero. This is due to the fact that

$$\frac{\partial\left(\beta\sum_{k=1}^{NSV} w_{lk}^{BSV}\left({}^{n}SV_{k}\right)\right)}{\partial^{n+1}\varepsilon_{j}^{NN}}=0$$

as described in (Hashash et al. 2004).

The order of growth of an algorithm defines the performance (time or space) of an algorithm as its input size grows. For example, the time consumed by an algorithm can linearly grow with its input n, so it is described as an algorithm with an order of growth of O(n); a better algorithm would grow logarithmically with its input and hence has an order of growth of O(log(n)). The order of growth becomes especially important when the input size grows to big numbers.

Both ways of forward propagation and assembling the matrices have theoretically the same order of growth $O(n^3)$ as they are at their core matrix multiplications, yet the vectorization and the efficient memory management alone resulted in significant computational improvements.

6.3.2 Deriving the Stiffness Matrix for Different Activation Functions

The implementation in (Hashash et al. 2004), the calculation assumed the use of the hyperbolic tangent activation function for all layers. However, since then, many more activation functions were introduced, and more activation functions are expected to emerge with the booming of the machine learning industry. Thus, it would be helpful to expand the derivation of the stiffness matrix to other activation functions, in case another activation function is to be used in one or more nodes or layers.

For a fully connected neural network with two hidden layers, the stiffness matrix as presented in (Hashash et al. 2004) is:

$$\frac{\partial^{n+1}\Delta\sigma_{i}}{\partial^{n+1}\Delta\varepsilon_{j}}=\frac{S_{j}^{\sigma}}{S_{j}^{\varepsilon}}\beta^{3}\sum_{k=1}^{NC}\left(\frac{\partial^{n+1}\sigma_{i}^{NN}}{\partial^{n+1}C_{k}}\times\left[\sum_{l=1}^{NB}\frac{\partial^{n+1}C_{k}}{\partial^{n+1}B_{l}}\frac{\partial^{n+1}B_{l}}{\partial^{n+1}\varepsilon_{j}^{NN}}\right]\right)$$

The effect of the activation function appears when taking the derivatives. For example, Since $\frac{\partial \tanh(f(x))}{\partial x} = \left(1-\left(\tanh f(x)\right)^2\right)\frac{\partial f(x)}{\partial x}$ the last term $\frac{\partial^{n+1} B_l}{\partial^{n+1}\varepsilon_j^{NN}}$ equates to $\left(1-\left({}^{n+1}B_l\right)^2\right)w_{lj}^{B\varepsilon}$.

Considering other activation functions is simple:

1- ReLU:

$$\mathrm{Re}LU\left(f(x)\right) = \begin{Bmatrix} f(x) \text{ if } f(x) > 0 \\ 0 \text{ if } f(x) < 0 \end{Bmatrix}$$

$$\frac{\partial \mathrm{Re}LU(f(x))}{\partial x} = \begin{Bmatrix} 1 \text{ if } f(x) > 0 \\ 0 \text{ if } f(x) < 0 \end{Bmatrix} \times \frac{\partial f(x)}{\partial x}$$

2- Sigmoid:

$$sigmoid\left(f(x)\right) = \frac{e^{f(x)}}{1+e^{f(x)}}$$

$$\frac{\partial sigmoid(f(x))}{\partial x} = sigmoid\left(f(x)\right)\times\left(1-sigmoid\left(f(x)\right)\right)\times\frac{\partial f(x)}{\partial x}$$

3- SELU:

$$SELU\left(f(x)\right) = \begin{Bmatrix} f(x) \text{ if } f(x) > 0 \\ \beta(e^{f(x)}-1) \text{ if } f(x) < 0 \end{Bmatrix}$$

$$\frac{\partial SELU(f(x))}{\partial x} = \begin{Bmatrix} 1 \text{ if } f(x) > 0 \\ \beta(e^{f(x)}) \text{ if } f(x) < 0 \end{Bmatrix} \times \frac{\partial f(x)}{\partial x}$$

The only term that changes is the term multiplied by $\frac{\partial f(x)}{\partial x}$. Thus if sigmoid for example is used instead of tanh, the last term $\frac{\partial^{n+1} B_l}{\partial^{n+1}\varepsilon_j^{NN}}$ equates to $sigmoid\left(B_l\right)\times\left(1-sigmoid\left(B_l\right)\right)w_{lj}^{B\varepsilon}$ rather than $\left(1-\left({}^{n+1}B_l\right)^2\right)w_{lj}^{B\varepsilon}$.

The full equation of the stiffness matrix using sigmoid will look like:

$$\frac{\partial^{n+1}\Delta\sigma_i}{\partial^{n+1}\Delta\varepsilon_j} = \frac{S_j^{\sigma}}{S_j^{\varepsilon}}\beta^3\sum_{k=1}^{NC}\left(\begin{Bmatrix} sigmoid\left(\sigma_i^{NN}\right)\times\left(1-sigmoid\left(\sigma_i^{NN}\right)\right)w_{ik}^{\sigma C} \end{Bmatrix} \times \left[\sum_{l=1}^{NB}\times\begin{Bmatrix} sigmoid\left(B_l\right)\times\left(1-sigmoid\left(B_l\right)\right)w_{lj}^{B\varepsilon} \end{Bmatrix}\right]\right)$$

The vectorized version of the stiffness matrix would look the same, however the scalar multiplied by the rows of the weights matrix will take the form of $sigmoid\left(f(x)\right)\times\left(1-sigmoid\left(f(x)\right)\right)$ rather than $\left(1-\left(\tanh f(x)\right)^2\right)$.

Note that this allows the calculation of the stiffness matrix which contains more than one type of activation functions. For example, the hidden layers can be of ReLU type, while the output layer can be of tanh type.

The main differences between the FEM user defined material implementations of the ANN model in SelfSim Gen2 and Gen3 are summarized in Table 6-3.

6.4 Development of the Pre-post Processor Module

The pre-post processor module enables processing of the results of the FEM analyses, automatically reading the analyses output files, plotting behavior and comparison plots then preparing for the next training cycle. All postprocessing functions were written in python, and the plots shown in chapters 7 and 8 are all auto generated by the functions in the pre-post processor. The main differences between the pre-post processor modules in SelfSim Gen2 and Gen3 are summarized in Table 6-4.

6.5 Development of the Control Panel

The control panel is the code that controls the flow of the SelfSim analysis. It starts by defining the ANN, either by asking for user's input to create one, or to read an existing one. The next step is an optional pretraining step, which either uses an autogenerated linear elastic dataset, where the user selected the linear elastic moduli, or a user-defined data set. Once pre-training is done, input files for LS-DYNA are generated, then the LS-DYNA analyses are automatically triggered. When the two parallel analyses are done, their results are processed and used for further training of the ANN, and the SelfSim learning loop continues. The control panel marches the analyses through their steps and repeats this process over the designated number of passes, or until the tolerance criteria are met.

This control panel automates the whole SelfSim analysis and does not require the user's input except when initiating the analysis. If occasionally an analysis was not able to converge, the code automatically repeats the ANN pretraining with the linear elastic dataset, and this is almost always effective in resolving the lack of convergence issue. The ANN is then trained with the extracted

dataset rather than the linear elastic dataset. The linear elastic data is well balanced as it is randomly generated in all directions, while the extracted data might be biased towards one or more directions, depending on the problem, which might make the extraction of a proper stiffness matrix vulnerable to overfitting, which is reflected in the occasional lack of convergence.

The control panel also manages the run files and folders, cleaning them before the analysis starts to prevent mixing of old results with new results.

All the input for the SelfSim analysis is read from the input file SSInput.py, and thus the user would just change the parameters in this file without needing to modify the code. The input parameters are the ANN parameters (architecture, number of path points, activation functions, use of biases), linear elastic set parameters (moduli and levels of strain), strain and stress scalars, LS-DYNA material required parameters (material ID, and density), training parameters (minimum and maximum number of epochs, error tolerance, batch size, optimizer type, loss function type), SelfSim parameters (number of passes, number of cycles, mode of operation, maximum and average displacement error tolerances) in addition to input needed for postprocessing the output of the FEM analyses (number of FEM steps, elements and nodes) and input needed for generating plots (monitored nodes numbers). Additional optional inputs include number of processors and amount of memory to be used per analysis, TensorFlow version to be used, option of regularization of the ANN, enriching the data set, balancing the data set, and saving results to CSV files. The ANN is then trained using actual data, if available, and requires user's input of training hyperparameters, such as learning rate, minimum and maximum number of epochs, tolerance, and optimization algorithm.

The main differences between the control panel in SelfSim Gen2 and Gen3 are summarized in Table 6-5.

6.6 Conclusions

The final code package contains around 5000 lines of python code (around 2000 lines for ANN, 2000 lines for preprocessing, postprocessing and plotting, and 1000 lines for the analysis control and file management) in addition to around 1000 lines of Fortran code (for the LS-DYNA UMAT). The overall performance comparison of the code is summarized in Table 6-6. The computational time for many analyses will be presented in CHAPTER 7 and 0. The overall speedup ratio comes up to between 100 and 1500 times, which could have a significant impact on the usability of this

method in research and industry. The code was tested in CHAPTER 7, then used for real conventional and modified triaxial tests in 0.

6.7 Tables

Table 6-1: Main differences between the training modules in SelfSim Gen2 and Gen3

Attribute	SelfSim Gen2	SelfSim Gen3
Platform	C++; written by UIUC students	Google's TensorFlow v1 and v2 (C++ with python API/wraper)
ANN architecture	nested – not fully connected	Supports nested and fully connected networks
ANN nodes per hidden layer	Adaptive	Fixed (to control computational cost), can be expanded in the future to support adaptive
ANN optimizer	RProp	State-of-the-art optimizers: ADAM, RMSProp … lead to much faster training convergence, in addition to implementing RProp in TensorFlow
ANN weight initialization	random between -1 and 1	Randomly sampled from a uniform distribution with standard deviation capped at $\pm \frac{\sqrt{6}}{\sqrt{n_i+n_{i+1}}}$ where n_i is the number of incoming network connections to the layer (Glorot and Bengio 2010 initialization)
Additional features		ANN model evaluator: evaluates the accuracy of an ANN given a test dataset, and plots comparisons External stiffness matrix calculators: implemented in python to allow user to check the matrix before using the ANN in FEM analyses Regularization and other overfitting prevention: to prevent overfitting of ANN training

Table 6-2: Timing of assembling 1000 stiffness matrices using the old and new assembly formulations

ANN Architecture	**Old Formulation (s)**	**New Formulation (s)**	**Speedup (ratio)**
18x30x30x6	21.929	0.09	244
18x50x50x6	62.414	0.101	617

Table 6-3: Main differences between the FEM user defined material implementations of the ANN model in SelfSim Gen2 and Gen3

Attribute	**SelfSim Gen2**	**SelfSim Gen3**
Platform	Abaqus (Fortran)	LS-DYNA (Fortran)
Forward propagation	Non-vectorized	Vectorized
Stiffness matrix calculation	Non-vectorized	Reformulated and vectorized
Additional features		Support for more activation functions Support for any number and size of hidden layers and any number of path points.

Table 6-4: Main differences between the pre-post processors in SelfSim Gen2 and Gen3

Attribute	SelfSim Gen2	SelfSim Gen3
Dataset management	Databases	Databases, RAM
Additional features		Automatic generation of plots Automatic generation of linear elastic data sets Automatic scaling of datasets Balancing of datasets: previously balancing was done manually in an ad hoc manner. Now balancing is automated.

Table 6-5: Main differences between the control panel in SelfSim Gen2 and Gen3

Attribute	SelfSim Gen2	SelfSim Gen3
Additional features		3 modes of operation Previously a lack of convergence (failure) would require repeating the analysis from the beginning. Now automatic retraining is implemented to overcome lack of convergence Simplified Interface

Table 6-6: Overall performance comparison in SelfSim Gen2 and Gen3

Attribute	SelfSim Gen2	SelfSim Gen3
Overall Performance	A SelfSim analysis for a lab test takes 4 hours if no failure occurs Failures occurred frequently and required manual adjustments to to address them then to restart the analysis. So, the analysis would take about 1 week to complete.	A SelfSim analysis for a lab test takes 5 to 10 minutes if no failure occurs Failures rarely occur, and can be addressed automatically

6.8 Figures

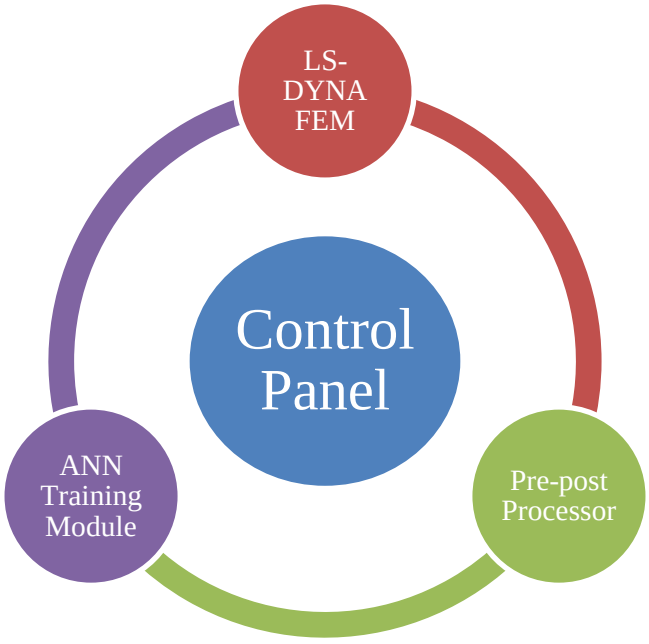


Figure 6-1: Schematic showing the main elements of the SelfSim code

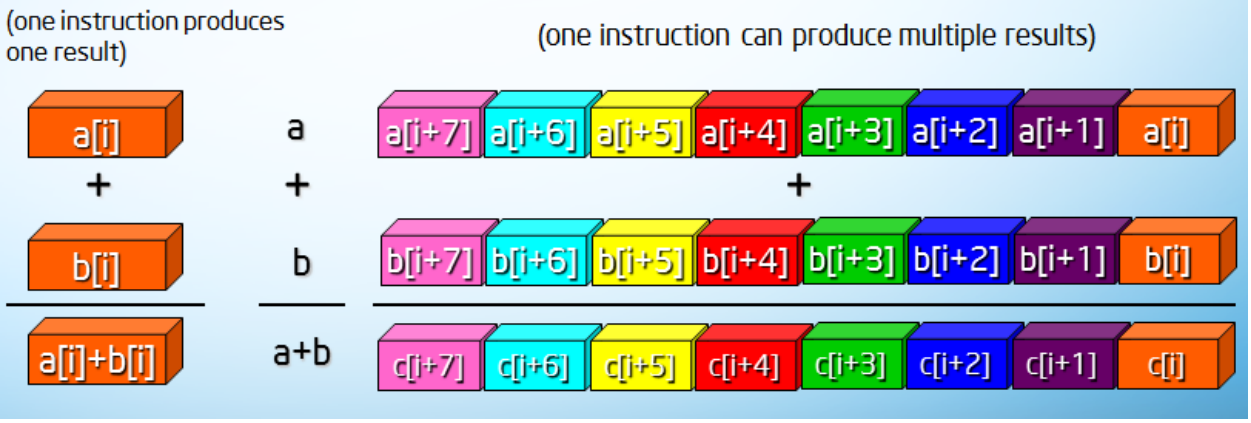


Figure 6-2: Illustration of the function of vectorization. Source: https://software.intel.com

CHAPTER 7. Testing of the New SelfSim Code Using Synthetic Data

The developed code underwent a multi-stage process of testing to verify its functionality. The testing aimed at verifying that the training module can train ANN models to learn material behavior, and that the ANN can be used in implicit and explicit FEM analyses. It also aimed at testing that the SelfSim learning can be performed.

Two triaxial tests were modeled in LS-DYNA: the first model is a conventional triaxial model, while the second is a modified triaxial model in which clamps were added to constrain the lateral deformation, as described in (Hashash and Asmar 2018). Those constrains subject the elements of the model to more complex stress-states, which increases the richness of the extracted dataset.

Two material models were used in each of the two FEM models, resulting in 4 models. The material models are the linear elastic model and power law plasticity model. The 4 models are termed:

1- Conventional Triaxial test modeled with a Linear Elastic material model: CTX-LE
2- Modified Triaxial test modeled with a Linear Elastic material model: MTX-LE
3- Conventional Unconfined Compressive strength test modeled with a Power Law Plasticity material model: CUC-PL
4- Modified Unconfined Compressive strength test modeled with a Power Law Plasticity material model: MUC-PL

The parameters selected for the material models were selected to be representative of loose to medium dense sand. The reason for using an unconfined compressive strength test for the power law plasticity model in LS-DYNA is its inability to be initialized with non-zero stresses in LS-DYNA. The modulus of elasticity for example was selected at 20 MPa in both models.

The testing was performed in two stages: forward analyses stage and SelfSim analyses stage. Table 7-1 provides a summary of the analyses performed in those stages, which will be described in the following sections.

7.1 Forward Analyses Testing Stage

The first stage is the forward analyses stage in which the FEM models were run using the linear elastic or the power law plasticity material models as described above. The results were processed, and the strains and stresses were extracted, creating training datasets. ANN models were trained using those datasets, then the FEM analyses were run, but this time using the ANN models as

constitutive models. The results of the FEM models with ANN constitutive models were then processed and compared to the results of the FEM models with the linear elastic or the power law plasticity constitutive models.

For the CTX-LE model, the results of the training of the ANN, are presented in Figure 7-1, which shows very good concordance correlation coefficients between the stresses from the training datasets and the stresses calculated using the ANN. Those stresses are not results of an FEM analyses but are calculated using the ANN evaluator written in TensorFlow. The trained ANN model is used in a FEM analysis with the same boundary conditions as the original model, and yielded the results shown in Figure 7-2 to Figure 7-4. The comparison shows that stress path, axial strains, volumetric strains, deformations, and stress space match very well those calculated by the linear elastic constitutive model. To compare the stresses and strains more closely, the stresses and strains calculated by the FEM analysis employing the ANN constitutive model were plotted against stresses and strains calculated by the FEM analysis employing the linear elastic constitutive model as shown in Figure 7-5 and Figure 7-6. The comparisons again show very good matches between the stresses and strains calculated by the ANN constitutive model and those calculated by the linear elastic constitutive model.

The same process was repeated for the other 3 models. The results for the MTX-LE models are presented in Figure 7-7 to Figure 7-12. The results for the CUC-PL models are presented in Figure 7-13 to Figure 7-18. The results for the MUC-PL models are presented in Figure 7-19 to Figure 7-24. All the figures show very good matches between the global and constitutive behavior of the FEM analyses employing linear elastic or power law plasticity constitutive models and the FEM analyses employing the ANN constitutive model.

Those plots show that if an NN is trained with the correct strain and stress datasets, the training module was able to train ANN well which is reflected by the excellent concordance correlation coefficients displayed on the ANN computed stresses vs the training dataset stresses. The figures also showed that the ANN constitutive model worked well in LS-DYNA as it yielded a very similar response compared with the original models. The pre-post processor also is shown to be functional, creating the training datasets, postprocessing the results and generating the comparison plots.

7.2 SelfSim Analyses Testing Stage

The second stage of testing included running SelfSim analyses. The loads and deformations calculated from the 4 models using the linear elastic or the power law plasticity constitutive

models, were used as boundary conditions for the 2 models of SelfSim. In the first model, the load boundary conditions were applied while in the second model the displacement boundary conditions were applied. In both FEM models, an ANN constitutive model was used. This model was pretrained using an auto-generated linear elastic dataset. The Young's modulus of the linear elastic training datasets was selected to be different from the actual moduli, typically being around 2 times more than the actual. A different Poisson's ratio was used as well. The maximum strain used in the linear elastic datasets was typically 10 to 100 times less than the maximum strain calculated in the actual model. Those restrictions on the linear elastic training datasets are to make sure that the behavior was learned from SelfSim rather than being provided for the ANN. The SelfSim algorithm being applied to this set of problems is illustrated in Figure 7-25.

The results for the CTX-LE models are presented in Figure 7-26 to Figure 7-30. The results for the MTX-LE models are presented in Figure 7-32 to Figure 7-36. The results for the CUC-PL models are presented in Figure 7-37 to Figure 7-41. The results for the MUC-PL models are presented in Figure 7-42 to Figure 7-46. All figure sets show very good matches of the stress path, axial strain, volumetric strain, and deformations between the FEM analyses employing linear elastic or power law plasticity constitutive models and the FEM analyses employing the ANN constitutive model trained through SelfSim.

It can be noticed that the stresses calculated by the ANN (Figure 7-30) are not identical to the stresses calculated by the linear elastic model, even though those the overall behavior is very close as shown in Figure 7-26 to Figure 7-29. The other models show also similar differences. This is suggesting that more than one set of stresses can satisfy the problem's conditions. This is probably due to the symmetry of the models. The obtained stress is dependent on the initialization of the ANN and the parameters of the autogenerated linear elastic set. To demonstrate this, another SelfSim analysis for the CTX-LE case was run, in which the autogenerated linear elastic dataset was kept in the training dataset during the SelfSim analysis; the overall behavior (figures not shown) was also very close to the target behavior, however, the stress distribution was closer to the stresses calculated by the linear elastic model as shown in Figure 7-31.

7.2.1 The Extracted Stiffness Matrix

The theoretical linear elastic stiffness matrix stiffness matrix for the linear elastic material used in the model CTX-LE ($E = 20$ MPa, $\upsilon = 0.3$) is first compared to the stiffness matrix of an ANN model trained by an autogenerated linear elastic training dataset that is generated by providing it

with the actual elastic moduli (E = 20 MPa, υ = 0.3). This check is necessary to verify that the stiffness matrix calculations in LS-DYNA are correct. The two matrices are extremely close:

- Linear elastic stiffness matrix:

$$M = \begin{bmatrix} 26.92 & 11.54 & 11.54 & 0 & 0 & 0 \\ 11.54 & 26.92 & 11.54 & 0 & 0 & 0 \\ 11.54 & 11.54 & 26.92 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15.38 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15.38 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15.38 \end{bmatrix}$$

- ANN stiffness matrix trained with auto-generated linear elastic dataset:

$$M = \begin{bmatrix} 27.04 & 11.56 & 11.55 & -0.009 & -0.005 & -0.005 \\ 11.56 & 27.04 & 11.55 & -0.005 & -0.008 & -0.002 \\ 11.55 & 11.55 & 27.04 & -0.004 & -0.01 & -0.004 \\ -0.002 & -0.003 & -0.005 & 15.41 & -0.003 & 0 \\ -0.002 & -0.004 & -0.004 & 0.002 & 15.42 & 0.007 \\ 0.003 & 0.002 & 0.001 & 0.004 & -0.005 & 15.42 \end{bmatrix}$$

The stiffness matrix of the ANN model which was trained by the stresses and strains extracted directly from the CTX-LE model is close but not identical to the theoretical stiffness matrix as shown below:

- ANN stiffness matrix trained by the stresses and strains dataset extracted from the CTX-LE model (not SelfSim).

$$M = \begin{bmatrix} 27.33 & 10.55 & 10.88 & 0.4 & 0.006 & 0.06 \\ 10.96 & 27.6 & 12.13 & -0.29 & -0.02 & 0.01 \\ 11.51 & 11.47 & 26.74 & -0.1 & -0.006 & -0.03 \\ 0.04 & -0.05 & -0.01 & 15.38 & -0.005 & 0.04 \\ 0.03 & 0.002 & -0.05 & -0.08 & 14.82 & -0.07 \\ 0.05 & -0.12 & 0.12 & -0.03 & -0.02 & 14.77 \end{bmatrix}$$

Those differences however, didn't prevent the trained ANN from capturing the global and constitutive behavior as was presented in Figure 7-1 to Figure 7-6.

For the stiffness matrix of the ANN model trained through the SelfSim analysis done for the CTX-LE model, significant differences can be observed when compared to the theoretical stiffness matrix:

- ANN stiffness matrix trained through SelfSim run for the CTX-LE model

$$M = \begin{bmatrix} 24.23 & -0.56 & 8.36 & -0.54 & -0.38 & -0.23 \\ -1.24 & 23.7 & 7.92 & -0.39 & 0.11 & -0.02 \\ 6.92 & 7.11 & 25.01 & 0.32 & 0.11 & -0.03 \\ 0.76 & 0.46 & 0.82 & 27.68 & -0.49 & -0.58 \\ -0.76 & 0.16 & 1.18 & 0.08 & 10.25 & -0.03 \\ 0.33 & 0.1 & 1.8 & 0.18 & 0.03 & 9.88 \end{bmatrix}$$

This is a result of the dataset with which the ANN was trained as it was extracted through SelfSim from a model that is not as rich in its stress and strain spaces as an autogenerated linear elastic dataset. The dataset extracted through SelfSim is affected by the symmetry of the model and the consequent collinearity of x and y stresses and strains. This however, didn't prevent this ANN from properly capturing the global behavior as was shown in Figure 7-26 to Figure 7-30.

7.2.2 Generalization Capability of the Trained ANN Models:

The generalization capability of the ANN models trained through SelfSim was checked by running the MTX FEM model using the ANN model trained through SelfSim conducted for the CTX-LE model and running the CTX FEM model using the ANN model trained through SelfSim conducted for the CTX-LE model. The MUC FEM model was also run using the ANN model trained through SelfSim conducted for the CUC-PL model, and the CUC FEM model was run using the ANN model trained through SelfSim conducted for the MUC-PL model.

The results for those comparisons are shown in Figure 7-47 to Figure 7-58. The figures show an acceptable generalization capability for the ANN models trained through SelfSim to learn the linear elastic behavior. The generalization capability for the ANN models trained through SelfSim to learn the power law plasticity behavior is poorer than that of the linear elastic behavior.

7.3 Conclusions

In this chapter, the results of the testing of the new code were demonstrated. In the forward analysis stage of testing, the performance of the code's components was verified and demonstrated through a set of forward analyses without employing the SelfSim algorithm. In the second stage of testing

the SelfSim algorithm was employed in a set of inverse analyses. Both stages of testing show that the code is capable of training ANN models that can approach the behavior of the theoretical models used in those tests. In addition, a brief study of the generalization capabilities of the trained ANN models was conducted. With this verification, the code is ready to be used with real problems. This is demonstrated in 0, through a set of conventional and modified triaxial tests performed on uniform sand.

7.4 Table

Table 7-1 Summary of analyses performed for testing the new code

Analysis Type	Analysis Name	Data Type	Data Source	Model Description
Forward	LE-CTX-F	Strains and Stresses	FEM Model using Linear Elastic model (synthetic)	2 to 1 (H/D), no clamps, drained, Triaxial Compression test
	LE-MTX-F	Strains and Stresses	FEM Model using Linear Elastic model (synthetic)	2 to 1 (H/D), with clamps, drained, Modified Triaxial Compression test
	PL-UC-F	Strains and Stresses	FEM Model using Power Law Plasticity model (synthetic)	2 to 1 (H/D), no clamps, drained, Unconfined Compression test
	PL-UC-F	Strains and Stresses	FEM Model using Power Law Plasticity model (synthetic)	2 to 1 (H/D), with clamps, drained, Modified Unconfined Compression test
SelfSim	LE-CTX-SS	Boundary Loads and Displacements	FEM Model using Linear Elastic model (synthetic)	2 to 1 (H/D), no clamps, drained, Triaxial Compression test
	LE-MTX-SS	Boundary Loads and Displacements	FEM Model using Linear Elastic model (synthetic)	2 to 1 (H/D), with clamps, drained, Modified Triaxial Compression test
	PL-UC-SS	Boundary Loads and Displacements	FEM Model using Power Law Plasticity model (synthetic)	2 to 1 (H/D), no clamps, drained, Unconfined Compression test
	PL-UC-SS	Boundary Loads and Displacements	FEM Model using Power Law Plasticity model (synthetic)	2 to 1 (H/D), with clamps, drained, Modified Unconfined Compression test

7.5 Figures

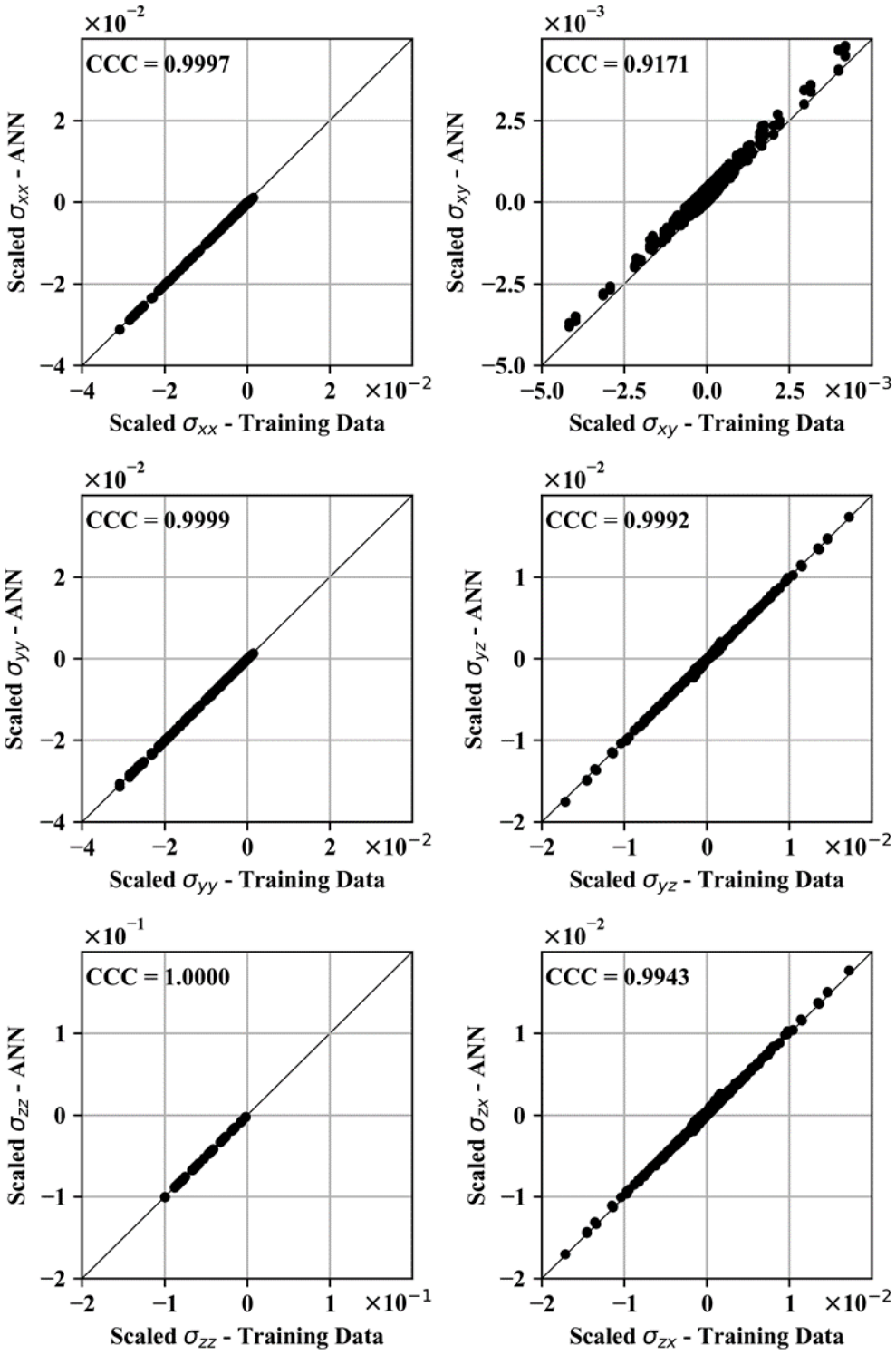


Figure 7-1: Comparison of the stresses computed by the ANN to the stresses in the training datasets extracted from the results of the CTX-LE model

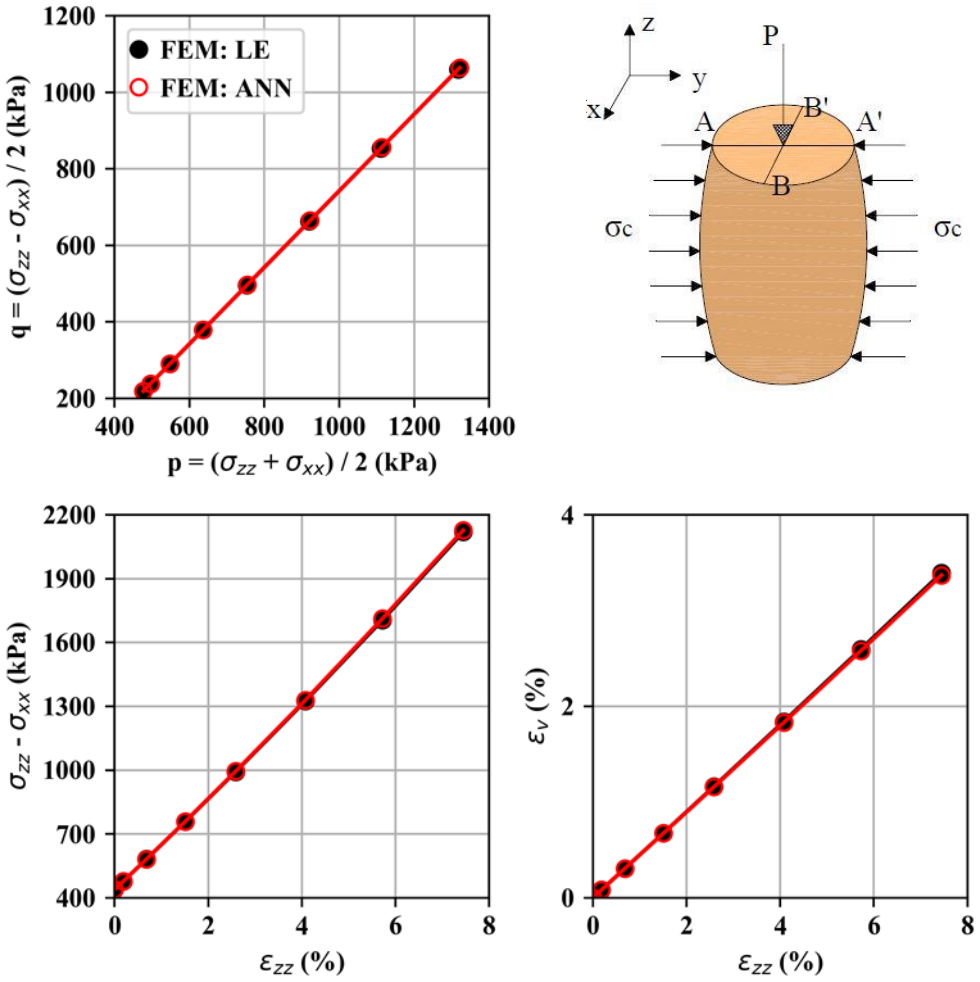


Figure 7-2: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

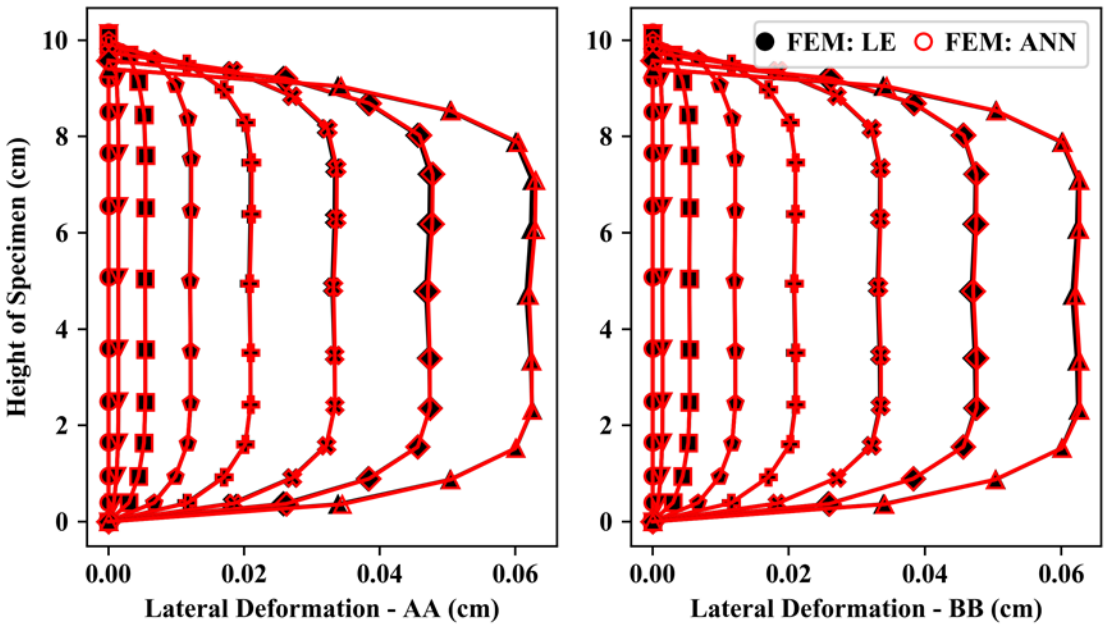


Figure 7-3: Comparison of the deformations calculated by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

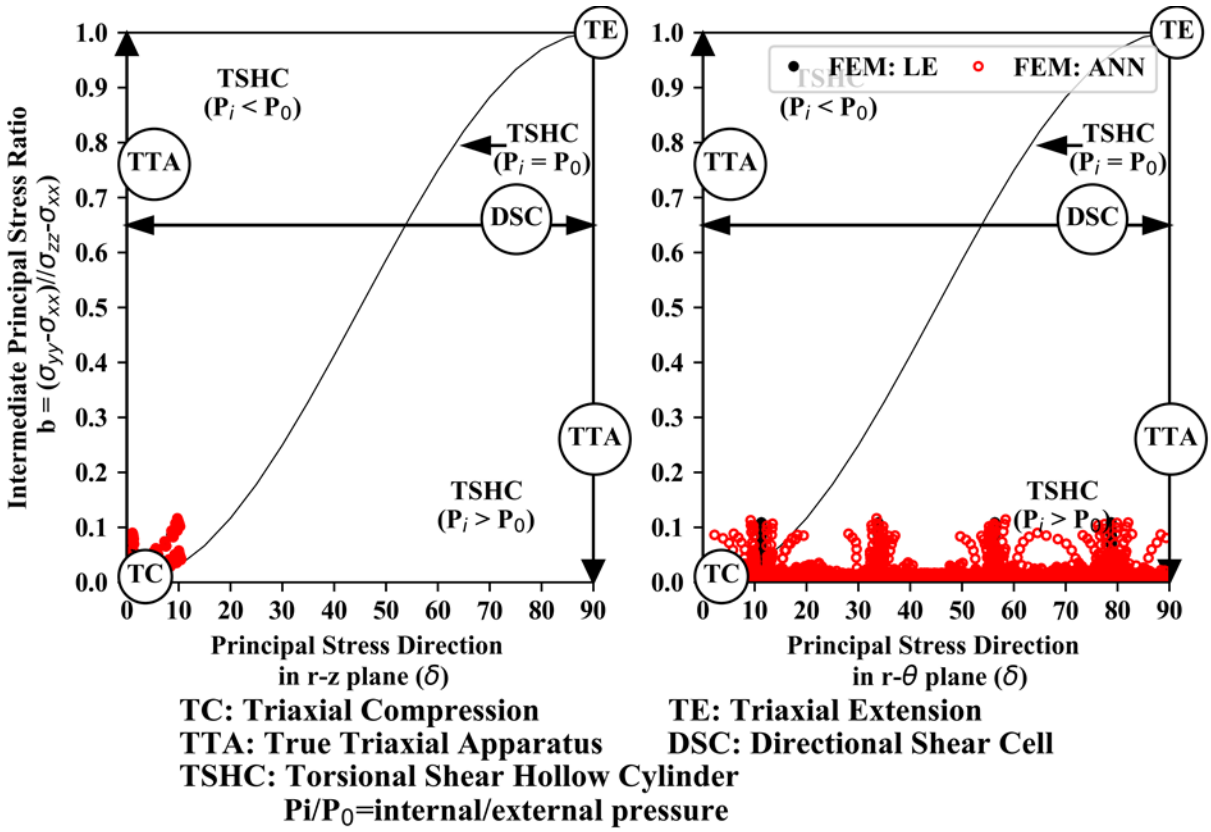


Figure 7-4: Comparison of the stress space calculated by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

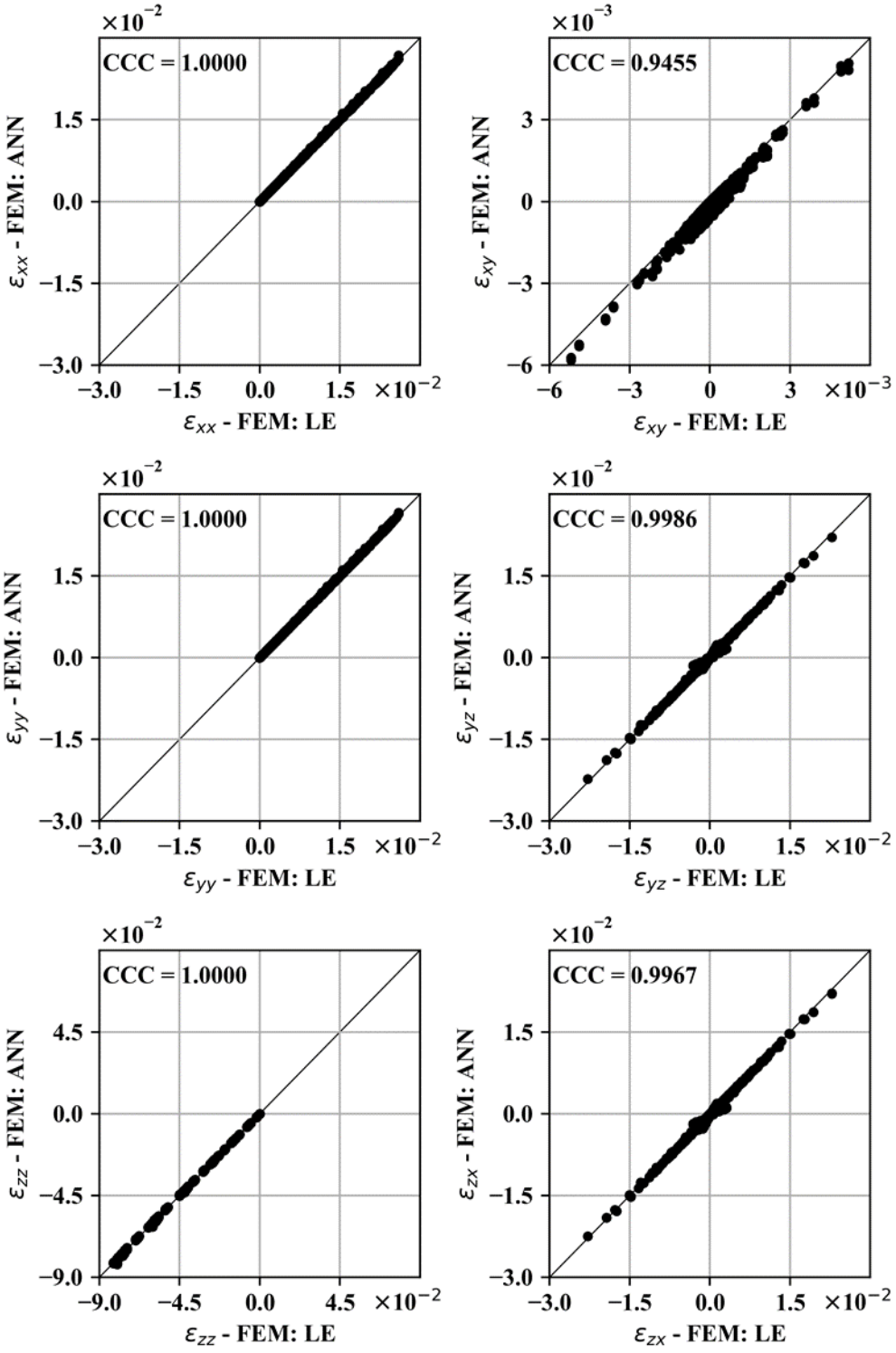


Figure 7-5: Comparison of the strains calculated the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

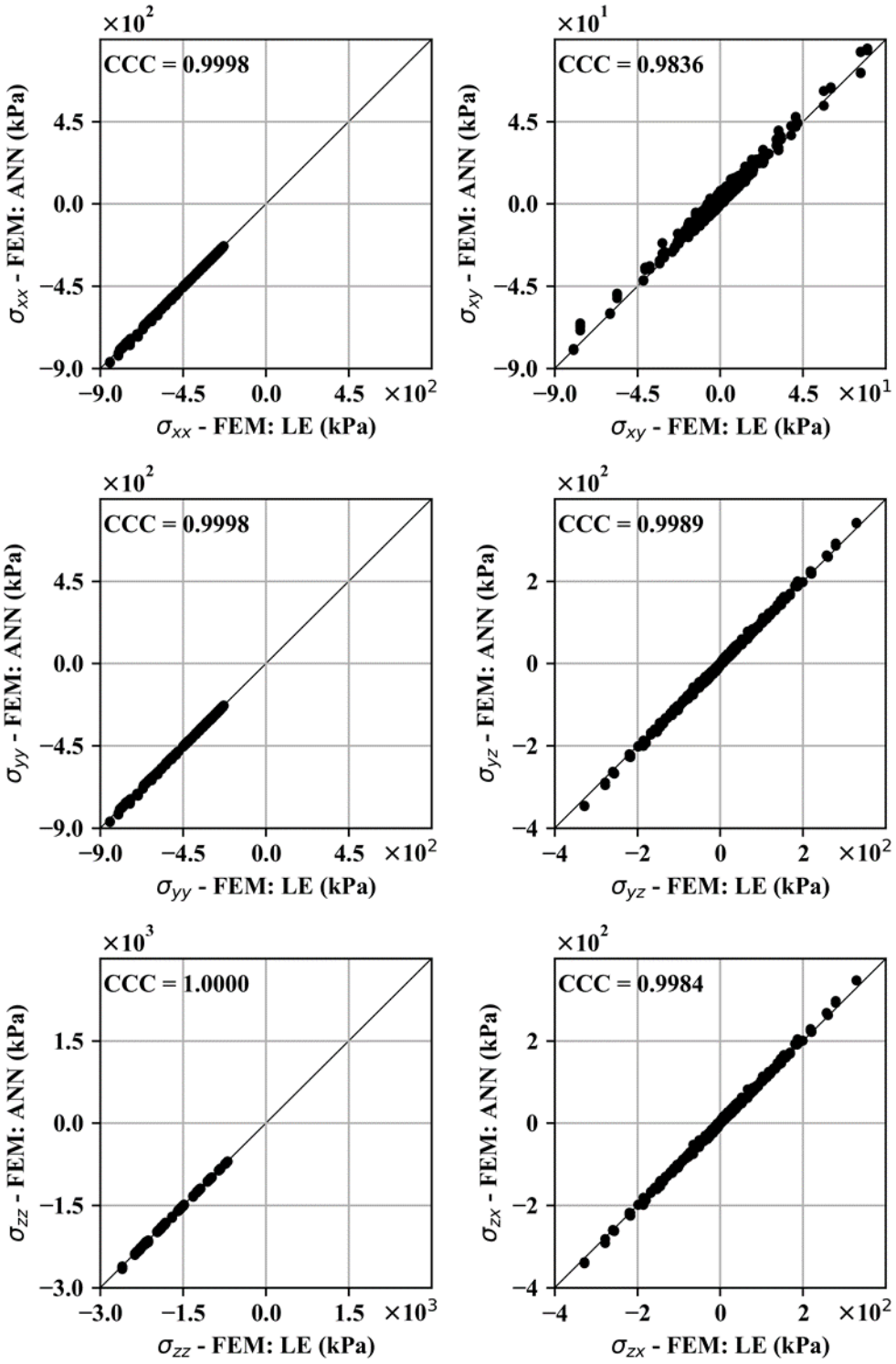


Figure 7-6: Comparison of the stresses calculated the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

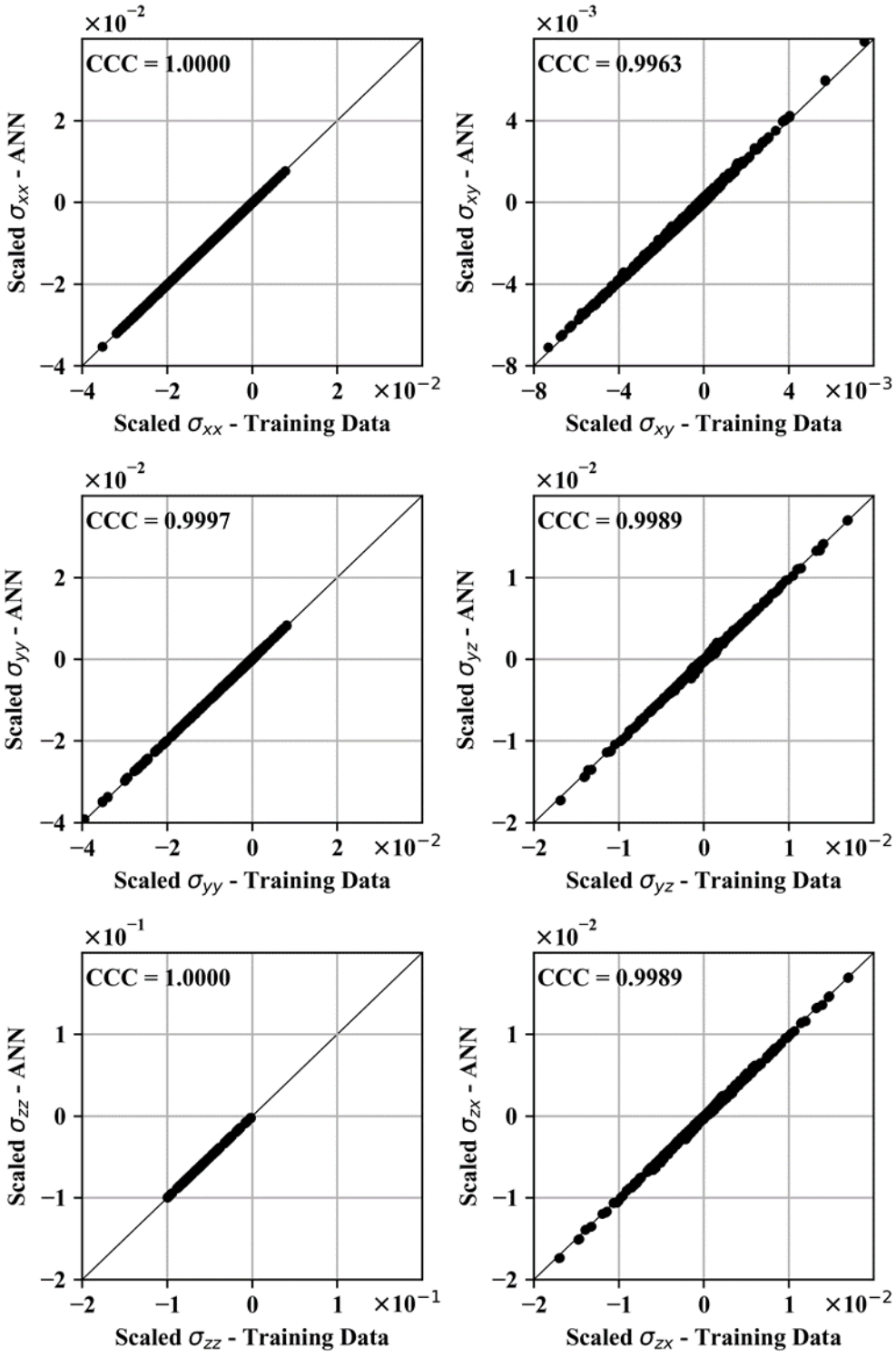


Figure 7-7: Comparison of the stresses computed by the ANN to the stresses in the training datasets extracted from the results of the MTX-LE model.

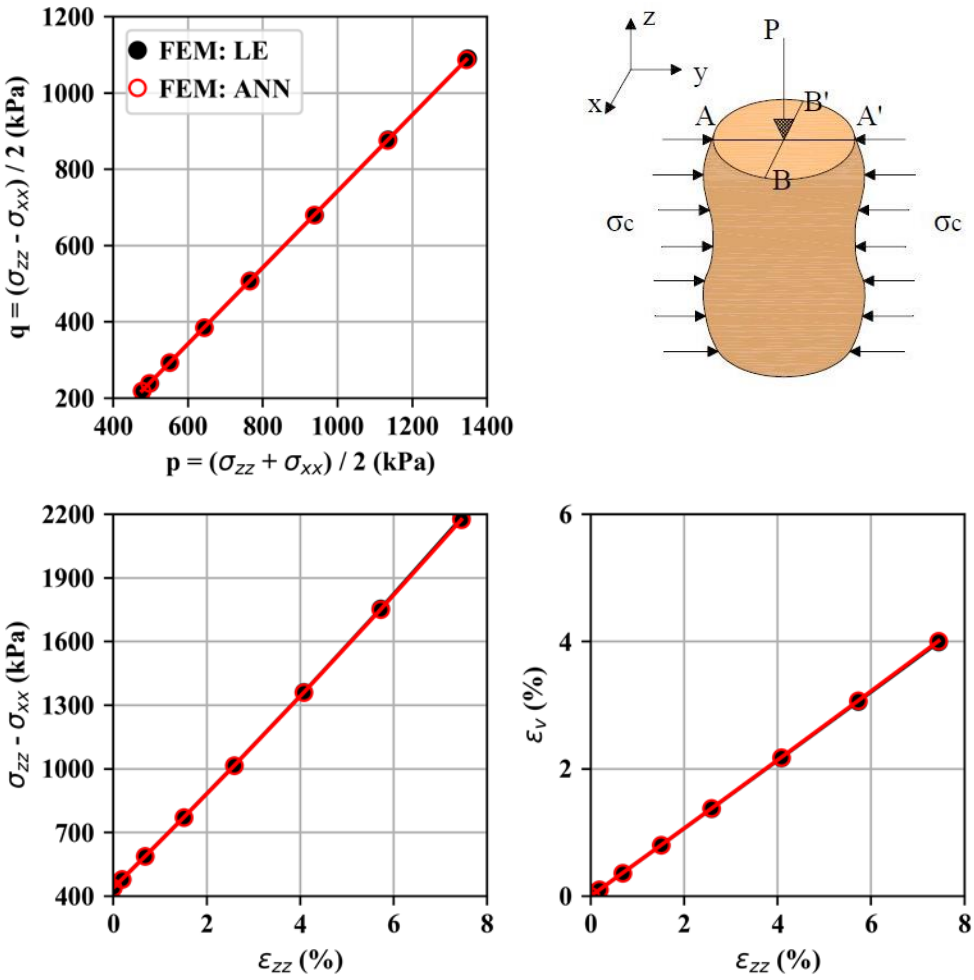


Figure 7-8: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

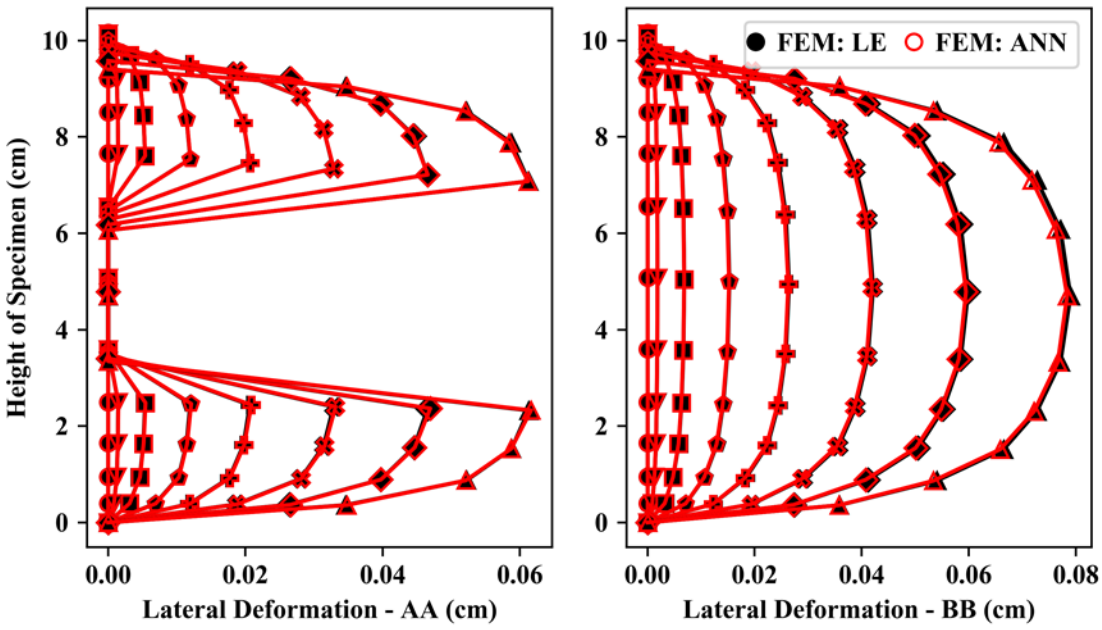


Figure 7-9: Comparison of the deformations calculated by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

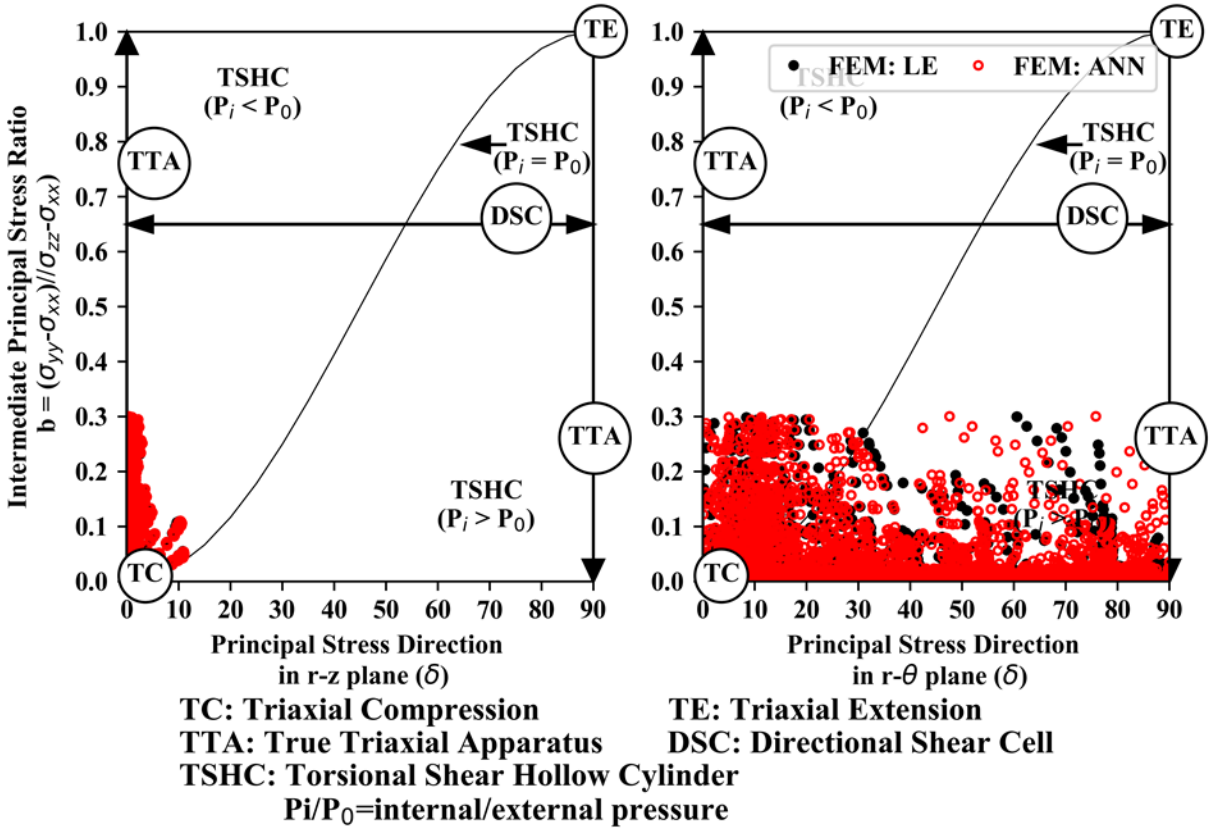


Figure 7-10: Comparison of the stress space calculated by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

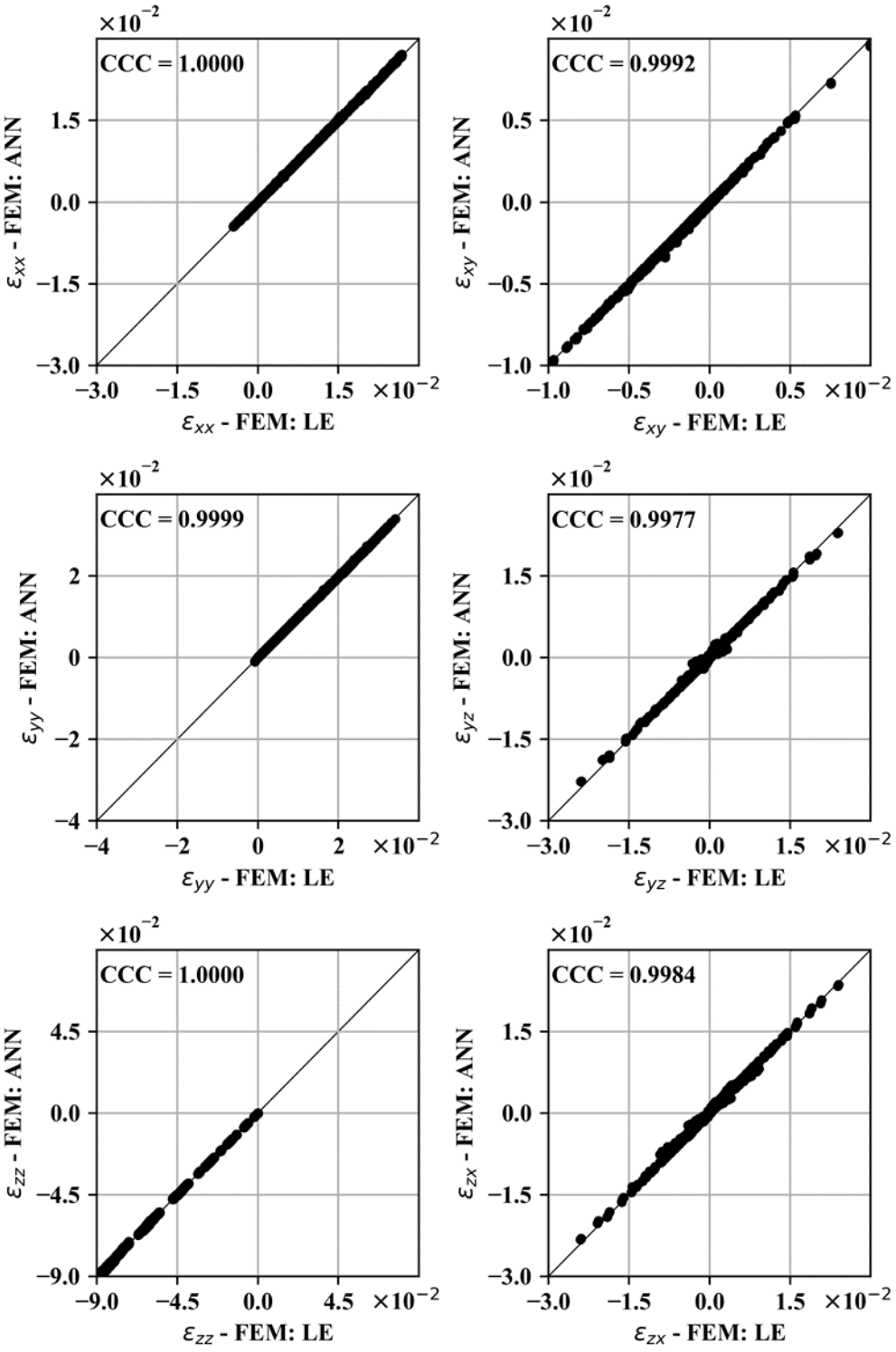


Figure 7-11: Comparison of the strains calculated the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

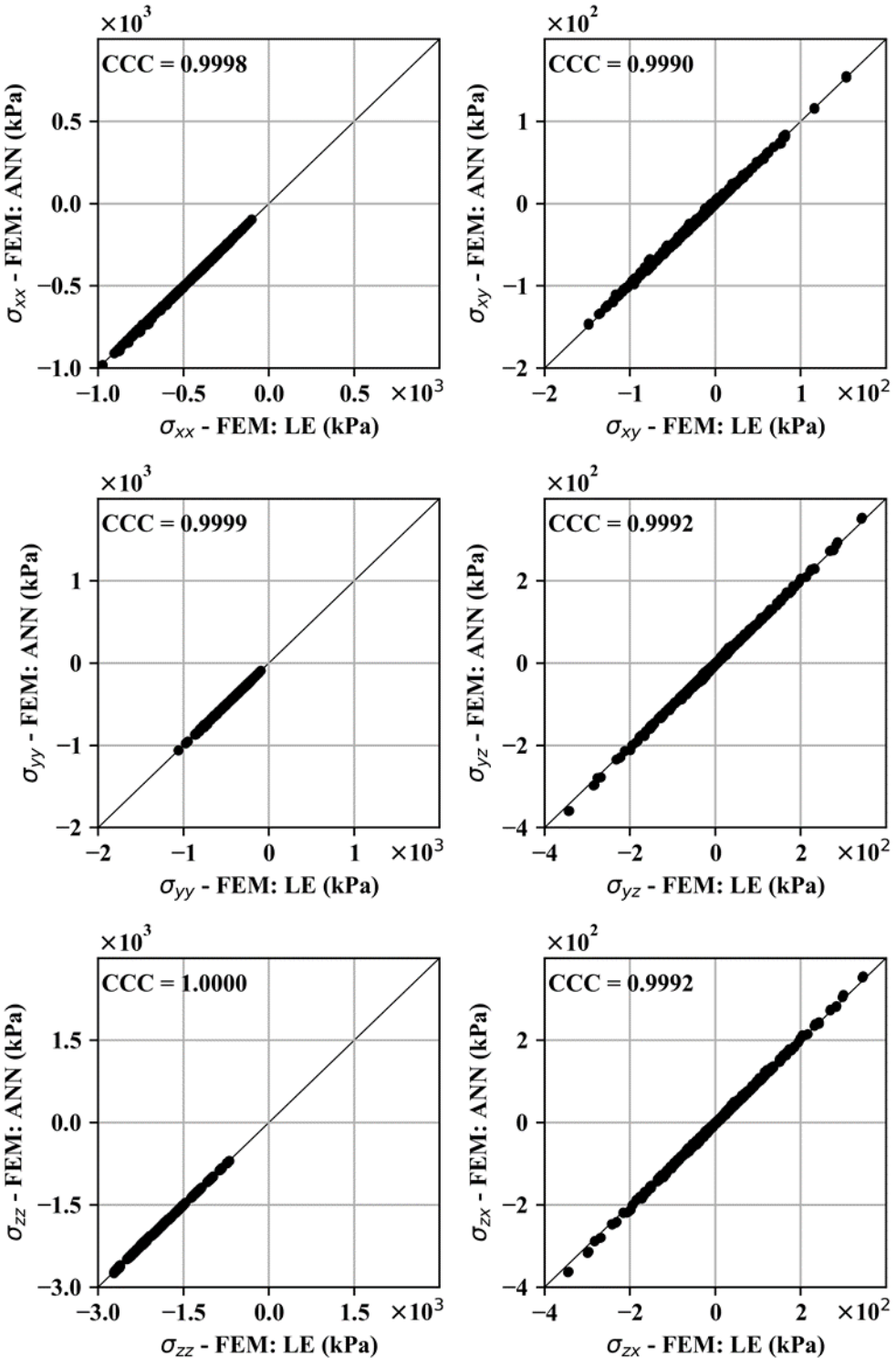


Figure 7-12: Comparison of the stresses calculated the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model.

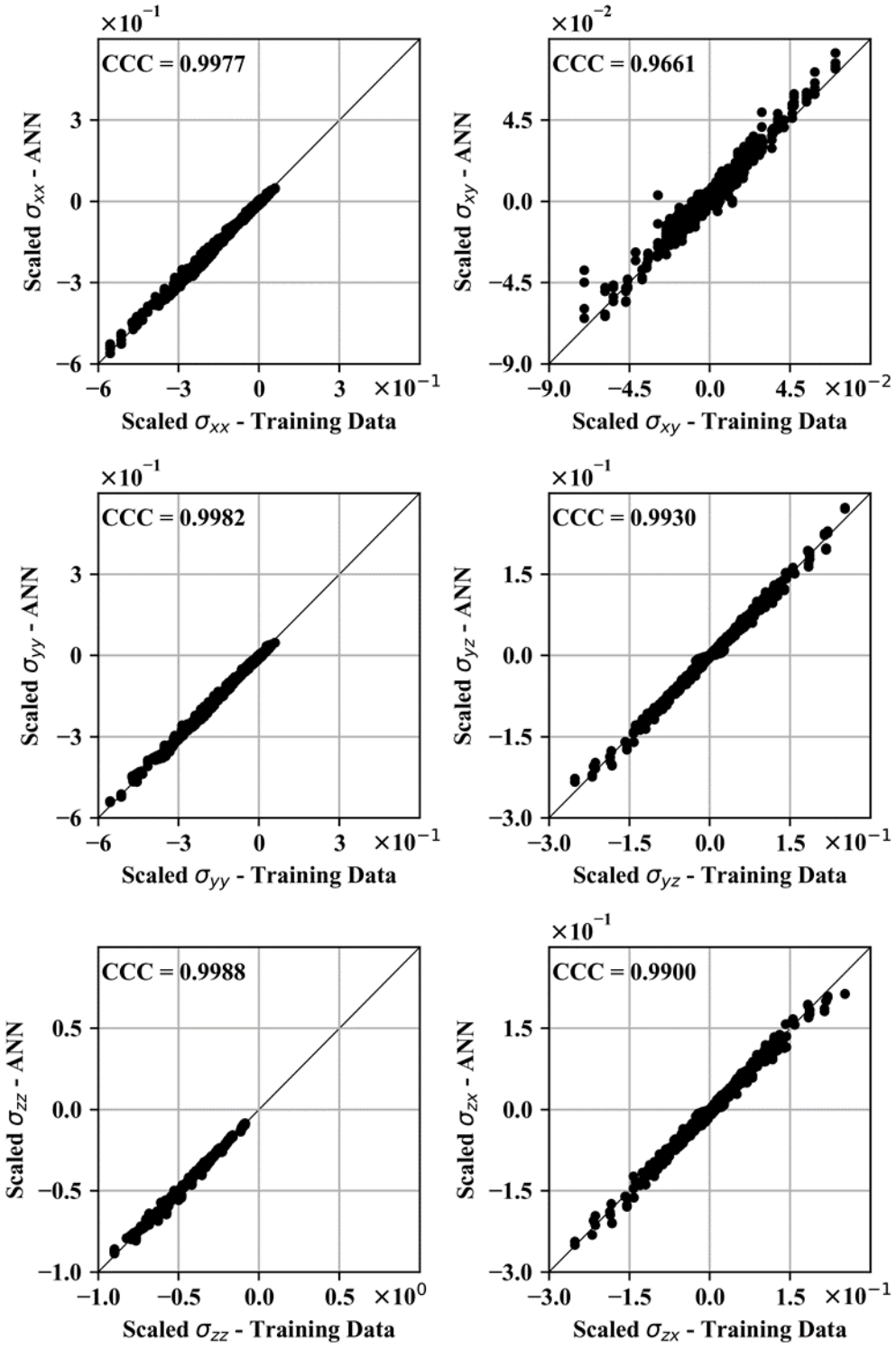


Figure 7-13: Comparison of the stresses computed by the ANN to the stresses in the training datasets extracted from the results of the CUC-PL model.

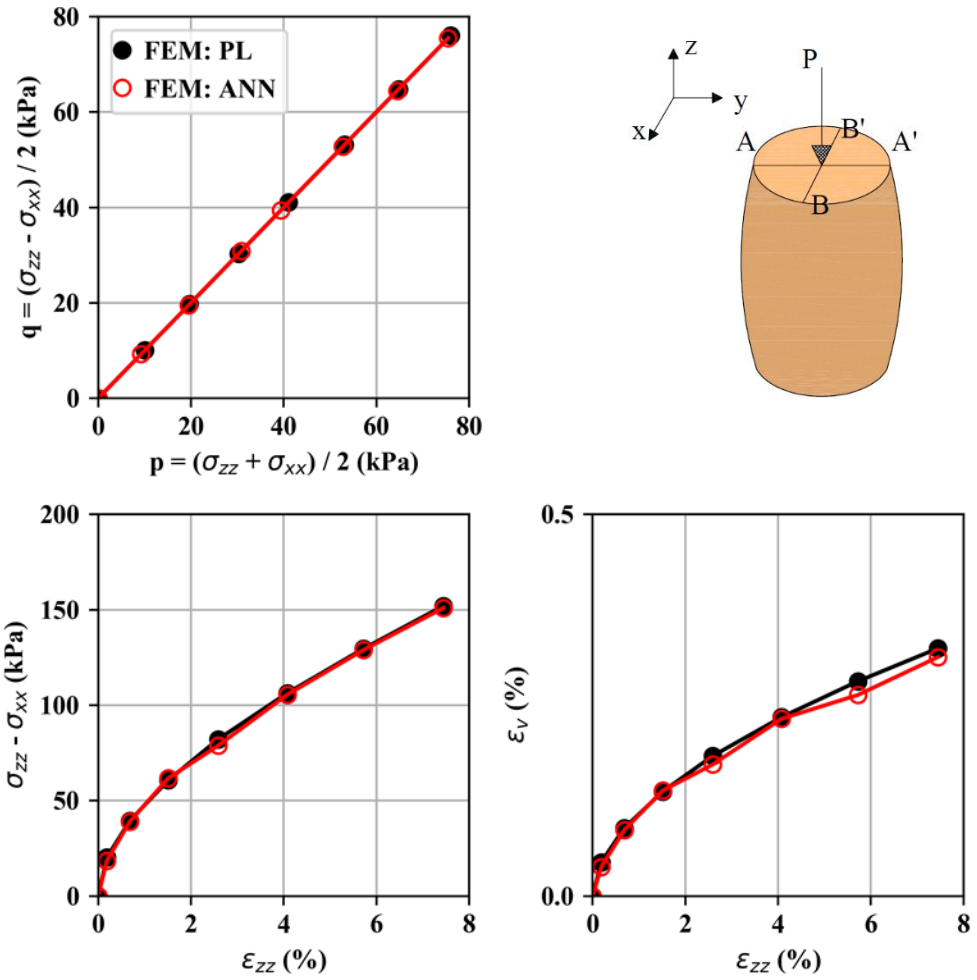


Figure 7-14: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

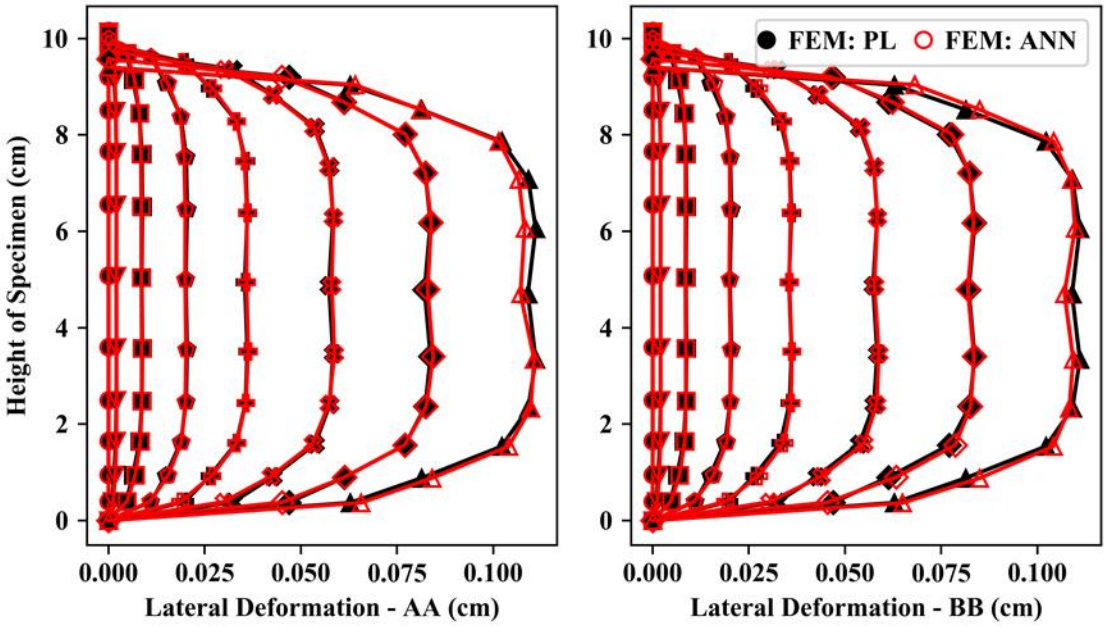


Figure 7-15: Comparison of the deformations calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

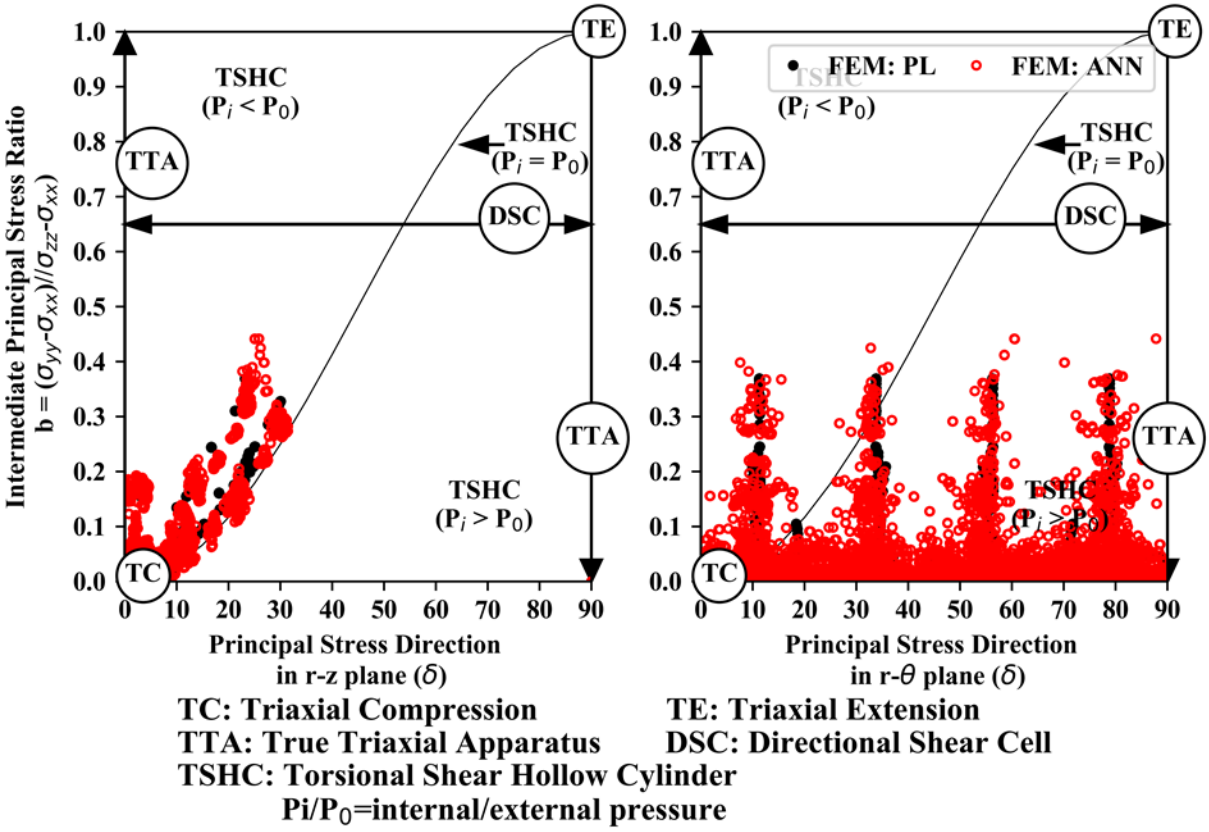


Figure 7-16: Comparison of the stress space calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

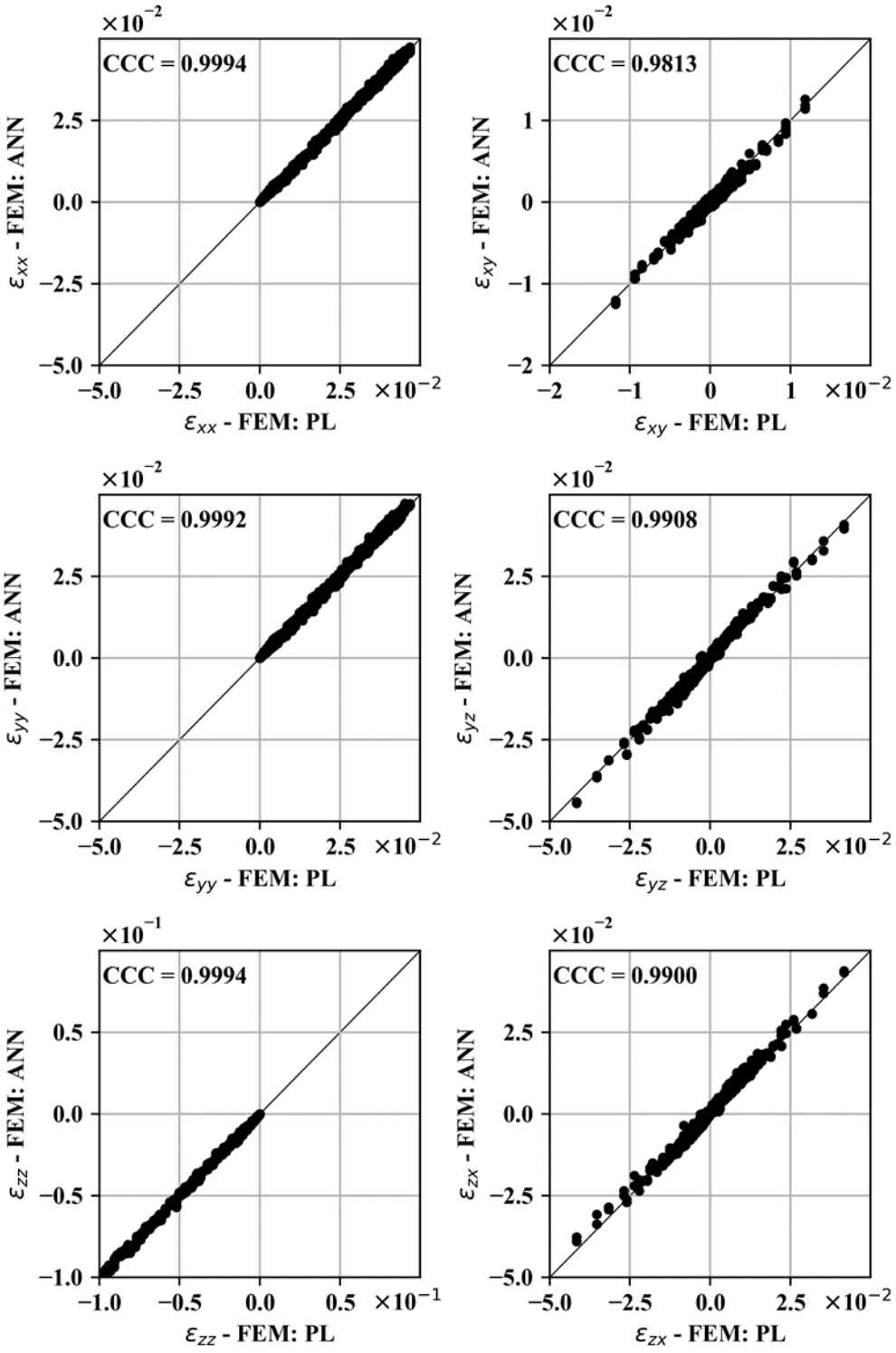


Figure 7-17: Comparison of the strains calculated the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

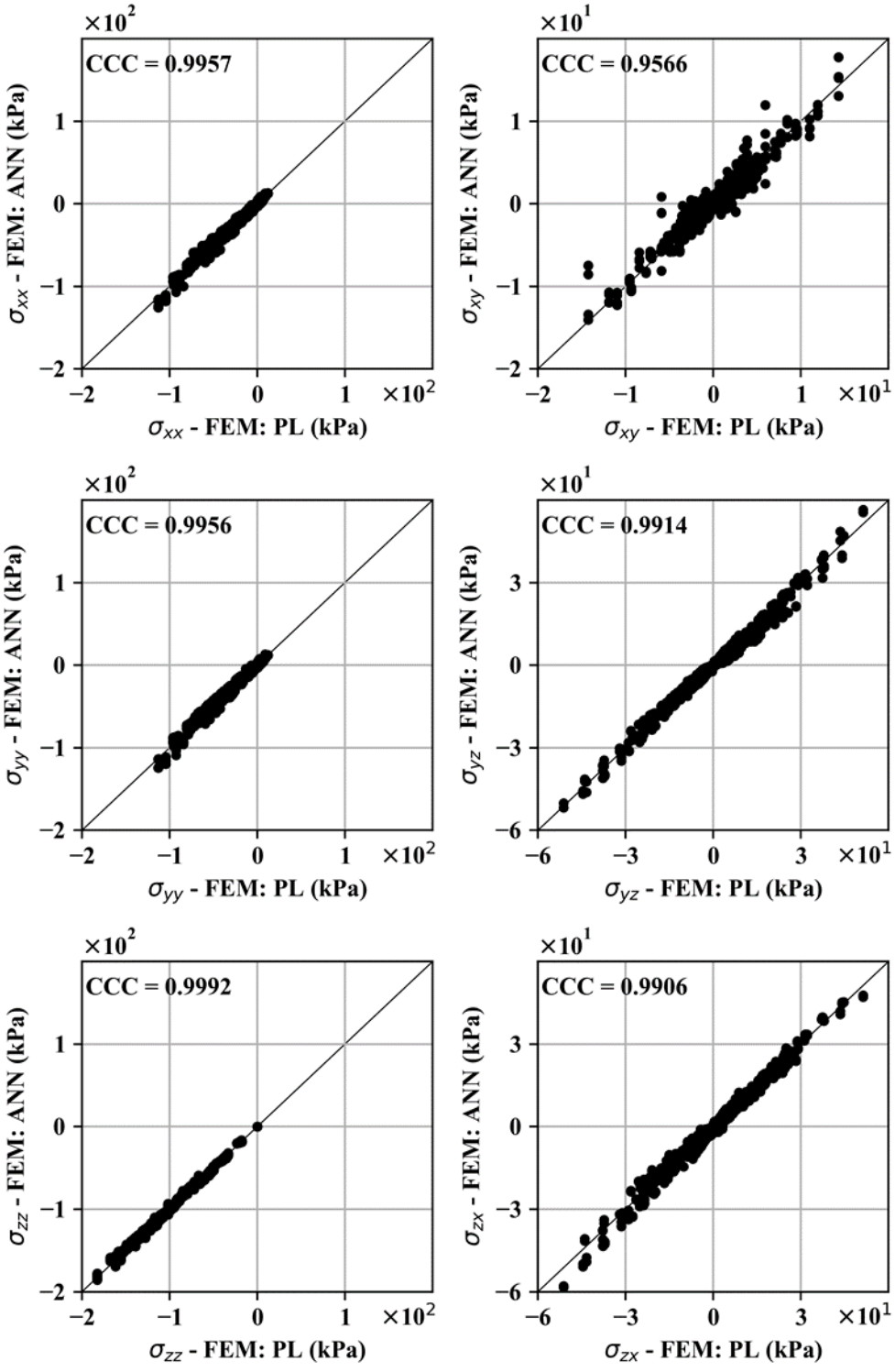


Figure 7-18: Comparison of the stresses calculated the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

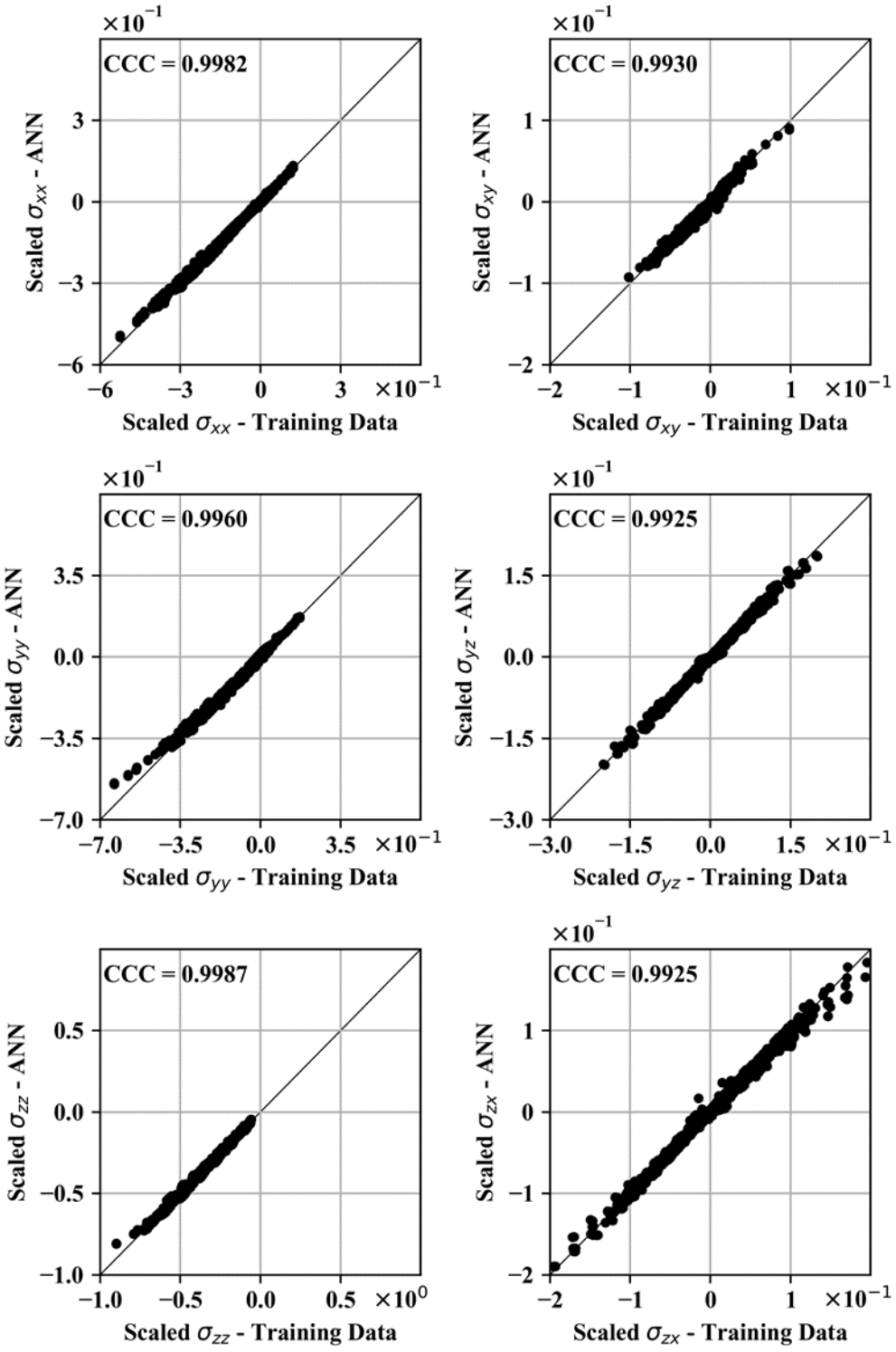


Figure 7-19: Comparison of the stresses computed by the ANN to the stresses in the training datasets extracted from the results of the MUC-PL model.

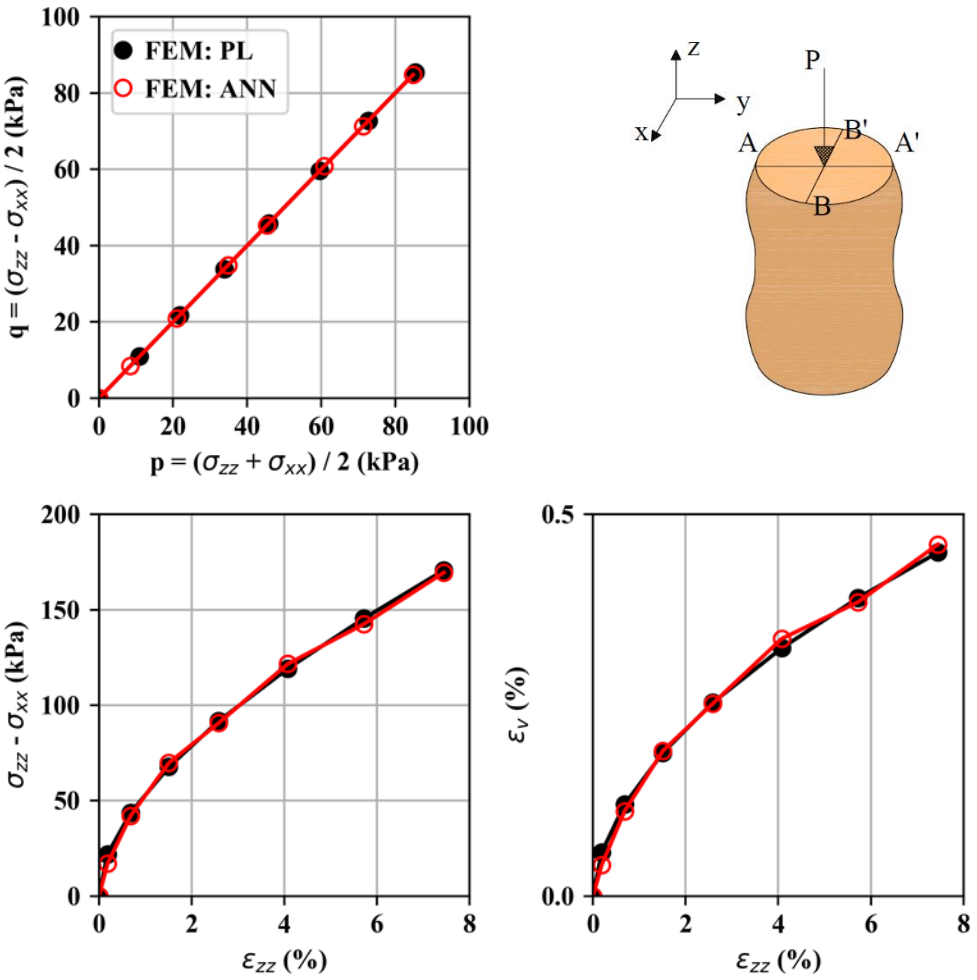


Figure 7-20: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

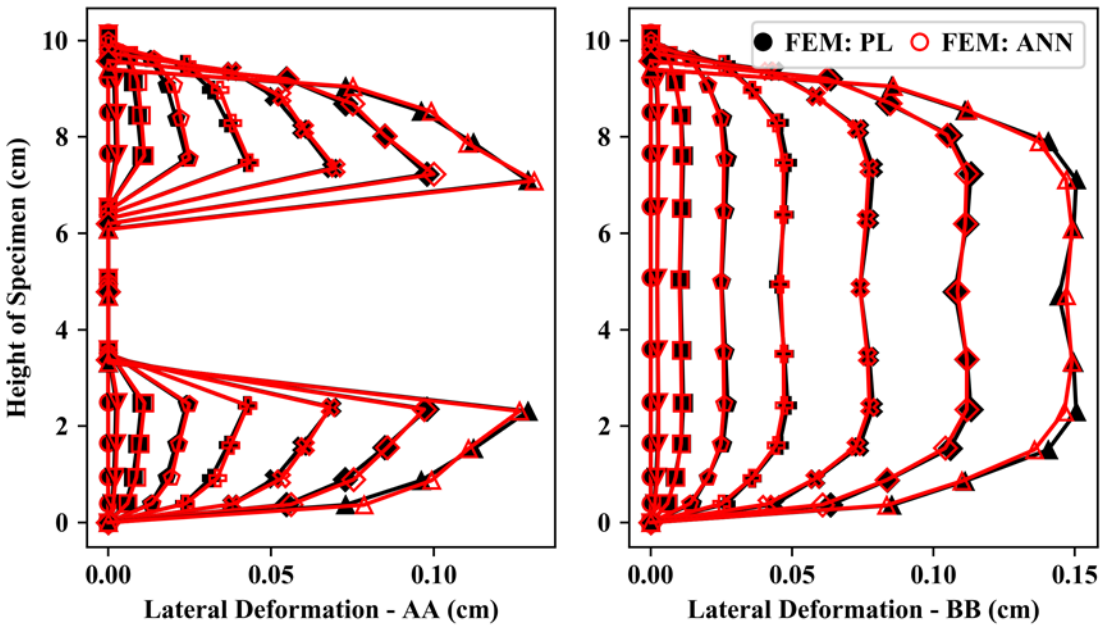


Figure 7-21: Comparison of the deformations calculated by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

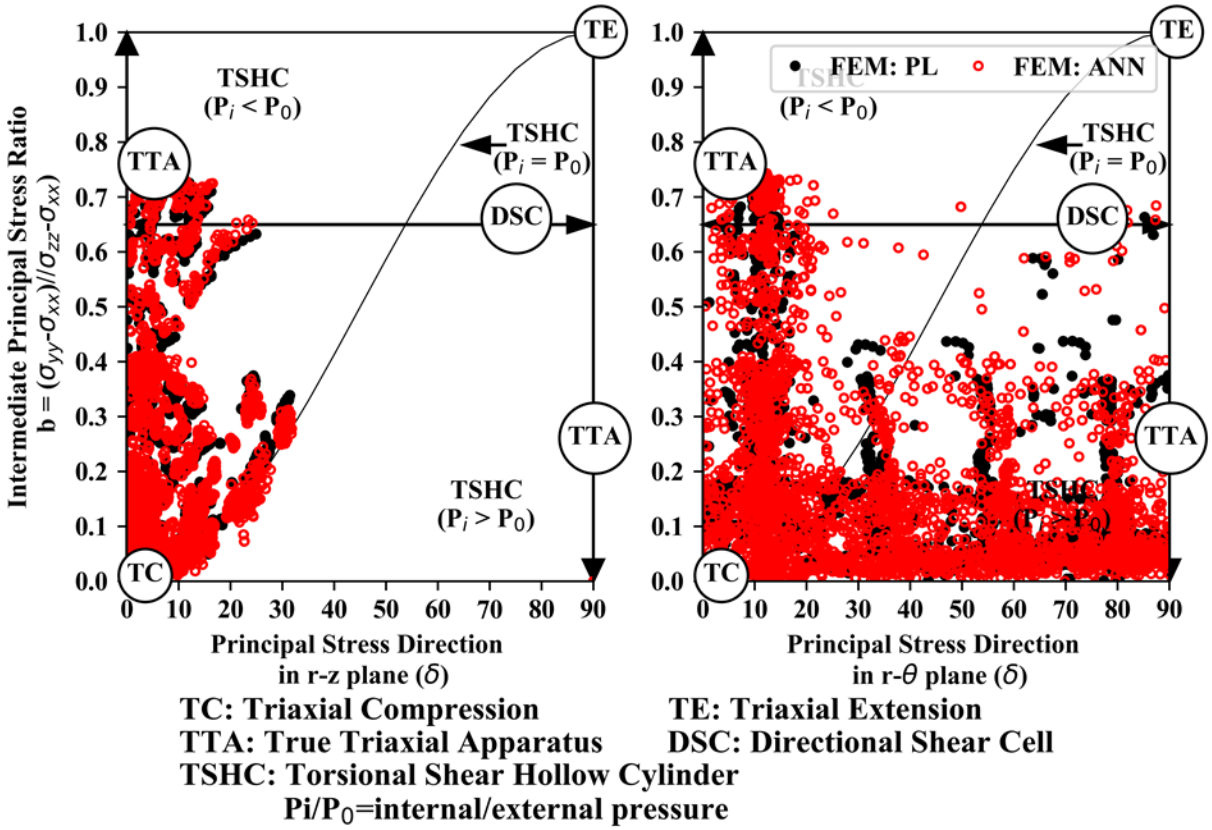


Figure 7-22: Comparison of the stress space calculated by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

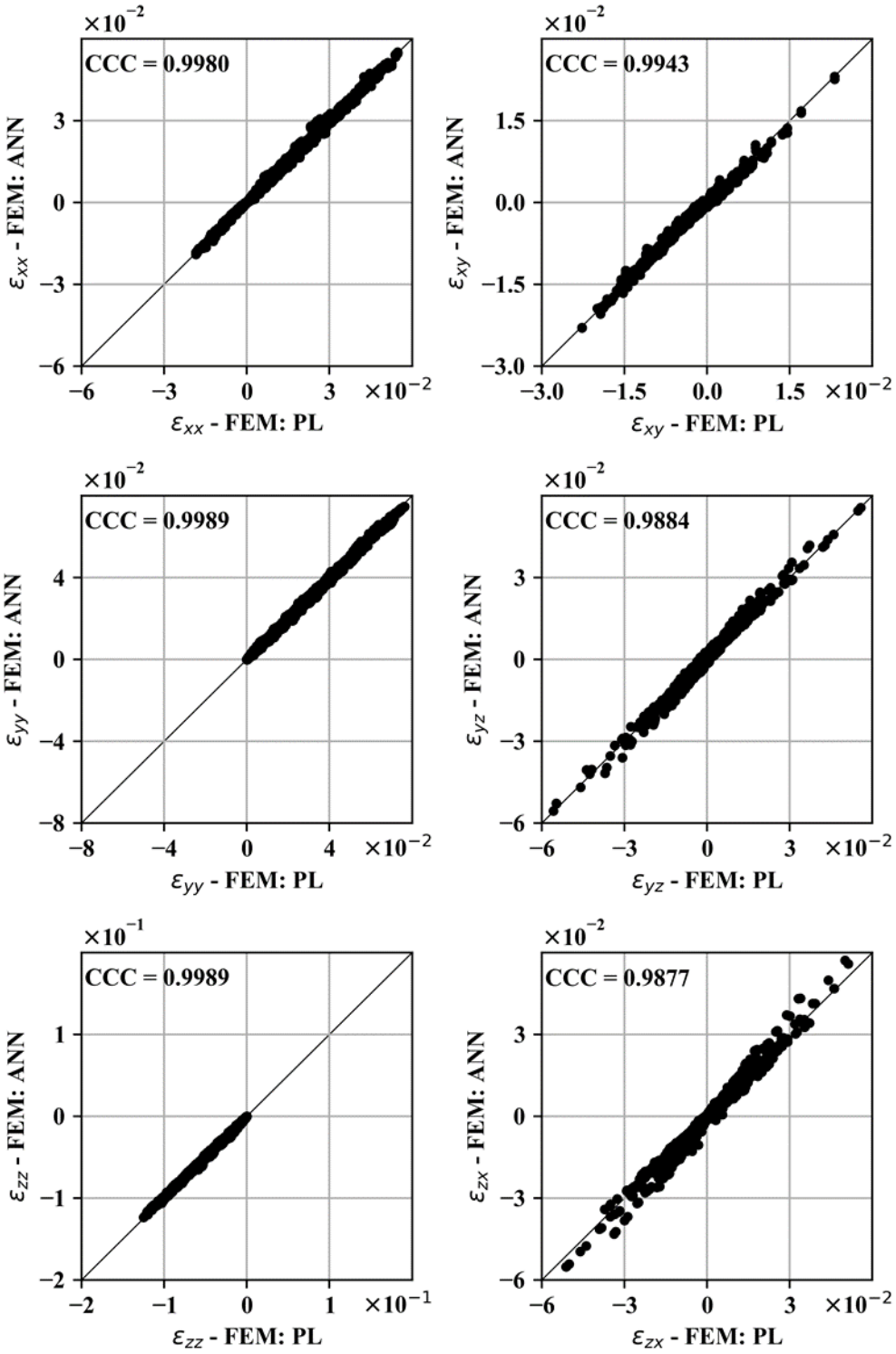


Figure 7-23: Comparison of the strains calculated the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

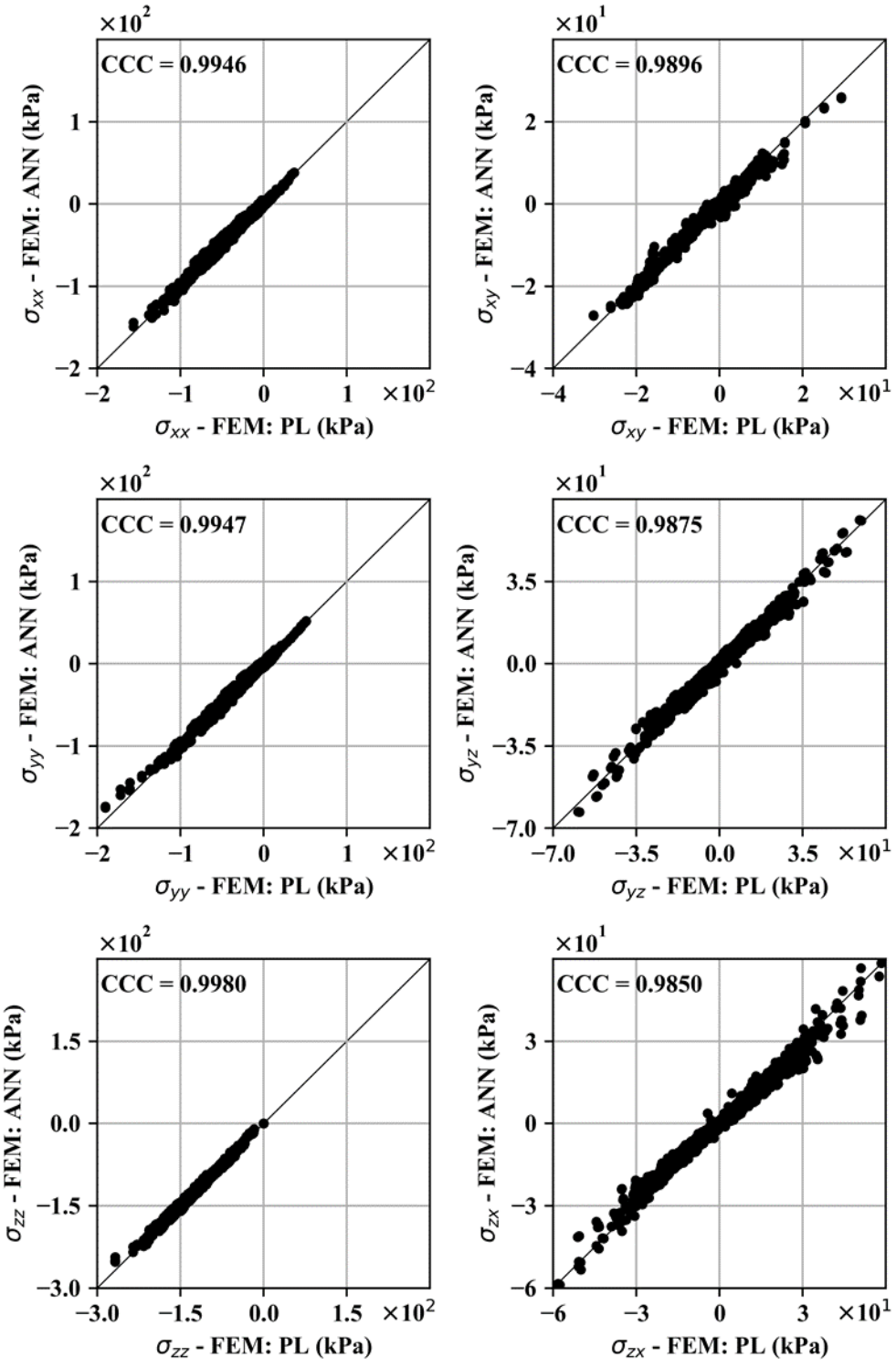


Figure 7-24: Comparison of the stresses calculated the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model.

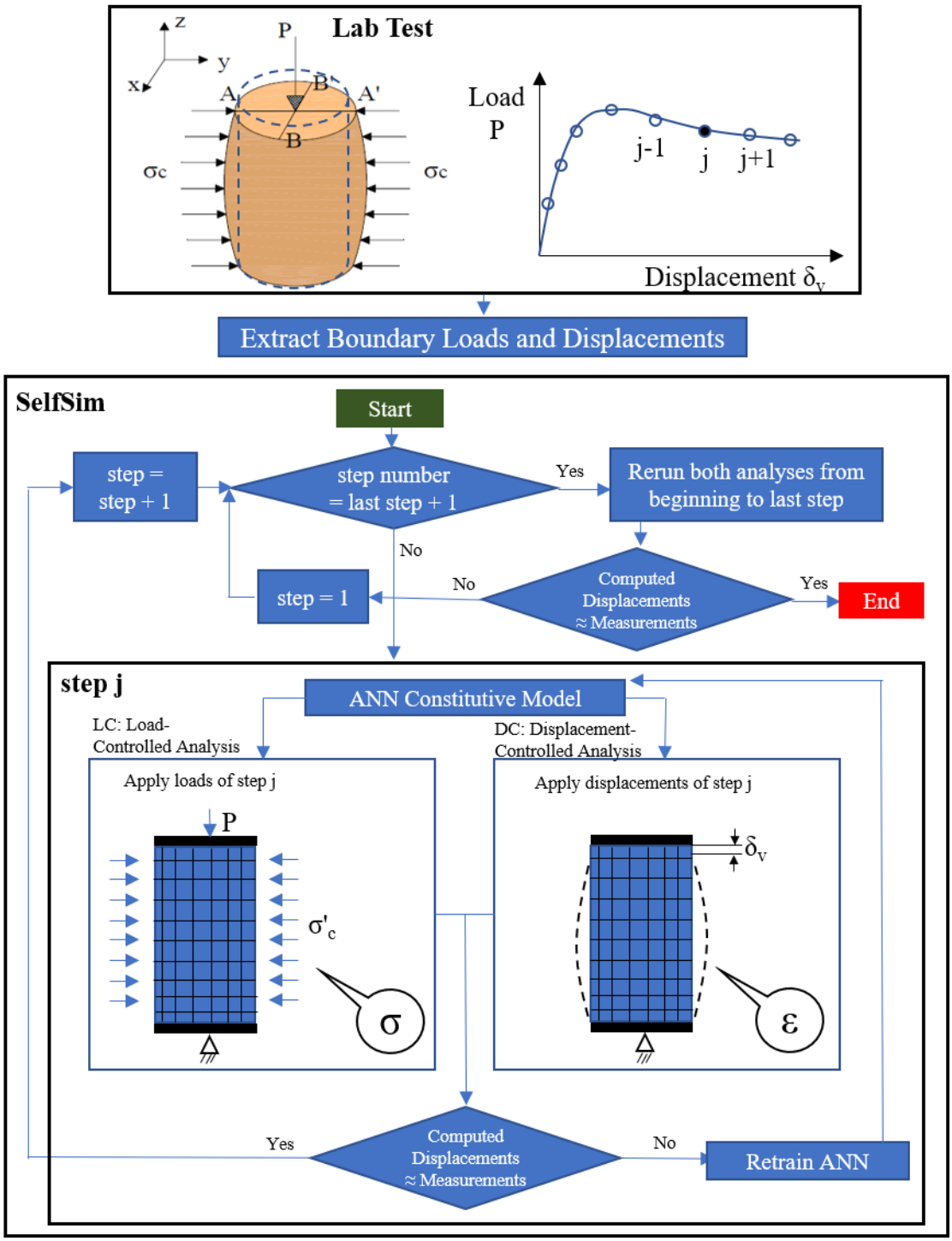


Figure 7-25: Schematic showing the steps of a SelfSim analysis performed for a lab test simulation problem. Source: modified from (Hashash and Asmar 2018)

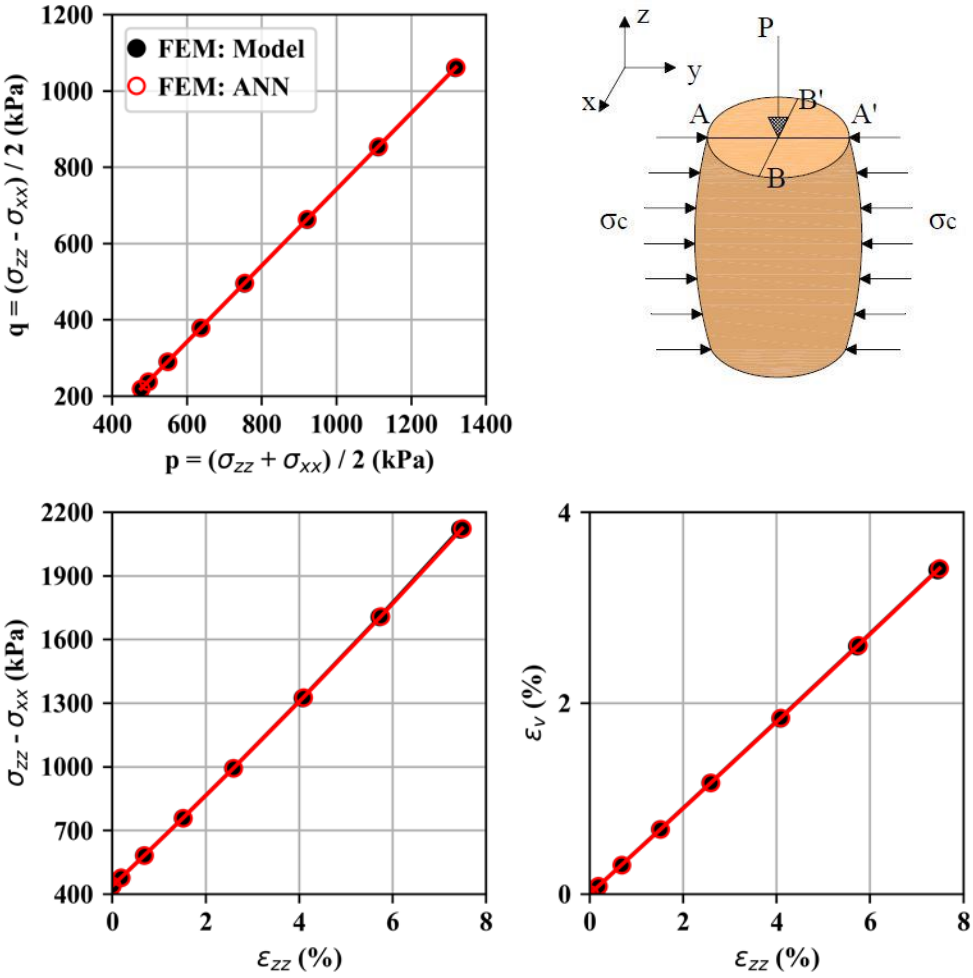


Figure 7-26: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

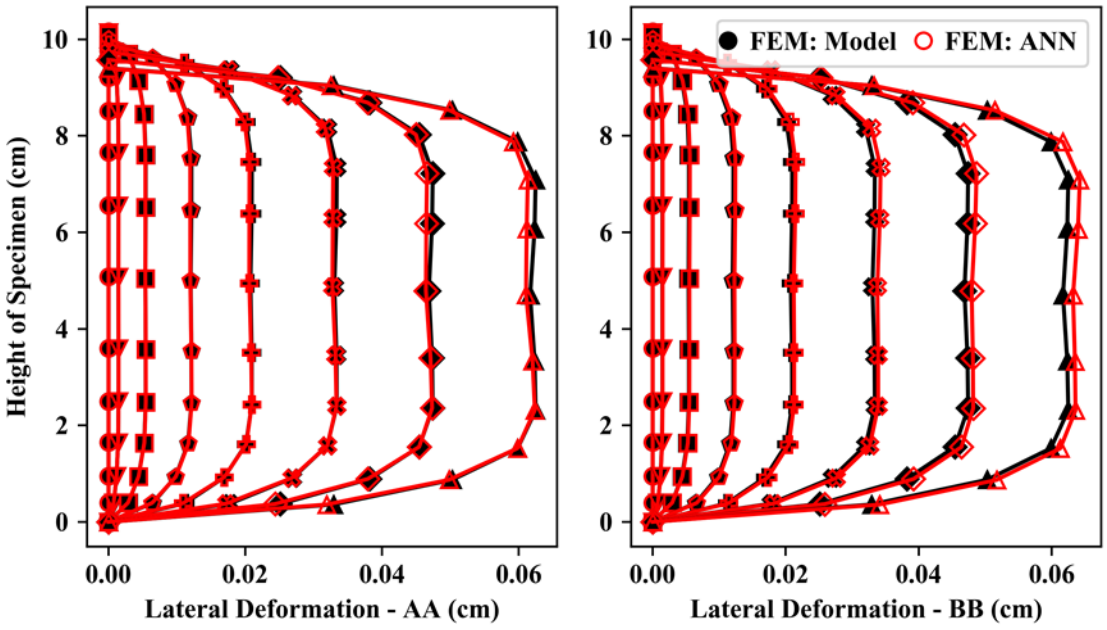


Figure 7-27: Comparison of the deformations calculated by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

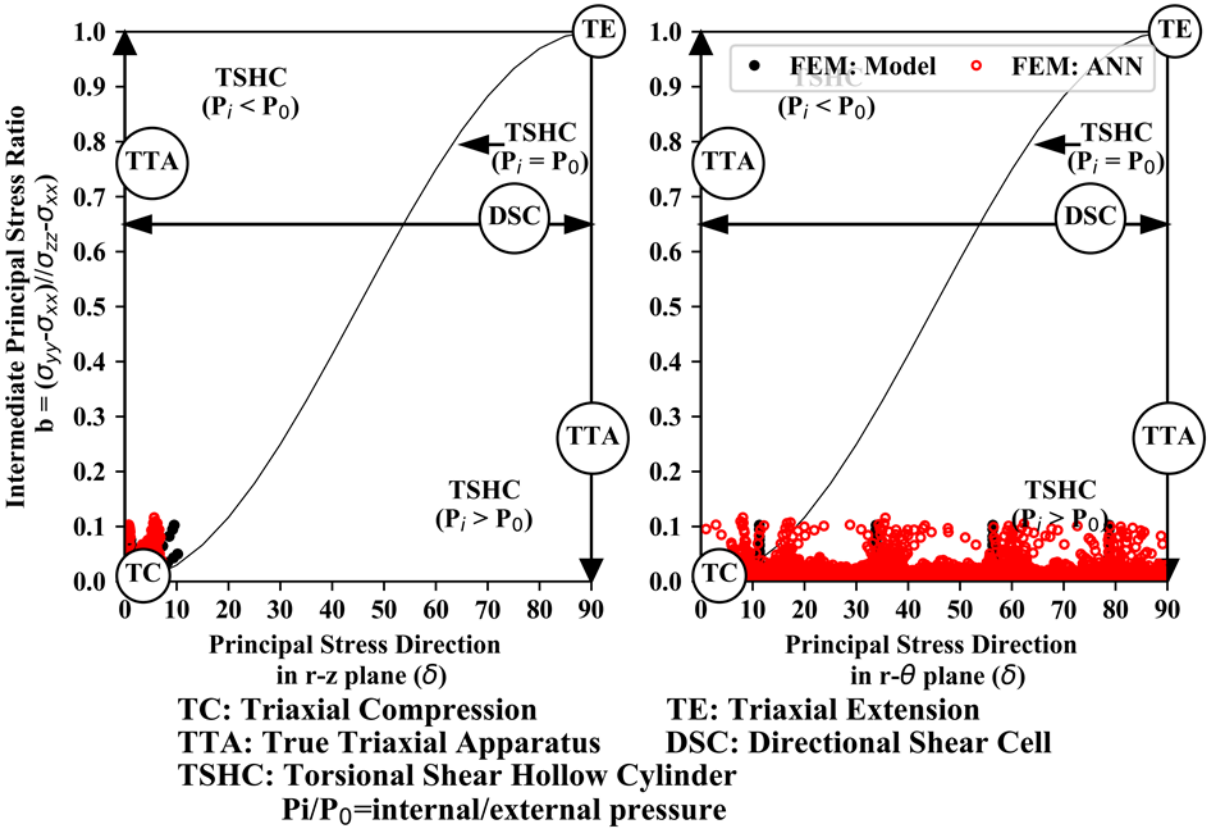


Figure 7-28: Comparison of the stress space calculated by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

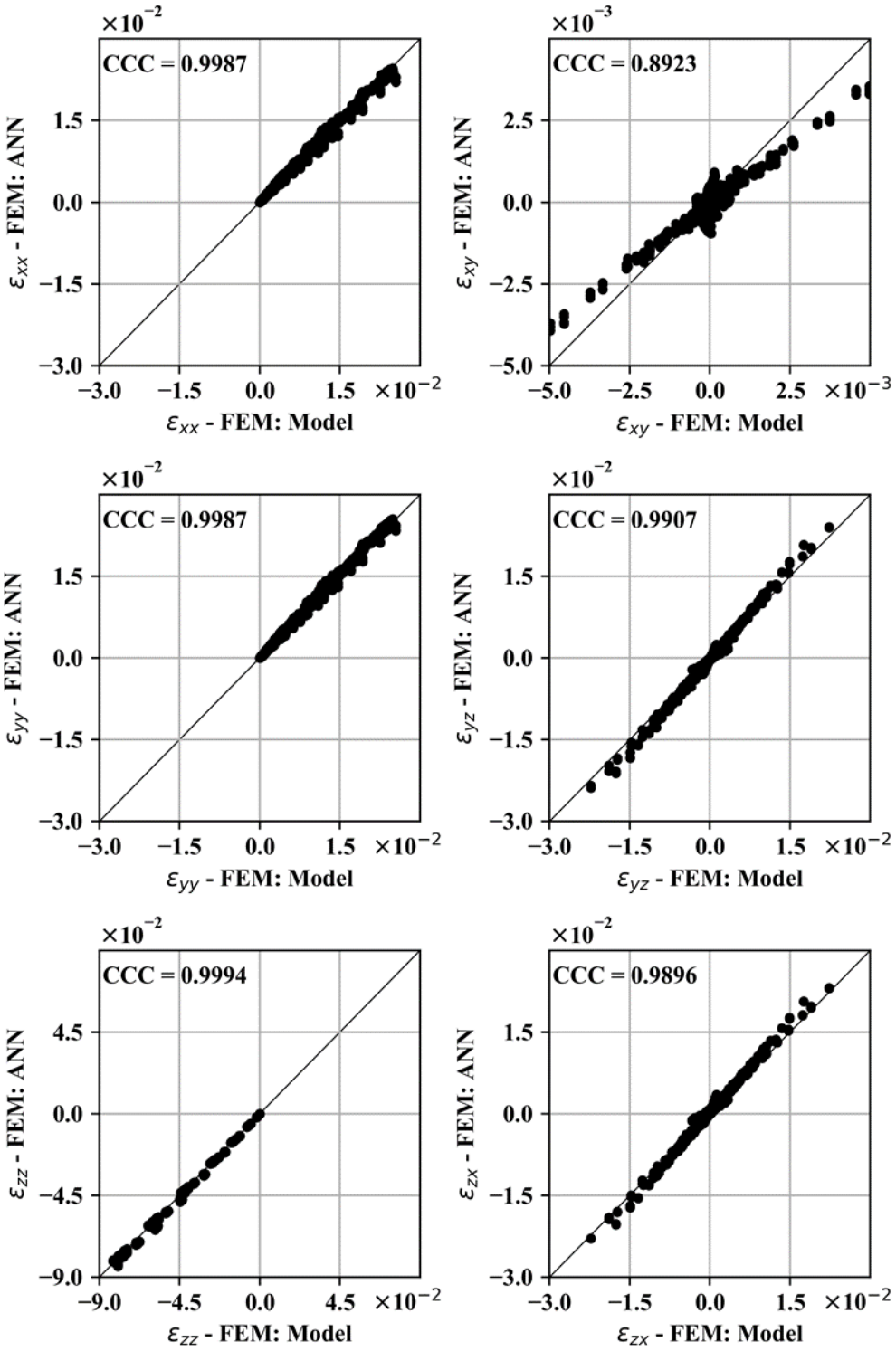


Figure 7-29: Comparison of the strains calculated the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

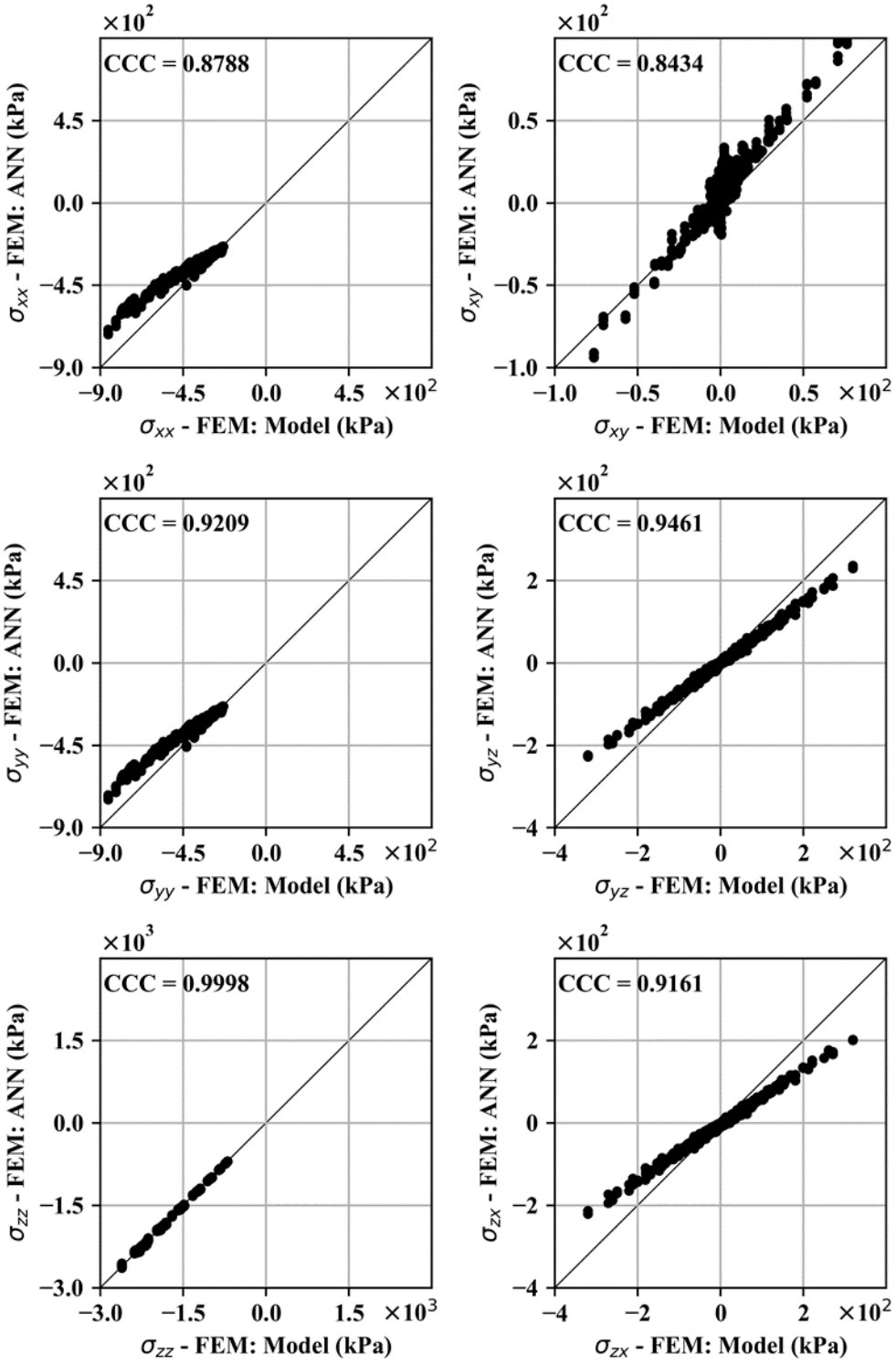


Figure 7-30: Comparison of the stresses calculated the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

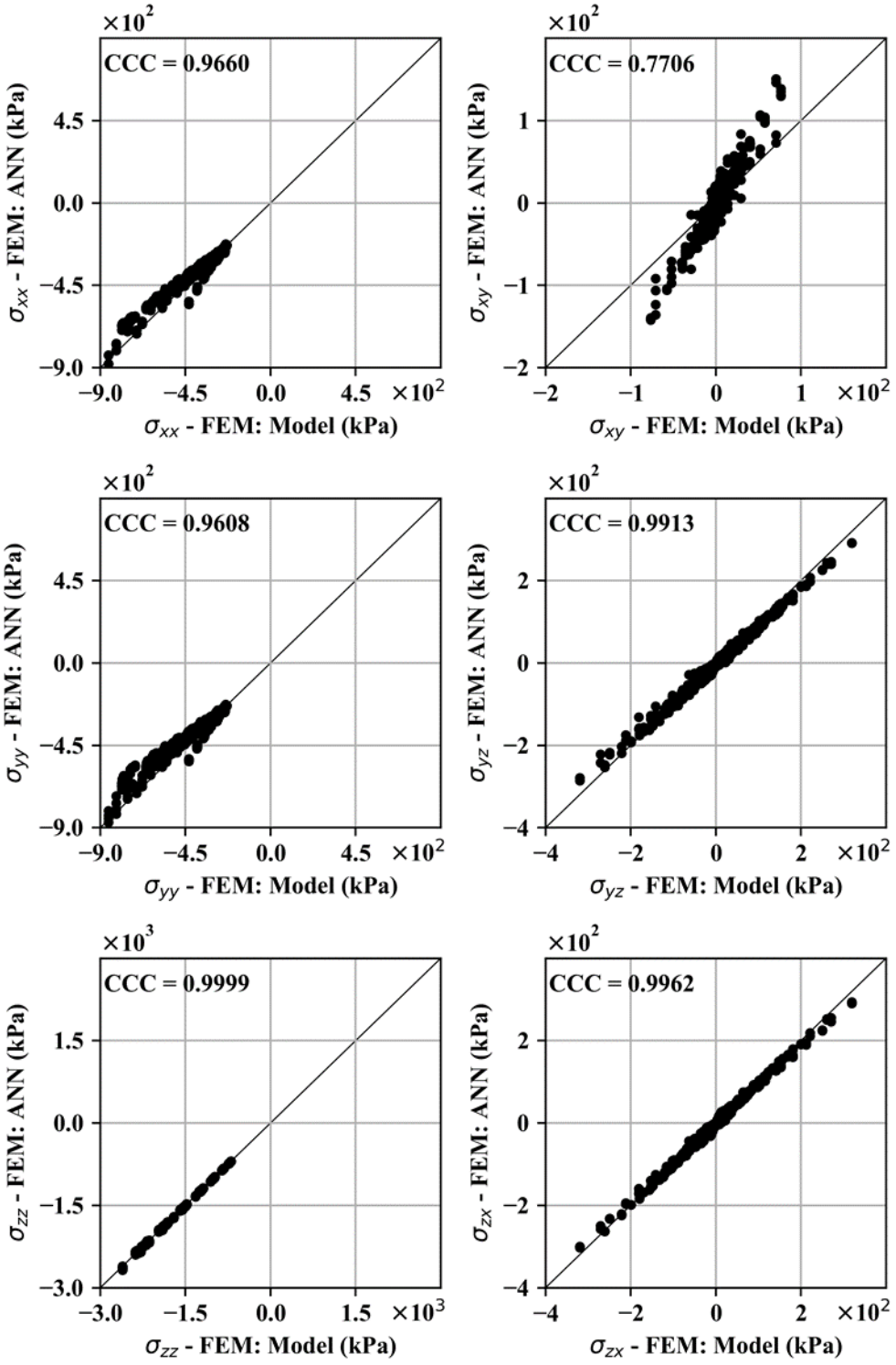


Figure 7-31: Comparison of the stresses calculated the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim in which the pretrained LE dataset was included in the training dataset during the SelfSim analysis

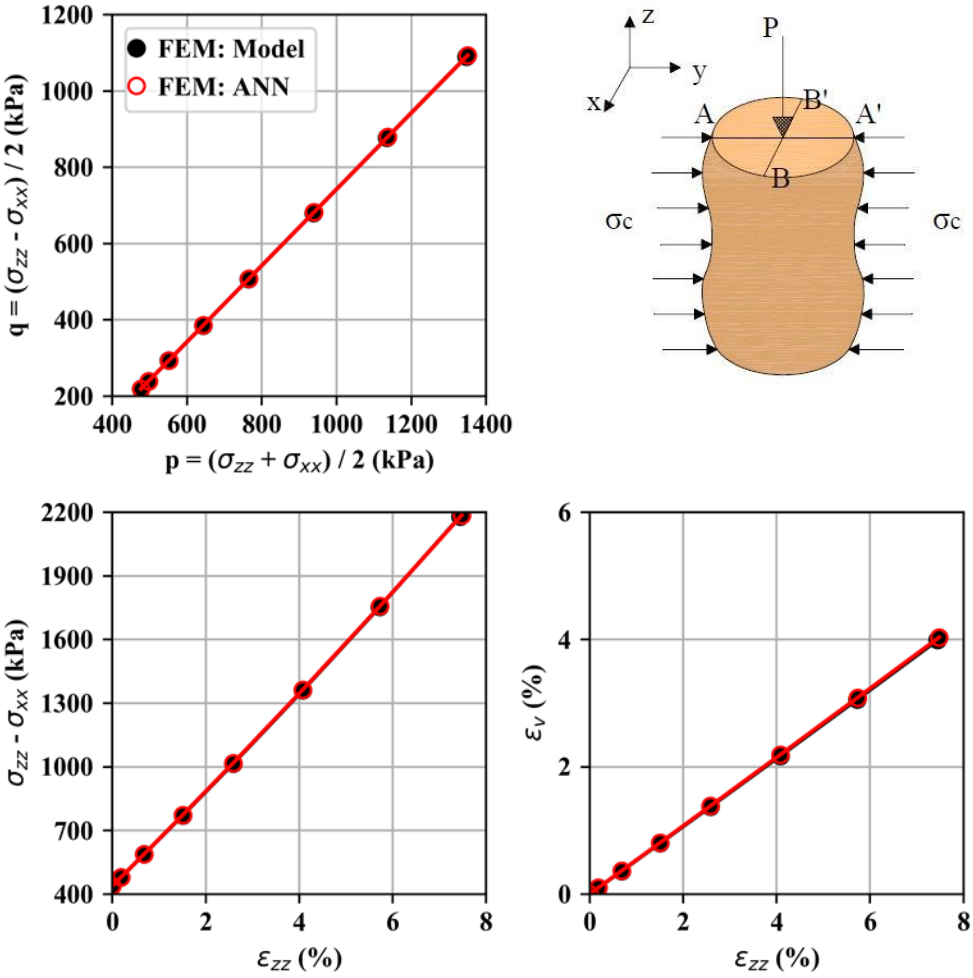


Figure 7-32: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

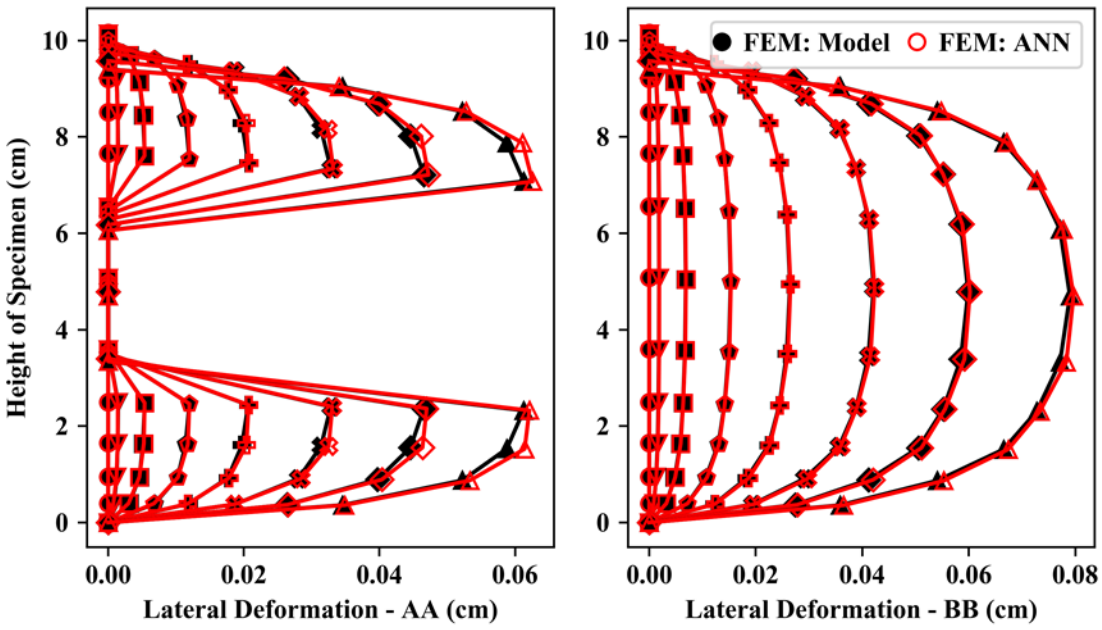


Figure 7-33: Comparison of the deformations calculated by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

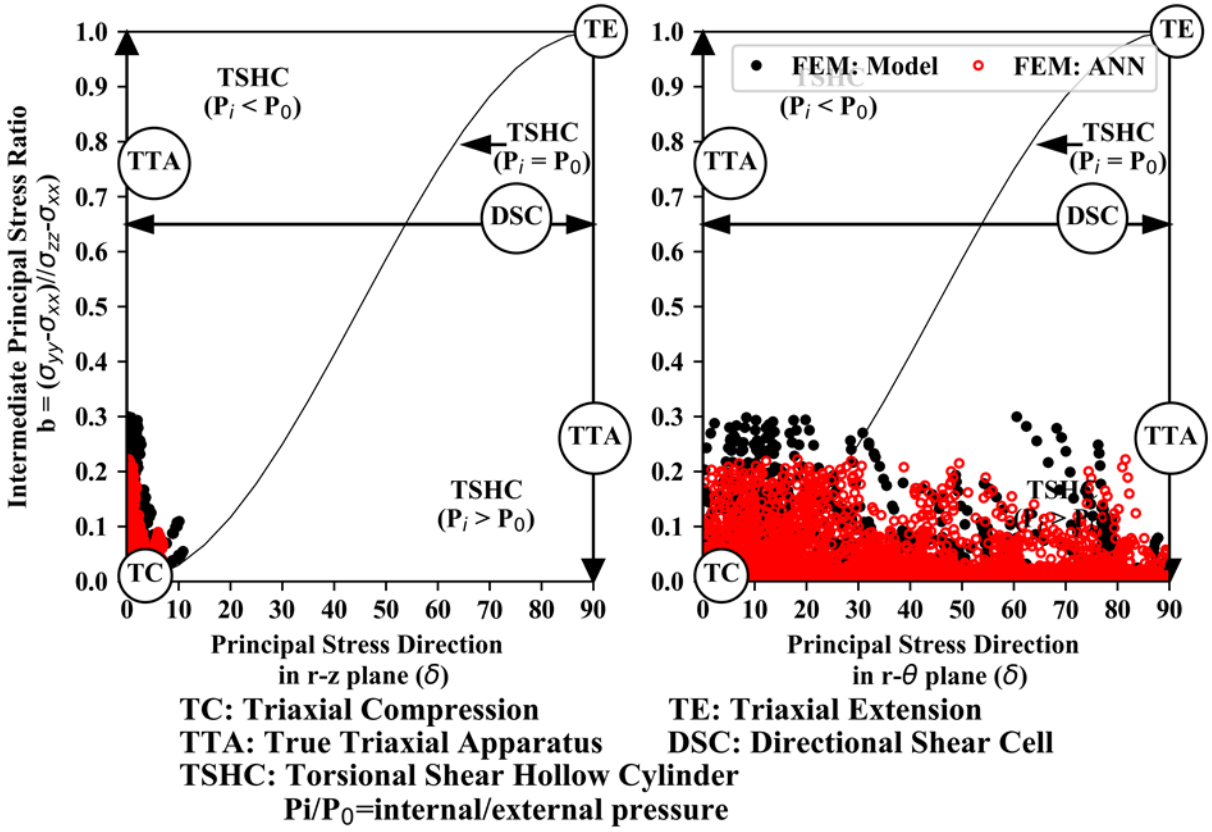


Figure 7-34: Comparison of the stress space calculated by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

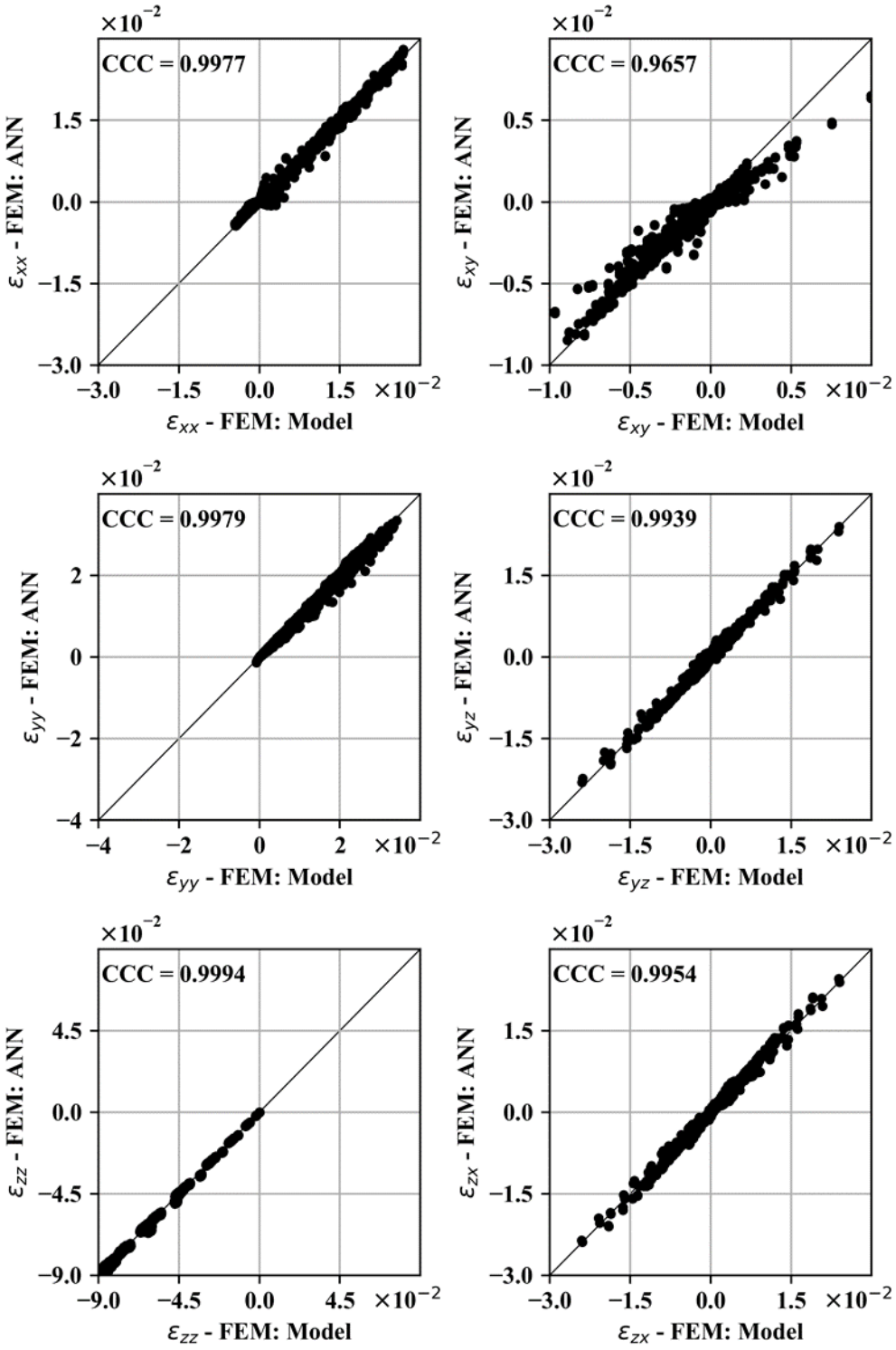


Figure 7-35: Comparison of the strains calculated the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

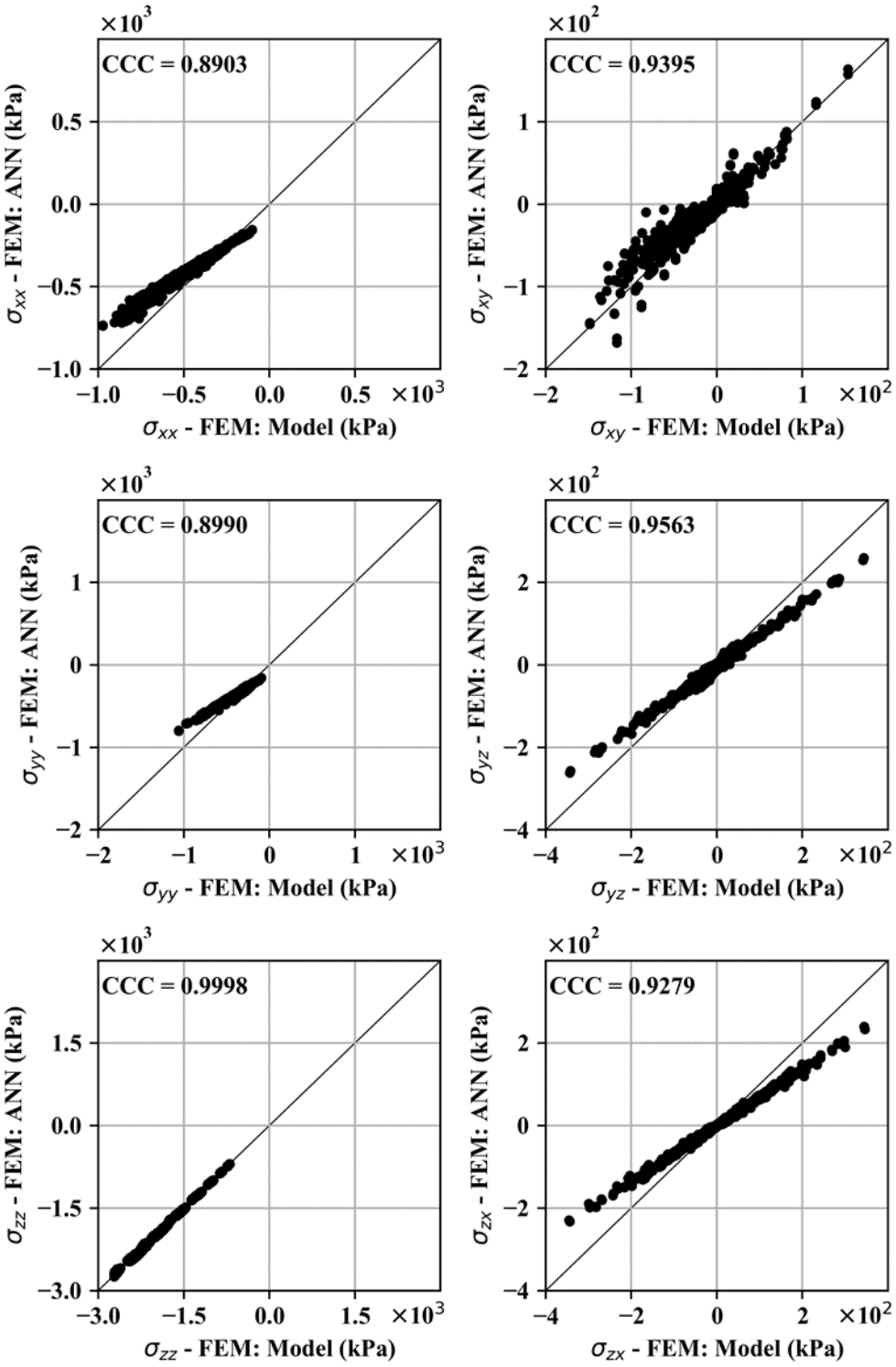


Figure 7-36: Comparison of the stresses calculated the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

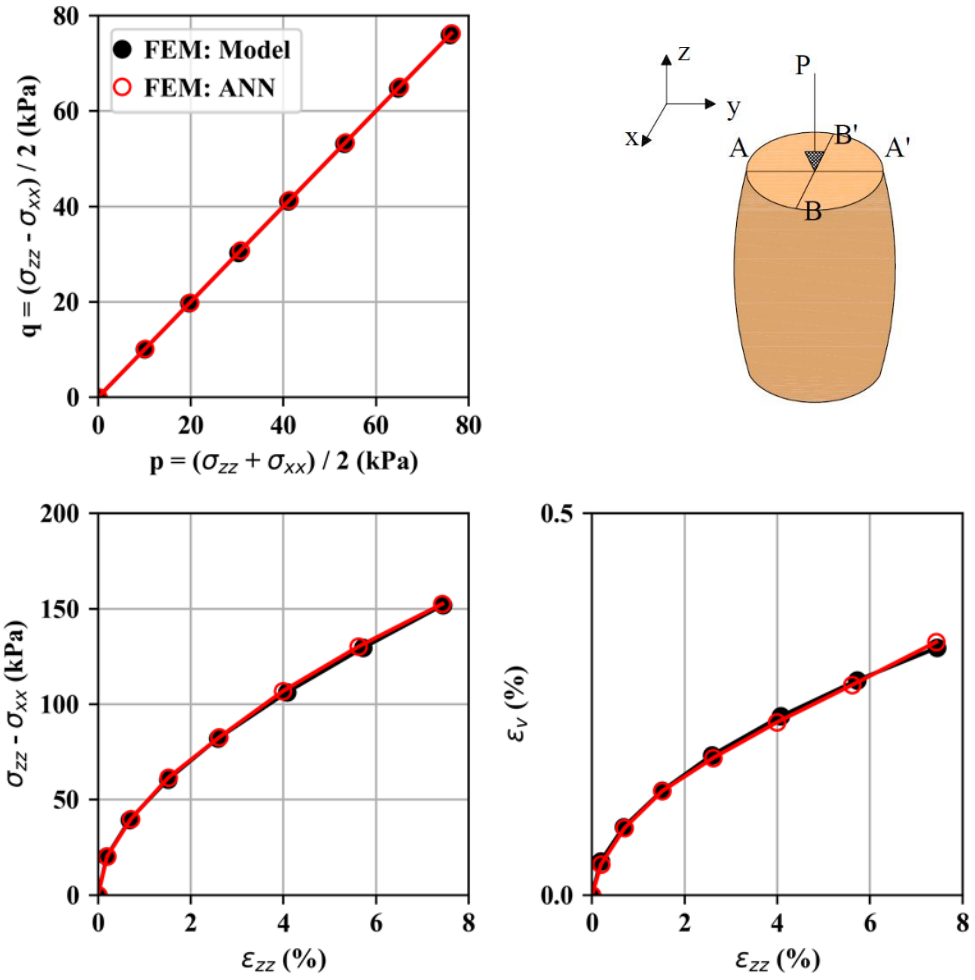


Figure 7-37: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

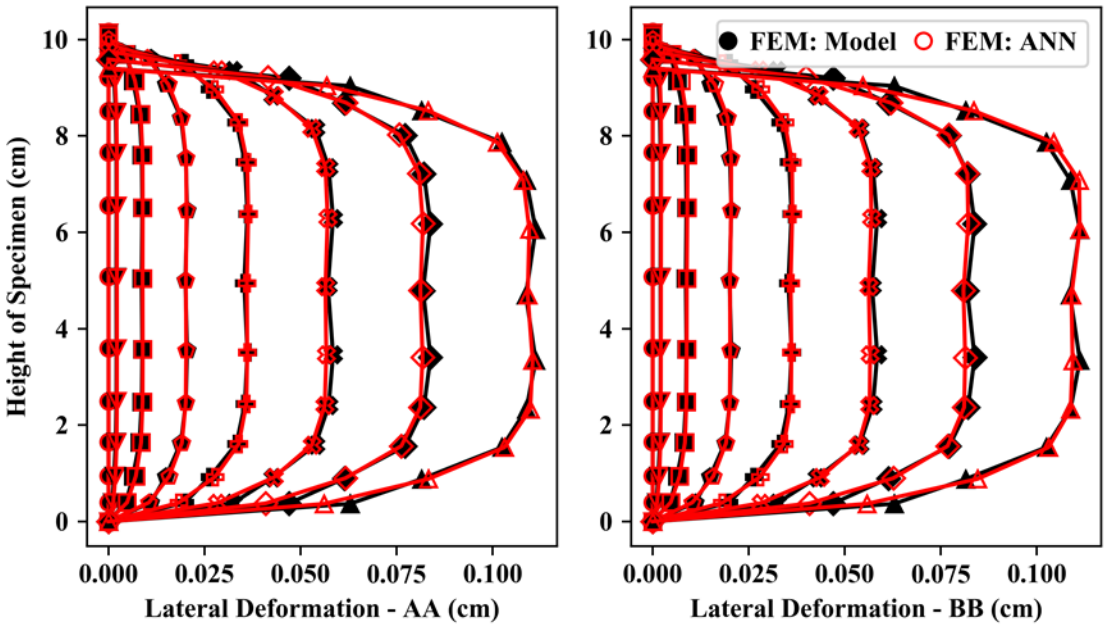


Figure 7-38: Comparison of the deformations calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

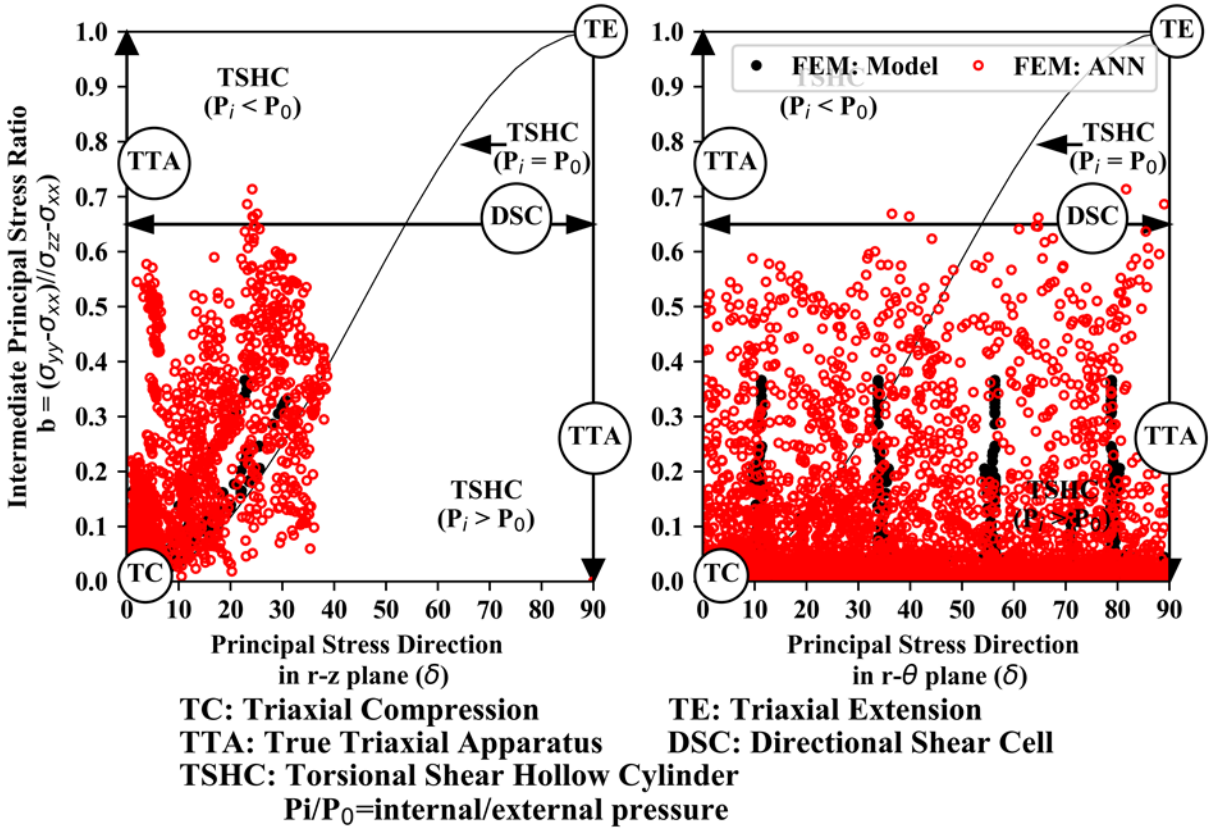


Figure 7-39: Comparison of the stress space calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

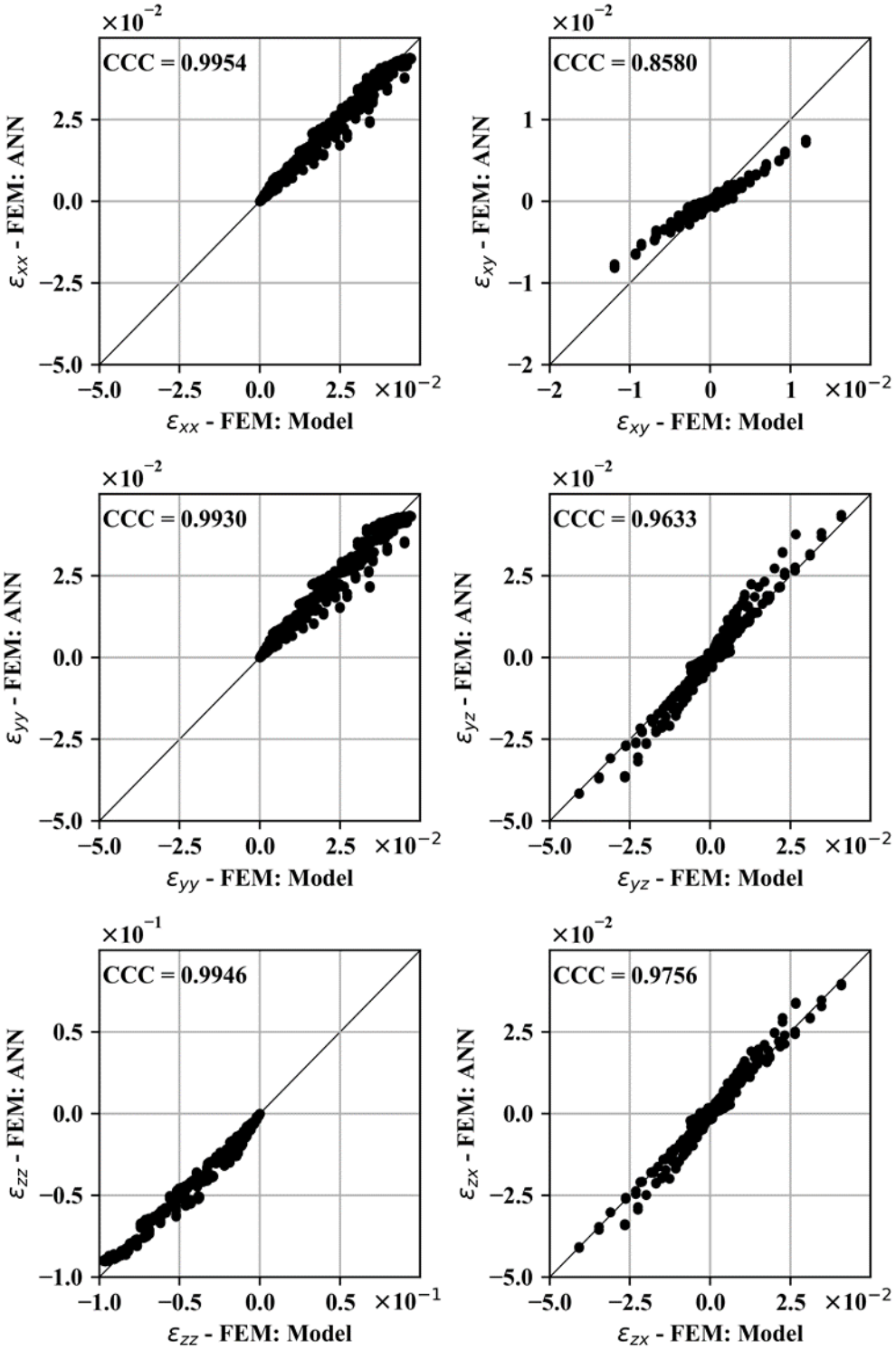


Figure 7-40: Comparison of the strains calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

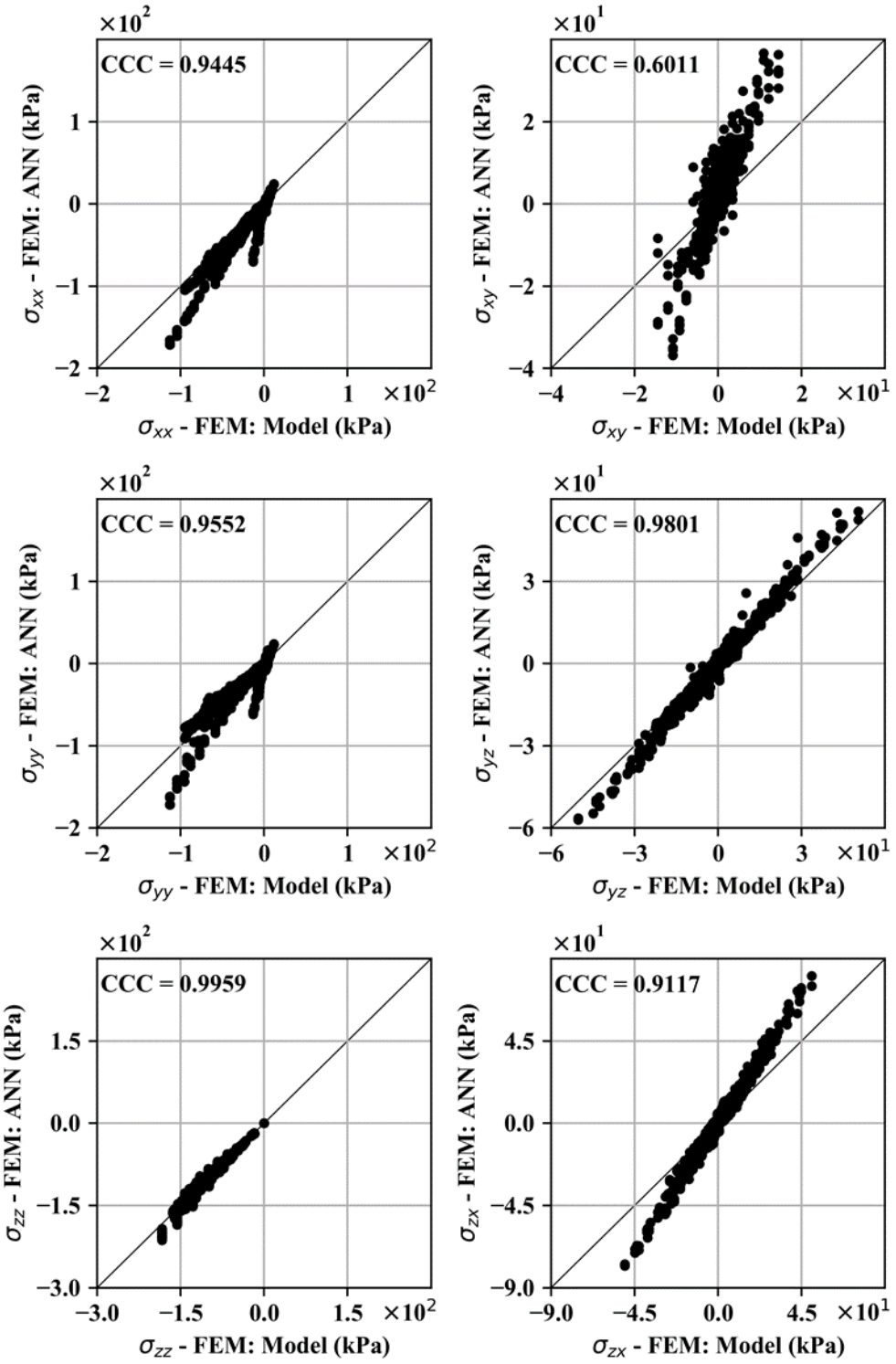


Figure 7-41: Comparison of the stresses calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

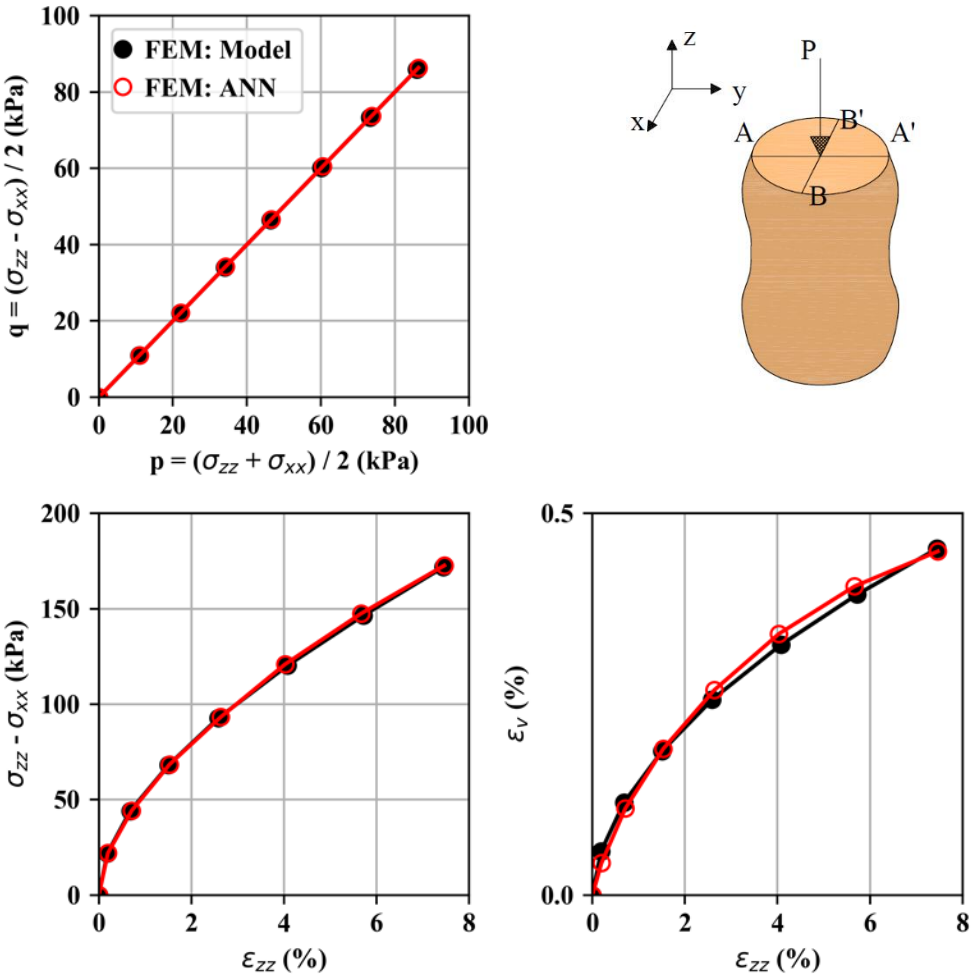


Figure 7-42: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

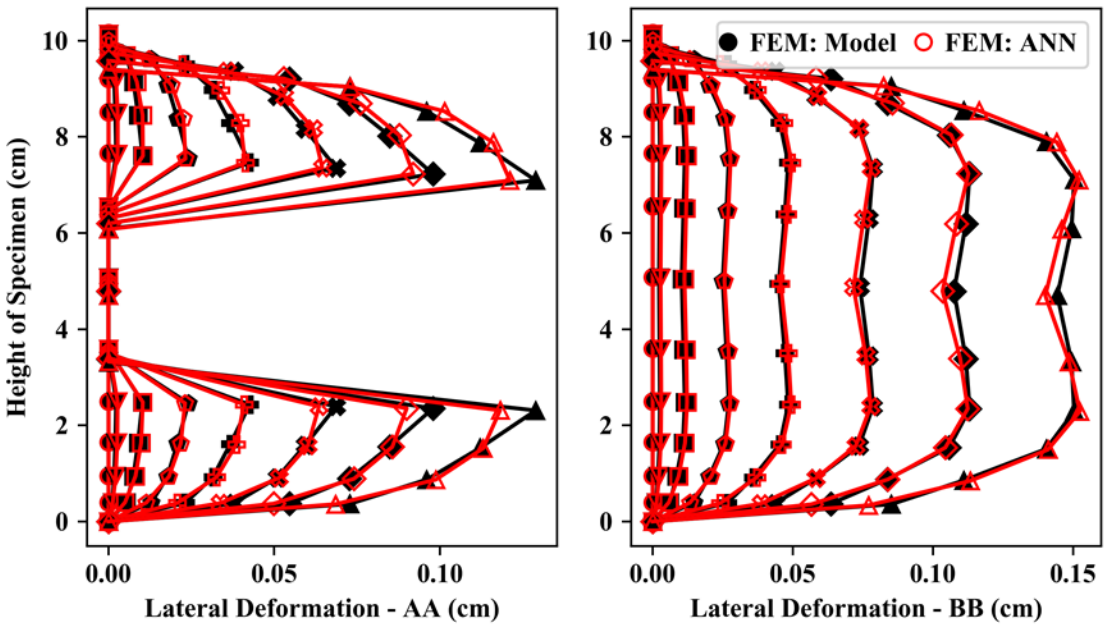


Figure 7-43: Comparison of the deformations calculated by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

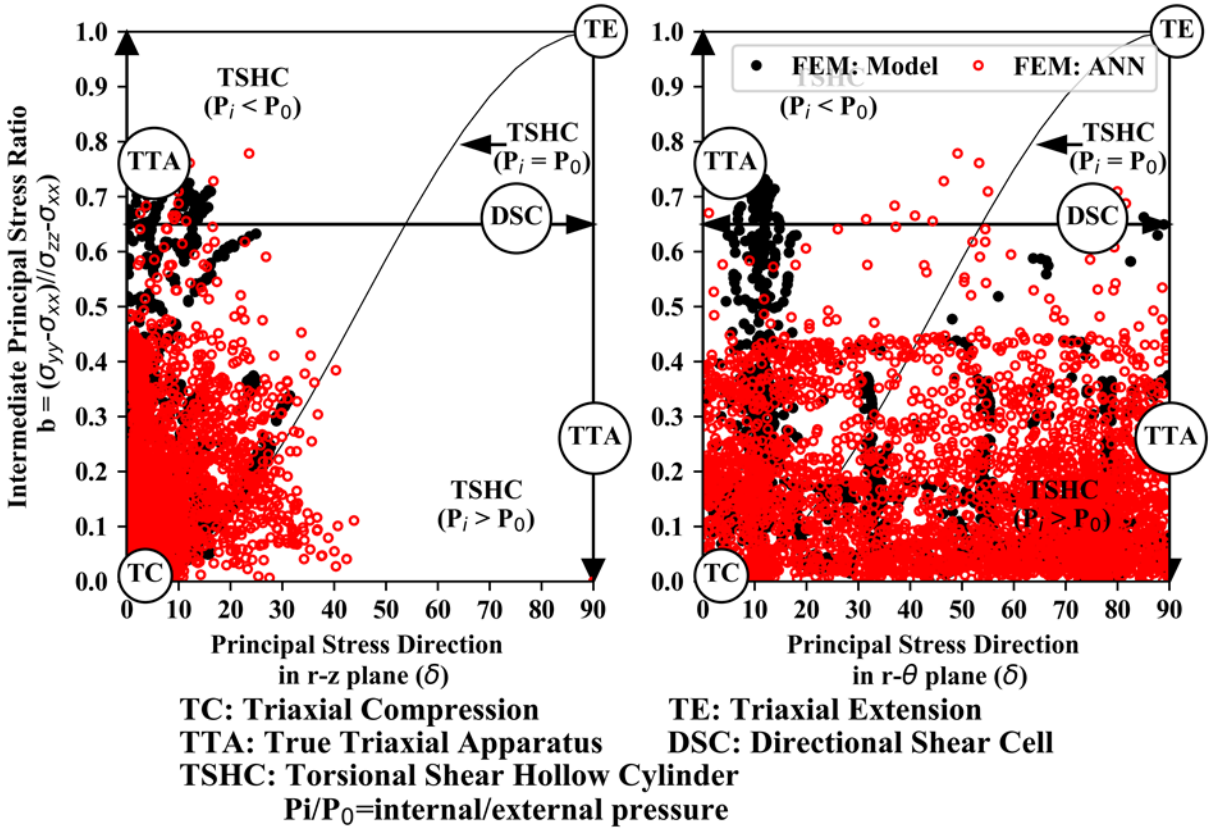


Figure 7-44: Comparison of the stress space calculated by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

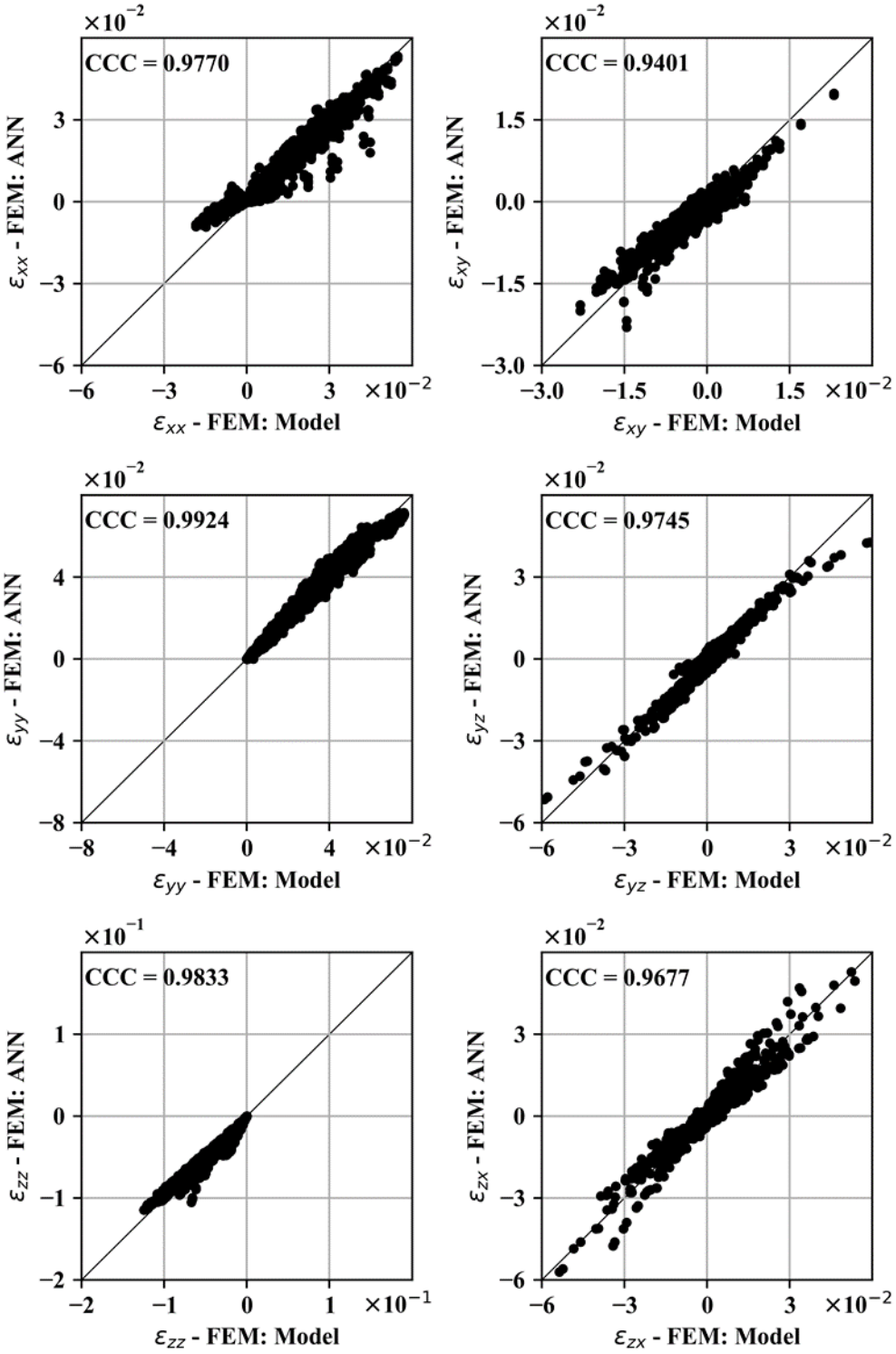


Figure 7-45: Comparison of the strains calculated the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

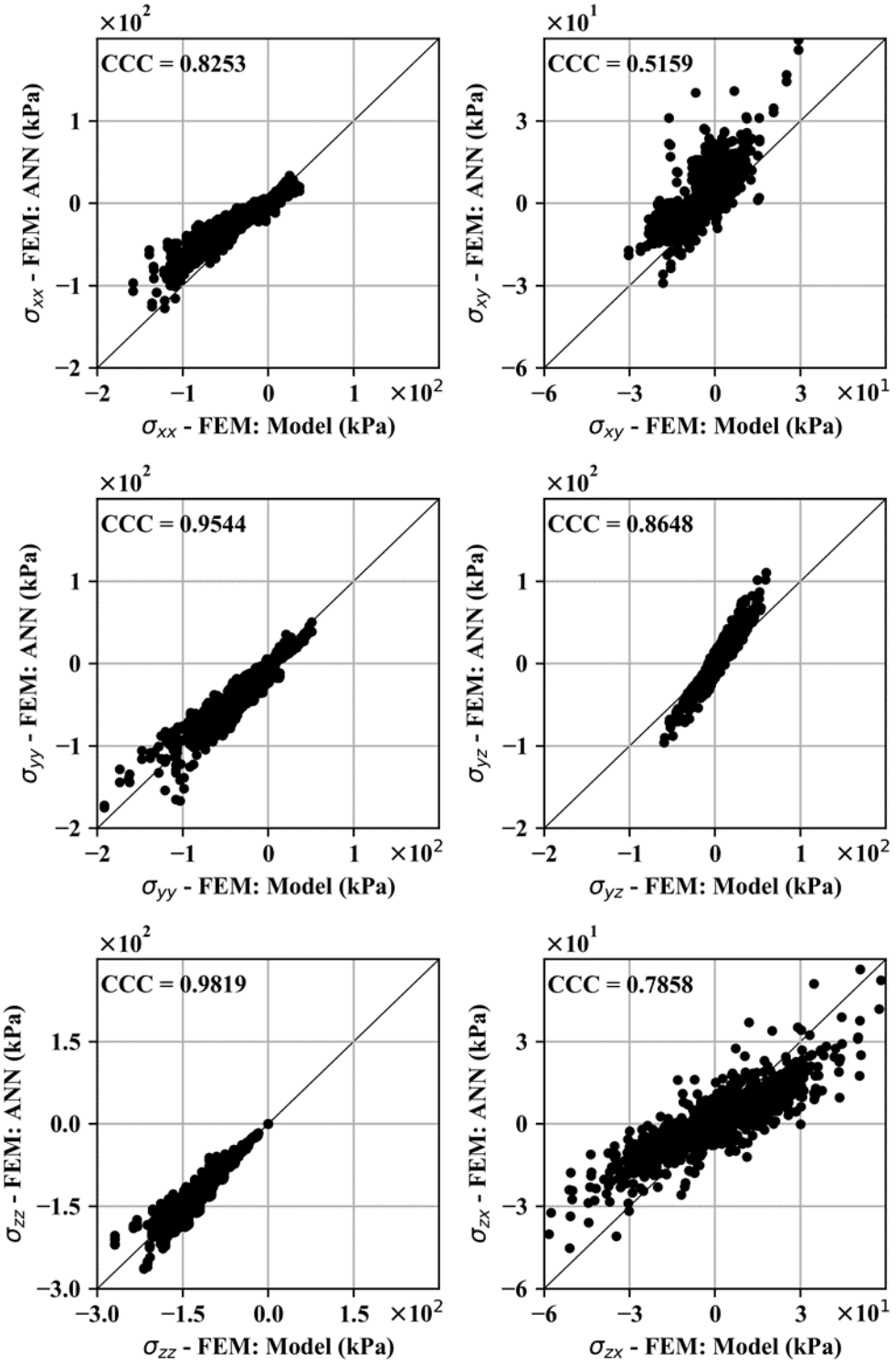


Figure 7-46: Comparison of the stresses calculated the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim.

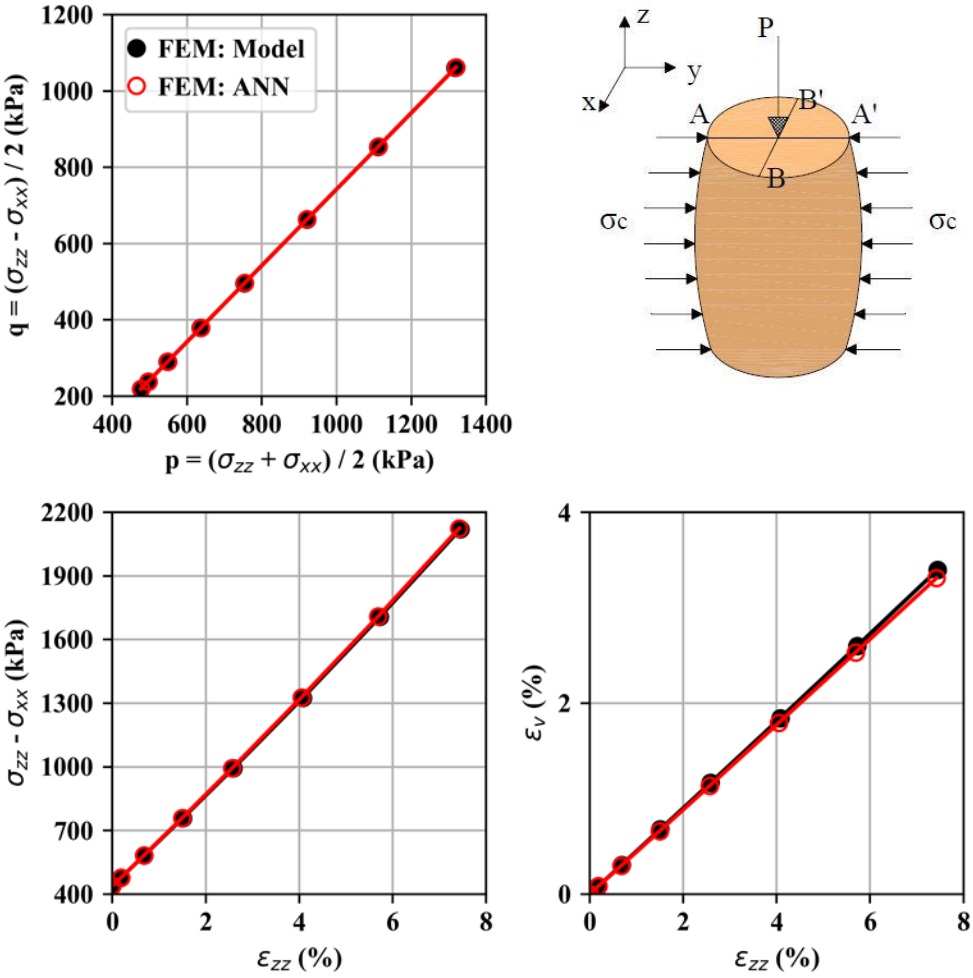


Figure 7-47: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim based of the MTX-LE model

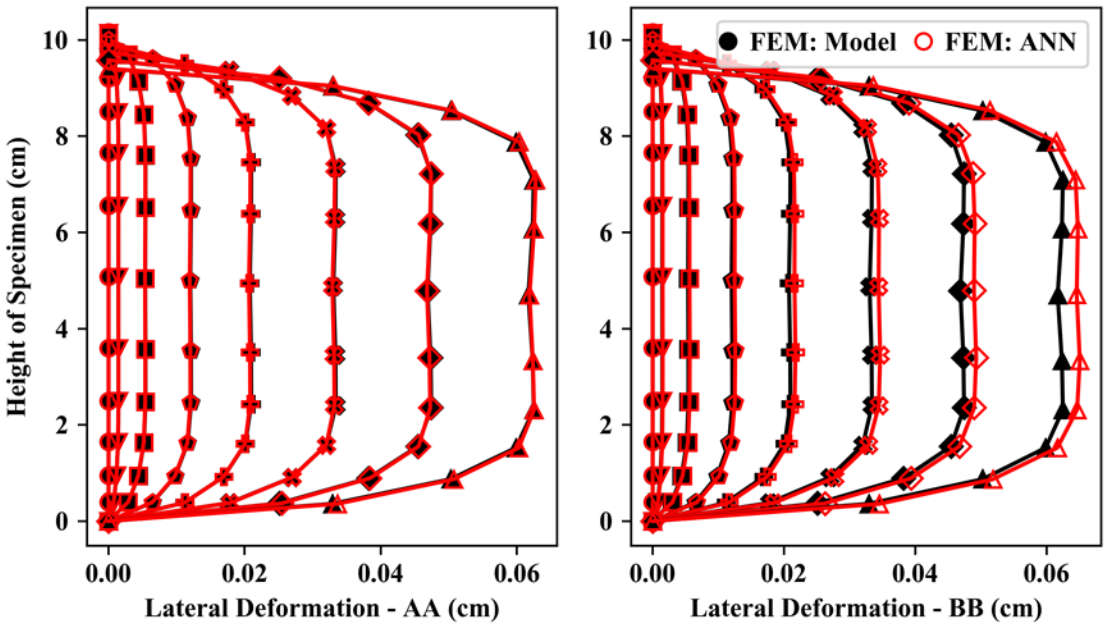


Figure 7-48: Comparison of the deformations by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the MTX-LE model

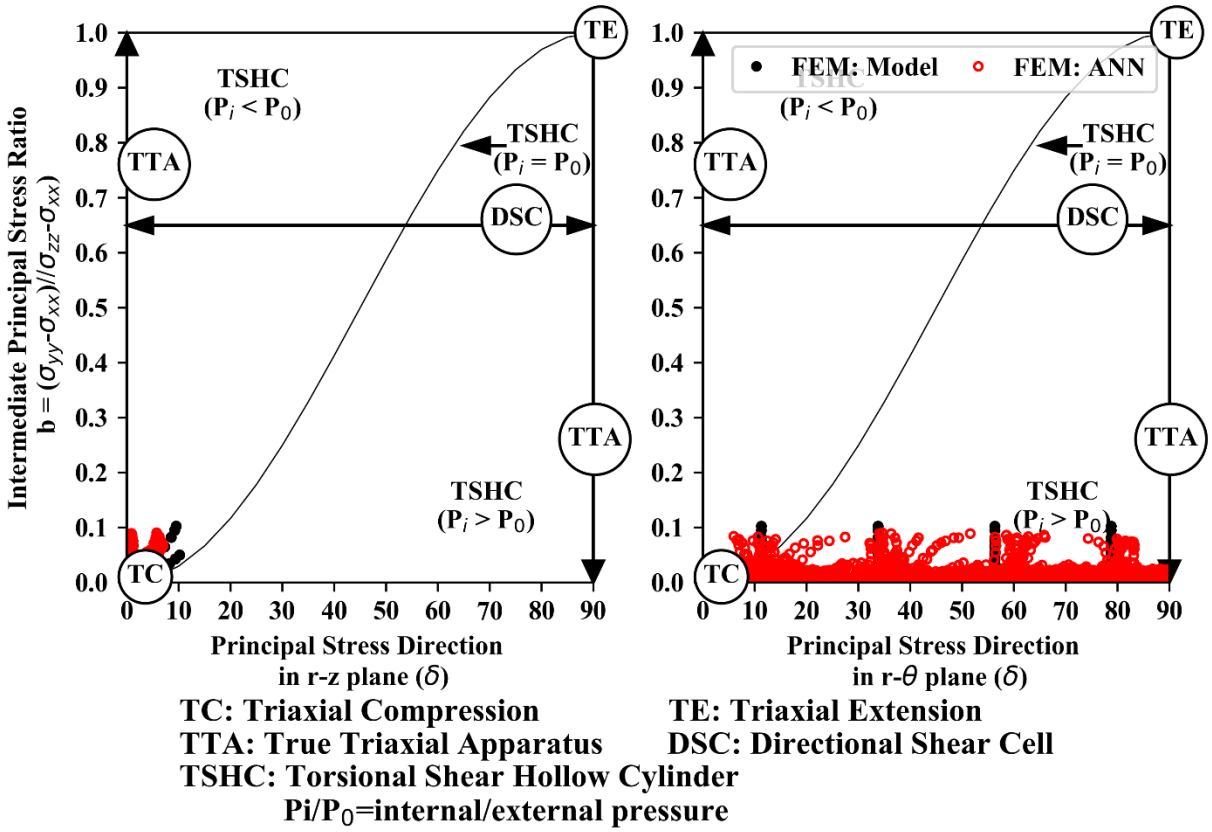


Figure 7-49: Comparison of the stress space by the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the MTX-LE model

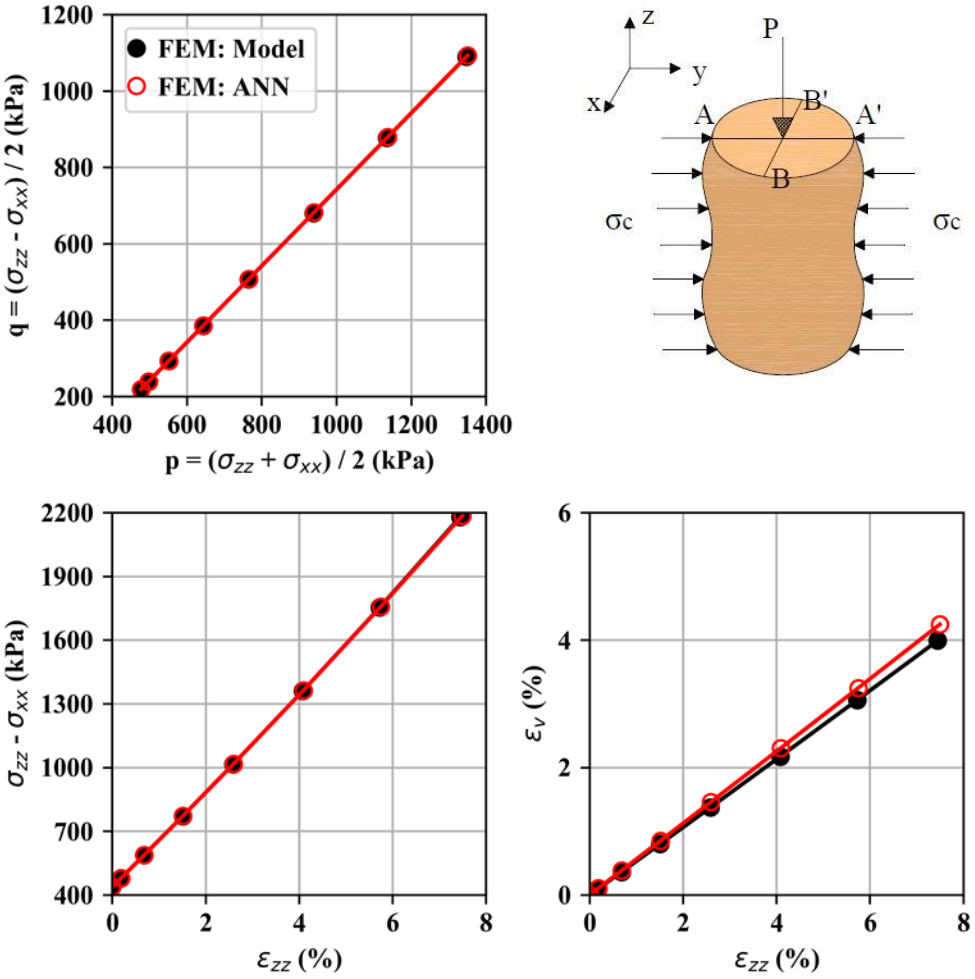


Figure 7-50: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim based of the CTX-LE model

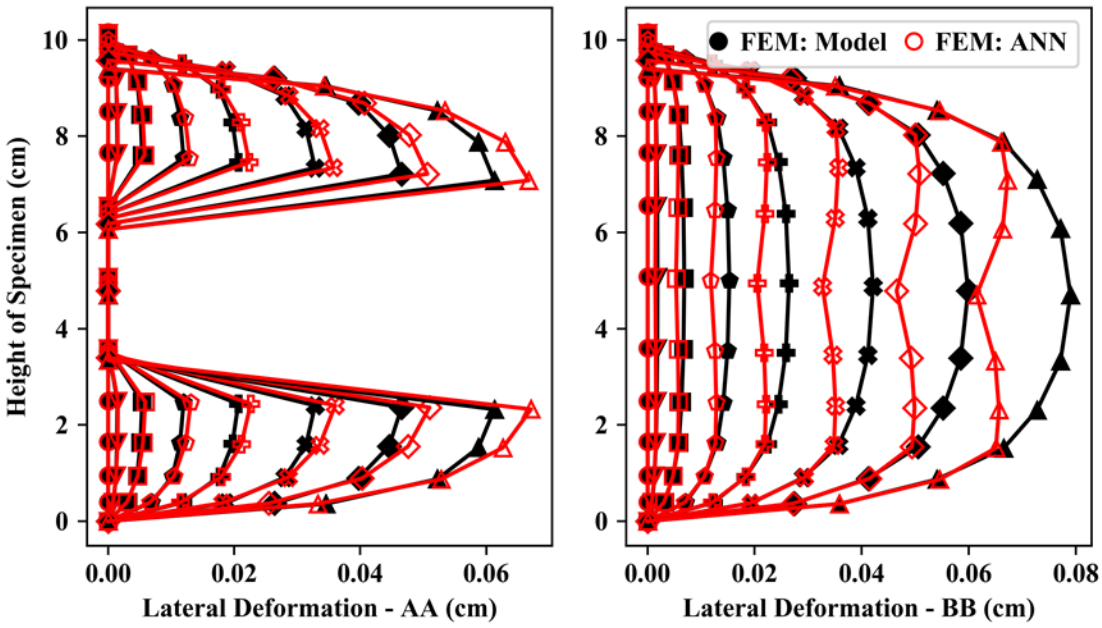


Figure 7-51: Comparison of the deformations by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the CTX-LE model

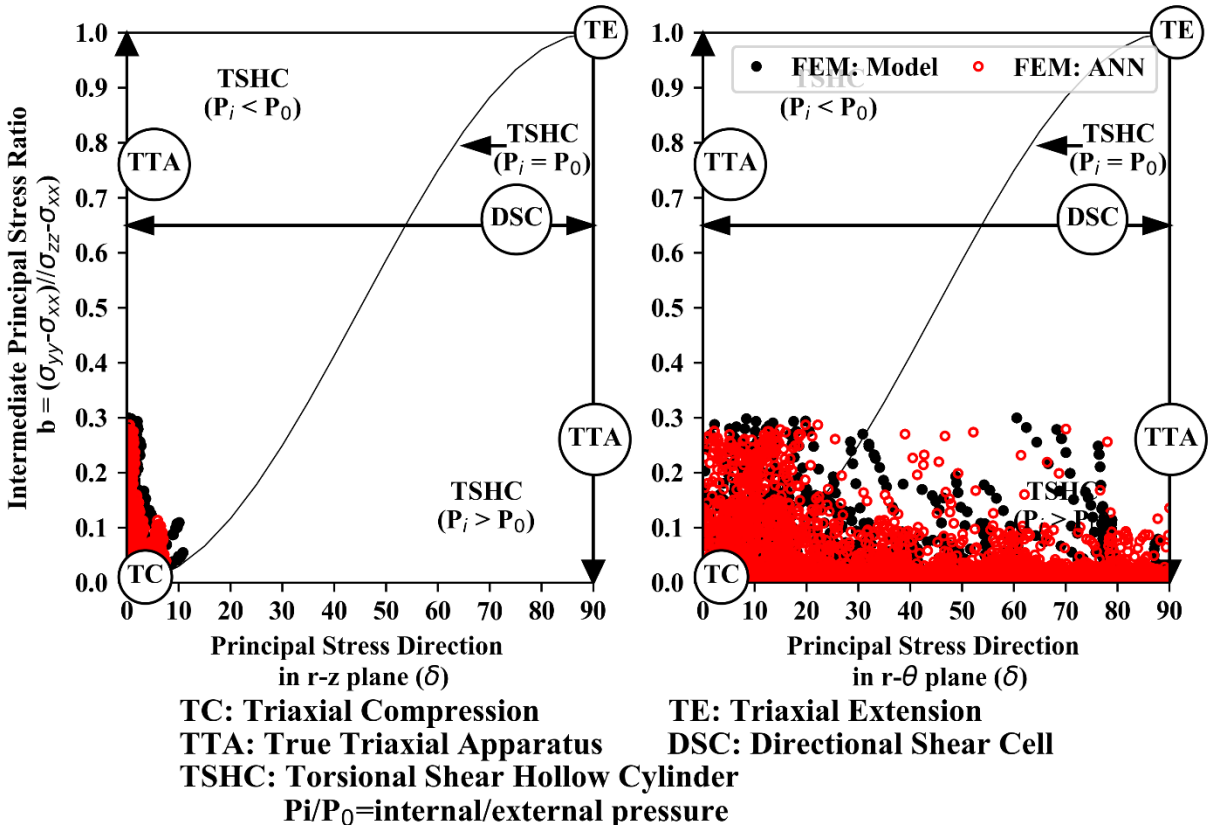


Figure 7-52: Comparison of the stress space by the MTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the CTX-LE model

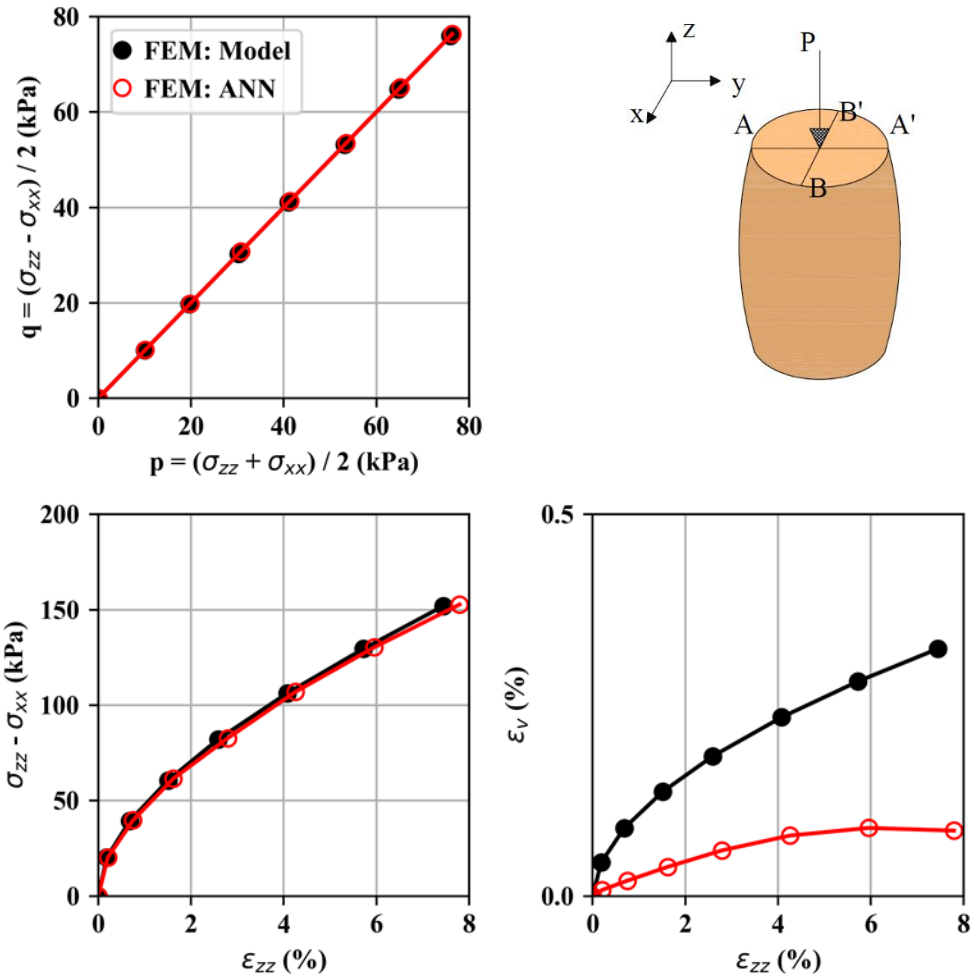


Figure 7-53: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim based of the MUC-PL model

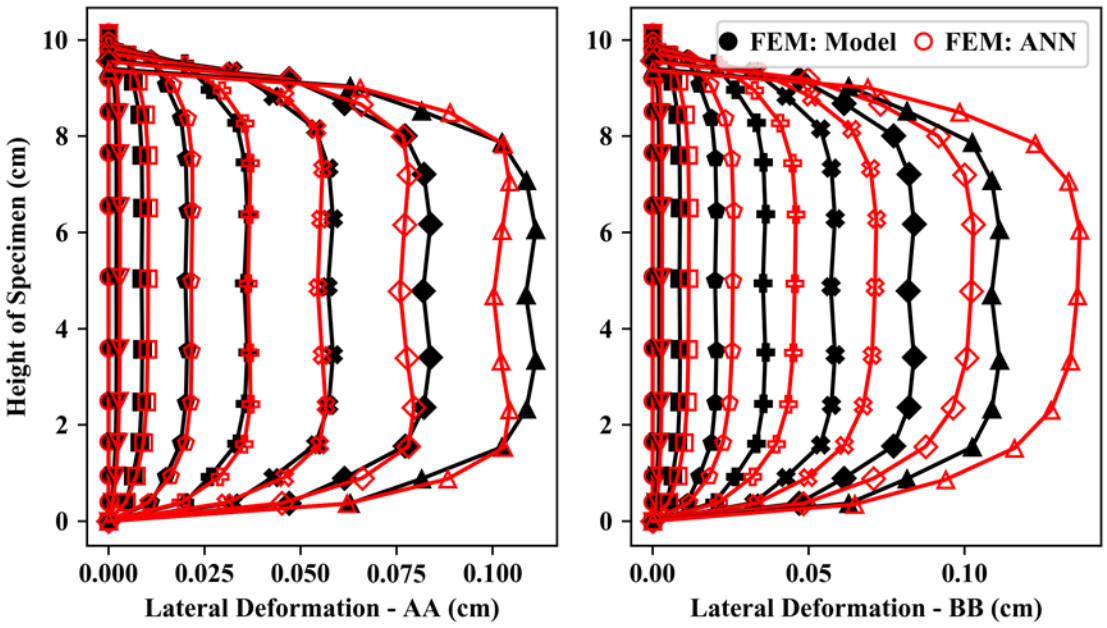


Figure 7-54: Comparison of the deformations by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the MUC-PL model

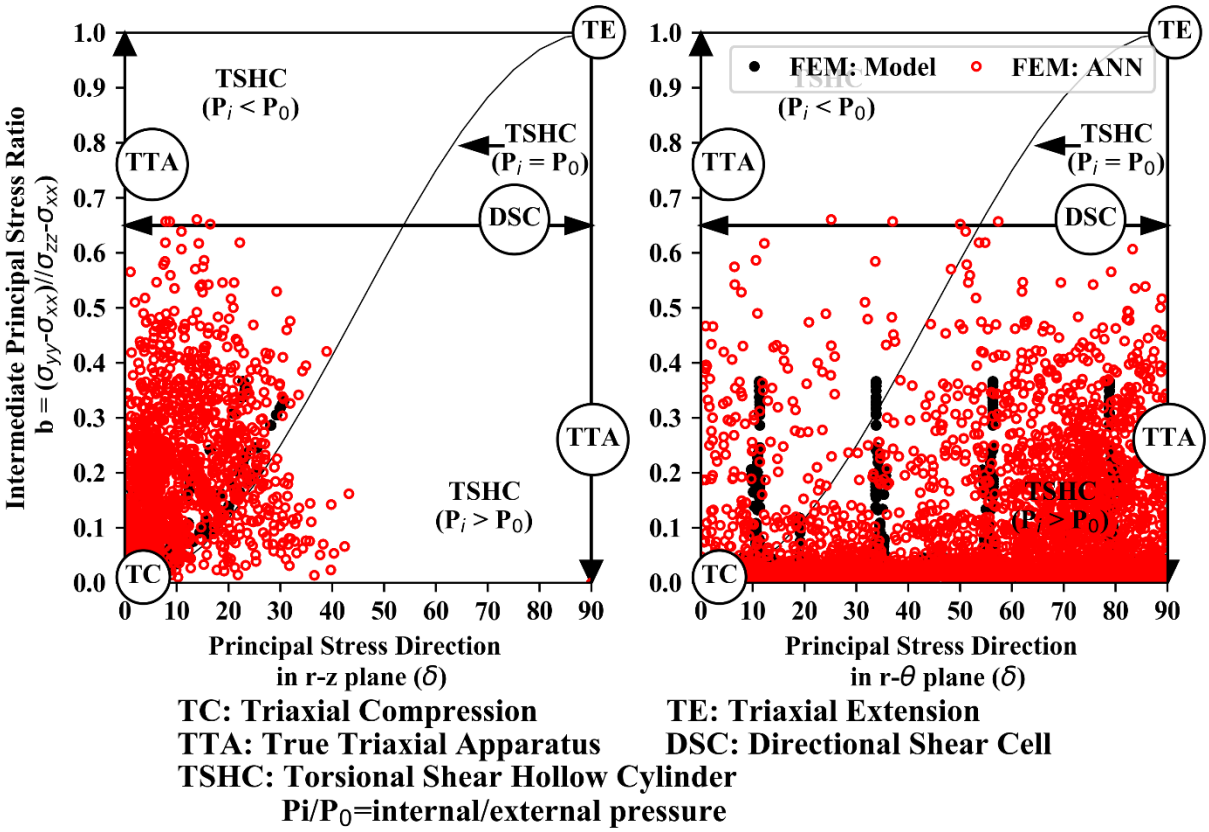


Figure 7-55: Comparison of the stress space by the CUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the MUC-PL model.

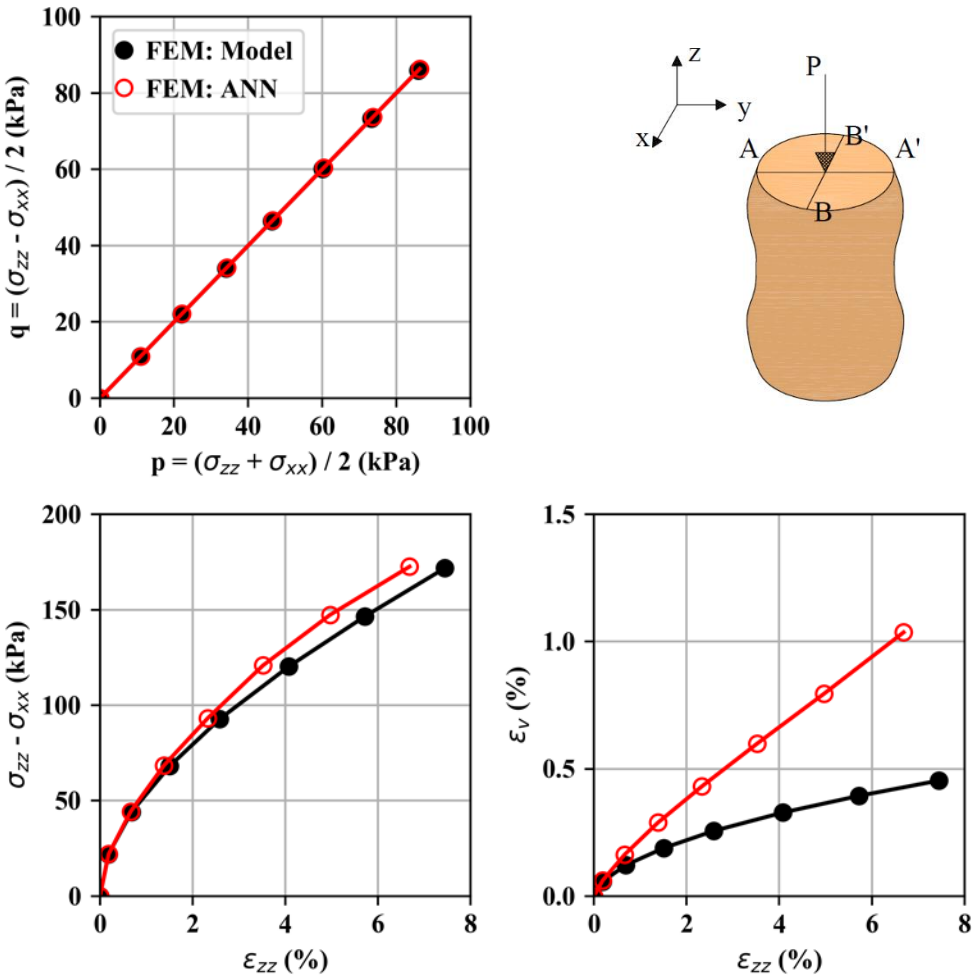


Figure 7-56: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain calculated by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim based of the CUC-PL model

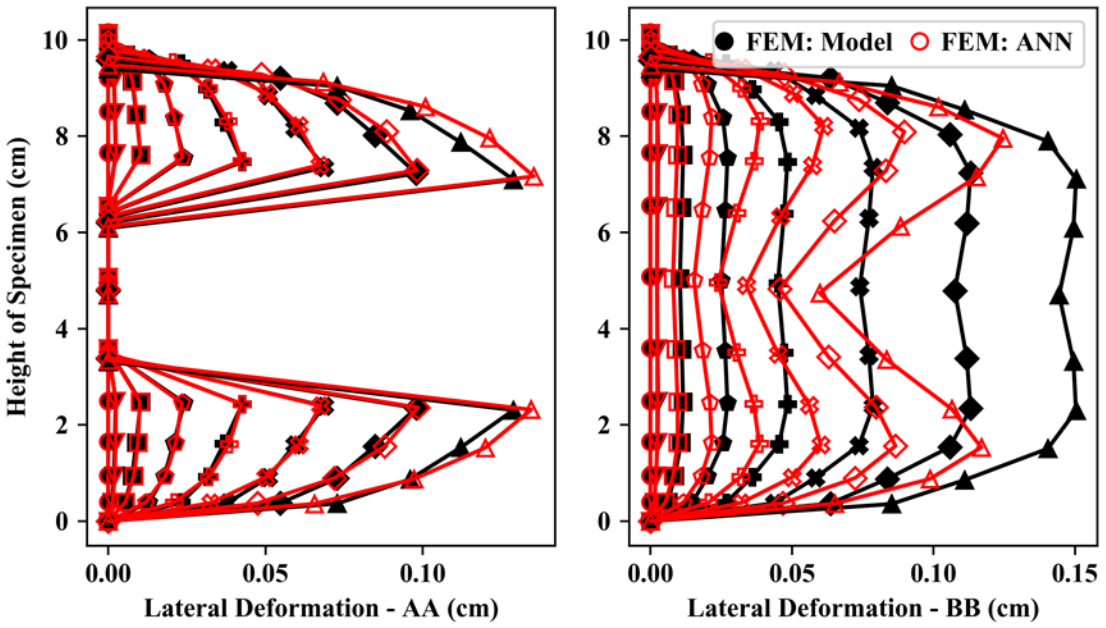


Figure 7-57: Comparison of the deformations by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the CUC-PL model

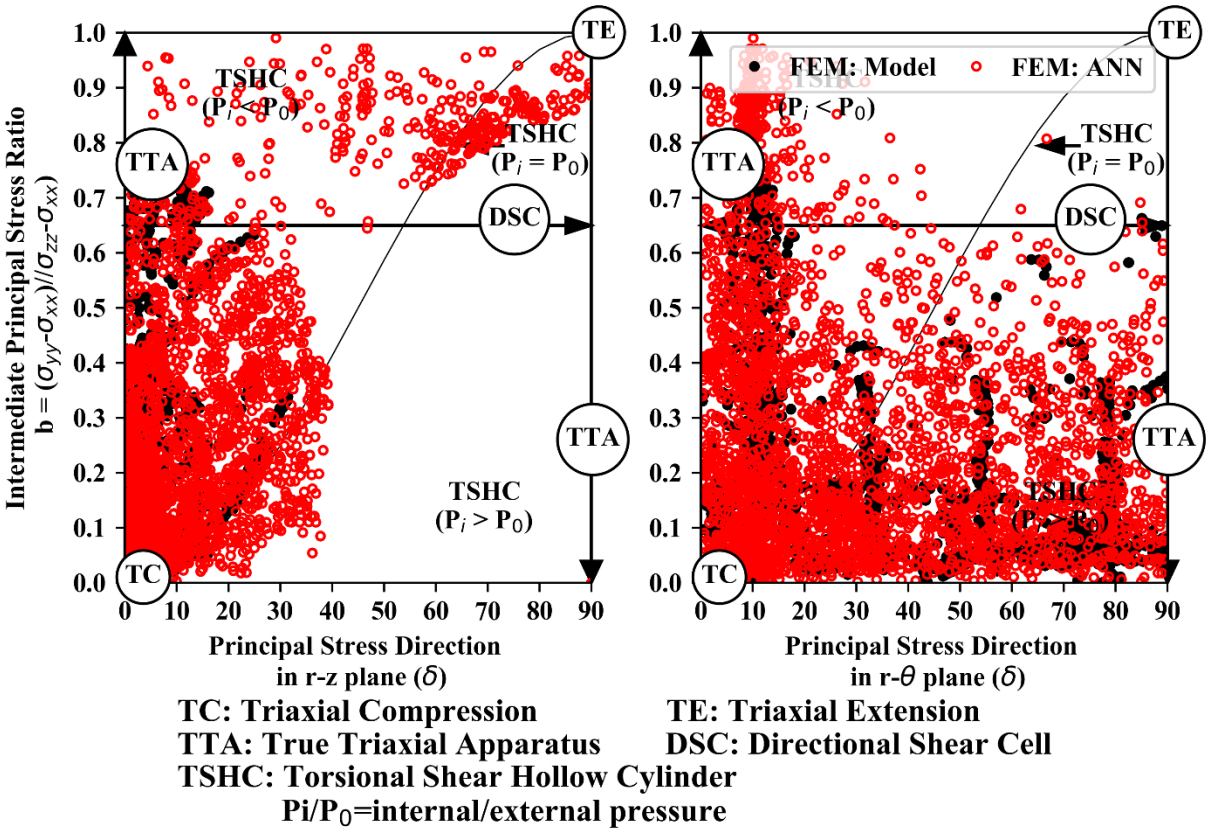


Figure 7-58: Comparison of the stress space by the MUC-PL FEM analysis vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the CUC-PL model

CHAPTER 8. Applying the New SelfSim Code to Ottawa Sand Triaxial Tests

The new code is to be used to learn real sand behavior tested at the University of Illinois through a series of advanced testing. A description of those tests and the SelfSim learning of their behavior is presented in this chapter.

8.1 Description of the Laboratory Tests

Asmar (2017) and Hashash and Asmar (2018) describe a set of 12 drained triaxial compression tests performed on Ottawa sand, which is a uniform, pure quartz sand with rounded particles. The grain size distribution of Ottawa sand in show in Figure 8-1. Six of those tests were conventional triaxial compressions tests, with a friction cap, while the other 6 were performed with a modified triaxial cell. The modification of the triaxial test cell was to increase the states of stress and generate multiple stress paths in one test. The modification encompassed adding lateral clamps that constrain lateral deformation around the mid height of the specimen. Hashash and Asmar (2018) describes the development of this new device: "*This device inherits all features of a conventional triaxial test and adds lateral restraint clamps to increase nonuniformity in specimen deformation, combined with a digital photogrammetry system to measure the 3-D deformed shape of the specimen. The design of the restraint clamps was optimized using numerical simulations which showed that the sheared specimen includes shear modes that cannot currently be mobilized with available testing devices*".

The tests varied in addition to the testing device, in two other properties: the height-to-diameter ratio of the samples (2 to 1 or 1 to 1), and the initial confining pressures. The relative densities of all tests were similar varying from 42 to 48%. A summary of the conducted tests is presented in Table 8-1.

8.2 SelfSim Analyses Using the Previous Code

In Hashash and Asmar (2018), the behavior of the sand was learned through SelfSim analyses for the 12 tests, using the previous code. Many challenges were faced, and some modifications were made to allow for the completion of those 12 SelfSim analyses:

- the data of 114 out 912 elements was used for training the ANNs. The rest of the data was not used to save on training time. The elements were selected to represent different zones

of the specimen attempting to cover most of the stress states experienced by the soil specimen.

- Data from transition zones next to the clamps were excluded from the training datasets due: "*it was found that the transition areas from the part of the samples that is covered with clamps and the part that is not covered with clamps are areas of complexity and including the calculated lateral displacements from these areas will result in global behaviors that doesn't match very well with the measured global behaviors. Therefore, these areas were excluded from the training data set which improved the SeflSim learning and resulted in better match with the global behavior*".

The architecture of the ANN in all tests was 18x12x12x6. The typical timing for successful runs of the SelfSim analyses was around 4 hours, however many unsuccessful runs were typically encountered before a set of hyperparameters result in a successful run.

8.3 SelfSim Analyses Using the New Code

Since the new code employs LS-DYNA, the 12 ABAQUS models were translated into LS-DYNA models preserving the mesh, node and element numbering other details of the models. Rather than the 20-node elements used in the previous study, 8-node elements were used in the LS-DYNA models. The reason for this switch is because non-symmetric stiffness matrix, needed for ANN constitutive models, is not supported for the 20-node elements in LS-DYNA, but supported for the selected 8-node elements. Section type "-2" was used and showed better performance compared ti the other sections with support for non-symmetric stiffness matrices as demonstrated in Appendix C: Effect of LS-DYNA Element Formulations on FEM Results. A photo of one of the LS-DYNA models in addition to the deformed meshes of models when subjected to the modified and conventional triaxial boundary conditions are shown in Figure 8-2 and Figure 8-3.

The new code was used to run SelfSim analyses for the 12 tests. The data of all 912 elements was used for training. All the displacement boundary conditions in x and y directions were used, and displacement errors were calculated at all points.

8.3.1 Preprocessing of Training Datasets

While creating the training datasets from the results of the FEM analyses, an important step is taken to enable better training. The initial stresses are deducted from the stresses, which enables the use of a large section of the activation function. For example, if the initial stress in a dataset

was 500 kPa while the maximum final stress was 900 kPa: scaling the stresses to a value less then one (standardization) by dividing them by 1000 would map the 500 to 0.5 and the 900 to 0.9. Hence the section of the activation function used is limited to 0.5 to 0.9. However, if the initial stresses are deducted, the 500 is mapped to 0 while the 900 is mapped to 0.9, and hence the section of the activation functions used varies between 0 and 0.9, and hence richer behavior can be learned. A similar scheme of preprocessing was adapted in the previous code.

To use the full range of the activation function, an attempt was made in which the average of the training dataset is deducted from all datapoints then scaling the resultants to be between -1 and 1 (normalization). This attempt suffered from convergence issues especially at the start of the FEM analyses, and hence it was not adopted.

It was noticed during running multiple SelfSim analyses, that training frequently was overfitting to the points with large magnitudes while giving a lesser priority to the points with smaller magnitudes. However, since the FEM analysis used previous stresses and strains, missing the small-strain range often resulted in missing the large-strain range. To overcome this, the datasets were balanced by adding more copies of the datasets containing the small magnitude points. This forced the training to give a similar importance to small and large magnitude points, as now the error of the small-magnitude points is multiplied by the number of copies. This was preferred over eliminating some of the large-magnitude points which would lead to loss of data.

The algorithm for adding the copies is:

- Calculate the average of the absolute magnitudes of stresses in each step.
- Divide the averages from all the steps by the maximum average. Round the numbers to the nearest integers. Those integers represent the number of copies of data of this step that should appear in the final dataset.
- If a big difference exists between the largest average and smallest average, the number of copies required for the step with the smallest average might be big, which would increase the training time. Set the maximum number of copies to a user-defined cap. The cap on the number of copies used in this study is 5.

8.3.2 Selection of Hyperparameters

A uniform architecture of 18x30x30x6 was used for the ANN in all tests. For all analyses, the following hyperparameters were chosen:

- The optimizer used was the ADAM optimizer. This selection is based on many studies that showed superior performance of this optimizer compared to other optimizers including (Ruder 2016).
- The loss function used was the mean absolute error "MAE". This selection was made to prevent training from being dominated by points of high magnitudes while giving less importance to points of lower magnitude. This is especially important in this application, because the behavior is history dependent, so deviations at low stresses might accumulate leading to big deviations at large stresses.
- Batch training was used (so the weights are updated once after calculating the error from all the training dataset examples). A maximum number of epochs of 50 to 200 was used. The batch training is used here benefitting from the relatively small size of the training datasets and the low number of epochs typically required. If a bigger dataset was encountered, the code, the batch size could be reduced switching to a mini-batch training.
- The ANN used were fully connected (no nesting was incorporated). This was based on testing both nested and fully connected networks. Both types of networks were yielding similar final results, however, the fully connected networks were converging much faster, which enables the completion of the analyses in fewer passes as described next. While the purpose of adding the path points to a nested neural network rather than a fully connected network, is to allow the error to be reduced as much as possible by the major input (the current strains) before allowing the path points to further reduce the error, it can be argued that the neural network can detect those relationships more efficiently if the path points data is not collinear with the current point data. This was evident in the calculation of a stiffness matrix that matched very closely with the theoretical stiffness matrix of the linear elastic model as shown in section 7.2.1.
- The (Glorot and Bengio 2010) uniform weight initialization scheme was used. This selection is based on many studies that showed that random weight initialization can slow or prevent the convergence of training, while using optimized initializers would accelerate training. The type selected is the one recommended by Keras, which is an open-source library that runs on top of TensorFlow and many other neural network libraries, and whose syntax is becoming the mainstream in the neural network industry.
- No biases were used in any of the ANNs.

8.3.3 Selection of SelfSim Parameters

For SelfSim, the following parameters were selected:

- The number of cycles used per step was capped at 3. Many steps would not require 3 cycles, because the displacement error drops below the set tolerances, and hence the analysis would move to the next step without reaching the cycle number cap.
- The displacement tolerances used to determine the success of the SelfSim analysis and to stop training in a certain step and moving to the next step without consuming all the set cycles are the following:
    - $Max\ Normalized\ Displacement\ Error = \frac{\max(\delta_i)}{\max(\widehat{U_i})} < 0.05$
    - $Average\ Normalized\ Displacement\ Error = \frac{\text{average}(\delta_i)}{\max(\widehat{U_i})} < 0.025$

  where: $Displacement\ Error\ \delta_i = Measured\ Displacement\ (\widehat{U_i}) - Calculated\ Displacement\ (U_i)$
- While many modes of operation of SelfSim were implemented in SelfSim, the mode of operation 3 is the one used for the triaxial test example. The modes of operation available are:
    - Mode 1: run all steps then train, then rerun. This turned out to be capable of very quickly learning simple behavior such as the linear elastic behavior which doesn't change with the level of strain
    - Mode 3: runs a step for a number of cycles then moved to the next step without repeating previous steps. This saves on the time consumed by restarting the analysis from the beginning while preventing the data of the previous points from being updated by values that may not satisfy the displacement tolerances of those steps
    - Mode 4: runs a step for a number of cycles then restarts the analysis from the beginning before moving to the next step. The advantage of this mode is that all the data in the dataset results from a single ANN, but its main disadvantage is that when rerunning the analysis from the beginning, the tight tolerances set during training might be exceeded, especially if the training was dominated by the latest added points.

While training, it is necessary to make sure that the ANN model is not overfitting to the training data, which might get it stuck in local minima, preventing it from learning new behavior, or might limit its extrapolation capability. To avoid overfitting, an early-stopping scheme was used, in which the training dataset is divided into two parts: the training set containing 98% of the points in the dataset and the validation dataset containing 2% of points in the dataset. The dataset should be randomly shuffled before splitting it into those two parts. The training dataset is used for training while the validation dataset is not included in the training. While training, the error of both sets is calculated. When the error of the training dataset doesn't change much over many epochs, or when the error of the training set drops while the error of the validation set increases over many epochs, overfitting is claimed to be happening, and the training in the current cycle is stopped. The SelfSim analysis continues to the next cycle or step.

After each SelfSim analysis was completed, a forward analysis was run using the trained ANN. This analysis was load-controlled: applying the load on the loading cap and the confining pressure on the curved boundary of the model. The results of those runs are presented in Figure 8-4 to Figure 8-39. The overall behavior is well captured by the trained ANN model.The number of passes required to achieve those results was 1 or 2 and the total run time for the SelfSim analyses varied between 6 and 15 minutes. The average time per pass was around 5 minutes. The number of passes and the run times for all tests are listed in Table 8-2. In addition, the stress space calculated by the current code was compared with that calculated using the previous code and presented in Figure 8-40 and Figure 8-51. The comparison shows that the stress spaces calculated by both codes are very similar.

Comparing the run times of the previous SelfSim code with the new code, it can be deduced that a significant speedup was gained with the optimizations applied in the new code which were described in CHAPTER 6. Some of the speedup is because of the use of a less-costly finite element, but the major speedup lies in the number of passes required to achieve the displacement tolerances set for SelfSim (5% maximum displacement error). Being able to train the ANN in 1 or 2 passes is a great improvement over the 8 passes required in the previous study.

8.4 Load-Driven vs Displacement-Driven Analyses

As mentioned, the results shown in Figure 8-4 to Figure 8-39 are for load-controlled forward analyses that employ the trained ANN models. It was noticed that driving the analysis with a displacement control, where all boundary displacements are applied and no load or confining

pressure is applied, instead of the load control was yielding sometimes different loads compared to the load-controlled analysis whose loads match those applied in the lab tests. This issue is still under study. Possible reasons include overfitting to one or more steps data which may cause bias in the stresses of other steps. The comparison between the load-controlled and displacement-controlled analyses for the MT1 test are shown in Figure 8-52 to Figure 8-54. Figure 8-52 shows the differences in the calculated loads, while Figure 8-53 and Figure 8-54 show the similarity of the deformations and stress-spaces of both models. This behavior does not show in other tests such as the CS3 test whose comparisons are shown in Figure 8-55 to Figure 8-57.

8.5 Generalization Capability of the Trained ANN Models

The generalization capability of the trained ANN models was again tested, but this time using the ANN models trained through SelfSim for the laboratory tests. The ANN trained through the SelfSim analysis on the MT3 test is used in a forward FEM analysis of the CT3 model, while the ANN trained through the SelfSim analysis on the CT3 test is used in a forward FEM analysis of the MT3 model. The results of those runs are shown in Figure 8-58 to Figure 8-61. As with the theoretical models, the generalization capability of the ANN trained with tests from the modified triaxial tests which are richer produced was better. Because of the complex soil behavior of real soil and possible measurement errors, the generalization quality was less than that of the theoretical models.

8.6 Attempts for Training a Unified ANN Model:

An attempt was made to train a single ANN that can learn the behavior of sand in all 12 tests. This was tried via 2 methods:

1- Compiling the learned strain-stress datasets then training a single ANN.
2- Running simultaneous SelfSim analyses

The first method didn't yield good matches. The second method on the other hand resulted in acceptable deformations for the 12 tests. Since the confining stresses varied from test to test, the stresses passed and calculated from the ANN had to be normalized by the confining stress. The confining stress was also passed as an input to the ANN. The architecture used is demonstrated in Figure 8-62. The deformations calculated by the unified ANN model for the 1 to 1 tests are shown in Figure 8-63. It is noteworthy that while the deformations calculated by the unified ANN model

were acceptable, the loads calculated had overall bigger differences. The development of a unified model should be further explored in future research.

8.7 Findings and Recommendations

The new code was used successfully to learn behavior of sand tested through conventional and modified laboratory testing. The speedup gain of the new code compared with the old code varied between 500 and 1500 times approximately. This allows for much faster testing and experimenting with the SelfSim method.

An attempt was made to develop a unified ANN model for the 12 tests. The calculated deformations were close to the target deformations while the calculated loads showed bigger differences to target loads.

While the new code resulted in significant speedups, it also posed additional questions. It pointed to the possibility of finding multiple slightly varying solutions for each problem. In addition, it showed that the displacement-controlled and the load-controlled analyses do not necessarily perfectly converge. It also extended the capability of the SelfSim framework by giving it access to additional optimizers, initializations schemes, modes of operation, etc, whose effect need to be studied.

In addition to the above questions, a deeper understanding of the ANN material model is needed. Great improvements to the inverse algorithm and to the training can be achieved if the ANN becomes more interpretable. For example, the significance of certain weights in the network, or their effect on the calculation of the stiffness matrix can direct the focus on these connections and suggest methods to deal with poor datasets that do not contain enough data for a general identification of the relationships between the inputs and the outputs and might hence not be enough for a good calculation of stiffness matrices. This understanding will also enhance the generalization capability of the ANNs by guiding the training and avoiding overfitting.

These questions, in addition to the development of a better unified model should be the subject of future research.

8.8 Tables

Table 8-1: Summary of the tests conducted on Ottawa Sand in (Asmar 2017)

Test ID	Test Type		Sample initial Size Diameter x Height (cm)	Initial Void Ratio (e_0)	Relative Density D_r (%)	Vertical Consolidation Pressure (kPa)	K_0
MS1	CK_0DMTXC-1:1		5.13x5.08	0.637	42	84.84	0.4
MS2			5.13x5.11	0.635	43	252.41	0.4
MS3			5.13x5.08	0.628	45	689.92	0.37
CS1	CK_0DTXC-1:1		5.11x5.08	0.623	47	88.23	0.43
CS2			5.11x5.08	0.627	46	247.87	0.38
CS3			5.11x5.08	0.631	44	700.06	0.4
MT1	CK_0DMTXC-2:1		5.11x10.16	0.628	45	100.67	0.42
MT2			5.11x10.16	0.621	48	250.65	0.37
MT3			5.11x10.16	0.631	44	688.91	0.33
CT1	CK_0DTXC-2:1		5.11x10.16	0.626	46	88.00	0.4
CT2			5.11x10.16	0.621	48	250.15	0.35
CT3			5.11x10.16	0.621	48	685.52	0.37

CK_0DMTxC-1:1: K_0 consolidated drained triaxial compression test using modified triaxial device on (1:1) sample

CK_0DTxC-1:1: K_0 consolidated drained triaxial compression test using conventional triaxial device on (1:1) sample

CK_0DMTxC-2:1: K_0 consolidated drained triaxial compression test using modified triaxial device on (2:1) sample

CK_0DTxC-2:1: K_0 consolidated drained triaxial compression test using conventional triaxial device on (2:1) sample

Table 8-2: Time consumed by the SelfSim analyses

Test	MS1	MS2	MS3	CS1	CS2	CS3	MT1	MT2	MT3	CT1	CT2	CT3
Time (min)	13.5	12	12.5	15	14.5	15	14	15.5	14.5	13.5	7	6
Number of Passes	2	2	2	2	2	2	2	2	2	1	1	1
Time of last successful run using old code (min)	247	219	221	271	313	252	271	216	218	218	226	233
Speedup compared to old code	18	18	18	18	22	17	19	14	15	16	32	39
Speedup considering number of failed attempts of old code	500 to 1500 times											

8.9 Figures

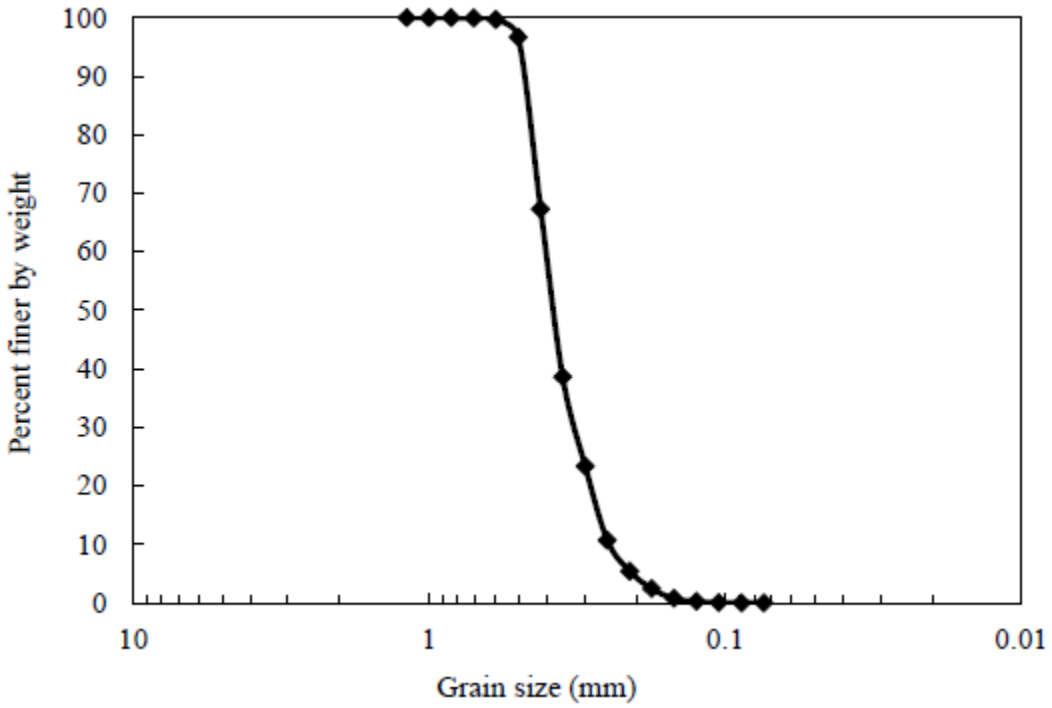


Figure 8-1: Particle size distribution for the Ottawa sand tested with the laboratory triaxial tests. Source: (Asmar 2017).

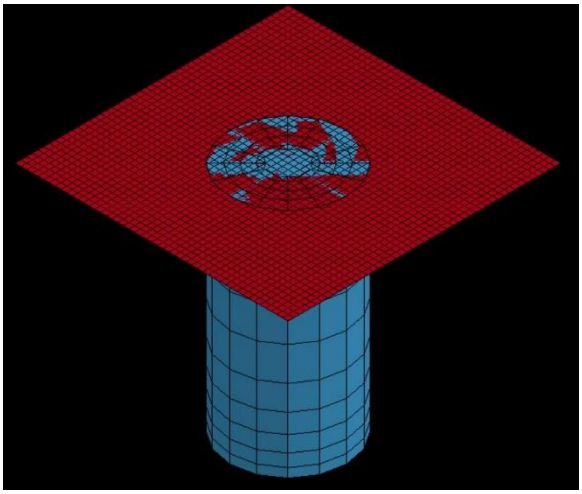


Figure 8-2: Photo of the 3D LS-DYNA model, showing the soil elements in blue and the rigid loading cap in red. The translation of the bottom boundary nodes are also constrained in all directions

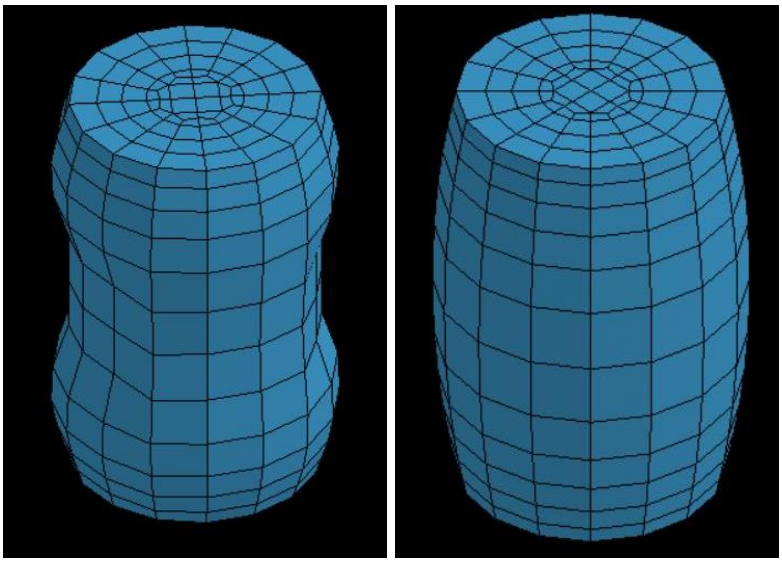


Figure 8-3: Photos of the deformed meshes of the 2 to 1 models: with clamps (left) and without clamps (right)

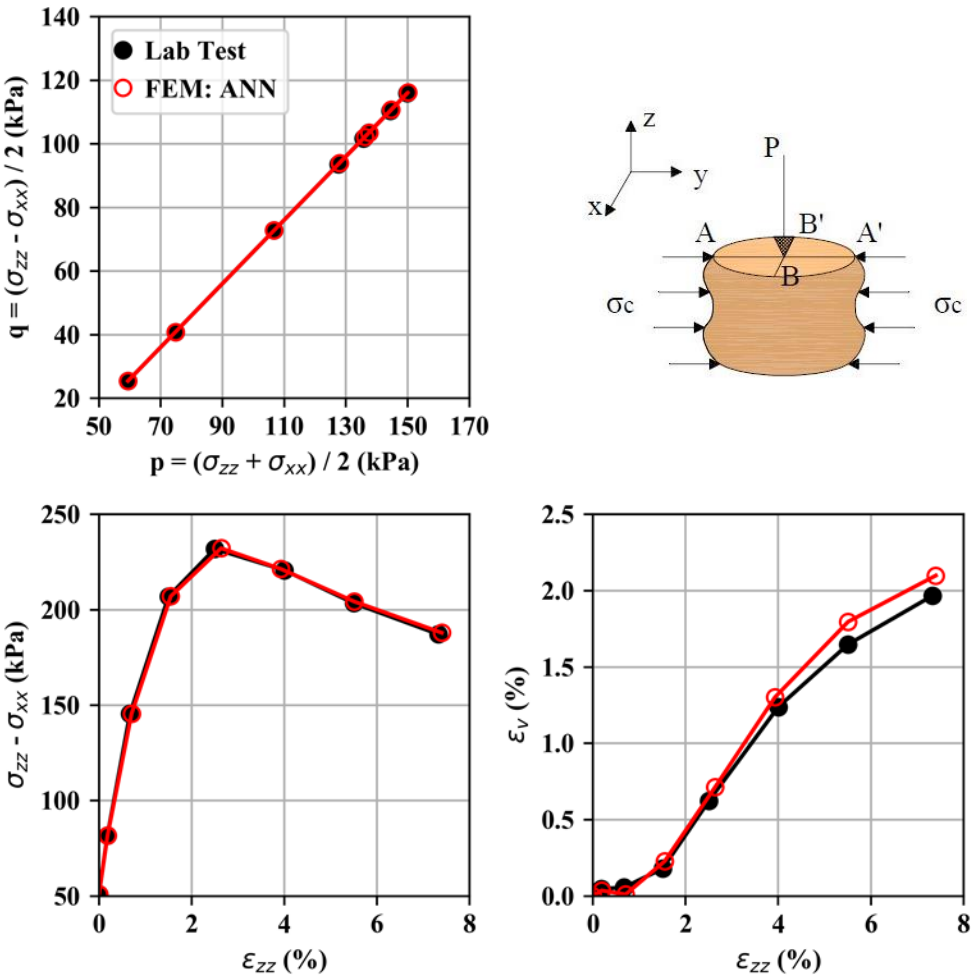


Figure 8-4: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the MS1 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

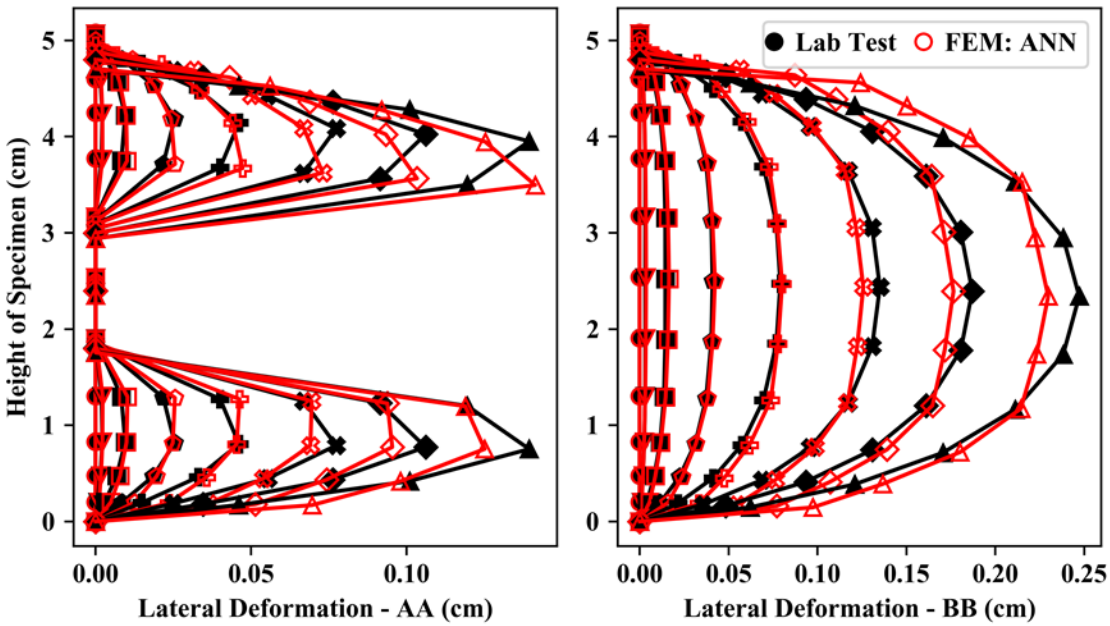


Figure 8-5: Comparison of the deformations measurements of the MS1 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

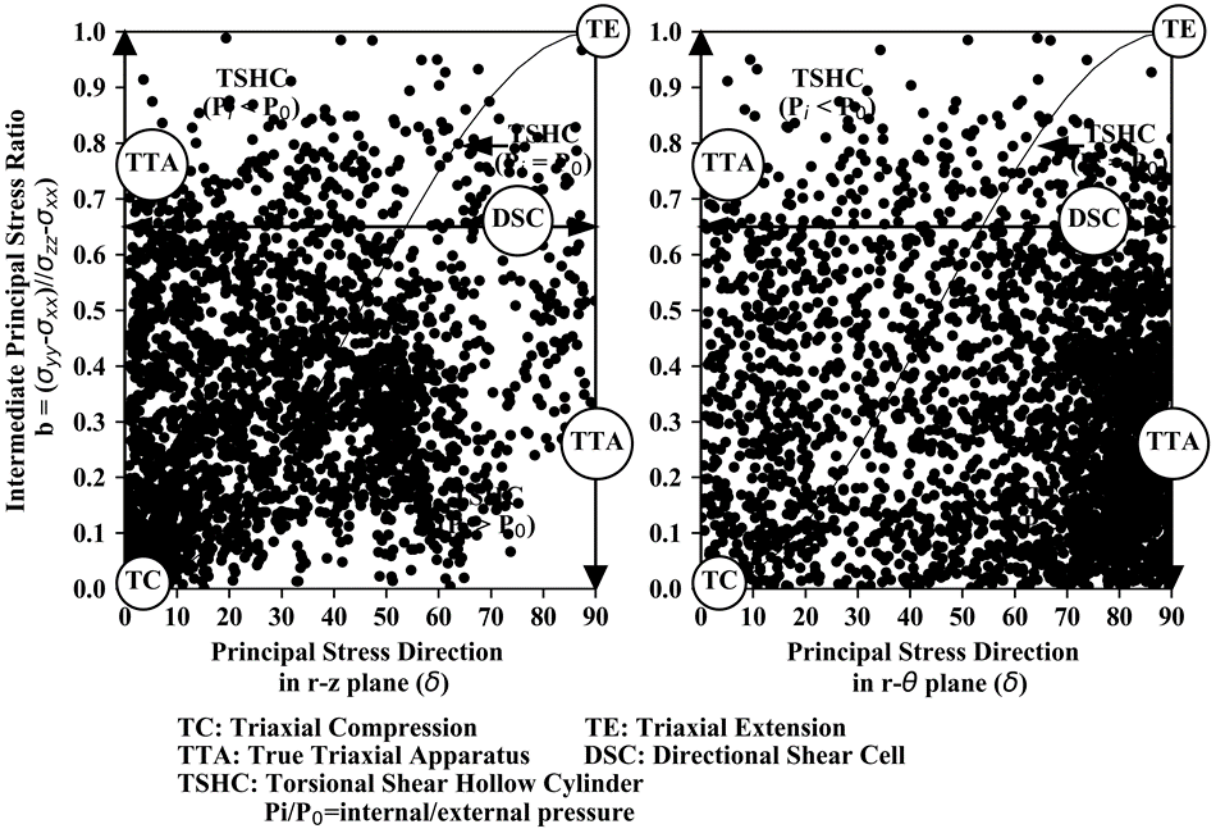


Figure 8-6: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the MS1 lab test

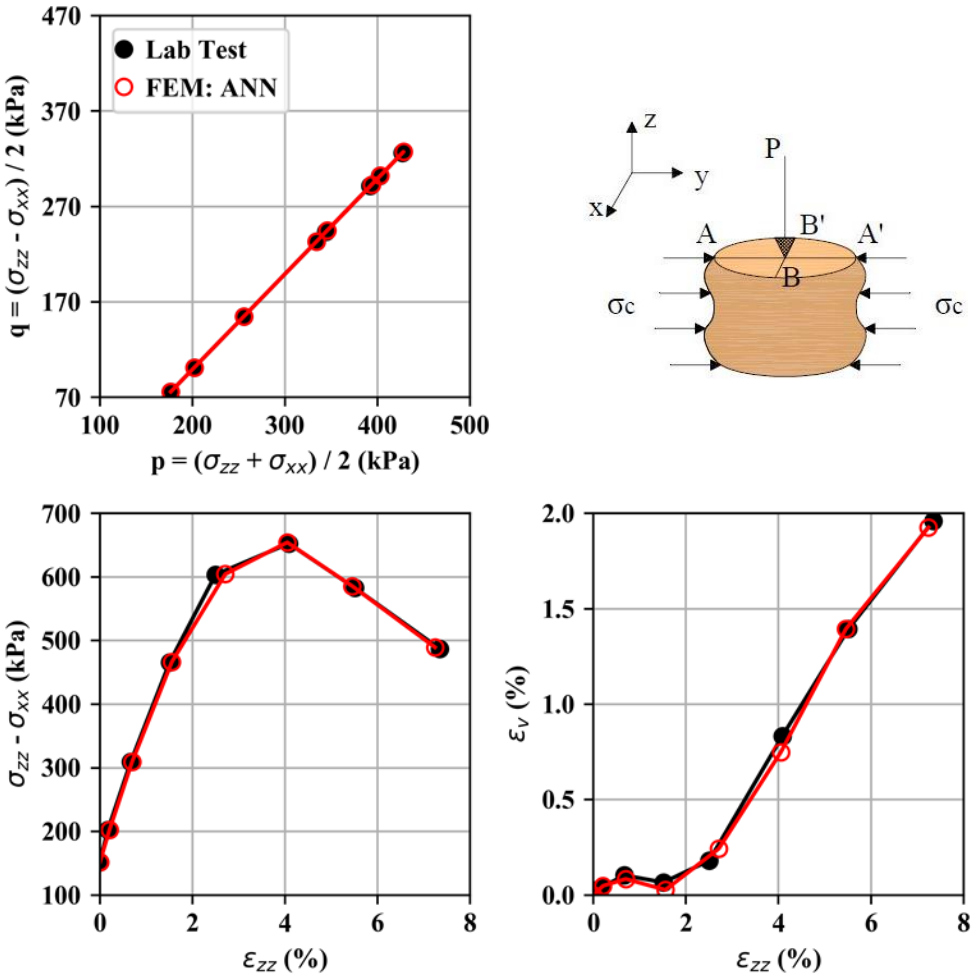


Figure 8-7: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the MS2 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

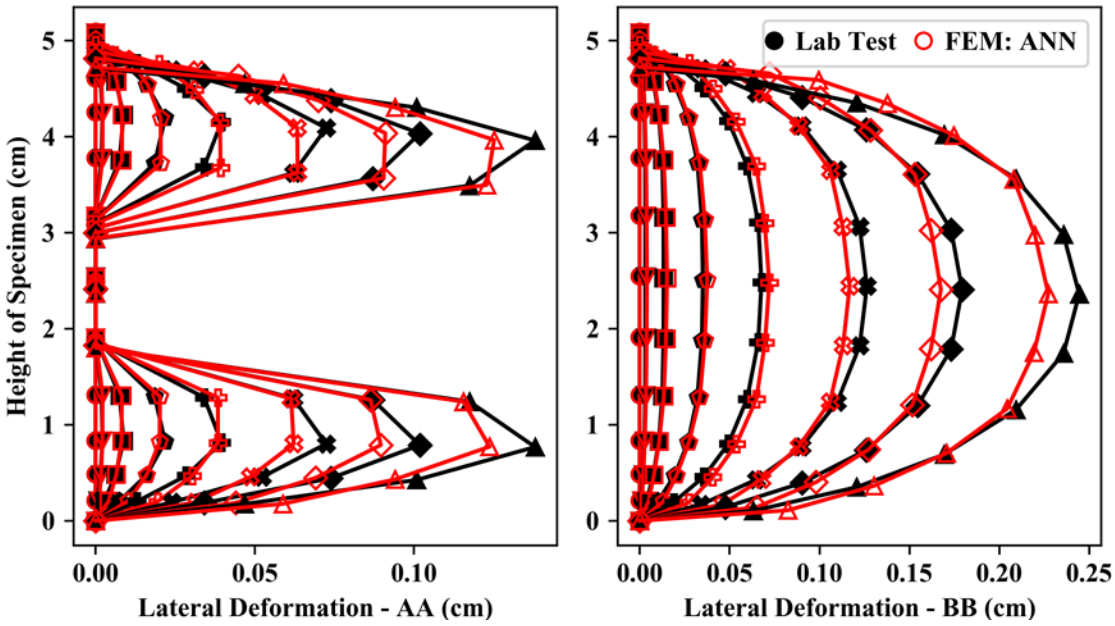


Figure 8-8: Comparison of the deformations measurements of the MS2 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

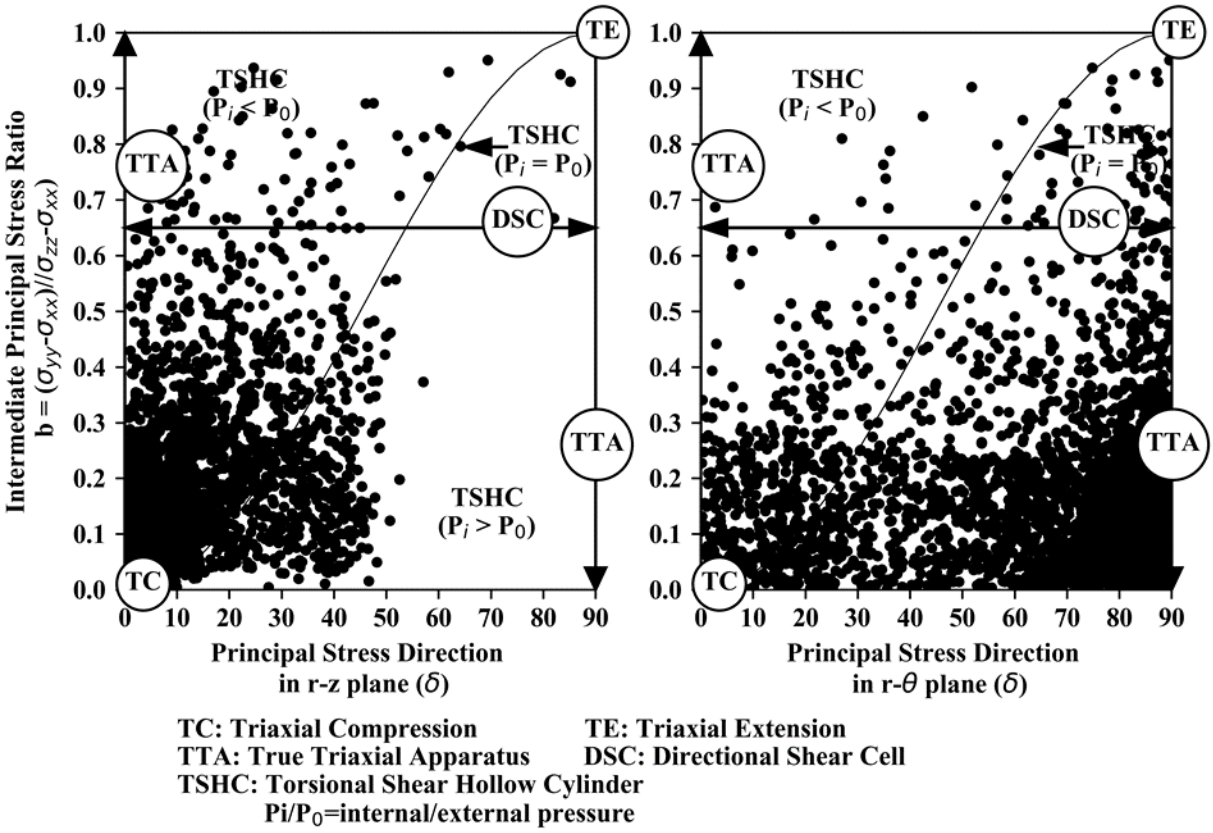


Figure 8-9: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the MS2 lab test

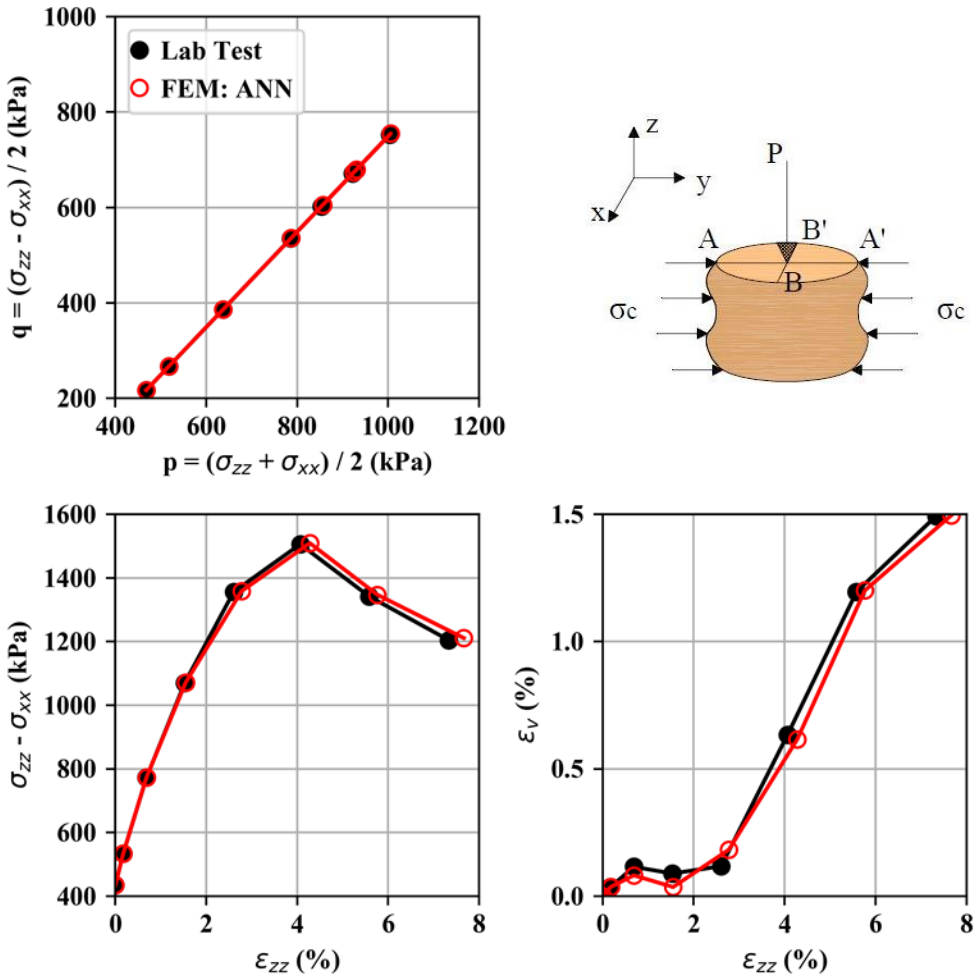


Figure 8-10: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the MS3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

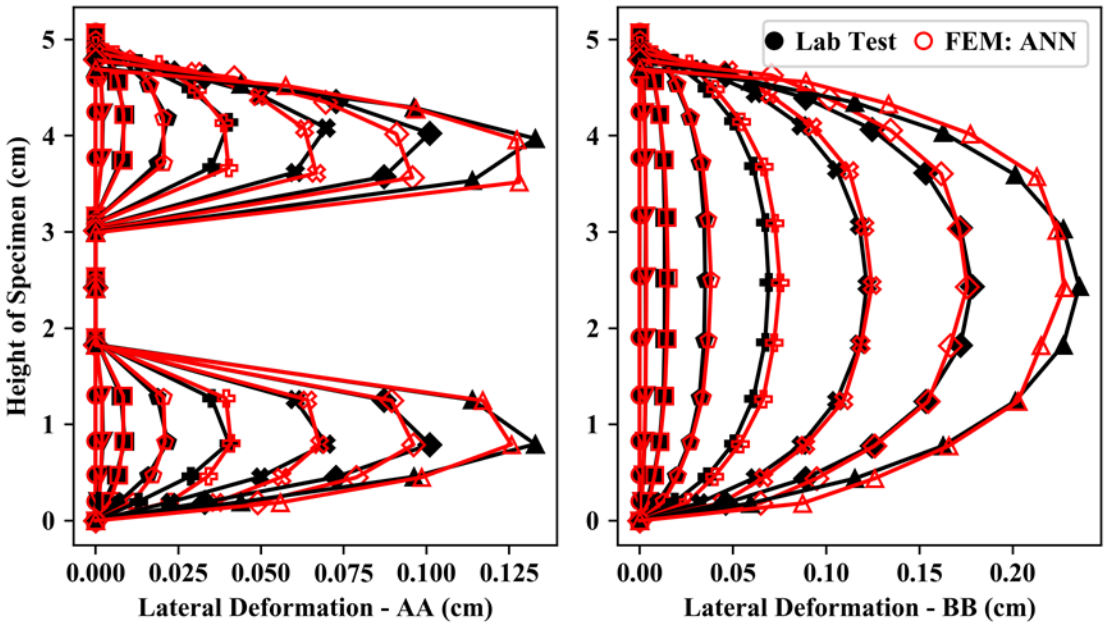


Figure 8-11: Comparison of the deformations measurements of the MS3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

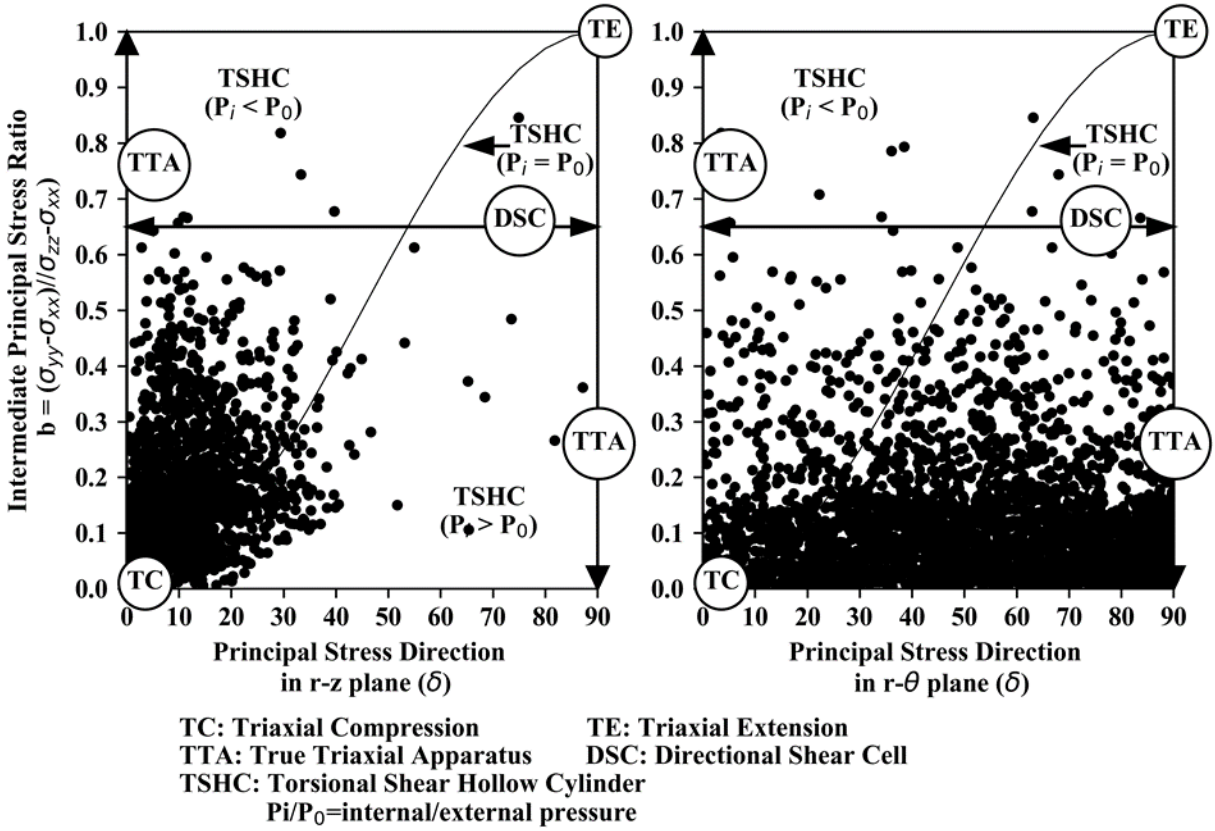


Figure 8-12: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the MS3 lab test

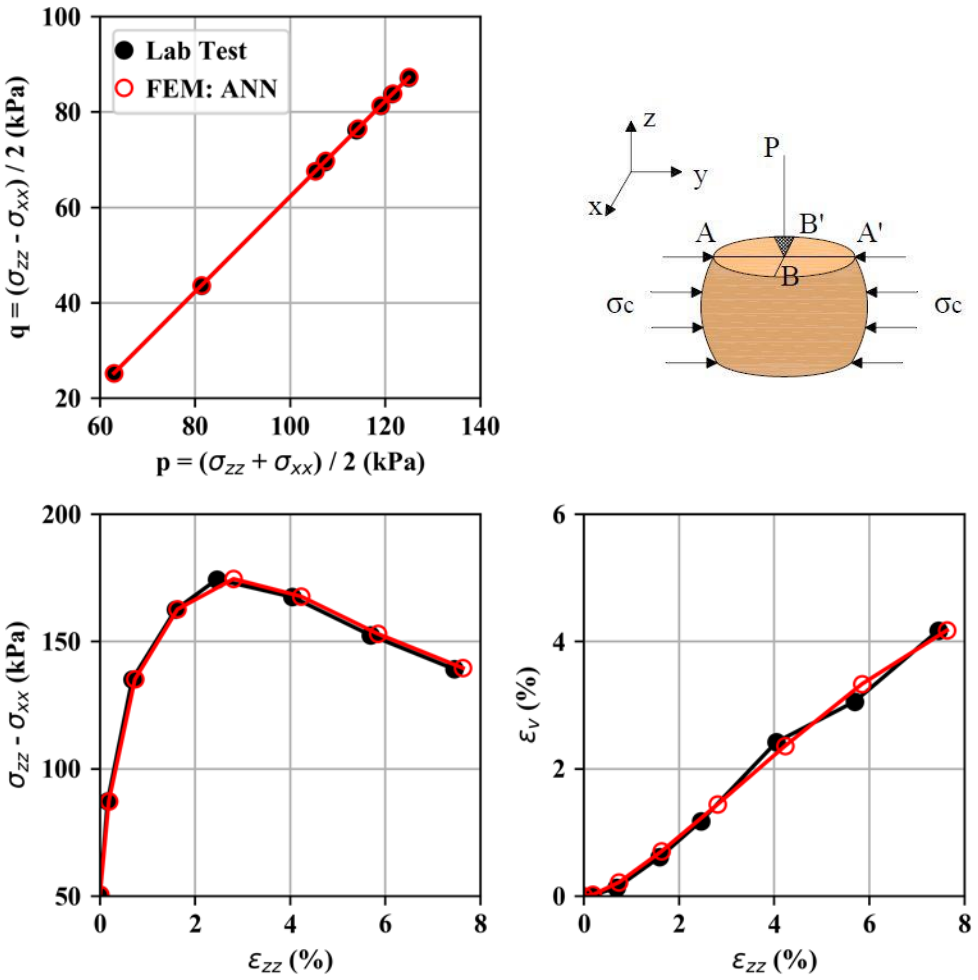


Figure 8-13: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the CS1 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

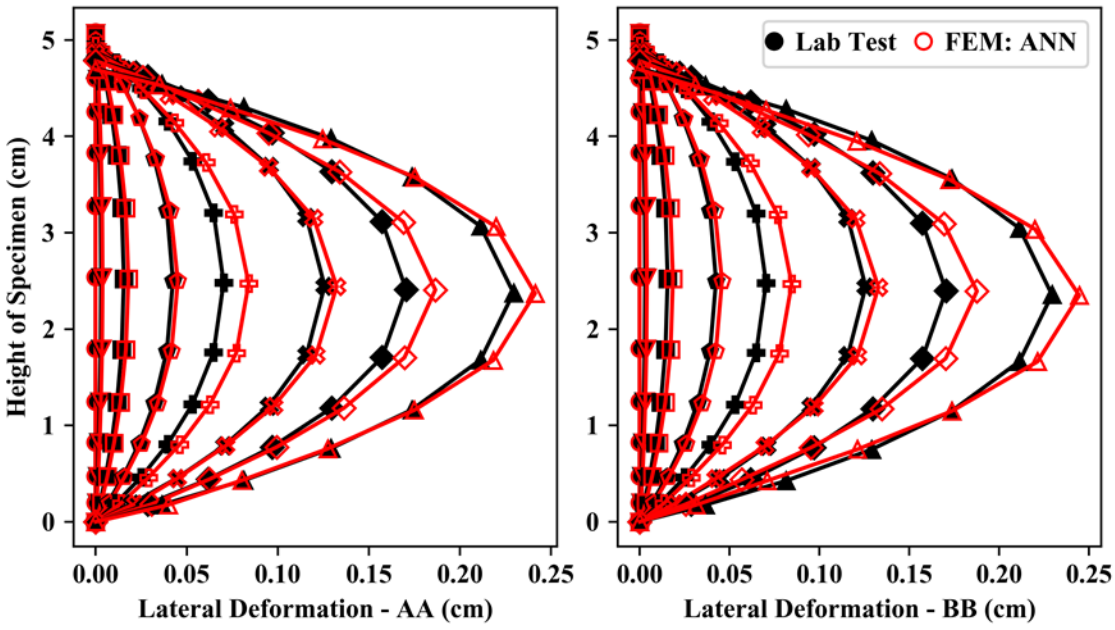


Figure 8-14: Comparison of the deformations measurements of the CS1 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

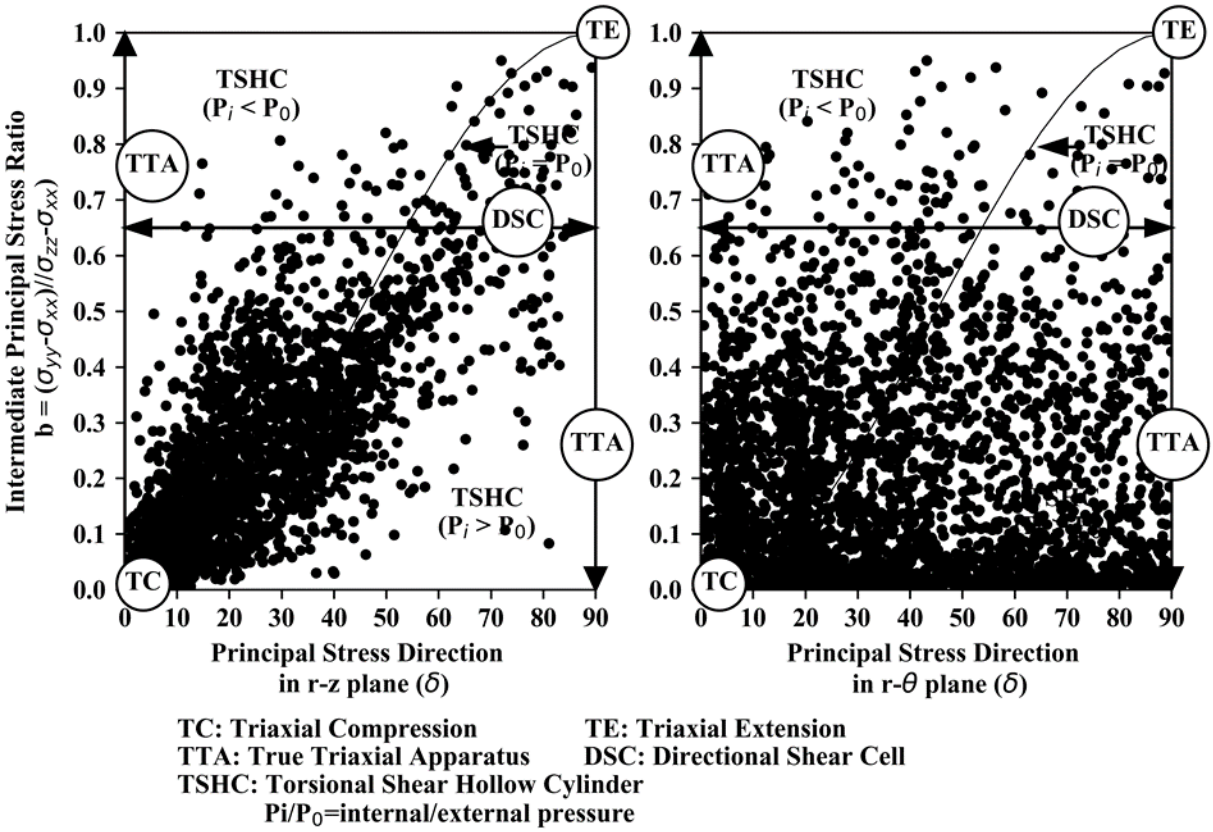


Figure 8-15: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the CS1 lab test

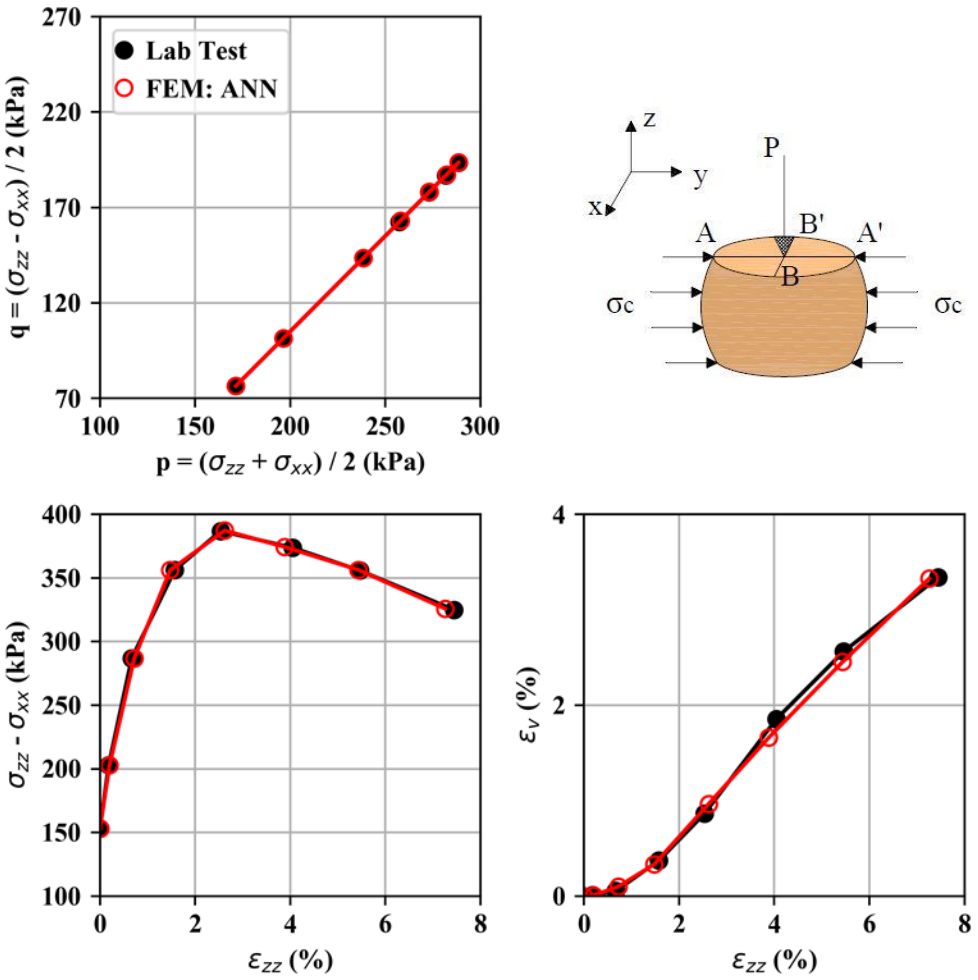


Figure 8-16: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the CS2 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

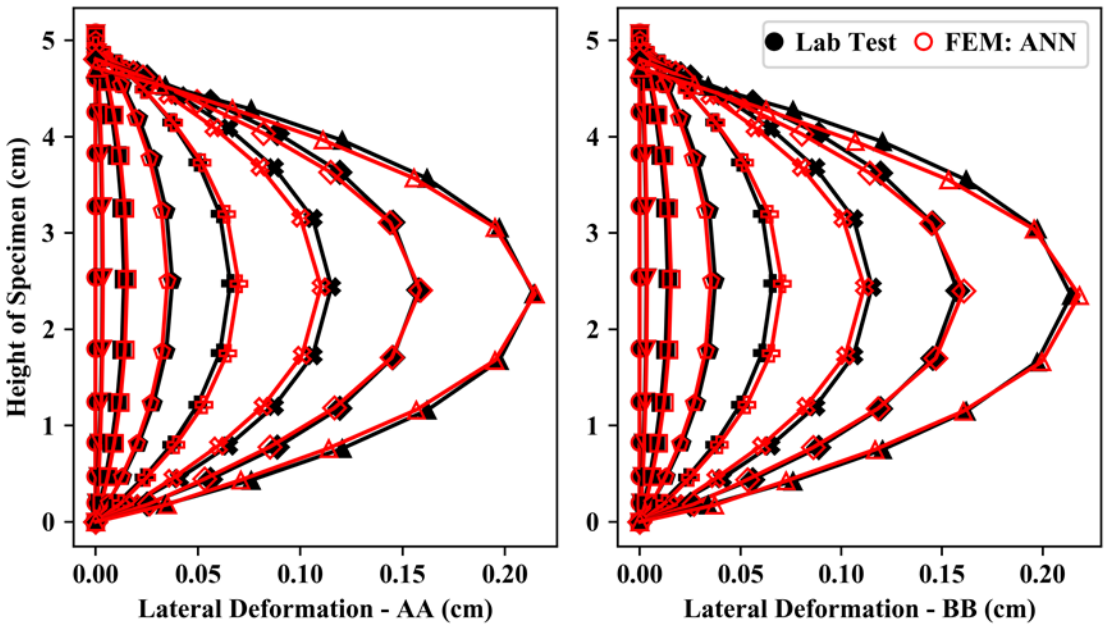


Figure 8-17: Comparison of the deformations measurements of the CS2 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

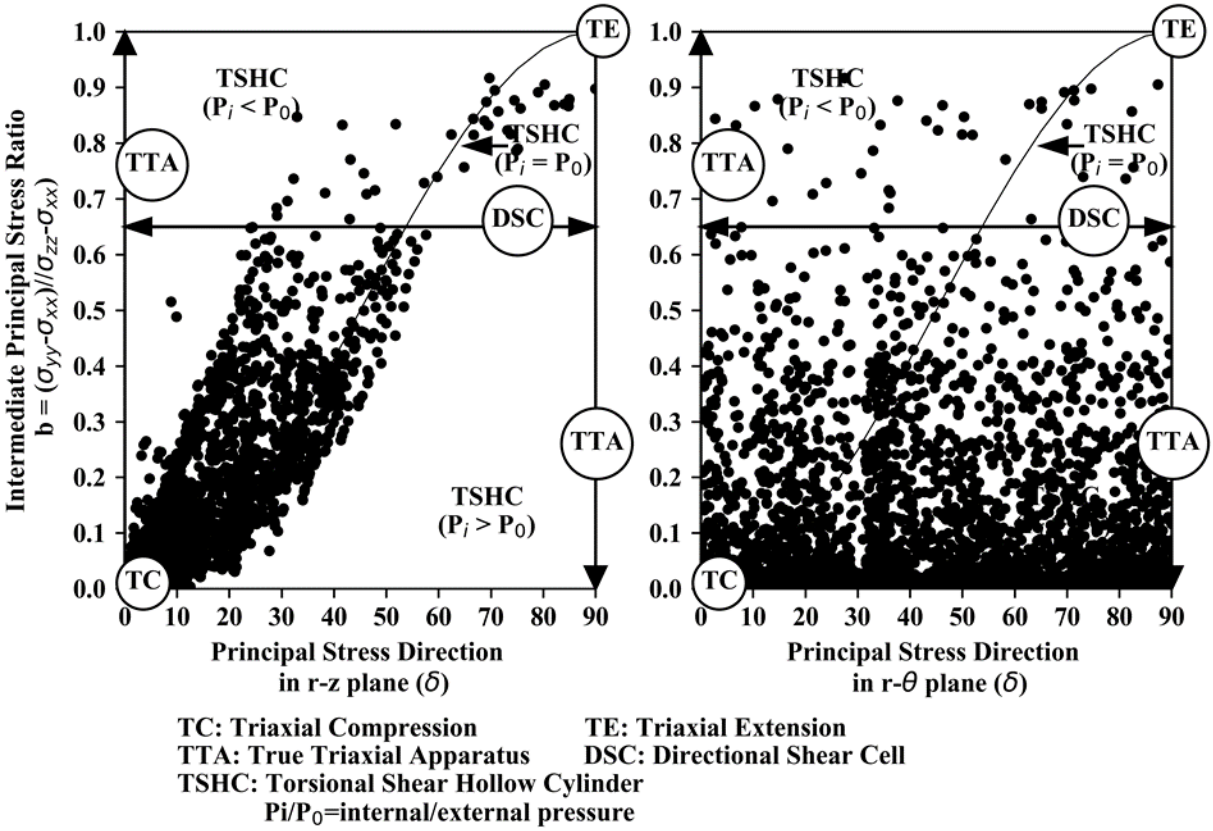


Figure 8-18: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the CS2 lab test

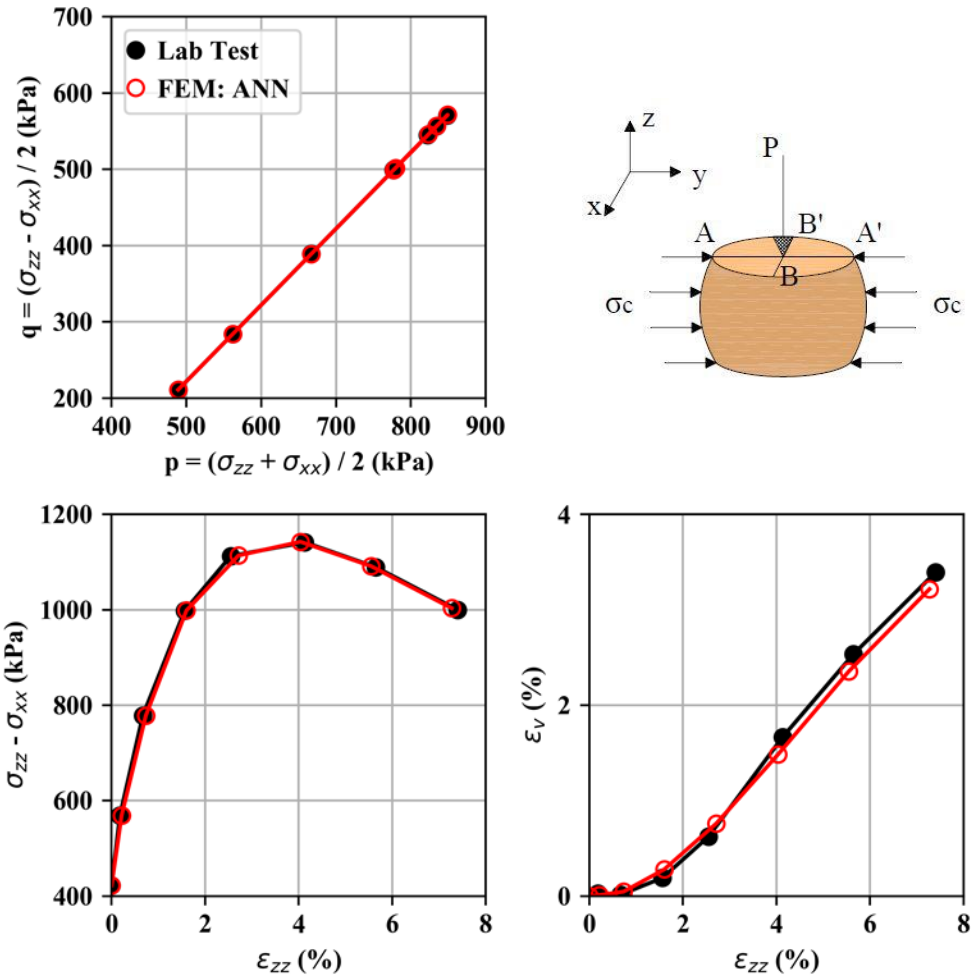


Figure 8-19: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the CS3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

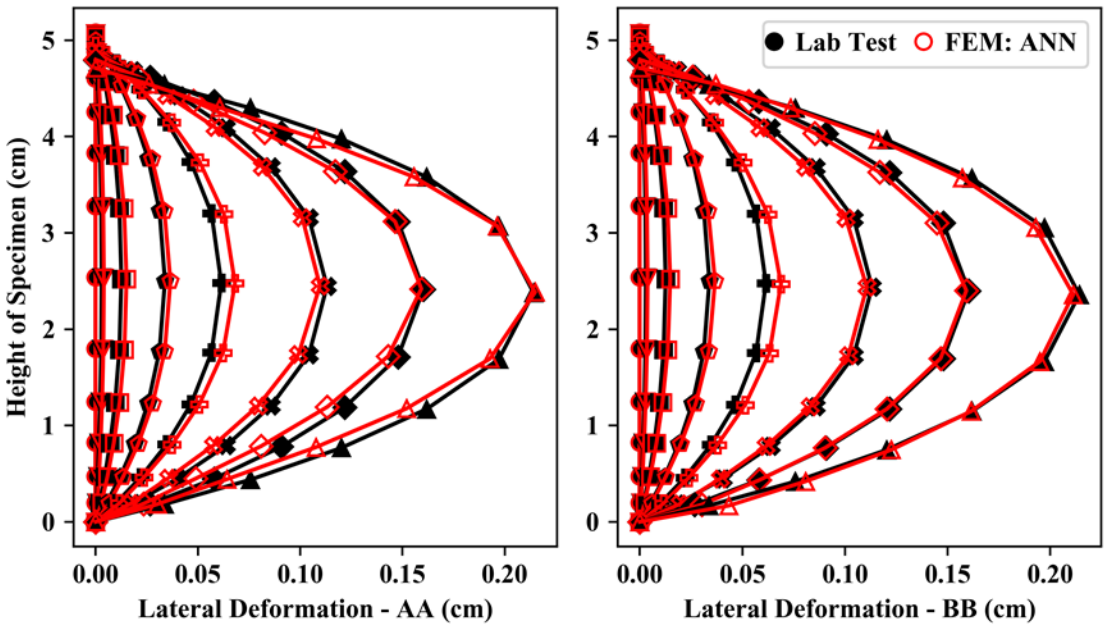


Figure 8-20: Comparison of the deformations measurements of the CS3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

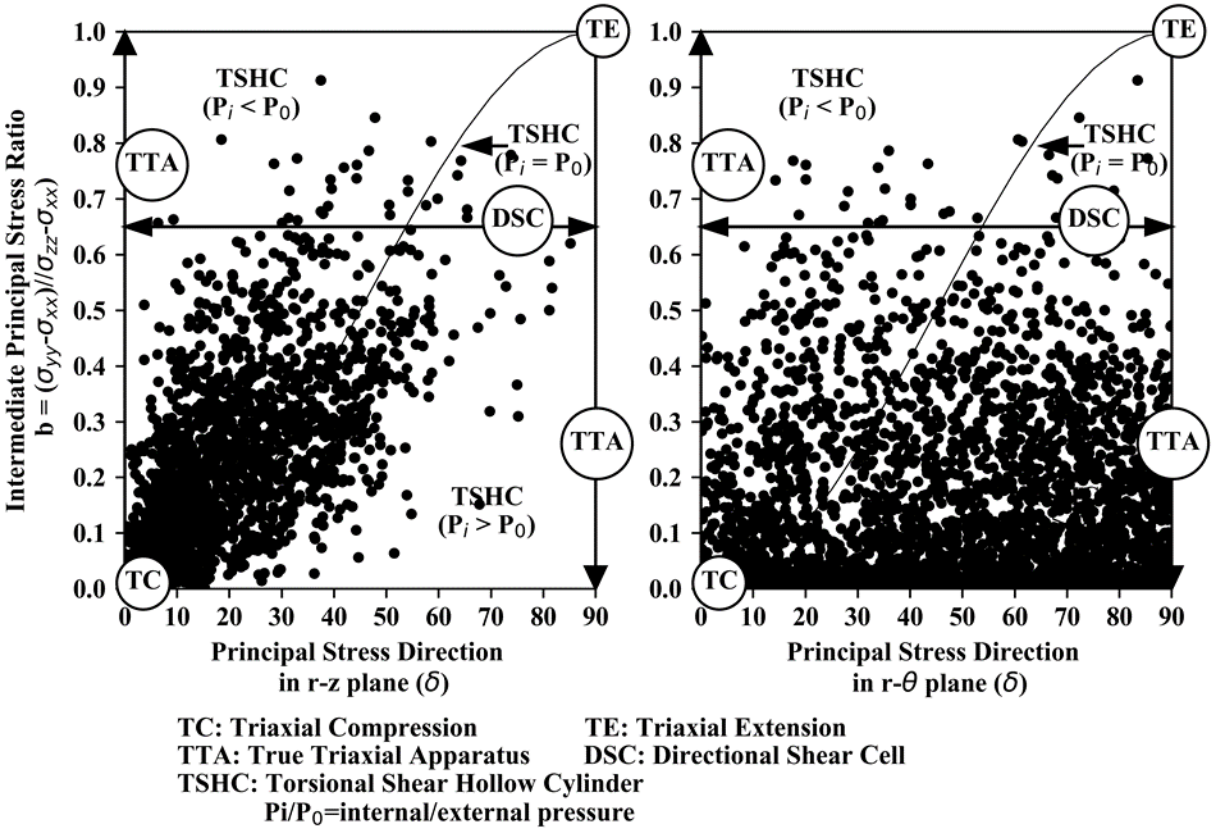


Figure 8-21: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the CS3 lab test

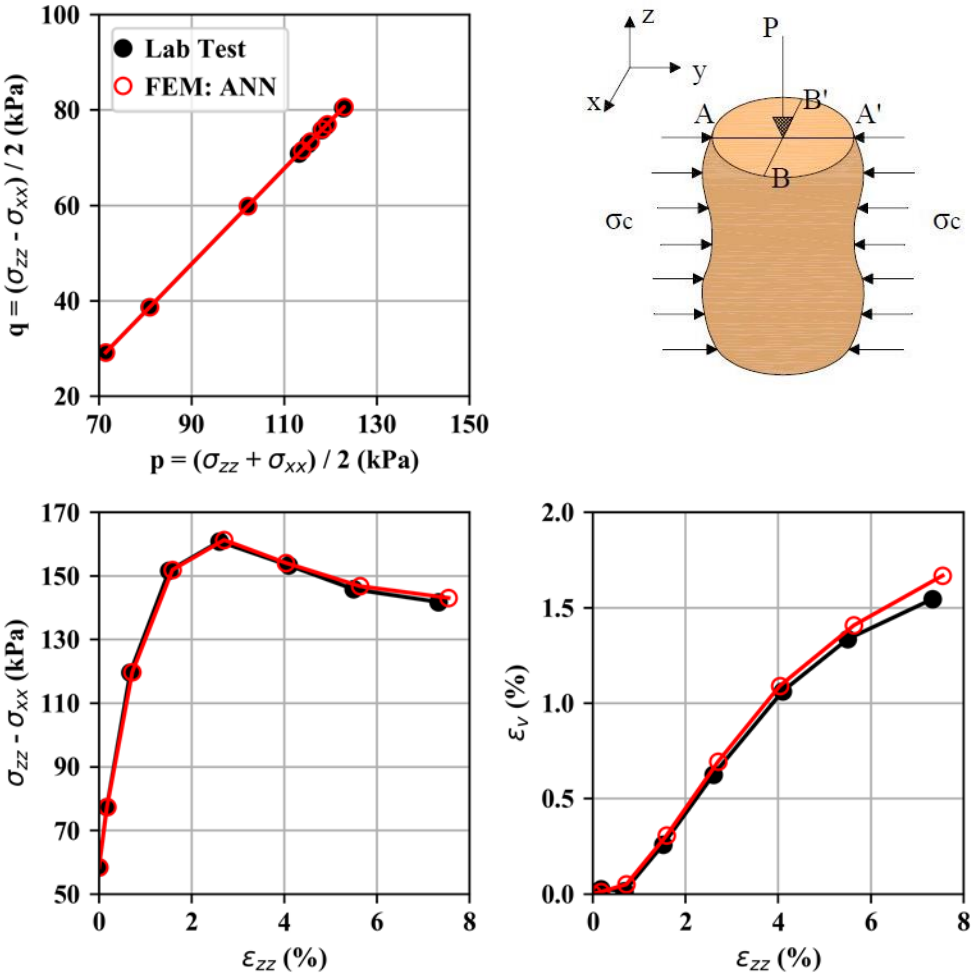


Figure 8-22: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the MT1 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

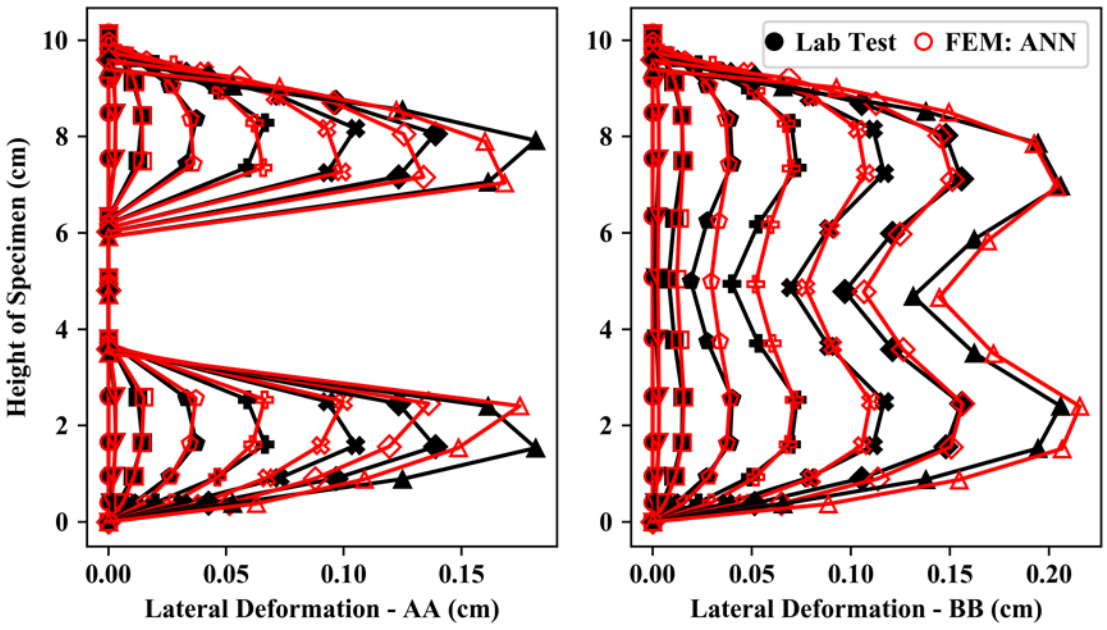


Figure 8-23: Comparison of the deformations measurements of the MT1 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

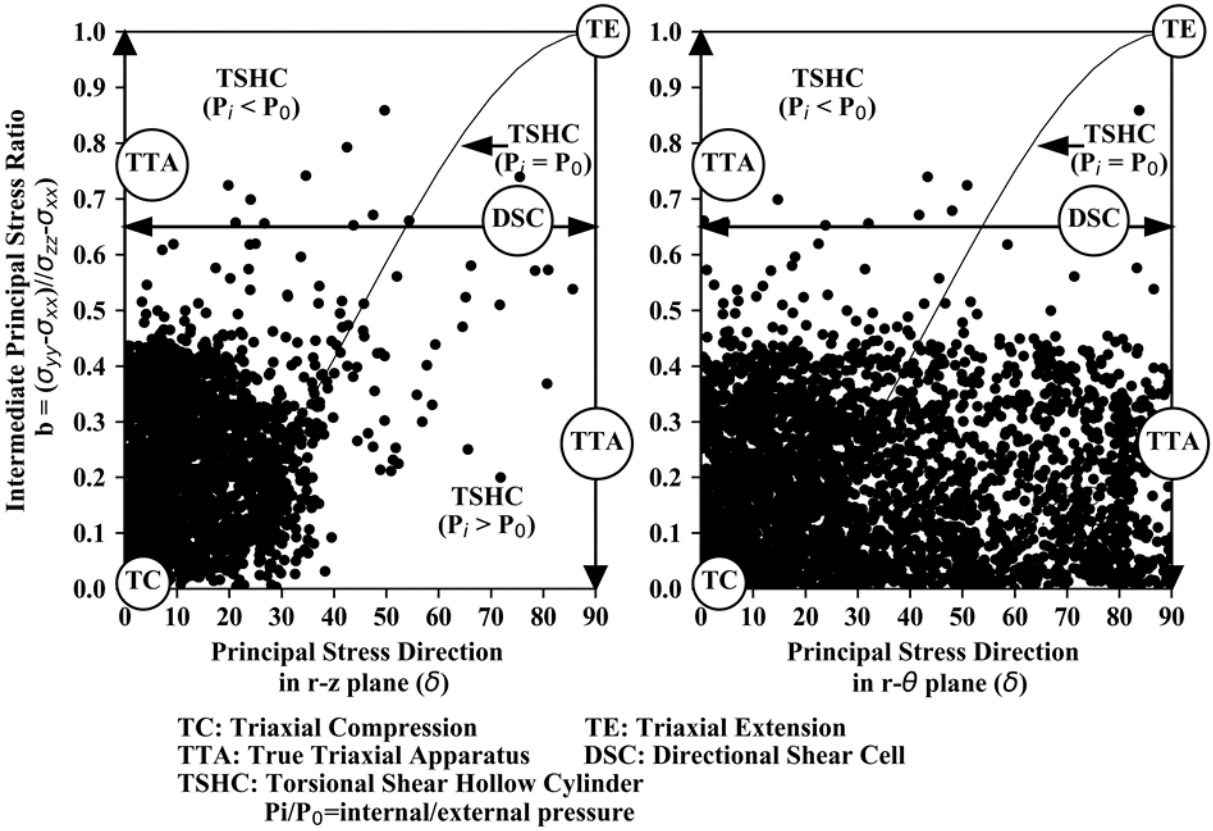


Figure 8-24: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the MT1 lab test

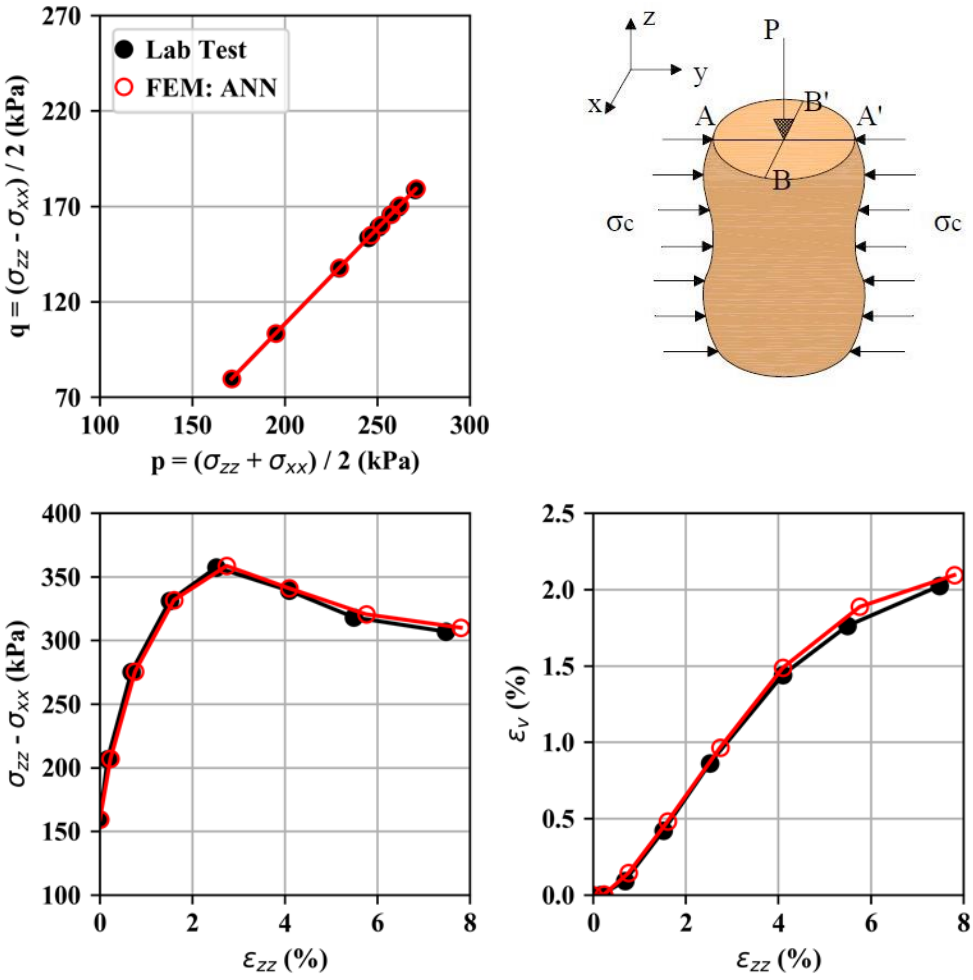


Figure 8-25: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the MT2 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

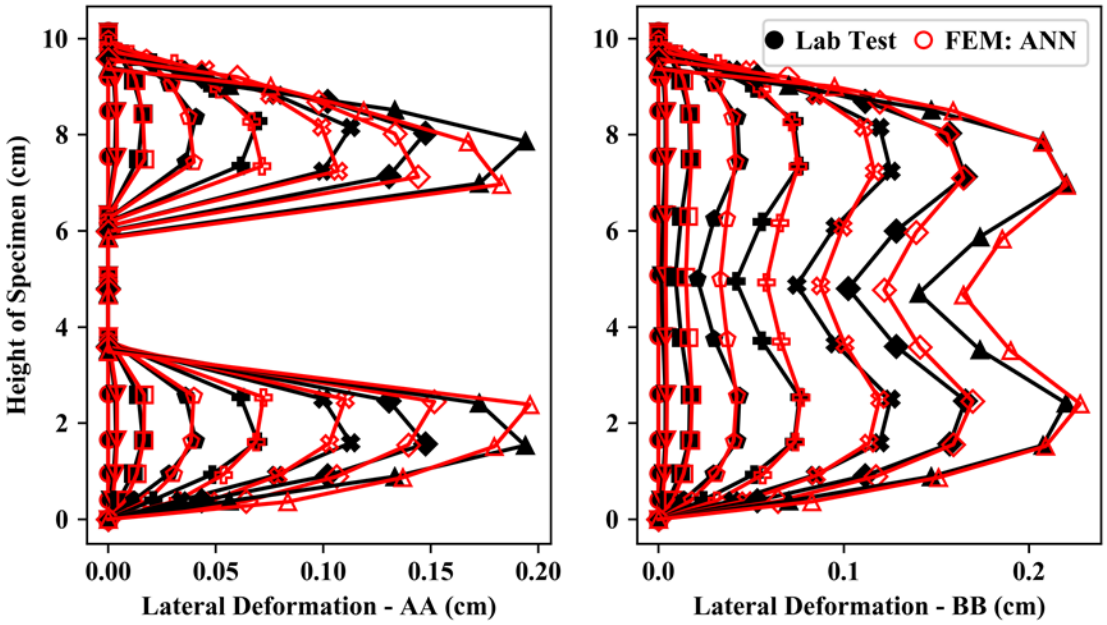


Figure 8-26: Comparison of the deformations measurements of the MT2 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

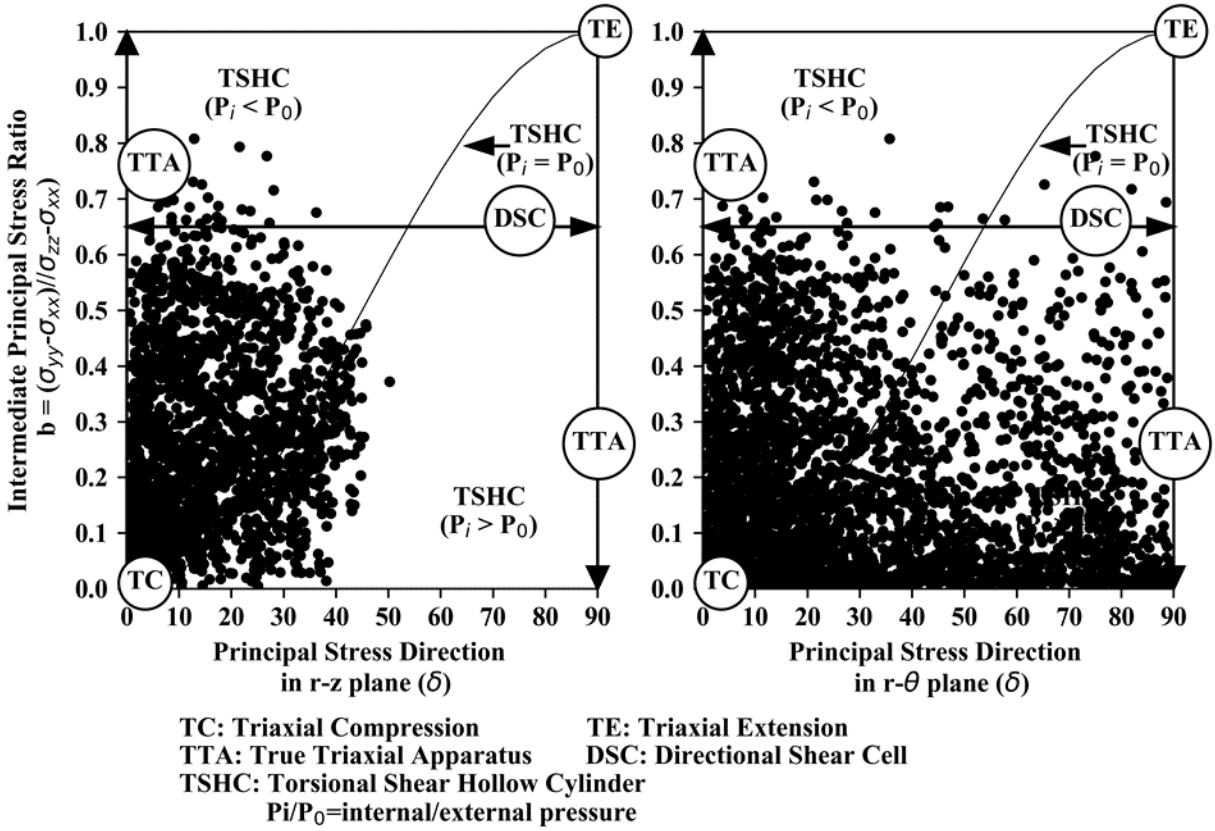


Figure 8-27: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the MT2 lab test

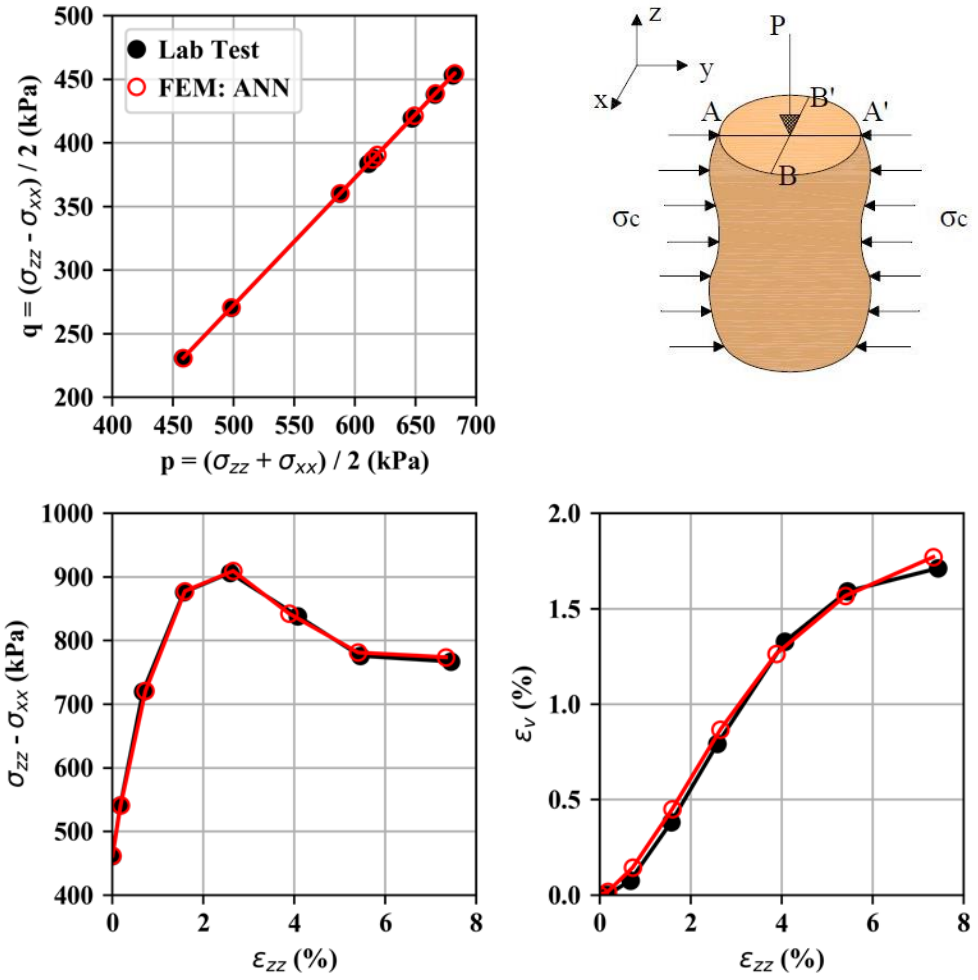


Figure 8-28: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the MT3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

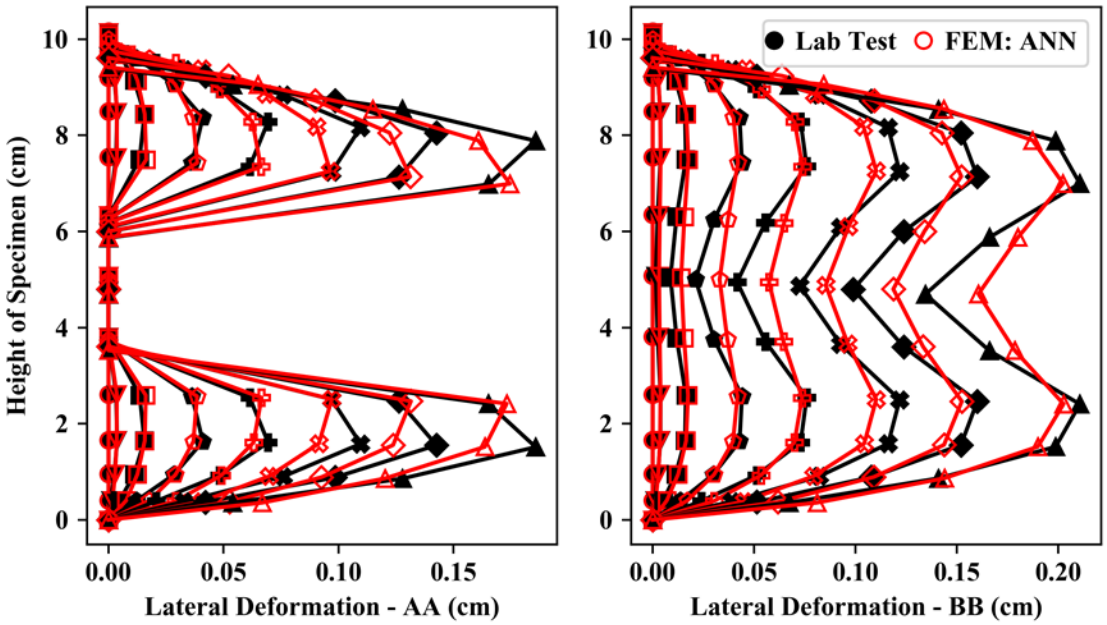


Figure 8-29: Comparison of the deformations measurements of the MT3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

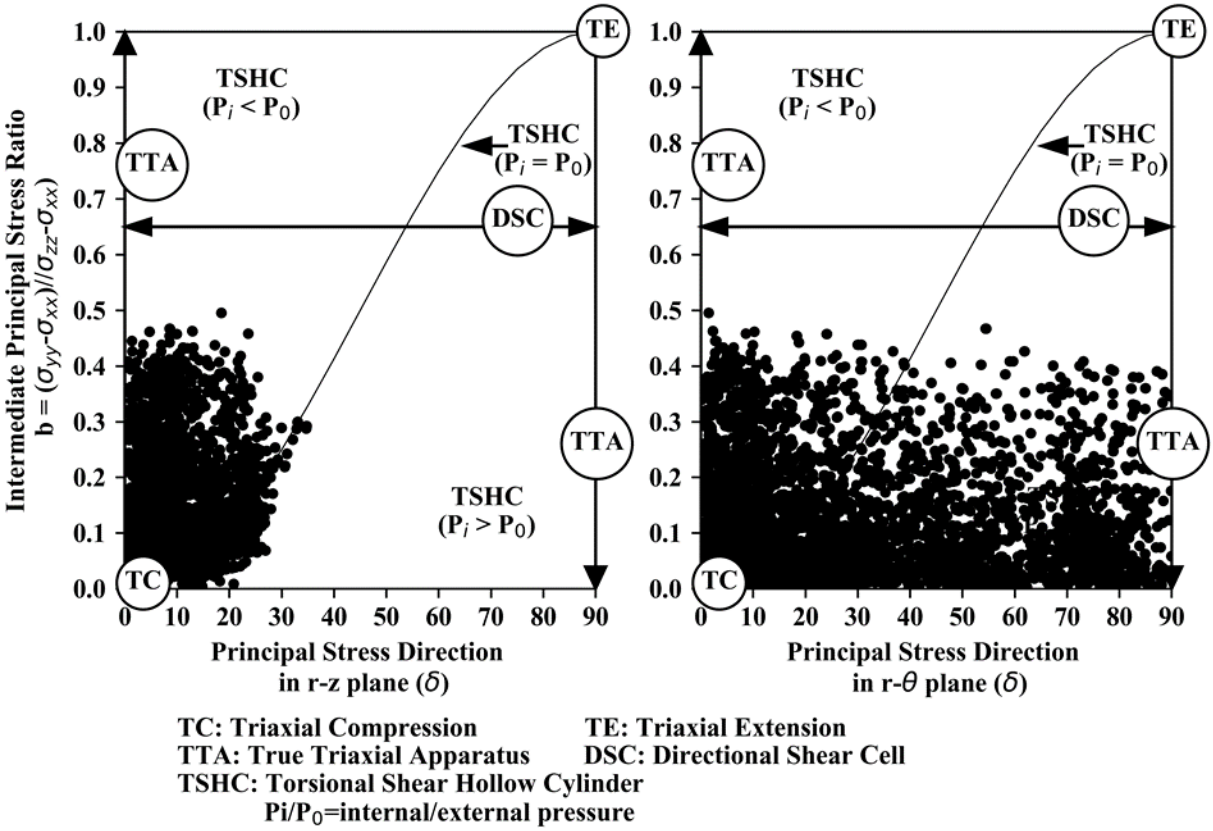


Figure 8-30: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the MT3 lab test

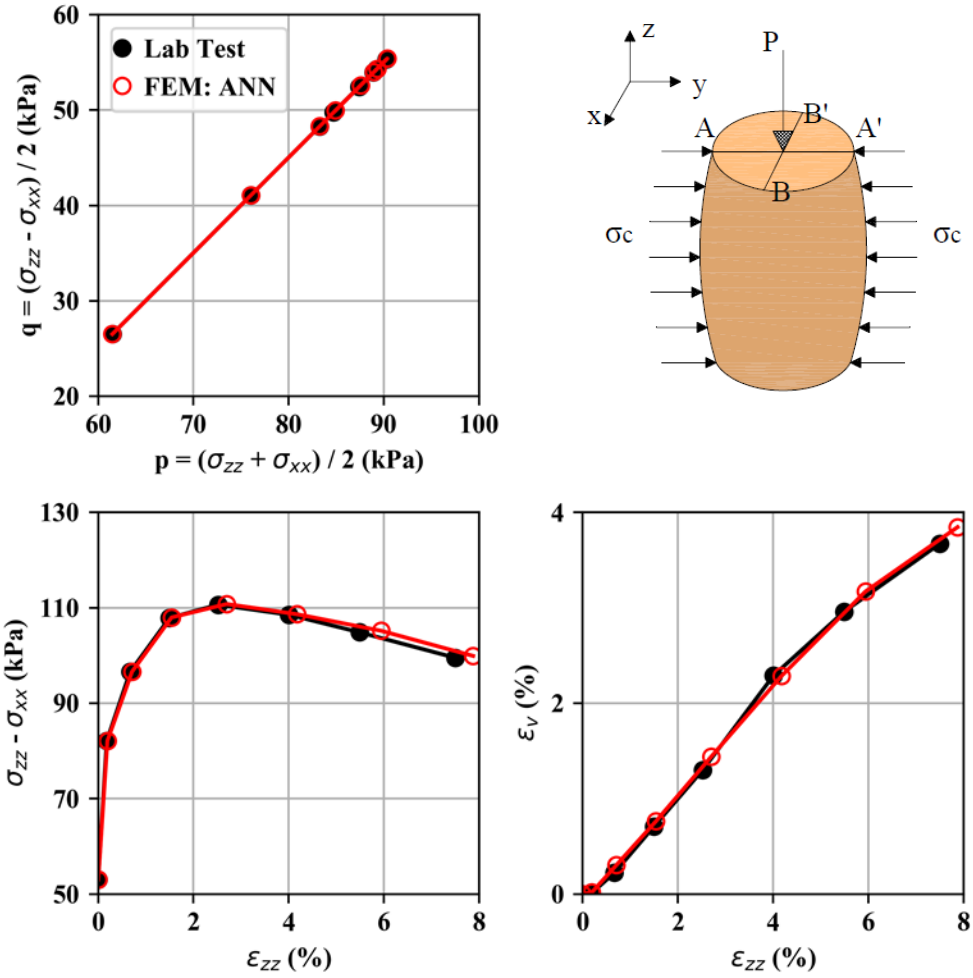


Figure 8-31: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the CT1 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

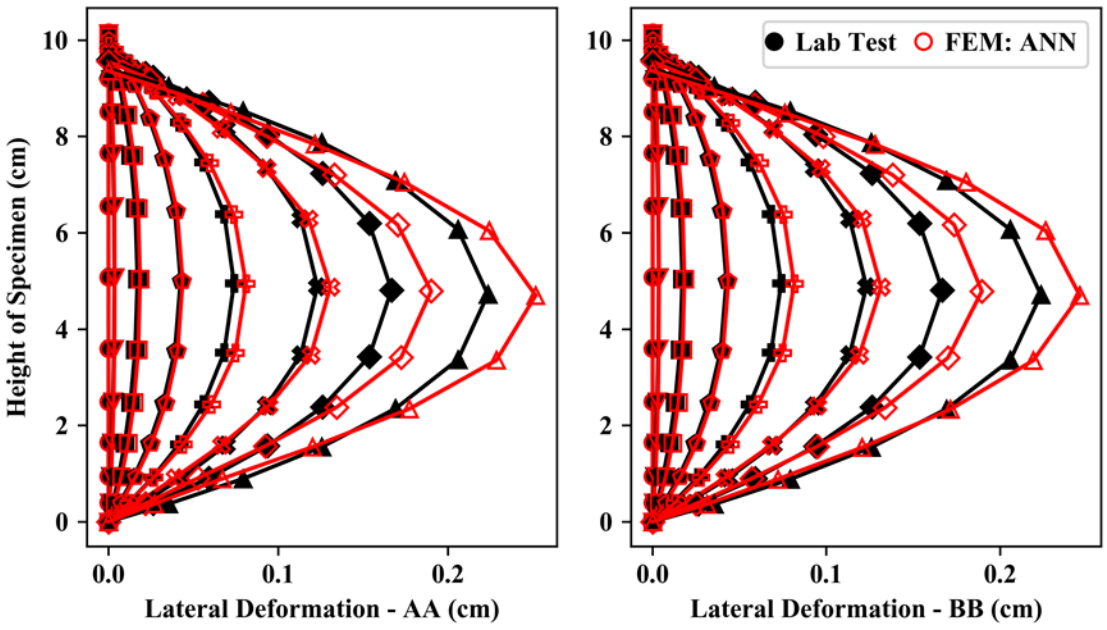


Figure 8-32: Comparison of the deformations measurements of the CT1 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

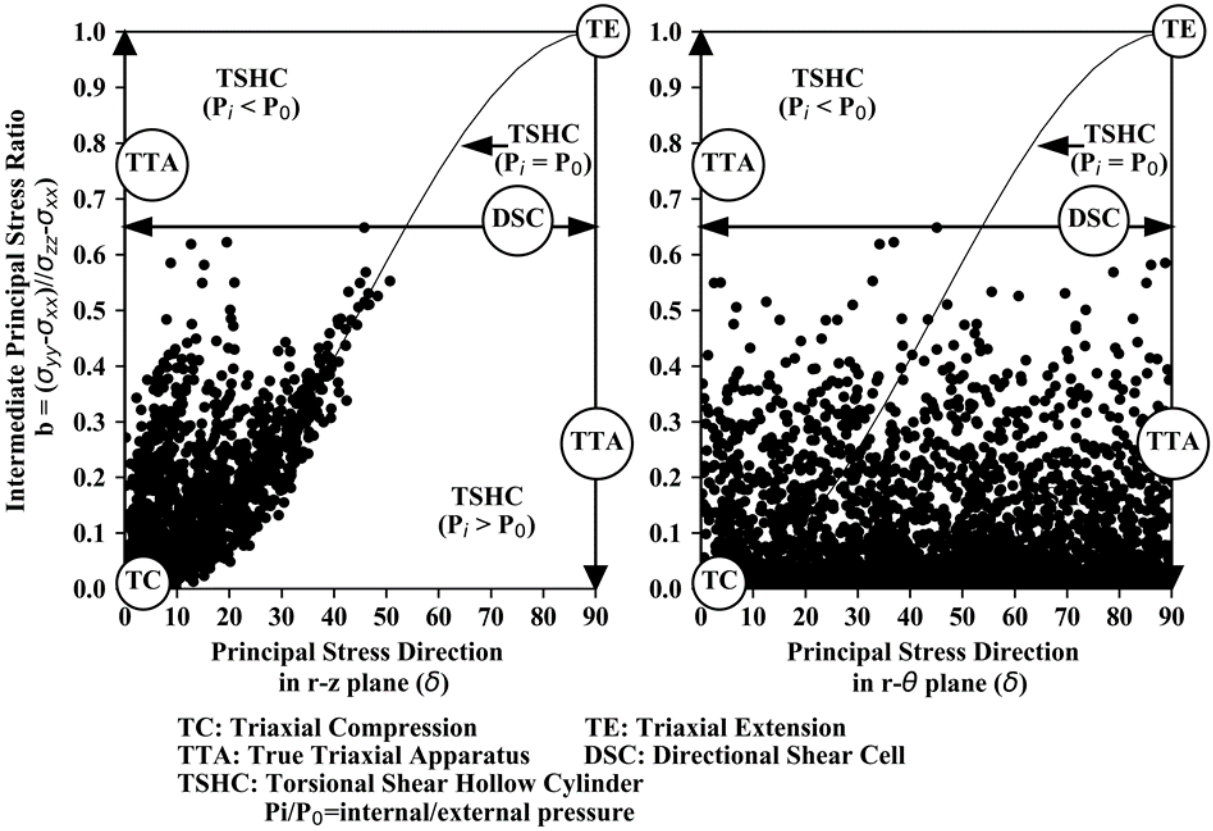


Figure 8-33: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the CT1 lab test

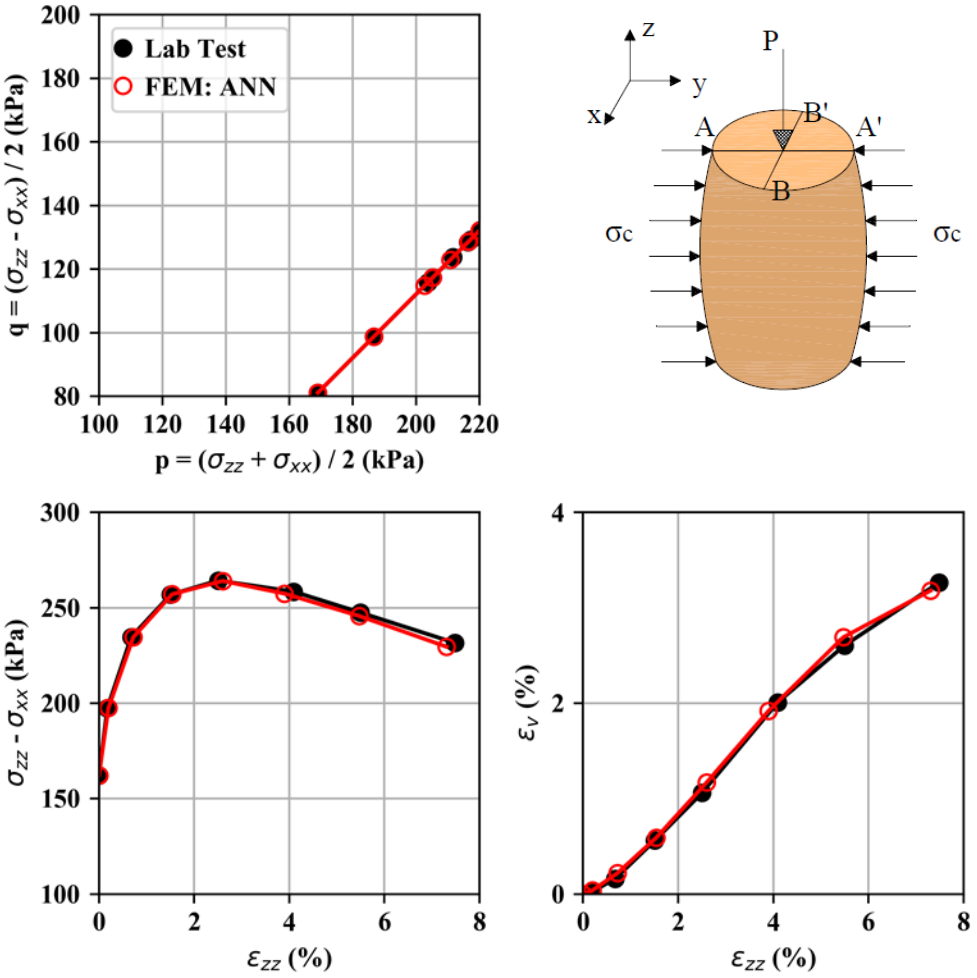


Figure 8-34: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the CT2 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

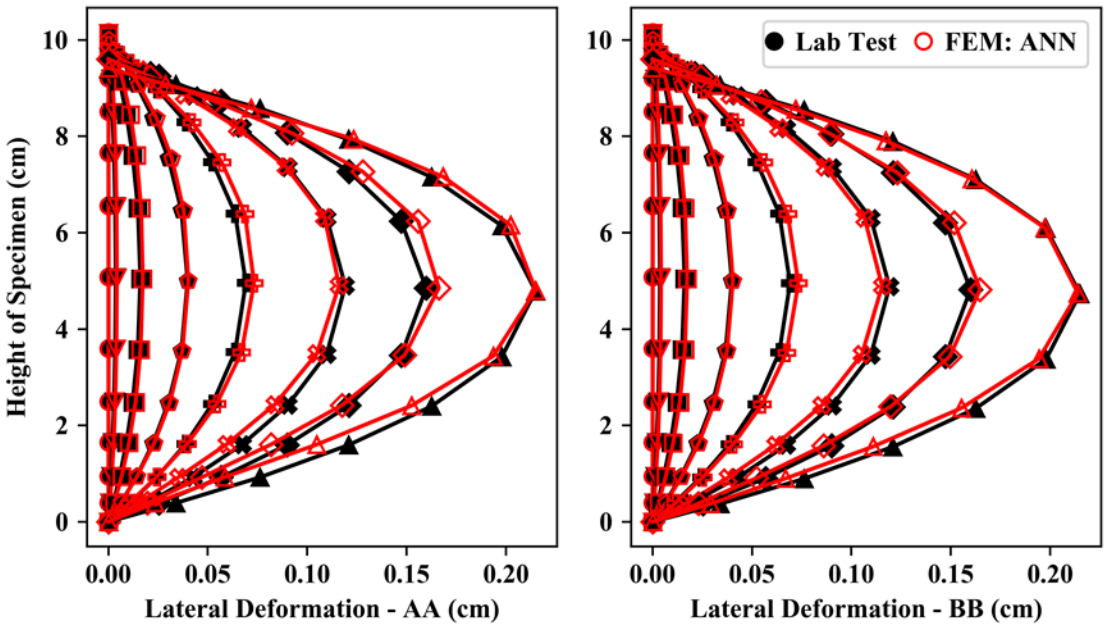


Figure 8-35: Comparison of the deformations measurements of the CT2 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

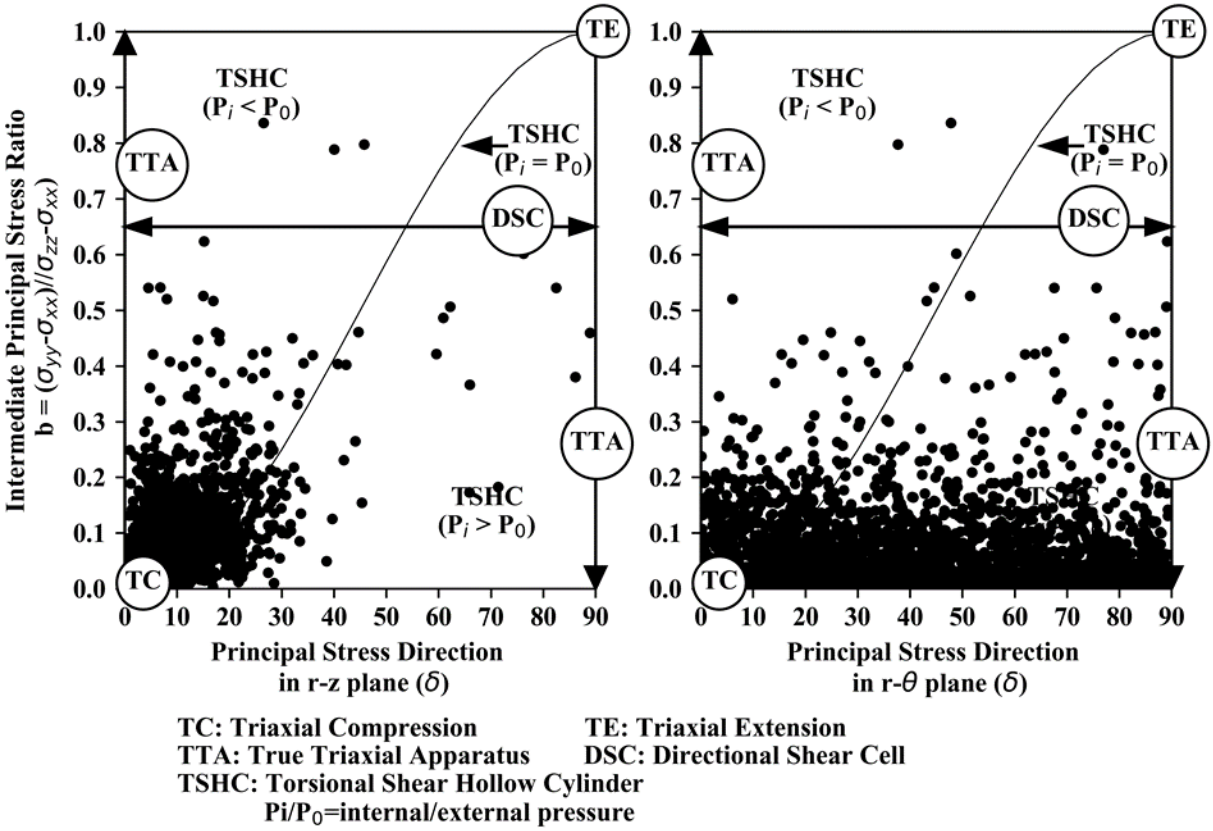


Figure 8-36: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the CT2 lab test

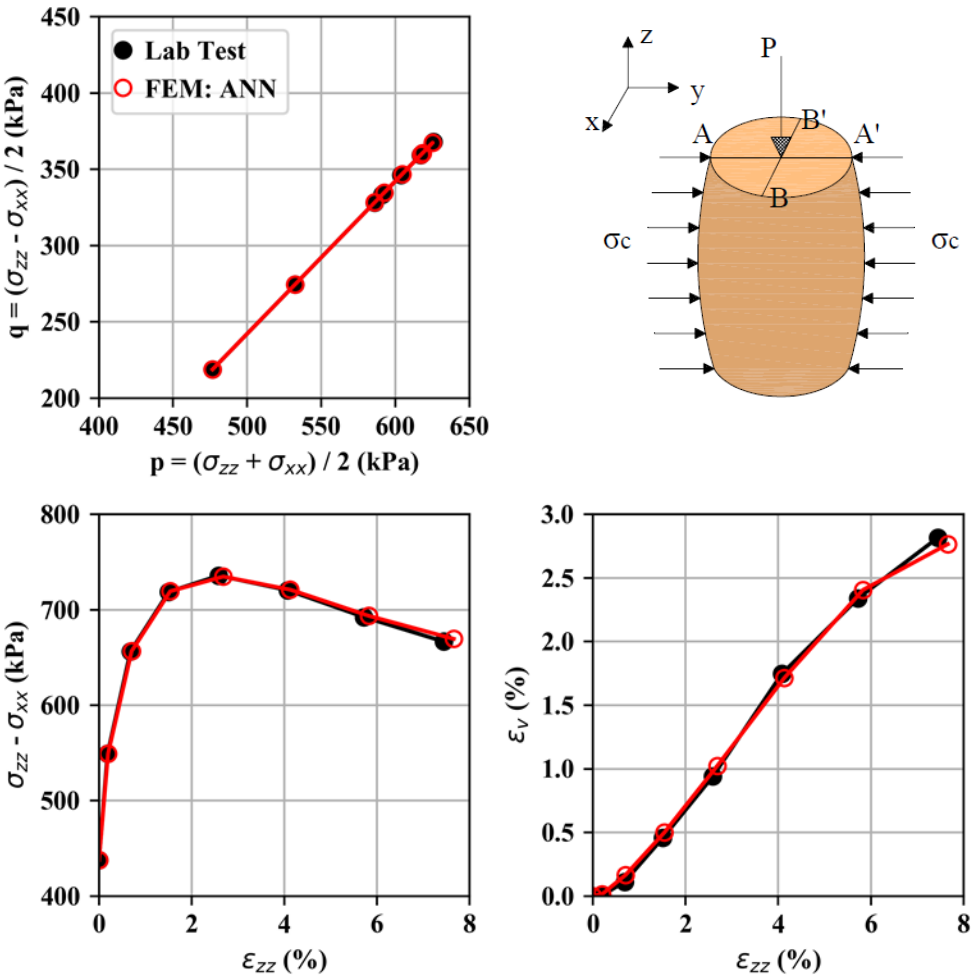


Figure 8-37: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the CT3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

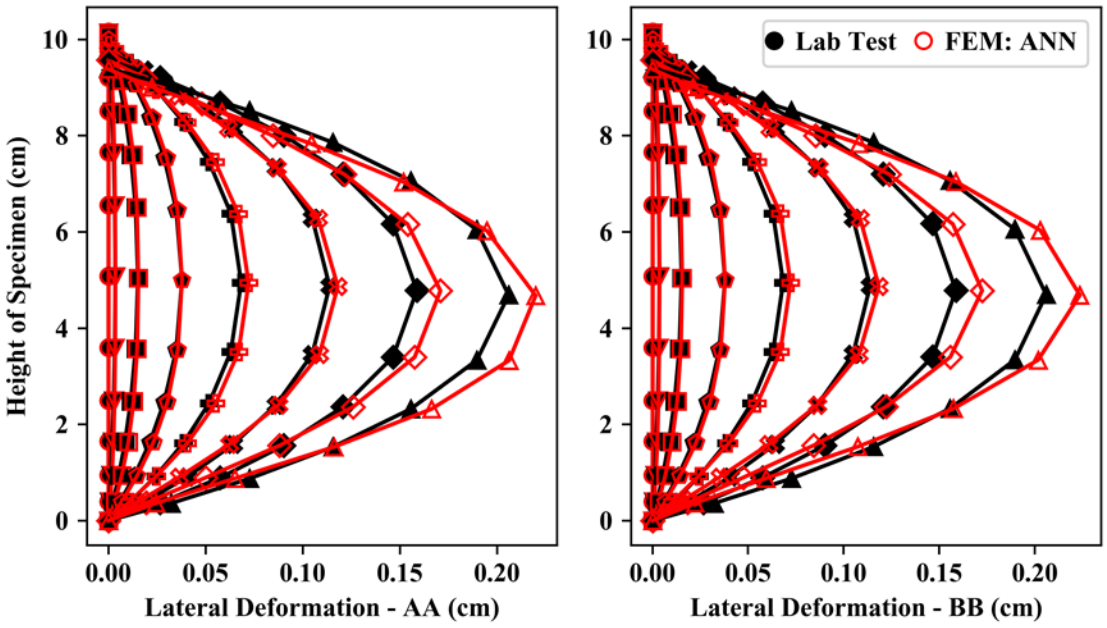


Figure 8-38: Comparison of the deformations measurements of the CT3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

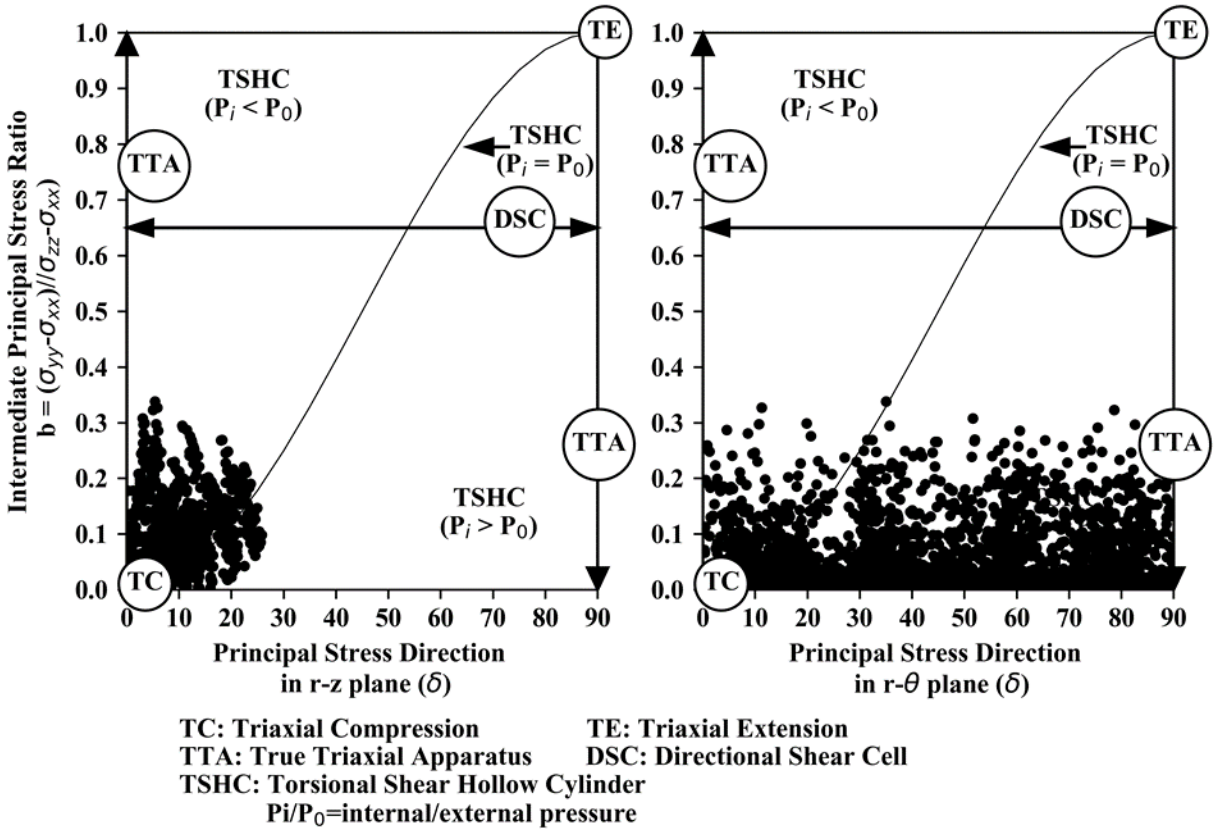


Figure 8-39: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the CT3 lab test

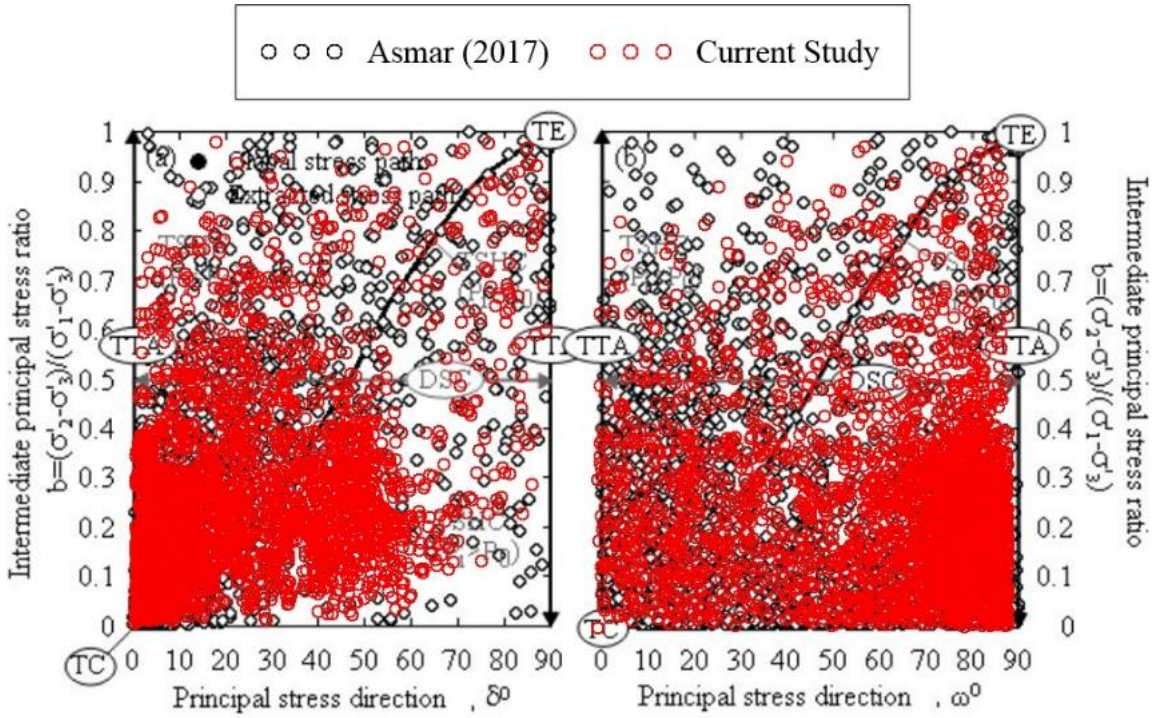


Figure 8-40: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the MS1 test.

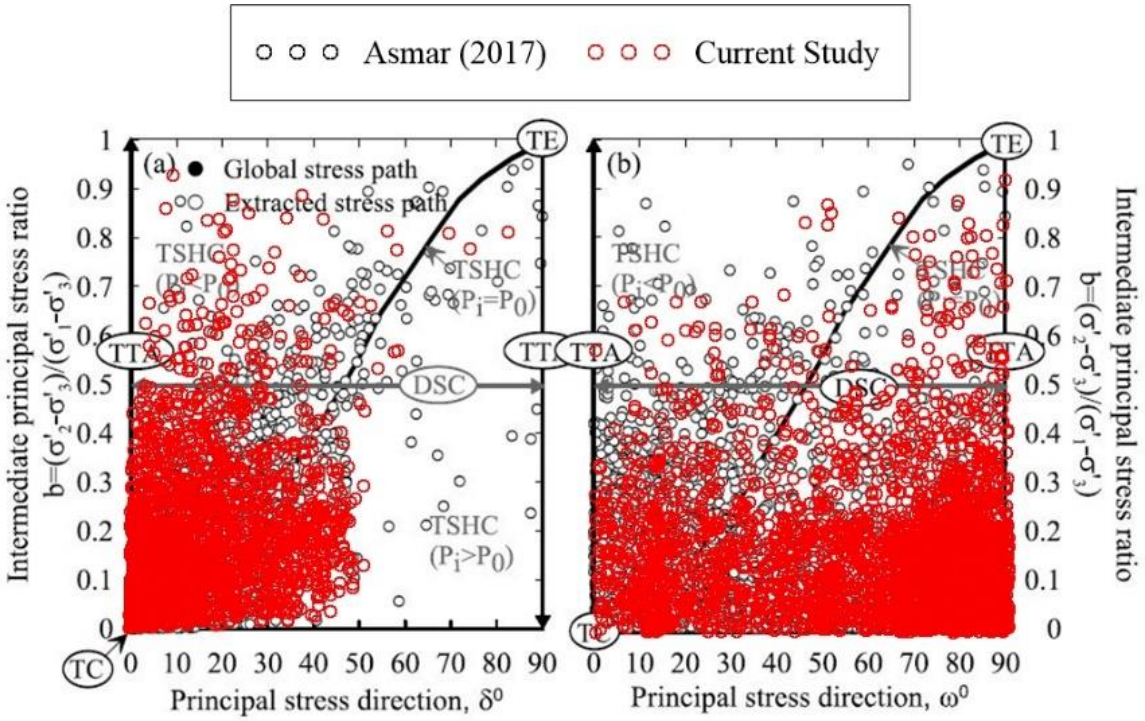


Figure 8-41: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the MS2 test.

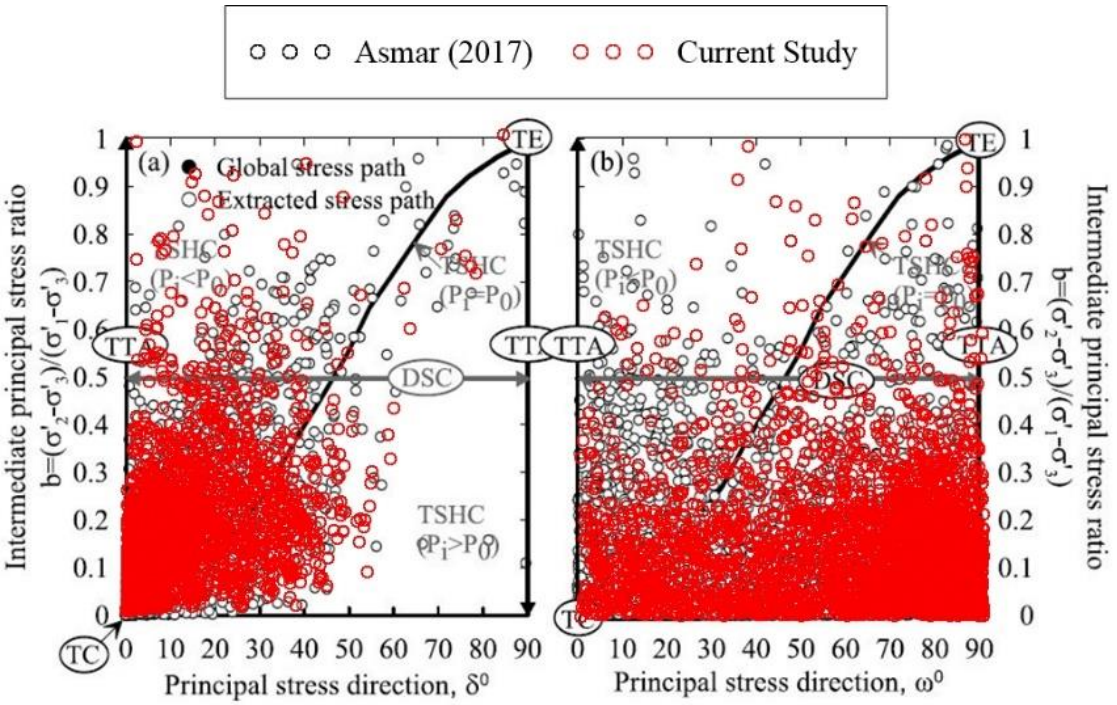


Figure 8-42: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the MS3 test.

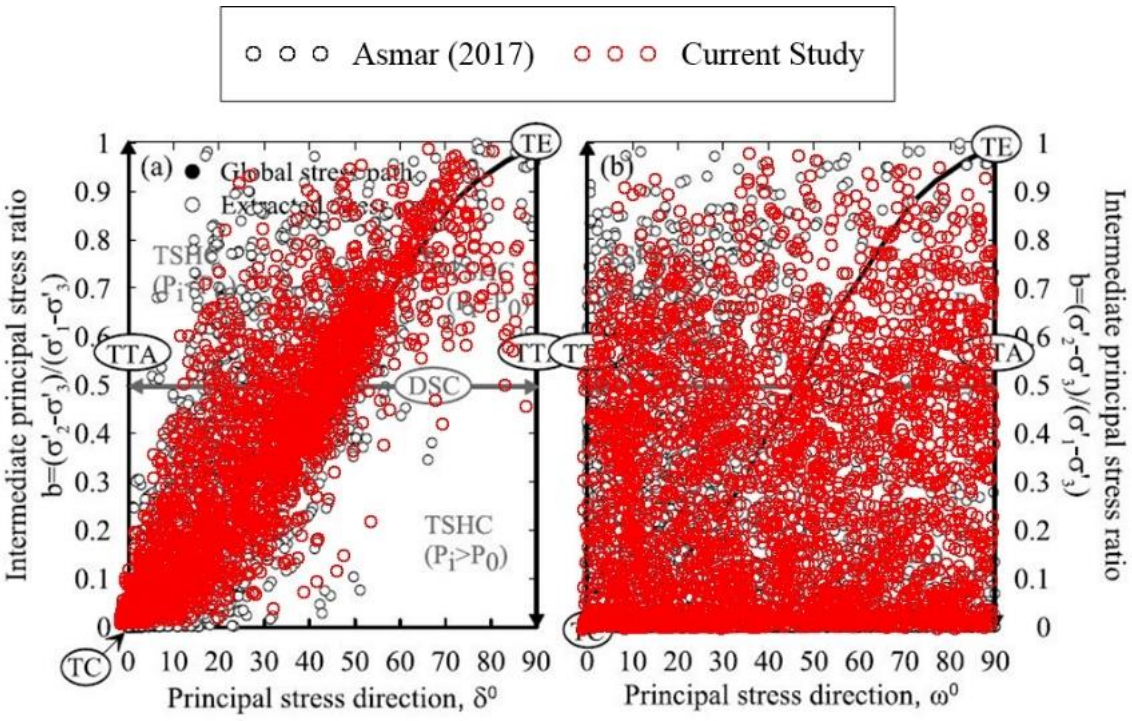


Figure 8-43: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the CS1 test.

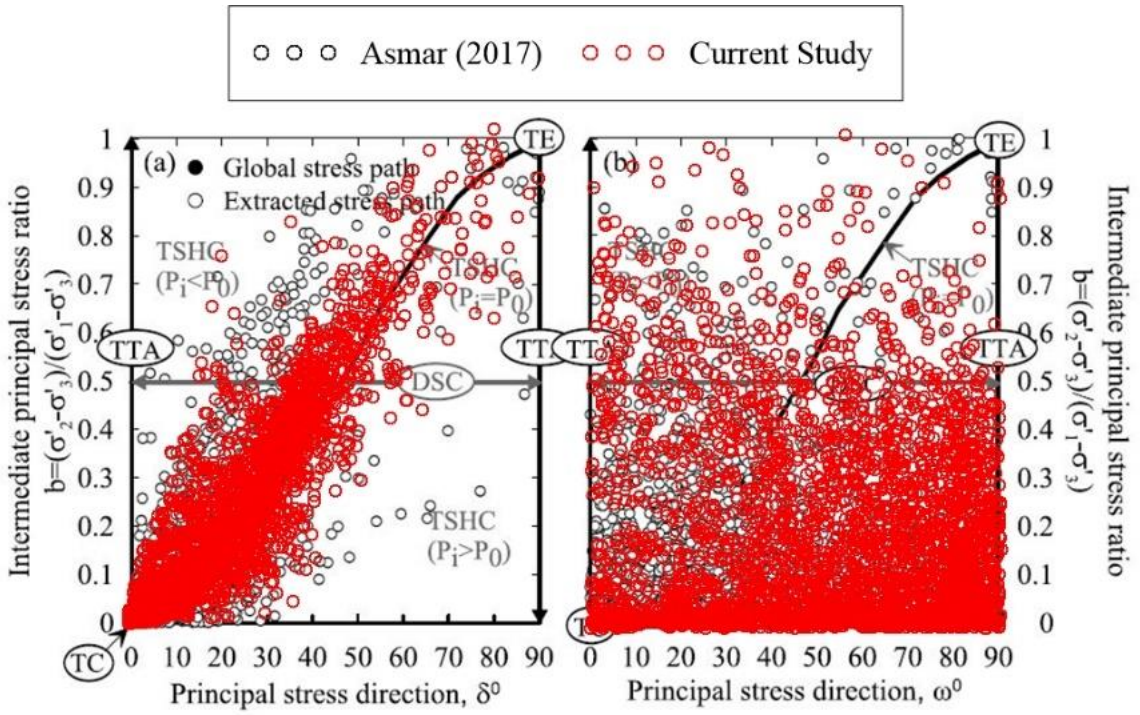


Figure 8-44: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the CS2 test.

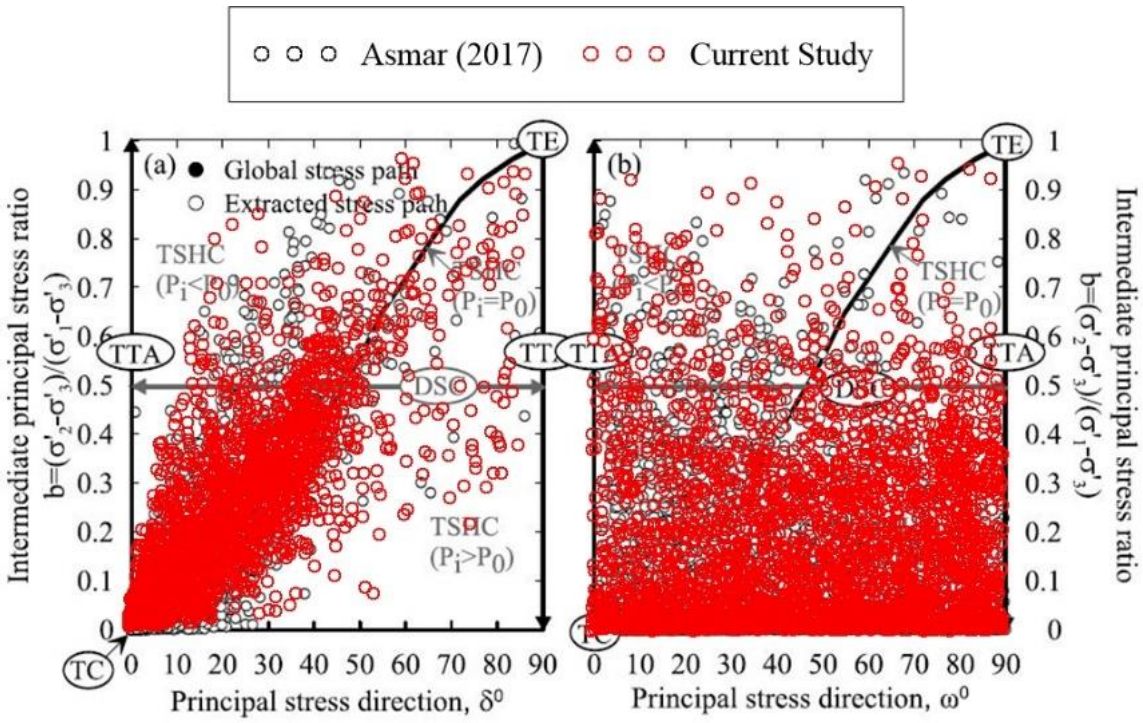


Figure 8-45: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the CS3 test.

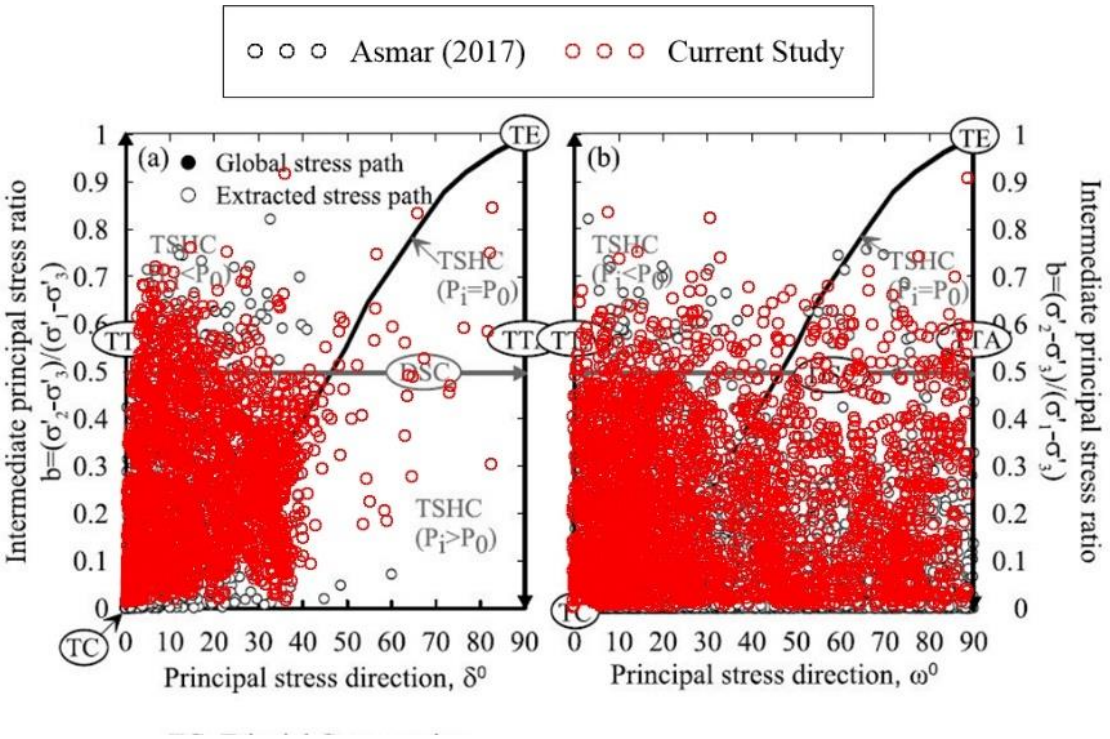


Figure 8-46: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the MT1 test.

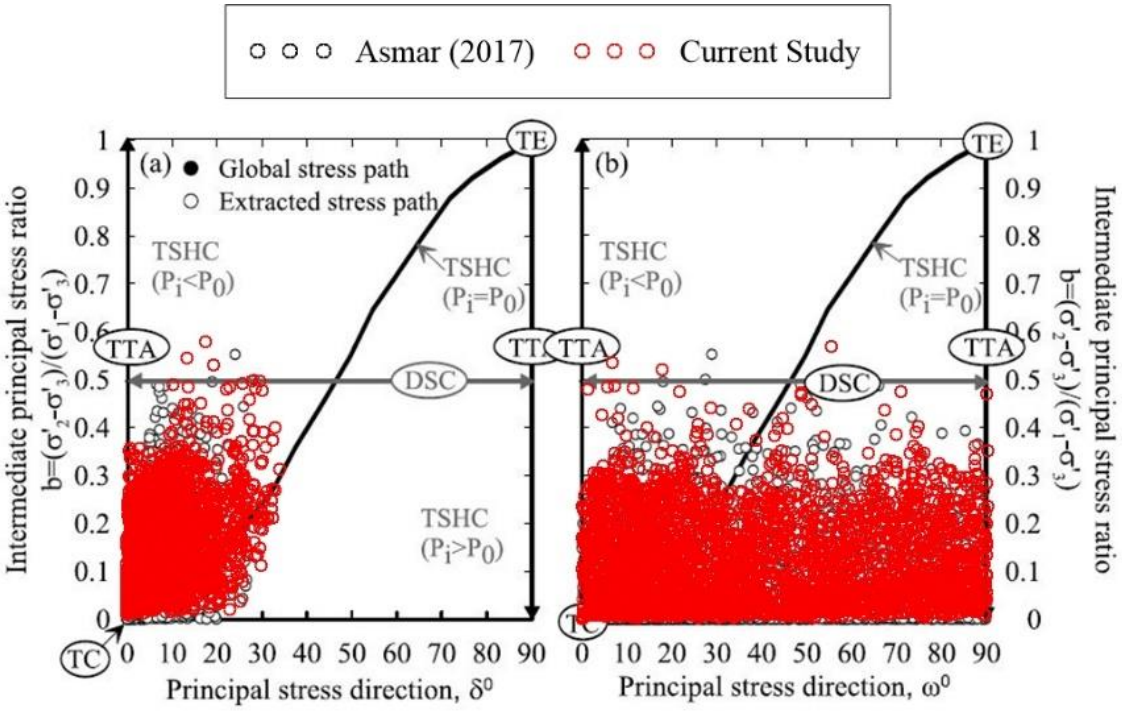


Figure 8-47: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the MT2 test.

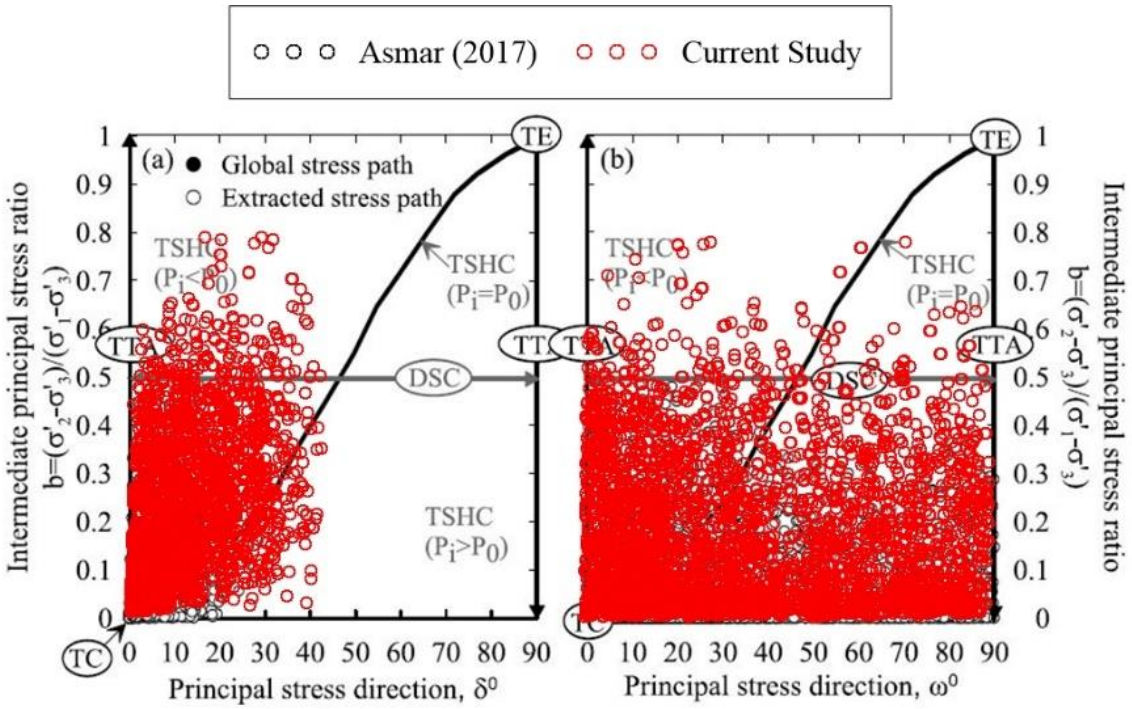


Figure 8-48: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the MT3 test.

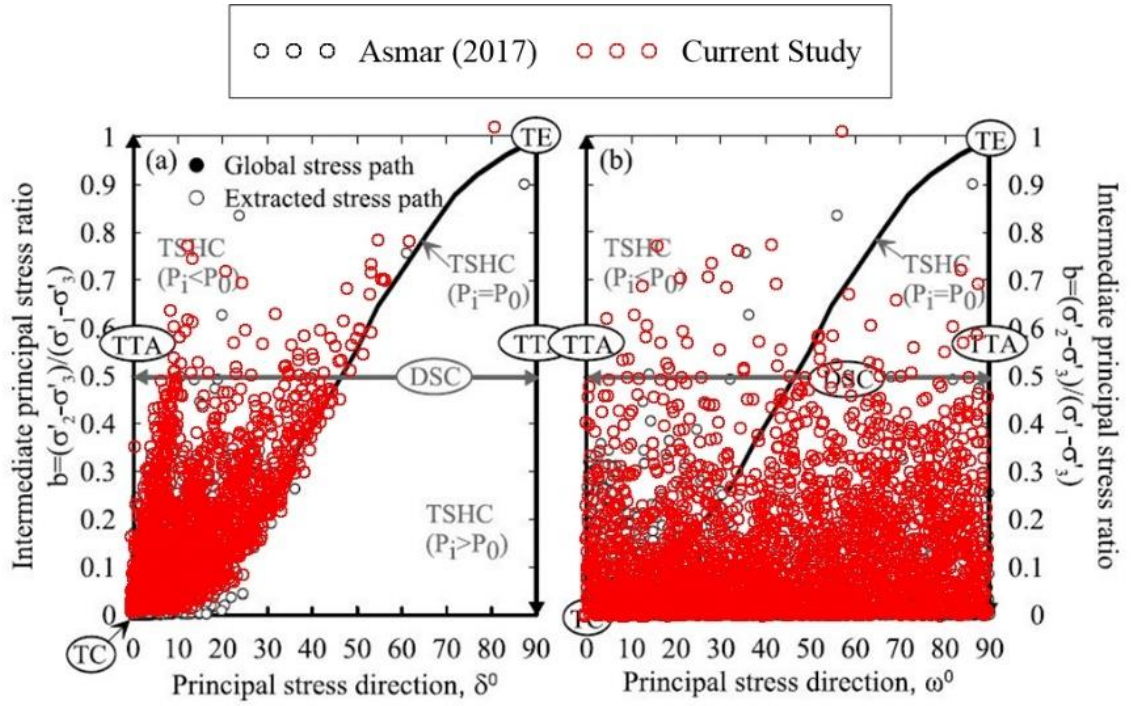


Figure 8-49: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the CT1 test.

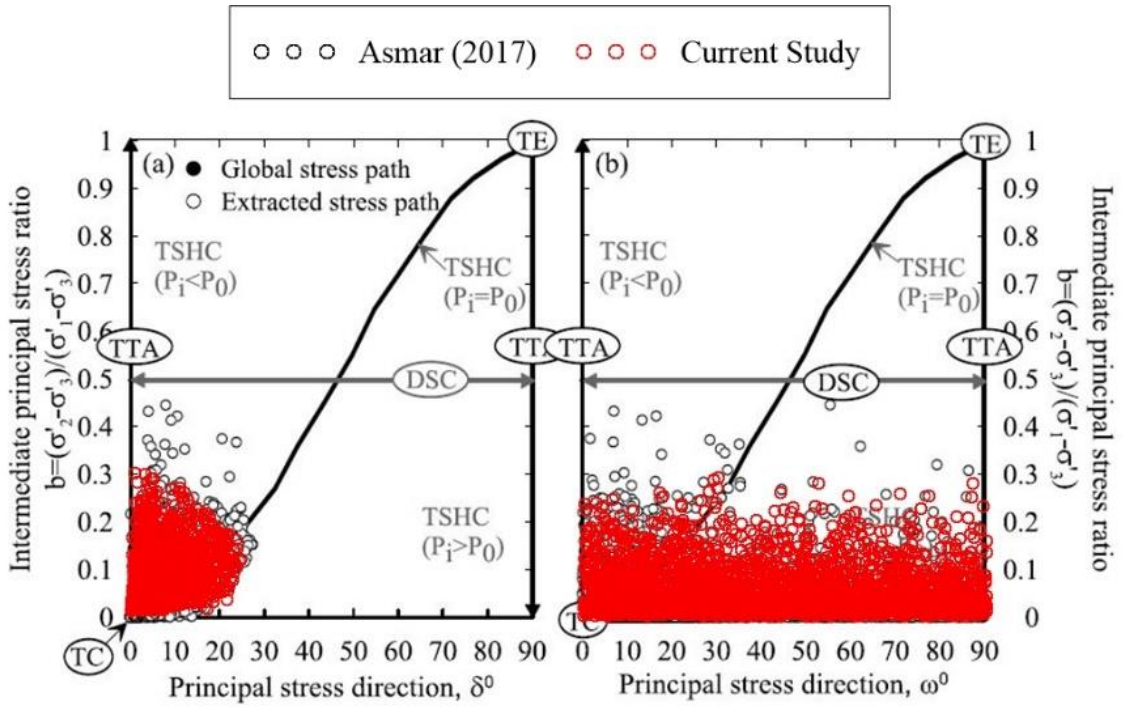


Figure 8-50: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the CT2 test.

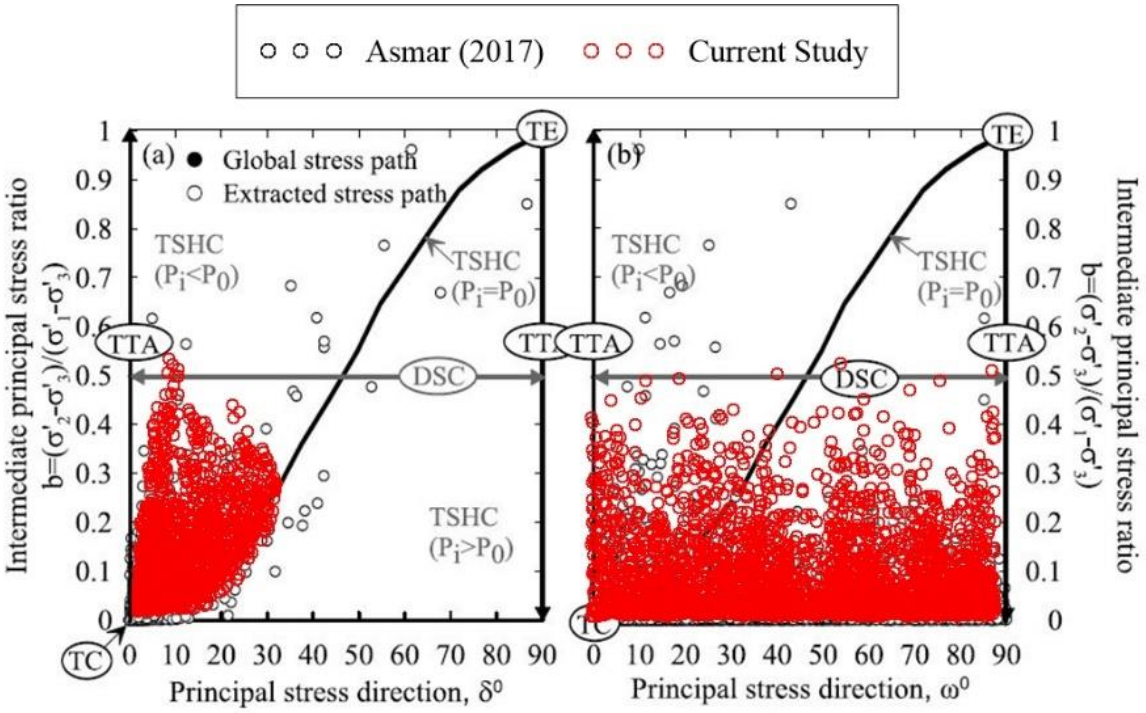


Figure 8-51: Comparison of the stress space calculated via SelfSim in the current study vs that calculated in (Asmar 2017) for the CT3 test.

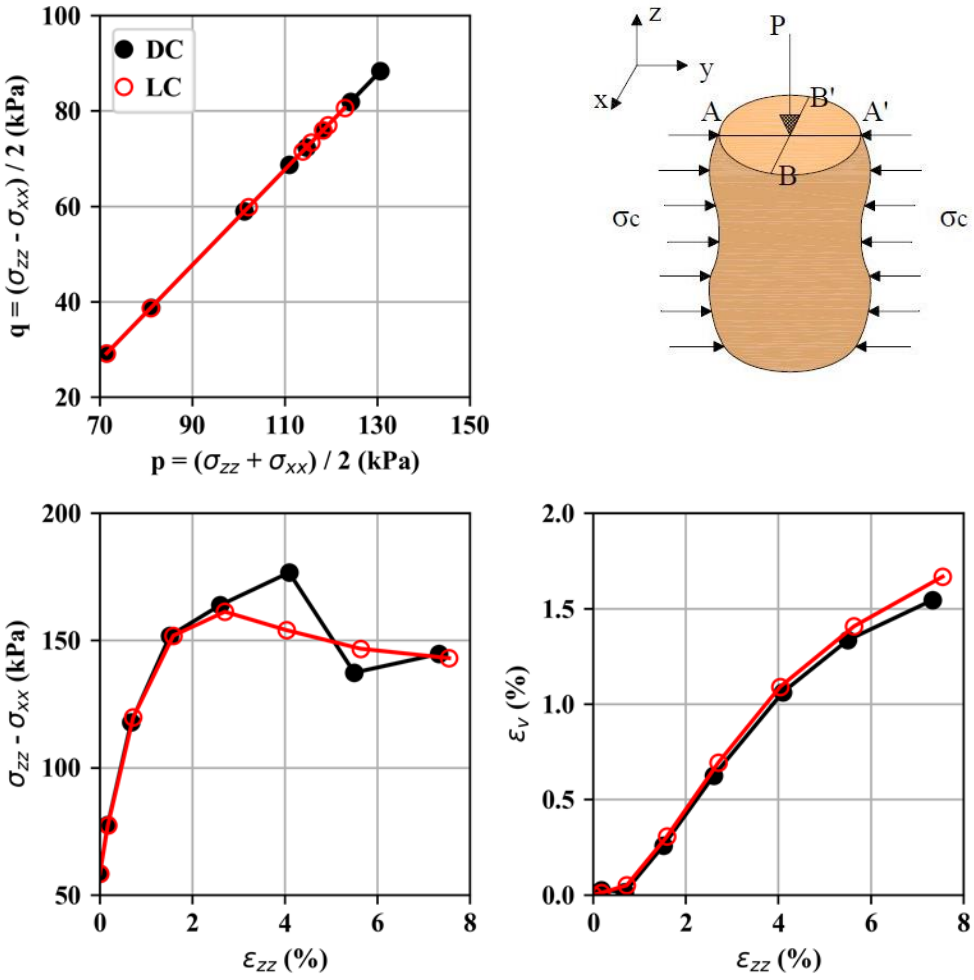


Figure 8-52: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for calculated by the load-controlled vs the displacement-controlled FEM analysis both employing the ANN model trained through SelfSim of the MT1 lab test

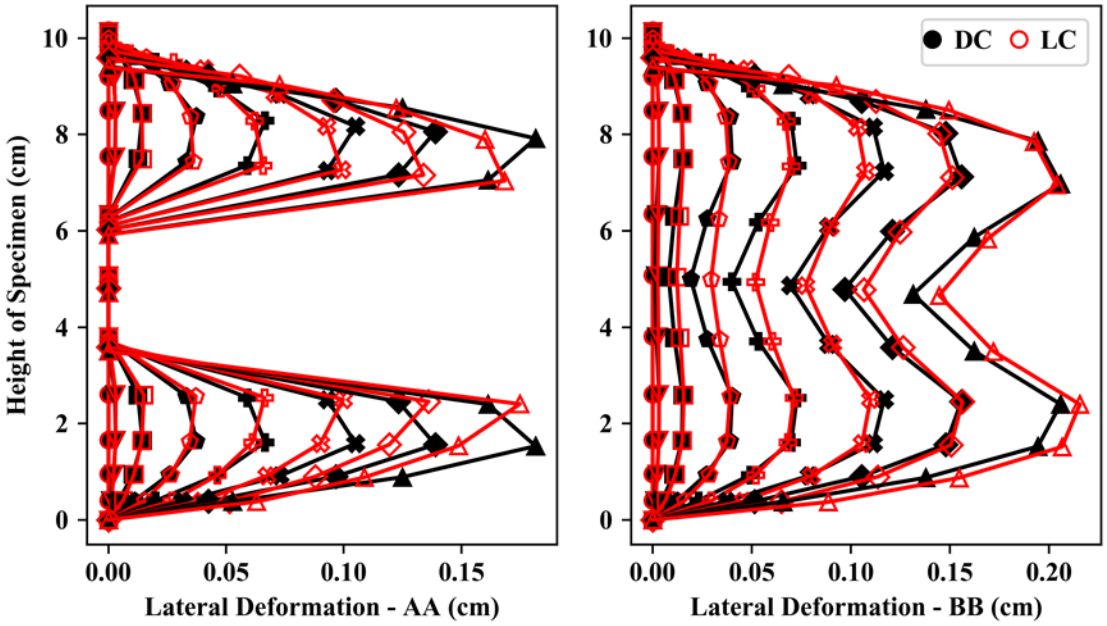


Figure 8-53: Comparison of the deformations calculated by the load-controlled vs the displacement-controlled FEM analysis both employing the ANN model trained through SelfSim of the MT1 lab test

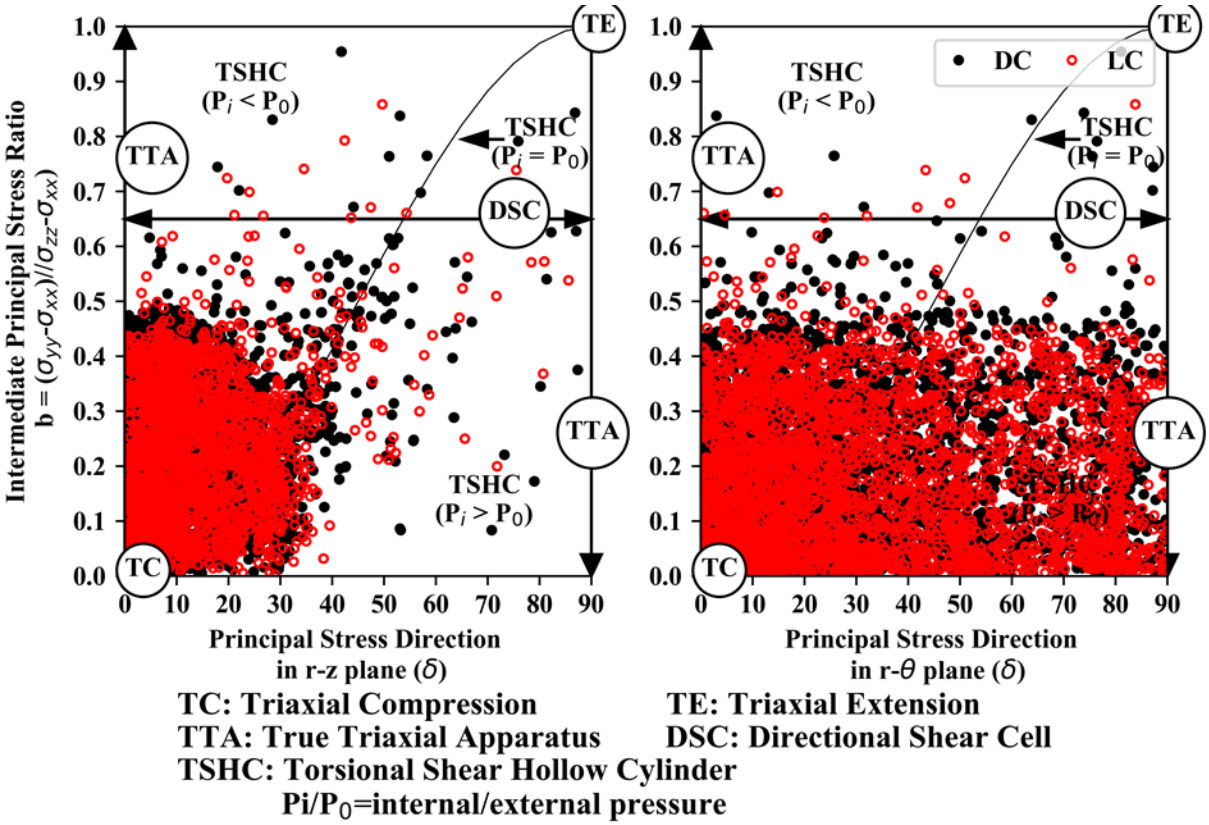


Figure 8-54: Comparison of the stress-space calculated by the load-controlled vs the displacement-controlled FEM analysis both employing the ANN model trained through SelfSim of the MT1 lab test

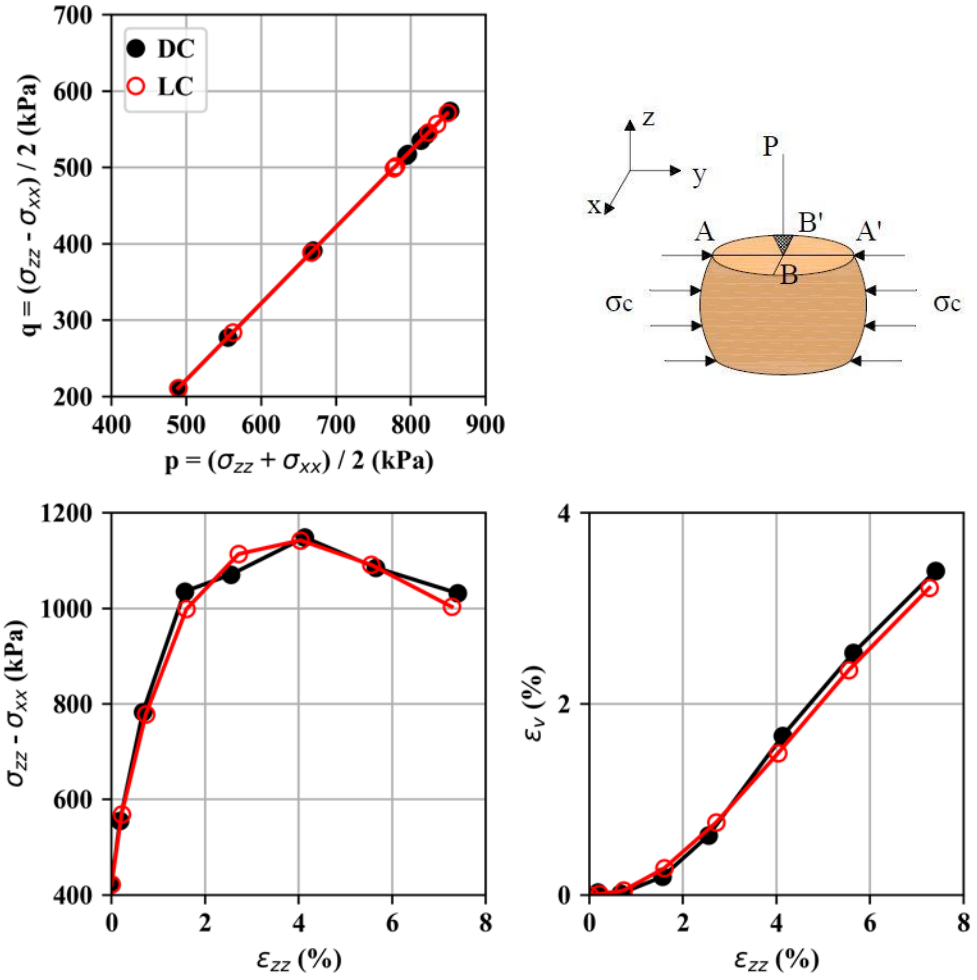


Figure 8-55: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for calculated by the load-controlled vs the displacement-controlled FEM analysis both employing the ANN model trained through SelfSim of the CS3 lab test

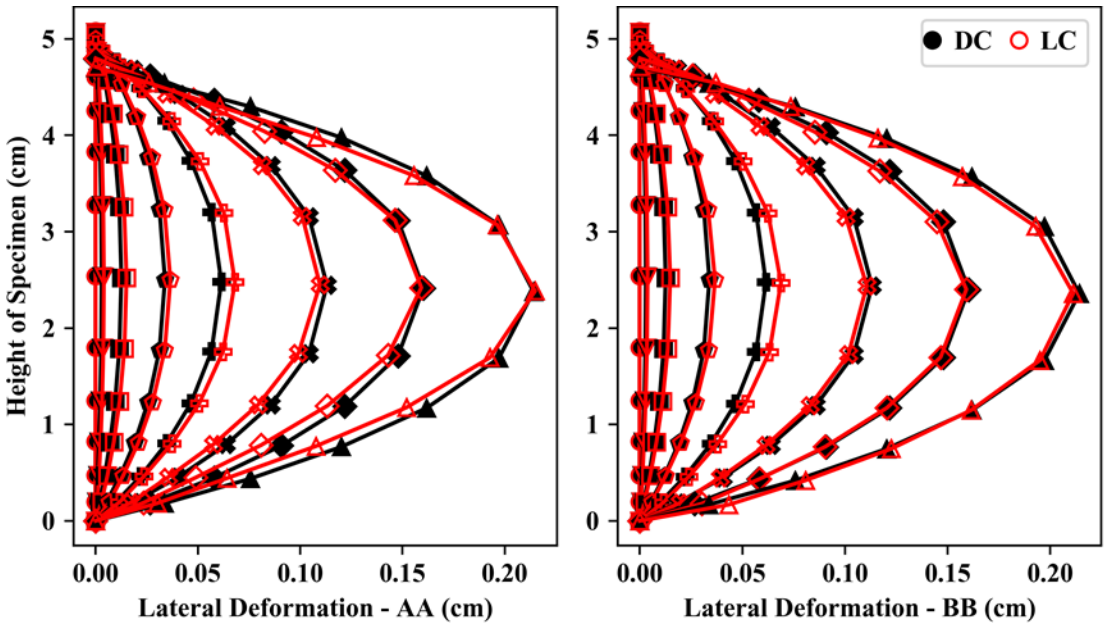


Figure 8-56: Comparison of the deformation calculated by the load-controlled vs the displacement-controlled FEM analysis both employing the ANN model trained through SelfSim of the CS3 lab test

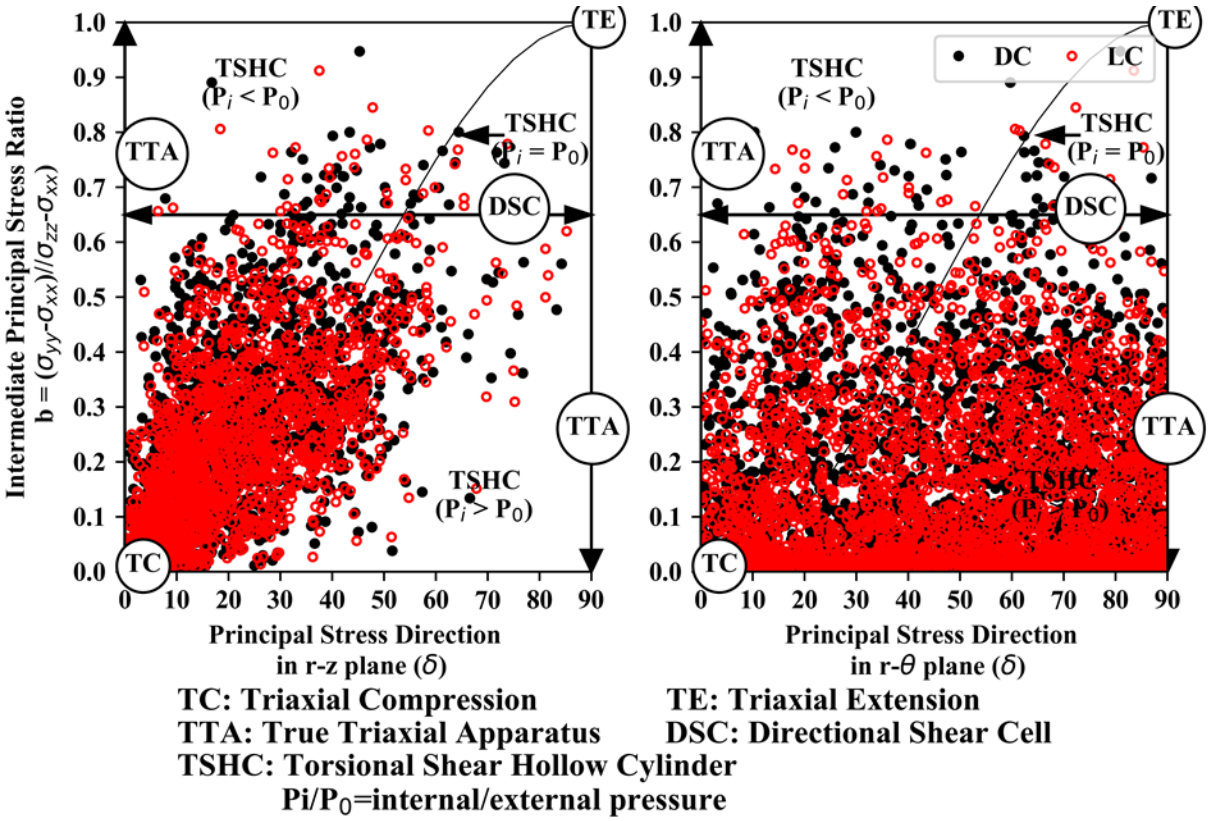


Figure 8-57: Comparison of the stress space calculated by the load-controlled vs the displacement-controlled FEM analysis both employing the ANN model trained through SelfSim of the CS3 lab test

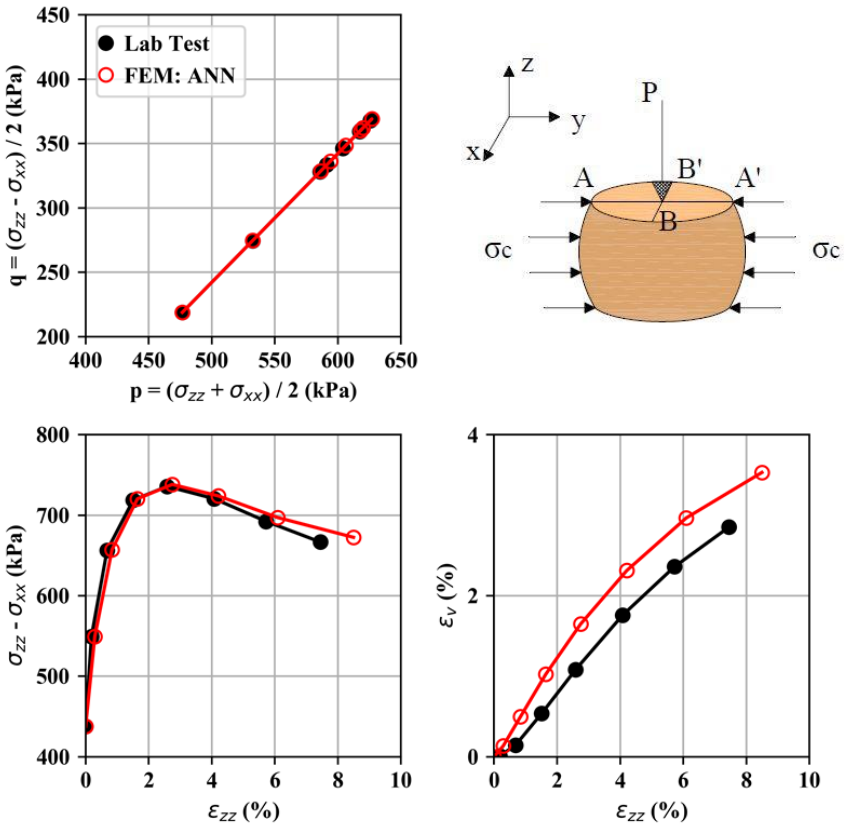


Figure 8-58: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain of the CT3 test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the MT3 test

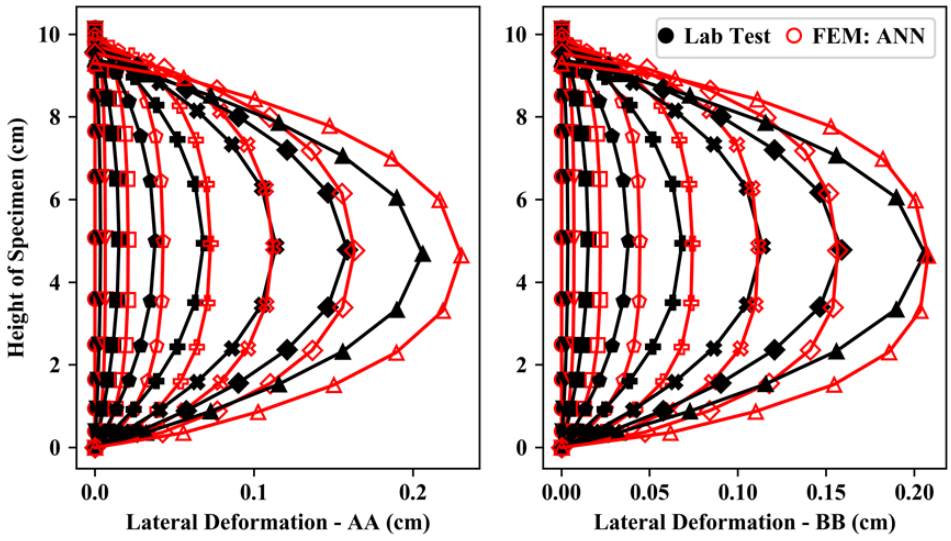


Figure 8-59: Comparison of the deformations of the CT3 test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the MT3 test

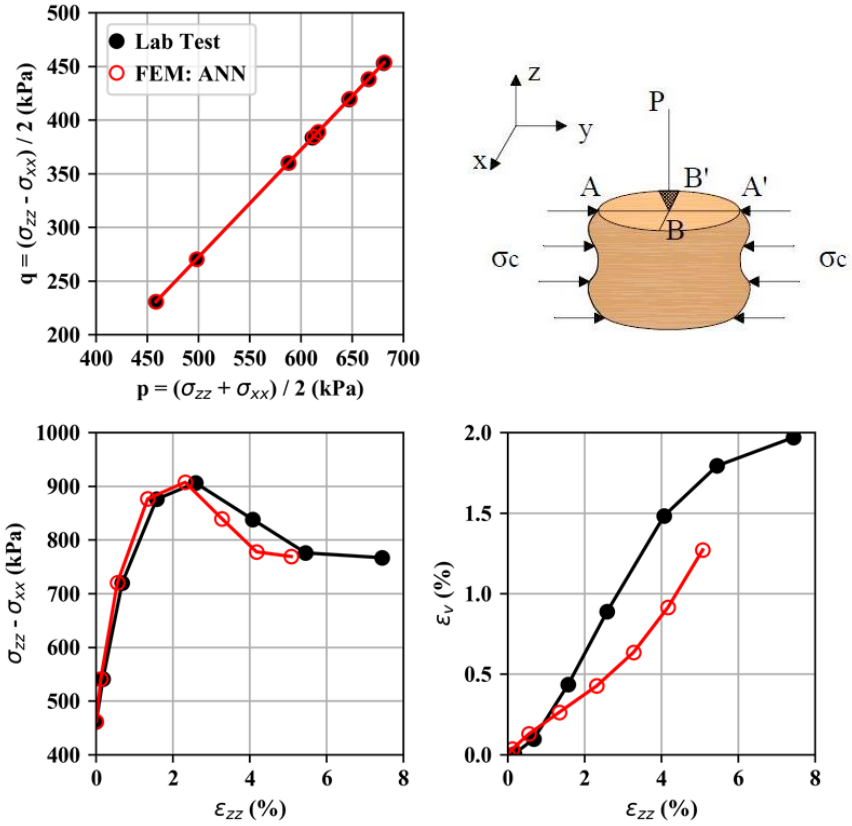


Figure 8-60: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain of the MT3 test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the CT3 test

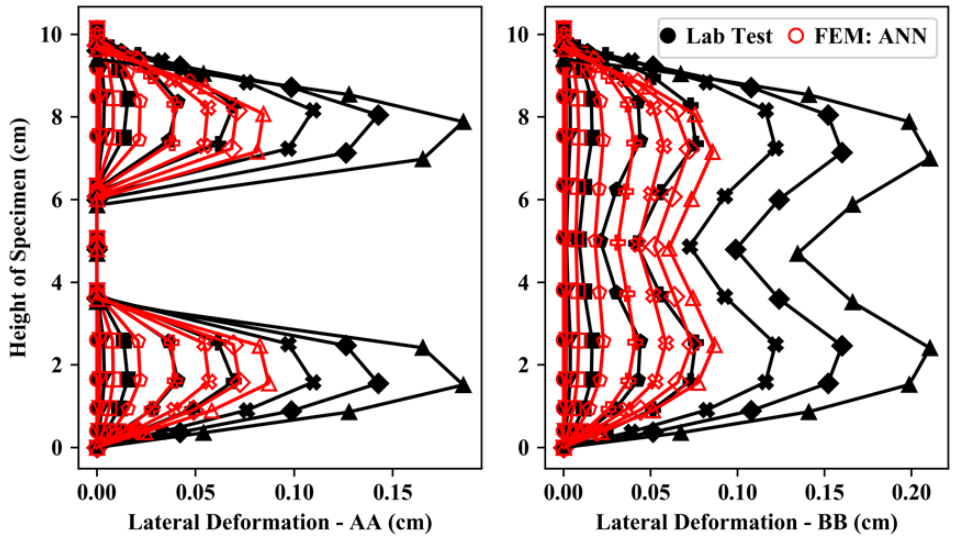


Figure 8-61: Comparison of the deformations of the MT3 test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim of the CT3 test

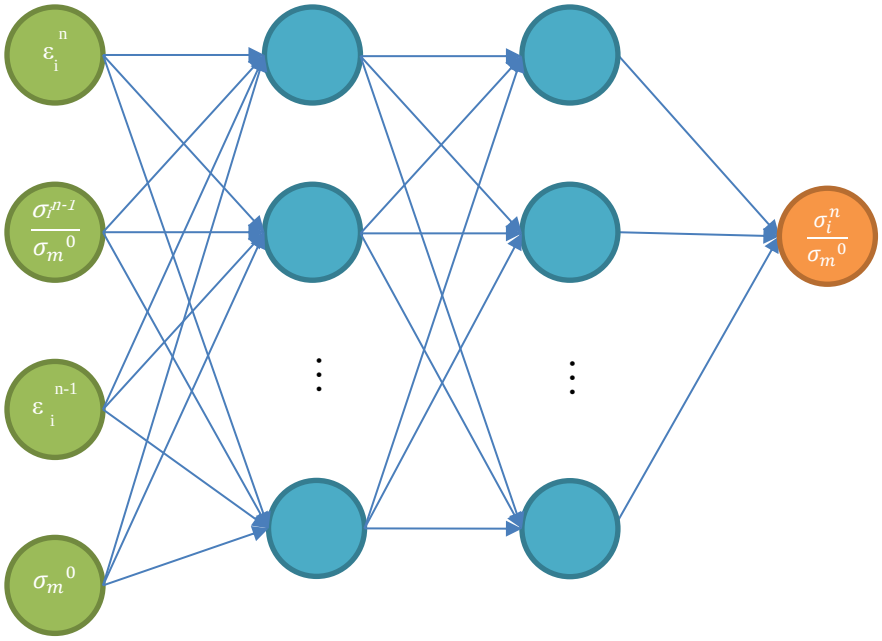


Figure 8-62: Schematic of the architecture of the tested unified ANN model. The calculated and previous stresses are normalized by the initial confining pressure. The initial confining pressure is also passed as an input to the ANN.

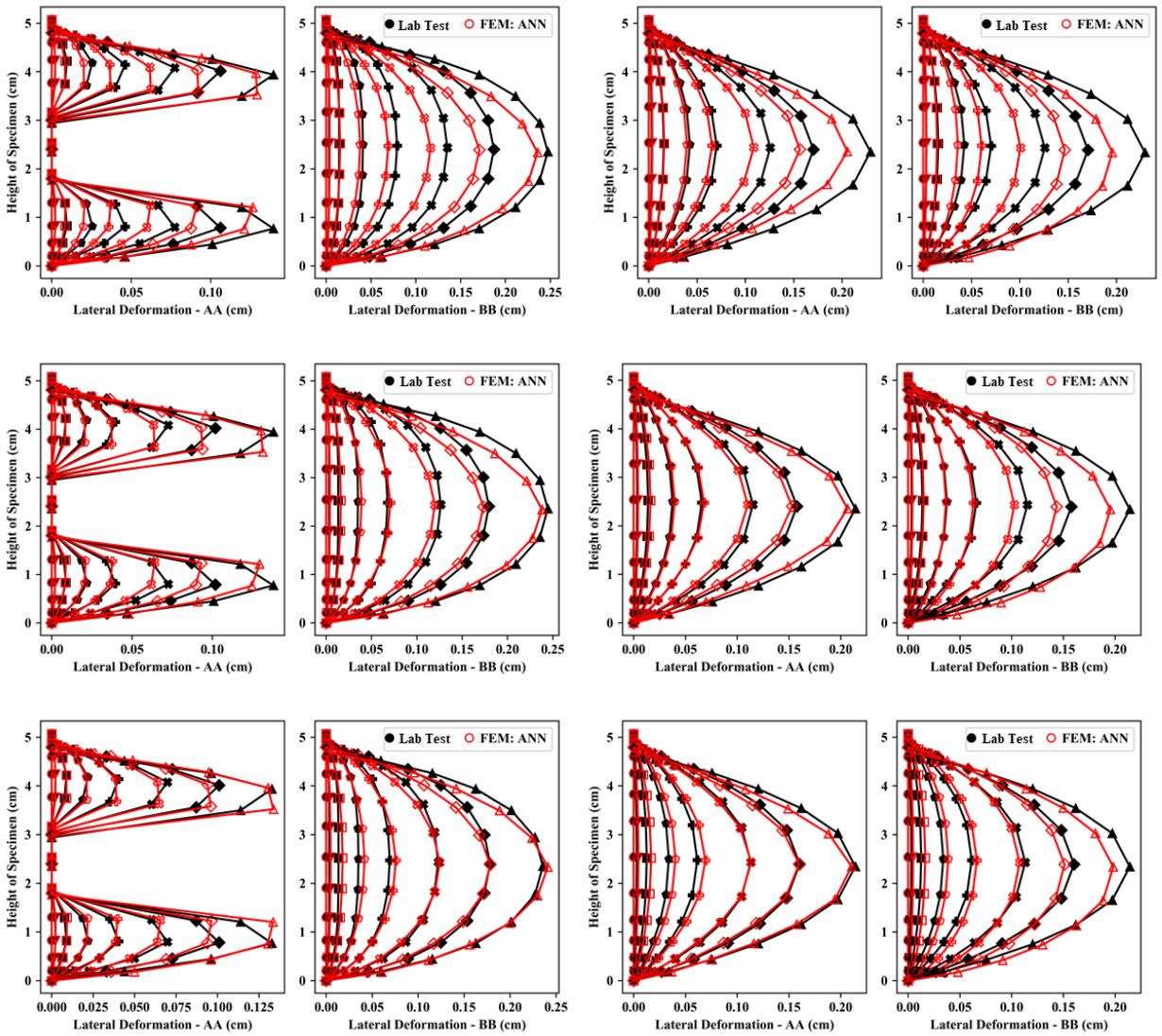


Figure 8-63: Comparison of the deformations of the (MT1, MT2, MT3, CS1, CS2 and CS3) tests test vs those calculated by the FEM analyses employing the unified ANN model trained through the simultaneous SelfSim.

CHAPTER 9. Conclusions, Limitations and Recommendations

9.1 Development of a Wireless Through-Soil Communication System

In this study a new through-soil communication system was developed that can communicate information wirelessly underground. This required understanding the properties of the electromagnetic and acoustic soil channels, selecting the acoustic channel and the frequency of operation, then designing the system by selecting some components while designing other components. Modern communication techniques such as channel characterization, equalization, error correction and energy-efficient modulation were incorporated in the system.

The communication system employs low-frequency acoustic waves which were found to benefit from low attenuation while being possible to generate using small sources. Validation tests were performed in lab and field. Communication up to a range of 50 meters was achieved. A prototype for a practical system that has a compact design enabling it to fit in medium-sized boreholes, and which has a communication range of 10 meters and a power efficiency enabling its batteries to last for 3 years of operation was developed. This communication system was used to transmit geotechnical sensor data, demonstrating its potential use in geotechnical engineering.

9.2 Development of the New SelfSim Code

A new highly efficient code capable of running SelfSim-type analyses was developed. The code was written in python, while employing Google's TensorFlow library for machine learning. The code employed LS-DYNA as its FEM framework, due its computational efficiency. A user-defined material was implemented in Fortran and integrated into LS-DYNA. A new formulation for the stiffness matrix that enables vectorization was introduced. The code was vectorized and optimized, enabling its run times to drop significantly compared to the previous version of the code. SelfSim analyses that used to take hours can now be completed in minutes.

9.3 Testing and Using the New SelfSim Code

The code was tested in multiple stages, verifying the functionality of its individual elements then the functionality of the whole interconnected code. The testing was performed by running forward and SelfSim analyses to learn behavior of two constitutive models then comparing the results of the trained models to the original models. The uniqueness of the solution, the generalization capability of the trained ANNs and the extracted stiffness matrices are topics that were investigated in this study.

The code was then used to learn the behavior of Ottawa sands which was measured through a set of 12 conventional and modified triaxial compression tests. The SelfSim analyses were preformed efficiently consuming in average 10 minutes per analysis compared to an average of 4 hours per analysis using the previous code. The new code arrived at well-trained ANNs in 2 or less passes compared with 8 passes using the previous code.

9.4 Potential of the Developed Tools

SoilComm and SelfSim Gen3 have the potential for impacting applications in multiple disciplines. In geotechnical engineering, SoilComm can be employed in the instrumentation systems for geotechnical construction, enhancing the robustness and versatility of these systems and likely reducing their costs. SelfSim can be employed by engineers to quickly update the estimates for future performance of a geotechnical system based on its previous performance, hence assisting them in evaluating options, making decisions, and early identifying and addressing possible issues. It can be used as a methodical way for benefitting from sensors data and can alleviate a lot of the human effort that goes into that. In addition, SelfSim can enable a new class of soil testing which extracts complex soil behavior from few tests due to the nonuniformity of their boundary conditions. SelfSim can be used to create site-specific soil models that capture better site-specific soil behavior, and are less affected by conventional material models generalizations, assumptions, and data biases.

SoilComm can be used in many other applications that use embedded sensors such as mining monitoring and communication, agricultural monitoring, landfills, tailing dams, environmental monitoring, and seismic monitoring. The technology can help in the research related to the subsurface such as soil studies, fossorial animal studies, seismology, groundwater studies, etc. In addition to its practical use, the developed system represents a starting point for future experimentation for enhancing the understanding of the mechanical wave propagation through soil and developing better communication systems.

SelfSim Gen3 can be used to quickly model different kinds of materials with complex behavior such as biological materials, new artificial materials, composites, etc. The flexibility of its modelling unit, the ANN, allows for quick integration of material testing and material modelling. The enhanced speed may be especially useful for time-sensitive applications such as biological imaging. It may also allow future researchers to experiment much faster with the technology to improve the understanding of ANN material modelling and the SeflSim inverse algorithm.

9.5 Recommendations for Future Work

As the field and lab testing of the developed through-soil communication system was limited to relatively shallow deployment depths not exceeding 2 meters, it is recommended to test the system in deeper deployment depths, in deposits of different properties, and specifically different saturation levels. Developing methods for dealing with ambient noise and algorithms for communicating data within clusters of instruments equipped with those wireless transceivers would increase the efficiency and rigorousness of the system. The developed communication system prototype can benefit from further optimization in its size, energy efficiency and design. Reducing the cost of this system while optimizing its properties will pave the way to wide adoption of the technology in geotechnical engineering and other industries that use underground sensors such as agriculture, and mining.

The new SelfSim code should be further tested, experimenting with different architectures, modes of operation and hyperparameters. A parametric study investigating the effect of each parameter would allow informed selection of those parameters. Studying the non-uniqueness of the solutions, and the generalization capabilities of the trained ANNs are important research topics that are highly recommended for study. Finding methods for increasing the generalization capabilities of trained models, probably by imposing constraints on the training process (regularization for example), or for limiting the non-unique solutions to stress spaces that are closer to the actual stress spaces will help make SelfSim more reliable and useful. Studying the properties of training datasets is one of the most important next steps, defining criteria for evaluating the completeness of datasets and their suitability to train ANNs that can be used in FEM analyses. This study should investigate the effect of dataset features collinearity defining thresholds which if exceeded would make identifying the actual relationships between inputs and outputs of the ANN difficult or impossible, which obstructs the assembly of a proper stiffness matrix capable of directing the FEM analysis. The study should also explore methods for augmenting incomplete datasets based on known behavior of the modeled materials.

It is recommended to test different kinds of ANNs, particularly long-short-term memory networks (LSTM), convolutional neural networks (CNN) and recurrent neural networks (RNN). This last type is particularly expected to be of use to SelfSim as the concept of path dependency is intrinsic to its architecture.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J. and Devin, M. (2016). "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467.

Aki, K. and Richards, P. G. (2002). Quantitative seismology.

Akyildiz, I. F. and Stuntebeck, E. P. (2006). "Wireless underground sensor networks: Research challenges." Ad Hoc Networks 4(6): 669-686.

Akyildiz, I. F., Sun, Z. and Vuran, M. C. (2009). "Signal propagation techniques for wireless underground communication networks." Physical Communication 2(3): 167-183.

Arbanas, Ž., Jagodnik, V., Ljutić, K., Vivoda, M., Dugonjić Jovančević, S. and Peranić, J. (2013). Remote monitoring of a landslide using integration of GPS, TPS and conventional geotechnical monitoring methods. Proceedings of the 1st symposium on landslides in Adriatic Balkan Region.

Asmar, R. K. (2017). Next generation triaxial apparatus using combined computational-experimental testing framework, University of Illinois at Urbana-Champaign.

ASTM-International (2015). "D4015-15e1 Standard Test Methods for Modulus and Damping of Soils by Fixed-Base Resonant Column Devices."

Azzam, R., Fernandez-Steeger, T., Arnhardt, C., Klapperich, H. and Shou, K.-J. (2011). Monitoring of landslides and infrastructures with wireless sensor networks in an earthquake environment. Memorias de la conferencia 5th International Conference on Earthquake Geotechnical Engineering. Chile: International Society of Soil Mechanics and Geotechnical Engineering.

Baltaji, O., Numanoglu, O., Veeraraghavan, S., Hashash, Y. M. A., Coleman, J. L. and Bolisetti, C. (2017). Non-linear time domain site response and soil structure analysis for nuclear facilities using MOOSE. 24th International Conference on Structural Mechanics in Reactor Technology (SMiRT 24). Busan, South Korea.

Baltaji, O., Yang, S., Hashash, Y. M. and Singer, A. (2019). Through-Soil Wireless Communication System for Embedded Geotechnical Instrumentation. Geo-Congress 2019: Soil Erosion, Underground Engineering, and Risk Assessment, American Society of Civil Engineers Reston, VA.

Basarah, Y. I., Numanoglu, O. A., Hashash, Y. M. and Dashti, S. (2019). Impact of Hysteretic Damping on Nonlinear Dynamic Soil-Underground Structure-Structure Interaction Analyses. Geo-Congress 2019: Earthquake Engineering and Soil Dynamics, American Society of Civil Engineers Reston, VA.

Beadman, D., Bailey, R., Roberts, T., Welman, S., Thurlow, P. and Fauschou, M. (2001). "The Copenhagen Metro—observational method at Norreport station." Proceedings of the Institution of Civil Engineers-Geotechnical Engineering 149(4): 231-236.

Bennett, S. (1996). "A brief history of automatic control." IEEE Control Systems Magazine 16(3): 17-25.

Bennett, W. (1954). "Sources and properties of electrical noise." Electrical Engineering 73(11): 1001-1006.

Benz, T., Vermeer, P. and Schwab, R. (2009). "A small-strain overlay model." International journal for numerical and analytical methods in geomechanics 33(1): 25-44.

Brekhovskikh, L. M. and Godin, O. A. (2012). Acoustics of layered media I: Plane and quasi-plane waves, Springer Science & Business Media.

Brekhovskikh, L. M., Lysanov, Y. P. and Beyer, R. T. (1991). Fundamentals of ocean acoustics, Acoustical Society of America.

Brunson, B. A. (1983). "A comparison between Biot model predictions and shear wave attenuation measurements in unconsolidated laboratory sediments." The Journal of the Acoustical Society of America 74(S1): S59-S59.

Bullen, K. E., Bullen, K. E., Bullen, K. A. and Bolt, B. A. (1985). An introduction to the theory of seismology, Cambridge university press.

BureauofReclamation (2014). Embankment Dams. Instrumentation and Monitoring, Bureau of Reclamation.

Butkovich, J. N. and Hashash, Y. M. A. (2005). Neural Network Material Model Enhancement: Optimization Through Visualization and Selective Data Removal. Geo-Frontiers 2005, Austin, TX.

Chapman, T. and Green, G. (2004). Observational method looks set to cut city building costs. Proceedings of the Institution of Civil Engineers-Civil Engineering, Thomas Telford Ltd.

Choa, V. (1994). "Application of the observational method to hydraulic fill reclamation projects." Geotechnique 44(4): 735-745.

Christian, J. T. (2004). "Geotechnical engineering reliability: How well do we know what we are doing?" Journal of geotechnical and geoenvironmental engineering 130(10): 985-1003.

Cocroft, R. B. and Rodríguez, R. L. (2005). "The behavioral ecology of insect vibrational communication." Bioscience 55(4): 323-334.

Darendeli, M. B. (2001). Development of a new family of normalized modulus reduction and material damping curves Ph. D., University of Texas at Austin.

Doyle, J. C., Francis, B. A. and Tannenbaum, A. R. (2013). Feedback control theory, Courier Corporation.

Dunnicliff, J. (1996). Geotechnical instrumentation for monitoring field performance, John Wiley and Sons.

Finno, R. J. and Calvello, M. (2005). "Supported excavations: observational method and inverse modeling." Journal of Geotechnical and Geoenvironmental Engineering 131(7): 826-836.

Fleming, K. L., Hashash, Y. M., McLandrich, S., O'Riordan, N. and Riemer, M. (2016). "Novel Technologies for Deep-Excavation Digital Construction Records." Practice Periodical on Structural Design and Construction 21(4): 05016002.

Ghaboussi, J. (2010). Advances in neural networks in computational mechanics and engineering. Advances of soft computing in engineering, Springer: 191-236.

Ghaboussi, J., Garrett, J. H. and Wu, X. (1991). "Knowledge-based modeling of material behaviour with neural networks." Journal of Engineering Mechanics Division 117(1): 132-153.

Ghaboussi, J., Pecknold, D. A., Zhang, M. F. and Haj-Ali, R. M. (1998). "Autoprogressive training of neural network constitutive models." International Journal for Numerical Methods in Engineering 42(1): 105-126.

Ghaboussi, J. and Sidarta, D. E. (1998). "New nested adaptive neural networks (NANN) for constitutive modeling." Computers and Geotechnics 22(1): 29-52.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics.

Grami, A. (2015). Introduction to digital communications, Academic Press.

Groholski, D. (2012). Seismic site response analysis and extraction of dynamic soil behavior and pore pressure response from downhole array measurements, University of Illinois at Urbana-Champaign.

Groholski, D., Hashash, Y. M. A., Kim, B., Musgrove, M., Harmon, J. and Stewart, J. (2016). "Simplified Model for Small-Strain Nonlinearity and Strength in 1D Seismic Site Response Analysis." Journal of Geotechnical and Geoenvironmental Engineering 142(9): 04016042.

Groholski, D. R. and Hashash, Y. M. A. (2013). "Development of an inverse analysis framework for extracting dynamic soil behavior and pore pressure response from downhole array measurements." International Journal for Numerical and Analytical Methods in Geomechanics 37(12): 1867-1890.

Groholski, D. R., Hashash, Y. M. A. and Matasovic, N. (2014). "Learning of pore pressure response and dynamic soil behavior from downhole array measurements." Soil Dynamics and Earthquake Engineering 61–62(0): 40-56.

Hardin, B. O. and Drenevich, V. P. (1972). "Shear modulus and damping in soils: Measurement and parameter effects." Journal of Soil Mechanics and Foundation Engineering Division 98(SM6): 603-624.

Hashash, Y. M. and Asmar, R. K. (2018). "A New Triaxial Apparatus Imposing Nonuniform Shearing for Deep Learning of Soil Behavior." Geotechnical Testing Journal 42(3).

Hashash, Y. M. A. (2014). Innovations in Modeling and Monitoring Technologies for Response of Deep Urban Excavations. Geocongress. Atlanta, Georgia, ASCE.

Hashash, Y. M. A. and Finno, R. J. (2008). "Development of new integrated tools for predicting, monitoring and controlling ground movement due to excavations." ASCE Practice Periodical on Structural Design and Construction Volume 13(Issue 1): pp. 4-10.

Hashash, Y. M. A., Jung, S. and Ghaboussi, J. (2004). "Numerical implementation of a neural network based material model in finite element analysis." International Journal for Numerical Methods in Engineering 59(7): 989-1005.

Hashash, Y. M. A., Levasseur, S., Osouli, A., Finno, R. and Malecot, Y. (2010). "Comparison of two inverse analysis techniques for learning deep excavation response." Computer and Geotechnics 37(3): 323-333.

Hashash, Y. M. A., Marulanda, C., Ghaboussi, J. and Jung, S. (2003). "Systematic update of a deep excavation model using field performance data." Computers and Geotechnics 30(6): 477-488.

Hashash, Y. M. A., Marulanda, C., Ghaboussi, J. and Jung, S. (2006). "Novel approach to integration of numerical modeling and field observations for deep excavations." Journal of Geotechnical and Geoenvironmental Engineering 132(8): 1019 - 1031.

Hashash, Y. M. A., Musgrove, M. I., Harmon, J. A., Groholski, D., Phillips, C. A. and Park, D. (2016). DEEPSOIL V6.1, User Manual. Urbana, IL, Board of Trustees of University of Illinois at Urbana-Champaign.

Haupt, R. W. and Rolt, K. D. (2005). "Standoff acoustic laser technique to locate buried land mines." Lincoln Laboratory Journal 15(1): 3-22.

Hill, P. S. (2009). "How do animals use substrate-borne vibrations as an information source?" Naturwissenschaften 96(12): 1355-1371.

Hinton, G. (2012). "Neural Networks for MachineLearning - Lecture 6a - Overview of mini-batch gradi-ent descent."

Ikrath, K. and Schneider, W. (1968). "Communications via seismic waves employing 80-Hz resonant seismic transducers." IEEE Transactions on Communication Technology 16(3): 439-444.

Jamiolkowski, M. (2014). "Soil mechanics and the observational method: challenges at the Zelazny Most copper tailings disposal facility." Géotechnique 64(8): 590-618.

Jensen, F. B., Kuperman, W. A., Porter, M. B. and Schmidt, H. (2011). Computational ocean acoustics, Springer Science & Business Media.

Jung, S. and Ghaboussi, J. (2006). "Neural network constitutive model for rate-dependent materials." Computers & Structures 84(15-16): 955-963.

Kibblewhite, A. C. (1989). "Attenuation of sound in marine sediments: A review with emphasis on new low-frequency data." The Journal of the Acoustical Society of America 86(2): 716-738.

Kim, D.-S. and Lee, J.-S. (2000). "Propagation and attenuation characteristics of various ground vibrations." Soil Dynamics and Earthquake Engineering 19(2): 115-126.

Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S. and Turon, M. (2007). Health monitoring of civil infrastructures using wireless sensor networks. Proceedings of the 6th international conference on Information processing in sensor networks, ACM.

Kingma, D. P. and Ba, J. (2014). "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980.

Kinsler, L. E., Frey, A. R., Coppens, A. B. and Sanders, J. V. (1999). Fundamentals of acoustics, John wiley & sons.

Klein, K. and Santamarina, J. C. (1997). "Methods for broad-band dielectric permittivity measurements (soil-water mixtures, 5 Hz to 1.3 GHz)." Geotechnical Testing Journal 20(2): 168-178.

Koetter, R., Singer, A. C. and Tuchler, M. (2004). "Turbo equalization." IEEE signal processing magazine 21(1): 67-80.

Kojima, W., Takanashi, T. and Ishikawa, Y. (2012). "Vibratory communication in the soil: pupal signals deter larval intrusion in a group-living beetle Trypoxylus dichotoma." Behavioral ecology and sociobiology 66(2): 171-179.

Kovári, K. and Lunardi, P. (2000). On the observational method in tunnelling. ISRM International Symposium, International Society for Rock Mechanics and Rock Engineering.

Kramer, S. L. (1996). Geotechnical earthquake engineering. Upper Saddle River, N.J., Prentice Hall.

Langton, C. (2012). Intuitive Guide to Principles of Communication.

Lowrie, W. and Fichtner, A. (2020). Fundamentals of geophysics, Cambridge university press.

Lurton, X. (2002). An introduction to underwater acoustics: principles and applications, Springer Science & Business Media.

Marr, W. A. (2007). Why Monitor Performance? 7th FMGM 2007: Field Measurements in Geomechanics: 1-27.

Marulanda, C. (2005). Integration of numerical modeling and field observations of deep excavations Ph.D. Thesis, University of Illinois at Urbana-Champaign.

McLandrich, S. M., Hashash, Y. M. A. and O'Riordan, N. J. (2013). Networked Geotechnical near Real-Time Monitoring for Large Urban Excavation using Multiple Wireless Sensors. CONFERENCE TO COMMEMORATE THE LEGACY OF RALPH B. PECK, SEVENTH INTERNATIONAL CONFERENCE ON CASE HISTORIES IN GEOTECHNICAL ENGINEERING AND SYMPOSIUM IN HONOR OF CLYDE BAKER – APRIL 29-MAY 4, 2013. WHEELING, IL (CHICAGO, IL AREA)

Moon, S.-w. and Hashash, Y. M. (2015). "From direct simple shear test to soil model development and supported excavation simulation: integrated computational-experimental soil behavior characterization framework." Journal of Geotechnical and Geoenvironmental Engineering 141(11): 04015050.

Moritz, B. and Schubert, W. (2009). "Application of the observational method in geotechnical safety management." Geomechanics and Tunnelling 2(3): 269-281.

Narins, P. M., Reichman, O., Jarvis, J. U. and Lewis, E. R. (1992). "Seismic signal transmission between burrows of the Cape mole-rat, Georychus capensis." Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology 170(1): 13-21.

Numanoglu, O. A. (2019). Numerical modeling and simulation of seismic settlements on dense sands, University of Illinois at Urbana-Champaign.

O'Connell-Rodwell, C., Hart, L. and Arnason, B. (2001). "Exploring the potential use of seismic waves as a communication channel by elephants and other large mammals." American Zoologist 41(5): 1157-1170.

O'Connell-Rodwell, C. E. (2007). "Keeping an "ear" to the ground: seismic communication in elephants." Physiology 22(4): 287-294.

Oelze, M. L., O'Brien, W. D. and Darmody, R. G. (2002). "Measurement of attenuation and speed of sound in soils." Soil Science Society of America Journal 66(3): 788-796.

Osouli, A. (2010). The interplay between field measurements and soil behavior for learning supported excavation response, University of Illinois at Urbana-Champaign.

Osouli, A., Hashash, Y. M. and Song, H. (2009). "Interplay between field measurements and soil behavior for capturing supported excavation response." Journal of geotechnical and geoenvironmental engineering 136(1): 69-84.

Osouli, A. and Hashash, Y. M. A. (2010). "Case studies of prediction of excavation response using learned performance of excavations." International Journal of Geoengineering Case Histories 4(2).

Payne, R. and Webb, D. (1971). "Orientation by means of long range acoustic signaling in baleen whales." Annals of the New York Academy of Sciences 188(1): 110-141.

Peck, R. (2001). "The observational method can be simple." Proceedings of the Institution of Civil Engineers-Geotechnical Engineering 149(2): 71-74.

Peck, R. B. (1969). "Advantages and limitations of the observational method in applied soil mechanics." Geotechnique 19(2): 171-187.

Phillips, C. and Hashash, Y. M. A. (2008). A new simplified constitutive model to simultaneously match modulus reduction and damping soil curves for nonlinear site response analysis. Geotechnical Earthquake Engineering & Soil Dynamics IV (GEESD IV), Sacramento, California.

Phillips, C. and Hashash, Y. M. A. (2009). "Damping formulation for non-linear 1D site response analyses." Soil Dynamics and Earthquake Engineering 29(7): Pages 1143-1158.

Phoon, K.-K. (2017). "Role of reliability calculations in geotechnical design." Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards 11(1): 4-21.

Phoon, K.-K. and Kulhawy, F. H. (1999a). "Characterization of geotechnical variability." Canadian geotechnical journal 36(4): 612-624.

Phoon, K.-K. and Kulhawy, F. H. (1999b). "Evaluation of geotechnical property variability." Canadian Geotechnical Journal 36(4): 625-639.

Powderham, A. (2004). "The observational method–learning from projects."

Powderham, A. J. (1994). "An overview of the observational method: Development in cut and cover and bored tunnelling projects." Geotechnique 44(4): 619-636.

Prästings, A., Müller, R. and Larsson, S. (2014). "The observational method applied to a high embankment founded on sulphide clay." Engineering geology 181: 112-123.

Rappaport, T. S. (1996). Wireless communications: principles and practice, prentice hall PTR New Jersey.

Riedl, T. and Singer, A. (2013). MUST-READ: Multichannel sample-by-sample turbo resampling equalization and decoding. OCEANS-Bergen, 2013 MTS/IEEE, IEEE.

Riedl, T. and Singer, A. (2014). Towards a video-capable wireless underwater modem: Doppler tolerant broadband acoustic communication. Underwater Communications and Networking (UComms), 2014, IEEE.

Riedmiller, M. and Braun, H. (1993). A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. IEEE International Conference on Neural Networks, San Francisco, IEEE, New York.

Ritsema, C. J., Kuipers, H., Kleiboer, L., Van Den Elsen, E., Oostindie, K., Wesseling, J. G., Wolthuis, J. W. and Havinga, P. (2009). "A new wireless underground network system for continuous monitoring of soil water contents." Water resources research 45(4).

Robinson, E. S. (1988). "Basic exploration geophysics."

Ruder, S. (2016). "An overview of gradient descent optimization algorithms." arXiv preprint arXiv:1609.04747.

Runesson, K., Steinmann, P., Ekh, M. and Menzel, A. (2006). "Constitutive modeling of engineering materials—theory and computation." Volume I General Concepts and Inelasticity.

Sayed, A. H. (2003). Fundamentals of adaptive filtering, John Wiley & Sons.

Schanz, T., Vermeer, P. and Bonnier, P. (1999). "The hardening soil model: formulation and verification." Beyond 2000 in computational geotechnics: 281-296.

Schubert, W. (2008). "The development of the observational method." Geomechanik und Tunnelbau: Geomechanik und Tunnelbau 1(5): 352-357.

Shearer, P. M. (2019). Introduction to seismology, Cambridge university press.

Sheriff, R. E. and Geldart, L. P. (1995). Exploration seismology, Cambridge university press.

Sidarta, D. E. and Ghaboussi, J. (1998). "Constitutive Modeling of Geomaterials from Non-Uniform Material Tests." International Journal of Computers and Geotechnics 22(1): 53-71.

Silva, A. R. and Vuran, M. C. (2009). Empirical Evaluation of Wireless Underground-to-Underground Communication in Wireless Underground Sensor Networks. DCOSS, Springer.

Silva, A. R. and Vuran, M. C. (2010). Communication with aboveground devices in wireless underground sensor networks: An empirical study. Communications (ICC), 2010 IEEE International Conference on, IEEE.

Sklar, B. and Harris, F. J. (1988). Digital communications: fundamentals and applications, Prentice-hall Englewood Cliffs, NJ.

Slabbekoorn, H., Dooling, R. J., Popper, A. N. and Fay, R. R. (2018). Effects of anthropogenic noise on animals, Springer.

Song, H. and Hashash, Y. M. A. (2009). "Characterization of stress-strain behavior of red blood cells from measured deformation response: Response for malaria infected red blood cells." submitted to Journal of Biomechanics.

Spross, J. (2014). A critical review of the observational method, KTH Royal Institute of Technology.

Spross, J., Johansson, F., Uotinen, L. K. T. and Rafi, J. Y. (2016). "Using observational method to manage safety aspects of remedial grouting of concrete dam foundations." Geotechnical and Geological Engineering 34(5): 1613-1630.

Stuntebeck, E. P., Pompili, D. and Melodia, T. (2006). Wireless underground sensor networks using commodity terrestrial motes. 2006 2nd IEEE Workshop on Wireless Mesh Networks, IEEE.

Sun, Z. and Akyildiz, I. F. (2010). "Magnetic induction communications for wireless underground sensor networks." IEEE Transactions on Antennas and Propagation 58(7): 2426-2435.

Tan, X., Sun, Z. and Akyildiz, I. F. (2015). "A testbed of magnetic induction-based communication system for underground applications." arXiv preprint arXiv:1503.02519.

Teal, D. (1990). Cordless geo data-link. Geotechnical instrumentation in practice: Purpose, performance and interpretation, Thomas Telford Publishing: 617-628.

Telford, W. M., Telford, W., Geldart, L., Sheriff, R. E. and Sheriff, R. E. (1990). Applied geophysics, Cambridge university press.

Tsai, C.-C. and Hashash, Y. M. A. (2008). "A novel framework integrating downhole array data and site response analysis to extract dynamic soil behavior." Soil Dynamics and Earthquake Engineering 28(3): 181-197.

Vasquez, J., Rodriguez, V. and Reagor, D. (2004). "Underground wireless communications using high-temperature superconducting receivers." IEEE Transactions on applied superconductivity 14(1): 46-53.

Vucetic, M. and Dobry, R. (1991). "Effect of soil plasticity on cyclic response." Journal of Geotechnical Engineering 117(1): 87-107.

Vuran, M. C. and Akyildiz, I. F. (2007). "A-MAC: Adaptive medium access control for next generation wireless terminals." IEEE/ACM Transactions on Networking 15(3): 574-587.

Wencai, Y. (2013). Reflection Seismology: theory, data processing and interpretation, Elsevier.

Whittle, A. J. and Kavvadas, M. J. (1994). "Formulation of MIT-E3 constitutive model for overconsolidated clays." Journal of Geotechnical engineering 120(1): 173-198.

Wu, T. H. (2011). "2008 Peck lecture: the observational method: case history and models." Journal of Geotechnical and Geoenvironmental Engineering 137(10): 862-873.

Wu, X. and Ghaboussi, J. (1995). Neural network-based material modeling, University of Illinois Engineering Experiment Station. College of ….

Yang, S., Baltaji, O., Singer, A. C. and Hashash, Y. M. (2019). "Development of an Underground Through-Soil Wireless Acoustic Communication System." IEEE Wireless Communications 27(1): 154-161.

Yeow, H.-C., Nicholson, D., Man, C.-L., Ringer, A., Glass, P. and Black, M. (2014). "Application of observational method at Crossrail Tottenham Court Road station, UK." Proceedings of the Institution of Civil Engineers-Geotechnical Engineering 167(2): 182-193.

Appendix A: Additional Photos for the SoilComm Research Project



Figure A-1: Photos of some of the laboratory tests conducted on field soil specimens



Figure A-2: Photos of data processing and a test on saturated sand in the lab



Figure A-3: Photos of distance measurements and boreholes digging using a motorized augur.



Figure A-4: UAV photos of the second testing site.

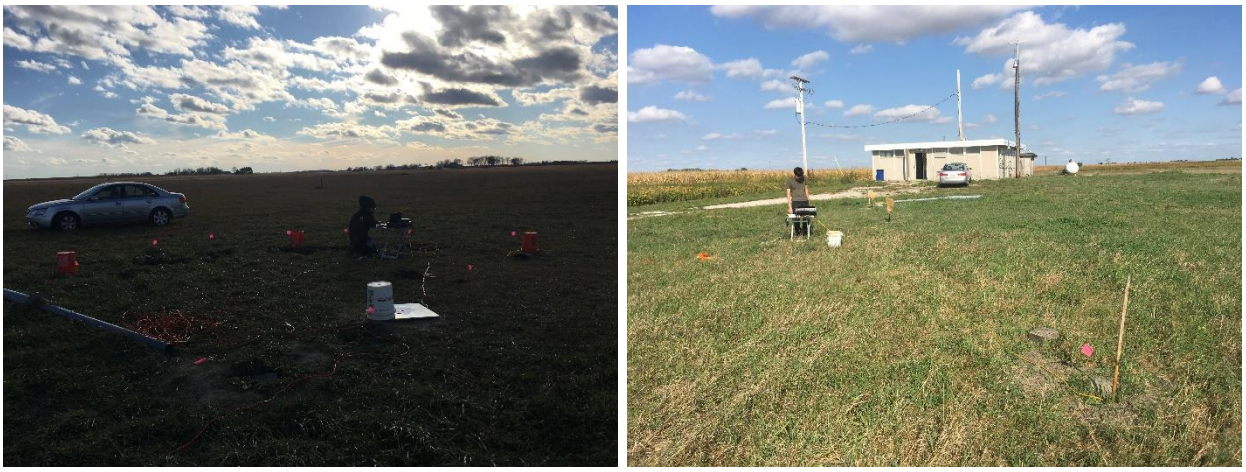


Figure A-5: photo of a test to check the directivity of the mechanical source and another to test medium range communication

Appendix B: New SelfSim Manual

The goal of this manual is to show how a SelfSim analysis is set up and run using SelfSim Gen 3.

Step 1: Setting the Environment

The first step for running a SelfSim analysis is to setup the environment.

- Python v3.7 should be installed with the following libraries:
    - TensorFlow v2
    - Numpy
    - Scipy
    - Pandas
    - Matplotlib
    - Sklearn

Some python IDEs (such as Anaconda and PyCharm), include most of those libraries (except TensorFlow), by default.

The SelfSim python code (most importantly the "Selfim" and the "Functions" modules) should be acquired and placed in a folder selected by the user.

- LS-DYNA v11 should be installed. The SelfSim executable should be placed in the directory containing default LS-DYNA executables (usually C:\LSDYNA\program) then selected as the solver for LS-DYNA.
- A folder should be created for the analysis. This folder should contain two subfolders called "dc" and "lc" and one python file called "SSInput.py". The contents of those folders and file are described in Step2.

Step 2: Preparing the FEM Analysis

The second step is to prepare the FEM analysis file. This is performed just like any other analysis. However, the loading boundary conditions to be used in the load-controlled analysis of SelfSim should be placed in a file called "LoadBC.k" while the displacements to be used in the displacement-controlled analysis of SelfSim should be placed in a file called "DisplacementBC.k". This can be done by applying all the loads and displacements when preparing the analysis initial keyword file then using a text editor (such as Notepad ++) to cut the keywords of those loads and displacements then to paste the keywords for the load boundary conditions in the "LoadBC.k" file and the keywords for the displacement boundary conditions in the "DisplacementBC.k" file. Those

latter files are ordinary text files that are saved with a ".k" extension (also using Notepad++ for example).

The FEM keyword file is then copied to two different folders: called "dc" and "lc" placed in the folder of the analysis. In the "dc" folder, the "DisplacementBC.k" is included while in the "lc" folder, the "LoadBC.k" is included. In the FEM keyword file placed in the "dc" folder, the following two lines are added:

*INCLUDE

DisplacementBC.k

While in the FEM keyword file placed in the "lc" folder, the following two lines are added:

*INCLUDE

LoadBC.k

The preparation of the "LoadBC.k" is typically easy but problem specific, while the preparation of the "DisplacementBC.k" can be challenging if plenty of displacements are to be included. To facilitate the preparation of this latter filer, a function called "GenerateDisplacementBCKeyword" was included in the "Functions" module. This function takes a csv file that has the following format:

node # 1	x-displacement at t = 0	y-displacement at t = 0	z-displacement at t = 0	x-displacement at t = 1	y-displacement at t = 1	z-displacement at t = 1	…
node # 2	x-displacement at t = 0	y-displacement at t = 0	z-displacement at t = 0	x-displacement at t = 1	y-displacement at t = 1	z-displacement at t = 1	
…	…	…	…	…	…	….	..

where a node number is the number of the node at which the corresponding displacement boundary conditions are to be applied. If a displacement in a certain node is not known or the user doesn't want to include it (for example if only x and y displacements are to be given to SelfSim), an "N/A" value should be given for that displacement.

This function can be run from an external Python shell, and it will generate the "DisplacementBC.k" file which is then copied to the "dc" folder.

The ANN keywords are not generated at the beginning of the analysis (they will be generated at the end of the pretraining), but the analysis should contain the names of the files that will be generated. For example, if the MaterialID to be used for the ANN material is to be 820 (should be a number), the following keywords should be included in the analysis keyword file:

*INCLUDE

820_Keyword.k

Step 3: Selecting SelfSim Parameters

The third step is to select the SelfSim parameters which will control the SelfSim analyses. Those parameters are included in the SSInput.py file. Following is the list of parameters needed:

Name	Description	Example
ANN Architecture Parameters		
Dimension	The dimension of the problem. Options are '3D', '2DPStrain', '2DPStress', '1D'. The current version of SelfSim is fully prepared and tested for the '3D' case. Other dimensions might be supported in future releases.	Dimension = '3D'
NumPathPoints	includes the current point and the history points	NumPathPoints = 1
NumNodes	number of nodes in hidden layers only	NumNodes = [30,30]
ActivationFunctionIndeces	activation functions for hidden layers and output layer (it should contain one entry more than the NumNodes list). The activation functions available are: 1: ReLU; 2: sigmoid; 3: tanh; 4: SELU	ActivationFunctionIndeces = [3,3,3]

UseBiases	A boolean flag that controls whether biases should be used in the artificial neural network	UseBiases = False
UseExistingNNCurveFile	A boolean flag to tell SelfSim to use an existing ANN model rather than constructing a new one via the parameters above.	UseExistingNNCurveFile = False
ExistingNNCurveFileName	Name of the existing ANN model text file. The UseExistingNNCurveFile Flag must be set to True to use this ANN model.	ExistingNNCurveFileName = 'ANN_model_23.txt'
Linear Elastic Pretraining Parameters		
E	The modulus of elasticity of the linear elastic dataset which will be autogenerated then used for ANN pretraining. It should have the same unit as the unit of stress used in the FEM analyses. A good estimate of E would enhance and accelerate rating. It should be at least close in its order of magnitude to actual material moduli.	E = 6000
v	The poisson's ratio of the linear elastic dataset which will be autogenerated then used for ANN pretraining	v = 0.2
LowStrain	The lowest strain in the elastic pretraining dataset	LowStrain = 1e-10
HighStrain	The highest strain in the elastic pretraining dataset	HighStrain = 1e-4

Size	The number of samples to be generated in the autogenerated elastic pretraining dataset	Size = 5000
Parameters of UMAT keyword		
MaterialID	An ID number given by the user for the ANN material model	MaterialID = 820
Density	The density of the material modeled by the ANN model. It should have consistent units with the FEM analyses	Density = 2e-5
InitialBulkModulus	The initial bulk modulus of the material. This is required by LS-DYNA for timestep calculation. It is set by default to E/(3*(1-2*v))	InitialBulkModulus = E/(3*(1-2*v))
InitialShearModulus	The initial shear modulus of the material. This is required by LS-DYNA for timestep calculation. It is set by default to E/(2*(1+v))	InitialShearModulus = E/(2*(1+v))
ANN Training Parameters		

LearningRate	This is the learning rate used by the optimizer. The ADAM optimizer adjusts the learning rate dynamically, so when using it, no need to provide this number.	LearningRate = " # when using ADAM optimizer
		LearningRate = 1e-3 # when using other optimizers

BatchSize	The training dataset is divided into batches each with a size (number of points) equal to BatchSize. The optimizer optimizes the ANN by calculating the error of the first batch then the second then the third, etc. If all the points are to be used at	BatchSize = 'All' BatchSize = 64

	once for optimization, the BatchSize can be set to ‘All’. The number of batches = Data Size/BatchSize	
MinNumEpochs	The minimum number of epochs of training. In one epoch, all training data is used once. Number of total optimizations = number of epochs x number of batches.	MinNumEpochs = 50
MaxNumEpochs	The maximum number or epochs of training. The training stops at this number of epochs even if the target error didn’t drop below the average epoch error tolerance.	MaxNumEpochs = 50
EpochAvgTolerance	The tolerated average error of all data points in an epoch. If the average error drops below this number, the training stops.	EpochAvgTolerance = 1e-15
LossFunctionType	The loss function used to calculate the error, which is then reduced by the optimizer. Many loss functions are available including the mean absolue error ‘MAE’, the mean square error ‘MSE’, etc.	LossFunctionType = 'MAE'
OptimizerType	The optimizer that is used to minimize the error defined by the loss function. Many optimizer types are available including the ‘ADAM’ optimizer, the ‘RMSProp” optimizer, the ‘RProp’ optimizer, the gradient decent optimizer ‘GD’, etc.	OptimizerType = 'ADAM'

DisplayStep	The number of epochs after which the training statistics (for example current error) are displayed.	DisplayStep = 50
UMAT Controllers		
SymmetrizeStiffnessMatrix	A flag to instruct the UMAT in LS-DYNA to symmetrize the stiffness matrix. This is an experimental feature and should be not used. (SymmetrizeStiffnessMatrix = False)	SymmetrizeStiffnessMatrix = False
CalculateIncrements	A flag that tells the UMAT whether the ANN provided is outputting a total stress or an increment stress. If CalculateIncrements is True, then the ANN is outputting stress increments.	CalculateIncrements = False
DeductInitialStresses	A flag that tells the UMAT whether the ANN was trained with a dataset in which the initial stresses were deducted from all stresses. This usually results in better training because it expands the range available for training.	DeductInitialStresses = True
UniformScaling	Tells SelfSim whether to scale all stresses using one scalar and all the strains using another one scalar. Using a uniform scaling is usually more robust.	UniformScaling = True
NNType	The type of the ANN being used. Types include: fully connected network ‘FC’ and ‘Nested’	

Data Scaling Parameters		
FinalScalars	A list containing the scalars to be applied to the strain and stress data sets. The list should be as long as the training dataset. For example, an ANN architecture with one path point has a training data set that looks like: previous strains (6), previous stresses (6), current strains (6), current stresses (6).	[7,7,7,7,7,7, 6.0e-3,6.0e-3,6.0e-3,6.0e-3,6.0e-3,6.0e-3 ,7,7,7,7,7,7 ,6.0e-3,6.0e-3,6.0e-3,6.0e-3,6.0e-3,6.0e-3,]
MinReductionFactor	defines the minimum reduction factor on an automatic scaling. Automatic scaling (if FinalScalars are not provided) is adjusted to scale the maximum strain to 1 and scale maximum stress to 1. To prevent saturation, a number lower than 1 is usually used, and this number is controlled via the reduction factor. The analysis starts with the MinReductionFactor and then adds the ReductionFactorIncrement after each step till the MaxReducionFactor is reached. This way of controlling scaling is an experimental feature and should not be used. FinalScalars should be directly defined instead.	
MaxReductionFactor	maximum reduction factor for automatic scaling (see MinReductionFactor)	

ReductionFactorIncrement	reduction factor increment for automatic scaling (see MinReductionFactor)	
The Autoprogressive algorithm parameters		
NumPasses	The maximum number of passes to be used. If the target error between the load-controlled and the displacement-controlled analyses is reached in less passes, the SelfSim analysis is stopped before.	NumPasses = 2
NumCycles	The maximum number of cycles to be used. If the target error between the load controlled and the displacement controlled analyses is reached in less cycles, the SelfSim analysis moves to the next step	NumCycles = 3
MaxDisplacementTolerance	The normalized maximum tolerated displacement error between the load-controlled and the displacement-controlled analyses	MaxDisplacementTolerance = 0.05
AvgDisplacementTolerance	The normalized average tolerated displacement error between the load-controlled and the displacement-controlled analyses	AvgDisplacementTolerance = 0.03
TrainingWindow	The number of previous steps to be included in the training.	TrainingWindow = 7
MaxRepititions	The number of repetitions allowed if the analysis fails to converge.	MaxRepititions = 2
ModeOfOperation	The mode of operation of the SelfSim analysis: # 1: run all steps then train	ModeOfOperation = 3

	# 2: run cycle then learn then run from beginning and move to the next cycle # 3: run step then learn, then move to the next step # 4: run step then learn then run from beginning and move to the next step	
UpdateHistoryAfterStep	A flag to tell the software to update the previously calculated training data points for all steps after each new step using the latest FEM results. This flag works only with ModeOfOperation = 4	UpdateHistoryAfterStep = True
FEM Analysis Parameters		
NumFEMSteps	Number of steps in the FEM analysis	NumFEMSteps = 7
NumFEMElements	Number of elements in the FEM analysis	NumFEMElements = 912
NumFEMNodes	Number of nodes in the FEM analysis	NumFEMNodes = 634
TimeStep	The timestep used in the analysis. For static analyses, use TimeStep = 1.	TimeStep = 1
MonitoredNodes	The IDs of nodes used to prepare problem specific plots.	MonitoredNodes2 = [12090,212090,412090,612090,812090,1012090,1212090, 1412090,1612090,1812090,2012090,2212090,2412090]

InitialStresses	The initial stresses of the material under consideration. This is used if the initial stresses are to be deducted using DeductInitialStresses = True	InitialStresses = [-3.497,-3.497,-8.800,0,0,0]
InputFileName	The name of the input file name without the "_dc" or "_lc"	InputFileName = "ap_abaq21nc3"
Memory	The RAM to be used in each FEM analysis (in mega byte)	Memory = 100
NCPU	The number of CPUs to be used in each FEM analysis	NCPU = 1
PathToSolver	The directory to the LS-DYNA solver to be used in the analyses	PathToSolver = r"C:\LSDYNA\program\lsdyna-05-17-2020.exe"
Additional Controllers		
TFVersion	Tells SelfSim which version of the Tensorflow to use (SelfSim functions are defined for both TFv1 and TFv2)	TFVersion = 2
Extra Features		
SavetoCSV	A flag to tell SelfSim to store processing results on CSV files. This feature was used to inspect the output of functions. It is recommended to turn off for typical SelfSim analyses to save time and storage.	SavetoCSV = False
ReinitializeNNAfterFailure	A flag to tell SelfSim whether to reinitialize the ANN if one of both of the FEM analyses fail to converge. The reinitialization is	ReinitializeNNAfterFailure = True

	done simply by discarding the old ANN model then pretraining a new one using the autogenerated linear elastic dataset	
BalanceClip	The limit on the number of copies of data of a certain step when balancing data.	BalanceClip = 5

Step 4: Running the SelfSim Analysis

The last step is to run the SelfSim analysis. Open the "SelfSim" module. Adjust the following three entries:

```
PathToSolver = r"C:\LSDYNA\program\lsdyna-05-17-2020.exe"
PathToMainFolder = r"C:\ \Desktop\LS-DYNA\SelfSim\2-XY\\"
Analyses = ['ap_abaq21nc1', 'ap_abaq21nc2', 'ap_abaq21nc3',
        'ap_abaq11nc1', 'ap_abaq11nc2', 'ap_abaq11nc3',
        'ap_abaq21c1', 'ap_abaq21c2', 'ap_abaq21c3',
        'ap_abaq11c1', 'ap_abaq11c2', 'ap_abaq11c3',]
```

where PathToSolver is the path to the solver – which has the ANN UMAT implementation – which should be placed in the same directory of the other common LS-DYNA solvers). PathToMainFolder is the path of the mother folder that contains the folders of the SelfSim analyses. This folder may contain one or more analysis folders (this is useful if the user wants to run multiple SelfSim analyses). Analyses is a list of the SelfSim analyses to be run. Each of the names included in the list should be a name of a folder included in the mother folder. Each of these folders will contain the 'lc' and 'dc' subfolders in addition to the "SSInput.py" file as described above.

Run the SelfSim code, and the SelfSim analysis (or analyses) should execute.

The SelfSim analysis will run till the given number of passes are completed or till the error drops below the determined tolerances. The plots generated are saved in the "lc" folder and can be explored there during and after the SelfSim analysis. The definition of the trained ANN model can be found in the MaterialId_Keyword.k and MaterialId_Parameters.k files. They can be used for forward analysis as needed.

Example of a SelfSim analysis performed for a Triaxial Test Problem

The CT3 test described in the thesis is used as an example in this tutorial. A finite element model was prepared for this test. The analysis folder is called ap_abaq21nc3. Inside this folder, two subfolders are found: ‘lc’ and ‘dc’ in addition to a file called ‘SSInput.py’. Inside the ‘lc’ folder there is the ‘ap_abaq21nc3_lc.k’ file and the ‘LoadBC.k’ file while in the ‘dc’ folder, there is the ‘ap_abaq21nc3_dc.k’ file and the ‘DisplacementBC.k’ file. The analysis folder is zipped and attached to this document.

The “SSInput.py” file is a simple “.py” file that contains the analysis parameters described in the above table. It looks like:

“Dimension = '3D'
NumPathPoints = 1
NumNodes = [30,30]
ActivationFunctionIndeces = [3,3,3]
UseBiases = False
…”

The “LoadBC.k” file looks like:

“*LOAD_RIGID_BODY
$# PID DOF LCID SF CID M1 M2 M3
1,3,5,1
*DEFINE_CURVE_TITLE
Axial Load
$# lcid sidr sfa sfo offa offo dattyp
5 0 1.000000 1.000000 0.000 0.000 0
$# a1 o1
0.000 -1428.0
1.000 -1656.13
2.000 -1876.26
3.000 -2004.44
4.000 -2039.30
5.000 -2007.95
6.000 -1950.21

7.000 -1897.35”

Note that the first column is the step number while the second column is the load applied in this step.

The “DisplacementBC.k” file looks like:

“*BOUNDARY_PRESCRIBED_MOTION_NODE
$# nid dof vad lcid sf vid death birth
112010,1,2,1120101,1,10000,0
*DEFINE_CURVE
$# lcid sidr sfa sfo offa offo dattyp
1120101,0,1,1,0,0,0
$# a1 o1
1,0.00028684
2,0.00122070
3,0.00303873
4,0.00563822
5,0.00939742
6,0.01331601
7,0.01590897
…”

The above keyword block defines the displacement boundary condition at the 7 steps of the analysis in one direction (x-direction) for one node (node 112010).

The SelfSim module is configured as described above, then the analysis is started. Running on a personal desktop computer, the analysis finished in 471 seconds. During the analysis, the screen shows updates on the training. All those updates are stored in the ‘log’ test file created by the software and placed in the analysis folder. At the end of the analysis, output figures can be found in the ‘lc’ folder, including figures of intermediate steps and figures at the end of the analysis.

The final ANN files generated by the analysis cane be found in the ‘lc’ or ‘dc’ folders. They are named “MaerialID_Keyword.k” and “MaterialID_Parameters.k”.

For demonstration of results, those two files are copied then used in a forward analysis (in a separate folder) in which only the load boundary conditions are applied, just like the analysis in

the ‘lc’ folder. All steps of the analyses are run without any training. The results are compared to deformations measured in the actual test. The results are plotted in the figures below.

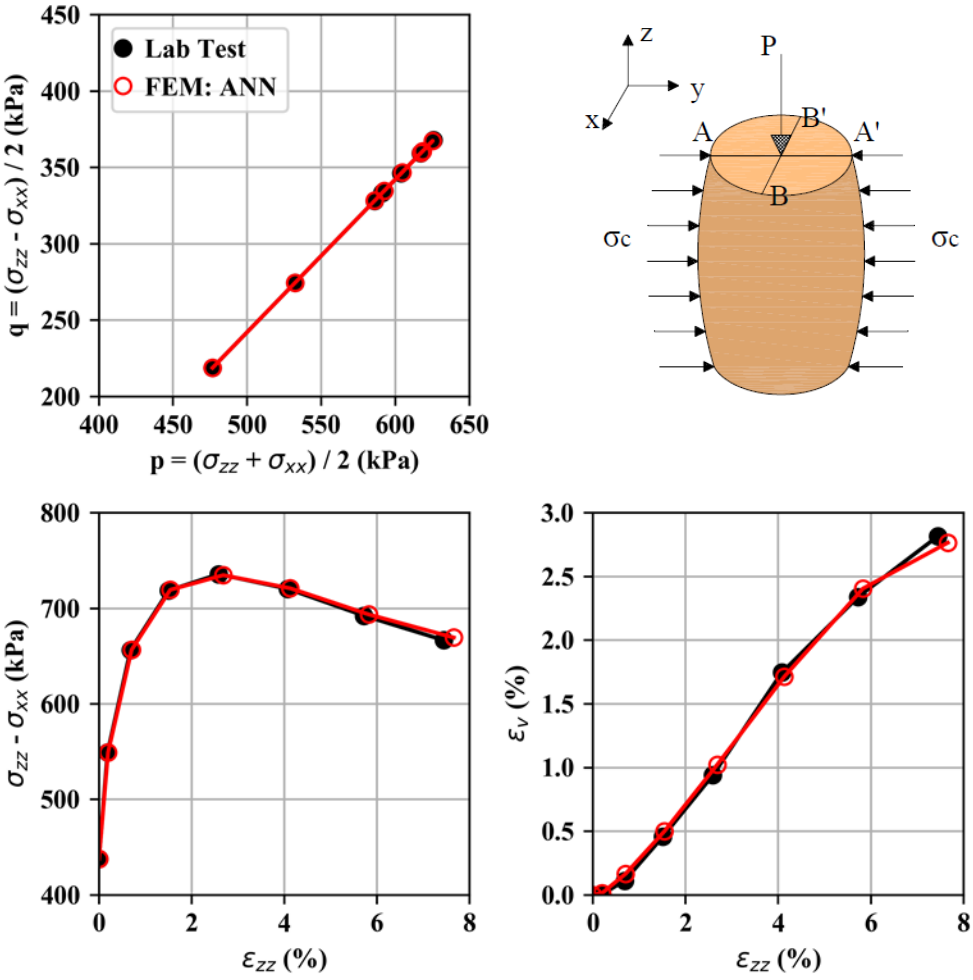


Figure B-1: Comparison of the stress path, deviatoric stress vs axial strain, and volumetric strain vs axial strain for the CT3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

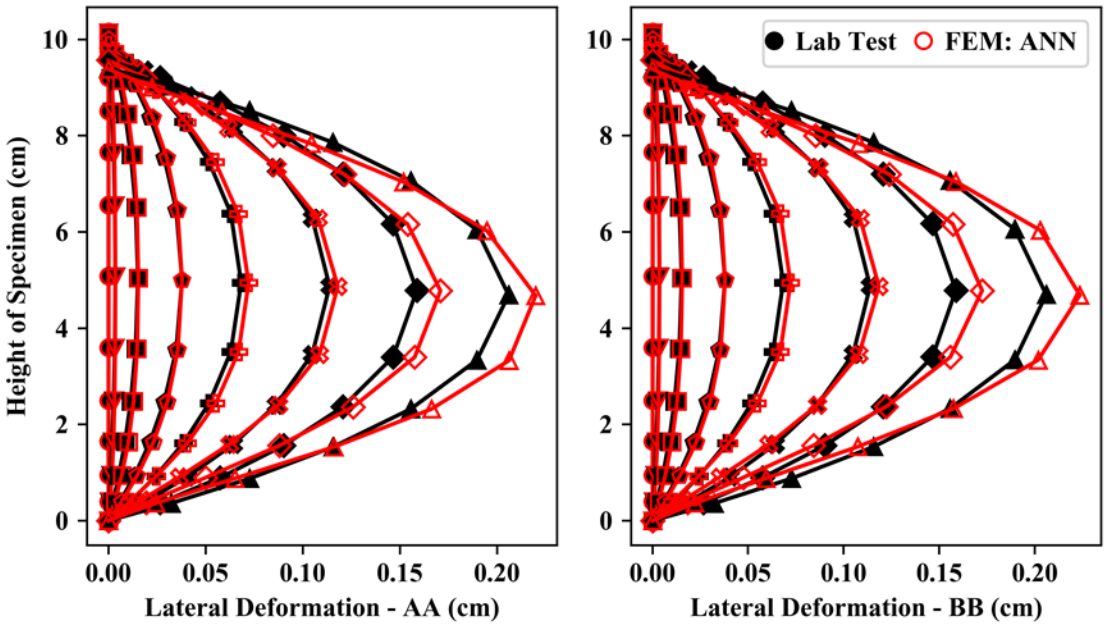


Figure B-2: Comparison of the deformations measurements of the CT3 lab test vs those calculated by the FEM analysis employing the ANN model trained through SelfSim

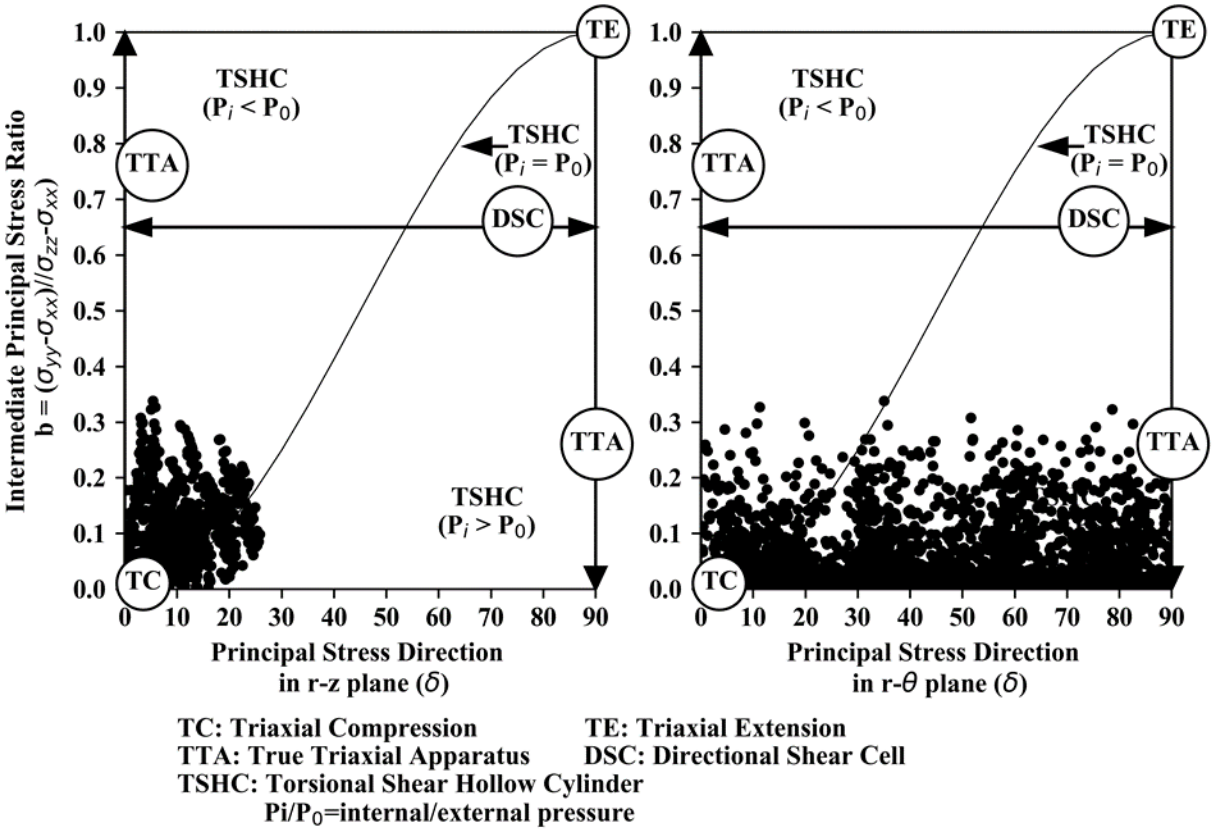


Figure B-3: Plot of the stress space calculated by the FEM analysis employing the ANN model trained through SelfSim performed for the CT3 lab test.

Appendix C: Effect of LS-DYNA Element Formulations on FEM Results

LS-DYNA provides tens of element formulations suitable for different applications. For the ANN material model, one of the most important required features for an element to have is to support non-symmetric stiffness matrices. This is crucial because the ANN stiffness matrices are often non-symmetric.

In LS-DYNA, 3 brick elements support the non-symmetric stiffness matrices: "2": fully integrated or selectively reduced integrated solid, "-1": fully integrated or selectively reduced integrated solid for elements with poor aspect ratio – efficient formulation and "-2" fully integrated or selectively reduced integrated solid for elements with poor aspect ratio – accurate formulation. The manual states the following: "*Solid formulations -1 and -2 may offer improved behavior over formulation 2 by accounting for poor element aspect ratios in a manner so as to reduce the transverse shear locking effects seen with formulation 2. Type -1 is a more computationally efficient implementation of type -2, but a side effect is that type -1's resistance to a particular deformation mode, similar to an hourglass mode, is weakened. This side effect is not truly hourglassing behavior, so there is no hourglass energy and the behavior is not affected by hourglass parameters.*"

Originally element type "2" was used which resulted in some noise especially at low strains. For example, for the forward analysis conducted for the linear elastic model for the conventional triaxial test model, although the ANN model was trained with the correct strains and stress calculated using the linear elastic model, the FEM analysis employing this ANN will always show noise in the xy strain and stress characterized by the least magnitude among the other strains and stresses as shown in the figures below. Using the element type"-2" resulted in the elimination of noise as was shown previously in Figure 7-5 and Figure 7-6. Similar improvement of the results were noticed for the other tests. This comes at an additional computation cost but is justified because of the enhanced accuracy.

It is hoped that in the future, LS-DYNA will extend the support for non-symmetric stiffness matrices for more of the available sections especially the 20-node section 23.

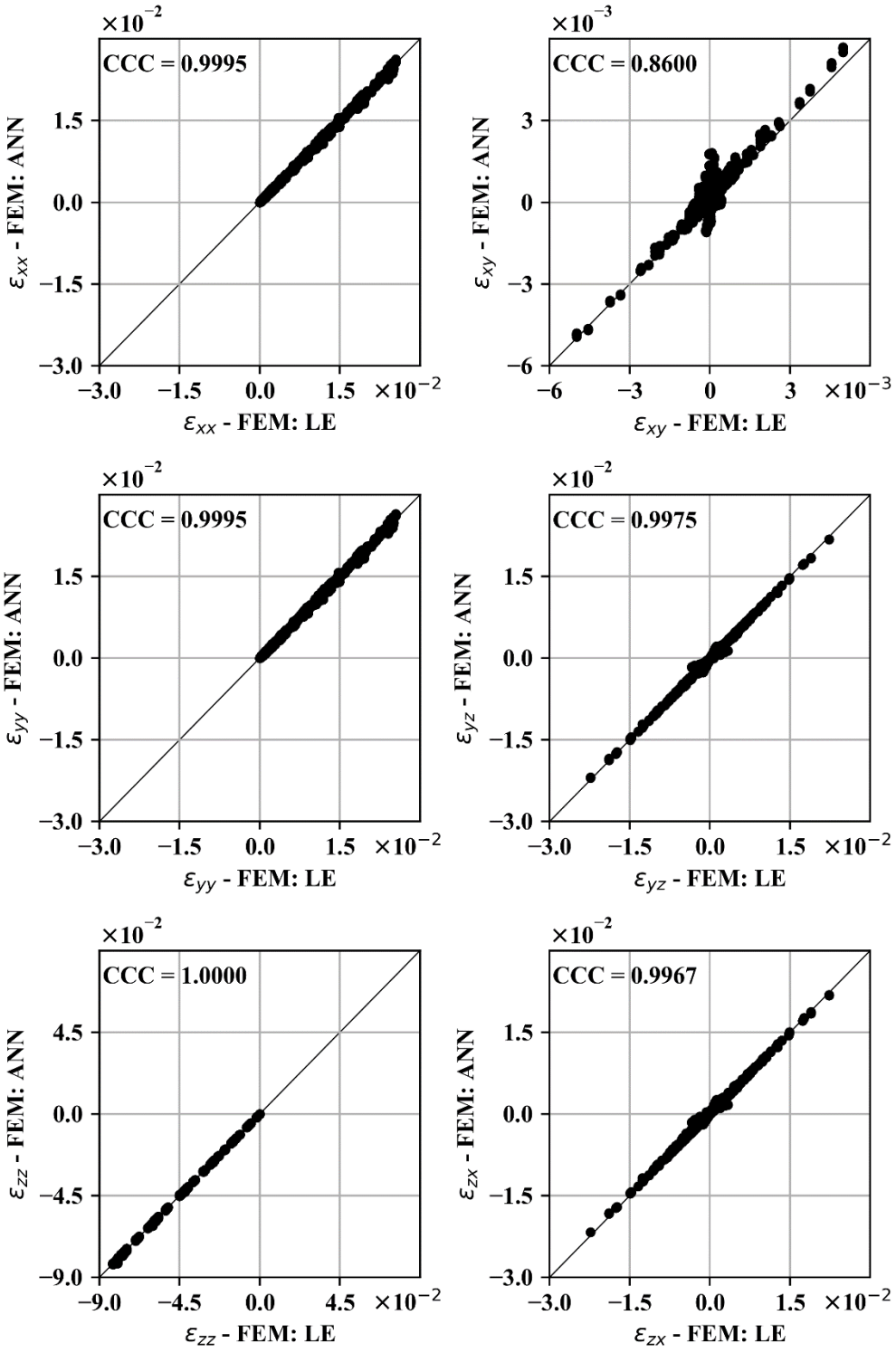


Figure C-1: Comparison of the strains calculated the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model. Element type “2” was used.

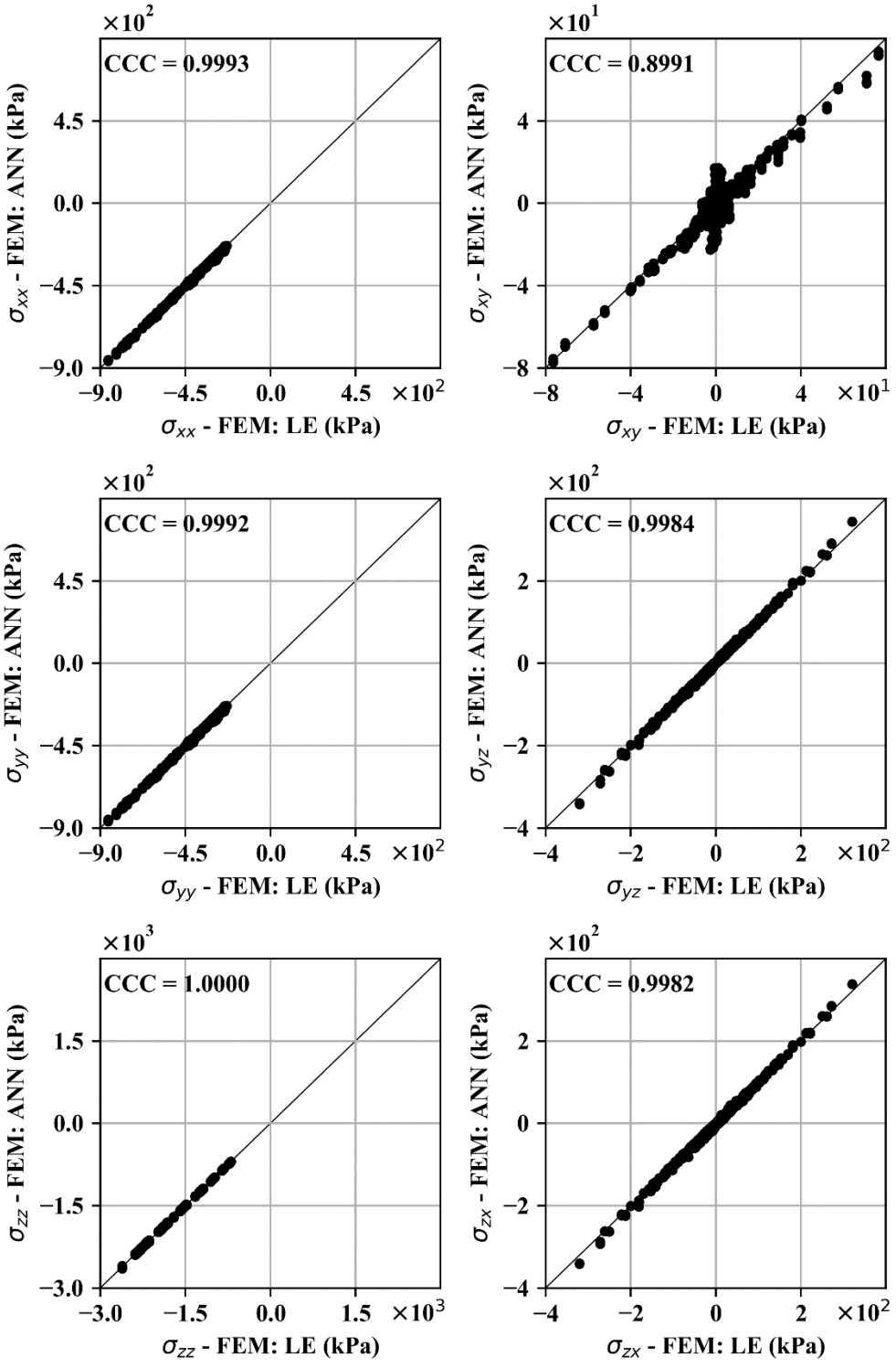


Figure C-2: Comparison of the stresses calculated the CTX-LE FEM analysis vs those calculated by the FEM analysis employing the ANN model. Element type “2” was used.