ISTANBUL TECHNICAL UNIVERSITY ★ EARTHQUAKE ENGINEERING AND DISASTER MANAGEMENT INSTITUTE

CPU-ACCELERATED EARTHQUAKE SIMULATIONS FOR LARGE SCALE URBAN CITIES

M.Sc. THESIS

Mert UYSAL

Department of Earthquake Engineering

Earthquake Engineering Programme

DECEMBER 2019

ISTANBUL TECHNICAL UNIVERSITY ★ EARTHQUAKE ENGINEERING AND DISASTER MANAGEMENT INSTITUTE

CPU-ACCELERATED EARTHQUAKE SIMULATIONS FOR LARGE SCALE URBAN CITIES

M.Sc. THESIS

Mert UYSAL
(802161213)

Department of Earthquake Engineering

Earthquake Engineering Programme

Thesis Advisor: Asst. Prof. Zeynep DEĞER
Thesis Co-Advisor: Assoc. Prof. Gian Paolo CIMELLARO

DECEMBER 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ DEPREM MÜHENDİSLİĞİ VE AFET YÖNETİMİ ENSTİTÜSÜ

BÜYÜK ÖLÇEKLİ ŞEHİRLER İÇİN CPU İLE HIZLANDIRILMIŞ DEPREM SİMULASYONLARI

YÜKSEK LİSANS TEZİ

Mert UYSAL
(802161213)

Deprem Mühendisliği Anabilim Dalı

Deprem Mühendisliği Programı

Tez Danışmanı: Dr. Öğr. Üyesi Zeynep DEĞER
Eş Danışman: Doç. Dr. Gian Paolo CIMELLARO

ARALIK 2019

Mert Uysal, a **M.Sc.** student of ITU **Institute of Earthquake Engineering and Disaster Management** student ID **802161213**, successfully defended the **thesis** entitled **"CPU-ACCELERATED EARTHQUAKE SIMULATIONS FOR LARGE SCALE URBAN CITIES"**, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Asst. Prof. Zeynep DEĞER** Istanbul Technical University	....
Co-advisor :	**Assoc. Prof. Gian Paolo CIMELLARO** Politecnico di Torino	....
Jury Members :	**Assoc. Prof. Ufuk YAZGAN** Istanbul Technical University	....
	Asst. Prof. Gülşen Taşkın KAYA Istanbul Technical University	....
	Asst. Prof. Oğuzhan CEYLAN Kadir Has University	....

Date of Submission : 15 November 2019
Date of Defense : 17 December 2019

To my mother,

FOREWORD

Firstly, I would like to thank my thesis advisor Asst. Prof. Zeynep DEĞER for her guidance, help and support.

I would like to thank my thesis co-advisor Assoc. Prof. Gian Paolo CIMELLARO for his help, guidance and hospitality in Politecnico di Torino, Italy. I am most grateful for making this thesis possible.

I would also like to thank jury members Assoc. Prof. Ufuk YAZGAN, Asst. Prof. Gülşen Taşkın KAYA and Asst. Prof. Oğuzhan CEYLAN.

Lastly, special thanks to my mother for her support, I would not have been able to do this without her. There are no words that can express my gratitude for what she has done for me.

December 2019 — Mert UYSAL (Civil Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ALU : Arithmetic Logic Unit
CPU : Central Processing Unit
CSI : Computers and Structures Inc.
CUDA : Compute Unified Device Architecture
DRAM : Dynamic Random Access Memory
DRSL : Disaster Resilience Simulation Lab
ERC : European Research Council
GPU : Graphics Processing Unit
IES : Integrated Earthquake Simulation
MDOF : Multi Degree of Freedom
NRC : Norcia
NRHA : Nonlinear Response History Analysis
OPENSEES : Open System for Earthquake Engineering Simulation
PEER : Pacific Earthquake Engineering Research Center
RAM : Random Access Memory
SDOF : Single Degree of Freedom

SYMBOLS

c_{eq}	: Equivalent damping
k	: Stiffness
k_{eq}	: Equivalent stiffness
m	: Mass
m_{eq}	: Equivalent mass
t	: Time
u	: Displacement
$\dot{u}$	: Velocity
$\ddot{u}$	: Acceleration
$\omega_{\max}$	: Maximum frequency
$\omega_{\min}$	: Minumum frequency
ξ	: Damping ratio

LIST OF TABLES

Page

LIST OF FIGURES

Page

CPU-ACCELERATED EARTHQUAKE SIMULATIONS FOR LARGE SCALE URBAN CITIES

SUMMARY

Number of individuals is increasing every moment due to high birth rate and low death rate. Because of economic and social circumstances, enlarging population often intend to migrate to large cities for better life expectancy. For this reason, large cities are becoming more complex urban environments which are vulnerable against natural disasters.

One of the most destructive natural disasters is earthquake. After an earthquake occurs, large scale simulation of building damage can potentially reveal possible consequences that are important for disaster mitigation and managing rescue plans for saving as much lives as possible. This study focuses on two challenging problems in large scale simulation under earthquake scenario. First of them is to demonstrate realistic response of huge quantity of building during seismic activity. Second one is to decrease the computational time that is highly significant to provide building damage information rapidly.

A virtual city with around 23000 buildings is modeled using Open System for Earthquake Engineering Simulation (OpenSees) software which is a widely accepted open source software to simulate behavior of structural and geotechnical systems during earthquake motion. Each building is represented as a single degree of freedom (SDOF) system.

Nonlinear response history analyses (NRHA) are applied to simulate seismic response of each building in the virtual city. The analyses are performed by OpenSees. Despite the analysis capability of OpenSees, its performance can be restrictive since all functions are implemented to use the CPU in the traditional way, which is single-processing. Therefore, to better exploit the computational capabilities of CPU, Python libraries can be used in conjunction with the new OpenSeesPy library which brought Python interface into use.

Python is an effective, flexible and user-friendly programming language. To reveal the full capacity of CPU, one of the useful packages of Python is *multiprocessing* package that supports spawning subprocesses. Therefore, underlying logic of the *multiprocessing* package fits well with the requirements of rapid large scale earthquake simulations that needs a lot of computational capacity.

According to results, CPU-multiprocessing application can be faster up to 37 times than CPU single-processing application. Independent of computational platform, Python scripts can be run on many computers and operating systems such as Windows, Linux, macOS. It can be used to solve different kind of civil engineering problems related with computational time.

BÜYÜK ÖLÇEKLİ ŞEHİRLER İÇİN CPU İLE HIZLANDIRILMIŞ DEPREM SİMULASYONLARI

ÖZET

Yüksek doğum oranı ve düşük ölüm oranı sebebi ile her an birey sayısı artmaktadır. Daha iyi bir hayat standartına ulaşabilmek için, genişleyen populasyon genellikle ekonomik ve sosyal sebepler ile büyük şehirlere göç etmeye meyillilerdir. Bu sebepten dolayı büyük şehirler doğal afetlere karşı kırılgan olan daha kompleks şehir alanları haline gelmektedirler.

En yıkıcı doğal afetlerden birinin deprem olduğu herkes tarafından bilinen bir gerçektir. Bir deprem meydana geldikten sonra, bina hasarlarının büyük ölçekli simülasyonu, doğal afet sonucu olusan hasarların azaltılması ve mümkün olduğunca hayat kurtarabilmek için kurtarma planlarının yönetilmesi için önemli olan birçok olası sonuçları ortaya çıkarabilir. Bu simülasyonlar sayesinde, doğal afetleri ve etkilerini daha iyi anlayıp bu konu üzerinde daha fazla çalışma yaparak doğal afetlere karşı daha hazırlıklı olunması mümkün hale getirilebilir. Bu çalışma, deprem senaryosu etkisinde gerçekleştirilmiş olan büyük ölçekli simülasyonda iki zorlu soruna odaklanmaktadır. Bunlardan birincisi, sismik faaliyet sırasında yüksek sayıda binanın gerçekçi tepkisini gösterebilmektir. İkincisi ise bina hasar ve deformasyon bilgilerini hızlı bir şekilde sağlamak için oldukça önemli olan hesaplama süresini olabildiğince azaltmaktır.

Yaklaşık 23000 binanın bulunduğu bir sanal şehir, yapısal ve jeoteknik sistemlerin deprem etkisi altındaki davranışlarını simüle etmek için yaygın olarak kabul gören ve açık kaynak kodlu bir yazılım olan Open System for Earthquake Engineering Simulation (OpenSees) ile modellenmiştir. Modellenmesi için seçilen şehir, İtalya’da bulunan Torino şehrinin verileri kullanılarak oluşturulmuş bir sanal şehirdir. Bu sanal şehirde, betornarme ve yığma binalar modellenmiştir. Modellemeler sırasında, betonarme binalar için OpenSees’in “MultiLinear” materyali, yığma binaları için ise “Hysteretic” materyali kullanılmıştır. Bu modelleme sırasında her bir bina tek serbestlik dereceli sistem olarak temsil edilmiştir. OpenSees yazılımı ile hem yüksek sayıdaki binaların eş zamanlı modellenebilmesi, hem de oluşturulmuş bu modellerin analizlerinin daha hızlı yapılabilmesi mümkün hale gelmiştir.

Binaların sismik haraket karşısındaki tepkilerini simüle etmek ve sanal şehrin sismik simülasyonunu gerçekleştirmek için doğrusal olmayan zaman tanım alanında analizler yapılmıştır. Bu analizlerde, adımla tekniği olarak Newmark yöntemi kullanılmıştır. Deprem kaydı seçiminde ise, 2016 yılının Ekim ayında meydana gelen Merkez İtalya depreminin Norcia istasyonu tarafında kayıt altınan alınan batı-doğu ve kuzey-güney bileşenleri tercih edilmiştir.

OpenSees yazılımının güçlü hesaplama yeteneklerine rağmen, geleneksel yöntem olan tekli işleme yöntemi ile merkezi işlem biriminin (CPU) hesaplama hızı limitler içinde sıkışmaktadır. Bu sıkışmanın sebeplerinden biri, TCL programlama dilinin OpenSees yazılımı için belirlenmiş yorumlayıcı olmasıdır. Her ne kadar TCL programlama dili

güçlü ve etkili bir dilde olsa, ileri seviye veri analizleri ve bilimsel hesaplama gibi konularda geri kalmaya başlamaktadır. Fakat, yakın zamanda yapılan OpenSees yorumlayıcı ile ilgili gelişmeler ışığında, Python programlama dili bir OpenSees yorumlayıcısı olarak kullanıma sunulmuştur. Bu nedenle, CPU'nun hesaplama kapasitesinden daha iyi yararlanabilmek için, etkili, esnek ve kullanıcı dostu bir programlama dili olan Python için OpenSees yorumlayıcısı ile birlikte Python kütüphaneleri kullanılabilir.

CPU'nun tam kapasitesini ortaya çıkarmak için, en faydalı Python paketlerinden biri alt işlem birimlerini çoğaltmaya ve bu işlem birimlerine eş zamanlı olarak görev atanmasını sağlayan multiprocessing paketidir. Python ile CPU çoklu işlem yapmanın ana kavramı "tek görev, tek mantıksal çekirdek" mantığı ile açıklanabilir. Her bir görev, bir binanın modellenmesi ve ardından zaman tanım alanında doğrusal olmayan analizinin gerçekleştirilmesi olarak tanımlanmıştır. Kullanılan CPU'nun sanal çekirdek sayısına paralel olarak görevler oluşturulmakta ve hesaplama hızı yine sanal çekirdek sayısına bağlı olarak lineer olarak bir artış göstermektedir. Bu durumlar göz önünde bulundurulduğunda, Python multiprocessing paketi çok fazla miktarda hesaplama kapasitesi gerektiren büyük ölçekli deprem simülasyonunda hesaplama süresini azaltmak için kullanılabilir.

Bu tez çalışmasının giriş bölümünde, bilgisayar bilimi alanında ne gibi gelişmeler olduğu, bu gelişmelerin deprem mühendisliği alanına etkileri ve kullanım alanları, yapılan çalışmalarla ilgili örnekler ve bu tez kapsamında yapılan çalışmanın hangi sebep doğrultusunda yapıldı ile ilgili bilgiler verilmiştir.

İkinci bölümünde, tekli işleme ve çoklu işleme tekniklerinin çalışma mantıklarının açıklanması, çoklu işleme için kullanılabilecek donanımlardan ve bu donanımların mimarilerinden bahsedilmiştir.

Üçüncü bölümde, modellenecek olan sanal şehrin genel özelliklerinden, modelleme yönteminden, şehirde bulunan binaların mimari oranlarından ve bina verilerinin nasıl elde edildiği ile ilgili bilgi verilmiştir.

Dördüncü bölümde, OpenSees yazılımı ve yorumlayıcıları olan TCL ve Python programlama dilleri hakkında bilgi, modelleme sırasında kullanılan elemanlar ve materyaller, seçilen deprem kaydının detaylı özellikleri ve grafikleri verilmiştir.

Beşinci bölümde, zaman tanım alanında doğrusal olmayan analizlerin, Python ve OpenSees kullanarak nasıl gerçekleştirildiği, tekli işleme ve çoklu işleme tekniğine göre akış şemaları, analizlerin yapıldığı bilgisayar hakkında bilgi, analizler sonucu hangi çıktıların elde edildiği ve bu çıktıların SAP2000 programı ile karşılaştırılması ele alınmıştır.

Altıncı bölümde ise CPU tekli işleme, CPU çoklu işleme ve aradaki farkları daha iyi gözlemleyebilmek için GPU ile yapılan analizler karşılaştırılmıştır.

Elde edilen sonuçlara göre CPU çoklu işleme uygulaması, CPU tekli işleme uygulamasından 37 kata kadar daha hızlı olabildiği gözlemlenmiştir. Elde edilen bu sonuç, toplamda 80 mantıksal çekirdek içeren çift Intel Xeon E5 işlemciye sahip bir bilgisayar ile elde edilmiştir. Bu tez kapsamında geliştirilmiş olan Python kodu, hesaplamayı yapacak olan bilgisayarın kullandığı Windows, Linux ve maxOs gibi işletim sistemlerinden bağımsız olarak uygulanabilir ve geliştirilebilir.

Son olarak, bu çoklu işleme uygulaması, sadece modelleme ve analizle sınırlı kalmamakla birlikte bir çok hesaplama hızı ile ilgili farklı türlerdeki inşaat mühendisliği problemlerine adapte edilip kullanılabilir.

1. INTRODUCTION

Many researchers have been studying [1, 2] in earthquake engineering field, including the topics of design and assessment of buildings, even simulation of building collapse. During the implementation progress of such studies, elapsed time of analysis depends on capacity and usage method of the computational platform. In the computer science field, computational capacity of hardware has advanced day by day, which means both number of cores and clock speed have increased. These hardware modules include central processing unit (CPU) and graphics processing unit (GPU).

In last decades, there were many applications that occurred with the hardware developments. These applications exploit the parallelized architecture of newly designed hardware rather than the clock speeds. Naturally, many applications that employ parallelism were applied primarily on computer science problems. The use of hardware accelerated multiprocessing for numerical computation has mainly driven by deep learning algorithms applied on the computer vision area. One of the earliest studies is AlexNet [3], in which the authors try to solve object classification problem on Imagenet Large Scale Visual Recognition Challenge dataset [4]. After this achievement, many technology companies such as Google and Microsoft were attracted to the field and they have established departments in order to research deep neural networks. In GoogleNet [5], authors have improved previous results. Similarly, Residual Neural Networks [6] from Microsoft Research had satisfactory results. Especially after these developments, parallel computing phenomenon became a popular topic in many fields including earthquake engineering.

With the hardware developments, various topics become possible such as large structure models and large-scale earthquake simulations that need huge computational capacity. For instance, one of the main studies, named “Integrated earthquake simulation (IES)”, has applied the simulation for Tokyo City [7]. In IES, nonlinear response history analysis (NRHA) for multi-scale analysis is proposed in order to calculate seismic response of buildings in selected area. However, IES study is applied with a super computer which is not easy to obtain due to its high operating costs.

Recently, Sahin et al. [8] developed and adapted the IES system to simulate the response of Zeytinburnu district of Istanbul under a given earthquake scenario. In order to enhance this new version of IES, they used MATLAB programming language with the data obtained from GIS databases.

One of the various applications is the new modeling method for mega-tall buildings. Xinzheng Lu et al. [9] proposed a modeling and analyzing method for the Shanghai Tower which has 27220 nodes and 632m height. For modeling, Open System for Earthquake Engineering Simulation (OpenSees) [10, 11] is used, which is an open source software that uses finite element method. For analysis, OpenSees is also used to solve the Shangai Tower's large mass, stiffness and damping matrices, with a GPU based solver named CuSPSolver [9], which is based on Cusp library [12]. Furthermore, in order to apply this framework, they utilized from GPU/CPU cooperative computing method [13].

For improving nonlinear response history analysis (NRHA), two usual problems were investigated [14]. These two problems are convergence problems related with nonlinearity at high levels and execution time. In order to improve computational time, the parallel computing versions of OpenSees (e.g., OpenSeesSP and OpenSeesMP) are used with NEEShub [15] which provided 4232 processors at the research date. The shortcuts of SP and MP refer Single Parallel interpreter and Multiple Parallel interpreter respectively. Xinzheng Lu et al. [16] proposed another method for both modelling and computing large scale earthquake simulation. They proposed two types of multi degree of freedom (MDOF) systems. First one is shear model for multi story buildings [17]. Second one is flexural shear model for tall buildings [18]. In addition, a GPU based solver [13, 19] which is able to do parallel computing, is proposed.

Another parallelization with OpenSees was applied using GPU [9]. In order to adapt for parallelization, a model for matrix and vector calculations was proposed, named the Bold Model. For this study, three different analyses were performed with the Bold Model. First one was traditional way which is single-processing. Second was CPU multi-processing and the last one was GPU multi-processing.

However, performing analysis with GPU based solvers requires Compute Unified Device Architecture (CUDA) programming [20] which provides a parallel computing platform developed by NVIDIA. Although GPU solvers provide a huge computational capacity, development of CUDA programming is not easy to apply.

On the other hand, recently, M. Zhu et al. [21] proposed a new OpenSees library interface that enables OpenSees to be used as a Python module. With the interpreter of OpenSees, highly effective and useful Python libraries can also be used. Among the Python libraries, especially the *multiprocessing* package can be used to reveal full capacity of CPU. The multiprocessing package supports spawning subprocesses, which means "many tasks on several logical cores at any given time''.

Marasco S. [22] proposed a novel method for assessing the performance of infrastructures and their interdependencies under the project called Integrated Design and Control of Sustainable Communities during Emergencies (IDEAL RESCUE). The project is funded by European Research Council (ERC). In this project, both modeling and analysis are performed by using SAP2000 [23] which is a commercial software for civil engineering.

This study aims to contribute the IDEAL RESCUE project with new a modeling and analyzing approach. Modeling system of the project with SAP2000 software is changed with OpenSees framework. Moreover, the analyzing method is renewed and multi-processing technique is adopted in order to speed up the project. Both modeling and analyzing process are developed using Python programming language environment [24].

2. MULTIPROCESSING TECHNIQUES

2.1 Description of Multiprocessing

Recently, central processing units (CPUs) with multi-cores are standard in modern devices such as mobile phones, computers, even in televisions. Although CPUs in computers have multi cores, most of the programming languages (e.g. Python, C, C++, Fortran, Matlab) are designed to process the instructions with single-processing technique by default, which operates one machine level operation at one time, on a single core. This is why some of the application frameworks are not able to use full computational capacity of CPU. The flowchart of single-processing is illustrated in Figure 2.1.

However, it is possible to reveal full capacity of CPUs by using multiprocessing technique. Basically, multiprocessing refers to a CPU ability to support more than one process at the same time. All programming languages have own packages or libraries to activate multiprocessing framework. In this study, all scripts are written in Python programming language [25]. There are different kind of packages that are related with multiprocessing in Python environment. One of them is the "*multiprocessing"* package [26] which is able to spawn subprocesses for each logical core. Main logic of multiprocessing is "many tasks on several logical cores at any given time''. The flowchart of multi-processing is illustrated in Figure 2.2.

On the other hand, multiprocessing applications are not limited with CPUs. Graphic processing units (GPUs) are also designed for multiple operation with their multi cores which are designed specifically for graphics rendering. In time, the idea of using GPUs is emerged due to the lack of computational capacity of CPUs. One of the biggest GPU producers is NVIDIA company, who also provides Compute Unified Device Architecture (CUDA) [20] library to run on its GPU devices. CUDA is a parallel computing platform and special programming model that enables programmers to speed up their computing applications significantly by exploiting power of GPUs. In computer science, the ability of GPUs is used for various areas such as image processing, deep learning, numerical analytics etc. It is also tailored for engineers who deal with simulations and heavy datasets.

In addition to multiprocessing techniques with one computer, computer clusters can also be used to speed up applications. Basically, computer clusters refer to a number of connected computers. Cluster systems are able to work together as a single system. Depending on the power of cluster system i.e., number of computers, processor core speed of each computer, applications can speed up the process significantly.

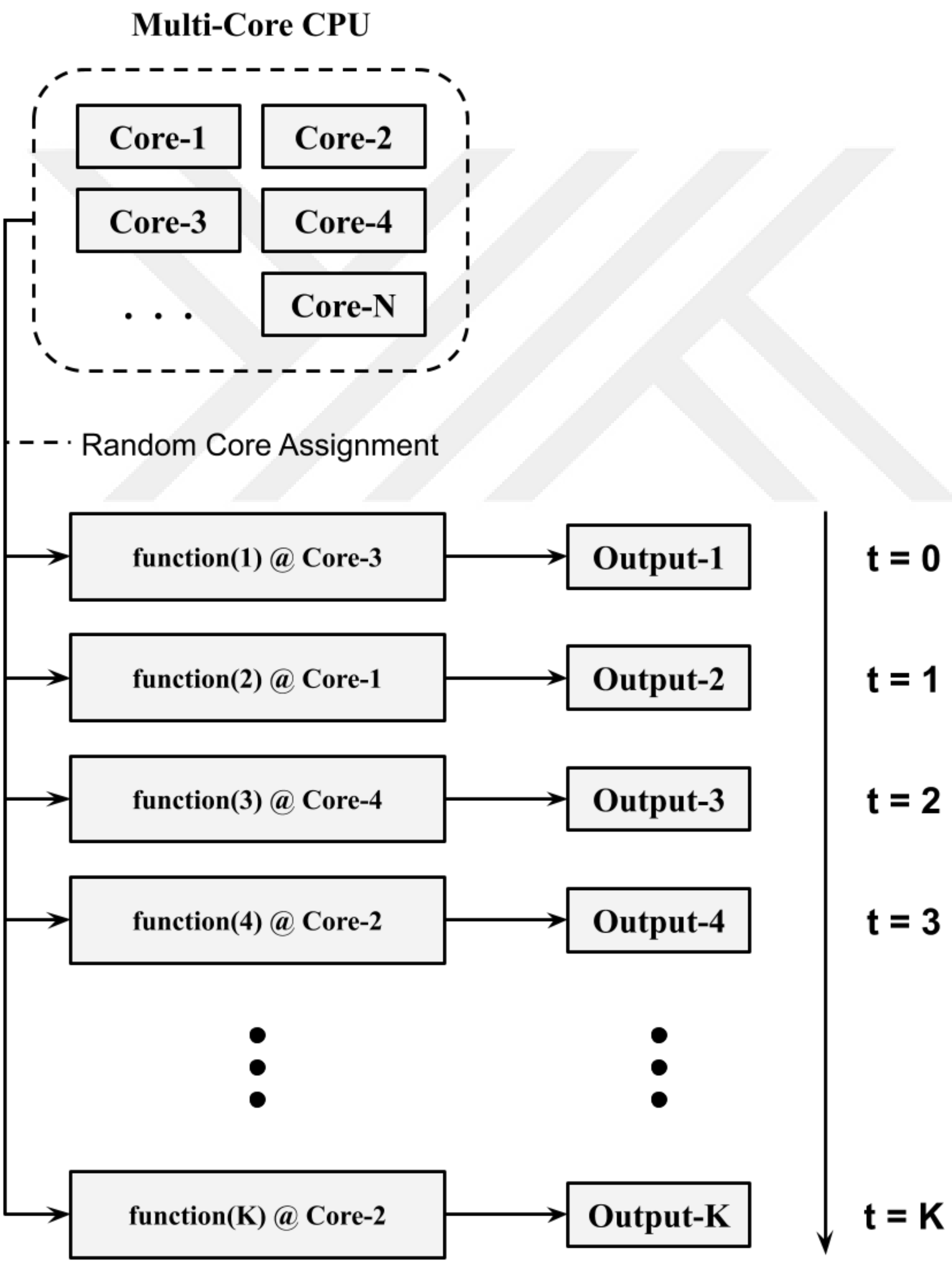

Figure 2.1 : Flowchart of single-processing.

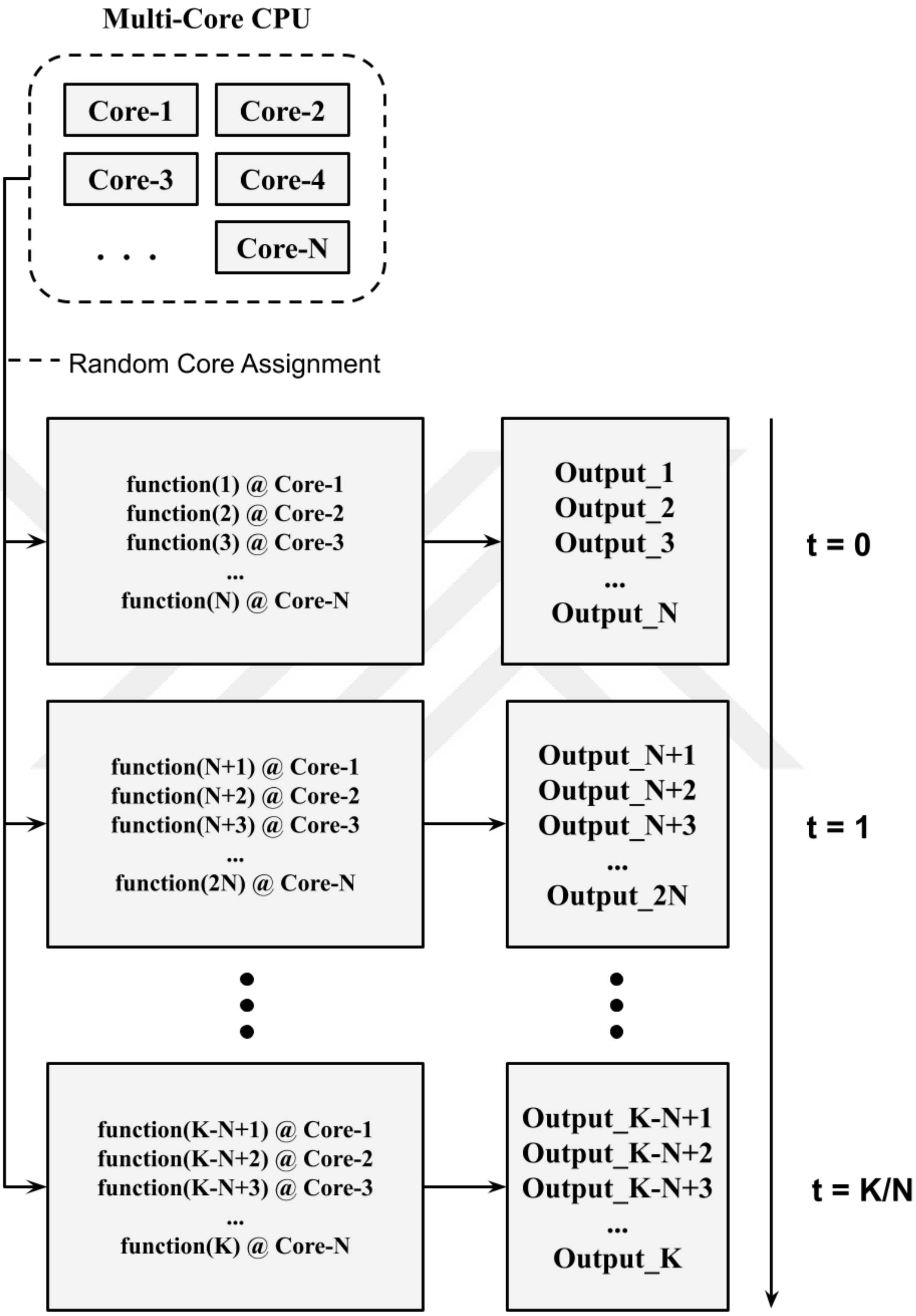

Figure 2.2 : Flowchart of multi-processing.

2.2 CPU and GPU Architectures

When the CPU and the GPU are compared, it is seen that both are microchips made with the same components and cooled by a fan. The components are control block, arithmetic logic unit (ALU), cache and dynamic random access memory (DRAM). Control block is responsible for the leading of related process. Mathematical and logical operations are performed in ALU of processor. Cache is a temporary memory which stores the program code. DRAM is a type of random access memory (RAM) with transistors and capacitors. CPU and GPU architectures are illustrated in Figure 2.3.

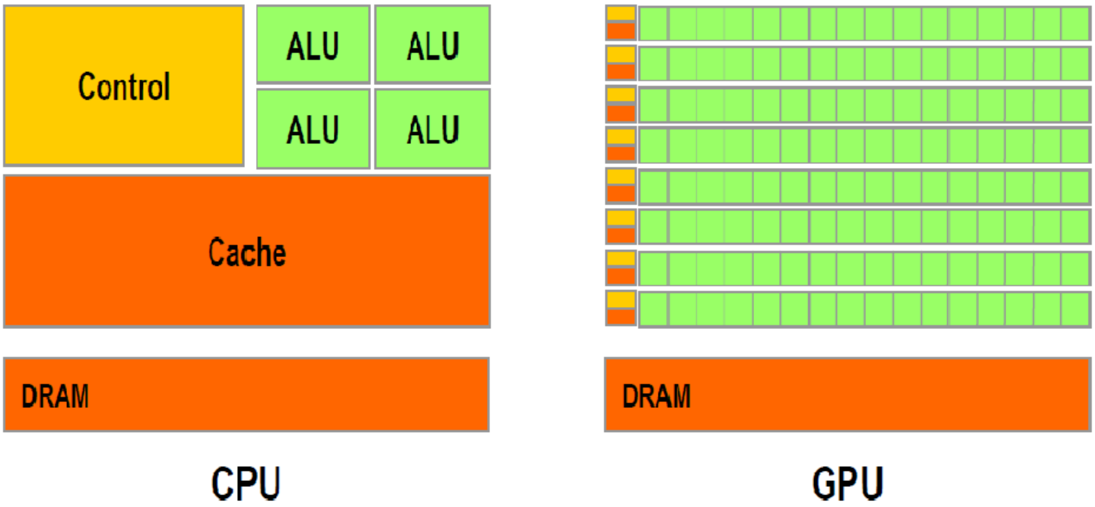

Figure 2.3 : CPU and GPU Architectures.

At the first glance, it can be seen that GPU has more cores than CPU. GPUs can have thousands of cores with slower clock speeds. However, these cores are less capable for general purpose programming.

2.3 Comparison of Feasibility Between CPU and GPU

CPUs are designed to run with mainstream operating systems such as MS Windows and Linux. Over the years, design is evolved according to the requirements. General purpose programming languages are also adapted to this phenomenon. Therefore, it is convenient to write a simple program that runs on CPUs with acceptable memory requirements and speed. There are already many specialized frameworks and libraries developed to run on CPUs for numerical computations.

Therefore it is easy to use these libraries to solve engineering problems. However, not all frameworks and libraries have enough performance to solve every problem in reasonable time.

On the other hand, GPUs are much more advanced hardware designed specifically to solve parallelizable applications. In spite of their computational power, it is much more difficult to adapt all engineering solutions since parallel programming is an advanced technique that requires its own knowledge.

3. DESCRIPTION OF THE VIRTUAL CITY

3.1 Ideal City Model

Modeling and analyzing various types of buildings has been researched in past decades. However, the topic of modeling and analyzing entire large scale cities is becoming an attractive subject for many researchers in last decade. For example, Xinzheng Lu et al., Hori, and Sahin et al. were described in Chapter 1. Marasco S. [22] modeled a virtual city, called IDEAL CITY, which represents the residential buildings of city of Turin, Italy. IDEAL CITY is established under the project called Integrated Design and Control of Sustainable Communities during Emergencies (IDEAL RESCUE) which proposes a novel method for assessing the performance of infrastructures and their interdependencies. In addition, The project of IDEAL RESCUE is funded by European Research Council (ERC). The plan view of the residential buildings is given in Figure 3.1.

Figure 3.1 : Residential buildings of IDEAL CITY modeled by Marasco S. [22].

In the approach of Marasco S., the process of the IDEAL CITY modeling have done in two main steps: (i) data collection and processing, (ii) developing a new building model. The data collection step includes following topics:

- Geometrical parameters
- Structural and nonstructural parameters
  - Building archetypes
  - Construction elements
- Population data
- Year of construction
- Additional information

After the data collection, a MATLAB algorithm has been developed to process the data. Data processing stage includes following assessment steps:

- Unknown years of construction
- Year of construction properties to the unprovided buildings
- Unknown number of story and building archetypes
- Construction elements
- Structural configuration and reinforcement
- Minumum design requirements (RC buildings)
- Minumum design requirements (Masonry buildings)
- Lateral resisting frame dimensions

The developed MATLAB algorithm is able to collect all the minimum design requirements related with the year of constructions. Then, these minimum design requirements are verified by the Italian Seismic Standards.

As second part of modeling step, a new building model is proposed to demonstrate the seismic response of buildings. The developing a new building model for IDEAL CITY includes following determinations for both reinforced concrete and masonry buildings:

- Elastic parameters
    - Stiffness properties
    - Mass properties
    - Multimodal pushover analysis
- Post-elastic parameters

In order to estimate the uncertainties related with both geometric and mechanical parameters, Monte Carlo Simulation is carried out. After these determinations, to demonstrate the seismic response of buildings, a four-linear backbone curve and a tri-linear backbone curve are proposed for reinforced concrete buildings and masonry buildings, respectively. These backbone curves describe the maximum and minumum global seismic capacity of each building.

Using these backbone curves, each building in IDEAL CITY is modeled as a equivalent single degree of freedom (SDOF) system which demonstrate the elastic and plastic behavior of the buildings. The proposed equivalent single degree of freedom model is illustrated in Figure 3.2.

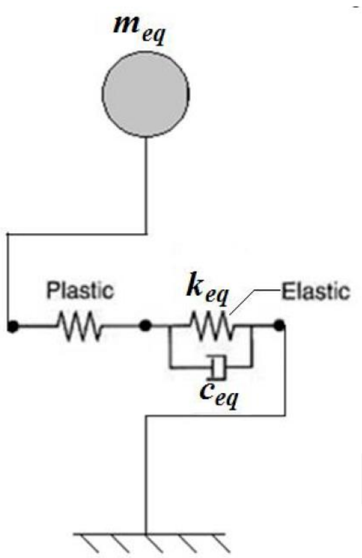

Figure 3.2 : Proposed equivalent SDOF model [22].

where m_{eq} is the equivalent mass of the model, k_{eq} is the equivalent elastic stiffness of the linear element and c_{eq} is the equivalent damping.

3.1.1 Archetypes of buildings

In IDEAL CITY, there are two types of residential buildings which are masonry and RC structures. Table 3.1 exhibits the residential buildings' quantities in IDEAL CITY by archetype. The distribution of the building archetypes is given in Figure 3.3.

Table 3.1 : Residential buildings quantity of IDEAL CITY by archetype.

Masonry	RC	Total
10,005	13,414	23,419

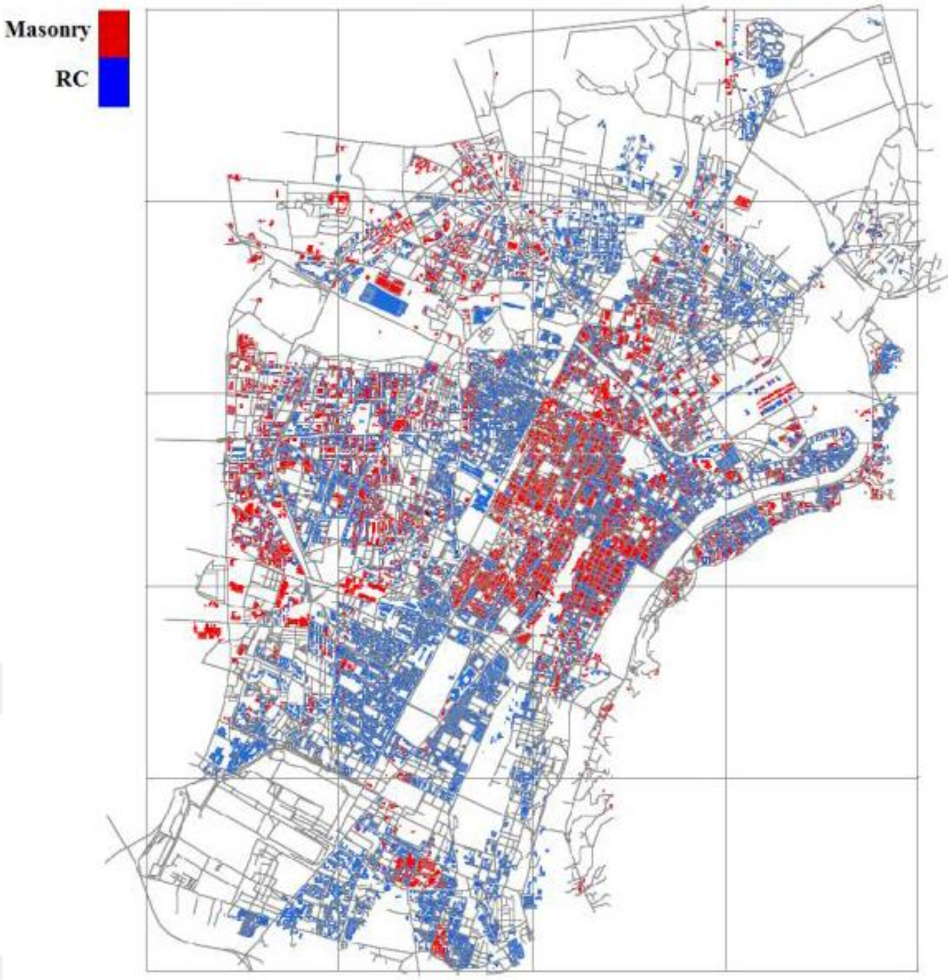

Figure 3.3 : Distribution of building archetypes [22].

3.1.2 Database of buildings

The buildings database of IDEAL CITY generated by Marasco S. [22] refers to backbone curves of SDOF which is defined as lateral load resisting system. For masonry buildings, tri-linear backbone curve is used. On the other hand, four-linear backbone curve is used for RC buildings. Depending on building archetypes, each building has its own specific backbone curve, which specifies possible seismic behavior of buildings. The database also includes mass data of buildings and soil amplification coefficient that is dependent on distance between related building and epicenter of ground motion.

4. OPENSEES MODEL OF VIRTUAL CITY

4.1 OpenSees

The Open System for Earthquake Engineering Simulation (OpenSees) [10] is a widely accepted open source software used to simulate the behavior of structural and geotechnical systems during earthquake motion. It has been under development by Pacific Earthquake Engineering Research Center (PEER) [27]. As a summary, it is a powerful open source software for performing finite element analysis. Because it is open source, all users can modify the the framework to improve their research. The source code of OpenSees is written in C++ programming language [28] mostly. Thanks to the effective and flexible environment of C++, developers are able to generate and perform models rapidly. However, users are not allowed to write their own code in C++ since the framework has no C++ interface. In order to use OpenSees, there are two interpreters: Tcl and Python.

4.2 OpenSees Interpreters

4.2.1 Tcl programming language

From the beginning, Tcl [29] has been the primary programming language of OpenSees. While Tcl is a user-friendly and effective programming language, it is not powerful for advanced computing applications. Furthermore, it is falling behind the new developments in computer science field.

4.2.2 Python programming language

Nowadays, one of the most popular programming languages is Python [25] since it is easy to use, versatile and capable of providing solutions for engineering problems and data science, compared to other programming languages such as Java and C++. Recently, M.Zhu et al. [21] have developed a Python library called OpenSeesPy that allows users to use Python as an interpreter of OpenSees.

With this library, many capable and powerful packages of Python (e.g. Numpy [30], Pandas [31], Matplotlib [32], Multiprocessing [26], etc.) can be used by users to improve their research.

Multiprocessing package of Python is used in this study, to better exploit the computational capabilities of CPUs. Assigning each task to each available logical core is the main idea of CPU-Multiprocessing framework with Python.

4.3 Elements

4.3.1 ZeroLength element

Since all buildings in virtual city are modeled as SDOF system, two nodes between lumped mass and support is defined with ZeroLength Element of OpenSees [33], which provides the possibility of assigning two nodes at same location. Lumped mass and support are connected with UniaxialMaterial object which is the representative of uniaxial force-deformation relationships. Four parameters are determined to define ZeroLength element: nodes number, defined-UniaxialMaterial, material directions and whether or not to include Rayleigh damping.

4.4 Materials

4.4.1 Hysteretic material

To demonstrate the seismic response of masonry buildings, tri-linear backbone curve is used. It is defined with Hysteretic material [33] of OpenSees. While defining Hysteretic material, Takeda model [34] is used for the dynamic strength degradation. Parameters of Hysteretic material related with dynamic strength degradation are determined according to Takeda model. The proposed tri-linear backbone curve for masonry buildings of IDEAL CITY is illustrated in Figure 4.1. The Hysteretic material of OpenSees is given in Figure 4.2.

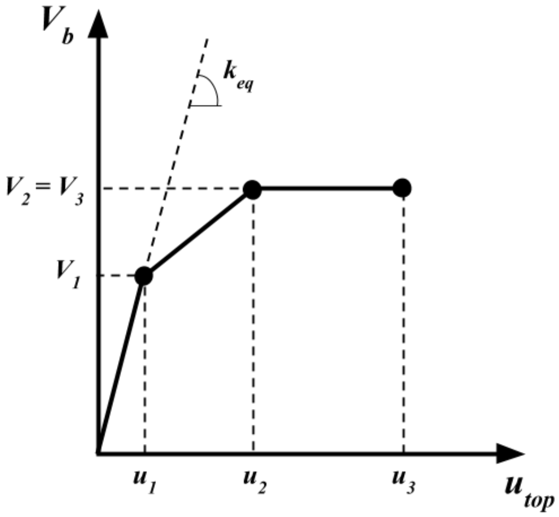

Figure 4.1 : Proposed tri-linear backbone curve for masonry buildings of IDEAL CITY.

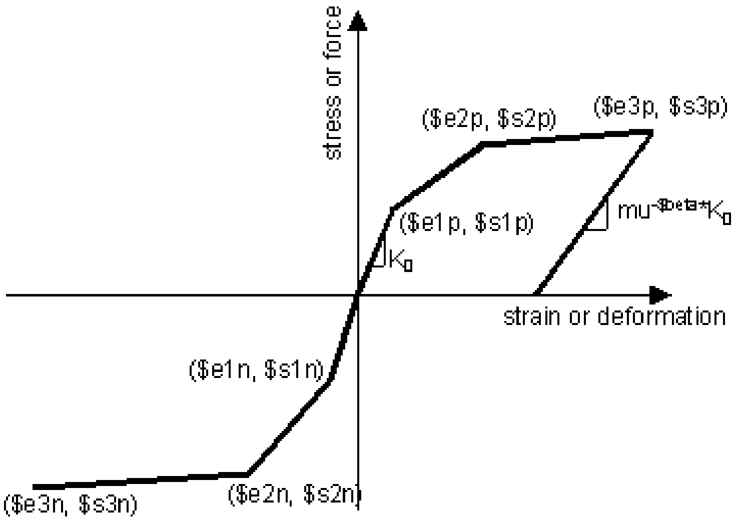

Figure 4.2 : Hysteretic material of OpenSees [33].

4.4.2 MultiLinear material

To demonstrate the seismic response of reinforced concrete buildings, four-linear backbone curve is used. It is defined with MultiLinear material [33] of OpenSees. Since four points input is not supported by Hysteretic material of OpenSees, MultiLinear material is used for reinforced concrete buildings which are modeled with four-linear backbone curve. Although there are no parameters for Takeda model [34] in MultiLinear material, the result between SAP2000 [23] with Takeda model and OpenSees were matched. Comparison between the SAP2000 and OpenSees results is given in Chapter 5.

The proposed four-linear backbone curve for reinforced concrete buildings of IDEAL CITY is illustrated in Figure 4.3. The MultiLinear material of OpenSees is given in Figure 4.4.

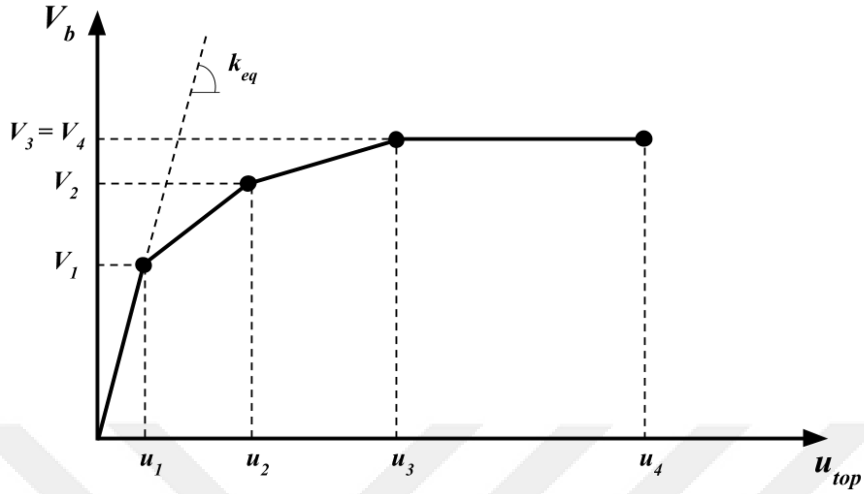

Figure 4.3 : Proposed four-linear backbone curve for reinforced concrete buildings of IDEAL CITY.

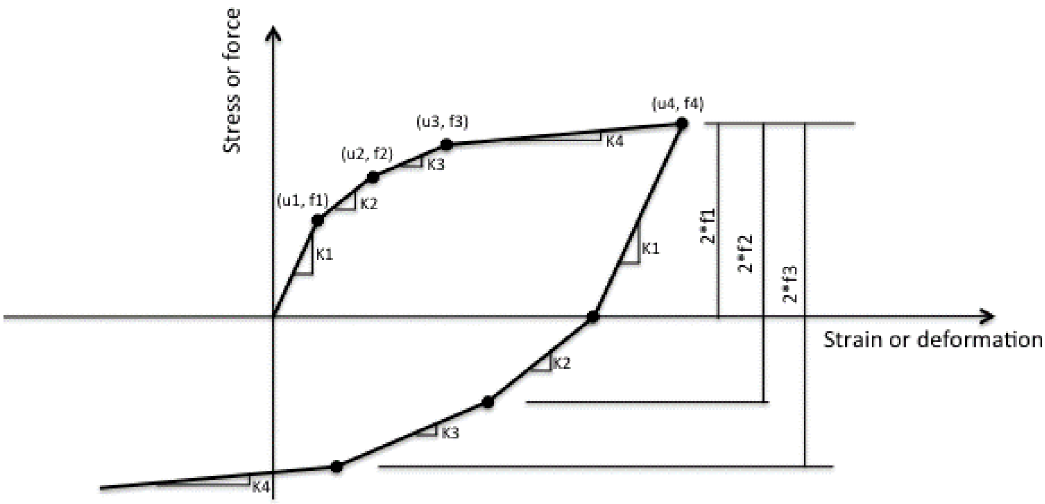

Figure 4.4 : MultiLinear material of OpenSees [33].

4.5 Rayleigh Method

To define %5 damping ratio for the both masonry and reinforced concrete buildings, Rayleigh damping formulation [35] is used. Rayleigh damping is determined according to the first eigen mode of both x and y directions. In order to apply Rayleigh damping formuation, eigenvalue analysis is performed for each building in virtual city.

The Rayleigh formulation is given in Equation 4.1. Details of eigenvalue analysis is shown in Chapter 5.

$$c_{eq} = \alpha.m_{eq} + \beta.k_{eq} \tag{4.1}$$

where c_{eq} is the equivalent damping, m_{eq} is the equivalent mass, k_{eq} is the equivalent elastic stiffness, the coefficient α is given in Equation 4.2.

$$\alpha = 2.\xi.\frac{\omega_{\max}.\omega_{\min}}{(\omega_{\max} + \omega_{\min})} \tag{4.2}$$

where $\omega_{\max}$, $\omega_{\min}$ are the predominant frequencies, ξ is the damping ratio, the coefficient β is given in Equation 4.3.

$$\beta = \frac{2.\xi}{(\omega_{\max} + \omega_{\min})} \tag{4.3}$$

4.6 Specifications of Ground Motion

The north-south and east-west acceleration time histories components recorded by Norcia (NRC) station during October 2016 Central Italy Earthquake (6.5 M_w) are used. One of the main reasons for the ground motion selection is to compare with the previous results obtained with SAP2000. In addition, the ground motion has a frequency bandwidth between 0.4 and 3.3 Hz. The wide frequency bandwidth causes significant dynamic amplification on the building. The graphics of both North-South and East-West components are given in Figure 4.5.

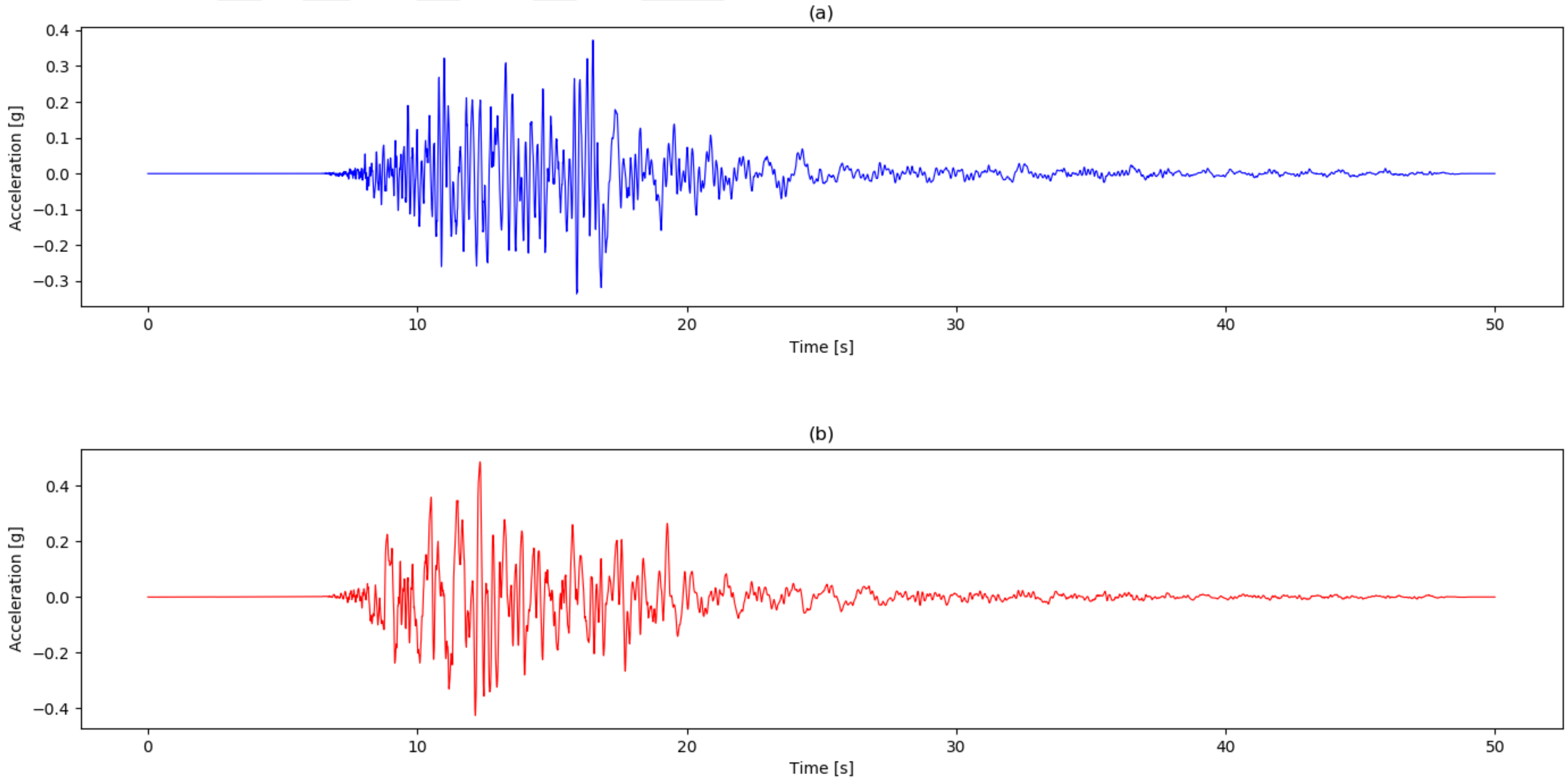

Figure 4.5 : North-South (a) and East-West (b) ground motion records recorded by Norcia station.

Maximum acceleration value of 0.37 g, maximum velocity value of 41.46 cm/sec and maximum displacement value of 0.08 m have been recorded by Norcia station for North-South component of the ground motion. In additon, maximum acceleration value of 0.49 g, maximum velocity value of 48.30 cm/sec and maximum displacement value of 0.18 m have been recorded by Norcia station for East-West component of the ground motion. The specifications of the ground motion is given in Table 4.1. In Table 4.1, further specification values are provided related with engineering intensities. In order to better understand the character of the ground motion, these engineering intensities can be used.

Table 4.1 : Specifications of the ground motion.

Parameter	**N-S component**	**E-W component**
Max. Acceleration (g)	0.37	0.49
Time of Max. Acceleration (sec)	16.52	12.33
Max. Velocity (cm/sec)	41.46	48.30
Time of Max. Velocity (sec)	16.58	10.22
Max. Displacement (m)	0.08	0.18
Time of Max. Displacement (sec)	16.89	11.55
Vmax / Amax: (sec)	0.11	0.10
Acceleration RMS: (g)	0.06	0.07
Velocity RMS: (cm/sec)	5.78	8.05
Displacement RMS: (m)	0.02	0.03
Arias Intensity: (m/sec)	2.40	3.63
Characteristic Intensity (Ic)	0.09	0.13
Specific Energy Density (cm^2/sec)	1670.63	3242.21
Cumulative Absolute Velocity (cm/sec)	1293.09	1558.43
Acceleration Spectrum Intensity (g*sec)	0.38	0.49
Velocity Spectrum Intensity (cm)	167.11	231.09
Housner Intensity (m)	151.48	203.00
Sustained Maximum Acceleration (g)	0.32	0.36
Sustained Maximum Velocity (cm/sec)	28.28	42.80
Effective Design Acceleration (g)	0.37	0.48
A95 parameter (g)	0.37	0.48
Predominant Period (sec)	0.22	0.26

Figure 4.6 shows the elastic acceleration response spectrum of the ground motion with %5 damping value.

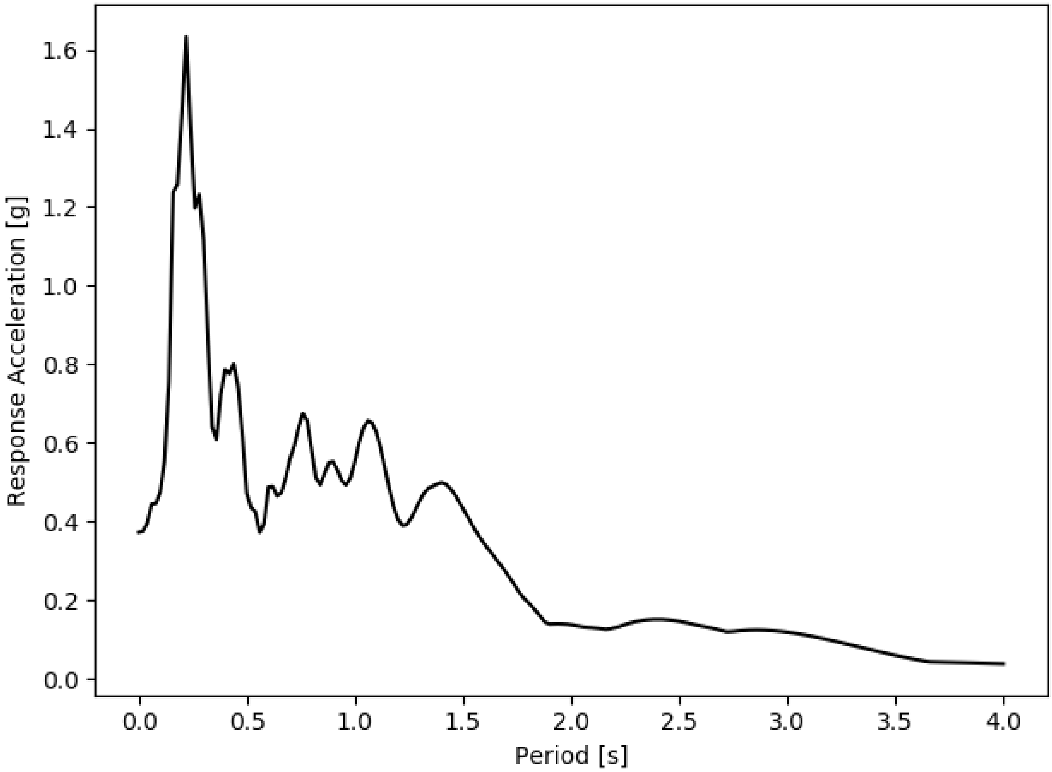

Figure 4.6 : Elastic acceleration response spectrum of the ground motion.

In Figure 4.6, maximum amplitude values can be seen between 0.3 s and 0.4 s. Also, the period range between 0.3 s and 1.4 s represent the fundamental periods of the buildings of IDEAL CITY. For this range, dynamic amplification values between 0.4 g and 0.7 g are observed.

On the other hand, the effect of the ground motion may vary depending on epicenter distance. Generally, the effect of ground motion increases with decreasing epicentral distance. Attenuation of ground motion represented by PGA versus distance. However, the ground motion parameter can be PGV, PGD, etc. Figure 4.7 shows the PGA map of IDEAL CITY.

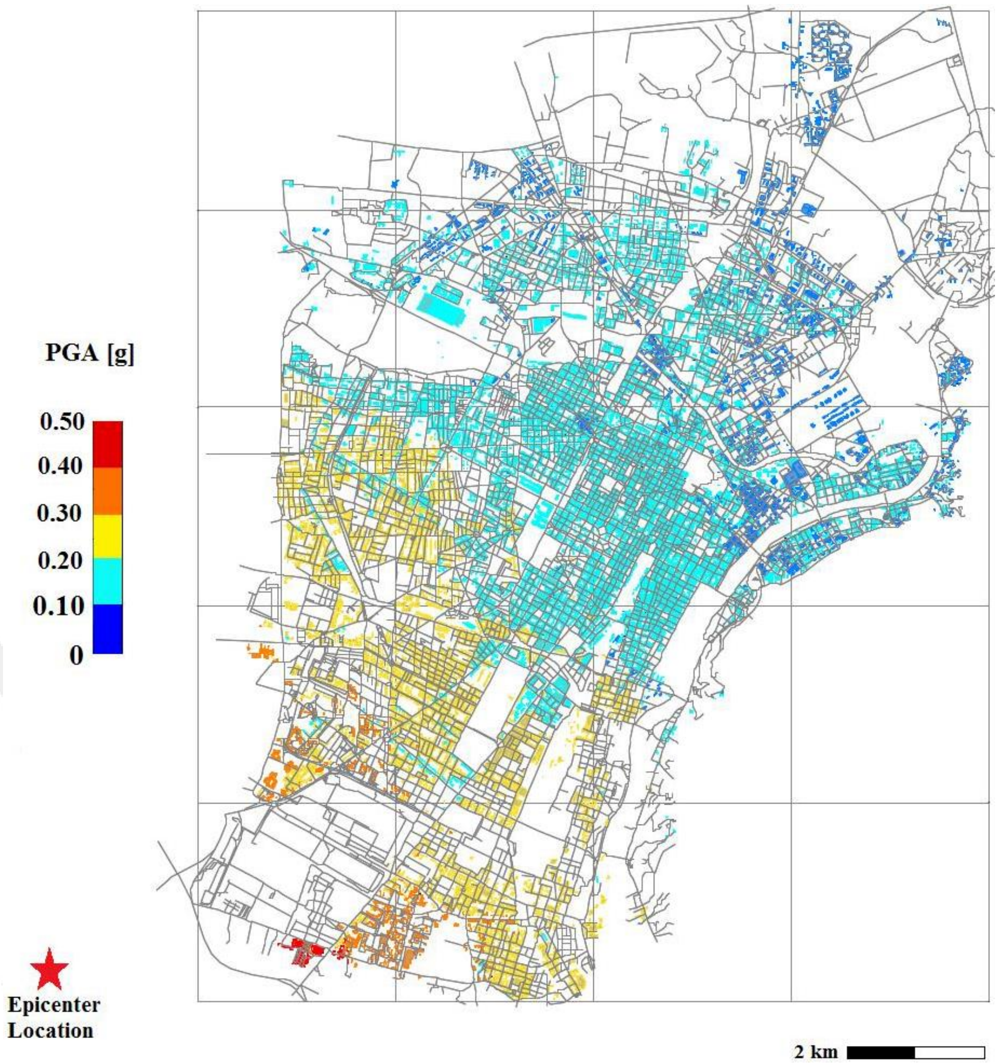

Figure 4.7 : PGA map of IDEAL CITY [22].

For the attenuation model, Boore-Atkinson [36] attenuation model is used. This model provide to estimate the equivalent shear wave velocity in the uppermost 30 m ($V_{s30,eq}$) at the location of each building.

5. NONLINEAR RESPONSE HISTORY ANALYSES USING PYTHON AND OPENSEES

5.1 Eigenvalue Analysis

To determine the period of buildings, eigenvalue analysis is performed. Since all buildings of virtual city modeled as SDOF systems, both x and y directions have only one period each. These calculated periods are used for determining the Rayleigh damping. Eigenvalue analyses are performed by using Eigen Command of OpenSees. The theory of eigen command is based on Equation 5.1.

$$\left(k-\lambda.m\right)\theta=0 \qquad \textbf{(5.1)}$$

5.2 Nonlinear Response History Analysis

To demonstrate the nonlinear behavior of each building, nonlinear response history analysis (NRHA) is carried out with given an earthquake motion. Both north-south component and east-west component of ground motion are applied independently. There are two reason for this. First one is to reduce the computation time. Second one is that if the building collapses in one direction, there is no need to check the other direction. As a stepping method, Newmark's average acceleration method [37] is applied with using Transient Integrator Command of OpenSees. The theory of Newmark's average acceleration method is shown as follows;

$$\ddot{u}\left(\tau\right)=\frac{1}{2}.\left(\ddot{u}_{i+1}+\ddot{u}_{i}\right) \qquad \textbf{(5.2)}$$

where $\ddot{u}_i$ is the acceleration at time i, $\ddot{u}_{i+1}$ is the acceleration at time $i+1$.

$$\dot{u}\left(\tau\right)=\dot{u}_{i}+\frac{\tau}{2}.\left(\ddot{u}_{i+1}+\ddot{u}_{i}\right) \qquad \textbf{(5.3)}$$

where $\dot{u}_i$ is the velocity at time i.

$$\dot{u}_{i+1} = \dot{u}_i + \frac{\Delta t}{2}.\left(\ddot{u}_{i+1} + \ddot{u}_i\right) \quad \textbf{(5.4)}$$

where $\dot{u}_{i+1}$ is the velocity at time $i+1$, Δt is the time step.

$$u\left(\tau\right) = u_i + \dot{u}_i.\tau + \frac{\tau^2}{4}.\left(\ddot{u}_{i+1} + \ddot{u}_i\right) \quad \textbf{(5.5)}$$

where u_i is the displacement at time i.

$$u_{i+1} = u_i + \dot{u}_i.\Delta t + \frac{\left(\Delta t\right)^2}{4}.\left(\ddot{u}_{i+1} + \ddot{u}_i\right) \quad \textbf{(5.6)}$$

where u_{i+1} is the displacement at time $i+1$.

5.2.1 NRHA using single-processing

In order to reveal the differences between single-processing and multi-processing, firstly, NRHA is applied using single-processing technique which is traditional way. To better understanding, flowchart of analysis of one building is shown in Figure 5.1.

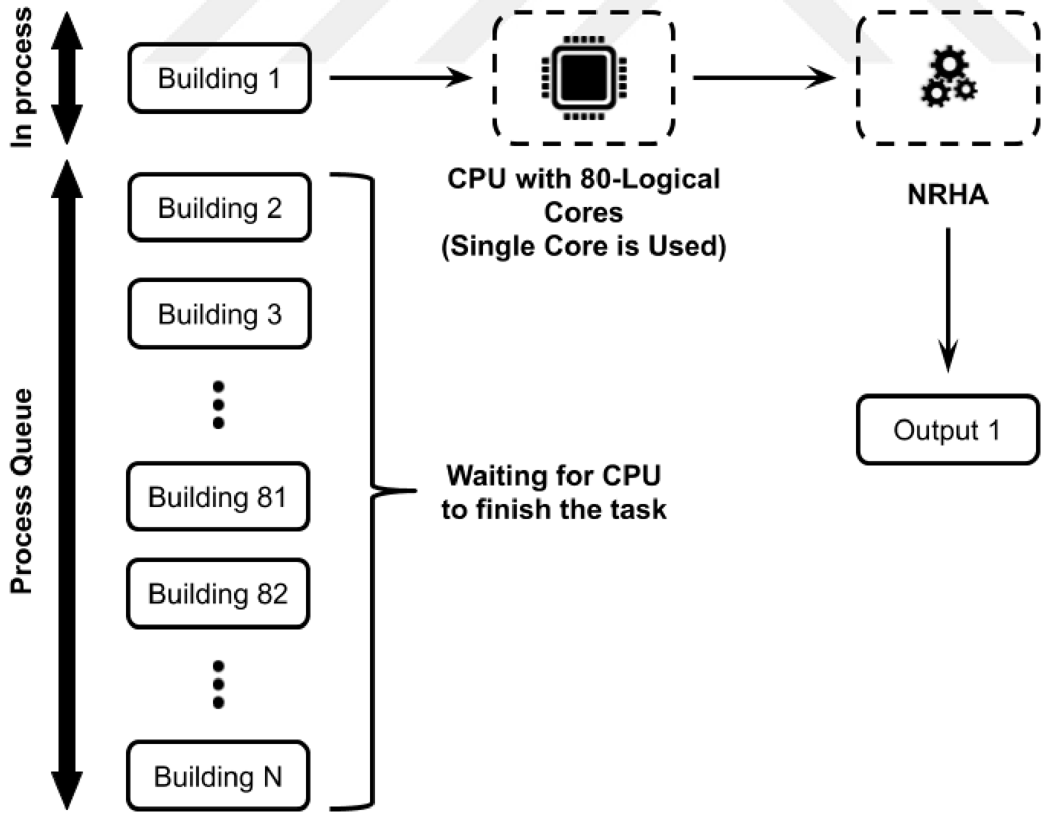

Figure 5.1 : Flowchart of analysis of one building using single-processing technique.

5.2.2 NRHA using multi-processing

Because the project of IDEAL RESCUE is a rescue project after earthquake event. After a seismic activity, one of the most important subjects is to save as much lives as possible. Therefore, the analyses need to be done rapidly. Damaged areas of cities can be estimated and revealed immediately using the accelerated earthquake simulations. Depending on computational time of analysis, the ratio of people rescue may increase. Against the single-processing technique, multi-processing technique is able to do task sharing between logical cores of CPU. In this study, Python code can identify how many logical cores are in the hardware of CPU automatically. After identification, it spawns processes for assigning tasks. The computational platform with CPU that includes 80 logical cores, called IDEAL SERVER, is used for the NRHA. Further information about IDEAL SERVER is provided in Chapter 5.3. While all processes are in process, remaining buildings wait available cores. When one of the logical cores is available for another task, python code assign to solve another task immediately. Flowchart of multi-processing for IDEAL CITY is given in Figure 5.2.

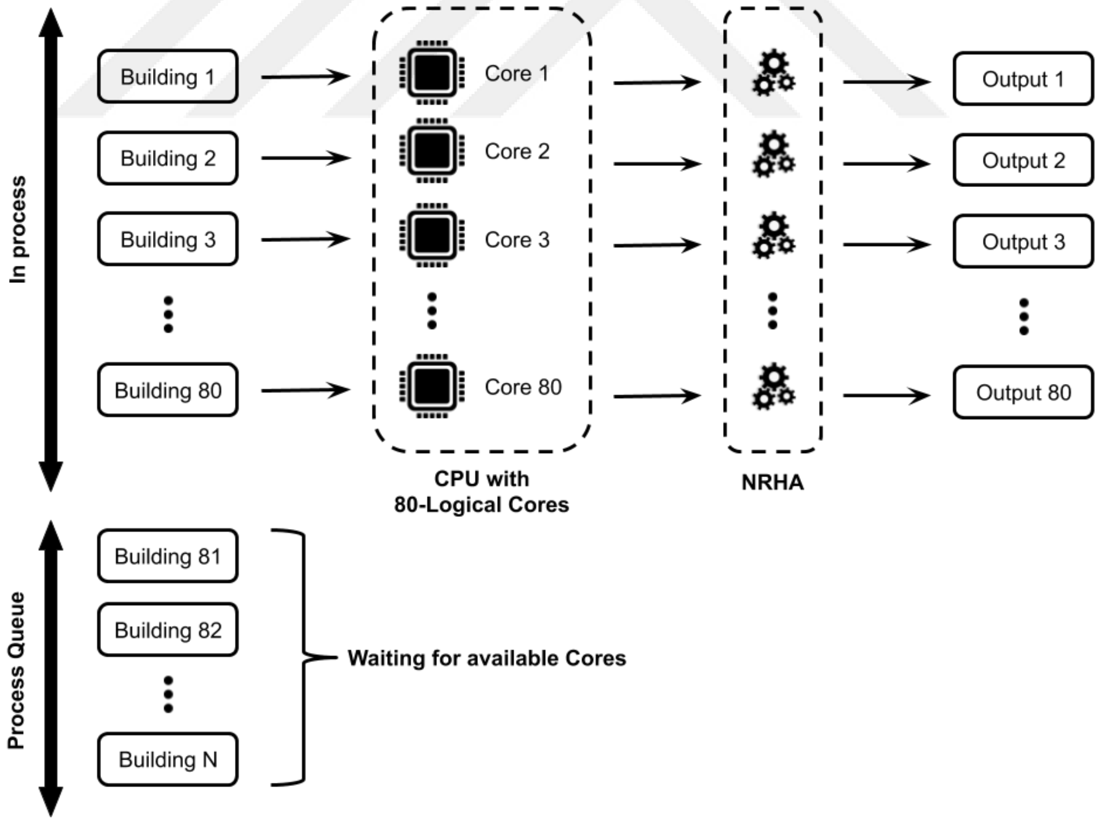

Figure 5.2 : Flowchart of multi-processing for IDEAL CITY.

5.3 Ideal Server

To perform all of the framework, the server of Disaster Resilience Simulation Lab (DRSL), called IDEAL SERVER, is used. It is located in the Structural and Geotechnical Department (DISEG) of the Politecnico di Torino. Specifications of the IDEAL SERVER is given in Table 5.1.

Table 5.1 : Specifications of IDEAL SERVER.

Used Hardware	Used Software
CPU: Intel Xeon E5-2698 v4 processor (x2) with 80 logical cores	OS: Windows 10 Pro
RAM: 256 GB	Python: 3.6.7
GPU: NVIDIA Tesla K40m	OpenSeesPy: 0.2.0

5.4 Outputs of Framework

After all analyses and calculations, the framework written in python is able to provide response of the SDOF system against the given earthquake motion. The outputs includes displacement response history and acceleration response history of each building. It is also possible to read maximum and minimum values of response directly. As an example, both x and y directions displacement outputs of the randomly selected building is given in Figure 5.3.

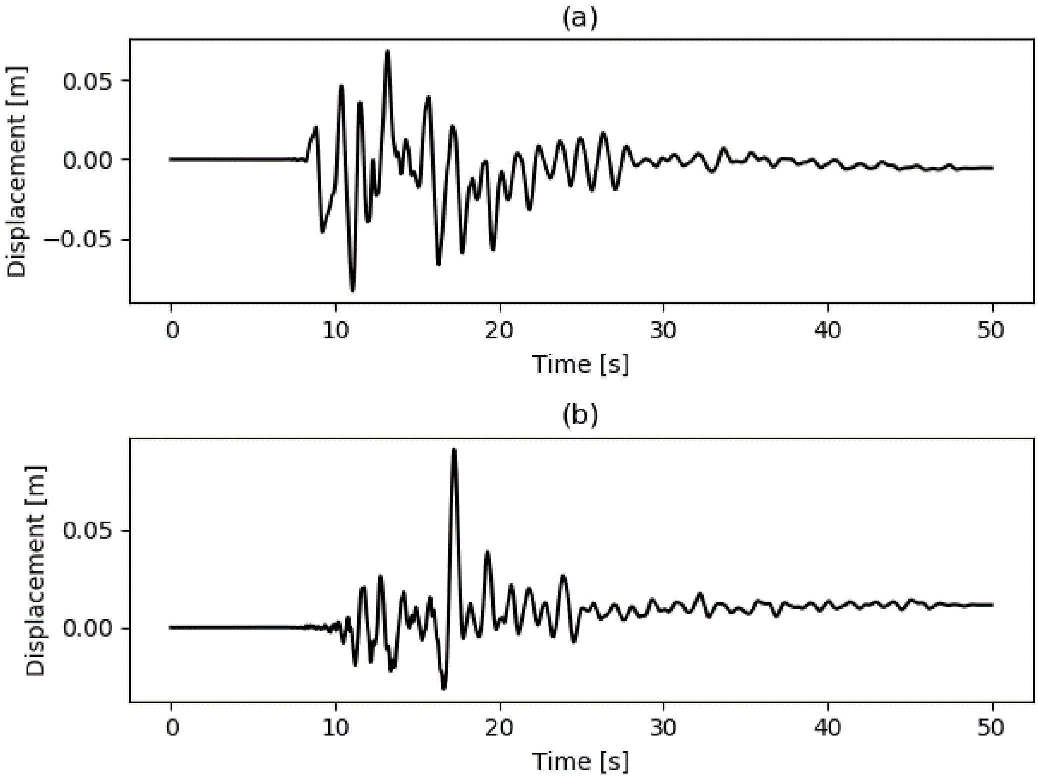

Figure 5.3 : An example of displacement outputs which include both x (a) and y (b) directions.

5.4.1 Comparison of outputs between OpenSees and SAP2000

SAP2000 [23] is the globally used civil engineering software for analyzing structural system using finite element method. Against the OpenSees software which is open source, SAP2000 software is commercial and produced by Computers and Structures Inc. (CSI). To reveal the differences between results of SAP2000 and OpenSees, the comparison graph is given in Figure 5.4.

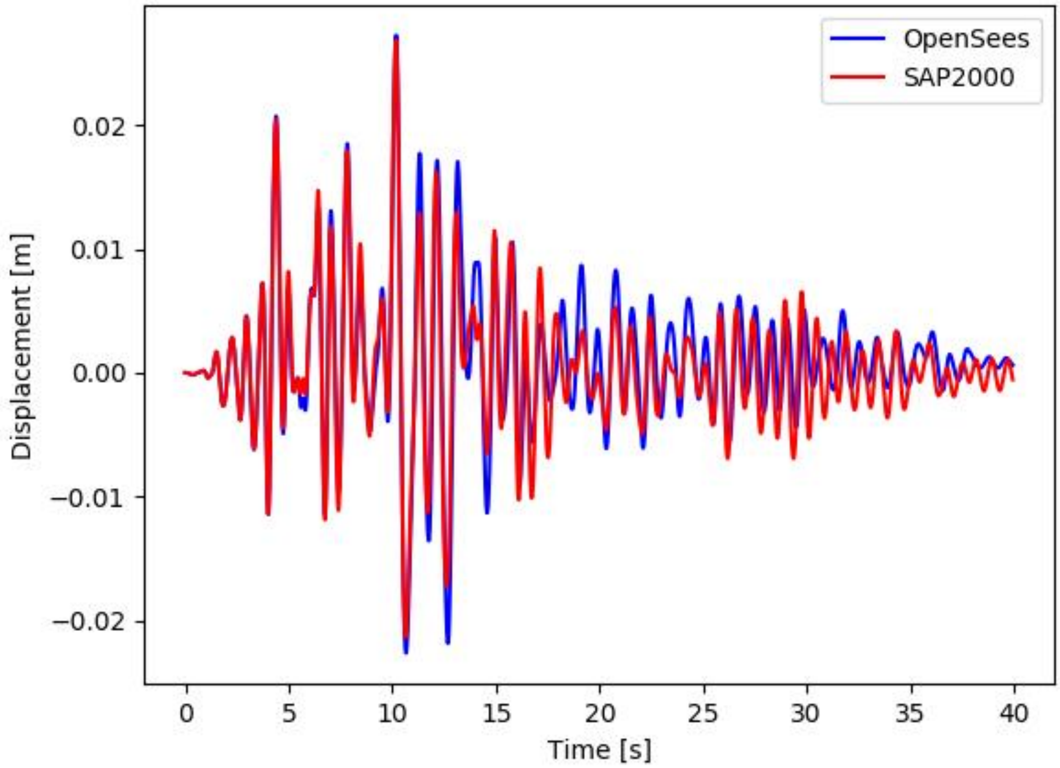

Figure 5.4 : Displacement comparison between OpenSees and SAP2000.

As shown in the Figure 5.4, behavior of randomly selected building under given earthquake motion is similar between both OpenSees and SAP2000 softwares.

6. RESULTS

Nonlinear response history analyses of the virtual city are carried out with both CPU single-processing and CPU multi-processing methods. Graphics processing unit (GPU) analysis with CuSPSolver of OpenSees [9] is also appended to see the further details of comparison. CuSPSolver is a based on Cusp library which is linear sparse system solver. CuSPSolver is developed by Xinzheng Lu et al. for OpenSees framework. In addition, Tcl programming language is required to use CuSPSolver. Comparison of the GPU, CPU-SP (CPU single-processing) and CPU-MP (CPU multi-processing) elapsed times for first 10 buildings is illustrated in Figure 6.1.

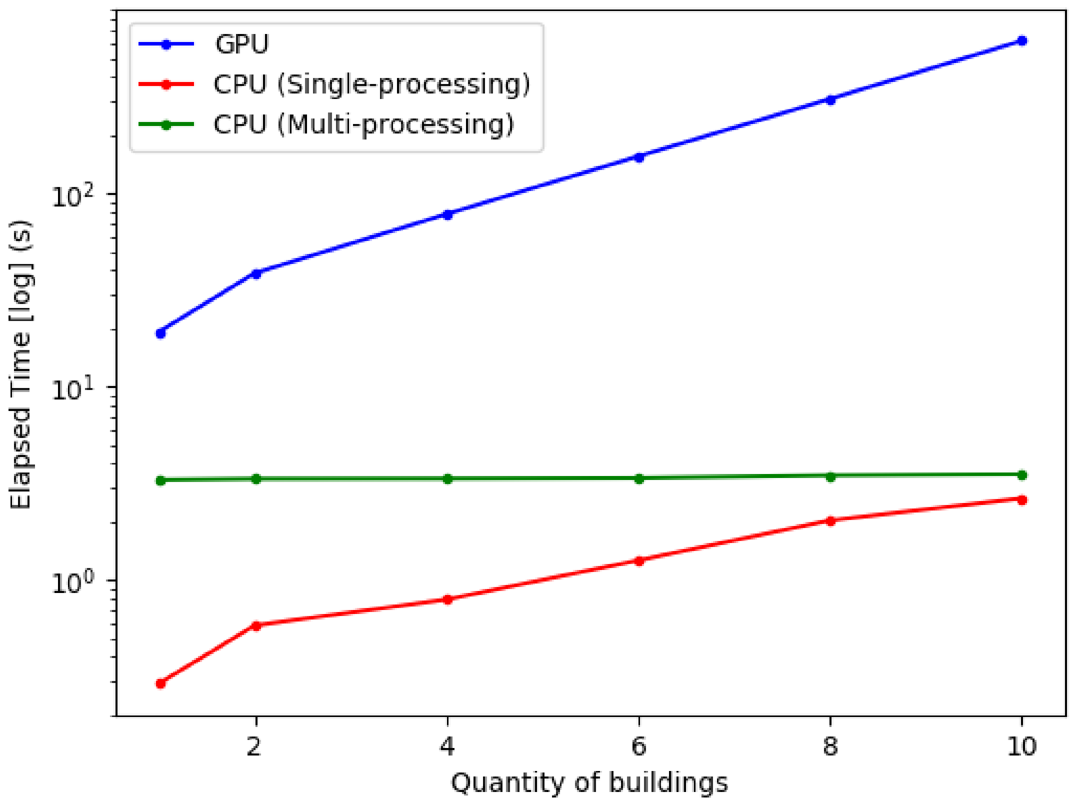

Figure 6.1 : Comparison of the GPU, CPU-SP and CPU-MP elapsed time for first 10 buildings.

At first glance, CPU single core application looks faster. This is because CPU-Multiprocessing framework needs time for spawning processes, assigning tasks, collecting data and closing processes and the time needed for these tasks can easily be seen in Figure 6.1. Once the processes are spawned, they can be used repeatedly without being closed. Therefore, Multiprocessing speed increases quickly. Comparison of the CPU-SP and CPU-MP elapsed times for all buildings is illustrated in Figure 6.2. On the other hand, GPU Solver fell behind both of the CPU applications. The reason of this is that the GPU Solver of OpenSees may not be designed to be applied repeatedly when doing analyses. It can be explained with the communication latency between CPU and GPU when data is being copied.

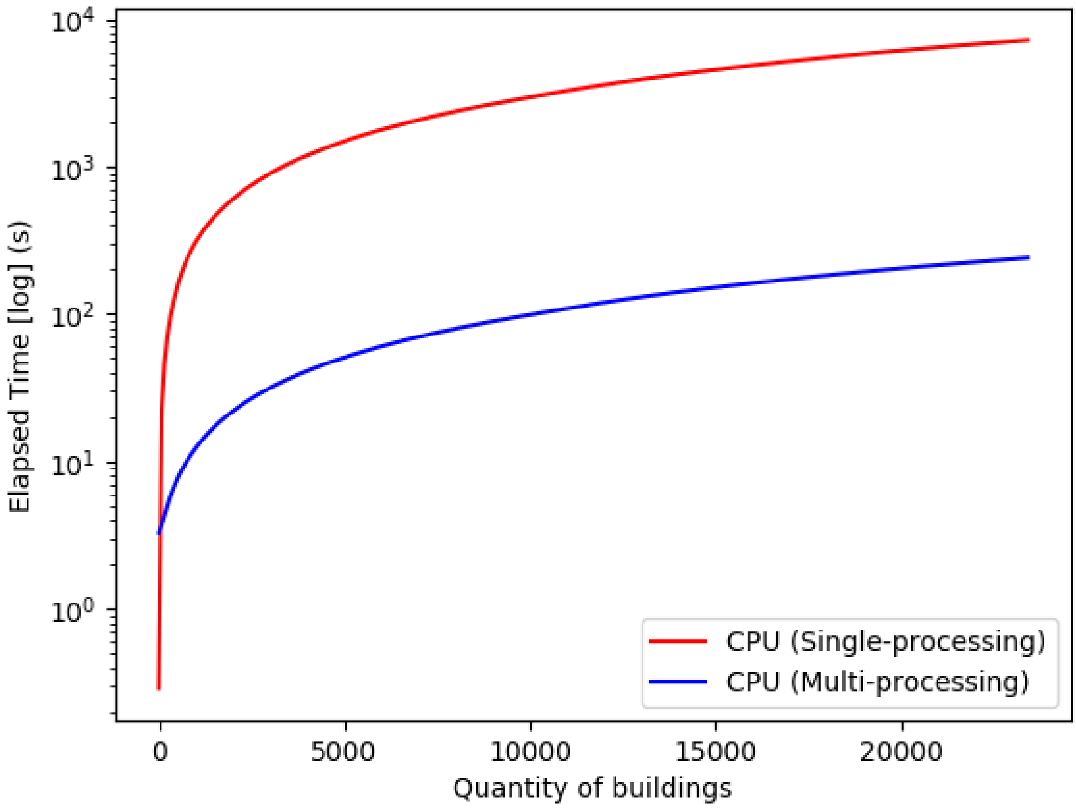

Figure 6.2 : Comparison of the CPU-SP and CPU-MP elapsed time for all buildings.

In the Figure 6.2, CPU-Multiprocessing framework's speed can be better observed with increasing number of buildings. With CPU single-processing application, the nonlinear response history analyses of 1, 10, 100, 1000, 10000 and 23419 buildings are completed in 0.29, 2.64, 28.42, 310.84, 2957.64 and 7223.64 seconds respectively. With CPU multi-processing application, the nonlinear response history analyses of 1, 10, 100, 1000, 10000 and 23419 buildings are completed in 3.29, 3.52, 3.79, 8.42, 98.95 and 240.64 seconds in respectively.

Elapsed time of 1, 10 and 100 buildings analyses have quite same elapsed time since processes needs to be spawn for multi-processing application. Therefore, in order to better understanding of differences, y-axis which is elapsed time axis is given in logarithmic scale. Logarithmic scale reveal the point where the differences appear clearly between CPU single-processing and CPU multi-processing applications. In addition, elapsed time ratios between CPU single-processing and CPU multi-processing is calculated and illustrated in Figure 6.3.

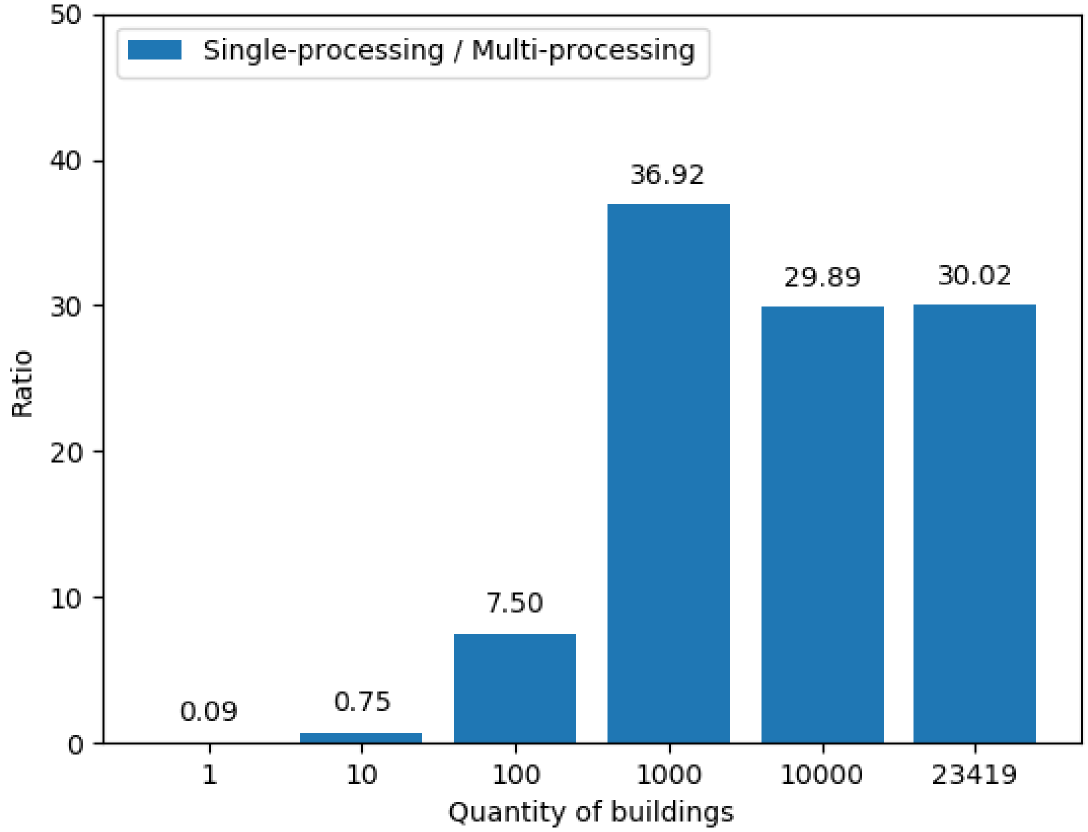

Figure 6.3 : Comparison of the CPU-SP and CPU-MP elapsed time ratio.

In Figure 6.3, it can be seen that the performance ratio decreases after 1000 buildings. There may be two reasons for this: (i) after a certain number of iterations, the temperature of the CPU cores reach to a point where cooling becomes a problem and performance decreases, (ii) after solving a certain number of buildings, CPU cache hit rates decrease. Each CPU is designed for running within a temperature range. While CPU runs with %100 performance, the temperature of CPU increases. After the above point of optimal temperature range, the performance of CPU decreases. To prevent this issue, additional cooling solutions can be applied.

Furthermore, CPU cache is the specialized type of memory in CPU. It reduces the time for accessing data from main memory. However, despite the analysis of IDEAL CITY includes repeatedly similar operations, each building has a different characteristics and corresponding data. Therefore, after solving a certain number of operations, CPU performance decreases depending on CPU cache hit rates.

Elapsed time of CPU-Multiprocessing application depends on different variables. One of the variables is number of cores used. Nonlinear response time history analyses were carried out using 20, 40 and 80 logical cores. Elapsed time differences between analyses using 20, 40 and 80 logical cores are given in Figure 6.4.

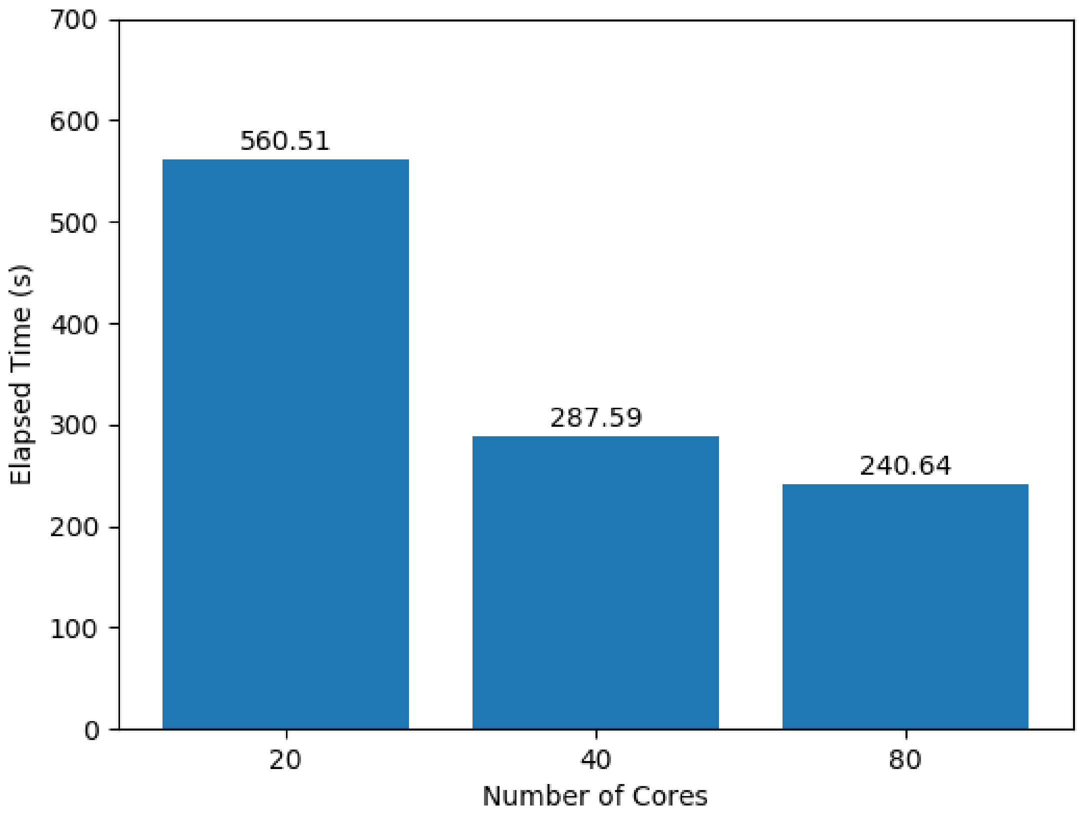

Figure 6.4 : Elapsed time differences between using 20, 40 and 80 logical cores.

In Figure 6.4, elapsed times of analyses are given in seconds. CPU performances were around %25, %50 and %100 using 20, 40 and 80 cores, respectively. As it can be seen, nonlinear response history analysis used 40 cores is approximately two times faster than analysis used 20 cores. Since number of cores used is increased 2 times, this result is acceptable. However, nonlinear response history analysis used logical 80 cores is approximately 1.2 faster than analysis used 40 logical cores. Although number of cores used is increased 2 times, elapsed time differences between using 40 and 80 logical cores were faced with hard drive bottleneck. While the CPU runs with %100 performance using 80 cores, hard drive performance is also %100. When the activity of hard drive reaches the point of the full performance capacity, a queue occurs to write outputs which are processed by CPU. Therefore, analysis used logical 80 cores was not able to reach two times faster than analysis used logical 40 cores.

Depending on the variety of simulation outputs, elapsed time of the analysis may vary. If the desired number of outputs decreases, the computational time also reduces. This is because more variety of simulation outputs require more computational effort. On the other hand, computational time is also related with duration of ground motion since nonlinear response history analysis is applied for IDEAL CITY. In this study, the ground motion recorded by Norcia station is used. The duration of the record is 50 seconds long with 10000 steps. When the duration of the record decreases, the computational time decreases since Newmark's average acceleration method is a time-stepping method.

Furthermore, the acceleration ratio may vary depending on degree of freedom of the buildings model which is single degree of freedom. If the degree of freedom increases, the acceleration ratio decreases. When the degree of freedom increases, the dimensions of matrices extend. Therefore, these matrices cause more computational requirement.

Lastly, if the analyses (e.g. damping, period, etc.) are calculated in advance, the acceleration ratio may increase.

Building responses which are displacement response history output, acceleration response history output and envelope values of both are obtained as a result of nonlinear response history analyses. In order to see the examples of the outputs, seven outputs of randomly selected masonry and concrete buildings are given in between Figure 6.4 and Figure 6.9. Maximum and minimum values of displacement response histories and acceleration response histories can be easily observed since the Python code provides these values directly as an output.

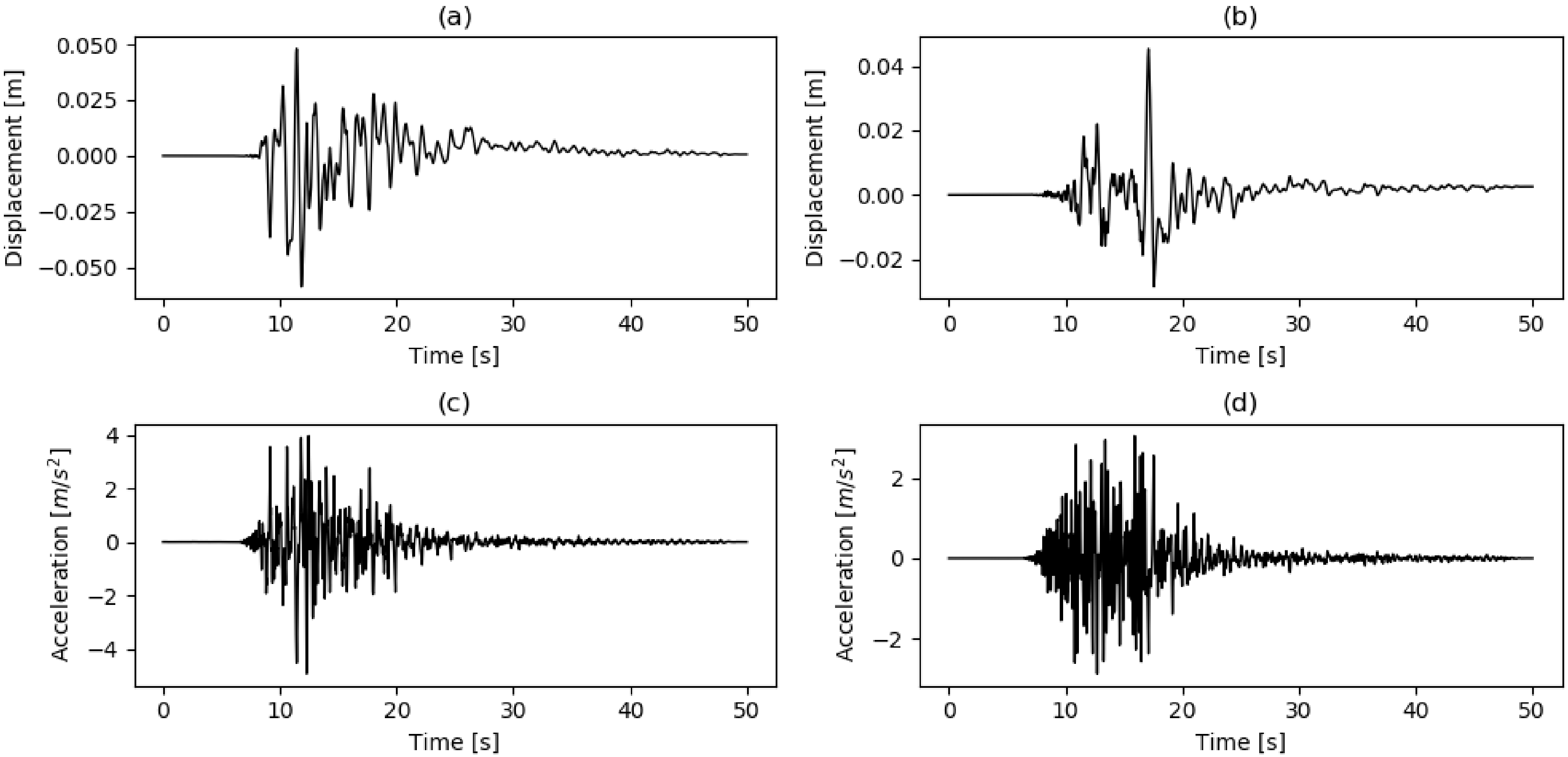

Figure 6.5 : Both displacement and acceleration response history outputs of building #22: x(a)(c) and y(b)(d) components.

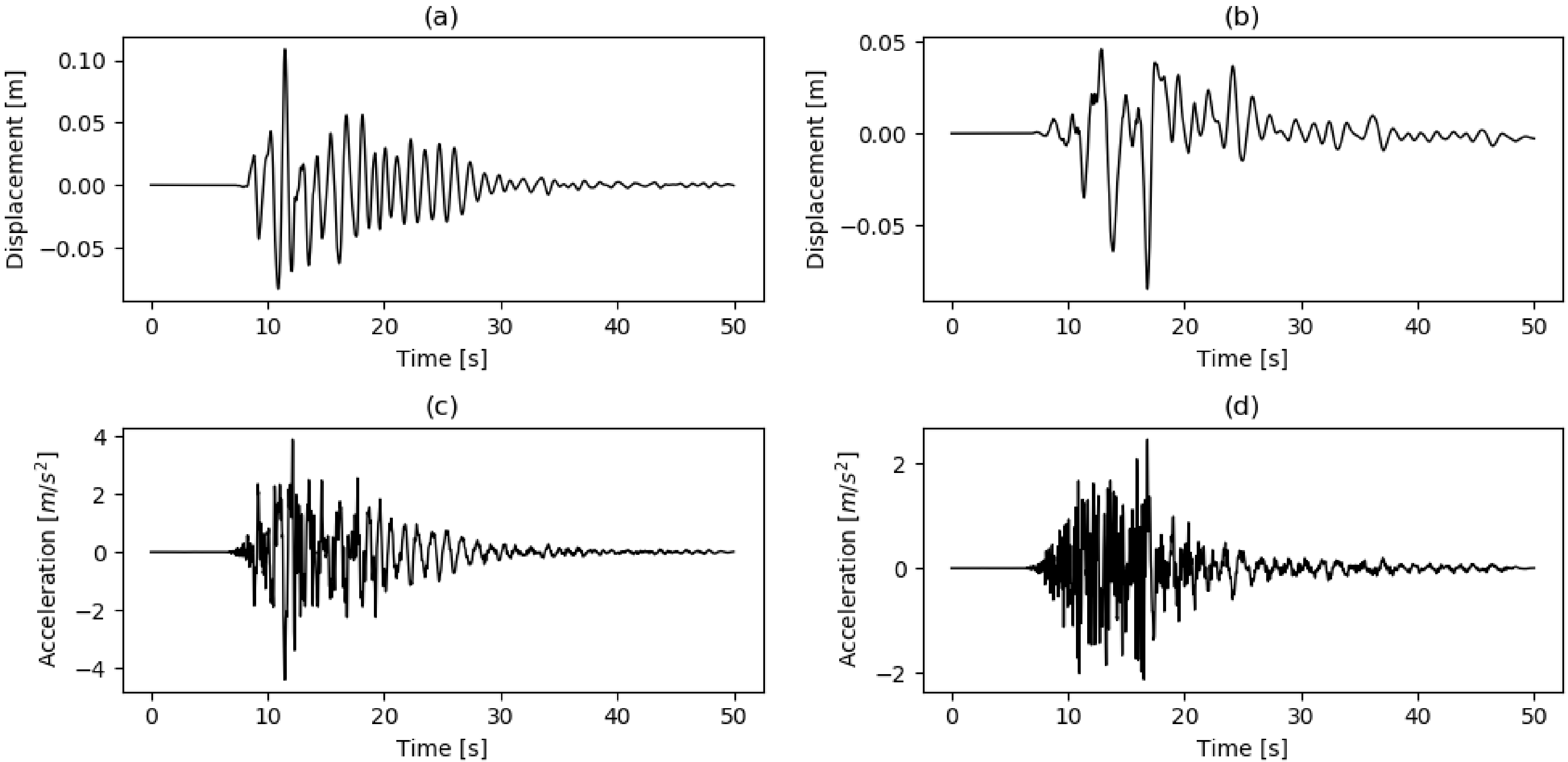

Figure 6.6 : Both displacement and acceleration response history outputs of building #1073: x(a)(c) and y(b)(d) components.

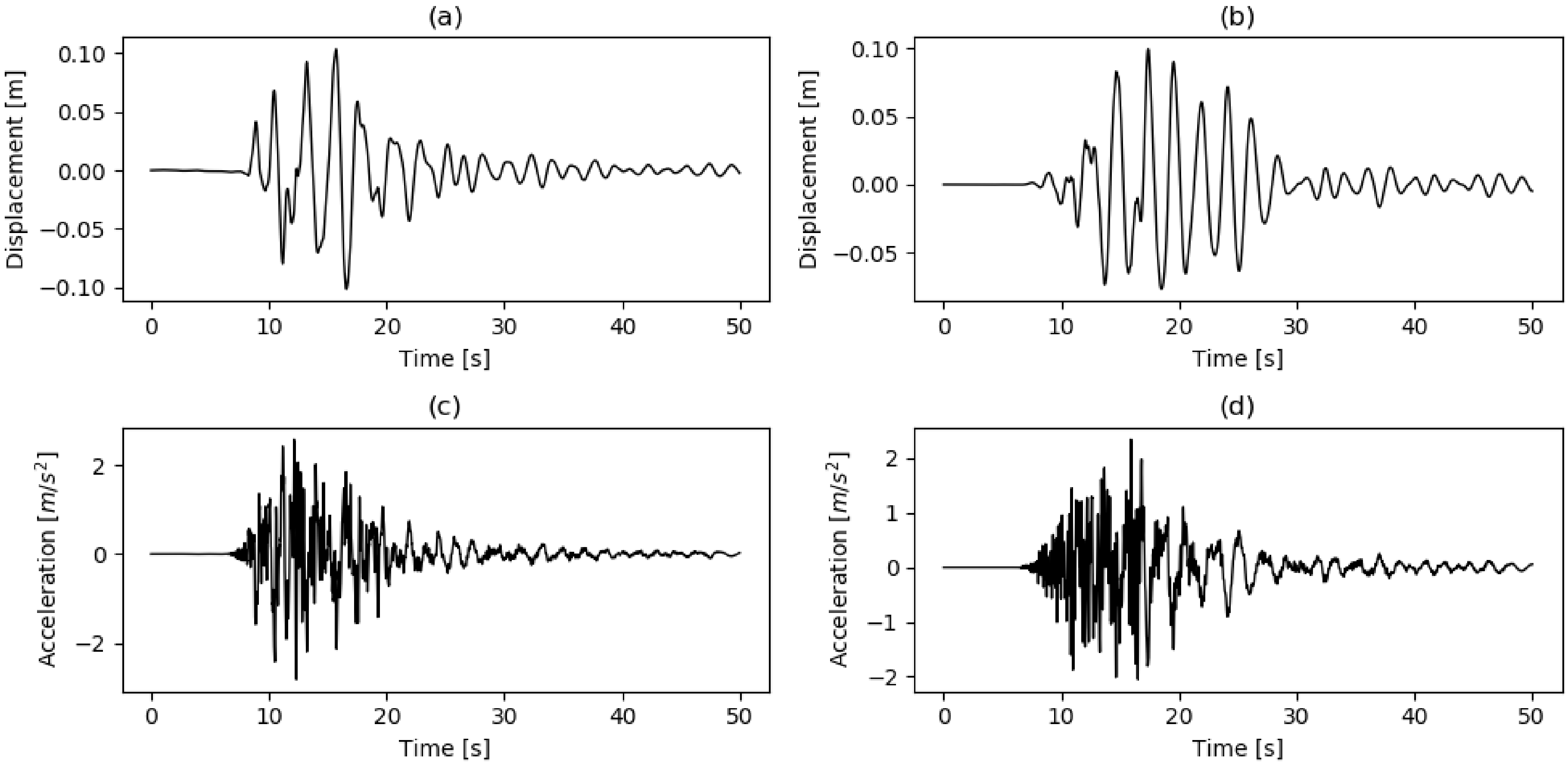

Figure 6.7 : Both displacement and acceleration response history outputs of building #7410: x(a)(c) and y(b)(d) components.

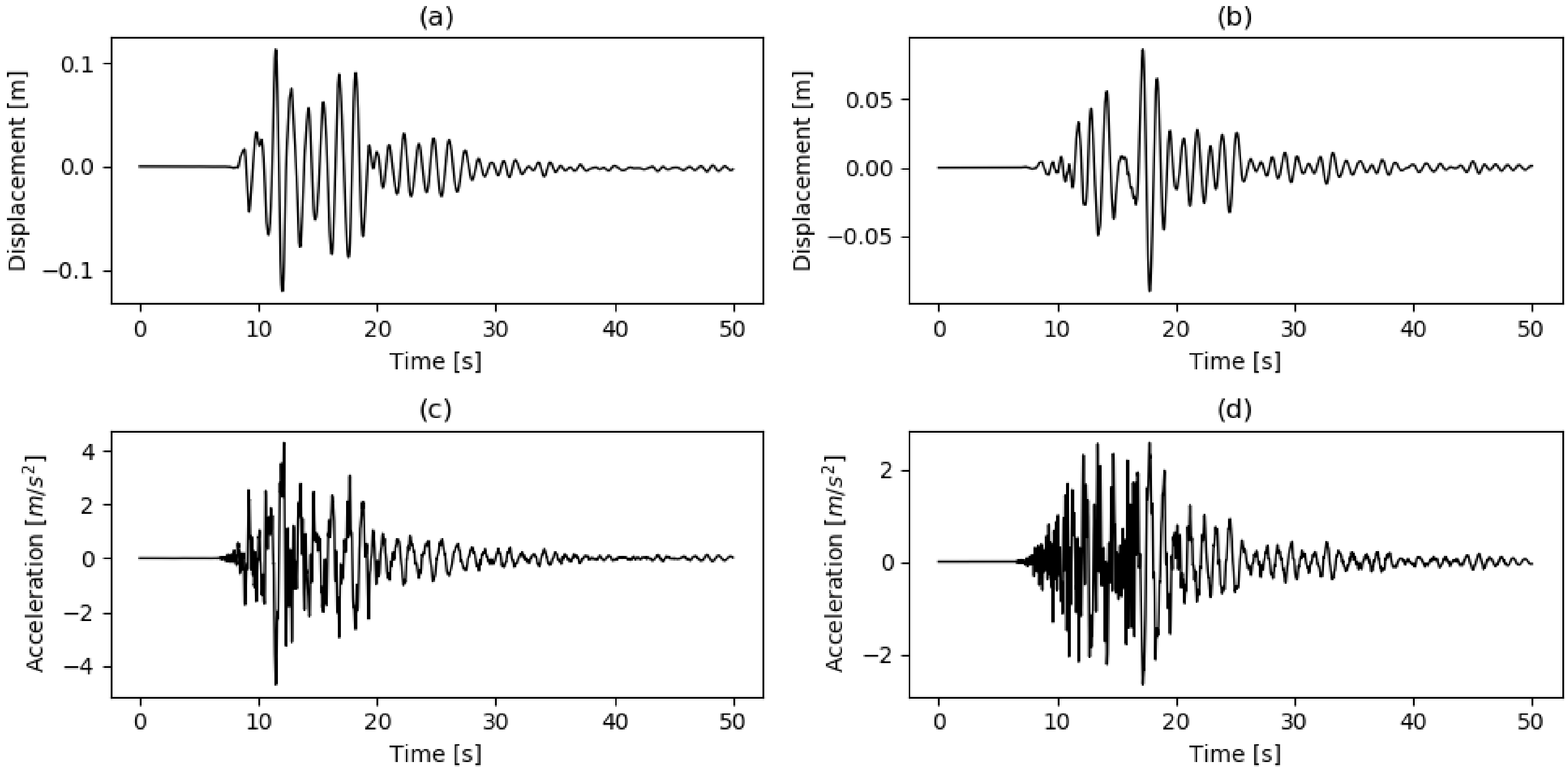

Figure 6.8 : Both displacement and acceleration response history outputs of building #13413: x(a)(c) and y(b)(d) components.

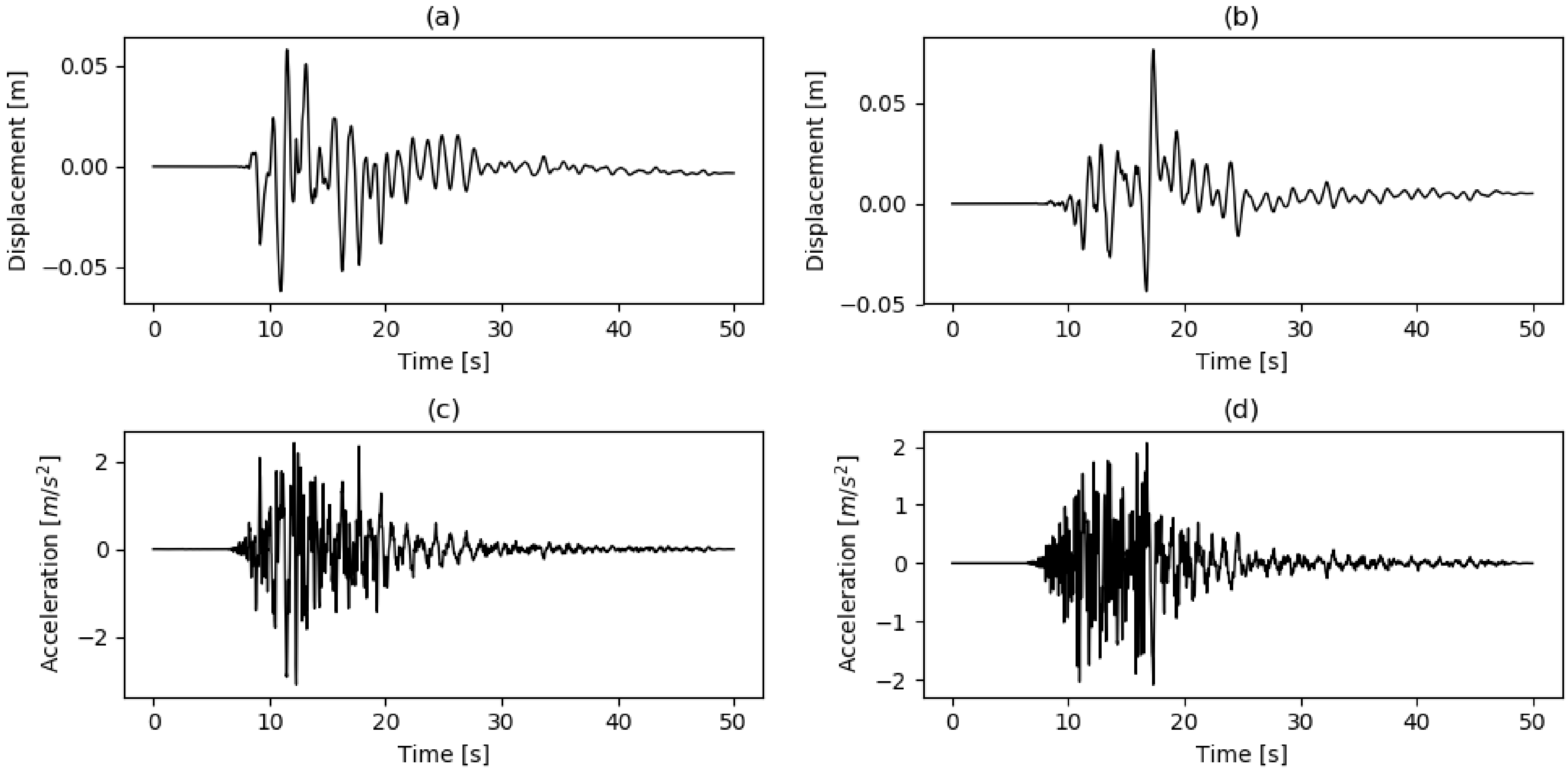

Figure 6.9 : Both displacement and acceleration response history outputs of building #15005: x(a)(c) and y(b)(d) components.

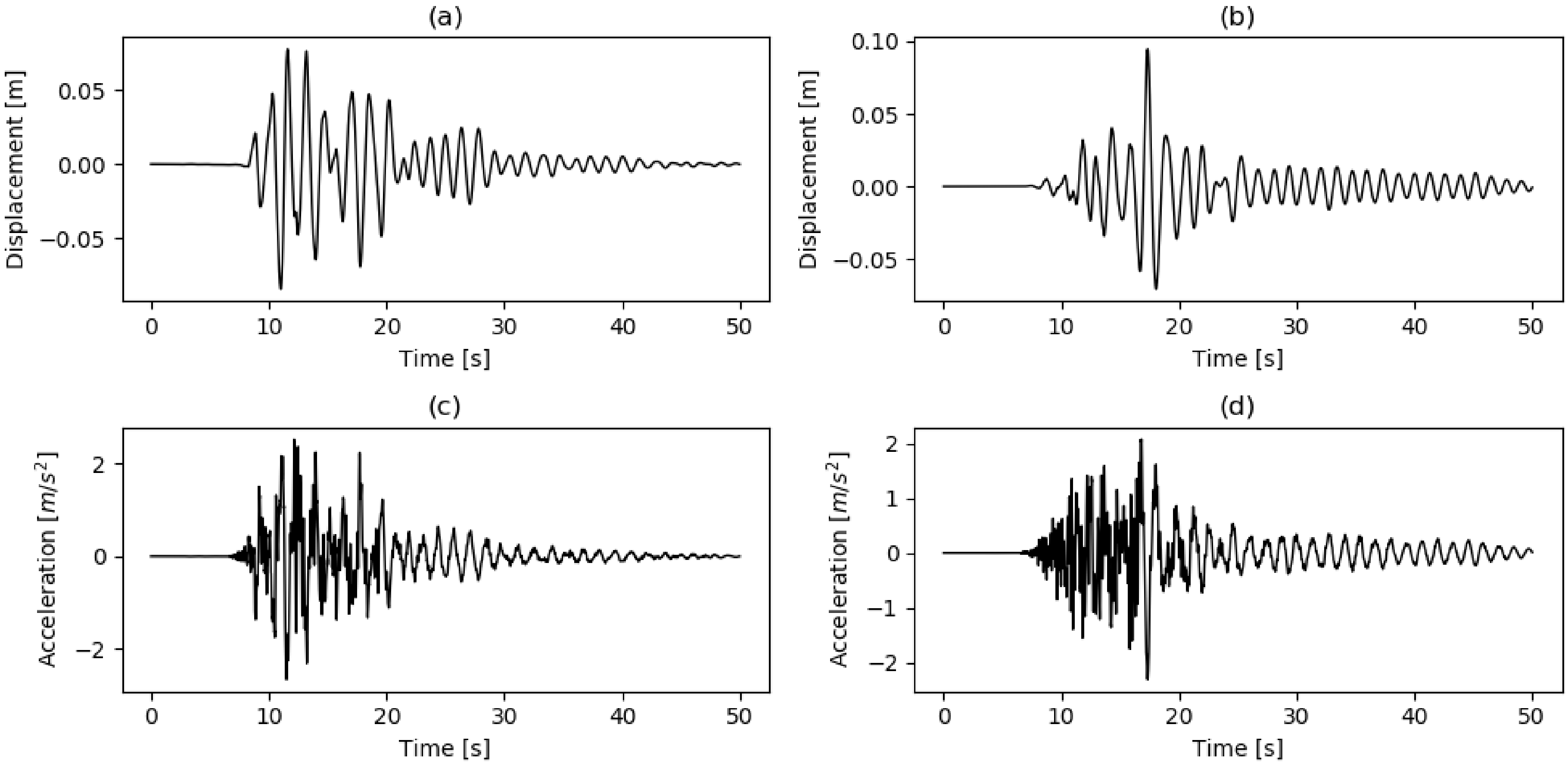

Figure 6.10 : Both displacement and acceleration response history outputs of building #20006 x(a)(c) and y(b)(d) components.

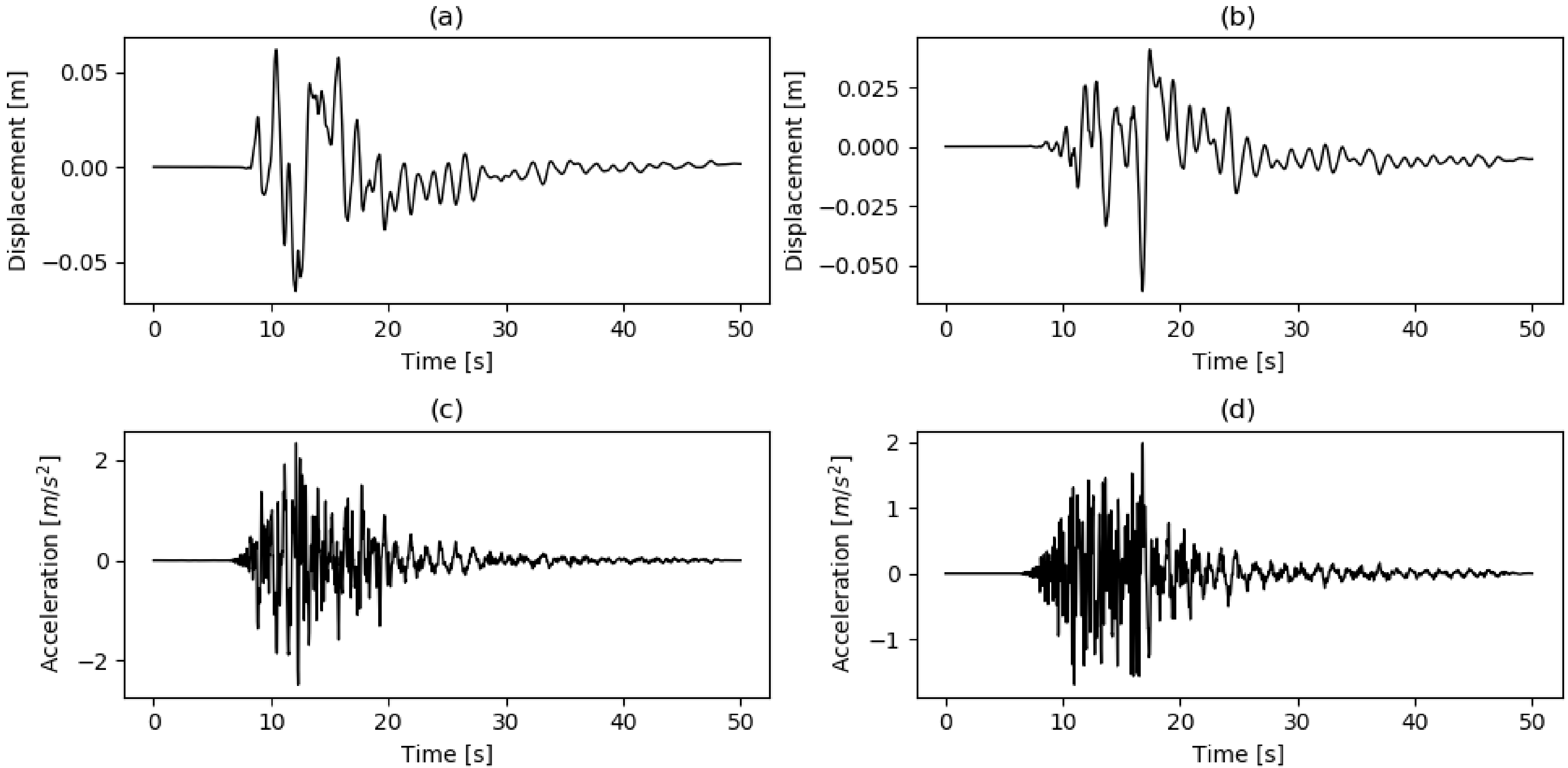

Figure 6.11 : Both displacement and acceleration response history outputs of building #22516: x(a)(c) and y(b)(d) components.

After the outputs of the nonlinear response history analysis of IDEAL CITY were obtained, behaviour type of the buildings were examined. Figure 6.12 shows the backbone curve of global capacity for masonry buildings.

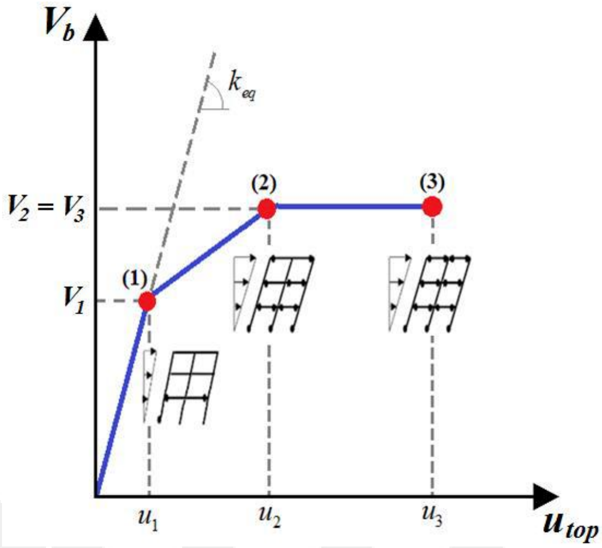

Figure 6.12 : Backbone curve of global capacity for masonry buildings [22].

The backbone curve of the global capacity is expressed between the top displacement (u_{top}) and the base shear (V_b). The first point of the backbone curve represents the yield point where happens the first plastic hinge. The second point refers to the point where the global capacity is reached. The third point shows the collapse point of the masonry buildings. In addition to the masonry buildings, Figure 6.13 shows the backbone curve of global capacity for reinforced concrete buildings.

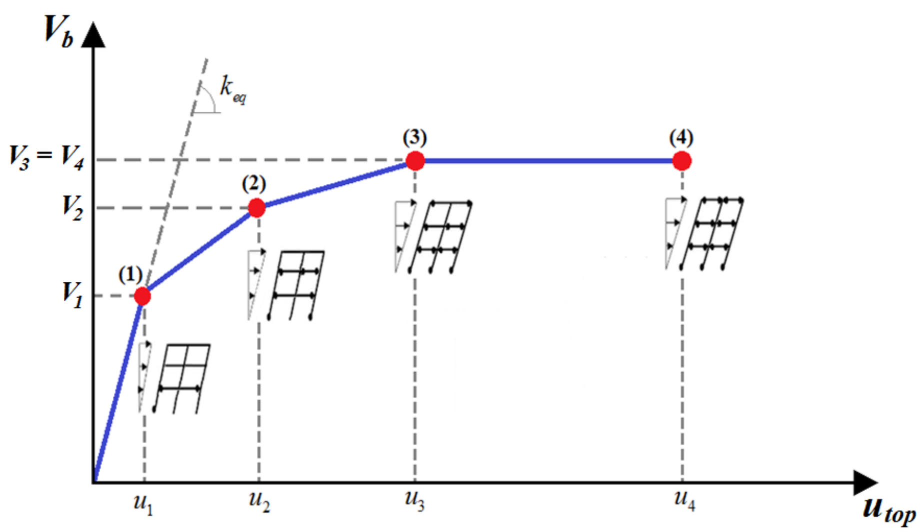

Figure 6.13 : Backbone curve of global capacity for RC buildings [22].

Similary to the masonry buildings, the backbone curve of the global capacity is expressed between the top displacement and the shear base. The first point of the backbone curve shows the yield point where happens the first plastic hinge. The global stiffness decreases between the first point and the second point. The third point refers to the point where the global capacity is reached. The fourth point represents the collapse point of the reinforced concrete buildings.

In order to identify the behaviour type of the buildings, the top displacements of the buildings obtained from the outputs are compared with the first points of the backbone curves for both masonry and reinforced concrete buildings. The meaning of exceeding the first point refers to nonlinear behaviour. The distribution of behaviour types of the buildings are given in Table 6.1.

Table 6.1 : Distribution of behaviour types of the buildings.

Linear [%]	Nonlinear [%]
9	91

The nonlinear behaviour was observed on %91 of the buildings. In addition, %9 of the buildings demonstrate linear behaviour. Especially, masonry buildings show more nonlinear behaviour than reinforced concrete buildings. Furthermore, depending on construction year, buildings shows different ductility performance.

7. CONCLUSIONS AND FUTURE WORK

At the beginning of Integrated Design and Control of Sustainable Communities during Emergencies (IDEAL RESCUE) project, the IDEAL CITY was modeled and analyzed with SAP2000 software. In this thesis, Python scripts are developed for modeling a virtual city with OpenSees framework. Accelerated nonlinear response history analysis is applied with *multiprocessing* package of Python, under the project of IDEAL RESCUE. According to the results, multi-processing application yields faster simulations up to approximately 37 times than single-processing application. This result is obtained by using dual Intel Xeon E5 CPUs (3.6 GHz) that include 80 logical cores. Depending on degree of freedom of the model, number of cores, memory size, operation system, interpreter of OpenSees and version of OpenSeesPy, the acceleration ratio may vary. Although GPUs may have thousands of cores, it is not convenient to use these cores parallely with OpenSees. This is because OpenSees is written in C++ programming language and designed to work compatible with CPU. In order to exploit the full capacity of GPUs, it is required to develop a framework written in a GPU programming environment such as CUDA. On the other hand, Python programming language is quite flexible and user-friendly, which makes it possible to create an accelerated framework rapidly. Independent of computational platform, Python scripts can be run on many computers and operating systems such as Windows, Linux, macOS thanks to the Python's cross-platform capabilities. Lastly, this multi-processing application is not restricted to modeling and analyses. It can be used to solve different kind of civil engineering problems where computation time is a constraint.

As for future studies, despite the CPU-Multiprocessing application have provided acceptable and sufficient solution for the cities which have around 23000 buildings, this method requires a more computational cost and effort for cities such as Istanbul which includes more than one million buildings. For the nonlinear response history analysis of large scale cities such as Istanbul, a advanced solution could be developed.

Graphic processing unit can be used instead of CPU to carry out the nonlinear response history analysis. Although the clock speed of each core of GPU weaker than CPU, GPU has many more cores. Therefore, GPU can be convenient solution due to its huge number of cores. In order to use full computational power of GPUs, a code could be written in programming language such as Python, C++ using CUDA environment. On the other hand, the computational cost of nonlinear response history analysis of large cities can be reduced using GPU.

REFERENCES

[1] **Cimellaro, G. P.; Marasco, S.** (2018). *Introduction to Dynamics of Structures and Earthquake Engineering*, Springer International Publishing

[2] **Lu, X.; Guan, H.** (2017). *Earthquake Disaster Simulation of Civil Infrastructures*, Springer Singapore

[3] **Krizhevsky, A.; Sutskever, I.; Hinton, G. E.** (2012). ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*

[4] **Li, L.-J.; Li, K.; Li, F. F.; Deng, J.; Dong, W.; Socher, R.; Fei-Fei, L.** (2009). ImageNet: A Large-Scale Hierarchical Image Database, *2009 IEEE Conference on Computer Vision and Pattern Recognition*

[5] **Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A.** (2015). Going deeper with convolutions, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

[6] **He, K.; Zhang, X.; Ren, S.; Sun, J.** (2016). Deep residual learning for image recognition, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. doi:10.1109/CVPR.2016.90

[7] **Hori, M.; Ichimura, T.** (2008). Current state of integrated earthquake simulation for earthquake hazard and disaster, *Journal of Seismology*, Vol. 12, No. 2, 307–321

[8] **Sahin, A.; Sisman, R.; Askan, A.; Hori, M.** (2016). Development of integrated earthquake simulation system for Istanbul the Next Marmara Earthquake: Disaster Mitigation, Recovery and Early Warning 4. Seismology, *Earth, Planets and Space*, Vol. 68, No. 1

[9] **Tian, Y.; Xie, L.; Xu, Z.; Lu, X.** (2015). GPU-powered high-performance computing for the analysis of large-scale structures based on opensees, *Congress on Computing in Civil Engineering, Proceedings*, Vols 2015-Janua, No. January, 411–418

[10] **McKenna, F.** (2011). OpenSees: A framework for earthquake engineering simulation, *Computing in Science and Engineering*, Vol. 13, No. 4, 58–66

[11] **McKenna, F.** (1997). *Object-oriented finite element programming frameworks for analysis algorithms and parallel computing*, PhD dissertation, University of California, Berkeley, California, USA,

[12] **Bell, N.; Garland, M.** (2012). Cusp: Generic Parallel Algorithms for Sparse Matrix and Graph Computations

[13] **Lu, X.; Han, B.; Hori, M.; Xiong, C.; Xu, Z.** (2014). A coarse-grained parallel approach for seismic damage simulations of urban areas based on refined models and GPU/CPU cooperative computing, *Advances in Engineering Software*, Vol. 70, 90–103

[14] **Mosalam, K. M.; Liang, X.; Günay, S.; Schellenberg, A.** (2013). Alternative integrators and parallel computing for efficient nonlinear response history analyses, *ECCOMAS Thematic Conference - COMPDYN 2013: 4th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, Proceedings - An IACM Special Interest Conference*, No. July 2017, 3116–3136

[15] **Anderson, T. L.; Pauschke, J. M.; Goldstein, S. N.; Nelson, P. P.** (2004). The George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES), *Structures - A Structural Engineering Odyssey, Structures 2001 - Proceedings of the 2001 Structures Congress and Exposition*

[16] **Lu, X.; Xu, Z.; Xiong, C.; Zeng, X.** (2017). High Performance Computing for Regional Building Seismic Damage Simulation, *Procedia Engineering*, Vol. 198, No. September 2016, 836–844

[17] **Xiong, C.; Lu, X.; Lin, X.; Xu, Z.; Ye, L.** (2017). Parameter Determination and Damage Assessment for THA-Based Regional Seismic Damage Prediction of Multi-Story Buildings, *Journal of Earthquake Engineering*

[18] **Xiong, C.; Lu, X.; Guan, H.; Xu, Z.** (2016). A nonlinear computational model for regional seismic simulation of tall buildings, *Bulletin of Earthquake Engineering*

[19] **Xu, Z.; Lu, X.; Law, K. H.** (2016). A computational framework for regional seismic simulation of buildings with multiple fidelity models, *Advances in Engineering Software*, Vol. 99, 100–110

[20] **Compute Unified Device Architecture (CUDA).** (2019), from https://developer.nvidia.com/cuda-zone, Date Accessed: 2019

[21] **Zhu, M.; McKenna, F.; Scott, M. H.** (2018). OpenSeesPy: Python library for the OpenSees finite element framework, *SoftwareX*, Vol. 7, 6–11

[22] **Marasco, S.** (2018). *Large Scale Simulation of IDEAL CITY under Seismic Scenario*, PhD Dissertation, Politecnico di Torino, Italy

[23] **CSI SAP2000.** (2016). Analysis Reference Manual, *CSI: Berkeley (CA, USA): Computers and Structures INC*

[24] **Uysal, M.; Deger, Z.; Cimellaro, G. P.** (2019). CPU-accelerated earthquake simulations for large scale urban cities, *Engineering Mechanics Institute Conference 2019 (EMI2019)*, ASCE, Caltech, Los Angeles, California

[25] **Python Programming Language.** (2019), from https://docs.python.org/3.6/, Date Accessed: 2019

[26] **Multiprocessing Package of Python.** (2019), from https://docs.python.org/3.6/library/multiprocessing.html, Date Accessed: 2019

[27] **The Pacific Earthquake Engineering Research Center (PEER).** (2019), from https://ngawest2.berkeley.edu/, Date Accessed: 2019

[28] **C++ Programming Language.** (2019), from https://isocpp.org/std/the-standard, Date Accessed: 2019

[29] **Tcl Programming Language.** (2019), from http://www.tcl.tk/, Date Accessed: 2019

[30] **Van Der Walt, S.; Colbert, S. C.; Varoquaux, G.** (2011). The NumPy array: A structure for efficient numerical computation, *Computing in Science and Engineering*

[31] **McKinney, W.** (2010). Data Structures for Statistical Computing in Python, *Proceedings of the 9th Python in Science Conference*

[32] **Hunter, J. D.** (2007). Matplotlib: A 2D graphics environment, *Computing in Science and Engineering*

[33] **OpenSees Command Manual.** (2019), from https://opensees.berkeley.edu/wiki/index.php/Command_Manual, Date Accessed: 2019

[34] **Takeda; Sozen; Nielsen.** (1970). Reinforced Concrete Response to Simulated Earthquakes, *Journal of the Structural Division*

[35] **Charney, F. A.** (2008). Unintended consequences of modeling damping in structures, *Journal of Structural Engineering*

[36] **Boore, D. M.; Atkinson, G. M.** (2008). Ground-motion prediction equations for the average horizontal component of PGA, PGV, and 5%-damped PSA at spectral periods between 0.01 s and 10.0 s, *Earthquake Spectra*. doi:10.1193/1.2830434

[37] **Newmark, N. M.** (1959). A Method of Computation for Structural Dynamics, *ASCE Journal of Engineering Mechanics Division*, Vol. 85

CURRICULUM VITAE

Name Surname: Mert Uysal

Place and Date of Birth: Antalya, 25 February 1992

E-Mail: mert.uysal@msn.com

EDUCATION:

B.Sc.: 2016, Akdeniz University, Faculty of Engineering, Civil Engineering

M.Sc.: 2019, Istanbul Technical University, Earthquake Engineering and Disaster Management Institute, Earthquake Engineering

PUBLICATIONS/PRESENTATIONS ON THE THESIS:

- Uysal, M., Deger, Z. T., and Cimellaro, G. P. (2019). "CPU-accelerated earthquake simulations for large scale urban cities." *Engineering Mechanics Institute Conference 2019 (EMI2019)*, ASCE, Caltech, Los Angeles, California, 18-21 June, 2019.